

**PENGEMBANGAN DAN ANALISIS KUALITAS SISTEM INFORMASI JADWAL
AKADEMIK BERBASIS *YII FRAMEWORK* DI JURUSAN PENDIDIKAN
TEKNIK ELEKTRONIKA 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:

Pradana Setialana

NIM 10520244004

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2014**

LEMBAR PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

**PENGEMBANGAN DAN ANALISIS KUALITAS SISTEM INFORMASI JADWAL
AKADEMIK BERBASIS *YII FRAMEWORK* DI JURUSAN PENDIDIKAN
TEKNIK ELEKTRONIKA FAKULTAS TEKNIK UNIVERSITAS NEGERI
YOGYAKARTA**

Disusun oleh:

Pradana Setialana
NIM 10520244004

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

Yogyakarta, 28 April 2014

Mengetahui,
Ketua Program Studi
Pendidikan Teknik Informatika,



Dr. Ratna Wardani, S.Si., M.T.
NIP. 19701218 200501 2 001

Disetujui,
Dosen Pembimbing,



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

HALAMAN PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Pradana Setialana

NIM : 10520244004

Program Studi : Pendidikan Teknik Informatika

Judul TAS : Pengembangan dan Analisis Kualitas Sistem Informasi

Jadwal Akademik Berbasis *Yii Framework* di Jurusan
Pendidikan Teknik Elektronika 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, 28 April 2014

Yang menyatakan,



Pradana Setialana

NIM. 10520244004

HALAMAN PENGESAHAN

Tugas Akhir Skripsi

**PENGEMBANGAN DAN ANALISIS KUALITAS SISTEM INFORMASI JADWAL
AKADEMIK BERBASIS YII FRAMEWORK DI JURUSAN PENDIDIKAN
TEKNIK ELEKTRONIKA FAKULTAS TEKNIK UNIVERSITAS NEGERI
YOGYAKARTA**




Disusun oleh:

Pradana Setialana

NIM 10520244004

Telah dipertahankan di depan Tim Penguji Tugas Akhir Skripsi Program Studi
Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta
pada tanggal 20 Mei 2014

TIM PENGUJI

Nama/Jabatan	Tanda Tangan	Tanggal
Handaru Jati, Ph.D Ketua Penguji/Pembimbing		02/06/2014
Pipit Utami, M.Pd Sekretaris		02/06/2014
Suparman, M.Pd Penguji		02/06/2014

Yogyakarta, 2 Juni 2014

Fakultas Teknik Universitas Negeri Yogyakarta

Dekan,




Dr. Moch Bruri Triyono

NIP. 19560216 198603 1 003 

HALAMAN MOTTO

Man Jadda Wa Jada – Barang siapa yang bersungguh-sungguh pasti akan berhasil.

Jika anda bekerja semata-mata untuk uang, anda tidak akan menjadi kaya kerananya. Tetapi jika anda mencintai pekerjaan yang anda lakukan itu, kejayaan akan menjadi milik anda - Ray Kroc

Jenius adalah 1 % inspirasi dan 99 % keringat. Tidak ada yang dapat menggantikan kerja keras. Keberuntungan adalah sesuatu yang terjadi ketika kesempatan bertemu dengan kesiapan. - Thomas A. Edison

HALAMAN PERSEMBAHAN

Karya ini dipersembahkan untuk:

- Allah SWT yang telah memberikan kehidupan, kesehatan, keselamatan, rezeki serta semua hal.
- Bapak Alm. Adreng Setiadi dan Ibu Marsiyem yang sangat saya cintai yang telah melahirkan saya, memberikan kasih sayang, mendidik saya, memberikan perlindungan serta selalu dan tak henti-hentinya mendoakan saya hingga saat ini.
- Seluruh keluarga Simbah Maridi dan Simbah Kakung Gelangan yang telah banyak memberikan dukungan serta doanya.
- Pambudi, Fuat, Nuning, Netrin, Tika serta semua teman-teman yang telah banyak membantu dalam penyelesaian skripsi
- Mahasiswa Matakuliah PSBO 2014 yang telah banyak membantu dalam mengisi kuesioner
- Nurissa Febriana yang telah selalu memberikan dukungan dan semangat.
- Seluruh teman - temanku seperjuangan PT. Informatika UNY.
- Seluruh Alamater Universitas Negeri Yogyakarta.

**PENGEMBANGAN DAN ANALISIS KUALITAS SISTEM INFORMASI JADWAL
AKADEMIK BERBASIS *YII FRAMEWORK* DI JURUSAN PENDIDIKAN
TEKNIK ELEKTRONIKA FAKULTAS TEKNIK UNIVERSITAS NEGERI
YOGYAKARTA**

Oleh:

Pradana Setialana

NIM 10520244004

ABSTRAK

Penelitian ini bertujuan untuk (1) mengembangkan sistem informasi jadwal akademik Jurusan Pendidikan Teknik Elektronika Universitas Negeri Yogyakarta; (2) menguji kualitas perangkat lunak yang dikembangkan berdasarkan standar kualitas perangkat lunak ISO 9126 yang mencakup aspek *functionality*, *reliability*, *usability*, *efficiency*, *maintainability* dan *portability*.

Model pengembangan perangkat lunak yang digunakan adalah model *waterfall* dan paradigma pemrograman berorientasi obyek. Tahapan pertama pengembangan perangkat lunak adalah analisis kebutuhan. Tahapan kedua adalah desain yang mencakup pemodelan UML, ERD dan desain *user interface*. Tahap ketiga adalah implementasi yaitu dengan penulisan kode program dengan *Yii Framework*. Tahap keempat adalah pengujian dimana pengujian *functionality* mencakup uji fungsi *software* dan *security* dengan *software Acunetix Web Vulnerability*, pengujian *reliability* menggunakan *software WAPT*, pengujian *usability* menggunakan kuesioner USE, pengujian *efficiency* menggunakan *GTMetrix*, pengujian *maintainability* menggunakan *Source Code SearchEngine* dan pengujian *portability* menggunakan *software BrowseEmAll*.

Berdasarkan hasil analisis dan pengujian dapat diambil kesimpulan yaitu, (1) dihasilkan *software* sistem informasi jadwal Jurusan Pendidikan Elektronika UNY berbasis *Yii Framework* dengan model pengembangan *waterfall* dan paradigma pemrograman berorientasi obyek. *Yii Framework* mempercepat proses pengembangan *software* karena adanya *extensions* dan modul; (2) hasil pengujian perangkat lunak pada aspek *functionality* didapatkan hasil 100% fungsi *software* berjalan baik dan rendah adanya kerentanan (*low vulnerability*). Pengujian *reliability* didapatkan hasil 100%. Pengujian *usability* didapatkan hasil 77,89%, pengujian *efficiency* didapatkan *grade Page Speed* yaitu A, *grade YSlow* yaitu B dan *load time* sebesar 1,64 detik. Pengujian *maintainability* didapatkan nilai *maintainability index* (MI) sebesar 70,03 (*moderate maintable*). Pengujian *portability* didapatkan *software* dapat berjalan di 8 *web browser dekstop* dan 8 *web browser mobile* tanpa terdapat *error*.

Kata kunci: sistem informasi, website, jadwal akademik, *yii framework*, ISO 9126, *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, *portability*.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT Tuhan Yang Maha Esa atas berkat rahmat dan karunia-Nya, Tugas Akhir Skripsi dalam rangka untuk memenuhi sebagian persyaratan untuk mendapatkan gelar Sarjana Pendidikan dengan judul “Pengembangan dan Analisis Kualitas Sistem Informasi Jadwal Akademik Berbasis *Yii Framework* di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta” dapat disusun sesuai dengan harapan. Tugas Akhir Skripsi ini dapat diselesaikan tidak lepas dari bantuan dan kerjasama dengan pihak lain. Berkenaan dengan hal tersebut, penulis menyampaikan ucapan terima kasih kepada yang terhormat:

1. Handaru Jati, Ph.D selaku Dosen Pembimbing TAS yang telah banyak memberikan semangat, dorongan, dan bimbingan selama penyusunan Tugas Akhir Skripsi ini.
2. Astri Ollivia selaku Validator instrumen penelitian TAS yang memberikan saran/masukan perbaikan sehingga penelitian TAS dapat terlaksana sesuai dengan tujuan.
3. Handaru Jati, Ph.D, Pipit Utami, M.Pd dan Suparman, M.Pd selaku Ketua Penguji, Sekretaris, dan Penguji yang memberikan koreksi perbaikan secara komprehensif terhadap TAS ini.
4. Muhammad Munir, M.Pd dan Dr. Ratna Wardani selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Ketua Program Studi Pendidikan Teknik Informatika beserta dosen dan staf yang telah memberikan bantuan dan fasilitas selama proses penyusunan pra proposal sampai dengan selesainya TAS ini.

5. Dr. Moch Bruri Triyono selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang memberikan persetujuan pelaksanaan Tugas Akhir Skripsi.
6. Muhammad Munir, M.Pd selaku Ketua Jurusan Pendidikan Teknik Elektronika Fakultas Teknik UNY yang telah memberi ijin dan bantuan dalam pelaksanaan penelitian Tugas Akhir Skripsi ini.
7. Para mahasiswa Pendidikan Teknik Elektronika dan Pendidikan Teknik Informatika yang telah memberi bantuan memperlancar pengambilan data selama proses penelitian Tugas Akhir Skripsi ini.
8. Semua pihak, secara langsung maupun tidak langsung, yang tidak dapat disebutkan di sini atas bantuan dan perhatiannya selama penyusunan Tugas Akhir Skripsi ini.

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

Yogyakarta, 28 April 2014

Penulis,

Pradana Setialana

NIM 10501244004

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
HALAMAN PERNYATAAN.....	iii
HALAMAN PENGESAHAN	iv
HALAMAN MOTTO	iv
HALAMAN PERSEMBAHAN.....	vi
ABSTRAK.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvii
DAFTAR LAMPIRAN.....	xix
BAB I. PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah	3
C. Batasan Masalah	4
D. Rumusan Masalah.....	4
E. Tujuan Penelitian	4
F. Spesifikasi Produk yang Dikembangkan	5
G. Manfaat Penelitian.....	5
BAB II. KAJIAN PUSTAKA	6
A. Kajian Teori	6
1. Pengertian Sistem Informasi	6
2. Pemrograman	7

3. Bahasa Pemodelan <i>Unified Modeling Language</i> (UML)	12
4. Pengembangan <i>Software Model Waterfall</i>	14
5. Standar Kualitas <i>Software</i>	15
B. Penelitian yang Relevan	27
C. Kerangka Pikir	28
D. Pertanyaan Penelitian	30
BAB III. METODE PENELITIAN	31
A. Model Pengembangan.....	31
B. Prosedur Pengembangan.....	33
1. Analisis Kebutuhan	33
2. Desain	33
3. Implementasi.....	33
4. Pengujian	34
C. Sumber Data / Subyek Penelitian.....	34
D. Metode dan Alat Pengumpul Data.....	35
E. Teknik Analisis Data	35
1. <i>Functionality</i>	35
2. <i>Reliability</i>	37
3. <i>Usability</i>	38
4. <i>Efficiency</i>	40
5. <i>Maintainability</i>	41
6. <i>Portability</i>	42
BAB IV. HASIL DAN PEMBAHASAN	44
A. Tahap Analisis Kebutuhan.....	44
1. Fitur Sistem yang Dikembangkan	44

2. Informasi yang Dibutuhkan	45
3. Spesifikasi Sistem.....	45
B. Tahap Desain	46
1. Use Case Diagram.....	46
2. <i>Class Diagram</i>	49
3. <i>Activity Diagram</i>	50
4. <i>Sequence Diagram</i>	55
5. <i>Entity Relational Diagram (ERD)</i>	59
6. Desain Tampilan <i>User Interface (UI)</i>	60
C. Tahap Implementasi	63
1. Implementasi Halaman Utama.....	63
2. Implementasi Halaman Profil	64
3. Implementasi Halaman Data Kelas.....	64
4. Implementasi Halaman Kelas yang Diikuti.....	65
5. Implementasi Halaman Jadwal Akademik	65
6. Implementasi Halaman Cari Jadwal.....	66
7. Implementasi Halaman Data Ruang	67
8. Implementasi Notifikasi	67
D. Tahap Pengujian	68
1. <i>Functionality</i>	68
2. <i>Reliability</i>	72
3. <i>Usability</i>	74
4. <i>Efficiency</i>	74
5. <i>Maintainability</i>	82
6. <i>Portability</i>	83

E. Pembahasan Hasil Penelitian.....	84
1. <i>Functionality</i>	84
2. <i>Reliability</i>	85
3. <i>Usability</i>	85
4. <i>Efficiency</i>	87
5. <i>Maintainability</i>	88
6. <i>Portability</i>	89
BAB V. SIMPULAN DAN SARAN.....	91
A. Simpulan.....	91
B. Keterbatasan Produk.....	92
C. Pengembangan Produk Lebih Lanjut	93
D. Saran	93
DAFTAR PUSTAKA	94
LAMPIRAN	98

