PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN JURUSAN DI PERGURUAN TINGGI

TUGAS AKHIR SKRIPSI

Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan Guna
Memperoleh Gelar Sarjana Pendidikan

Oleh :
Yustie Aprissya Dhewi
NIM. 09520241017

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2014

LEMBAR PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN JURUSAN DI PERGURUAN TINGGI

Disusun oleh:

Yustie Aprissya Dhewi
NIM. 09520241017

Telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan Ujian Akhir Tugas Akhir Skripsi bagi yang bersangkutan.

Yogyakarta, 23 Desember 2013

Mengetahui, Ketua Program Studi Pendidikan Teknik Informatika	Disetujui, Pembimbing Skripsi
Dr. Ratna Wardani NIP. 19701218 200501 2 001	Handaru Jati, Ph.D NIP. 197405111999031002

HALAMAN PENGESAHAN

Tugas Akhir Skripsi

PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN JURUSAN DI PERGURUAN TINGGI

Disusun oleh:

Yustie Aprissya Dhewi
NIM. 09520241017

Telah dipertahankan didepan TimPenguji Tugas Akhir Skripsi Program Studi Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta

Pada tanggal 16 Desember 2013

TIM PENGUJI

Nama/Jabatan	Tanda Tangan	Tanggal
Handaru Jati, Ph.D	.....	21/1 14
Ketua Penguji/Pembimbing		
Totok Sukardiyono, M.T.	.....	21/1 14
Sekretaris		
Dessy Irmawati, M.T.	.....	21/1 14
Penguji		

Yogyakarta,21Januari 2014
Fakultas TeknikUniversitas Negeri Yogyakarta
Dekan,

Dr. Mochamad Bruri Triyono, M.Pd.
NIP. 19560216 198603 1 003

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini:

Nama : Yustie Aprissya Dhewi

NIM : 09520241017

Program Studi : Pendidikan Teknik Informatika

Fakultas : Fakultas Teknik UNY

Judul TAS : Pengembangan dan Analisis Kualitas Aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi

Menyatakan bahwa yang tertulis di dalam skripsi ini benar-benar hasil karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan oleh orang lain, kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim dan telah disebutkan dalam daftar pustaka.

Yogyakarta, 23 Desember 2013
Yang Menyatakan

Yustie Aprissya Dhewi
NIM. 09520241017

MOTTO DAN PERSEMBAHAN

A. MOTTO

"***Your time is limited, so don't waste it living someone else's life***" Steve Jobs

"***Sometimes life hits you in the head with a brick. Don't lose faith. I'm convinced that the only thing that kepft me going was that I loved what I did.***" Steve Jobs

"***Being the richest man in the cemetery doesn't matter to me. Going to bed at night saying we've done something wonderful... that's what matters to me.***" Steve Jobs

"***A person who never made mistake never tried anything new***" Albert Einstein

B. PERSEMBAHAN

Karya ini saya persembahkan untukBapak dan Ibu tercinta yang selalu memberikan dukungan dan doa yang tiada henti demi terselesaikannya karya ini, semoga selalu diberkahi Allah SWT. Adik-adik dan keluarga yang selalu memberikan support, semoga selalu diberi kelancaran dalam menempuh setiap pilihan hidupnya. Rekan-rekan akademik kelas E Pendidikan Teknik Informatika 2009 serta seluruh pihak yang telah membantu terselesaikannya penelitian ini.

PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PENDUKUNG KEPUTUSAN PEMILIHANJURUSAN DI PERGURUAN TINGGI

Oleh :
Yustie Aprissya Dhewi
NIM 09520241017

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi menggunakan bahasa pemograman Visual Basic dan *database* MySQL. Penelitian ini meliputi perancangan sistem serta pengujian unjuk kerja sistem perangkat lunak pada faktor kualitas *correctness*, *functionality*, *usability*,dan*maintainability*. Perangkat lunak dari penelitian ini diharapkan dapat membantu mengarahkan siswa SMA dalam memilih jurusan kuliah yang tepat berdasarkan minat dan jenis kecerdasan yang dimiliki oleh masing-masing siswa.

Jenis penelitian ini adalah *Research and Development.* Metode pengembangan perangkat lunak menggunakan *Waterfall Model.* Tahapan pertama dilakukan analisa kebutuhan serta melakukan kajian literatur pendukung penelitian. Tahapan kedua dilakukan dengan membuat desain perangkat lunak. Tahapan ketiga yaitu implementasi atau pengkodean. Tahapan terakhiradalah pengujian. Analisis faktor kualitas*correctness* dilakukan dengan perhitungan jumlah *error* / KLOC. Analisis faktor kualitas *functionality* dilakukan dengan pengujian setiap fungsi aplikasi. Analisis faktor kualitas *usability* dilakukan dengan metode kuesioner yang dibagikan kepada mahasiswa dan siswa kelas XII SMA Negeri 2 Wates.Analisis faktor kualitas *maintainability* dilakukan dengan melakukan pengujian pada tiga aspek *maintainability* yaitu *instrumentation*, *consistency* dan *simplicity*.

Berdasarkan hasil pengujian perangkat lunak yang dikembangkan menunjukkan hasil: 1) perancangan aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi telah melalui serangkaian proses rekayasa perangkat lunak dari analisis kebutuhan hingga pengujian; 2) hasil analisis kualitas dari aplikasi Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memenuhi standar faktor kualitas *correctness*, *functionality*, *usability*,dan *maintainability*.

Kata Kunci:sistem pendukung keputusan, faktor kualitas, *correctness*,*functionality*, *usability*, *maintainability*

KATA PENGANTAR

Puji Syukur keharidat Allah SWTatas berkat rahmat dan karunia-Nya, Tugas Akhir Skripsi dalam rangka untuk memenuhi sebagian persyaratan untuk mendapatkan gelar Sarjana Pendidikan dengan judul “Pengembangan dan Analisis Kualitas Aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi” dapat disusun sesuai dengan harapan. Tugas Akhir Skripsi ini dapat diselesaikan tidak lepas dari bantuan dan kerjasama dengan pihak lain. Berkenaan dengan hal tersebut, penulis menyampaikan ucapan terima kasih kepada yang terhormat:

1. Handaru Jati, Ph.D selaku Dosen Pembimbing TAS yang telah banyak memberikan semangat, dorongan, dan bimbingan selama penyusunan Tugas Akhir Skripsi ini.
2. Handaru Jati, Ph.D, Totok Sukardiyono, M.T., dan Dessy Irmawati, M.T. selaku Ketua Penguji, Sekretaris, dan Penguji yang memberikan koreksi perbaikan secara komprehensif terhadap TAS ini.
3. Muhammad Munir, M.Pd dan Dr. Ratna Wardani selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Ketua Program Studi Pendidikan Teknik Informatika beserta dosen dan staf yang telah memberikan bantuan dan fasilitas selama proses penyusunan pra proposal sampai dengan selesainya TAS ini.
4. Dr. Mochamad Bruri Triyono, M.Pdselaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang memberikan persetujuan pelaksanaan Tugas Akhir Skripsi.

5. Drs. H. Mudjijono,M.M selaku Kepala SMA Negeri 2 Wates yang telah memberi ijin dan bantuan dalam pelaksanaan penelitian Tugas Akhir Skripsi ini.
6. Para guru dan staf SMA Negeri 2 Wates yang telah memberi bantuan memperlancar pengambilan data selama proses penelitian Tugas Akhir Skripsi ini.
7. Semua pihak, secara langsung maupun tidak langsung, yang tidak dapat disebutkan di sini atas bantuan dan perhatiannya selama penyusunan Tugas Akhir Skripsi ini.

Akhirnya, semoga segala bantuan yang telah berikan semua pihak di atas menjadi amalan yang bermanfaat dan mendapatkan balasan dari Allah SWT dan Tugas Akhir Skripsi ini menjadi informasi bermanfaat bagi pembaca atau pihak lain yang membutuhkannya.

Yogyakarta,Desember2013

Penulis

Yustie Aprissya Dhewi
NIM. 09520241017

DAFTAR ISI

Halaman

HALAMAN JUDUL i
LEMBAR PERSETUJUAN ii
HALAMAN PENGESAHAN iii
SURAT PERNYATAAN iv
MOTTO DAN PERSEMBAHAN v
ABSTRAK vi
KATA PENGANTAR vii
DAFTAR ISI ix
DAFTAR GAMBAR xi
DAFTAR TABEL xv
DAFTAR LAMPIRAN xix

BAB I PENDAHULUAN 1
A. Latar Belakang 1
B. Identifikasi Masalah 4
C. Batasan Masalah 5
D. Rumusan Masalah 5
E. Tujuan Penelitian 6
F. Manfaat Penelitian 6

BAB II KAJIAN TEORI 7
A. Landasan Teori 7
1. Kecerdasan Buatan (*Artificial Intelligence*) 7
2. Sistem Pakar *(Expert System)* 9
3. Teori Holland (RIASEC) 28
4. Microsoft Visual Basic 39
5. MySQL 44
6. Basis Data 45
7. *Waterfall Process Model* 50
8. *Software Testing* 51
9. Kualitas Perangkat Lunak (*Software Quality*) 54
10. Faktor Kualitas *Correctness* 58
11. Faktor Kualitas *Functionality* 60
12. Faktor Kualitas *Usability* 61
13. Faktor Kualitas *Maintainability* 63
B. Penelitian yang Relevan 65
C. Kerangka Berfikir 66
D. Pertanyaan Penelitian 68

BAB III METODE PENELITIAN **69**
A. Desain Penelitian 69
1. Analisis Kebutuhan (*Software Requirement Analysis*)..... 71
2. Desain (*Design*) 71
3. Pengkodean (*Coding*) 72
4. Pengujian (*Testing*)..... 72
B. Variable Penelitian 75
C. Teknik Pengumpulan Data 76
1. Studi Pustaka 76
2. Observasi 77
3. Kuesioner..... 77
D. Skala Pengukuran 78
1. Skala Guttman 78
2. Skala Likert 79
E. Objek Penelitian 80
F. Instrumen Penelitian..... 80
1. Observasi 80
2. Angket 83
G. Teknik Analisis Data 98
1. Analisis Faktor Kualitas *Correctness*..... 98
2. Analisis Faktor Kualitas *Functionality*..... 98
3. Analisis Faktor Kualitas *Usability*..... 99
4. Analisis Faktor Kualitas *Maintainability* 104

BAB IV HASIL PENELITIAN DAN PEMBAHASAN **105**
A. Pengembangan Perangkat Lunak 105
1. Analisa Kebutuhan (*Requirement Analysis*)..... 105
2. Desain (*Design*) 109
3. Pengkodean (*Coding*) 148
4. Pengujian (*Testing*)..... 170
B. Analisis Kualitas Perangkat Lunak 185
1. Analisis Faktor Kualitas *Correctness*..... 185
2. Analisis Faktor Kualitas *Functionality*..... 194
3. Analisis Faktor Kualitas *Usability*..... 197
4. Analisis Faktor Kualitas *Maintainability* 200
C. Implementasi Pengujian 202

BAB V SIMPULAN DAN SARAN..... **207**
A. Kesimpulan..... 207
B. Keterbatasan Produk..... 208
C. Saran 208

DAFTAR PUSTAKA..... **209**
LAMPIRAN..... **211**

DAFTAR GAMBAR

Halaman

Gambar 1. Ruang Lingkup Kecerdasan Buatan.....8

Gambar 2. Struktur Sistem Pakar (Turban, 2005) 12

Gambar 3. Contoh Jaringan Semantik 20

Gambar 4. Bingkai Komputer.....21

Gambar 5. Pohon Keputusan Identifikasi Pesawat 24

Gambar 6. Proses Reduksi Pohon Keputusan.....25

Gambar 7. Pohon Keputusan Hasil Reduksi.....26

Gambar 8. Diagram Kategorisasi Faktor Kualitas Perangkat Lunak.....55

Gambar 9. Kerangka Berfikir.....68

Gambar 10. *Waterfall's Model*.....71

Gambar 11. DFD Level 0.....111

Gambar 12. DFD Level 1.....112

Gambar 13. DFD Level 1 Proses 3.0 113

Gambar 14. *Entity Relationship Diagram* 114

Gambar 15. Potongan *Rule Base* 119

Gambar 16. *Storyboard* Halaman *Login*.....120

Gambar 17. *Storyboard* Halaman Pendaftaran 121

Gambar 18. *Storyboard* Halaman *About* (Tentang).....121

Gambar 19. *Storyboard* Halaman Diagnosa Untuk *Normal User* 122

Gambar 20. *Storyboard* Halaman Diagnosa Untuk *Admin*.....122

Gambar 21. *Storyboard* Fitur Penjelas (Kenapa?).....123

Gambar 22. *Storyboard* Fitur Detail Info.....123

Gambar 23. *Storyboard* Halaman Tambah Awal.....124

Gambar 24. *Storyboard* Halaman Ubah Fakta.....124

Gambar 25. *Storyboard* Halaman Tambah Fakta 125

Gambar 26. *Storyboard* Halaman Sisipkan Fakta.....125

Gambar 27. Desain Halaman Bantuan (*Help*) 126

Gambar 28. Desain Notifikasi Sistem.....126

Gambar 29. *Storyboard* Notifikasi Tambah Fakta.....127

Gambar 30. *Storyboard* Notifikasi Keluar Aplikasi 127

Gambar 31. Desain Arsitektur 130

Gambar 32. Modul *Log In* 130

Gambar 33. Modul *Sign Up* 131

Gambar 34. Modul *About* 131

Gambar 35. Modul Diagnosa *Normal User* 132

Gambar 36. Modul Diagnosa Admin.....133

Gambar 37. Modul Detail Info.....134

Gambar 38. Modul Ubah Fakta.....134

Gambar 39. Modul Tambah Awal 135

Gambar 40. Modul Kosongkan.....135

Gambar 41. Modul Tambah Fakta 136

Gambar 42. Modul *Help* 136

Gambar 43. Modul Sisipkan Fakta 137

Gambar 44. Modul Hapus Fakta 138

Gambar 45. *Flowchart* Sistem 139

Gambar 46. *Flowchart Log In*.....140

Gambar 47. *Flowchart Sign Up* 141

Gambar 48. *Flowchart About* 142

Gambar 49. *Flowchart* Kosongkan.....142

Gambar 50. *Flowchart* Diagnosa *Normal User* 143

Gambar 51. *Flowchart* Diagnosa *Admin* 144

Gambar 52. *Flowchart* Tambah Awal 145

Gambar 53. *Flowchart* Tambah Fakta 146

Gambar 54. *Flowchart* Hapus Fakta 147

Gambar 55. *Flowchart* Ubah Fakta 148

Gambar 56. *Flowchart* Sisipkan Fakta 149

Gambar 57. Tampilan Halaman *About* 151

Gambar 58. Tampilan Diagnosa *Admin*.....153

Gambar 59. Tampilan Diagnosa *User*.....156

Gambar 60. Tampilan Kotak Tambah 159

Gambar 61. Tampilan Halaman *Login*.....160

Gambar 62. Tampilan Halaman *Signup*.....162

Gambar 63. Tampilan Halaman Penyisipan Fakta.....163

Gambar 64. Tampilan Halaman Tambah Fakta 166

Gambar 65. Tampilan Halman Tambah Awal 169

Gambar 66. Tampilan Halaman Ubah Fakta 170

Gambar 67. Perhitungan Jumlah LOC (1) 188

Gambar 68. Perhitungan Jumlah LOC (2) 188

Gambar 69. Perhitungan Jumlah LOC dengan *VB6 Pure Code Lines Calculator* 190

Gambar 70. Perhitungan Jumlah *Error*.....192

Gambar 71. Perhitungan Jumlah *Error* dengan *VB Watch Debugger* 193

DAFTAR TABEL

Halaman

Tabel 1. Contoh Tabel Keputusan.....22

Tabel 2. Tabel Keputusan Identifikasi Pesawat.....23

Tabel 3. Tipe Data MySQL.....44

Tabel 4. Komponen *Entity Relationship Diagram* (ERD).....49

Tabel 5. Perkiraan Jumlah *Error* McConnell 58

Tabel 6. Metode Perkiraan *Error* Tiap KLOC.....59

Tabel 7. Kriteria Lolos/Gagal pada program *Windows Logo Certification*.....61

Tabel 8. Keterhubungan antara faktor kualitas perangkat lunak dengan ukuran-ukuran (*metrics*) (Pressman, 2001, hal. 512).....64

Tabel 9. Interval Skala Likert 80

Tabel 10. Format *Test Case* yang Digunakan dalam Pengujian 83

Tabel 11. Instrumen *Maintainability* 84

Tabel 12. Spesifikasi Uji Membuka Aplikasi 85

Tabel 13. Spesifikasi Uji Halaman *Login* 85

Tabel 14. Spesifikasi Uji Halaman *Login* (lanjutan).....86

Tabel 15. Spesifikasi Uji Halaman *About*.....86

Tabel 16. Spesifikasi Uji Halaman Pendaftaran 86

Tabel 17. Spesifikasi Uji Halaman Diagnosa Untuk *Normal User* 87

Tabel 18. Spesifikasi Uji Halaman Diagnosa Untuk *Admin*.....89

Tabel 19. Spesifikasi Uji Halaman Diagnosa Untuk *Admin* (lanjutan) 90

Tabel 20. Spesifikasi Uji Fitur Kembali Pada Halaman Diagnosa.....91

Tabel 21. Spesifikasi Uji Fitur Detail Info Pada Halaman Diagnosa 91

Tabel 22. Spesifikasi Uji Fitur Mengapa Pada Halaman Diagnosa 92

Tabel 23. Spesifikasi Uji Halaman Ubah Fakta 92

Tabel 24. Spesifikasi Uji Halaman Penyisipan Fakta 93

Tabel 25. Spesifikasi Uji Halaman Penyisipan Fakta (lanjutan) 94

Tabel 26. Spesifikasi Uji Halaman Tambah Fakta 94

Tabel 27. Spesifikasi Uji Halaman Tambah Fakta (lanjutan).....95

Tabel 28. Spesifikasi Uji Fitur Bantuan (*Help*) 95

Tabel 29. Spesifikasi Uji Halaman Tambah Awal.....96

Tabel 30. Spesifikasi Uji Fitur *Logout*.....96

Tabel 31. *Computer System Usability Questionnaire*.....98

Tabel 32. Standar kriteria faktor kualitas *functionality* dalam *Microsoft Certification Logo* (Bach, 2005).....100

Tabel 33. Konversi Jawaban Item Kuesioner ke Nilai Kuantitatif 102

Tabel 34. Kriteria Kategori Penilaian Ideal 103

Tabel 35. Kategori Penilaian Faktor Kualitas *Usability* 104

Tabel 36. Standar Kriteria Faktor Kualitas *Maintainability* 105

Tabel 37. Kamus Data Tabel Rule 115

Tabel 38. Kamus Data Tabel Login 115

Tabel 39. Uji *Alpha* Spesifikasi Uji Membuka Aplikasi 174

Tabel 40. Uji *Alpha* Spesifikasi Uji Halaman *Login* 174

Tabel 41. Uji *Alpha* Spesifikasi Uji Halaman *Login* (lanjutan).....175

Tabel 42. Uji *Alpha* Spesifikasi Uji Halaman *About* 175

Tabel 43. Uji *Alpha* Spesifikasi Uji Halaman Pendaftaran.....175

Tabel 44. Uji *Alpha* Spesifikasi Uji Halaman Diagnosa Untuk *Normal User*.....176

Tabel 45. Uji *Alpha* Spesifikasi Uji Halaman Diagnosa Untuk *Admin* 178

Tabel 46. Uji *Alpha* Spesifikasi Uji Halaman Diagnosa Untuk *Admin* (lanjutan) 179

Tabel 47. Uji *Alpha* Spesifikasi Uji Fitur Kembali Pada Halaman Diagnosa 180

Tabel 48. Uji *Alpha* Spesifikasi Uji Fitur Detail Info Pada Halaman Diagnosa..180

Tabel 49. Uji *Alpha* Spesifikasi Uji Fitur Mengapa Pada Halaman Diagnosa 181

Tabel 50. Uji *Alpha* Spesifikasi Uji Halaman Ubah Fakta 181

Tabel 51. Uji *Alpha* Spesifikasi Uji Halaman Penyisipan Fakta 182

Tabel 52. Uji *Alpha* Spesifikasi Uji Halaman Penyisipan Fakta (lanjutan).....183

Tabel 53. Uji *Alpha* Spesifikasi Uji Halaman Tambah Fakta.....183

Tabel 54. Uji *Alpha* Spesifikasi Uji Halaman Tambah Fakta (lanjutan) 184

Tabel 55. Uji *Alpha* Spesifikasi Uji Fitur Bantuan (*Help*).....184

Tabel 56. Uji *Alpha* Spesifikasi Uji Halaman Tambah Awal 185

Tabel 57. Uji *Alpha* Spesifikasi Uji Fitur *Logout* 185

Tabel 58. Penjabaran Hasil Uji Validitas.....189

Tabel 59. Penjabaran Hasil Uji Reliabilitas 191

Tabel 60. Penjabaran Hasil Uji Validitas Jumlah *Error* 194

Tabel 61. Hasil Uji Reliabilitas *Code Advisor for Visual Basic 6* 194

Tabel 62. Perbandingan Hasil Pengujian Faktor Kualitas *Correctnes* dengan Standar yang Digunakan.....195

Tabel 63. Rangkuman *Test Case* Fungsi Primer.....196

Tabel 64. Rangkuman *Test Case* Fungsi Pendukung.....197

Tabel 65. Perbandingan Hasil Pengujian Faktor Kualitas *Functionality*.....198

Tabel 66. Jawaban Responden Terhadap Pertanyaan Kuesioner *Usability*.....199

Tabel 67. Konversi Jawaban Item Kuesioner ke Nilai Kuantitatif 200

Tabel 68. Kategori Penilaian Faktor Kualitas *Usability* 201

Tabel 69. Hasil Uji *Maintainability* 202

Tabel 70. Hasil Uji *Maintainability* (lanjutan) 202

Tabel 71. Kecocokan Jawaban Responden Terhadap Minat Jurusan dengan Keluaran Program.....205

Tabel 72. Kecocokan Jawaban Responden Terhadap Minat Jurusan dengan Keluaran Program (lanjutan) 206

Tabel 73. Interpretasi Persentase Likert.....207

Tabel 74. Penyesuaian Interpretasi Persentase Likert.....208

DAFTAR LAMPIRAN

Halaman

Lampiran 1.*Knowledge Base* 212

Lampiran 2.Hasil Uji *Alpha* Perangkat Lunak.....224

Lampiran 3.Angket Uji *Beta*.....237

Lampiran 4.Rekap Hasil Uji *Beta* Perangkat Lunak.....239

Lampiran 5.*Test Case* Pengujian Faktor *Functionality* 241

Lampiran 6. Surat Keputusan Pengangkatan Pembimbing.....273

Lampiran 7. Surat Keterangan Ijin Penelitian.....274

BAB I
PENDAHULUAN

A. Latar Belakang

Setiap tahun, sedikitnya 1,5 juta siswa lulus ujian nasional tingkat SMA dan sederajat. Menteri Pendidikan dan Kebudayaan menyatakan angka kelulusan ujian nasional tingkat SMA dan sederajat tahun 2013 mencapai 99, 48% atau 1.573.036 siswa(2013). Lulusan yang berencana melanjutkan jenjang pendidikannya ke perguruan tinggi harus menentukankan pilihan jurusan kuliah yang akanditempuh kelak. Hal ini merupakan pengambilan keputusan yang cukup sulit bagi kebanyakan siswa Sekolah Menengah Atas, terutama bagi siswa yang tidak mempunyai cukup referensi maupun kesulitan dalam mencari informasi terkait jurusan kuliah di perguruan tinggi.

Kecenderungan yang terjadi saat ini, banyak siswa kelas XII yang tidak mengetahui minat dan bakatnya. Siswabimbang akan memilih jurusan apa selepas SMA nanti. Hal ini merupakan khas remaja Indonesia karena tidak terdidik untuk mengambil keputusan sendiri. Bahkan terkadang keputusan para siswa dipengaruhi oleh obsesi orang tua, pendapat teman atau figur-figur yang mereka idolakan. Hanya dengan berdasar kepadadorongan dan pendapat tersebut, seorang siswa dapat membuat keputusan yang tidak sesuai dengan minat dan bakatnya. Akibat yang buruk terjadi setelah itu, yaitu keengganan belajar dan menurunnya kualitas serta prestasi akademik karena siswa merasa salah dalam memilih jurusan kuliah.

Terdapat begitu banyak penelitian yang memaparkan tentang hubungan antara minat siswa dengan kesuksesannya di bangku kuliah. Secara umum, hasil penelitian itu menunjukkan bahwa keselarasan bakat dengan minat dapat membuat anak mencapai keberhasilan di bangku kuliah. Sebaliknya, jika seseorang memiliki minat pada suatu bidang namun tidak berbakat pada bidang tersebut, maka kemungkinan besar ia akan mendapatkan nilai indeks prestasi yang minim pada bidang tersebut.

Kemajuan ilmu komputer telah banyak mempengaruhi perkembangan diberbagai bidang termasuk bidang psikologi khususnya bimbingan konseling dan kecerdasan anak. Sistem pakar (*expert system*) sebagai salah satu hasil dari perkembangan ilmu komputer, khususnya di bidang kecerdasan buatan (*artificial intelegence*) dapat memberikan solusi untuk mengatasi masalah tersebut. Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti layaknya para pakar. Pengetahuan yang terkait dalam pemilihan jurusan seperti kecenderungan bakat dan jenis kecerdasan dari para ahli disimpan, kemudian diolah sedemikian rupa sehingga komputer dapat menghasilkan solusi yang tepat.

Sistem yang dibuat bukan berarti menggantikan peran guru bimbingan konseling di Sekolah Menengah Atas atau psikolog di masyarakat, melainkan berfungsi sebagai bahan pengetahuan masyarakat terhadap permasalahan yang berhubungan dengan pemilihan jurusan kuliah bagi siswa lulusan Sekolah Menengah Atas.

Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi ini dapat membantu siswa kelas XII Sekolah Menengah Atas(SMA) yang sedang dihadapkan pada kesulitan dalam memilih jurusan kuliah yang sesuai dengan kemampuan mereka. Pengguna dapat melakukan konsultasi dirumah tanpa harus datang atau berkonsultasi langsung dengan guru bimbingan konseling maupun psikolog sehingga dapat menekan biaya pengeluaran serta dapat mengatasi rasa malu siswa dalam melakukan konsultasi.

Pengguna hanya diminta menjawab rangkaian pertanyaan dasar yang berhubungan dengan minat dan jenis kecerdasan,kemudian komputer akan memproses data dan memunculkan solusi berupa jurusan yang sesuai bagi pengguna. Setelah mendapatkan jurusan yang dianjurkan, pengguna dapat melihat keterangan singkat terkait jurusan tersebut untuk selanjutnya dipelajari lebih lanjut oleh pengguna.

Seperti halnya perangkat lunak lain, aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi ini dibuat berdasarkan kaidah rekayasa perangkat lunak (*software engineering*) mulai dari proses awal analisis kebutuhan hingga akhir pengujian. Pada akhirnya, aplikasi sistem pendukung keputusan yang dibuat diharapkan dapat memenuhi standar kualitas perangkat lunak dalam kaidah rakayasa perangkat lunak (*software engineering*). Kualitas perangkat lunak ditentukan oleh berbagai faktor. Beberapa ahli maupun organisasi telah merumuskan kriteria-keriteria pengujian kualias perangkat lunak. McCall, Richards, dan Walter merumuskan kriteria-kriteria untuk melakukan pengujian kualitas perangkat lunak yang terdiri dari beberapa faktor kualitas yaitu :

maintainability, flexibility, testability, portability, reusability, interoperability, correctness, reliability, usability, integrity, dan efficiency. Selain itu, *International Standard Organization* (ISO) juga mengeluarkan standar ISO-9126 yang terdiri dari enam faktor kualitas yaitu *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, dan *portability*.

Berdasarkan latar belakang tersebut, melalui Tugas Akhir ini Penulis bermaksud untuk mengembangkan aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi dan melakukan analisis kualitas aplikasi tersebut.

B. Identifikasi Masalah

Berdasarkan uraian latar belakang di atas, maka ada beberapa permasalahan yang dapat diidentifikasi adalah sebagai berikut :

1. Banyaknya siswa kelas XII yang tidak mengetahui minat dan bakatnya serta akan memilih jurusan apa selepas lulus SMA.
2. Upaya yang dilakukan oleh pihak sekolah untuk membimbing anak didiknya dalam menentukan pilihan jurusan kuliah belum maksimal.
3. Belum adanya aplikasi sistem pakar yang dapat membantu siswa SMA dalam memilih jurusan kuliah yang sesuai dengan minat dan bakatnya serta memberi info tentang jurusan tersebut.
4. Belum adanya analisis kualitas pada aplikasi sistem pendukung keputusan pemilihan jurusan di perguruan tinggi.

C. Batasan Masalah

Aplikasi sistem pakar mencakup bidang yang cukup luas, Untuk lebih memfokuskan permasalahan yang akan diteliti,maka permasalahannya dibatasi sebagai berikut:

1. Sistem pakar yang membantu mengarahkan siswa sekolah menengah atas dalam memilih jurusan yang ada di perguruan tinggi agar sesuai dengan minat dan tipe kepribadian (RIASEC) yang dimiliki.
2. Metode representasi pengetahuan yang digunakan adalah sistem pakar berbasis *Rule* dengan metode inferensi *Forward Chaining.*
3. *Software* yang akan digunakan dalam aplikasi ini adalah Visual Basic Versi 6.0 dan Mysql.
4. Uji kelayakan yang akan dipakai hanya terbatas pada empatfaktor yaitu*correctness, functionality,usability,* dan*maintainability.*

D. Rumusan Masalah

Dari uraian latar belakang, identifikasi masalah serta batasan masalah diatas maka dirumuskan permasalahan sebagai berikut:

1. Bagaimanakah perancangan Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi.
2. Bagaimanakahanalisiskelayakan dari Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggiberdasarkan faktor*correctness, functionality,usability*serta *maintainability*dalam *softwarequalitytesting*.

E. Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

1. MengembangkanSistem Pendukung Keputusan Pemilihan Jurusan Di Perguruan Tinggi menggunakan Visual Basic dan MySQL.
2. Mengetahui analisisSistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggidari segi*correctness*,*functionality*,*usability*dan*maintainability*dalam *softwarequalitytesting*.
3. Diharapkan program ini dapat membantu mengarahkan siswa SMA dalam memilih jurusan kuliah yang tepat berdasarkan minat dan jenis kecerdasan yang dimiliki oleh masing-masing siswa.

F. Manfaat Penelitian

Berikut merupakan beberapa manfaat dari penulisan tugas akhir ini:

1. Memberikan kemudahan bagisiswa dan orang tua siswa dalam konsultasi pemilihan jurusan kuliah yang tepat.
2. Bagi pakar dan guru sebagai media penyimpanan kemampuan dan keahlian pakar dalam jangka waktu panjang.
3. Bagi Sekolah Menengah Atas meningkatkan rasa percaya diri lulusan yang akan melanjutkan studi sehingga lulusan tersebut dapat berprestasi dan meningkatkan kualitas lulusan.
4. Aplikasi sistem pakar yang telah dibuat dapat menjadi referensi untuk pengembangan aplikasi sistem pakar sejenis di kemudian hari.

BAB II
KAJIAN TEORI

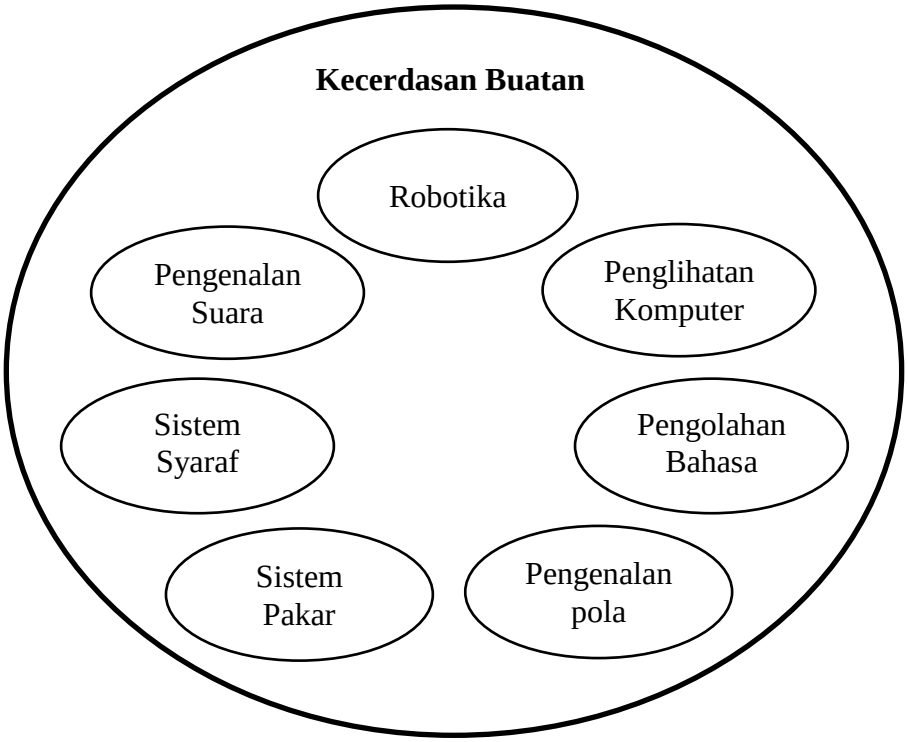
A. Landasan Teori

1. Kecerdasan Buatan(*Artificial Intelligence*)

Menurut Hartati dan Iswanti(2008), kecerdasan buatan (*Artificial Intelegence*) adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Ilmu komputer yang dimaksud adalah mengembangkan perangkat keras(*hardware*) dan perangkat lunak (*software*) untuk menirukan tindakan manusia. Tindakan menusia yang ditirukan antara lain penalaran, penglihatan, pembelajaran, pemecahan masalah, pemahaman bahasa alami dan sebagainya. Sesuai dengan definisi tersebut, maka kecerdasan buatan dipelajari dalam berbagai bidang.

Berdasarkan definisi tersebut, maka dapat dibedakan antara kecerdasan buatan dengan program atau aplikasi konvensional. Kecerdasan buatan memungkinkan program komputer dapat menyimpan informasi baru dalam sebuah basis pengetahuan (*knowledge base*) yang dapat digunakan pada masa yang akan datang. Sedangkan program konvensional hanya dapat menyelesaikan persoalan yang diprogram secara spesifik. Jika ada informasi baru, sebuah program konvensional harus diubah untuk menyesuaikan diri dengan informasi baru tersebut. Selain diperlukan waktu yang relatif lama, kemungkinan terjadinya *error* juga cukup besar.Hal itu menyebabkan kecerdasan buatan menjadi cabang ilmu komputer yang ramai dikembangkansaat ini.

Bidang-bidang yang termasuk dalam kecerdasan buatan terlihat pada Gambar 1 (Giarratano & Riley, 2005). Kecerdasan buatan menyelesaikan permasalahan dengan mendayagunakan komputer untuk memecahkan masalah yang komplek dengan cara mengikuti proses penalaran manusia. Salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia dan merupakan penyelesaian pendekatan yang sangat bagus untuk permasalahan kecerdasan buatan klasik dari pemrograman *intelligent* (cerdas) adalah sistem pakar.



Gambar 1. Ruang Lingkup Kecerdasan Buatan

Pemecahan masalah yang komplek biasanya hanya dapat dilakukan oleh sejumlah orang yang sangat terlatih, yaitu pakar. Dengan penerapan teknik kecerdasan buatan, sistem pakar menirukan apa yang dikerjakan oleh pakar berdasarkan pengetahuan yang dimilikinya.

2. Sistem Pakar(*Expert System*)

a. Pengertian Sistem Pakar (*Expert System*)

Sistem pakar adalah salah satu cabang kecerdasan buatan yang menggunakan pengetahuan-pengetahuan khusus yang dimiliki oleh seorang ahli untuk menyelesaikan suatu masalah tertentu(Giarratano & Riley, 2005). Pengetahuan dalam sistem pakar diperoleh dari pakar bidang tertentu, buku, jurnal ilmiah, majalah maupun dokumentasi yang tercetak lainnya.Sumber pengetahuan tersebut biasa disebut dengan sumber keahlian. Pengetahuan-pengetahuan tersebut kemudian direpresentasikan dalam format tertentu dan dikumpulkan dalam suatu basis pengetahuan. Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rule*)dengan basis pengetahuan tertentu yang diberikan oleh pakar dalam bidang tersebut. Kombinasi dari kedua hal tersebutdisimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah yang dihadapinya.

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tertentu(Martin & Oxman, 1988). Sistem pakar dirancang agar dapat melakukan penalaran seperti seorang pakar sehingga pengguna dapat menyelesaikan masalah yang biasanya hanya dapat diselesaikan oleh pakar. Tujuan pengembangan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mendistribusikan pengetahuan manusia ke dalam bentuk sistem sehingga dapat lebih mudah digunakan oleh banyak orang.

b. Manfaat Sistem Pakar

Sistem pakar (*expert system*) merupakan salah satu cabang kecerdasan buatan yang digunakan untuk menyelesaikan masalah tertentu pada bidang yang digeluti oleh para ahli atau pakar. Dengan kata lain sistem pakar merupakan paket perangkat lunak yang ditujukan sebagai penyedia nasihat dan sarana bantu dalam memecahkan berbagai masalah di bidang spesialisasi tertentu seperti kedokteran, pendidikan dan lain sebagainya. Hal ini berarti sistem pakar dibuat dengan harapan dapat bermanfaat bagi kehidupan manusia dalam memecahkan masalah. Menurut Sutojo, Mulyanto dan Suhartono(2011), sistem pakar mempunyai beberapa manfaat antara lain :

1) Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2) Membuat seorang yang awam menjadi bekerja seperti layaknya seorang pakar.
3) Meningkatkan kualitas, dengan memberi nasihat yang konsisten dan mengurangi kesalahan.
4) Mempu menangkap pengetahuan dan kepakaran seseorang.
5) Dapat beroperasi dilingkungan yang berbahaya.
6) Memudahkan akses pengetahuan seorang pakar.
7) Handal, karena sistem pakar tidak pernah menjadi bosan, lelah atau sakit.
8) Meningkatkan kapabilitas sistem komputer.
9) Bisa digunakan sebagai media pelengkap atau pelatihan.
10) Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari pakar.

c. Struktur Sistem Pakar

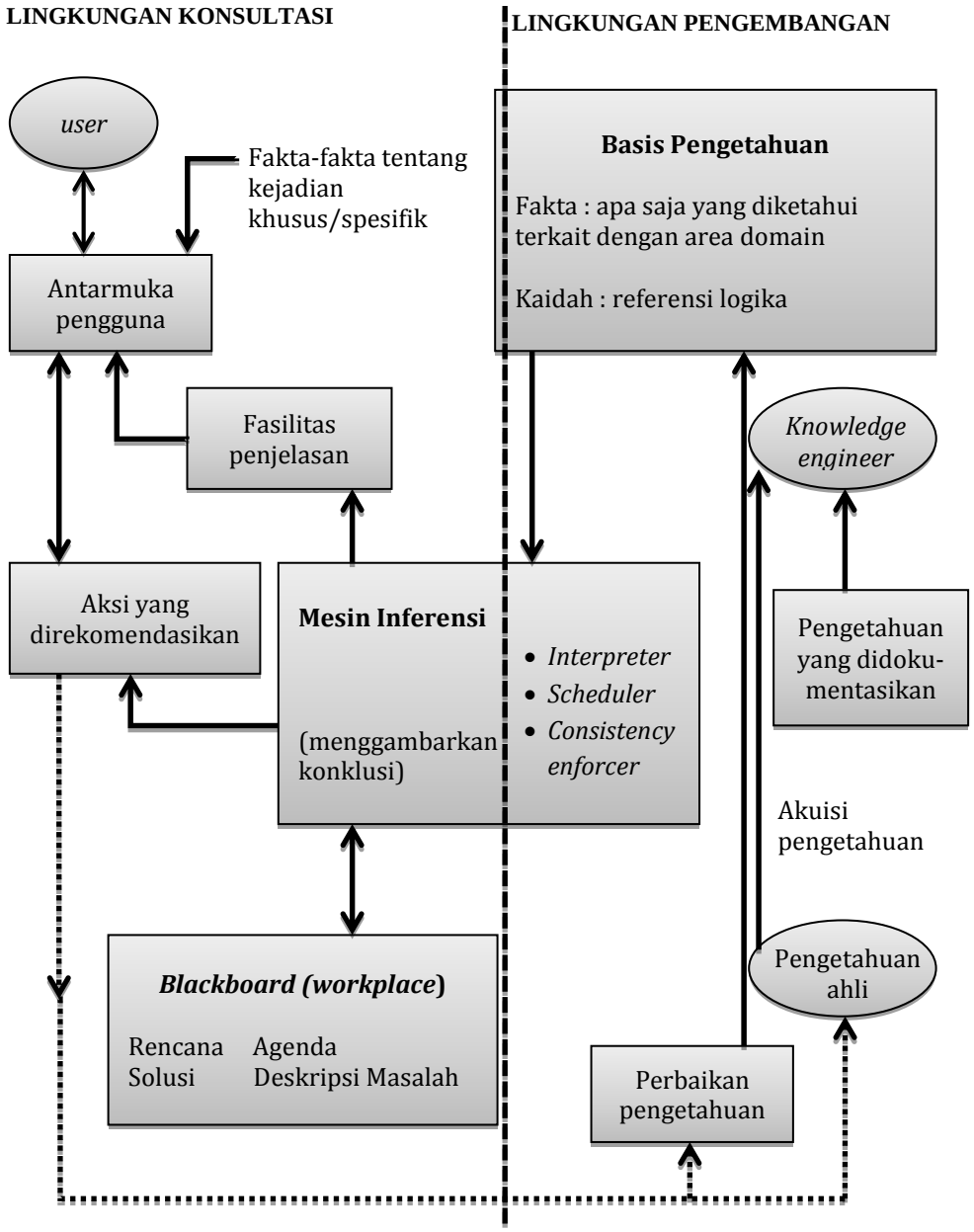
Sistem pakar dapat dilihat dari sudut pandang lingkungan (*environment*) dalam sistem. Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembang (*development environment*) dan lingkungan konsultasi (*consultation environment*)(Turban, 2005). Lingkungan pengembang ditujukan bagi pembangun sistem pakar untuk memasukkan pengetahuan hasil akuisisi pengetahuan pakar kedalam basis pengetahuan. Sedangkan lingkungan konsultasi diperuntukkan bagi pengguna yang bukan pakar untuk melakukan konsultasi dengan sistem yang tujuannya adalah memperoleh pengetahuan pakar dan mendapatkan nasehat pakar untuk solusi permasalahnnya. Secara lengkap struktur sistem pakar yang menekankan pada lingkungan yang ada dalam sistem dapat dilihat pada Gambar 2.

Berdasarkan gambar struktur sistem pakar pada gambar 2, dapat dijelaskan bahwa hasil pemrosesan yang dilakukan oleh mesin inferensi dari sudut pandang pengguna non pakar berupa aksi/konklusi yang direkomendasikan oleh sistem pakar atau dapat juga berupa penjelasan jika memang dibutuhkan oleh pengguna. Sedangkan dari sudut pandang pembangun sistem dalam lingkup pengembangan, mesin inferensi terdiri dari 3 elemen penting yaitu:

1) *Interpreter*/ interpreter kaidah terdapat pada sebagian besar sistem, elemen ini mengeksekusi item-item agenda yang terpilih dengan menggunakan kaidah berbasis pengetahuan yang bersesuaian.

2) Penjadwalan / *scheduller*, *e*lemen ini mengelola pengontrolan agenda. Penjadwalan memperkirakan pengaruh dari pengguna kaidahinferensi pada prioritas item atau kriteria lain pada agenda.

3) Pelaksanaan konsistensi /*consistency enforcer*, *e*lemen ini berusaha untuk mengelola penyajian solusi secara konsisten.



Gambar 2. Struktur Sistem Pakar(Turban, 2005**)**

d. Komponen Sistem Pakar

Sistem pakar merupakan sebuah program untuk menyimpan pengetahuan pakar agar dapat digunakan seolah-olah sedang berkonsultasi dengan seorang pakar. Sistem pakar sebagai sebuah program yang difungsikan untuk menirukan pakar, harus bisa melakukan hal-hal yang dapat dikerjakan oleh seorang pakar. Untuk membangun sistem seperti itu maka komponen-komponen yang harus dimiliki sistem pakar adalah sebagai berikut (Giarratano & Riley, 2005):

1) Antar Muka Pengguna (*User Interface*)

Sistem pakar menggantikan seorang pakar dalam suatu situasi tertentu, maka sistem harus menyediakan komunikasi antara sistem dan pemakainyayang disebut sebagai *user interface*. *User interface*atau antar muka pengguna menerima informasi dari pengguna dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu *user interface*juga menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pengguna.Antar muka yang efektif dan ramah pengguna (*user-friendly*) penting sekali bagipengguna yang tidak ahli atau awam dalam bidang yang diterapkan pada sistem pakar tersebut.

2) Basis Pengetahuan (*Knowledge Base*)

Menurut Hartati dan Iswanti(2008), basis pengetahuan (*knowledge base*) merupakan kumpulan pengetahuan bidang tertentu pada tingkatan pakar dalam format tertentu. Pengetahuan tersebut diperoleh dari akumulasi pengetahuan pakar dan sumber-sumber pengetahuan lainya.

*Knowledge base*bersifat dinamis, bisa berkembang dari waktu ke waktu. Perkembangan ini disebabkan karena pengetahuan selalu bertambahseiring perkembangan jaman. Dalam sistem pakar, basis pengetahuan terpisah dari mesin inferensi (*inference engine*). Hal ini dilakukan agar pengembangan sistem pakar dapat dilakukan secara leluasa tanpa mengganggu mesin inferensi.

3) Mekanisme Inferensi (*Inference Machine*)

Menurut Sutojo, Mulyanto, dan Suhartono(2011), mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan.

Mekanisme ini akan menganalisa masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik, kemudian dimulai pelacakannya dengan mencocokan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.Dalam prosesnya, mesin inferensi menggunakan strategi pengendalian, yaitu strategi yang berfungsi sebagai panduan arah dalam melakukan penalaran. Terdapat 2 pendekatan untuk mengontrol inferensi sistem pakar berbasis aturan, yaitu *forward chaining*dan *backward chaining*.

a) *Forward Chaining*(Runut Maju)

Sutojo, Mulyanto dan Suhartono (2011) menjelaskan bahwa *forward chaining* adalah teknik inferensi yang dimulai dengan pengumpulan fakta – fakta yang diketahui, kemudian mencocokkan fakta – fakta tersebut dengan *rules* yang ada sampai akhirnya didapat konklusi akhir.

b) *Backward Chaining*(Runut Balik)

Sutojo, Mulyanto dan Suhartono (2011) menjelaskan bahwa *backward chaining* adalah teknik inferensi yang bekerja mundur. Proses dimulai dari *goal* (konklusi), kemudian pencarian dijalankan untuk mencocokkan apakah fakta – fakta cocok dengan *rules*.

4) Memori Kerja (*Working Memory*)

Menurut Hartati dan Iswanti(2008), memori kerja (*working memory*) merupakan bagian dari sistem pakar yang menyimpan fakta-fakta yang diperoleh saat dilakukan proses konsultasi. Fakta-fakta yang diperoleh dari proses konsultasi nantinya akan diolah oleh mesin inferensi berdasarkan pengetahuan yang disimpan dalam basis pengetahuan (*knowledge base*) untuk menentukan suatu keputusan pemecahan masalah. Keputusan pemecahan masalah atau konlusi yang dihasilkan dapat berupa hasil diagnosa, tindakan, maupun akibat.

Sedangkan untuk menjadikan sistem pakar menjadi lebih menyerupai seorang pakar yang berinteraksi dengan pemakai, maka dilengkapi dengan fasilitas berikut:

1) Fasilitas Penjelas (*Explanation Facility*)

Proses menentukan keputusan yang dilakukan oleh mesin inferensi selama sesi konsultasi mencerminkan proses penalaran seorang pakar. Kadangkala pemakai sistem pakar bukanlah ahli dalam bidang tersebut, oleh karena itu dibuatlah fasilitas penjelas. Hartati dan Iswanti(2008)menjelaskan bahwa fasilitas penjelas inilah yang dapat memberikan informasi kepada pemakai mengenai jalannya penalaran sehingga dihasilkan suatu keputusan. Penjelasan dapat berupa

keterangan setelah pertanyaan diajukan, yaitu penjelasan atas pernyataan mengapa, atau penjelasan atas pertanyaan bagaimana sistem mencapai konklusi.

Setiap saat pemakai dapat menanyakan kepada sistem pakar bagaimana konklusi didapatkan, kemudian fasilitas penjelas akan memberikan penjelasan yang telah diformat dengan bagus. Tujuan adanya fasilitas penjelas ini antara lain untuk membuat sistem menjadi lebih cerdas dan memuaskan psikologis pemakai ,karena menunjukkan adanya proses analisa pengetahuan.

2) Fasilitas Akuisisi Pengetahuan (*Knowledge Acquisition Facility*)

Pengetahuan pada sistem pakar dapat berubah sewaktu-waktu sesuai dengan perkembangan pengetahuan pada saat itu, pengetahuan dapat ditambahkan saat pengetahuan baru diperoleh atau pengetahuan dapat diubah saat pengetahuan yang sudah ada tidak berlaku lagi. Hal ini dilakukan agar pengguna dapat menggunakan sistem pakar yang lengkap dan sesuai dengan perkembangan pengetahuan. Untuk melakukan proses perubahan ini sistem pakar dilengkapi dengan fasilitas akuisisi pengetahuan.

Akuisisi pengetahuan adalah proses pengumpulan, pemindahan, dan transportasi dari keahlian/kepakaran pemecahan masalah yang berasal dari beberapa sumber pengetahuan ke dalam bentuk yang dimengerti oleh komputer (Hartati & Iswanti, 2008). Dengan adanya fasilitas ini pada sistem maka seorang pakar dapat menambahkan pengetahuan maupun kaidah baru pada sistem pakar dengan mudah. Fasilitas akuisisi pengetahuan hanya bisa diakses oleh pakar untuk menjamin bahwa pengetahuan dalam sistem pakar ini *up to date* dan valid.

e. Representasi Pengetahuan

1) Pengetahuan

Pemrosesan yang dilakukan oleh sistem pakar merupakan pemrosesan pengetahuan, bukan pemrosesan data seperti yang dikerjakan dengan pemrograman konvensional yang kebanyakan dilakukan oleh sistem informasi. Hartati dan Iswanti(2008) menjelaskan bahwa pengetahuan (*knowledge*) adalah pemahaman secara praktis maupun teoritis terhadap suatu obyek atau domain tertentu. Pemahaman tentang domain dapat diperoleh dari pelatihan, membaca, pendidikan maupun dari pengalaman. Pengetahuan yang digunakan pada sistem pakar merupakan serangkaian informasi mengenai gejala-diagnosa, sebab-akibat, aksi-reaksi tentang domain tertentu. Oleh karena itu pengetahuan merupakan hal yang sangat penting dalam sistem pakar, menurut ekspresi klasis Wirth (Giarratano & Riley, 2005) dituliskan sebagai berikut :

Algoritma + Struktur Data = Program

Pengetahuan + Inferensi = Sistem Pakar

Menurut Hartati dan Iswanti(2008) pengetahuan dapat digolongkan menjadi 3 kategori yaitu: pengetahuan deklaratif, pengetahuan prosedural, dan pengetahuan tacit (*tacit knowledge*). Berikut merupakan penjelasan mengenai 3 kategori pengetahuan tersebut:

a) Pengetahuan Deklaratif

Pengetahuan deklaratif terkait dengan nilai kebenaran, apakah suatu itu bernilai benar atau salah. Pengetahuan deklaratif mengacu pada fakta dan assersi serta diasosiasikan/dihubungkan dengan apa yang terlibat dalam pemecahan masalah (Hartati & Iswanti, 2008). Pengetahuan ini dalam penyajiannya

menggunakan basis logika dan pendekatan relasi. Representasi logika menggunakan logika proporsional dan logika predikat, sedangkan pendekatan relasi menggunakan model jaringan semantik, *graphs* dan pohon keputusan (*decision tree*).

b) Pengetahuan Prosedural

Pengetahuan prosedural terkait dengan bagaimana melakukan sesuatu. Kategori pengetahuan ini mengacu pada serangkaian tindakan dan konsekuensinya serta diasosiasikan dengan bagaimana menerapkan strategi atau prosedur penggunaan pengetahuan yang tepat untuk memecahkan masalah (Hartati & Iswanti, 2008). Pengetahuan ini menggunakan algoritma sebagai prosedur pemecahan masalah.

c) Pengetahuan Tacit

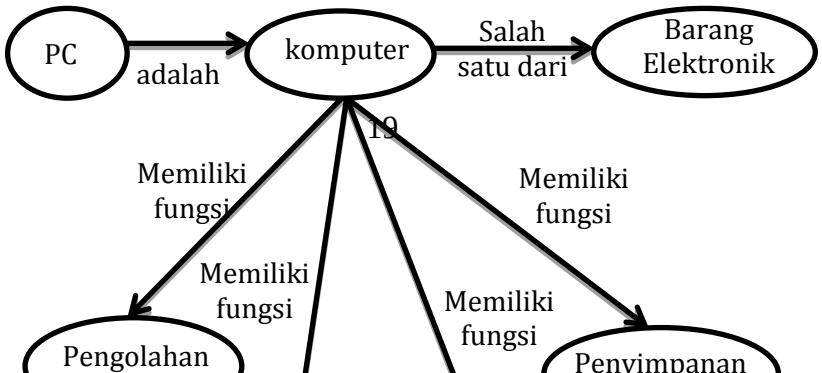
Pengetahuan tacit (*tacit knowledge*) disebut juga pengetahuan "tidak sadar" (*unconscious knowledge*), karena tidak dapat diekspresikan dengan bahasa. Istilah "tacit" sendiri mengandung arti dipahami tetapi tidak dapat dikatakan (Hartati & Iswanti, 2008). Contoh pengetahuan tacit adalah bagaimana menggerakkan tangan. Pertanyaan ini dapat dijawab dengan cara mengencangkan atau mengendorkan otot dan tendon tertentu. Pertanyaan akan berkembang, bagaimana untuk membuat otot dan tendon tertentu tersebut menjadi kencang atau kendor. Hal ini sulit diungkapkan dengan bahasa dalam format tertentu, tetapi menjadi mudah dipahami jika dilakukan atau ditunjukkan contohnya, seperti pada Sistem Jaringan Tiruan.

2) Model Representasi Pengetahuan

Sistem pakar merupakan program komputer yang menggunakan pengetahuan dari ahli atau pakar untuk menyelesaikan suatu permasalahan. Pengetahuan sangat penting dalam sistem pakar, oleh karena itu dibutuhkan format tertentu agar pengetahuan dari ahli tersebut dapat dimengerti oleh komputer. Representasi pengetahuan dimaksudkan untuk mengorganisasikan pengetahuan dalam bentuk dan format tertentu agar bisa dimengerti oleh komputer. Mengingat pentingnya representasi pengetahuan dalam sistem pakar, untuk membuat sistem pakar yang efektif harus dipilih representasi pengetahuan yang tepat. Pemilihan representasi pengetahuan yang tepat akan membuat sistem pakar dapat mengakses basis pengetahuan tersebut untuk keperluan pembuatan keputusan. Beberapa model representasi pengetahuan antara lain :

a) Jaringan Semantik (*Semantic Nets*)

Menurut Hartati dan Iswanti(2008)jaringan semantik merupakan penggambaran grafis dari pengetahuan yang memperlihatkan hubungan hirarkis dari obyek-obyek. Komponen dasar untuk merepresentasikan pengetahuan dalam bentuk jaringan semantik adalah simpul (*node*) dan penghubung (*link*). Simpul merepresentasikan obyek dan hubungan antar obyek dinyatakan oleh penghubung yang diberi label untuk menyatakan hubungan yang direpresentasikan. Jaringan ini digunakan untuk informasi proporsional. Informasi proporsional merupakan bahasa deklaratif karena menyatakan fakta. Contoh jaringan semantik dapat dilihat pada Gambar 3.



Gambar 3. Contoh Jaringan Semantik

b) Bingkai (*Frame*)

Hartati dan Iswanti(2008) menjelaskan bahwa bingkai berupa kumpulan slot-slot yang berisi atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berupa kejadian, lokasi, situasi maupun elemen-elemen lainnya.

Bingkai digunakan untuk representasi pengetahuan deklaratif. Bingkai memuat deskripsi sebuah objek dengan menggunakan tabulasi informasi yang berhubungan dengan obyek. Dengan kata lain bingkai mengelompokkan atribut sebuah obyek dan membantu menirukan cara seseorang mengorganisasi informasi tentang sebuah obyek menjadi kumpulan data. Representasi pengetahuan dengan bingkai ini sesuai untuk jenis pengetahuan yang memiliki subyek sempit, lebih bersifat pasti dan jarang berubah-ubah isinya kecuali terdapat kondisi khusus. Contoh representasi pengetahuan dengan bingkai terlihat pada Gambar 4.

Nama	Komputer
Spesialisasi	Jenis barang elektronik
Produk	Apple, Hp, Dell, Toshiba, Acer, Asus, Sony, Compaq, IBM
Bahan tenaga	Listrik

Gambar 4. Bingkai Komputer

c) Kaidah Produksi (*Production Rule*)

Hartati dan Iswanti(2008) menjelaskan bahwa kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dalam bentuk *if-then* (jika-maka). Kaidah *if-then* menghubungkan *antecendent* (entesenden) dengan konsekuensi yang diakibatkannya. Terdapat beberapa struktur kaidah *if-then* yang menghubungkan obyek atau atribut, contoh yang sering digunakan antara lain:

IF premis THEN konklusi

IF aksi THEN reaksi

IF sebab THEN akibat

IF gejala THEN diagnose

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat tertentu. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan.

Pengetahuan domain tertentu yang diperoleh harus melalui beberapa langkah sebelum sampai pada bentuk kaidah produksi. Langkah tersebut adalah

menyajikan pengetahuan yang didapatkan dalam bentuk tabel keputusan (*decision table*) kemudian dari tabel keputusan dibuat pohon keputusan (*decision tree*).

(1) Tabel Keputusan (*Decision Table*)

Tabel keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Hartati dan Iswanti(2008) menjelaskan bahwa tabel keputusan merupakan matrik kondisi yang dipertimbangkan dalam pendeskripsian kaidah. Kaidah yang disajikan dalam bentuk kaidah produksi dibentuk dari pengubahan tabel keputusan. Pembuatan kaidah sendiri dilakukan melalui beberapa tahapan. Sebagai contoh perhatikan pembuatan kaidah dari tabel keputusan pada Tabel 1.

Tabel 1.Contoh Tabel Keputusan

Goal / Kondisi	Goal 1	Goal 2
Kondisi 1	√	
Kondisi 2	√	√
Kondisi 3		√

Goal 1 merupakan konklusi dari kaidah 1 dan Goal 2 merupakan konklusi dari kaidah 2. Konklusi dapat dicapai bila kondisi-kondisi yang mendukung terpenuhi. Tanda centang (√) pada kolom Goal menunjukkan kondisi tersebut harus dipenuhi untuk mencapai konklusi tersebut. Setelah melihat data tabeldiatasmaka didapatkan kaidah seperti berikut ini :

Kaidah 1 : Goal 1 IFKondisi 1 AND Kondisi 2

Kaidah 2 : Goal 2 IFKondisi 2 AND Kondisi 3

(2) Pohon Keputusan (*Decision Tree*).

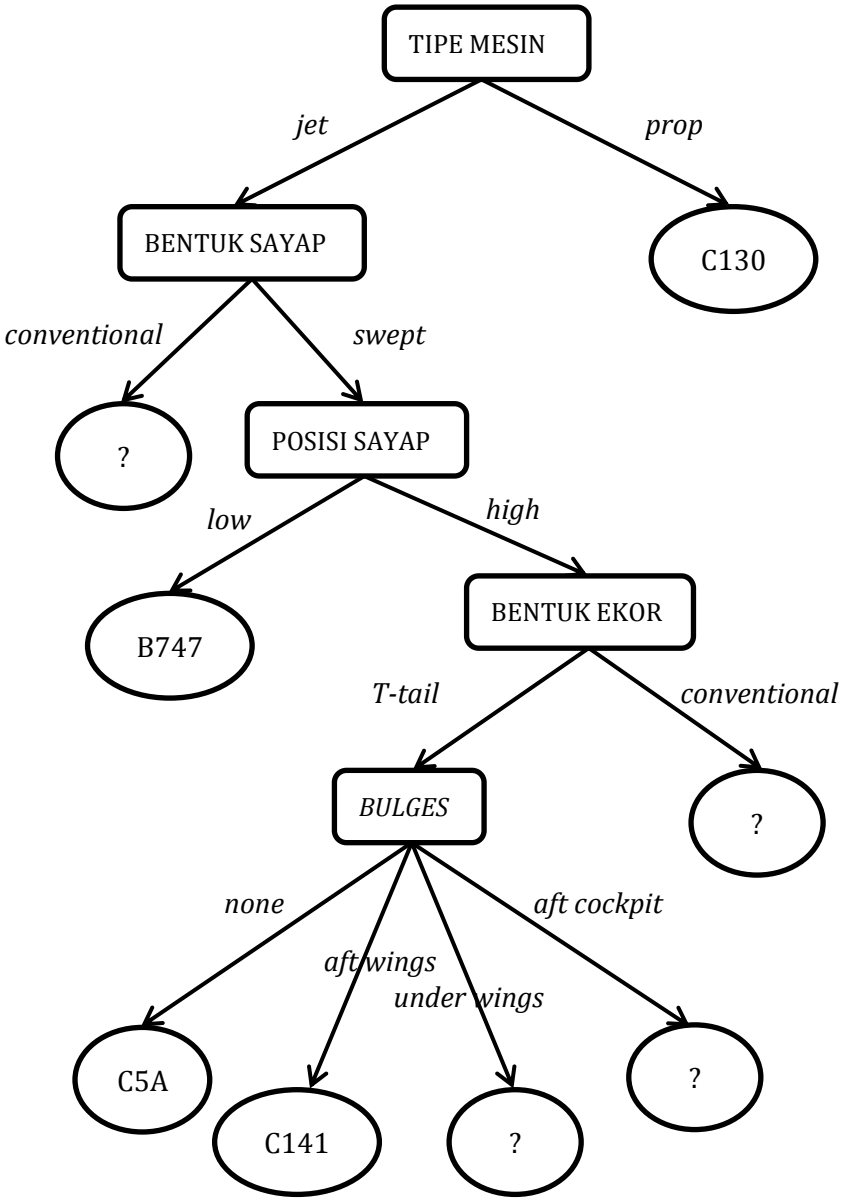
Meskipun kaidah dapat diperoleh secara langsung dari tabel keputusan tetapi untuk menghasilkan kaidah yang efisien terdapat suatu langkah yang harus ditempuh yaitu membuat pohon keputusan terlebih dahulu. Hartati dan Iswanti(2008) menjelaskan bahwa dari pohon keputusan dapat diketahui atribut (kondisi) yang dapat direduksi sehingga menghasilkan kaidah yang efisien dan optimal. Sebagai contoh akan disajikan tabel keputusan identifikasi pesawat pada Tabel 2. Tabel keputusan tersebutyang nantinya akan diubah menjadi pohon keputusan identifikasi pesawat pada gambar 5..

Tabel 2. Tabel Keputusan Identifikasi Pesawat

Tipe Pesawat / Atribut	C130	C141	C5A	B747
Tipe Mesin	*Prop*	*Jet*	*Jet*	*Jet*
Posisi Sayap	*High*	*High*	*High*	*Low*
Bentuk Sayap	*Conventional*	*Swept-back*	*Swept-back*	*Swept-back*
Bentuk Ekor	*Conventional*	*T-tail*	*T-tail*	*Conventional*
Bulges	*Under wings*	*Aft wings*	*None*	*Aft cockpit*

Sesuai data pada tabel keputusan diatas, sebuah pesawat termasuk tipe B747 jika memiliki ciri-ciri: tipe mesinnya *jet*, posisi sayap *low*, bentuk sayap *swept-back*, bentuk ekor *conventional* dan bulges *aft-cockpit*. Identifikasi pesawat tipe tertentu dapat dengan mudah diketahui setelah melihat tabel keputusan di atas. Menurut data pada tabel keputusan diatas identifikasi pesawat dilakukan dengan memperhatikan 5 atribut yaitu tipe mesin, posisi sayap, bentuk sayap, bentuk ekor, dan bbulges. Namun pada kenyataannya identifikasi juga dapat dilakukan hanya dengan memperhatikan atribut-atribut yang benar-benar membedakan tipe pesawat satu dengan lainnya. Berikut merupakan pohon keputusan yang dibuat

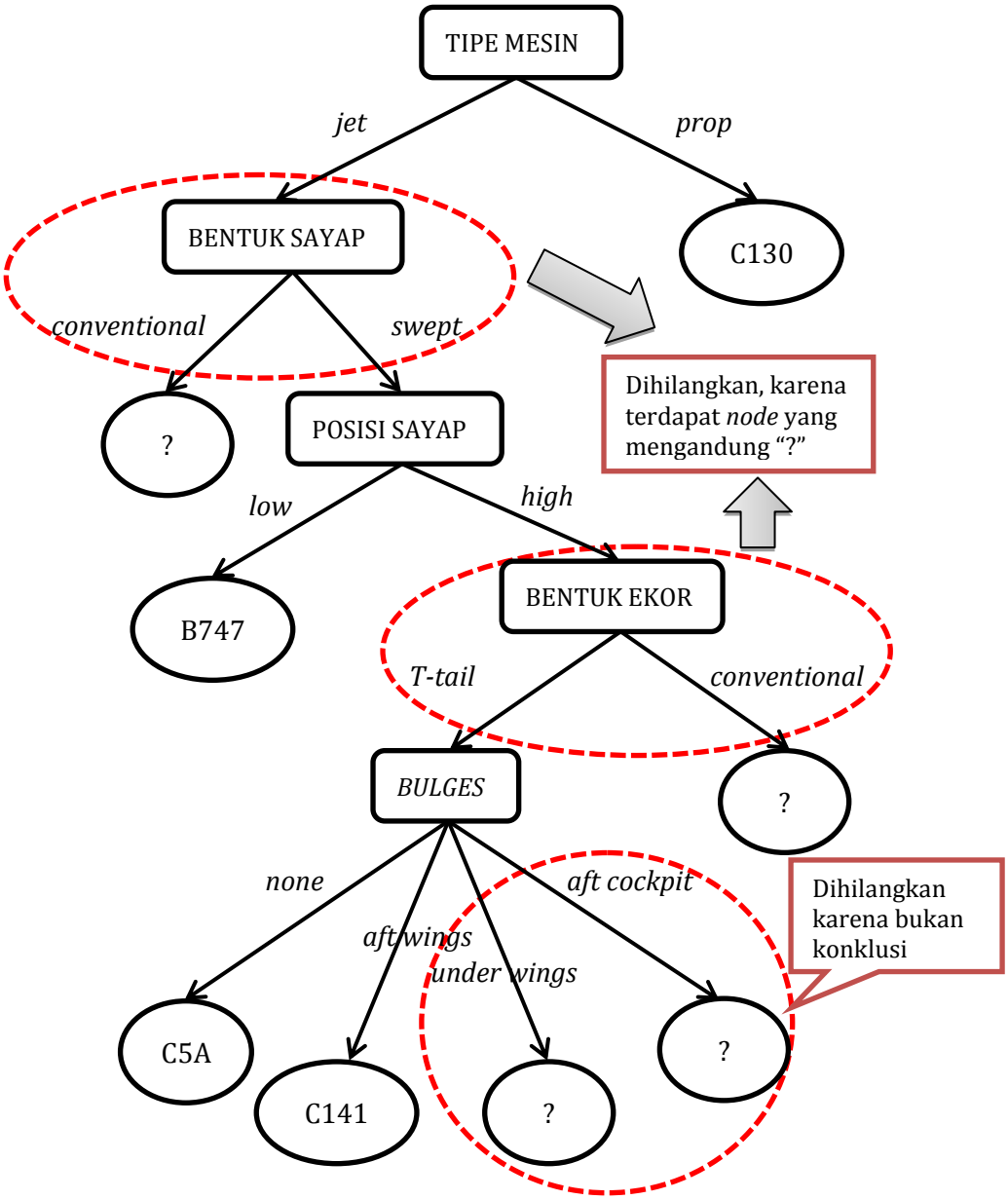
berdasarkan 5 atribut yang telah disajikan pada tabel keputusan identifikasi pesawat.



Gambar 5. Pohon Keputusan Identifikasi Pesawat

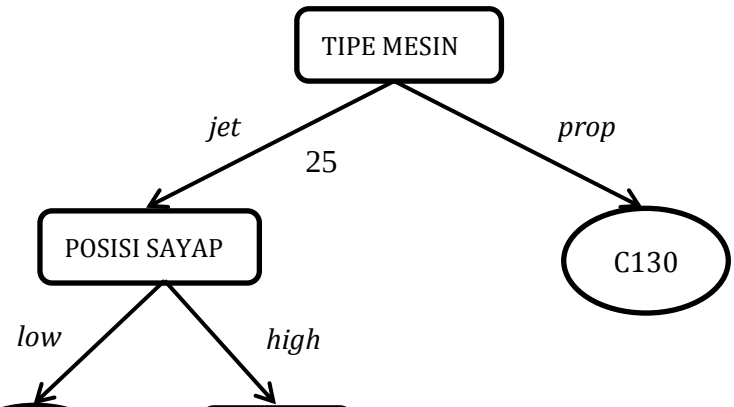
Pohon keputusan diatas dapat digunakan sebagai acuan untuk mereduksi atribut-atribut yang sebenarnya dapat dihilangkan dalam proses identifikasi pesawat. Atribut yang dapat dihilangkan adalah atribut-atribut yang mengandung

node dengan notasi tanda tanya kecuali dari atribut tersebut dapat disimpulkan suatu konklusi. Proses reduksi atribut dapat dilihat pada Gambar 6.



Gambar 6. Proses Reduksi Pohon Keputusan

Pohon keputusan yang telah mengalami reduksi dari atribut-atribut yang dapat dihilangkan dapat terlihat pada Gambar 7.



Gambar 7. Pohon Keputusan Hasil Reduksi

Pohon keputusan pada Gambar 7 lebih baik dari pada pohon keputusan pada Gambar 5. Terlihat pada Gambar 7 hanya dengan menggunakan tiga buah atribut saja sudah dapat digunakan untuk mengidentifikasi jenis pesawat. Hal ini akan membantu pada saat proses yang dilakukan komputer dari sisi kecepatan proses dan penyimpanan. Dari sisi sistem pakar itu sendiri dengan adanya atribut yang direduksi akan mengurangi premis sehingga membantu pada sesi konsultasi.

Setelah melewati tahap pembuatan tabel keputusan dan pohon keputusan dihasilkan kaidah yang efektif. Kaidah yang dihasilkan dari pohon keputusan pada Gambar 7 adalah sebagai berikut:

Kaidah 1 : IF tipe mesin *prop*THEN tipe pesawat C130

Kaidah 2 : IF tipe mesin *jet* AND posisi sayap *low*

THEN tipe pesawat B747

Kaidah 3 : IF tipe mesin *jet* AND posisi sayap *high*

AND *bulges none* THEN tipe pesawat C5A

Kaidah 4 : IF tipe mesin *jet* AND posisi sayap *high*

AND *bulges aft wings*THEN tipe pesawat C141

d) Logika Predikat (*Predicate Logic*)

Logika yang pertama kali dikembangkan Aristoteles merupakan logika formal. Logika formal adalah logika yang berhubungan dengan bentuk (*syntax*) pernyataan bukan pada arti (*semantic*) dari pernyataan (Hartati & Iswanti, 2008).Kemudian pada tahun 1847 George Boole mengemukakan konsep logika simbolis yang mengenal aksioma yaitu symbol-simbol yang merepresentasikan obyek dan kelas serta operasi aljabar untuk memanipulasi simbol-simbol tersebut. Logika proporsional adalah logika simbolis yang memanipulasi proposisi. Logika proporsional akan menangani kalimat deklaratif namun hanya mampu menangani pernyataan yang komplit dan tidak bisa menganalisa struktur internal sebuah pernyataan. Maka dikembangkan logika predikat untuk menganalisa kasus yang lebih umum dan dapat menganalisis struktur internal kalimat.

Bentuk paling sederhana dari logika predikat adalah logika derajat pertama (*first order logic*) yang terbentuk dengan menambahkan fungsi atau analisis lain pada kalkulus predikat. Logika predikat berdasarkan pada kebenaran dan kaidah inferensi untuk merepresentasikan simbol-simbol dan hubungannya satu dengan yang lain. Logika predikat selain digunakan untuk menentukan kebenaran (*truthfulness*) atau kesalahan (*falsity*) sebuah pernyataan juga dapat digunakan untuk merepresentasikan pernyataan tentang obyek tertentu.

