

**PENGEMBANGAN APLIKASI ALURADMI SEBAGAI INFORMASI ALUR  
ADMINISTRASI MAHASISWA BERBASIS ANDROID  
DI FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

**TUGAS AKHIR SKRIPSI**

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



Oleh:

Banu Kusuma

NIM 12520244039

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
2017**

## LEMBAR PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

**PENGEMBANGAN APLIKASI ALURADMI SEBAGAI INFORMASI ALUR  
ADMINISTRASI MAHASISWA BERBASIS ANDROID  
DI FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

Disusun oleh:

Banu Kusuma

NIM 12520244039

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

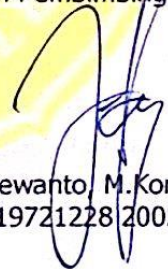
Yogyakarta, 9 Juni 2017

Mengetahui,  
Ketua Program Studi  
Pendidikan Teknik Informatika,



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

Disetujui,  
Dosen Pembimbing,



Adi Dewanto, M.Kom.  
NIP. 19721228 200501 1 001



## **SURAT PERNYATAAN**

Yang bertanda tangan di bawah ini:

Nama : Banu Kusuma

NIM : 12520244039

Program Studi : Pendidikan Teknik Informatika

Judul TAS : Pengembangan Aplikasi Aluradmi sebagai Informasi Alur  
Administrasi Mahasiswa Berbasis Android di Fakultas  
Teknik Universitas Negeri Yogyakarta

menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Yogyakarta, 9 Juni 2017

Yang menyatakan,



Banu Kusuma

NIM 12520244039

## HALAMAN PENGESAHAN

Tugas Akhir Skripsi

### PENGEMBANGAN APLIKASI ALURADMI SEBAGAI INFORMASI ALUR ADMINISTRASI MAHASISWA BERBASIS ANDROID DI FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

Disusun oleh:

Banu Kusuma

NIM 12520244039

Telah Dipertahankan di Depan Tim Penguji Tugas Akhir Tugas Akhir Skripsi  
Program Studi Pendidikan Informatika Fakultas Teknik Universitas Negeri  
Yogyakarta pada Tanggal 21 Juni 2017

#### TIM PENGUJI

Nama/Jabatan

Tanda Tangan


Tanggal

Adi Dewanto, M.Kom  
Ketua Penguji/Pembimbing



14/7/2017

Ponco Wali Pranoto, M.Pd  
Sekretaris



11/7-2017

Djoko Santoso, M.Pd  
Penguji Utama

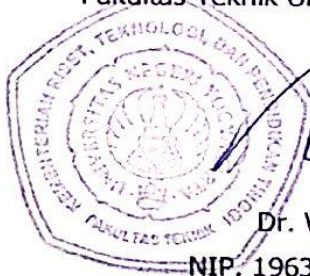


11/7-2017

Yogyakarta, Juli 2017

Fakultas Teknik Universitas Negeri Yogyakarta

Dekan,



Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001



## **MOTTO**

Sesungguhnya bersama kesulitan itu ada kemudahan

(Al-Insyiraah : 6)

Tidak apa-apa jika kita ingin merayakan kesuksesan, tetapi akan lebih penting lagi untuk mengambil pelajaran dari kegagalan

(Bill Gates)

Tidak ada yang mudah, tetapi tidak ada yang tidak mungkin

(Napoleon Bonaparte)

## **HALAMAN PERSEMBAHAN**

Dengan mengucapkan syukur Alhamdulillah kehadiran Allah SWT, kupersembahkan Tugas Akhir Skripsi ini untuk:

1. Kedua orang tuaku, Bapak Setiyono dan Ibu Indarti tercinta yang selalu memberikan kasih sayang, dukungan, semangat, pengorbanan dan tidak pernah lelah untuk selalu mendoakan demi keberhasilan anaknya.
2. Kakakku, Dora Puspita yang menjadi contoh figure yang baik sebagai seorang kakak dan selalu kuat dalam menghadapi segala macam cobaan.
3. Adikku, Rossi Andy Setiawan yang selalu sabar menerima setiap kejahilanku, memberikan canda tawa, dukungan dan semangat kepadaku.
4. Nurdiana, yang selalu mengingatkan, membantu, menerima keluhan kesah, memberi dukungan dan semangat untuk secepatnya menyelesaikan skripsi.
5. Teman-teman SPP Production, Yusta, Anno, Irfan, Hizba, Helmi, Laras yang memberikan pengalaman baru mengenai dunia perfilman dalam produksi serial "LOS(T)MEN" disela-sela mengerjakan skripsi.
6. Teman-temanku yang telah membantu dan memberikan saran dalam mengerjakan skripsi ini, Tia, Mas Kholis, Mas Irfan Lutfi, David, Topik, Ofani, Rio, Said, Icha, Farah, Esti, Lathif, Deni, Mas Rizki, Leila.
7. Teman-teman seperjuangan Kelas PTI G 2012 yang telah memberikan banyak cerita dan telah berjuang bersama.
8. Dan semua pihak yang tidak bisa saya sebutkan satu per satu, terima kasih atas doa, semangat, motivasi dan dukungannya untuk menyelesaikan skripsi ini.



**PENGEMBANGAN APLIKASI ALURADMI SEBAGAI INFORMASI ALUR  
ADMINISTRASI MAHASISWA BERBASIS ANDROID  
DI FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

Oleh:

BANU KUSUMA  
NIM 12520244039

**ABSTRAK**

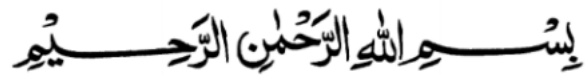
Penelitian ini bertujuan untuk: (1) mengembangkan aplikasi Aluradmi yang berisi informasi alur administrasi, tempat mengurus administrasi, dan berkas yang dibutuhkan bagi mahasiswa di Fakultas Teknik Universitas Negeri Yogyakarta; (2) mengetahui kualitas aplikasi Aluradmi berdasarkan aspek *functional suitability*, *compatibility*, *usability* dan *performance efficiency* pada ISO 25010.

Model pengembangan yang digunakan untuk mengembangkan aplikasi Aluradmi adalah model pengembangan *waterfall* yang terdiri dari analisis, desain, implementasi dan pengujian.

Hasil penelitian ini sebagai berikut: (1) aplikasi Aluradmi yang berisi informasi mengenai alur administrasi, tempat mengurus administrasi, dan berkas yang dibutuhkan bagi mahasiswa FT UNY. Informasi yang dimiliki oleh aplikasi Aluradmi tersinkronisasi dengan *server* melalui *web service* sehingga informasi yang terdapat dalam aplikasi dapat diperbarui; (2) aplikasi Aluradmi telah memenuhi standar ISO 25010 dengan hasil pengujian *functional suitability* dinyatakan sangat layak dikarenakan 100% fungsi yang terdapat dalam aplikasi berjalan dengan baik. Pada pengujian *usability* didapatkan nilai sebesar 82,2% dan dinyatakan sangat layak. Pada pengujian *compatibility*, aplikasi Aluradmi dapat diinstall dan dijalankan pada seluruh perangkat pengujian sehingga dapat dinyatakan sangat layak. Pengujian *performance efficiency* dalam hal penggunaan *resource*, aplikasi Aluradmi rata-rata memelurkan *resource CPU* sebesar 38,4% dan *Memory* sebesar 22,36 MB sehingga dinyatakan layak dikarenakan tidak terlalu memerlukan *resource* yang besar. Pengujian *performance efficiency* dalam hal waktu respon, aplikasi Aluradmi memerlukan rata-rata waktu sebesar 4,76 detik dan dinyatakan dengan predikat "Puas".

Kata Kunci: alur administrasi, Android, ISO 25010

## KATA PENGANTAR



Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir Skripsi ini dengan judul "Pengembangan Aplikasi Aluradmi sebagai Informasi Alur Administrasi Mahasiswa Berbasis Android di Fakultas Teknik Universitas Negeri Yogyakarta" dalam rangka memenuhi sebagian persyaratan untuk mendapatkan gelar Sarjana Pendidikan di Universitas Negeri Yogyakarta.

Penyusunan Tugas Akhir Skripsi ini tidak lepas dari bantuan dan kerjasama dari berbagai pihak. Maka dari itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Adi Dewanto, M.Kom. selaku Dosen Pembimbing Tugas Akhir Skripsi (TAS) yang telah banyak memberikan semangat, petunjuk, masukan, dan bimbingan selama penyusunan Tugas Akhir Skripsi (TAS) ini.
2. Bapak Dr. Fatchul Arifin selaku Ketua Jurusan dan Pendidikan Teknik Elektronika dan Informatika yang telah memberikan bantuan selama proses penyusunan proposal sampai dengan selesainya TAS ini.
3. Bapak Handaru Jati, Ph.D. selaku Ketua Program Studi Pendidikan Teknik Informatika yang telah memberikan bantuan selama proses penyusunan proposal sampai dengan selesainya TAS ini.
4. Bapak Dr. Widarto, M. Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang memberikan persetujuan dalam pelaksanaan TAS ini.



5. Ibu Nur Hasanah, S.T., M.Cs., Saudara David Surya Aji S., S.Pd., dan Saudara Ofani Dariyan, S.Pd. yang telah berkenan untuk mengisi angket pengujian *functional suitability* pada penelitian TAS ini.
6. Mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta yang telah berkenan untuk mengisi angket pengujian *usability* pada penelitian TAS ini.
7. Seluruh pihak, yang telah memberikan bantuan dan kerjasamanya selama penyusunan TAS ini.

Semoga segala bantuan yang telah diberikan semua pihak di atas menjadi amalan yang bermanfaat dan mendapatkan balasan dari Allah SWT.

Penulis menyadari bahwa Tugas Akhir Skripsi (TAS) ini tidak sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun untuk kesempurnaan TAS ini. Semoga TAS ini dapat menjadi informasi bermanfaat bagi pembaca atau pihak lain yang membutuhkan.

Yogyakarta, 9 Juni 2017

Penulis,

Banu Kusuma

NIM 12520244039

## DAFTAR ISI

HALAMAN SAMPUL .....	i
LEMBAR PERSETUJUAN.....	ii
SURAT PENYATAAN .....	iii
HALAMAN PENGESAHAN .....	iv
HALAMAN MOTTO .....	v
HALAMAN PERSEMBAHAN .....	vi
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR LAMPIRAN.....	xviii

### **BAB I PENDAHULUAN**

A. Latar Belakang Masalah .....	1
B. Identifikasi Masalah .....	4
C. Batasan Masalah .....	5
D. Rumusan Masalah .....	5
E. Tujuan Penelitian .....	5
F. Spesifikasi Produk .....	6
G. Manfaat Penelitian.....	6

### **BAB II KAJIAN TEORI**

A. Deskripsi Teori .....	8
1. Sistem Informasi.....	8
2. Administrasi.....	9
3. Android .....	9
4. Aplikasi <i>Mobile</i> .....	13
5. SQLite.....	15
6. Google Maps .....	19
7. Algoritma Dijkstra .....	20
8. Haversine Formula .....	22



9. <i>Software Development Life Cycle (SDLC)</i> .....	22
10. <i>Unified Modeling Language</i> .....	25
11. Kualitas Perangkat Lunak .....	29
B. Penelitian yang Relevan .....	34
C. Kerangka Pikir .....	35
<b>BAB III METODE PENELITIAN</b>	
A. Model Pengembangan .....	38
B. Prosedur Pengembangan .....	38
1. Analisis .....	38
2. Desain .....	39
3. Pengodean .....	39
4. Pengujian .....	39
C. Subjek, Tempat dan Waktu Penelitian .....	40
D. Metode Pengumpulan Data .....	40
1. Observasi .....	40
2. Kuisisioner (Angket) .....	41
E. Instrumen Penelitian .....	41
1. Instrumen <i>Functional Suitability</i> .....	41
2. Instrumen <i>Compatibility</i> .....	42
3. Instrumen <i>Usability</i> .....	42
4. Instrumen <i>Performance Efficiency</i> .....	43
F. Teknik Analisis Data .....	43
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN</b>	
A. Hasil Penelitian .....	45
1. Analisis .....	45
2. Desain .....	47
3. Implementasi .....	55
4. Pengujian .....	71
B. Pembahasan .....	78
<b>BAB V KESIMPULAN DAN SARAN</b>	
A. Kesimpulan .....	80
B. Keterbatasan Produk .....	81

C. Pengembangan Produk Lebih Lanjut .....	81
D. Saran .....	82
<b>DAFTAR PUSTAKA .....</b>	<b>83</b>
<b>LAMPIRAN-LAMPIRAN.....</b>	<b>86</b>



## DAFTAR TABEL

Tabel 1. Instrumen <i>Functional Suitability</i> .....	41
Tabel 2. Skor Instrumen <i>Functional Suitability</i> .....	42
Tabel 3. <i>USE Questionnaire</i> .....	42
Tabel 4. Skor Instrumen <i>Usability</i> .....	43
Tabel 5. Kriteria Interpretasi Skor .....	44
Tabel 6. Kepuasan terhadap Waktu Respon .....	44
Tabel 7. Spesifikasi Perangkat.....	46
Tabel 8. Daftar <i>Sequence Diagram</i> .....	50
Tabel 9. Daftar <i>Layout</i> .....	59
Tabel 10. Daftar File Java .....	60
Tabel 11. Contoh Koordinat .....	61
Tabel 12. Hasil Konversi .....	61
Tabel 13. Contoh Titik di Dalam Grafik.....	63
Tabel 14. Ringkasan Hasil Pengujian <i>Usability</i> .....	72
Tabel 15. Penguji <i>Functional Suitability</i> .....	73
Tabel 16. Hasil Pengujian <i>Functional Suitability</i> .....	73
Tabel 17. Daftar <i>Smartphone</i> Pengujian <i>Compatibality</i> .....	74
Tabel 18. Ringkasan Hasil Pengujian <i>Compatibality</i> .....	75
Tabel 19. Spesifikasi Perangkat Pengujian.....	76
Tabel 20. Hasil Pengujian Penguunaan <i>Resource</i> .....	76
Tabel 21. Hasil Pengujian Waktu Respon .....	77
Tabel 22. Hasil Seluruh Pengujian .....	79
Tabel 23. Definisi Aktor .....	93
Tabel 24. Definisi <i>Use Case</i> .....	93
Tabel 25. Skenario Penytetelan Awal Data .....	96
Tabel 26. Skenario Memeriksa Status Data.....	97
Tabel 27. Skenario Sinkronisasi Data .....	97
Tabel 28. Skenario Mengecek Data.....	98
Tabel 29. Skenario Menambah Data .....	98
Tabel 30. Skenario Mengubah Data .....	99

Tabel 31. Skenario Menghapus Data .....	99
Tabel 32. Skenario Memilih Jurusan.....	100
Tabel 33. Skenario Memeriksa Status Jurusan .....	100
Tabel 34. Skenario Melihat Daftar Administrasi.....	101
Tabel 35. Skenario Melihat Alur Administrasi .....	101
Tabel 36. Skenario Melihat Detail Langkah Administrasi .....	102
Tabel 37. Skenario Melihat Peta .....	103
Tabel 38. Skenario Melihat Rute.....	103
Tabel 39. Skenario Melihat Beranda .....	103
Tabel 40. Skenario Melihat Bantuan .....	104
Tabel 41. Skenario Cari Langkah .....	104

## DAFTAR GAMBAR

Gambar 1. Statistik Perangkat Android.....	10
Gambar 2. Arsitektur Android.....	11
Gambar 3. Arsitektur <i>Serverless</i> SQLite.....	16
Gambar 4. Algoritma Dijkstra.....	20
Gambar 5. Diagram UML .....	27
Gambar 6. <i>Use Case Diagram</i> .....	28
Gambar 7. <i>Class Diagram</i> .....	28
Gambar 8. <i>Sequence Diagram</i> .....	29
Gambar 9. Karakteristik ISO 25010 .....	32
Gambar 10. Bagan Kerangka Pikir .....	37
Gambar 11. Model <i>Waterfall</i> .....	38
Gambar 12. Diagram <i>Use Case</i> .....	48
Gambar 13. <i>Class Diagram</i> Aluradmi.....	49
Gambar 14. <i>Class Diagram</i> Web Service .....	50
Gambar 15. Desain Sistem .....	52
Gambar 16. Desain Antarmuka Halaman Beranda.....	53
Gambar 17. Desain Antarmuka Menu .....	53
Gambar 18. Desain Antarmuka Halaman Alur Administrasi .....	54
Gambar 19. Desain Antarmuka Halaman Peta .....	54
Gambar 20. Desain Antarmuka Halaman Jurusan .....	55
Gambar 21. <i>Workspace Designer Tool</i> Android Studio.....	56
Gambar 22. <i>Palette</i> .....	57
Gambar 23. <i>Layout View</i> .....	57
Gambar 24. <i>Component Tree</i> .....	57
Gambar 25. <i>Properties</i> .....	57
Gambar 26. <i>Text View</i> .....	58
Gambar 27. Gradle .....	59
Gambar 28. Potongan Kode Program.....	60
Gambar 29. Formula Haversine.....	62
Gambar 30. Contoh Grafik.....	64

Gambar 31. Cara Kerja Algoritma Dijkstra .....	65
Gambar 32. Implementasi Node.....	65
Gambar 33. Menghubungkan Node .....	66
Gambar 34. Implementasi Algoritma Dijkstra .....	66
Gambar 35. Antarmuka Halaman Beranda .....	67
Gambar 36. Antarmuka Menu .....	68
Gambar 37. Antarmuka Halaman Alur Administrasi .....	68
Gambar 38. Antarmuka Halaman Detail Langkah Administrasi .....	69
Gambar 39. Antarmuka Halaman Tentang .....	70
Gambar 40. Implementasi Basis Data Aplikasi .....	70
Gambar 41. Implementasi Basis Data <i>Server</i> .....	71
Gambar 42. Surat Keputusan Dosen Pembimbing .....	87
Gambar 43. <i>Sequence Diagram</i> Cari Langkah.....	105
Gambar 44. <i>Sequence Diagram</i> Melihat Daftar Administrasi.....	105
Gambar 45. <i>Sequence Diagram</i> Memilih Jurusan .....	106
Gambar 46. <i>Sequence Diagram</i> Melihat Peta .....	107
Gambar 47. <i>Sequence Diagram</i> Melihat Alur Administrasi .....	108
Gambar 48. <i>Sequence Diagram</i> Melihat Detail Langkah Administrasi .....	109
Gambar 49. <i>Sequence Diagram</i> Menghapus Data .....	110
Gambar 50. <i>Sequence Diagram</i> Mengubah Data.....	111
Gambar 51. <i>Sequence Diagram</i> Tambah Data .....	112
Gambar 52. <i>Sequence Diagram</i> Melihat Rute.....	113
Gambar 53. <i>Sequence Diagram</i> Beranda.....	113
Gambar 54. <i>Sequence Diagram</i> Bantuan.....	114
Gambar 55. <i>Sequence Diagram</i> Tentang.....	114
Gambar 56. <i>Sequence Diagram</i> Penyetelan Awal Data .....	115
Gambar 57. Desain Antarmuka Bantuan .....	116
Gambar 58. Desain Antarmuka Detail Langkah Administrasi .....	116
Gambar 59. Desain Antarmuka Halaman Pencarian.....	116
Gambar 60. Desain Antarmuka Halaman Tentang.....	116
Gambar 61. Halaman Antarmuka Bantuan .....	122
Gambar 62. Halaman Antarmuka Pengaturan.....	123



Gambar 63. Halaman Antarmuka Pilihan Jurusan.....	123
Gambar 64. Halaman Antarmuka Peta .....	124
Gambar 65. Lokasi Titik.....	125
Gambar 66. Tes 1 Uji Penggunaan <i>Resouce</i> .....	127
Gambar 67. Tes 2 Uji Penggunaan <i>Resouce</i> .....	127
Gambar 68. Tes 3 Uji Penggunaan <i>Resouce</i> .....	128
Gambar 69. Tes 4 Uji Penggunaan <i>Resouce</i> .....	128
Gambar 70. Tes 5 Uji Penggunaan <i>Resouce</i> .....	129
Gambar 71. Desain Basis Data Aplikasi .....	135
Gambar 72. Desain Basis Data <i>Server</i> .....	136

## DAFTAR LAMPIRAN

Lampiran 1. Surat Keputusan Dosen Pembimbing .....	87
Lampiran 2. Instrumen Uji <i>Functional Suitability</i> .....	88
Lampiran 3. Instrumen Uji <i>Usability</i> .....	92
Lampiran 4. Definisi <i>Use Case</i> dan Skenario <i>Use Case</i> .....	94
Lampiran 5. Desain <i>Sequence Diagram</i> .....	105
Lampiran 6. Desain Antarmuka Aplikasi .....	116
Lampiran 7. Daftar Layout.....	117
Lampiran 8. Daftar File Java .....	119
Lampiran 9. Implementasi Antarmuka .....	122
Lampiran 10. Letak Lokasi Titik.....	125
Lampiran 11. Hasil Pengujian <i>Usability</i> .....	126
Lampiran 12. Hasil Pengujian <i>Performa Efficiency</i> .....	127
Lampiran 13. Hasil Pengujian <i>Compatibility</i> .....	130
Lampiran 14. Desain Basis Data .....	135

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Universitas Negeri Yogyakarta (UNY) merupakan salah satu dari Perguruan Tinggi Negeri (PTN) yang ada di wilayah Daerah Istimewa Yogyakarta. Dilansir dari situs [webometrics.info](http://webometrics.info), UNY menempati peringkat 29 dari 487 universitas yang ada di Indonesia untuk edisi Januari 2017. Dari data tersebut membuktikan bahwa UNY merupakan salah satu Perguruan Tinggi Negeri yang patut diperhitungkan di Indonesia sehingga banyak calon mahasiswa yang menjadikan UNY sebagai tujuan untuk melanjutkan pendidikan mereka.

Fakultas Teknik (FT) UNY adalah salah satu dari beberapa fakultas yang ada di UNY. Berdasarkan data dari situs [sbmptn.ac.id](http://sbmptn.ac.id), jumlah peminat pada tahun 2016 untuk jenjang pendidikan Strata 1 (S1) yang berada di FT UNY baik itu dalam bidang saintek maupun soshum berjumlah 8352 orang. Sementara itu, jumlah daya tampung untuk program studi S1 yang berada di FT UNY melalui jalur seleksi SBMPTN hanya berjumlah 283 orang. Hal tersebut membuktikan bahwa FT UNY menjadi salah satu fakultas di UNY yang diminati oleh calon mahasiswa.

Dewasa ini teknologi di dunia berkembang dengan sangat pesat, salah satunya adalah teknologi perangkat *mobile*. Pada awal perkembangannya, perangkat *mobile* hanya bisa digunakan untuk mengirim pesan, melakukan panggilan telepon dan lain sebagainya. Namun sejak berkembangnya teknologi telepon pintar (*smartphone*), teknologi *mobile* dapat digunakan untuk mencari

informasi dari berbagai sumber di internet, mengakses peta, mengirim *email*, berinteraksi melalui sosial media dan lain sebagainya.

Baidu Indonesia, Indonesia Digital Assosiation, dan Indonesia Creative Economy Academy merilis hasil riset mereka mengenai konsumsi berita online di Indonesia. Salah satu hasil dari riset tersebut menyebutkan bahwa 96% masyarakat Indonesia mengakses informasi melalui *smartphone* (Baidu, 2016). Hal tersebut dikarenakan salah satu keunggulan yang dimiliki oleh perangkat *mobile (smartphone)* yaitu dapat mengakses informasi dimana saja dan kapan saja. Android adalah sistem operasi *smartphone* yang paling banyak digunakan di Indonesia yaitu sebesar 78,78% (Statcounter, 2017).

Sebuah lembaga atau organisasi tidak dapat lepas dari administrasi dikarenakan organisasi itu sendiri termasuk di dalam unsur administrasi (Wirman Syafri, 2012:12). Pengertian dari administrasi yang sering kita sebut diambil dari Bahasa Inggris yaitu *administration* yang berarti proses (rangkaian) kegiatan usaha kerja sama sekelompok orang secara terorganisasi untuk mencapai tujuan tertentu secara efisien (Wirman Syafri, 2012:4). Pada perguruan tinggi yang produknya (*output*) menghasilkan lulusan berkualitas, perlu adanya proses yang harus dilakukan oleh para pendidik maupun tenaga kependidikan. Proses-proses yang sering kita jumpai antara lain penentuan jumlah mahasiswa yang masuk, mekanisme seleksi, pembuatan soal seleksi, pembayaran biaya dan pengisian KRS, persiapan bahan ajar oleh dosen, pelaksanaan proses belajar mengajar, ujian dan tugas, bimbingan dan ujian akhir skripsi serta wisuda dan penyerahan ijazah kepada mahasiswa yang lulus (Wirman Syafri, 2012:6).

Mahasiswa di perguruan tinggi juga memiliki keperluan terkait administrasi yang harus diselesaikan sampai lulus. Berdasarkan observasi yang dilakukan oleh penulis di Fakultas Teknik UNY, informasi terkait administrasi mahasiswa tersedia dalam bentuk *hardcopy* ditempel di papan pengumuman di Gedung Kantor Pusat Layanan Terpadu (KPLT) FT UNY dan jurusan sehingga informasi tersebut kurang fleksibel. Belum adanya pemanfaatan teknologi *mobile* dalam menyampaikan informasi terkait administrasi mahasiswa di FT UNY. Dapat terjadinya *miss document* administrasi dalam proses pengurusannya oleh mahasiswa.

Aplikasi adalah sebuah produk di mana produk yang berguna memberikan konten, fungsi dan fitur yang diinginkan oleh pengguna tetapi yang paling penting adalah dapat diandalkan dan bebas dari *error* (Pressman, 2015:415). Oleh sebab itu, perlu dilakukan suatu proses untuk menilai kelayakan dari sebuah aplikasi agar terjamin kualitasnya. ISO 9126 adalah standar untuk mengukur kualitas perangkat lunak yang paling terkenal tetapi standar ISO 9126 telah diperbarui menjadi ISO 25010(Wagner, 2013:10).

Dengan demikian penulis mengembangkan sebuah aplikasi sebagai solusi dari permasalahan diatas. Aplikasi tersebut diberi nama Aluradmi yang merupakan gabungan dua kata yaitu alur dan administrasi. Nama tersebut dipilih agar mudah diingat dan kontekstual dengan informasi yang disajikan dalam aplikasi Aluradmi yaitu alur administrasi. Aplikasi Aluradmi dikembangkan dengan basis *mobile* yang berfokus pada sistem operasi Android agar dapat digunakan oleh banyak mahasiswa. Informasi yang disajikan dalam aplikasi Aluradmi berfokus pada alur administrasi, tempat mengurus administrasi, dan berkas yang dibutuhkan untuk mahasiswa FT UNY. Aluradmi perlu diuji kelayakannya



menggunakan 4 aspek yaitu *functional suitability*, *compatibility*, *usability* dan *performance efficiency* pada standar ISO 25010.

## **B. Identifikasi Masalah**

Berdasarkan uraian latar belakang tersebut, ada beberapa masalah yang dapat kita identifikasi sebagai berikut:

1. Informasi mengenai administrasi mahasiswa tidak dapat diakses kapan saja dan dimana saja karena informasinya berbentuk *hardcopy*.
2. Belum ada pemanfaatan teknologi *mobile* dalam menyampaikan informasi administrasi mahasiswa di Fakultas Teknik Universitas Negeri Yogyakarta.
3. Dapat terjadinya *miss document* administrasi dalam proses pengurusannya.
4. Perlunya aplikasi melewati proses pengujian agar terbukti keandalannya dan terhindar dari kesalahan atau *error*.

## **C. Batasan Masalah**

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Mengembangkan aplikasi Aluradmi yang berisi informasi alur administrasi, tempat mengurus administrasi, dan berkas yang dibutuhkan untuk mahasiswa FT UNY.
2. Mengetahui kualitas aplikasi Aluradmi berdasarkan aspek *functional suitability*, *compatibility*, *usability* dan *performance efficiency* pada ISO 25010.

#### **D. Rumusan Masalah**

1. Bagaimanakah pembuatan aplikasi Aluradmi yang berisi informasi alur administrasi, tempat mengurus administrasi, dan berkas yang dibutuhkan bagi mahasiswa di Fakultas Teknik Universitas Negeri Yogyakarta.
2. Bagaimanakah kualitas aplikasi Aluradmi di Fakultas Teknik Universitas Negeri Yogyakarta berdasarkan aspek *functional suitability*, *compatibility*, *usability* dan *performance efficiency* pada ISO 25010.

#### **E. Tujuan Penelitian**

Berdasarkan rumusan masalah, tujuan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan aplikasi Aluradmi yang berisi informasi alur administrasi, tempat mengurus administrasi, dan berkas yang dibutuhkan bagi mahasiswa di Fakultas Teknik Universitas Negeri Yogyakarta.
2. Mengetahui kualitas aplikasi Aluradmi berdasarkan aspek *functional suitability*, *compatibility*, *usability* dan *performance efficiency* pada ISO 25010.

#### **F. Spesifikasi Produk**

Spesifikasi produk yang dikembangkan adalah sebagai berikut:

1. Aplikasi berbasis Android dengan versi minimum 4.1 atau *Jelly Bean*.
2. Informasi yang ditampilkan dalam aplikasi berupa alur administrasi, tempat mengurus administrasi, dan berkas yang dibutuhkan. Informasi tersebut terbatas dalam lingkungan Fakultas Teknik Universitas Negeri Yogyakarta dan dikhususkan pada informasi administrasi bagi mahasiswa.

3. Informasi yang berada di aplikasi Aluradmi didapatkan dari *server* melalui sebuah *Web Service* lalu di simpan ke dalam basis data aplikasi Aluradmi.
4. Peta yang digunakan dalam aplikasi Aluradmi menggunakan Google Maps.
5. Terdapat petunjuk singkat penggunaan aplikasi Aluradmi pada saat pertama kali dijalankan

## **G. Manfaat**

Manfaat yang diharapkan dari Pengembangan Aplikasi Aluradmi Sebagai Informasi Alur Administrasi Mahasiswa Berbasis Android di Fakultas Teknik Universitas Negeri Yogyakarta adalah sebagai berikut:

### **1. Bagi pengguna:**

- a. Memudahkan mahasiswa untuk mengetahui proses pengurusan administrasi mahasiswa di Fakultas Teknik Universitas Negeri Yogyakarta.
- b. Memudahkan mahasiswa dalam menemukan tempat untuk mengurus keperluan administrasi mahasiswa.
- c. Memudahkan mahasiswa dalam mempersiapkan syarat maupun berkas yang dibutuhkan untuk keperluan pengurusan administrasi mahasiswa.
- d. Dapat meminimalkan terjadinya *miss document* bagi mahasiswa yang sedang mengurus administrasi.

### **2. Bagi peneliti:**

- a. Dapat mengamalkan ilmu yang sudah didapat dari Perguruan Tinggi dibidang Teknik Informatika.
- b. Mengetahui teknik pengembangan perangkat lunak secara umum.
- c. Mengetahui teknik pengembangan aplikasi Android.

- d. Mengetahui teknik pengujian dengan menggunakan standar ISO 25010 untuk menguji aplikasi berbasis Android.
- e. Memberikan bahan referensi dan perbandingan untuk peneliti yang mengembangkan aplikasi Android.

## **BAB II**

### **KAJIAN TEORI**

#### **A. Deskripsi Teori**

##### **1. Sistem Informasi**

Perkembangan teknologi informasi yang sangat pesat saat ini terutama dibidang *mobile computing* membuat sistem informasi tidak hanya berada pada komputer *desktop* tetapi sudah berkembang dalam teknologi *mobile*. O'brien & Marakas (2010:4), mendefinisikan sistem informasi sebagai kombinasi yang terorganisir dari manusia, *hardware*, *software*, jaringan kominukasi, sumber data, prosedur dan kebijakan yang digunakan untuk mengumpulkan, menerima, mengubah dan menyebarkan informasi di dalam sebuah organisasi. Menurut Laudon dan Laudon (2012:15), sistem informasi dapat didefinisikan secara teknis sebagai suatu set komponen yang saling berinteraksi untuk mengumpulkan atau menerima, memproses, menyimpan, dan mendistribusikan informasi untuk membantu pengambilan keputusan dan kontrol di dalam sebuah organisasi. Sementara itu menurut Hariyanto (2008:143-144), sistem informasi memiliki arti sebagai:

suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Berdasarkan definisi yang sudah dipaparkan di atas dapat dirangkum bahwa sistem informasi adalah komponen-komponen yang saling berinteraksi dan membentuk satu kesatuan. Berfungsi untuk mengumpulkan, memproses,

menyimpan dan menyebarkan informasi sebagai pembantu dalam mengambil keputusan.

## **2. Administrasi**

Mahasiswa di Fakultas Teknik Universitas Negeri Yogyakarta perlu untuk menyelesaikan berbagai urusan administrasi mulai dari terdaftar sebagai mahasiswa sampai lulus kelak. Menurut Syafri (2012:11), administrasi adalah rangkaian kegiatan (proses) usaha kerja sama sekelompok orang dalam suatu organisasi untuk mencapai tujuan tertentu secara efisien. Administrasi dalam pengertian sempitnya yaitu segala kegiatan tulis-menulis, surat-menyurat dan kegiatan ketatausahaan (Marbun, 2003:33). Administrasi pada umumnya berkaitan dengan surat menyurat dan data administrasi tersebut direkam ke dalam sebuah buku khusus (Irmawati & Indrihapsari, 2014).

Berdasarkan berbagai penjelasan di atas dapat dirangkum bahwa administrasi adalah rangkaian kegiatan yang dilakukan oleh sekelompok orang dalam suatu organisasi untuk mencapai tujuan tertentu secara efisien. Administrasi biasanya berhubungan dengan kegiatan surat menyurat.

## **3. Android**

Pengembangan aplikasi Aluradmi menggunakan Android sebagai basis sistem operasinya. Android merupakan sistem operasi berbasis Linux yang digunakan untuk telepon genggam contohnya smartphone yang telah berkembang dengan pesat (Agustina, Darmawiguna, & Sunarya, 2017). Seperti sistem operasi linux, Android merupakan sistem operasi terbuka atau *open source* sehingga semua orang dapat melakukan pengembangan secara bebas terhadap sistem operasi Android.



### a. Perkembangan Android

Android telah mengalami banyak perkembangan sejak diluncurkan pada tanggal 7 November 2007. Dibawah ini adalah data statistik versi Android digunakan dari situs resmi pengembang Android (developer.Android.com).