DAFTAR GAMBAR

	Halaman
Gambar 1. Model ISO 9126	16
Gambar 2. Model dan Karakteristik ISO 9126.....	16
Gambar 3. Model <i>Maintainability</i>	24
Gambar 4. <i>Mapping</i> Karakteristik <i>Maintainability</i>	25
Gambar 5. Kerangka Pikir	29
Gambar 6. Model <i>Waterfall</i>	32
Gambar 7. <i>Use Case Diagram</i> Sistem Informasi Jadwal	46
Gambar 8. <i>Class Diagram</i> Sistem Informasi Jadwal Akademik	49
Gambar 9. <i>Activity Diagram Register</i>	50
Gambar 10. <i>Activity Diagram Update Profil</i>	50
Gambar 11. <i>Activity Diagram Lihat Profil User Lain</i>	51
Gambar 12. <i>Activity Diagram Tambah Teman</i>	51
Gambar 13. <i>Activity Diagram Unfollow</i>	52
Gambar 14. <i>Activity Diagram Mencari Jadwal</i>	52
Gambar 15. <i>Activity Diagram Menambah Jadwal</i>	53
Gambar 16. <i>Activity Diagram Mengubah Jadwal</i>	53
Gambar 17. <i>Activity Diagram Menghapus Jadwal</i>	54
Gambar 18. <i>Activity Diagram Mencari Ruang Kosong</i>	54
Gambar 19. <i>Sequence Diagram Cari Teman</i>	55
Gambar 20. <i>Sequence Diagram Follow Teman</i>	55
Gambar 21. <i>Sequence Diagram Unfollow Teman</i>	56
Gambar 22. <i>Sequence Diagram Join Kelas</i>	56

Gambar 23. <i>Sequence Diagram</i> Unjoin Kelas.....	57
Gambar 24. <i>Sequence Diagram</i> Tambah Jadwal	57
Gambar 25. <i>Sequence Diagram</i> Hapus Jadwal	57
Gambar 26. <i>Sequence Diagram</i> Cari Jadwal	58
Gambar 27. <i>Sequence Diagram</i> Cari Ruang Kosong	58
Gambar 28. <i>Entity Relational Diagram</i> (ERD)	59
Gambar 29. Desain UI Halaman Home	60
Gambar 30. Desain UI Halaman Teman	60
Gambar 31. Desain UI Halaman Jadwal Akademik	61
Gambar 32. Desain UI Halaman Jadwal Pribadi	61
Gambar 33. Desain UI Halaman Cari Jadwal.....	62
Gambar 34. Desain UI Halaman Informasi Ruangan	62
Gambar 35. Halaman Utama	63
Gambar 36. Halaman Profil.....	64
Gambar 37. Halaman Data Kelas.....	65
Gambar 38. Halaman Data Kelas yang Diikuti	65
Gambar 39. Halaman Jadwal Akademik	66
Gambar 40. Halaman Cari Jadwal	66
Gambar 41. Halaman Data Ruang	67
Gambar 42. Tampilan Notifikasi Jadwal	68
Gambar 43. Hasil Pengujian Keamanan Sistem	72
Gambar 44. Grafik Hasil <i>Stress Testing</i>	73
Gambar 45. Jumlah <i>Session</i> , <i>Pages</i> dan <i>Hits</i> Setiap Menit.....	73
Gambar 46. Hasil Pengujian Reliabilitas Kuesioner dengan SPSS	74
Gambar 47. Hasil GTMatrix Halaman Utama.....	75

Gambar 48. Hasil GTMatrix Halaman Cari Teman.....	75
Gambar 49. Hasil GTMatrix Halaman <i>Following</i>	76
Gambar 50. Hasil GTMatrix Halaman <i>Follower</i>	77
Gambar 51. Hasil GTMatrix Halaman Data Kelas.....	77
Gambar 52. Hasil GTMatrix Halaman Kelasku.....	78
Gambar 53. Hasil GTMatrix Halaman Jadwal Akademik	79
Gambar 54. Hasil GTMatrix Halaman Waktu Kuliah	79
Gambar 55. Hasil GTMatrix Halaman Jadwal Pribadi.....	80
Gambar 56. Hasil GTMatrix Halaman Cari Jadwal.....	81
Gambar 57. Hasil GTMatrix Halaman Data Ruang	81
Gambar 58. Hasil Pengujian <i>Portability</i> pada Perangkat <i>Desktop</i>	83
Gambar 59. Hasil Pengujian <i>Portability</i> pada Perangkat <i>Mobile</i>	83
Gambar 60. Kontinum Skor Jawaban.....	86

DAFTAR TABEL

	Halaman
Tabel 1. Perbandingan Model Kualitas <i>Software</i> dengan ISO 9126.	15
Tabel 2. Penjelasan Sub Kategori ISO 9126.....	17
Tabel 3. <i>Internal Consistency</i>	21
Tabel 4. Presentase Minat <i>User</i> Berdasarkan <i>Load Time</i>	23
Tabel 5. Rata-Rata Kecepatan Akses <i>Website</i>	23
Tabel 6. <i>Maintainability Index</i>	25
Tabel 7. <i>SQL Injection String</i>	36
Tabel 8. Angket <i>Usefulness, Satisfaction, and Ease of use (USE)</i>	38
Tabel 9. Keterangan Poin Skala Likert.....	40
Tabel 10. Teknik Analisis Data Setiap Aspek ISO 9126.....	43
Tabel 11. Definisi Aktor	46
Tabel 12. Definisi <i>Use Case Manage</i> Teman.....	46
Tabel 13. Definisi <i>Use Case Manage</i> Kelas	47
Tabel 14. Definisi <i>Use Case Manage</i> Jadwal.....	47
Tabel 15. Definisi <i>Use Case Manage</i> Ruangan	48
Tabel 16. Hasil Pengujian <i>Functionality</i>	68
Tabel 17. Jumlah Total <i>Success</i> dan <i>Failed</i>	73
Tabel 18. Jumlah Jawaban Poin Skala Likert.....	74
Tabel 19. Hasil Pengujian dengan Source Code SearchEngine	82
Tabel 20. Hasil Uji Fungsi <i>Software</i>	84
Tabel 21. Perbandingan Hasil dengan Standar <i>Functionality</i>	85
Tabel 22. Persentase Keberhasilan dalam <i>Stress Testing</i>	85

Tabel 23. Jumlah Skor Skala Likert.....	86
Tabel 24. Hasil Pengujian GTMetrix.....	87
Tabel 25. Perbandingan Hasil dengan Standar Kualitas <i>Maintainability</i>	88
Tabel 26. Tabel Hasil Pengujian <i>Portability</i>	89
Tabel 27. Perbandingan Hasil Pengujian dengan Standar <i>Portability</i>	90

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Keputusan Pembimbing.....	99
Lampiran 2. Surat Permohonan Ijin Observasi.....	100
Lampiran 3. Surat Permohonan Ijin Penelitian	101
Lampiran 4. Instrumen <i>Functionality</i>	102
Lampiran 5. Hasil Pengujian <i>Functionality</i>	105
Lampiran 6. Validasi Bahasa Instrumen <i>Usability</i>	108
Lampiran 7. Hasil Kuesioner <i>Usability</i>	111

BAB I

PENDAHULUAN

A. Latar Belakang

Universitas Negeri Yogyakarta (UNY) merupakan salah satu perguruan tinggi yang membutuhkan jadwal untuk manajemen waktu jalannya kegiatan perkuliahan. Berdasarkan hasil pengamatan yang dilakukan di Fakultas Teknik UNY, jadwal kegiatan kuliah yang disediakan ditulis dalam lembaran kertas yang kemudian ditempel di papan informasi. Selain dalam bentuk kertas, jadwal biasanya disebarluaskan dalam bentuk *file Portable Document Format* (PDF) yang kemudian diunggah di website. Tujuan dari adanya jadwal tersebut dimaksudkan untuk mempermudah mahasiswa dalam merencanakan kegiatan perkuliahan yang akan dilaksanakan. Penulisan jadwal dalam bentuk kertas maupun *file PDF* tersebut merupakan bentuk konvensional dalam penyebaran informasi. Permasalahan yang timbul dari bentuk jadwal tersebut adalah informasi yang statis sehingga perubahan pada informasi jadwal tidak dapat diterima dengan cepat.

Selain informasi yang statis, berdasarkan hasil pengamatan di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik UNY (Pend. Teknik Elektronika FT UNY) menunjukkan bahwa jadwal yang tersedia tidak memberikan informasi aktivitas setiap dosen secara detail karena informasi yang ada hanya jadwal kuliah yang dikelompokkan berdasarkan kelas dan dosen. Hal tersebut menyulitkan mahasiswa yang ingin mencari dosen. Hal ini dibuktikan dengan hasil wawancara yang dilakukan pada mahasiswa di Jurusan Pend. Teknik Elektronika FT UNY menyatakan bahwa mahasiswa mengalami kesulitan dalam mencari dosen dikarenakan tidak adanya informasi yang lengkap mengenai jadwal pelaksanaan kegiatan dosen.

Tugas seorang dosen bukan hanya melakukan pengajaran dan pendidikan tetapi juga penelitian dan pengabdian kepada masyarakat sehingga menyebabkan mahasiswa kesulitan dalam menemui dosen. Hal tersebut berpengaruh terhadap pelaksanaan bimbingan yang dilakukan untuk mahasiswa seperti bimbingan skripsi. Berdasarkan penelitian yang dilakukan Mujiyah yang tertuang dalam Sigit Suryadi (2008), kendala eksternal dalam penyelesaian skripsi yang berasal dari dosen pembimbing skripsi meliputi sulit ditemui sebesar (36,7%), minimnya waktu bimbingan sebesar (23,3%), kurang koordinasi dan kesamaan persepsi antara pembimbing satu dan pembimbing dua sebesar (23,3%), kurang jelas memberi bimbingan sebesar (26,7%), dan dosen terlalu sibuk sebesar (13,3%). Kesulitan tersebut bertambah ketika mahasiswa ingin bertemu lebih dari satu dosen dikarenakan mahasiswa harus mencari waktu senggang yang bersamaan pada setiap dosen.

Selain kesulitan mencari dosen, mahasiswa juga mengalami kesulitan dalam mencari ruangan kosong serta waktu yang tersedia untuk suatu ruangan karena sedikitnya informasi jadwal penggunaan ruangan. Berdasarkan wawancara yang dilakukan pada mahasiswa Program Studi Pend. Teknik Informatika FT UNY menyatakan bahwa mahasiswa sulit mencari ruangan yang akan digunakan untuk kegiatan diluar jadwal perkuliahan rutin misalnya untuk melaksanakan *workshop*, kegiatan perkuliahan diluar jadwal rutin atau diskusi bersama dosen. Untuk mengetahui adanya ruangan kosong, mahasiswa harus melihat setiap ruangan satu persatu sehingga memakan waktu dan tenaga.

Bentuk jadwal yang berupa kertas dan *file PDF* juga menyulitkan dosen jika mengganti waktu mengajar pada saat tertentu. Menurut keterangan dari dosen Jurusan Pendidikan Teknik Elektronika Fakultas Teknik UNY menyatakan bahwa

sulitnya mengatur jadwal pengganti kuliah karena harus menyesuaikan dengan jadwal kelas dan jadwal sendiri. Permasalahan bertambah jika dosen belum dapat bertemu dengan mahasiswa di kelas tersebut untuk saling berdiskusi mengenai waktu pengganti. Selain itu, informasi perubahan jadwal juga harus disebarluaskan yang tentu saja menyebabkan tidak efektifnya penyebaran informasi. Hal tersebut tentunya merepotkan dosen ketika terjadi perubahan jadwal mengajar.

Permasalahan bentuk jadwal tersebut dikarenakan tidak tersedianya sistem informasi yang mampu memberikan informasi jadwal secara dinamis yang sesuai dengan karakteristik jadwal di UNY. Selain itu, belum tersedianya sistem informasi yang dikembangkan dengan *Yii Framework* dan memenuhi standar kualitas perangkat lunak ISO 9126. Berdasarkan permasalahan tersebut maka diperlukan suatu sistem informasi jadwal akademik yang dikembangkan dengan *Yii Framework* sehingga mempermudah dalam penggantian jadwal, penyebaran informasi, pencarian informasi jadwal dosen dan penggunaan ruangan. Sistem informasi tersebut juga harus memenuhi standar kualitas suatu *software* ISO 9126.