Contoh logika proporsional : Bujur sangkar mempunyai empat sisi. Kalimat tersebut merupakan logika proporsional karena mengandung pernyataan yang mempunyai nilai kebenaran. Contoh logika predikat : Semua segitiga adalah poligon. Logika predikat menganalisa struktur internal kalimat tersebut, ditunjukkan dengan penggunaan kata “semua” yang merupakan *quantifier*.

3. Teori Holland (RIASEC)

Teori Holland mengemukakan enam lingkungan okupasional dan enam tipe kepribadian. Pada tahun 1966, Holland berpendapat bahwa enam tipe kepribadiantersebut adalah Realistik, Intelektual, Artistik, Sosial, Pengusaha, dan Konvensional, demikian juga lingkungan okupasional diberi nama yang sama. Tingkatan orientasi kepribadian individu menentukan lingkungan yang dipilihnya, semakin jelas tingkatannya, maka makin efektif pencarian lingkungan yang sesuai (Manrihu, 1992, p. 71). Pengetahuan individu tentang diri dan lingkungannya diperlukan untuk menetapkan pilihan yang sesuai.

Teori Holland direvisi pada tahun 1973, tipe-tipe kepribadian dan lingkungan okupasional tersebut adalah Realistik, Investigatif, Artistik, Sosial, Pengusaha, dan Konvensional (Manrihu, 1992, p. 71). Pandangan Holland yang digunaanuntuk mengemukakan enam tipe tersebut berakar pada psikologi diferensial, terutama penelitian dan pengukuran terhadap minat, dan psikologi kepribadian yang mempelajari tipe kepribadian. Dua sumber pengaruh ini mendorong Holland untuk mengasumsikan bahwa orang yang memiliki minat dan bekerja dalam lingkungan yang berbeda-beda, sebenarnya adalah orang yang berkepribadian berbeda-beda dan mempunyai sejarah hidup yang berbeda pula.

Kepribadian seseorang menurut John Holland merupakan hasil dari keturunan dan pengaruh lingkungan (Osipow, 1983, p. 84). Secara garis besar pandangan Holland mencakup tiga ide dasar, yaitu :

a. Setiap orang dapat digolongkan tipe kepribadiannya menurut patokan sampai seberapa jauh mereka mendekati salah satu di antara enam tipe kepribadian. Enam tipe kepribadian tersebut adalah: Tipe Realistik (*The Realistic Type*), Tipe Peneliti/Pengusut (*The Investigative Type*), Tipe Seniman (*The Artistic Type*), Tipe Sosial (*The Social Type*), Tipe Pengusaha (*The Enterprising Type*), dan Tipe Orang Rutin (*Conventional Type*). Semakin mendekati atau mirip seseorang dengan salah satu di antara enam tipe tersebut, maka makin tampaklah ciri-ciri dan corak perilaku yang khas untuk tipe tersebut. Setiap tipe kepribadian adalah suatu tipe ideal atau tipe teoritisyang merupakan hasil dari interaksi antara faktor-faktor internal dan eksternal. Berdasarkan interaksi tersebut manusia belajar lebih menyukai kegiatan/aktivitas tertentu, yang kemudian melahirkan suatu minat kuat yang pada gilirannya menumbuhkan kemampuan dan keterampilan tertentu. Kombinasi dari minat dan kemampuan tersebut menciptakan suatu disposisi yang bersifat sangat pribadi untukbersikap, berpikir, bertindak,dan menafsirkan dengan cara-cara tertentu. Sebagai sebuah contoh : seseorang dengan tipe sosial yang notabennya lebih peka terhadap kebutuhan orang lain maka ia lebih cerderung memasuki lingkungan okupasi yang mengandung unsur pelayanan sosial seperti perawat, guru, pekerja sosial, dan pemuka agama. Membandingkan sikap dan cara bertindak seseorang dengan pola sikap dan kebiasaan bertindak yang khusus

untuk setiap tipe kepribadian dapat menentukan tipe manakah yang cocok dengan orang tersebut, dalam urutan mana yang sesuai , mana yang sesuai dalam urutan pertama, kedua, ketiga dan seterusnya. Dengan demikian, seseorang dapat dinyatakan paling mendekati tipe sosial, namun masih memiliki kemiripan dengan tipe pengusaha dan tipe seniman. Hal ini dapat dilanjutkan untuk mengetahui urutan tipe kepribadian selanjutnya dengan cara mengidentifikasikan kemiripan dengan tipe-tipe lain yang belum teridentifikasi. Profil total dari keseluruhan kemiripan dalam urutan pertama ke bawah, menampakkan pola kepribadian seseorang (*the individual's personality pattern*). Usaha untuk menentukan profil total tersebut dapat menggunakan berbagai metode seperti testing psikologis dan analisis sejarah hidup sehubungan dengan aspirasi okupasi.

b. Berbagai lingkungan yang di dalamnya terdapat orang hidup dan bekerja, dapat digolongkan menurut patokan sampai seberapa jauh suatu lingkungan tertentu mendekati salah satu model lingkungan (*a model environment*). Enam model lingkungan tersebutadalah: Lingkungan Realistik (*The Realistic Environment*), Lingkungan Penelitian/Pengusutan (*The Investigative Environment*), Lingkungan Kesenian (*The Artistic Environment*), Lingkungan Pengusaha (*The Enterprising Environment*), Lingkungan Pelayanan Sosial (*The Social Environment*), Lingkungan Bersuasana Kegiatan Rutin (*The Conventional Environment*). Semakin mirip atau mendekati lingkungan tertentu dengan salah satu di antara enam model lingkungan tersebut, maka makin tampaklah corak dan suasana kehidupan yang khas untuk lingkungan

tersebut. Masing-masing model lingkungan hiduptermasuk lingkungan okupasi, didominasi oleh orang yang bertipe kepribadian tertentu. Sebagai sebuah contoh, lingkungan kesenian didominasi oleh orang yang bertipe kepribadian seniman. Hal ini berarti kebanyakan orang yang hidup dan bekerja di lingkungan itu termasuk tipe kepribadian itu. Masing-masing model lingkungan hidup memberikan kesempatan tertentu dan menimbulkan tantangan tertentu pula. Mengingat keenam tipe kepribadian menunjukkan pola minat dan kompetensi tertentu, maka jika banyak orang dari tipe kepribadian tertentu berkumpul untuk hidup dan bekerja sama, mereka menciptakan suasana yang mencerminkan tipe kepribadian mereka dan menarik orang lain yang bertipe kepribadian sama untuk bergabung dengan mereka. Salah satu metode yang digunakan untuk meneliti lingkungan tertentu ialah menghitung jumlah orang dari berbagai tipe kepribadian yang hidup dan bekerja di lingkungan tersebut. Hasil perhitungan ini ditransformasi menjadi presentase. Presentase tertinggi dari tipe kepribadian tertentu menciptakan suasana yang khas pada lingkungan tersebut.

c. Perpaduan antara tipe kepribadian tertentu dan model lingkungan yang sesuaimenghasilkan keselarasan dan kecocokan okupasional (*occupational homogeneity*), sehingga seseorang dapat mengembangkan diri dengan optimal dalam lingkungan tersebut dan merasa puas. Perpaduan dan pencocokan antara tiap kepribadian dan tiap model lingkungan memungkinkan meramalkan pilihan okupasi, keberhasilan, stabilitas seseorang dalam okupasi yang didiami. Sebagai contoh, seseorang yang diketahui paling mendekati tipe

sosial akan lebih cenderung memasuki okupasi yang bertipe lingkungan sosial sepertipelayanan sosial karena okupasi tersebut diketahui paling sesuai dengan kepribadiannya sendiri dan paling memuaskan baginya.Sedangkan orang lain yang diketahui paling mendekati tipe kepribadian orang rutin akan lebih cenderung memilih okupasi dengan lingkungan yang bersuasana kegiatan rutin seperti pegawai di kantor, resepsionis, dan akuntan. Sebaliknya, orang yang memasuki lingkungan okupasi yang jauh dari tipe kepribadian yang paling dominan atau khas baginya akan mengalami konflik, tidak dapat mengembangkan diri secara optimal dan tidak akan merasa puas sehingga cenderung untuk meninggalkan lingkungan okupasi tersebut dan mencari lingkungan lain yang lebih cocok dengan tipe kepribadian domiannya.

Berdasarkan penjelasan diatas, dapat diambil kesimpulan bahwa terdapatempat asumsi yang merupakan jantung teori Holland, yaitu :

a. Kebanyakan orang dapat dikategorikan sebagai salah satu dari 6 tipe kepribadian, yaitu: Realistik, Investigatif, Artistik, Sosial, Giat (suka ber-usaha), dan Konvensional.
b. Ada enam jenis lingkungan : Realistik, Investigatif, Artistik, Sosial, Giat (suka berusaha), dan Konvensional.
c. Orang menyelidiki lingkungan-lingkungan yang akan memungkinkannya melatih keterampilan dan kemampuannya, mengekspresikan sikap dan nilainya, dan menerima masalah serta perananyang sesuai.
d. Perilaku seseorang ditentukan oleh interaksi antara kepribadiannya dan ciri-ciri lingkungannya.

Holland berkeyakinan bahwa suatu minat yang menyangkut pekerjaan dan okupasi adalah hasil perpaduan dari sejarah hidup seseorang dan keseluruhan kepribadiannya, sehingga minat tertentu akhirnya menjadi suatu ciri kepribadian yang berupa ekspresi diri dalam bidang pekerjaan, bidang studi akademik, hobi inti, berbagai kegiatan rekreatif dan banyak kesukaan yang lain (Winkel & Hastuti, 2005, pp. 636-637).

Pada halaman tiga buku *"Making Vocational Choices : A Theory of Vocational Personalities and Work Environments*" , Holland menuliskan : *"In short, what we have called vocational interests are an important aspect of personality"*, karena itu alat tes yang dikenal dengan nama *interest inventory* dipandang sebagai tes kepribadian (Winkel & Hastuti, 2005, p. 637).Salah satu indikasi dari minat adalah kesukaan pada kegiatan tertentu,sedangkan ketidaksukaan menjadi kontraindikasinya. Holland mengembangkan beberapa tes yang dapat membantu orang mengenal dirinya sendiri, seperti: *The Vocational Preference Inventory* di tahun 1977 dan *Self-directed Search* di tahun 1979.

Holland juga berefleksi tentang hubungan antara tipe-tipe kepribadian dan antara model-model lingkungan, yang dituangkan dalam bagan yang disebut Hexagonal Model. Model ini menggambarkan jarak psikologis antara tipe-tipe kepribadian dan model-model lingkungan, makin pendek jarak (menurut garis-garis dalam model) antara dua tipe kepribadian maka makin dekat kedua tipe itu dalam makna psikologisnya dan makin panjang jarak (menurut garis-garis dalam model) maka makin jauh kedua tipe itu dalam makna psikologisnya.

Korelasi suatu tipe dengan tipe lainnya, misalnya tipe realistik dekat dengan tipe investigatif di satu sisi dan dengan tipe konvensional di sisi lainnya (korelasinya 0,46 dan 0,36), sedangkan dengan tipe sosial korelasinya 0,21 (Osipow, 1983, p. 83). Contoh lainnya, tipe artistik dekat hubungannya dengan

tipe investigatif dan tipe sosial (korelasinya 0,34 dan 0,42), tetapi jauh sekali dari tipe konvensional yang korelasinya 0,11. Keadaan tersebut tidak dapat disesuaikan secara tepat pada hexagon jika dimasukkan dalam ukuran skala, hal ini lebih merupakan sekedar suatu percobaan dari Holland untuk mempertalikan antara yang satu dengan yang lain (Osipow, 1983, p. 90).

Perkembangan tipe kepribadian adalah hasil dari interaksi faktor bawaan dan lingkungan yang membawa kepada kecenderungan untuk jenis aktivitas khusus yang pada gilirannya mengarahkan individu kepada tipe perilaku atau kepribadian tertentu. Berikut penjelasan untuk setiap tipe dari ke-enam tipe kepribadian Holland:

a. Tipe Realistik memiliki preferensi pada aktivitas yang memerlukan manipulasi eksplisit, teratur, atau sistematik terhadap obyek, alat, mesin, dan binatang. Tidak menyukai aktivitas pemberian bantuan atau pendidikan. Preferensi tersebut menyebabkan pengembangan kompetensi dalam bekerja dengan benda, binatang, alat dan perlengkapan teknik, serta mengabaikan kompetensi sosial dan pendidikan. Menganggap diri baik dalam kemampuan mekanikal dan atletik tetapi tidak cakap dalam keterampilan sosial hubungan insani. Menilai tinggi benda-benda nyata, seperti : uang dan kekuasaan. Ciri-ciri khususnya adalah praktikalitas, stabilitas, konformitas. Cenderung lebih menyukai keterampilan dan okupasi teknik.
b. Tipe Investigatif memiliki preferensi untuk aktivitas-aktivitas yang memerlukan penyelidikan observasional, simbolik, sistematik, dan kreatif terhadap fenomena fisik, biologis, dan kultural agar dapat memahami dan

mengontrol fenomena tersebut, dan tidak menyukai aktivitas-aktivitas persuasif, sosial, dan repetitif. Contoh okupasi yang memenuhi kebutuhan tipe ini adalah ahli kimia dan ahli fisika.

c. Tipe Artistik lebih menyukai aktivitas-aktivitas yang ambiguous, bebas, dan tidak tersistematisasi untuk menciptakan produk-produk artistik seperti lukisan, drama, dan karangan. Tidak menyukai aktivitas-aktivitas yang sistematik, teratur, dan rutin. Kompetensi dalam upaya artistik dikembangkan sedangkan keterampilan yang rutin, sistematik, klerikal diabaikan. Memandang diri sebagai ekspresif, murni, independen, dan memiliki kemampuan-kemampuan artistik. Beberapa ciri khususnya adalah emosional, imaginatif, impulsif, dan murni. Okupasi artistik biasanya adalah lukisan, karangan, akting, dan seni pahat.

d. Tipe Sosial lebih menyukai aktivitasyang melibatkan orang lain dengan penekanan pada membantu, mengajar, atau menyediakan bantuan. Tidak menyukai aktivitas-aktivitas rutin dan sistematik yang melibatkan obyek-obyek dan materi-materi. Kompetensi sosial cenderung dikembangkan, sedangkan hal-hal yang bersifat manual & teknik diabaikan. Menganggap diri kompeten dalam membantu dan mengajar orang lain serta menilai tinggi aktivitas hubungan sosial. Beberapa ciri khususnya adalah kerja sama, bersahabat, persuasif, dan bijaksana. Okupasi sosial mencakup pekerjaan-pekerjaan seperti mengajar, konseling, dan pekerjaan kesejahteraan sosial.

e. Tipe Enterprising lebih menyukai aktivitas yang melibatkan manipulasi terhadap orang lain untuk perolehan ekonomik atau tujuan-tujuan organisasi.

Tidak menyukai aktivitas yang sistematik, abstrak, dan ilmiah. Kompetensikepemimpinan, persuasif dan yang bersifat supervisi dikembangkan, sedangkan kompetensi yang ilmiah diabaikan. Memandang diri sebagai orang yang agresif, populer, percaya diri, dan memiliki kemampuan memimpin. Keberhasilan politik dan ekonomik dinilai tinggi. Ciri khasnya adalah ambisi, dominasi, optimisme, dan sosiabilitas.

f. Tipe Konvensional lebih menyukai aktivitas-aktivitas yang memerlukan manipulasi data yang eksplisit, teratur, dan sistematik guna memberikan kontribusi kepada tujuan-tujuan organisasi. Tidak menyukai aktivitas-aktivitas yang tidak pasti, bebas dan tidak sistematik. Kompetensi-kompetensi dikembangkan dalam bidang-bidang klerikal, komputasional, dan sistem usaha. Aktivitas-aktivitas artistik dan semacamnya diabaikan. Memandang diri sebagai teratur, mudah menyesuaikan diri, dan memiliki keterampilan-keterampilan klerikal dan numerikal. Beberapa ciri khasnya adalah efisiensi, keteraturan, praktikalitas, dan kontrol diri. Okupasi-okupasi yang sesuai adalah bankir, penaksir harga, ahli pajak, dan pemegang buku.

Terdapat tiga asumsi tambahan tentang orang dan lingkungan yang dikemukakan Holland, asumsi tersebut adalah sebagai berikut:

a. Konsistensi, pada diri seseorang atau lingkungan beberapa pasangan tipe lebih dekat hubungannya daripada yang lainnya. Misalnya, tipe realistik dan investigatif lebih banyak persamaannya daripada tipe konvensional dan artistik. Konsistensi adalah tingkat hubungan antara tipe kepribadian atau antara model lingkungan. Taraf konsistensi atau keterhubungan diasumsikan

mempengaruhi preferensi vokasional. Misalnya, orang yang paling menyerupai tipe realistik dan paling menyerupai berikutnya dengan tipe investigatif (realistik-investigatif) seharusnya lebih dapat diramalkan daripada orang realistik-sosial.

b. Diferensiasi, beberapa orang atau lingkungan lebih dibatasi secara jelas daripada yang lainnya. Misalnya, seseorang mungkin sangat menyerupai suatu tipe dan menunjukkan sedikit kesamaan dengan tipelainnya, atau suatu lingkungan mungkin sebagian besar didominasi oleh suatu tipe tunggal. Sebaliknya, orang yang menyerupai banyak tipe atau suatu lingkungan yang bercirikan kira-kira sama dengan keenam tipe tersebut tidak terdiferensiasi atau kurang terdefinisikan. Taraf di mana seseorang atau suatu lingkungan terdefinisikan dengan baik adalah taraf diferensiasinya.

c. Kongruensi, tipe kepribadian tertentu memerlukan lingkungan tertentu. Misalnya tipe realistik tumbuh dengan baik dalam lingkungan realistik karena dapat memberikan kesempatan dan menghargai kebutuhan tipe realistik. Ketidakharmonisan (*incongruence*) terjadi bila suatu tipe hidup dalam suatu lingkungan yang menyediakan kesempatan dan penghargaan yang asing bagi preferensi atau kemampuan orang itu. Misalnya tipe realistik di lingkungan sosial.

Pada tahun 1978, Holland mengembangkan suatu Sistem Klasifikasi Okupasi (*The Classification System*) yang menggolongkan 500 okupasi dalam enam kategori okupasi, yaitu: *RealisticOccupations*, *Investigative Occupations*, *Artistic Occupation*, *Social Occupations*, *Entreprising Occupations*,

dan*Conventional Occupations*. Klasifikasi ini terdapat dalam *The Occupations Finder* yang juga mencantumkan nomor-nomor kode dari *Dictionary of Occupational Titles* dan tingkat pendidikan sekolah yang umumnya dituntut supaya mampu memegang okupasi tertentu (Winkel & Hastuti, 2005, p. 637). Sistem Klasifikasi Okupasi tersebut diterapkan dalam *The Self-directed Search* yang dirancang untuk membantu orang agar lebih mengenal dirinya sendiri dan menemukan bidang okupasi yang cocok dengan tipe kepribadiannya atau paling sedikit untuk dipertimbangkan.

Teori Holland oleh banyak pakar psikologi vokasional dinilai sebagai teori yang komprehensif karena meninjau pilihan okupasi sebagai bagian dari keseluruhan pola hidup seseorang. Teori Holland dianggap sebagai teori yang mendapat banyak dukungan dari hasil penelitian yang berhubungan atau menyangkut tentang model-model lingkungan serta tipe-tipe kepribadian.

Teori Holland pada penelitian ini digunakan sebagai dasar pengetahuan untuk kemudian disajikan dalam bentuk*rule base*. Tipe kepribadian menurut Holland ada 6, yaitu : Tipe Realistik (*The Realistic Type*), Tipe Peneliti/Pengusut (*The Investigative Type*), Tipe Seniman (*The Artistic Type*), Tipe Sosial (*The Social Type*), Tipe Pengusaha (*The Enterprising Type*), dan Tipe Orang Rutin (*Conventional Type*). Setiap tipe kepribadian dibandingkan satu persatu sesuai dengan korelasinya di Hexagonal Model. Setelah ditemukan 3 tipe kepribadian dominan maka dapat dicocokkan dengan jurusan kuliah yang sesuai dengan gabungan 3 tipe kepribadian paling dominan tersebut.

4. Microsoft Visual Basic

Microsoft Visual Basic atau sering disingkat sebagai VB saja, merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows. Visual Basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat.Beberapa bahasa skrip seperti *Visual Basic for Applications* (VBA) dan *Visual Basic Scripting Edition* (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda.*Programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic. Program yang ditulis dengan Visual Basic juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi luar tambahan.

Visual Basic 1.0 dikenalkan pada tahun 1991. Konsep pemrograman dengan metode *drag-and-drop* untuk membuat tampilan aplikasi Visual Basic ini diadaptasi dari *prototype generator form*bernama Tripod yang dikembangkan oleh Alan Cooper dan perusahaannya. Microsoft kemudian mengontrak Cooper dan perusahaannya untuk mengembangkan Tripod menjadi sistem *form* yang dapat diprogram untuk Windows 3.0, di bawah kode nama Ruby.Tripod tidak memiliki bahasa pemrograman sama sekali.Hal ini menyebabkan Microsoft memutuskan untuk mengkombinasikan Ruby dengan bahasa pemrograman Basic untuk membuat Visual Basic.

Visual basic dari waktu ke waktu :

a. Proyek Thunder dimulai.

b. Visual Basic 1.0 dirilis untuk Windows pada Comdex/Windows Trade Show di Atlanta, Georgia pada Mei 1991. Visual Basic 1.0 untuk DOS dirilis pada September 1992. Bahasa pemrogramannya tidak terlalu kompatibel dengan Visual Basic untuk Windows, karena sesungguhnya itu adalah versi selanjutnya dari kompiler BASIC berbasis DOS yang dikembangkan oleh Microsoft, yaitu QuickBASIC. Antarmuka program menggunakan teks, dan memanfaatkan kode ASCII tambahan untuk mensimulasikan tampilan GUI.
c. Visual Basic 2.0 dirilis pada November 1992. Lingkungan pemrogramannya lebih mudah untuk digunakan, dan kecepatannya lebih ditingkatkan.
d. Visual Basic 3.0 rilistahun 1993 dan hadir dalam dua versi, yaitu Standar dan Professional. VB3 juga menyertakan versi 1.1 dari Microsoft Jet Database Engine yang dapat membaca dan menulis database Jet/Access 1.x.
e. Visual Basic 4.0 (Agustus 1995) adalah versi pertama yang dapat membuat program 32-bit seperti program 16-bit. VB4 juga memperkenalkan kemampuannya dalam membuat aplikasi non-GUI. Sejak VB4 dirilis Visual Basic tidak menggunakan control VBX lagi melainkan kontrol OLE (dengan ekstensi file *.OCX), yang lebih dikenal kemudian dengan kontrol ActiveX.
f. Visual Basic 5.0 (Februari 1997), rilis Visual Basic eksklusif untuk versi 32-bit dari Windows. Para programmer dapat meng-impor program VB4 ke versi VB5, dan program VB5 dapat dikonversi ke dalam format VB4.
g. Visual Basic 6.0 (pertengahan 1998) telah diimprovisasi di beberapa bagian, termasuk kemampuan membuat aplikasi web. Meskipun kini VB6 sudah tidak didukung lagi, tetapi *file runtime*-nya masih didukung hingga Windows 7.

Visual basic 6.0 mempunyai 3 edisi yang menunjukkan fasilitas dan kemampuannya masing-masing. Berikut 3 edisi Visual Basic 6.0:

a. *Learning Edition*

Memungkinkan *programmer* untuk dengan mudah membuat aplikasi yang kuat untuk Microsoft Windows dan Windows NT ®. Ini mencakup semua kontrol intrinsik, ditambah *grid*, *tab*, dan kontrol *data-bound*. Dilengkapi dokumentasi Learn VB Now (judul CD-ROM multimedia), ditambah CD *Microsoft Developer Network* yang mengandung dokumentasi *online* penuh.

b. *Profesional Edition*

Menyediakan seluruh fitur *learning edition* ditambah dengan fiturActiveX controls, The Internet Information Server Application Designer, Integrated Data Tools and Data Environment, dan The Dynamic HTML Page Designer. Dilengkapi dokumentasi melalui buku *Visual Studio Professional Features book* dan CD *Microsoft Developer Network*yang mendukung dokumentasi *online* secara penuh.

c. *Enterprise Edition*

Visual Basic Enterprise edition berisi semua fitur yang terdapat pada *learning edition* dan *professional edition*. Selain itu, terdapat pula fitur tambahan yang hanya terdapat pada VB *Enterprise Edition*. Fitur tersebut adalah :

1) *Remote Data Control* (RDC)

RDC mengimplementasikan akses data dengan menggunakan sistem RDO, *Microsoft Remote Data Objects*.

2) SQL *Debugging*

T-SQL *debugger* memungkinkan untuk secara interaktif *debug* prosedur yang tersimpan secara terpencil ditulis dalam *Microsoft SQL server* dengan *dialog Transact SQL*, dari dalam lingkungan pengembangan Visual Basic. *Toolkit Microsoft SQL Server Programmer* mencakup Transact-SQL, pemrograman ODBC untuk SQL, dan banyak lagi.

3) Stored Procedure Editor

Editor SQL memungkinkan untuk membuat dan mengubah prosedur dan dapat menyimpan pada keduanya, yaitu SQL Server dan Oracle dari dalam lingkungan *Visual Basic Developer*.

4) Visual Database Integration

Visual *database* desainer memungkinkan secara visual untuk membuat dan memodifikasi skema *database* dan *query*, membuat SQL Server dan tabel *database* Oracle, *drag* dan *drop* untuk membuat pandangan, dan secara otomatis mengubah tipe.

Microsoft visual basic memiliki berbagai kelebihan. Berikut adalah beberapa keunggulan Microsoft Visual Basic 6.0:

a. Kurva pembelajaran dan pengembangan yang lebih singkat dibandingkan bahasa pemrograman yang lain seperti C/C++, Delphi atau bahkan PowerBuilder sekalipun.
b. Menghilangkan kompleksitas pemanggilan fungsi windows API, karena banyak fungsi-fungsi tersebut sudah di "*embedded*" kedalam *syntax* visual basic.

c. Cocok digunakan untuk mengembangkan aplikasi atau program yang bersifat *Rapid Application Development*.

d. Sangat cocok digunakan untuk membuat program atau aplikasi bisnis.

e. Digunakan oleh hampir semua Microsoft Office sebagai bahasa *macro* dan segera akan diikuti oleh yang lainnya.

f. Dapat membuat *ActiveX Control*, yaitu *project* yang digunakan untuk membuat komponen kontrol *ActiveX* sehingga dapat menyisipkan komponen pada sebuah program aplikasi.

g. Dapat menggunakan OCX atau komponen yang disediakan oleh pihak ketiga (*Third Party*) sebagai *tools* pengembang.

h. Menyediakan *wizard* yang sangat berguna untuk mempersingkat atau mempermudah pengembangan aplikasi.*Project* VB *Aplication Wizard* merupakan *project* yang digunakan untuk membuat sebuah kerangka suatu program atau aplikasi.

i. Mendekati *Object Oriented Programming*.

j. Dapat di-integrasikan dengan internet, baik itu pada sisi *Client* maupun pada sisi *Server*.

k. Dapat membuat *ActiveX Automation Server*.

l. Integrasi dengan *Microsoft Transaction Server*.

m. Dapat menjalankan *server* tersebut dari mesin yang sama atau bahkan dari mesin atau komputer yang lain.

n. Memiliki *compiler* yang dapat menghasilkan *output file executable* (exe).

o. Membuat *flat form* pembuatan pogram yang diberi nama *developer studio*.

5. MySQL

MySQL adalah sebuah program pembuat *database* yang bersifat *open source* sehingga dapat dijalankan pada semua *platform* baik windows maupun linux . MySQL merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *multi user*. Kelebihan lain dari MySQL adalah MySQL menggunakan bahasa *query* standar yang dimiliki SQL.

MySQL tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (*interface*). MySQL dapat didukung oleh hampir semua program aplikasi baik yang *open source* maupun yang tidak. Program aplikasi pendukung MySQL antara lain PHP, Visual Delphi, Visual Basic dan Cold Fusion. MySQL memiliki beberapa tipe data, berikut tipe data dalam MySQL yang sering dipakai:

Tabel 3.Tipe Data MySQL

Tipe data	Keterangan
INT(M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT(M,D)	Angka pecahan
DATE	Tanggal, Format: YYYY-MM-DD
DATETIME	Tanggal dan Waktu, Format: YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1-255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LONGBLOB	Teks dengan panjang maksimum 4294967295 karakter

6. Basis Data

a. Pengertian Basis Data

Basis data adalah penggunaan bersama dari data yang terhubung secara logis dan deskripsi dari data, yang dirancang untuk keperluan informasi(Cannolly & Carolyn, 2005).Penyusunan suatu basis data digunakan untuk mengatasi masalah-masalah yang timbul pada penyusunan data. Masalah-masalah tersebut adalah sebagai berikut :

1) *Redudancy* data. *Redudancy* data adalah munculnya data-data secara berulang-ulang pada *file* basis data yang seharusnya tidak diperlukan.
2) *Inconsistency* data. Terjadi karena kesalahan dalam pemasukan data atau proses *update* data yang mengakibatkan munculnya data tidak konsisten.
3) *Isolation* data. Program aplikasi tidak mampu mengakses *file* tertentu dalam basis data karena pemakaian beberapa *file*, perlu mengubah agar seolah-olah ada *file* yang terpisah (terisolasi) dengan *file*lain.
4) Pengaksesan data. Data harus siap diakses oleh siapa saja yang mempunyai hak akses. perlu dibuat program untuk penolakan dan penerimaan hak akses data.Salah satu cara untuk mengakses data adalah DBMS.
5) Masalah keamanan. Hanya pengguna tertentu yang diperbolehkan untuk mengakses data tertentu, misalnya data gaji pegawai hanya boleh diakses oleh bagian keuangan dan personalia.
6) Masalah integritas (*integrity)*. Basis data berisi *file* yang saling berhubungan, kaitan antar *file* tersebut secara teknis dapat dilakukan dengan adanya sebuah *field* kunci yang mengaitkan kedua *file* tersebut.

7) *Multiple user*. Salah satu alasan basis data dibangun adalah untuk penggunaan data oleh banyak orang baik dalam waktu bersamaan atau tidak.
8) *Independent*data. Perubahan apapun yang terjadi dalam basis data, semua perintah harus stabil tanpa ada yang diubah.

Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya, penjelasan ini disebut skema yang menggambarkan objek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk memodelkan struktur basis data, ini dikenal sebagai model basis data atau model data.Model yang umum digunakan sekarang adalah model rasional, informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Jenis basis data, antara lain basis data hierarkis, basis data jaringan, dan basis data relasional.

b. Terminologi Basis Data

Pada terminologi basis data terdapat istilah tabel, baris, dan kolom. Tabel menyatakan bentuk berdimensi dua yang mewakili suatu kelompok data sejenis. Tabel berisi sejumlah kolom (*field*) dan baris (*record*). Menurut konsep basis data relasional, setiap tabel memiliki kunci primer (*primary key*), walaupun dalam praktiknya bisa saja tidak memiliki.Kunci primer (*primary key)* adalah suatu nilai yang bersifat unik sehingga dapat digunakan sebagaipembeda antar baris. Selain *primary key*, kunci dibagi menjadi 4, yaitu:

1) *Candidate Key*. *Candidate Key* adalah satu atau lebih atribut yang mendefinisikan sebuah baris secara unik yang berfungsi sebagai calon *primary key* dan mempunyai nilai unik pada hampir setiap barisnya.

2) *Alternate Key***.** *Alternate Key* adalah *candidat key* yang tidak dipilih untuk mendefinisikan sebuah baris secara unik, tetapi perlu dicatat. Tidak menutupkemungkinan menjadi *primary key* di tabel lain.

3) *Foreign Key***.** *Foreign Key* adalah sebuah kolom yang menunjuk ke kunci primer (*primary key*) milik tabel lain.

4) *Composite Key. Composite Key* adalah kunci primer yang tidak terbentuk oleh sebuah kolom, melainkan tersusun atas beberapa kolom.

c. *Data Manipulation Language* (DML)

DML terdiri dari *select, insert, update*, dan *delete*. Dengan kata lain DML digunakan untuk menampilkan, menambah, mengubah, dan menghapus baris (*record*) dalam tabel.

1) Menampilkan Data (*Select*)

Select field1,field2,…,fieldn *from*nama_tabel*where* [kondisi] *orderby*nama_field *asc*|*desc limit* [batasan];

2) Memasukkan Data (*Insert*)

*Insert into*nama_tabel*values* (isi_field1, isi_field2,…,isi_fieldn);

3) Mengubah Data (*Update*)

*Update*nama_tabel set nama_field1 = isi_baru, nama_field2 = isi_baru *where* [kondisi];

4) Menghapus Data (*Delete*)

*Delete from*nama_tabel *where* [kondisi];

Keterangan:[kondisi] : nama_field = “isi baris/*record*”

[batasan] : n (jumlah *record* yang akan ditampilkan)

d. *Entity Relationship Diagram* (ERD)

Entity relationship diagram adalah model data untuk menggambarkan hubungan antara satu entitas dengan entitas lain yang mempunyai hubungan (relasi) dengan batasan-batasan tertentu. Hubungan tersebut akan menyangkut dua komponen yang menyatakan jalinan ikatan yang terjadi, yaitu derajat hubungan dan partisipasi hubungan.Derajat hubungan menyatakan jumlah anggota entitas yang terlibat didalam ikatan yang terjadi dalam membentuk hubungan.Derajat hubungan pada *relationship type*memiliki 3 jenis yaitu (Waljiyanto, 2003):

1) 1:1 (*One to one relationship*)

Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu. Derajat hubungan antar *entity* 1:1 terjadi apabila tiap anggota *entity* A berpasangan dengan salah satu anggota dari *entity* B, begitu pula sebaliknya setiap anggota dari *entity* B hanya boleh berpasangan dengan satu anggota *entity* A.

2) 1:M (*One to many relationship*)

Hubungan antar *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik menjadi banyak berbanding satu. Derajat hubungan ini terjadi bila setiap anggota *entity* A boleh berpasangan dengan lebih dari satu anggota B. sebaliknya tiap anggota *entity* B hanya boleh berpasangan dengan satu anggota *entity* A.

3) M:M (*Many to many relationship*)

Hubungan antar satu atribut dengan atribut yang lain dalam satu *file* yang sama mempunyai hubungan banyak lawan banyak. Hal ini terjadi bila tiap anggota *entity* A boleh berpasangan dengan lebih dari satu anggota B dan sebaliknya.

Tabel 4. Komponen *Entity Relationship Diagram* (ERD)

Komponen	**Notasi**	**Fungsi**
Entity		Menyatakan himpunan entitas
Option Symbol		Menyatakan atribut (atribut yang berfungsi sebagai *key*namanyadigaris bawahi)
Relationship		Menyatakan himpunan relasi
Relationship Line		Sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya
Composite entity		Menyatakan himpunan entitas yang terbentuk dari relasi banyak-kebanyak.

7. *Waterfall Process Model*

*Software Development Life Cycle*merupakan proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem-sistem perangkat lunak(Rosa, 2011, p. 24). Model proses perangkat lunak atau biasa disebut juga dengan paradigma rekayasa perangkat lunak adalah suatu strategi pengembangan yang memadukan lapisan proses, metode, dan alat serta tahap-tahap generik. Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat proyek dan perangkat lunak, metode dan alat yang digunakan, serta pengendalian dan hasil yang diinginkan. *Software Development Life Cycle* sendiri memiliki beberapa model dalam penerapannya,antara lain : *Waterfall's Models, Prototyping Model, Rapid Application Development Model, Incremental Model,* dan*Spiral Model.*

W*aterfall's model* merupakan salah satu model pengembangan perangkat lunak klasik. Model ini juga sering disebut dengan model sekuensial linier dikarenakan proses pengembangannya yang dilakukan secara bertahap. W*aterfall's model* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau berurutan dimulai dari tahap analisis, tahap desain, tahap pengkodean dan tahap pengujian. Rosa dan Salahuddin menjelaskan bahwa kelebihan dari *waterfall's model*adalah merupakan model proses yang paling sederhana dan cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-ubah (Rosa, 2011, p. 29).

8. *Software Testing*

Tahap akhir dari proses pengembangan perangkat lunak adalah tahap pengujian. Pengujian perangkat lunak menurut Hetzel merupakan aktivitas-aktivitas yang bertujuan untuk mengevaluasi atribut-atribut atau kemampuan sebuah program atau sistem dan penentuan apakah sesuai dengan hasil yang diharapkan(Hetzel, 1988, p. 88). Sedangkan menurut Myers,*testing* merupakan proses eksekusi sebuah program atau sistem dengan maksud menemukan kesalahan(Myers, 1979).

Menurut Pressman pengujian perangkat lunak merupakan salah satu elemen dari rekayasa perangkat lunak yang sering disebut dengan *verification and validation testing*(V&V)(2001, hal. 479). Menurut pressman verifikasi mengacu pada serangkaian kegiatan yang memastikan perangkat lunak dapat melakukan fungsi-fungsi tertentu yang telah ditentukan, sedangkan validasi mengacu pada suatu set yang memastikan perangkat lunak sudah sesuai dengan kebutuhan pengguna(Pressman, 2001).

Tujuan dari pengujian perangkat lunak itu sendiri adalah untuk meningkatkan kualitas perangkat lunak, dengan cara melakukan uji verifikasi dan validasi perangkat lunak yang dikembangkan serta menguji reliabilitas perangkat lunak yang dikembangkan. *Verification testing*dan*validationtesting*(V&V) merupakan satu bagian dari set yang disebut dengan *Software Quality Assurance*

(SQA). Berikut penjelasan lebih lanjut mengenai *verification testing* dan *validation testing*(V&V):

a. *Verification Testin*g

Verification : "*Are we building the product right?*"(Boehm, 1981, hal. 81). Sesuai pernyataan diatas, pada tahapan uji verifikasi ini ditanyakan apakah kita telah membuat produk secara benar?. Hal ini memiliki arti bahwa tahap uji verifikasi dilakukan untuk memastikan apakah pengembang telah mengembangkan perangkat lunak dengan spesifikasi dan cara yang benar. Pada tahapan *verification testing* ini terdapat dua metode *testing* yaitu *black box test* dan *white box test*.

Pengujian *white box* menurut Irena merupakan pengujian untuk melakukan verifikasi program secara terstruktur, program di lakukan uji untuk menentukan benar tidaknya program secara *logic*, seperti *basis path testing, loop testing,* dan *structure control.*(Irena, 2009). Pengujian ini memfokuskan pada *test case* pada kondisi dan perulangan program untuk mendapatkan *logic* program secara benar. Dengan kata lain uji *white box testing* merupakan proses pengujian perangkat lunak dari sisi desain dan kode program.

Pengujian *black box* menurut Irena merupakan pengujian berdasarkan "*output requirement*" tanpa pengetahuan struktur internal program perangkat lunak (Irena, 2009). Pengujian ini digunakan untuk menguji semua fungsionalitas perangkat lunak yang dikembangkan. Pengujian *black box* dilakukan oleh pengguna akhir yang nantinya akan menggunakan perangkat lunak tersebut.

Pengembang tidak termasuk dalam pengguna yang akan menguji perangkat lunak pada pengujian *black box*.

b. *Validation Testing*

Validation : "*Are we building the right product?*"(Boehm, 1981, hal. 81). Sesuai pernyataan diatas, pada tahapan uji validasi ini ditanyakan apakah kita sebagai pengembang telah membuat produk yang benar sesuai dengan kebutuhan pengguna?. *Validationtesting* merupakan pengujian yang dilakukan untuk menguji apakah perangkat lunak yang dikembangkan dapat melakukan apa yang benar-benar dibutuhkan oleh pengguna secara fungsionalitas. *Validationtesting*akan memastikan apakah perangkat lunak yang dkembangkan telah sesuai dengan kebutuhan pengguna. Pada tahapan ini terdapat dua metode *testing* yaitu *alpha test* dan *beta test*.

Alpha test merupakan proses pengujian validasi perangkat lunak dengan perspektif pengembang perangkat lunak. Semua pengujian *alpha* dilakukan oleh pengembang atau ahli rekayasa perangkat lunak pada lingkungan uji pengembang yang memadai. Sedangkan *beta test* merupakan pengujian validasi perangkat lunak oleh beberapa pengguna yang dipilih sesuai dengan karakteristik pengguna perangkat lunak nantinya. Pada pengujian *beta* ini pengembang tidak terlibat dalam rangkaian *testing*.

Hasil dari uji validasi dilaporkan secara berkala kepada pengembang perangkat lunak untuk mengetahui permasalahan yang dilakukan selama proses pengujian. Hasil dari pengujian nantinya akan digunakan untuk menentukan

kualitas perangkat lunak serta persiapan sebelum perangkat lunak akan dirilis secara keseluruhan.

9. Kualitas Perangkat Lunak (*Software Quality*)

Agarwal, Tayal dan Gupta(2010)menjelaskan bahwa kualitas perangkat lunak merupakan kesesuaian terhadap persyaratan fungsional dan kinerja secara eksplisit, standar pengembangan yang terdokumentasi secara eksplisit, dan karakteristik implisit yang diharapkan dari perangkat lunak yang dikembangkan secara profesional.

Teori-teori tentang kualitas perangkat lunak telah dikemukakan oleh beberapa ahli. Teori-teori tersebut antara lain :

a. McCall Quality Factors

McCall, Richards, dan Walter merumuskan serangkaian faktor yang menunjukan kualitas perangkat lunak. Faktor-faktor kualitas tersebut terkategorisasi menjadi tiga aspek penting dari sebuah perangkat lunak yaitu : karakteristik operasional, kemampuan untuk dalam menangani perubahan, dan kemampuan beradaptasi dengan lingkungan baru(Pressman, 2001, hal. 509).

Gambar 8. Diagram Kategorisasi Faktor Kualitas Perangkat Lunak

Faktor-faktor kualitas yang menunjukan kualitas aplikasi tersebut adalah :

1) *Correctness*:berkaitan dengan bagaimana program mampu memenuhi spesifikasi dan tujuan yang ingin dicapai oleh pengguna.
2) *Reliability* : berkaitan dengan bagaimana sebuah program mampu beroperasi dalam sebuah kondisi yang menuntut presisi tertentu.
3) *Usability* : berkaitan dengan usaha yang diperlukan pengguna untuk mengoperasikan, menyiapkan *input*, dan menginterpretasikan *output* dari program.
4) *Integerity* : berkaitan dengan tingkat kontrol terhadap program oleh pengguna, baik yang mendapatkan otorisasi atau tidak.
5) *Efficiency* : berkaitan dengan jumlah sumber daya komputer yang digunakan serta kode yang diperlukan di dalam program untuk menjalankan setiap fungsinya.
6) *Maintainability* : berkaitan dengan usaha yang diperlukan untuk menemukan dan mengatasi kesalahan di dalam program.

7) *Flexibility* : berkaitan dengan usaha yang diperlukan untuk mengubah program yang beroperasi.
8) *Testability* : berkaitan dengan usaha yang diperlukan untuk menguji sebuah program untuk memastikan bahwa program tersebut berfungsi sebagaimana mestinya.
9) *Portability* :berkaitan dengan usaha yang diperlukan untuk dapat mentransfer sebuah program dari sebuah lingkungan perangkat keras atau lunak tertentu ke lingkungan yang lain.
10) *Reusability* : berkaitan dengan bagaimana sebuah bagian program dapat digunakan kembali di dalam program lain.
11) *Interoperability* : berkaitan dengan usaha yang diperlukan untuk menghubungkan sebuah sistem dengan sistem yang lain.

b. ISO 9126 Quality Factors

International StandardOrganization (ISO) atau organisasi internasional untuk standarisasi telah mendefinisikan satu set standar yang berkaitan dengan perangkat lunak. Satu set standar tersebut merupakan ISO 9126. Set standar ISO 9126 sendiri pertama kali diperkenalkan pada tahun 1991 melalui pertanyaan tentang definisi kualitas perangkat lunak. Saat ini ISO 9126 merupakan salah satu set standar yang digunakan secara luas oleh banyak orang. ISO 9126 banyak digunakan secara luas karena mencakup model kualitas dan metrik.Set standar ISO 9126 sendiri mengidentifikasi karakteristik kualitas perangkat lunak menjadi enam bagian.

International StandardOrganization(ISO) mengembangkan Standar ISO 9126 yang mengidentifikasikan enam faktor kualitas yang menentukan kualitas suatu perangkat lunak(Pressman, 2001, hal. 513). Faktor-faktor kualitas tersebut adalah sebagai berikut :

1) *Functionality* : Kemampuan menutupi fungsi produk perangkat lunak yang menyediakan kepuasan kebutuhan *user*. Faktor ini dapat ditunjukan oleh beberapa sub faktor yaitu : *suitability, accuracy, interoperability, security* , dan *functionality compliance*.

2) *Reliability* : Kemampuan perangkat lunak untuk perawatan dengan level performansi. Faktor ini dapat ditunjukan oleh beberapa sub faktor yaitu : *maturity, fault tolerance, recoverability,* dan *reliability compliance.*

3) *Usability* : Kemampuan yang berhubungan dengan penggunaan perangkat lunak. Faktor ini dapat ditunjukan oleh beberapa sub faktor yaitu : *understandability, learnability, operability, attractiveness,* dan *usabilitycompliance*.

4) *Efficiency* : Kemampuan software memanfaatkan secara optimal *resource* yang digunakan, Faktor ini ditunjukan oleh beberapa sub faktor yaitu : *time behavior, resource utilization,* dan *efficiencycompliance.*

5) *Maintainability* :Kemudahan suatu perangkat lunak untuk diperbaiki dikemudian hari. Faktor ini ditunjukan oleh beberapa sub faktor antara lain : *analyzability, changeability, stability, testability,* dan *maintainability compliance*.

6) *Portability* :Kemampuan yang berhubungan dengan kemampuan perangkat lunak yang dikirim ke lingkungan berbeda. Faktor ini dapat ditunjukan oleh beberapa sub faktor yaitu : *adaptability, installability, conformance, replaceability,* dan *portabilitycompliance*.

Faktor kualitas perangkat lunak yang dianalisis pada penelitian ini adalah faktor c*orrectness, functionality, usability,*dan *maintainability.* Faktor tersebut dipilih dengan pertimbangan bahwa perangkat lunak harus memiliki kualitas *source code* yang baik, semua fungsi yang dibutuhkan pengguna berjalan dengan baik, mudah digunakan, dan mudah dipelajari untuk pemeliharaan sistem.

10. Faktor Kualitas *Correctness*

Pressman (2001, hal. 509) menjelaskan bahwa *correctness* merupakan faktor kualitas yang menunjukkan tingkat bagaimana perangkat lunak menjalankan fungsi yang dibutuhkannya. Faktor kualitas *correctness* dapat diukur dengan analisis *defect per* KLOC (cacat / *error* pada setiap KLOC/*Kilo Line of Code*).

McConnell (2004, p. 689) dalam bukunya menjelaskan bahwa jumlah *error* yang terjadi dalam pengembangan perangkat lunak, terutama yang kaitannya dengan penulisan kode, dapat diperkirakan berdasarkan besar kecilnya *project* perangkat lunak yang sedang dikembangkan. Rentang kemukinan *error* yang tejadi dalam suatu *project* digambarkan dalam tabel berikut:

Tabel 5.Perkiraan Jumlah *Error* McConnell

Ukuran Project (*Line of Code*/LOC)	**Perkiraan Jumlah *Error***
<2K	0 - 25 *error* / KLOC
2K – 16K	0 - 40 *error* / KLOC
16K – 64K	0,5 - 50 *error* / KLOC

64K – 512K	2 - 70 *error* / KLOC
>512K	4 - 100 *error* / KLOC

Selain perkiraan rentang jumlah *error* yang dikemukakan McConnell untuk *project* perangkat lunak yang sedang dikembangkan, Pressman dalam bukunya (2001, hal. 133) menjelaskan beberapa metode lain yang dapat digunakan untuk menentukan perkiraan jumlah *error* dalam sebuah *project* perangkat lunak yang sedang dikembangkan. Metode tersebut beserta rumusnya dapat dilihat pada tabel di bawah ini:

Tabel 6. Metode Perkiraan *Error* Tiap KLOC

No	Metode	Rumus
1.	Watson-Felix Model	$E = 5.2 \times (KLOC)^{0,91}$
2.	Bailey-Basili Model	$E = 5.5 + 0.73 \times (KLOC)^{1.18}$
3.	Boehm Simple Model	$E = 3.2 \times (KLOC)^{1,05}$
4.	Doty Model (untuk KLOC > 9)	$E = 5.28 \times (KLOC)^{1,047}$

McConnell (2004, p. 564)juga menjelaskan bahwa kemungkinan *error* yang dapat ditemukan dalam sebuah *project* tergantung pada kualitas pengembangan perangkat lunak yang dilakukan. Semakin baik kualitas pengembangan perangkat lunak tersebut, maka semakin kecil ditemukan *error* dalam *project* tersebut. Berikut adalah beberapa rentang kemungkinan *error* tersebut :

a. *Industry Average* : 1-25 *error* tiap 1 KLOC
b. *Microsoft Application* : 10-20 *error* tiap 1 KLOC pada tiap tahap pengujian *in house* dan 0.5 *error* tiap KLOC pada tahap peluncuran

Penelitian ini menggunakan *tool Code Line Counter*untuk menghitung jumlah *lines of code* (LOC) pada aplikasi yang dikembangkan. Sedangkan jumlah *error*aplikasi yang dikembangkan dihitung menggunakan *tool Code Advisor for Visual Basic 6*. Setelah didapatkan nilai KLOC dari aplikasiSistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggikemudian dibandingkan dengan *standard Industry Average* dan *Microsoft Application*. Aplikasi akan dikatakan LOLOS pengujian jika jumlah *error*≤ (lebih kecil sama dengan) standar yang digunakan.

11. Faktor Kualitas *Functionality*

Pressman dalam bukunya menjelaskan bahwa *functionality* merupakan faktor kualitas yang menunjukkan tingkat kemampuan menyediakan fungsi-fungsi yang diharapkan sehingga dapat memberikan kepuasan kepada pengguna(Pressman, 2001, hal. 513).

Faktor kualitas *functionality* dapat diuji dengan analisis fungsionalitas dari setiap komponen pada suatu perangkat lunak. Metode *black-box testing* merupakan metode yang cocok untuk melakukan pengujian fungsionalitas perangkat lunak. Dalam bukunya, Pressman(2001, hal. 459)menjelaskan bahwa *black-box testing*, atau juga disebut *behavioral testing*, fokus pada kebutuhan fungsional dari suatu perangkat lunak. Pengujian ini memungkinkan analis sistem memperoleh kumpulan kondisi *input* yang akan mengerjakan seluruh keperluan fungsional program dan *output* yang akan dihasilkan pada kondisi *input* tertentu.

James Bach dalam tulisanya "*General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo Desktop Applications Edition*"(2005, hal. 4) membagi fungsi dalam sebuah perangkat lunak menjadi dua yaitu : *primaryfunction* (fungsi primer) dan *contributingfunction* (fungsi pendukung). Fungsi primer merupakan fungsi yang utama dalam perangkat lunak, kesalahan dalam fungsi ini akan membuat perangkat lunak tidak layak. Sedangkan fungsi pendukung merupakan fungsi yang memberikan kontribusi pada perangkat lunak, tetapi bukan merupakan fungsi utama. Kesalahan pada fungsi pendukung belum tentu akan membuat perangkat lunak tidak layak.

Dalam pengujian diperlukan standar yang digunakan untuk menentukan apakah sebuah perangkat lunak lolos dalam pengujian faktor kualitas tersebut atau tidak. James Bach(2005) dalam tulisanya yang berjudul "*General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo*" memberikan gambaran bagaimana suatu perangkat lunak dapat dikatakan memenuhi kriteria lolos faktor kualitas *functionality* dalam program *Windows Logo Certification*. Berikut tabel kriterianya :

Tabel 7. Kriteria Lolos/Gagal pada program *Windows Logo Certification*

Kriteria Lolos	**Kriteria Gagal**
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya. 2. Jika ada fungsi pendukung yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya. 2. Jika ada fungsi pendukung yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan

	normal.

12. Faktor Kualitas *Usability*

Agarwal, Tayal, dan Gupta (2010)menjelaskan bahwa *usability* merupakan faktor kualitas perangkat lunak yang menunjukan kapabilitas untuk dapat dimengerti, dipahami dan digunakan oleh pengguna. Sementara itu, Anne Mette Jonassen Hass(2008) menjelaskan bahwa *usability* merupakan faktor kualitas yang menunjukan kecocokan perangkat lunak dengan penggunanya, dalam hal efektivitas, efisiensi, dan kepuasan pengguna.

Usability berkaitan tentang bagaimana perangkat lunakdapat dimengerti, dipahami dan digunakan oleh pengguna. Standar ISO 9126 mengkategorikan *usability* sebagai faktor kualitas nonfungsional.Standar ISO 9126 membagi faktor kualitas *usability* menjadi beberapa subfaktor yaitu *understandability*, *learnability*, *operability* dan *attractiveness* (Hass, 2008).

Setiap sub faktor dalam faktor kualitas *usability* mempunyai wilayah kualitasnya masing-masing. *Understandability* berkaitan dengantingkat kesulitan pengguna untuk mengerti bagaimana menggunakan suatu perangkat lunak dalam konsep logis. *Learnability* berkaitan dengan bagaimana pengguna dapat mempelajari penggunakan suatu perangkat lunak. *Operability* berkaitan dengan bagaimana pengguna dapat menggunakan fungsi-fungsi yangada dalam suatu perangkat lunak. Sedangkan*attractiveness* berhubungan dengan bagaimana kemenarikan perangkat lunak sehingga pengguna tertarikdan mau menggunakannya.

Anne Mette Jonassen Hass (2008)menjelaskan bahwa faktor kualitas *usability* dapat diuji dengan metode *survey* atau kuesioner. Metode *survey* atau kuisioner digunakanuntuk menganalisa faktor kualitas *usability* dari sisi subjektif pengguna.Pengguna yang dimaksud adalah pengguna akhir yang nantinya akan menggunakan perangkat lunak tersebut. Pertanyaan-pertanyaan yang digunakan dalam kuisioneratau angket harus mencerminkanpresepsi pengguna terhadap perangkat lunak yang dikembangkan. Pertanyaan-pertanyaan tersebut juga seharusnya mencakup pada sub faktor kualitas *usability* yaitu *understandability*, *learnability*, *operability* dan *attractiveness*(Hass, 2008, hal. 254).

13. Faktor Kualitas *Maintainability*

Aspek *maintainability* dijelaskan sebagai usaha yang diperlukan untuk mencari dan membetulkan kesalahan pada sebuah program (Pressman, 2001, hal. 510). Sedangkan standar ISO 9126 mendefinisikan aspek *maintainability* sebagai kemudahan sebuah perangkat lunak untuk diperbaiki dikemudian hari. Beberapa sub faktor kriteria faktor kualitas *maintainability* yang dinilai menurut standar ISO 9126 antara lain adalah *analyzability*, *changeability*, *stability*, *testability*, dan *maintainability compliance*.

Faktor kualitas *maintainability* merupakan faktor yang hanya diukur secara tidak langsung (Pressman, 2001, hal. 509). Oleh karena itu, dibutuhkan beberapa ukuran (*metric*) yang didefinisikan dan penilaiannya diukur secara objektif. Pengukuran biasanya dapat berbentuk*checklist*. McCall menetapkan beberapa pengukuran yang dapat digunakan dalam pengujian kualitas perangkat lunak. Metrik faktor kualitas perangkat lunak tersebut dapat dilihat pada tabel 8.

Berdasarkan tabel 8, maka pengujian untuk aspek *maintainability* ini menggunakan ukuran-ukuran (*metrics*) antara lain : *Concision, Consistency, Instrumentation, Modularity, Self-Documentation,* dan *Simplicity*.Aspek *maintainability*yang diuji oleh peneliti pada penelitian kali ini adalah aspek *Consistency, Instrumentation,* dan *Simplicity*.Pengujian aspek *maintainability*dilakukan secara operasional oleh pengembang.

Tabel 8. Keterhubungan antara faktor kualitas perangkat lunak dengan ukuran-ukuran (*metrics*)**(**Pressman, 2001, hal. 512**)**

Quality factor \ Software quality metric	Correctness	Reliability	Efficiency	Integrity	Maintainability	Flexibility	Testability	Portability	Reusability	Interoperability	Usability
Auditability				x			x				
Accuracy		x									
Communication commonality										x	
Completeness	x										
Complexity		x				x	x				
Concision			x		x	x					
Consistency	x	x			x	x					
Data commonality										x	
Error tolerance		x									
Execution efficiency			x								
Expandability						x					
Generality						x		x	x	x	
Hardware Indep.								x	x		
Instrumentation				x	x		x				
Modularity		x			x	x	x	x	x	x	
Operability			x								x
Security				x							
Self-documentation					x	x	x	x	x		
Simplicity		x			x	x	x				
System Indep.								x	x		
Traceability	x										
Training											x

(Adapted from Arthur, L. A., *Measuring Programmer Productivity and Software Quality*, Wiley-Interscience, 1985.)

B. Penelitian yang Relevan

1. **Skripsi oleh mahasiswa Universitas Islam Negeri Malang atas nama Alfi Sukmawan (2008)dengan judul Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi.**

Dalam penelitian tersebut, sistem pakar dibuat dengan bahasa PHP. *Knowledge base* yang digunakan disimpan dalam dalam *database* MySQL yang dijadikan satu paket dengan PHP Triad/Apache. Pengetahuan yang digunakan dalam sistem penelitian tersebut menggunakan tes DAT (*Differential Aptitude Test*).

2. **Skripsi mahasiswa UIN Jakarta , Idam Kusumo Wardono (2011) yang berjudul Pengembangan Sistem Pendukung Keputusan dalam Pemilihan Fakultas Perkuliahan Berbasis Mobile Web.**

Dalam penelitian tersebut, sistem pakar dibuat dengan bahasa PHP. Penulis menggunakan metode *incremental* sebagai metode pengembangan sistem dan metode *analytical hierarchy process* (AHP) sebagai metode inferensi.

3. **Skripsi mahasiswi Universitas Siliwangi, Nina Nurfitriana (2012)dengan judul Rancang Bangun Aplikasi Sistem Pakar Berbasis Web untuk Rekomendasi Pemilihan Jurusan di Universitas Siliwangi.**

Dalam penelitian tersebut, sistem pakar dibuat dengan bahasa PHP. *Knowledge base* yang digunakan disimpan dalam dalam *database* MySQL yang dijadikan satu paket dengan PHP Triad/Apache. Representasi pengetahuan menggunakan teknik *Forward Chaining*.

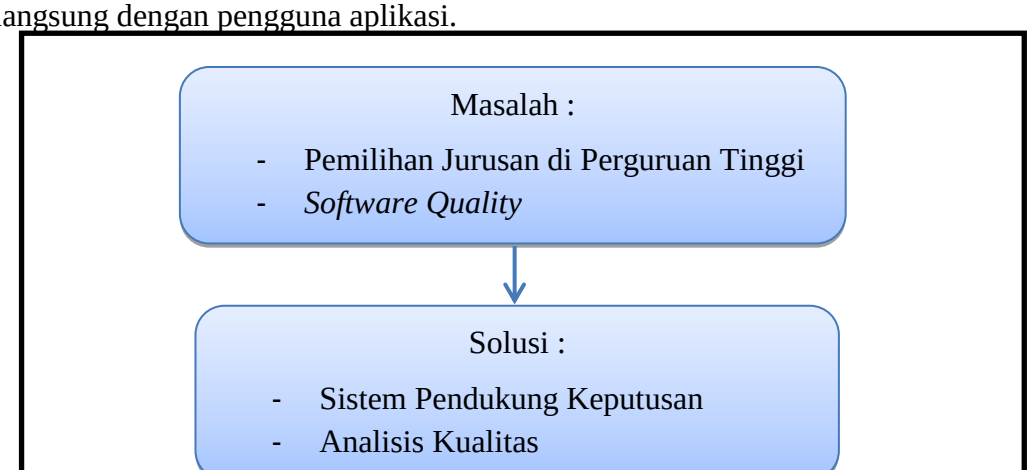
C. Kerangka Berfikir

Pemilihan jurusan kuliah merupakan keputusan yang sulit bagi sebagian besar siswa SMA. Sistem pakar dapat menjadi solusi dari permasalahan tersebut. Penelitian ini bermaksud untuk mengembangkan sebuah aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi. Dengan adanya sistem pakar ini, diharapkan siswa SMA dapat memilih jurusan kuliah yang sesuai dengan jenis kecerdasannya.

Aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi akan dibuat untuk dapat dijalankan dikomputer. Pada pengembangan aplikasi tersebut, digunakan bahasa pemrograman Visual Basic dan *database*MySQL.

Sebagai sebuah produk pengembangan perangkat lunak, aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi diharapkan dapat memenuhi standar kualitas perangkat lunak (*Software Quality)*. Kualitas tersebut dapat dianalisa dari faktor-faktor yang disebutkan dalam *McCall Quality Factors* ataupun *ISO-9126 Quality Factors*.

Penelitian ini menggunakan empat faktor kualitas untuk analisis kualitas perangkat lunak yang dibuat, yaitu faktor kualitas *correctness, functionality, usability,*dan *maintainability*. Pengujian dilakukan dari sisi internal dan eksternal perangkat lunak. Pengujian dari sisi internal perangkat lunak meliputi pengujian faktor kualitas *correctness,functionality*, dan*maintainability*. Sementara pengujian dari sisi ekternal meliputi pengujian faktor kualitas *usability* yang berkaitan langsung dengan pengguna aplikasi.



Gambar 9. Kerangka Berfikir

D. Pertanyaan Penelitian

Berdasarkan kerangka berpikir yang telah dijelasakan sebelumnya, Penulis merumuskan beberapa pertanyaan penelitian antara lain :

1. Apakah aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi yang dikembangkan dalam penelitian ini memenuhi standar faktor kualitas *correctness*?
2. Apakah aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi yang dikembangkan dalam penelitian ini memenuhi standar faktor kualitas *functionality*?
3. Apakah aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi yang dikembangkan dalam penelitian ini memenuhi standar faktor kualitas *usability*?
4. Apakah aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi yang dikembangkan dalam penelitian ini memenuhi standar faktor kualitas*maintainability*?

BAB III
METODE PENELITIAN

A. Desain Penelitian

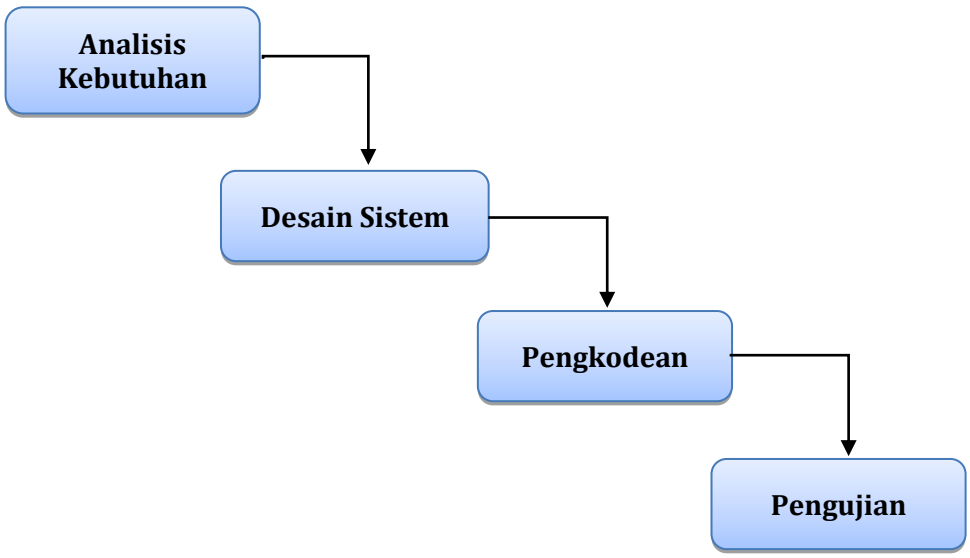
Penelitian dan pengembangan aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi ini menggunakan model penelitian *Research and Development*(R&D). Penulis menggunakan kerangka berpikir sesuai dengan penelitian *Research and Development*(R&D). Borg dan Gall (1996) menyebutkan bahwa yang dimaksud dengan peneltian *Research and Development*adalah "*a process used develop and validate educational product*". Borg dan Gall menjelaskan bahwa penelitian R*esearch and Development* merupakan sebuah proses untuk mengembangkan dan melakukan uji validitas produk yang dikembangkan untuk suatu kepentingan tertentu.Hal ini berarti bahwa penelitian *Research and Development* merupakan suatu proses dalam mengembangkan sebuah produk serta melakukan pengujian terhadap validitas produk yang dikembangkan.

Penelitian *Research and Development*lebih lanjut dijelaskan oleh Sugiyono (2011, hal. 407) dalam bukunya bahwa penelitian *research and development* merupakan sebuah metode penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut. Penelitian R&D lebih dikhususkan untuk menghasilkan sebuah produk yang nantinya akan diuji keefektifitasannya. Produk yang dihasilkan dapat berupa produk baru (*original*) maupun produk pengembangan atau penyempurnaan dari produk yang telah ada sebelumnya agar lebih efektif (Sujadi, 2003).

Berdasarkan beberapa sumber mengenai penelitian dengan model *Research and Development*, dalam penelitian ini penulis ingin melakukan pengembangan dan pengujian perangkat lunak berupa aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi.

Dalam melakukan pengembangan perangkat lunak tersebut, penulis menggunakan panduan *software engineering* karangan Roger S. Pressman yang menjelaskan bahwa tahapan dalam pengembangan perangkat lunak yaitu (1) Analisis Kebutuhan, (2) Desain Sistem, (3) Pengkodean dan (4) Pengujian.Mengacu pada buku *software engineering* tersebut penulis mengembangkan perangkat lunak menggunakan salah satu model proses klasik yaitu *waterfall'smodel*

. *Waterfall'smodel* merupakan suatu proses model dalam mengembangkan perangkat lunak yang memiliki sifat sistematis, berurutan dalam membangun suatu software (Pressman, 2001, hal. 29).



Gambar 10. *Waterfall's Model*

Metode *waterfall* dipilih karena setiap tahapan prosesnya mudah dipahami dan memiliki struktur sistem yang jelas, selain itu metode *waterfall's model* memiliki pendekatan secara sistematis dan urut sehingga penelitian yang dilakukan lebih terkontrol serta terjadwal dengan baik.

Sistematika *waterfall's model* sendiri memiliki beberapa tahapandalam mengembangkan produk penelitian. Mengacu pada model tersebut, beberapa hal yang akan dilakukan dalam mengembangkan produk penelitian dalam hal ini perangkat lunak antara lain :

1. Analisis Kebutuhan (*Software Requirement Analysis*)

*Requirement analysis*atau analisis kebutuhan merupakan sebuah tahapan dimana pengembang menganalisa kebutuhan atau keinginan yang diharapkan oleh pengguna untuk perangkat lunak yang akan dikembangkan. Tahap analisis kebutuhan mencakup analisis fungsional perangkat lunak, desain antar muka (*user interface*)perangkat lunak dan lain-lain. Pada tahapan ini diharapkan segala kebutuhan mengenai pengembangan perangkat lunak dapat dirumuskan sebagai dasar pengembangan perangkat lunak nantinya.

2. Desain (*Design*)

Setelah melakukan analisa kebutuhan pada langkah sebelumnya maka pada tahap selanjutnya yaitu desain sistem akan dilakukan pemodelan terhadap sistem yang akan dibuat. Sebuah bahasapemodelandapatberupa grafis atautekstual, dalam hal ini penulis menggunakan permodelan sistem dengan bahasa grafis yaitu *flowchart*.*Flowchart*adalah representasiskematis darisuatu algoritmaatau prosesbertahap.Bahasa pemodelangrafismenggunakan teknikdiagram

dengansimbolbernamayang mewakilikonsep-konsep, garis yangmenghubungkansimbol danmewakili hubunganmaupun notasigrafislainnyauntuk mewakiliberbagaikendala.Dengan dibuatnya desain pengembangan sistem perangkat lunak nantinya akan mempermudah pengembang dalam proses pengkodean perangkat lunak tersebut.

3. Pengkodean (*Coding*)

Proses pengkodean merupakan tindak lanjut dari proses desain sistem. Tahap pengkodean merupakan tahap implementasi dari perangkat lunak yang akan dibuat. Proses implementasi dilakukan sesuai dengan desain yang telah dibuat sebelumnya. Pengembang menterjemahkan desain model ke dalam kode-kode untuk membangun perangkat lunak. Pada tahapan ini pengembang juga melakukan testing pada tiap bagian implementasi yang dibuatnya secara langsung (*white-box test*).

4. Pengujian (*Testing*)

Setelah implementasi produk perangkat lunak selesai maka tahapan selanjutnya adalah proses pengujian. Sebuah perangkat lunak yang dikembangkan harus diuji kelayakannya sebelum dilepas dan digunakan oleh pengguna. Proses *testing* atau pengujian merupakan sebuah proses penting dimana nantinya akan menentukan kualitas atau kelayakan dari perangkat lunak yang dikembangkan. Pada tahapan pengujian ini penulis mengacu pada Mc Call's *software quality factors* dan ISO 9126 *quality*sebagai standar yang digunakan dalam melakukan uji kualitas perangkat lunak.

Analisis kualitas pada penelitian ini difokuskan pada empat faktor kualitas perangkat lunak yaitu faktor*correctness, functionality, usability*dan*maintainability*. Pemilihan beberapa faktor kualitas perangkat lunak tersebut didasari pada beberapa pertimbangan yaitu :

a. *Correctness*

Faktor kualitas *correctness* dipilih dengan pertimbangan pengujian pada faktor ini dapat menunjukan bagaimana kualitas *source code* aplikasi yang dikembangkan. Hal ini dikarenakan dalam pengujianya dilakukan analisis jumlah *error* pada tiap *kilo line of codes* (*KLOC*).

b. *Functionality*

Fakor kualitas *functionality* dipilih dengan pertimbangan sebuah perangkat lunak harus bisa memenuhi semua fungsi-fungsi yang diharapkan dan memastikan bahwa fungsi-fungsi tersebut berjalan dengan baik dan benar.

c. *Usability*

Faktor kualitas *usability* dipilih dengan pertimbangan bahwa aplikasi harus dapat digunakan oleh pengguna dengan mudah. Pengujian fakor kualitas *usability* akan menunjukan bagaimana aplikasi dapat dimengerti, dipahami, dan digunakan oleh pengguna.

d. *Maintainability*

Faktor kualitas *maintainability* dipilih karena dapat menunjukan kemudahan aplikasi untuk dipahami, dikembangkan, dan diperbaiki. Hal ini bermanfaat untuk pemeliharaan sistem nantinya.

Adapun dalam pengujiannya, empat faktor kualitas tersebut dilaksanakan dengan cara yang berbeda. Berikut analisis kualitas perangkat lunak pada penelitian ini:

a. Faktor kualitas *correctness* diuji dengan analisis *error per kilo line of codes* (*KLOC*), yang akan dibandingkan dengan standar*error/KLOC*pada *industry average* dan *Microsoft Application*. Untuk menghitung jumlah *lines of code* (LOC) digunakan *tool Code Line Counter*. Sedangkan jumlah *error* dihitung menggunakan *tool Code Advisor for Visual Basic 6.* Perangkat lunak dinyatakan layak apabila *error per kilo line of codes* (*KLOC*)perangkat lunak yang dikembangkan lebih kecil sama dengan standar *error/KLOC*pada *industry average* dan *Microsoft Application*.
b. Faktor kualitas *functionality* diuji dengan melakukan pengujian pada setiap fungsi pada aplikasi yang dikembangkan. Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi pada aplikasi berkerja sebagai mana mestinya. Standar*Windows Logo Certification* digunakan sebagai standarkriteria lolos faktor kualitas *functionality*
c. Faktor kualitas *usability* dikaji dari penilaian pengguna akhir (*end user*) yang didapat melalui kuesioner. Kuesioner yang digunakan mengacu pada *Computer System Usability Questionnaire* yang dipublikasi oleh J.R. Lewis. Kuesioner ini akan disebar kepada 60 responden yang terdiri dari mahasiswa dan siswa SMA kelas XII.
d. Faktor kualitas *maintainability*diuji secara operasional oleh pengembang sendiri dengan menggunakan kriteria lolos untuk tiga aspek faktor

*maintainability*yaitu aspek c*onsistency, instrumentation,* dan s*implicity.* Aplikasi dinyatakan lolos apabila memenuhi kriteria lolos ke-tiga aspek faktor *maintainability*tersebut.

B. Variable Penelitian

Variabel pengujian yang digunakan dalam pengujian kualitas perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggimengacu pada standar *ISO-9126* dan*McCall*. Berikut definisi operasional setiap variabel yang digunakan:

1. *Correctness*

Correctness merupakan faktor kualitas yang menunjukan tingkat bagaimana perangkat lunak menjalankan fungsi yang dibutuhkannya. Hal ini berkaitan dengan bagaimana program mampu memenuhi spesifikasi dan tujuan yang ingin dicapai oleh pengguna.