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	3.7%
4.2.x		17	5.4%
4.3		18	1.5%
4.4	KitKat	19	20.8%
5.0	Lollipop	21	9.4%
5.1		22	23.1%
6.0	Marshmallow	23	31.3%
7.0	Nougat	24	2.4%
7.1		25	0.4%

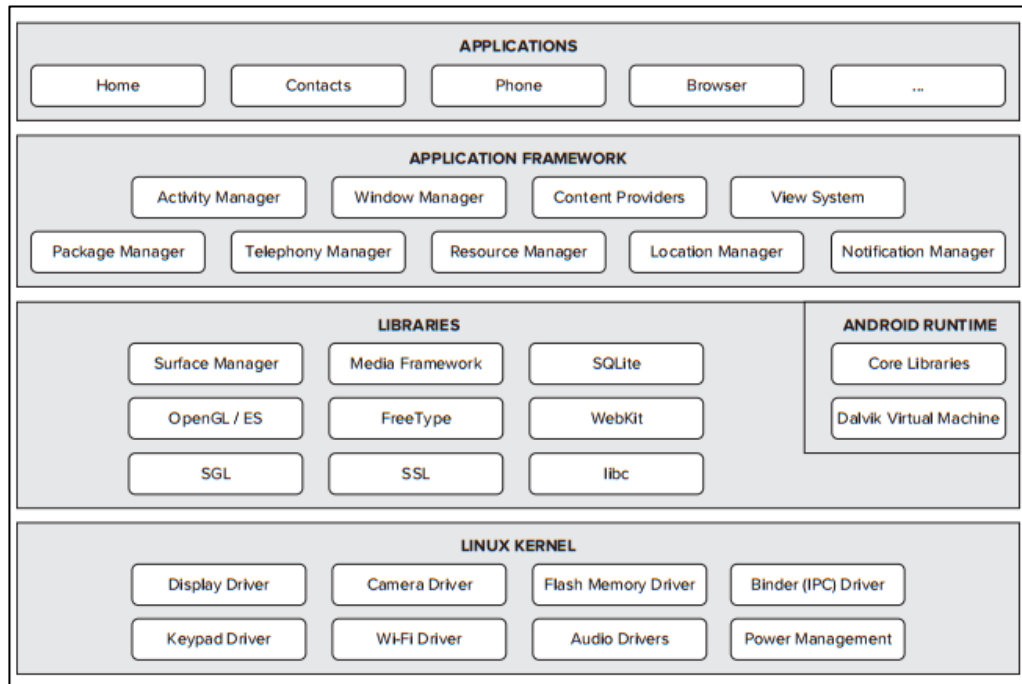
Gambar 1. Statistik Perangkat Android (Google, 2017)

Berdasarkan data yang tertera pada Gambar 2 OS Android versi *Jelly Bean*, *Kitkat*, *Lollipop* dan *Marshmallow* adalah 4 besar versi yang paling banyak digunakan oleh pengguna.

### b. Arsitektur Android

Secara umum *software stack* Android terdiri dari 4 lapisan dan terdapat 5 bagian di dalamnya. Bagian yang paling mendasar adalah *linux kernel*, kemudian

di atasnya terdapat *libraries* dan *Android runtime*, di atasnya lagi terdapat *application framework* dan pada lapisan paling atas terdapat aplikasi.



Gambar 2. Arsitektur Android (Meeng Lee W, 2012)

Berikut adalah penjelasan tentang lapisan yang terdapat dalam Android menurut Meeng Lee W (2012):

#### 1) *Linux Kernel*

Kernel ini adalah dasar dari Android itu sendiri. Di dalam lapisan *kernel* terdapat berbagai macam *driver* untuk menjalankan *hardware* dalam perangkat Android.

#### 2) *Libraries*

Di dalam bagian *libraries* terdapat semua kode yang menjalankan fitur utama dari sistem operasi Android.

### 3) *Android Runtime*

Pada bagian Android runtime yang terdapat di lapisan yang sama dengan libraries terdapat 2 fitur penting yaitu *core libraries* dan *dalvik virtual machine*. *Android runtime* menyediakan *core libraries* sehingga pengembang aplikasi dapat mengembangkan aplikasi Android menggunakan Bahasa Pemrograman Java. *Dalvik Virtual Machine* membuat aplikasi Android dapat menjalankan semua proses yang terdapat di dalam aplikasi tersebut.

### 4) *Application Framework*

Di dalam *Application Framework* terdapat berbagai macam kelas yang sudah disediakan oleh Android. Para pengembang aplikasi dapat menggunakannya untuk membuat aplikasi Android.

### 5) *Application*

Pada lapisan paling atas dari arsitektur Android terdapat lapisan aplikasi dimana semua aplikasi yang dibangun oleh pengembang terdapat di lapisan ini. Pada lapisan aplikasilah semua interaksi antara pengguna dan sistem operasi terjadi.

## **c. Android SDK**

Pengembangan aplikasi Android pada umumnya menggunakan Bahasa pemrograman java dan *Software Development Kit (SDK) tools* yang dapat diunduh dari situs resmi Android. SDK Android berisi beberapa komponen yaitu *Application Framework, Platform Tools, SDK Tools , extras* dan contoh aplikasi. Untuk mengembangkan aplikasi Android diperlukan sebuah perangkat khusus yang disebut dengan IDE (*Integrated Development Environment*). Pihak Android

menyarankan pengembang untuk menggunakan IDE Android Studio yang telah terintegrasi langsung dengan Android SDK.

Berdasarkan penjelasan diatas dapat dirangkum bahwa Android merupakan sistem operasi *mobile* berbasis Linux dan memiliki pengguna terbesar didunia. Sistem Operasi Android tersedia dalam berbagai versi dengan 4 versi yang paling banyak digunakan yaitu *Jelly Bean*, *Kitkat*, *Lollipop*, dan *Marshmallow*. Arsitektur Android terdiri dari *Linux Kernel*, *Libraries*, *Android Runtime*, *Application Framework*, dan *Application*. Pengembangan Aplikasi Aluradmi akan menggunakan IDE Android Studio dan Android SDK. Aplikasi Aluradmi dikembangkan dengan minimal versi *Jelly Bean* dan Maksimal *Marshmallow*.

#### **4. Aplikasi *Mobile***

Perkembangan teknologi *smartphone* yang pesat di dunia tidak terlepas dari perkembangan aplikasi *mobile* yang membuat *smartphone* dapat digunakan dengan maksimal. Aplikasi *mobile* dapat didefinisikan sebagai sebuah *software* yang di desain khusus untuk berjalan pada perangkat *mobile* seperti *smartphone* atau *tablet* (Salz & Moranz,2013). Aplikasi *mobile* tidak hanya dapat melakukan tugas-tugas dasar seperti menelepon maupun mengakses layanan pesan, tetapi aplikasi *mobile* sudah dapat melaksanakan tugas-tugas yang rumit seperti mencari posisi pengguna, mengakses dan menampilkan informasi seperti peta digital.

Berikut adalah jenis dari aplikasi *mobile* menurut Peggy & Jennifer (2013):

**a. *Mobile Web Apps***

*Mobile web* merujuk pada konten dilihat melalui *web browser* yang dimiliki oleh *smartphone*. Tingkat kompatibilitas yang tinggi menjadi kelebihan utama dari jenis aplikasi ini dikarenakan dapat dijalankan disemua perangkat *smartphone* dengan berbagai macam spesifikasi. Namun *mobile web apps* juga memiliki kekurangan tidak dapat mengakses informasi dalam keadaan *offline*.

**b. *Native Apps***

*Native Apps* merupakan jenis aplikasi yang terbaik dan dapat diakses tanpa terkoneksi dengan website. Kelebihan aplikasi jenis ini adalah dapat menggunakan semua fitur yang tersedia di perangkat *mobile* melalui *Application Programming Interface* (API) dan memiliki performa yang paling baik. *Native Apps* dibuat secara khusus untuk suatu platform tertentu dan spesifikasi tertentu, seperti aplikasi untuk sistem operasi berbasis Android, iOS maupun Blackberry.

**c. *HTML 5 Apps***

HTML 5 Apps merupakan aplikasi yang dikembangkan menggunakan pemrograman berbasis HTML 5. Kelebihan dari jenis aplikasi ini adalah sedikitnya proses adaptasi yang dibutuhkan untuk setiap *platform* sehingga dapat untuk proses pengembangan dan penyebarannya terhitung cepat dengan biaya yang efektif. Namun aplikasi jenis ini masih belum bisa menyaingi *Native Apps* dalam bidang performa dan fungsionalitas.

#### **d. *Hybrid Apps***

*Hybrid Apps* merupakan aplikasi yang dikembangkan dengan menggabungkan teknologi HTML 5 dan *Native*. *Hybrid Apps* dapat mengakses beberapa API seperti lokasi dan sensor.

#### **e. *Cross-Platform Apps***

Aplikasi *Cross-Platform* dibuat agar dapat dijalankan ke dalam berbagai platform yang berbeda. Aplikasi jenis *cross platform* dibuat dengan menggunakan sebuah *framework* seperti appcelerator titanium, Rhodes maupun Phonegap. Kelebihan dari aplikasi jenis ini adalah pihak *developer* hanya menulis kode satu kali dan dapat digunakan kembali untuk pengembangan aplikasi pada platform yang berbeda.

Berdasarkan penjelasan di atas dapat dirangkum bahwa aplikasi *mobile* adalah aplikasi yang di desain secara khusus untuk perangkat *mobile*. Dalam pengembangannya terdapat berbagai jenis aplikasi *mobile* seperti aplikasi *web mobile*, aplikasi *native*, aplikasi HTML 5 , aplikasi *Hybrid* dan aplikasi lintas *platform*. Pengembangan aplikasi Aluradmi akan memakai jenis aplikasi *native* dikarenakan pada performa dan fungsionalitas lebih baik dari jenis aplikasi yang lain.

### **5. SQLite**

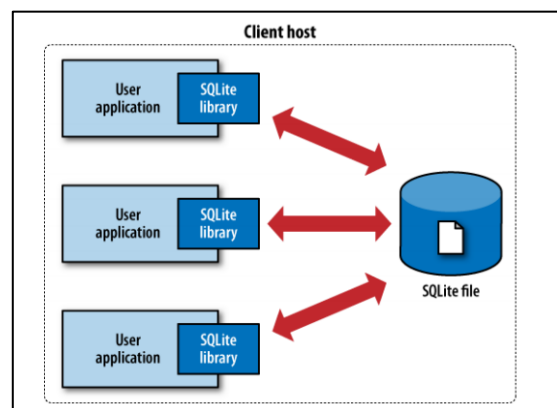
Sistem informasi tanpa informasi tidak akan berguna sehingga informasi tersebut harus disimpan ke dalam penyimpanan yang aman dan handal agar tidak hilang atau rusak, salah satunya adalah dengan menggunakan basis data. Salah satu basis data yang banyak digunakan untuk aplikasi *mobile* adalah SQLite. SQLite adalah sebuah *software library* yang ditulis dalam Bahasa C dan



bersifat *public domain* sehingga semua orang bisa memakainya (Feiler, 2015:10). Sementara itu menurut Kreibich (2010:1) "SQLite is a public-domain software package that provides a relational database management system, or RDBMS". Berikut adalah fitur yang dimiliki oleh SQLite menurut Kreibich (2010):

**a. Self-Contained, No Server Required**

SQLite tidak memiliki server terpisah, seluruh *engine database* terintegrasi dengan aplikasi apa saja yang membutuhkan akses ke *database*. RDBMS yang memiliki arsitektur *client-server* akan membutuhkan *advance multitasking* dan *high-performance inter-process communication* sementara SQLite hanya membutuhkan kemampuan untuk membaca dan menulis ke dalam tipe penyimpanan tertentu. Hal tersebut membuat SQLite dapat diintegrasikan ke dalam *enviromtent* apa saja seperti *mobile phones*, *game console* dan perangkat lainnya.



Gambar 3. Arsitektur *Serverless* SQLite (Kreibich, 2010)

**b. Zero Configuration**

SQLite tidak perlu dipasang sebelum digunakan, tidak ada prosedur untuk mempersiapkan. Tidak membutuhkan konfigurasi *file* dan tidak perlu memberitahu sistem kalau SQLite sedang berjalan (SQLite, 2017).

**c. *Single File Database***

SQLite mempacketkan seluruh *database* ke dalam satu *file*.

**d. *Cross-Platform***

Format *file database* SQLite dapat digunakan diberbagai *platform* (SQLite, 2017).

**e. *Unique Feature***

Salah satu fitur yang paling penting adalah SQLite menggunakan dynamic-type systems untuk table dimana pengembang tidak perlu mendefinisikan tipe data untuk suatu kolom dalam tabel dikarenakan SQLite yang akan menangani hal tersebut. Fitur lain dari SQLite yang sangat berguna adalah kemampuan untuk memanipulasi lebih dari satu *database* dalam satu waktu.

**f. *Compatible Licence***

SQLite dan SQLite *code* tidak terdaftar dalam *open source license* manapun seperti GNU *General Public Licence* dan tidak memiliki *user licence*.

**g. *Support Full ACID Transactional***

*ACID* yang dimaksud adalah acronim dari *atomicity, consistency isolation, durability*. Berikut adalah penjelasannya (Feiler, 2015):

1) *Atomicity*

*Atomicity* adalah salah satu aspek dalam sebuah transaksi dimana hanya terdapat transaksi tersebut semuanya sukses atau semua transaksi tersebut gagal.

2) *Consistency*

Di dalam proses transaksi, database memulai dengan keadaan yang valid dan berakhir dengan keadaan yang valid.

### 3) *Isolation*

Isolation dapat diartikan bahwa transaksi dapat dijalankan secara berurutan ataupun bersamaan. Transaksi tersebut tidak akan saling mengganggu satu sama lain.

### 4) *Durability*

Data disimpan oleh sistem dalam keadaan yang tepat, bahkan saat terjadi kegagalan dan sistem melakukan restart data tetap tersedia dalam keadaan yang tepat.

## **h. Full-Featured**

Kata "Lite" dalam SQLite bukan berarti ada pembatasan terhadap implementasi dari SQL. SQLite memiliki *full-feature* dari SQL.

## **i. Highly Reliable**

SQLite adalah solusi penyimpanan yang sangat bisa diandalkan. SQLite sudah di buktikan dalam prakteknya tanpa mengalami masalah di jutaan perangkat seperti *smartphone*, perangkat *IoT*, dan aplikasi *desktop* di dunia selama beberapa dekade (SQLite, 2017).

Berdasarkan penjelasan di atas dapat dirangkum bahwa SQLite sudah diimplementasikan ke dalam jutaan *smartphone* dan tidak mengalami masalah sedikitpun. SQLite memiliki kelebihan tidak membutuhkan *server*, tidak ada konfigurasi, dalam berjalan di lintas *platform*, *single file database*, *compatible license*, dapat diandalkan, memiliki fitur yang unik, mendukung *ACID transaction*, dan *Full Feature SQL*. SQLite sangat cocok digunakan sebagai *database* bagi aplikasi Android yang akan dikembangkan oleh peneliti.

## 6. Google Maps

Google Maps merupakan salah satu produk dari Google Inc yang banyak digunakan untuk mengakses peta digital. Google Maps awal mulanya dikembangkan oleh Lars and Jens Rasmusse. Dua bersaudara tersebut mendirikan sebuah perusahaan yang bernama Where 2 Technologies yang berfokus pada solusi pemetaan dan akhirnya diakuisisi oleh Google pada Oktober 2004. Google Maps v3 merupakan versi Google Maps yang paling baru dan memiliki performa yang lebih bagus daripada versi Google Maps sebelumnya (Svennerberg, 2010).

Google Maps menyediakan layanan seperti pemetaan jalan, rute dan navigasi untuk berbagai rute yang dikehendaki oleh penggunanya serta dapat menampilkan waktu tempuh dari perjalanan tersebut. Google Maps menggunakan teknologi pencitraan satelit untuk melakukan proses pemetaan objek yang terdapat di permukaan bumi secara *realtime*. Oleh karena itu dalam kurun waktu tertentu peta pada Google Maps akan di update dalam kurun waktu tertentu.

Saat ini para pengembang *website*, *blog* maupun aplikasi *mobile* dapat menambahkan Google Maps ke dalam aplikasi mereka dengan menggunakan Google Maps API. Google Maps API resmi dirilis ke publik pada bulan juni 2005 (Svennerberg, 2010). Google Maps dapat disesuaikan dengan kebutuhan aplikasi yang akan dibuat oleh para pengembang aplikasi (Anupriya & Saxena, 2013).

Berdasarkan penjelasan diatas dapat dirangkum bahwa Google Maps merupakan teknologi dalam bidang pemetaan bumi yang memiliki fitur pemetaan jalan, rute dan navigasi ke tempat tujuan dengan pemanfaatan citra satelit dalam

proses pemetaannya. Saat ini Google Maps dapat digunakan dalam teknologi website maupun aplikasi *mobile* dengan menggunakan Google Maps API. Oleh karena itu, Google Maps cocok digunakan dalam pengembangan aplikasi Aluradmi pada fitur peta.

## 7. Algoritma Dijkstra

Rute adalah salah satu hal yang dicari orang ketika ingin melakukan perjalanan ke suatu tempat. Terdapat berbagai macam algoritma yang dikembangkan untuk mengatasi masalah pencarian rute, salah satunya adalah Algoritma Dijkstra. Algoritma Dijkstra merupakan "algoritma pencarian graf untuk menyelesaikan permasalahan *single-source shortest path* pada sebuah graf berbobot tak-negatif" (Riski, 2015:171). Algoritma Dijkstra biasanya digunakan pada *routing* untuk menemukan lintasan terpendek dengan *cost* yang paling sedikit (Singal & Chhillar, 2014).

### ALGORITHM 1 Dijkstra's Algorithm.

```

procedure Dijkstra( $G$ : weighted connected simple graph, with
    all weights positive)
{ $G$  has vertices  $a = v_0, v_1, \dots, v_n = z$  and lengths  $w(v_i, v_j)$ 
  where  $w(v_i, v_j) = \infty$  if  $\{v_i, v_j\}$  is not an edge in  $G$ }
for  $i := 1$  to  $n$ 
   $L(v_i) := \infty$ 
 $L(a) := 0$ 
 $S := \emptyset$ 
{the labels are now initialized so that the label of  $a$  is 0 and all
  other labels are  $\infty$ , and  $S$  is the empty set}
while  $z \notin S$ 
   $u :=$  a vertex not in  $S$  with  $L(u)$  minimal
   $S := S \cup \{u\}$ 
  for all vertices  $v$  not in  $S$ 
    if  $L(u) + w(u, v) < L(v)$  then  $L(v) := L(u) + w(u, v)$ 
    {this adds a vertex to  $S$  with minimal label and updates the
    labels of vertices not in  $S$ }
return  $L(z)$  { $L(z)$  = length of a shortest path from  $a$  to  $z$ }
  
```

Gambar 4. Algoritma Dijkstra (Rosen, 2012)

Berikut cara kerja dari Algoritma Dijkstra:

1. Berikan nilai pada titik awal 0 dan berikan nilai pada titik yang lain dengan nilai maksimal. Inisialisasi S sebagai sebuah tempat untuk menyimpan titik-titik yang terpilih.
2. Pilih titik dengan nilai terkecil (U) lalu masukkan titik U ke dalam S.
3. Kunjungi salah satu titik yang terhubung dengan titik U tersebut namun tidak ada di dalam S (V).
4. Tambahkan jarak antara titik U dan titik V ke dalam nilai titik U. Lalu bandingkan nilai antara titik U dan titik V. Apabila nilai titik V lebih besar daripada nilai titik U maka nilai titik V diganti dengan nilai yang dimiliki titik U.
5. Ulangi Langkah 3 dan 4 sampai semua titik yang terhubung dengan titik U namun tidak ada di dalam S dikunjungi.
6. Ulangi langkah 2 sampai 5 hingga titik tujuan dikunjungi.
7. S akan menunjukkan titik-titik mana saja yang harus dilalui untuk sampai di titik tujuan dengan jarak tempuh paling kecil.

Berdasarkan penjelasan diatas dapat disimpulkan Algoritma Dijkstra merupakan algoritma untuk mencari rute tercepat berdasarkan sebuah graf berbobot tak negative. Penggunaan Algoritma Dijkstra di dalam aplikasi Aluradmi adalah untuk menentukan rute di dalam Fakultas Teknik UNY dari satu lokasi ke lokasi yang lainnya menggunakan Google Maps sebagai pemetaannya. *Vertex* atau titik pada Algoritma Dijkstra dibuat menggunakan koordinat peta yaitu *latitude* dan *longitude*.

## 8. Haversine Formula

Haversine Formula digunakan untuk menghitung jarak dari 2 lokasi di permukaan bumi dengan menggunakan garis bujur dan garis lintang sebagai inputannya (Saputra & Ardana, 2016). Haversine Formula ditemukan oleh James Andrew pada tahun 1805 dan menjadi sangat populer untuk para pelaut (Brummelen, 2013: 160). Berikut adalah rumus dari Haversine Formula dengan titik 1( $\Phi_1, \gamma_1$ ) dan titik 2( $\Phi_2, \gamma_2$ ):

$$d = 2r \sin^{-1} \left( \sqrt{\sin^2 \left( \frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left( \frac{\gamma_2 - \gamma_1}{2} \right)} \right)$$

Keterangan:

$\Phi = \text{Latitude}$

$\gamma = \text{Longitude}$

$d = \text{Jarak}$

$r = \text{Radius Bumi} = 6371 \text{ km}$

Berdasarkan penjelasan diatas dapat dirangkum bahwa Formula Haversine dapat digunakan untuk menentukan jarak antara 2 titik di permukaan bumi berdasarkan koordinat peta yaitu *latitude* dan *longitude*. Formula Haversine dalam penelitian ini digunakan untuk menentukan bobot atau jarak antara 2 titik yang digunakan dalam Algoritma Dijkstra.

## 9. *Software Development Life Cycle (SDLC)*

Pengembangan software atau aplikasi tentunya memerlukan berbagai tahapan dari awal pembuatan sampai software atau aplikasi tersebut jadi. Dalam pengembangan software berbagai tahapan itu disebut dengan *Software Development Life Cycle (SDLC)*. Definisi dari *Software Development Life Cycle*

adalah “proses mengembangkan atau mengubah suatu system perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan system-sistem perangkat lunak sebelumnya” (Rosa & Shalahuddin, 2013:26).

Berikut beberapa model *software development life cycle*:

**a. Model *Waterfall* (Air Terjun)**

Model *waterfall* sering disebut sebagai *the classic life cycle* atau alur hidup klasik yang melakukan pendekatan alur secara sekuensial atau terurut dari setiap tahapan pengembangan. Tahapan dari model *waterfall* dimulai dari analisis kebutuhan , desain , pengodean, dan pengujian. Setiap tahapan dari model *waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap pengembangan yang berikutnya.

Menurut Rosa & shalahuddin (2013:30-31), model *waterfall* memiliki beberapa kelemahan dan kelebihan, namun model *waterfall* ini menjadi dasar dari model pengembangan yang lainnya. Model *waterfall* merupakan model SDLC yang paling sederhana. Model ini cocok digunakan untuk mengembangkan perangkat lunak dengan spesifikasi yang tidak berubah-ubah.

**b. Model *Prototype***

Model *prototype* merupakan SDLC yang dimulai dari tahap komunikasi dengan pengguna untuk mengumpulkan kebutuhan dari pengguna tersebut, kemudian pihak pengembang membuat model purwarupa yang akan ditunjukkan kepada pengguna. Kemudian purwarupa tersebut dievaluasi oleh pengguna sampai ditemukan spesifikasi yang sesuai keinginan dari pengguna (user).



Rosa & Shalahuddin (2013:31-33), menyebutkan bahwa model *prototype* cocok digunakan untuk membantu pengembang dalam mengetahui kebutuhan pengguna secara detail tetapi memiliki resiko besar terhadap biaya pengembangan dan waktu pengerjaannya. Model *prototype* kurang cocok digunakan untuk mengembangkan aplikasi dengan skala besar karena akan memakan waktu dan tenaga.

#### **c. *Rapid Application Development (RAD)***

Model Rapid Application Development (RAD) merupakan model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Model RAD membagi tim menjadi beberapa komponen sehingga pengerjaan perangkat lunak dapat dilakukan secara parallel (Rosa & Shalahuddin, 2013:34).

Menurut Rosa & Shalahuddin (2013), model RAD cocok diterapkan apabila memenuhi beberapa kriteria sebagai berikut:

1. Anggota tim sudah memiliki pengalaman dalam mengembangkan perangkat lunak sejenis
2. Pengembang sudah memiliki komponen-komponen sistem yang bisa digunakan kembali dalam proyek tersebut.

#### **d. *Model Iteratif***

Model Iteratif menggabungkan antara model *waterfall* dan *prototype*. Model iteratif dibuat untuk mengatasi kelemahan dalam model *waterfall* yang tidak mengakomodasi iterasi dan model *prototype* yang tidak selalu menghasilkan produk dalam proses iteratifnya. Model ini sangat cocok digunakan pengembang dengan pergantian staff yang tinggi (Rosa & Shalahuddin, 2013:38-39).

#### **e. Model *Spiral***

Model *Spiral* merupakan model pengembangan yang memasangkan *iterative* pada model *prototype* dengan control dan sistematik dari model waterfall. Model Spiral menyediakan pengembangan dengan cara cepat dan pada setiap versi perangkat lunak memiliki tambahan fungsi (Rosa & Shalahuddin, 2013:39).

Menurut Rosa & Shalahuddin (2013), Model *Spiral* cocok digunakan untuk mengembangkan aplikasi skala besar tetapi dengan target waktu dan biaya yang tidak mengikat. Dari beberapa SDLC model *spiral* adalah model yang dapat memberikan jaminan kualitas yang paling baik untuk aplikasi skala besar.

Berdasarkan beberapa penjelasan tentang model *Software Delevopment Life Cycle* di atas. Model yang cocok digunakan untuk penelitian ini adalah model *waterfall* dikarenakan model tersebut paling sederhana, cocok digunakan untuk mengembangkan aplikasi yang spesifikasinya tidak berubah-ubah dan keterbatasan waktu serta biaya yang dimiliki oleh peneliti.

#### **10. Unified Modeling Language**

Desain adalah salah satu tahapan yang diperlukan dalam pembuatan *software* atau aplikasi. Salah satu cara untuk mendesain sebuah *software* adalah dengan melakukan pemodelan. Bahasa yang digunakan untuk melakukan pemodelan *software* atau aplikasi yang berbasis objek adalah Unified Modelling Language (*UML*). Fowler (2003:1), menjelaskan bahwa *Unified Modeling Languange (UML)* adalah "*family of graphical notations, backed by single meta-model, that help in describing and designing software systems, particularly software systems built using the object-oriented (OO) style*". Definisi lain dari

*UML* adalah sebuah bahasa pemodelan visual yang memungkinkan pembuat sistem untuk membuat *blueprint* yang menangkap visi mereka dalam sebuah standar, mudah dipahami dan menyediakan mekanisme untuk membagi dan mengkomunikasikan visi mereka secara efektif (Schmuller, 2004). Menurut Mellor dan Fowler (Fowler, 2003:2) terdapat tiga cara penggunaan *UML* yaitu sebagai sketsa, sebagai cetak biru dan sebagai bahasa pemrograman.

Menurut Rosa & Shalahuddin (2013,141), diagram *UML* dapat dibagi menjadi 3 kategori yaitu:

**a. *Structure Diagrams***

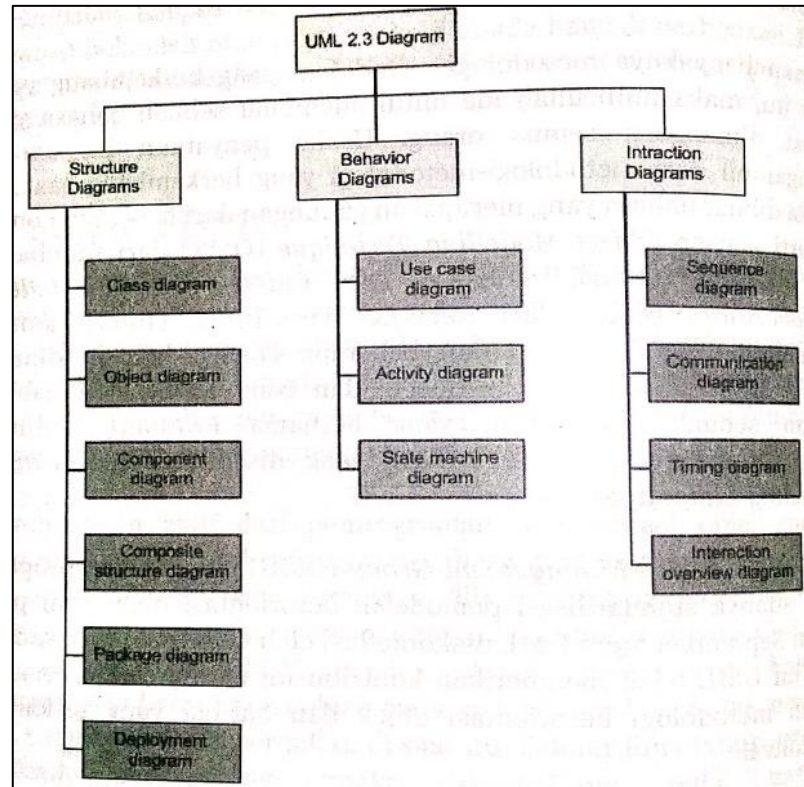
Struktur diagram adalah kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari suatu sistem yang dimodelkan. Diagram yang masuk kategori ini adalah *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.

**b. *Behavior diagrams***

*Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan rangkaian perubahan yang terjadi dalam sistem. Diagram yang masuk dalam kategori ini adalah *use case diagram*, *activity diagram*, dan *state machine diagram*.

**c. *Intraction Diagrams***

*Intraction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi antar sistem, interaksi antar subsistem pada suatu sistem. Diagram yang masuk dalam kategori ini adalah *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction overview diagram*.

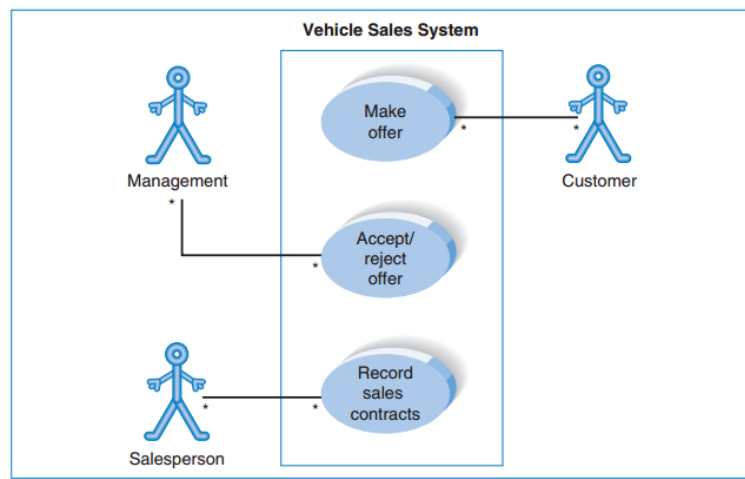


Gambar 5. Diagram *UML* (Rosa & Shalahuddin,2013)

Berdasarkan penjelasan diatas dapat dirangkum bahwa *UML* merupakan bahasa pemodelan visual yang digunakan untuk mendeskripsikan dan mendesain sebuah sistem agar mudah dipahami oleh orang lain. Terdapat 3 kategori dalam *UML* yaitu *Structure Diagram*, *Behaviour Diagram*, dan *Intraction Diagram*. Proses pengembangan aplikasi Aluradmi hanya menggunakan 3 diagram yaitu use case diagram, class diagram dan sequence diagram dikarenakan ketiga diagram tersebut telah mewakili masing-masing kategori diagram yang telah dijelaskan di atas. Berikut adalah penjelasan dari masing-masing diagram tersebut:

1) *Use Case Diagram*

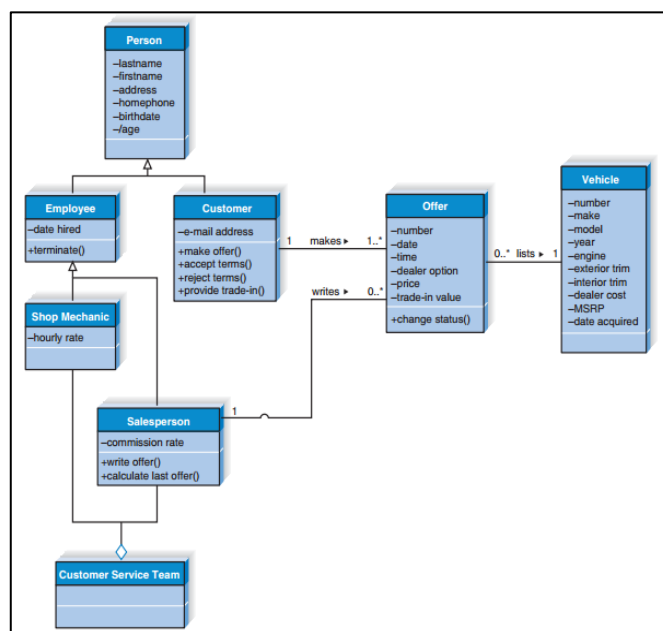
Menurut Rosa & Shalahuddin (2014:155), use case diagram merupakan “permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. Contoh diagram *use case* dapat dilihat pada Gambar 7.



Gambar 6. Use Case Diagram (Dennis, Wixom & Roth, 2012)

## 2) Class Diagram

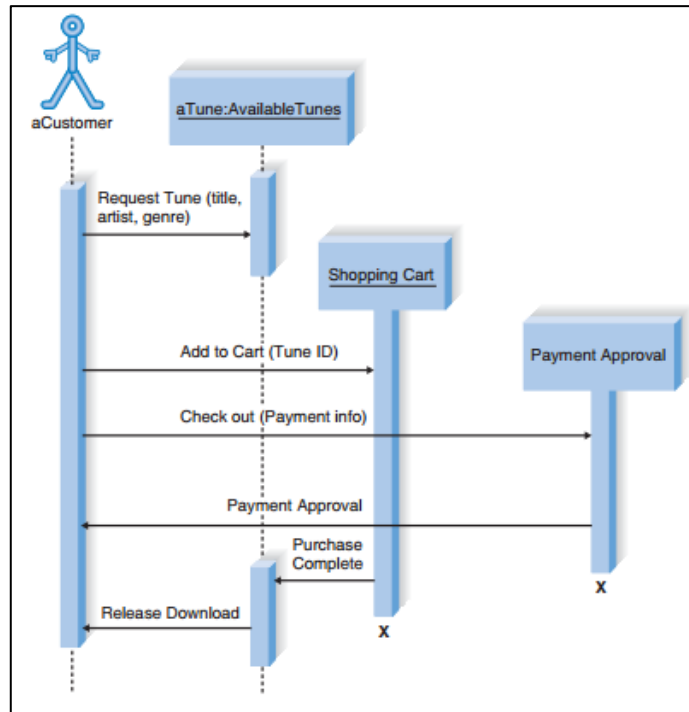
*Class Diagram* mendeskripsikan jenis objek dalam sistem dan jenis hubungan statis antar objek tersebut serta menunjukkan properti, operasi dan batasan dalam kelas yang diterapkan dalam hubungan antar objek tersebut (Fowler, 2004:35).



Gambar 7. *Class Diagram* (Dennis, Wixom & Roth, 2012)

### 3) Sequence Diagram

Menurut Dennis, Wixom & Roth (2012:530), *sequence diagram* menggambarkan objek yang berpartisipasi dalam *use case* dan message yang melewati mereka dari waktu ke waktu dalam sekali penggunaan *use case*.



Gambar 8. *Sequence Diagram* (Dennis, Wixom & Roth, 2012)

## 11. Kualitas Perangkat Lunak

Pengguna tentunya akan merasa nyaman menggunakan suatu *software* apabila *software* tersebut mudah digunakan dan tidak mengalami masalah pada saat digunakan. Untuk itu pengujian *software* untuk mengukur kualitasnya menjadi komponen yang penting. Menurut Pressman (2015:414) Tujuan penting dari pengembangan perangkat lunak adalah menghasilkan perangkat lunak yang berkualitas. Sedangkan Rosa & Shalahuddin (2014:5) berkata bahwa "perangkat lunak yang baik adalah perangkat lunak yang fokus pada pengguna atau

pelanggan". Untuk itu diperlukan pengujian terhadap perangkat lunak sebelum diberikan kepada pelanggan untuk meminimalkan kesalahan teknis maupun non teknis dalam perangkat lunak tersebut (Rosa & Shalahuddin, 2014:271-272).