B. Identifikasi Masalah

1. Jadwal yang berbentuk kertas dan *file PDF* sehingga informasi yang ada bersifat statis dan tidak dapat mengetahui perubahan jadwal secara cepat.
2. Tidak adanya informasi jadwal penggunaan ruangan kuliah sehingga menyulitkan untuk mencari ruangan kosong untuk kegiatan - kegiatan tertentu di luar jadwal kuliah rutin.
3. Dosen mengalami kesulitan menentukan waktu pengganti kuliah di suatu kelas dikarenakan harus berkoordinasi dengan kelas agar tidak bertabrakan dengan jadwal mata kuliah lain.

4. Tidak terdapatnya sistem informasi jadwal digital yang dapat mempermudah dalam penyebaran jadwal, pencarian dosen dan informasi penggunaan ruang.
5. Tidak terdapatnya sistem informasi jadwal akademik di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta (FT UNY) yang dikembangkan dengan *Yii Framework*.
6. Tidak terdapatnya sistem informasi akademik yang memenuhi standar kualitas *software* ISO 9126.

C. Batasan Masalah

1. Pengembangan sistem informasi jadwal akademik di Jurusan Pendidikan Teknik Elektronika FT UNY berbasis *Yii Framework*.
2. Analisis kualitas sistem informasi jadwal akademik Jurusan Pendidikan Teknik Elektronika FT UNY berdasarkan standar kualitas *software* ISO 9126.

D. Rumusan Masalah

1. Bagaimana mengembangkan sistem informasi jadwal akademik berbasis *Yii Framework* Jurusan Pendidikan Elektronika FT UNY?
2. Bagaimana kualitas sistem informasi jadwal akademik Jurusan Pendidikan Teknik Elektronika FT UNY berdasarkan standar kualitas *software* ISO 9126?

E. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Untuk mengembangkan sistem informasi jadwal akademik berbasis *Yii Framework* di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik UNY.
2. Untuk mengetahui kualitas sistem informasi jadwal akademik Jurusan Pendidikan Teknik Elektronika FT UNY berdasarkan standar kualitas *software* ISO 9126.

F. Spesifikasi Produk yang Dikembangkan

Adapun spesifikasi produk yang dikembangkan adalah :

1. Sistem informasi berbasis web dengan menggunakan *Yii PHP Framework* dan *database MySQL*.
2. *Database* yang digunakan berisi informasi data mahasiswa, dosen, matakuliah, jadwal kuliah dan ruang di Jurusan Pendidikan Elektronika Fakultas Teknik Universitas Negeri Yogyakarta.
3. Fitur yang dimiliki sistem informasi ini adalah manajemen jadwal, pencarian ruang, pencarian jadwal dan pencarian lokasi dosen berdasarkan jadwal.

G. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat bagi :

1. Universitas Negeri Yogyakarta untuk meningkatkan pelayanan dengan menyediakan informasi jadwal yang dinamis dan mempermudah pencarian baik dosen, mahasiswa maupun ruangan.
2. Pembaca untuk menambah pengetahuan mengenai perancangan *software* berbasis web dengan *Yii Framework*.
3. Peneliti untuk mengetahui pengujian kualitas suatu *software* berbasis web.

BAB II

KAJIAN PUSTAKA

A. Kajian Teori

1. Pengertian Sistem Informasi

Informasi adalah data yang sudah diolah sehingga menjadi sesuatu yang bermakna dan berguna bagi manusia. Sedangkan sistem informasi adalah komponen yang saling terikat dan bekerjasama untuk mengumpulkan, memproses, menyimpan, menyebarluaskan informasi sehingga mendukung dalam mempermudah pengambilan keputusan (Laudon, 2012:15). Rainer (2011:38) menjelaskan bahwa sistem informasi terdiri dari berbagai kegiatan yaitu mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi.

Menurut Rainer (2011:38), perkembangan teknologi komputer yang cepat dan semakin kompleksnya informasi yang ada memunculkan istilah baru yaitu *computer-based information system* (CBIS) atau sistem informasi berbasis komputer. Sistem informasi berbasis komputer adalah sistem informasi yang menggunakan komputer dalam menyelesaikan beberapa tugas atau keseluruhan tugas. Pada saat ini, hampir semua sistem informasi menggunakan komputer sehingga istilah sistem informasi berbasis komputer bersinonim dengan sistem informasi (Rainer, 2011:38).

Berdasarkan pendapat tersebut maka dapat disimpulkan bahwa sistem informasi adalah suatu kesatuan sistem atau komponen yang saling bekerjasama dalam memproses dan mengolah data menjadi informasi dimana pada saat ini hampir semua proses dilakukan dengan bantuan komputer. Suatu informasi dapat

dengan mudah dan cepat dikumpulkan, disebarluaskan ataupun disimpan dengan bantuan dari sistem informasi. Sehingga sistem informasi merupakan suatu sistem yang berperan penting dalam pengolahan data menjadi informasi.

2. Pemrograman

a. *PHP: Hypertext Preprocessor* (PHP)

PHP merupakan salah satu dari berbagai macam pemrograman web. Menurut Loka (2010:3), PHP awalnya ditemukan pada 1995 oleh Rasmus Lerdorf ketika ingin mengetahui jumlah pengunjung yang membaca *resume* onlinenya. *Script* yang dibuat Rasmus Lerdorf tersebut menarik *developer* sehingga Rasmus mengembangkan PHP menjadi suatu bahasa tersendiri yang mungkin dapat mengkonversikan data yang diinputkan melalui *form* HTML menjadi suatu variabel, yang dapat dimanfaatkan oleh sistem lainnya. Loka (2010:3) menerangkan bahwa pengguna PHP yang semakin banyak membuat Zeev Suraski dan Andi Gutsman selaku *core developer* (*programmer* inti) mencoba untuk menulis ulang *PHP Parser*, dan diintegrasikan dengan menggunakan *Zend scripting engine*, dan mengubah jalan alur operasi PHP.

Banyaknya pengguna PHP dalam pemrograman web bukan tanpa alasan. Butzon (2002) menjelaskan bahwa PHP adalah pilihan tepat dalam pemrograman web dibandingkan dengan bahasa pemrogram lainnya. Menurut Butzon (2002:8), PHP merupakan pemrograman dengan biaya rendah dibandingkan ASP dan memiliki waktu eksekusi yang lebih cepat dibandingkan Java. PHP sebagai bahasa pemrograman web memiliki berbagai keunggulan. Loka (2010:4) menjelaskan berbagai macam keunggulan PHP yaitu,

1) *Open Source*

PHP merupakan pemrograman yang bebas digunakan dan dikembangkan tanpa harus membayar lisensi.

2) *Cross Platform*

PHP dapat dijalankan hampir di semua sistem operasi.

3) Mendukung banyak database

PHP mendukung banyak database seperti MySQL, PostgreSQL, Oracle dan masih banyak lagi.

4) *On The Fly*

PHP memiliki kemampuan untuk membuat dokumen word, excel, PDF dan masih banyak lagi.

Dari berbagai macam kelebihan dari PHP tersebut dapat diambil kesimpulan bahwa PHP merupakan bahasa pemrograman web yang cepat, mudah, gratis dan dapat mendukung banyak database. Sehingga PHP merupakan bahasa pemrograman yang baik digunakan dalam pengembangan *software* berbasis *website* dibandingkan bahasa pemrograman yang lain.

b. *Yii Framework*

Menurut Winesett (2012:9), *web framework* digunakan untuk membantu proses pembuatan aplikasi berbasis web dengan menggunakan fungsi yang telah disediakan dalam *framework* tersebut. Aplikasi yang dibangun menggunakan suatu *framework* akan memiliki sifat-sifat dan kelebihan dari *framework* tersebut. Terdapat banyak *framework* untuk setiap bahasa pemrograman, salah satunya dari pemrograman PHP adalah *Yii Framework*.

Yii merupakan singkatan dari *Yes It Is* adalah *framework* yang diciptakan oleh Qiang Xue mulai tanggal 1 Januari 2008 dan dirilis tanggal 3 Desember 2008

(Winesett, 2012:9). Menurut Winesett (2012:9), Yii merupakan *framework* aplikasi berbasis web yang dibangun dengan dasar PHP 5 sehingga memiliki kinerja yang tinggi, mudah, efisien dan *extensible*. Konsep yang digunakan *Yii Framework* adalah *DRY (Don't Repeat Yourself)* yang berarti bahwa tidak ada duplikasi kode dan fungsi dalam Yii sehingga suatu fungsi yang telah dibuat dapat digunakan oleh kelas-kelas lain (Winesett, 2012:10).

Berbagai macam kelebihan tersebut menjadikan *Yii Framework* sebagai *framework* PHP yang tepat digunakan untuk pengembangan software berbasis web dengan mudah, cepat dan efisien karena konsep DRY dalam *Yii Framework*. Konsep DRY dalam *Yii Framework* tersebut menyebabkan kode program dapat tersusun dengan rapi karena tidak terdapatnya duplikasi kode dan fungsi yang sama.

c. *HyperText Markup Language (HTML)*

HTML (*HyperText Markup Language*) menurut Robbins (2007:165) adalah bahasa pemrograman untuk membuat halaman web. HTML ditemukan buat oleh Tim Berners-Lee pada tahun 1991 di laboratorium CERN. Robbins (2007:165) mengungkapkan awalnya HTML ditujukan untuk berbagai dokumen penelitian melalui sistem *hypertext*. Perkembangan dari HTML adalah XHTML (*extensible HTML*) yang memiliki aturan yang sama. Menurut Robbins (2007:165) pemrograman HTML di standarisasi oleh *World Wide Web Consortium (W3C)* agar dapat pada banyak *browser*. Perkembangan yang pesat dari HTML tersebut menjadikan HTML sebagai pemrograman yang tepat digunakan sebagai halaman tampilan dari web. Selain itu, HTML juga merupakan pemrograman yang telah menjadi standar dalam W3C sehingga dapat digunakan diberbagai macam *web browser*.

d. *Cascading Style Sheets (CSS)*

Cascading Style Sheets (CSS) merupakan suatu bahasa yang diciptakan untuk memungkinkan desainer menempatkan informasi *styling* untuk seluruh situs web ke dalam file eksternal terpisah (Jenkins, 2009:273). Menurut Jenkins (2009:273), tujuan dari dibuatnya CSS adalah untuk mengembangkan tampilan HTML dan memperkecil ukuran *file* serta mengurangi jumlah kode pada HTML. Menurut Jenkins (2009:274), CSS memiliki berbagai keunggulan diantaranya,

- 1) Mempercepat membuka halaman
- 2) Meningkatkan *site access* untuk pengunjung dengan *device* berkemampuan terbatas.
- 3) Meningkatkan manajemen visual dan modifikasi *style* tampilan web.
- 4) Mempermudah perawatan setelah web dipublikasi.

Berbagai macam keunggulan dari CSS tersebut menjadikan CSS sebagai pemrograman yang tepat digunakan untuk mendesain tampilan halaman web menjadi lebih indah.

e. JavaScript

Pemrograman JavaScript tidak sama dengan pemrograman Java. Menurut Haverbeke (2011:6), nama JavaScript dibuat mirip dengan Java dengan pertimbangan pemasaran. Pada tahun 1995 bahasa pemrograman Java sangat populer. Untuk itu nama javascript dibuat mirip java ketika diperkenalkan oleh Netscape. Javascript merupakan bahasa script yang berjalan pada *browser* (Haverbeke, 2011:6).

JavaScript merupakan bahasa pemrograman yang digunakan untuk mendukung suatu halaman web sehingga dapat memberikan animasi visual pada halaman web. Oleh karena itu, JavaScript merupakan bahasa pemrograman yang

tepat digunakan dalam pengembangan web bersama dengan bahasa pemrograman lain seperti PHP, HTML dan CSS.

f. MySQL

Loka (2010:6) mendefinisikan MySQL adalah basis data (*database*). *Database* menurut Loka (2010:6) merupakan suatu jalan untuk dapat menyimpan berbagai informasi dengan membaginya berdasarkan kategori-kategori tertentu. Dimana informasi-informasi tersebut saling berkaitan, satu dengan yang lainnya. Loka (2010) menjelaskan bahwa MySQL bersifat *Relational Database Management System* (RDBMS). RDBMS memungkinkan seorang *admin* dapat menyimpan banyak informasi ke dalam tabel, dimana tabel tersebut saling berkaitan satu sama lain. Keuntungan RDBMS sendiri adalah kita dapat memecah *database* ke dalam tabel - tabel yang berbeda. setiap tabel memiliki informasi yang berkaitan dengan tabel yang lainnya (Loka, 2010:6).