2. *Functionality*

Functionality merupakan faktor kualitas yang menunjukan tingkat kemampuan menyediakan fungsi-fungsi yang diharapkan dan memenuhi segala kebutuhan yang diperlukan oleh pengguna dalam segi fungsi sehingga dapat memberikan kepuasan kepada pengguna.

3. *Usability*

Usability merupakan faktor kualitas perangkat lunak yang menunjukan kapabilitas untuk dapat dimengerti, dipahami dan digunakan oleh pengguna. Dengan kata lain *usability* merupakan faktor dimana perangkat lunak dilihat dari sisi kemudahan pengguna.

4. *Maintainability*

*Maintainability*merupakan faktor kualitas perangkat lunak yang menunjukan kemudahan suatu sistem untuk dipahami dan diperbaiki dikemudian hari.Hal ini berhubungan dengan usaha yang diperlukan untuk mencari dan membetulkan kesalahan pada sebuah program.

C. Teknik Pengumpulan Data

1. Studi Pustaka

Dalam penelitian ini studi pustaka dilakukan penulis pada tahap pengembangan perangkat lunak yaitu pada proses perancangan *knowledge base*. Studi pustaka dilakukan dengan mempelajari pustaka yang berkaitan dengan teori kepribadian RIASEC karangan psikolog John L. Holland sebagai panduan ilmu psikologi untuk minat bakat dan jenis kecerdasan.

Hubungan teori kepribadian RIASEC dengan pemilihan jurusan di perguruan tinggi terletak padatiga kode dominan Holland untuk setiap bidang. Data tiga kode dominan Holland untuk setiap daftar jurusandiambil dari *free online database*yang digunakan oleh *US Department of Labor/Employment and Training Administration* (USDOL/ETA) yaitu *The Occupational Information Network* (O*NET).Data tersebut dicocokkan dengan beberapa *research report*lain dan dari beberapa universitas diluar negeri yang menggunakan tiga kode dominan Holland pada setiap jurusan yang disediakan oleh universitas tersebut. Universitas yang menggunakan tiga kode dominan Holland tersebut antara lain : University of Missouri, Salisbury University, Virginia Commonwealth University, The University of Oklahoma, dan Huntingdon Collage.

2. Observasi

Teknik pengumpulan data observasi merupakan salah satu teknik pengumpulan data dengan melakukan pengamatan secara langsung terhadap objek yang diteliti. Observasi yaitu melakukan pengamatan secara langsung ke objek penelitian untuk melihat dari dekatkegiatan yang dilakukan (Riduwan, 2011, p. 76).Metode observasi digunakan untuk melakukan pengamatan objek penelitian aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi. Teknik observasi dalam penelitian ini dilakukan untuk membantu mengumpulkan data pada proses uji kelayakan perangkat lunak yang terkait dengan faktorkualitas *correctness,functionality* dan*maintainability.*

3. Kuesioner

Angket adalah daftar pertanyaan yang diberikan kepada orang lain yang bersedia memberikan respon (responden). Tujuan penyebaran angket ini adalah mencari informasi lengkap dari responden mengenai suatu permasalahan tanpa merasa khawatir bila jawaban yang diberikan tidak sesuai dengan kenyataan dalam pengisian daftar pertanyaan.

Angket atau *questionnaire* merupakan metode pengumpulan data dengan menggunakan daftar pertanyaan. Jenis angket sendiri dibedakan menjadi dua jenis. Angket dibagi menjadi angket terbuka dan angket tertutup (Riduwan, 2011, p. 71). Teknik pengumpulan data kuesioner pada penelitian ini menggunakan jenis angket tertutup.

Angket tertutup atau yang sering disebut dengan angket terstruktur merupakan angket yang disajikan dalam bentuk sedemikian rupa sehingga

responden diminta untuk memilih satu jawaban yang sesuai dengan karakteristik dirinya dengan cara memberikan tanda *checklist*pada pilihan yang diinginkan (Riduwan, 2011, p. 72).

Dalam penelitian ini teknik pengumpulan data menggunakan kuisioner dilakukan untuk analisa faktor kualitas *usability*. Kuesioner dibagikan kepada 60 responden yang dipilih berdasarkan teknik *purposive sampling*.

D. Skala Pengukuran

1. Skala Guttman

Skala Guttman merupakan skala kumulatif yang mengukur suatu dimensi saja dari suatu variabel yang multidimensi. Skala Guttman disebut juga skala *Scalogram* yang sangat baik untuk meyakinkan peneliti tentang kesatuan dimensi dan sikap atau sifat yang diteliti, yang sering disebut dengan atribut *universal*(Riduwan, 2011, p. 90). Skala Guttman digunakan untuk jawaban yang bersifat jelas (tegas) dan konsisten sehingga digunakan dalam penelitian yang menginginkan jawaban yang tegas dan konsisten terhadap suatu permasalahan yang ditanyakan. Contoh penggunaan dari Skala Guttman misal: Yakin – Tidak yakin, Ya – Tidak, Benar – Salah, dan lain-lain.

Penelitian ini menggunakan skala Guttman untuk mendapatkan data pada proses pengujian *alpha*. Uji *alpha* ini nantinya akan dilakukan oleh ahli pada lingkungan pengembang. Pada pengujian *alpha* nantinya akan dibuat sebuah tabel spesifikasi pengujian yang digunakan untuk menguji kinerja perangkat lunak. Skala Guttman juga digunakan pada *test case* untuk menguji salah satu faktor kualitas perangkat lunak yaitu*functionality*. Tabel spesifikasi dan *test case*yang

digunakan pada penelitian ini menggunakan skala Guttman karenadiinginkan data hasil unjuk kerja yang bersifat jelas dan tegas.

2. Skala Likert

Skala likert merupakan skala pengukuran yang digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok tentang kejadian atau gejala sosial. Dengan gejala sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian(Riduwan, 2011, p. 87).

Skala likert menjadikan variabel yang akan diukur dijabarkan menjadi dimensi, dimensi dijabarkan menjadi sub variabel kemudian sub variabel dijabarkan kembali menjadi indikatar-indikator yang dapat diukur. Indikator-indikator yang terukur ini dapat dijadikan titik tolak untuk membuat item instrumen yang berupa pertanyaan atau pernyataan yang perlu dijawab oleh responden. Setiap jawaban dihubungkan dengan bentuk pertanyaan atau pernyataan dukungan sikap yang diungkapkan dengan kata-kata

Interval pada skala likert dibagi menjadi dua untuk pernyataan positif dan pernyataan negatif. Contoh dari interval skala Likert untuk kedua pernyataan tersebut adalah sebagai berikut :

Tabel 9. Interval Skala Likert

Pernyataan Positf		Pernyataan Negatif	
Sangat Setuju (SS)	5	Sangat Setuju (SS)	1
Setuju (S)	4	Setuju (S)	2
Netral (N)	3	Netral (N)	3
Tidak Setuju (TS)	2	Tidak Setuju (TS)	4
Sangat Tidak Setuju (STS)	1	Sangat Tidak Setuju (STS)	5

Skala likert digunakan untuk mendapatkan data pada uji validitas perangkat lunak. Skala Likert digunakan untuk mengukur faktor validitas perangkat lunak pada tahap pengujian*beta*. Skala Likert nantinya akan digunakan pada kuesioner yang dibagikan kepada responden. Kuesioner tersebut digunakan untuk menguji faktor *usability*.

E. Objek Penelitian

Fokus objek penelitian yang dilakukan pada penelitian ini adalah pengembangan perangkat lunakSistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi.

F. Instrumen Penelitian

Instrumen yang digunakan dalam penelitian ini mengikuti pada teknik pengumpulan data, yaitu observasi dan angket. Untuk melakukan pengujian tersebut instrumen yang digunakan antara lain :

1. Observasi

a. *Code Line Counter*

Code Line Counter adalah sebuah aplikasi untuk menghitung baris kode. Mampu menghitung baris kode pada bahasa C, C + +, C #, Java, Delphi atau Pascal, COBOL, VB, PHP, ASP, XML, Perl, Fortran, SQL Script dan lain sebagainya. Termasuk jumlah total dan persentase untuk komentar, baris kosong dan *source lines*. Dapat menghitung dalam bentuk*subfolders*dan *multi-types count*. *Tool* ini digunakan untuk menghitung *lines of code* dari *source code* aplikasi yang dikembangkan dalam penelitian ini. Hal ini berhubungan dengan pengujian faktor kualitas *correctness*.

b. *Code Advisor for Visual Basic 6*

Code Advisor for Visual Basic 6 merupakan *freewaretools* yang dikembangkan oleh *Microsoft*. Dalam *website* resmi *Microsoft*(2012) dijelaskan bahwa *Code Advisor for Visual Basic 6* merupakan aplikasi *plugs-in* untuk Visual Basic 6.0 untuk menganalisa kode perangkat lunak yang dikembangkan dan menyarankan perbaikan.

Code Advisor for Visual Basic 6 adalah *add-in* yang digunakan dalam menganalisa jumlah*error* kode yang ditemukan dalam *source code*untuk memastikan bahwa kode telah memenuhi standar pengkodean yang telah ditentukan.*Tool* ini digunakan untuk menghitung jumlah *error* kode pada perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi yang dikembangkan dalam penelitian ini. Hal ini berkaitan dengan pengujian faktor kualitas *correctness*.

c. *Test case*

Dalam pengujian faktor kualitas *functionality* dengan metode *black-box testing*, dibutuhkan *test case*. Argawal, Tayal dan Gupta (2010) menjelaskan bahwa *test case* merupakan seperangkat instruksi yang didesain untuk mengetahui kesalahan yang terdapat dalam perangkat lunak.

Seperangkat *test case* perlu didokumentasikan dengan baik.*Test case* sebaiknya dibuat dalam format yang sama agar mempermudah proses pengujian dan dokumentasi menjadi mudah dimengerti. Dalam penelitian ini, *test case* digunakan penulis untuk pengujian faktor kualitas *functionality*. Berikut adalah format *test case*yang digunakan dalam penelitian ini :

Tabel 10. Format *Test Case* yang Digunakan dalam Pengujian

Test case id	Nama yang unik untuk identifikasi *test case*
Purpose	Tujuan dari *test case*
Assumptions	Syarat kondisi awal yang harus terpenuhi sebelum *test* dapat dijalankan.
Test data	Variabel atau kondisi yang akan di *test*.
Steps	Langkah-langkah yang dijalankan.
Expected result:	Hasil yang seharusnya didapatkan (yang menunjukkan bahwa tidak ada kesalahan dalam perangkat lunak)
Actual result:	Hasil yang didapat dalam pengujian.
Pass/Fail:	Keterangan : Lolos atau Gagal.

d. Instrumen *Maintainability*

Pengujian untuk faktor kualitas*maintainability*pada penelitian ini menggunakan ukuran-ukuran (*metrics*) yang berhubungan dengan faktor kualitas *maintainability*itu sendiri. Aspek faktor kualitas*maintainability*yang diuji pada penelitian ini adalah aspek c*onsistency, instrumentation,* dan s*implicity.* Pengujian faktor kualitas *maintainability*dilakukan oleh pengembang sendiri secara operasional. Pengujian ke-tiga aspek faktor kualitas *maintainability*tersebutmembutuhkan alat ukur yang menentukan apa saja yang harus dinilai dan standar yang digunakan sebagai patokan keberhasilan pengujian untuk setiap aspek faktor *maintainability* yang diuji. Instrumen yang digunakan untuk pengujian faktor kualitas *maintainability*dapat dilihat pada tabel di bawah ini :

Tabel 11.Instrumen *Maintainability*

Aspek	Aspek yang dinilai	Kriteria Lolos
Instrumentation	Terdapat peringatan pada sistem pengolah data untuk mengidentifikasi kesalahan	Ketika ada kesalahan yang dilakukan oleh *user*, maka sistem akan mengeluarkan peringatan untuk meng-identifikasi kesalahan.
Consistency	Penggunaan satu bentuk rancangan pada seluruh rancangan sistem	Bentuk rancangan sistem pengolah data mempunyai satu bentuk yang sama. Hal ini dapat dilihat pada bagian implementasi sistem.
Simplicity	Kemudahan dalam pengelolaan, perbaikan, dan pengembangan sistem	Mudah untuk dikelola, diperbaiki, dan dikem-bangkan. Hal ini dapat dilihat pada tahapan-tahapan proses penulisan kode program.

2. Angket

Instrumen lain yang digunakan penulis dalam penelitian ini untuk melakukan pengujian adalah angket. Instrumen angket digunakan untuk melakukan pengujian *alpha* dan pengujian *beta*.

a. Pengujian *Alpha*

Pengujian*alpha* dilakukan dengan bantuan tabel spesifikasi yang menggunakan skala Guttman. Pengujian ini dilakukan oleh ahli dalam lingkungan pengembang karena *alpha test* merupakan proses pengujian validasi perangkat lunak dengan perspektif pengembang perangkat lunak.Tabel spesifikasi pengujian yang digunakan adalah sebagai berikut :

Tabel 12. Spesifikasi Uji Membuka Aplikasi

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Membuka Aplikasi	Apabila MySQL belum aktif maka akan muncul notifikasi "Ada kesalahan dengan server, periksa apakah server sudah berjalan !".		
		Pengguna masuk ke halaman *login* dan muncul menu fitur *login* (beserta *textbox* nama pengguna dan kata sandi), daftar, *about* (tentang), *help* (bantuan), dan batal.		

Tabel 13.Spesifikasi Uji Halaman *Login*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Halaman Login	Pengguna dapat masuk ke semua fitur menu (daftar, tentang, batal, danbantuan) dengan memilih tombol-tombol tersebut yang ada pada halaman *login*		
2	Login	Pengguna dapat *login* dengan cara menuliskan data diri pada *textbox* nama pengguna dan *textbox*kata sandi kemudian memilih tombol masuk		
		Apabila pengguna memilih tombol masuk ketika*textbox*nama pengguna atau kata sandi atau keduanya belum terisi maka akan muncul notifikasi "login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar"		
		Apabila data diri yang ditulis oleh pengguna tidak cocok dengan data pengguna yang tersimpan dalam *database* maka akan muncul notifikasi "login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar"		
		Pengguna dapat kembali ke halaman *login* dengan cara memilih tombol ok pada notifikasi "login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar"		
		Apabila data diri yang ditulis oleh pengguna cocok dengan data pengguna yang tersimpan dalam *database* maka akan muncul notifikasi "login berhasil"		

Tabel 14.Spesifikasi Uji Halaman *Login*(lanjutan)

3	Keluar aplikasi	Apabila pengguna memilih tombol batal pada halaman *login* maka akan muncul notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"		
		Pengguna dapat keluar aplikasi dengan memilih tombol ya pada notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"		
		Pengguna dapat batal keluar aplikasi dengan cara memilih tombol tidak pada notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"		

Tabel 15.Spesifikasi Uji Halaman *About*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman *about*	Pengguna dapat masuk ke halaman*about* dengan cara memilih tombol tentang pada halaman *login*		
2	Halaman *about*	Muncul label keterangan yang berisi nama aplikasi, versi, diskripsi singkat dan nama pengembang		
		Pengguna dapat masuk ke semua fitur menu (info sistem dan bantuan) dengan memilih tombol-tombol tersebut yang ada pada halaman *about*		
3	Fitur info sistem	Pengguna dapat melihat info sistem dengan cara memilih tombol info sistem pada halaman *about*		
		Jendela info sistem dapat ditutup dengan cara memilih *icon* tutup (x) pada jendela info sistem		
4	Keluar halaman *about*	Pengguna dapat keluar dari halaman *about* dengan memilih tombol ok pada halaman *about*, maka akan muncul halaman *login*		

Tabel 16.Spesifikasi Uji Halaman Pendaftaran

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman pendaftaran	Pengguna dapat masuk ke halamanpendaftaran dengan cara memilih tombol daftar pada halaman *login*		
2	Pendaftaran	Pengguna dapat melakukan pendaftaran dengan mengisi data diri pada *textbox* nama pengguna dan *textbox*kata sandi kemudian memilih tombol simpan		
		Apabila data pengguna belum lengkap maka akan muncul notifikasi “Data masih kosong. Isilah dengan lengkap”		
		Apabila nama pengguna atau kata sandi yang dituliskan lebih dari 15 karakter maka akan muncul notifikasi “Maaf data yang anda masukkan lebih dari 15 karakter”		
		Apabila nama pengguna yang dituliskan telah terdaftar dalam database maka akan muncul notifikasi “Maaf nama pengguna tersebut sudah terdaftar”		
		Pengguna dapat kembali ke halaman pendaftaran dengan memilih tombol ok pada setiap notifikasi peringatan gagal pendaftaran		
		Apabila data pengguna sudah lengkap, tidak ada data yang lebih dari 15 karakter, dan nama pengguna yang dituliskan belum terdaftar dalam database maka data pengguna tersebut akan tersimpan dalam database dan akan muncul notifikasi “Pendaftaran berhasil!”		
		Pengguna dapat memasuki halaman diagnosa untuk *normal user* dengan cara memilih tombol ok pada notifikasi “Pendaftaran berhasil!”		
3	Keluar halaman pendaftaran	Pengguna dapat keluar dari halaman pendaftaran dengan memilih tombol kembali pada halaman pendaftaran, maka akan muncul halaman *login*		

Tabel 17.Spesifikasi Uji Halaman Diagnosa Untuk *Normal User*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman diagnosa untuk *normal user*	Pengguna dapat masuk ke halaman diagnosa untuk *normal user* dengan cara berhasil *login* atau daftar dengan data tipe pengguna "user"		
2	Diagnosa*normal user*	Muncul pernyataan awal pada label pernyataan untuk diagnosa awal, serta tombol ya, tidak, kenapa?, bantuan, dan logout (setiap tombol berfungsi dengan benar dan melakukan tugas masing-masing fitur)		
		Pengguna dapat menjawab pernyataan yang muncul dengan cara memilih tombol ya atau tidak, sesuai dengan keadaan pengguna, maka akan muncul pernyataan selanjutnya (sesuai jawaban yang dipilih)		
		Apabila pengguna sudah menjawab pernyataan maka tombol kembali akan muncul dan dapat dipilih		
		Apabila data untuk pernyataan selanjutnya belum tersedia padahal data sebelumnya merupakan jenis data pertanyaan maka akan muncul notifikasi "Maaf data masih kosong"		
		Pengguna dapat kembali ke halaman diagnosa dengan cara memilih tombol ok pada notifikasi "Maaf data masih kosong"		
		Pengguna akan mendapatkan solusi dari diagnosa jika tidak terdapat lagi pernyataan yang perlu dijawab oleh pengguna karena pernyataan tersebut merupakan jenis data jawaban		
		Apabila pengguna mendapatkan solusi diagnosa (jenis data jawaban) maka tombol jawab (ya dan tidak) tidak dapat di pilih lagi, sedangkan tombol detail info muncul dan dapat dipilih		
		Apabila data *rulebase* pada *database* masih kosong maka akan muncul notifikasi "rulebase masih kosong"		
		Aplikasi akan tertutup apabila pengguna memilih tombol ok pada notifikasi "rulebase masih kosong"		
3	Ulang diagnosa	Pengguna dapat melakukan diagnosa ulang dengan cara memilih tombol ulang pada halaman diagnosa. Maka label pernyataan pada halaman diagnosa akan kembali pada pernyataan awal proses diagnosa		

Tabel 18.Spesifikasi Uji Halaman Diagnosa Untuk *Admin*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman diagnosa untuk *admin*	Pengguna dapat masuk ke halaman diagnosa untuk *admin* dengan cara berhasil *login* sebagai akun dengan data tipe pengguna“admin”		
2	Diagnosa*admin*	Muncul pernyataan awal pada label pernyataan untuk diagnosa awal, serta tombol ya, tidak, kenapa?, ubah, kosongkan, bantuan, dan logout (setiap tombol berfungsi dengan benar dan melakukan tugas masing-masing fitur)		
		Pengguna dapat menjawab pernyataan yang muncul dengan cara memilih tombol ya atau tidak, sesuai dengan keadaan pengguna, maka akan muncul pernyataan selanjutnya (sesuai dengan jawaban yang dipilih)		
		Apabila pengguna sudah menjawab pernyataan maka tombol kembali, hapus, sisipkan fakta, dan ulang akan muncul dan dapat dipilih (setiap tombol dapat melakukan tugas fitur masing-masing dengan benar)		
		Apabila data untuk pernyataan selanjutnya belum tersedia padahal data sebelumnya merupakan jenis data pertanyaan maka akan muncul notifikasi “Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?”		
		Pengguna dapat masuk ke halaman tambah fakta dengan cara memilih tombol ya pada notifikasi “Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?”		
		Pengguna dapat kembali ke halaman diagnosa *admin* tanpa menambah fakta baru dengan cara memilih tombol tidak pada notifikasi “Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?”, maka akan muncul pernyataan sebelumnya pada halaman diagnosa		
		Pengguna akan mendapatkan solusi dari diagnosa jika tidak terdapat lagi pernyataan yang perlu dijawab oleh pengguna karena pernyataan tersebut merupakan jenis data jawaban		

Tabel 19.Spesifikasi Uji Halaman Diagnosa Untuk *Admin* (lanjutan)

		Apabila pengguna mendapatkan solusi diagnosa (jenis data jawaban) maka tombol jawab (ya dan tidak) tidak dapat di pilih lagi, sedangkan tombol detail info muncul dan dapat dipilih (melakukan tugas fitur detail info)		
		Apabila data *rulebase* pada *database* masih kosong maka akan muncul notifikasi "rulebase masih kosong"		
		Pengguna dapat masuk ke halaman tambah awal untuk memasukkan data awal pada *rulebase* dengan cara memilih tombol ok pada notifikasi "rulebase masih kosong"		
3	Ulang diagnosa	Pengguna dapat melakukan diagnosa ulang dengan cara memilih tombol ulang pada halaman diagnosa. Maka label pernyataan pada halaman diagnosa akan kembali pada pernyataan awal proses diagnosa		
4	Fitur ubah data	Pengguna dapat masuk ke halaman ubah data untuk mengubah data yang muncul dalam label pernyataan dengan cara memilih tombol ubah pada halaman diagnosa *admin*		
5	Fitur hapus data	Pengguna dapat menghapus data yang muncul dalam label pernyataan (beserta anak-anaknya) dengan cara memilih tombol hapus pada halaman diagnosa *admin*, maka akan muncul notifikasi "Data berhasil dihapus!"		
		Pengguna akan kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi "Data berhasil dihapus!", maka pada halaman diagnosa akan muncul pernyataan yang ditampilkan sebelum pernyataan yang dihapus tersebut		
6	Fitur sisipkan fakta	Pengguna dapat masuk ke halaman penyisipan fakta untuk menyisipkan data diantara data yang telah tersedia dengan cara memilih tombol sisipkan fakta pada halaman diagnosa *admin*		
7	Fitur kosongkan rulebase	Pengguna dapat mengosongkan data yang terdapat pada tabel *rulebase* di *database* dengan cara memilih tombol kosongkan yang ada di halaman diagnosa *admin*, maka akan muncul notifikasi "Tabel rule berhasil dikosongkan"		
		Muncul halaman login setelah memilih tombol ok pada notifikasi "Tabel rule berhasil dikosongkan"		

Tabel 20. Spesifikasi Uji Fitur Kembali Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur kembali pada halaman diagnosa	Tombol kembali pada halaman diagnosa (untuk *normal user* dan *admin*) akan muncul jika pengguna telah menjawab pernyataan		
		Pengguna dapat mengulang pernyataan sebelumnya dengan cara memilih tombol kembali pada halaman diagnosa		
		Apabila pengguna memilih tombol kembali padahal pernyataan yang dituju merupakan awal data maka akan muncul pemberitahuan "Anda saat ini berada di awal data"		
		Pengguna dapat melanjutkan proses diagnosa dengan cara memilih tombol ok pada notifikasi "Anda saat ini berada di awal data", maka akan muncul kembali halaman diagnosa		
		Tombol kembali pada halaman diagnosa (untuk *normal user* dan *admin*) akan hilang jika berada di awal data pernyataan diagnosa		

Tabel 21.Spesifikasi Uji Fitur Detail Info Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur detail info pada halaman diagnosa	Tombol detail info pada halaman diagnosa (untuk *normal user* dan *admin*) akan muncul hanya jika proses diagnosa berada pada solusi atau kesimpulan diagnosa (jenis data jawaban)		
		Pengguna dapat melihat detail info mengenai hasil kesimpulan diagnosa dengan cara memilih tombol detail info pada halaman diagnosa, maka akan muncul *frame* info penyedia jurusan yang menjadi hasil kesimpulan tersebut		
		Pengguna dapat kembali ke halaman diagnosa dengan cara memilih *icon* tutup (x) pada *frame* info jurusan		
		Tombol detail info pada halaman diagnosa (untuk *normal user* dan *admin*) akan hilang jika berada pada pernyataan diagnosis dengan jenis data pertanyaan		

Tabel 22. Spesifikasi Uji Fitur Mengapa Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur mengapa pada halaman diagnosa	Tombol kenapa? akan selalu muncul pada halaman diagnosa (untuk *normal user* dan *admin*)		
		Pengguna dapat menanyakan mengapa pernyataan tersebut diberikan dengan cara memilih tombol kenapa? pada halaman diagnosa, maka akan muncul *frame* penjelasan mengenai pernyataan tersebut		
		Pengguna dapat kembali ke halaman diagnosa dengan memilih *icon* tutup (x) pada *frame* penjelasan		

Tabel 23.Spesifikasi Uji Halaman Ubah Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman ubah fakta	Pengguna dapat masuk ke halaman ubah fakta dengan cara memilih tombol ubah pada halaman diagnosa *admin*		
2	Ubah fakta	Muncul data yang akan diubah pada label pernyataan dan label keterangan pada halaman ubah fakta		
		Pengguna dapat langsung mengubah data pada label pernyataan dan label keterangan sesuai keinginan		
		Perubahan data akan tersimpan di *database* dengan cara memilih tombol simpan yang ada di halaman ubah fakta, maka akan muncul notifikasi “Data berhasil diubah”		
		Pengguna dapat kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi “Data berhasil diubah”, maka akan muncul data yang telah berhasil diubah tersebut pada label pernyataan di halaman diagnosa *admin*		
3	Batal ubah fakta	Pengguna dapat kembali ke halaman diagnosa *admin* tanpa jadi mengubah fakta dengan cara memilih tombol kembali pada halaman ubah fakta, maka akan muncul data yang tidak jadi diubah pada label pernyataan di halaman diagnosa *admin*		

Tabel 24. Spesifikasi Uji Halaman Penyisipan Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman penyisipan fakta	Pengguna dapat masuk ke halaman penyisipan fakta dengan cara memilih tombol sisipkan fakta pada halaman diagnosa *admin*		
2	Penyisipan fakta	Muncul data kode pernyataan yang muncul saat tombol sisipkan fakta pada halaman diagnosa *admin* dipilih (pernyataan yang nantinya akan diletakkan dibawah data sisipan baru)		
		Pengguna dapat mengisi pernyataan dan keterangan dari fakta yang akan disisipkan pada *textbox* pertanyaan dan keterangan yang telah disediakan		
		Pengguna dapat memilih letak data pernyataan yang muncul saat tombol sisipkan fakta pada halaman diagnosa *admin* dipilih, yaitu pada pilihan ya atau tidak untuk jawaban dari fakta yang akan disisipkan		
		Data yang akan disisipkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman penyisipan fakta		
		Apabila pernyataan belum dituliskan namun pengguna memilih tombol simpan, maka akan muncul notifikasi "Maaf tuliskan terlebih dahulu fakta di tempat yang sudah disediakan!"		
		Pengguna dapat kembali ke halaman penyisipan fakta dengan cara memilih tombol ok pada notifikasi "Maaf tuliskan terlebih dahulu fakta di tempat yang sudah disediakan!"		
		Apabila pengguna belum memilih letak data sebelumnya namun pengguna memilih tombol simpan, maka akan muncul notifikasi "Pilih salah satu Option yang tersedia"		
		Pengguna dapat kembali ke halaman penyisipan fakta dengan cara memilih tombol ok pada notifikasi "Pilih salah satu Option yang tersedia"		
		Apabila pernyataan telah ditulis dan pilihan letak data sebelumnya telah dipilih maka data tersimpan dalam *database* dan muncul notifikasi “data berhasil di sisipkan"		

Tabel 25. Spesifikasi Uji Halaman Penyisipan Fakta (lanjutan)

		Pengguna dapat kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi “data berhasil di sisipkan", maka akan muncul data yang telah berhasil disisipkan pada label pernyataan di halaman diagnosa *admin*		
3	Batal menyisipan fakta	Pengguna dapat melanjutkan proses diagnosa kembali di halaman diagnosa *admin* tanpa jadi menyisipkan fakta dengan cara memilih tombol kembali pada halaman penyisipan fakta		

Tabel 26.Spesifikasi Uji Halaman Tambah Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman tambah fakta	Pengguna dapat masuk ke halaman tambah fakta dengan cara memilih tombol ya pada notifikasi “Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?” pada halaman diagnosa *admin*		
2	Penambahan fakta	Pengguna dapat mengisi pernyataan dan keterangan dari fakta yang akan ditambahkan pada *textbox* fakta dan *textbox* keterangan		
		Pengguna dapat memilih jenis fakta yang akan ditambahkan, yaitu jenis pertanyaan atau jawaban		
		Data yang akan ditambahkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman tambah fakta		
		Apabila pernyataan belum dituliskan kemudian pengguna memilih tombol simpan maka akan muncul notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"		
		Pengguna dapat kembali ke halaman tambah fakta dengan cara memilih tombol ok pada notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"		
		Apabila pengguna belum memilih jenis fakta yang akan ditambah namun memilih tombol simpan maka akan muncul notifikasi "Anda belum memilih jenis masukan, pilih terlebih dahulu jenis data yang diinginkan"		

Tabel 27. Spesifikasi Uji Halaman Tambah Fakta (lanjutan)

		Pengguna dapat kembali ke halaman tambah fakta dengan cara memilih tombol ok pada notifikasi "Anda belum memilih jenis masukan, pilih terlebih dahulu jenis data yang diinginkan"		
		Apabila pernyataan telah ditulis dan jenis fakta yang akan ditambahkan telah dipilih maka data tersimpan dalam *database* dan muncul notifikasi "Fakta berhasil disimpan!"		
		Pengguna dapat kembali ke halaman diagnosa admin dengan cara memilih tombol ok pada notifikasi "Fakta berhasil disimpan!", maka akan muncul data yang telah ditambahkan pada label pernyataan di halaman diagnosa *admin*		
3	Batal tambah fakta	Pengguna dapat melanjutkan proses diagnosa kembali di halaman diagnosa *admin* tanpa jadi menambahkan fakta dengan cara memilih tombol kembali pada halaman penyisipan fakta		

Tabel 28.Spesifikasi Uji Fitur Bantuan (*Help*)

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Bantuan	Tombol bantuan dengan icon “?” muncul di setiap halaman aplikasi		
		Pengguna dapat melihat halaman bantuan yang berisi petunjuk mengenai penggunaan halaman yang sedang diakses dengan cara memilih *icon* bantuan (?) yang ada pada halaman tersebut		
		Pengguna dapat kembali ke halaman yang sedang diakses dengan memilih *icon* tutup (x) pada halaman bantuan		
		Khusus pada halaman bantuan utama untuk halaman diagnosa *admin* pengguna dapat masuk ke 2 jenis kategori petunjuk. Kategori tersebut adalah diagnosis dan pengaturan		
		Pengguna pada halaman kategori bantuan untuk halaman diagnosis *admin* dapat kembali ke halaman bantuan utama dengan memilih *icon* tutup (x) pada halaman kategori bantuan tersebut		

Tabel 29.Spesifikasi Uji Halaman Tambah Awal

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman tambah awal	Pengguna dapat masuk ke halaman tambah awal dengan cara memilih tombol ok pada notifikasi “rulebase masih kosong” pada halaman diagnosa *admin*		
2	Isi fakta awal	Pengguna dapat mengisi pernyataan dan keterangan dari fakta awal pada *textbox* fakta dan *textbox* keterangan		
		Data yang akan dimasukkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman tambah awal		
		Apabila pernyataan belum dituliskan namun pengguna memilih tombol simpan maka akan muncul notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan!"		
		Pengguna dapat kembali ke halaman tambah awal dengan cara memilih tombol ok pada notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"		
		Apabila pernyataan telah ditulis dan pengguna memilih tombol simpan maka data tersimpan dalam *database* dan muncul notifikasi "Fakta pertama berhasil disimpan"		
		Pengguna dapat masuk kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi "Fakta pertama berhasil disimpan"		
3	Keluar halaman tambah fakta	Pengguna dapat tidak menambahkan data awal dengan cara memilih tombol batal pada halaman tambah awal, maka akan muncul halaman *login*		

Tabel 30.Spesifikasi Uji Fitur *Logout*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	*Logout*	Tombol *logout* muncul di setiap halaman setelah pengguna berhasil *login*		
		Pengguna dapat *logout* dengan cara memilih tombol *logout*, maka akan muncul kembali ke halaman *login*		

b. Pengujian *Beta*

Angket digunakan pada tahapan pengujian *beta*untuk mengukur kualitasperangkat lunak yang dikembangkan oleh peneliti. Penelitian ini menggunakan angket untuk pengujian faktor kualitas *usability* pada perangkat lunak yang dikembangkan. Kuesioner yang digunakan untuk pengujian faktor kualitas *usability* ini mengacu pada *Computer System Usability Questionnaire* yang dipublikasi oleh J.R. Lewis. Kuesioner *Computer System Usability*yang dipublikasikan oleh J.R. Lewis tersebut diterjemahkan dan disesuaikan dengan keadaan perangkat lunak yang dikembangkan oleh penulis. Hal ini dilakukan oleh penulis agar proses pengisian kuesioner oleh responden dapat dilakukan dengan baik. Responden mudah mengerti maksud dari setiap pertanyaan pada kuesioner dan tidak menimbulkan kebingungan sehingga responden dapat menjawab setiap pertanyaan pada lembar kuesioner dengan tegas dan jelas.

Kuesioner yang telah disesuaikan tersebut kemudian dibagikan kepada 60 responden yang terdiri dari mahasiswa dan siswa SMA kelas XII. Hal ini dikarenakan *beta test* merupakan pengujian validasi perangkat lunak oleh beberapa pengguna yang dipilih sesuai dengan karakteristik pengguna perangkat lunak nantinya. Dengan kata lain pada uji ini pengembang tidak terlibat dalam rangkaian *testing*. Mahasiswa dan siswa SMA kelas XII merupakan responden yang sesuai dengan karakteristik pengguna aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi.Instrumen angket yang digunakan pada pengujian *beta* untuk pengujian faktor kualitas *usability*pada penelitian ini adalah sebagai berikut:

Tabel 31.*Computer System Usability Questionnaire*

No	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1.	Secara keseluruhan, saya merasa puas dengan kemudahan penggunaan sistem ini.	…	…	…	…	…
2.	Cara penggunaan sistem ini sangat simpel.	…	…	…	…	…
3.	Saya dapat memutuskan pilihan saya dengan efektif ketika menggunakan sistem ini.	…	…	…	…	…
4.	Saya dapat dengan cepat memutuskan pilihan saya menggunakan sistem ini.	…	…	…	…	…
5.	Saya dapat memutuskan pilihan saya dengan efisien ketika menggunakan sistem ini.	…	…	…	…	…
6.	Saya merasa nyaman menggunakan sistem ini.	…	…	…	…	…
7.	Sistem ini sangat mudah dipelajari.	…	…	…	…	…
8.	Saya yakin saya akan lebih percaya diri dengan keputusan saya ketika menggunakan sistem ini.	…	…	…	…	…
9.	Jika terjadi error, sistem ini memberikan pesan pemberitahuan tentang langkah yang saya lakukan untuk mengatasi masalah.	…	…	…	…	…
10.	Kapanpun saya melakukan kesalahan, sistem bisa kembali digunakan dengan cepat	…	…	…	…	…
11.	Informasi yang disediakan sistem ini sangat jelas.	…	…	…	…	…
12.	Mudah untuk menemukan informasi yang saya butuhkan.	…	…	…	…	…
13.	Informasi yang diberikan oleh sistem ini sangat mudah dipahami.	…	…	…	…	…
14.	Informasi yang diberikan sangat efektif dalam membantu memutuskan pilihan saya.	…	…	…	…	…
15.	Tata letak informasi yang terdapat di layar monitor sangat jelas.	…	…	…	…	…
16.	Tampilan sistem ini sangat memudahkan.	…	…	…	…	…
17.	Saya suka menggunakan tampilan sistem semacam ini.	…	…	…	…	…
18.	Sistem ini memberikan semua fungsi dan kapabilitas yang saya perlukan.	…	…	…	…	…
19.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini.	…	…	…	…	…

Keterangan :

SS : Sangat Setuju ST : Setuju RG : Ragu – ragu

TS : Tidak setuju STS : Sangat tidak setuju

G. Teknik Analisis Data

1. Analisis Faktor Kualitas *Correctness*

Faktor kualitas *correctness* dianalisa dengan cara menghitung jumlah *error* tiap *kilo lines of code (KLOC)* pada perangkat lunak yang dikembangkan. Jumlah *lines of code* dapat dihitung menggunakan *Code Line Counter*. Sedangkan jumlah *error* dalam suatu perangkat lunak, dalam hal ini aplikasiVisual Basic, dapat dihitung menggunakan*Code Advisor for Visual Basic 6*yang merupakan *freewaretools* yang dikembangkan oleh *Microsoft*. Jumlah *error/KLOC* yang didapatkan dalam pengujian kemudian dibandingkan dengan standar *error/KLOC* pada *industry average* dan standar *Microsoft Application*. Perangkat lunak dinyatakan lolos uji kualitas *correctness* apabila *error per kilo line of codes* (*error/KLOC*) ≤ (lebih kecil sama dengan) standar*error per kilo line of codes (error/KLOC)* pada *industry average* dan *Microsoft Application*.

2. Analisis Faktor Kualitas *Functionality*

Pengujian faktor kualitas *functionality* dilakukan dengan melakukan tes pada setiap fungsi perangkat lunak. Tes yang dilakukan didokumentasikan dalam *test case*. Setiap *test case* menggambarkan apakah suatu fungsi berjalan sebagaimana mestinya atau tidak.

Berkaitan dengan standar yang digunakan dalam menentukan apakah perangkat lunak telah memenuhi syarat faktor kualitas *functionality*, penulis menggunakan standar *functionality* yang ditetapkan oleh *Microsoft* dalam *Microsoft Certification Logo*.

Tabel 32. Standar kriteria faktor kualitas *functionality* dalam *Microsoft Certification Logo* (Bach, 2005)

Kriteria Lolos	Kriteria Gagal
1. Setiap fungsi primer yang diuji berjalan sebagaimana mestinya.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalan sebagaimana mestinya.
2. Jika ada fungsi pendukung yang tidak berjalan sebagaimana mestinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	2. Jika ada fungsi pendukung yang tidak berjalan sebagaimana mestinya dan itu merupakan kesalahan yang serius dan berpengaruh pada penggunaan normal.

3. Analisis Faktor Kualitas *Usability*

Pengujian faktor kualitas *usability* dilakukan dengan menggunakan metode kuesioner. Kuesioner akan dibagikan kepada 60 responden pengguna Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi. Pengambilan sampel dilakukan dengan teknik *purposive sampling*. Teknik *purposive sampling* yaitu teknik penentuan sampel dengan pertimbangan tententu(Sugiyono, 2011).

Pengguna sistem yang dikembangkan dalam penelitian ini terdiri dari admin dan pengguna umum, oleh karena itu sampel penelitian ini diambil mahasiswa dan siswa SMA kelas XII. Hal tersebut didasari pertimbangan bahwa mahasiswalebih memahami tentang jurusan kuliah dan sistem pendukung keputusan sehingga dapat menjadi pengguna sistem jenis admin.Sedangkan siswa SMA kelas XII diambil dengan pertimbangan bahwa mereka lah yang akan melanjutkan studi ke perguruan tinggi sehingga dapat menjadi pengguna umum sistem.

Penentuan jumlah sampel tersebut berdasarkan pedoman yang dikembangkan oleh Roscoe (1982) yaitu: (Sugiyono, 2011)

a. Ukuran sampel yang layak dalam penelitian adalah antara 30 sampai dengan 500 orang.
b. Apabila sampel didasari dari kategori (seperti pria-wanita, pegawai negeri-pegawai swasta) maka jumlah anggota setiap kategori minimal 30 orang.
c. Pada penelitian multivariate (misalnya korelasi atau regresi ganda) ukuran sampel minimal 10 kali dari jumlah variabel yang akan diteliti.
d. Untuk penelitian eksperimen yang sederhana, yang menggunakan kelompok eksperimen dan kelompok kontrol, ukuran sampel masing-masing 10 sampai dengan 20.

Data yang dihasilkan dari kuesioner tersebut merupakan gambaran pendapat atau persepsi pengguna perangkat lunak, dalam hal ini yang berkaitan dengan faktor kualitas *usability* perangkat lunak yang dikembangkan. Data yang dihasilkan dari kuisioner merupakan data yang bersifat kualititatif. Data tersebut dapat dikonversi ke data kuantatif dalam bentuk data interval atau rasio menggunakan Skala Likert.

Menurut Sugiyono (2011), Skala Likert digunakan untuk mengukur sikap, pendapat atau presepsi seseorang atau kelompok terhadap sesuatu, dalam hal pendapat pengguna terhadap perangkat lunak yang dikembangkan. Data hasil kuesioner yang berupa jawaban-jawaban pengguna terhadap setiap item pertanyaan dalam kuesioner mempunyai gradasi nilai dari sangat positif sampai sangat negatif. Dalam kaitanya dengan kuesioner yang digunakan yaitu, *Computer*

System Usability Questionnaire (CSUQ) yang dikembangkan oleh J.R. Lewis, terdapat 5 macam jawaban dalam setiap item pertanyaan dalam keusioner. Data tersebut diberi skor sebagai berikut :

Tabel 33. Konversi Jawaban Item Kuesioner ke Nilai Kuantitatif

Jawaban	**Skor**
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

Setelah merubah data kualitatif dari jawaban semua responden untuk setiap item pertanyaan pada kuesioner, data tersebut dapat dihitung rata-ratanya. Skor yang didapatkan dari hasil kuesioner tersebut kemudian dihitung nilai rata-rata totalnya dengan membagi jumlah skor dengan jumlah penilai. Dari data yangtelah dikumpulkan, dapatdihitung rata-ratanya dengan rumus sebagai berikut:

$$\bar{X} = \frac{\sum X}{N}$$

Keterangan:

$\bar{X}$: Skor rata-rata

$\sum X$: Jumlah skor

N : Jumlah penilai

Selanjutnya dari data yang diperoleh baik dari mahasiswa maupun siswa SMA kelas XII diubah menjadi nilai kualitatif berdasarkan kriteria penilaian ideal. Ketentuan kriteria penilaian ideal untuk faktor kualitas *usability*ditunjukkan dalam tabel34 berikut ini:

Tabel 34.Kriteria Kategori Penilaian Ideal

Rentang skor kualitatif	Nilai
$\bar{X} > M_i + 1{,}8\ SB_i$	Sangat Baik
$M_i + 0{,}6\ SB_i < \bar{X} \leq M_i + 1{,}8\ SB_i$	Baik
$M_i - 0{,}6\ SB_i < \bar{X} \leq M_i + 0{,}6\ SB_i$	Cukup
$M_i - 1{,}8\ SB_i < \bar{X} \leq M_i - 0{,}6\ SB_i$	Kurang
$\bar{X} \leq M_i - 1{,}8\ SB_i$	Sangat Kurang

(Sukarjo, 2006, p. 53)

Keterangan:

M_i : rata-rata ideal

$$M_i = \frac{1}{2} x(\textit{skor maksimal ideal} + \textit{skor minimum ideal})$$

SB_i : simpangan baku ideal

$$SB_i = (\frac{1}{2} . \frac{1}{3})\ x(\textit{skor maksimal ideal - skor minimumideal})$$

(Sukarjo, 2006, p. 52)

Skor maksimal ideal : $\sum$ butir kriteria x skor tertinggi

Skor minimum ideal : $\sum$ butir kriteria x skor terendah

Diketahui pada penelitian ini skor tertinggi adalah 5 dan skor terendah adalah 1.Jumlah butir kriteria dalam kuesioner adalah 19. Maka dapat dihitung berdasarkan rumus diatas :

Skor maksimal ideal =$\sum$ butir kriteria x skor tertinggi

= 19 x 5 = 95

Skor minimum ideal = $\sum$ butir kriteria x skor terendah

= 19 x 1 = 19

$M_i = \frac{1}{2}$x(*skor maksimal ideal + skor minimum ideal*)

$= \frac{1}{2}$x(95 + 19)

= 57

$SB_i = (\frac{1}{2} . \frac{1}{3})$ x(*skor maksimal ideal - skor minimumideal*)

$= (\frac{1}{2} . \frac{1}{3})$ x(95 - 19)

= 12,67

Kelayakan aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi pada faktor kualitas *usability*ini ditentukan dengan cara menghitung rata-rata nilai yang diperoleh dari jawaban kuesioner60 responden. 60 responden yang mengisi kuesioner untuk aspek kualitas *usability* ini terdiri dari mahasiswa dan siswa SMA kelas XII. Nilai rata-rata dari jawaban kuesioner 60 responden kemudian dibandingkan dengan tabel kategori penilaian faktor kualitas *usability*.Kriteria kelayakan tersebut dapat dilihat pada tabel 35.

Tabel 35. Kategori Penilaian Faktor Kualitas *Usability*

Rentang skor kualitatif	**Nilai**
$\bar{X}$>79,8	Sangat Layak
64,6<$\bar{X} \leq$ 79,8	Layak
49,4<$\bar{X} \leq$ 64,6	Cukup Layak
34,2<$\bar{X} \leq$ 49,4	Tidak Layak
$\bar{X} \leq$ 34,2	Sangat Tidak Layak

4. Analisis Faktor Kualitas *Maintainability*

Pengujian aspek *maintainability* dapat dilakukan secara dinamis dalam arti bahwa prosedur *maintainability* ditetapkan, dijalankan dan dibandingkan dengan persyaratan (Hass, 2008). Oleh karena itu, pengujian aspek *maintainability* pada penelitian yang penulis lakukan difokuskan untuk menjawab pertanyaan atas ukuran-ukuran (*metrics*) yang berhubungan dengan faktor kualitas *maintainability*.

Aspek *maintainability*yang diuji adalah *instrumentation*, *consistency*, dan *simplicity*. Hasil pengujian Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi dibandingkan dengan standar kriterialolos untuk masing-masing aspek *maintainability*. Sistem dikatakan lolos jika memenuhi standar uji faktor kualitas *maintainability*.

Tabel 36. Standar Kriteria Faktor Kualitas *Maintainability*

Aspek	Aspek yang Dinilai	Kriteria Lolos
Instrumentation	Terdapat peringatan pada sistem pengolah data untuk mengidentifikasi kesalahan	Ketika ada kesalahan yang dilakukan oleh *user*, maka sistem akan mengeluarkan peringatan untuk mengidentifikasi kesalahan.
Consistency	Penggunaan satu bentuk rancangan pada seluruh rancangan sistem	Bentuk rancangan sistem pengo-lah data mempunyai satu bentuk yang sama. Hal ini dapat dilihat pada bagian implementasi sistem.
Simplicity	Kemudahan dalam pengelolaan, perbaikan, dan pengembangan sistem	Mudah untuk dikelola, diperbaiki, dan dikembangkan. Hal ini dapat dilihat pada tahapan-tahapan proses penulisan kode program.

BAB IV
HASIL DAN PEMBAHASAN

A. Pengembangan Perangkat Lunak

Salah satu tujuan dari penelitian ini adalah untuk mengembangkan produk perangkat lunak yaitu aplikasi Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi.Proses pengembangan perangkat lunak dalam penelitian ini mengacu pada *softwareengineering*. Penulis mengembangkan perangkat lunak menggunakan salah satu model proses klasik yaitu *waterfall'smodel*. *Waterfall's model* merupakan suatu *processmodel* dalam mengembangkan perangkat lunak yang memiliki sifat sistematis, berurutan dalam membangun suatu *software*(Pressman, 2001, hal. 29). Mengacu pada model tersebut terdapat 4 tahap pada proses pengembangan perangkat lunak. Beberapa tahap yangdilakukan dalam mengembangkan produk perangkat lunak pada penelitian ini antara lain :

1. Analisa Kebutuhan (*Requirement Analysis*)

Proses pertama yang dilakukan dalam mengembangkan perangkat lunak adalah analisa kebutuhan. Analisa kebutuhan dilakukan agar segala kebutuhan mengenai pengembangan perangkat lunak dapat dirumuskan sebagai dasar pengembangan perangkat lunak nantinya, sehingga dapat memenuhi segala kebutuhan pengguna.

Proses analisa kebutuhan dilaksanakan dengan mencari dan mengkaji informasi mengenai perangkat lunak yang akan dikembangkan baik secara langsung, maupun dari media cetak dan elektronik. Analisa kebutuhan yang dilakukan oleh penulis antara lain :

a. Observasi

Observasi merupakan salah satu metode yang digunakan penulis untuk melakukan analisa kebutuhan. Observasi dilakukan untuk mengetahui apa saja fitur yang dibutuhkan oleh pengguna, kaitanya dengan pengembangan perangkat lunakSistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi.Observasi juga dilakukan terhadap perangkat lunak serupa yang sudah ada. Dengan melakukan observasi pada beberapa poin diatas maka didapatkan pendekatan bahwa :

1) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi merupakan *expert system* yang dibuat menggunakan Microsoft Visual Basic 6.0 dengan *database* MySQL.
2) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur *login* untuk membedakan hak akses antara *admin* dan*normal user*.
3) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur pendaftaran untuk pengguna yang belum terdaftar dan ingin menggunakan aplikasi ini.
4) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur bantuan (*help*) di setiap halaman untuk memberi petunjukcara penggunaan halaman terkait.
5) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur tentang (*about*)yang menyediakan informasi tentang aplikasi ini.

6) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur diagnosa yang dapat diakses oleh semua jenis pengguna (*admin* dan *normal user*) yang telah berhasil *login*maupun berhasilmelakukan pendaftaran. Fitur ini merupakan fitur utama dalam aplikasi ini, yaitu fitur yang digunakan untuk proses konsultasi pengguna dalam mencari jurusan yang sesuai dengan jenis kecerdasannya.
7) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur penjelas (kenapa?) dalam proses diagnosa. Fitur ini berfungsi untuk memberikan penjelasan yang berupa keterangan, yaitu penjelasan atas pertanyaan mengapa pernyataan tersebut diajukan dan mengapa konklusi tersebut dihasilkan.
8) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur detail info yang dapat diakses ketika proses diagnosa telah mencapai konklusi. Fitur ini berfungsi untuk memberikan detail info mengenai daftar Perguruan Tinggi Negeri di Indonesia yang menyediakan jurusan yang menjadi hasil diagnosa pengguna.
9) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur perubahan *knowledge base*hanya untuk pengguna dengan hak akses*admin*. Fitur perubahan *knowledge base*tersebut meliputi tambah awal, ubah fakta, tambah fakta, sisip fakta, hapus fakta, dan kosongkan *rulebase.*
10) Perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi memiliki fitur *logout*.

b. Studi Literatur

Observasi yang dilakukan untuk mendukung landasan pembuatan perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi adalah studi literatur. Studi literatur digunakan untuk mencari data mengenai konsep-konsep dan teori-teori yang sudah ada untuk mendukung pengembangan perangkat lunak yang dikembangkan oleh peneliti.

Studi literatur yang dilakukan oleh peneliti antara lain mendapati konsep teori sebagai berikut :

1) Pemilihan jurusan di Perguruan Tinggi menggunakan pengetahuan mengenaiteori kepribadian RIASEC Holland dari buku karangan Winkel dan Osipow.
2) Penyusunan *knowledge base* menggunakan metode representasi pengetahuan kaidah produksi, yaitu pohon keputusan (*decision tree*) dengan teknik inferensi *forward chaining*(runut maju).
3) Daftar jurusan dan penyedia jurusan diambil dari 61 Perguruan Tinggi Negeri yang ada di Indonesia.
4) Daftar *Holland Three Digit Code* untuk setiap jurusan diambil dari berbagai sumber yang mengacu pada *The Occupational Information Network* (O*NET) yang digunakan oleh *US Department of Labor/Employment and Training Administration*. Sumber tersebut berupa penelitian dari beberapa lembaga atau psikolog dan universitas di luar negeri yang mencantumkan *Holland Three Digit Code*di setiap jurusan yang disediakan.

5) Aplikasi yang dikembangkan menggunakan *database*MySQL sebagai media penyimpanan data r*ulebase* dan data pengguna.

Selain kajian literatur mengenai konsep pengembangan pemilihan jurusan di perguruan tinggi dari sisi fitur dan pengkodean, dilakukan kajian terhadap *user interface* perangkat lunak juga. Dengan dilakukannya kajian literatur sebelum proses pengembangan perangkat lunak diharapkan akan mempermudah dan mendasari tahapan-tahapan pengembangan perangkat lunakSistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi.

2. Desain (*Design*)

Tahapan ke-dua dalam *waterfall's model process* ialah proses desain pengembangan perangkat lunak. Desain pengembangan perangkat lunak adalah proses multilangkah yang fokus pada desain pembuatan program. Fokus desain pada proses pengembangan perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggiini meliputi desain sistem, desain basis data, perancangan *knowledge base*, desain antar muka (*interface*), desain arsitektur, dan desain prosedural.

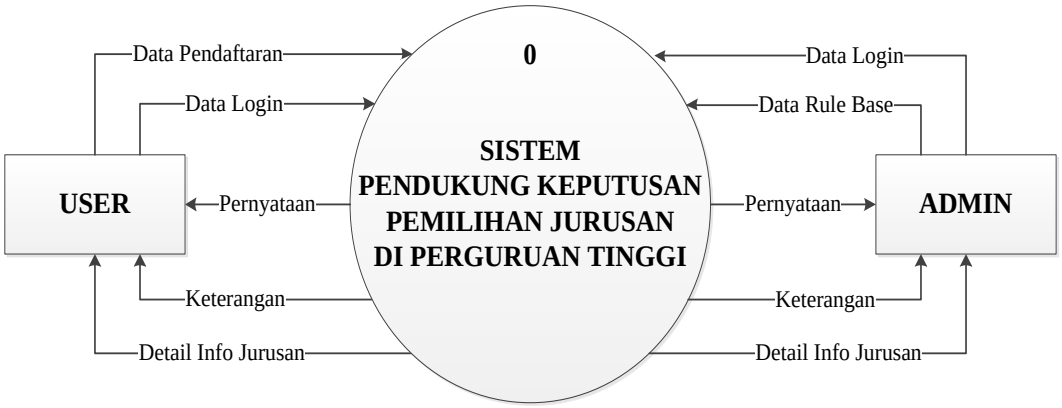
a. Desain Sistem dengan DFD (*Data Flow Diagram*)

Desain model yang digunakan nantinya harus sesuai dengan karakteristik perangkat lunak yang dikembangkan dan tentunya mempermudah pengembang dalam mengembangkan perangkat lunak. Penelitian ini menggambarkan desain sistem perangkat lunak dengan metode *Data Flow Diagram (DFD)*.Langkah awal dalam pembuatan *data flow diagram*atau diagram alir data yaitu dengan membuat

diagram konteks secara keseluruhan, seperti pembuatan *data flow diagram level* 0, *data flow diagram level* 1 dan seterusnya hingga proses tidak dapat diuraikan lagi.

1) *Data Flow Diagram Level 0*

Pada *data flow diagram level 0* ini terdapat dua entitas yaitu *user* sebagai pengguna sistem (pengguna umum) dan *admin* sebagai pengelola sistem. *Normal user* memberi masukan kepada sistem berupa data pendaftaran dan data *login*. Sedangkan *admin* memberi masukan kepada sistem berupa data *login* dan data *rule base*. Sebagai timbal balik dari masukan entitas maka sistem memberi keluaran berupa pernyataan, keterangan dan detail info jurusan kepada *normal user* dan *admin*.

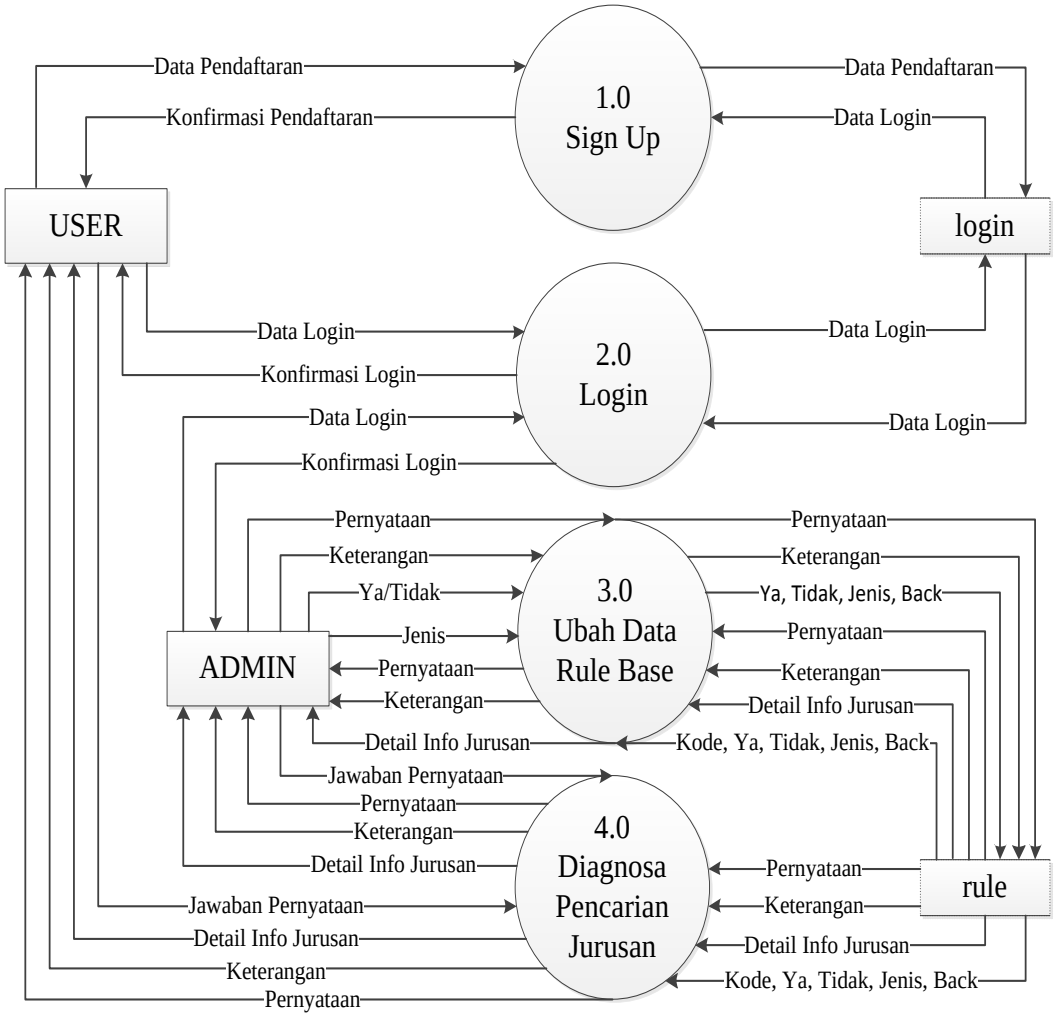


Gambar 11. DFD Level 0

2) *Data Flow Diagram Level 1*

*Data flow diagram level 1*merupakan representasi dari *data flow diagram level 0*yang sudah dipartisi untuk memberikan penjelasan yang lebih detail. DFD Level 1 memperlihatkan proses utama pembentuk proses sistem. Proses pada DFD Level 1 perangkat lunakSistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi dibagi menjadi 4 yaitu, *Sign Up*, *Login*, Ubah Data *Rule Base*,

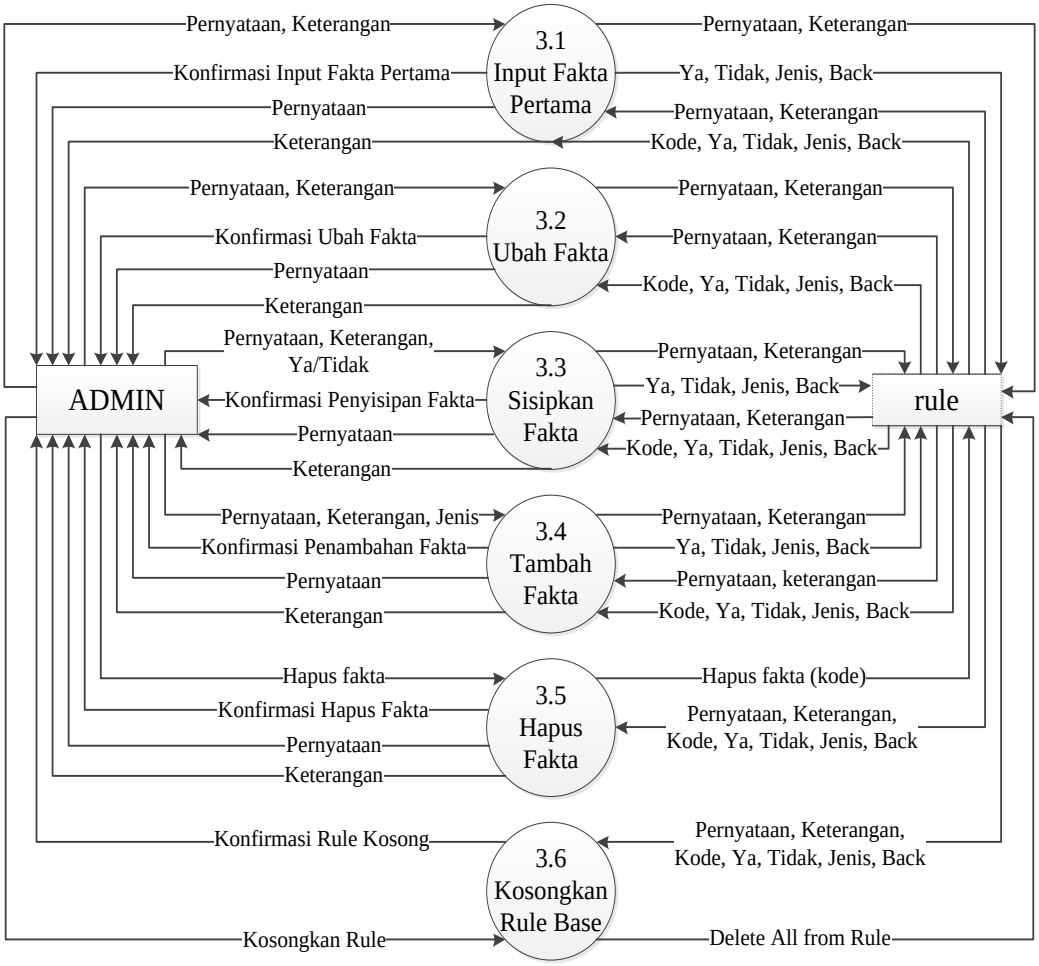
dan Diagnosa Pencarian Jurusan.*Data Flow Diagram Level 1* dapat dilihat pada gambar 12.



Gambar 12. DFD Level 1

Kemudian dari *data flow diagram level 1*yang dapat dilihat pada gambar diatas, proses Ubah Data *Rule Base* dapat dikembangkan menjadi *Data Flow Diagram* Level 1 Proses 3.0. Proses Ubah Data *Rule Base* diturunkan menjadi 6 proses yaitu, Input Fakta Pertama, Ubah Fakta, Sisipkan Fakta, Tambah Fakta, Hapus Fakta, dan Kosongkan *Rule Base*. Proses 3.0 hanya mempunyai 1 entitas

yaitu *admin*, dan hanya memiliki 1 tabel yaitu tabel*rule*. DFD Level 1 Proses 3.0 dapat dilihat pada gambar 13.



Gambar 13. DFD Level 1 Proses 3.0

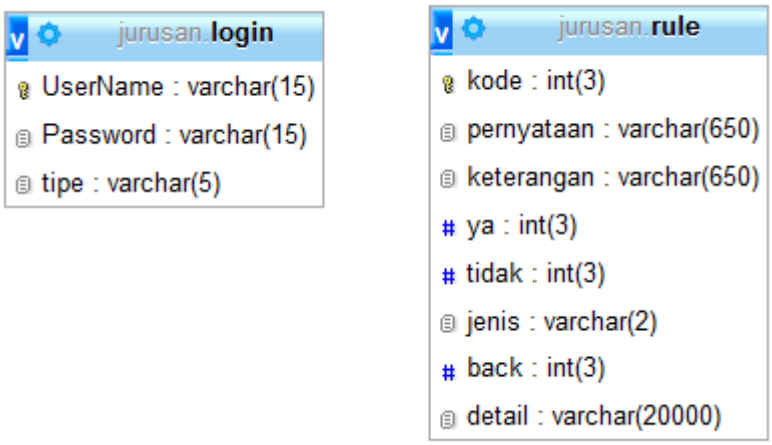
b. Desain Basis Data

Sebagai sarana untuk mempermudah proses pembaruan data pada aplikasi Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi ini maka digunakan *database* sebagai media penyimpanan data. Perancangan basis data aplikasi Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi didasarkan pada *data flow diagram* yang telah dirancang sebelumnya.

Berdasarkan proses serta interaksi antar entitas maka dibuatlah tabel-tabel yang nantinya akan digunakan sebagai *data storage* perangkat lunak.

1) *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah diagram bantu untuk menggambarkan hubungan antar entitas dalam basis data. Diagram hubungan entitas digambarkan sebagai sekumpulan tabel yang saling berelasi dalam basis data berdasarkan hubungan entitas yang ditentukan oleh atribut-atributnya. *Entity Relationship Diagram*(ERD) untuk aplikasi yang dikembangkan dapat dilihat pada gambar 14.



Gambar 14. *Entity Relationship Diagram*

2) Kamus Data

Kamus data dari masing-masing tabel yang terdapat pada basis data aplikasi Sistem Pendukung Keputusan Pemilihan Jurusandi Perguruan Tinggi yang dikembangkan dalam penelitian ini adalah sebagai berikut:

a) Tabel rule

Tabel rule merupakan tabel yang digunakan untuk menyimpan data pohon keputusan (*decision tree*) yang digunakan dalam proses diagnosa.Data pohon keputusan merupakan data utama dari aplikasi ini karena dengan data tersebut aplikasi ini dapat melakukan konsultasi dengan pengguna. Kolom kode pada tabel rule dibuat sebagai *primary key* dan menggunakan *auto increment* agar penulisan kode dilakukan otomatis oleh *database*.

Tabel 37. Kamus Data Tabel Rule

Field	**Type**	**Null**	**Key**	**Default**
kode	int (3)	No	Primary	*None*
pernyataan	varchar (650)	No	-	*None*
keterangan	varchar (650)	No	-	*None*
ya	int (3)	No	-	*None*
tidak	int(3)	No	-	*None*
jenis	varchar(2)	No	-	*None*
back	Int(3)	No	-	*None*
detail	varchar (20000)	No	-	*None*

b) Tabel login

Tabel login merupakan tabel yang digunakan untuk menyimpan data pengguna, baik itu pengguna umum maupun admin.Data yang terdapat pada tabel login adalah *username* (nama pengguna) yang merupakan *primary key*, *password* (kata sandi), dan tipe yang digunakan untuk membedakan jenis pengguna umum dengan admin.

Tabel 38. Kamus Data Tabel Login

Field	**Type (Length)**	**Null**	**Key**	**Default**
username	varchar (15)	No	Primary	*None*

password	varchar (15)	No	-	*None*
tipe	varchar (5)	No	-	*None*

c. Knowledge Base

Untuk membuat sebuah *expert system* diperlukan *knowladge base* yang digunakan sebagai aturan jalannya diagnosa sistem. Penjabaran kebutuhan data untuk *knowladge base* dalam aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi ini menggunakan metode akuisisi pengetahuan. Akuisisi pengetahuan merupakan suatu proses untuk mengumpulkan data pengetahuan akan suatu masalah dari pakar pada bidang tersebut. Pengetahuan tersebut dapat dikumpulkan melalui berbagai cara contohnya melalui wawancara dari seorang pakar, dari buku, dari jurnal, dari majalah, artikel dari internet dan lain sebagainya.

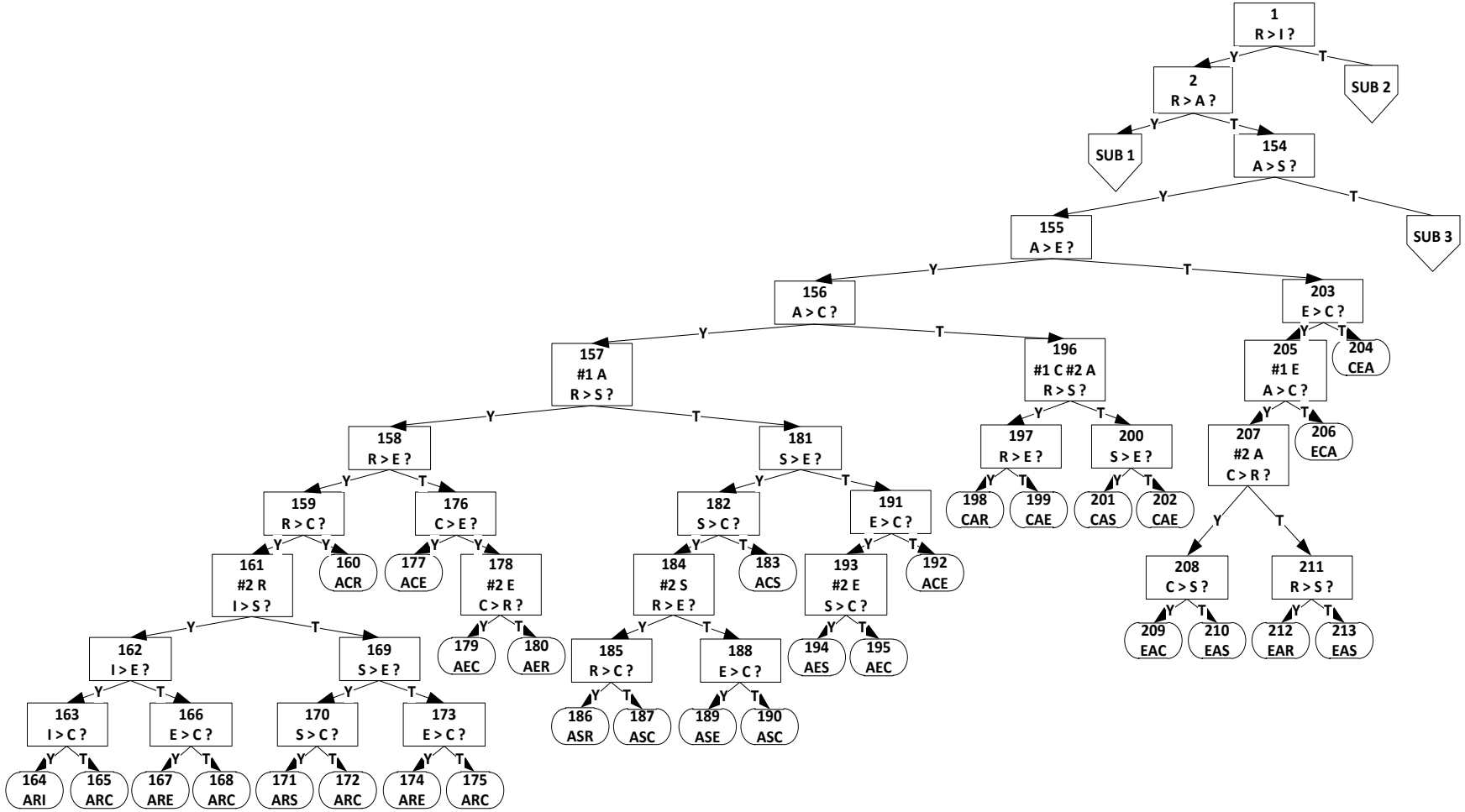
Data pengetahuan yang digunakan dalam penelitian ini adalah teori kepribadian RIASEC yang dikemukakan oleh psikolog John L. Holland. Data pengetahuan lain yang berkaitan dengan pengembangan perangkat lunak ini adalah data jurusan dari 61 Perguruan Tinggi Negeri di Indonesia yang masing-masing memiliki 3 kode dominan Holland.. *Tree digit code* Holland untuk setiap jurusan tersebut diambil dari berbagai penelitian yang mengacu pada teori kepribadian Holland. *Rule base* dirancang dengan pendekatan *Rule-Based Reasoning* dan menggunakan metode inferensi *forward chaining*.