Kualitas perangkat lunak dapat diukur menggunakan beberapa model seperti McCall dan ISO 9126 (Pressman, 2015:416-418). Pada tahun 2011 *International Organization of Standardization* merilis ISO 25010 yang menggantikan ISO 9126 dan akan menjadi tipe model kualitas perangkat lunak yang paling terkenal (Wagner, 2013:60).

Dalam penelitian ini peneliti menggunakan model ISO 25010 untuk mengukur kualitas perangkat lunak. ISO 25010 menetapkan 8 karakteristik dalam standarnya yaitu *functional suitability, reliability, performance efficiency, usability, maintainability, security, compatibility, portability* (Wagner, 2013:62). Berikut adalah penjelasan dari masing-masing karakteristik:

**a. *Functional Suitability***

*Functional Suitability* berarti sejauh mana produk yang dihasilkan dapat memenuhi fungsi yang dibutuhkan oleh pengguna.

**b. *Performance Efficiency***

*Performance efficiency* adalah sebaik apa respon dari produk terhadap permintaan pengguna dan seefisien apa dalam eksekusinya.

**c. *Compatibility***

*Compatibility* dapat didefinisikan sebagai sejauh apa suatu sistem dapat berbagi informasi, komponen bahkan menjalankan fungsi yang diperlukan dengan produk lain pada saat berbagi lingkungan perangkat keras atau perangkat lunak yang sama.

**d. *Reliability***

*Reliability* berarti sejauh mana kemampuan dari suatu sistem untuk menjalankan fungsi tertentu dibawah kondisi tertentu dan dalam jangka waktu yang ditetapkan.

**e. *Security***

*Security* dapat didefinisikan sebagai sejauh mana kemampuan dari suatu sistem dapat memberikan perlindungan terhadap informasi , data atau sistem yang memiliki tingkat akses data yang telah disesuaikan dengan tingkat otoritas dari pengguna.

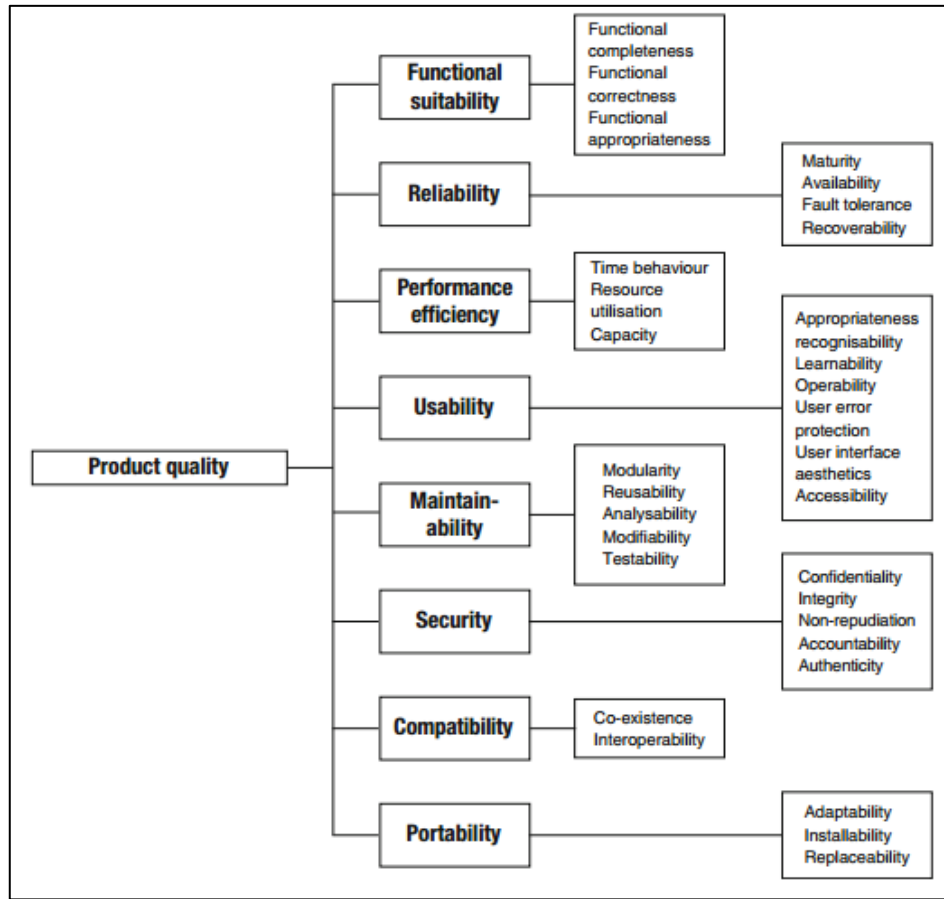
**f. *Maintainability***

*Maintainability* berarti tingkat efektivitas dan efisiensi dari suatu sistem atau produk dapat dimodifikasi untuk tujuan perbaikan.

**g. *Portability***

*Portability* berarti tingkat efektivitas dan efisiensi system atau produk dapat ditranster ke dalam platform lain misalnya Bahasa pemrograman, sistem operasi dan hardware.





Gambar 9. Karakteristik ISO 25010 (Wagner, 2013)

Menurut David (2011), pengujian untuk mobile application meliputi 4 aspek yaitu *functional testing*, *compatibility testing*, *usability testing* dan *performance testing*. Berikut adalah penjelasannya:

#### a. **Functional Testing**

*Functional testing* adalah metode tradisional yang digunakan untuk memvalidasi pemenuhan aplikasi terhadap kebutuhan fungsional dan kebutuhan bisnis. Dalam standar ISO 25010 yang mewakili aspek ini adalah *functional suitability*.

#### **b. *Compatibility Testing***

*Compatibility testing* adalah sejenis tes yang dilakukan oleh aplikasi atau web dengan menggunakan variasi *browser*, OS, jenis perangkat, ukuran perangkat dan kecepatan koneksi. Dalam standar ISO 25010 yang mewakili aspek ini adalah *compatibility*

#### **c. *Usability Testing***

*Usability testing* digunakan untuk memastikan pengalaman pengguna dalam menggunakan aplikasi efektif, efisien dan terpuaskan. Dalam standar ISO 25010 yang mewakili aspek ini adalah *usability*

#### **d. *Performance Testing***

Performance testing mengetes penggunaan CPU/memory, konsumsi baterai, dan pemuatan data dalam server di bawah kondisi tertentu. Dalam standar ISO 25010 yang mewakili aspek ini adalah *performance efficiency*.

Berdasarkan penjelasan diatas dapat dirangkum bahwa perangkat lunak yang baik adalah perangkat lunak yang fokus kepada pelanggan sehingga dibutuhkan pengujian agar terhindar dari kesalahan teknis maupun non teknis. Salah satu standar yang digunakan untuk mengukur kualitas perangkat lunak adalah ISO 25010. ISO 25010 memiliki aspek yaitu *Functional Suitability*, *Reliability*, *Compatibility*, *Security*, *Performance Efficiency*, *Maintainability*, *Usability*, dan *Portability*. Aplikasi Aluradmi hanya akan diuji kualitasnya menggunakan 4 aspek saja di dalam ISO 25010 yaitu *Functional Suitability*, *Compatibility*, *Usability*, dan *Performance Efficiency*. Hal tersebut dikarenakan 4 aspek yang telah disebutkan sudah mencukupi untuk pengujian aplikasi *mobile*.

## **B. Penelitian yang Relevan.**

1. Hasil penelitian dengan judul "Pengembangan Aplikasi "Annuncio" Sebagai Media Untuk Menyebarkan Atau Berlangganan Informasi Kampus" oleh Refany Anhar pada tahun 2014. Tujuan dari penelitian ini adalah untuk merancang aplikasi "Annuncio" sebagai media berlangganan informasi kampus. Hasil dari penelitian ini berupa aplikasi yang bernama Annuncio yang terdiri dari dua bagian yaitu sistem admin merupakan sistem pengelola aplikasi yang berbasis website dan sistem member yang merupakan aplikasi *mobile* bagi *client* dengan *platform* Android.
2. Hasil penelitian dengan judul "Penerapan Algoritma *Dijkstra* pada Aplikasi Pencarian Rute Bus Trans Semarang" oleh Dwi Ardana dan Ragil Saputra yang diseminarkan pada Seminar Nasional Ilmu Komputer (SNIK 2016) dengan nomor ISBN 978-602-1034-40-8 di Semarang pada tanggal 10 Oktober 2016. Tujuan dari penelitian ini adalah menghasilkan aplikasi yang mampu memberikan informasi rute dan penjadwalan Bus Trans Semarang. Hasil dari penelitian ini berupa Aplikasi Pencarian Rute Bus Trans Semarang dengan memanfaatkan Algoritma Dijkstra untuk menentukan *shelter* terdekat dan perpindahan koridor pada *transfer point*.
3. Hasil penelitian dengan judul "Visualisasi Prosedur Administrasi Di Lingkungan Universitas Menggunakan Adobe Flash" oleh Wildan Zulmaniar Noer pada tahun 2015. Tujuan dari penelitian ini adalah membangun aplikasi visual dalam bentuk video yang digunakan untuk penggambaran prosedur-prosedur akademik. Hasil dari penelitian ini berupa aplikasi video yang digunakan

untuk memberi gambaran dari langkah-langkah prosedur yang terdapat dalam kegiatan akademik.

### **C. Kerangka Pikir**

Pengembangan aplikasi ini didasari oleh permasalahan yang timbul di lingkungan kampus. Permasalahannya adalah informasi tentang administrasi mahasiswa tersebut kurang fleksibel. Dapat terjadinya *miss document* administrasi dalam proses pengurusannya. Pihak FT UNY belum memanfaatkan teknologi informasi berbasis *mobile* untuk menyampaikan informasi administrasi mahasiswa.

Berdasarkan observasi yang telah dilakukan penulis, pihak FT UNY telah memberikan 2 solusi untuk menangani masalah tersebut . Pertama, membuat papan pengumuman di masing-masing jurusan dan beberapa titik di lingkungan FT UNY. Kedua, pihak FT UNY memberikan informasi mengenai administrasi melalui website tetapi tidak semua informasi mengenai administrasi mahasiswa ada didalam website tersebut. Untuk itu penulis memberikan solusi tambahan yang memanfaatkan teknologi *mobile* dengan mengembangkan sebuah aplikasi *mobile* untuk menyampaikan informasi terkait admininstrasi mahasiswa. Aluradmi adalah solusi yang ditawarkan penulis untuk mengatasi masalah tersebut.

Aluradmi memiliki beberapa keunggulan dibanding dengan solusi yang sudah ada yaitu:

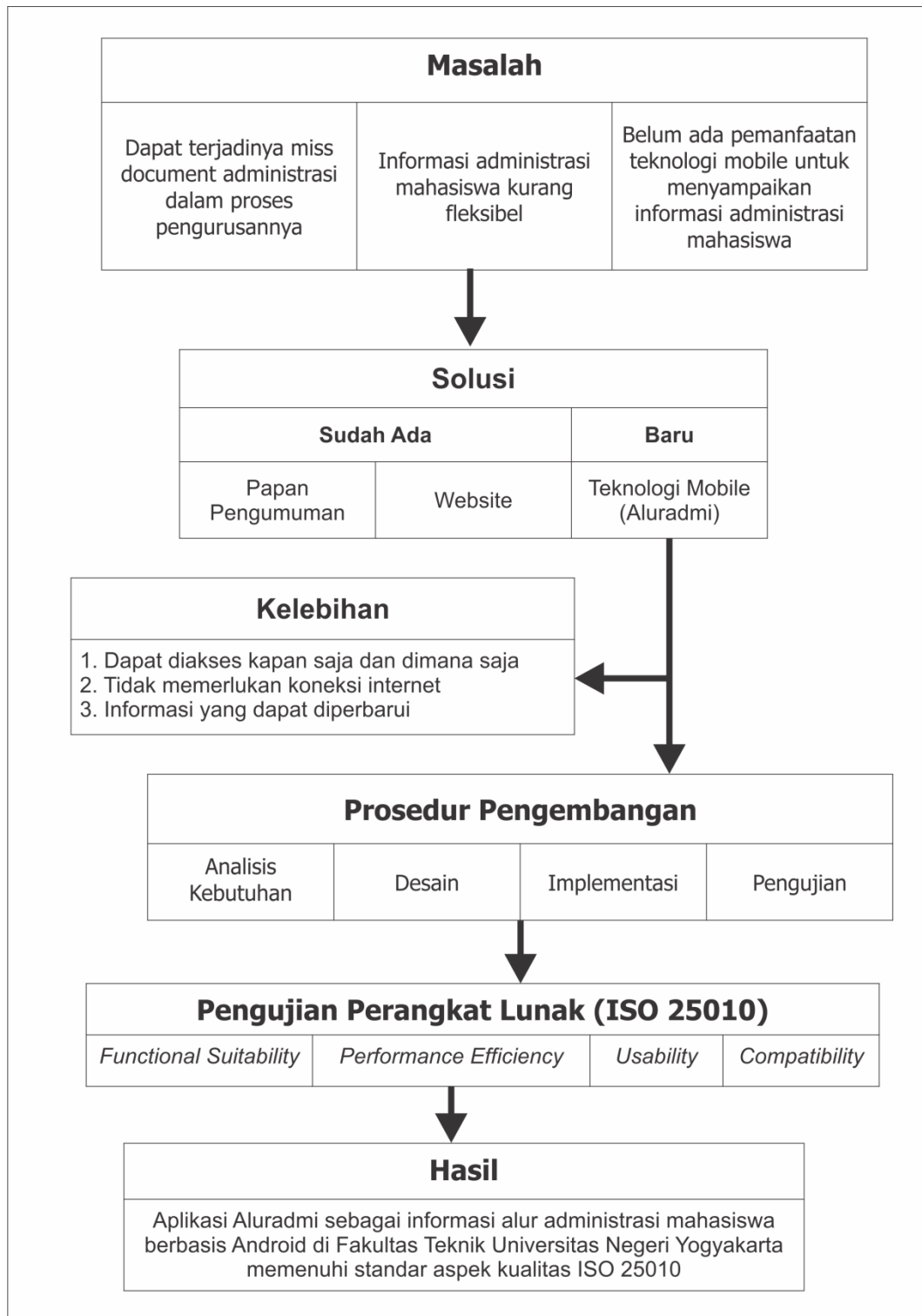
1. Informasi yang dapat diakses kapan saja dan dimana saja.
2. Informasi mengenai administrasi mahasiswa disimpan dalam basis data aplikasi sehingga tidak memerlukan koneksi internet.

3. Informasi yang ada di dalam basis data aplikasi dapat diperbarui dengan melakukan sinkronisasi data dengan basis data server.

Pengembangan aplikasi ini menggunakan pendekatan metode *waterfall*. Tahap-tahap pengembangannya adalah analisis kebutuhan, desain, implementasi dan pengujian. Tahapan pengembangan aplikasi tersebut dilakukan secara terurut atau runtut.

Aplikasi Aluradmi sebagai informasi alur administrasi mahasiswa berbasis Android tersebut diharapkan menjadi aplikasi yang berkualitas tinggi. Untuk itu diperlukan pengujian dengan menerapkan standar kualitas perangkat lunak yang relevan dengan teknologi pada saat ini. Salah satunya adalah standar ISO 25010. Standar tersebut menggantikan standar kualitas perangkat lunak ISO 9126.

Standar ISO 25010 menerapkan delapan karakteristik dalam standar kualitas perangkat lunaknya yaitu *functional suitability, reliability, performance efficiency, usability, maintainability, security, compatibility*, dan *portability* yang dibagi lagi kedalam subkarakteristik seperti pada gambar. Tetapi dalam penelitian ini, peneliti hanya akan menggunakan empat karakteristik saja dalam ISO 25010. Karakteristik tersebut adalah *functional suitability, performance efficiency, compatibility*, dan *usability* karena sudah cukup untuk menentukan kualitas perangkat lunak berbasis *mobile*.



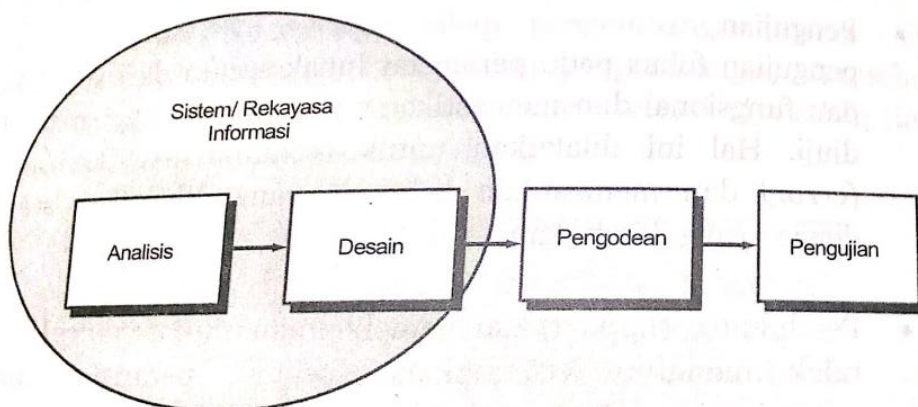
Gambar 10. Bagan Kerangka Pikir

## BAB III

### METODE PENELITIAN

#### A. Model Pengembangan

Penelitian "Pengembangan Aplikasi Aluradmi Sebagai Informasi Alur Administrasi Mahasiswa Berbasis Android di Fakultas Teknik Universitas Negeri Yogyakarta" menggunakan model pengembangan *waterfall*. Tahapan model pengembangan *waterfall* meliputi analisis, desain, pengodean, dan pengujian.



Gambar 11. Model *Waterfall* (Rosa & Shalahuddin, 2013)

#### B. Prosedur Pengembangan

Prosedur pengembangan aplikasi Aluradmi yang mengacu pada model waterfall adalah sebagai berikut:

##### 1. Analisis

Menurut Rosa & Shalahuddin (2013:29) pada tahapan analisis dilakukan proses pengumpulan kebutuhan agar perangkat lunak dapat dispesifikasi seperti yang dibutuhkan oleh *user*. Pada tahap analisis, akan dilakukan proses observasi dan studi literatur untuk mendapatkan informasi untuk keperluan pengembangan aplikasi. Analisis terhadap informasi yang telah didapatkan akan menghasilkan kebutuhan yang berkaitan dengan pengembangan perangkat lunak termasuk

kegunaan dan batasan dari perangkat lunak agar sesuai dengan kebutuhan dari pengguna. Tahapan analisis tersebut terdiri dari analisis fungsional perangkat lunak, analisis desain perangkat lunak, dan lain-lain. Pada tahapan ini diharapkan semua kebutuhan dari perangkat lunak yang akan dikembangkan dapat diketahui.

## **2. Desain**

Menurut Rosa & Shalahuddin (2013:29) pada desain pengembang berfokus pada desain pembuatan perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean. Representasi perangkat lunak yang akan dikembangkan digambarkan dalam sebuah pemodelan. Pemodelan tersebut menggunakan UML (*Unified Modelling Language*). Pada tahapan ini juga akan dilakukan perancangan basis data dan perancangan *user interface* yang disesuaikan dengan kebutuhan pengguna.

## **3. Pengodean**

Tahap pengodean atau biasa disebut implementasi merupakan tahapan penerjemahan desain sistem ke dalam perangkat lunak yang diinginkan sesuai dengan analisis kebutuhan sistem yang telah dilakukan. Tahapan ini bertujuan untuk menghasilkan produk yang siap diuji atau dievaluasi.

## **4. Pengujian**

Tahapan pengujian merupakan tahapan dimana aplikasi yang sudah dikembangkan diuji berdasarkan 4 aspek dari ISO 25010 yang telah disebutkan dalam bab 2. Dalam tahapan ini diharapkan aplikasi layak untuk digunakan oleh pengguna (*user*).



### **C. Subjek, Tempat, dan Waktu Penelitian**

Subjek penelitian digunakan untuk menguji aspek *functional suitability* dan *usability* dari perangkat lunak yang telah dikembangkan. Subjek pengujian aspek *functional suitability* adalah 3 responden dari berbagai profesi di bidang IT yang merupakan ahli dalam pengembangan perangkat lunak. Pengujian *usability* menggunakan 5 mahasiswa untuk setiap jurusan di FT UNY sehingga didapatkan total 30 mahasiswa sebagai responden.

Penelitian akan dilakukan di Fakultas Teknik Universitas Negeri Yogyakarta. Penelitian dimulai pada bulan Juni 2016 sampai dengan bulan Mei 2017.

### **D. Metode Pengumpulan Data**

Terdapat tiga teknik pengumpulan data yang sering digunakan yaitu teknik wawancara, teknik observasi dan teknik kuisisioner (Rosa & Shalahuddin, 2013:19). Metode pengumpulan data digunakan untuk keperluan analisis kebutuhan dan pengujian beberapa aspek kelayakan pada penelitian ini. Berikut adalah metode pengumpulan data yang digunakan dalam penelitian ini:

#### **1. Observasi**

Menurut Hadi yang dikutip oleh Sugiyono(2016:145), observasi merupakan suatu proses kompleks yang terdiri dari berbagai proses biologis dan psikologis dimana dua hal yang terpenting dari proses tersebut adalah proses-proses pengamatan dan ingatan. Teknik observasi dilakukan untuk memperoleh data untuk analisis kebutuhan dan pengujian pada aspek *compatibility* dan *performance efficiency*.

## 2. Kuisioner(angket)

Kuesioner atau angket adalah teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawab oleh responden tersebut. (Sugiyono, 2016:142). Teknik kuisioner digunakan untuk menguji aspek *usability* dan *functional suitability* pada perangkat lunak.

### E. Instrumen Penelitian

#### 1. Instrumen *Functional Suitability*

Instrumen untuk pengujian aspek *functional suitability* berupa test case yang berisi daftar fungsi aplikasi berdasarkan analisis pada kebutuhan fungsional. Instrumen *functional suitability* dapat dilihat pada Tabel 1.

Tabel 1. Instrumen Functional Suitability

No.	Aktivitas/ Menu	Skenario Pengujian	Hasil yang Diharapkan	Lolos	
				Ya	Tidak
1.	Menjalankan aplikasi aluradmi pertama kali	Membuka aplikasi Aluradmi untuk pertama kali dalam keadaan tidak tersambung dengan internet	Halaman peringatan bahwa penyetelan aplikasi membutuhkan koneksi internet sudah tampil dengan benar		
		Membuka aplikasi Aluradmi dalam keadaan tersambung dengan internet	Menampilkan halaman proses pengunduhan data dari server, apabila berhasil mengunduh data akan menuju halaman memilih jurusan		
Selengkapnya dapat dilihat pada Lampiran 2					

Jawaban pada instrumen *functional suitability* menggunakan skala Guttman. Skala Guttman digunakan untuk mendapat jawaban tegas dari suatu

permasalahan yang ditanyakan (Sugiyono, 2016). Setiap jawaban pada instrumen akan diberi skor agar dapat diukur.

Tabel 2. Skor Instrumen *Functional Suitability*

No	Pernyataan	Skor
1.	Ya	1
2.	Tidak	0

## 2. Instrumen *Compatibility*

Pengujian pada aspek *compatibility* dilakukan dengan melakukan instalasi aplikasi Aluradmi pada perangkat Android. Perangkat Android yang digunakan berjumlah 10 dengan spesifikasi minimal versi OS 4.1 (Jelly Bean). Hasil pengujian didokumentasikan lalu dilakukan analisis.

## 3. Instrumen *Usability*

Pengujian aspek *usability* pada aplikasi Aluradmi menggunakan angket *USE Questionnaire* yang dibuat oleh A.M. Lund (2001). *USE Questionnaire* berjumlah 30 pertanyaan yang dibagi ke dalam 4 kriteria yaitu *usefulness*, *ease of use*, *ease of learning*, dan *satisfaction*. Angket tersebut di terjemahkan terlebih dahulu ke dalam Bahasa Indonesia agar mudah dipahami sebelum dibagikan kepada responden. Instrumen *USE Questionnaire* dapat dilihat pada Tabel 3.

Tabel 3. *USE Questionnaire*

No.	Pernyataan
<b><i>USEFULLNESS</i></b>	
1.	Aplikasi ini membantu saya menjadi lebih efektif
2.	Aplikasi ini membantu saya menjadi lebih produktif
3.	Aplikasi ini bermanfaat
Selengkapnya dapat dilihat pada Lampiran 3	

Jawaban pada instrumen *usability* menggunakan skala Likert dengan pilihan jawaban 4. Skala Likert digunakan untuk mengukur persepsi atau sikap seseorang (Sugiyono,2016). Setiap pilihan jawaban diberikan skor agar dapat dilakukan proses analisis. Pemberian skor pada jawaban instrumen *usability* dapat dilihat di tabel 4.

Tabel 4. Skor Instrumen Usability

No	Pernyataan	Skor
1.	Sangat Setuju (SS)	4
2.	Setuju(S)	3
3.	Tidak Setuju (TS)	2
4.	Sangat Tidak Setuju (STS)	1

#### 4. Instrumen Performance Efficiency

Pada aspek ini, dilakukan pengujian dengan menjalankan aplikasi Aluradmi pada perangkat Android lalu dilakukan teknik *profiling* untuk mengetahui penggunaan CPU , penggunaan Memory dan waktu respon aplikasi.

#### F. Teknik Analisis Data

Berikut adalah teknik analisis data yang digunakan untuk pengujian aspek ISO 25010:

##### 1. Analisis Aspek Functional Suitability, Usability dan Compatibility

Proses analisis data pada aspek *functional suitability*, *usability* dan *compatibility* dilakukan dengan analisis deskriptif. Data skor yang didapatkan dari hasil pengujian akan dihitung menggunakan rumus sebagai berikut:

$$Presentase Kelayakan(\%) = \frac{\text{jumlah skor yang didapatkan}}{\text{jumlah skor yang diharapkan}} \times 100\%$$

Hasil yang didapatkan dari perhitungan diatas selanjutnya dikonversikan ke dalam sebuah pernyataan berdasarkan tabel kriteria interpretasi skor seperti yang tersaji dalam Tabel 5 yang telah dilakukan penyesuaian.

Tabel 5. Kriteria Interpretasi Skor (Guritno, Sudaryono, & Rahardja, 2011)

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

## 2. Analisis *Aspek Performance Efficiency*

*Performance efficiency* dinilai dengan melihat seberapa efisien penggunaan *hardware resources* (CPU dan Memory) dan seberapa cepat aplikasi merespon permintaan pengguna (Wagner, 2013). Analisis aspek *performance efficiency* dilakukan dengan cara menghitung rata-rata dari penggunaan *CPU*, penggunaan *Memory*, dan waktu respon yang dibutuhkan aplikasi pada saat menjalankan setiap aktivitas. Pengujian aspek *performance efficiency* dilakukan minimal sejumlah 5 kali (Niknejad, 2011). Untuk hasil pengujian dalam hal waktu respon aplikasi akan dibandingkan dengan tabel kepuasan pengguna terhadap respon waktu.

Tabel 6. Kepuasan terhadap Waktu Respon (Hoxmeier & DiCesare, 2000)

Respon Waktu	Predikat
<3	Sangat Puas
3-9	Puas
9-12	Cukup Puas
>12	Tidak Puas

## **BAB IV**

### **HASIL PENELITIAN DAN PEMBAHASAN**

#### **A. Hasil Penelitian**

##### **1. Analisis**

Analisis kebutuhan adalah tahapan awal dalam proses pengembangan aplikasi Aluradmi. Tahap analisis kebutuhan terdiri dari analisis kebutuhan fungsional, analisis kebutuhan spesifikasi dan analisis kebutuhan perangkat keras dan perangkat lunak.

##### **a. Analisis Kebutuhan Fungsional**

Berdasarkan observasi yang telah dilakukan, didapatkan informasi mengenai administrasi mahasiswa sebagai berikut:

- 1) Terdapat lebih dari 1 administrasi mahasiswa.
- 2) Setiap administrasi mahasiswa memiliki alur masing-masing.
- 3) Setiap jurusan di Fakultas Teknik UNY memiliki alur administrasi mahasiswa yang berbeda.
- 4) Beberapa langkah dalam suatu alur administrasi memiliki rangkaian proses lagi di dalamnya.
- 5) Beberapa alur administrasi memerlukan berbagai macam berkas maupun surat keterangan.
- 6) Beberapa alur memiliki informasi mengenai lokasi tempat mengurus administrasi.
- 7) Alur administrasi mahasiswa dapat berubah sewaktu-waktu.

Melalui uraian data diatas, didapatkan fungsi minimal yang harus dimiliki aplikasi Aluradmi sebagai berikut:

- 1) Pengguna dapat melihat daftar administrasi mahasiswa yang ada di Fakultas Teknik.
- 2) Pengguna dapat melihat alur dan informasi yang berkaitan dengan administrasi mahasiswa.
- 3) Pengguna dapat mengetahui berkas-berkas yang dibutuhkan pada saat mengurus proses administrasi mahasiswa.
- 4) Pengguna dapat mengetahui lokasi untuk mengurus administrasi mahasiswa.
- 5) Pengguna dapat mengubah jurusan.
- 6) Pengguna dapat melakukan *update* informasi mengenai administrasi mahasiswa.
- 7) Pengguna dapat mengetahui rute dari posisi pengguna menuju lokasi mengurus administrasi mahasiswa.
- 8) Pengguna dapat melakukan pencarian informasi alur administrasi mahasiswa.
- 9) Pengguna dapat melihat bantuan penggunaan aplikasi Aluradmi
- 10) Pengguna dapat melihat informasi tentang pengembang aplikasi

#### **b. Analisis Kebutuhan Spesifikasi**

Aplikasi Aluradmi membutuhkan spesifikasi perangkat seperti yang tertera pada Tabel 7.

Tabel 7. Spesifikasi Perangkat

Komponen	Spesifikasi Minimal	Spesifikasi Rekomendasi
Sistem Operasi	Jelly Bean	Lollipop
Versi	4.1	5.0
RAM	256 MB	1 GB
CPU	800 MHz	1.3 GHz

### **c. Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak**

#### **1) Perangkat Keras**

- a) Laptop dengan spesifikasi processor intel i3 dan RAM 4 GB.
- b) Perangkat android dengan sistem operasi *minimum* Jelly Bean (4.1)

#### **2) Perangkat Lunak**

- a) Android Studio 2.0
- b) Android SDK
- c) Star UML
- d) AutoCAD
- e) Adobe Illustrator CS6
- f) MySQL Workbench

## **2. Desain**

Setelah tahapan analisis kebutuhan selesai dilakukan, tahapan selanjutnya adalah tahapan desain. Tahapan desain terdiri dari desain *Unified Modelling Language* (UML) , desain basis data, desain sistem dan desain antarmuka aplikasi.

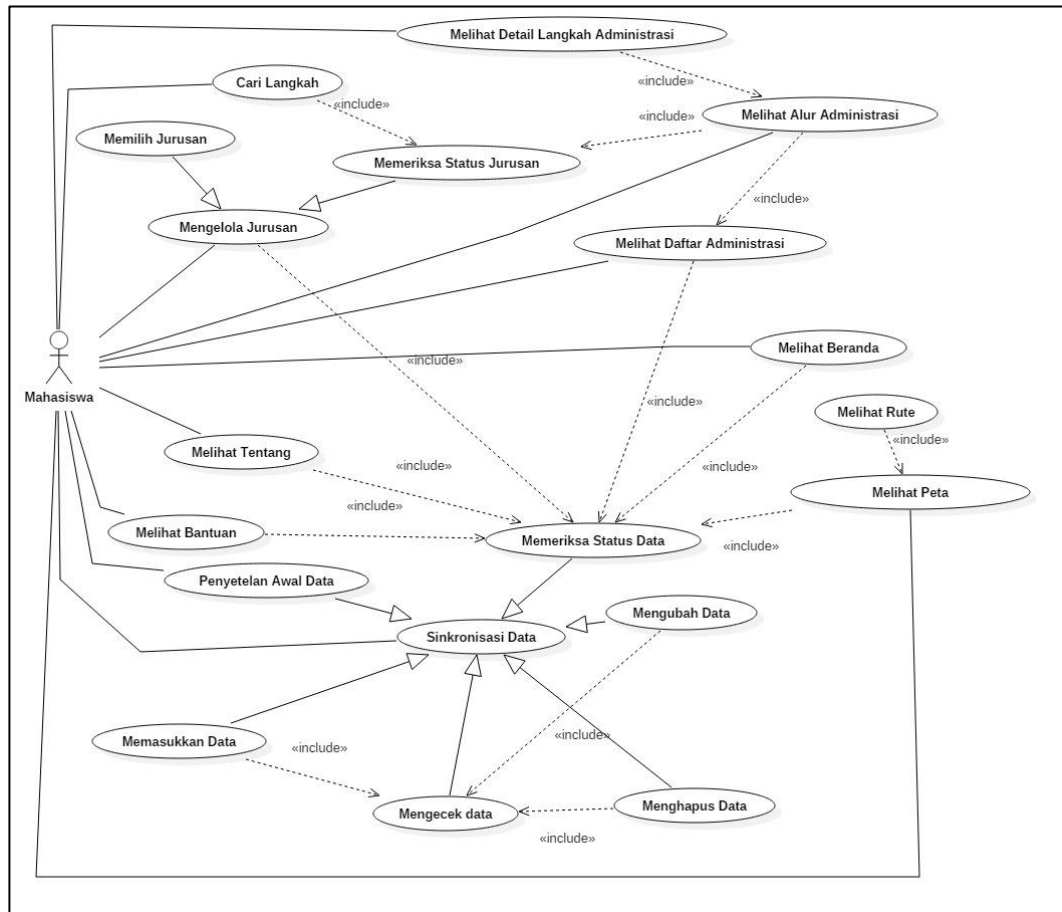
### **a. Desain *Unified Modelling Language* (UML)**

#### **1) *Use Case Diagram***

*Use Case* adalah teknik untuk mengetahui kebutuhan fungsional dengan mendeskripsikan interaksi antara pengguna dengan sistem dan antara sistem itu sendiri (Fowler, 2004). Diagram *Use Case* dapat digunakan untuk menjelaskan fungsi – fungsi yang terdapat di dalam aplikasi beserta siapa saja yang berhak menggunakannya.