Menurut Loka (2010:6), MySQL memiliki berbagai keunggulan sebagai RDMS yaitu,

- 1) Gratis
- 2) *Cross platform*
- 3) Lengkap dan cepat

MySQL merupakan basis data tipe RDBMS yang cocok digunakan dalam *software* berbasis web karena kemampuannya untuk dapat digunakan dalam platform yang berbeda, cepat dan sifatnya yang gratis. Selain itu, MySQL juga didukung oleh banyak *web server* sehingga MySQL merupakan database yang tepat digunakan untuk website.

g. *Asynchronous JavaScript and XML (AJAX)*

York (2009:219) mendefinisikan *AJAX* sebagai teknologi yang mencakup kemampuan untuk membuat *HTTP requests* dari javascript untuk memperoleh data baru tanpa proses *reload* halaman web. *AJAX* merupakan singkatan dari *Asynchronous JavaScript and XML* tetapi menurut York (2009:219) singkatan tersebut menyesatkan karena *XML* merupakan format opsional yang bisa diganti dengan format lain seperti *JSON*.

York (2009:219) menjelaskan berbagai keuntungan dari penggunaan *AJAX* yaitu,

- 1) Halaman web sebagai aplikasi bukan dokumen.
- 2) Dapat berjalan disemua *platform (cross platform)*.
- 3) Banyak *browser* mendukung.

AJAX merupakan teknologi yang membuat halaman web menjadi interaktif sehingga halaman web tersebut dapat berfungsi tanpa proses *reload*. Halaman web yang menggunakan *AJAX* akan membuat web menjadi suatu *web application* atau aplikasi berbasis web.

3. Bahasa Pemodelan *Unified Modeling Language (UML)*

Desain sistem merupakan hal penting yang harus dilakukan dalam proses perancangan sistem maupun *software*. Menurut Hamilton (2006:2), desain sistem akan menghasilkan model yang mempermudah dalam pengembangan sistem secara berkelompok, manajemen kompleksitas sistem, membuat sistem sesuai dengan analisis kebutuhan serta menghasilkan dokumen proses pengembangan *software*. Terdapat berbagai macam bahasa pemodelan yang salah satunya adalah *Unified Modeling Language (UML)*. *UML* merupakan standar bahasa pemodelan yang terdiri dari berbagai macam diagram-diagram dan digunakan dalam pengembangan *software* (Hamilton, 2006:9). Menurut Alhir (2003), *UML*

adalah bahasa visual yang memodelkan komunikasi tentang sistem yang baik digunakan pada sistem berorientasi obyek.. Diagram-diagram dalam bahasa pemodelan UML tersebut diantaranya adalah *use case*, *activity*, *class* dan *sequence*.

a. Use case diagram

Menurut Hamilton (2006:20), diagram *use case* merupakan diagram yang memodelkan interaksi antara sistem dengan pengguna. Selain memodelkan interaksi, *use case* juga sangat membantu dalam memetakan kebutuhan dalam sistem. *Use case* termasuk dalam pemodelan struktural yang mendeskripsikan kebutuhan fungsional seperti keamanan, fungsionalitas, memasukan data, proses data, menghasilkan laporan dan lainnya (Alhir, 2003).

b. Activity Diagram

Activity diagram merupakan diagram yang menunjukkan spesifikasi bagaimana suatu sistem mencapai tujuannya (Hamilton, 2006:43). Menurut Alhir (2003), *activity diagram* termasuk dalam *behavioral modeling* yang memodelkan aktivitas dan jalannya elemen dalam sistem.

c. Sequence Diagram

Sequence diagram merupakan bagian dari diagram interaksi yaitu diagram yang menggambarkan interaksi antara sistem dengan tampilan (Hamilton, 2006:108). *Sequence diagram* merupakan bagian dari *behavior modeling* yang menggambarkan perilaku dinamis dari unsur-unsur pembentuk sistem yang selalu berinteraksi (Alhir, 2003). Menurut Alhir (2003), *sequence diagram* digambarkan dalam garis sumbu horisontal dan vertikal. Garis sumbu horisontal menggambar elemen sistem yang berubah ketika berinteraksi sedangkan garis sumbu vertikal sistem menggambarkan waktu dalam proses interaksi.

d. Class Diagram

Menurut Alhir (2003), *class diagram* termasuk dalam pemodelan struktural yang menunjukkan tentang struktur umum dari suatu sistem. Dalam bahasa pemodelan UML, *class diagram* digambarkan dengan garis berbentuk kotak yang terbagi menjadi tiga bagian dan dipisahkan oleh garis horisonal. Bagian paling atas dari kelas diagram menunjukkan nama kelas, bagian tengah menunjukkan daftar *attribute* dan dibagian paling bawah menunjukkan operasi (Alhir, 2003).

4. Pengembangan Software Model Waterfall

Menurut Munassar (2010), proses pengembangan *software* terdiri dari berbagai kegiatan yang mencakup spesifikasi, desain, validasi dan evolusi. Proses pengembangan *software* tersebut kemudian terbagi menjadi berbagai model pengembangan *software* yang antara lain adalah model *waterfall*, *iteration*, *V-shaped*, *spiral* dan *extreme*. Dari berbagai macam model pengembangan tersebut, model *waterfall* merupakan model pengembangan *software* tertua dan paling banyak digunakan dalam instansi pemerintahan maupun perusahaan besar.

Model *waterfall* atau model air terjun adalah model klasik dalam rekayasa perangkat lunak. Model ini menekankan pada perencanaan tahap awal dan pembuatan desain sebelum mulai mengembangkan *software* (Munassar, 2010). Model *waterfall* terdiri dari berbagai tahap yang harus dilakukan secara berurutan yaitu *system requirements*, *software requirements*, *architectural design*, *detailed design*, *coding*, *testing* dan *maintance*. Menurut Munassar (2010), model pengembangan *waterfall* memiliki beberapa keunggulan yaitu,

- a. Mudah dipahami dan diimplementasikan
- b. Banyak diketahui dan digunakan
- c. Menghasilkan dokumentasi yang baik

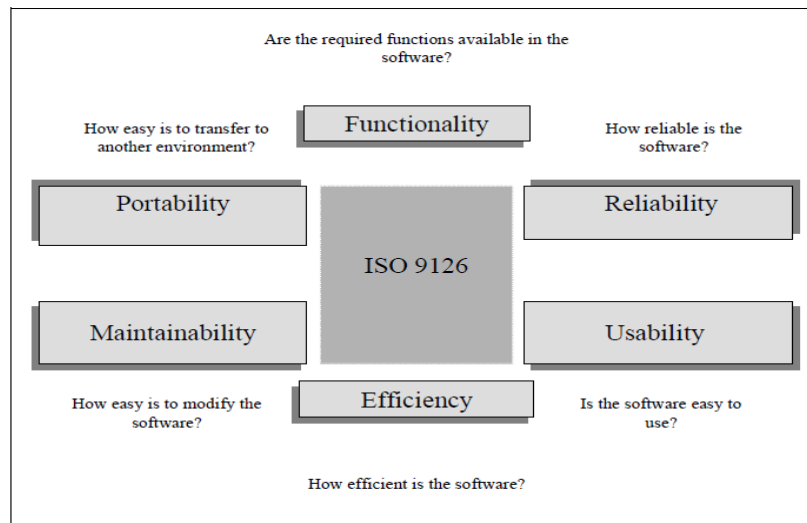
5. Standar Kualitas *Software*

Menurut Fahmy (2012), ISO 9126 adalah standar internasional untuk evaluasi *software*. ISO 9126 merupakan perkembangan dari model McCall, Boehm, FURPS dimana dasar dari model ini adalah menentukan dan mengevaluasi kualitas produk perangkat lunak dalam segi internal maupun eksternal. Atribut-atribut kualitas *software* diklasifikasikan dalam bentuk hirarki karakteristik dan setiap karakteristik memiliki sub karakteristik. Sebagai suatu model standar kualitas *software*, ISO 9126 memiliki berbagai keunggulan dibandingkan dengan model yang lainnya. Tabel 1 berikut ini merupakan perbandingan berbagai model kualitas *software*.

Tabel 1. Perbandingan Model Kualitas *Software* dengan ISO 9126.

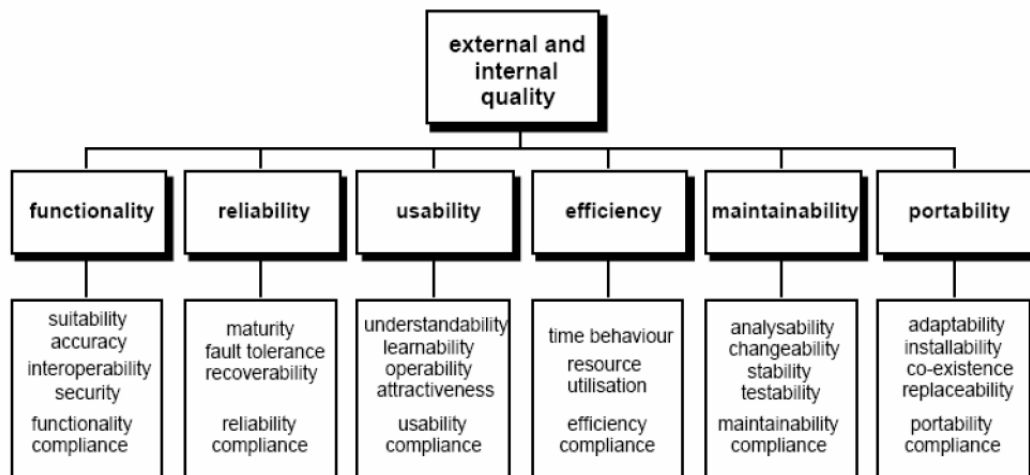
CHARACTERISTICS MODEL	MCCALL	BOEHM	FURPS	DROMEY	BBN	ISO 9126
STRUCTURE	Hierarchical	Hierarchical	Hierarchical	Hierarchical	Non-Hierarchical	Hierarchical
NUMBER OF LEVELS	2	3	2	2	n/a	3
RELATIONSHIP	Many-Many	Many-Many	One-Many	One-Many	Many-Many	One-Many
MAIN ADVANTAGE	Evaluation Criteria	Hardware Factors Included	Separation of FR & NFR	Different Systems	Weighted Factors	Evaluation Criteria
MAIN DISADVANTAGE	Components Overlapping	Lack of Criteria	Portability not Considered	Comprehensiveness	Lack of Criteria	Generality

Menurut Chua (2004), ISO 9126 memiliki kelebihan karena mudah beradaptasi dengan semua sistem karena ISO 9126 tidak mendefinisikan persyaratan kualitas yang spesifik melainkan hanya definisi umum untuk evaluasi perangkat lunak. Chua (2004) menggambarkan model dari ISO 9126 memiliki 6 karakteristik dalam gambar 1.



Gambar 1. Model ISO 9126 (Chua, 2004)

Cote (2006) menjelaskan bahwa ISO 9126 dibagi menjadi 6 karakteristik pada gambar 2.



Gambar 2. Model dan Karakteristik ISO 9126. (Cote, 2006)

Fahmy (2012) menjelaskan 6 karakteristik dari ISO 9126 yang terdiri dari *Functionality*, *Reliability*, *Usability*, *Efficiency*, *Maintainability* dan *Portability*. Chua (2004) menambahkan dari 6 karakteristik ISO tersebut kemudian dapat dijabarkan menjadi beberapa sub kategori yang dapat dijabarkan pada tabel 2. Setiap

kategori memiliki sub kategori yang masing-masing memiliki penjelasan masing-masing.

Tabel 2. Penjelasan Sub Kategori ISO 9126 (Chua, 2004).

Characteristic	Sub-characteristic	Explanation
Functionality	Suitability	Can software perform the tasks required?
	Accurateness	Is the result as expected?
	Interoperability	Can the system interact with another system?
	Security	Does the software prevent unauthorised access?
Reliability	Maturity	Have most of the faults in the software been eliminated over time?
	Fault tolerance	Is the software capable of handling errors?
	Recoverability	Can the software resume working and restore lost data after failure?
Usability	Understandability	Does the user comprehend how to use the system easily?
	Learnability	Can the user learn to use the system easily?
	Operability	Can the user use the system without much effort?
	Attractiveness	Does the interface look good?
Efficiency	Time Behaviour	How quickly does the system respond?
	Resource Utilisation	Does the system utilise resources efficiently?
Maintainability	Analysability	Can faults be easily diagnosed?
	Changeability	Can the software be easily modified?
	Stability	Can the software continue functioning if changes are made?
	Testability	Can the software be tested easily?
Portability	Adaptability	Can the software be moved to other environments?
	Installability	Can the software be installed easily?
	Conformance	Does the software comply with portability standards?
	Replaceability	Can the software easily replace other software?
All characteristics	Compliance	Does the software comply with laws or regulations?

a. Aspek *Functionality*

Functionality adalah kemampuan perangkat lunak untuk menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat dari pengguna di bawah kondisi tertentu penggunaan (Fahmy, 2012). Suatu *software* dikatakan memiliki tingkat fungsionalitas yang baik jika *software* tersebut dapat menjalankan fungsi – fungsi yang ada pada *software* tersebut (ISO, 2003). Dalam ISO (2003) dijelaskan bahwa *software* yang baik dapat diukur dengan menguji fungsi – fungsi yang ada pada *software* dalam bentuk *test case* pada *software developer*. Hasil dari pengujian tersebut menghasilkan nilai *functionality* dimana dikatakan memenuhi standar jika diatas 0,5 dan semakin mendekati 1 maka menunjukkan bahwa tingkat *functionality* dari *software* semakin baik.