Rancangan *rule base* menggunakan kode huruf yang menggambarkan setiap tipe kepribadian (R, I, A, S, E, dan C) untuk menyingkat pernyataan. R mewakili pernyataan “kegiatan di luar ruangan, bekerja dengan objek, mesin, peralatan,

tumbuhan atau hewan dan mempunyai kemampuan atletik". I mewakili pernyataan "melakukan observasi, mempelajari, menginvestigasi, menganalisis, mengevaluasi atau memecahkan persoalan perhitungan/penelitian". A mewakili pernyataan "bekerja di situasi tak terduga dengan menggunakan imajinasi dan kreativitas atau ber jiwa seni, berinovasi atau kemampuan intuitif (memahami sesuatu tanpa melalui penalaran rasional)". S mewakili pernyataan "bekerjasama dengan orang lain untuk memberi pencerahan, memberi informasi, memberikan pertolongan, melatih, merawat orang lain atau cakap berbicara". E mewakili pernyataan "bekerjasama dengan banyak orang, mempengaruhi, membujuk, memimpin/mengelola tujuan organisasi, atau mendapat keuntungan finansial". C mewakili pernyataan "bekerja dengan data, administrasi dan pengolahan angka, melaksanakan tugas secara rinci, atau menindaklanjuti instruksi orang lain".

Rule base dirancang untuk membandingkan setiap tipe kepribadian,antara satu tipe kepribadian dengan tipe kepribadian lainnya dibandingkan sehingga menemukan 3 kode dominan Holland untuk jurusan yang cocok dengan pengguna. Pada *database* dan program, kode yang mewakili pernyataan ditulis lengkap. Sebagai salah satu contoh pada kode 1 *rule base* ditulis pernyataan "R > I ?", maka pada *database* disimpan pernyataan dengan kode 1 yaitu "Apakah anda LEBIH MENYUKAI kegiatan di luar ruangan, bekerja dengan objek, mesin, peralatan, tumbuhan atau hewan dan mempunyai kemampuan atletik DARIPADA melakukan observasi, mempelajari, menginvestigasi, menganalisis, mengevaluasi atau memecahkan persoalan perhitungan/penelitian?. Berikut ini adalah potongan

rule base yang telah dirancang untukSistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi:



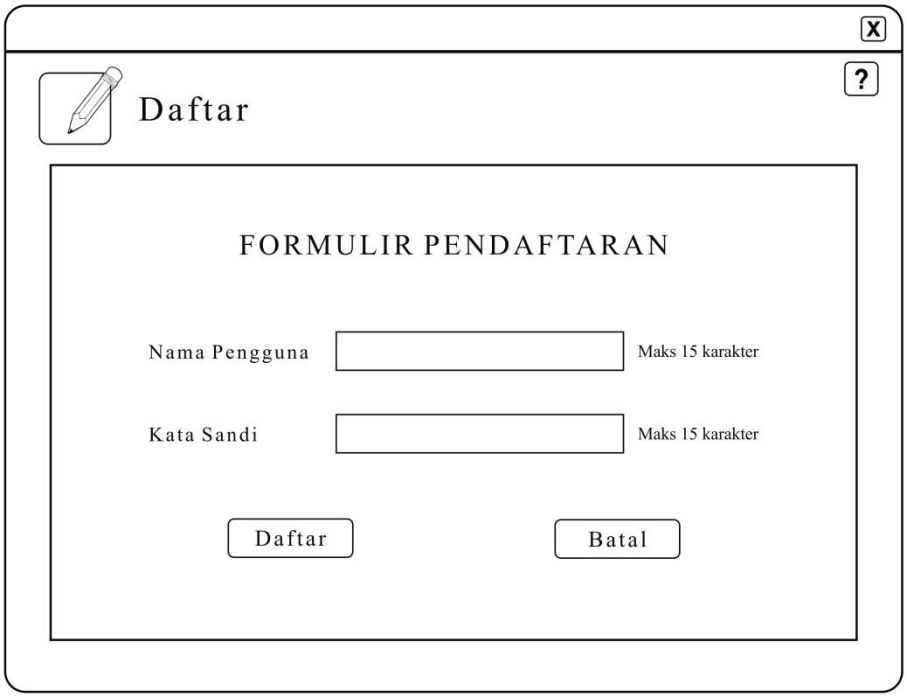
Gambar 15. Potongan *Rule Base*

d. Desain Antar Muka

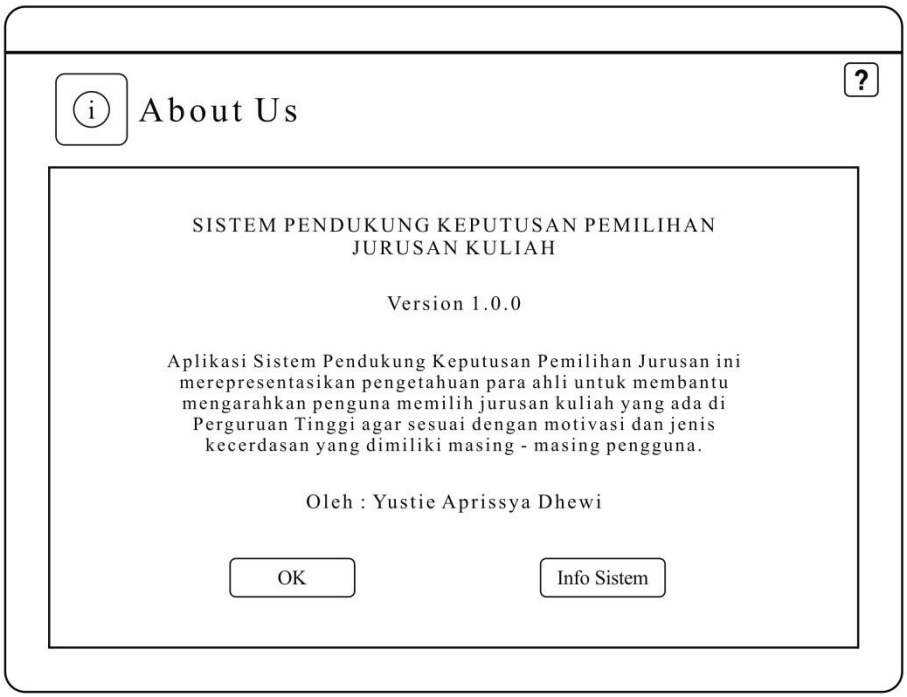
*Interfacedesign*atau yang biasa disebut rancangan antarmuka berfungsi untuk memudahkan pengguna berinteraksi dengan sistem. Rancangan antarmuka yang baik dapat membuat pengguna mudah mengerti dan nyaman dalam penggunaan perangkat lunak tersebut. Dalam rancangan antarmuka ini segala kesulitan yang ada pada sistem akan disembunyikan sehingga pengguna hanya mendapatkan tampilan yang bersifat interaktif, komunikatif, menarik dan mudah dalam pemakaiannya.

Berikut rancangan antar muka(*interfacedesign*) dari Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi yang penulis rancang melalui *storyboard* :

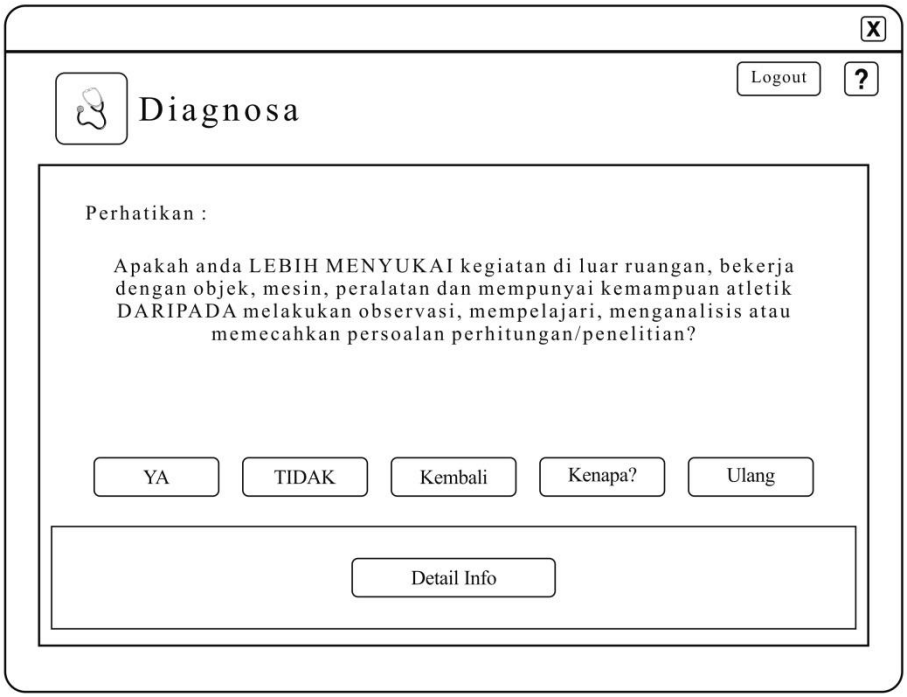
Gambar 16.*Storyboard* Halaman *Login*



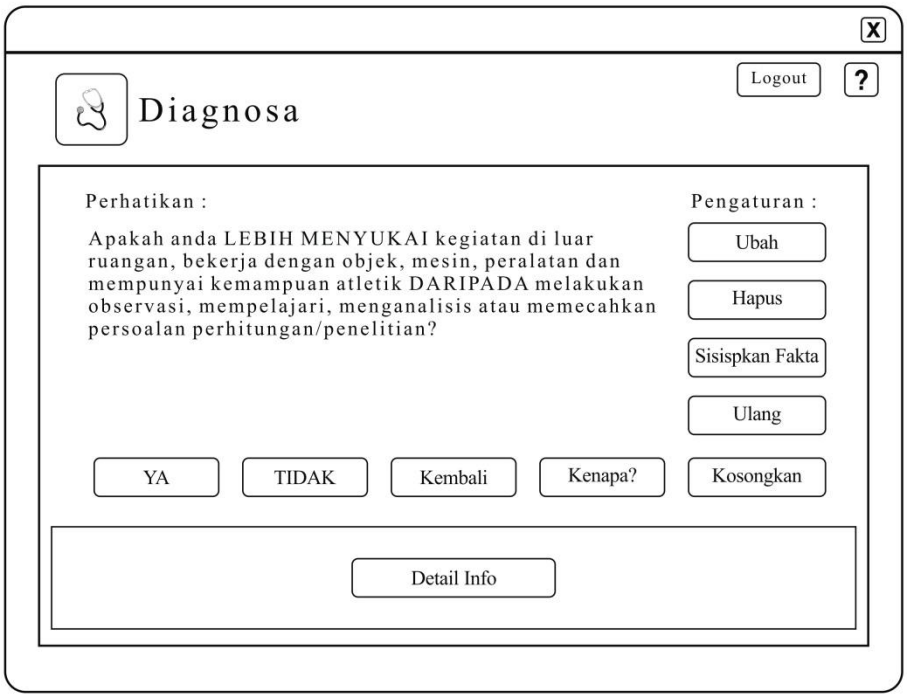
Gambar 17. *Storyboard* Halaman Pendaftaran



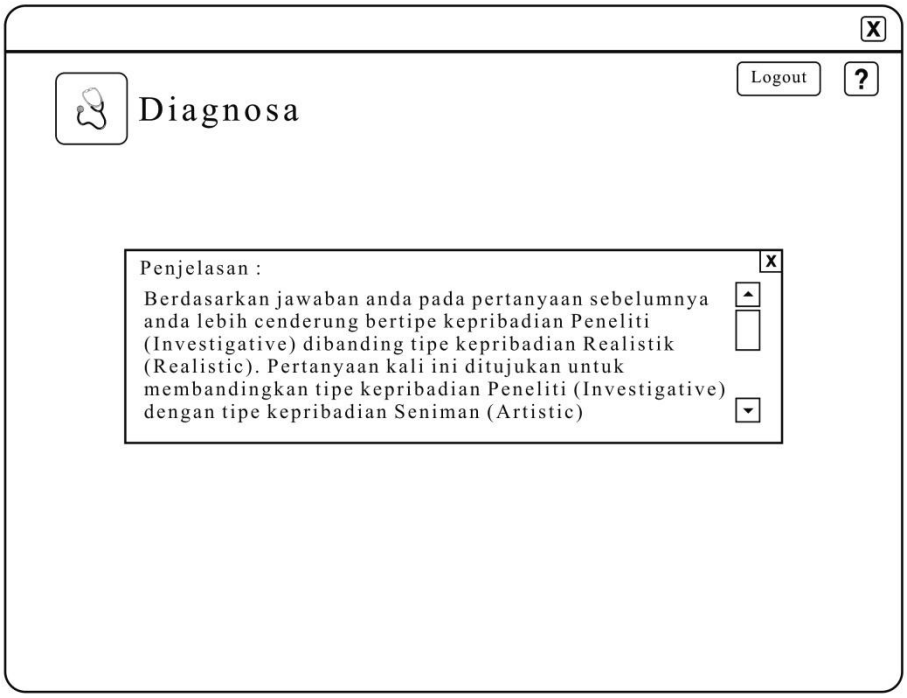
Gambar 18. *Storyboard* Halaman *About* (Tentang)



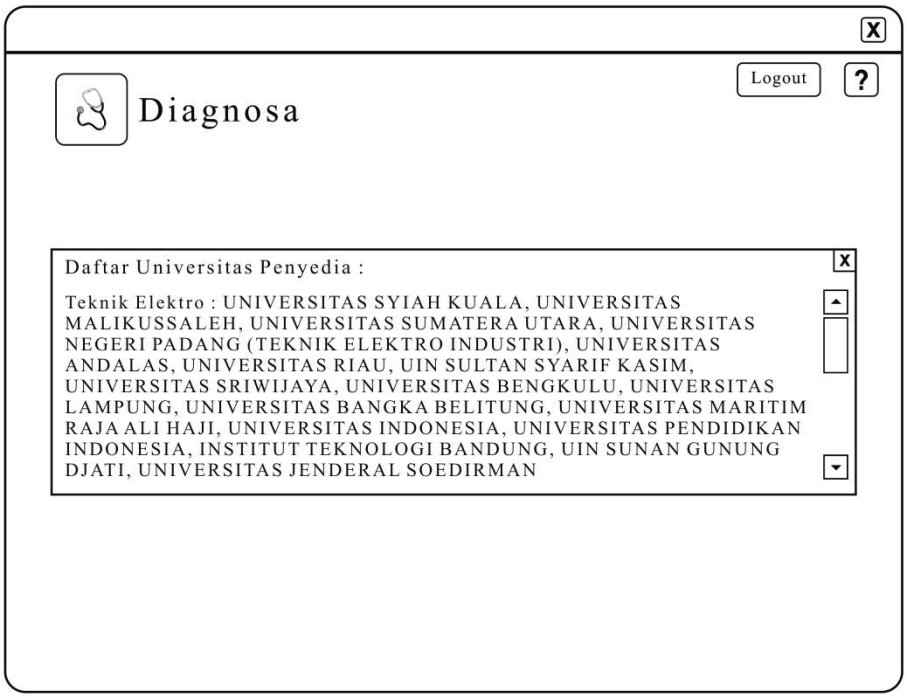
Gambar 19. *Storyboard* Halaman Diagnosa Untuk *Normal User*



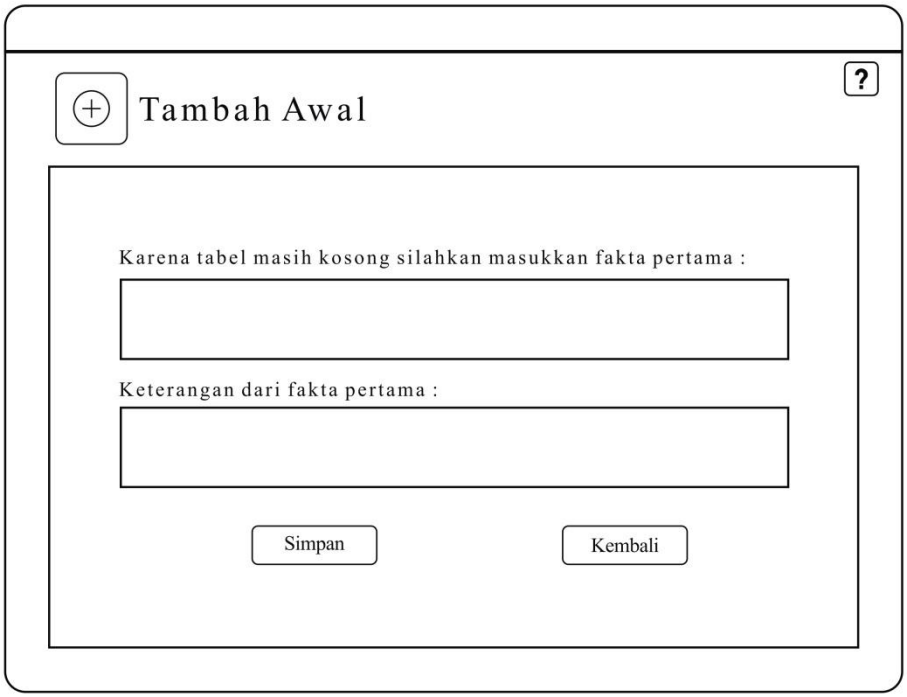
Gambar 20. *Storyboard* Halaman Diagnosa Untuk *Admin*



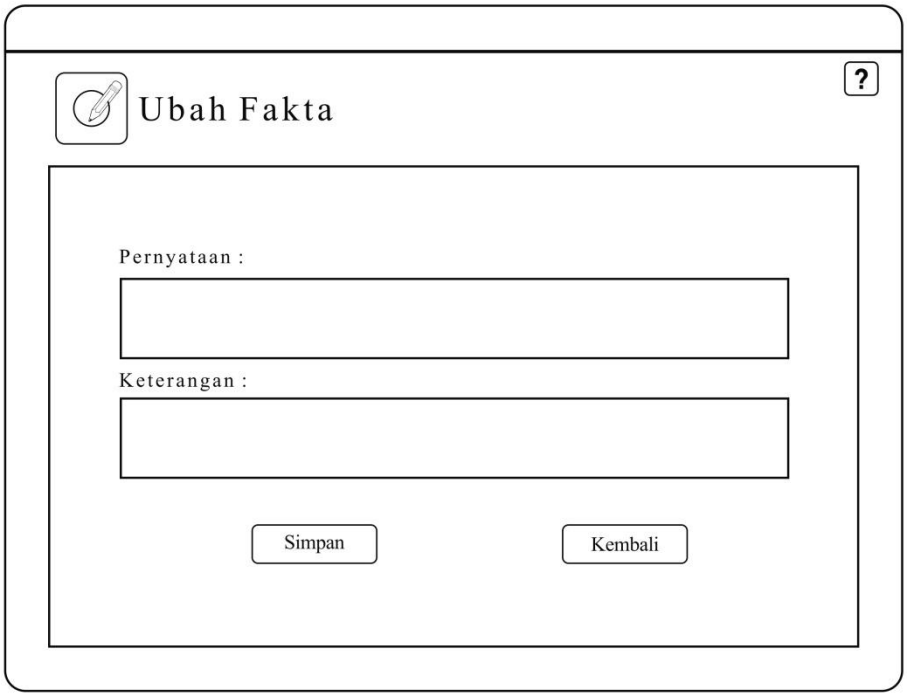
Gambar 21. *Storyboard* Fitur Penjelas (Kenapa?)



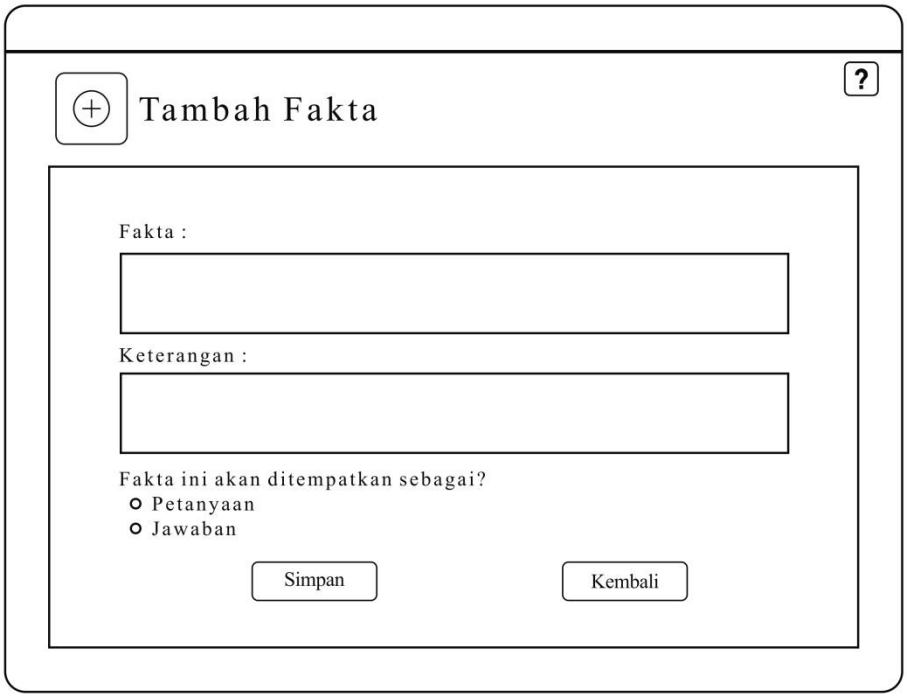
Gambar 22. *Storyboard* Fitur Detail Info



Gambar 23. *Storyboard* Halaman Tambah Awal

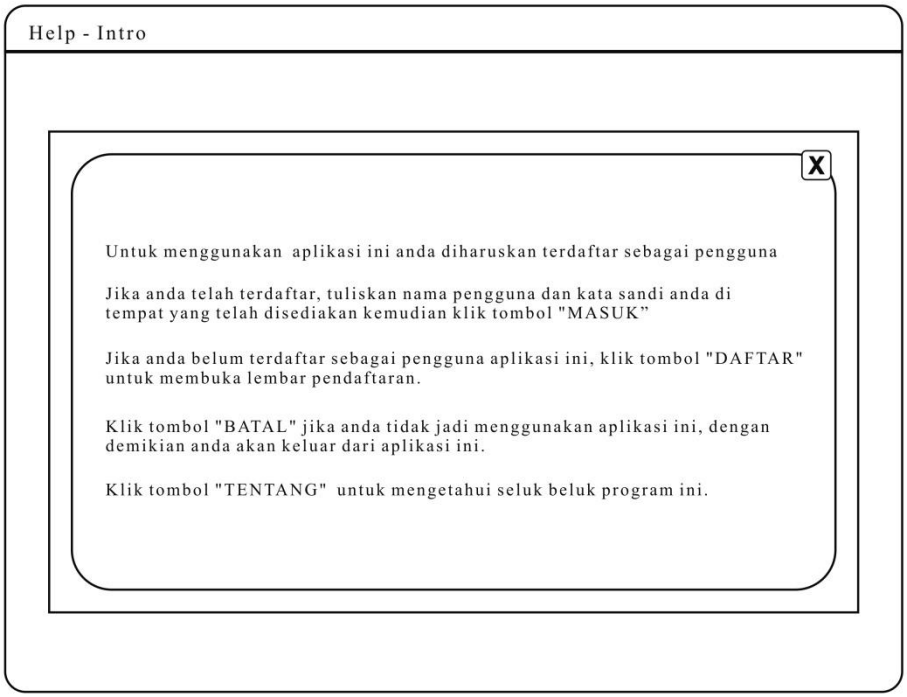


Gambar 24. *Storyboard* Halaman Ubah Fakta

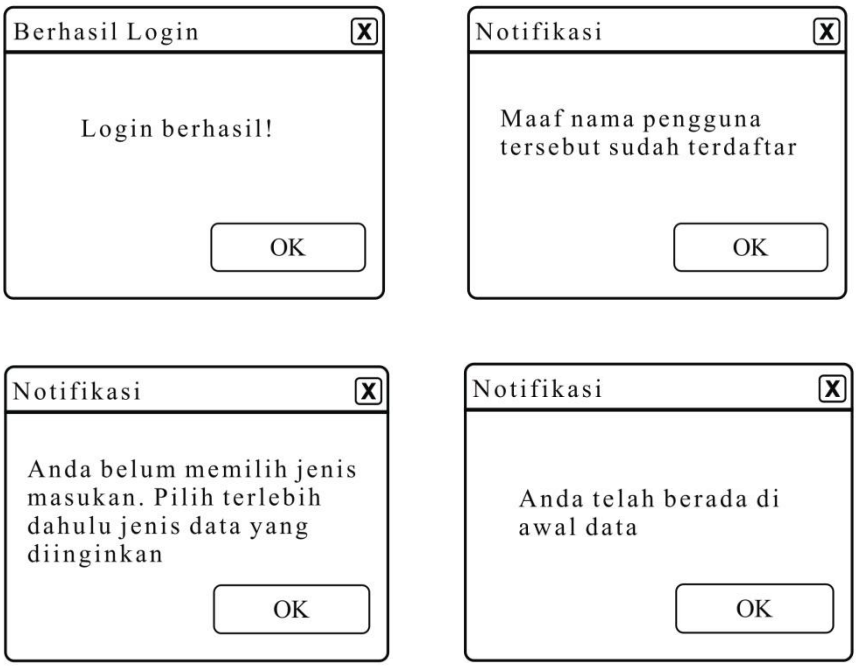


Gambar 25. *Storyboard* Halaman Tambah Fakta

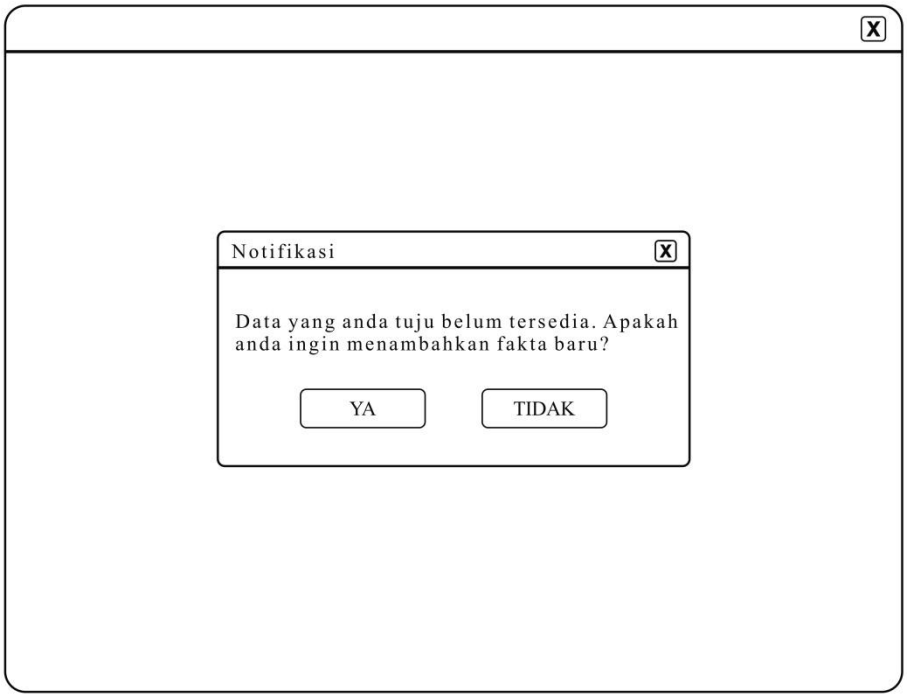
Gambar 26. *Storyboard* Halaman Sisipkan Fakta



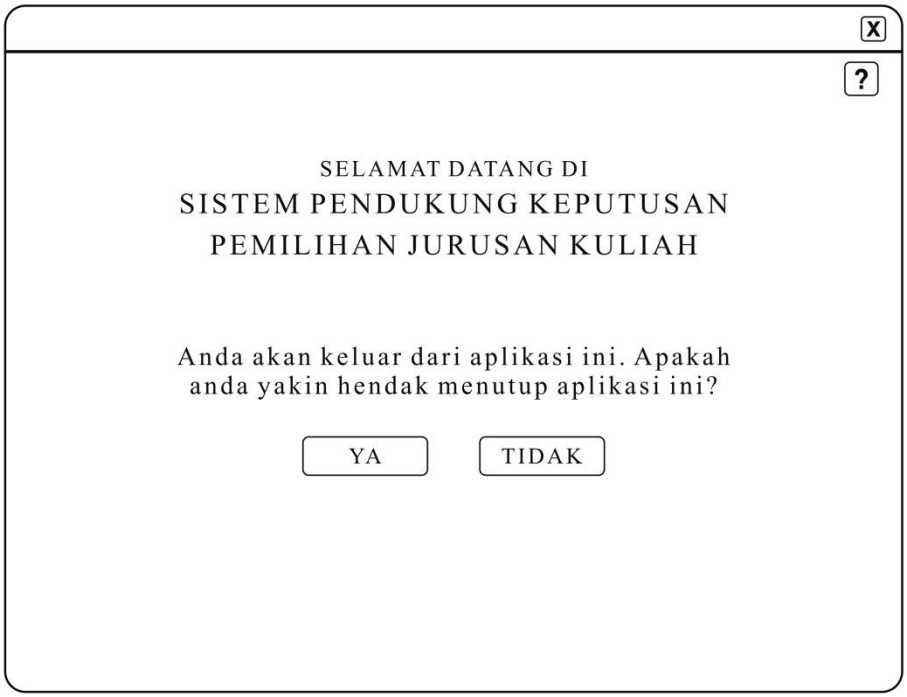
Gambar 27. Desain Halaman Bantuan (*Help*)



Gambar 28. DesainNotifikasi Sistem



Gambar 29. *Storyboard* Notifikasi Tambah Fakta



Gambar 30. *Storyboard* Notifikasi Keluar Aplikasi

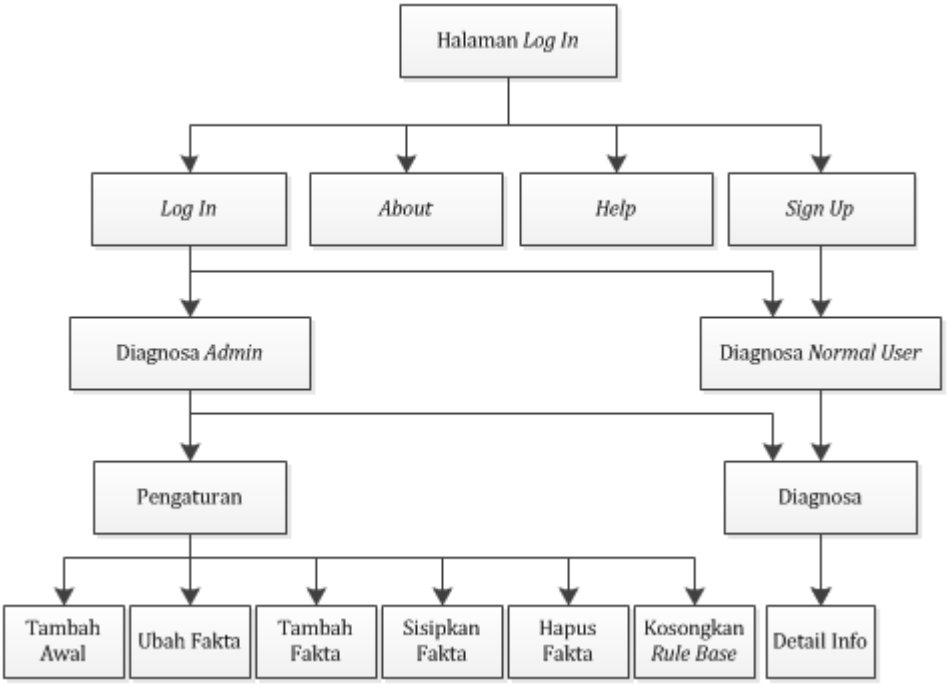
e. Desain Arsitektur

Desain arsitektur merupakan sebuah gambaran untuk menjelaskan struktur modul program dan merepresentasikan relasi antar modul. Deskripsi fungsi dari modul-modul yang dikembangkan dalamSistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi adalah sebagai berikut:

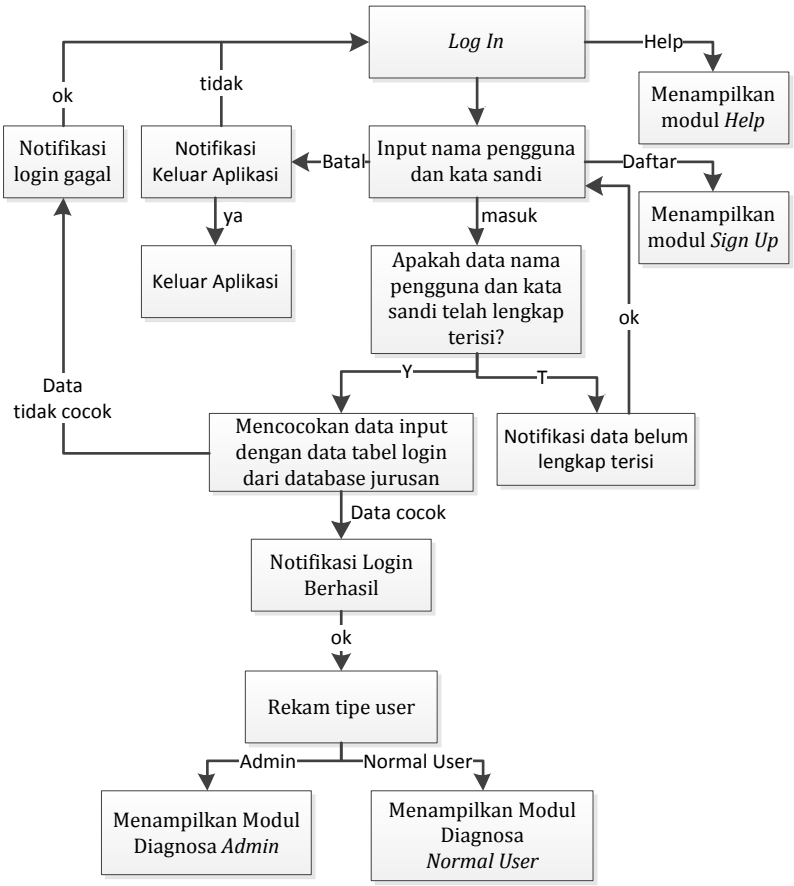
1) *Login* digunakan untuk verifikasi pengguna yang hendak menggunakan aplikasi ini sehingga pihak yang tidak berhak mengakses tidak bisa masuk. Selain itu *login* juga digunakan untuk membedakan hak akses antara*admin* dan *normal user*.
2) *Signup*digunakan untuk pengguna yang hendak menggunakan aplikasi ini namun belum terdaftar sebagai pengguna aplikasi ini.
3) *About* digunakan untuk memberi info tentang aplikasi ini
4) *Help* digunakan untuk memberi info cara penggunaan aplikasi
5) Diagnosa*Admin*digunakan untuk mendiagnosa tipe pengguna *admin* dengan menampilkan serangkaian pertanyaan yang nantinya akan memunculkan solusi di akhir diagnosa dan dapat melakukan perubahan pada *rule base*
6) Diagnosa *Normal User* digunakan untuk mendiagnosa tipe pengguna *user* dengan menampilkan serangkaian pertanyaan yang nantinya akan memunculkan solusi di akhir diagnosa
7) Detail Info digunakan untuk menampilkan detail info mengenai jurusan yang dikeluarkan sebagai hasil diagnosa.

8) Hapus Fakta digunakan untuk menghapus fakta yang sedang ditampilkan beserta anak-anak faktanya yang terhubung dalam *rule base*. Menu ini hanya dapat diakses oleh *admin*.
9) Ubah Fakta digunakan untuk mengubah fakta yang sudah ada dalam *rulebase*. Menu ini hanya dapat diakses oleh *admin*.
10) Tambah Fakta digunakan untuk menambahkan fakta baru di ujung *rule base*. Menu ini hanya dapat diakses oleh *admin*.
11) Sisipkan Fakta digunakan untuk menyisipkan fakta baru di antara fakta yang telah ada dalam *rulebase*. Menu ini hanya dapat diakses oleh *admin*.
12) Tambah Awal digunakan untuk mengisikan fakta pertama ke dalam *rule base* saat *rule base* dalam keadaan masih kosong.Menu ini hanya dapat diakses oleh *admin*.
13) Kosongkan digunakan untuk menghapus semua fakta yang ada pada *rulebase*. Menu ini hanya dapat diakses oleh *admin*.

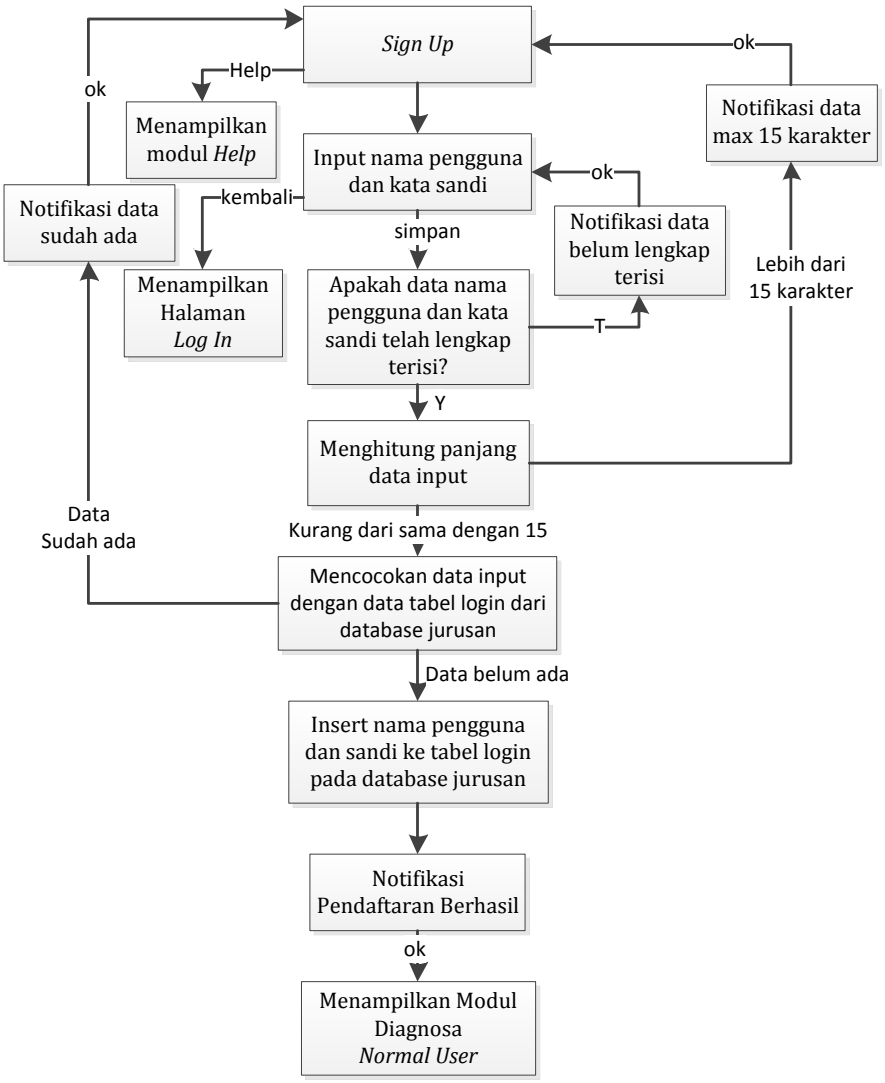
Berdasarkan penjelasan diatas, terdapat 13 modul pada Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi. Diantara 13 modul tersebut terdapat 6 modul yang hanya dapat diakses oleh jenis pengguna *admin* yaitu, Hapus Fakta, Ubah Fakta, Tambah Fakta, Sisipkan Fakta, Tambah Awal, dan Kosongkan. Untuk lebih jelasnya gambar arsitektur sistem dapat dilihat pada gambar 31. Detail masing-masing modul dapat dilihat pada gambar 32 sampai dengan gambar 44.



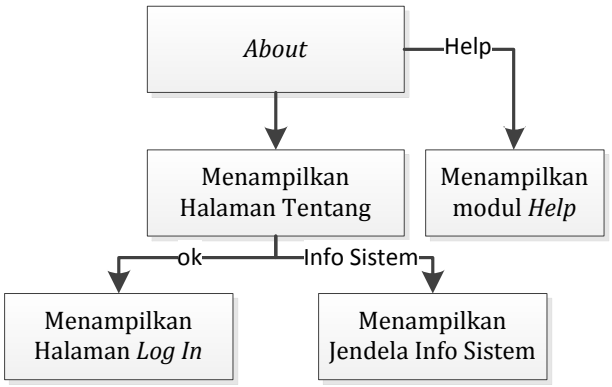
Gambar 31. Desain Arsitektur



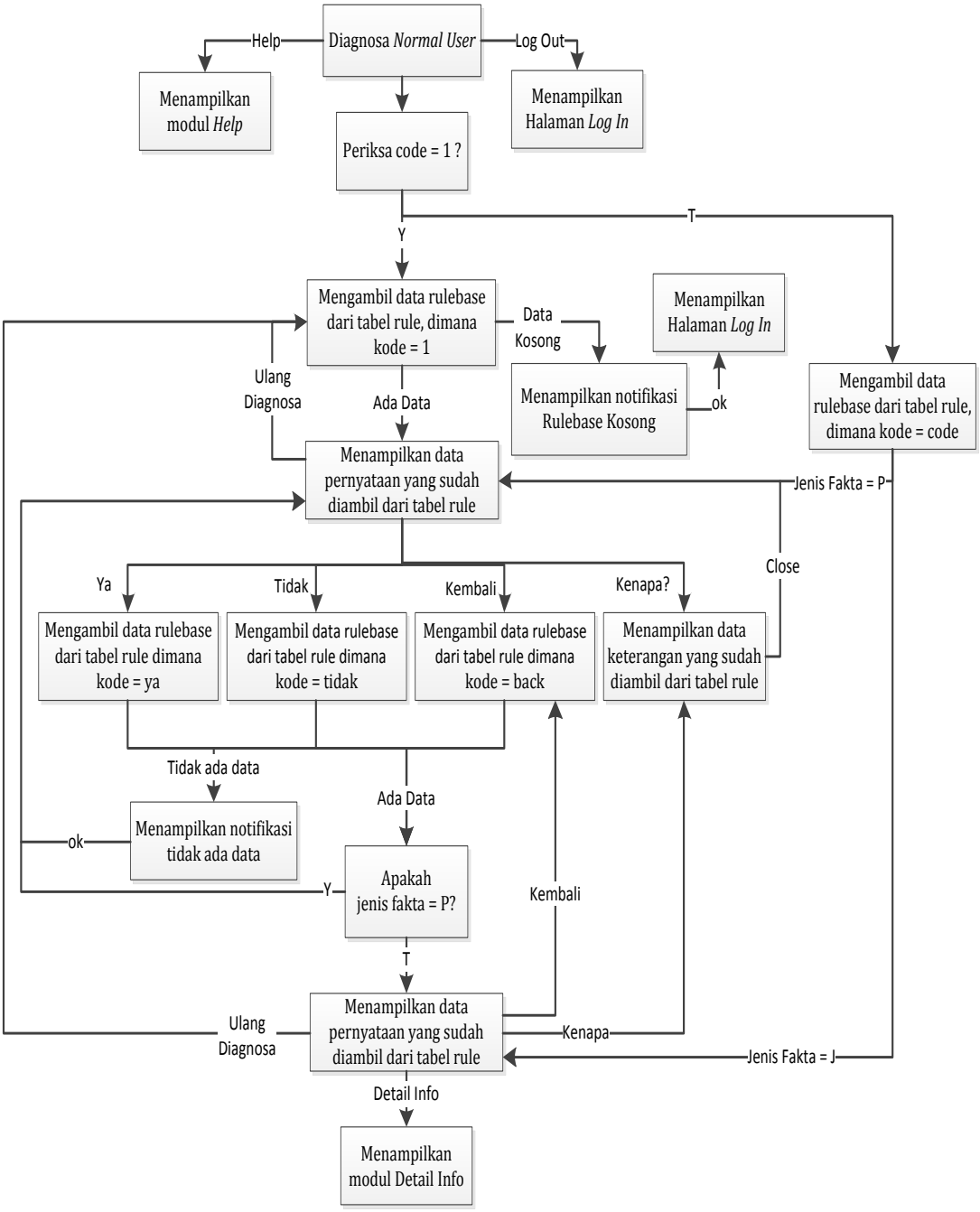
Gambar 32. Modul *Log In*



Gambar 33. Modul *Sign Up*

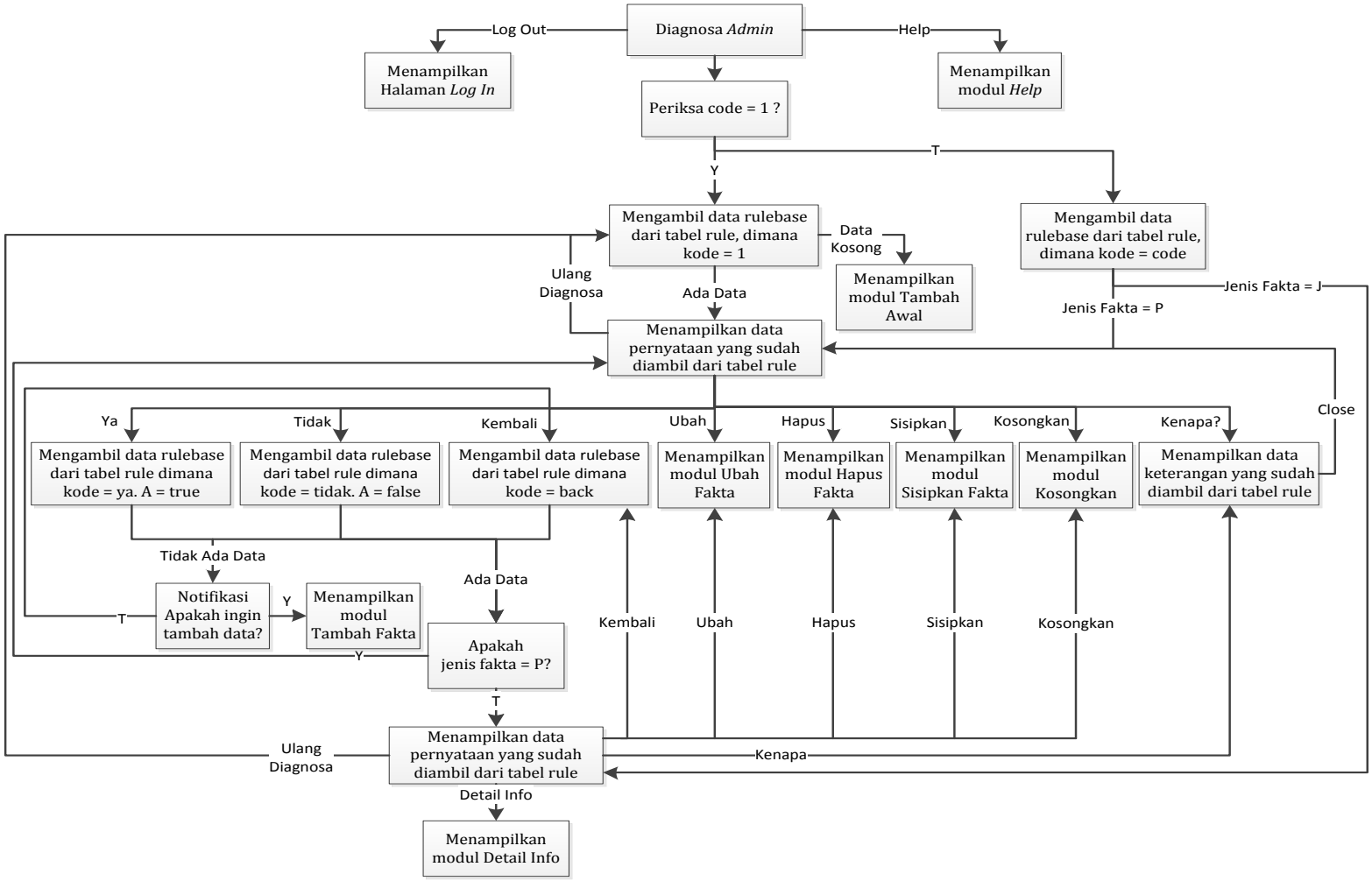


Gambar 34. Modul *About*

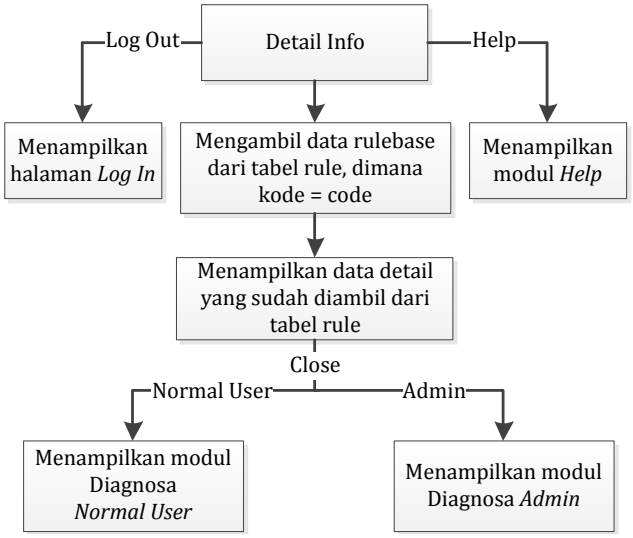


Gambar 35. Modul Diagnosa *Normal User*

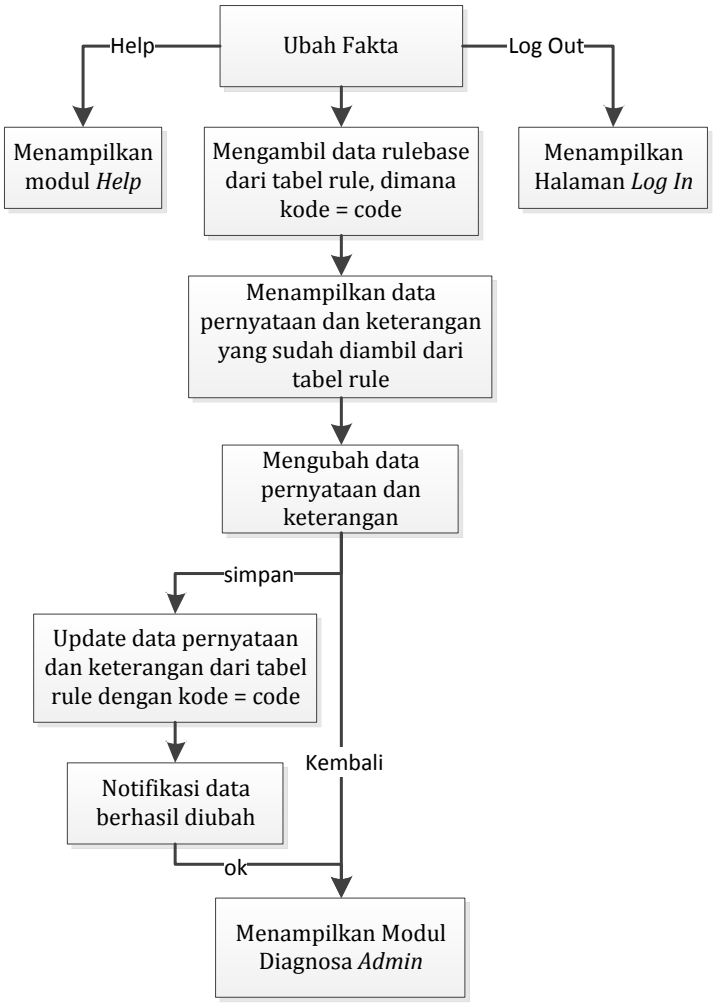
Gambar diatas merupakan detail modul diagnosa untuk jenis pengguna *normal user*. Modul ini hanya dapat diakses oleh pengguna dengan tipe *normal user*untuk melakukan konsultasi guna mendapatkan jurusan kuliah yang sesuai.



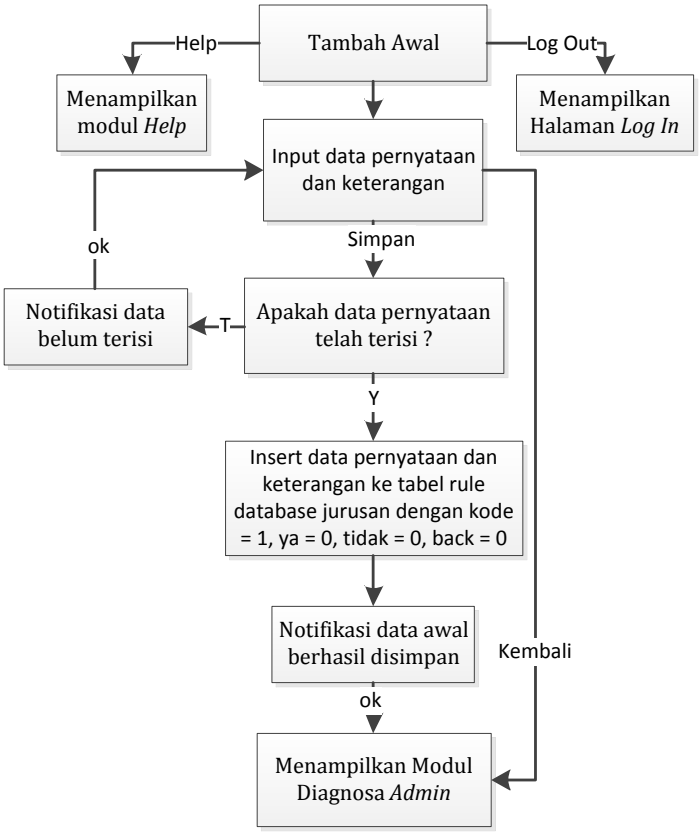
Gambar 36. Modul Diagnosa Admin



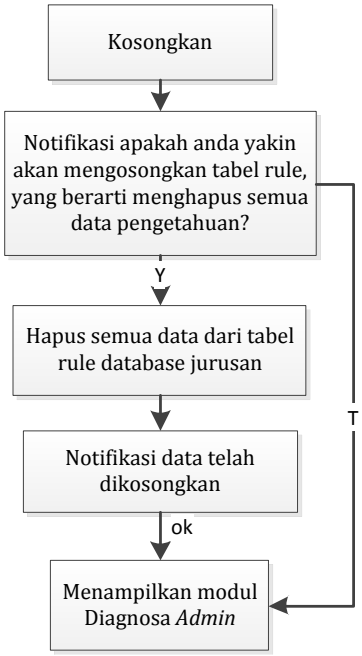
Gambar 37. Modul Detail Info



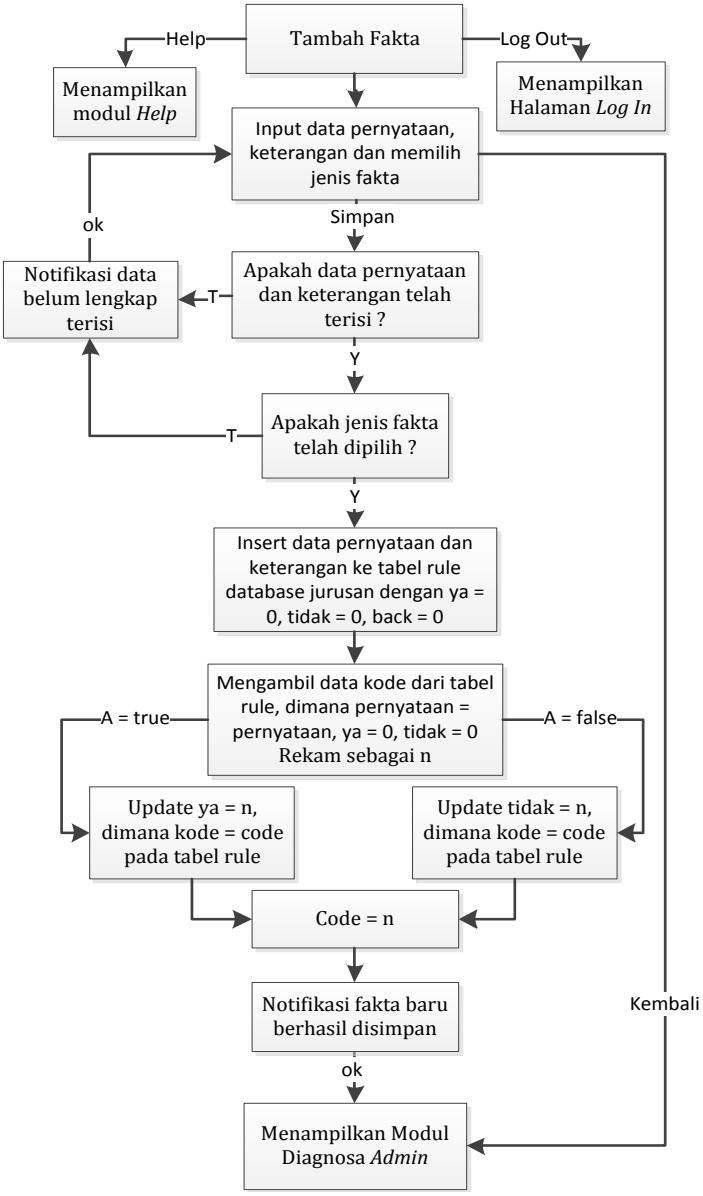
Gambar 38. Modul Ubah Fakta



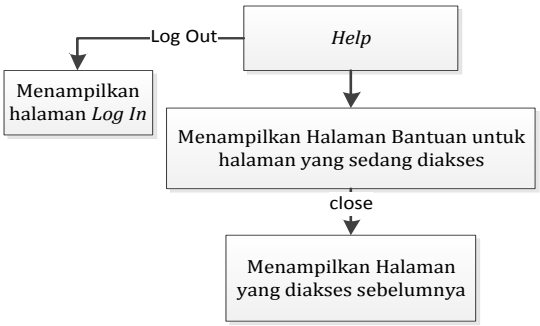
Gambar 39. Modul TambahAwal



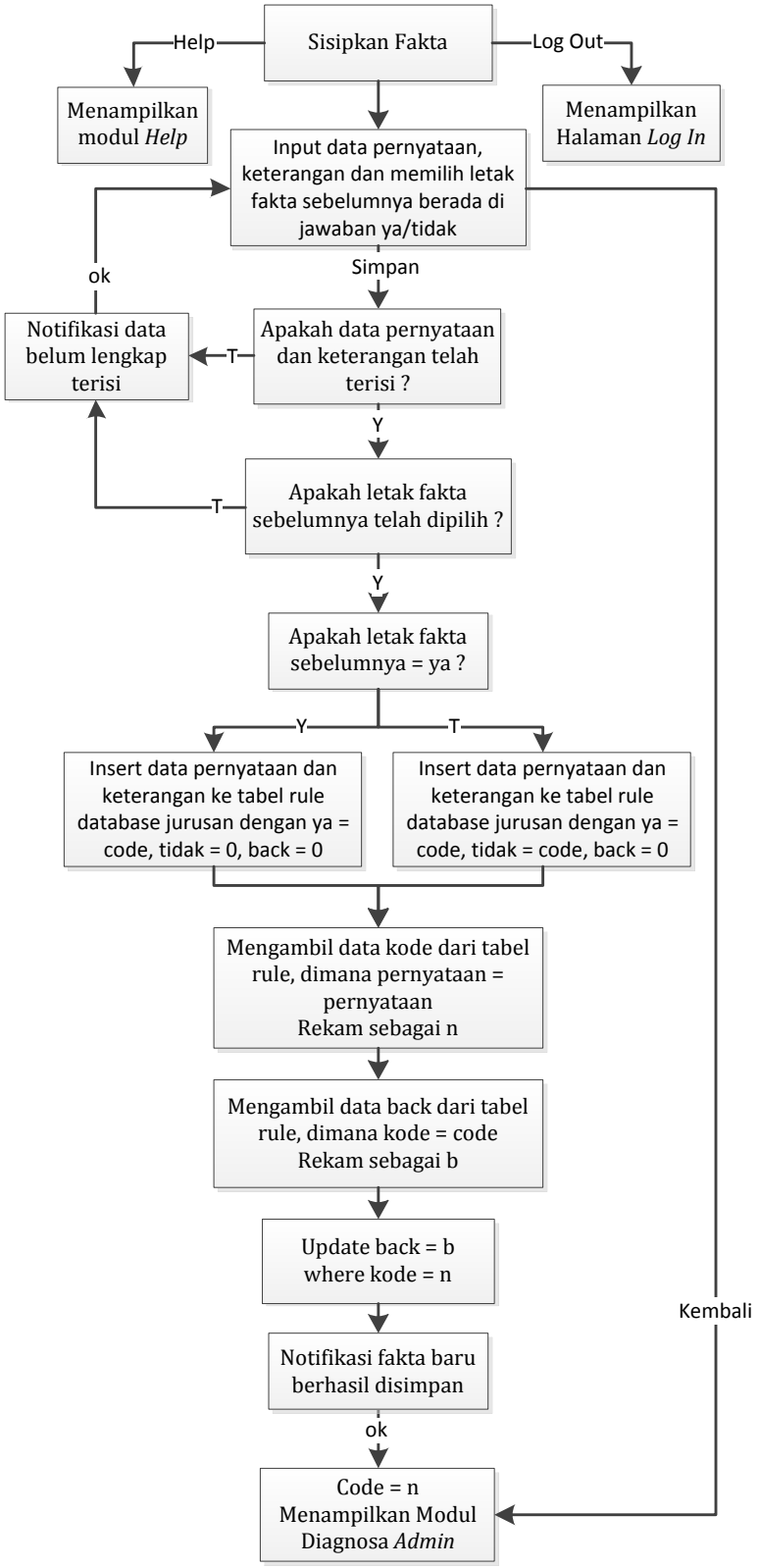
Gambar 40. Modul Kosongkan



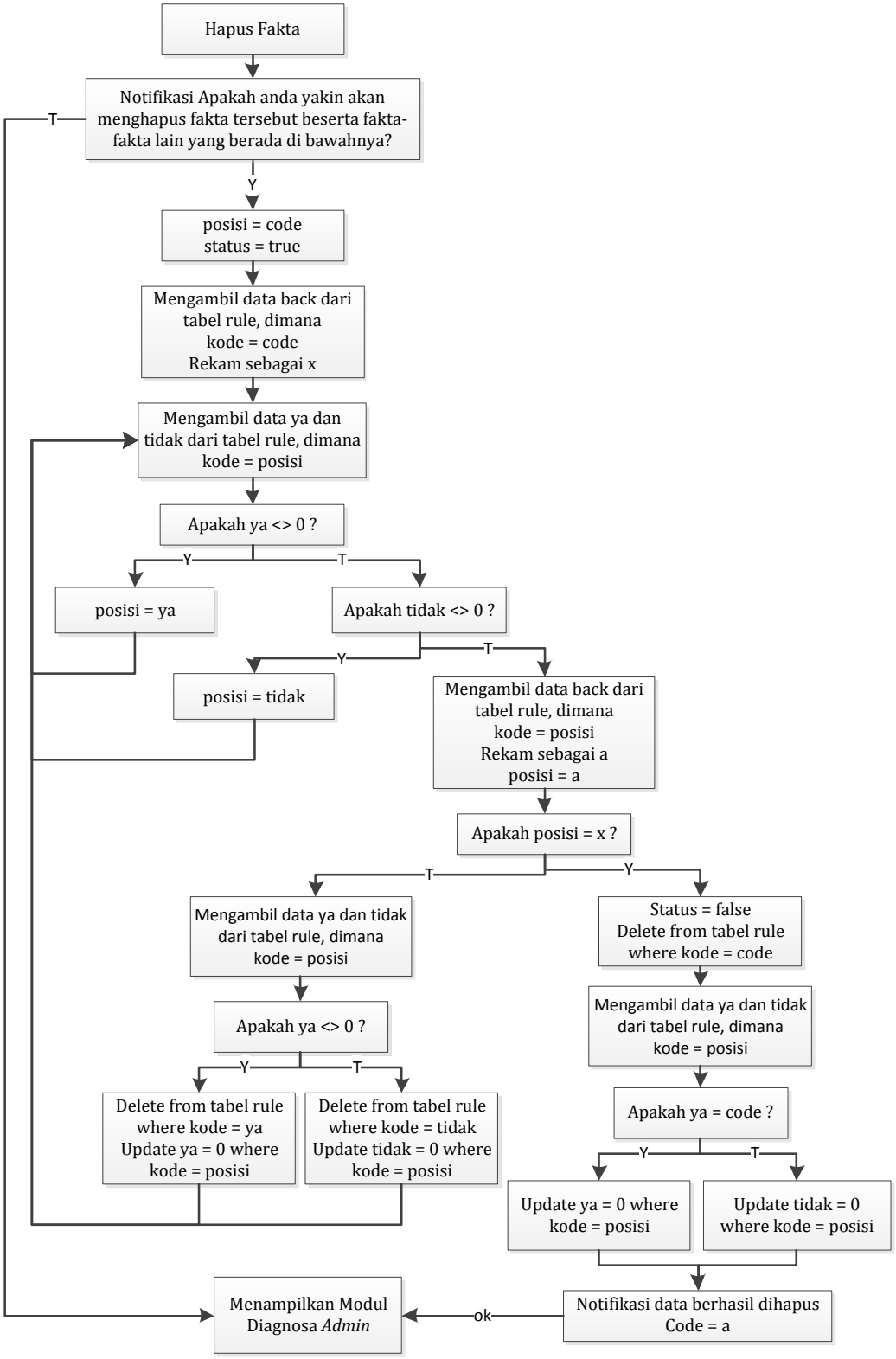
Gambar 41. Modul Tambah Fakta



Gambar 42. Modul *Help*



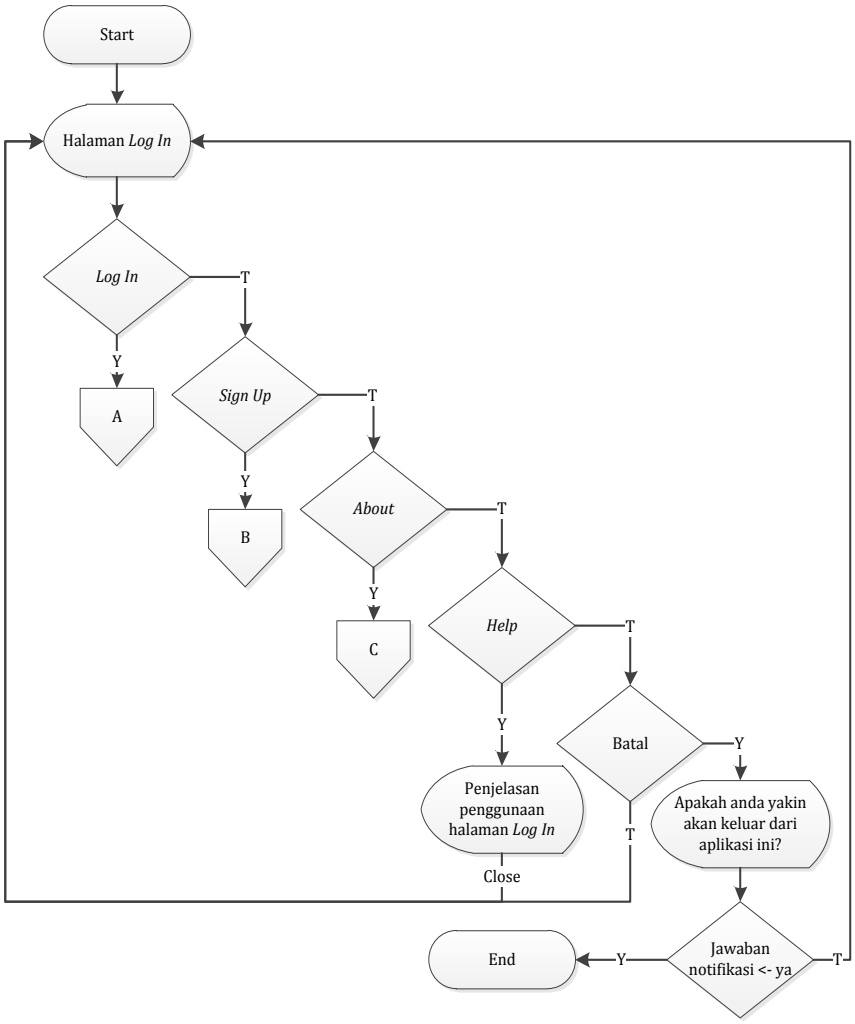
Gambar 43. Modul SisipkanFakta



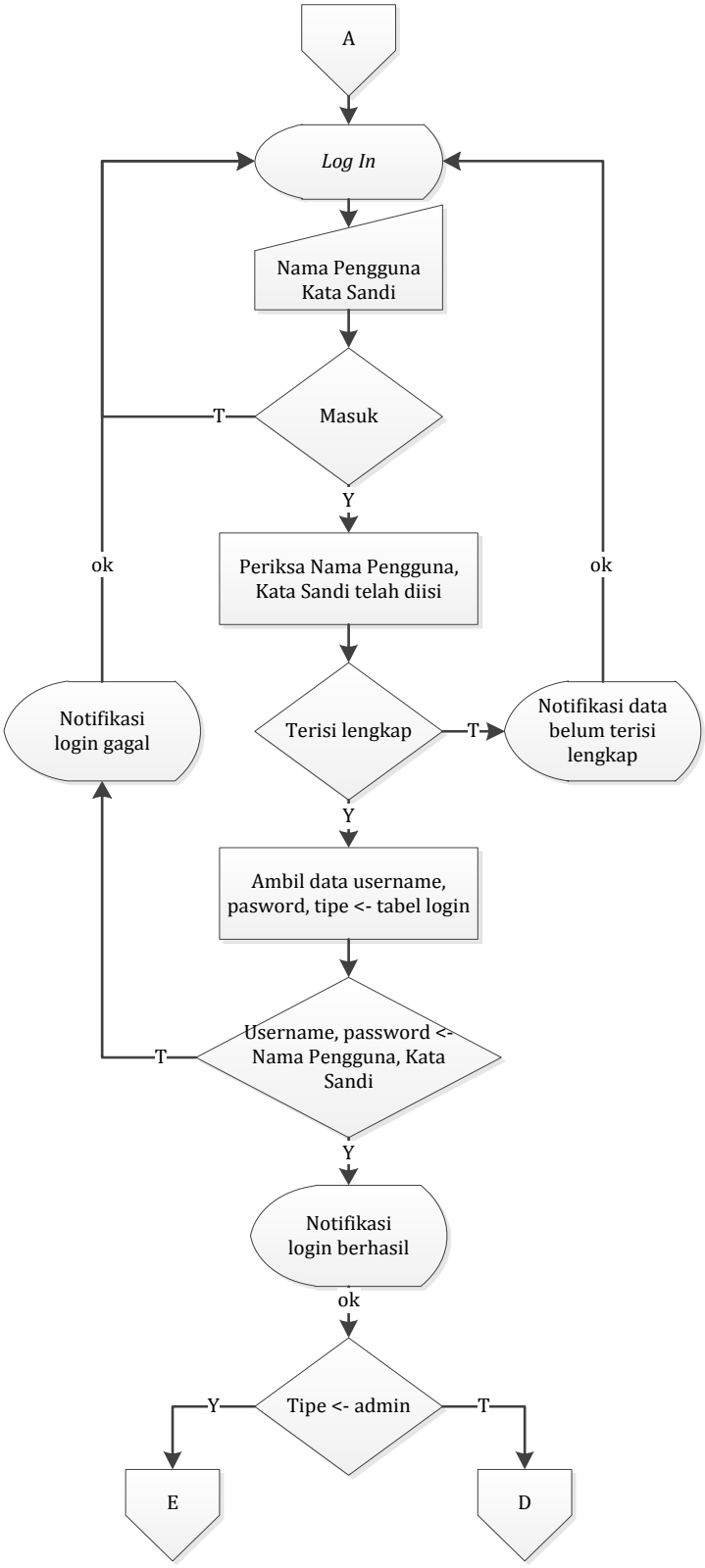
Gambar 44. Modul Hapus Fakta

f. Desain Prosedural

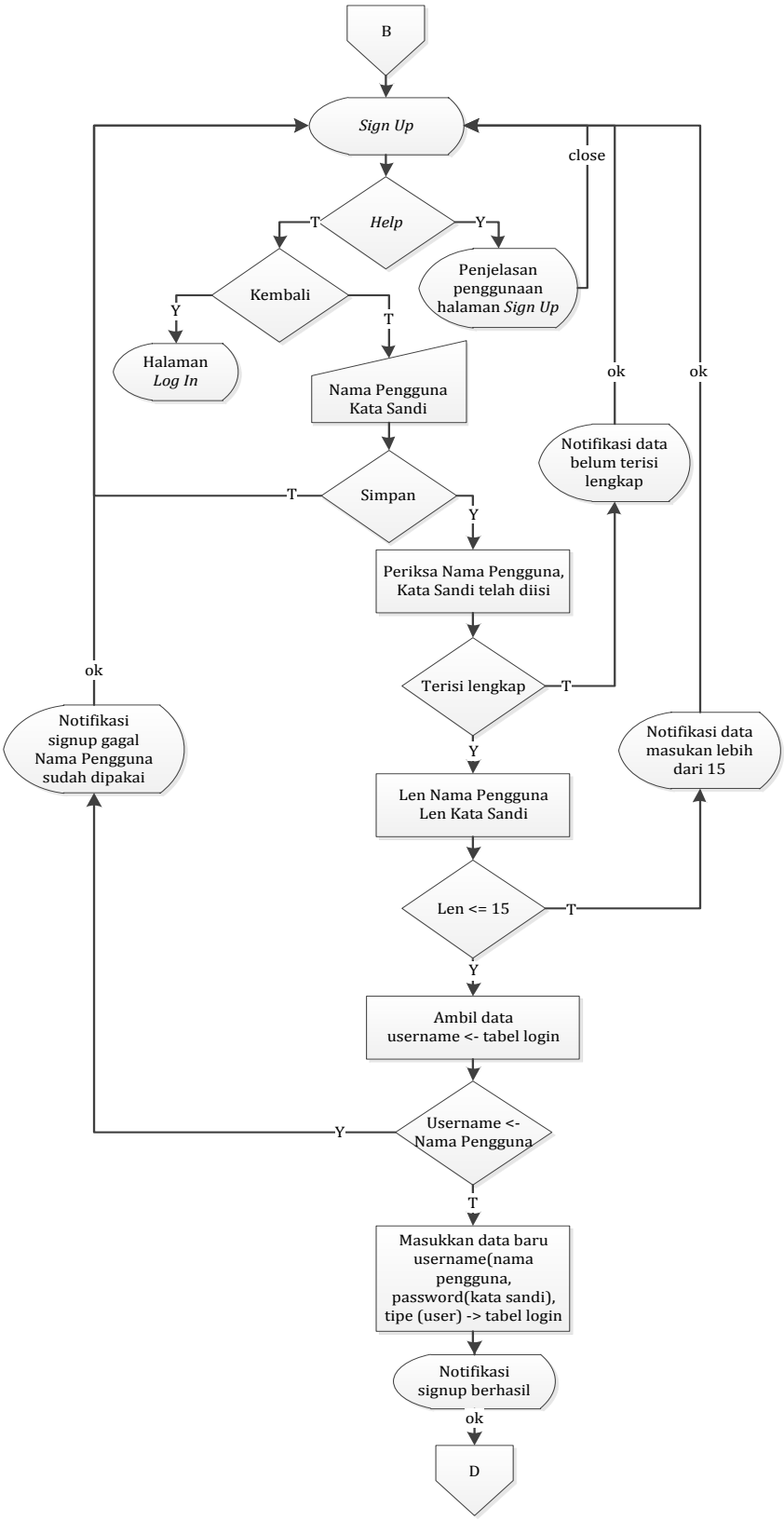
Procedural design difokuskan kepada detail pemrosesan dari setiap modul secara individu. Modul disini merupakan elemen-elemen dari sebuah software yang dapat dipanggil secara terpisah yang berfungsi sebagai tambahan fungsi pada perangkat lunak tersebut. Desain prosedural Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi ini mengacu dari arsitektural desain dan direpresentasikan dengan menggunakan *flowchart*.*Flowchart* sistem dapat dilihat pada gambar 45.