Berikut adalah diagram *Use Case* dalam pengembangan aplikasi Aluradmi:



Gambar 12. Diagram Use Case

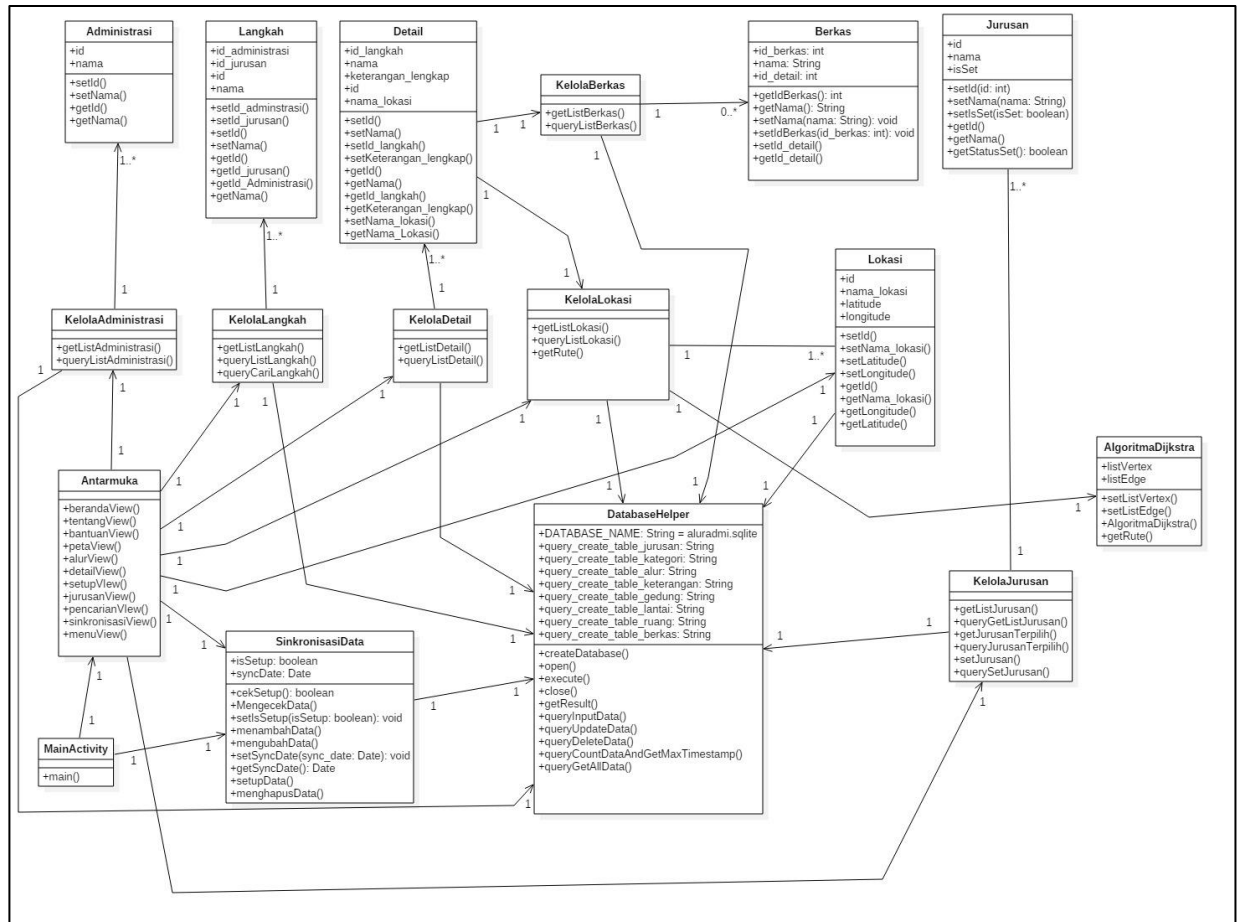
Definisi Aktor, Definisi Use Case dan Skenario Use Case selengkapnya dapat dilihat pada lampiran 4.

## 2) *Class Diagram*

*Class Diagram* digunakan untuk menggambarkan struktur sistem yang akan dikembangkan dengan mendefinisikan kelas yang diperlukan dan hubungan antar kelas tersebut (Fowler, 2004). Berikut adalah *Class Diagram* dari pengembangan aplikasi Aluradmi:

### a) Class Diagram Aplikasi

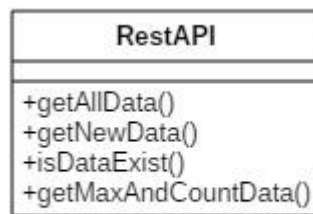
Gambar 13 menunjukkan desain class diagram dari aplikasi Aluradmi. Di dalamnya terdapat 17 kelas yang saling berinteraksi membentuk sebuah sistem guna menampilkan informasi kepada pengguna.



Gambar 13. Class Diagram Aluradmi

### b) Class Diagram Web Service

Gambar 14 merupakan desain *class diagram* dari *web service* yang ada di dalam *server*. Hanya terdapat 1 kelas di dalam diagram tersebut yang memiliki fungsi utama untuk melakukan sinkronisasi data antara *server* dengan aplikasi.



Gambar 14. Class Diagram Web Service

### 3) *Sequence Diagram*

*Sequence Diagram* digunakan untuk menggambarkan interaksi antar objek dan pertukaran pesan yang terjadi antar objek tersebut. *Sequence Diagram* yang digunakan dalam pengembangan aplikasi Aluradmi dapat dilihat pada Tabel 8 (Diagram dapat dilihat pada Lampiran 5):

Tabel 8. Daftar *Sequence Diagram*

No	Nama Sequence Diagram	Fungsi
1.	Penyetelan Awal Data	Berisi interaksi antar kelas dan proses yang terjadi dalam pengunduhan data pada pertama kali menjalankan aplikasi
2.	Beranda	Berisi interaksi antar kelas dan proses tampilnya halaman beranda pada mahasiswa
3.	Tentang	Berisi interaksi antar kelas dan proses tampilnya halaman yang berisi informasi pengembang pada mahasiswa
4.	Bantuan	Berisi interaksi antar kelas dan proses tampilnya halaman bantuan pada mahasiswa
5.	Cari Langkah	Berisi interaksi antar kelas dan proses yang terjadi dalam proses pencarian informasi langkah administrasi.

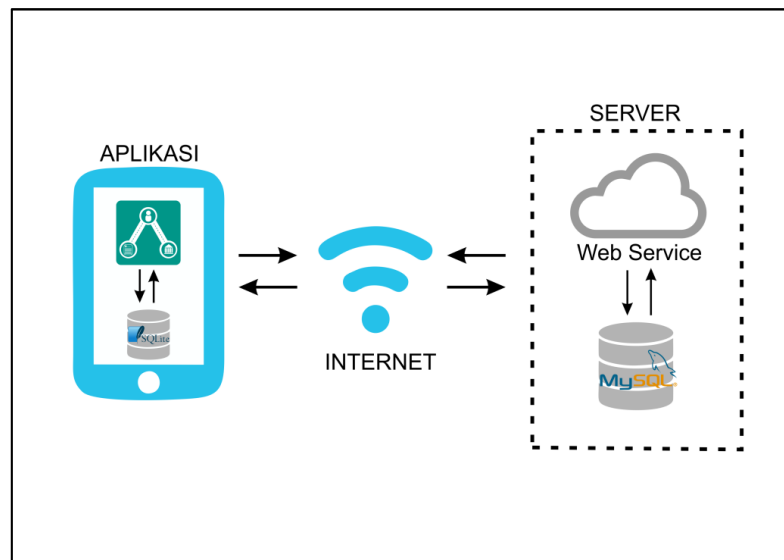
No	Nama Sequence Diagram	Fungsi
6.	Melihat Alur Administrasi	Sequence diagram ini berisi interaksi antar kelas dari proses pengambilan data dari basis data hingga menampilkan informasi alur administrasi ke mahasiswa
7.	Melihat Daftar Administrasi	Interaksi antar kelas yang terdapat dalam sequence diagram ini menjelaskan proses dalam menampilkan informasi daftar administrasi kepada mahasiswa
8.	Melihat Detail Langkah	Interaksi antar kelas dalam sequence diagram ini berfungsi untuk menampilkan informasi detail langkah administrasi setelah mahasiswa memilih salah satu langkah dalam halaman alur administrasi
9.	Melihat Peta	Di dalam sequence diagram ini terdapat interaksi antar kelas dan proses tampilnya halaman peta
10.	Melihat Rute	Sequence diagram ini menjelaskan interaksi antar kelas dan proses dalam menampilkan rute di peta pada mahasiswa
11.	Memilih Jurusan	Interaksi antar kelas dalam diagram ini menunjukkan proses yang terjadi ketika mahasiswa ingin mengganti jurusan
12.	Tambah Data	Interaksi antar kelas dalam diagram ini menangani penambahan data dalam basis data aplikasi
13.	Menghapus Data	Interaksi antar kelas dalam diagram ini menangani penghapusan data dalam basis data aplikasi
14.	Mengubah Data	Interaksi antar kelas dalam diagram ini menangani perubahan data dalam basis data aplikasi

### b. Desain Basis Data

Pengembangan aplikasi Aluradmi menggunakan 2 buah basis data yaitu basis data di dalam aplikasi *mobile* dan basis data di *server*. Perancangan kedua basis data tersebut menggunakan tool yang bernama *MySQL Workbench*. Desain basis data yang digunakan dalam penelitian ini dapat dilihat pada Lampiran 14.

### c. Desain Sistem

Penulis merancang aplikasi Aluradmi agar dapat memperbarui data didalam basis data aplikasinya dengan melakukan proses sinkronisasi dengan basis data *server* melalui internet. Untuk itu dibutuhkan interaksi antara aplikasi dan *server*. Desain sistem aplikasi Aluradmi dapat dilihat pada Gambar 15.



Gambar 15. Desain Sistem

### d. Desain Antarmuka

Berikut adalah desain antarmuka dari aplikasi Aluradmi yang berbentuk wireframe atau kerangka dasar dari tampilan aplikasi (selengkapnya dapat dilihat pada Lampiran 6):

### 1) Halaman Beranda & Menu

Desain halaman beranda dapat dilihat pada Gambar 16. Di dalam desain tersebut terdapat logo UNY, informasi jurusan dan tanggal sinkronisasi data.



Gambar 16. Desain Antarmuka Halaman Beranda

### 2) Menu

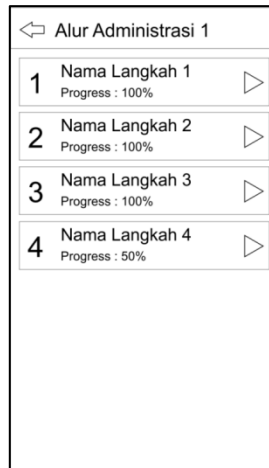
Desain menu dapat dilihat pada Gambar 17. Terdapat 5 menu utama didalamnya yaitu menu beranda, menu daftar administrasi, menu peta, menu bantuan, dan menu tentang.



Gambar 17. Desain Antarmuka Menu

### 3) Halaman Alur Administrasi

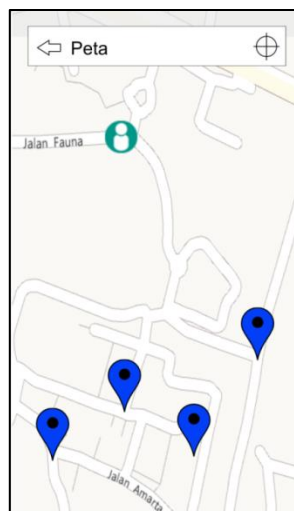
Gambar 18 menunjukkan desain halaman alur administrasi, di dalamnya terdapat judul alurnya, urutan langkah penyelesaian, nama langkahnya dan progress dari masing-masing langkah.



Gambar 18. Desain Antarmuka Halaman Alur Administrasi

### 4) Halaman Peta

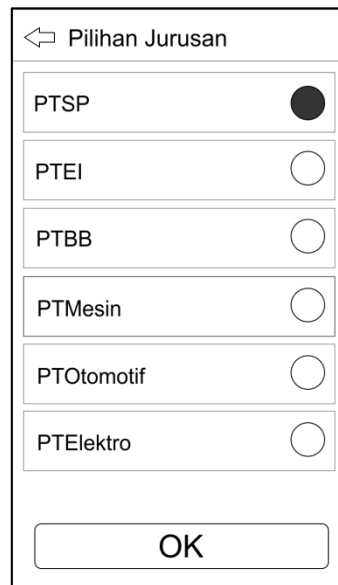
Desain halaman peta dapat dilihat pada Gambar 19. Berisi *marker* lokasi pengguna, tampilan peta, dan *marker* yang menunjukkan lokasi gedung.



Gambar 19. Desain Antarmuka Halaman Peta

### 5) Halaman Jurusan

Desain halaman jurusan dapat dilihat pada Gambar 20. Di dalamnya terdapat 6 jurusan dan tombol "OK" untuk memilih jurusan.



Pilihan Jurusan	
PTSP	<input checked="" type="radio"/>
PTEI	<input type="radio"/>
PTBB	<input type="radio"/>
PTMesin	<input type="radio"/>
PTOtomotif	<input type="radio"/>
PTElektro	<input type="radio"/>
<input type="button" value="OK"/>	

Gambar 20. Desain Antarmuka Halaman Jurusan

## 3. Implementasi

Tahap implementasi merupakan tahap selanjutnya setelah tahap desain yang telah dilakukan. Pada tahap implementasi, desain yang sudah dibuat diwujudkan ke dalam bentuk perangkat lunak. Tahap implementasi terdiri dari implementasi sistem, implementasi algoritma pemrograman, implementasi antarmuka dan implementasi basis data.

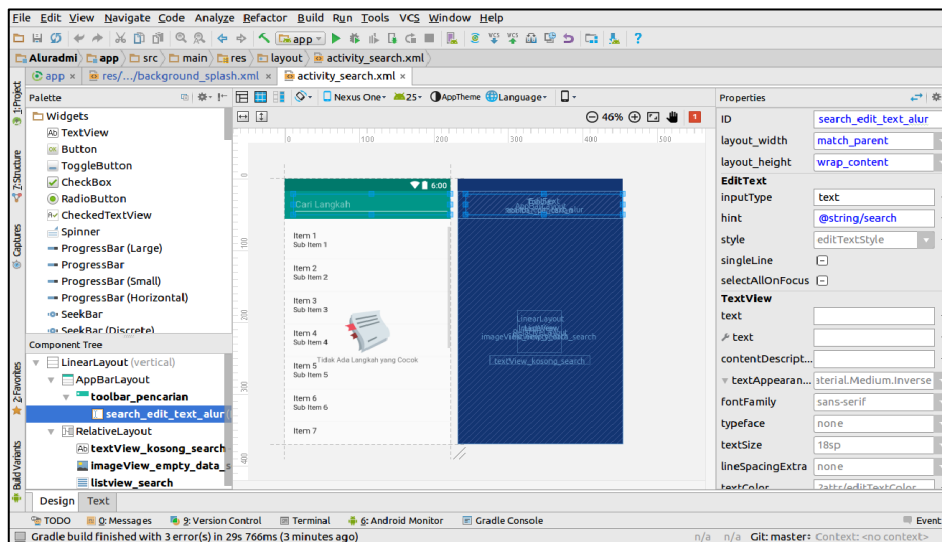
### a. Implementasi Sistem

Perangkat lunak yang digunakan dalam proses implementasi sistem aplikasi Aluradmi menggunakan *Integrated Development Environment* (IDE) Android Studio 2.2 yang dikembangkan oleh Google.



## 1) Implementasi *layout* aplikasi

Tahap implementasi *layout* merupakan tahap mewujudkan desain yang telah diwujudkan ke dalam bentuk *UI/UX* dalam aplikasi. Dalam pengembangannya, *layout* aplikasi dibuat dengan Bahasa XML. Aplikasi Aluradmi menggunakan *material design* sebagai dasar pembuatan *layout* sesuai aturan dari Google. *Designer Tool* yang disediakan oleh Android Studio digunakan dalam pembuatan *layout* aplikasi Aluradmi. *Workspace Designer Tool* dari Android Studio dapat dilihat pada gambar dibawah:

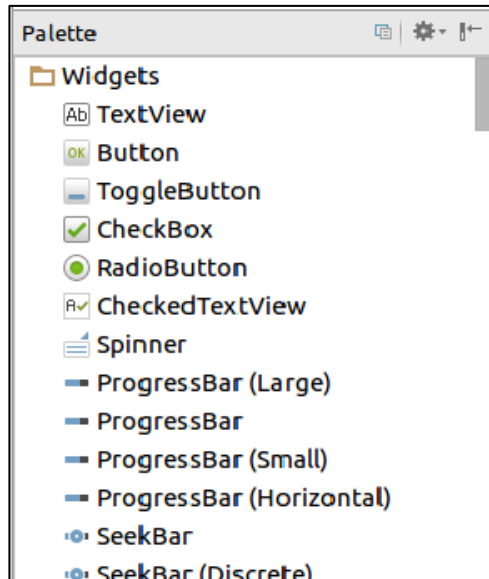


Gambar 21. *Workspace Designer Tool* Android Studio

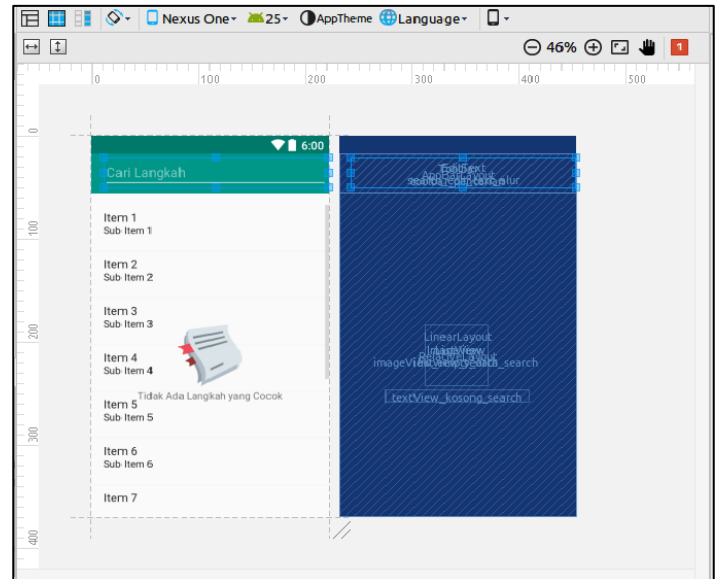
Di dalam pengembangan *layout* aplikasi Aluradmi menggunakan 2 cara yaitu dengan menggunakan *Design View* dan *Text View* pada *Designer Tool*.

### a) *Design View*

*Design View* digunakan untuk membuat *layout* statis yang digunakan dalam aplikasi Aluradmi. Cara penggunaan adalah dengan *drag and drop* komponen *layout* di dalam *Palette* yang dibutuhkan ke dalam *Layout View* yang dapat dilihat langsung hasilnya.

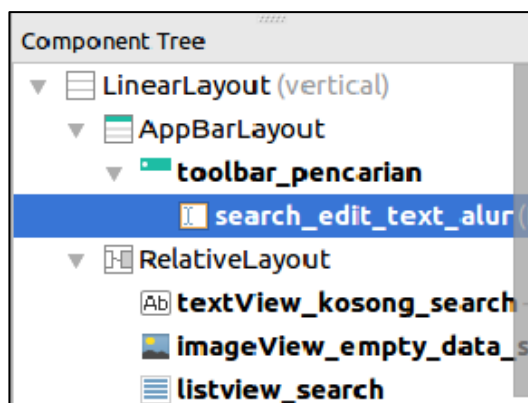


Gambar 22. Palette

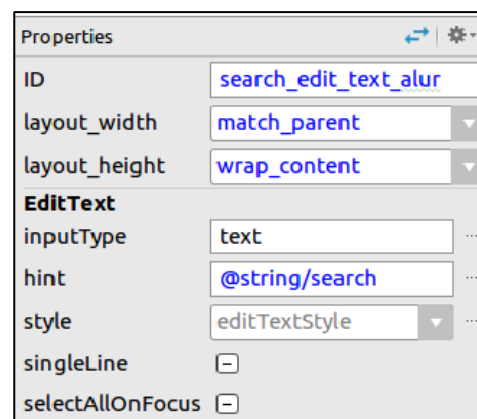


Gambar 23. Layout View

Komponen *layout* tersebut dapat dirubah susunannya dengan merubah susunan pada *tool Component Tree*. Komponen layout juga dapat dimodifikasi dengan menggunakan *Properties tool*.



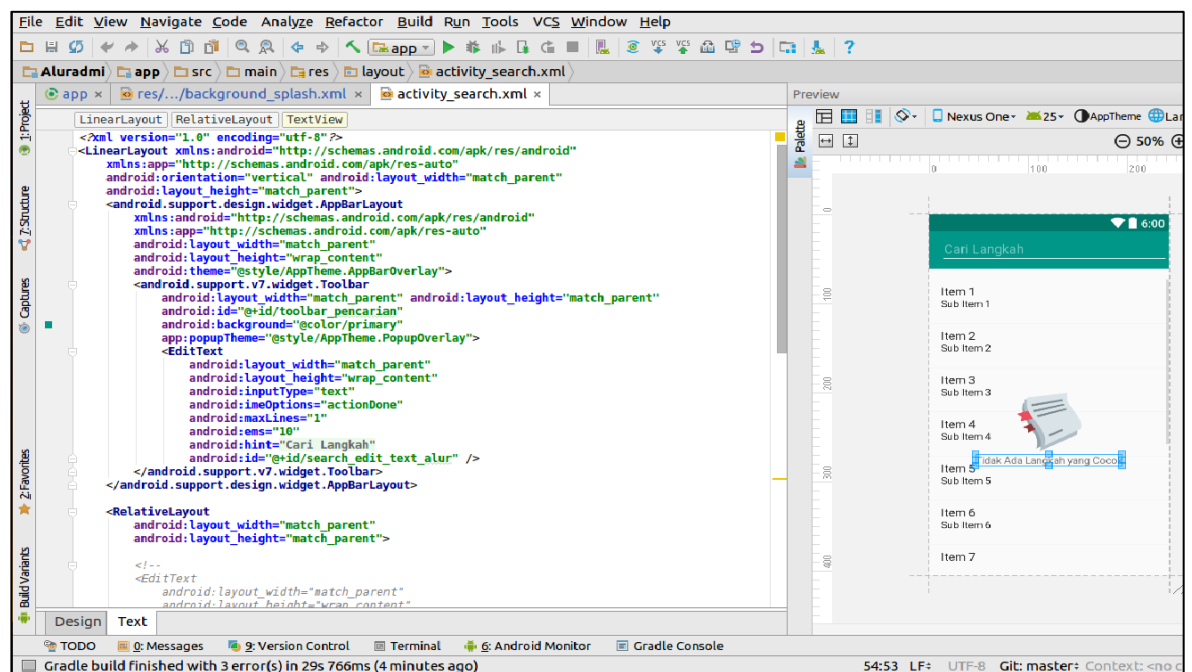
Gambar 24. Component Tree



Gambar 25. Properties

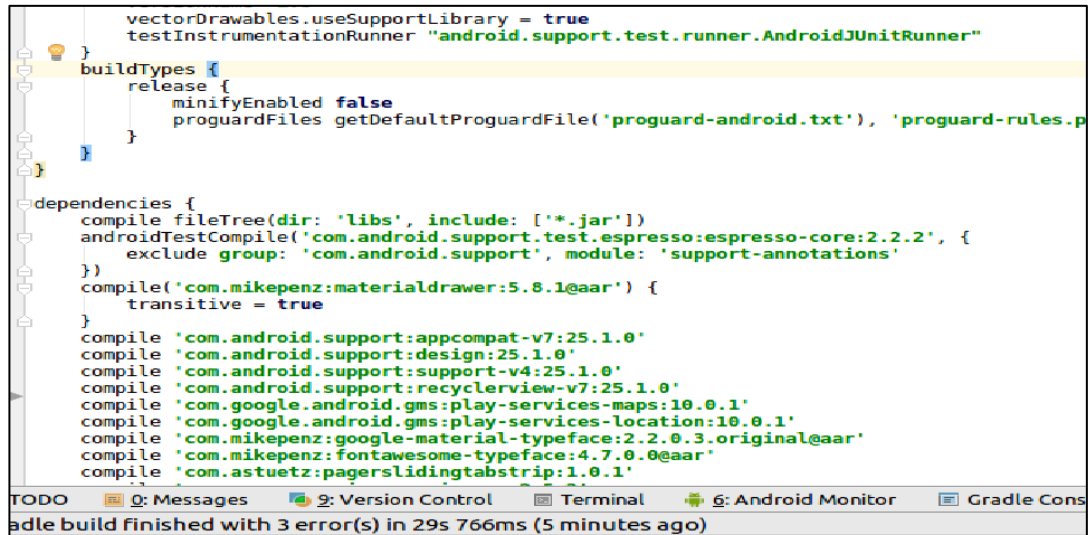
## b) *Text View*

Metode yang digunakan dalam mendesain *layout* menggunakan *Text View* adalah dengan menuliskan kode XML yang mendefinisikan komponen *layout*. Di dalam mode *Text View* juga terdapat *Layout View* jadi tidak perlu untuk berpindah ke dalam mode *Design View* untuk melihat hasil dari modifikasi *layout* yang dibuat.



Gambar 26. *Text View*

Apabila komponen *layout* yang ada didalam *Palette* di rasa kurang, maka dapat menambahkan komponen yang di dapatkan dari internet. Penambahan komponen tersebut melalui dependensi *file Gradle* aplikasi untuk selanjutnya dilakukan sinkronisasi agar dapat digunakan.



Gambar 27. Gradle

*Layout* yang digunakan dalam aplikasi Aluradmi dapat dilihat pada Tabel 9.

Tabel 9. Daftar *Layout*

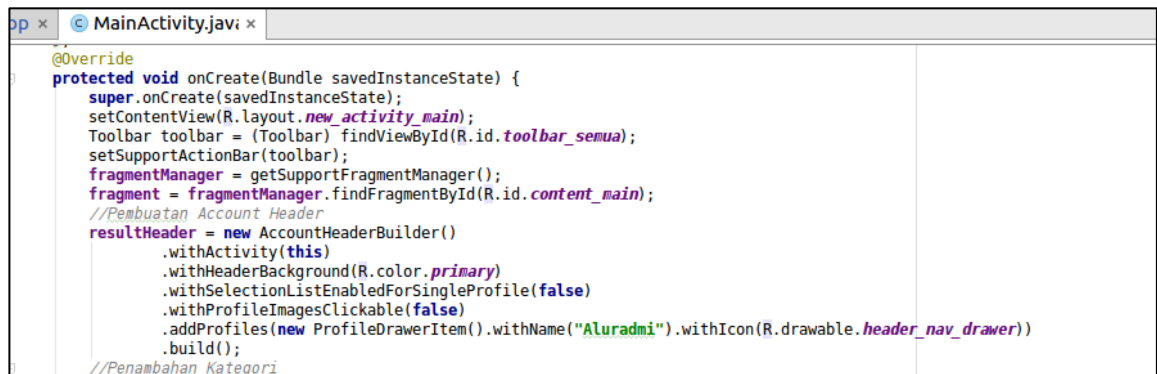
No	Nama	Role
1.	activity_bantuan_lengkap_baru.xml	Layout untuk menampilkan halaman bantuan lengkap
2.	Activity_bantuan_singkat.xml	Layout untuk menampilkan halaman panduan singkat
3.	Activity_fragment.xml	Layout yang digunakan untuk menempatkan halaman alur dan detail
4.	Activity_fullscreen.xml	Layout untuk menampilkan halaman denah
5.	Activity_maps2.xml	Layout untuk menampilkan halaman peta
6.	Activity_pengaturan.xml	Layout untuk menampilkan halaman pengaturan
Selengkapnya dapat dilihat pada Lampiran 7		

## 2) Implementasi Fitur Aplikasi

Pengembangan fitur aplikasi menggunakan Bahasa pemrograman Java.

*Layout* yang telah dibuat sebelumnya dapat dirubah menjadi dinamis melalui java

dengan menghubungkan basis data dan *layout* statis yang telah dibuat. Berikut adalah potongan kode program yang dikembangkan:



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.new_activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar_semua);
    setSupportActionBar(toolbar);
    fragmentManager = getSupportFragmentManager();
    fragment = fragmentManager.findFragmentById(R.id.content_main);
    //Pembuatan Account Header
    resultHeader = new AccountHeaderBuilder()
        .withActivity(this)
        .withHeaderBackground(R.color.primary)
        .withSelectionListEnabledForSingleProfile(false)
        .withProfileImagesClickable(false)
        .addProfiles(new ProfileDrawerItem().withName("Aluradmi").withIcon(R.drawable.header_nav_drawer))
        .build();
    //Penambahan Kategori

```

Gambar 28. Potongan Kode Program

*File* java yang dibuat dalam pengembangan aplikasi Aluradmi dapat dilihat pada Tabel 10.

Tabel 10. Daftar File Java

No	Nama File	Fungsi
1.	MainActivity.java	Aktivitas utama yang menangani menu, pergantian antara halaman beranda dan about.
2.	AlurListActivity.java	Aktivitas yang menangani halaman alur administrasi.
3.	KeteranganPagerActivity.java	Aktivitas yang menangani halaman detail langkah administrasi.
4.	AlurSearchActivity.java	Aktivitas yang menangani halaman pencarian langkah administrasi.
5.	JurusanActivity.java	Aktivitas yang menangani halaman pilihan jurusan.
Selengkapnya dapat dilihat pada Lampiran 8		

### 3) Implementasi Algoritma Pemrograman

Aplikasi Aluradmi menggunakan dua buah algoritma dalam pembuatan rute di dalam Fakultas Teknik Universitas Negeri Yogyakarta. Algoritma yang digunakan yaitu Algoritma Dijkstra yang digunakan untuk mencari rute terpendek dari posisi pengguna ke gedung tujuan dan Formula Haversine untuk mengukur bobot atau jarak dari titik yang saling terhubung di dalam grafik yang akan digunakan dalam Algoritma Dijkstra.

Berikut ini adalah implementasi dari kedua algoritma yang telah disebutkan diatas:

#### a) Formula Haversine

Proses menentukan bobot atau jarak antara dua titik yang saling terhubung di dalam sebuah grafik yang digunakan dalam penentuan rute di aplikasi Aluradmi adalah sebagai berikut:

##### (1) Persyaratan

Formula haversine dapat digunakan untuk menentukan bobot atau jarak antara dua titik apabila kedua titik tersebut memiliki koordinat. Koordinat tersebut terdiri dari *latitude* (garis lintang bumi) dan *longitude* (garis bujur bumi) yang telah dideklarasikan sebelum dilakukan perhitungan.

##### (2) Perhitungan

Perhitungan dengan formula haversine diawali dengan pendeklarasian dua buah koordinat. Titik koordinat yang dideklarasikan dapat dilihat dalam Tabel 11.

Tabel 11. Contoh Koordinat

Koordinat A		Koordinat B	
<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>
-7.768979	110.3879	-7.769133	110.3877

Setelah dilakukannya pendeklarasian titik koordinat, kemudian konversikan koordinat tersebut ke dalam radian seperti pada Tabel 12.

Tabel 12. Hasil Konversi

Koordinat A		Koordinat B	
<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>
-0.13559426307	1.9266323094	-0.13559695088	1.9266288187

Selanjutnya dua titik tersebut dimasukkan de dalam rumus perhitungan. Perhitungan disesuaikan menggunakan Bahasa pemrograman dengan radius bumi 6731 KM.

Delta Latitude =  $(-0.13559695088) - (-0.13559426307) = -0.00000268781$

Delta Longitude =  $1.9266288187 - 1.9266323094 = 0.0000034907$

Jarak =  $6371 * 2 * \text{ASIN}(\text{SQRT}(\text{POWER}(\text{SIN}(0.00000268781 / 2), 2) + \text{COS}(-0.13559426307) * \text{COS}(-0.13559695088) * \text{POWER}(\text{SIN}(0.0000034907 / 2), 2)))$

Jarak koordinat A dan koordinat B = 0,027 km atau 27 meter

### (3) Implementasi Program

Implementasi formula haversine pada program terdapat dalam salah satu metode kelas Edge di aplikasi Aluradmi. Tampilan kode program yang menggunakan formula haversine dapat dilihat pada Gambar 29.

```

public static double haversine(double lat1, double lon1, double lat2, double lon2) {
    double dLat = Math.toRadians(lat2 - lat1);
    double dLon = Math.toRadians(lon2 - lon1);
    lat1 = Math.toRadians(lat1);
    lat2 = Math.toRadians(lat2);

    double a = Math.pow(Math.sin(dLat / 2), 2) + Math.pow(Math.sin(dLon / 2), 2) * Math.cos(lat1) * Math.cos(lat2);
    double c = 2 * Math.asin(Math.sqrt(a));
    return R * c;
}

```

Gambar 29. Formula Haversine

b) Algoritma Dijkstra

Proses pemilihan rute dari posisi pengguna ke gedung tempat mengurus administrasi adalah sebagai berikut:

(1) Persyaratan

Untuk menggunakan Algoritma Dijkstra sebagai pemilihan rute terlebih dahulu perlu disiapkan sebuah grafik dengan titik-titik yang saling terhubung namun tidak memiliki arah serta memiliki jarak yang positif antar titik tersebut. Titik – titik tersebut memiliki koordinat yang terdiri dari latitude dan longitude.

(2) Penentuan Rute

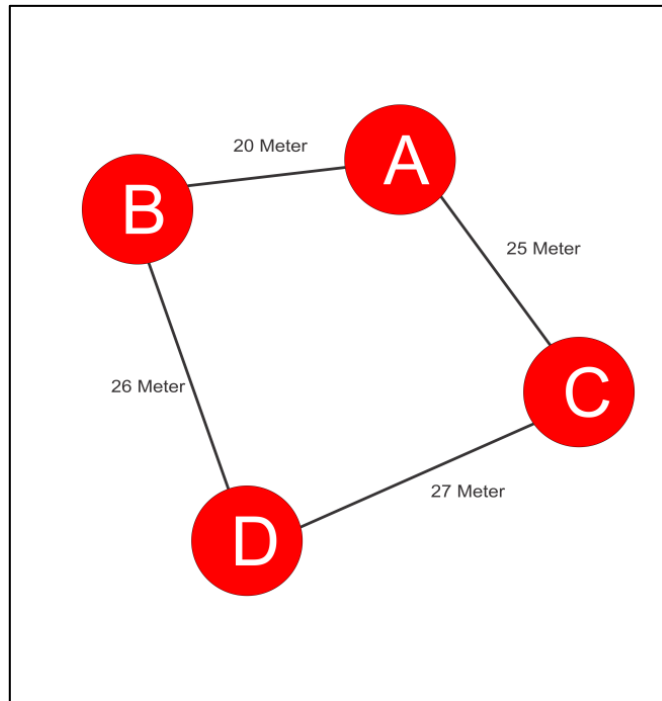
Hal yang dilakukan pertama kali dalam penentuan rute menggunakan algoritma Dijkstra adalah dengan mendeklarasikan titik-titik atau node didalam sebuah grafik. Pendeklarasian titik – titik tersebut dapat dilihat dalam Tabel 13.

Tabel 13. Contoh Titik di Dalam Grafik

<b>Nama Titik (Node)</b>	<b>Latitude</b>	<b>Longitude</b>
A	-7.76864445327121	110.38788914680481
B	-7.768702920282386	110.38772016763687
C	-7.768827827051751	110.38803398609161
D	-7.768923500296765	110.38780868053436

Selanjutnya adalah dengan menghubungkan titik – titik tersebut dan memberikan bobot atau jarak antar titik yang saling terhubung. Perhitungan jarak antar kedua titik menggunakan formula haversine yang sudah dijelaskan sebelumnya. Grafik selengkapya dapat dilihat pada Gambar 30.



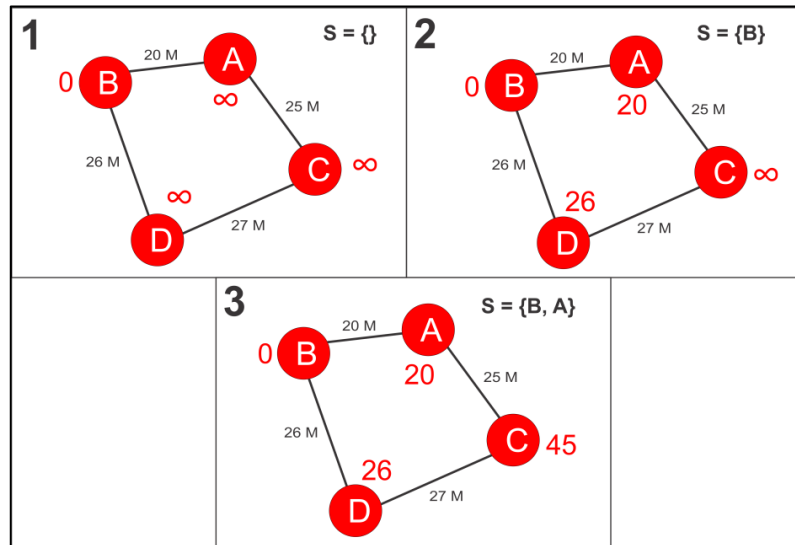


Gambar 30. Contoh Grafik

Berikut adalah cara penggunaan Algoritma Dijkstra untuk menentukan rute terpendek dari titik B menuju titik C:

1. Berikan label nilai pada titik A dengan 0 dan titik yang lain dengan nilai  $\infty$  (maksimal). Inisialisasi variable S sebagai tempat titik-titik yang terpilih.
2. Masukkan titik yang tidak ada di dalam S namun memiliki label nilai terkecil ke dalam S yaitu titik A.
3. Tambahkan jarak titik A dengan B ke dalam label titik B. Rubah label nilai titik A dengan nilai yang dimiliki titik B karena memiliki nilai yang lebih besar daripada label titik B. Lakukan juga pada titik D yang terhubung dengan titik B.
4. Ulangi langkah 2 dengan memasukkan titik A ke dalam S.
5. Ulangi langkah 3 dengan terhadap titik-titik terhubung dengan A yaitu titik C.

6. S akan menunjukkan rute yang akan dilalui dan label nilai C akan menunjukkan jarak yang ditempuh.



Gambar 31. Cara Kerja Algoritma Dijkstra

### (3) Implementasi Program

Lokasi titik yang digunakan dalam aplikasi Aluradmi dapat dilihat pada Lampiran 10. Implementasi titik-titik yang digunakan dalam aplikasi Aluradmi dapat dilihat pada Gambar 32.