Dalam ISO 9125, *functionality* memiliki sub karakteristik *security* yang menurut Chua (2004) adalah kemampuan untuk pembatasan akses pada

pengakses yang tidak disetujui. Menurut Huang (2003), untuk mengetahui keamanan suatu web dapat dideteksi dengan metode *SQL injection* dan *cross-site scripting* (XSS). Dua permasalahan keamanan tersebut dapat digunakan untuk menguji keamanan suatu web karena banyak web yang masih mengalami permasalahan tersebut dan permasalahan tersebut tergolong rumit ditangani (Huang, 2003).

SQL injection menurut Halfond (2006) adalah suatu kode yang berikan pada database yang menyebabkan terjadinya proses manipulasi data pada *database* tanpa mendapat melalui hak akses yang sah. Menurut Halfond (2006), kerentanan terhadap *SQL Injection* merupakan ancaman serius dalam web. *SQL Injection* dapat menyebabkan penyerang mendapatkan akses penuh terhadap data dalam *database*. Untuk mengatasi ini, pengembang sudah memberikan teknik penanganan terhadap *SQL Injection*. Pada kenyataannya, teknik penangan ini jarang digunakan sehingga masih memungkinkan terjadinya *SQL Injection* (Halfond, 2006).

Cross-site scripting berhubungan dengan aliran data pada web (Huang, 2003). Menurut Huang (2003), ketika suatu halaman web dibuka maka akan menghasilkan halaman respon yang dikirim pada *browser*. Halaman tersebut akan menghasilkan *cookies* yang dapat dibaca. Hal ini dapat menyebabkan web disusupi *script* berbahaya. Menurut Spett (2005), kerentanan terhadap *Cross-site scripting* ini biasanya terjadi pada pesan kesalahan dan form pengisian data. Suatu web dikatakan memenuhi aspek kewanaman jika dapat menangkis adanya *cross-site scripting*.

Menurut Doupe (2010), terdapat berbagai macam alat uji keamanan suatu website yang berupa *software* diantaranya yaitu Acunetix Web Vulnerability

Scanner, AppScan, NTOSpider dan Webinspect. Dari berbagai macam *software* uji kemanan website tersebut, Acunetix Web Vulnerability Scanner merupakan *software* terbaik yang mampu melakukan uji kemanan website karena kemampuan menguji setiap celah kemanan website yang berupa *SQL Injection*, *Cross-site scripting*, *code injection* dan *broken access control*. Suatu *software* yang diuji dengan Acunetix Web Vulnerability Scanner dikatakan memenuhi aspek security jika memiliki tingkat kerentanan (*vulnerability level*) dibawah *Level 2 Medium*.

Jadi dalam aspek *functionality* terdapat dua macam uji yaitu menguji fungsi yang ada pada *software* dalam bentuk *test case* pada *software developer* dan menguji keamanan website dengan menggunakan *software* Acunetix Web Vulnerability Scanner. Hasil dari uji fungsi *software* tersebut dikatakan memenuhi standar jika menghasilkan nilai *functionality* lebih dari 0,5 dan hasil dari uji kemanan website dikatakan memenuhi standar jika menghasilkan tingkat kerentanan dibawah *Level 2 Medium*.

b. Aspek *Reliability*

Menurut Fahmy (2012), *Reliability* adalah kemampuan dari produk perangkat lunak untuk mempertahankan tingkat tertentu dari kinerja bila digunakan dalam kondisi tertentu. *Reliability* menurut Chua (2004) memiliki sub karakteristik yaitu *maturity*, *fault tolerance* dan *recoverability*. Dari beberapa sub karakteristik *reliability* tersebut, sub karakteristik *maturity* yang paling berpengaruh terhadap *software* berbasis web (Olsina, 2006).

Maturity merupakan kemampuan *software* untuk berjalan secara stabil jika digunakan dalam jangka waktu tertentu (Chua, 2004). Dalam ISO (2003) disebutkan untuk mengetahui tingkat *maturity* suatu *software* dapat menggunakan *stress testing*. *Stress testing* bekerja dengan menguji skenario (*test case*)

berdasarkan *user* yang mengakses bersamaan dan dalam waktu tertentu. Hasil dari pengujian tersebut kemudian dibandingkan dengan standar Telcordia dalam Asthana (2009) yang menyatakan bahwa semua *test case* (100%) dalam *software* harus diuji dan 95% *test case* harus berhasil.

Untuk mengetahui persentase tingkat *reliability* tersebut dapat digunakan *software Web Application Load, Stress and Performance Testing* (WAPT). WAPT merupakan *software* yang mampu membebani sistem dengan kondisi tertentu untuk menguji apakah *software* tersebut masih dapat berjalan dengan baik ketika terdapat beban pada sistem. Menurut Rina (2013), WAPT merupakan *software* untuk mengetahui tingkat kesetabilan *software* pada tingkat tertentu dengan menjalankan *load agent* secara bersamaan. *Load agent* merupakan suatu *virtual user* yang ukurannya telah ditetapkan secara standar dalam WAPT.

Jadi suatu *software* telah memenuhi aspek *reliability* jika *software* tersebut memiliki persentase *reliability* diatas 95%. Persentase tersebut didapatkan dari hasil uji *reliability software* dengan menggunakan *software Application Load, Stress and Performance Testing* (WAPT). WAPT tersebut bekerja dengan menjalankan *virtual user* sehingga memberikan beban terhadap jalannya fungsi yang ada dalam *software*.

c. Aspek *Usability*

Menurut Fahmy (2012), *usability* adalah kemampuan *software* untuk mudah dipelajari, digunakan dan menarik berdasarkan kondisi tertentu. *Usability* berhubungan dengan interaksi manusia dan komputer dengan memiliki konsep "*quality in use*" (kualitas penggunaan) yang berfokus pada tampilan (Seffah, 2006). *Usability* berkaitan dengan pengalaman pengguna *software* yang menurut Seffah (2006) berelasi dengan kemudahan pengguna dalam menggunakan *software*

maupun kemampuan *software* dalam memberikan pemahaman kepada pengguna.

Menurut Joo (2011) tingkat *usability* dari suatu *software* berbasis website dapat diketahui dengan menggunakan kuesioner atau angket yang telah di validasi. Terdapat berbagai macam kuesioner yang dikemukakan oleh berbagai ahli salah satunya adalah *Usefulness, Satisfaction, and Ease of use* (USE) yang dikemukakan oleh Arnold M. Lund (2001). USE merupakan kuesioner yang terdiri dari 3 dimensi yaitu *usefulness, satisfaction* dan *ease of use* yang terdiri dari 30 pernyataan dan dapat digunakan untuk menguji *usability* pada *software, hardware* maupun *service* (Lund, 2001). Menurut Lund (2001), USE menggunakan skala Likert yang terdiri dari 7 poin dimana semakin tinggi poin maka menyatakan sangat setuju. Untuk mengetahui *reliabilitas* kuesioner USE, maka hasil dari kuesioner dicari nilai *alpha cronbach* dan kemudian dibandingkan dengan tabel 3 yang merupakan standar *internal consistency* (George, 2003).

Tabel 3. *Internal Consistency* (George, 2003)

Cronbach's Alpha	Internal Consistency
$\alpha \geq .9$	<i>Excellent</i>
$.9 > \alpha \geq .8$	<i>Good</i>
$.8 > \alpha \geq .7$	<i>Acceptable</i>
$.7 > \alpha \geq .6$	<i>Questionable</i>
$.6 > \alpha \geq .5$	<i>Poor</i>
$.5 > \alpha$	<i>Unacceptable</i>

Sedangkan untuk mengetahui apakah *software* yang dikembangkan memenuhi aspek *usability* maka dapat diketahui dengan menghitung persentase jawaban dari responden pada jawaban yang menyatakan setuju maupun tidak

setuju (Sugiyono, 2012). Untuk menghitung persentase jawaban dari responden dapat digunakan persamaan berikut ini,

$$\text{Persetujuan terhadap sistem} = \frac{\text{Jumlah skor yang diperoleh}}{\text{Jumlah skor maksimal}} \times 100\%$$

Hasil dari persamaan tersebut akan menghasilkan persentase yang menunjukkan tingkat kemudahan penggunaan software (*usability*).

d. Aspek *Efficiency*

Efficiency adalah kemampuan produk *software* untuk memberikan kinerja yang tepat, relatif terhadap jumlah sumber daya yang digunakan, dinyatakan dalam kondisi (Fahmy, 2012). Chua (2004) menjelaskan bahwa dalam aspek *efficiency* terdapat sub karakteristik *time behaviour* dan *resource utilisation*. *Time behaviour* dapat juga disebut *performance* yang menurut Chua (2004) adalah kemampuan *software* untuk dapat merespon dengan cepat.

Kecepatan *software* untuk merespon merupakan faktor penting dalam menjaga pengguna untuk menggunakan *software* tersebut (Subraya, 2006). Menurut Subraya (2006) dalam *software* berbasis web, kecepatan akses web yang lambat akan mempengaruhi psikologi pengunjung untuk berhenti mengakses web, tidak pernah mengunjungi web tersebut lagi dan mempengaruhi orang lain untuk tidak mengunjungi.

Berdasarkan pada penelitian eMarketer dan dijelaskan Subraya (2006) diketahui presentase minat seseorang terhadap web berdasarkan dengan kecepatan akses web tersebut pada tabel 4.

Tabel 4. Presentase Minat *User* Berdasarkan *Load Time* (Subraya, 2006)

Load Time	Percentage of Users Waiting
10 seconds	84%
15 seconds	51%
20 seconds	26%
30 seconds	5%

Dari tabel 3 tersebut dapat diketahui bahwa rata-rata pengunjung suatu *website* akan menunggu loading web hingga 15 detik. *Website* dengan waktu *load* lebih dari 20 detik kemungkinan hanya akan ditunggu *user* kurang dari 30%. Selain persentase waktu tunggu *user*, juga perlu dipertimbangkan dalam kecepatan web adalah rata-rata *load time website* di dunia. Berdasarkan data yang dirilis majalah Fortune tahun 2010 (Tabel 5), *website* lokal memiliki kecepatan akses rata-rata adalah 7 detik sedangkan untuk *website* internasional memiliki kecepatan akses rata-rata 9.5 detik.

Tabel 5. Rata-Rata Kecepatan Akses *Website* (Fortune, 2010)

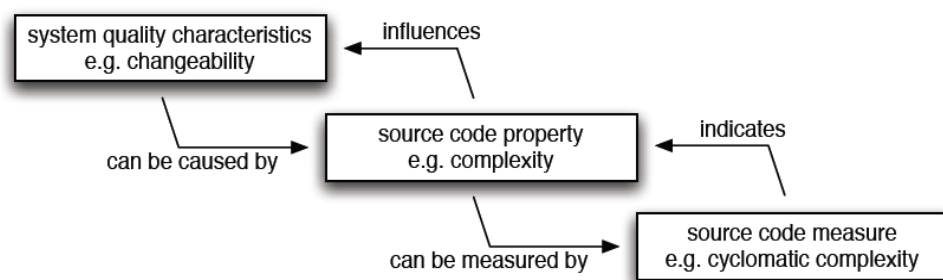
#	Benchmark	Load Time
1	First view load time (Domestic)	7.066 seconds
2	First view load time (International)	9.462 seconds
2	Repeat view load time	3.970 seconds
3	Start render time	2.960 seconds

Dari tabel tabel 5 tersebut maka suatu *web application* dikatakan memenuhi aspek efisiensi harus memiliki *load time* kurang dari 7 detik. Sedangkan data pada tabel 4 Kecepatan *load time* lebih dari 10 detik akan menyebabkan berkurangnya minat *user* untuk menggunakan *web* tersebut. Untuk dapat mengetahui besarnya *load time* pada suatu web dapat digunakan *software* berbasis web yaitu GTMetrix. GTMetrix merupakan *software* berbasis yang mampu melakukan pengujian

terhadap *software* berbasis web dengan hasil berupa peringkat Google Page Speed serta Yahoo YSlow dan *load time* dalam ukuran detik.

e. Aspek *Maintainability*

Maintainability menurut Fahmy (2012) adalah kemampuan *software* untuk dapat dengan mudah dipahami, dikembangkan, dan dimodifikasi. Modifikasi meliputi *correction, improvements or adaptation to changes in the environment, in requirements* dan *functional specifications*. Dalam pengukuran *maintainability*, Heitlager (2007) mendeskripsikan dalam beberapa langkah yang digambarkan dalam model pada gambar 3. Langkah pengukuran menurut Heitlager (2007) yaitu melihat karakteristik level sistem pada *sifat source* kode kemudian dilanjutkan dengan mengukur properti pada *source code*.