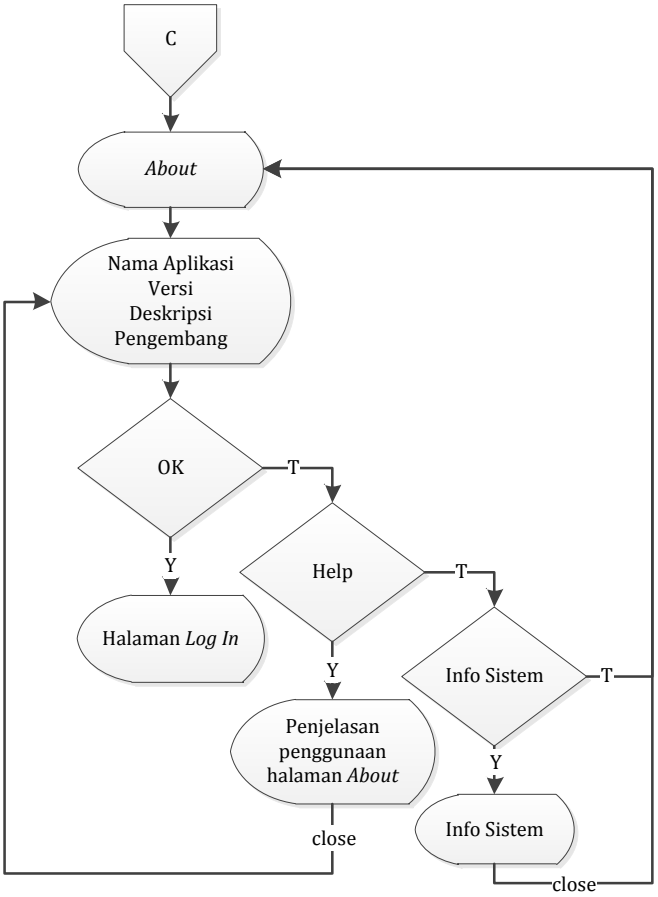
Gambar 45. *Flowchart* Sistem



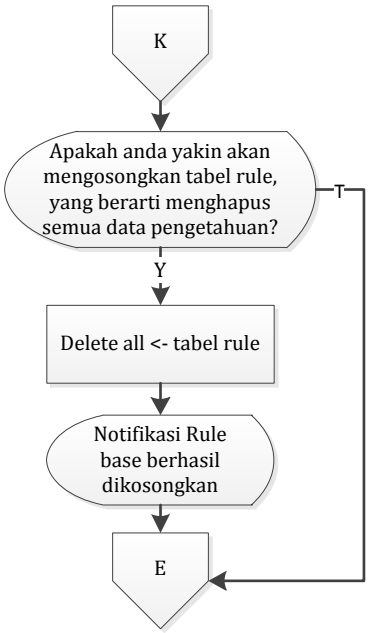
Gambar 46.*Flowchart Log In*



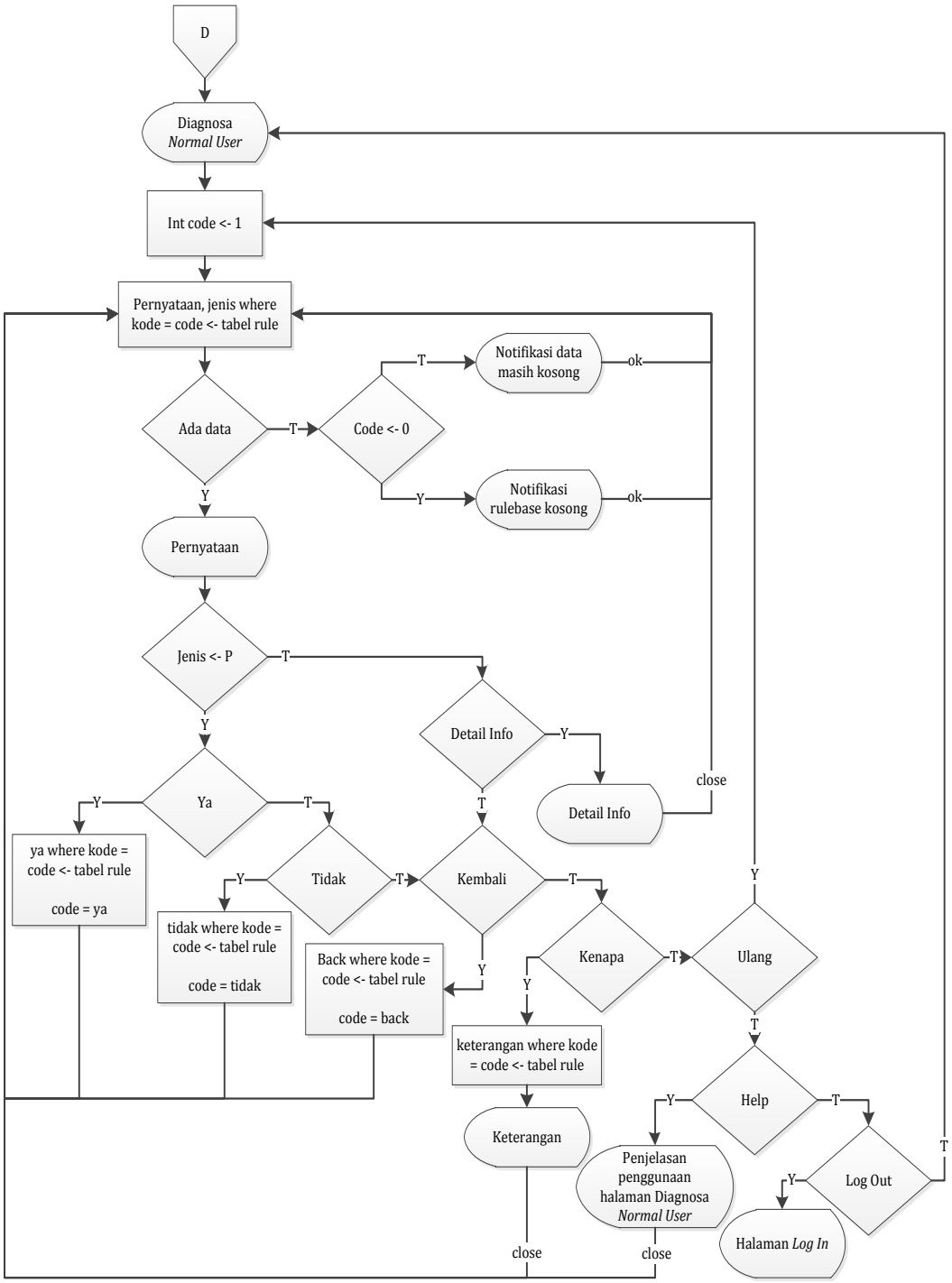
Gambar 47. *Flowchart Sign Up*



Gambar 48.*Flowchart About*

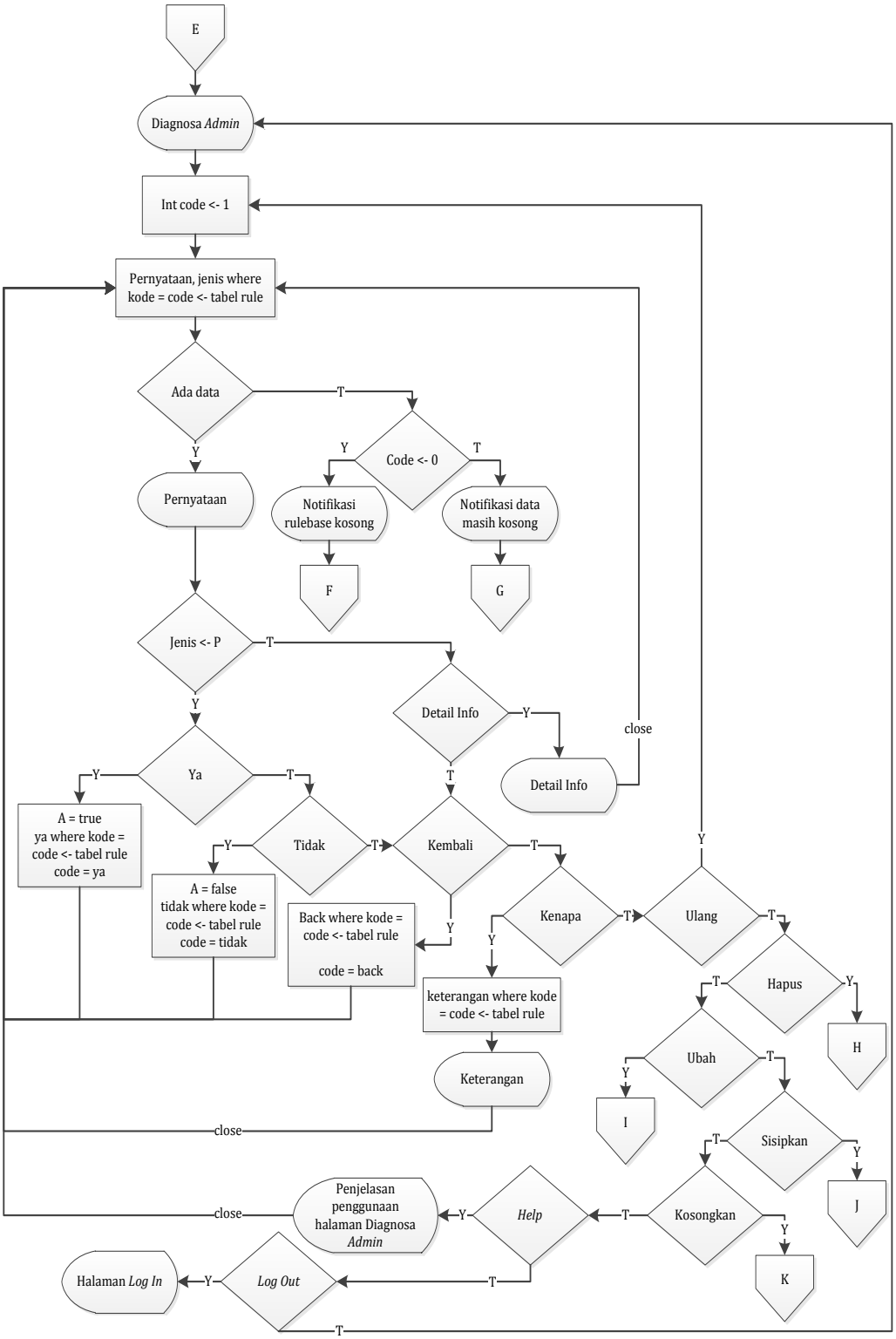


Gambar 49.*Flowchart* Kosongkan

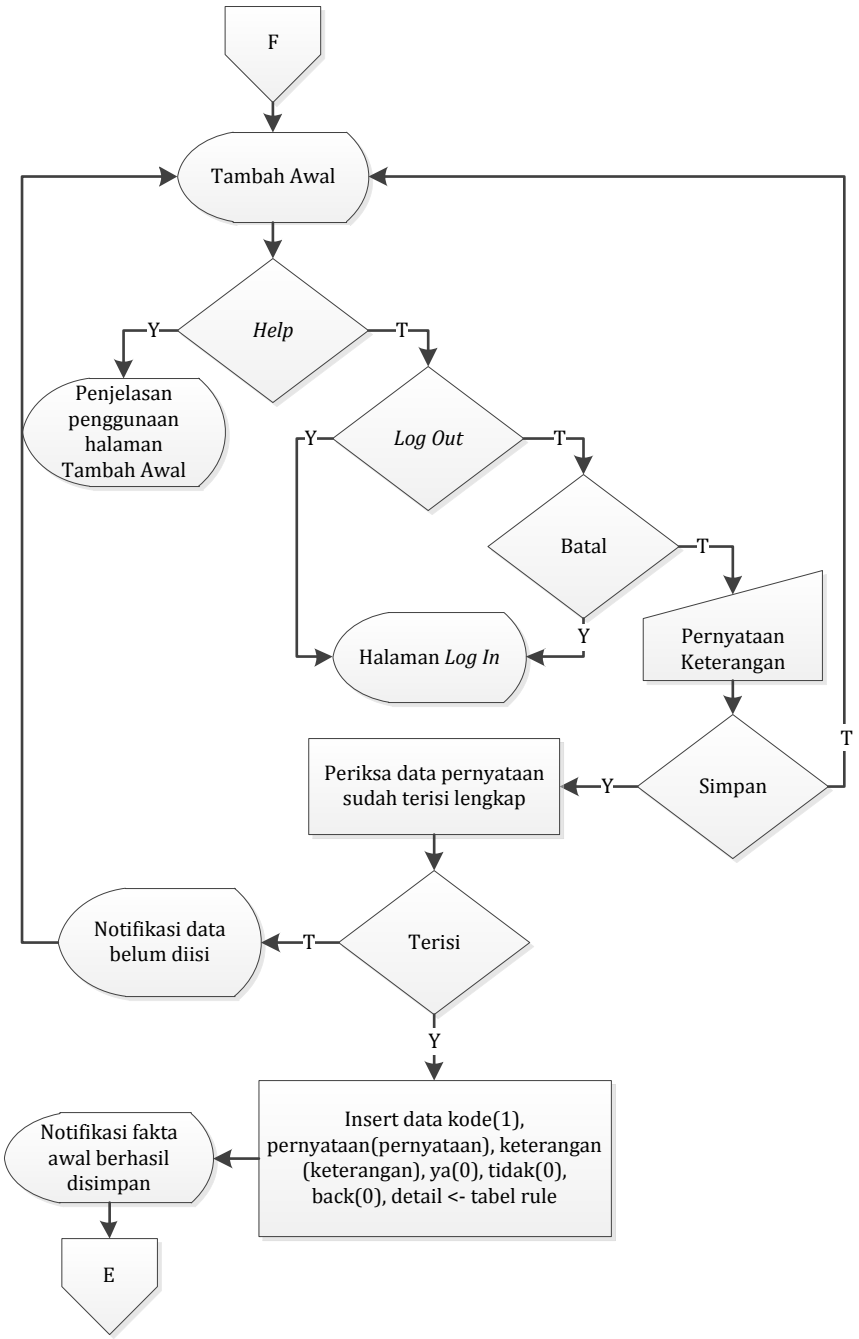


Gambar 50.*Flowchart* Diagnosa *Normal User*

Gambar diatas merupakan *flowchart* untuk proses diagnosa yang diakses oleh tipe pengguna *normal user*untuk melakukan konsultasi.

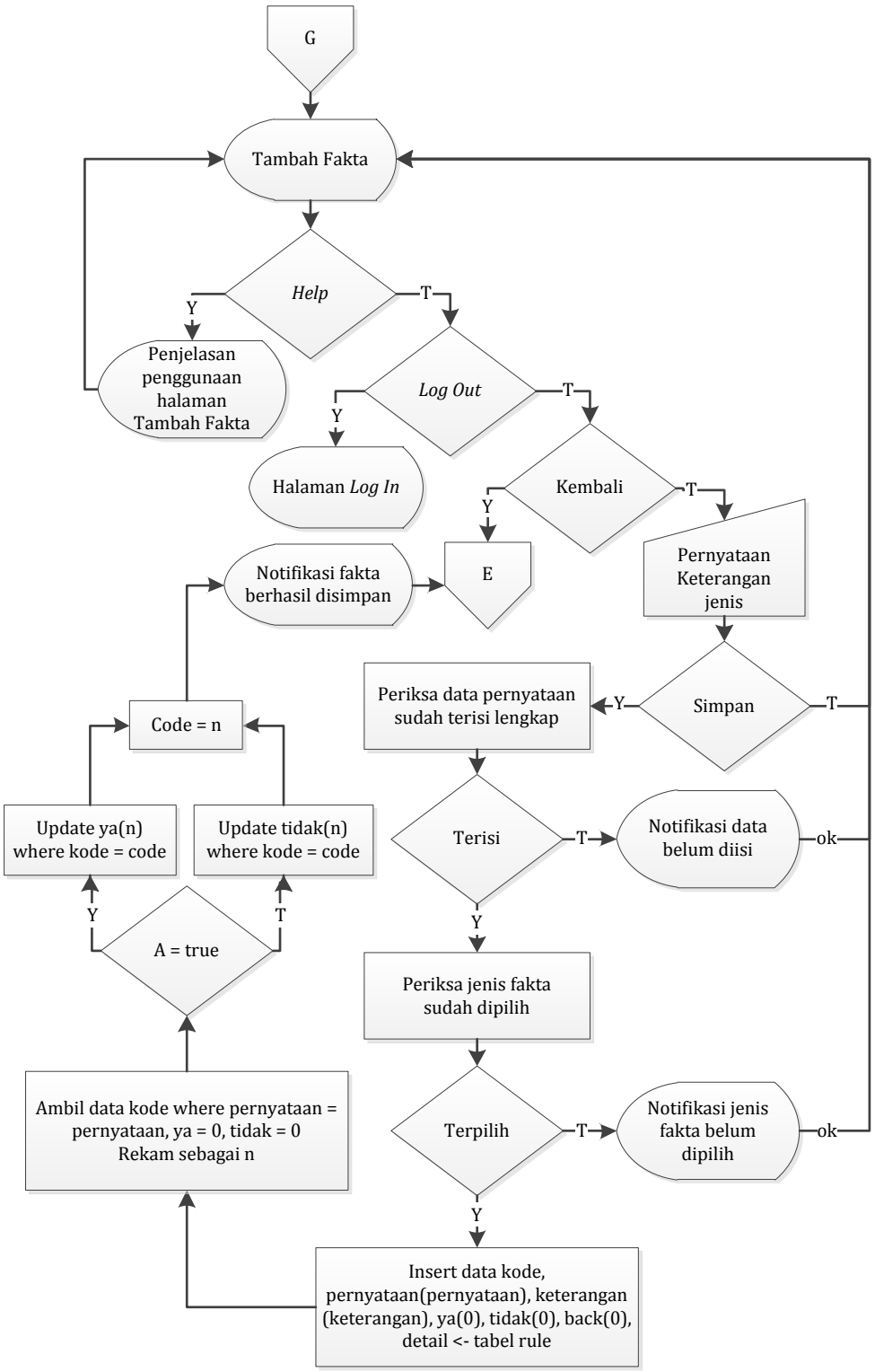


Gambar 51.*Flowchart* Diagnosa *Admin*

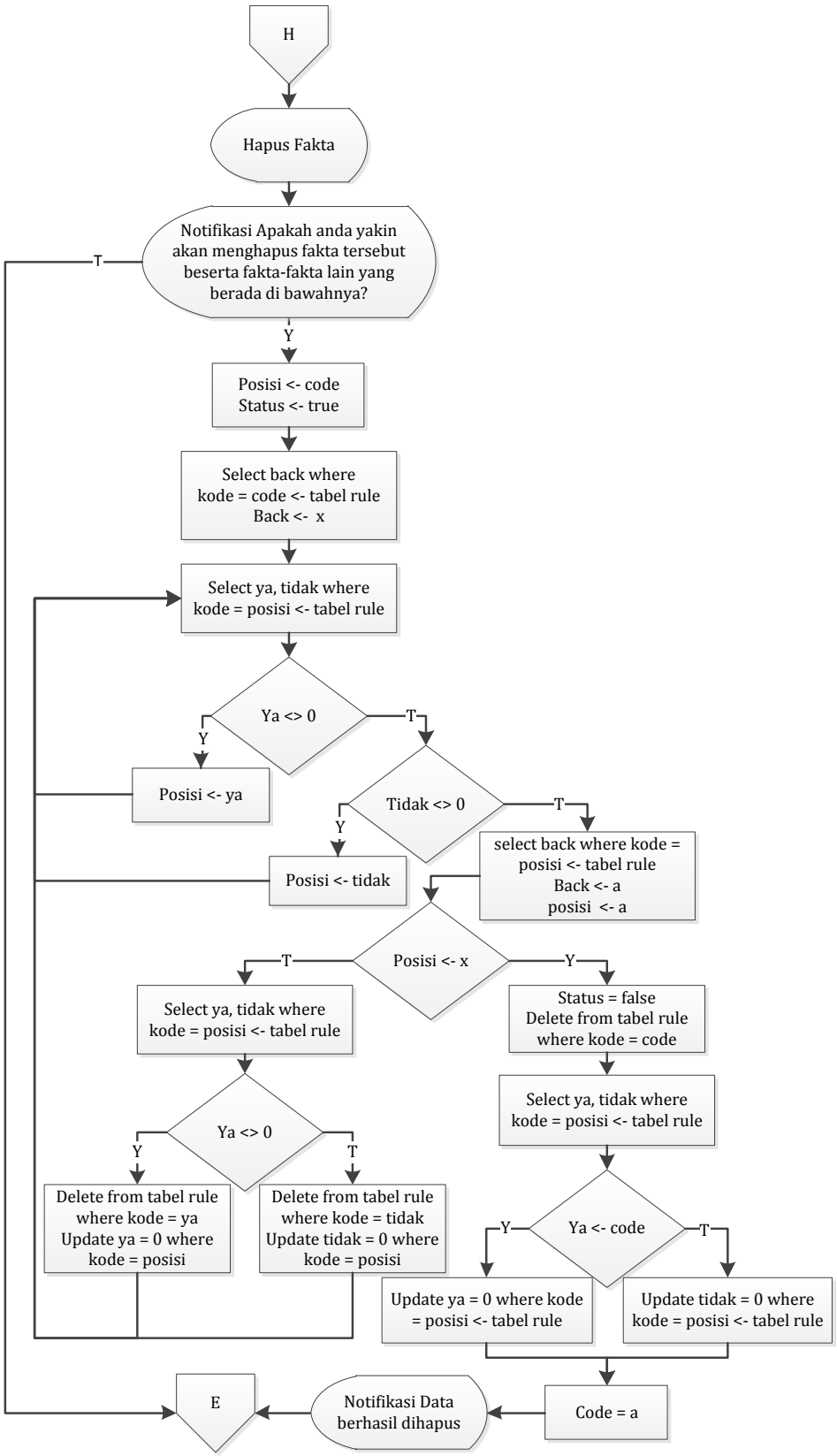


Gambar 52.*Flowchart* Tambah Awal

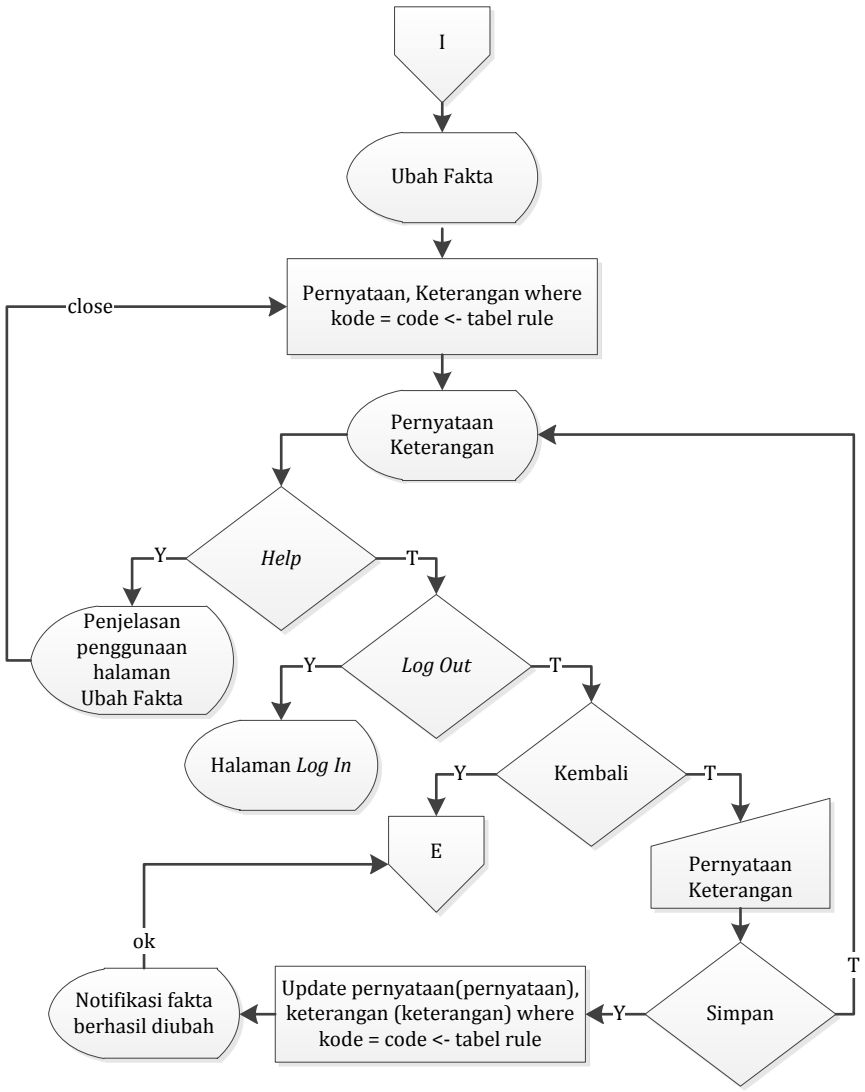
Gambar diatas merupakan *flowchart* untuk proses tambah awal yang hanya dapat diakses oleh tipe pengguna *admin*. Proses tambah awal berfungsi untuk memasukkan data awal bila *rulebase* masih kosong.



Gambar 53.*Flowchart* Tambah Fakta

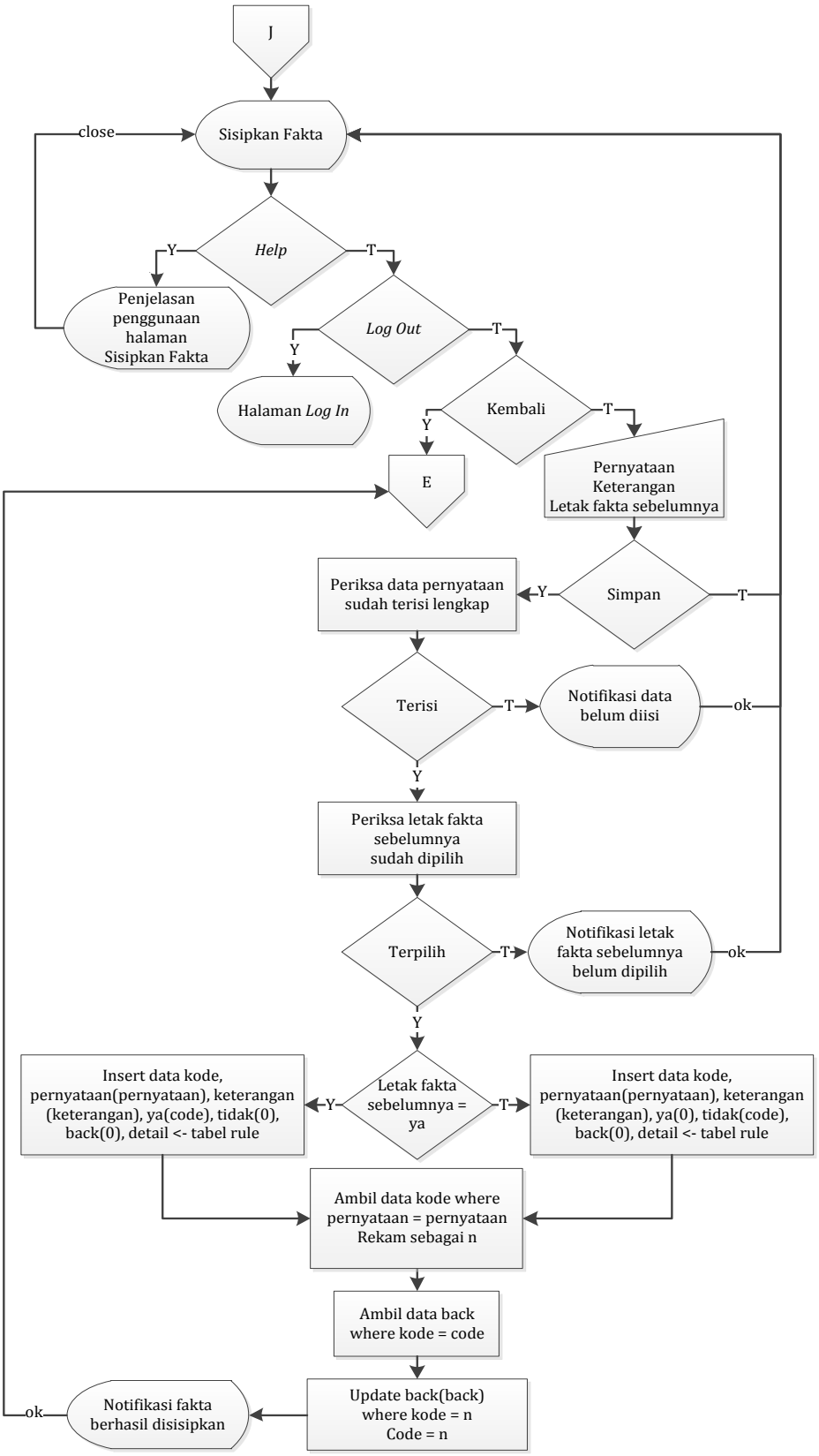


Gambar 54.*Flowchart* Hapus Fakta



Gambar 55.*Flowchart* Ubah Fakta

Gambar diatas merupakan *flowchart* ubah fakta yang hanya dapat diakses oleh pengguna dengan tipe *admin*. Proses ubah fakta berfungsi untuk mengubah fakta yang sudah ada dalam *rulebase*. Fungsi lain yang juga hanya dapat diakses oleh tipe pengguna *admin* adalah sisipkan fakta. *Flowchart* sisipkan fakta dapat dilihat pada gambar 56. Proses sisipkan fakta berfungsi untuk menyisipkan fakta diantara fakta-fakta yang telah tersimpan dalam*database*.



Gambar 56. *Flowchart* Sisipkan Fakta

3. Pengkodean (*Coding*)

Proses pengembangan perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi ini merupakan proses implementasi sistem. Tahapan implementasi merupakan translasi desain ke dalam program perangkat lunak. Hasil dari tahapan pengkodean ini adalah program komputer yang sesuai dengan analisis kebutuhan dan desain yang telah di buat pada tahap sebelumnya. Desain sistem yang telah dibuat pada tahap desain di implementasikan dalam bentuk kode atau program. *Flowchart* yang telah di buat di transformasikan ke dalam implementasi kode program. Pengembang melakukan implementasi dari perangkat lunak yang akan dibuat. Pengembang membuat bagian per bagian sistem yang ada sehingga menjadi satu kesatuan sistem yang utuh.

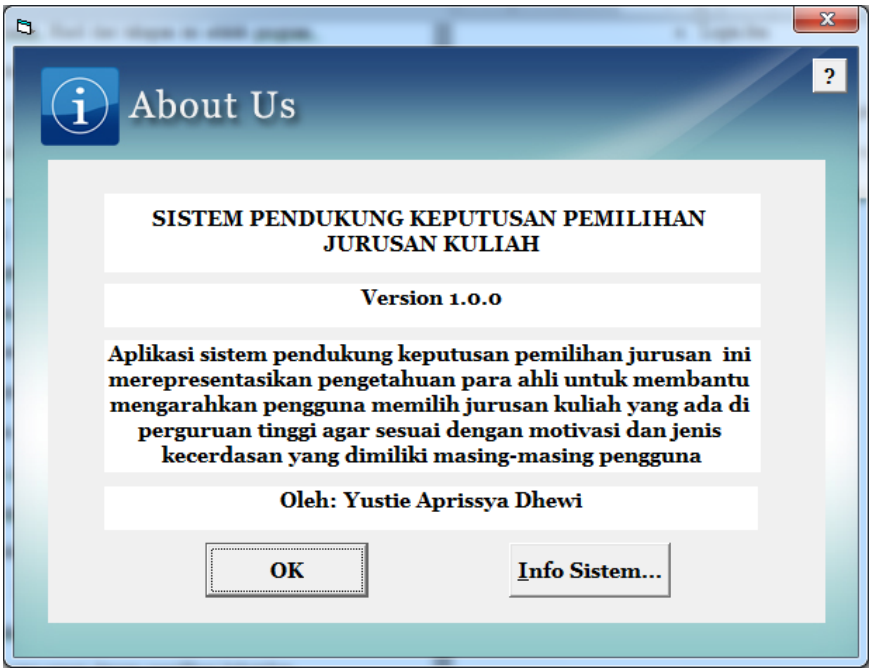
Pengembang juga melakukan pengujian *correctness* berupa pengujian *white box* dan pengujian *test case*berupa pengujian *black box* pada proses implementasi ini.Pengembang melakukan uji *white box* untuk mengecek implementasi program yang dibuat. Pengujian *white box* ini menguji implementasi program dari segi desain dan kode program. Pengujian *white box* dilakukan dengan cara memeriksa lojik dari kode program. Sedangkan uji *black box* dilakukan untuk menguji apakah program mampu menghasilkan fungsi-fungsi, masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan.Pengkodean yang dilakukan oleh peneliti dilakukan dengan bahasa pemrograman VB yang dieksekusi dengan *softwareVisual Basic 6*.

Berikut adalah daftar *form* yang digunakan dalam *project* Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi :

a. About.frm
b. Diagnosa_admin.frm
c. Diagnosa_user.frm
d. Kotak_tambah.frm
e. Login.frm
f. Signup.frm
g. Sisipkan.frm
h. Tambah.frm
i. Tambah_awal.frm
j. Ubah.frm
k. Module1.bas

Berikut adalah implementasi antar muka beserta potongan*source code* utama dari setiap *form* yang menyusun *project*Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi :

a. About.frm

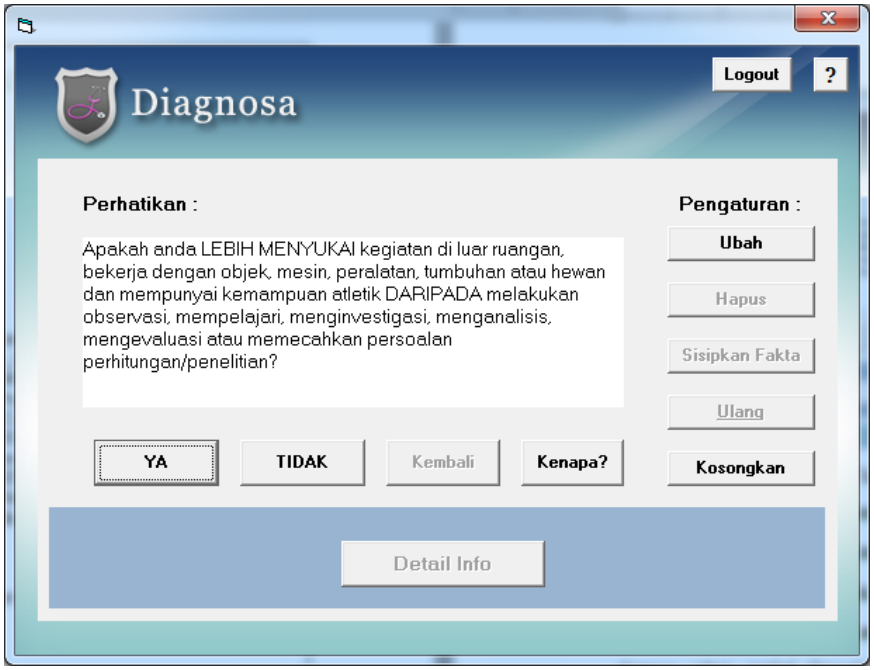


Gambar 57. Tampilan Halaman *About*

```
Private Sub Form_Load()
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "."
    & App.Revision
End Sub

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr
    Dim rc As Long
    Dim SysInfoPath As String

    ' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
    gREGVALSYSINFO, SysInfoPath) Then
     ' Try To Get System Info Program Path Only From Registry...
    ElseIf GetKeyValue(HKEY_LOCAL_MACHINE,
    gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC, SysInfoPath) Then
         ' Validate Existance Of Known 32 Bit File Version
        If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
            SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
            ' Error - File Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If
        ' Error - Registry Entry Can Not Be Found...
    Else
        GoTo SysInfoErr
    End If
    Call Shell(SysInfoPath, vbNormalFocus)
    Exit Sub
    SysInfoErr:
        MsgBox "System Information Is Unavailable At This Time",
        vbOKOnly
End Sub
```

Halaman *about* digunakan untuk memberi info kepada pengguna tentang aplikasi ini. Pada halaman *about* terdapat 3 tombol yaitu, “OK” untuk kembali ke halaman *Login*, “Info Sistem” untuk melihat Info Perangkat, dan “?” untuk melihat petunjuk penggunaan halaman *about*.

b. Diagnosa_admin.frm



Gambar 58. Tampilan Diagnosa *Admin*

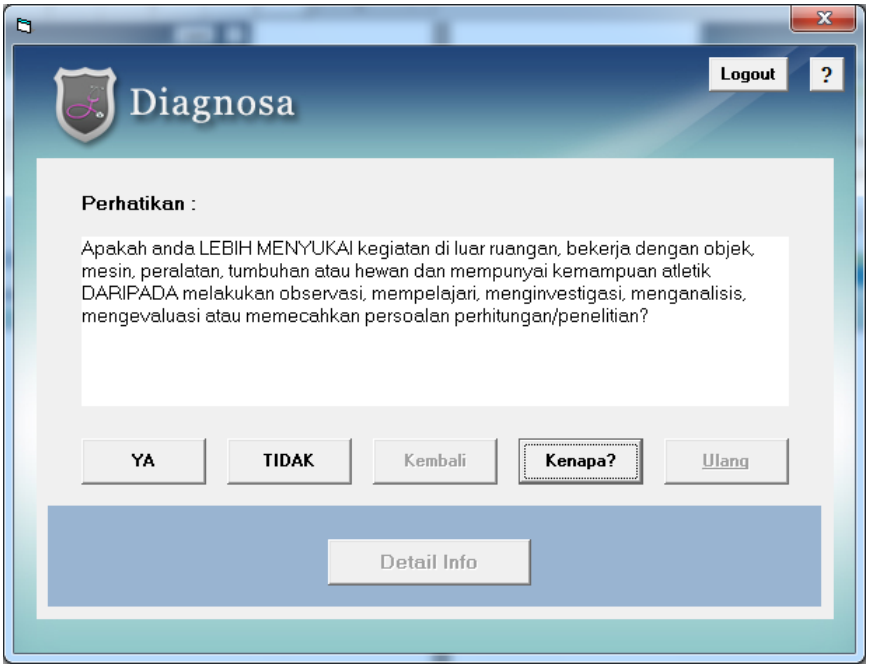
Halaman diagnosa *admin* merupakan halaman dimana tipe pengguna *admin* dapat melakukan diagnosa atau konsultasi serta melakukan perubahan *knowledge*yang ada di dalam *database*. Pada halaman diagnosa *admin* terdapat label yang berisi pernyataan, tombol "?" untuk melihat petunjuk penggunaan halaman diagnosa *admin*, tombol "logout", kumpulan tombol diagnosa, dan kumpulan tombol pengaturan.

Kumpulan tombol diagnosa terdiri dari tombol "YA", "TIDAK", "Kembali", "Kenapa?", "Detail Info" dan "Ulang". Sedangkan kumpulan tombol pengaturan terdiri dari tombol "Ubah", "Hapus", "Sisipkan Fakta", dan "Kosongkan". Setiap tombol berfungsi sebagai mana mestinya dengan algoritma yang digambarkan pada *flowchart.* Berikut adalah potongan *source code*untuk tombol "Hapus" yang ada pada halaman diagnosa *admin* :

```
Private Sub Command22_Click()
'menghapus fakta beserta anak-anaknya
 posisi = Code
strSelect6 = "SELECT back FROM rule WHERE kode='" & Code &
"'"
Set rs = Conn.Execute(strSelect6)
 x = rs.Fields(0).Value
 Status = True

Do Until (posisi = x And Status = True)
    strSelect6 = "SELECT ya FROM rule WHERE kode='" & posisi &
    "'"
    Set rs = Conn.Execute(strSelect6)
    ya = rs.Fields(0).Value
    strSelect6 = "SELECT tidak FROM rule WHERE kode='" & posisi
    & "'"
    Set rs = Conn.Execute(strSelect6)
    tidak = rs.Fields(0).Value

 If ya <> 0 Then
        posisi = ya
    ElseIf tidak <> 0 Then
        posisi = tidak
    Else
        strSelect6 = "SELECT back FROM rule WHERE kode='" &
        posisi & "'"
        Set rs = Conn.Execute(strSelect6)
        a = rs.Fields(0).Value
        posisi = a
        If posisi = x Then
            Status = False
            strDelete = "DELETE FROM rule WHERE kode='" &
            Code & "'"
            Set rs = Conn.Execute(strDelete)
            strSelect6 = "SELECT ya FROM rule WHERE kode='" &
            posisi & "'"
            Set rs = Conn.Execute(strSelect6)
            ya = rs.Fields(0).Value
            strSelect6 = "SELECT tidak FROM rule WHERE kode='"
            & posisi & "'"
            Set rs = Conn.Execute(strSelect6)
            tidak = rs.Fields(0).Value
            If ya = Code Then
                strUpdate = "UPDATE rule SET ya='0' WHERE
                kode='" & posisi & "'"
                Set rs = Conn.Execute(strUpdate)
```

```
                ElseIf tidak = Code Then
                    strUpdate = "UPDATE rule SET tidak='0' WHERE
                    kode='" & posisi & "'"
                    Set rs = Conn.Execute(strUpdate)
                End If
    Exit Do
            Else
                strSelect6 = "SELECT ya FROM rule WHERE kode='" &
                posisi & "'"
                Set rs = Conn.Execute(strSelect6)
                 ya = rs.Fields(0).Value
                strSelect6 = "SELECT tidak FROM rule WHERE kode='"
                & posisi & "'"
                 Set rs = Conn.Execute(strSelect6)
                tidak = rs.Fields(0).Value
                If ya <> 0 Then
                    strDelete = "DELETE FROM rule WHERE kode='" &
                    ya & "'"
                    Set rs = Conn.Execute(strDelete)
                    strUpdate = "UPDATE rule SET ya='0' WHERE
                    kode='" & posisi & "'"
                    Set rs = Conn.Execute(strUpdate)
                ElseIf tidak <> 0 Then
                    strDelete = "DELETE FROM rule WHERE kode='" &
                    tidak & "'"
                    Set rs = Conn.Execute(strDelete)
                    strUpdate = "UPDATE rule SET tidak='0' WHERE
                    kode='" & posisi & "'"
                    Set rs = Conn.Execute(strUpdate)
                End If
            End If
        End If

    Loop
    MsgBox "Data berhasil dihapus!", vbInformation, "Perhatian!"

    Code = a
    strSelect2 = "SELECT pernyataan FROM rule WHERE kode='" &
    Code & "'"
    Set rs = Conn.Execute(strSelect2)
    strSelect2 = rs.Fields(0).Value
    Baca = strSelect2
    Label2.Caption = Baca
    Command1.Enabled = True
    Command2.Enabled = True
End Sub
```

c. Diagnosa_user.frm



Gambar 59. Tampilan Diagnosa *User*

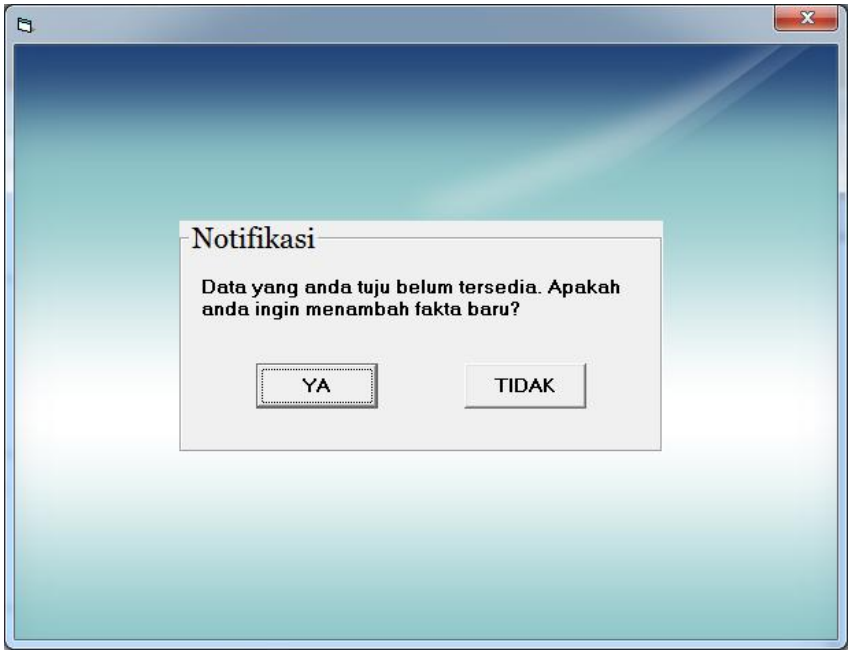
Halaman diagnosa *user* merupakan halaman dimana tipe pengguna *user* dapat melakukan diagnosa atau konsultasi mengenai jurusan kuliah yang sesuai dengan jenis kecerdasannya. Terdapat label yang berisi pernyataan, tombol "?" untuk melihat petunjuk penggunaan halaman diagnosa *user*, tombol "logout", dan kumpulan tombol diagnosa. Kumpulan tombol diagnosa terdiri dari tombol "YA", "TIDAK", "Kembali", "Kenapa?", "Detail Info" dan "Ulang". Setiap tombol berfungsi sebagai mana mestinya. Berikut adalah potongan *source code* yang ada pada halaman diagnosa *user* :

```
Private Sub Form_Load()
    If Code = 1 Then
        Command5.Enabled = False
        Command7.Enabled = False
        Command3.Enabled = False
        Command11.Enabled = False
    End If
```

```
    strSelect2 = "SELECT pernyataan FROM rule WHERE kode='" &
    Code & "'"
    Set rs = Conn.Execute(strSelect2)
    If Not rs.EOF And Not rs.BOF Then
        strRekam = rs.Fields(0).Value
        Baca = strRekam
        Label2.Caption = Baca
        'menentukan tipe pernyataan. jika jawaban maka tombol ya tidak
        akan hilang
        strSelect4 = "SELECT jenis from rule where kode='" & bil & "'"
        Set rs = Conn.Execute(strSelect4)
        ‘jika data sudah ada
        If rs.EOF = False Then
            strJenis = rs.Fields(0).Value
            perjab = strJenis
            If perjab = "P" Then
                Command1.Enabled = True
                Command2.Enabled = True
                Command11.Enabled = False
            ElseIf perjab = "J" Then
                Command1.Enabled = False
                Command2.Enabled = False
                Command11.Enabled = True
            End If
        End If
    End If
End Sub
Public Sub Perjalanan()
    'menghitung jumlah data
    strSelect3 = "SELECT COUNT(kode) from rule where kode = '" &
    Code & "' "
    Set rs = Conn.Execute(strSelect3)
    strTambah = rs.Fields(0).Value
    bil = strTambah
    bil = bil + 1
    strback = Code
    beck = strback
    'membaca kode pernyataan yang dituju sesuai jawaban pengguna
    If tanda = 1 Then
        strSelect2 = "SELECT ya FROM rule WHERE kode='" &Code&
        "'"
        Set rs = Conn.Execute(strSelect2)
        If rs.EOF = True Or rs.BOF = True Then
            Form1.Visible = False
            Form5.Visible = True
        Else
```

```
            strAnswer = rs.Fields(0).Value
            Code = strAnswer
        End If
    ElseIf tanda = 0 Then
        strSelect2 = "SELECT tidak FROM rule WHERE kode='"
        &Code& "'"
        Set rs = Conn.Execute(strSelect2)
        If IsNull(rs.Fields(0).Value) Then
            Form1.Visible = False
            Form5.Visible = True
        Else
            strAnswer = rs.Fields(0).Value
            Code = strAnswer
        End If
    End If
    'membaca pernyataan dari kode yang didapat
    strSelect1 = "SELECT pernyataan FROM rule WHERE kode='" &
    Code & "'"
    Set rs = Conn.Execute(strSelect1)
    If Not rs.EOF And Not rs.BOF Then
         strMorse = rs.Fields(0).Value
        Baca = strMorse
        Label2.Caption = Baca
    Else
        Form5.Visible = True
        Unload Me
    End If
    'membaca jenis fakta yang didapat
    strSelect4 ="SELECT jenis from rule where kode='"&Code&"'"
    Set rs = Conn.Execute(strSelect4)
    strJenis = rs.Fields(0).Value
    perjab = strJenis
    If perjab = "P" Then
        Command1.Enabled = True
        Command2.Enabled = True
        Command11.Enabled = False
    ElseIf perjab = "J" Then
        Command1.Enabled = False
        Command2.Enabled = False
        Command11.Enabled = True
    End If
    If Code = 0 Or Code = 1 and ksg = false Then
         Command7.Enabled = False
         Command3.Enabled = False
    End If
End Sub
```

d. Kotak_tambah.frm



Gambar 60. Tampilan Kotak Tambah

Notifikasi tambah fakta muncul apabila data yang dituju belum ada. Terdapat 2 tombol yaitu, tombol "YA" untuk menuju halaman tambah fakta, dan tombol "TIDAK" untuk kembali ke halaman diagnosa *admin*. Berikut potongan *source code* untuk halaman kotak tambah :

```
Private Sub Command2_Click()
    strSelect2 = "SELECT pernyataan FROM rule WHERE kode='" &
    strback & "'"
    Set rs = Conn.Execute(strSelect2)
    strSelect2 = rs.Fields(0).Value
    Baca = strSelect2
    Form1.Label2.Caption = Baca
    Form1.Visible = True
    Form5.Visible = False
    Form1.Command1.Enabled = True
    Form1.Command2.Enabled = True
    Form1.Command3.Enabled = True
    Form1.Command5.Enabled = True
    Form1.Command7.Enabled = True
    Form1.Command4.Enabled = True
End Sub
```

e. Login.frm

Gambar 61. Tampilan Halaman *Login*

Halaman *login* merupakan halaman pertama yang muncul setelah membuka aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi. Terdapat 5 tombol yaitu, “Daftar”, “Masuk”, “Batal”, “Tentang”, dan “?”. Setiap tombol berfungsi dengan baik sesuai dengan *flowchart*. Berikut adalah potongan *source code* untuk fungsi tombol “Masuk” pada halaman *login* :

```
Private Sub cmdOK_Click()
    nama = txtUserName.Text
    pssword = txtPassword.Text
    strSel = "select UserName from login where UserName='" & nama &
    "'"
    Set rs = Conn.Execute(strSel)
    If rs.BOF And rs.EOF = True Then
        MsgBox "Login gagal. Pastikan anda menuliskan nama pengguna
        dan kata sandi dengan benar", vbOKOnly + vbInformation, "Batal
        login"
    Else
        strUsername = rs.Fields(0).Value
        strSelect = "select Password from login where UserName='" &
        nama & "'"
```

```
        Set rs = Conn.Execute(strSelect)
        strPassword = rs.Fields(0).Value
        If pssword <> strPassword Then
            MsgBox "Login gagal. Pastikan anda menuliskan nama
            pengguna dan kata sandi dengan benar", vbOKOnly +
            vbInformation, "Batal login"
        Else
            strSelect = "select tipe from login where UserName='" &
            nama & "'"
            Set rs = Conn.Execute(strSelect)
            strTipe = rs.Fields(0).Value
            If strTipe = "admin" Then
                MsgBox "Login berhasil", vbOKOnly + vbInformation,
                "Berhasil login"
                strSelect = "SELECT pernyataan FROM rule WHERE
                kode='" & Code & "'"
                Set rs = Conn.Execute(strSelect)
                If rs.BOF = True Or rs.EOF = True Then
                    MsgBox "rulebase masih kosong", vbInformation,
                    "Notifikasi"
                    Form7.Visible = True
                    Form3.Visible = False
                Else
                    Form3.Visible = False
                    Form1.Visible = True
                End If
            Else
                MsgBox "Login berhasil", vbOKOnly + vbInformation,
                "Berhasil login"
                strSelect = "SELECT pernyataan FROM rule WHERE
                kode='" & Code & "'"
                Set rs = Conn.Execute(strSelect)
                If rs.BOF = True Or rs.EOF = True Then
                    MsgBox "rulebase masih kosong", vbInformation,
                    "Notifikasi"
                    End
                    Close All
                Else
                    Form3.Visible = False
                    Form9.Visible = True
                End If
            End If
        End If
    End If
End Sub
```

f. Signup.frm

Gambar 62. Tampilan Halaman *Signup*

Halaman *signup* digunakan untuk pendaftaran bagi pengguna yang belum terdaftar sebagai pengguna aplikasi ini namun hendak menggunakan aplikasi ini. Halaman *signup* muncul dengan cara klik tombol "Daftar" pada halaman *login*. Terdapat *2 textbox* dan 3 tombol pada halaman *signup* yaitu, *textbox* nama pengguna dan kata sandi, tombol "Simpan", "Kembali", dan "?". Setiap tombol berfungsi sebagai mana mestinya sesuai *flowchart*. Berikut adalah potongan *source code* dari fungsi tombol "Simpan" pada halaman *signup*:

```
Private Sub Command1_Click()
    If Len(Text1.Text) <= 15 And Len(Text2.Text) <= 15 Then
        nm = Text1.Text    'cek panjang data
        pss = Text2.Text
        strSelect7 = "SELECT * from login where UserName='" & nm & "'"
        Set rs = Conn.Execute(strSelect7)
        If rs.EOF Then      'cek ketersediaan nama pengguna
            If Text1.Text = "" Or Text2.Text = "" Then
                MsgBox "Data masih kosong. Isilah dengan lengkap!",
                vbInformation, "Notifikasi"
```

```
                Text1.Text = ""
                Text2.Text = ""
            Else
                strInput1 = "INSERT INTO login VALUES ('" & Text1.Text
                & "', '" & Text2.Text & "' ,'user')"
                Set rs = Conn.Execute(strInput1) 'input data pengguna
                MsgBox "Pendaftaran berhasil!", vbInformation, "Notifikasi"
                Form8.Visible = False
                Form9.Visible = True
            End If
        Else
            MsgBox "Maaf nama pengguna tersebut sudah terdaftar",
            vbInformation, "Notifikasi"
            Text1.Text = ""
            Text2.Text = ""
        End If
    Else
        MsgBox "Maaf data yang Anda masukkan lebih dari 15 karakter.",
        vbInformation, "Notifikasi"
        Text1.Text = ""
        Text2.Text = ""
    End If
End Sub
```

g. Sisipkan.frm

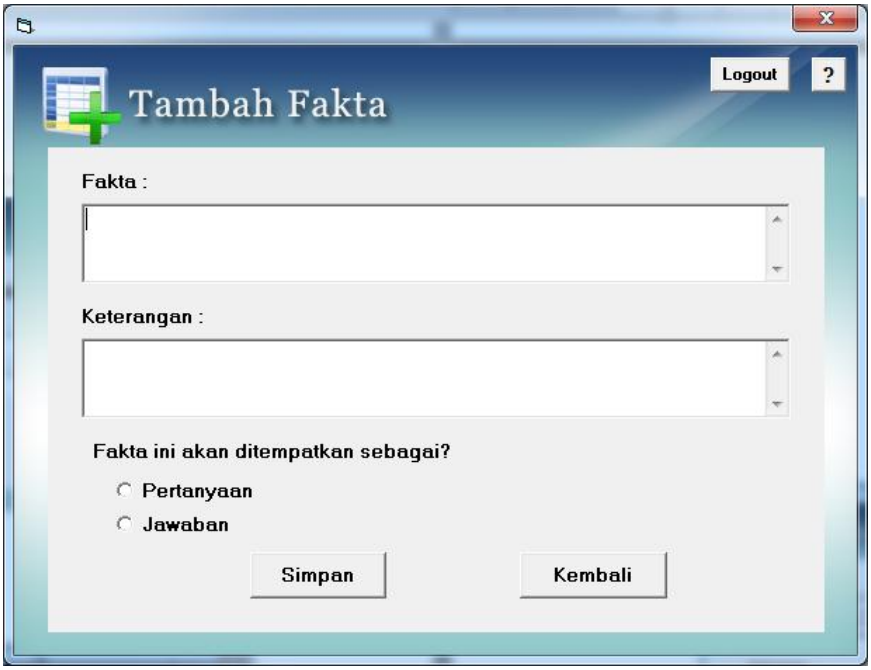
Gambar 63. Tampilan Halaman Penyisipan Fakta

Halaman penyisipan fakta digunakan untuk menyisipkan fakta diantara fakta-fakta yang sudah ada di tabel rule dalam *database* perangkat lunak ini. Berikut merupakan potongan *source code* untuk tombol “Simpan” pada halaman penyisipan fakta:

```
Private Sub Command1_Click()
    strSelect = "SELECT * from rule WHERE kode='" & Code & "' "
    Set rs = Conn.Execute(strSelect)
    If rs.BOF = False And rs.EOF = False Then
        If Text1.Text = "" Then
            MsgBox "Maaf tuliskan terlebih dahulu fakta di tempat yang
            sudah disediakan!", vbInformation, "Notifikasi"
        Else
            strSelect6 = "SELECT back FROM rule WHERE kode='" &
            Code & "'"
            Set rs = Conn.Execute(strSelect6)
            strback = rs.Fields(0).Value
            ‘menyisipkan fakta baru
            If Option1.Value = True Then
                a = True
                strInput1 = "INSERT INTO rule VALUES ('', '" &
                Text1.Text & "', '" & Text3.Text & "', '" & Code &
                "','0','P','" & strback & ", ’’)"
                Set rs = Conn.Execute(strInput1)
                strSelect6 = "SELECT kode FROM rule WHERE
                pernyataan='" & Text1.Text & "' and ya = '" & Code & "'"
                Set rs = Conn.Execute(strSelect6)
                strnilai = rs.Fields(0).Value
                ‘update data diatasnya
                If tanda = 1 Then
                    strUpdate = "UPDATE rule SET ya='" & strnilai & "'
                    WHERE kode='" & strback & "'"
                    Set rs = Conn.Execute(strUpdate)
                ElseIf tanda = 0 Then
                    strUpdate = "UPDATE rule SET tidak='" & strnilai &
                    "' WHERE kode='" & strback & "'"
                    Set rs = Conn.Execute(strUpdate)
                End If
                ‘update data dibawahnya
                strUpdate = "UPDATE rule SET back='" & strnilai & "'
                WHERE kode='" & Code & "'"
                Set rs = Conn.Execute(strUpdate)
```

```
                MsgBox "data berhasil di sisipkan", vbInformation, "Notifikasi"
            ElseIf Option2.Value = True Then
                a = True
                strInput1 = "INSERT INTO rule VALUES ('', '" & Text1.Text & "', '" & Text3.Text & "', '0','" & Code & "','P','" & strback & ",'')"
                Set rs = Conn.Execute(strInput1)
                strSelect6 = "SELECT kode FROM rule WHERE pernyataan='" & Text1.Text & "' and tidak = '" & Code & "'"
                Set rs = Conn.Execute(strSelect6)
                strnilai = rs.Fields(0).Value
                If tanda = 1 Then
                    strUpdate = "UPDATE rule SET ya='" & strnilai & "' WHERE kode='" & strback & "'"
                    Set rs = Conn.Execute(strUpdate)
                ElseIf tanda = 0 Then
                    strUpdate = "UPDATE rule SET tidak='" & strnilai & "' WHERE kode='" & strback & "'"
                    Set rs = Conn.Execute(strUpdate)
                End If
                strUpdate = "UPDATE rule SET back='" & strnilai & "' WHERE kode='" & Code & "'"
                Set rs = Conn.Execute(strUpdate)
                MsgBox "data berhasil di sisipkan", vbInformation, "Notifikasi"
            Else
                a = False
                MsgBox "Pilih salah satu Option yang tersedia", vbInformation, "Notifikasi"
            End If
            'membaca data yang baru disisipkan untuk ditampilkan
            If a = True Then
                Unload Me
                Code = strnilai
                strSelect2 = "SELECT pernyataan FROM rule WHERE kode='" & Code & "'"
                Set rs = Conn.Execute(strSelect2)
                strSelect2 = rs.Fields(0).Value
                Baca = strSelect2
                Form1.Label2.Caption = Baca
                Form1.Visible = True
            End If
        End If
    End If
End Sub
```

h. Tambah.frm



Gambar 64. Tampilan Halaman Tambah Fakta

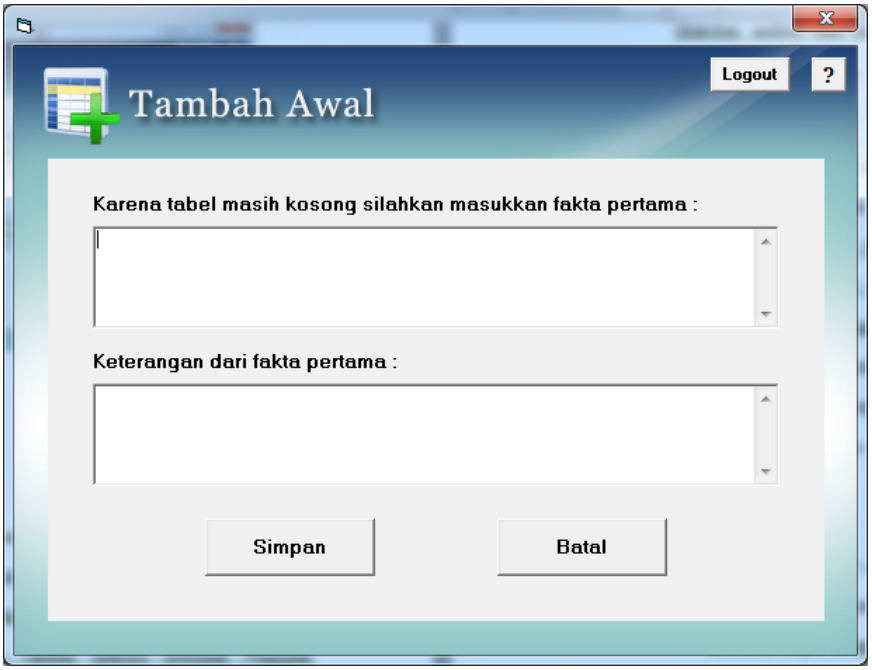
Halaman tambah fakta digunakan untuk menambahkan fakta pada ujung *rulebase*. Ujung *rulebase* yang dimaksud adalah *node* yang dituju oleh pernyataan dengan jenis pertanyaan namun data pada *node*yang dituju tersebut belum tersedia pada tabel *rule*dalam*database* perangkat lunak ini. *Node* yang kosong tersebut menyebabkan tidak adanya pernyataan yang ditampilkan pada diagnosa.

Terdapat 4 tombol pada halaman tambah fakta yaitu, tombol “Simpan”, “Kembali”, “Logout”, dan “?”. Untuk kembali ke halaman diagnosa *admin* tanpa jadi menambah fakta klik tombol “Kembali”. Untuk menambahkan fakta pengguna diminta mengisi *textbox* fakta dan keterangan, memilih jenis fakta yang akan ditambahkan kemudian klik tombol “Simpan”. Berikut adalah potongan *source code* untuk tombol “Simpan” pada halaman tambah fakta :

```
Private Sub Command1_Click()
    If Text1.Text = "" Then
        MsgBox "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada
        tempat yang telah disediakan", vbInformation, "Notifikasi"
    Else
        If strback = "1" Then       'input data baru (letak atas data semula)
            If Option1.Value = True Then
                a = True
                strInput1 = "INSERT INTO rule VALUES ('', '" &
                Text1.Text & "', '" & Text2.Text & "','0','0','P','1,’’)"
                Set rs = Conn.Execute(strInput1)
            ElseIf Option2.Value = True Then
                a = True
                strInput1 = "INSERT INTO rule VALUES ('', '" &
                Text1.Text & "', '" & Text2.Text & "','0','0','J','1,’’)"
                Set rs = Conn.Execute(strInput1)
                Form1.Command1.Enabled = False
                Form1.Command2.Enabled = False
                Form1.Command3.Enabled = True
                Form1.Command5.Enabled = True
            Else
                MsgBox "Anda belum memilih jenis masukan, pilih terlebih
                dahulu jenis data yang diinginkan", vbInformation,
                "Notifikasi"
                a = False
            End If
        Else
            If Option1.Value = True Then            ‘input data baru
                a = True
                strInput1 = "INSERT INTO rule VALUES ('', '" & Text1.Text
                & "', '" & Text2.Text & "','0','0','P','" & strback & ",’’)"
                Set rs = Conn.Execute(strInput1)
            ElseIf Option2.Value = True Then
                a = True
                strInput1 = "INSERT INTO rule VALUES ('', '" &
                Text1.Text & "', '" & Text2.Text & "','0','0','J','" & strback
                & ",’’)"
                Set rs = Conn.Execute(strInput1)
                Form1.Command1.Enabled = False
                Form1.Command2.Enabled = False
                Form1.Command3.Enabled = True
                Form1.Command5.Enabled = True
            Else
                MsgBox "Anda belum memilih jenis masukan, pilih terlebih
                dahulu jenis data yang anda inginkan", vbInformation,
                "Notifikasi"
```

```
                a = False
            End If
        End If
        If a = True Then   'mencari kode data yang baru ditambahkan
            strSelect6 = "SELECT kode FROM rule WHERE
            pernyataan='" & Text1.Text & "' and ya='0' and tidak='0'"
            Set rs = Conn.Execute(strSelect6)
            strnilai = rs.Fields(0).Value
            If tanda = 1 Then       'mengubah data sebelumnya
                strUpdate = "UPDATE rule SET ya='" & strnilai & "'
                WHERE kode='" & strback & "'"
                Set rs = Conn.Execute(strUpdate)
            ElseIf tanda = 0 Then
                strUpdate = "UPDATE rule SET tidak='" & strnilai & "'
                WHERE kode='" & strback & "'"
                Set rs = Conn.Execute(strUpdate)
            End If
            MsgBox "Fakta berhasil disimpan!", vbInformation,
            "Notifikasi"
            Unload Me
            Code = strnilai
            strSelect2 = "SELECT pernyataan FROM rule WHERE
            kode='" & Code & "'"
            Set rs = Conn.Execute(strSelect2)
            strSelect2 = rs.Fields(0).Value
            Baca = strSelect2
            Form1.Label2.Caption = Baca
            Form1.Command7.Enabled = True
            strSelect4 = "SELECT jenis from rule where pernyataan='" &
            Baca & "'"
            Set rs = Conn.Execute(strSelect4)
            strJenis = rs.Fields(0).Value
            perjab = strJenis       'membaca jenis data
            If perjab = "J" Then
                Form1.Command1.Enabled = False
                Form1.Command2.Enabled = False
                Form1.Command11.Enabled = True
            ElseIf perjab = "P" Then
                Form1.Command1.Enabled = False
                Form1.Command2.Enabled = False
                Form1.Command11.Enabled = True
            End If
            Form1.Visible = True
        End If
    End If
End Sub
```

i. Tambah_awal.frm

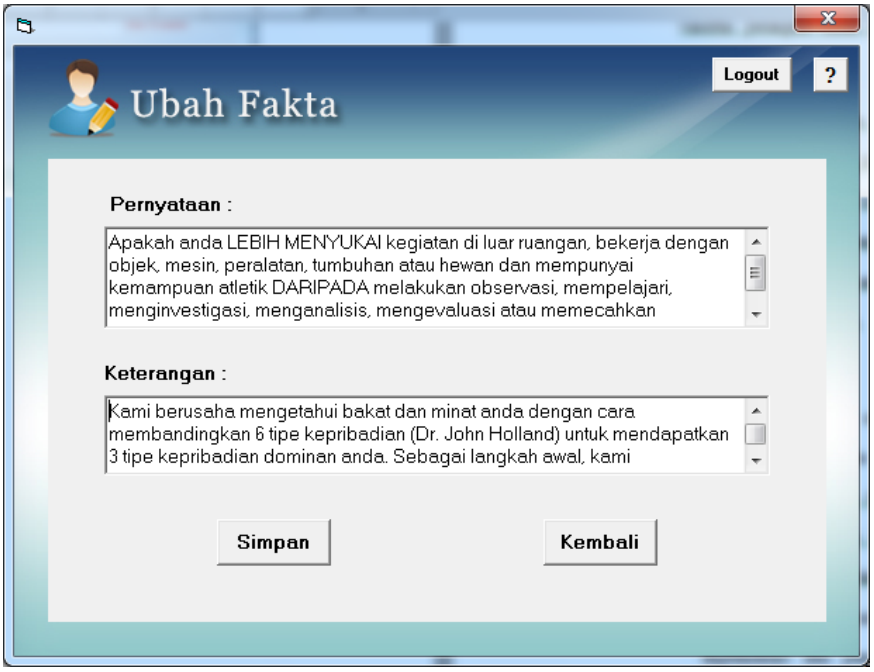


Gambar 65. Tampilan Halman Tambah Awal

Halaman tambah awal muncul ketika *admin* berhasil *login* namun data *rulebase* pada *database* masih kosong. Halaman tambah awal digunakan untuk memasukkan pernyataan awal yaitu pertanyaan yang akan ditanyakan pertama ke dalam tabel*rule*. Berikut adalah potongan *source code* untuk tombol "Simpan" yang ada pada halaman tambah awal :

```
Private Sub Command1_Click()
    If Text1.Text = "" Then
        MsgBox "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada
        tempat yang telah disediakan!", vbInformation, "Notifikasi"
    Else
        strInput1 = "INSERT INTO rule VALUES ('', '" & Text1.Text & "',
        '" & Text2.Text & "' ,'0','0','P','0','')"
        Set rs = Conn.Execute(strInput1)
        MsgBox "Fakta pertama berhasil disimpan", vbInformation,
        "Notifikasi"
        Unload Me
        Form1.Visible = True
    End If
End Sub
```

j. Ubah.frm



Gambar 66. Tampilan Halaman Ubah Fakta

Halaman ubah fakta digunakan untuk mengubah fakta yang sudah terdapat pada *database*. Berikut adalah potongan *source code* untuk tombol "Simpan" pada halaman ubah fakta :

```
Private Sub em()
    strEdit = "UPDATE rule SET pernyataan='" & Text1.Text & "'
    WHERE kode='" & Code & "' "
    Set rs = Conn.Execute(strEdit)

    strEdit1 = "UPDATE rule SET keterangan='" & Text2.Text & "'
    WHERE kode='" & Code & "' "
    Set rs = Conn.Execute(strEdit1)
    MsgBox "data berhasil di ubah", vbInformation, "Notifikasi"
End Sub

Private Sub Command1_Click()
    Call em
    Call Form1.rel
    Unload Me
    Form1.Visible = True
End Sub
```

k. Module1.bas

Aplikasi ini menggunakan*tool*MyODBC untuk koneksi antara bahasa pemrograman Visual Basicdengan *database* MySql. Module1.bas digunakan untuk mengkoneksikan *database*yang ada pada *local host*dengan perangkat lunak agar aplikasi dapat mengakses data dari *database* tersebut. Pada Module1.bas dibuat koneksi aplikasi dengan *database* jurusan yang terdapat pada *server localhost*. Dalam *database* jurusan terdapat 2 tabel yaitu, login dan rule.Tabel tersebut digunakan untuk proses diagnosa dan daftar pengguna. Berikut adalah potongan *source code* yang terdapat pada Module1.bas :

```
Public Conn As New ADODB.Connection
Public Sub buat_koneksi()
    Dim ConnString As String
    Dim db_server As String
    Dim db_port As String
    Dim db_user As String
    Dim db_pass As String
    Dim db_name As String
    On Error GoTo buat_koneksi_Error
    db_server = "localhost"
    db_port = "3306"
    db_user = "root"
    db_pass = ""
    db_name = "jurusan"
        ConnString = "DRIVER={MySQL ODBC 5.1 Driver};SERVER="
        & db_server & ";DATABASE=" & db_name & ";UID=" &
        db_user & ";PWD=" & db_pass & ";PORT=" & db_port &
        ";OPTION=3"
    With Conn
        .ConnectionString = ConnString
        .Open
    End With
    On Error GoTo 0
    Exit Sub
    buat_koneksi_Error:
        MsgBox "Ada kesalahan dengan server, periksa apakah server
        sudah berjalan !", vbInformation, "Cek Server"
End Sub
```

4. Pengujian (*Testing*)

Setelah melalui proses implementasi sistem pada tahap pengkodean maka tahap selanjutnya yang dilakukan ialah melakukan uji perangkat lunak. Pengujian perangkat lunak yang dilakukan dalam penelitian ini adalah uji verifikasi dan validitas perangkat lunak. Pengujian verifikasi perangkat lunak meliputi *white box testing* dan *black box testing*. Sedangkan pengujian validitas perangkat lunak meliputi pengujian *alpha* dan pengujian *beta*.