```

Aluradmi | app | src | main | java | com | spp | banu | aluradmi | MapsActivity
└─ MainActivity.java | Edge.java | MapsActivity.java
pointToUseInRoute = new ArrayList<>();
//menggambar titik persimpangan node

nodes.add(new Vertex("1", new LatLng(-7.76864445327121, 110.38788914680481)));
nodes.add(new Vertex("2", new LatLng(-7.768702920282386, 110.38772016763687)));
nodes.add(new Vertex("3", new LatLng(-7.768827827051751, 110.38803398609161)));
nodes.add(new Vertex("4", new LatLng(-7.768923500296765, 110.38780868053436)));
nodes.add(new Vertex("5", new LatLng(-7.768979309679639, 110.38797497749329)));
nodes.add(new Vertex("6", new LatLng(-7.769029803876789, 110.38819491863251)));
nodes.add(new Vertex("7", new LatLng(-7.769016515930754, 110.38755655288696)));
nodes.add(new Vertex("8", new LatLng(-7.769133449841402, 110.387744307518)));
nodes.add(new Vertex("9", new LatLng(-7.769223807840961, 110.3881761431694)));
nodes.add(new Vertex("10", new LatLng(-7.769348714455395, 110.38810640573502)));
nodes.add(new Vertex("11", new LatLng(-7.769399208608119, 110.38764506578445)));
nodes.add(new Vertex("12", new LatLng(-7.769497539309186, 110.38795620203018)));
nodes.add(new Vertex("13", new LatLng(-7.769566636544804, 110.38787841796875)));
nodes.add(new Vertex("14", new LatLng(-7.769641048939684, 110.38749754428864)));
nodes.add(new Vertex("15", new LatLng(-7.769654336865949, 110.38732320070267)));
nodes.add(new Vertex("16", new LatLng(-7.769805819195691, 110.3874546289444)));
nodes.add(new Vertex("17", new LatLng(-7.769808476779934, 110.38769602775574)));
nodes.add(new Vertex("18", new LatLng(-7.769803161611436, 110.38782745599747)));
nodes.add(new Vertex("19", new LatLng(-7.769896177050572, 110.38778990507126)));
nodes.add(new Vertex("20", new LatLng(-7.769811134364164, 110.38801789283752)));
nodes.add(new Vertex("21", new LatLng(-7.7699519863041635, 110.38796693086624)));

```

Gambar 32. Implementasi Node

Kode program yang digunakan untuk menghubungkan titik-titik yang telah dibuat dapat dilihat pada Gambar 33.

```

//menghubungkan node tersebut dikurangi 1
edges = new ArrayList<>();

addLane("1-2", 0, 1);
addLane("1-3", 0, 2);

addLane("2-1", 1, 0);
addLane("2-4", 1, 3);
addLane("2-7", 1, 6);

addLane("3-1", 2, 0);
addLane("3-5", 2, 4);

addLane("4-5", 3, 4);
addLane("4-2", 3, 1);
addLane("4-7", 3, 6);
addLane("4-8", 3, 7);

addLane("5-3", 4, 2);
addLane("5-4", 4, 3);
addLane("5-6", 4, 5);

addLane("6-5", 5, 4);
addLane("6-9", 5, 8);

```

Gambar 33. Menghubungkan Node

Perhitungan Algoritma Dijkstra diimplementasikan ke dalam sebuah kelas yang bernama DijkstraAlgoritm. Implementasinya dapat dilihat pada Gambar 34.

```

private Set<Vertex> unSettledNodes;
private Map<Vertex, Vertex> predecessors;
private Map<Vertex, Double> distance;

public DijkstraAlgoritm(Graph graph) {
    // create a copy of the array so that we can operate on this array
    this.nodes = new ArrayList<>(graph.getVertexes());
    this.edges = new ArrayList<>(graph.getEdges());
}