Gambar 3. Model Maintainability (Heitlager, 2007)

Heitlager (2007) juga membuat *mapping* dari karakteristik sistem ke dalam *source code* yang terdiri baris dan kolom. Setiap kolom terdiri dari level kode seperti *volume, complexity, duplication, unit length, number of units, and number of modules*. Gambar 4 merupakan *mapping* karakteristik *maintainability* yang berpengaruh terhadap tingkat *maintainability*.

		source code properties				
ISO 9126 maintainability		volume	complexity per unit	duplication	unit size	unit testing
	analysability	x		x	x	x
	changeability		x	x		
	stability					x
	testability		x		x	x

Gambar 4. *Mapping Karakteristik Maintainability*. (Heitlager, 2007)

Untuk mengetahui tingkat *maintainability* dari suatu *software*, dapat dilakukan dengan menghitung *Maintainability Index* (Ganpati, 2012). Ganpati (2012) menjelaskan semakin tinggi nilai *Maintainability Index* (MI) hingga mendekati 100 menunjukkan bahwa *software* mudah dilakukan perawatan sedangkan nilai MI sama dengan 0 menunjukkan *software* sulit untuk dirawat. Menurut Ganpati (2012), suatu *software* dapat dibedakan menjadi tiga macam berdasarkan nilai *maintainability index*nya sesuai dengan tabel 6.

Tabel 6. *Maintainability Index* (Ganpati, 2012)

Maintability Index	Sifat	Keterangan
86 - 100	<i>Highly Maintainable</i>	Sangat mudah dirawat
66 - 85	<i>Moderate Maintainable</i>	Normal untuk dirawat
0 - 65	<i>Difficult to Maintain</i>	Sulit untuk dirawat

Tabel 6 tersebut menjelaskan tiga macam tingkat MI yaitu sulit untuk dirawat, layak atau normal untuk dirawat dan sangat mudah untuk dirawat. Suatu *software* dikatakan telah memenuhi aspek *maintainability* jika nilai MI berada dikisaran layak atau normal untuk dirawat dan sangat mudah untuk dirawat atau dengan nilai MI sama dengan atau diatas 66. Untuk menghitung nilai MI dapat dilakukan dengan

bantuan *software* Source Code SearchEngine. Soure Code SearchEngine merupakan *software* yang mampu menguji tingkat *maintainability* dari suatu *software* yang dikembangkan dengan berbagai macam bahasa pemrograman seperti PHP, HTML, Java dan C++.

f. Aspek Portability

Menurut Fahmy (2012), portabilitas (*portability*) adalah kemampuan suatu *software* untuk dapat berjalan di lingkungan yang berbeda. *Portability* dalam ISO 9126 memiliki 4 sub-karakteristik yaitu *adaptability*, *install-ability*, *co-existence*, *replace-ability* dan *portability compliance*. Suatu *software* dapat dikatakan *portable* jika *software* tersebut dapat berjalan dengan normal pada banyak tipe komputer yang berbeda tanpa adanya perubahan pada kode program tersebut (Brown, 2003).

Menurut Salonen (2012), pada awalnya *portability* hanya berfokus pada *software* yang digunakan pada tipe *hardware* yang berbeda. Pengertian *portability* berkembang mengarah ke *software* berbasis web yang dapat berjalan di berbagai *web browser*, *server* maupun kecepatan akses internet. Menurut Salonen (2012), *portability* merupakan salah satu faktor utama yang dimiliki *software*. *Software* yang memiliki kemampuan dapat berjalan di sistem lain tanpa adanya perubahan kode, dapat menghemat biaya pembuatan *software*.

Salonen (2012) menambahkan bahwa suatu *software* berbasis web telah memenuhi aspek *portability* jika *software* berbasis web tersebut dapat dijalankan paling sedikit di 7 *web browser desktop* dan 5 *web browser mobile*. Menurut Larsen (2013:230), untuk mengetahui apakah suatu *software* berbasis web dapat berjalan di *web browser* lain maka dapat digunakan *software* BrowseEmAll. BrowseEmAll

adalah *software* yang mampu menjalankan *virtual browser* dari browser yang paling banyak digunakan.

B. Penelitian yang Relevan

Berikut ini merupakan beberapa penelitian relevan yang dilakukan oleh peneliti lain :

1. Sistem Penyusunan Jadwal Pelajaran Berbasis Web merupakan penelitian yang dilakukan oleh Fajar Kurniawan (2012) yang membuat sistem informasi pembuat jadwal secara otomatis dengan menggunakan algoritma *blind search*. Sistem informasi yang dikembangkan berbasis web dengan menggunakan bahasa pemrograman PHP dan MySQL. Perbedaan dengan penelitian yang dilakukan adalah pada fitur dari *software* yaitu pencarian jadwal dan manajemen jadwal yang dapat langsung dilakukan oleh pemilik jadwal dan tidak terdapatnya fitur pembuatan jadwal secara otomatis.
2. Rancang Bangun Sistem Penjadwalan Perkuliahan dan Ujian Akhir Semester dengan Pendekatan Algoritma Genetika merupakan dari Sam'ani (2012) yang membuat sistem penjadwalan otomatis dengan algoritma genetika dan hasilnya dalam bentuk file Microsoft Excel. Perbedaan dengan penelitian yang dilakukan adalah informasi jadwal dapat dimasukan oleh pemilik jadwal tersebut sendiri dan tidak di buat secara otomatis oleh admin.
3. Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika oleh Andhika Lady Maharsi (2013) yang membuat *software* penjadwalan secara otomatis berbasis java dengan menggunakan algoritma genetika. Perbedaan dari penelitian ini adalah sistem yang dikembangkan hanya memberikan informasi jadwal sehingga mempermudah pencarian,

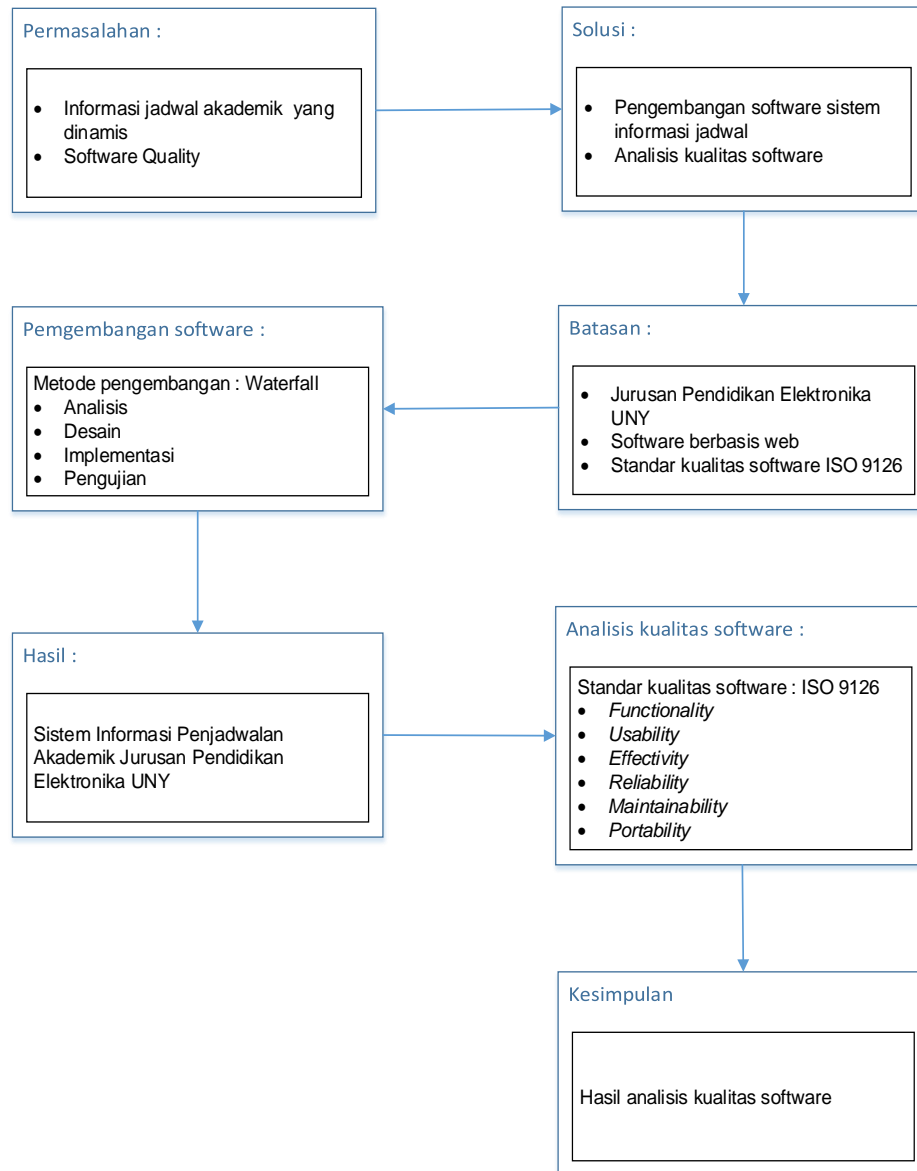
pengubahan maupun penambahan dan bukan untuk membuat jadwal secara otomatis.

C. Kerangka Pikir

Jurusan Pendidikan Teknik Elektronika Universitas Negeri Yogyakarta (UNY) menggunakan jadwal dalam mengatur setiap kegiatan seperti pelaksanaan kuliah, pelaksanaan rapat maupun event kegiatan tertentu. Akan tetapi jadwal tersebut masih berbentuk statis sehingga jika terdapat perubahan maupun penambahan jadwal, informasi tersebut tidak dengan mudah diterima oleh mahasiswa maupun dosen.

Dengan adanya sistem informasi jadwal ini diharapkan informasi jadwal lebih bersifat dinamis. *Software* sistem informasi jadwal ini merupakan *software* berbasis web yang menggunakan PHP (Yii Framework), MySQL dan HTML 5. Sistem informasi yang baik sebaiknya memenuhi standar kualitas *software* maka dari itu sistem informasi ini dianalisis dan dikembangkan sesuai standar kualitas *software* ISO 9126 yang memiliki karakteristik *functionality, usability, effectivity, reliability, maintainability dan portability*.

Diagram kerangka pikir digambarkan dalam gambar 5. Gambar 5 menggambarkan hubungan variabel yang diteliti dalam penelitian ini. Terdapat 6 macam variabel yang berdasarkan pada karakteristik dalam ISO 9126 yaitu *functionality, reliability, usability, efficiency, maintainability dan portability*. Variabel - variabel tersebut merupakan karakteristik dalam mengembangkan *software* dengan kualitas baik sesuai standar ISO 9126.



Gambar 5. Kerangka Pikir

Setiap karakteristik dari ISO 9126 memiliki bentuk pengujian yang berbeda-beda. Hasil dari pengujian tersebut kemudian dibandingkan dengan standar nilai pada setiap karakteristik sehingga dapat diketahui apakah *software* tersebut memenuhi standar kualitas *software* sesuai dengan ISO 9126.

D. Pertanyaan Penelitian

Berdasarkan kerangka pikir yang telah dijelaskan maka dapat dirumuskan pertanyaan penelitian sebagai berikut :

1. Bagaimanakah cara mengembangkan sistem informasi jadwal akademik berbasis *Yii Framework*?
2. Apakah sistem informasi jadwal akademik memenuhi standar kualitas *software* berdasarkan faktor *functionability*?
3. Apakah sistem informasi jadwal akademik memenuhi standar kualitas *software* berdasarkan faktor *reliability*?
4. Apakah sistem informasi jadwal akademik memenuhi standar kualitas *software* berdasarkan faktor *efficiency*?
5. Apakah sistem informasi jadwal akademik memenuhi standar kualitas *software* berdasarkan faktor *maintability*?
6. Apakah sistem informasi jadwal akademik memenuhi standar kualitas *software* berdasarkan faktor *usability*?
7. Apakah sistem informasi jadwal akademik memenuhi standar kualitas *software* berdasarkan faktor *portability*?