Pengujian validitas yang dilakukan mengacu pada ISO 9126 tentang *software quality factors*. Uji validitas perangkat lunak berfokus pada faktor kualitas*usability*. Dalam pengujian faktor kualitas *usability*digunakan angket penelitian sebagai instrumen pengumpulan data. Setelah data terkumpul kemudian dilakukan analisis data untuk mengetahui tingkat kelayakan perangkat lunak yang dikembangkan pada faktor kualitas *usability*.

a. White box Testing

Pengujian verifikasi perangkat lunak yang pertama adalah pengujian *white box.White box testing* merupakan proses pengujian perangkat lunak dari sisi desain dan kode program. Program diuji apakah mampu menghasilkan fungsi-fungsi yang sesuai dengan kebutuhan dan tidak mengalami kesalahan dari sisi program. Pengujian ini dilakukan dengan melakukan pemeriksaan *logic* dari kode program. Pengujian ini dilakukan secara langsung pada saat proses implementasi atau pengkodean perangkat lunak oleh pengembang. Pengujian ini dilakukan dengan cara uji faktor kualitas *correctness*.

b. Black Box Testing

Pengujian verifikasi perangkat lunak setelah pengujian *white box* adalah pengujian *black box. Black box testing* merupakan proses pengujian perangkat lunak dari segi spesifikasi fungsional program. Pengujian ini dilakukan tanpa menguji desain dan kode program. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran perangkat lunak sudah sesuai dengan spesifikasi yang diperlukan.Pada tahapan pengujian *black box*dilakukan pengujian*test case*. Hasil uji *test case* pada *black box testing* dibahas pada analisis faktor kualitas *functionality*.

c. Alpha Testing

Tahapan uji *alpha* merupakan salah satu tahapan *validation testing* yang dilakukan oleh pihak pengembang atau ahli. Pengujian *alpha*pada penelitian ini dilakukan oleh ahli pada lingkungan pengembang yang memadai. Ahli melakukan pengujian *alph*a kepada perangkat lunak yang dikembangkan untuk mengetahui permasalahan perangkat lunak pada lingkungan yang paling memadai dalam hal ini lingkungan perspektif pengembang.

Pada tahapan uji *alpha* ini penguji ahli akan memberikan *report* berupa kesalahan-kesalahan yang terjadi serta penyelesaian masalah bagi pengembangan perangkat lunak sebelum dilanjutkan ke pengujian *beta.*Proses pengujian perangkat lunak ini dilakukan pengujian *alpha* dengan melibatkan seorang ahli dalam bidang rekayasa perangkat lunak. Hasil dari pengujian *alpha* ialah seperti berikut ini :

Tabel 39.Uji *Alpha*Spesifikasi Uji Membuka Aplikasi

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Membuka Aplikasi	Apabila MySQL belum aktif maka akan muncul notifikasi “Ada kesalahan dengan server, periksa apakah server sudah berjalan !”.	√	
		Pengguna masuk ke halaman *login* dan muncul menu fitur *login* (beserta *textbox* nama pengguna dan kata sandi), daftar, *about* (tentang), *help* (bantuan), dan batal.	√	

Tabel 40.Uji *Alpha* Spesifikasi Uji Halaman *Login*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Halaman Login	Pengguna dapat masuk ke semua fitur menu (daftar, tentang, batal, dan bantuan) dengan memilih tombol-tombol tersebut yang ada pada halaman *login*	√	
2	Login	Pengguna dapat *login* dengan cara menuliskan data diri pada *textbox* nama pengguna dan *textbox*kata sandi kemudian memilih tombol masuk	√	
		Apabila pengguna memilih tombol masuk ketika *textbox* nama pengguna atau kata sandi atau keduanya belum terisi maka akan muncul notifikasi “login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar”	√	
		Apabila data diri yang ditulis oleh pengguna tidak cocok dengan data pengguna yang tersimpan dalam *database* maka akan muncul notifikasi “login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar”	√	
		Pengguna dapat kembali ke halaman *login* dengan cara memilih tombol ok pada notifikasi “login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar”	√	
		Apabila data diri yang ditulis oleh pengguna cocok dengan data pengguna yang tersimpan dalam *database* maka akan muncul notifikasi “login berhasil”	√	

Tabel 41.Uji *Alpha* Spesifikasi Uji Halaman *Login*(lanjutan)

3	Keluar aplikasi	Apabila pengguna memilih tombol batal pada halaman *login* maka akan muncul notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"	√	
		Pengguna dapat keluar aplikasi dengan memilih tombol ya pada notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"	√	
		Pengguna dapat batal keluar aplikasi dengan cara memilih tombol tidak pada notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"	√	

Tabel 42.Uji *Alpha* Spesifikasi UjiHalaman *About*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman *about*	Pengguna dapat masuk ke halaman*about* dengan cara memilih tombol tentang pada halaman *login*	√	
2	Halaman *about*	Muncul label keterangan yang berisi nama aplikasi, versi, diskripsi singkat dan nama pengembang	√	
		Pengguna dapat masuk ke semua fitur menu (info sistem dan bantuan) dengan memilih tombol-tombol tersebut yang ada pada halaman *about*	√	
3	Fitur info sistem	Pengguna dapat melihat info sistem dengan cara memilih tombol info sistem pada halaman *about*	√	
		Jendela info sistem dapat ditutup dengan cara memilih *icon* tutup (x) pada jendela info sistem	√	
4	Keluar halaman *about*	Pengguna dapat keluar dari halaman *about* dengan memilih tombol ok pada halaman *about*, maka akan muncul halaman *login*	√	

Tabel 43.Uji *Alpha* Spesifikasi UjiHalaman Pendaftaran

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman pendaftaran	Pengguna dapat masuk ke halamanpendaftaran dengan cara memilih tombol daftar pada halaman *login*	√	
2	Pendaftaran	Pengguna dapat melakukan pendaftaran dengan mengisi data diri pada *textbox* nama pengguna dan *textbox*kata sandi kemudian memilih tombol simpan	√	
		Apabila data pengguna belum lengkap maka akan muncul notifikasi “Data masih kosong. Isilah dengan lengkap”	√	
		Apabila nama pengguna atau kata sandi yang dituliskan lebih dari 15 karakter maka akan muncul notifikasi “Maaf data yang anda masukkan lebih dari 15 karakter”	√	
		Apabila nama pengguna yang dituliskan telah terdaftar dalam database maka akan muncul notifikasi “Maaf nama pengguna tersebut sudah terdaftar”	√	
		Pengguna dapat kembali ke halaman pendaftaran dengan memilih tombol ok pada setiap notifikasi peringatan gagal pendaftaran	√	
		Apabila data pengguna sudah lengkap, tidak ada data yang lebih dari 15 karakter, dan nama pengguna yang dituliskan belum terdaftar dalam database maka data pengguna tersebut akan tersimpan dalam database dan akan muncul notifikasi “Pendaftaran berhasil!”	√	
		Pengguna dapat memasuki halaman diagnosa untuk *normal user* dengan cara memilih tombol ok pada notifikasi “Pendaftaran berhasil!”	√	
3	Keluar halaman pendaftaran	Pengguna dapat keluar dari halaman pendaftaran dengan memilih tombol kembali pada halaman pendaftaran, maka akan muncul halaman *login*	√	

Tabel 44. Uji *Alpha* Spesifikasi Uji Halaman Diagnosa Untuk *Normal User*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman diagnosa untuk *normal user*	Pengguna dapat masuk ke halaman diagnosa untuk *normal user* dengan cara berhasil *login* atau daftar dengan data tipe pengguna "user"	√	
2	Diagnosa *normal user*	Muncul pernyataan awal pada label pernyataan untuk diagnosa awal, serta tombol ya, tidak, kenapa?, bantuan, dan logout (setiap tombol berfungsi dengan benar dan melakukan tugas masing-masing fitur)	√	
		Pengguna dapat menjawab pernyataan yang muncul dengan cara memilih tombol ya atau tidak, sesuai dengan keadaan pengguna, maka akan muncul pernyataan selanjutnya (sesuai jawaban yang dipilih)	√	
		Apabila pengguna sudah menjawab pernyataan maka tombol kembali akan muncul dan dapat dipilih	√	
		Apabila data untuk pernyataan selanjutnya belum tersedia padahal data sebelumnya merupakan jenis data pertanyaan maka akan muncul notifikasi "Maaf data masih kosong"	√	
		Pengguna dapat kembali ke halaman diagnosa dengan cara memilih tombol ok pada notifikasi "Maaf data masih kosong"	√	
		Pengguna akan mendapatkan solusi dari diagnosa jika tidak terdapat lagi pernyataan yang perlu dijawab oleh pengguna karena pernyataan tersebut merupakan jenis data jawaban	√	
		Apabila pengguna mendapatkan solusi diagnosa (jenis data jawaban) maka tombol jawab (ya dan tidak) tidak dapat di pilih lagi, sedangkan tombol detail info muncul dan dapat dipilih	√	
		Apabila data *rulebase* pada *database* masih kosong maka akan muncul notifikasi "rulebase masih kosong"	√	
		Aplikasi akan tertutup apabila pengguna memilih tombol ok pada notifikasi "rulebase masih kosong"	√	
3	Ulang diagnosa	Pengguna dapat melakukan diagnosa ulang dengan cara memilih tombol ulang pada halaman diagnosa. Maka label pernyataan pada halaman diagnosa akan kembali pada pernyataan awal proses diagnosa	√	

Tabel 45.Uji *Alpha* Spesifikasi Uji Halaman Diagnosa Untuk *Admin*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman diagnosa untuk *admin*	Pengguna dapat masuk ke halaman diagnosa untuk *admin* dengan cara berhasil *login* sebagai akun dengan data tipe pengguna "admin"	√	
2	Diagnosa *admin*	Muncul pernyataan awal pada label pernyataan untuk diagnosa awal, serta tombol ya, tidak, kenapa?, ubah, kosongkan, bantuan, dan *logout* (setiap tombol berfungsi dengan benar dan melakukan tugas masing-masing fitur)	√	
		Pengguna dapat menjawab pernyataan yang muncul dengan cara memilih tombol ya atau tidak, sesuai dengan keadaan pengguna, maka akan muncul pernyataan selanjutnya (sesuai dengan jawaban yang dipilih)	√	
		Apabila pengguna sudah menjawab pernyataan maka tombol kembali, hapus, sisipkan fakta, dan ulang akan muncul dan dapat dipilih (setiap tombol dapat melakukan tugas fitur masing-masing dengan benar)	√	
		Apabila data untuk pernyataan selanjutnya belum tersedia padahal data sebelumnya merupakan jenis data pertanyaan maka akan muncul notifikasi "Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?"	√	
		Pengguna dapat masuk ke halaman tambah fakta dengan cara memilih tombol ya pada notifikasi "Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?"	√	
		Pengguna dapat kembali ke halaman diagnosa *admin* tanpa menambah fakta baru dengan cara memilih tombol tidak pada notifikasi "Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?", maka akan muncul pernyataan sebelumnya pada halaman diagnosa	√	
		Pengguna akan mendapatkan solusi dari diagnosa jika tidak terdapat lagi pernyataan yang perlu dijawab oleh pengguna karena pernyataan tersebut merupakan jenis data jawaban	√	

Tabel 46.Uji *Alpha* Spesifikasi Uji Halaman Diagnosa Untuk *Admin* (lanjutan)

		Apabila pengguna mendapatkan solusi diagnosa (jenis data jawaban) maka tombol jawab (ya dan tidak) tidak dapat di pilih lagi, sedangkan tombol detail info muncul dan dapat dipilih (melakukan tugas fitur detail info)	√	
		Apabila data *rulebase* pada *database* masih kosong maka akan muncul notifikasi "rulebase masih kosong"	√	
		Pengguna dapat masuk ke halaman tambah awal untuk memasukkan data awal pada *rulebase* dengan cara memilih tombol ok pada notifikasi "rulebase masih kosong"	√	
3	Ulang diagnosa	Pengguna dapat melakukan diagnosa ulang dengan cara memilih tombol ulang pada halaman diagnosa. Maka label pernyataan pada halaman diagnosa akan kembali pada pernyataan awal proses diagnosa	√	
4	Fitur ubah data	Pengguna dapat masuk ke halaman ubah data untuk mengubah data yang muncul dalam label pernyataan dengan cara memilih tombol ubah pada halaman diagnosa *admin*	√	
5	Fitur hapus data	Pengguna dapat menghapus data yang muncul dalam label pernyataan (beserta anak-anaknya) dengan cara memilih tombol hapus pada halaman diagnosa *admin*, maka akan muncul notifikasi "Data berhasil dihapus!"	√	
		Pengguna akan kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi "Data berhasil dihapus!", maka pada halaman diagnosa akan muncul pernyataan yang ditampilkan sebelum pernyataan yang dihapus tersebut	√	
6	Fitur sisipkan fakta	Pengguna dapat masuk ke halaman penyisipan fakta untuk menyisipkan data diantara data yang telah tersedia dengan cara memilih tombol sisipkan fakta pada halaman diagnosa *admin*	√	
7	Fitur kosongkan rulebase	Pengguna dapat mengosongkan data yang terdapat pada tabel *rulebase* di *database* dengan cara memilih tombol kosongkan yang ada di halaman diagnosa *admin*, maka akan muncul notifikasi "Tabel rule berhasil dikosongkan"	√	
		Muncul halaman login setelah memilih tombol ok pada notifikasi "Tabel rule berhasil dikosongkan"	√	

Tabel 47. Uji *Alpha* Spesifikasi Uji Fitur Kembali Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur kembali pada halaman diagnosa	Tombol kembali pada halaman diagnosa (untuk *normal user* dan *admin*) akan muncul jika pengguna telah menjawab pernyataan	√	
		Pengguna dapat mengulang pernyataan sebelumnya dengan cara memilih tombol kembali pada halaman diagnosa	√	
		Apabila pengguna memilih tombol kembali padahal pernyataan yang dituju merupakan awal data maka akan muncul pemberitahuan "Anda saat ini berada di awal data"	√	
		Pengguna dapat melanjutkan proses diagnosa dengan cara memilih tombol ok pada notifikasi "Anda saat ini berada di awal data", maka akan muncul kembali halaman diagnosa	√	
		Tombol kembali pada halaman diagnosa (untuk *normal user* dan *admin*) akan hilang jika berada di awal data pernyataan diagnosa	√	

Tabel 48.Uji *Alpha* Spesifikasi Uji Fitur Detail Info Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur detail info pada halaman diagnosa	Tombol detail info pada halaman diagnosa (untuk *normal user* dan *admin*) akan muncul hanya jika proses diagnosa berada pada solusi atau kesimpulan diagnosa (jenis data jawaban)	√	
		Pengguna dapat melihat detail info mengenai hasil kesimpulan diagnosa dengan cara memilih tombol detail info pada halaman diagnosa, maka akan muncul *frame* info penyedia jurusan yang menjadi hasil kesimpulan tersebut	√	
		Pengguna dapat kembali ke halaman diagnosa dengan cara memilih *icon* tutup (x) pada *frame* info jurusan	√	
		Tombol detail info pada halaman diagnosa (untuk *normal user* dan *admin*) akan hilang jika berada pada pernyataan diagnosis dengan jenis data pertanyaan	√	

Tabel 49. Uji *Alpha* Spesifikasi Uji Fitur Mengapa Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur mengapa pada halaman diagnosa	Tombol kenapa? akan selalu muncul pada halaman diagnosa (untuk *normal user* dan *admin*)	√	
		Pengguna dapat menanyakan mengapa pernyataan tersebut diberikan dengan cara memilih tombol kenapa? pada halaman diagnosa, maka akan muncul *frame* penjelasan mengenai pernyataan tersebut	√	
		Pengguna dapat kembali ke halaman diagnosa dengan memilih *icon* tutup (x) pada *frame* penjelasan	√	

Tabel 50. Uji *Alpha* Spesifikasi UjiHalaman Ubah Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman ubah fakta	Pengguna dapat masuk ke halaman ubah fakta dengan cara memilih tombol ubah pada halaman diagnosa *admin*	√	
2	Ubah fakta	Muncul data yang akan diubah pada label pernyataan dan label keterangan pada halaman ubah fakta	√	
		Pengguna dapat langsung mengubah data pada label pernyataan dan label keterangan sesuai keinginan	√	
		Perubahan data akan tersimpan di *database* dengan cara memilih tombol simpan yang ada di halaman ubah fakta, maka akan muncul notifikasi “Data berhasil diubah”	√	
		Pengguna dapat kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi “Data berhasil diubah”, maka akan muncul data yang telah berhasil diubah tersebut pada label pernyataan di halaman diagnosa *admin*	√	
3	Batal ubah fakta	Pengguna dapat kembali ke halaman diagnosa *admin* tanpa jadi mengubah fakta dengan cara memilih tombol kembali pada halaman ubah fakta, maka akan muncul data yang tidak jadi diubah pada label pernyataan di halaman diagnosa *admin*	√	

Tabel 51. Uji *Alpha* Spesifikasi Uji Halaman Penyisipan Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman penyisipan fakta	Pengguna dapat masuk ke halaman penyisipan fakta dengan cara memilih tombol sisipkan fakta pada halaman diagnosa *admin*	√	
2	Penyisipan fakta	Muncul data kode pernyataan yang muncul saat tombol sisipkan fakta pada halaman diagnosa *admin* dipilih (pernyataan yang nantinya akan diletakkan dibawah data sisipan baru)	√	
		Pengguna dapat mengisi pernyataan dan keterangan dari fakta yang akan disisipkan pada *textbox* pertanyaan dan keterangan yang telah disediakan	√	
		Pengguna dapat memilih letak data pernyataan yang muncul saat tombol sisipkan fakta pada halaman diagnosa *admin* dipilih, yaitu pada pilihan ya atau tidak untuk jawaban dari fakta yang akan disisipkan	√	
		Data yang akan disisipkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman penyisipan fakta	√	
		Apabila pernyataan belum dituliskan namun pengguna memilih tombol simpan, maka akan muncul notifikasi "Maaf tuliskan terlebih dahulu fakta di tempat yang sudah disediakan!"	√	
		Pengguna dapat kembali ke halaman penyisipan fakta dengan cara memilih tombol ok pada notifikasi "Maaf tuliskan terlebih dahulu fakta di tempat yang sudah disediakan!"	√	
		Apabila pengguna belum memilih letak data sebelumnya namun pengguna memilih tombol simpan, maka akan muncul notifikasi "Pilih salah satu Option yang tersedia"	√	
		Pengguna dapat kembali ke halaman penyisipan fakta dengan cara memilih tombol ok pada notifikasi "Pilih salah satu Option yang tersedia"	√	
		Apabila pernyataan telah ditulis dan pilihan letak data sebelumnya telah dipilih maka data tersimpan dalam *database* dan muncul notifikasi “data berhasil di sisipkan"	√	

Tabel 52. Uji *Alpha* Spesifikasi Uji Halaman Penyisipan Fakta (lanjutan)

		Pengguna dapat kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi “data berhasil di sisipkan", maka akan muncul data yang telah berhasil disisipkan pada label pernyataan di halaman diagnosa *admin*	√	
3	Batal menyisipan fakta	Pengguna dapat melanjutkan proses diagnosa kembali di halaman diagnosa *admin* tanpa jadi menyisipkan fakta dengan cara memilih tombol kembali pada halaman penyisipan fakta	√	

Tabel 53. Uji *Alpha* Spesifikasi Uji Halaman Tambah Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman tambah fakta	Pengguna dapat masuk ke halaman tambah fakta dengan cara memilih tombol ya pada notifikasi “Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?” pada halaman diagnosa *admin*	√	
2	Penambahan fakta	Pengguna dapat mengisi pernyataan dan keterangan dari fakta yang akan ditambahkan pada *textbox* fakta dan *textbox* keterangan	√	
		Pengguna dapat memilih jenis fakta yang akan ditambahkan, yaitu jenis pertanyaan atau jawaban	√	
		Data yang akan ditambahkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman tambah fakta	√	
		Apabila pernyataan belum dituliskan kemudian pengguna memilih tombol simpan maka akan muncul notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"	√	
		Pengguna dapat kembali ke halaman tambah fakta dengan cara memilih tombol ok pada notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"	√	
		Apabila pengguna belum memilih jenis fakta yang akan ditambah namun memilih tombol simpan maka akan muncul notifikasi "Anda belum memilih jenis masukan, pilih terlebih dahulu jenis data yang diinginkan"	√	

Tabel 54. Uji *Alpha* Spesifikasi Uji Halaman Tambah Fakta (lanjutan)

		Pengguna dapat kembali ke halaman tambah fakta dengan cara memilih tombol ok pada notifikasi "Anda belum memilih jenis masukan, pilih terlebih dahulu jenis data yang diinginkan"	√	
		Apabila pernyataan telah ditulis dan jenis fakta yang akan ditambahkan telah dipilih maka data tersimpan dalam *database* dan muncul notifikasi "Fakta berhasil disimpan!"	√	
		Pengguna dapat kembali ke halaman diagnosa admin dengan cara memilih tombol ok pada notifikasi "Fakta berhasil disimpan!", maka akan muncul data yang telah ditambahkan pada label pernyataan di halaman diagnosa *admin*	√	
3	Batal tambah fakta	Pengguna dapat melanjutkan proses diagnosa kembali di halaman diagnosa *admin* tanpa jadi menambahkan fakta dengan cara memilih tombol kembali pada halaman penyisipan fakta	√	

Tabel 55.Uji *Alpha*Spesifikasi Uji FiturBantuan (*Help*)

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Bantuan	Tombol bantuan dengan *icon* “?” muncul di setiap halaman aplikasi	√	
		Pengguna dapat melihat halaman bantuan yang berisi petunjuk mengenai penggunaan halaman yang sedang diakses dengan cara memilih *icon* bantuan (?) yang ada pada halaman tersebut	√	
		Pengguna dapat kembali ke halaman yang sedang diakses dengan memilih *icon* tutup (x) pada halaman bantuan	√	
		Khusus pada halaman bantuan utama untuk halaman diagnosa *admin* pengguna dapat masuk ke 2 jenis kategori petunjuk. Kategori tersebut adalah diagnosis dan pengaturan	√	
		Pengguna pada halaman kategori bantuan untuk halaman diagnosis *admin* dapat kembali ke halaman bantuan utama dengan memilih *icon* tutup (x) pada halaman kategori bantuan tersebut	√	

Tabel 56.Uji *Alpha* Spesifikasi UjiHalaman Tambah Awal

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman tambah awal	Pengguna dapat masuk ke halaman tambah awal dengan cara memilih tombol ok pada notifikasi “rulebase masih kosong” pada halaman diagnosa *admin*	√	
2	Isi fakta awal	Pengguna dapat mengisi pernyataan dan keterangan untuk fakta awal pada *textbox* fakta dan *textbox* keterangan	√	
		Data yang akan dimasukkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman tambah awal	√	
		Apabila pernyataan belum dituliskan namun pengguna memilih tombol simpan maka akan muncul notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan!"	√	
		Pengguna dapat kembali ke halaman tambah awal dengan cara memilih tombol ok pada notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"	√	
		Apabila pernyataan telah ditulis dan pengguna memilih tombol simpan maka data tersimpan dalam *database* dan muncul notifikasi "Fakta pertama berhasil disimpan"	√	
		Pengguna dapat masuk kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi "Fakta pertama berhasil disimpan"	√	
3	Keluar halaman tambah fakta	Pengguna dapat tidak menambahkan data awal dengan cara memilih tombol batal pada halaman tambah awal, maka akan muncul halaman *login*	√	

Tabel 57.Uji *Alpha*Spesifikasi UjiFitur *Logout*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	*Logout*	Tombol *logout* muncul di setiap halaman setelah pengguna berhasil *login*	√	
		Pengguna dapat *logout* dengan cara memilih tombol *logout*, maka akan muncul kembali ke halaman *login*	√	

Berdasarkan tabel hasil pengujian *alpha* diatas, penguji menyimpulkan bahwa perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi memiliki unjuk kerja yang baik.Selain itu penguji menemukan apabila pengguna telah mengetikkan nama dan kata sandi di halaman login namun pengguna tersebut sebenarnya ingin mendaftar, aplikasi membuka halaman pendaftaran namun dengan *textbox* nama pengguna dan kata sandi yang masih kosong. Berdasarkan penemuan tersebut perangkat lunak direvisi untuk bisa menampilkan nama pengguna dan kata sandi yang telah diketik pada halaman *login* ke halaman pendaftaran karena memilih tombol daftar bukan masuk.

Dengan kesimpulan unjuk kerja yang baik dan dengan adanya revisi *textbox* pendaftaran maka perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi telah siap untuk menempuh proses pengujian validitas yang ke dua yaitu *beta testing.*

d. Beta Testing

Proses uji *beta* merupakan tahapan pengujian yang dilakukan oleh pengguna yang dipilih (*selected users*). Pengguna terpilih tersebut nantinya akan menguji perangkat lunak untuk mengetahui kualitas perangkat lunak pada aspek *usability*. Pada proses uji *beta* ini pengguna melakukan pengujian menggunakan bantuan kuesioner yang mengacu pada *Computer System Usability Questionnaire (CSUQ)* yang dikembangkan oleh J.R. Lewis. Proses pengumpulan data pada uji *beta* dilakukan oleh 60 pengguna yang terdiri dari 15 Mahasiswa di Yogyakarta dan 45 Siswa Kelas XII SMA N 2 Wates. Hasil pengujian *beta testing* selanjutnya dibahas pada bagian analisa faktor kualitas *usability*.

B. Analisis Kualitas Perangkat Lunak

Setelah proses pengembangan perangkat lunak selesai, proses selanjutnya adalah proses analisis kualitas perangkat lunak. Dalam penelitian ini terbatas pada beberapa faktor kualitas perangkat lunak yaitu : *correctness*, *functionality*, *usability*, dan*maintainability*.

1. Analisis Faktor Kualitas *Correctness*

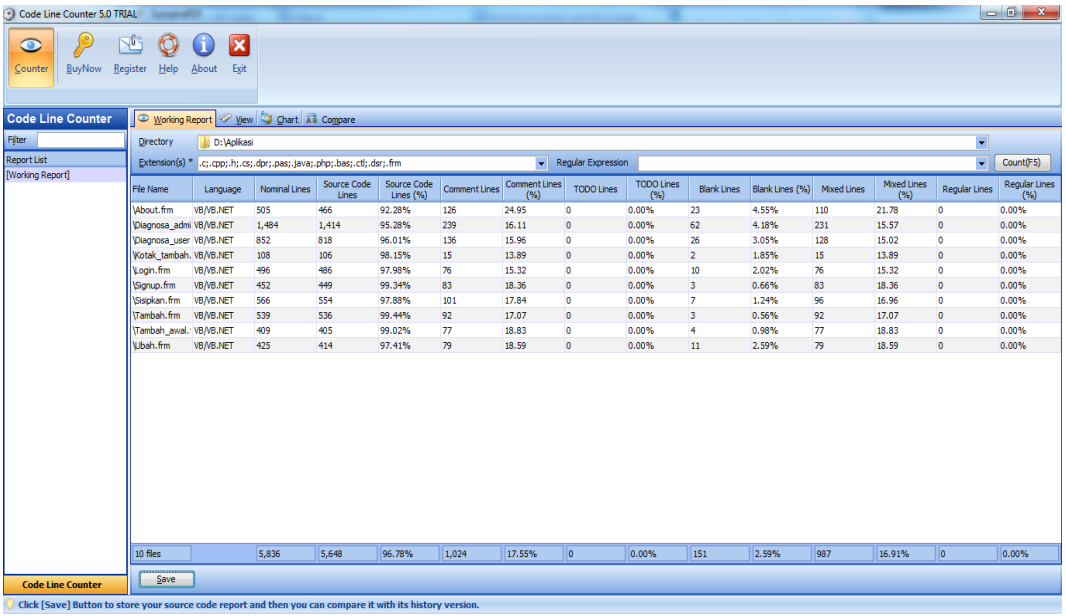
Faktor kualitas *correctness* dapat diukur dengan analisisis*defect per KLOC* (cacat/*error*pada setiap *KLOC*/*Kilo Line of Code*). Untuk mendapatkan nilai *error per KLOC*, diperlukan penghitungan jumlah *Kilo Lines of Code* (*KLOC*), kemudian dilakukan perhitungan jumlah *error*pada *source code* sistem menggunakan *Code Advisor for Visual Basic 6*.

a. Menghitung Jumlah *Lines Of Code* (LOC)

Untuk menghitung jumlah *lines of code (LOC)*, digunakan *tool Code Line Counter*. Penghitungan *lines of code (LOC)* dilakukan pada *source code* Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi. Berikut adalah daftar *file*yang dihitung :

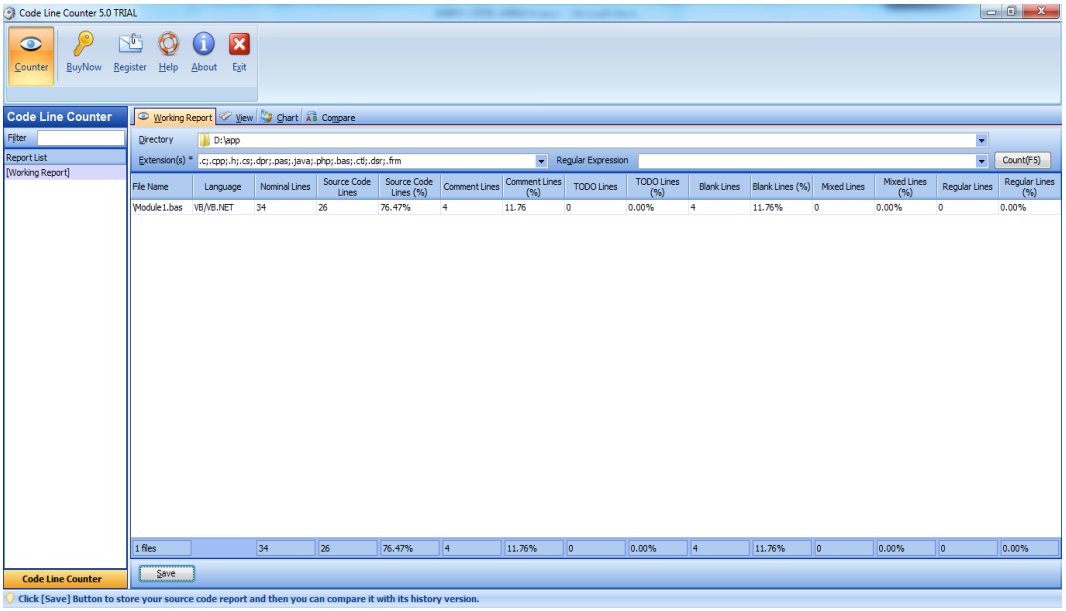
1) About.frm
2) Diagnosa_admin.frm
3) Diagnosa_user.frm
4) Kotak_tambah.frm
5) Login.frm
6) Signup.frm
7) Sisipkan.frm
8) Tambah.frm
9) Tambah_awal.frm
10) Ubah.frm
11) Module1.bas

Screenshoot hasil penghitungan *lines of code (LOC) file source code*Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi dapat dilihat pada gambar 67 dan gambar 68.



File Name	Language	Nominal Lines	Source Code Lines	Source Code Lines (%)	Comment Lines	Comment Lines (%)	TODO Lines	TODO Lines (%)	Blank Lines	Blank Lines (%)	Mixed Lines	Mixed Lines (%)	Regular Lines	Regular Lines (%)
\About.frm	VB/VB.NET	505	466	92.28%	126	24.95	0	0.00%	23	4.55%	110	21.78	0	0.00%
\Diagnosa_admi	VB/VB.NET	1,484	1,414	95.28%	239	16.11	0	0.00%	62	4.18%	231	15.57	0	0.00%
\Diagnosa_user	VB/VB.NET	852	818	96.01%	136	15.96	0	0.00%	26	3.05%	128	15.02	0	0.00%
\Kotak_tambah.	VB/VB.NET	108	106	98.15%	15	13.89	0	0.00%	2	1.85%	15	13.89	0	0.00%
\Login.frm	VB/VB.NET	496	486	97.98%	76	15.32	0	0.00%	10	2.02%	76	15.32	0	0.00%
\Signup.frm	VB/VB.NET	452	449	99.34%	83	18.36	0	0.00%	3	0.66%	83	18.36	0	0.00%
\Sisipkan.frm	VB/VB.NET	566	554	97.88%	101	17.84	0	0.00%	7	1.24%	96	16.96	0	0.00%
\Tambah.frm	VB/VB.NET	539	536	99.44%	92	17.07	0	0.00%	3	0.56%	92	17.07	0	0.00%
\Tambah_awal.	VB/VB.NET	409	405	99.02%	77	18.83	0	0.00%	4	0.98%	77	18.83	0	0.00%
\Ubah.frm	VB/VB.NET	425	414	97.41%	79	18.59	0	0.00%	11	2.59%	79	18.59	0	0.00%
10 files		5,836	5,648	96.78%	1,024	17.55%	0	0.00%	151	2.59%	987	16.91%	0	0.00%

Gambar 67. Perhitungan Jumlah LOC (1)



File Name	Language	Nominal Lines	Source Code Lines	Source Code Lines (%)	Comment Lines	Comment Lines (%)	TODO Lines	TODO Lines (%)	Blank Lines	Blank Lines (%)	Mixed Lines	Mixed Lines (%)	Regular Lines	Regular Lines (%)
\Module1.bas	VB/VB.NET	34	26	76.47%	4	11.76%	0	0.00%	4	11.76%	0	0.00%	0	0.00%
1 files		34	26	76.47%	4	11.76%	0	0.00%	4	11.76%	0	0.00%	0	0.00%

Gambar 68. Perhitungan Jumlah LOC (2)

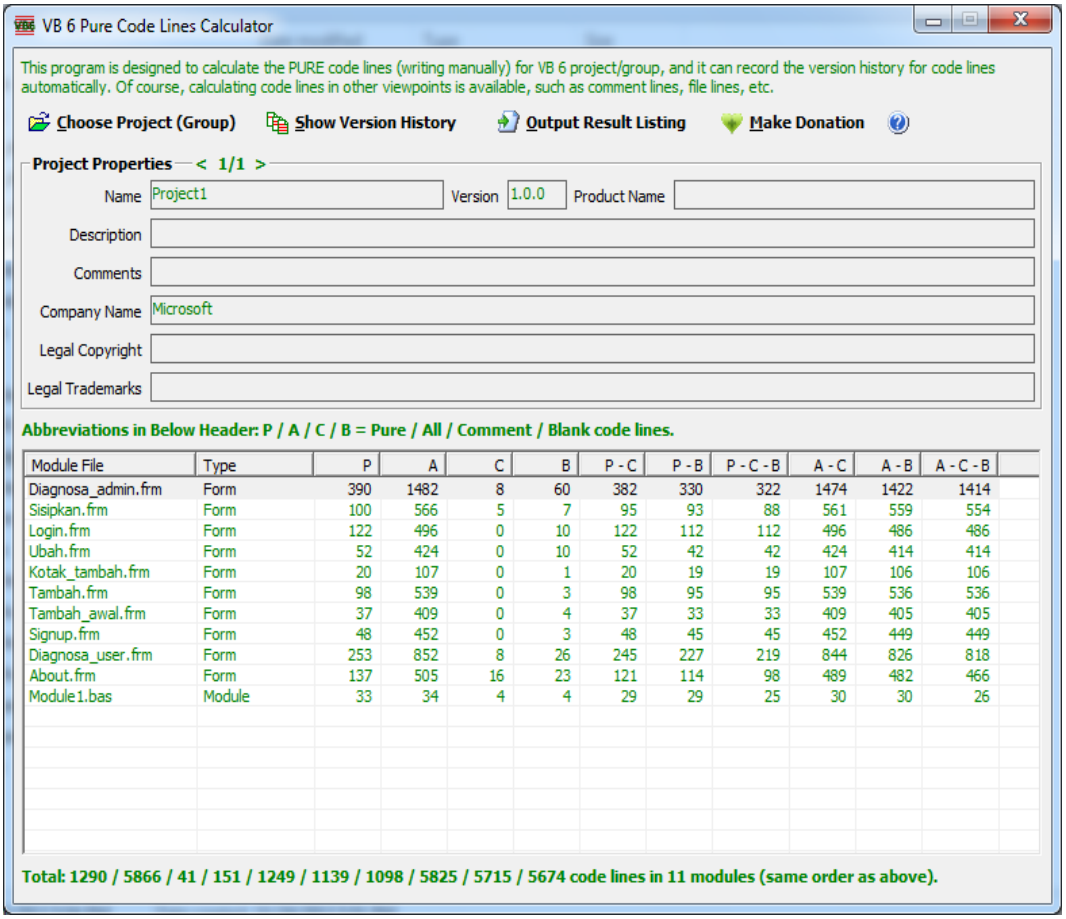
Berdasarkan perhitungan menggunakan instrumen*Code Line Counter*, *lines of code* (LOC) *source code*perangkat lunak Sistem PendukungKeputusan Pemilihan Jurusan di Perguruan Tinggi adalah 5648LOC + 26 LOC = 5,674 KLOC (*Kilo lines of code*).

Keabsahan hasil analisa sangat diperlukan dalam suatu penelitian. Oleh karena itu hasil dari perhitungan jumlah *lines of code* (LOC) *source code*perangkat lunak Sistem PendukungKeputusan Pemilihan Jurusan di Perguruan Tinggidengan *Code Line Counter*sebagai instrumen pengumpulan data harus diuji validitasnya. Validitas merupakan suatu standar ukur yang menunjukkan tingkat ketepatan suatu alat ukur. Alat ukur dapat dikatakan valid jika dapat digunakan untuk mengukur apa yang seharusnya diukur. Oleh karena itu peneliti melakukan uji validitas dengan menggunakan *tool* lain yakni *VB6 Pure Code Lines Calculator*. Hasil perhitungan jumlah *lines of code*menggunakan *tool VB6 Pure Code Lines Calculator* dapat dilihat pada gambar 69.

Berdasarkan hasil uji validitas tersebut maka dapat dijabarkan perbandingan hasil perhitungan jumlah*lines of code*perangkat lunak yang dihitung menggunakan*Code Line Counter* dan *VB6 Pure Code Lines Calculator*adalah sebagai berikut :

Tabel 58. Penjabaran Hasil Uji Validitas

Instrumen	**Hasil Perhitungan**	**Keterangan**
Code Line Counter	5674 LOC = 5,674 KLOC	Valid
VB6 Pure Code Lines Calculator	5674 LOC = 5,674 KLOC	



Module File	Type	P	A	C	B	P - C	P - B	P - C - B	A - C	A - B	A - C - B
Diagnosa_admin.frm	Form	390	1482	8	60	382	330	322	1474	1422	1414
Sisipkan.frm	Form	100	566	5	7	95	93	88	561	559	554
Login.frm	Form	122	496	0	10	122	112	112	496	486	486
Ubah.frm	Form	52	424	0	10	52	42	42	424	414	414
Kotak_tambah.frm	Form	20	107	0	1	20	19	19	107	106	106
Tambah.frm	Form	98	539	0	3	98	95	95	539	536	536
Tambah_awal.frm	Form	37	409	0	4	37	33	33	409	405	405
Signup.frm	Form	48	452	0	3	48	45	45	452	449	449
Diagnosa_user.frm	Form	253	852	8	26	245	227	219	844	826	818
About.frm	Form	137	505	16	23	121	114	98	489	482	466
Module1.bas	Module	33	34	4	4	29	29	25	30	30	26

Gambar 69. Perhitungan Jumlah LOC dengan *VB6 Pure Code Lines Calculator*

Selain validitas, karakter lain dari evaluasi adalah reliabilitas. Penguji juga melakukan uji reliabilitas instrumen *Code Line Counter*. Reliabilitas dapat juga diartikan dengan konsistensi atau keajegan. Bila terdapat kesamaan data dalam waktu yang berbeda, hasil penelitian dapat dikatakan reliabel. Oleh karena itu, uji reliabilitas dapat dilakukan dengan melakukan pembagian waktu pengujian. Daftar *file* yang dihitung jumlah LOC–nya berjumlah 11. 11 *file* tersebut dibagi menjadi 4 bagian untuk dihitung masing-masing LOC-nya,kemudian dijumlahkan hasilnya. Penjabaran hasil uji reliabilitas dapat dilihat pada tabel 59.

Tabel 59. Penjabaran Hasil Uji Reliabilitas

No.	Uji Reliabilitas	Hasil (LOC)
1.	Tahap I (3 form : About.frm, Diagnosa_admin.frm, Diagnosa_user.frm)	2698
2.	Tahap II (3 form: Kotak_tambah.frm, Login.frm, Signup.frm)	1041
3.	Tahap III (3 form: Sisipkan.frm, Tambah.frm, Tambah_awal.frm)	1495
4.	Tahap IV (2 form: Ubah.frm, Module1.bas)	440
Jumlah		**5674**

Hasil uji reliabilitas di atas menunjukkan bahwa jumlah *lines of code dari source code*perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi adalah 5674 LOC = 5,674 KLOC. Berdasarkan data tersebut, dapat disimpulkan bahwa instrumen penelitian ini untuk menghitung jumlah *lines of code*(LOC)yaitu *code line counter*reliabel karena hasil pengujian yang telah dilakukan menghasilkan data yang sama. Instrumen penelitian ini untuk menghitung jumlah *lines of code*(LOC)yaitu *code line counter* valid karena hasil pengujian menggunakan *tool* lain hasilnya sama.

b. Perhitungan Jumlah *Error*

Perhitungan jumlah *error* pada *source code*Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi dilakukan menggunakan *Code Advisor for Visual Basic 6*. *Code Advisor for Visual Basic 6* merupakan *freeware tools* yang dikembangkan oleh Microsoft. *Code Advisor for Visual Basic 6* merupakan aplikasi *plugs-in*gratis dari Microsoft untuk aplikasi Visual Basic 6.0. *Code Advisor for Visual Basic 6*digunakan untuk menganalisis kode perangkat lunak yang dikembangkan dan menyarankan perbaikan.

Berikut adalah *screenshoot* penghitungan jumlah *error*Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi menggunakan *Code Advisor for Visual Basic 6*:

Code Advisor for Visual Basic 6

Project:	Project1
Project Path:	D:\SKRIPSI\Aplikasi\Project1.vbp
Scope:	Active Project
Rule Count:	5
Component Count:	11
Issue Count:	0
Report Date:	11/24/2013
Processing Time:	00:00:18

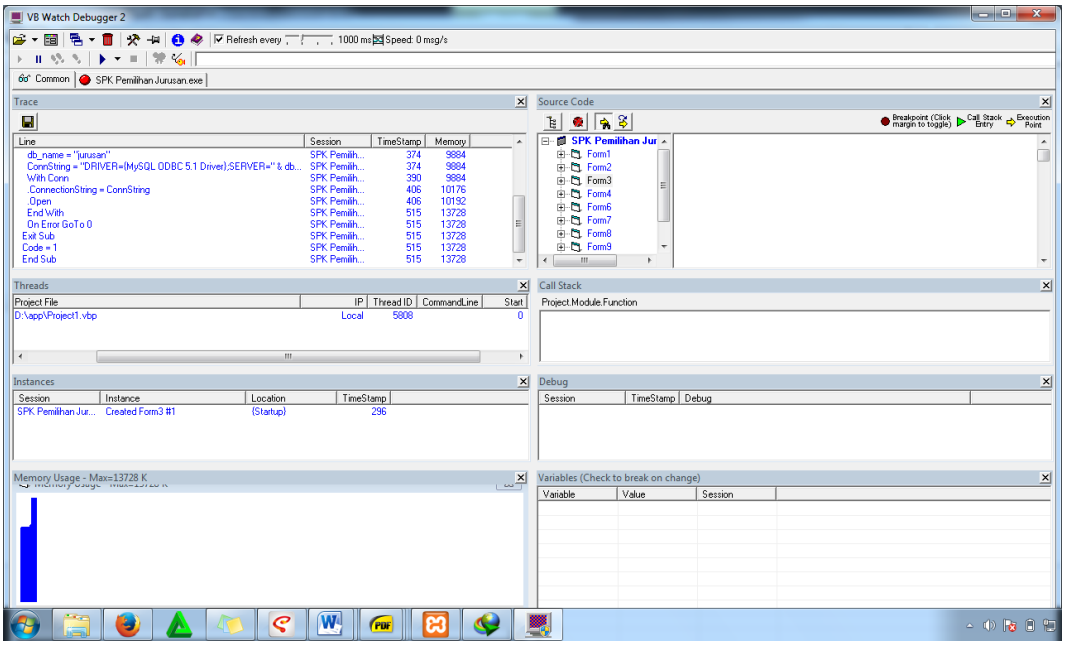
Issues Total Count:0

Rules Total Count:5

Name	Description	Issue Count
Variant-Returning String Function	Variant-returning string functions are not supported in Visual Basic .NET. Use the String-returning version of the function, which has a '$' suffix.	0
Incorrect Use Of Enumeration	Incorrectly used enumeration constants will not up upgraded.	0
DAO Data Binding	DAO data binding is deprecated in Visual Basic 6. Consider using ADO data binding instead.	0
RDO Data Binding	Replace RDO data binding with ADO	0
Single Threaded Controls	Controls designed for Visual Basic 6 have better performance , since they are apartment threaded.	0

Gambar 70. Perhitungan Jumlah *Error*

Gambar 70 menunjukan bahwa jumlah *error* untuk pengujian fakor kualitas *correctness* menggunakan *Code Advisor for Visual Basic 6* adalah 0. Seperti halnya instrumen penguji jumlah *lines of code*, instrumen penguji jumlah *error* juga harus dilakukan uji validitas dan reliabilitas. Untuk menunjukkan tingkat ketepatan alat ukur, maka *Code Advisor for Visual Basic 6* dibandingkan dengan *VB Watch Debugger*. Dapat dikatakan valid apabila hasil yang diperoleh dari masing-masing alat ukur adalah sama. Berikut ini merupakan hasil perhitungan jumlah *error* perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggimenggunakan*VB Watch Debugger* :



Gambar 71. Perhitungan Jumlah *Error* dengan *VB Watch Debugger*

Berdasarkan hasil uji validitas tersebut maka dapat dijabarkan perbandingan hasil perhitungan jumlah *error* perangkat lunak menggunakan *Code Advisor for Visual Basic 6* dan *VB Watch Debugger* sebagai berikut :

Tabel 60. Penjabaran Hasil Uji Validitas Jumlah *Error*

Instrumen	Hasil Perhitungan	Keterangan
Code Advisor for Visual Basic 6	0	Valid
VB Watch Debugger	0	

Selain validitas, instrumen juga dilakukan uji reliabilitas. Instrumen yang reliabel adalah instrumen yang digunakan beberapa kali untuk mengukur obyek yang sama, akan menghasilkan data yang sama. Berikut ini tabel hasil pengujian reliabilitas *Code Advisor for Visual Basic 6*:

Tabel 61. Hasil Uji Reliabilitas *Code Advisor for Visual Basic 6*

No.	Uji Reliabilitas	Hasil
1.	Tahap I (3 form : About.frm, Diagnosa_admin.frm, Diagnosa_user.frm)	0
2.	Tahap II (3 form: Kotak_tambah.frm, Login.frm, Signup.frm)	0
3.	Tahap III (3 form: Sisipkan.frm, Tambah.frm, Tambah_awal.frm)	0
4.	Tahap IV (2 form: Ubah.frm, Module1.bas)	0
Jumlah		**0**

Hasil uji reliabilitas *Code Advisor for Visual Basic 6*pada tabel 61 menunjukkan bahwa jumlah *error* dari *source code*perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi adalah 0. Berdasarkan data tersebut, dapat disimpulkan bahwa instrumen penelitian ini reliabel karena hasil pengujian yang telah dilakukan menghasilkan data yang sama.

c. Perbandingan Hasil Pengujian dengan S*tandard* yang Telah Ditentukan

Dari hasil pengujian sebelumnya, didapatkan bahwa nilai *Error/Kilo Line Of Code*perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi adalah 0 *Error/KLOC*. Aplikasi akan dikatakan lolos pengujian jika jumlah *error*≤ (kurang dari sama dengan) standar yang digunakan. Sebaliknya, aplikasi akan dikatatakan gagal jika jumlah *error* melebihi standar yang digunakan.

Tabel 62. Perbandingan Hasil Pengujian Faktor Kualitas *Correctnes*dengan Standar yang Digunakan

Nama Standar	**Nilai Standar (*error/KLOC*)**	**Hasil Pengujian Aplikasi**	**Keterangan**
Industry Average	1-25	$\frac{0}{5,674} = 0$	**LOLOS**. Jumlah *error* lebih sedikit dari standar. Lebih baik.
Microsoft Application	0,5		**LOLOS**. Jumlah *error* lebih sedikit dari standar. Lebih baik.

Tabel perbandingan hasil pengujian dengan standar yang digunakan pada tabel 62 menujukan bahwa aplikasi Sistem Pendukung Keputusan Pemilihan

Jurusan di Perguruan Tinggi memenuhi standar faktor kualitas *correctness* baik dari *Standar Industry Average* maupun *Microsoft Application*.

2. Analisis Faktor Kualitas *Functionality*

Faktor kualitas *functionality* diuji dengan melakukan tes pada setiap fungsi yang terdapat pada aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi. Hasil pengujian kemudian dibandingkan dengan standar *functionality* yang ditetapkan oleh Microsoft dalam program *Microsoft Certification Logo.*

Pengujian faktor kualitas *functionality*dilakukan oleh pengembang dengan mengunakan format *test case* seperti yang telah dijelaskan sebelumnya. Uji faktor kualitas*functionality* yang dilakukan dengan menggunakan *test case* dibagi menjadi 2 jenis *test case*, yaitu *test case* untuk fungsi primer dan *test case* untuk fungsi pendukung. *Test case* fungsi primer dibuat untuk menguji fungsi utama sistem yang terhubung dengan *database*. Sedangkan *test case* fungsi pendukung dibuat untuk menguji fungsi pendukung yang tidak ada kaitannya dengan *database*.

Berikut ini adalah tabel (Tabel 63 dan Tabel 64) rangkuman *test case* pengujian faktor kualitas *functionality*pada aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggiuntukfungsi primer dan fungsi pendukung (Hasil *test case* lengkap dilampirkan) :

Tabel 63. Rangkuman *Test Case* Fungsi Primer

No	Nama Tes	Lolos / Gagal
1	*LoginTest*1	Lolos
2	*LoginTest*2	Lolos
3	*SignupTest*1	Lolos
4	*SignupTest*2	Lolos
5	*SignupTest*3	Lolos
6	*SignupTest*4	Lolos
7	Diagnosa*UserTest*	Lolos
8	Diagnosa*AdminTest*	Lolos
9	*PrimaryFunctionality*1	Lolos
10	*PrimaryFunctionality*2	Lolos
11	*PrimaryFunctionality*3	Lolos
12	*PrimaryFunctionality*4	Lolos
13	*PrimaryFunctionality*5	Lolos
14	*PrimaryFunctionality*6	Lolos
15	*PrimaryFunctionalityAdmin*1	Lolos
16	*PrimaryFunctionalityAdmin*2	Lolos
17	*PrimaryFunctionalityAdmin*3	Lolos
18	*PrimaryFunctionalityAdmin*4	Lolos
19	*PrimaryFunctionalityAdmin*5	Lolos
20	*PrimaryFunctionalityAdmin*6	Lolos
21	*PrimaryFunctionality*7	Lolos
22	*PrimaryFunctionalityAdmin*7	Lolos
23	*PrimaryFunctionalityAdmin*8	Lolos

Tabel 64. Rangkuman *Test Case* Fungsi Pendukung

No	Nama Tes	Lolos / Gagal
1	*AboutTest*	Lolos
2	*HelpLoginTest*	Lolos
3	*HelpSignupTest*	Lolos
4	*Help*Diagnosa*UserTest*	Lolos
5	*Help*Diagnosa*AdminTest*1	Lolos
6	*Help*Diagnosa*AdminTest*2	Lolos
7	*Help*Diagnosa*AdminTest*3	Lolos
8	*Help*UbahFakta*Test*	Lolos
9	*Help*SisipkanFakta*Test*	Lolos
10	*Help*TambahAwal*Test*	Lolos
11	*HelpAboutTest*	Lolos

Hasil *test case*fungsi primer dan fungsi pendukung kemudian dibandingkan dengan standar *functionality* dalam program *Microsoft Certification Logo*untuk mengetahui apakah aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi lolos uji faktor kualitas *functionality*. Perbandingan tersebutdapat dilihat pada tabel 65.

Berdasarkan perbandingan hasil pengujian faktor kualitas *functionality* dengan standar kriteria faktor kualitas *functionality* dalam *Microsoft Certification Logo,* maka dapat disimpulkan bahwa aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi LOLOS pengujian faktor kualitas *functionality.*

Tabel 65. Perbandingan Hasil Pengujian Faktor Kualitas *Functionality*

Kriteria Lolos	Kriteria Gagal	Hasil Pengujian	Keterangan
1. Setiap fungsi primer yang diuji berjalan sebagai-mana mestinya. 2. Jika ada fungsi pendukung yang tidak berjalan se-bagaimana mes-tinya, tetapi itu bukan kesalahan yang serius dan tidak berpengaruh pada penggunaan normal.	1. Paling tidak ada satu fungsi primer yang diuji tidak berjalansebagai-mana mestinya. 2. Jika ada fungsi pendukung yang tidak berjalan se-bagaimana mes-tinya dan itu merupakan kesa-lahan yang serius dan berpengaruh pada penggunaan normal.	Semua fungsi primer dan fungsi pendukung berjalan dengan baik.	**LOLOS**

3. Analisis Faktor Kualitas *Usability*

Pengujian faktor kualitas *usability*pada penelitian ini dilakukan dengan menggunakan metode kuesioner. Kuesioner yang digunakan mengacu pada *Computer System Usability Questionnaire (CSUQ)* yang dikembangkan oleh J.R. Lewis. Kuesioner tersebut diterjemahkan dengan bahasa yang disesuaikan dengan keadaan perangkat lunak yang dikembangkan. Kuesioner tersebut diberikan kepada 60 responden yang terdiri dari 45 siswa kelas XII SMA N 2 Wates dan 15 mahasiswa di Yogyakarta. Berikut adalah tabel jawaban responden terhadap setiap pertanyaan dalam kuesioner.

Tabel 66. Jawaban Responden Terhadap Pertanyaan Kuesioner *Usability*

Pertanyaan	Jawaban Responden				
	Sangat Setuju	**Setuju**	**Ragu-ragu**	**Tidak Setuju**	**Sangat Tidak Setuju**
Pertanyaan 1	9	27	18	6	0
Pertanyaan 2	34	26	0	0	0
Pertanyaan 3	4	18	26	11	1
Pertanyaan 4	3	17	24	16	0
Pertanyaan 5	4	18	27	11	0
Pertanyaan 6	18	26	9	7	0
Pertanyaan 7	29	27	4	0	0
Pertanyaan 8	0	21	29	9	1
Pertanyaan 9	12	28	13	7	0
Pertanyaan 10	17	33	9	1	0
Pertanyaan 11	14	33	11	2	0
Pertanyaan 12	13	32	13	2	0
Pertanyaan 13	16	34	10	0	0
Pertanyaan 14	5	23	25	7	0
Pertanyaan 15	20	36	4	0	0
Pertanyaan 16	15	42	2	1	0
Pertanyaan 17	11	37	10	2	0
Pertanyaan 18	1	29	25	5	0
Pertanyaan 19	6	30	16	8	0
Jumlah	**231**	**537**	**275**	**95**	**2**

Data yang dihasilkan dari kuisioner merupakan data yang bersifat kualitatif. Data tersebut dapat dikonversi ke dalam data kuantitatif dalam bentuk data interval atau rasio menggunakan Skala Likert. Konversi jawaban item kuesioner menjadi nilai kuantitatif dapat dilihat pada tabel67 dibawah ini:

Tabel 67.Konversi Jawaban Item Kuesioner ke Nilai Kuantitatif

Jawaban	Skor
Sangat setuju	5
Setuju	4
Ragu – ragu	3
Tidak setuju	2
Sangat tidak setuju	1

Berikut perhitungan jumlah skor yang didapat dari hasil kuesioner :

- Jumlah jawaban “Sangat Setuju” = 231 x 5 = 1155
- Jumlah jawaban “Setuju” = 537 x 4 = 2148
- Jumlah jawaban “Ragu - Ragu” = 275 x 3 = 825
- Jumlah jawaban “Tidak Setuju” = 95 x 2 = 190
- Jumlah jawaban “Sangat Tidak Setuju” = 2 x 1 = 2

Jumlah Total = 4320

Skor yang didapatkan dari hasil kuesioner tersebut kemudian dihitung nilai rata-rata totalnya. Dari data yangtelah dikumpulkan, dapat dihitung rata-ratanya sebagai berikut:

$$\bar{X} = \frac{\sum X}{N}$$

$$= \frac{4320}{60}$$

$$= \mathbf{72}$$

Rata-rata total yang didapat kemudian dibandingkan dengan kategori penilaian faktor kualitas *usability*. Kategori penilaian faktor kualitas *usability*dapat dilihat pada tabel 68.

Tabel 68.Kategori Penilaian Faktor Kualitas *Usability*

Rentang skor kualitatif	Nilai
$\bar{X}$>79,8	Sangat Layak
64,6<$\bar{X}$ ≤ 79,8	Layak
49,4<$\bar{X}$ ≤ 64,6	Cukup Layak
34,2<$\bar{X}$ ≤ 49,4	Tidak Layak
$\bar{X}$ ≤ 34,2	Sangat Tidak Layak

Dari data di atas, hasil analisis faktor kualitas *usability* menunjukan bahwa Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi masuk dalam kriteria **"Layak"**.

4. Analisis Faktor Kualitas *Maintainability*

Pengujian aspek *maintainability* dilaksanakan peneliti dengan cara diuji secara operasional. Berikut ini tabel hasil uji faktor kualitas*maintainability* :

Tabel 69. Hasil Uji *Maintainability*

Aspek	Kriteria Lolos	Hasil Uji
Instrumentation	Ketika ada kesalahan yang dilakukan oleh *user*, maka sistem akan mengeluarkan peringatan untuk mengidentifikasi kesalahan.	Hasil pengujian yang telah dilakukan peneliti menunjukkan bahwa ketika ada kesalahan yang dilakukan oleh *user*, maka sistem akan mengeluarkan pesan peringatan untuk mengidentifikasi kesalahan. Contoh: ketika user melakukan pendaftaran dan masih ada data yang kosong atau data lebih dari 15 karakter maka akan muncul peringatan mengenai kesalahan tersebut.

Tabel 70. Hasil Uji *Maintainability*(lanjutan)

Consistency	Bentuk rancangan sistem pengolah data mempunyai satu bentuk yang sama. Hal ini dapat dilihat pada bagian imple-mentasi sistem.	Hasil pengujian yang telah dilakukan peneliti menunjukkan bahwa bentuk rancangan sistem mempunyai satu bentuk yang sama. Hal ini dapat dilihat pada bagian implementasi sistem, di mana tampilan halaman aplikasi dari satu halaman ke halaman lainnya memiliki kemiripan, bentuk yang serupa dan konsisten.
Simplicity	Mudah untuk dike-lola, diperbaiki, dan dikembangkan. Hal ini dapat dilihat pada tahapan-tahapan proses penulisan kode program.	Sistem ini mudah dikelola karena jika ingin mengembangkan fungsi-fungsi yang belum ada, pengembang hanya perlu membuat tombolatau fungsi baru tanpa mengubah komponen yang lain.
		Sistem ini mudah diperbaiki karenatelah mengikuti pedoman pengkodean, seperti penggunaan konvensi penamaan yang standar sehingga pengembang selanjutnya dapat membedakan antara variabel, konstanta dan dapat lebih mudah mendeteksi *error*. *Source code* telah dilengkapi dengan *comment lines* sehingga mudah dipahami logikanya.
		Hasil pengujian menunjukkan bahwa sistem mudah untuk dikembangkan, karena dibuat menggunakan visual basic 6.0 yang menggunakan bahasa pemograman BASIC (*Begin-ners All-purpose Symbolic Instruction Code*) yang cukup mudah dipelajari, populer dan digunakan oleh banyak orang.

Berdasarkan tabel di atas, diperoleh data bahwa hasil uji perangkat lunak Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi telah sesuai dengan kriteria lolos faktor kualitas *maintainability*.

C. Implementasi Pengujian

Setelah perangkat lunak mendapatkan pengujian *beta*oleh 60 responden menggunakan kuesioner yang mengacu pada *Computer System Usability Questionnaire (CSUQ)* yang dikembangkan oleh J.R. Lewis, data penelitian yang didapat tidak hanya berupa jawaban terhadap pertanyaan kuesioner. Penulis juga mendapatkan data kecocokan minat jurusan responden dengan jurusan hasil keluaran program yang dicoba oleh masing-masing responden. Data tersebut nantinya akan digunakan untuk menguji apakah *knowledge base* yang terdapat pada perangkat lunakSistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi layak digunakan.

Pada lembar penilaian yang diberikan kepada 60 responden disediakan tempat untuk menuliskan 3 jurusan yang menjadi minat responden. Setelah responden menggunakan aplikasiSistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi, responden diminta menuliskan jurusan hasil keluaran program di halaman paling bawah. Berdasarkan data tersebut penulis merangkum kecocokan minat jurusan. Berikut adalah data 60 responden terhadap kecocokan minat jurusan responden dengan jurusan hasil keluaran aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi :

Tabel 71. Kecocokan Jawaban Responden Terhadap Minat Jurusan dengan Keluaran Program

Responden	Kecocokan Minat Jurusan Responden dengan Program		
	Minat Jurusan 1	Minat Jurusan 2	Minat Jurusan 3
Responden 1			Sesuai
Responden 2			
Responden 3	Sesuai	Sesuai	
Responden 4	Sesuai		
Responden 5	Sesuai		
Responden 6	Sesuai		
Responden 7	Sesuai		
Responden 8	Sesuai		
Responden 9	Sesuai		Sesuai
Responden 10	Sesuai	Sesuai	
Responden 11	Sesuai		
Responden 12	Sesuai		
Responden 13	Sesuai		
Responden 14			
Responden 15			
Responden 16			
Responden 17	Sesuai		
Responden 18		Sesuai	Sesuai
Responden 19	Sesuai		
Responden 20	Sesuai		
Responden 21	Sesuai		
Responden 22	Sesuai		Sesuai
Responden 23	Sesuai		
Responden 24	Sesuai		Sesuai
Responden 25	Sesuai		
Responden 26	Sesuai		
Responden 27	Sesuai		
Responden 28	Sesuai		
Responden 29	Sesuai		
Responden 30	Sesuai		
Responden 31	Sesuai		
Responden 32	Sesuai	Sesuai	Sesuai
Responden 33	Sesuai	Sesuai	
Responden 34			
Responden 35			
Responden 36		Sesuai	
Responden 37			

Tabel 72.Kecocokan Jawaban Responden Terhadap Minat Jurusan dengan Keluaran Program (lanjutan)

Responden 38			
Responden 39			
Responden 40			
Responden 41	Sesuai		
Responden 42	Sesuai		
Responden 43			
Responden 44	Sesuai		
Responden 45	Sesuai	Sesuai	Sesuai
Responden 46		Sesuai	
Responden 47	Sesuai		
Responden 48		Sesuai	Sesuai
Responden 49			Sesuai
Responden 50		Sesuai	
Responden 51	Sesuai		
Responden 52			Sesuai
Responden 53	Sesuai		
Responden 54		Sesuai	Sesuai
Responden 55		Sesuai	
Responden 56		Sesuai	Sesuai
Responden 57		Sesuai	
Responden 58	Sesuai		
Responden 59		Sesuai	Sesuai
Responden 60		Sesuai	Sesuai
Jumlah Responden Cocok			**46**

Berdasarkan data pada tabel diatas dapat dihitung jumlah jurusan responden yang cocok dengan hasil keluaran program. Dengan asumsi minat jurusan 3 adalah jurusan cadangan saja maka untuk menjaga validitas perhitungan hanya minat jurusan 1 dan 2 saja yang dihitung. Apabila terdapat salah satu atau keduanya diantara minat jurusan 1 dan 2 yang sesuai dengan hasil keluaran program maka minat jurusan responden tersebut dinyatakan cocok dengan jurusan hasil keluaran program.

Untuk melakukan penghitungan deskriptif persentase dari data hasil pengujian kecocokan minat jurusan responden dengan hasil keluaran aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggidapat dihitung sebagai berikut :

1. Rumus perhitungan persentase skor ditulis dengan rumus berikut :

$$Persentase\ Kelayakan = \frac{Skor\ yang\ diobservasi}{Skor\ yang\ diharapkan} x\ 100\%$$

2. Hasil perhitungan yang didapatkan :

$$Persentase\ Kelayakan\ (\%) = \frac{46}{60} x\ 100\%$$

$$Persentase\ Kelayakan\ = 76.67\%$$

Setelah didapatkan hasil persentase dari perhitungan diatas, kemudian persentase tersebut dikonversi ke dalam pernyataan predikat. Untuk melakukan konversi dari hasil perhitungan data ke dalam pernyataan predikat digunakan skala linkert. Konversi persentase ke pernyataan predikat dapat dilihat pada tabel dibawah ini (Riduwan, 2011):

Tabel 73. Interpretasi Persentase Likert

No	Presentase	Interpretasi
1	0% - 20%	Sangat Lemah
2	21% - 40%	Lemah
3	41% - 60%	Cukup
4	61% - 80%	Kuat
5	81% - 100%	Sangat Kuat

Supaya konversi persentase ke dalam bentuk pernyataan predikat lebih sesuai dengan penelitian yang dilakukan oleh penulis, maka skala konversi persentase diatas disesuaikan interpretasinya. Penyesuaian interpretasi tersebut dikarenakan penelitian ini melakukan uji kelayakan perangkat lunak yang dikembangkan oleh penulis. Penyesuaian interpretasi pada skala konversi persentase ke dalam bentuk pernyataan predikat dapat dilihat pada tabel 74.

Tabel 74. Penyesuaian Interpretasi Persentase Likert

No	Presentase	Interpretasi
1	81% - 100%	SangatLayak
2	61% - 80%	Layak
3	41% - 60%	Cukup Layak
4	21% - 40%	Tidak Layak
5	0% - 20%	Sangat Tidak Layak

Dari data di atas, hasil implementasi pengujian kecocokan minat jurusan dengan hasil keluaran program menunjukan bahwa Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi masuk dalam kriteria **“Layak”**.

BAB V
KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pembahasan yang telah diuraikan sebelumnya dapat diambil beberapa kesimpulan sebagai berikut :

1. Hasil dari pengembangan perangkat lunak adalah Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi yang dikembangkan menggunakan Visual Basic 6.0 dan *database* MySQL. *Knowledge base*dibangun berdasarkanteori kepribadian RIASEC dari John L. Holland.

2. Hasil unjuk kerja Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi memenuhi standar faktor kualitas *correctness*, *functionality*, *usability*,dan *maintainability*. Nilai *error/KLOC* aplikasi ini adalah 0 sehingga memenuhi standar faktor kualitas *correctness*.Seluruh fungsi primer dan fungsi pendukung aplikasi ini berjalan dengan sebagaimana mestinya sehingga memenuhi standar faktor kualitas *functionality*.Hasil analisis faktor kualitas *usability* menunjukan bahwa aplikasi inimasuk dalam kriteria “Layak”. Hasil pengujian tiga aspek*instrumentation*, *consistency* dan *simplicity* menunjukkan bahwa aplikasi ini telah sesuai dengan kriteria lolos faktor kualitas *maintainability*.

3. Hasil analisis implementasi pengujian menunjukan bahwa Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi masuk dalam kriteria “Layak”.

B. Keterbatasan Produk

Penelitian yang telah dilakukan menghasilkan suatu produk yang masih memiliki beberapa keterbatasan. Keterbatasan penelitian itu adalah sebagai berikut :

1. Aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi hanya menggunakan *connector ODBC*untukmenghubungkansistem dengan *database.*
2. Aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi hanya berdasar pada tipe kepribadian RIASEC karangan psikolog John L. Holland untuk pencarian jurusan yang sesuai.
3. Aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi hanya menyediakan 61 Universitas Negeri di Indonesia sebagai universitas penyedia jurusan keluaran program.

C. Saran

1. Dalam pembuatan koneksi Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi dengan jaringan, penulis hanya menggunakan *connector ODBC*. Penelitian selanjutnya dapat mengembangkan sistem tambahan untuk mengatur *connectivity* terhadap jaringan.
2. Dalam penelitian ini penulis hanya melakukan pengujian pada faktor kualitas *correctness*, *functionality*, *usability*, dan *maintainability*. Penelitian selanjutnya dapat dilakukan pengujian pada faktor kualitas yang lain.

DAFTAR PUSTAKA

Agarwal, B. B., Tayal, S. P., & Gupta, M. (2010). *Software Engineering and Testing.* Sudbury: Jones and Bartlett Publishers.

Anonim. (2010). *Holland Three Digit Code.* Dipetik Juli 15, 2013, dari O*NET OnLine: http://www.onetonline.org/

Anonim. (2012). *Code Advisor for Visual Basic 6*. Dipetik September 13, 2012, dari Microsoft: http://www.microsoft.com/en-us/download/details.aspx?id=1222#Overview

Bach, J. (2005). *General Functionality and Stability Test Procedure for Certified for Microsoft Windows Logo.* Dipetik Maret 4, 2013, dari Satisfice,Inc: http://www.satisfice.com/tools/procedure.pdf

Boehm, B. W. (1981). *Software Engineering Economics.* Prentice-Hall.

Borg, Walter R, Gall, M, & D. (1996). *Educational research: An introduction (6th ed).* England: Longman Publishing.

Cannolly, T. M., & Carolyn, B. E. (2005). *Database Systems: A Practical Approach to Design, Implementationand Management. Fourth Edition.* Addison-Wesley.

Giarratano, J., & Riley, G. (2005). *Expert System Principles and Programming.* Boston: PWS-KENT Publishing Company.

Hartati, S., & Iswanti, S. (2008). *Sistem Pakar & Pengembangannya.* Yogyakarta: Graha Ilmu.

Hass, A. M. (2008). *Guide to Advanced Software Testing.* Norwood: Artech House.

Hetzel, B. (1988). *The Complete Guide to Software Testing, Second Edition.* New York: John Wiley.

Irena, J. (2009). *Software Testing Methods and Techniques.* Dipetik Juli 22, 2013, dari www.internetjournals.net/journals/tir/2009/January/Paper%2006.pdf

Laela, Z. (2013). *Tingkat kelulusan UN SMA 2013 turun 0,02 persen*. Dipetik Mei 25, 2013, dari SINDOnews.com: http://nasional.sindonews.com/read/2013/05/24/15/730411/tingkat-kelulusan-un-sma-2013-turun-0-02-persen

Manrihu, M. T. (1992). *Pengantar Bimbingan dan Konseling Karier.* Jakarta: Bumi Aksara.

Martin, J., & Oxman, S. (1988). *Building Expert Systems a Tutorial.* New Jersey: Prentice Hall.