public void execute(Vertex source) {
    settledNodes = new HashSet<>();
    unSettledNodes = new HashSet<>();
    distance = new HashMap<>();
    predecessors = new HashMap<>();
    distance.put(source, 0.0);
    unSettledNodes.add(source);
    while (unSettledNodes.size() > 0) {
        Vertex node = getMinimum(unSettledNodes);
        settledNodes.add(node);
        unSettledNodes.remove(node);
        findMinimalDistances(node);
    }
}

```

Gambar 34. Implementasi Algoritma Dijkstra

## **b. Implementasi Antarmuka**

Berikut adalah beberapa hasil antarmuka yang telah dikembangkan untuk aplikasi Aluradmi (selengkapnya dapat dilihat pada Lampiran 9):

### **1) Halaman Beranda**

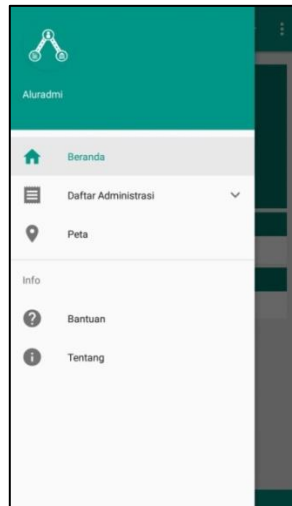
Gambar 35 merupakan hasil implementasi antarmuka untuk halaman beranda. Di dalam halaman beranda terdapat ucapan selamat datang, jurusan yang dipilih oleh pengguna dan tanggal terakhir kali aplikasi melakukan sinkronisasi data dengan server.



Gambar 35. Antarmuka Halaman Beranda

### **2) Tampilan Menu**

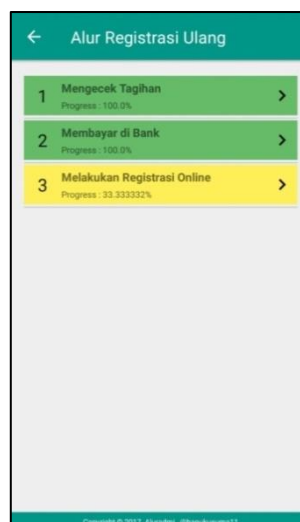
Gambar 36 merupakan hasil implementasi antarmuka untuk menu utama di dalam aplikasi. Tampilan menu sesuai dengan yang telah di desain dalam tahapan sebelumnya.



Gambar 36. Antarmuka Menu

### 3) Halaman Alur Administrasi

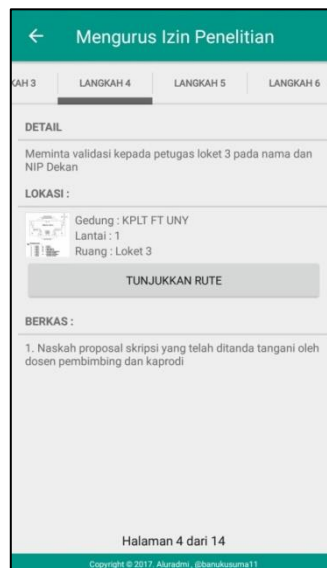
Gambar 37 merupakan hasil implementasi antarmuka untuk halaman alur administrasi. Di dalam halaman alur administrasi terdapat nomor yang menunjukkan urutan langkah dalam penyelesaian administrasi, terdapat keterangan progress atau kemajuan di dalam setiap langkah, dan warna latar belakang tiap langkah akan berubah sesuai dengan kemajuan yang telah dilakukan oleh pengguna.



Gambar 37. Antarmuka Halaman Alur Administrasi

#### 4) Halaman Detail Langkah Administrasi

Gambar 38 merupakan hasil implementasi antarmuka untuk halaman detail langkah administrasi. Halaman detail langkah administrasi dapat diakses ketika pengguna menekan salah satu langkah di dalam halaman alur administrasi. Halaman detail langkah administrasi berisi penjelasan lengkap mengenai langkah administrasi tersebut seperti lokasi dan berkas yang harus dimiliki. Terdapat fitur checkbox "selesai" di dalam halaman ini. Apabila pengguna menekan tombol selesai maka di halaman alur administrasi progress langkah tersebut akan bertambah.



Gambar 38. Antarmuka Halaman Detail Langkah Administrasi

#### 5) Halaman Tentang

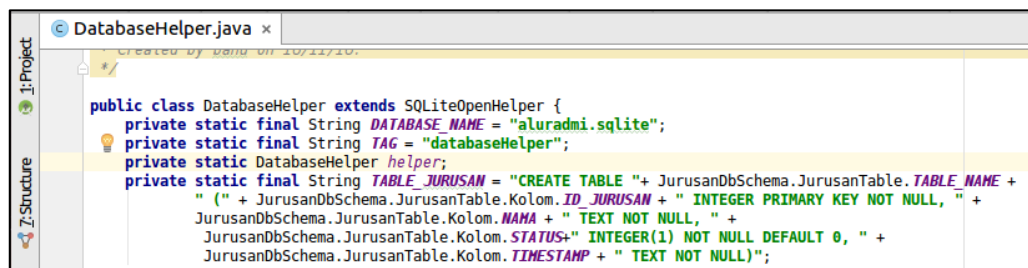
Gambar 39 merupakan hasil implementasi antarmuka dari halaman tentang. Halaman tentang berisi informasi pengembang dan informasi dimana pengguna dapat memberikan kritik dan saran terhadap aplikasi Aluradmi.



Gambar 39. Antarmuka Halaman Tentang

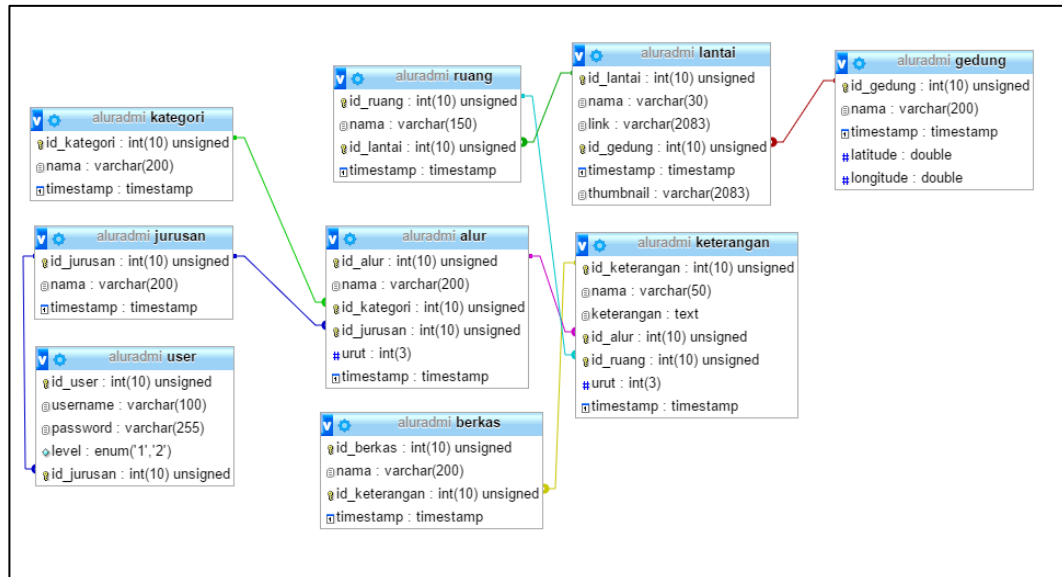
### c. Implementasi Basis Data

Basis data aplikasi Aluradmi menggunakan SQLite. Pendefinisian tabel dan relasi yang digunakan dalam aplikasi melalui sebuah kelas java yang bernama DatabaseHelper. Pendefinisian tabel dan relasi antar tabel dapat dilihat pada Gambar 40.



Gambar 40. Implementasi Basis Data Aplikasi

Sementara itu basis data di dalam *server* menggunakan MySQL. Implementasinya dapat dilihat pada Gambar 41.



Gambar 41. Implementasi Basis Data Server

#### 4. Pengujian

Tahapan yang terakhir adalah tahapan pengujian. Tahapan pengujian dimaksudkan untuk mengetahui kelayakan aplikasi berdasarkan 4 aspek dalam standar ISO 25010. Pengujian yang akan dilakukan antara lain pengujian *usability*, pengujian *functional suitability*, pengujian *compatibility* dan pengujian *performance efficiency*.

##### a. Pengujian Usability

Pengujian aspek *usability* aplikasi Aluradmi dilakukan kepada 30 mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta. Pengujian *usability* menggunakan instrumen *USE Questionnaire* yang memiliki 30 pertanyaan. Kuisisioner *usability* diberikan kepada mahasiswa masing-masing jurusan yang ada di FT UNY dengan rincian setiap jurusan 5 responden sehingga total mendapat 30 responden mahasiswa. Ringkasan hasil pengujian aspek *usability* dapat dilihat pada Tabel 14 (Hasil keseluruhan dapat dilihat dalam Lampiran 11).



Tabel 14. Ringkasan Hasil Pengujian *Usability*

No	Jawaban	Jumlah	Skor	Jumlah Skor
1.	Sangat Setuju	322	4	1288
2.	Setuju	523	3	1569
3.	Tidak Setuju	51	2	102
4.	Sangat Tidak Setuju	4	1	4
Nilai Total				2963
Nilai Maksimal				3600

Berdasarkan ringkasan hasil pengujian usability pada Tabel 14, dapat diperoleh presentase kelayakan sebagai berikut:

$$\begin{aligned}
 \text{Presentase Kelayakan (\%)} &= \frac{\text{skor yang diobservasi}}{\text{skor yang diharapkan}} \times 100\% \\
 &= \frac{2963}{3600} \times 100\% \\
 &= 82,2\%
 \end{aligned}$$

Presentase kelayakan dari pengujian *usability* adalah 82,2%, sehingga dapat disimpulkan bahwa aplikasi Aluradmi memiliki nilai "**Sangat Layak**" pada aspek *usability*.

#### **b. Pengujian *Functional Suitability***

Pengujian aspek *functional suitability* menggunakan lembar *test-case* yang berisi 32 fungsi yang harus diuji. Tiga orang ahli yang memiliki berbagai profesi di dalam bidang IT dan memahami pengembangan perangkat lunak yang menjadi penguji fungsi aplikasi Aluradmi. Daftar penguji fungsi aplikasi Aluradmi dapat dilihat pada Tabel 15.

Tabel 15. Penguji *Functional Suitability*

No	Nama	Profesi	Instansi
1.	Nur Hasanah, S.T, M.Cs.	Dosen	Universitas Negeri Yogyakarta
2.	David Surya Aji S S.Pd.	Back End Developer	PT. Erporate
3.	Ofani Dariyan S.Pd	Web Developer	PT. Digital Skyshi Indonesia

Hasil pengujian dari *functional suitability* dapat dilihat pada Tabel 16.

Tabel 16. Hasil pengujian Functional Suitability

No	Nama Penguji	Fungsi yang lolos uji	Fungsi yang tidak lolos uji
1.	Nur Hasanah, S.T, M.Cs.	32	0
2.	David Surya Aji S S.Pd.	32	0
3.	Ofani Dariyan S.Pd	32	0
<b>Total</b>		96	0

Berikut adalah persentase kelayakan yang didasarkan pada hasil pengujian *functional suitability* yang tertera pada Tabel 16:

$$\begin{aligned}
 \text{Persentase Kelayakan (\%)} &= \frac{\text{skor yang didapatkan}}{\text{skor maksimal}} \times 100\% \\
 &= \frac{96}{96} \times 100\% \\
 &= 100\%
 \end{aligned}$$

Hasil persentase kelayakan dalam aspek *functional suitability* adalah 100%, sehingga dapat dikatakan bahwa seluruh fungsi dalam aplikasi Aluradmi dapat berjalan dengan baik. Oleh karena itu, dapat disimpulkan aplikasi Aluradmi memiliki nilai **"Sangat Layak"** pada aspek *functional suitability*.

### c. Pengujian *Compatibility*

Pengujian *compatibility* dilakukan dengan dengan cara menginstall aplikasi Aluradmi pada perangkat *smartphone* Android dan menjalankannya. *Smartphone* Android yang digunakan berjumlah 10 perangkat dengan spesifikasi minimal untuk melakukan penginstallan aplikasi Aluradmi. Daftar *smartphone* Android yang digunakan dalam pengujian *compatibility* dapat dilihat dalam Tabel 17.

Tabel 17. Daftar *Smartphone* Pengujian *Compatibility*

No	Perangkat	Versi OS	CPU	RAM	Ukuran Layar
1.	Samsung Galaxy Ace 3	4.2	1 Ghz	1 GB	4"
2.	Lenovo A6010	5.0	1.2 Ghz	2 GB	5"
3.	Lenovo P70	4.4	1.7 Ghz	2 GB	5"
4.	Samsung Galaxy J5	5.1	1.2 Ghz	1.5 GB	5"
5.	Oppo A39	5.1	1.5 Ghz	3 GB	5.2"
6.	Oppo Joy R1001	4.2	1.3 Ghz	512 MB	4"
7.	Asus Zenfone 4C	4.3	1.2 Ghz	1 GB	4"
8.	Asus Zenfone 2 Laser	5.0	1.2 Ghz	2 GB	5"
9.	Xiaomi Mi 4C	5.1.1	1.44 Ghz	2 GB	5"
10.	Sony Experia T3	4.4.2	1.4 Ghz	1 GB	5.3"

Ringkasan hasil pengujian aspek *compatibility* dapat dilihat pada Tabel 18 (hasil pengujian selengkapnya terdapat pada Lampiran 13).

Tabel 18. Ringkasan Hasil Pengujian Compatibility

No	Pengujian	Jumlah Skor	Berhasil	Gagal
1.	Instalasi aplikasi pada perangkat	10	10	0
2.	Menjalankan aplikasi pada perangkat	10	10	0
<b>Total</b>		20	20	0

Setelah mendapatkan hasil pengujian, dilakukan perhitungan presentase kelayakan sebagai berikut:

$$\begin{aligned}
 \text{Presentase Kelayakan (\%)} &= \frac{\text{skor yang didapatkan}}{\text{skor maksimal}} \times 100\% \\
 &= \frac{20}{20} \times 100\% \\
 &= 100\%
 \end{aligned}$$

Hasil presentase kelayakan dari pengujian aspek *compatibility* adalah 100%, sehingga dapat disimpulkan aplikasi Aluradmi memiliki nilai "**Sangat Layak**" pada aspek *compatibility*.

#### d. Pengujian *Performance Efficiency*

Pengujian *performance efficiency* dari aplikasi Aluradmi terdiri dari 2 hal yaitu menghitung rata-rata penggunaan *resource* (*CPU* dan *Memory*) dan menghitung rata-rata waktu respon aplikasi dalam menjalankan setiap aktifitas. *Tool* yang digunakan dalam pengujian ini adalah *CPU Monitor*, *Memory Monitor*, dan *Traceview* yang ada di dalam Android Studio.

Pengujian dilakukan dengan menjalankan aplikasi Aluradmi pada perangkat pengujian yang tersambung ke dalam laptop menggunakan kabel USB. Perangkat tersebut tersambung dengan koneksi internet HSDPA yang memiliki kecepatan unduh 1,8 Mbps dan kecepatan unggah 1,1 Mbps dari *provider* IndosatOoredoo. Spesifikasi perangkat yang digunakan dalam pengujian dapat dilihat pada Tabel 19.

Tabel 19. Spesifikasi Perangkat Pengujian

Perangkat	CPU	RAM	Versi OS
Oppo Joy R1001	1.3 Ghz	512 MB	4.1.1 (Jelly Bean)

#### 1) Pengujian Penggunaan *Resource*

Hasil pengujian *performance efficiency* aplikasi Aluradmi dalam penggunaan *resource CPU* dan *Memory* dapat dilihat pada Tabel 20 (selengkapnya dapat dilihat pada Lampiran 12).

Tabel 20. Hasil Pengujian Penggunaan Resource

No	Pengujian	CPU	Memory
1.	Pengujian 1	38%	17,92 MB
2.	Pengujian 2	33%	22,09 MB
3.	Pengujian 3	29%	20,85 MB
4.	Pengujian 4	50%	26,92 MB
5.	Pengujian 5	42%	24,00 MB
<b>Rata-rata</b>		38,4%	22,36 MB

## 2) Pengujian Waktu Respon

Hasil pengujian dari waktu respon aplikasi Aluradmi dapat dilihat pada Tabel 21.

Tabel 21. Hasil Pengujian Waktu Respon

No	Aktifitas / Tugas	Waktu Respon (detik)				
		Uji 1	Uji 2	Uji 3	Uji 4	Uji 5
1.	Halaman Beranda	2,3	2,5	2,7	2,8	2,5
2.	Halaman Penyetelan Awal	7,3	3,7	6,6	6,8	6,6
3.	Halaman Bantuan	3,8	3,8	4,3	4,4	4,1
4.	Halaman Tentang	3,9	4,3	3,9	4	3,5
5.	Halaman Detail Langkah Administrasi	3	3,1	2,4	2,5	3
6.	Halaman Pengaturan	4,8	4	3,7	3,7	4,2
7.	Halaman Peta	10	16,4	8	3,6	3,8
8.	Tunjukkan Rute	10,8	10,4	8,2	3,7	4,3
9.	Halaman Denah	7,1	5,1	7	4,4	4,7
10.	Sinkronisasi Data	7,8	7,6	7,2	8,2	6,6
11.	Pencarian Langkah	6,6	6,2	5,4	5,9	5,6
12.	Halaman Pilihan Jurusan	3,8	4,4	4,2	4,3	4,4
Rata-rata		6,5	4	6,33	4	3
<b>Rata-rata waktu respon</b>		<b>4,76</b>				

Berdasarkan pengujian penggunaan resource yang tertera pada Tabel 20, didapatkan hasil berupa aplikasi Aluradmi rata-rata menggunakan *resource CPU* sebesar 38,4% dan *resource Memory* sebesar 22,36 MB. Berdasarkan hasil pengujian penggunaan *resource* aplikasi Aluradmi tidak memerlukan *resource* yang besar untuk menjalankan fungsinya sehingga **Layak** digunakan. Sementara itu, aplikasi Aluradmi memiliki rata-rata waktu respon sebesar detik untuk

menjalankan setiap aktifitas nya seperti yang tertera pada Tabel 21. Hasil rata-rata waktu respon aplikasi Aluradmi mendapatkan predikat "**Puas**" berdasarkan tabel kepuasan pelanggan terhadap waktu respon.

## **B. Pembahasan**

Aluradmi adalah aplikasi berbasis Android yang memberikan informasi mengenai alur administrasi mahasiswa di lingkungan Fakultas Teknik Universitas Negeri Yogyakarta (FT UNY). Aplikasi Aluradmi telah melalui tahapan pengembangan yang meliputi analisis kebutuhan, desain, implementasi dan pengujian.

Tahap analisis dilakukan di FT UNY dengan melakukan observasi terhadap pengumuman yang berhubungan dengan administrasi mahasiswa di papan pengumuman maupun website yang berada di lingkungan FT UNY. Informasi tersebut kemudian dianalisis untuk mendapatkan kebutuhan yang akan digunakan dalam proses desain.

Tahapan kedua adalah desain. Tahapan ini terdiri dari pemodelan sebagai acuan dalam pembuatan aplikasi , desain basis data, desain antarmuka aplikasi dan desain sistem yang akan dikembangkan.

Tahapan ketiga yaitu tahap implementasi, desain yang telah dibuat diwujudkan ke dalam bentuk aplikasi. Tahap implementasi meliputi implementasi layout, implementasi fungsi, implementasi algoritma pemrograman dan implementasi basis data.

Tahapan terakhir yaitu tahapan pengujian. Tahapan ini dilakukan pada aplikasi yang sudah siap untuk diuji. Pengujian yang dilakukan antara lain pengujian *usability*, pengujian *functional suitability* , pengujian *compatibility* dan

pengujian *performance efficiency*. Keempat jenis pengujian tersebut didasarkan pada aspek yang ada di dalam standar ISO 25010. Hasil dari pengujian dapat dilihat pada Tabel 22.

Tabel 22. Hasil Seluruh Pengujian

No	Aspek	Hasil	Predikat
1.	Usability	Aplikasi Aluradmi mendapatkan skor 82,2% setelah diujikan pada 30 mahasiswa di FT UNY	<b>Sangat Layak</b>
2.	Functional Suitability	Seluruh fungsi berjalan 100% berdasarkan pengujian yang dilakukan 3 ahli di berbagai bidang IT.	<b>Sangat Layak</b>
3.	Compatibility	Aplikasi Aluradmi dapat di install dan dijalankan pada 10 Smartphone Android. Smartphone pengujian memiliki ukuran layar, versi OS, CPU dan RAM yang berbeda-beda dan memenuhi spesifikasi minimum untuk menginstall aplikasi Aluradmi	<b>Sangat Layak</b>
4.	Performance Efficiency	Aplikasi Aluradmi menggunakan rata-rata CPU sebesar 38,4% dan <i>Memory</i> sebesar 22,36 MB. Sehingga aplikasi Aluradmi tidak memerlukan <i>resource</i> yang besar untuk menjalankannya.	<b>Layak</b>
		Aplikasi Aluradmi memiliki rata-rata waktu respon sebesar 4,76 detik dalam menjalankan setiap aktifitas	<b>Puas</b>



## BAB V

### KESIMPULAN & SARAN

#### A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Penelitian ini menghasilkan aplikasi Aluradmi yang berisi informasi mengenai alur administrasi, tempat mengurus administrasi, dan berkas yang dibutuhkan bagi mahasiswa FT UNY. Informasi yang dimiliki oleh aplikasi Aluradmi tersinkronisasi dengan *server* melalui *web service* sehingga informasi yang terdapat dalam aplikasi dapat diperbarui.
2. Aplikasi Aluradmi telah melalui serangkaian pengujian berdasarkan 4 aspek ISO 25010 yaitu *functional suitability*, *usability*, *compatibility*, dan *performance efficiency*. Pada pengujian *functional suitability*, aplikasi Aluradmi dinyatakan **sangat layak** dikarenakan 100% fungsi yang terdapat dalam aplikasi berjalan dengan baik. Pada pengujian *usability* didapatkan nilai sebesar 82,2% dan dinyatakan **sangat layak**. Pada pengujian *compatibility*, aplikasi Aluradmi dapat diinstall dan dijalankan pada seluruh perangkat pengujian sehingga dapat dinyatakan sangat layak. Terakhir pada pengujian *performance efficiency* dalam hal penggunaan *resource*, aplikasi Aluradmi rata-rata memerlukan *resource CPU* sebesar 38,4% dan *Memory* sebesar 22,36 MB sehingga dapat dinyatakan **layak** dikarenakan tidak terlalu memerlukan *resource* yang besar. Pengujian *performance efficiency* dalam hal waktu respon, aplikasi memerlukan rata-rata waktu sebesar 4,76 detik dan dinyatakan dengan predikat "**Puas**". Berdasarkan pengujian yang telah

dilakukan, maka dapat disimpulkan bahwa aplikasi Aluradmi memenuhi standar 4 aspek dari ISO 25010 dan layak untuk digunakan.

## **B. Keterbatasan Produk**

Aplikasi yang telah dikembangkan memiliki keterbatasan produk sebagai berikut:

1. Rute yang menggunakan Algoritma Dijkstra sebagai dasarnya hanya tersedia di lingkungan Fakultas Teknik Universitas Negeri Yogyakarta. Rute diluar Fakultas Teknik Universitas Negeri Yogyakarta menggunakan *Google Direction API*.
2. Informasi administrasi mahasiswa yang lengkap hanya terdapat pada 2 jurusan yaitu Jurusan Pendidikan Teknik Elektronika dan Informatika, dan Jurusan Pendidikan Teknik Sipil dan Perencanaan.
3. Pengingat kemajuan penyelesaian administrasi aplikasi Aluradmi tidak disimpan didalam basis data *server*, hanya disimpan dalam basis data aplikasi

## **C. Pengembangan Produk Lebih Lanjut**

Pengembangan aplikasi selanjutnya adalah sebagai berikut:

1. Menambahkan rute diluar Fakultas Teknik Universitas Negeri Yogyakarta.
2. Menambahkan penanda informasi baru atau informasi telah diperbarui agar memudahkan pengguna dalam membedakan informasi yang terdapat dalam aplikasi.
3. Memperluas informasi administrasi bagi mahasiswa bukan hanya di Fakultas Teknik namun di seluruh fakultas di Universitas Negeri Yogyakarta.
4. Perbaiki tampilan untuk aplikasi Aluradmi.

#### **D. Saran**

Saran dari penulis adalah sebagai berikut:

1. Pengembangan aplikasi disarankan tidak dilakukan oleh 1 orang saja melainkan sebuah tim sehingga pengembangan aplikasi menjadi efektif , efisien serta memiliki hasil yang maksimal.
2. Untuk Fakultas Teknik Universitas Negeri Yogyakarta agar menggunakan aplikasi Aluradmi dalam menyampaikan informasi mengenai administrasi mahasiswa.
3. Pengembangan aplikasi *mobile* diharapkan tidak hanya berfokus pada platform Android melainkan *platform* lainnya seperti IOS dan Windows Phone juga diperhatikan sehingga aplikasi yang dikembangkan dapat digunakan oleh lebih banyak orang.

## DAFTAR PUSTAKA

- A.S, Rosa, & Shalahuddin, M. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Agustina, Gede D., Darmawiguna, I Gede M., & Sunarya, I Made G. (2017). *Developing An Android Based Application As Balinese Shadow Puppet Educational Media*. Jurnal Pendidikan Teknologi dan Kejuruan. 23(III). Hlm. 242.
- Ardana, Dwi, & Saputra, Ragil. (2016). *Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang*. Prosiding, Seminar Nasional. Semarang: DIPK Undip.
- Anupriya, & Saxena, Mansi. (2014). *An Android Application for Google Maps Navigation System Implementing Travelling Salesman Problem*. International Journal of Computer & Organization Trends. 3(IV). Hlm. 70.
- Baidu. (2015). *Indonesian Online News Consumption Study*. Jakarta: Baidu.
- Brummelen, Glen Van. (2013). *Heavenly Mathematics The Forgotten Art of Spherical Trigonometry*. New Jersey: Princeton University Press.