BAB III

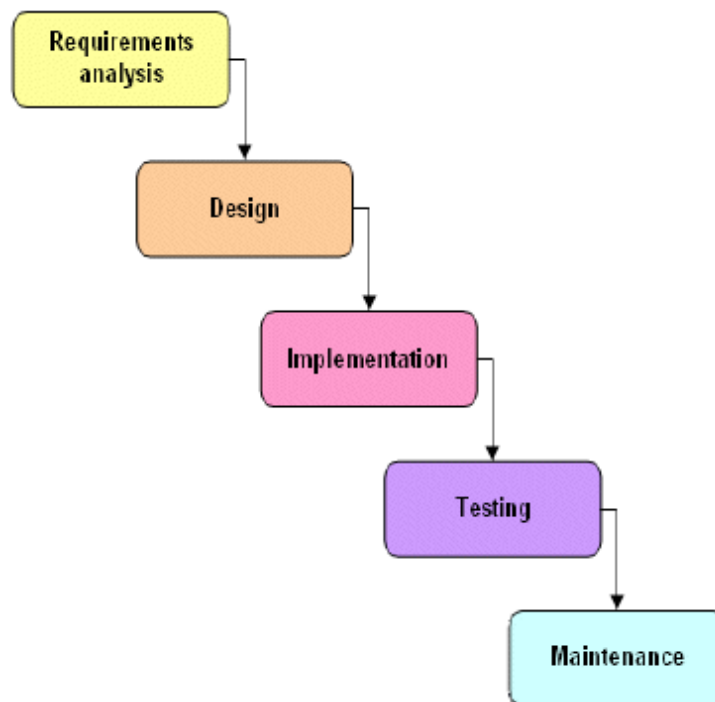
METODE PENELITIAN

A. Model Pengembangan

Metode penelitian yang digunakan dalam penelitian ini adalah penelitian dan pengembangan atau *Research and Development* (R&D). Menurut Sugiyono (2012), metode penelitian dan pengembangan adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut. Metode tersebut telah banyak digunakan di hampir semua produk teknologi.

Senada dengan metode penelitian dan pengembangan, dalam rekayasa perangkat lunak (*software engineering*) terdapat model pengembangan software yang terdiri dari berbagai aktifitas yaitu analisis kebutuhan, spesifikasi, desain arsitektur program, implementasi, pengujian, dokumentasi, pelatihan dan perawatan (Maheshwari, 2012). Proses dan kegiatan rekayasa perangkat lunak tersebut disusun dalam tahapan-tahapan yang disebut dengan *software development life cycle* (SDLC) atau daur hidup pengembangan *software*. Terdapat berbagai macam model dari SDLC diantaranya yaitu *waterfall*, *iterative*, *prototype* dan *spiral* (Maheshwari, 2012).

Dalam mengembangkan sistem informasi jadwal akademik ini menggunakan model *waterfall* (air terjun). Menurut Maheshwari (2012), model *waterfall* adalah model SDLC konvensional, linier dan berurutan yang dimulai dari analisis kebutuhan, desain, implementasi, pengujian dan perawatan. Maheshwari (2012) menggambarkan model dari *waterfall* seperti terdapat pada gambar 6.



Gambar 6. Model *Waterfall* (Maheshwari, 2012)

Sesuai dengan model *waterfall* tersebut, pengembangan sistem informasi jadwal akademik juga dimulai dari analisis kebutuhan. Pada tahap analisis kebutuhan, dilakukan observasi dan analisis terhadap permasalahan yang jadwal. Pada tahap berikutnya yaitu tahap desain, dirancang desain dari *software* yang akan dikembangkan dengan menggunakan bahasa pemodelan UML dan ERD. Tahap implementasi dilakukan dengan menuliskan kode program sistem informasi jadwal akademik sesuai dengan desain yang dibuat. Pada tahap berikutnya, *software* yang telah jadi kemudian diuji sesuai dengan standar kualitas perangkat lunak. Tahap terakhir yaitu perawatan dilakukan dengan memberikan update *software* untuk meningkatkan kualitas *software*. Dalam penelitian ini, tahap terakhir yaitu perawatan atau *maintenance* tidak dilakukan karena tahap perawatan dilakukan ketika suatu *software* telah digunakan secara rutin. Sehingga model *waterfall* dalam penelitian ini hanya sampai tahap pengujian.

B. Prosedur Pengembangan

1. Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahapan pertama dalam pengembangan perangkat lunak yang digunakan untuk mendeskripsikan data – data yang digunakan dalam perangkat lunak yang akan dibuat. Dalam tahap ini, dilakukan beberapa langkah seperti observasi maupun wawancara untuk mendapatkan merancang bagaimana cara sistem tersebut dikembangkan, siapa pengguna sistem tersebut dan informasi apa saja yang dibutuhkan untuk mengembangkan sistem tersebut.

2. Desain

Desain merupakan tahap kedua dari proses pengembangan sistem. Hasil dari tahap analisis kebutuhan, akan diubah menjadi model dari sistem informasi yang akan dikembangkan. Pada tahap desain, digunakan pemodelan agar proses perancangan sistem dapat terencana dengan baik. Bahasa pemodelan yang digunakan adalah *Unified Modelling Language (UML)*. *Unified Modelling Language (UML)* merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasi, merancang serta mendokumentasi perangkat lunak.

3. Implementasi

Tahap implementasi adalah tahap penerapan desain sistem yang telah dibuat menjadi kode program sehingga menghasilkan perangkat lunak yang dapat berfungsi sesuai dengan yang diinginkan. Untuk mempermudah dan menjaga agar *software* yang dikembangkan terjamin kualitasnya maka digunakan *framework Yii*. *Framework Yii* akan membantu agar *software* berjalan dengan cepat dan aman.

Sedangkan pada sisi tampilan, digunakan *Twitter Bootstrap* yang merupakan *framework Cascading Style Sheets (CSS)* untuk mempermudah

desain tampilan. Bootstrap telah menyediakan berbagai *library* yang dapat digunakan untuk mendukung proses pembuatan tampilan. Dengan menggunakan *framework* Yii dan *Twitter Bootstrap* maka tahap implementasi dapat berjalan dengan lebih cepat.

4. Pengujian

Tahap pengujian dilakukan untuk menguji sistem apakah sistem dapat digunakan dan berfungsi dengan baik. Dengan adanya pengujian maka dapat diketahui apakah terdapat permasalahan (*error*) atau fungsi *software* yang tidak memenuhi standar kualitas perangkat lunak.

C. Sumber Data / Subyek Penelitian

Subyek penelitian untuk pengujian kualitas *software* karakteristik *reliability*, *efficiency*, *maintainability* dan *portability* adalah sistem informasi jadwal akademik yang dikembangkan. Untuk aspek *functionality* terdapat dua subyek pengujian yaitu sistem informasi jadwal akademik dan *software developer* sebanyak 3 orang. Sedangkan untuk aspek *usability*, subyek penelitian adalah mahasiswa dan dosen Jurusan Pendidikan Teknik Elektronika UNY.

Subyek penelitian yang diambil dari penelitian ini adalah dengan menggunakan sampel. Berdasarkan Roscoe dalam Sugiono (2012), jumlah sampel yang layak dalam penelitian adalah 30 hingga 500. Selain itu, menurut Nielsen (2012), untuk mendapatkan data yang signifikan secara statistik, maka jumlah sampel paling sedikit adalah 20 orang. Sehingga dalam penelitian ini, jumlah sampel yang digunakan adalah sebanyak 30 orang dimana terdiri dari dosen sebanyak 3 orang dan mahasiswa sebanyak 27 orang.

D. Metode dan Alat Pengumpul Data

1. Observasi

Observasi dilakukan untuk mengetahui kualitas suatu *software* pada aspek *functionality*, *reliability*, *efficiency* dan *maintability*. Observasi dilakukan pada hasil pengembangan sistem informasi dimana data informasi yang dicari merupakan data-data untuk mendukung terhadap aspek kualitas *software* seperti jumlah kode (KLOC), jumlah modul dan sebagainya.

2. Angket

Dalam ISO 9126 terdapat karakteristik *usability* dimana cara pengujiannya yaitu dengan menggunakan angket atau kuesioner. Sehingga teknik pengumpulan data yang digunakan adalah menggunakan angket.

E. Teknik Analisis Data

Untuk menguji kualitas dari *software* yang dibuat maka digunakan standar ISO 9126. Teknik analisis kualitas perangkat lunak berdasarkan ISO 9126 yang akan digunakan meliputi *functionality*, *reliability*, *efficiency* dan *maintability*.

1. *Functionality*

Untuk menguji fungsionalitas suatu *software* berbasis web, dapat dilakukan dengan menguji fungsi-fungsi yang ada pada website oleh yang ahli di bidangnya dalam bentuk *test case*. Menurut ISO 9126 (2003), suatu *software* dapat diketahui tingkat fungsionalitasnya dengan persamaan berikut ini.

$$X = 1 - \frac{A}{B}$$

Keterangan :

X = Tingkat *functionality*

A = Jumlah fungsi yang bermasalah

B = Jumlah keseluruhan fungsi yang diuji.

Hasil dari pengujian fungsionalitas tersebut menghasilkan nilai X dimana X lebih besar atau sama dengan 1 dan X kurang dari atau sama dengan 1. Software dikatakan telah memenuhi standar *functionality* jika nilai X lebih dari 0,5 dan nilai X semakin mendekati 1 maka *functionality* dari *software* akan semakin baik (ISO, 2003).

Selain itu, dalam ISO 9126 juga terdapat sub karakteristik *security*. *Security* merupakan aspek yang sangat penting dalam menjaga keamanan data dalam *software*. Untuk memenuhi aspek keamanan dalam fungsionalitas web, maka web harus dapat mengatasi serangan *SQL Injection* (Dougherty, 2012). Untuk menguji *SQL Injection*, menurut Huang (2003) dapat dilakukan dengan cara menuliskan *injection* teks pada input *username* dan *password*. Masukan tersebut pada SQL akan menghasilkan query berikut.

“*SELECT * FROM Users WHERE (strUserName='X' OR 'A' = 'A')
AND (Password='X' OR 'A' = 'A');*”

Hasil dari *injection* tersebut akan membuat *username* dan *password* bernilai *true* walaupun tidak sesuai dengan database (Huang, 2003). Untuk menguji fungsionalitas sistem pada aspek keamanan (*security*), maka digunakan *software* Acunetix Web Vulnerability Scanner. *Software* tersebut bekerja dengan cara memindai seluruh direktori dan link yang ada pada sistem dan kemudian menginjeksi setiap *link* dengan *script SQL Injection* dan *cross site-scripting*. Prinsip kerja dari *SQL Injection* adalah dengan mengirimkan *query SQL* ke sistem seperti pada tabel 7.

Tabel 7. *SQL Injection String*

No	SQL Injection String
1	admin' --
2	admin' #
3	admin'/*
4	' or 1=1--

Acunetix Web Vulnerability Scanner akan melakukan pemindaian secara otomatis terhadap kerentanan tersebut. Hasil dari pemindaian tersebut dapat ditampilkan dalam bentuk *threat level* yang menunjukkan kerentanan keamanan (*Vulnerability*) dalam sistem. Terdapat berbagai macam *threat level* dalam Acunetix Web Vulnerability Scanner yaitu *Level 1 Low* yang berarti rendah kerentanan, *Level 2 Medium* yang berarti memiliki kerentanan sedang dan *Level 3 High* yang berarti tinggi kerentanan. *Software* dikatakan aman dari kerentanan jika memiliki *threat level* dibawah *Level 2 Medium*.

2. Reliability

Reliability dalam ISO 9126 memiliki komponen *maturity* yang menunjukkan ketahanan *software* dalam waktu tertentu. Dalam ISO Metrics (2003) dijelaskan bahwa tingkat *maturity* suatu *software* dapat diuji dengan menggunakan *stress testing*. Untuk mengetahui nilai *maturity* dari *software* dapat dihitung dengan persamaan berikut ini (ISO, 2003).

$$X = \frac{A}{B}$$

Keterangan :

X = Nilai *maturity*.

A = Jumlah lolos dalam pengujian

B = Jumlah pengujian keseluruhan.

Nilai *maturity* tersebut berkisar antara 0 hingga 1 dimana semakin mendekati 1 maka tingkat *maturity* akan semakin baik. Nilai tersebut kemudian diubah dalam bentuk persentase yang kemudian dibandingkan dengan standar Telcordia.

Untuk menguji Stress testing dari suatu *software* berbasis web dapat dilakukan dengan menggunakan *software Web Application Load, Stress and Performance Testing (WAPT)*. WAPT merupakan *software* yang dapat menguji

kehandalan dari suatu website dengan menjalankan *virtual user agent* sebanyak kriteria yang ditentukan baik secara bersamaan maupun bertahap dalam jangka waktu tertentu.