McConnell, S. C. (2004). *Code Complete.* Redmond: Microsoft Press.

Myers, G. J. (1979). *The Art of Software Testing.* New York: John Wiley.

Osipow, S. H. (1983). *Theories of Career Development.* New Jersey: Prentice Hall.

Pressman, R. (2001). *Software Engineering : A Practitioner's Approach* (5 ed.). New York: McGraw Hill.

Riduwan. (2011). *Belajar Mudah Penelitian untuk Guru-Karyawan dan Peneliti Pemula.* Bandung: Alfa Beta.

Rosa, S. (2011). *Rekayasa Perangkat Lunak.* Bandung: Modula.

Sugiyono. (2011). *Metode Penelitian Pendidikan.* Bandung: Alfabeta.

Sujadi. (2003). *Metodologi Penelitian Pendidikan.* Jakarta: Rineka Cipta.

Sukarjo. (2006). *Kumpulan Materi Evaluasi Pembelajaran.* Yogyakarta: Jurusan Teknologi Pembelajaran Program Pascasarjana Universitas Negeri Yogyakarta.

Sutojo, T., Mulyanto, E., & Suhartono, V. (2011). *Kecerdasan Buatan.* Yogyakarta: Andi.

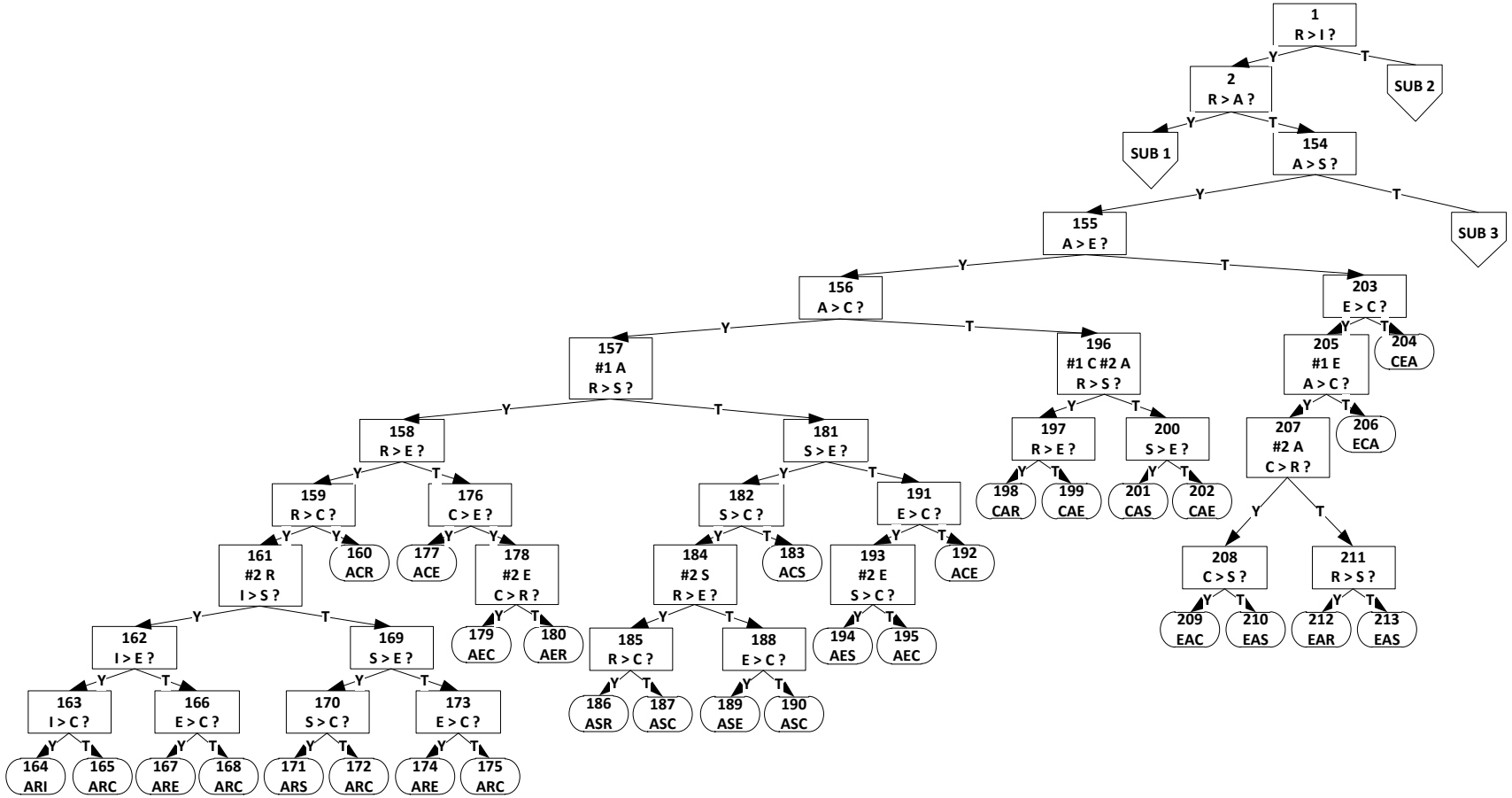
Turban. (2005). *Decision Support Systems and Intelligent Systems.* Yogyakarta: Penerbit Andi.

Waljiyanto. (2003). *Sistem Basis Data: Analisis dan Pemodelan Data.* Yogyakarta: Graha Ilmu.

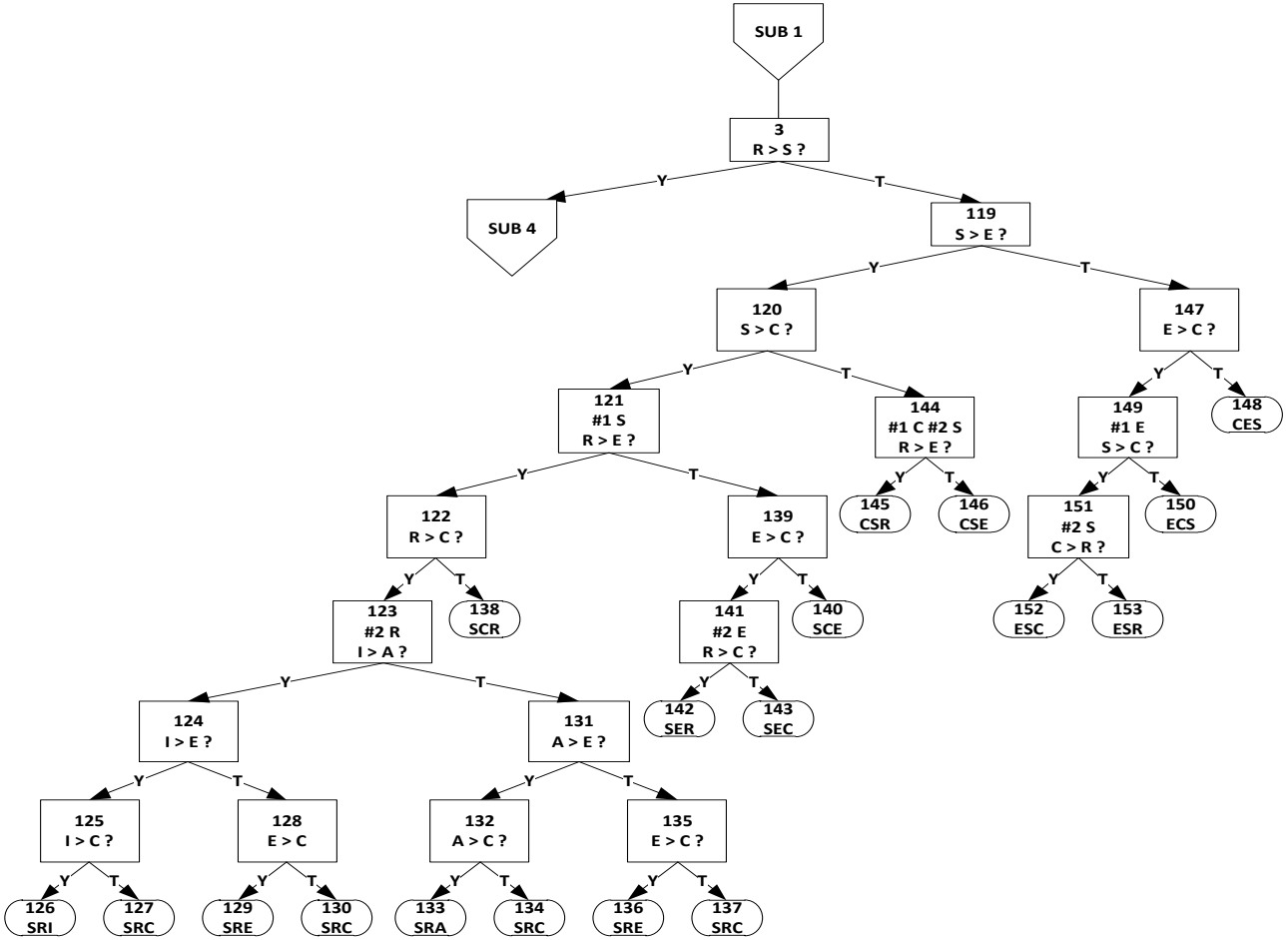
Winkel, W. S., & Hastuti, S. (2005). *Bimbingan dan Konseling di Institusi Pendidikan.* Jakarta: PT. Grasindo.

LAMPIRAN

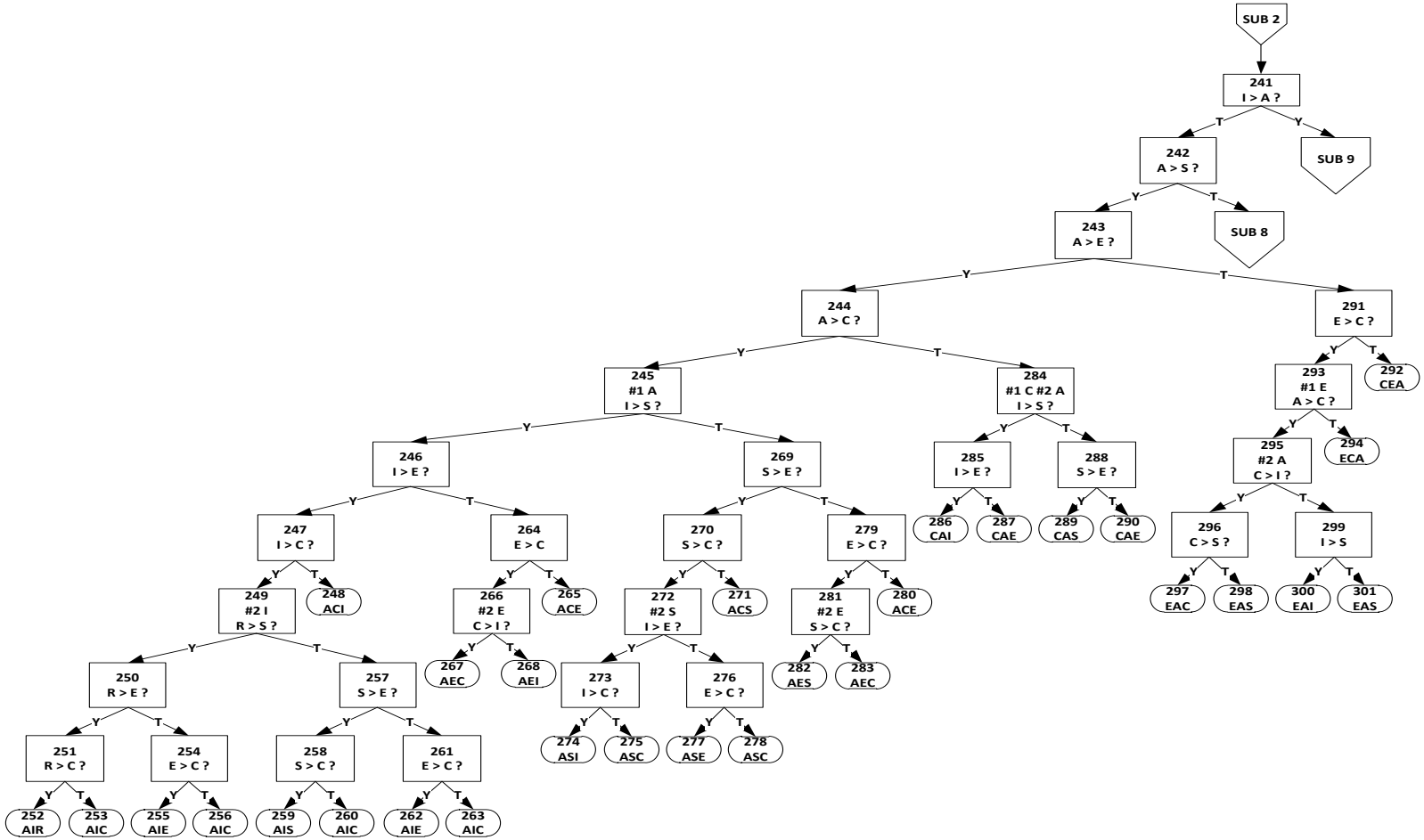
Lampiran 1.*Knowledge Base*



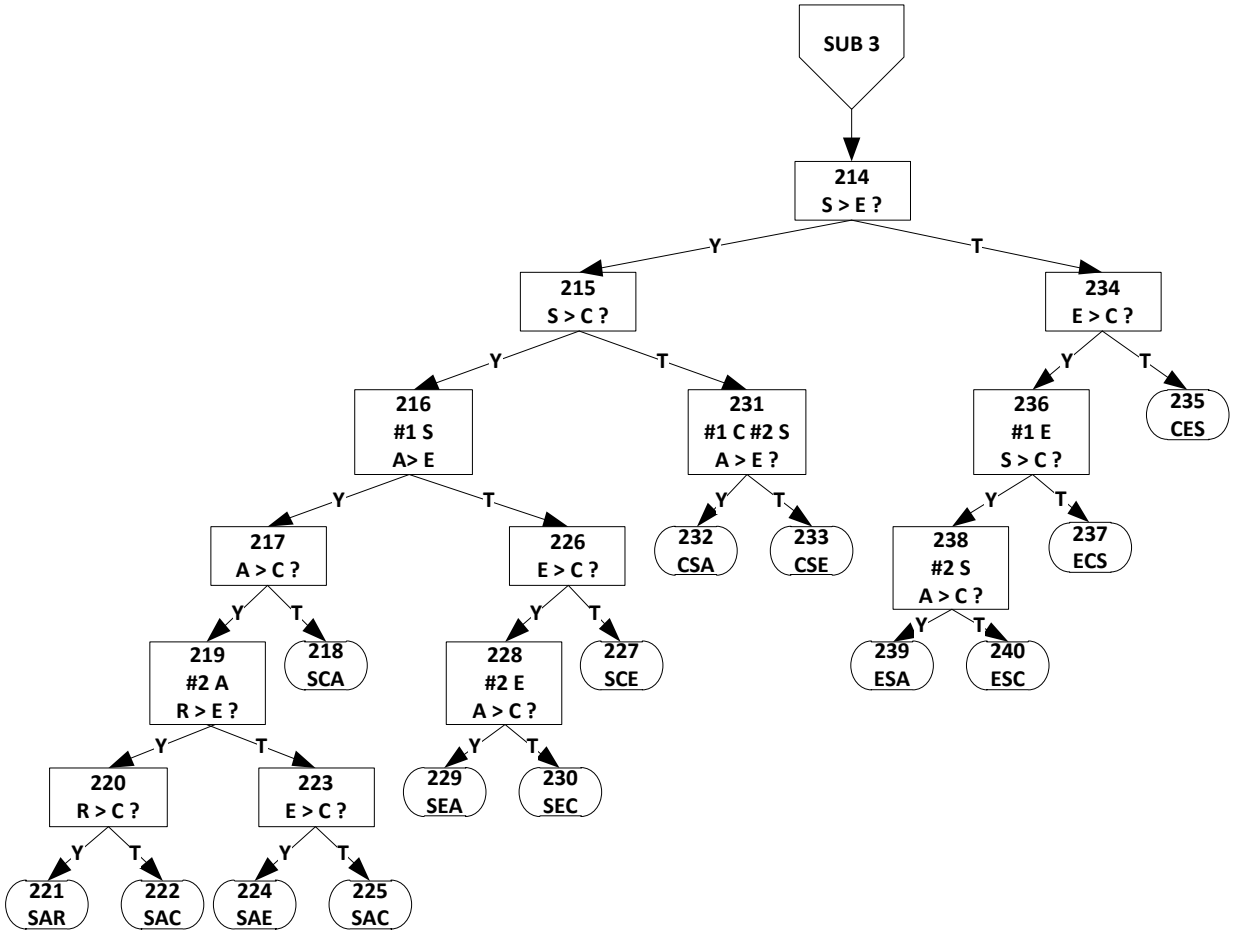
Gambar 1. *Rulebase* Sistem



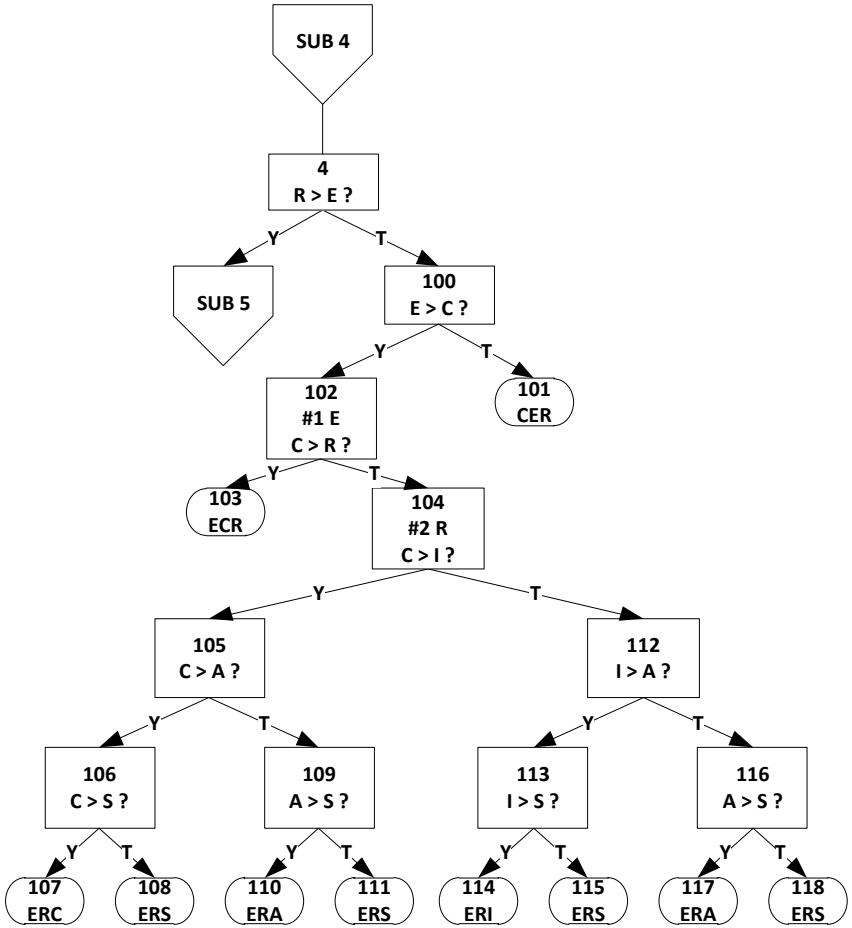
Gambar 2. *Rulebase* Sub 1



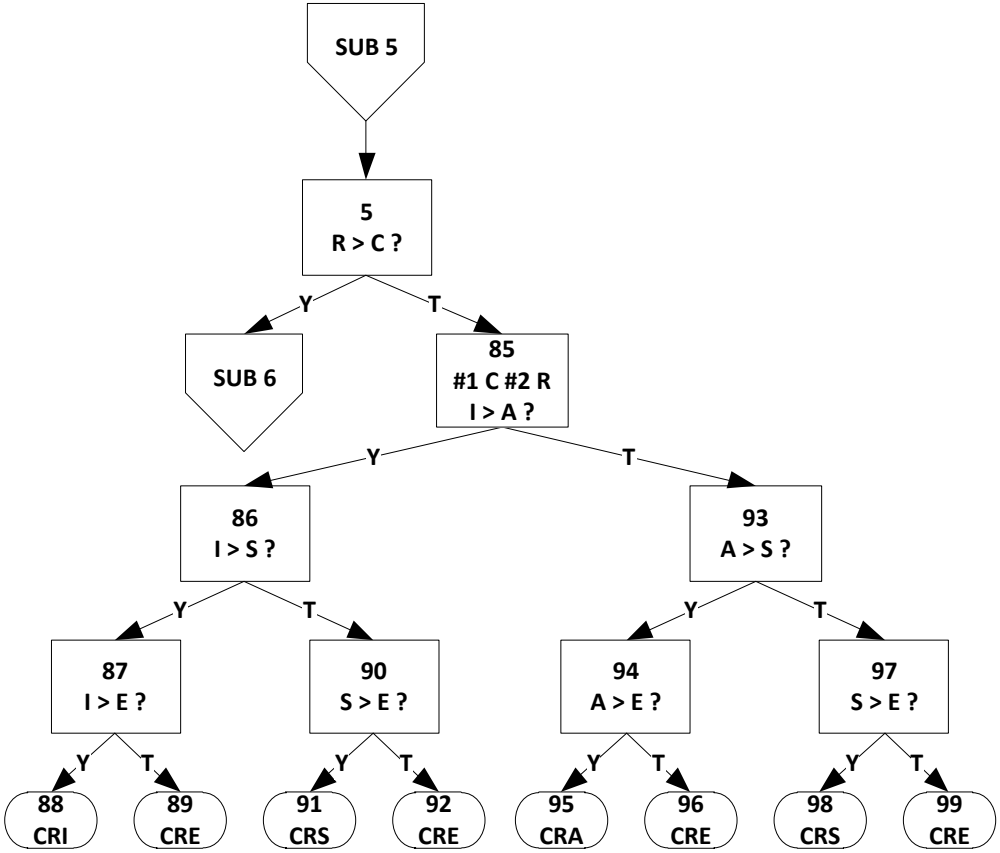
Gambar 3. *Rulebase* Sub 2



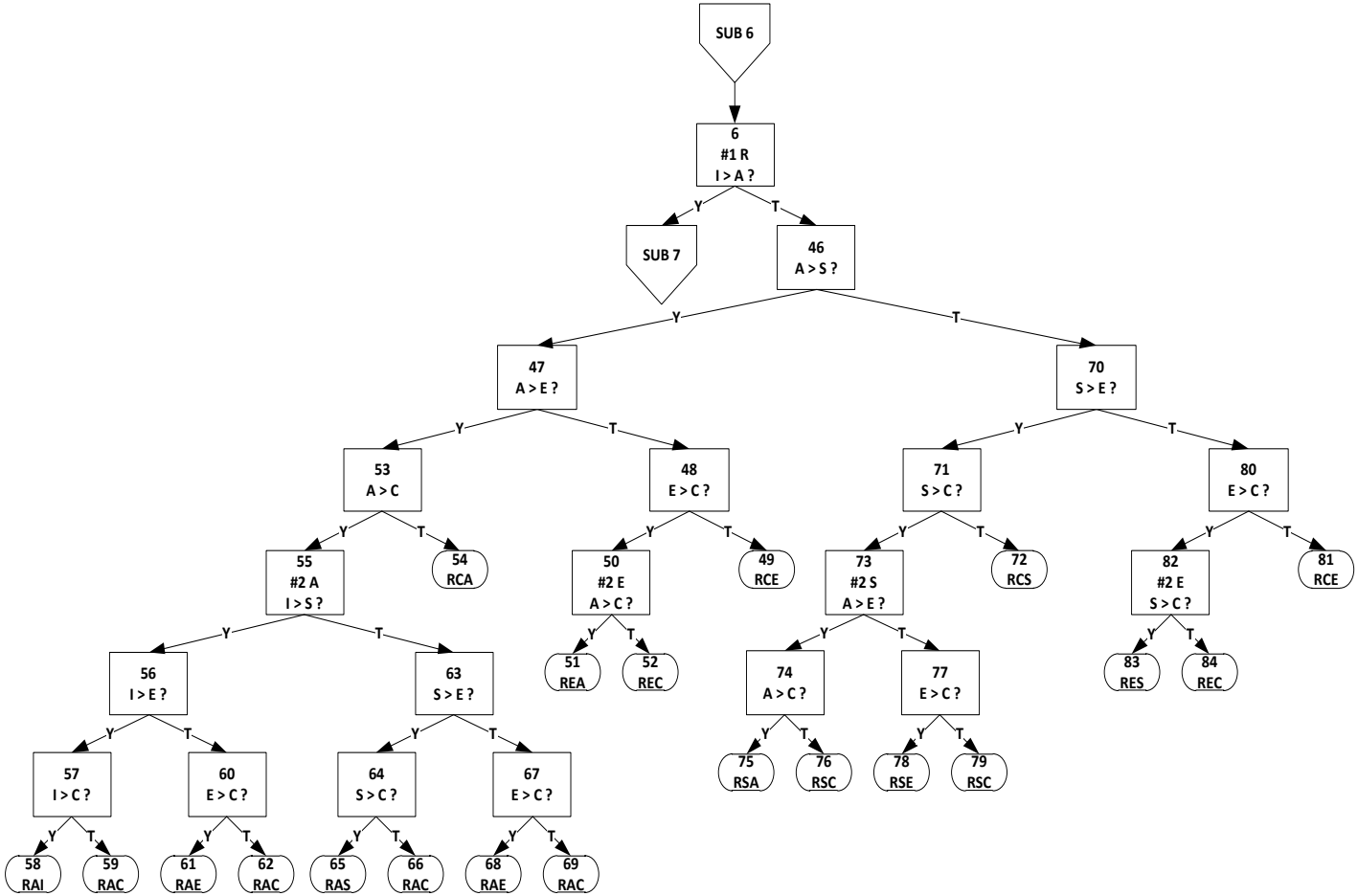
Gambar 4. *Rulebase* Sub 3



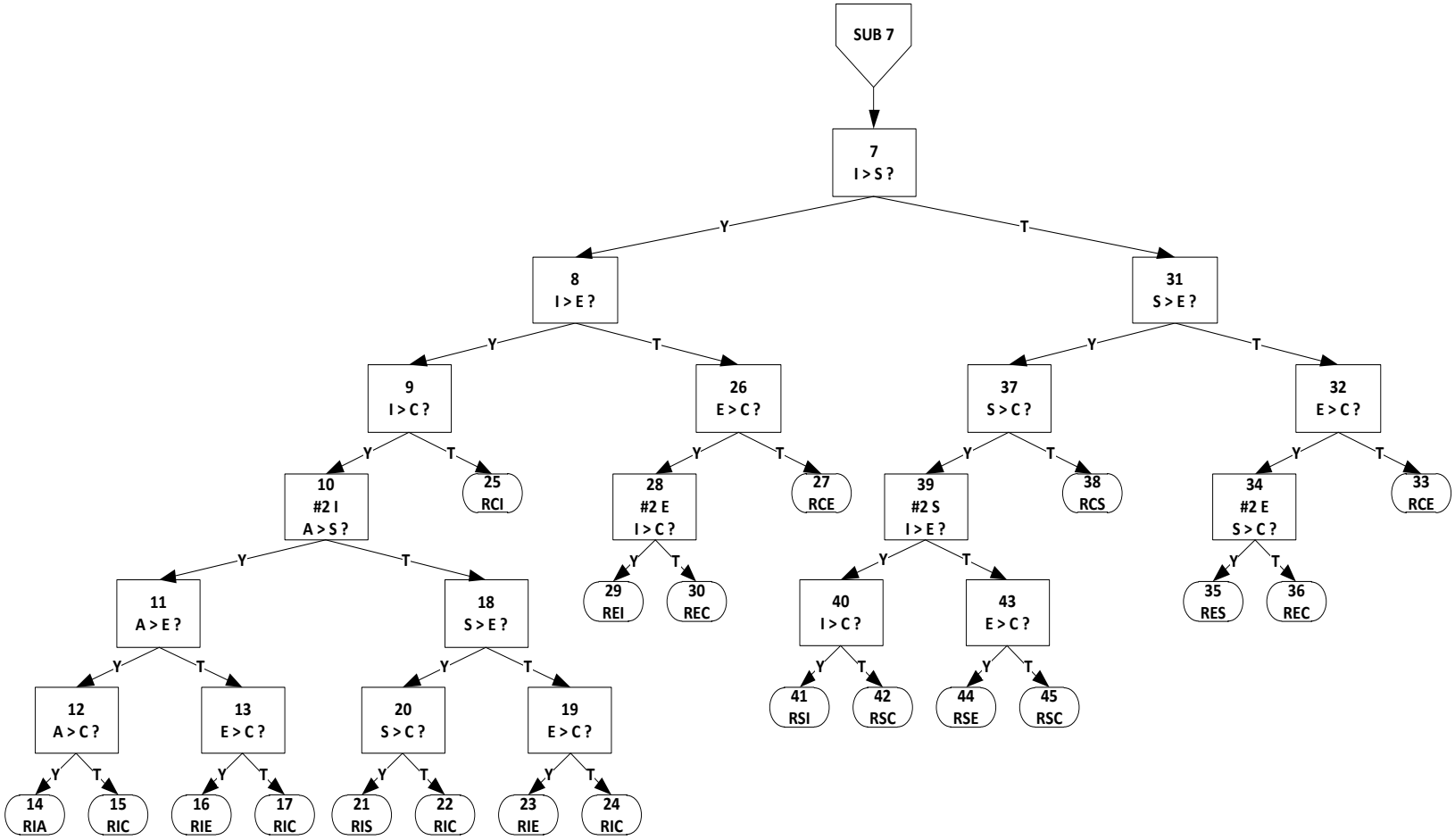
Gambar 5. *Rulebase* Sub 4



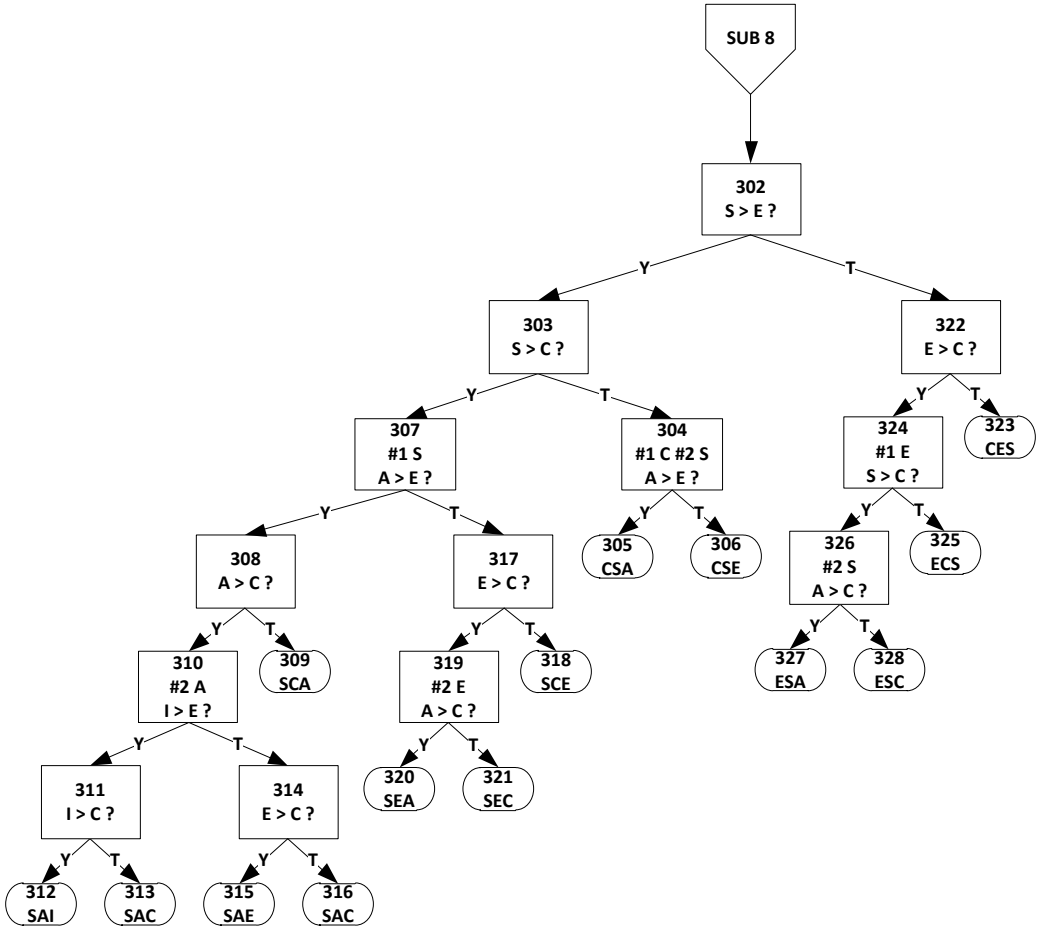
Gambar 6. *Rulebase* Sub 5



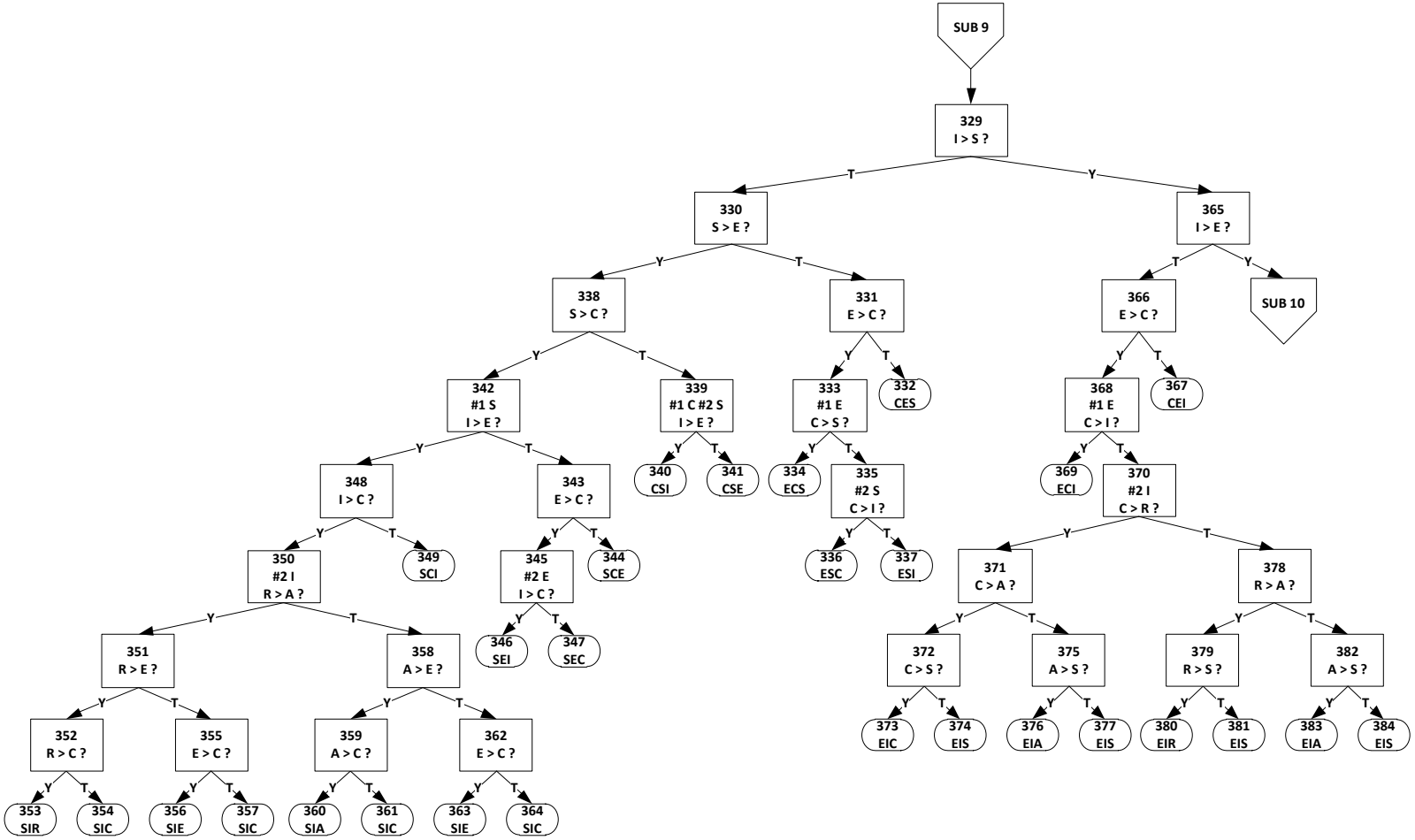
Gambar 7. *Rulebase* Sub 6



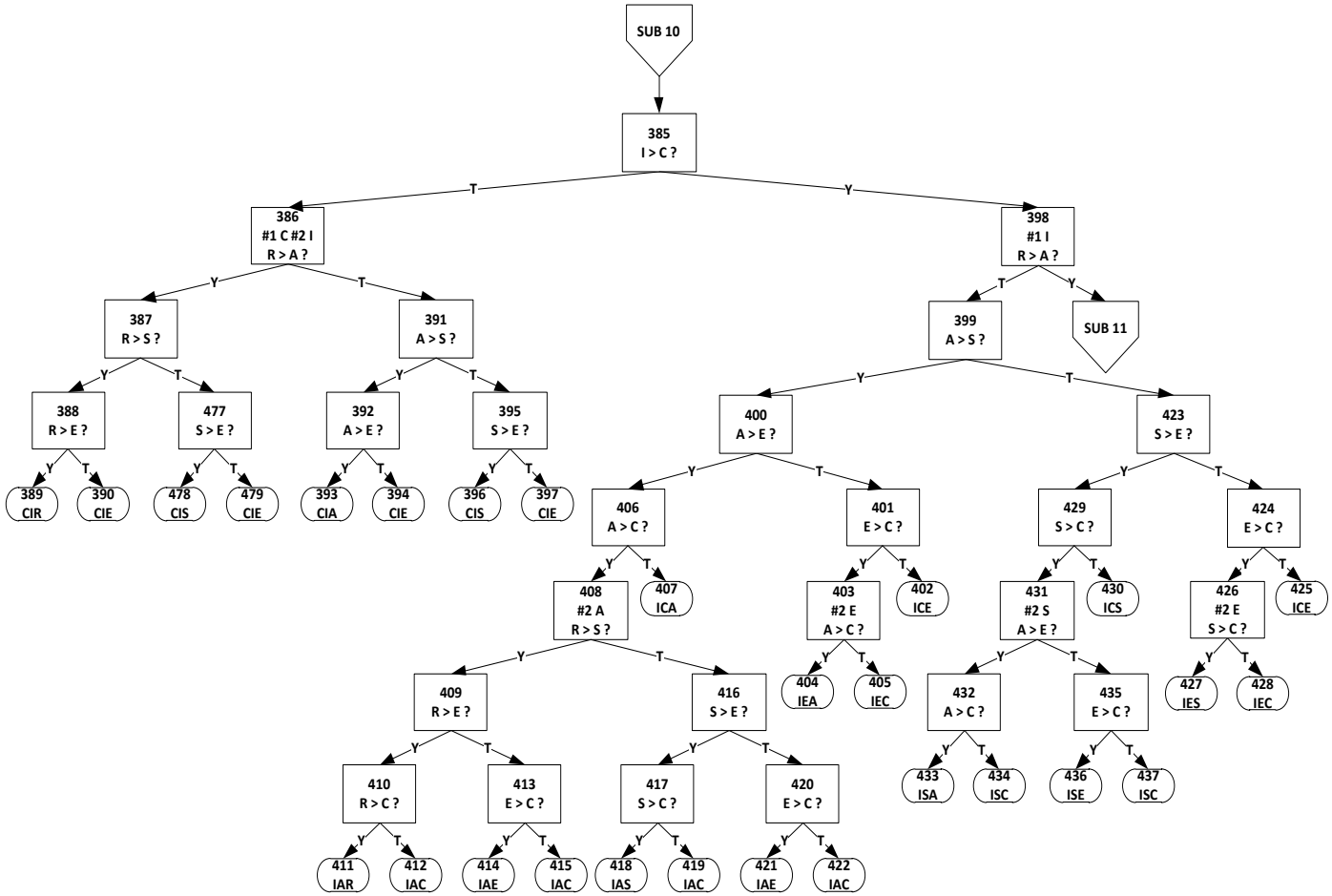
Gambar 8. *Rulebase* Sub 7



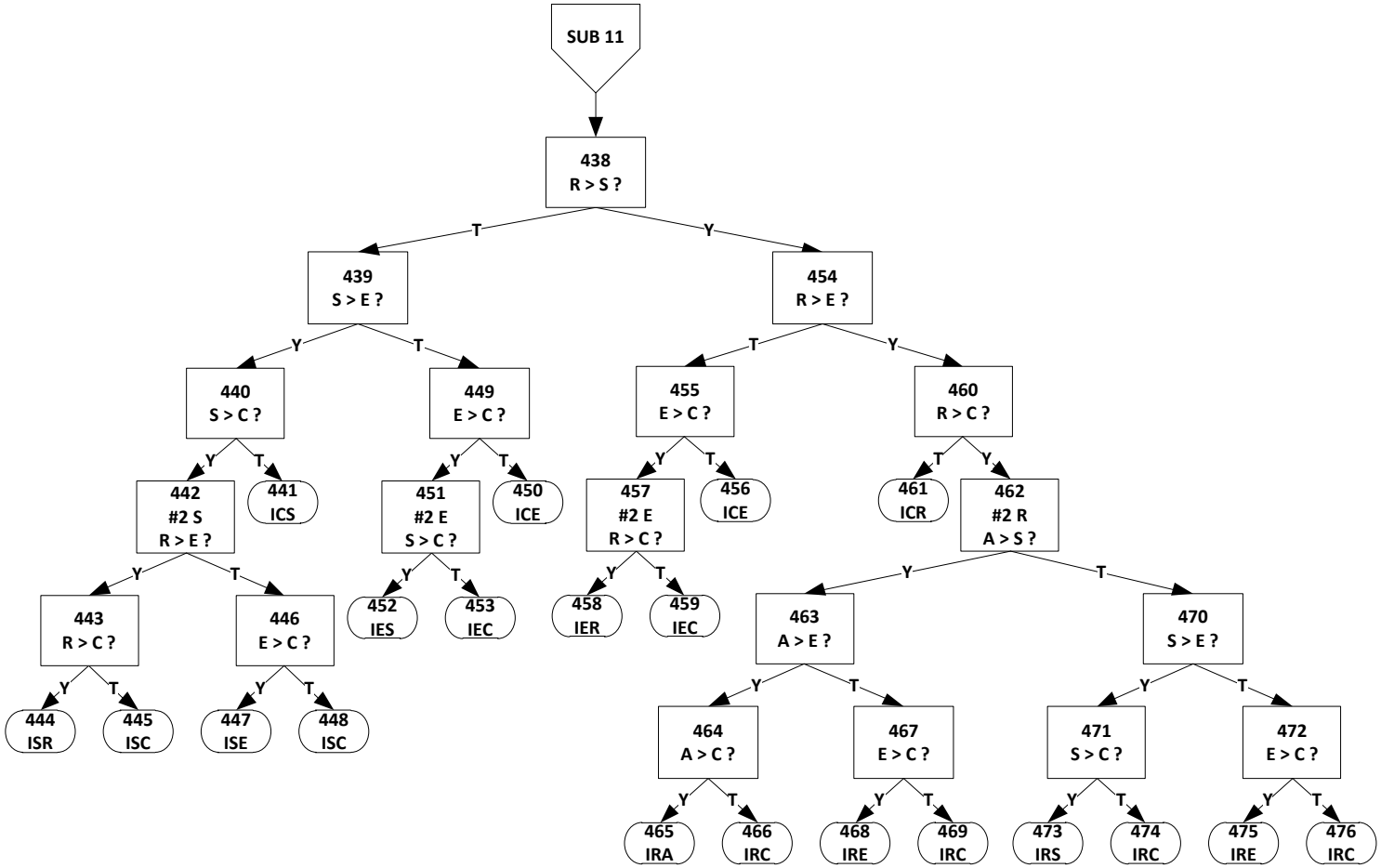
Gambar 9. *Rulebase* Sub 8



Gambar 10. *Rulebase* Sub 9



Gambar 11. *Rulebase* Sub 10



Gambar 12. *Rulebase* Sub 11

Lampiran 2.Hasil Uji *Alpha* Perangkat Lunak

LEMBAR PENGUJIAN AHLI REKAYASA PERANGKAT LUNAK

Berilah tanda *checklist* (√) pada kolom pilihan '"Ya" atau "Tidak" pada kolom "Taraf ketercapaian" yang ada pada tabel spesifikasi pengujian perangkat lunak *Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi*.

A. Tabel Spesifikasi Pengujian

Tabel 1. Spesifikasi Uji Membuka Aplikasi

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Membuka Aplikasi	Apabila MySQL belum aktif maka akan muncul notifikasi "Ada kesalahan dengan server, periksa apakah server sudah berjalan !".	✓	
		Pengguna masuk ke halaman *login* dan muncul menu fitur *login* (beserta *textbox* nama pengguna dan kata sandi), daftar, *about* (tentang), *help* (bantuan), dan batal.	✓	

Tabel 2. Spesifikasi Uji Halaman *About*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman *about*	Pengguna dapat masuk ke halaman *about* dengan cara memilih tombol tentang pada halaman *login*	✓	
2	Halaman *about*	Muncul label keterangan yang berisi nama aplikasi, versi, diskripsi singkat dan nama pengembang	✓	
		Pengguna dapat masuk ke semua fitur menu (info sistem dan bantuan) dengan memilih tombol-tombol tersebut yang ada pada halaman *about*	✓	
3	Fitur info sistem	Pengguna dapat melihat info sistem dengan cara memilih tombol info sistem pada halaman *about*	✓	
		Jendela info sistem dapat ditutup dengan cara memilih *icon* tutup (x) pada jendela info sistem	✓	
4	Keluar halaman *about*	Pengguna dapat keluar dari halaman *about* dengan memilih tombol ok pada halaman *about*, maka akan muncul halaman *login*	✓	

Tabel 3. Spesifikasi Uji Halaman *Login*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Halaman Login	Pengguna dapat masuk ke semua fitur menu (daftar, tentang, batal, dan bantuan) dengan memilih tombol-tombol tersebut yang ada pada halaman *login*	✓	
2	Login	Pengguna dapat *login* dengan cara menuliskan data diri pada *textbox* nama pengguna dan *textbox* kata sandi kemudian memilih tombol masuk	✓	
		Apabila pengguna memilih tombol masuk ketika *textbox* nama pengguna atau kata sandi atau keduanya belum terisi maka akan muncul notifikasi "login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar"	✓	
		Apabila data diri yang ditulis oleh pengguna tidak cocok dengan data pengguna yang tersimpan dalam *database* maka akan muncul notifikasi "login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar"	✓	
		Pengguna dapat kembali ke halaman *login* dengan cara memilih tombol ok pada notofikasi "login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar"	✓	
		Apabila data diri yang ditulis oleh pengguna cocok dengan data pengguna yang tersimpan dalam *database* maka akan muncul notifikasi "login berhasil"	✓	
3	Keluar aplikasi	Apabila pengguna memilih tombol batal pada halaman *login* maka akan muncul notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"	✓	
		Pengguna dapat keluar aplikasi dengan memilih tombol ya pada notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"	✓	
		Pengguna dapat batal keluar aplikasi dengan cara memilih tombol tidak pada notifikasi "Anda akan keluar dari aplikasi ini. Apakah anda yakin hendak menutup aplikasi ini?"	✓	

Tabel 4. Spesifikasi Uji Halaman Pendaftaran

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman pendaftaran	Pengguna dapat masuk ke halaman pendaftaran dengan cara memilih tombol daftar pada halaman *login*	✓	
2	Pendaftaran	Pengguna dapat melakukan pendaftaran dengan mengisi data diri pada *textbox* nama pengguna dan *textbox* kata sandi kemudian memilih tombol simpan	✓	
		Apabila data pengguna belum lengkap maka akan muncul notifikasi "Data masih kosong. Isilah dengan lengkap"	✓	
		Apabila nama pengguna atau kata sandi yang dituliskan lebih dari 15 karakter maka akan muncul notifikasi "Maaf data yang anda masukkan lebih dari 15 karakter"	✓	
		Apabila nama pengguna yang dituliskan telah terdaftar dalam database maka akan muncul notifikasi "Maaf nama pengguna tersebut sudah terdaftar"	✓	
		Pengguna dapat kembali ke halaman pendaftaran dengan memilih tombol ok pada setiap notifikasi peringatan gagal pendaftaran	✓	
		Apabila data pengguna sudah lengkap, tidak ada data yang lebih dari 15 karakter, dan nama pengguna yang dituliskan belum terdaftar dalam database maka data pengguna tersebut akan tersimpan dalam database dan akan muncul notifikasi "Pendaftaran berhasil!"	✓	
		Pengguna dapat memasuki halaman diagnosa untuk *normal user* dengan cara memilih tombol ok pada notifikasi "Pendaftaran berhasil!"	✓	
3	Keluar halaman pendaftaran	Pengguna dapat keluar dari halaman pendaftaran dengan memilih tombol kembali pada halaman pendaftaran, maka akan muncul halaman *login*	✓	

Tabel 5. Spesifikasi Uji Halaman Diagnosa Untuk *Normal User*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman diagnosa untuk *normal user*	Pengguna dapat masuk ke halaman diagnosa untuk *normal user* dengan cara berhasil *login* atau daftar dengan data tipe pengguna "user"	✓	
2	Diagnosa *normal user*	Muncul pernyataan awal pada label pernyataan untuk diagnosa awal, serta tombol ya, tidak, kenapa?, bantuan, dan *logout* (setiap tombol berfungsi dengan benar dan melakukan tugas masing-masing fitur)	✓	
		Pengguna dapat menjawab pernyataan yang muncul dengan cara memilih tombol ya atau tidak, sesuai dengan keadaan pengguna, maka akan muncul pernyataan selanjutnya (sesuai jawaban yang dipilih)	✓	
		Apabila pengguna sudah menjawab pernyataan maka tombol kembali akan muncul dan dapat dipilih	✓	
		Apabila data untuk pernyataan selanjutnya belum tersedia padahal data sebelumnya merupakan jenis data pertanyaan maka akan muncul notifikasi "Maaf data masih kosong"	✓	
		Pengguna dapat kembali ke halaman diagnosa dengan cara memilih tombol ok pada notifikasi "Maaf data masih kosong"	✓	
		Pengguna akan mendapatkan solusi dari diagnosa jika tidak terdapat lagi pernyataan yang perlu dijawab oleh pengguna karena pernyataan tersebut merupakan jenis data jawaban	✓	
		Apabila pengguna mendapatkan solusi diagnosa (jenis data jawaban) maka tombol jawab (ya dan tidak) tidak dapat di pilih lagi, sedangkan tombol detail info muncul dan dapat dipilih	✓	
		Apabila data *rulebase* pada *database* masih kosong maka akan muncul notifikasi "rulebase masih kosong"	✓	
		Aplikasi akan tertutup apabila pengguna memilih tombol ok pada notifikasi "rulebase masih kosong"	✓	
3	Ulang diagnosa	Pengguna dapat melakukan diagnosa ulang dengan cara memilih tombol ulang pada halaman diagnosa. Maka label pernyataan pada halaman diagnosa akan kembali pada pernyataan awal proses diagnosa	✓	

Tabel 6. Spesifikasi Uji Halaman Diagnosa Untuk *Admin*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman diagnosa untuk *admin*	Pengguna dapat masuk ke halaman diagnosa untuk *admin* dengan cara berhasil *login* sebagai akun dengan data tipe pengguna "admin"	✓	
2	Diagnosa *admin*	Muncul pernyataan awal pada label pernyataan untuk diagnosa awal, serta tombol ya, tidak, kenapa?, ubah, , kosongkan, bantuan, dan *logout* (setiap tombol berfungsi dengan benar dan melakukan tugas masing-masing fitur)	✓	
		Pengguna dapat menjawab pernyataan yang muncul dengan cara memilih tombol ya atau tidak, sesuai dengan keadaan pengguna, maka akan muncul pernyataan selanjutnya (sesuai dengan jawaban yang dipilih)	✓	
		Apabila pengguna sudah menjawab pernyataan maka tombol kembali, hapus, sisipkan fakta, dan ulang akan muncul dan dapat dipilih (setiap tombol dapat melakukan tugas fitur masing-masing dengan benar)	✓	
		Apabila data untuk pernyataan selanjutnya belum tersedia padahal data sebelumnya merupakan jenis data pertanyaan maka akan muncul notifikasi "Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?"	✓	
		Pengguna dapat masuk ke halaman tambah fakta dengan cara memilih tombol ya pada notifikasi "Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?"	✓	
		Pengguna dapat kembali ke halaman diagnosa *admin* tanpa menambah fakta baru dengan cara memilih tombol tidak pada notifikasi "Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?", maka akan muncul pernyataan sebelumnya pada halaman diagnosa	✓	
		Pengguna akan mendapatkan solusi dari diagnosa jika tidak terdapat lagi pernyataan yang perlu dijawab oleh pengguna karena pernyataan tersebut merupakan jenis data jawaban	✓	

Tabel 7. Spesifikasi Uji Halaman Diagnosa Untuk *Admin* (lanjutan)

		Apabila pengguna mendapatkan solusi diagnosa (jenis data jawaban) maka tombol jawab (ya dan tidak) tidak dapat di pilih lagi, sedangkan tombol detail info muncul dan dapat dipilih (melakukan tugas fitur detail info)	✓	
		Apabila data *rulebase* pada *database* masih kosong maka akan muncul notifikasi "rulebase masih kosong"	✓	
		Pengguna dapat masuk ke halaman tambah awal untuk memasukkan data awal pada *rulebase* dengan cara memilih tombol ok pada notifikasi "rulebase masih kosong"	✓	
3	Ulang diagnosa	Pengguna dapat melakukan diagnosa ulang dengan cara memilih tombol ulang pada halaman diagnosa. Maka label pernyataan pada halaman diagnosa akan kembali pada pernyataan awal proses diagnosa	✓	
4	Fitur ubah data	Pengguna dapat masuk ke halaman ubah data untuk mengubah data yang muncul dalam label pernyataan dengan cara memilih tombol ubah pada halaman diagnosa *admin*	✓	
5	Fitur hapus data	Pengguna dapat menghapus data yang muncul dalam label pernyataan (beserta anak-anaknya) dengan cara memilih tombol hapus pada halaman diagnosa *admin*, maka akan muncul notifikasi "Data berhasil dihapus!"	✓	
		Pengguna akan kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi "Data berhasil dihapus!", maka pada halaman diagnosa akan muncul pernyataan yang ditampilkan sebelum pernyataan yang dihapus tersebut	✓	
6	Fitur sisipkan fakta	Pengguna dapat masuk ke halaman penyisipan fakta untuk menyisipkan data diantara data yang telah tersedia dengan cara memilih tombol sisipkan fakta pada halaman diagnosa *admin*	✓	
7	Fitur kosongkan rulebase	Pengguna dapat mengosongkan data yang terdapat pada tabel *rulebase* di *database* dengan cara memilih tombol kosongkan yang ada di halaman diagnosa *admin*, maka akan muncul notifikasi "Tabel rule berhasil dikosongkan"	✓	
		Muncul halaman login setelah memilih tombol ok pada notifikasi "Tabel rule berhasil dikosongkan"	✓	

Tabel 8. Spesifikasi Uji Fitur Kembali Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur kembali pada halaman diagnosa	Tombol kembali pada halaman diagnosa (untuk *normal user* dan *admin*) akan muncul jika pengguna telah menjawab pernyataan	✓	
		Pengguna dapat mengulang pernyataan sebelumnya dengan cara memilih tombol kembali pada halaman diagnosa	✓	
		Apabila pengguna memilih tombol kembali padahal pernyataan yang dituju merupakan awal data maka akan muncul pemberitahuan "Anda saat ini berada di awal data"	✓	
		Pengguna dapat melanjutkan proses diagnosa dengan cara memilih tombol ok pada notifikasi "Anda saat ini berada di awal data", maka akan muncul kembali halaman diagnosa	✓	
		Tombol kembali pada halaman diagnosa (untuk *normal user* dan *admin*) akan hilang jika berada di awal data pernyataan diagnosa	✓	

Tabel 9. Spesifikasi Uji Fitur Detail Info Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur detail info pada halaman diagnosa	Tombol detail info pada halaman diagnosa (untuk *normal user* dan *admin*) akan muncul hanya jika proses diagnosa berada pada solusi atau kesimpulan diagnosa (jenis data jawaban)	✓	
		Pengguna dapat melihat detail info mengenai hasil kesimpulan diagnosa dengan cara memilih tombol detail info pada halaman diagnosa, maka akan muncul *frame* info penyedia jurusan yang menjadi hasil kesimpulan tersebut	✓	
		Pengguna dapat kembali ke halaman diagnosa dengan cara memilih *icon* tutup (x) pada *frame* info penyedia jurusan	✓	
		Tombol detail info pada halaman diagnosa (untuk *normal user* dan *admin*) akan hilang jika berada pada pernyataan diagnosis dengan jenis data pertanyaan	✓	

Tabel 10. Spesifikasi Uji Fitur Mengapa Pada Halaman Diagnosa

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Fitur mengapa pada halaman diagnosa	Tombol kenapa? akan selalu muncul pada halaman diagnosa (untuk *normal user* dan *admin*)	✓	
		Pengguna dapat menanyakan mengapa pernyataan tersebut diberikan dengan cara memilih tombol kenapa? pada halaman diagnosa, maka akan muncul *frame* penjelasan mengenai pernyataan tersebut	✓	
		Pengguna dapat kembali ke halaman diagnosa dengan memilih *icon* tutup (x) pada *frame* penjelasan pernyataan tersebut	✓	

Tabel 11. Spesifikasi Uji Halaman Ubah Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman ubah fakta	Pengguna dapat masuk ke halaman ubah fakta dengan cara memilih tombol ubah pada halaman diagnosa *admin*	✓	
2	Ubah fakta	Muncul data yang akan diubah pada label pernyataan dan label keterangan pada halaman ubah fakta	✓	
		Pengguna dapat langsung mengubah data pada label pernyataan dan label keterangan sesuai keinginan	✓	
		Perubahan data akan tersimpan di *database* dengan cara memilih tombol simpan yang ada di halaman ubah fakta, maka akan muncul notifikasi "Data berhasil diubah"	✓	
		Pengguna dapat kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi "Data berhasil diubah", maka akan muncul data yang telah berhasil diubah tersebut pada label pernyataan di halaman diagnosa *admin*	✓	
3	Batal ubah fakta	Pengguna dapat kembali ke halaman diagnosa *admin* tanpa jadi mengubah fakta dengan cara memilih tombol kembali pada halaman ubah fakta, maka akan muncul data yang tidak jadi diubah pada label pernyataan di halaman diagnosa *admin*	✓	

Tabel 12. Spesifikasi Uji Halaman Penyisipan Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman penyisipan fakta	Pengguna dapat masuk ke halaman penyisipan fakta dengan cara memilih tombol sisipkan fakta pada halaman diagnosa *admin*	✓	
2	Penyisipan fakta	Muncul data kode pernyataan yang muncul saat tombol sisipkan fakta pada halaman diagnosa *admin* dipilih (pernyataan yang nantinya akan diletakkan dibawah data sisipan baru)	✓	
		Pengguna dapat mengisi pernyataan dan keterangan dari fakta yang akan disisipkan pada *textbox* pertanyaan dan keterangan yang telah disediakan	✓	
		Pengguna dapat memilih letak data pernyataan yang muncul saat tombol sisipkan fakta pada halaman diagnosa *admin* dipilih, yaitu pada pilihan ya atau tidak untuk jawaban dari fakta yang akan disisipkan	✓	
		Data yang akan disisipkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman penyisipan fakta	✓	
		Apabila pernyataan belum dituliskan namun pengguna memilih tombol simpan, maka akan muncul notifikasi "Maaf tuliskan terlebih dahulu fakta di tempat yang sudah disediakan!"	✓	
		Pengguna dapat kembali ke halaman penyisipan fakta dengan cara memilih tombol ok pada notifikasi "Maaf tuliskan terlebih dahulu fakta di tempat yang sudah disediakan!"	✓	
		Apabila pengguna belum memilih letak data sebelumnya namun pengguna memilih tombol simpan, maka akan muncul notifikasi "Pilih salah satu Option yang tersedia"	✓	
		Pengguna dapat kembali ke halaman penyisipan fakta dengan cara memilih tombol ok pada notifikasi "Pilih salah satu Option yang tersedia"	✓	
		Apabila pernyataan telah ditulis dan pilihan letak data sebelumnya telah dipilih maka data tersimpan dalam *database* dan muncul notifikasi "data berhasil di sisipkan"	✓	

Tabel 13. Spesifikasi Uji Halaman Penyisipan Fakta (lanjutan)

		Pengguna dapat kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi "data berhasil di sisipkan", maka akan muncul data yang telah berhasil disisipkan pada label pernyataan di halaman diagnosa *admin*	✓	
3	Batal menyisipan fakta	Pengguna dapat melanjutkan proses diagnosa kembali di halaman diagnosa *admin* tanpa jadi menyisipkan fakta dengan cara memilih tombol kembali pada halaman penyisipan fakta	✓	

Tabel 14. Spesifikasi Uji Halaman Tambah Fakta

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman tambah fakta	Pengguna dapat masuk ke halaman tambah fakta dengan cara memilih tombol ya pada notifikasi "Data yang anda tuju belum tersedia. Apakah anda ingin menambah fakta baru?" pada halaman diagnosa *admin*	✓	
2	Penambahan fakta	Pengguna dapat mengisi pernyataan dan keterangan dari fakta yang akan ditambahkan pada *textbox* fakta dan *textbox* keterangan	✓	
		Pengguna dapat memilih jenis fakta yang akan ditambahkan, yaitu jenis pertanyaan atau jawaban	✓	
		Data yang akan ditambahkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman tambah fakta	✓	
		Apabila pernyataan belum dituliskan kemudian pengguna memilih tombol simpan maka akan muncul notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"	✓	
		Pengguna dapat kembali ke halaman tambah fakta dengan cara memilih tombol ok pada notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"	✓	
		Apabila pengguna belum memilih jenis fakta yang akan ditambah namun memilih tombol simpan maka akan muncul notifikasi "Anda belum memilih jenis masukan, pilih terlebih dahulu jenis data yang diinginkan"	✓	

Tabel 15. Spesifikasi Uji Halaman Tambah Fakta (lanjutan)

		Pengguna dapat kembali ke halaman tambah fakta dengan cara memilih tombol ok pada notifikasi "Anda belum memilih jenis masukan, pilih terlebih dahulu jenis data yang diinginkan"	✓	
		Apabila pernyataan telah ditulis dan jenis fakta yang akan ditambahkan telah dipilih maka data tersimpan dalam *database* dan muncul notifikasi "Fakta berhasil disimpan!"	✓	
		Pengguna dapat kembali ke halaman diagnosa admin dengan cara memilih tombol ok pada notifikasi "Fakta berhasil disimpan!", maka akan muncul data yang telah ditambahkan pada label pernyataan di halaman diagnosa *admin*	✓	
3	Batal tambah fakta	Pengguna dapat melanjutkan proses diagnosa kembali di halaman diagnosa *admin* tanpa jadi menambahkan fakta dengan cara memilih tombol kembali pada halaman penyisipan fakta	✓	

Tabel 16. Spesifikasi Uji Fitur Bantuan (*Help*)

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Bantuan	Tombol bantuan dengan icon "?" muncul di setiap halaman aplikasi	✓	
		Pengguna dapat melihat halaman bantuan yang berisi petunjuk mengenai penggunaan halaman yang sedang diakses dengan cara memilih icon bantuan (?) yang ada pada halaman tersebut	✓	
		Pengguna dapat kembali ke halaman yang sedang diakses dengan memilih *icon* tutup (x) pada halaman bantuan	✓	
		Khusus pada halaman bantuan utama untuk halaman diagnosa *admin* pengguna dapat masuk ke 2 jenis kategori petunjuk. Kategori tersebut adalah diagnosis dan pengaturan	✓	
		Pengguna pada halaman kategori bantuan untuk halaman diagnosis *admin* dapat kembali ke halaman bantuan utama dengan memilih icon tutup (x) pada halaman kategori bantuan tersebut	✓	

Tabel 17. Spesifikasi Uji Halaman Tambah Awal

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk halaman tambah awal	Pengguna dapat masuk ke halaman tambah awal dengan cara memilih tombol ok pada notifikasi "rulebase masih kosong" pada halaman diagnosa *admin*	✓	
2	Isi fakta awal	Pengguna dapat mengisi pernyataan dan keterangan untuk fakta awal pada *textbox* fakta dan *textbox* keterangan	✓	
		Data yang akan dimasukkan dapat tersimpan dalam *database* dengan cara memilih tombol simpan pada halaman tambah awal	✓	
		Apabila pernyataan belum dituliskan namun pengguna memilih tombol simpan maka akan muncul notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan!"	✓	
		Pengguna dapat kembali ke halaman tambah awal dengan cara memilih tombol ok pada notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"	✓	
		Apabila pernyataan telah ditulis dan pengguna memilih tombol simpan maka data tersimpan dalam *database* dan muncul notifikasi "Fakta pertama berhasil disimpan"	✓	
		Pengguna dapat masuk kembali ke halaman diagnosa *admin* dengan cara memilih tombol ok pada notifikasi "Fakta pertama berhasil disimpan"	✓	
3	Keluar halaman tambah fakta	Pengguna dapat tidak menambahkan data awal dengan cara memilih tombol batal pada halaman tambah awal, maka akan muncul halaman *login*	✓	

Tabel 18. Spesifikasi Uji Fitur *Logout*

No	Aktivitas / Menu	Hasil unjuk kerja yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	*Logout*	Tombol *logout* muncul di setiap halaman setelah pengguna berhasil *login*	✓	
		Pengguna dapat *logout* dengan cara memilih tombol *logout*, maka akan muncul kembali ke halaman *login*	✓	

B. Kesimpulan

Berdasarkan hasil dari uji ahli rekayasa perangkat lunak yang dilakkan dengan panduan tabel spesifikasi diatas, maka perangkat lunak *Sistem Pendukung Keputusan Pemilihan Jurusan di Perguruan Tinggi* dinyatakan:

- ✓ Memiliki unjuk kerja Baik
- □ Memiliki unjuk kerja Kurang Baik dan Butuh Revisi
- □ Tidak Berfungsi

*beri tanda *checklist* pada salah satu pilihan

C. Saran

Pengujian fungsi yang ada sebaiknya tidak perlu menguji fitur untuk user yang berbeda (cukup fungsinya ada).

Yogyakarta, September 2013

Handaru Jati, Ph.D
NIP. 19740511 199903 1 002

Lampiran 3. Angket Uji *Beta*

LEMBAR PENILAIAN APLIKASI

Nama : ……..

Profesi : *) Mahasiswa / Siswa Sekolah Menengah Atas

Instansi : ……..

Jurusan Yang Diminati : 1. ……

2. ……

3. ……

Berilah tanda centang (√) pada kolom jawaban yang telah disediakan sesuai dengan penilaian anda.

Keterangan :

SS : Sangat Setuju ST : Setuju RG : Ragu – ragu

TS : Tidak setuju STS : Sangat tidak setuju

No	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1.	Secara keseluruhan, saya merasa puas dengan kemudahan penggunaan sistem ini.	…	…	…	…	…
2.	Cara penggunaan sistem ini sangat simpel.	…	…	…	…	…
3.	Saya dapat memutuskan pilihan saya dengan efektif ketika menggunakan sistem ini.	…	…	…	…	…
4.	Saya dapat dengan cepat memutuskan pilihan saya menggunakan sistem ini.	…	…	…	…	…
5.	Saya dapat memutuskan pilihan saya dengan efisien ketika menggunakan sistem ini.	…	…	…	…	…
6.	Saya merasa nyaman menggunakan sistem ini.	…	…	…	…	…
7.	Sistem ini sangat mudah dipelajari.	…	…	…	…	…
8.	Saya yakin saya akan lebih percaya diri dengan keputusan saya ketika menggunakan sistem ini.	…	…	…	…	…
9.	Jika terjadi error, sistem ini memberikan pesan pemberitahuan tentang langkah yang saya lakukan untuk mengatasi masalah.	…	…	…	…	…

10.	Kapanpun saya melakukan kesalahan, sistem bisa kembali digunakan dengan cepat	…	…	…	…	…
11.	Informasi yang disediakan sistem ini sangat jelas.	…	…	…	…	…
12.	Mudah untuk menemukan informasi yang saya butuhkan.	…	…	…	…	…
13.	Informasi yang diberikan oleh sistem ini sangat mudah dipahami.	…	…	…	…	…
14.	Informasi yang diberikan sangat efektif dalam membantu memutuskan pilihan saya.	…	…	…	…	…
15.	Tata letak informasi yang terdapat di layar monitor sangat jelas.	…	…	…	…	…
16.	Tampilan sistem ini sangat memudahkan.	…	…	…	…	…
17.	Saya suka menggunakan tampilan sistem semacam ini.	…	…	…	…	…
18.	Sistem ini memberikan semua fungsi dan kapabilitas yang saya perlukan.	…	…	…	…	…
19.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini.	…	…	…	…	…

Setelah memakai aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan, muncul hasil akhir berupa jurusan yang sesuai dengan tipe kepribadian anda.Jurusan tersebut adalah :

……

……

……

……

……

……

……

Tanda Tangan

……..

Keterangan :

*)Coret yang tidak perlu.