- David, A.B. (2011). *Mobile Application Testing Best Practices to Ensure Quality*. Diakses pada tanggal 12 Januari 2017 pukul 16.11 WIB dari [http://www.globaltelecomsbusiness.com/pdf/AMDOCS%20WHITEPAPER\\_%20Mobile%20application%20testing%20whitepaper.pdf](http://www.globaltelecomsbusiness.com/pdf/AMDOCS%20WHITEPAPER_%20Mobile%20application%20testing%20whitepaper.pdf)
- Dennis, Alan, Wixom, B.H., & Roth, R.M. (2012). *System Analysis and Design Fifth Edition*. New Jersey: John Wiley & Sons, Inc.
- Feiler, Jesse. (2015). *Introducing SQLite for Mobile Developers*. New York: Appress.
- Fowler, Martin. (2004). *UML Distilled Third Edition*. Boston: Pearson Education, Inc.
- Google. (2017). *Dashboard*. Diakses dari <https://developer.android.com/about/dashboards/index.html> pada tanggal 22 Februari 2017, jam 14.57.
- Guritno S, Sudaryono, & Raharja, U. (2011). *Theory and Application of IT Research*. Yogyakarta: Andi.
- Hariyanto, Didik. (2008). *Pengembangan Sistem Informasi Akademik Mahasiswa Berbasis Teknologi WAP (Wireless Application Protocol) Di Jurusan Pendidikan Teknik Elektro FT UNY*. Jurnal Pendidikan Teknologi dan Kejuruan. 17(II). Hlm. 143-144.

- Hoaxmeier, John A., & Dicesare, Chris. (2000). *System Response Time and User Satisfication: An Experimental Study of Browser-based Applications*. AMCIS 2000 Proceedings. Hlm. 347.
- Irmawati, Dessy, & Indrihapsari, Yuniar. (2014). *Sistem Informasi Kearsipan untuk Meningkatkan Kualitas Pelayanan*. *Jurnal Pendidikan Teknologi dan Kejuruan*. 22(II). Hlm. 136.
- Kreibich, Jay A. (2010). *Using SQLite*. California: O'Reilly Media, Inc.
- Laudon, Kenneth C., & Laudon, Jane P. (2012). *Management Information Systems*. New Jersey: Pearson Education, Inc.
- Lee, Wei-Meng. (2012). *Beginning Android 4 Application Development*. Indiana: John Wiley & Sons, Inc.
- Lund, A.M. (2001). *Measuring Usability with the USE Questionnaire*. STC and Usability SIG Newslettter.
- Marbun, S.F. (2003). *Peradilan Administrasi dan Upaya Administrasi di Indonesia*. Yogyakarta: Liberty.
- Niknejad, A. (2011). *A Quality Evaluation of an Android Smartphone Application*. Sweden: University of Gothenburg.
- O'Brien, James A., & Marakas, George M. (2010). *Introduction to Information Systems*. New York: McGraw-Hill Companies, Inc.
- Pressman, Roger S. (2015). *Software Engineering: A Practicioner's Approaches Eight Edition*. New York: McGraw-Hill Companies, Inc.
- Riski, Abduh. (2015). *Pensejajaran Rantai DNA Menggunakan Algoritma Dijkstra*. *prosiding*, Seminar Nasional. Yogyakarta: FMIPA UNY.
- Rosen, Kenneth H. (2012). *Discrete Mathematics and Its Applications Seventh Edition*. New York: McGraw-Hill Companies, Inc.
- Salz, Peggy A., & Moranz, Jennifer. (2013). *The Everything Guide To Mobile Apps*. Massachusetts: Adams Media.
- SBMPTN. (2015). *Daftar Prodi UNY*. Diakses dari <https://sbmptn.ac.id/index.php?mid=14&ptn=46> pada tanggal 7 April 2017, jam 15.06 WIB.
- Schmuller, Joseph. (2004). *Sams Teach Yourself UML in 24 Hours Third Edition*. Indiana: Sams Publising.

- Singal, Pooja, & Chhillar, R.S. (2014). *Dijkstra Shortest Path Algorithm using Global Positioning System*. International Journal of Computer Applications. Vol. 101(No-6). Hlm. 12.
- SQLite. (2017). *Home Page*. Diakses dari <https://www.sqlite.org/> pada tanggal 10 Januari 2017, jam 10.30.
- StatCounter. *Mobile Operating System Market Share Indonesia Maret*. Diakses dari <http://gs.statcounter.com/os-market-share/mobile/indonesia> pada tanggal 10 Maret 2017, jam 15.30 WIB.
- Sugiyono. (2016). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- Svennerberg, Gabriel. (2010). *Beginning Google Maps API 3*. New York: Appress.
- Syafri, Wirman. (2012). *Studi Tentang Administrasi Publik*. Bandung: Erlangga.
- Wagner, Stefan. (2013). *Software Product Quality Control*. Berlin: Springer.

# LAMPIRAN

## Lampiran 1. Surat Keputusan Dosen Pembimbing

**KEPUTUSAN DEKAN  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
NOMOR : 112/ELK/Q-1/VI/2016  
TENTANG  
PENGANGKATAN 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.

**MEMUTUSKAN**

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

Nama Pembimbing : **Adi Dewanto, M.Kom**  
Bagi mahasiswa :  
Nama/No. Mahasiswa : **Banu Kusuma /12520244039**  
Jurusan/Prodi : Pendidikan Teknik Elektronika / Pendidikan Teknik Informatika  
Judul Skripsi : *Pengembangan Aplikasi Aluradmi sebagai Media Informasi Alur Administrasi Mahasiswa Berbasis Android Di Fakultas Teknik Universitas Negeri Yogyakarta*

Kedua : Dosen pembimbing disertai 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.

Dibuatkan : di Yogyakarta  
Pada tanggal : 6 Juni 2016  
Dekan  
*[Signature]*  
Dr. Roch Bruri Triyono  
16560216 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

Gambar 42. Surat Keputusan Dosen Pembimbing



## Lampiran 2. Instrumen Uji *Functional Suitability*

No.	Aktivitas/ Menu	Skenario Pengujian	Hasil yang Diharapkan	Lolos	
				Ya	Tidak
1.	Membuka aplikasi aluradmi pertama kali	Membuka aplikasi Aluradmi untuk pertama kali dalam keadaan tidak tersambung dengan internet	Halaman peringatan bahwa penyetelan aplikasi membutuhkan koneksi internet sudah tampil dengan benar		
		Membuka aplikasi Aluradmi dalam keadaan tersambung dengan internet	Menampilkan halaman proses pengunduhan data dari server, apabila berhasil mengunduh data akan menuju halaman memilih jurusan		
2.	Memilih jurusan pertama kali	Memilih salah satu jurusan yang tersedia dan menekan tombol "OK"	Halaman memilih jurusan sudah tampil dan dapat berfungsi dengan benar		
3.	Panduan Singkat	Menekan tombol selanjutnya	Fungsi untuk berpindah halaman ke halaman selanjutnya sudah berfungsi dengan benar		
		Menekan tombol sebelumnya	Fungsi untuk berpindah halaman ke halaman sebelumnya sudah berfungsi dengan benar		
		Menekan tombol "Ulangi Lagi" pada halaman terakhir panduan singkat	Menampilkan halaman awal dari panduan singkat		
		Menekan tombol "Saya Pahami" pada halaman terakhir panduan singkat	Menampilkan halaman beranda dari aplikasi aluradmi		
4.	Beranda	Memilih menu "Beranda" pada menu utama	Menampilkan halaman beranda yang berisi ucapan selamat datang, jurusan yang telah dipilih dan tanggal terakhir kali proses sinkronisasi dilakukan dengan benar		

No.	Aktivitas/ Menu	Skenario Pengujian	Hasil yang Diharapkan	Lolos	
				Ya	Tidak
5.	Daftar Administrasi	Memilih menu "Daftar Administrasi" pada menu utama	Menampilkan daftar administrasi yang tersedia dengan benar		
		Memilih salah satu menu administrasi yang tersedia	Menampilkan halaman alur dari administrasi yang telah dipilih dengan benar		
6.	Peta	Memilih menu "Peta" pada menu utama	Menampilkan halaman peta, marker lokasi gedung dan marker lokasi pengguna dengan benar		
		Menekan ikon lokasi	Menampilkan marker yang menunjukkan lokasi pengguna		
		Menekan salah satu marker yang menunjukkan lokasi gedung dan menekan info window dari marker yang ditekan	Menampilkan rute pada peta dari posisi pengguna menuju gedung yang dipilih dengan benar		
7.	Bantuan	Memilih menu "Bantuan"	Menampilkan halaman daftar bantuan yang tersedia dengan benar		
		Memilih salah satu dari daftar bantuan	Menampilkan informasi mengenai bantuan yang telah dipilih dengan benar		
8.	Tentang	Memilih menu "Tentang"	Menampilkan halaman informasi mengenai aplikasi Aluradmi dengan benar		
9.	Alur Administrasi	Memilih salah satu administrasi yang tersedia dalam menu "Daftar Administrasi"	Menampilkan alur dari administrasi yang telah dipilih dengan benar		
10.	Detail alur administrasi	Menekan salah satu langkah pada halaman alur administrasi	Menampilkan detail informasi dari langkah administrasi yang dipilih dengan benar		

No.	Aktivitas/ Menu	Skenario Pengujian	Hasil yang Diharapkan	Lolos	
				Ya	Tidak
11.	Progress	<ul style="list-style-type: none"> <li>Melakukan aktivitas nomor 9</li> <li>Lanjutkan dengan memilih langkah pertama pada alur administrasi</li> <li>Menekan checkbox "Selesai"</li> <li>Tekan tombol back</li> </ul>	Progress pada langkah administrasi yang telah dipilih bertambah dan latar belakang pada langkah administrasi yang telah dipilih berwarna hijau, apabila progress telah mencapai 100% sedangkan berwarna kuning apabila progress lebih dari 0% dan kurang dari 100%		
		<ul style="list-style-type: none"> <li>Membuat progress pada langkah pertama mencapai 100%</li> <li>Memilih langkah selanjutnya pada alur administrasi</li> </ul>	Menampilkan checkbox "Selesai" pada detail informasi langkah administrasi, selain langkah pertama		
		<ul style="list-style-type: none"> <li>Memilih langkah administrasi yang telah mencapai progress 100% dan pada langkah selanjutnya telah terdapat progress lebih dari 0%</li> <li>Membatalkan checkbox "Selesai"</li> </ul>	Menampilkan peringatan apabila membatalkan status selesai pada langkah tersebut akan membuat progress pada langkah selanjutnya kembali ke 0%		
		Menekan tombol "Ya" pada tampilan peringatan pembatalan status selesai	Membuat progress pada langkah selanjutnya kembali ke 0%		
12.	Pengaturan	Memilih menu "Pengaturan"	Menampilkan halaman pengaturan dengan benar		
		Merubah jumlah hari dalam pengaturan sinkronisasi otomatis	Sinkronisasi otomatis telah berfungsi dengan benar		

No.	Aktivitas/ Menu	Skenario Pengujian	Hasil yang Diharapkan	Lolos	
				Ya	Tidak
13.	Pencarian Langkah Administrasi	Menekan ikon "Search"	Menampilkan halaman pencarian langkah administrasi dengan benar		
		Memasukkan beberapa huruf dengan "soft input keyboard" pada kolom pencarian	Menampilkan daftar langkah administrasi sesuai dengan huruf yang dimasukan dengan benar		
		Menekan salah satu langkah administrasi yang muncul	Menampilkan halaman detail informasi langkah administrasi sesuai yang dipilih dengan benar		
14.	Sinkronisasi Data	Menekan ikon "Sinkronisasi Data" dengan tidak tersambung pada internet	Menampilkan informasi bahwa sinkronisasi data membutuhkan koneksi internet		
		Menekan ikon "Sinkronisasi Data" dengan tersambung pada internet	Menampilkan informasi bahwa sinkronisasi data sedang dilakukan, apabila sinkronisasi data berhasil maka tanggal terakhir sinkronisasi pada halaman beranda akan berubah		
15.	Ganti Jurusan	Memilih menu "Ganti Jurusan"	Menampilkan daftar jurusan dengan benar		
		Memilih salah satu jurusan yang tersedia dan menekan tombol "OK"	Informasi jurusan pada halaman beranda berubah sesuai dengan jurusan yang dipilih		
16.	Berikan Review dan Rating	Memilih menu "Berikan Review dan Rating"	Menampilkan halaman Aluradmi pada aplikasi Google Play berfungsi dengan benar		

### Lampiran 3. Instrumen Uji Usability

No.	Pernyataan	SS	S	TS	STS
<b>USEFULLNESS</b>					
1.	Aplikasi ini membantu saya menjadi lebih efektif				
2.	Aplikasi ini membantu saya menjadi lebih produktif				
3.	Aplikasi ini bermanfaat				
4.	Aplikasi ini memberi saya dampak yang besar terhadap tugas yang saya lakukan dalam hidup saya				
5.	Aplikasi ini memudahkan saya mencapai hal-hal yang saya inginkan				
6.	Aplikasi ini menghemat waktu ketika saya menggunakannya				
7.	Aplikasi ini sesuai dengan kebutuhan saya				
8.	Aplikasi ini bekerja sesuai apa yang saya harapkan				
<b>EASE OF USE</b>					
9.	Aplikasi ini mudah digunakan				
10.	Aplikasi ini praktis untuk digunakan				
11.	Aplikasi ini mudah dipahami				
12.	Aplikasi ini memerlukan langkah-langkah yang praktis untuk mencapai apa yang ingin saya kerjakan				
13.	Aplikasi ini dapat disesuaikan dengan kebutuhan				
14.	Tidak kesulitan menggunakan aplikasi ini				
15.	Saya dapat menggunakan tanpa instruksi tertulis				
16.	Saya tidak melihat adanya ketidakkonsistenan selama saya menggunakannya				

No.	Pernyataan	SS	S	TS	STS
17.	Pengguna yang jarang maupun rutin menggunakan akan menyukai sistem ini				
18.	Saya dapat kembali dari kesalahan dengan cepat dan mudah				
19.	Saya dapat menggunakan sistem ini dengan berhasil setiap kali saya menggunakannya				
<b>EASE OF LEARNING</b>					
20.	Saya belajar menggunakan aplikasi ini dengan cepat				
21.	Saya mudah mengingat bagaimana cara menggunakan aplikasi ini				
22.	Mudah untuk mempelajari penggunaan aplikasi ini				
23.	Saya cepat menjadi terampil dengan aplikasi ini				
<b>SATISFACTION</b>					
24.	Saya puas dengan aplikasi ini				
25.	Saya akan merekomendasikan aplikasi ini kepada teman saya				
26.	Aplikasi ini menyenangkan untuk digunakan				
27.	Aplikasi ini bekerja seperti yang saya inginkan				
28.	Aplikasi ini sangat bagus				
29.	Saya merasa saya harus memiliki aplikasi ini				
30.	Aplikasi ini nyaman untuk digunakan				

#### Lampiran 4. Definisi *Use Case* dan Skenario *Use Case*

##### 1. Definisi Aktor

Berikut adalah deskripsi pendefinisian aktor pada aplikasi Aluradmi:

Tabel 23. Definisi Aktor

No	Aktor	Deskripsi
1.	Mahasiswa	Orang yang menggunakan aplikasi

##### 2. Definisi Use Case

Berikut adalah deskripsi dari *use case* pada aplikasi Aluradmi:

Tabel 24. Definisi *Use Case*

No	<i>Use Case</i>	Deskripsi
1.	Sinkronisasi Data	Use case ini merupakan generalisasi dari 6 use case lain yaitu penyetelan awal data, memeriksa status data, mengecek data, menambah data, mengubah data, menghapus data.
2.	Penyetelan Awal Data	Use case ini berfungsi untuk mengunduh data dari server dan menyimpannya di basis data aplikasi pada saat pertama kali aplikasi dijalankan.
3.	Memeriksa Status Data	Use case ini berfungsi untuk memeriksa apakah data sudah diunduh pada saat pertama kali aplikasi dijalankan.
4.	Mengecek Data	Use case ini berfungsi untuk memeriksa data di dalam aplikasi dengan data di server selaras atau tidak.
5.	Menambah Data	Use case ini berfungsi menambah data dalam basis data aplikasi apabila terjadi perubahan data di dalam server
6.	Mengubah Data	Use case ini berfungsi mengubah data dalam basis data aplikasi apabila terjadi perubahan data di dalam server
7.	Menghapus Data	Use case ini berfungsi menghapus data dalam basis data aplikasi apabila terjadi perubahan data di dalam server
8.	Memeriksa Status Jurusan	Use case ini berfungsi untuk memeriksa jurusan yang dipilih oleh pengguna.

No	Use Case	Deskripsi
9.	Memilih Jurusan	Use case ini berfungsi untuk membuka halaman pilihan jurusan yang tersedia.
10.	Mengelola Jurusan	Use case ini adalah generalisasi untuk pengelolaan informasi jurusan. Di dalamnya terdapat memilih jurusan dan memeriksa status jurusan.
11.	Melihat Daftar Administrasi	Use case ini berfungsi menampilkan daftar administrasi yang tersedia. Apabila salah satu administrasi ditekan maka akan menampilkan halaman alur administrasinya.
12.	Melihat Alur Administrasi	Use case ini berfungsi menampilkan alur administrasi yang berisikan langkah yang harus dilakukan dalam menyelesaikan administrasi. Apabila salah satu langkah ditekan akan menampilkan detail informasi langkah tersebut.
13.	Melihat Detail Langkah Administrasi	Use case ini berfungsi menampilkan halaman detail informasi dari langkah dalam menyelesaikan administrasi. Apabila tombol tunjukkan rute ditekan maka akan membuka halaman peta dan apabila gambar denah gedung ditekan maka akan menampilkan denah gedung tersebut.
14.	Melihat Peta	Use case ini merupakan generalisasi dari tampilan informasi peta. Berisi melihat lokasi gedung dan melihat rute.
15.	Melihat Rute	Use case ini berfungsi untuk menampilkan rute dari posisi pengguna sampai pada gedung yang dituju pada halaman peta.
16.	Melihat Beranda	Use case ini berfungsi untuk menampilkan halaman beranda.
17.	Melihat Bantuan	Use case ini berfungsi untuk menampilkan halaman bantuan.
18.	Lihat Tentang	Use case ini berfungsi untuk menampilkan halaman informasi pengembang aplikasi.
19.	Cari Langkah	Use case ini berfungsi untuk mencari langkah administrasi. Hasil pencarian berupa daftar. langkah administrasi



### 3. Skenario *Use Case*

Skenario *use case* merupakan alur jalannya proses *use case* dari sisi aktor dan sistem. Berikut adalah skenario *use case* dari aplikasi aluradmi:

#### a. Skenario Penyetelan Awal Data

Tabel 25. Skenario Penyetelan Awal Data

Aksi Aktor	Reaksi Sistem
Skenario Utama	
Membuka Aplikasi	
	Menampilkan halaman pengunduhan data
	Mengunduh data dari server dan menyimpan data ke dalam basis data aplikasi
	Mengubah isi variable "status data"
	Menampilkan halaman pilihan jurusan
Skenario Alternatif	
Membuka Aplikasi	
	Menampilkan peringatan penyetelan aplikasi membutuhkan koneksi internet
Menekan "Buka Setting"	
	Menampilkan halaman pengaturan sambungan internet
Menekan tombol "back"	
	Menampilkan halaman pengunduhan data
	Mengunduh data dari server dan menyimpannya di dalam basis data aplikasi
	Mengubah isi variable "Status Data"
	Menampilkan halaman pilihan jurusan

b. Skenario Memeriksa Status Data

Tabel 26. Skenario Memeriksa Status Data

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa ke dalam variable "status data" sebagai penanda apakah telah melakukan pengunduhan dan penyimpanan data di dalam basis data pada saat aplikasi pertama kali dijalankan
	Mengembalikan status data , sudah terdapat data atau belum di dalam aplikasi

c. Skenario Sinkronisasi Data

Tabel 27. Skenario Sinkronisasi Data

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
Menekan menu "Sinkronisasi Data"	
	Menampilkan informasi "Sedang melakukan sinkronisasi data".
	Mengecek ketersediaan koneksi internet
	Mengecek jumlah data dan tanggal maksimal timestamp di basis data dan dibandingkan dengan data dalam server
	Membandingkan id pada setiap data dalam tabel dengan server
	Mengunduh data terbaru
	Memasukkan data terbaru, mengubah data, atau menghapus data di dalam basis data aplikasi

Aksi Aktor	Reaksi Sistem
Skenario Alternatif	
	Memeriksa Status Data
Menekan menu "Sinkronisasi Data"	
	Menampilkan informasi "Sedang melakukan sinkronisasi data".
	Mengecek ketersediaan koneksi internet
	Menampilkan pesan "Sinkronisasi data membutuhkan koneksi internet"

d. Skenario Mengecek Data

Tabel 28. Skenario Mengecek Data