Lampiran 4.Rekap Hasil Uji *Beta* Perangkat Lunak

RESPONDEN	PERTANYAAN																		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
1	3	5	4	4	5	5	5	4	3	3	5	5	5	5	5	5	5	4	5
2	4	5	2	2	3	3	5	3	3	3	3	3	4	3	4	4	3	2	3
3	4	4	5	5	5	4	5	4	3	3	5	5	5	5	4	4	4	4	4
4	2	5	3	4	4	5	5	3	5	5	5	5	5	5	5	5	5	4	4
5	5	5	3	3	3	5	5	4	4	4	4	4	5	4	5	5	5	4	4
6	5	5	3	3	3	5	5	4	4	4	4	4	5	4	5	5	5	4	4
7	2	5	4	4	4	5	5	4	4	5	5	4	4	3	4	4	4	3	3
8	5	5	4	3	3	4	4	2	2	5	4	4	4	4	4	4	4	4	4
9	3	4	2	2	2	3	4	2	4	4	4	3	4	2	4	4	4	3	3
10	3	4	2	2	2	2	3	2	2	4	3	4	4	2	4	4	2	3	2
11	3	4	1	2	2	2	4	1	3	4	3	3	3	2	4	4	4	3	4
12	2	5	2	2	2	2	5	2	4	3	3	3	4	2	4	5	4	3	2
13	4	4	3	3	3	4	4	4	3	3	3	4	4	3	4	4	4	3	4
14	3	4	2	2	2	2	4	2	2	5	4	4	4	3	4	4	4	4	3
15	4	5	3	3	3	5	5	3	4	4	3	4	5	3	4	4	4	4	4
16	4	5	3	3	3	5	5	3	5	5	5	5	5	3	5	5	3	3	4
17	5	5	2	2	3	5	5	3	5	5	5	5	5	3	5	5	4	3	4
18	3	5	2	2	2	4	4	3	4	3	4	4	4	2	4	4	3	2	2
19	5	5	3	3	3	5	5	3	4	4	4	4	5	4	5	5	5	3	4
20	2	5	2	2	2	2	5	2	2	2	3	2	4	2	4	4	4	2	2
21	2	4	2	2	2	4	4	3	4	4	3	4	4	3	4	4	3	3	2
22	3	5	4	4	4	4	4	3	4	4	4	4	4	4	4	4	4	3	3
23	3	4	3	2	2	3	4	3	4	4	4	3	4	3	4	4	3	2	2
24	5	5	4	4	4	5	5	4	4	5	5	5	5	4	5	5	5	4	5
25	4	5	4	4	4	4	5	4	4	4	4	4	4	4	5	5	4	4	4
26	3	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
27	3	4	2	2	3	3	4	3	3	4	4	4	3	3	4	4	4	3	4
28	3	4	2	2	2	3	4	3	4	4	4	3	3	3	4	4	4	3	3

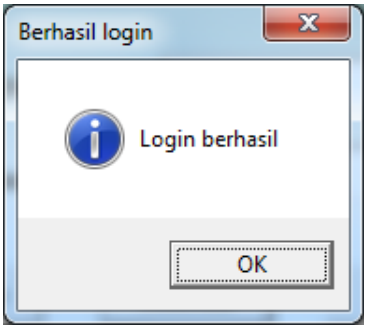
29	4	4	3	3	3	4	4	3	3	4	4	4	3	3	4	4	4	3	4
30	4	4	3	3	3	4	4	3	4	4	4	4	4	3	4	4	4	4	4
31	4	5	4	4	4	5	5	4	3	4	4	4	4	4	4	4	5	4	4
32	4	5	3	3	3	4	4	3	3	4	4	3	4	3	5	5	4	4	4
33	3	4	3	3	3	4	4	3	4	4	4	3	4	3	4	4	4	4	3
34	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	3	4	4
35	4	5	3	3	3	4	5	3	4	5	5	5	5	3	5	4	4	4	4
36	3	4	3	3	3	3	4	3	3	4	3	3	3	3	4	4	4	4	2
37	3	5	3	3	3	4	4	3	5	4	4	4	3	3	4	4	4	3	3
38	3	5	3	2	3	3	5	3	3	4	3	3	3	3	4	4	4	4	2
39	4	5	3	3	3	4	5	3	4	5	5	5	5	4	4	4	4	4	5
40	4	4	4	4	4	4	3	4	4	4	4	4	4	4	4	4	3	3	3
41	4	4	4	3	4	4	4	3	4	4	4	4	4	4	4	4	3	4	4
42	3	4	3	5	3	4	5	2	2	4	4	4	4	3	3	4	4	3	3
43	3	5	3	3	3	2	3	2	2	4	2	2	3	2	3	2	2	2	3
44	4	4	3	3	3	3	4	3	4	4	4	4	4	3	4	4	4	3	3
45	2	4	3	2	2	2	5	3	5	5	2	3	3	3	5	3	3	3	3
46	5	4	3	3	3	4	4	4	5	5	5	3	4	3	4	4	4	3	3
47	3	4	4	4	4	3	4	3	2	3	4	3	4	4	4	4	3	3	3
48	4	5	3	3	3	5	4	2	3	3	4	4	3	4	5	4	4	3	4
49	4	5	4	4	4	5	5	3	4	4	3	4	4	4	4	3	4	4	4
50	4	5	4	4	3	4	4	4	3	3	4	4	5	4	5	4	4	4	3
51	4	5	3	4	5	4	5	4	5	5	4	4	5	4	4	4	4	3	4
52	4	5	4	4	4	4	5	3	4	5	4	4	4	4	5	5	4	4	4
53	4	4	3	2	3	4	4	3	5	5	4	4	4	3	4	4	4	3	4
54	4	5	4	4	4	5	5	4	5	5	5	4	4	5	5	5	4	5	4
55	4	5	5	5	5	5	5	4	5	5	5	5	5	4	5	5	5	4	5
56	4	5	5	4	3	4	4	3	5	4	5	5	4	4	3	4	5	3	5
57	4	4	4	3	4	5	3	4	4	4	4	5	4	4	3	4	5	4	4
58	4	4	4	3	4	5	5	4	4	4	4	5	4	4	5	4	4	4	4
59	5	5	5	3	4	5	5	4	5	5	5	5	5	5	5	5	4	4	5
60	5	5	4	4	4	4	5	4	4	4	4	4	4	4	5	4	5	4	4
JUMLAH	219	274	193	187	195	235	265	190	225	246	239	236	246	206	256	251	237	206	214

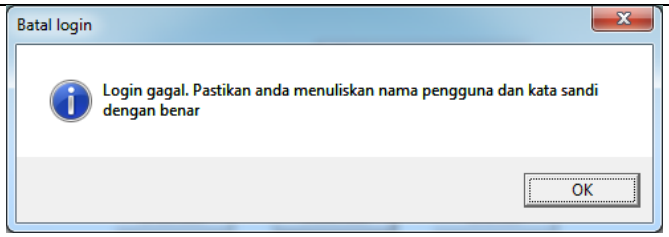
Lampiran 5.*Test Case* Pengujian Faktor *Functionality*

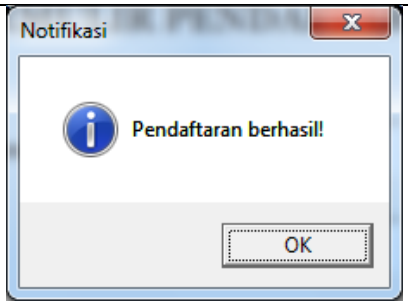
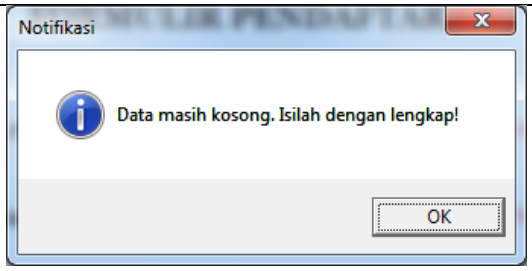
Pengujian faktor kualitas *functionality* dengan menggunakan *tase case* dibagi menjadi 2 bagian yaitu *Test Case* Pengujian Faktor Kualitas *Functionality Primer dan Test Case* Pengujian Faktor Kualitas *Functionality Contributing.* Berikut merupakan rincian hasil pengujian terhadap 2 bagian tersebut :

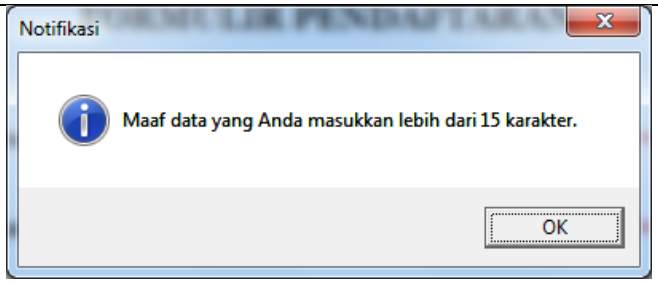
Tabel 19.*Test Case* Pengujian Faktor Kualitas *Functionality Primer*

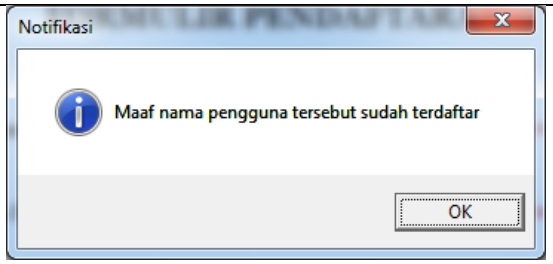
No.	**Pengujian**		**Lolos / Gagal**
1.	Nama tes	*LoginTest*1	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat terhubung dengan *database* dan dapat mencocokan *username* serta *password* sehingga *login* berhasil	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar	
	Data tes	Tombol “Masuk” dengan kombinasi nama pengguna dan kata sandi yang benar	
	Langkah	Mengisi *textbox* nama pengguna dan kata sandi kemudian klik tombol “Masuk”	
	Hasil yang diharapkan	- Proses *login* dijalankan - Muncul notifikasi “Login berhasil” - Tampil halaman Diagnosa Admin jika tipe pengguna “admin” - Tampil halaman Diagnosa User jika tipe pengguna “user”	
	Hasil pengamatan	Sesuai dengan harapan	

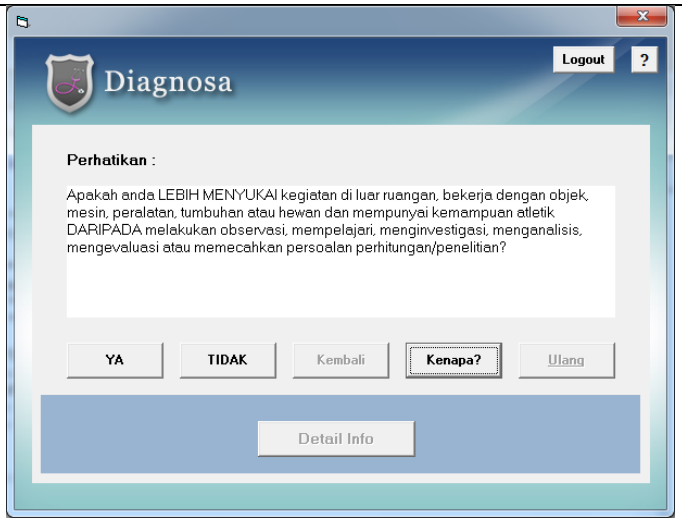
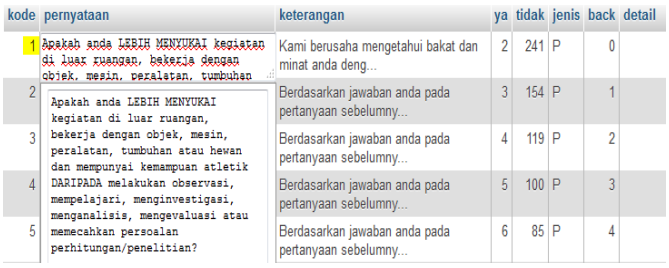
	Screenshoot	 	
2.	Nama tes	*LoginTest*2	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan pesan peringatan jika *login* gagal	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar	
	Data tes	Tombol "Masuk" dengan kombinasi nama pengguna dan kata sandi yang salah	
	Langkah	Mengisi *textbox* nama pengguna dan kata sandi kemudian klik tombol "Masuk"	
	Hasil yang diharapkan	- Proses *login* dijalankan - Muncul notifikasi "Login gagal. Pastikan anda menuliskan nama pengguna dan kata sandi dengan benar"	
	Hasil pengamatan	Sesuai dengan harapan	

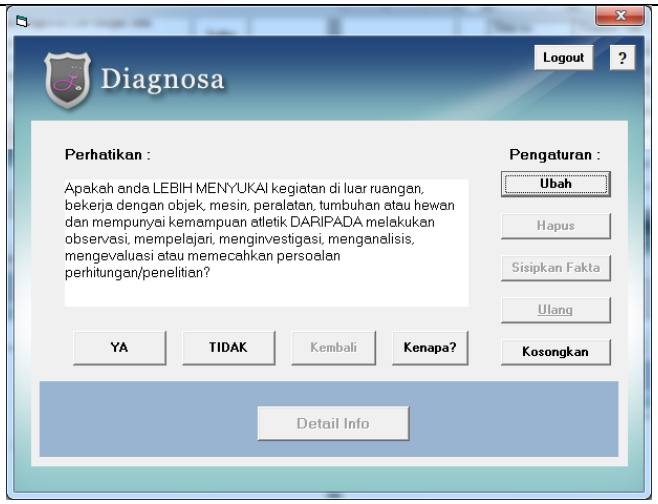
	Screenshoot		
3.	Nama tes	*SignupTest*1	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat terhubung dengan *database* dan dapat menyimpan data pendaftaranpada tabel *login* sehingga pendaftaran berhasil	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman pendaftaran dengan cara klik tombol "Daftar" pada halaman *Login.*	
	Data tes	Tombol "Simpan" dengan nama pengguna dan kata sandi yang diisi lengkap, nama pengguna belum dipakai, dan tidak lebih dari 15 karakter	
	Langkah	Mengisi *textbox* nama pengguna dan kata sandi kemudian klik tombol "Simpan"	
	Hasil yang diharapkan	- Proses *signup* dijalankan - Muncul notifikasi "Pendaftaran berhasil" - Tampil halaman Diagnosa User	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

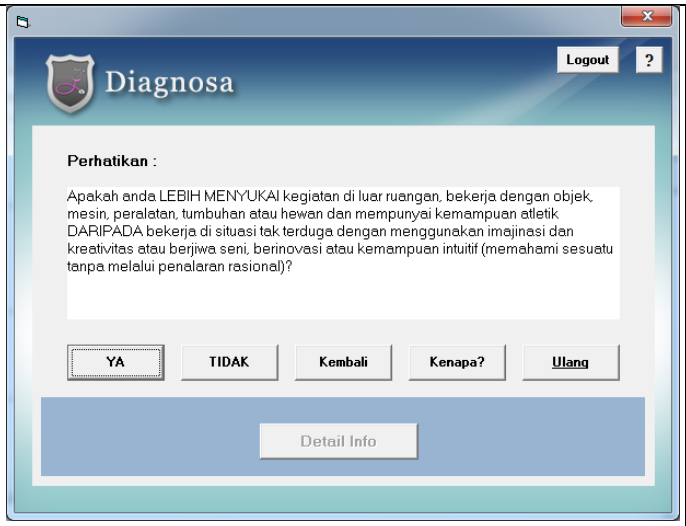
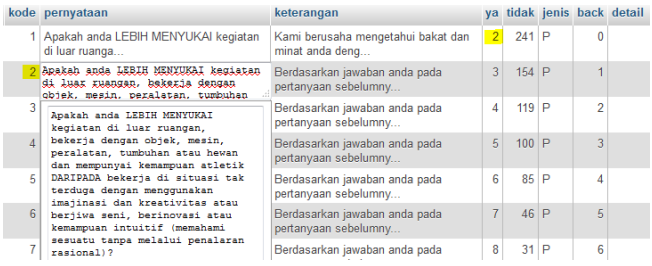
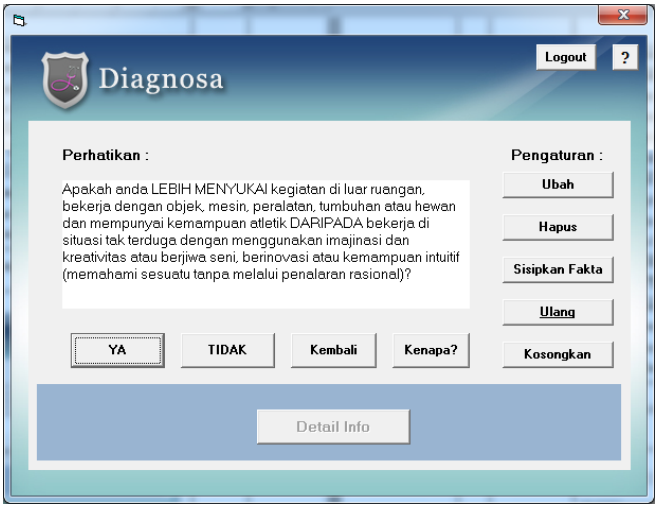
			
4.	Nama tes	*SignupTest*2	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan pesan peringatan jika data pendaftaran tidak diisi dengan lengkap	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman pendaftaran dengan cara klik tombol "Daftar" pada halaman *Login.*	
	Data tes	Tombol "Simpan" dengan nama pengguna dan kata sandi yang tidak diisi lengkap	
	Langkah	Mengisi *textbox* nama pengguna atau *textbox* kata sandi atau tidak mengisi sama sekali kemudian klik tombol "Simpan"	
	Hasil yang diharapkan	- Proses *signup* dijalankan - Muncul notifikasi "Data masih kosong. Isilah dengan lengkap! "	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
5.	Nama tes	*SignupTest*3	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan pesan peringatan jika data pendaftaran lebih dari 15 karakter	

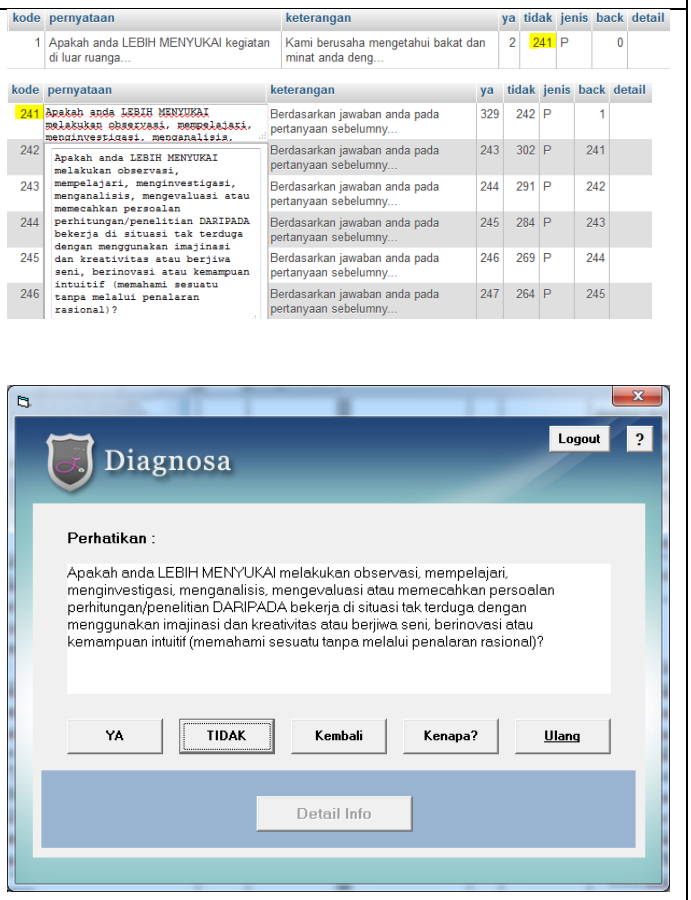
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman pendaftaran dengan cara klik tombol “Daftar” pada halaman *Login.*	
	Data tes	Tombol “Simpan” dengan nama pengguna atau kata sandi atau keduanya yang diisi lebih dari 15 karakter	
	Langkah	Mengisi *textbox* nama pengguna atau *textbox* kata sandi atau keduanya dengan lebih dari 15 karakter kemudian klik tombol “Simpan”	
	Hasil yang diharapkan	- Proses *signup* dijalankan - Muncul notifikasi “Maaf data yang Anda masukkan lebih dari 15 karakter ”	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
6.	Nama tes	*SignupTest*4	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan pesan peringatan jika nama pengguna telah digunakan dan ada pada *database* tabel *login*	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar Aplikasi sudah masuk pada halaman pendaftaran dengan cara klik tombol “Daftar” pada halaman *Login*	
	Data tes	Tombol “Simpan” dengan nama pengguna yang sudah digunakan dan ada dalam *database* tabel *login*	
	Langkah	Mengisi *textbox* nama pengguna dengan *username* yang sudah ada di *database* tabel *login* kemudian klik tombol	

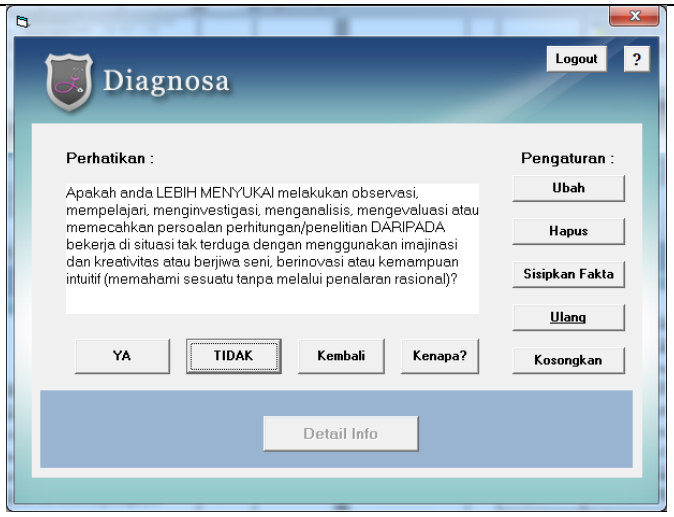
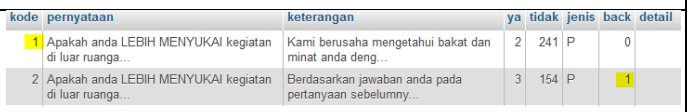
		"Simpan"	
	Hasil yang diharapkan	- Proses *signup* dijalankan - Muncul notifikasi "Maaf nama pengguna tersebut sudah terdaftar "	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
7.	Nama tes	Diagnosa*UserTest*	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat mengambil data *rulebase* dari *database* dan menampilkan pada halaman Diagnosa *User* jika berhasil *signup* atau *login* dengan tipe pengguna "user"	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar Pengguna telah berhasil *signup* atau *login* sebagai *user*	
	Data tes	Tombol "OK" pada notifikasi "Pendaftaran berhasil " atau notifikasi "Login berhasil"	
	Langkah	Melakukan *signup* atau *login* kemudian klik tombol "OK" pada notifikasi "Pendaftaran berhasil " atau notifikasi "Login berhasil"	
	Hasil yang diharapkan	- Halaman *Login* maupun halaman *Signup* tertutup - Muncul halaman Diagnosa *User* dengan data pertama (kode = 1) *rulebase*	
	Hasil pengamatan	Sesuai dengan harapan	

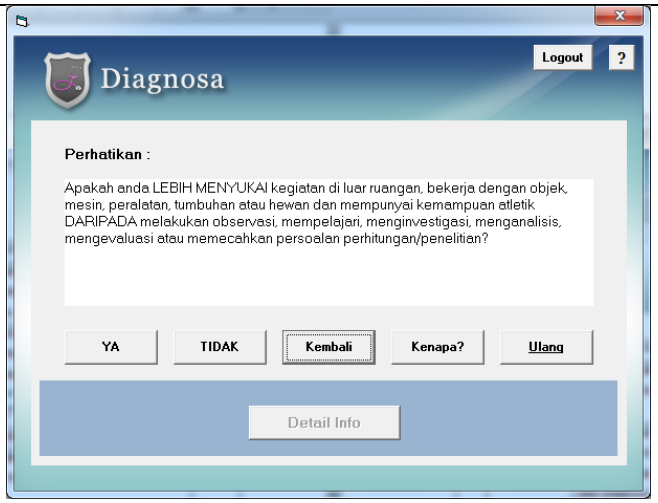
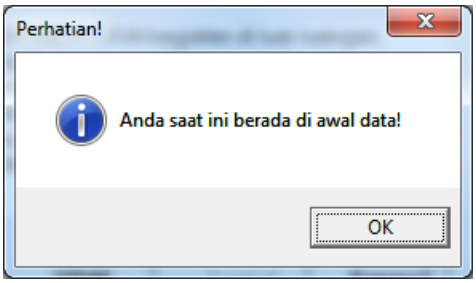
	Screenshoot	 	
8.	Nama tes	Diagnosa*AdminTest*	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat mengambil data *rulebase* dari *database* dan menampilkan pada halaman Diagnosa *Admin* jika berhasil *login* dengan tipe pengguna "admin"	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar Pengguna telah berhasil *login* sebagai *admin*	
	Data tes	Tombol "OK" pada notifikasi "Login berhasil"	
	Langkah	Melakukan *login* kemudian klik tombol "OK" pada notifikasi "Login berhasil"	
	Hasil yang diharapkan	- Halaman *Login* tertutup - Muncul halaman Diagnosa *Admin* dengan data pertama (kode = 1) *rulebase*	
	Hasil pengamatan	Sesuai dengan harapan	

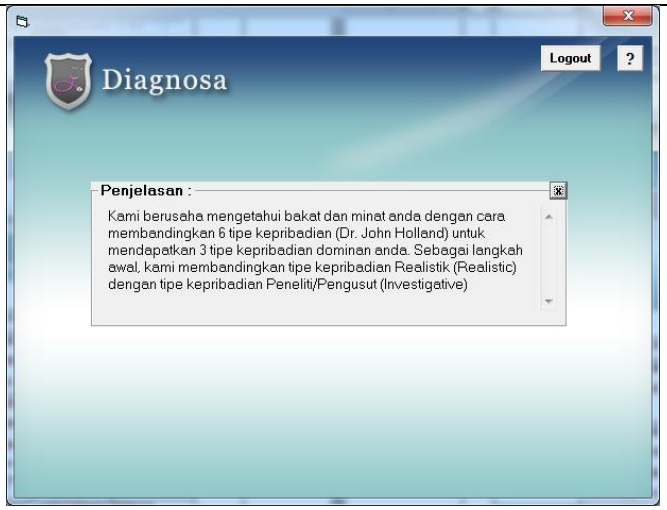
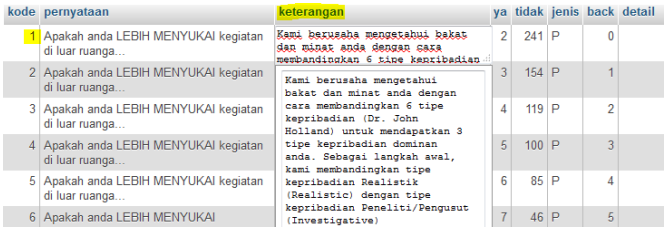
	Screenshoot		
9.	Nama tes	*PrimaryFunctionality*1	**Lolos**
	Tujuan	Menguji apakah pernyataan yang ditampilkan pada halaman Diagnosa berganti dengan pernyataan selanjutnya untuk jawaban ya (sesuai kode *rulebase* untuk ya pada pernyataan tersebut) jika memilih tombol "YA"	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan file database yang benar. Aplikasi sudah masuk pada halaman Diagnosa *Admin* atau Diagnosa *User*.	
	Data tes	Tombol "YA" pada halaman Diagnosa	
	Langkah	Klik tombol "YA" pada halaman Diagnosa	
	Hasil yang diharapkan	- Proses jawaban ya dijalankan - Muncul pernyataan selanjutnya untuk jawaban ya pada pertanyaan tersebut - Tombol "Kembali" , "Ulang", "Hapus", dan "Sisipkan Fakta" muncul	
	Hasil pengamatan	Sesuai dengan harapan	

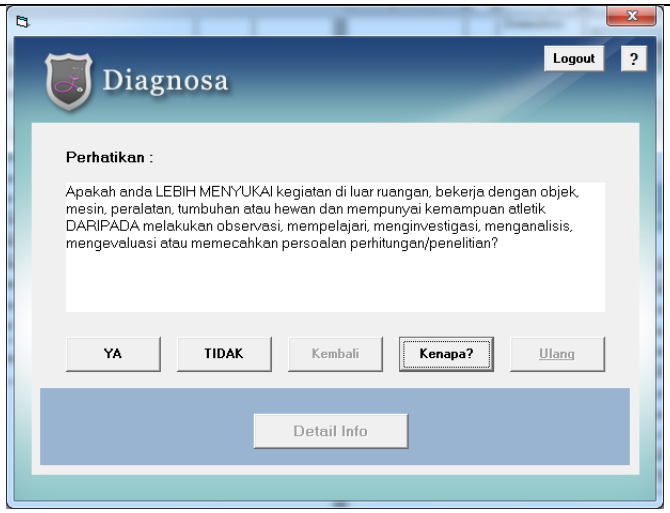
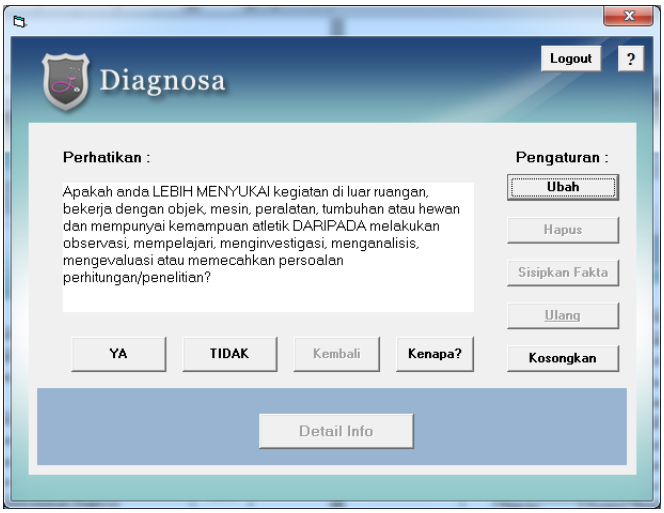
	Screenshoot	  kode pernyataan keterangan ya tidak jenis back detail 1 Apakah anda LEBIH MENYUKAI kegiatan di luar ruanga... Kami berusaha mengetahui bakat dan minat anda deng... 2 241 P 0 2 Apakah anda LEBIH MENYUKAI kegiatan di luar ruangan, bekerja dengan objek, mesin, peralatan, tumbuhan Berdasarkan jawaban anda pada pertanyaan sebelumny... 3 154 P 1 3 Berdasarkan jawaban anda pada pertanyaan sebelumny... 4 119 P 2 4 Berdasarkan jawaban anda pada pertanyaan sebelumny... 5 100 P 3 5 Berdasarkan jawaban anda pada pertanyaan sebelumny... 6 85 P 4 6 Berdasarkan jawaban anda pada pertanyaan sebelumny... 7 46 P 5 7 Berdasarkan jawaban anda pada 8 31 P 6 Apakah anda LEBIH MENYUKAI kegiatan di luar ruangan, bekerja dengan objek, mesin, peralatan, tumbuhan atau hewan dan mempunyai kemampuan atletik DARIPADA bekerja di situasi tak terduga dengan menggunakan imajinasi dan kreativitas atau berjiwa seni, berinovasi atau kemampuan intuitif (memahami sesuatu tanpa melalui penalaran rasional)? 	
10.	Nama tes	*PrimaryFunctionality*2	**Lolos**
	Tujuan	Menguji apakah pernyataan yang ditampilkan pada halaman Diagnosa berganti dengan pernyataan selanjutnya untuk jawaban tidak (sesuai kode *rulebase* untuk tidak pada pernyataan tersebut) jika memilih	

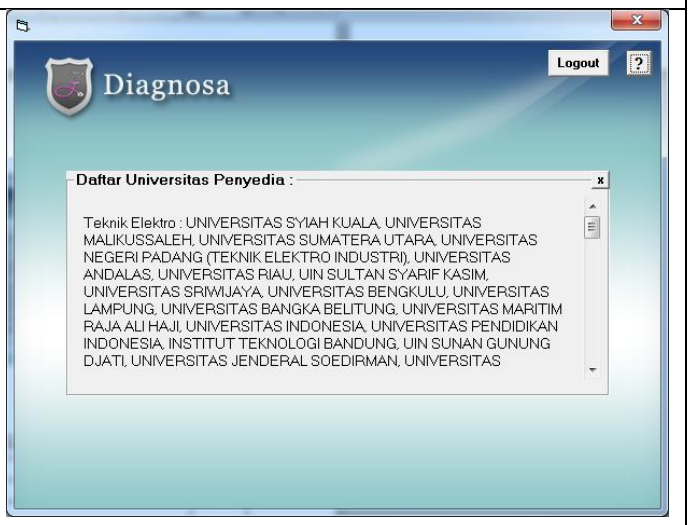
		tombol “TIDAK”	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Diagnosa *Admin* atau Diagnosa *User*.	
	Data tes	Tombol “TIDAK” pada halaman Diagnosa	
	Langkah	Klik tombol “TIDAK” pada halaman Diagnosa	
	Hasil yang diharapkan	- Proses jawaban tidak dijalankan - Muncul pernyataan selanjutnya untuk jawaban tidak pada pertanyaan tersebut - Tombol “Kembali” , “Ulang”, “Hapus”, dan “Sisipkan Fakta” muncul	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

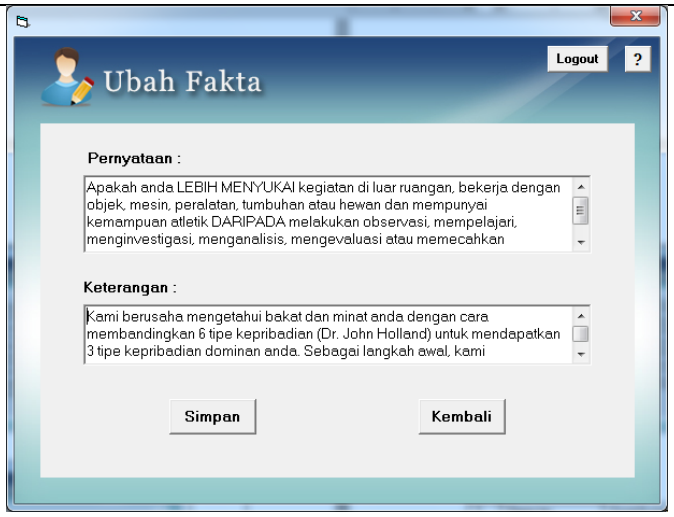
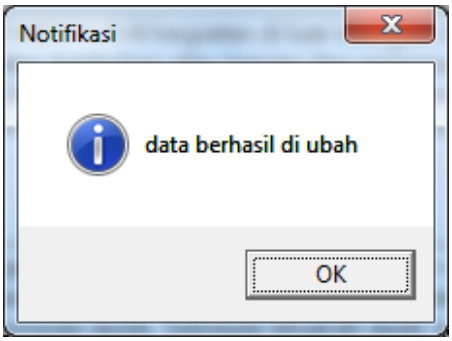
<table>
<tr><td></td><td></td><td> 
</td><td></td></tr>
<tr><td rowspan="8">11.</td><td>Nama tes</td><td>PrimaryFunctionality3</td><td rowspan="8">Lolos</td></tr>
<tr><td>Tujuan</td><td>Menguji apakah pernyataan yang ditampilkan pada halaman Diagnosa berganti pernyataan sebelumnya (sesuai kode rulebase untuk back pada pernyataan tersebut) jika memilih tombol “Kembali”</td></tr>
<tr><td>Asumsi</td><td>Aplikasi sudah berjalan dan terhubung dengan file database yang benar.
Aplikasi sudah masuk pada halaman Diagnosa Admin atau Diagnosa User.</td></tr>
<tr><td>Data tes</td><td>Tombol “Kembali” pada halaman Diagnosa</td></tr>
<tr><td>Langkah</td><td>Klik tombol “Kembali” pada halaman Diagnosa</td></tr>
<tr><td>Hasil yang diharapkan</td><td>- Proses back dijalankan
- Muncul pernyataan sebelumnya
- Muncul notifikasi “Anda saat ini berada di awal data!” ketika pernyataan yang dituju adalah awal data</td></tr>
<tr><td>Hasil pengamatan</td><td>Sesuai dengan harapan</td></tr>
<tr><td>Screenshoot</td><td> 
</td></tr>
</table>

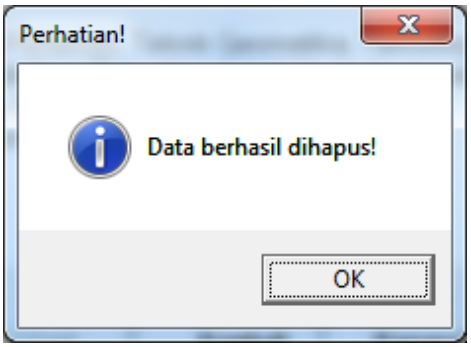
		 	
12.	Nama tes	*PrimaryFunctionality*4	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan penjelasan jika memilih tombol “Kenapa?” pada halaman Diagnosa	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Diagnosa *Admin* atau Diagnosa *User*	
	Data tes	Tombol “Kenapa?” pada halaman Diagnosa	
	Langkah	Klik tombol “Kenapa?” pada halaman Diagnosa	
	Hasil yang diharapkan	- Proses keterangan dijalankan - Muncul *frame* penjelasan mengenai pernyataan tersebut - Apabila memilih tombol “x” pada *frame* penjelasan maka akan kembali ke halaman Diagnosa	
	Hasil pengamatan	Sesuai dengan harapan	

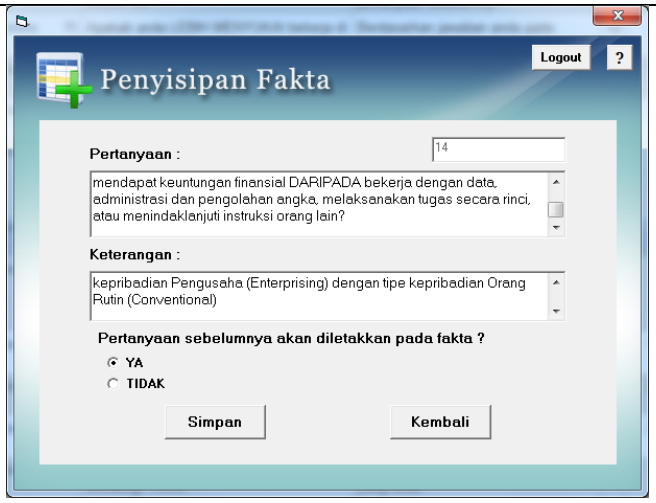
	Screenshoot	 	
13.	Nama tes	*PrimaryFunctionality*5	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat kembali ke awal proses Diagnosa jika memilih tombol “Ulang” pada halaman Diagnosa	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Diagnosa *Admin* atau Diagnosa *User*	
	Data tes	Tombol “Ulang” pada halaman Diagnosa	
	Langkah	Klik tombol “Ulang” pada halaman Diagnosa	
	Hasil yang diharapkan	- Proses ulang diagnosa dijalankan - Muncul pernyataan awal dari proses diagnosa - Tombol “Kembali”, “Ulang”, “Hapus”, dan “Sisipkan Fakta” hilang	
	Hasil pengamatan	Sesuai dengan harapan	

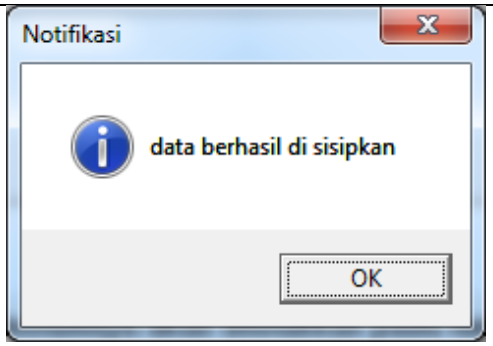
	Screenshoot	 	
14.	Nama tes	*PrimaryFunctionality*6	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan *frame* daftar universitas penyedia dengan data detail dari tabel *rule* pada *database* jika memilih tombol “Detail Info” pada halaman Diagnosa	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Diagnosa *Admin* atau Diagnosa *User*	
	Data tes	Tombol “Detail Info” pada halaman Diagnosa	
	Langkah	Klik tombol “Detail Info” pada halaman Diagnosa	
	Hasil yang	- Proses detail info dijalankan	

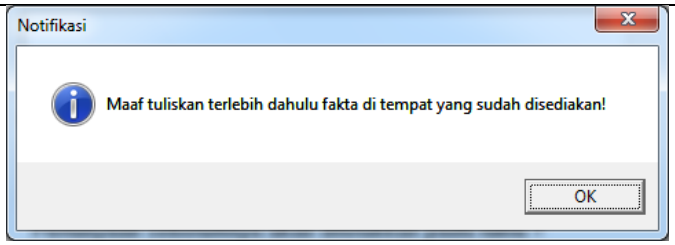
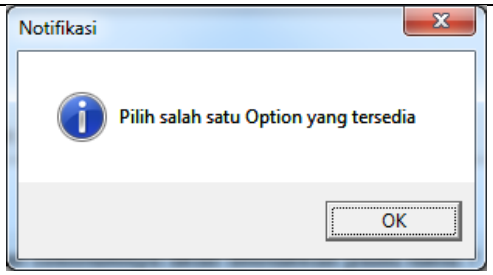
	diharapkan	- Muncul frame daftar universitas penyedia mengenai pernyataan tersebut - Apabila memilih tombol “x” pada *frame* daftar universitas penyedia maka akan kembali ke halaman Diagnosa	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
15.	Nama tes	*PrimaryFunctionalityAdmin*1	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat mengubah data *rulebase* yang sudah ada dalam *database*	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Ubah Fakta dengan cara klik tombol “Ubah” pada halaman Diagnosa *Admin* saat pernyataan yang ingin diubah muncul	
	Data tes	Tombol “Simpan” dengan pernyataan dan keterangan yang diisi dengan benar	
	Langkah	Mengisi *textbox* pernyataan dan keterangan kemudian klik tombol “Simpan”	
	Hasil yang diharapkan	- Proses ubah fakta dijalankan - Muncul notifikasi “Data berhasil diubah” - Muncul halaman Diagnosa Admin dengan pernyataan yang telah berhasil diubah jika klik	

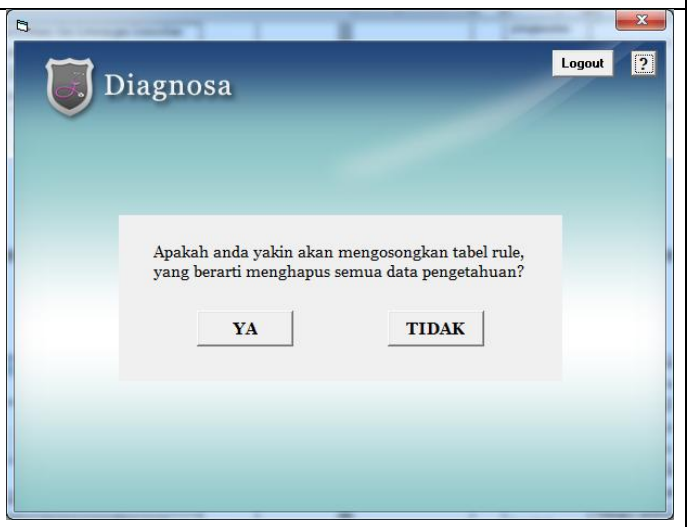
		tombol OK pada notifikasi “Data berhasil diubah” - Halaman Ubah Fakta akan tertutup dan kembali ke halaman Diagnosa *Admin* dengan pernyataan yang belum jadi diubah apabila klik tombol “Kembali” pada halaman Ubah Fakta	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot	 	
16.	Nama tes	*PrimaryFunctionalityAdmin*2	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menghapus data *rulebase* yang sudah ada dalam *database*	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Diagnosa *Admin*	
	Data tes	Tombol “Hapus” pada halaman Diagnosa *Admin*	
	Langkah	Klik tombol “Hapus” pada halaman Diagnosa	

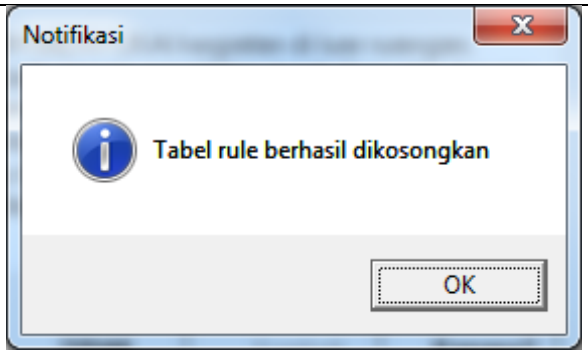
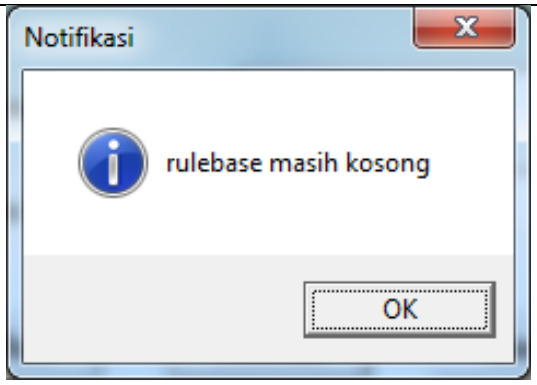
		*Admin*ketika pernyataan yang ingin dihapus muncul	
	Hasil yang diharapkan	- Muncul Notifikai “Apakah anda yakin akan menghapus fakta tersebut beserta fakta-fakta lain yang berada di bawahnya?” - Klik “YA” proses hapus fakta dijalankan - Muncul notifikasi “Data berhasil dihapus” - Halaman Diagnosa *Admin* akan menampilkan pernyataan sebelum pernyataan yang telah dihapus - Klik “TIDAK” muncul halaman Diagnosa *Admin* dengan data yang tidak jadi dihapus	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot	 	
17.	Nama tes	*PrimaryFunctionalityAdmin*3	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menyisipkan fakta diantara data *rulebase* yang sudah ada dalam *database*	

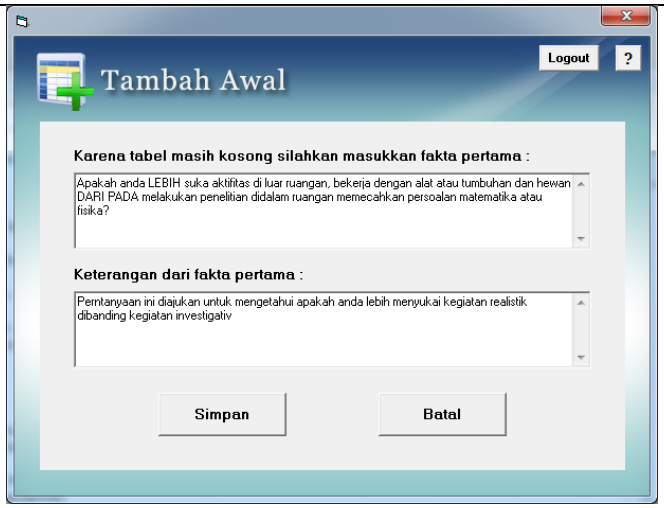
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Sisipkan Fakta dengan cara klik tombol “Sisipkan Fakta” pada halaman Diagnosa *Admin* saat pernyataan yang muncul akan diletakkan setelah fakta sisipan	
	Data tes	Tombol “Simpan” dengan pernyataan dan keterangan yang diisi dengan benar serta salah satu pilihan letak pernyataan setelahnya	
	Langkah	Mengisi *textbox* pernyataan dan keterangan, memilih letak pernyataan setelahnya kemudian klik tombol “Simpan”	
	Hasil yang diharapkan	- Proses sisipkan fakta dijalankan - Muncul notifikasi “Data berhasil disisipkan” - Muncul halaman Diagnosa Admin dengan pernyataan yang telah berhasil disisipkan jika klik tombol OK pada notifikasi “Data berhasil disisipkan” - Halaman Sisipkan Fakta akan tertutup dan kembali ke halaman Diagnosa *Admin* dengan data yang tidak jadi disisipi apabila klik tombol “Kembali” pada halaman Ubah Fakta	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

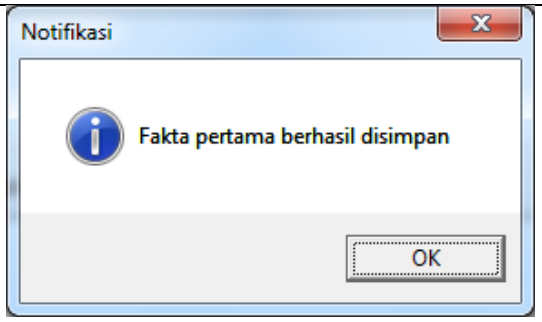
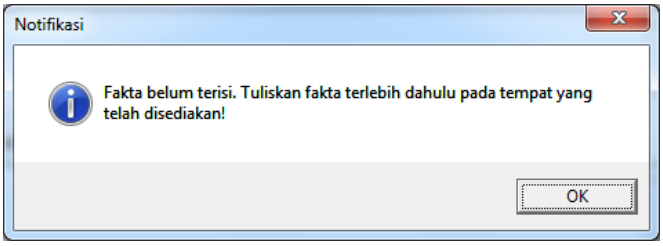
			
18.	Nama tes	*PrimaryFunctionalityAdmin*4	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan pesan peringatan jika pernyataan pada halaman Penyisipan Fakta tidak diisi	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Sisipkan Fakta dengan cara klik tombol "Sisipkan Fakta" pada halaman Diagnosa *Admin* saat pernyataan yang muncul akan diletakkan setelah fakta sisipan	
	Data tes	Tombol "Simpan" dengan pernyataan yang tidak diisi	
	Langkah	Mengisi *textbox* keterangan, memilih letak pernyataan setelahnya kemudian klik tombol "Simpan"	
	Hasil yang diharapkan	- Proses sisipkan fakta dijalankan - Muncul notifikasi "Maaf tuliskan terlebih dahulu fakta di tempat yang sudah disediakan!" - Aplikasi akan kembali ke halaman Sisipkan Fakta jika klik tombol OK pada notifikasi "Maaf tuliskan terlebih dahulu fakta di tempat yang sudah disediakan!"	
	Hasil pengamatan	Sesuai dengan harapan	

	Screenshoot		
19.	Nama tes	*PrimaryFunctionalityAdmin*5	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan pesan peringatan jika pilihan letak fakta setelahnya pada halaman Penyisipan Fakta tidak dipilih	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Sisipkan Fakta dengan cara klik tombol "Sisipkan Fakta" pada halaman Diagnosa *Admin* saat pernyataan yang muncul akan diletakkan setelah fakta sisipan	
	Data tes	Tombol "Simpan" dengan pernyataan dan keterangan yang diisi dengan benar tanpa memilih salah satu pilihan letak pernyataan setelahnya	
	Langkah	Mengisi *textbox* pernyataan dan keterangan kemudian klik tombol "Simpan"	
	Hasil yang diharapkan	- Proses sisipkan fakta dijalankan - Muncul notifikasi "Pilih salah satu Option yang tersedia" - Aplikasi akan kembali ke halaman Sisipkan Fakta jika klik tombol OK pada notifikasi "Pilih salah satu Option yang tersedia"	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

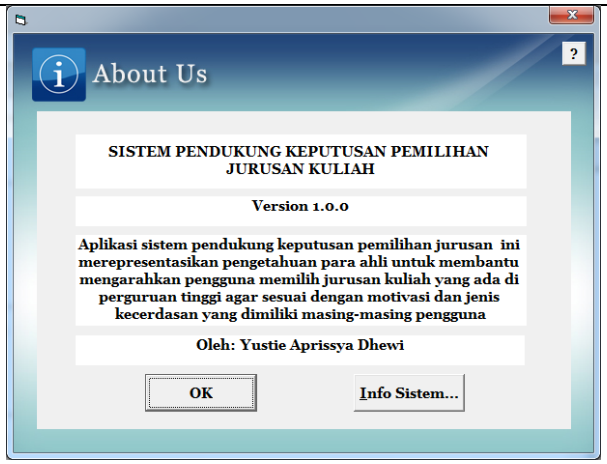
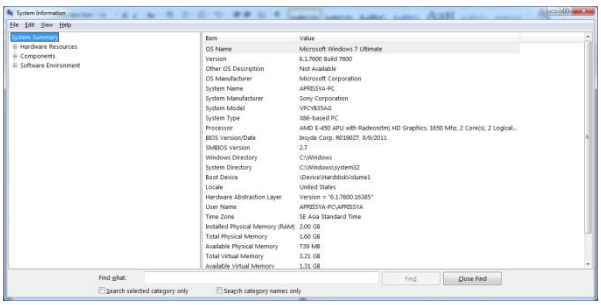
20.	Nama tes	*PrimaryFunctionalityAdmin*6	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menghapus semua data *rulebase* yang sudah ada dalam *database*	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Diagnosa *Admin*	
	Data tes	Tombol "Kosongkan" pada halaman Diagnosa *Admin*	
	Langkah	Klik tombol "Kosongkan" pada halaman Diagnosa *Admin*	
	Hasil yang diharapkan	- Muncul notifikasi "Apakah anda yakin akan mengosongkan tabel rule, yang berarti menghapus semua data pengetahuan? - Klik "YA" proses kosongkan *rulebase* dijalankan - Muncul notifikasi "Tabel rule berhasil dikosongkan" - Halaman Diagnosa *Admin* akan tertutup dan menampilkan halaman *Login* - Klik "TIDAK" maka muncul halaman Diagnosa *Admin*dengan pernyataan yang muncul sebelumnya	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

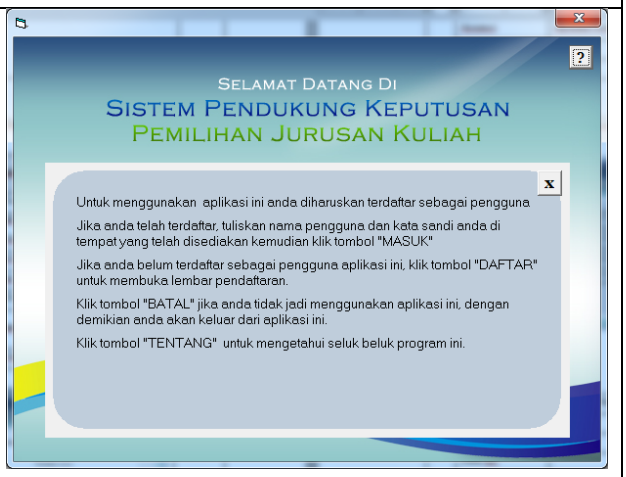
			
21.	Nama tes	*PrimaryFunctionality7*	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan pesan peringatan apabila data *rulebase* pada *database* masih kosong (tidak ada data sama sekali)	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Pengguna telah berhasil *signup* atau *login*	
	Data tes	Tombol "OK" pada notifikasi "Pendaftaran berhasil " atau notifikasi "Login berhasil" dengan *rulebase* kosong	
	Langkah	Melakukan *signup* atau *login* kemudian klik tombol "OK" pada notifikasi "Pendaftaran berhasil " atau notifikasi "Login berhasil"	
	Hasil yang diharapkan	- Halaman *Login* maupun halaman *Signup* tertutup - Muncul notifikasi "rulebase masih kosong"	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

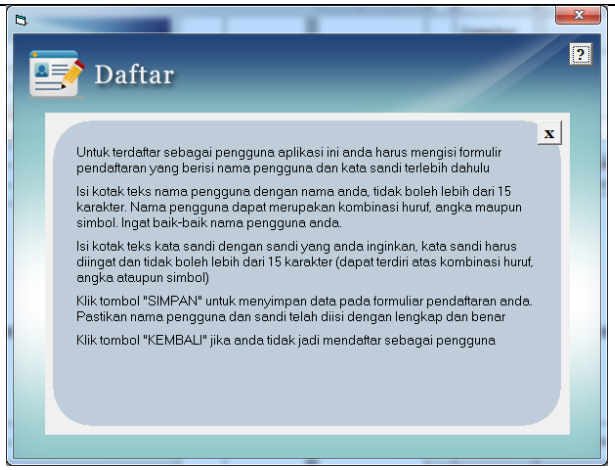
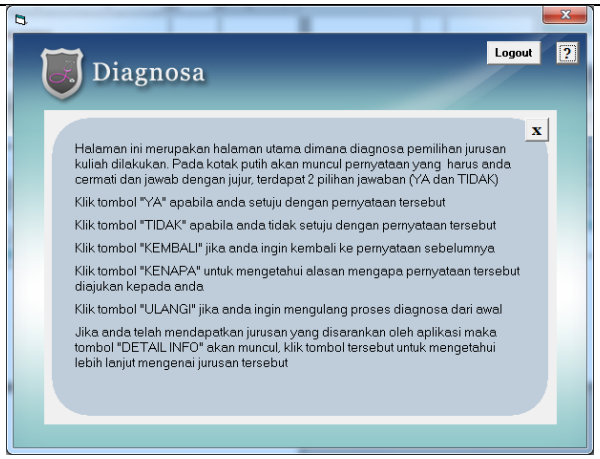
22.	Nama tes	*PrimaryFunctionalityAdmin*7	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat memasukkan data pertama dan menyimpannya dalam tabel *rule* di *database* jika belum ada data *rulebase* sama sekali	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Pengguna telah berhasil *login* sebagai admin	
	Data tes	Tombol "Simpan" dengan pernyataan dan keterangan yang disii dengan benar	
	Langkah	Mengisi *textbox* pernyataan dan keterangan kemudian klik tombol "Simpan"	
	Hasil yang diharapkan	- Proses tambah awal dijalankan - Muncul notifikasi "Fakta pertama berhasil disimpan" - Halaman Diagnosa *Admin* akan muncul apabila memilih tombol "OK" pada notifikasi "Fakta pertama berhasil disimpan"	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

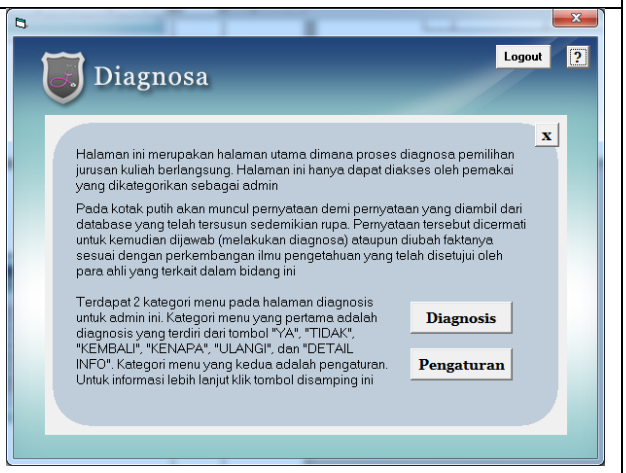
			
23.	Nama tes	*PrimaryFunctionalityAdmin*8	**Lolos**
	Tujuan	Menguji apakah aplikasi dapat menampilkan pesan peringatan jika pernyataan pada halaman Tambah Awal tidak diisi	
	Asumsi	Aplikasi sudah berjalan dan terhubung dengan *file database* yang benar. Aplikasi sudah masuk pada halaman Tambah Awal	
	Data tes	Tombol "Simpan" dengan pernyataan yang tidak disii	
	Langkah	Mengisi *textbox* keterangan kemudian klik tombol "Simpan"	
	Hasil yang diharapkan	- Proses tambah awal dijalankan - Muncul notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan" - Aplikasi akan kembali pada halaman Tambah Awal apabila memilih tombol "OK" pada notifikasi "Fakta belum terisi. Tuliskan fakta terlebih dahulu pada tempat yang telah disediakan"	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

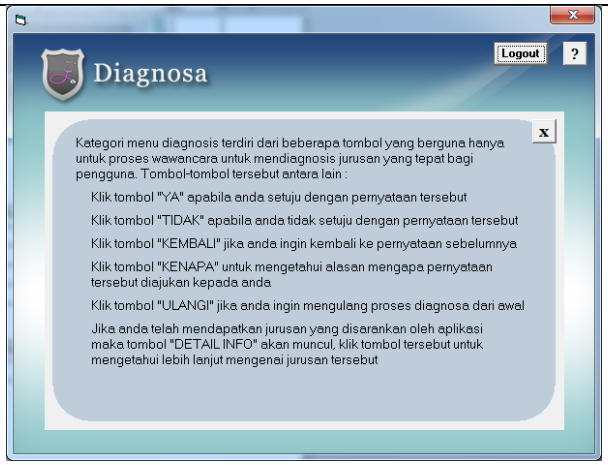
Tabel 20. *Test Case* Pengujian Faktor Kualitas *Functionality Contributing*

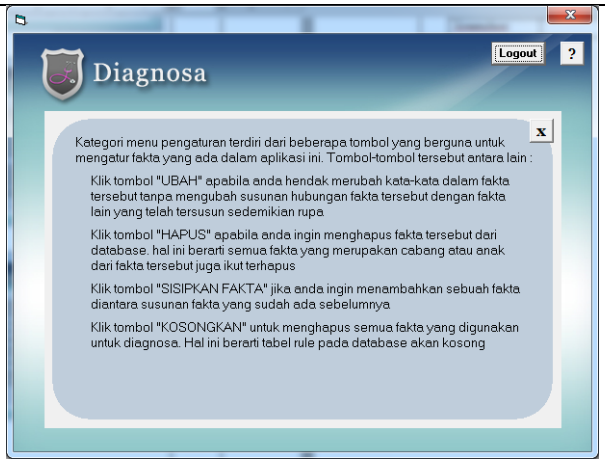
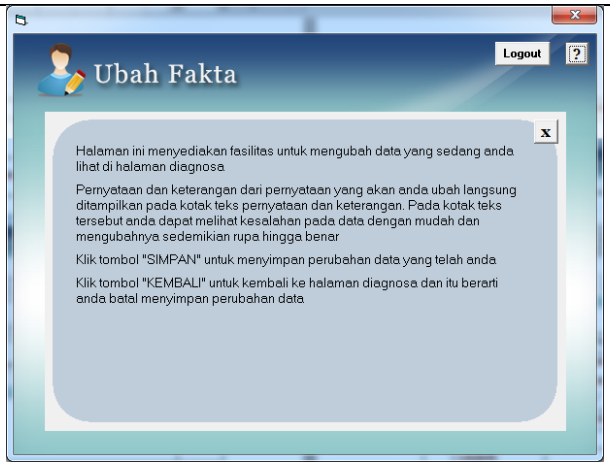
No.	Pengujian		Lolos / Gagal
1.	Nama tes	*AboutTest*	Lolos
	Tujuan	Menguji apakah tombol tentang telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol "Tentang"	
	Langkah	Halaman *Login* -> klik tombol "Tentang"	
	Hasil yang diharapkan	- Muncul halaman *About Us* - Muncul jendela Info Sistem jika klik tombol "Info Sistem…" pada halamn About Us - Halaman *About Us* akan tertutup dan kembali ke halaman *Login* apabila klik tombol "OK" pada halaman *About Us*	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot	 	

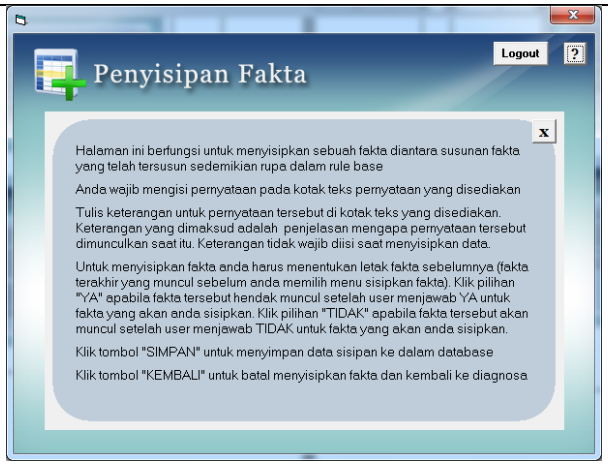
2.	Nama tes	*HelpLoginTest*	**Lolos**
	Tujuan	Menguji apakah tombol “?” pada halaman *Login* untuk melihat panduan penggunaan halaman tersebut telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “?” pada halaman *Login*	
	Langkah	Halaman *Login* -> klik tombol “?”	
	Hasil yang diharapkan	- Muncul halaman *HelpLogin* - Halaman *HelpLogin* akan tertutup dan kembali ke halaman *Login* apabila klik tombol “x” pada halaman *HelpLogin*	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
3.	Nama tes	*HelpSignupTest*	**Lolos**
	Tujuan	Menguji apakah tombol “?” pada halaman *Signup* untuk melihat panduan penggunaan halaman tersebut telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “?” pada halaman *Signup*	
	Langkah	Halaman *Signup* -> klik tombol “?”	
	Hasil yang diharapkan	- Muncul halaman *HelpSignup* - Halaman *HelpSignup* akan tertutup dan kembali ke halaman *Signup* apabila klik tombol “x” pada halaman *HelpSignup*	
	Hasil pengamatan	Sesuai dengan harapan	

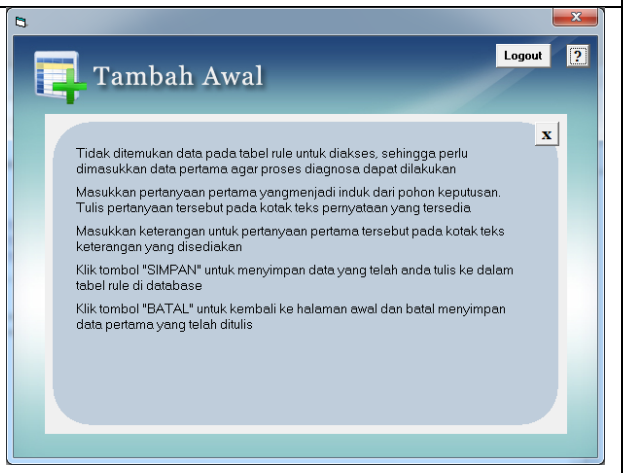
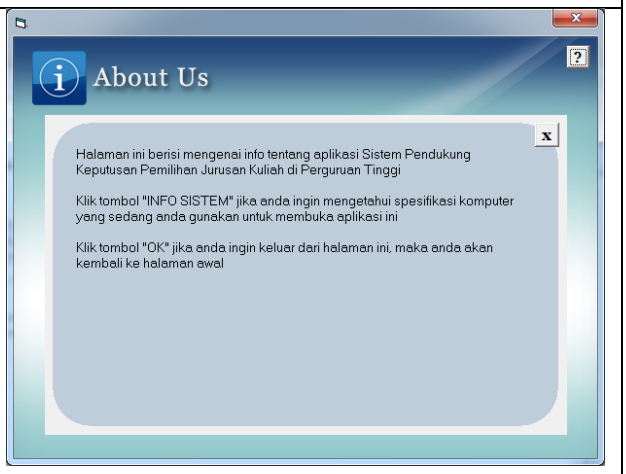
	Screenshoot		
4.	Nama tes	*Help*Diagnosa*UserTest*	**Lolos**
	Tujuan	Menguji apakah tombol “?” pada halaman Diagnosa *User* untuk melihat panduan penggunaan halaman tersebut telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “?” pada halamanDiagnosa *User*	
	Langkah	Halaman Diagnosa *User* -> klik tombol “?”	
	Hasil yang diharapkan	- Muncul halaman *Help* Diagnosa *User* - Halaman *Help* Diagnosa *User* akan tertutup dan kembali ke halaman Diagnosa *User* apabila klik tombol “x” pada halaman *Help* Diagnosa *User*	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

5.	Nama tes	*Help*Diagnosa*AdminTest*1	**Lolos**
	Tujuan	Menguji apakah tombol “?” pada halaman Diagnosa *Admin* untuk melihat panduan penggunaan halaman tersebut telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “?” pada halamanDiagnosa *Admin*	
	Langkah	Halaman Diagnosa *Admin*-> klik tombol “?”	
	Hasil yang diharapkan	- Muncul halaman *Help* Diagnosa *Admin* - Halaman *Help* Diagnosa *Admin* akan tertutup dan kembali ke halaman Diagnosa *Admin* apabila klik tombol “x” pada halaman *Help* Diagnosa *Admin*	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
6.	Nama tes	*Help*Diagnosa*AdminTest*2	**Lolos**
	Tujuan	Menguji apakah tombol “Diagnosis” pada halaman *Help* Diagnosa *Admin* untuk melihat panduan penggunaan fungsi diagnosis pada halaman Diagnosa *Admin* telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “Diagnosis” pada halaman *Help* Diagnosa *Admin*	
	Langkah	Halaman *Help* Diagnosa *Admin* -> klik tombol	

		"Diagnosis"	
	Hasil yang diharapkan	- Muncul halaman *Help* Diagnosa *Admin* Diagnosis - Halaman *Help* Diagnosa *Admin* Diagnosis akan tertutup dan kembali ke halaman *Help* Diagnosa *Admin* apabila klik tombol "x" pada halaman *Help* Diagnosa *Admin* Diagnosis	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
7.	Nama tes	*Help*Diagnosa*AdminTest*3	**Lolos**
	Tujuan	Menguji apakah tombol "Pengaturan" pada halaman *Help* Diagnosa *Admin* untuk melihat panduan penggunaan fungsi pengaturan pada halaman Diagnosa *Admin* telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol "Pengaturan" pada halaman *Help* Diagnosa *Admin*	
	Langkah	Halaman *Help* Diagnosa *Admin* -> klik tombol "Pengaturan"	
	Hasil yang diharapkan	- Muncul halaman *Help* Diagnosa *Admin* Pengaturan - Halaman *Help* Diagnosa *Admin* Pengaturan akan tertutup dan kembali ke halaman *Help* Diagnosa *Admin* apabila klik tombol "x" pada	

		halaman *Help* Diagnosa *Admin* Pengaturan	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
8.	Nama tes	*Help*UbahFakta*Test*	**Lolos**
	Tujuan	Menguji apakah tombol “?” pada halaman Ubah Fakta untuk melihat panduan penggunaan halaman tersebut telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “?” pada halaman Ubah Fakta	
	Langkah	Halaman Ubah Fakta -> klik tombol “?”	
	Hasil yang diharapkan	- Muncul halaman *Help* Ubah Fakta - Halaman *Help* Ubah Fakta akan tertutup dan kembali ke halaman Ubah Fakta apabila klik tombol “x” pada halaman *Help* Ubah Fakta	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

9.	Nama tes	*Help*SisipkanFakta*Test*	**Lolos**
	Tujuan	Menguji apakah tombol “?” pada halaman Sisipkan Fakta untuk melihat panduan penggunaan halaman tersebut telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “?” pada halaman Sisipkan Fakta	
	Langkah	Halaman Sisipkan Fakta -> klik tombol “?”	
	Hasil yang diharapkan	- Muncul halaman *Help* Sisipkan Fakta - Halaman *Help* Sisipkan Fakta akan tertutup dan kembali ke halaman Sisipkan Fakta apabila klik tombol “x” pada halaman *Help* Sisipkan Fakta	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
10.	Nama tes	*Help*TambahAwal*Test*	**Lolos**
	Tujuan	Menguji apakah tombol “?” pada halaman Tambah Awal untuk melihat panduan penggunaan halaman tersebut telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “?” pada halaman Tambah Awal	
	Langkah	Halaman Tambah Awal -> klik tombol “?”	
	Hasil yang diharapkan	- Muncul halaman *Help* Tambah Awal - Halaman *Help* Tambah Awal akan tertutup dan kembali ke halaman Tambah Awal apabila klik tombol “x” pada halaman *Help* Tambah Awal	

	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		
11.	Nama tes	*HelpAboutTest*	**Lolos**
	Tujuan	Menguji apakah tombol “?” pada halaman *About* untuk melihat panduan penggunaan halaman tersebut telah berjalan sesuai dengan fungsinya?	
	Asumsi	Aplikasi sudah berjalan	
	Data tes	Tombol “?” pada halaman *About*	
	Langkah	Halaman *About* -> klik tombol “?”	
	Hasil yang diharapkan	- Muncul halaman *HelpAbout* - Halaman *HelpAbout* akan tertutup dan kembali ke halaman *About* apabila klik tombol “x” pada halaman *HelpAbout*	
	Hasil pengamatan	Sesuai dengan harapan	
	Screenshoot		

Lampiran 6. Surat Keputusan Pengangkatan Pembimbing

KEPUTUSAN DEKAN
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
NOMOR : 162/ELK/Q-I/VIII/2013
TENTANG
PENGANGKATA, N PEMBIMBING TUGAS AKHIR SKRIPSI
BAGI MAHASISWA FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA

Menimbang :
1. Bahwa sehubungan dengan telah dipenuhi syarat untuk penulisan Tugas Akhir Skripsi bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta, perlu diangkat pembimbing.
2. Bahwa untuk keperluan dimaksud perlu ditetapkan dengan Keputusan Dekan.

Mengingat :
1. Undang-undang Nomor 20 tahun 2003.
2. Peraturan Pemerintah RI Nomor 60 tahun 1999.
3. Keputusan Presiden RI: a. Nomor 93 tahun 1999; b. 305/M tahun 1999.
4. Keputusan Menteri Pendidikan dan Kebudayaan RI: Nomor 274/O/1999.
5. Keputusan Mendiknas RI Nomor 003/O/2001.
6. Keputusan Rektor UNY Nomor : 1160/UN34/KP/2011

M E M U T U S K A N

Menetapkan

Pertama : Mengangkat Pembimbing Tugas Akhir Skripsi bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta sebagai berikut :

Nama Pembimbing	: Handaru Jati, Ph.D
Bagi mahasiswa	:
Nama/No.Mahasiswa	: Yustie Aprissya Dhewi / 09520241017
Jurusan/ Prodi	: Pendidikan Teknik Elektronika / Pendidikan Teknik Informatika
Judul Skripsi	: *Pengembangan dan Analisis Kualitas aplikasi Sistem Pendukung Keputusan Pemilihan Jurusan Kuliah Di Perguruan Tinggi*

Kedua : Dosen pembimbing diserahi tugas membimbing penulisan Tugas Akhir Skripsi sesuai dengan Pedoman Tugas Akhir Skripsi.

Ketiga : Keputusan ini berlaku sejak ditetapkan

Keempat : Segala sesuatu akan diubah dan dibetulkan sebagaimana mestinya apabila di kemudian hari ternyata terdapat kekeliruan dalam Keputusan ini.

Ditetapkan : di Yogyakarta
Pada tanggal : 28 Agustus 2013
Dekan

Dr. Moch. Bruri Triyono
NIP. 19560216 198603 1 003

Tembusan Yth :
1. Wakil Dekan II FT UNY
2. Ketua Jurusan Pendidikan Teknik Elektronika
3. Kasub. Bag Pendidikan FT UNY
4. Yang bersangkutan

Lampiran 7. Surat Keterangan Ijin Penelitian

website : http://ft.uny.ac.id e-mail: ft@uny.ac.id ; teknik@uny.ac.id
Certificate No. QSC 00592

Nomor : 2375/UN34.15/PL/2013 01 Oktober 2013
Lamp. : 1 (satu) bendel
Hal : Permohonan Ijin Penelitian

Yth.

1. Gubernur Provinsi DIY c.q. Ka. Biro Administrasi Pembangunan Setda Provinsi DIY
2. Bupati Kulon Progo c.q. Kepala Bappeda Kabupaten Kulon Progo
3. Kepala Dinas Pendidikan, Pemuda dan Olahraga Propinsi DIY
4. Kepala Dinas Pendidikan Kabupaten Kulon Progo
5. Kepala / Direktur/ Pimpinan : SMA Negeri 2 Wates

Dalam rangka pelaksanaan Tugas Akhir Skripsi kami mohon dengan hormat bantuan Saudara memberikan ijin untuk melaksanakan penelitian dengan judul **"PENGEMBANGAN DAN ANALISIS KUALITAS APLIKASI SISTEM PENDUKUNG KEPUTUSANPEMILIHAN JURUSAN DI PERGURUAN TINGGI",** bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta tersebut di bawah ini:

No.	Nama	NIM	Jurusan/Prodi	Lokasi Penelitian
1	Yustie Aprissya D	09520241017	Pend. Teknik Informatika - S1	SMA NEGERI 2 WATES

Dosen Pembimbing/Dosen Pengampu : Handaru Jati, Ph.D.
NIP : 19740511 199903 1 002

Adapun pelaksanaan penelitian dilakukan mulai tanggal 01 Oktober 2013sampai dengan selesai.

Demikian permohonan ini, atas bantuan dan kerjasama yang baik selama ini, kami mengucapkan terima kasih.

Dekan,
Wakil Dekan I,