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Mengecek jumlah data dan tanggal maksimal timestamp di basis data dan dibandingkan dengan data dalam server
	Mengembalikan status apakah perlu dilakukan sinkronisasi data

e. Skenario Menambah Data

Tabel 29. Skenario Menambah Data

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Mengecek Data
	Menambahkan data terbaru ke dalam basis data aplikasi
	Mengembalikan status data telah diperbarui

f. Skenario Mengubah Data

Tabel 30. Skenario Mengubah Data

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Mengecek Data
	Mengubah data yang terdapat dalam basis data aplikasi agar sesuai dengan yang berada pada server
	Mengembalikan status data telah diperbarui

g. Skenario Menghapus Data

Tabel 31. Skenario Menghapus Data

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Mengecek Data
	Menghapus data di dalam basis data aplikasi yang tidak terdapat dalam server
	Mengembalikan status data telah diperbarui

h. Skenario Memilih Jurusan

Tabel 32. Skenario Memilih Jurusan

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
	Menampilkan daftar jurusan yang tersedia
Menekan salah satu jurusan	
	Memberi tanda pada jurusan yang ditekan
Menekan tombol "OK"	
	Menampilkan pesan jurusan telah terpilih
	Menampilkan halaman beranda
Skenario Alternatif	
Menekan menu "Ganti Jurusan"	
	Menampilkan daftar jurusan yang tersedia
Menekan salah satu jurusan	
	Memberi tanda pada jurusan yang ditekan
Menekan tombol "OK"	
	Menampilkan pesan jurusan telah terpilih
	Menampilkan halaman beranda aplikasi Aluradmi

i. Skenario Memeriksa Status Jurusan

Tabel 33. Skenario Memeriksa Status Jurusan

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
	Memeriksa jurusan yang dipilih di dalam basis data
	Mengembalikan informasi jurusan yang telah dipilih oleh pengguna

j. Skenario Melihat Daftar Administrasi

Tabel 34. Skenario Melihat Daftar Administrasi

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
Menekan tombol menu utama	
	Menampilkan daftar menu utama
Menekan menu "Daftar Administrasi"	
	Memunculkan daftar administrasi yang tersedia di bawah menu "Daftar Administrasi"

k. Skenario Melihat Alur Administrasi

Tabel 35. Skenario Melihat Alur Administrasi

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
Menekan salah satu administrasi yang ada di menu "Daftar Administrasi"	
	Memeriksa status jurusan
	Menampilkan halaman alur administrasi

# I. Skenario Melihat Detail Langkah Administrasi

Tabel 36. Skenario Melihat Detail Langkah Administrasi

Aksi Aktor	Reaksi Sistem
<b>Skenario Utama</b>	
	Memeriksa Status Data
Menekan salah satu administrasi yang ada di menu "Daftar Administrasi"	
	Memeriksa status jurusan
	Menampilkan halaman alur administrasi
Menekan salah satu langkah pada halaman alur administrasi	
	Menampilkan halaman detail langkah administrasi
<b>Skenario Alternatif</b>	
	Memeriksa Status Data
Memilih menu "Pencarian Langkah"	
	Menampilkan halaman pencarian
Memasukkan nama langkah yang dicari pada kolom pencarian	
	Menampilkan daftar pencarian yang sesuai
Menekan salah satu hasil pencarian	
	Menampilkan halaman detail langkah administrasi

m. Skenario Melihat Peta

Tabel 37. Skenario Melihat Peta

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
Memilih menu "Peta"	
	Menampilkan halaman peta dengan marker yang menunjukkan lokasi gedung

n. Skenario Melihat Rute

Tabel 38. Skenario Melihat Rute

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
Memilih menu "Peta"	
	Menampilkan halaman peta dengan marker yang menunjukkan lokasi gedung
Menekan salah satu marker	
	Menampilkan info window
Menekan info window	
	Menampilkan rute dari posisi pengguna ke lokasi gedung

o. Skenario Melihat Beranda

Tabel 39. Skenario Melihat Beranda

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
	Menampilkan halaman beranda
Skenario Alternatif	
	Memeriksa Status Data
Memilih menu "Beranda"	
	Menampilkan halaman beranda



p. Skenario Melihat Bantuan

Tabel 40. Skenario Melihat Bantuan

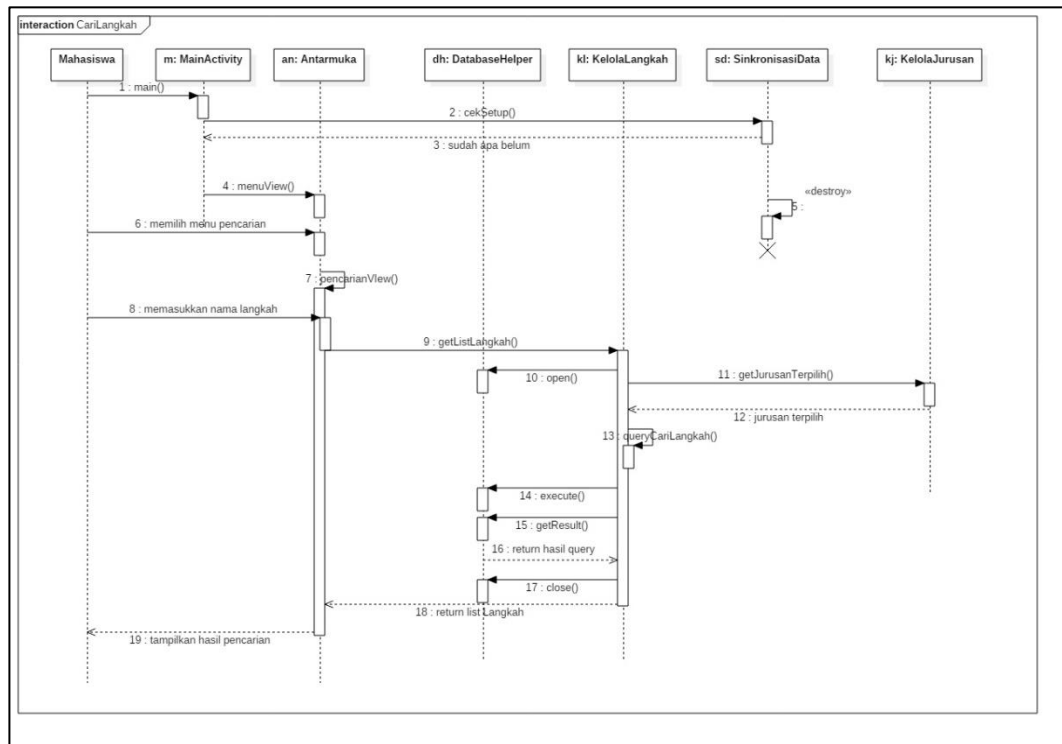
Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
Memilih menu "Bantuan"	
	Menampilkan halaman bantuan yang berisi daftar bantuan
Menekan salah satu daftar bantuan	
	Menampilkan detail informasi bantuan

q. Skenario Cari Langkah

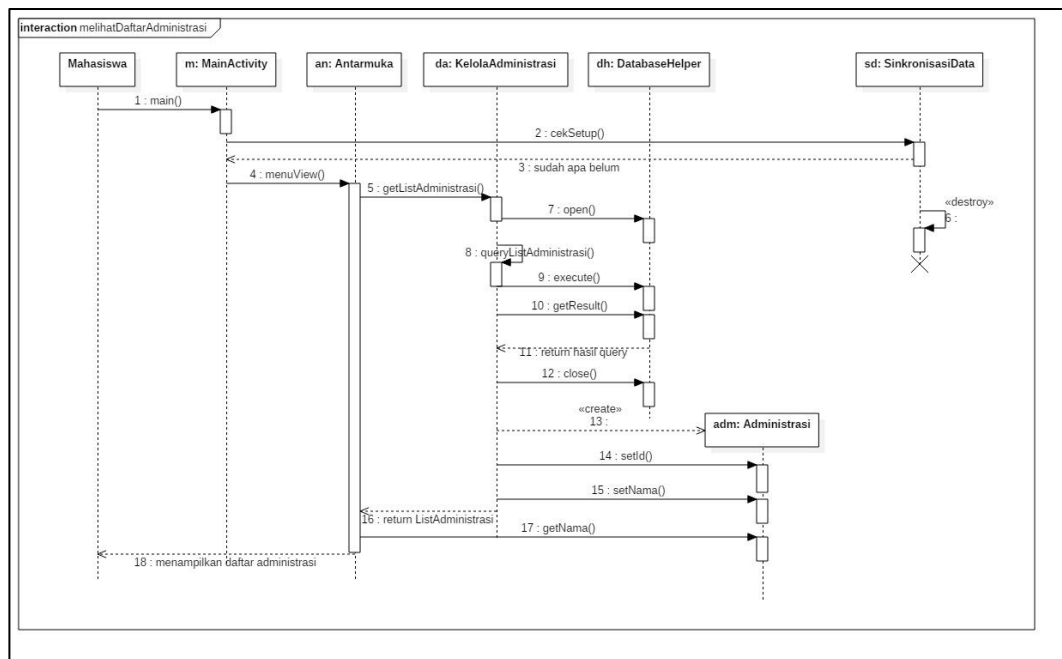
Tabel 41. Skenario Cari Langkah

Aksi Aktor	Reaksi Sistem
Skenario Utama	
	Memeriksa Status Data
Memilih menu "Pencarian Langkah"	
	Menampilkan halaman pencarian
Memasukkan nama langkah yang dicari pada kolom pencarian	
	Menampilkan daftar pencarian yang sesuai

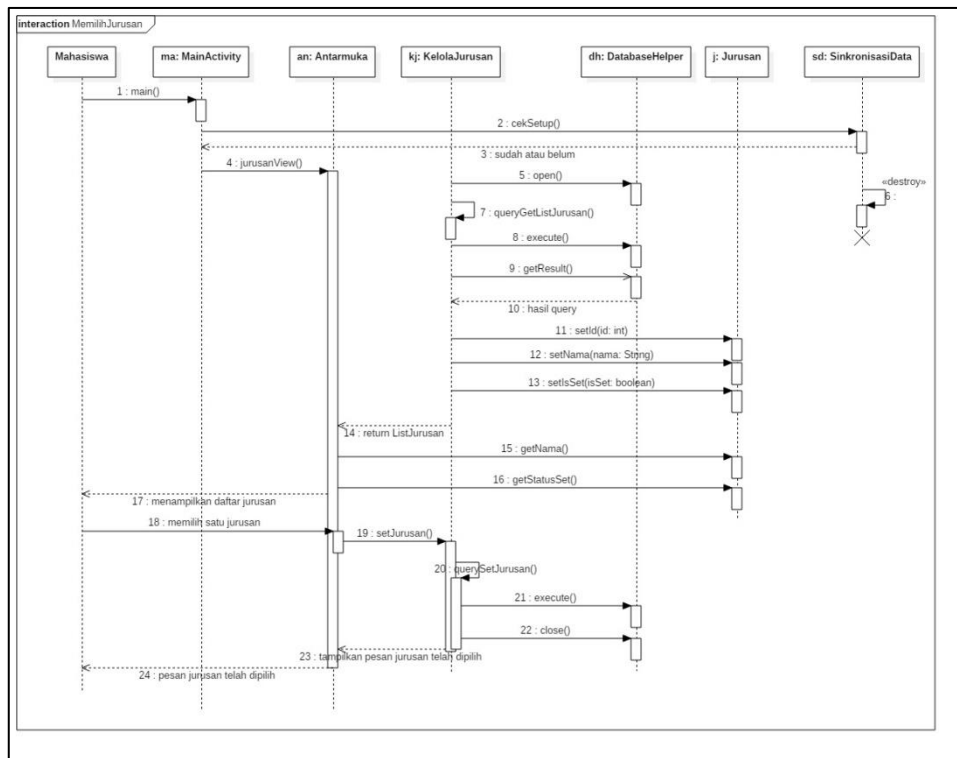
## Lampiran 5. Desain *Sequence Diagram*



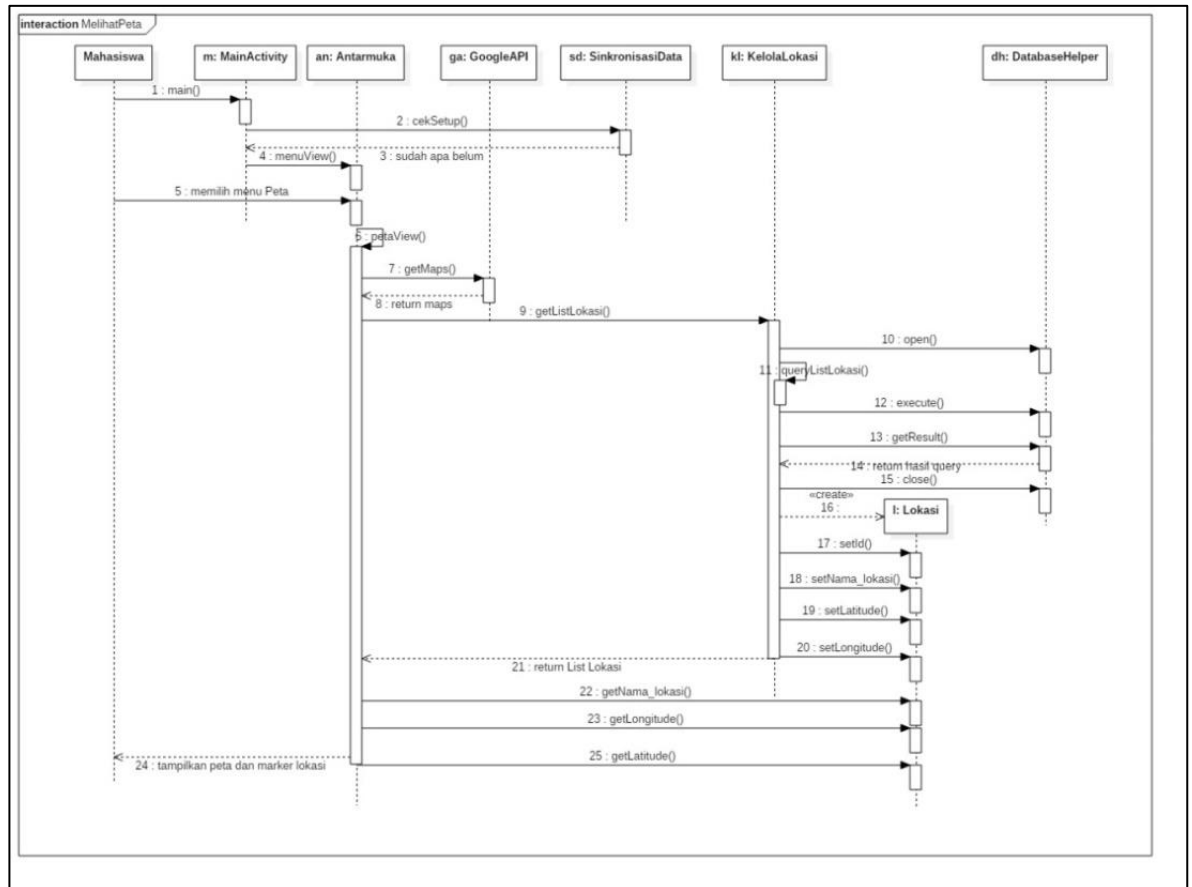
Gambar 43. *Sequence Diagram* Cari Langkah



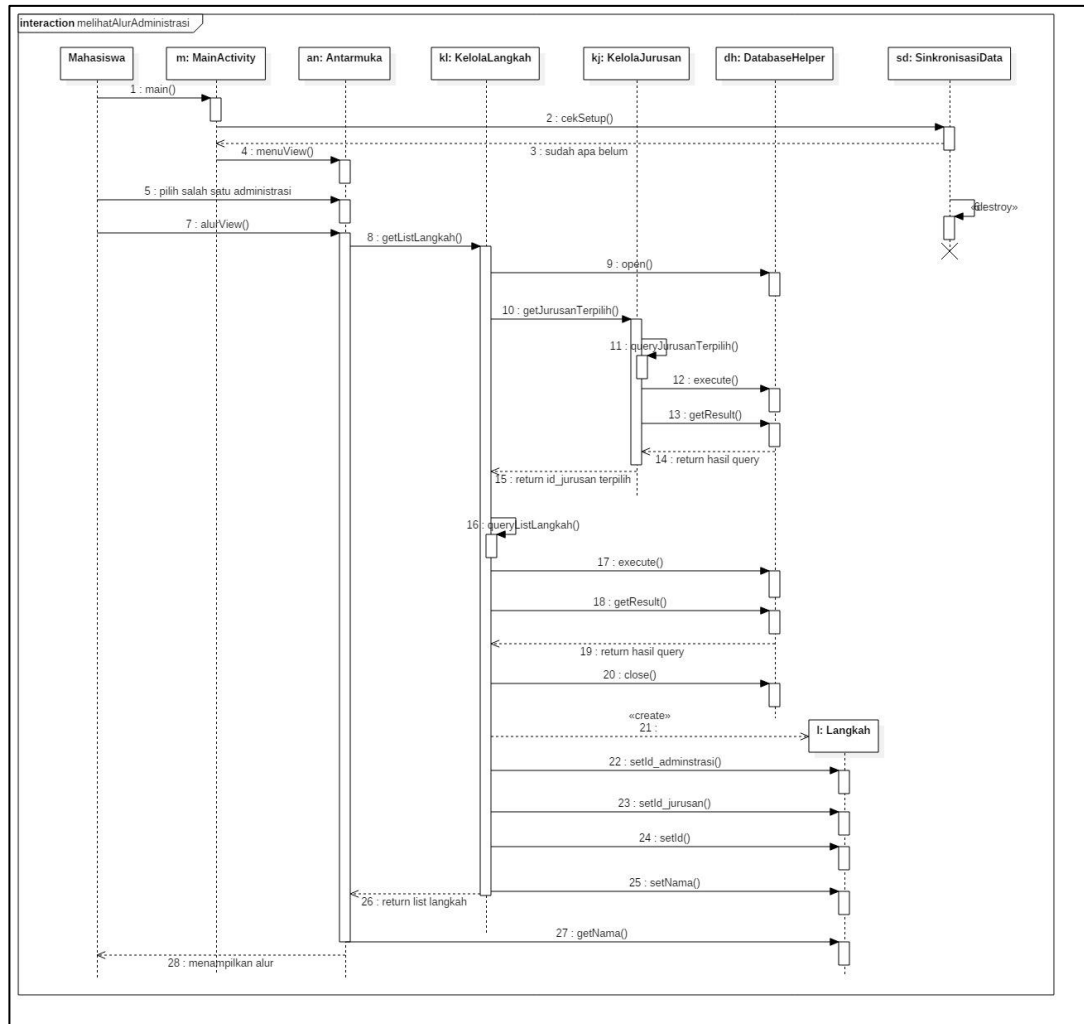
Gambar 44. *Sequence Diagram* Melihat Daftar Administrasi



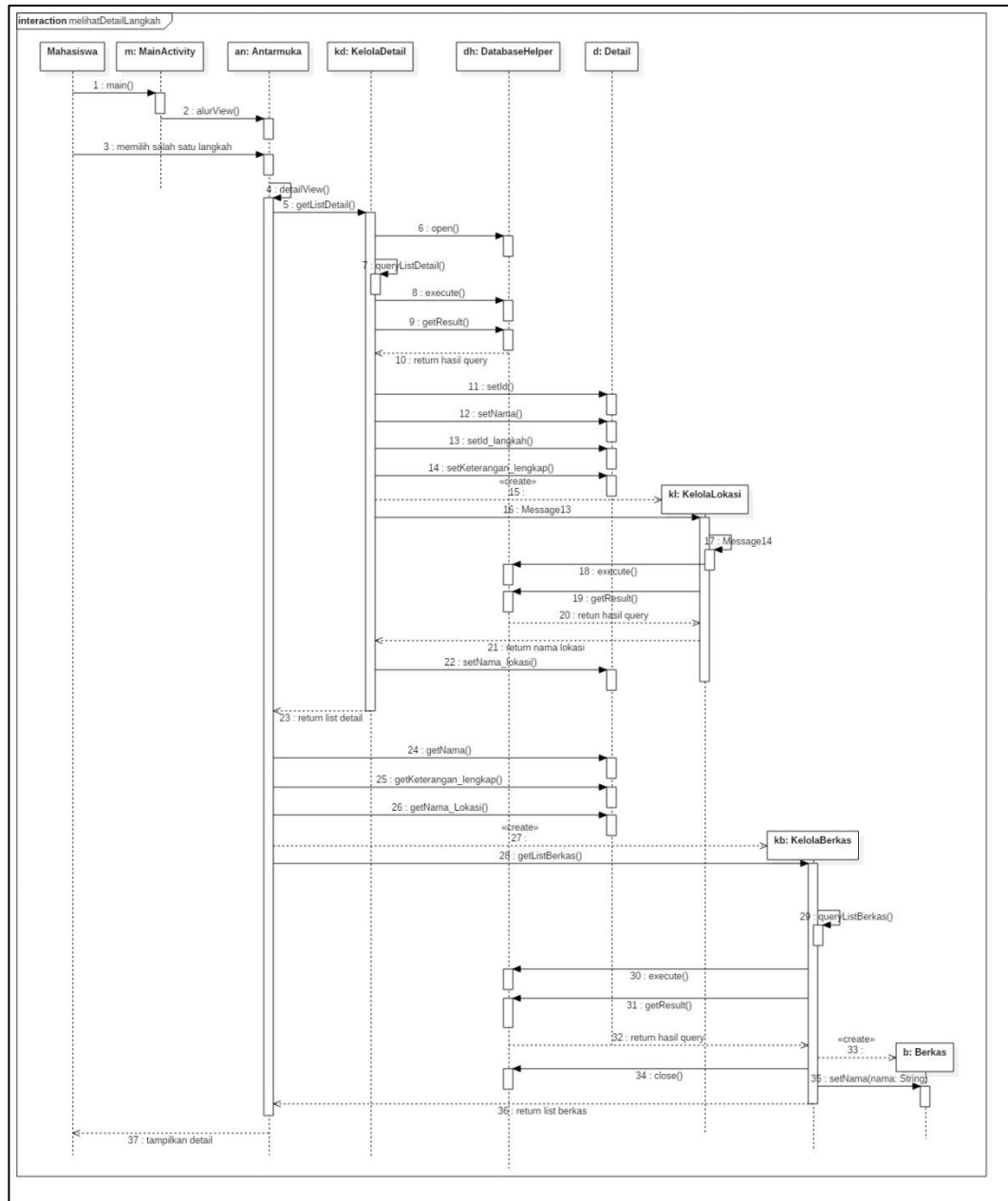
Gambar 45. *Sequence Diagram* Memilih Jurusan



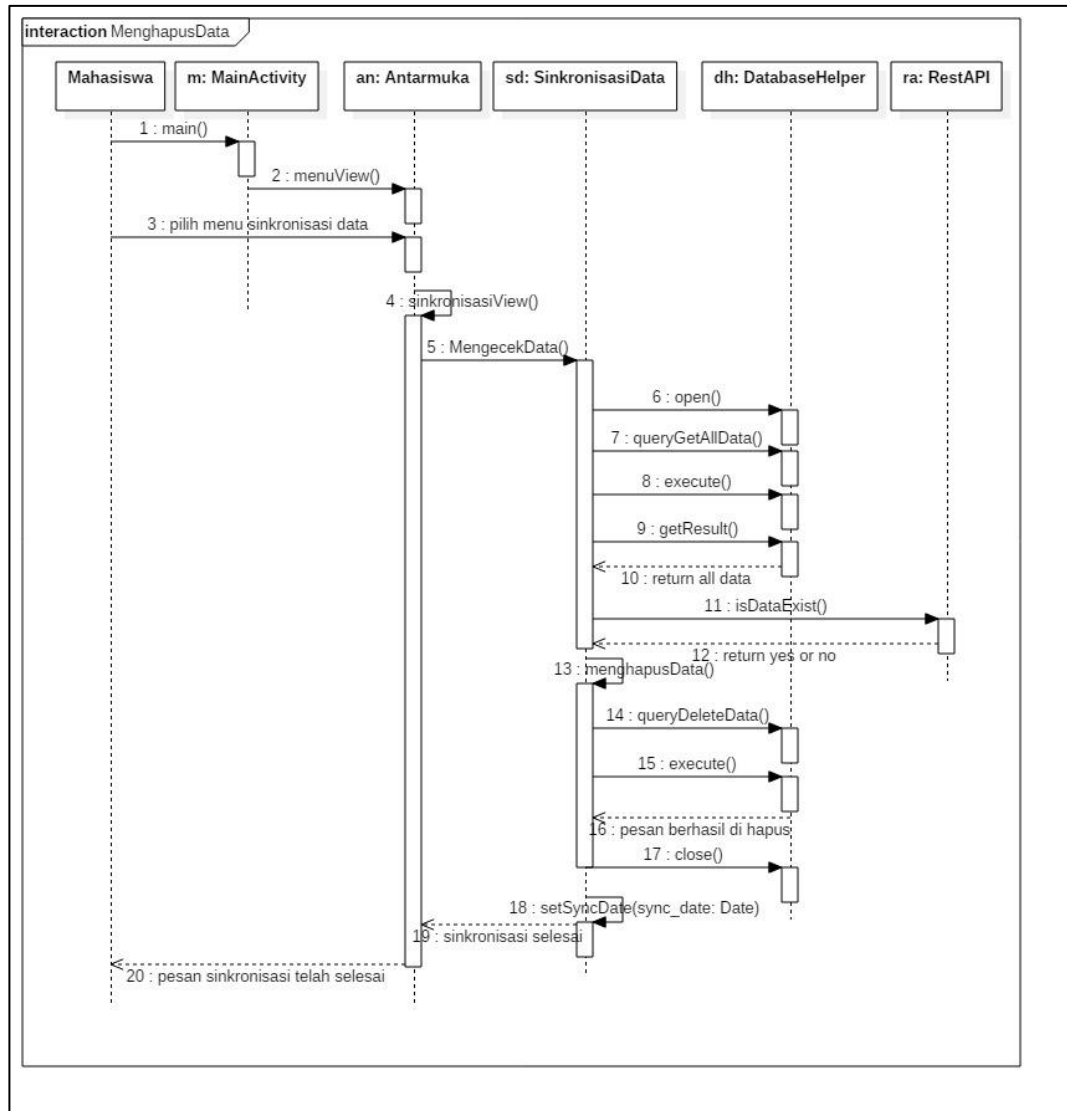
Gambar 46. *Sequence Diagram* Melihat Peta



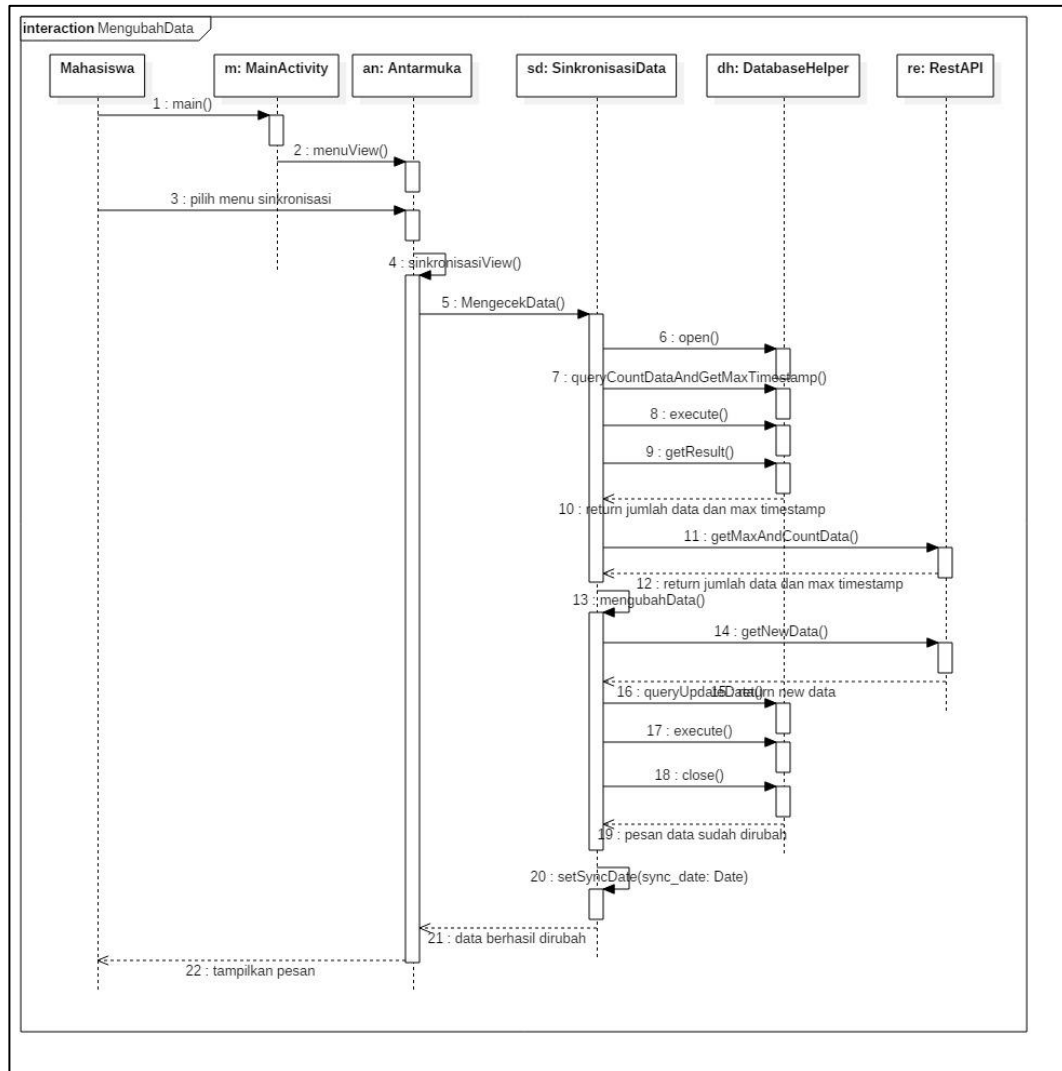
Gambar 47. *Sequence Diagram* Melihat Alur Administrasi



Gambar 48. *Sequence Diagram* Melihat Detail Langkah Administrasi

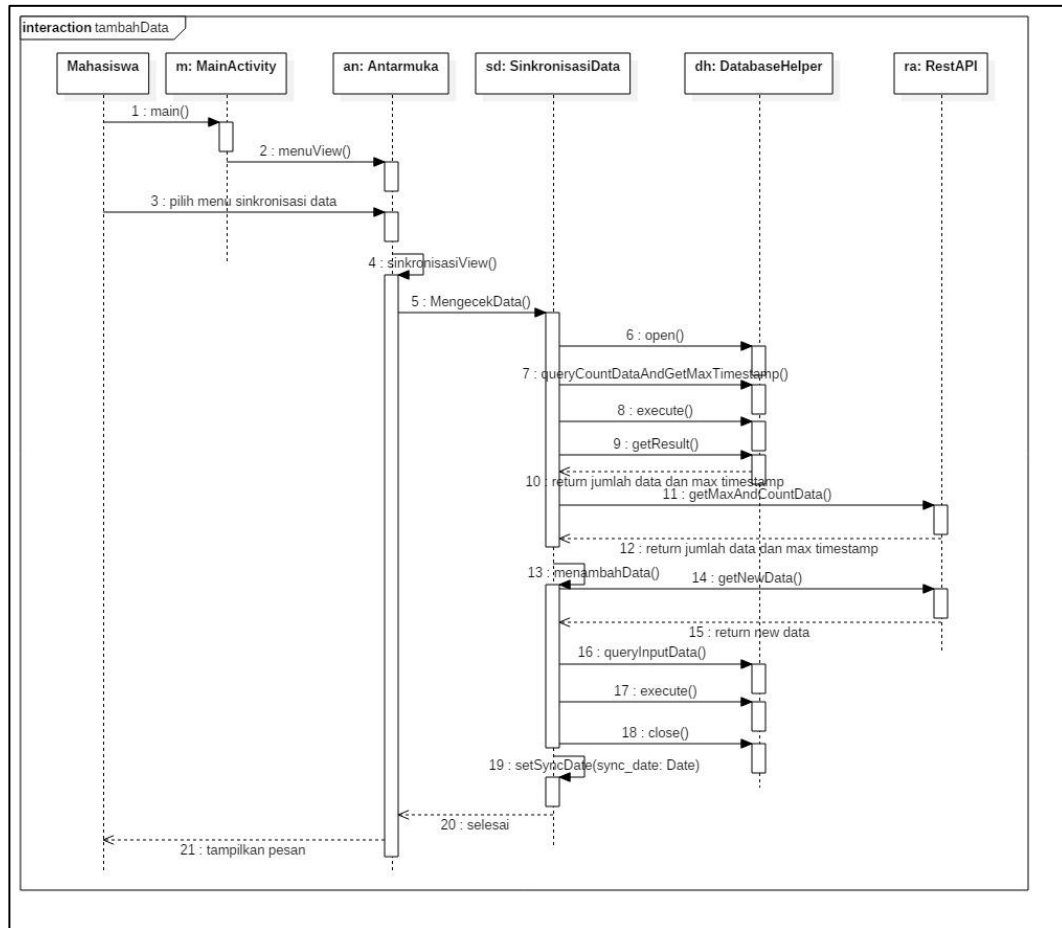


Gambar 49. *Sequence Diagram* Menghapus Data

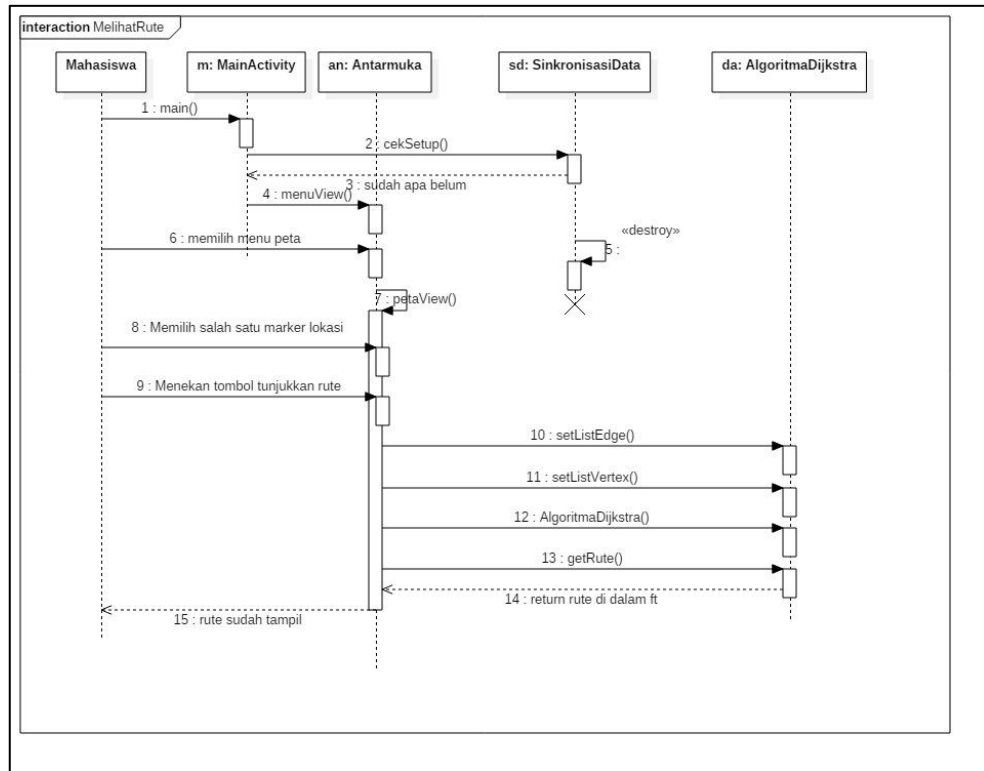


Gambar 50. *Sequence Diagram* Mengubah Data

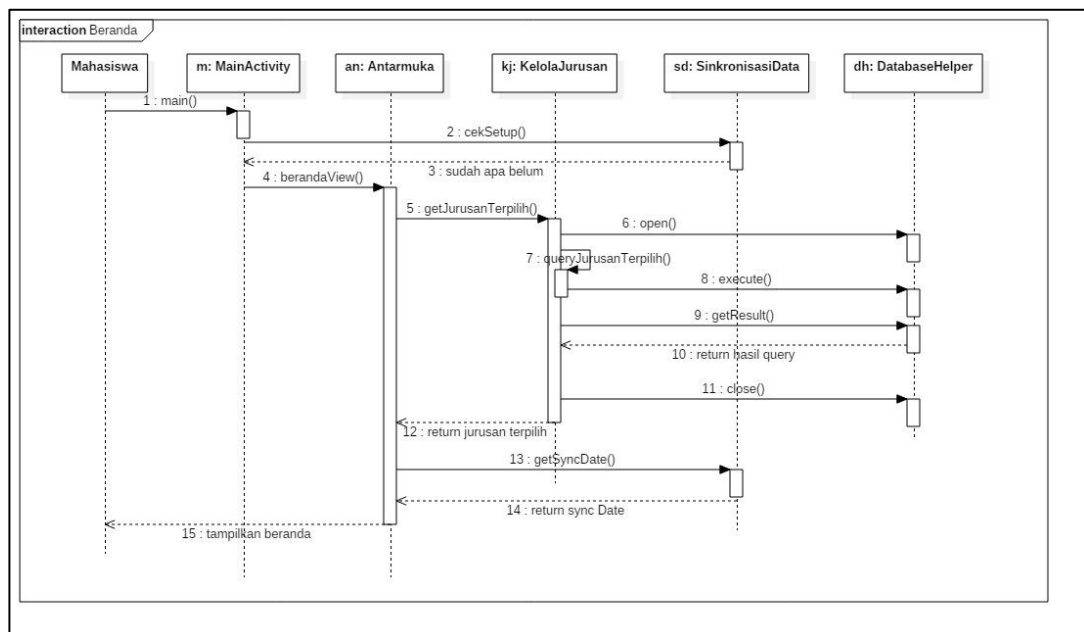




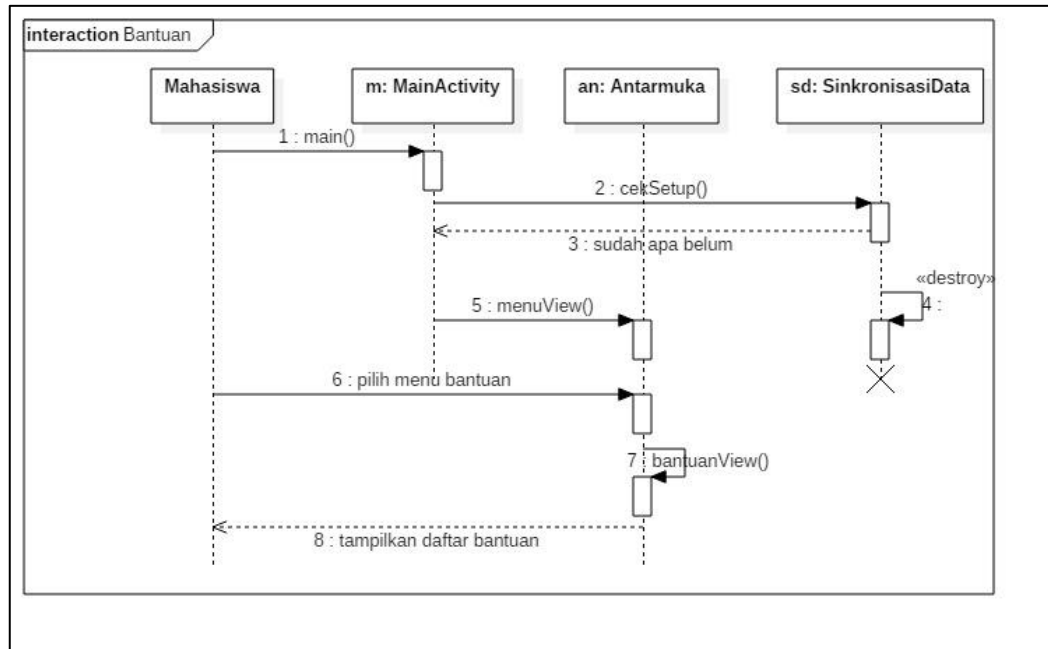
Gambar 51. *Sequence Diagram* Tambah Data



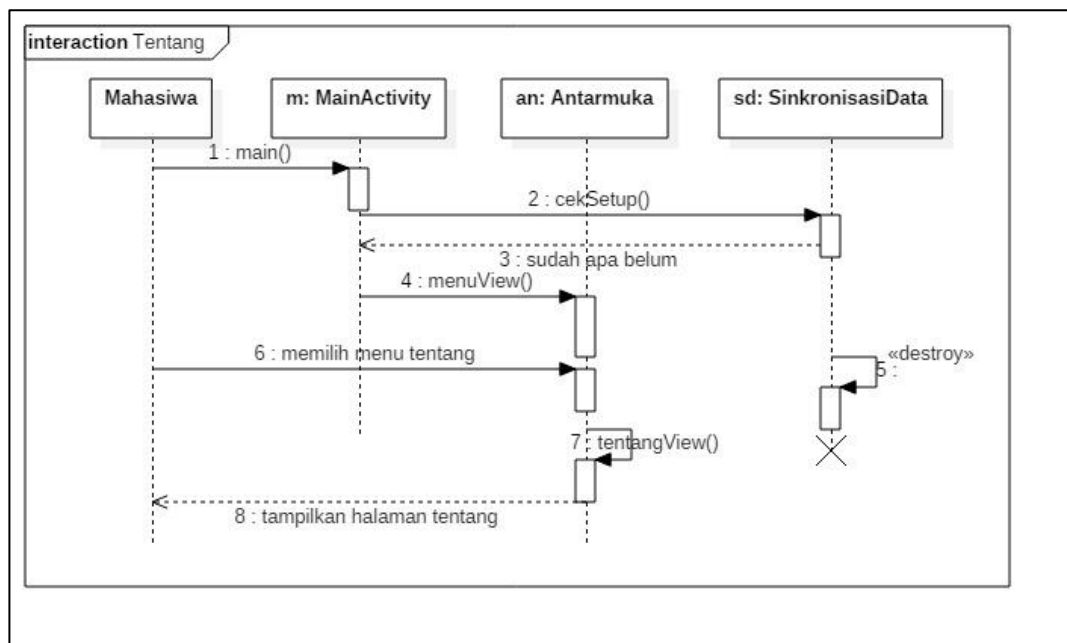
Gambar 52. *Sequence Diagram* Melihat Rute



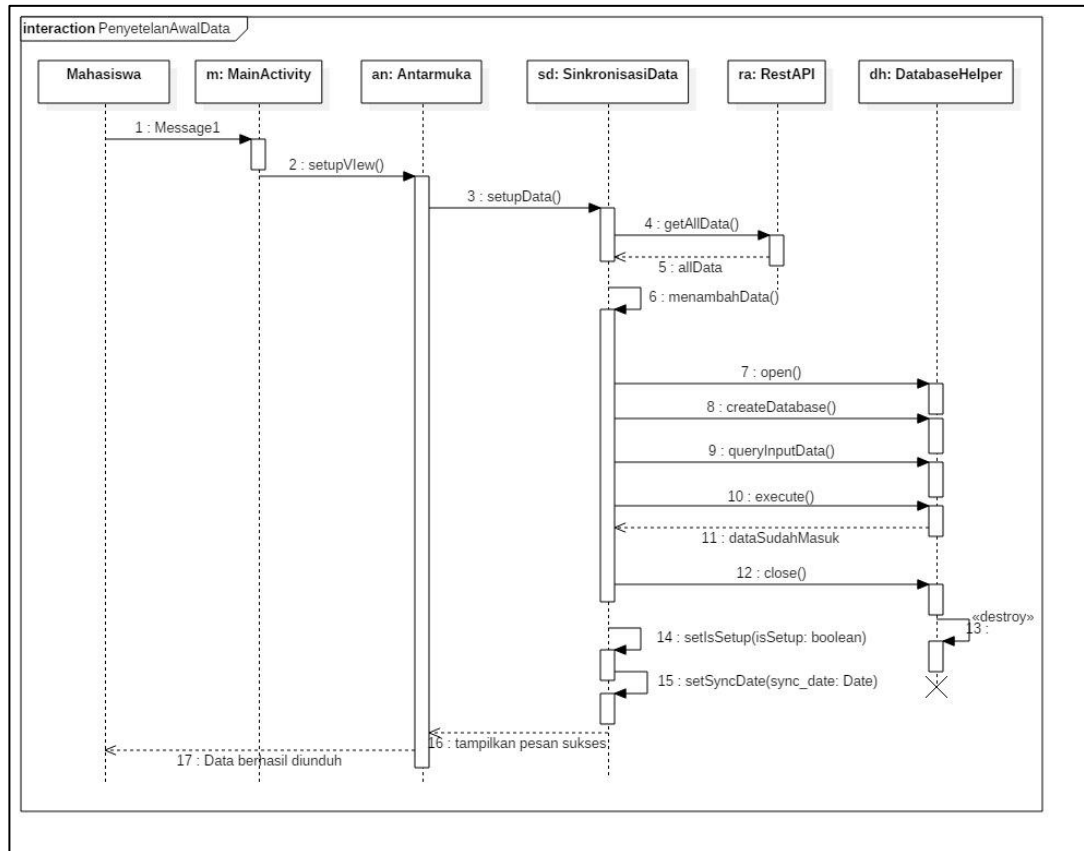
Gambar 53. *Sequence Diagram* Beranda



Gambar 54. *Sequence Diagram* Bantuan

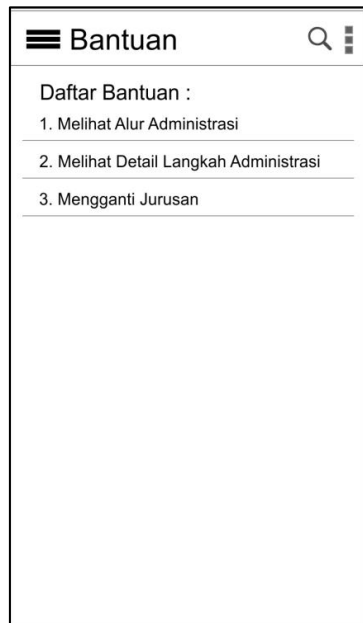


Gambar 55. *Sequence Diagram* Tentang

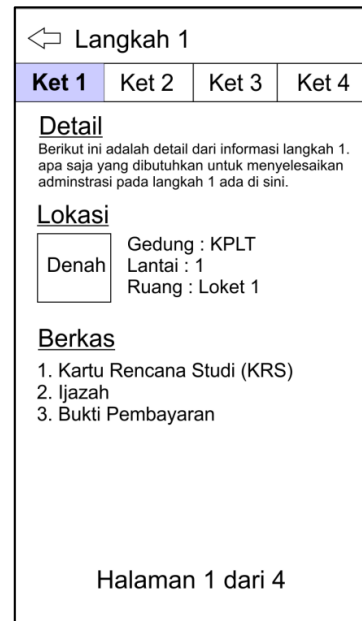


Gambar 56. *Sequence Diagram* Penyetelan Awal Data

## Lampiran 6. Desain Antarmuka Aplikasi



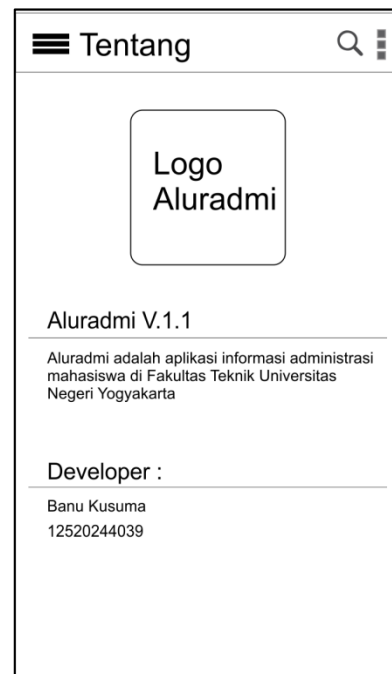
Gambar 57. Desain Antarmuka Bantuan



Gambar 58. Desain Antarmuka Detail Langkah Administrasi



Gambar 59. Desain Antarmuka Halaman Pencarian



Gambar 60. Desain Antarmuka Halaman Tentang

## Lampiran 7. Daftar Layout

No	Nama	Role
1.	activity_bantuan_lengkap_baru.xml	Layout untuk menampilkan halaman bantuan lengkap
2.	Activity_bantuan_singkat.xml	Layout untuk menampilkan halaman panduan singkat
3.	Activity_fragment.xml	Layout yang digunakan untuk menempatkan halaman alur dan detail
4.	Activity_fullscreen.xml	Layout untuk menampilkan halaman denah
5.	Activity_maps2.xml	Layout untuk menampilkan halaman peta
6.	Activity_pengaturan.xml	Layout untuk menampilkan halaman pengaturan
7.	Activity_search.xml	Layout untuk menampilkan halaman pencarian langkah
8.	Activity_setup.xml	Layout untuk menampilkan halaman penyetelan awal data
9.	Copyright_layout.xml	Layout untuk menampilkan copyright aplikasi
10.	Ending_bantuan_singkat.xml	Layout untuk menampilkan halaman terakhir panduan singkat aplikasi
11.	Fragment_alur_list.xml	Layout untuk menempatkan item dari halaman alur
12.	Fragment_bantuan.xml	Layout untuk menempatkan item dari halaman bantuan
13.	Fragment_home.xml	Layout untuk menempatkan item yang terdapat pada halaman beranda
14.	Home_layout.xml	Layout yang berisi ucapan selamat datang dan logo uny
15.	Info_window.xml	Layout info window pada marker halaman peta
16.	Item_bantuan_5.xml	Layout isi bantuan halaman 5
17.	Item_bantuan_6.xml	Layout isi bantuan halaman 6
18.	Item_bantuan_7.xml	Layout isi bantuan halaman 7
19.	Item_bantuan_8.xml	Layout isi bantuan halaman 8
20.	Item_bantuan_9.xml	Layout isi bantuan halaman 9

No	Nama	Role
21.	Item_bantuan_baru_1.xml	Layout isi bantuan halaman 1
22.	Item_bantuan_baru_2.xml	Layout isi bantuan halaman 2
23.	Item_bantuan_baru_3.xml	Layout isi bantuan halaman 3
24.	Item_bantuan_baru_4.xml	Layout isi bantuan halaman 4
25.	Item_home.xml	Layout yang berisi jurusan yang dipilih dan tanggal sinkronisasi
26.	Item_jurusan.xml	Layout dari masing-masing pilihan jurusan
27.	Item_keterangan.xml	Layout dari tiap halaman detail
28.	item_list_bantuan.xml	Layout dari tiap item yang terdapat dalam halaman awal bantuan
29.	Item_search.xml	Layout dari item langkah yang terdapat dalam halaman pencarian
30.	Jurusan_activity.xml	Layout halaman jurusan
31.	Keterangan_pager.xml	Layout untuk menempatkan item detail langkah
32.	Layout_about.xml	Layout halaman tentang
33.	List_item_alur.xml	Layout dari item langkah yang terdapat dalam halaman alur
34.	New_activity_main.xml	Layout halaman utama
35.	Opening_bantuan_lengkap.xml	Layout halaman awal bantuan
36.	Opening_bantuan_singkat.xml	Layout halaman awal panduan singkat
37.	Toolbar_layout.xml	Layout untuk menempatkan menu yang digunakan dalam aplikasi

## Lampiran 8. Daftar File Java

No	Nama File	Fungsi
1.	MainActivity.java	Aktivitas utama yang menangani menu, pergantian antara halaman beranda dan about.
2.	AlurListActivity.java	Aktivitas yang menangani halaman alur administrasi.
3.	KeteranganPagerActivity.java	Aktivitas yang menangani halaman detail langkah administrasi.
4.	AlurSearchActivity.java	Aktivitas yang menangani halaman pencarian langkah administrasi.
5.	JurusanActivity.java	Aktivitas yang menangani halaman pilihan jurusan.
6.	MapsActivity.java	Aktivitas yang menangani halaman peta.
7.	SetupActivity.java	Aktivitas yang menangani halaman awal pengunduhan data dari server.
8.	SettingActivity.java	Aktivitas yang menangani halaman pengaturan yang berisi sinkronisasi otomatis.
9.	SplashActivity.java	Aktivitas yang menangani halaman <i>splash screen</i> .
10.	DenahActivity.java	Aktivitas yang menangani halaman denah.
11.	AboutFragment.java	Menangani tampilan about.
12.	BantuanLengkapMaster.java	Menangani tampilan bantuan lengkap.
13.	EndingBantuanSingkatFragment.java	Menangani tampilan halaman ending panduan singkat.
14.	HomeFragment.java	Menangani tampilan beranda.
15.	KeteranganFragment.java	Menangani setiap halaman di dalam detail langkah administrasi
16.	OpeningBantuanLengkap.java	Menangani tampilan list bantuan
17.	OpeningBantuanSingkatFragment.java	Menangani halaman awal panduan singkat
18.	ReuniAlur.java	Menghubungkan data alur di basis data dengan model alur



No	Nama File	Fungsi
19.	ReuniBerkas.java	Menghubungkan data berkas di basis data dengan model berkas
20.	ReuniGedung.java	Menghubungkan data gedung di basis data dengan model gedung
21.	ReuniJurusan.java	Menghubungkan data jurusan di basis data dengan model jurusan
22.	ReuniKategori.java	Menghubungkan data kategori di basis data dengan model kategori
23.	ReuniKeterangan.java	Menghubungkan data keterangan di basis data dengan model keterangan
24.	ReuniRuang.java	Menghubungkan data ruang di basis data dengan model ruang
25.	Alur.java	Sebagai model data alur
26.	Berkas.java	Sebagai model data berkas
27.	Edge.java	Sebagai model data edge atau lintasan
28.	Gedung.java	Sebagai model data gedung
29.	Graph.java	Sebagai model data grafik
30.	Jurusan.java	Sebagai model data jurusan
31.	Kategori.java	Sebagai model data kategori
32.	Keterangan.java	Sebagai model data keterangan
33.	Ruang.java	Sebagai model data ruang
34.	Rute.java	Sebagai model data rute yang di dapat dari google direction API
35.	Vertex.java	Sebagai model vertex atau titik
36.	DatabaseHelper.java	Sebagai penghubung antara aplikasi dengan SQLite basis data
37.	DijkstraAlgorithm.java	Menangani pembuatan rute tercepat di dalam fakultas teknik
38.	DirectionFinderListener.java	Listener dari pencarian rute pada halaman peta
39.	onSearchListSelected.java	Listener yang menghubungkan SearchAdapter dengan SearchActivity
40.	AluradmiRestClient.java	Menangani akses ke webservice

No	Nama File	Fungsi
41.	BootBroadcastReceiver.java	Receiver yang akan menjalankan SinkronisasiService ketika device melakukan booting
42.	SearchAdapter.java	Sebagai pengolah data hasil pencarian dan menampilkan di layout item pencarian
43.	SinkronisasiService.java	Menangani sinkronisasi data antara aplikasi dengan server
44.	PeriodicTaskReceiver.java	Receiver yang akan menjalankan SinkronisasiService ketika menu Sinkronisasi Data dipilih
45.	CheckNetwork.java	Sebagai pengecek koneksi internet
46.	DirectionFinder.java	Menangani pencarian rute dengan menggunakan google direction API
47.	GetAllData.java	Menangani data yang didapat pertama kali untuk dimasukkan ke dalam basis data

## Lampiran 9. Implementasi Antarmuka

### 1. Halaman Pencarian Alur

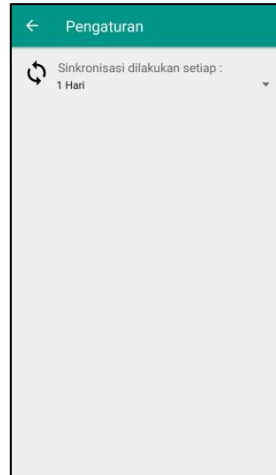
Implementasi antarmuka halaman pencarian langkah dapat dilihat pada Gambar 61. Apabila pengguna memasukkan huruf atau kata di dalam kolom pencarian maka akan muncul daftar langkah administrasi yang sesuai dengan yang dimasukkan. Apabila hasil pencarian di tekan oleh pengguna maka akan menuju ke halaman detail langkah administrasi.



Gambar 61. Halaman Pencarian Langkah

## 2. Halaman Pengaturan

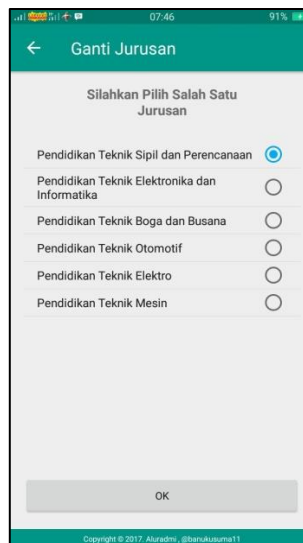
Implementasi antarmuka halaman pengaturan dapat dilihat di Gambar 62. Di dalamnya terdapat pilihan jumlah hari untuk melakukan sinkronisasi data secara otomatis.



Gambar 62. Halaman Pengaturan

## 3. Halaman Pilihan Jurusan

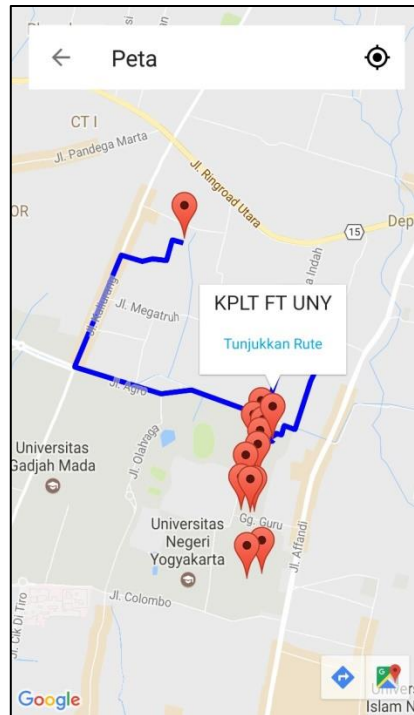
Implementasi antarmuka halaman pilihan jurusan dapat dilihat pada Gambar 63. Di dalamnya terdapat pilihan jurusan yang tersedia.



Gambar 63. Halaman Pilihan Jurusan

#### 4. Halaman Peta

Implementasi antarmuka halaman peta dapat dilihat pada Gambar 64. Didalamnya terdapat *marker* yang menunjukkan lokasi gedung. Apabila info window pada *marker* ditekan maka akan mengakses fitur pencarian rute dari posisi pengguna ke lokasi gedung tersebut.



Gambar 64. Halaman Peta

## Lampiran 10. Letak Lokasi Titik

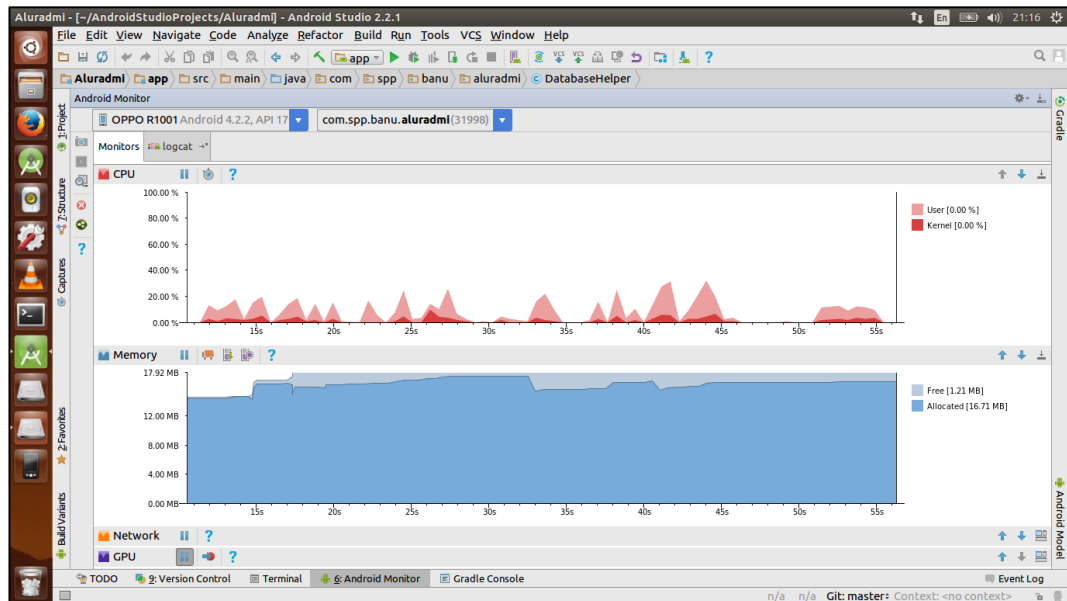


Gambar 65. Lokasi Titik

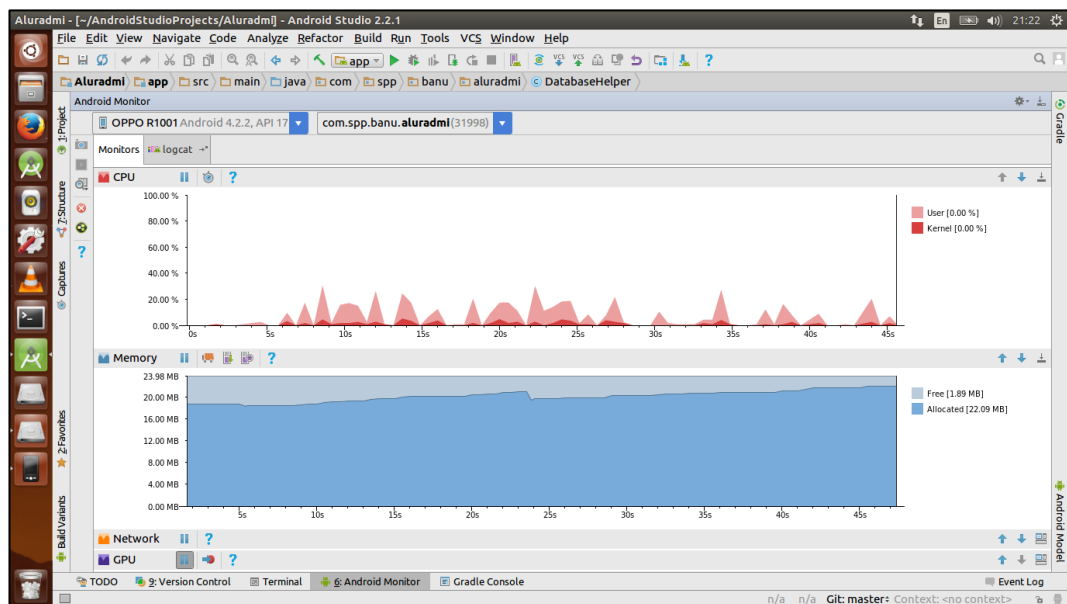
**Lampiran 11. Hasil Pengujian *Usability***

NO	JURUSAN	NIM	Nomor soal																														Total Skor
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
1	PTEI	12520241030	3	3	3	3	3	3	3	3	4	4	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	94	
2	PTEI	12520244035	3	2	3	2	3	3	3	3	3	3	3	4	4	4	2	3	3	4	2	4	4	4	3	3	3	3	4	3	2	4	94
3	PTEI	12520241062	3	3	4	3	3	3	3	3	4	4	4	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	95
4	PTEI	12520241054	3	3	3	2	3	3	4	4	3	3	3	4	4	4	3	3	3	3	4	3	4	3	3	3	3	3	3	3	3	3	96
5	PTEI	12520244011	3	3	4	4	3	4	4	3	3	3	4	3	3	3	4	4	3	4	3	4	4	4	3	3	4	3	4	3	3	3	103
6	PTSP	13505241065	4	4	4	3	4	3	4	3	3	3	4	4	4	3	4	3	4	3	3	3	4	4	4	4	4	4	4	4	4	4	110
7	PTSP	13505241044	4	3	4	3	4	4	4	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	4	4	3	4	4	4	101
8	PTSP	13505241063	4	3	4	3	3	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	116
9	PTSP	13505241078	4	3	4	3	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	117
10	PTSP	13505244025	4	4	4	3	4	3	4	3	3	3	3	4	4	3	4	3	4	4	4	4	3	4	4	3	3	3	4	4	4	4	108
11	PTBB	13514134027	4	4	4	3	4	4	4	4	4	4	3	4	4	4	3	4	4	4	3	4	4	4	4	4	4	4	3	4	4	4	115
12	PTBB	14511241028	3	3	4	4	3	4	4	3	4	4	4	4	4	4	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	114
13	PTBB	14511241023	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	89
14	PTBB	14511241031	3	3	4	4	3	3	3	3	4	4	4	4	3	3	3	2	3	2	3	3	4	3	3	3	3	3	3	3	3	3	95
15	PTBB	14511241013	4	4	4	3	3	3	4	4	4	4	4	4	4	3	3	3	3	3	3	4	4	4	4	3	3	3	4	4	4	4	108
16	PTE	12518241034	4	4	4	4	3	4	4	3	4	3	4	3	3	3	3	3	3	3	3	4	4	4	4	3	3	2	3	2	3	3	100
17	PTE	12518241013	4	3	4	3	3	3	3	4	4	3	3	4	3	3	4	4	3	4	3	3	3	3	3	4	3	3	3	3	3	4	100
18	PTE	12501241029	3	3	3	3	3	3	3	2	3	3	3	2	3	4	3	3	2	3	3	3	3	3	2	3	3	3	2	3	3	3	86
19	PTE	13501241060	3	3	3	3	3	4	4	3	3	4	3	3	3	3	4	3	3	3	3	3	4	4	3	3	3	3	3	3	3	4	97
20	PTE	12501241012	3	3	3	3	3	3	3	3	4	3	4	3	2	3	3	3	2	3	3	3	3	3	3	3	3	3	2	3	3	4	90
21	PTOto	16509134028	4	3	3	3	3	4	3	3	4	4	4	3	3	3	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	96
22	PTOto	16509134025	4	4	3	4	3	3	3	3	3	3	4	4	4	4	3	4	3	4	3	3	3	4	3	3	4	3	4	3	4	3	103
23	PTOto	16509134017	4	3	4	3	3	3	3	3	4	4	3	3	3	3	4	3	3	3	3	4	4	3	3	3	3	3	3	3	3	3	97
24	PTOto	16509134030	3	3	3	3	3	3	3	4	4	4	4	3	3	3	4	3	3	4	4	4	4	4	4	3	3	4	3	4	3	3	103
25	PTOto	16509134019	3	3	3	4	3	3	4	4	4	4	3	3	4	3	3	3	3	3	3	4	4	4	3	3	3	3	3	4	3	3	100
26	PTM	14503241038	3	3	3	4	2	3	2	3	3	3	3	2	3	4	3	2	2	3	3	4	3	3	3	3	3	2	3	3	2	3	86
27	PTM	14503241032	3	4	4	3	3	3	3	3	3	4	3	3	3	2	3	2	2	3	2	3	3	2	2	3	3	2	3	3	3	3	86
28	PTM	14503244016	3	3	4	2	3	3	3	3	4	4	4	3	3	4	3	3	3	3	3	4	4	4	4	3	3	3	3	4	3	3	99
29	PTM	14503244008	3	2	3	2	2	2	1	4	2	2	3	3	2	3	2	2	2	1	2	3	2	2	1	2	3	2	1	3	2	2	66
30	PTM	10503241031	4	3	4	3	3	4	4	3	3	3	4	3	3	3	3	3	3	3	3	4	3	4	3	3	4	3	3	3	4	3	99
JUMLAH			103	95	107	93	93	99	101	98	105	104	105	101	101	99	98	91	89	97	93	105	106	104	96	94	99	93	95	100	97	102	2963

## Lampiran 12. Hasil Pengujian *Performa Efficiency*

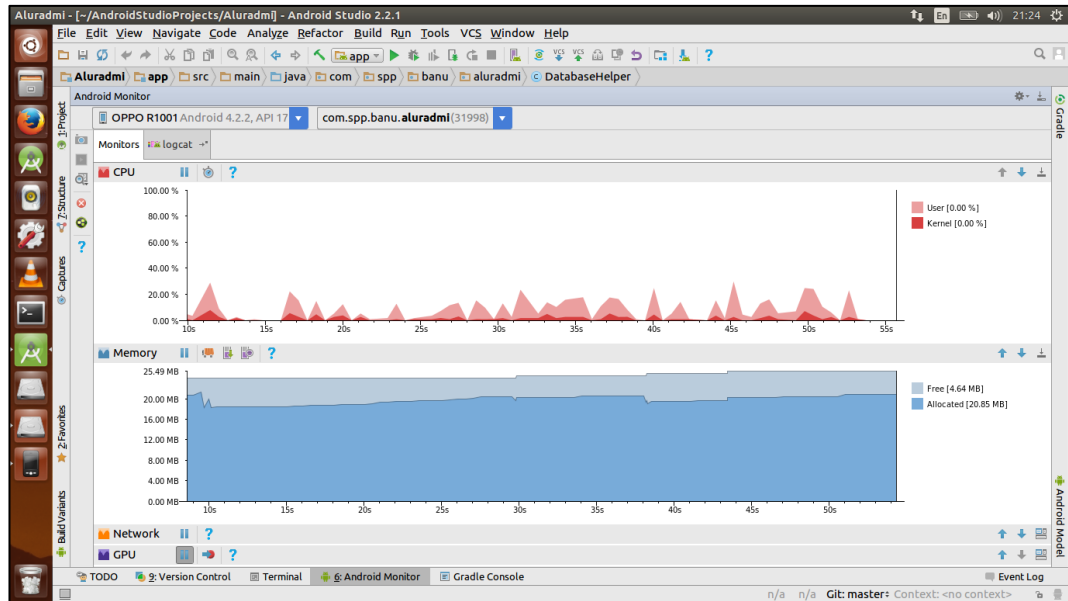


Gambar 66. Tes 1 Uji Penggunaan *Resource*

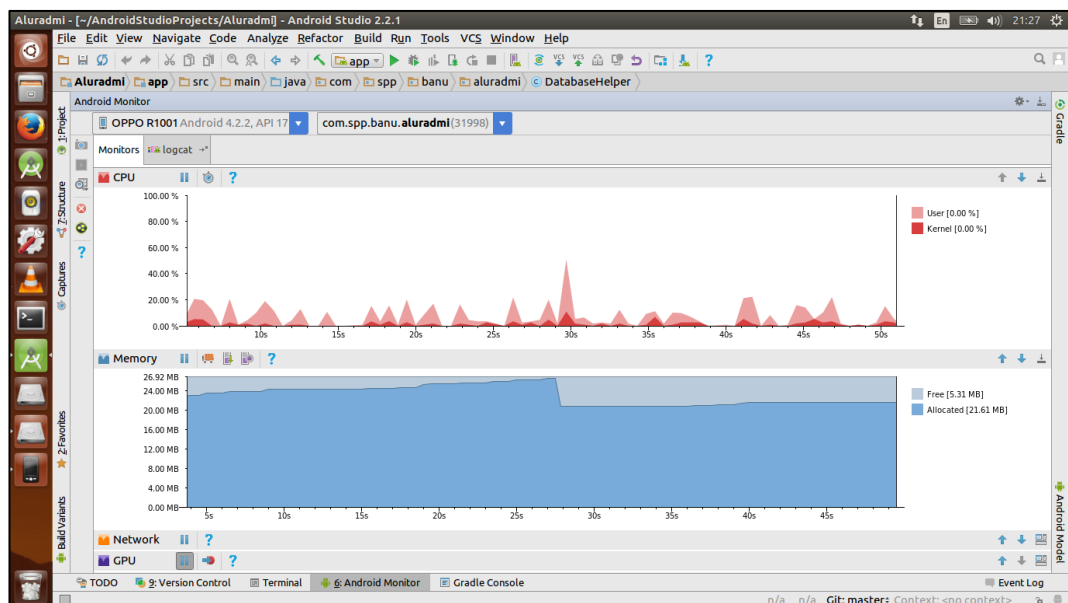


Gambar 67. Tes 2 Uji Penggunaan *Resource*

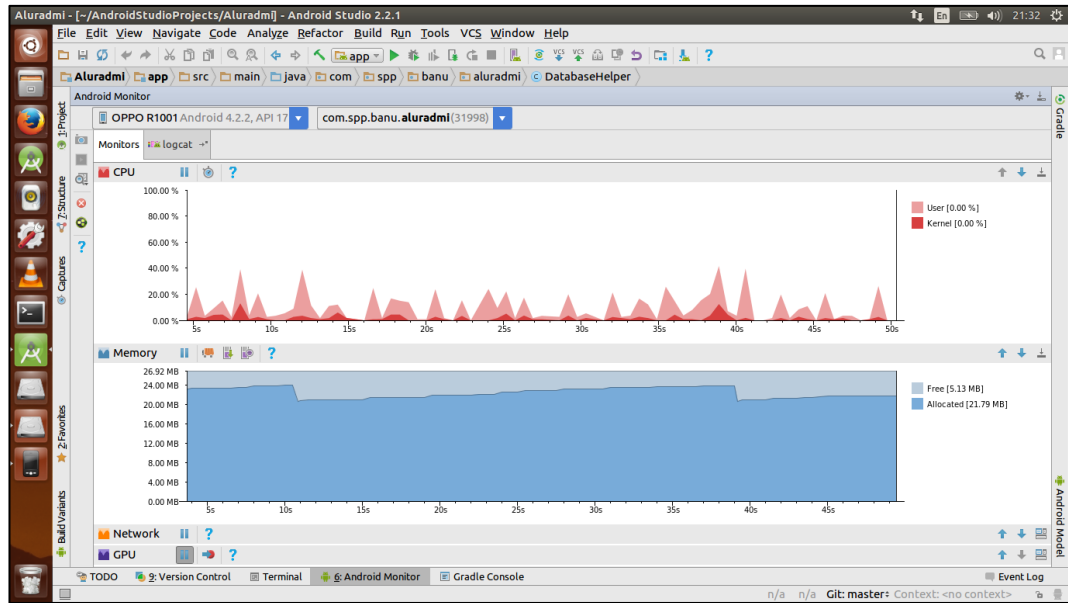




Gambar 68. Tes 3 Uji Penggunaan *Resource*

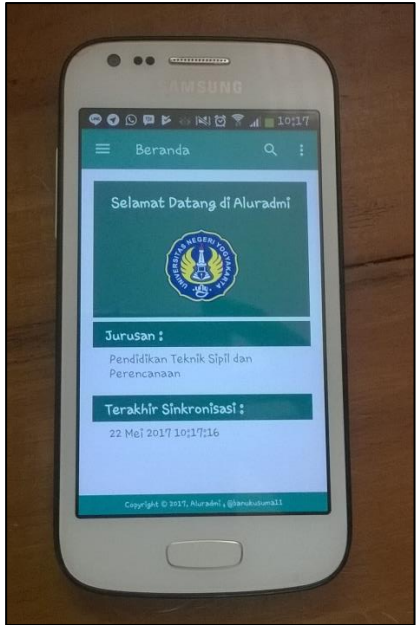
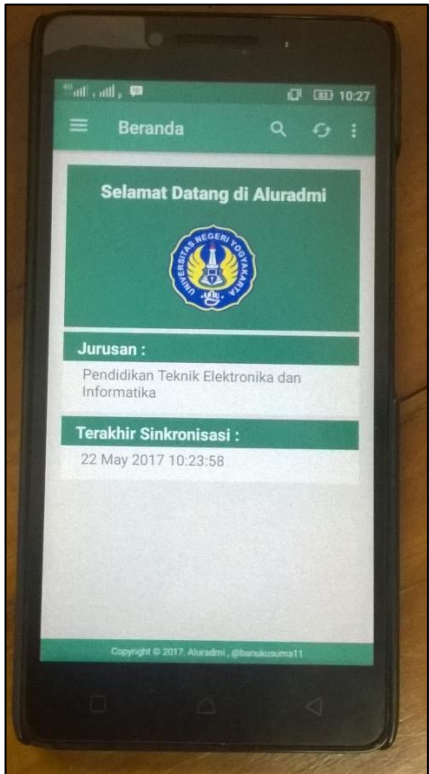


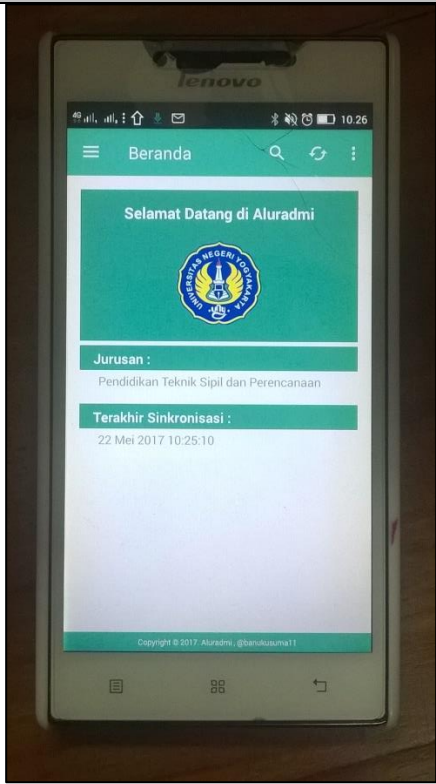
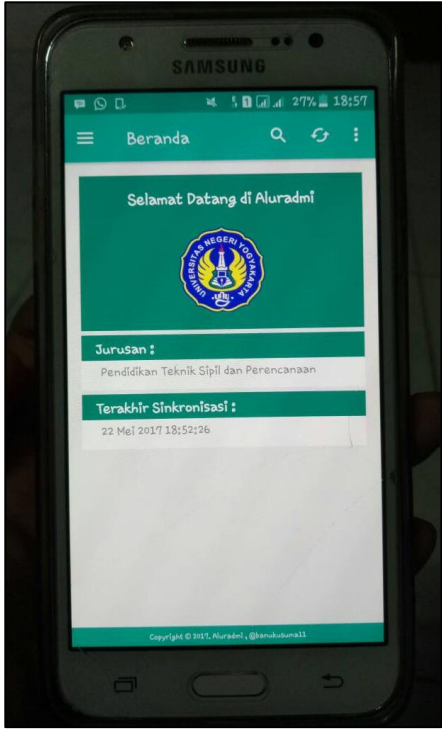
Gambar 69. Tes 4 Uji Penggunaan *Resource*

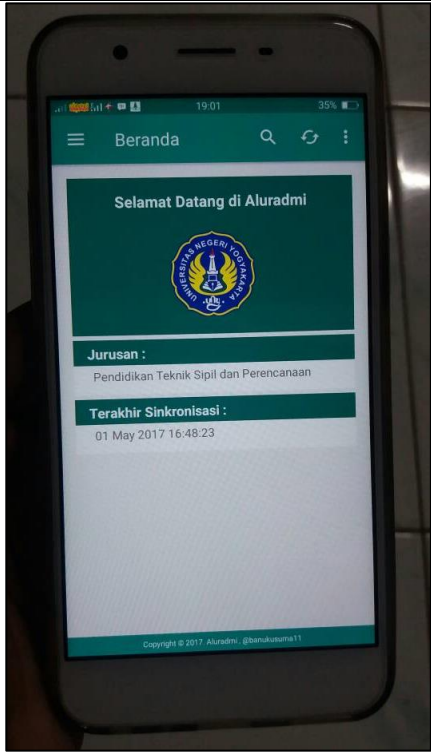
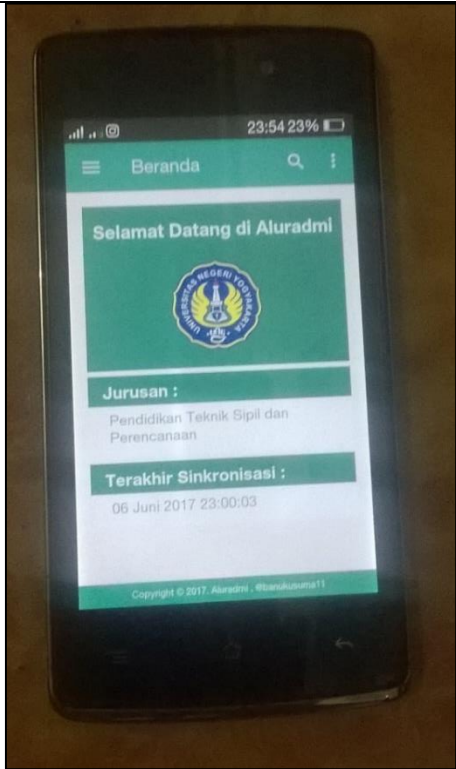



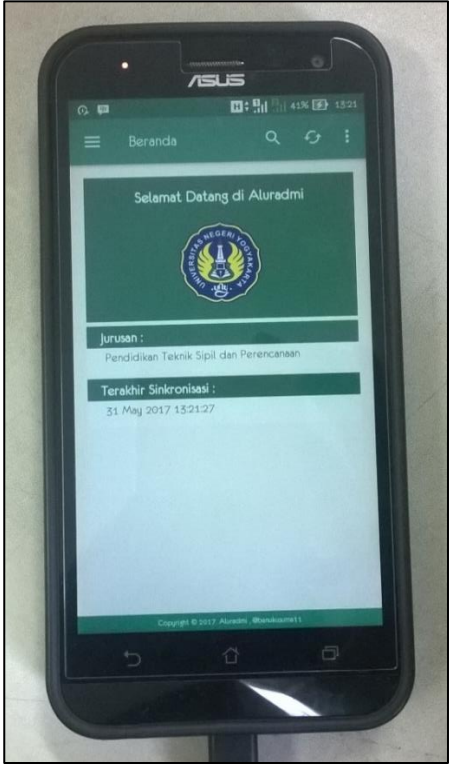
Gambar 70. Tes 5 Uji Penggunaan *Resource*



### Lampiran 13. Hasil Pengujian *Compatibility*

No	Perangkat	Foto
1.	Samsung Galaxy Ace 3	
2.	Lenovo A6010	

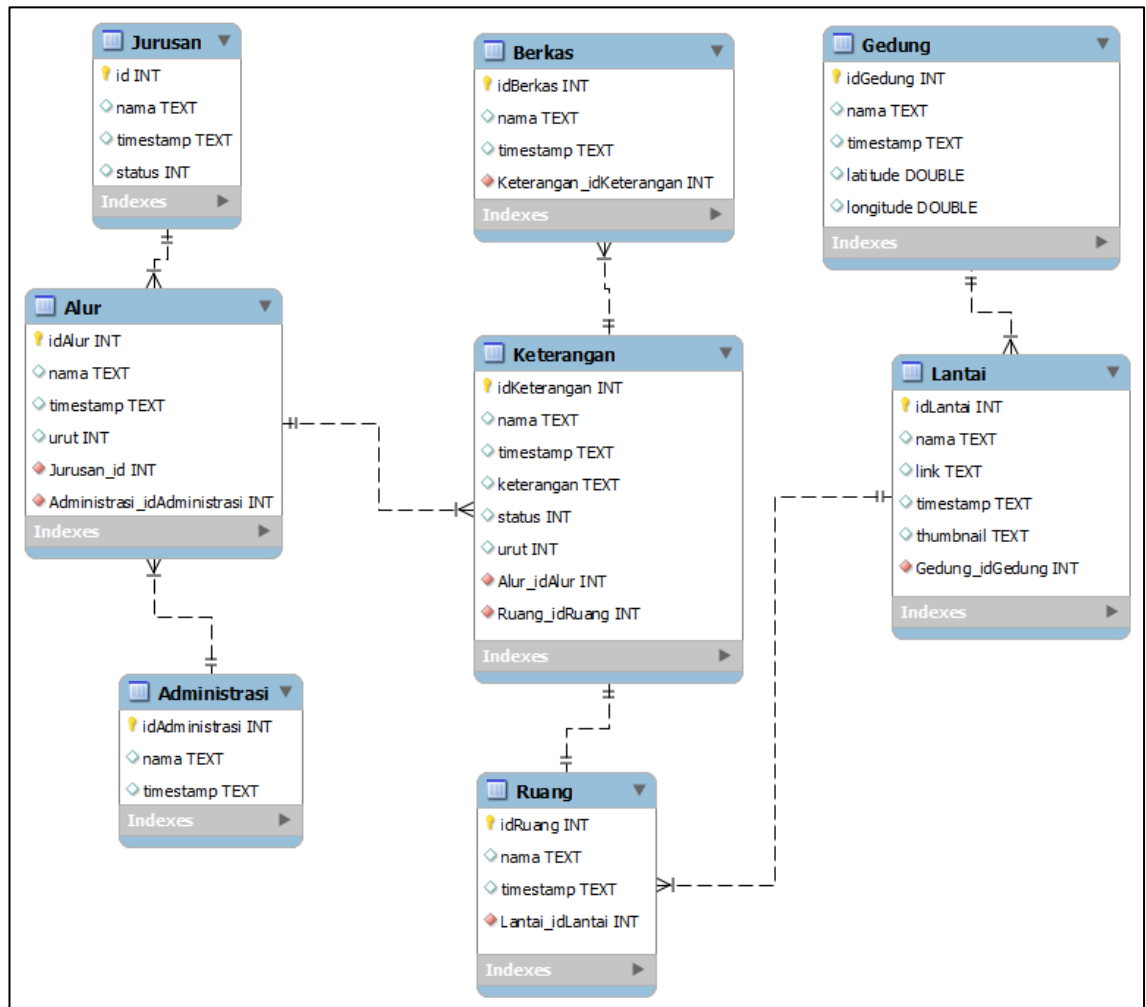
No	Perangkat	Foto
3.	Lenovo P70	
4.	Samsung Galaxy J5	

No	Perangkat	Foto
5.	Oppo A39	
6.	Oppo Joy R1001	

No	Perangkat	Foto
7.	Asus Zenfone 4C	
8.	Asus Zenfone 2 Laser	

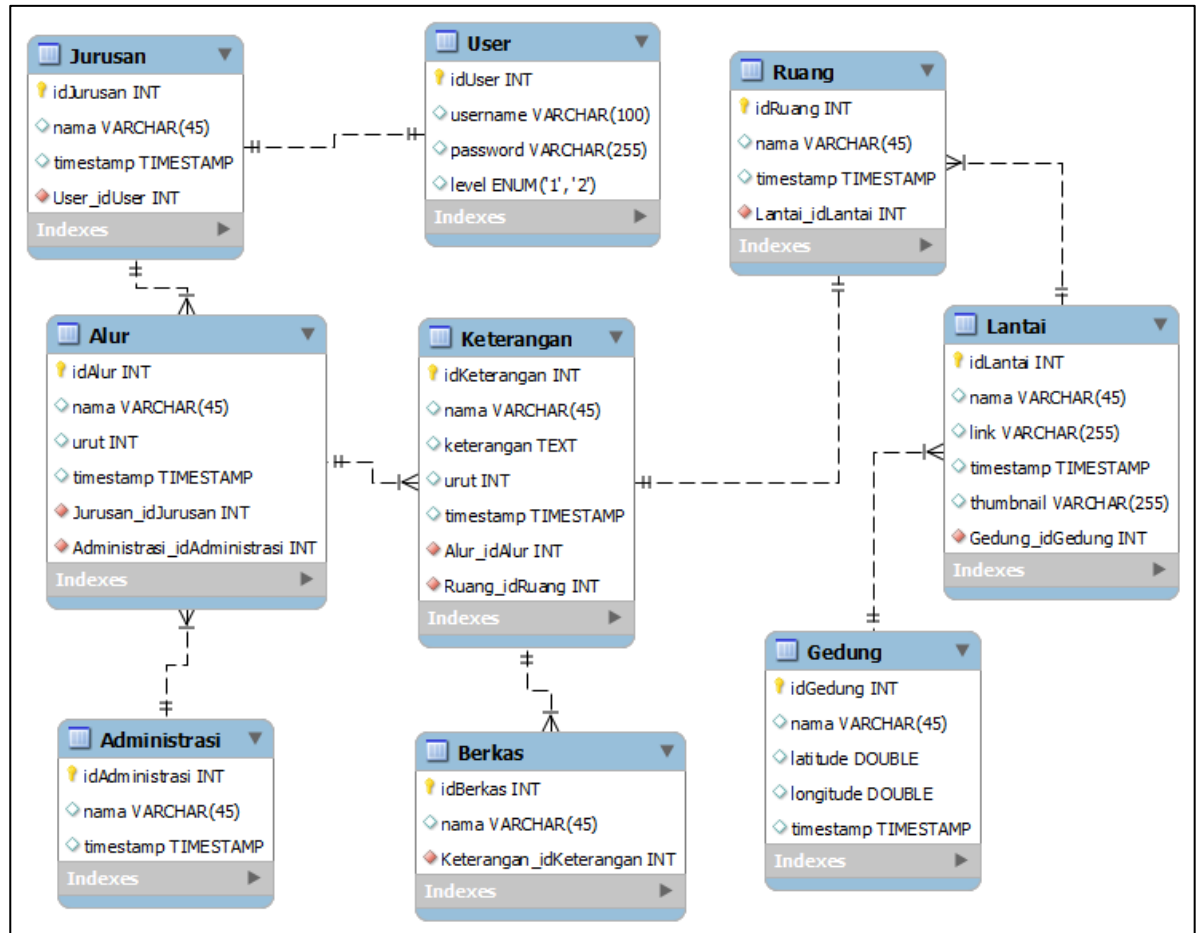
No	Perangkat	Foto
9.	Xiaomi Mi 4C	
10.	Sony Xperia T3	

## Lampiran 14. Desain Basis Data



Gambar 71. Desain Basis Data Aplikasi





Gambar 72. Desain Basis Data *Server*