3. Usability

Untuk mengetahui suatu sistem komputer telah memenuhi aspek *usability* dapat dilakukan dengan menggunakan kuesioner *Usefulness, Satisfaction, and Ease of use (USE)* yang dikemukakan oleh Lund (2001). Kuesioner USE memiliki 30 butir pertanyaan dengan skala Linkert 1, 2, 3, 4, 5, 6, 7. Tabel 8 berikut ini merupakan kuesioner *Usefulness, Satisfaction, and Ease of use (USE)*

Tabel 8. Angket *Usefulness, Satisfaction, and Ease of use (USE)*

No	Pernyataan	Jawaban						
Usefulness								
1	Sistem ini membantu saya menjadi lebih efektif	1	2	3	4	5	6	7
2	Sistem ini membantu saya menjadi lebih produktif	1	2	3	4	5	6	7
3	Sistem ini bermanfaat	1	2	3	4	5	6	7
4	Sistem ini memberikan kontrol lebih besar terhadap aktivitas saya	1	2	3	4	5	6	7
5	Sistem ini membuat hal-hal yang ingin saya lakukan menjadi lebih mudah	1	2	3	4	5	6	7
6	Sistem ini menghemat waktu saya ketika saya gunakan	1	2	3	4	5	6	7
7	Sistem ini memenuhi apa yang saya butuhkan	1	2	3	4	5	6	7
8	Sistem ini melakukan segala sesuatu yang saya harapkan untuk dilakukan	1	2	3	4	5	6	7
Ease of Use								
9	Sistem ini mudah digunakan	1	2	3	4	5	6	7
10	Sistem ini praktis untuk digunakan	1	2	3	4	5	6	7
11	Sistem ini <i>user friendly</i> (mudah dipahami pengguna)	1	2	3	4	5	6	7
12	Sistem ini hanya membutuhkan sedikit langkah-langkah untuk mencapai apa yang saya ingin lakukan	1	2	3	4	5	6	7
13	Sistem ini dapat disesuaikan dengan kebutuhan (fleksibel)	1	2	3	4	5	6	7

14	Saya menggunakan sistem ini tanpa kesulitan	1	2	3	4	5	6	7
15	Saya dapat menggunakan sistem ini tanpa panduan tertulis	1	2	3	4	5	6	7
16	Saya tidak melihat adanya ketidakkonsistenan (inkonsistensi) ketika saya menggunakan sistem ini	1	2	3	4	5	6	7
17	Pengguna yang jarang maupun rutin menggunakan akan menyukai sistem ini	1	2	3	4	5	6	7
18	Saya dapat mengetahui adanya kesalahan dan memperbaikinya dengan cepat dan mudah	1	2	3	4	5	6	7
19	Saya dapat menggunakan sistem ini dengan sukses setiap kali saya menggunakannya	1	2	3	4	5	6	7
Ease of Learning								
20	Saya belajar menggunakan sistem ini dengan cepat	1	2	3	4	5	6	7
21	Saya mudah mengingat bagaimana cara menggunakan sistem ini	1	2	3	4	5	6	7
22	Sistem ini mudah untuk dipelajari penggunaannya	1	2	3	4	5	6	7
23	Saya dengan cepat menjadi mahir menggunakan sistem ini	1	2	3	4	5	6	7
Satisfaction								
24	Saya puas menggunakan sistem ini	1	2	3	4	5	6	7
25	Saya akan merekomendasikan sistem ini kepada teman saya	1	2	3	4	5	6	7
26	Sistem ini menyenangkan untuk digunakan	1	2	3	4	5	6	7
27	Sistem ini bekerja seperti apa yang saya inginkan	1	2	3	4	5	6	7
28	Sistem ini sangat bagus	1	2	3	4	5	6	7
29	Saya merasa memerlukan sistem ini	1	2	3	4	5	6	7
30	Sistem ini nyaman untuk digunakan	1	2	3	4	5	6	7

Skala Likert pada kuesioner USE terdiri dari 7 poin skala yang semakin tinggi poinnya menunjukkan semakin setuju sedangkan semakin rendah poin yang dipilih menunjukkan semakin tidak setuju. Skala tersebut dapat diubah menjadi kata-kata seperti tabel 9 berikut ini (Vagias, 2006).

Tabel 9. Keterangan Poin Skala Likert

Poin	Keterangan
1	Sangat tidak setuju
2	Tidak setuju
3	Sedikit tidak setuju
4	Ragu – ragu
5	Sedikit setuju
6	Setuju
7	Sangat setuju

Untuk mengetahui apakah software telah memenuhi aspek usability yaitu dengan menghitung jumlah dari perkalian poin Likert dengan jumlah penjawab.

Sedangkan untuk mengetahui reliabilitas instrumen *usability*, maka dapat dilakukan dengan menghitung nilai *koefisien alpha cronbach* dari kuesioner USE yang sudah dikemukakan oleh Lund (2001). Hasil dari jawaban angket tersebut kemudian dihitung dengan rumus *internal consistency alpha cronbach* berikut ini.

$$\alpha = \frac{K}{K - 1} \left(1 - \frac{\sum_{i=1}^K \sigma_{Yi}^2}{\sigma_X^2} \right)$$

Untuk mempermudah perhitungan *apha cronbach* maka dapat digunakan *software* IBM SPSS Statistics versi 20. Dari uji reliabilitas dengan SPSS tersebut kemudian nilainya dibandingkan dengan tabel *internal consistency* sehingga dapat diketahui hasilnya baik atau tidak.

4. Efficiency

Fahmy (2012) menjelaskan bahwa *efficiency* dalam ISO 9126 memiliki komponen *time behaviour (performance)*. Menurut Mulugenta (2008), *performance* suatu web adalah seberapa cepat beberapa aspek dalam web untuk melakukan beban kerja tertentu. Dalam web, *performance* mengarah kepada kecepatan *website* untuk diakses dan menampilkan kontennya pada *web browser*.

Untuk menguji suatu kecepatan web, dapat dilakukan dengan menggunakan *web application* GTMetrix yang dapat diakses secara online di <http://gtmetrix.com/>.

GTMetrix merupakan aplikasi berbasis web yang memiliki kemampuan untuk menganalisa performa dari suatu website sehingga menghasilkan kecepatan akses (*page load time*), total ukuran website (*total page size*) dan jumlah *request* (*number of request*). Selain itu, GTMetrix juga memberikan peringkat (*grade*) berdasarkan Google (*Page Speed*) dan Yahoo (*YSlow*). Prinsip kerja dari GTMetrix adalah menghitung waktu yang dibutuhkan setiap *file* dalam web untuk tampil secara penuh dalam *web browser*.

5. *Maintainability*

Untuk memenuhi standar sebagai suatu *software* berdasarkan ISO 9126, maka harus dipenuhi aspek *maintability*. Menurut Ganpati (2012), *maintability* dapat diukur untuk mengetahui kemudahan perawatan *software* dengan *Maintability Index (MI)*. *Maintability Index (MI)* dapat dihitung dengan persamaan berikut ini (Ganpati, 2012).

$$MI = 171 - 5.2 * \ln(aveV) - 0.23 * aveV(g) - 16.2 * \ln(aveLOC)$$

Keterangan :

MI = *Maintability Index*

aveV = Rata-rata *Hasteed Volume*

aveV(g) = Rata-rata *Cyclomatic Complexity* setiap modul

aveLOC = Rata-rata *Line of Code* (baris kode) setiap modul

Dalam persamaan tersebut terdapat persamaan lain yaitu *V(g)* atau *Cyclocmatic Compelxity*. *Cyclomatic Complexity* menurut Watson (1996) adalah salah satu cara menguji *software* dengan menggunakan bagian *source code* program. Menurut Watson (1996) untuk menghitung *Cyclomatic Complexity* dapat dilakukan dengan persamaan berikut ini.

$$V(g) = E - N + 2$$

Keterangan :

$V(g)$ = *Cyclomatic Complexity*

E = Banyak *edge* (panah)

N = Banyak *node* (lingkaran)

Sedangkan untuk mengetahui *Halstead Volume* (V) dapat digunakan persamaan berikut ini (Gorman, 2006).

$$V = N * \log(n)$$

Keterangan :

V = *Hastead Volume*

N = Jumlah operator

n = Jumlah operator yang berbeda

Dari persamaan *Halstead Volume* dan *Cyclomatic Complexity* tersebut dapat diketahui *Maintability Index* (MI) dengan nilai index antara 0 hingga 100 (Ganpati, 2012). Untuk mempermudah pencarian *Maintainability Index* (MI) dapat digunakan *software* Source Code SearchEngine. Source Code SearchEngine merupakan *software* yang dapat digunakan untuk mengetahui jumlah LOC, *cyclomatic complexy* dan *halstead volume* dari setiap modul pada program.

6. Portability

Menurut Salonen (2012) untuk mengetahui apakah suatu *software* berbasis web memenuhi aspek *portability*, *software* harus diuji dengan menggunakan 7 versi *browser* dari 3 *browser* yang paling banyak digunakan oleh pengguna pada perangkat desktop dan 5 *web browser* pada perangkat *mobile*. Untuk mengetahui sistem berbasis website memenuhi aspek *portability*, dapat dilakukan dengan menggunakan *software* yang dapat menguji sistem tersebut pada berbagai

browser yang berbeda. *Software* yang memiliki kemampuan untuk menguji *portability* suatu sistem berbasis web adalah BrowseEmAll. Cara kerja BrowseEmAll adalah dengan menjalankan *virtual web browser* yang kemudian akan menguji baik resolusi, CSS maupun JavaScript yang ada dalam website.

Dari semua teknik analisis data untuk menguji setiap aspek dalam ISO 9126 yang telah dijabarkan tersebut, maka dapat disederhanakan dalam bentuk tabel 10 berikut ini.

Tabel 10. Teknik Analisis Data Setiap Aspek ISO 9126

Aspek	Teknik Analisis	Intepretasi Hasil	Alat Pengumpul Data
<i>Functionality</i>	$X = 1 - \frac{A}{B}$	X lebih besar dari 0,5. Semakin mendekati 1 semakin baik.	<i>Test case</i>
	Mengetahui <i>level vulnerability</i>	<i>Level vulnerability</i> dibawah <i>Level 2 - Medium</i>	Acunetix Web Vulnerability Scanner
<i>Reliability</i>	$\frac{\sum success}{\sum total} \times 100\%$	Persentase sukses diatas 95%	Web Application Load, Stress and Performance Testing (WAPT)
<i>Usability</i>	Skala Likert	Skala Likert	<i>Usefulness, Satisfaction, and Easy of use (USE) Questionnaire</i>
<i>Efficiency</i>	Menghitung <i>load time</i> dengan GTMetrix	Kurang dari dari 10 detik	GTMetrix
<i>Maintainability</i>	Menghitung nilai <i>Maintainability Index (MI)</i>	Nilai MI diatas 65	Source Code SearchEngine
<i>Portability</i>	Mengetahui <i>error</i> yang terjadi	Tidak terdapat <i>error</i> di 7 <i>browser desktop</i> dan 5 <i>browser mobile</i>	BrowseEmAll

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Tahap Analisis Kebutuhan

Tahap analisis kebutuhan pengembangan sistem informasi jadwal akademik ini dilakukan analisis terhadap hasil observasi masalah-masalah bentuk informasi jadwal akademik yang ada di Jurusan Pendidikan Teknik Elektronika Universitas Negeri Yogyakarta. Dari hasil observasi tersebut didapatkan permasalahan sebagai berikut ini.

1. Informasi jadwal yang bersifat statis yaitu berupa kertas atau file PDF
2. Informasi jadwal hanya dikelompokkan berdasarkan kelas
3. Tidak terdapatnya informasi detail setiap jadwal dosen
4. Informasi perubahan jadwal tidak secara cepat diterima
5. Sulitnya mencari ruang kosong maupun jadwal dosen yang kosong

Dari permasalahan – permasalahan tersebut maka dapat dianalisis sehingga menghasilkan fitur dari sistem yang akan dikembangkan, sasaran pengguna sistem, informasi yang dibutuhkan dan spesifikasi sistem.

1. Fitur Sistem yang Dikembangkan

Sistem yang akan dikembangkan memiliki berbagai fitur untuk mengatasi masalah – masalah yang didapat dari hasil observasi yaitu,

- a. Informasi jadwal akademik dikelompokkan berdasarkan kelas
- b. Informasi jadwal mengajar dosen maupun jadwal aktivitas mahasiswa
- c. Informasi jadwal diluar kegiatan akademik
- d. Penambahan, pengubahan serta penghapusan jadwal secara dinamis
- e. Pemberitahuan terhadap perubahan jadwal

- f. Pencarian jadwal dosen maupun mahasiswa berdasarkan jadwal
- g. Informasi penggunaan ruangan
- h. Pencarian ruangan yang tidak digunakan pada waktu tertentu

2. Informasi yang Dibutuhkan

Informasi yang dibutuhkan dalam sistem informasi ini yaitu,

- a. Informasi data diri pengguna sistem berupa *username*, *password*, nama, alamat rumah, nomor telepon, alamat *email*, deskripsi diri serta foto profil
- b. Informasi jadwal meliputi informasi jam kuliah, informasi mata kuliah serta pelaksanaan kuliah
- c. Informasi ruangan meliputi nama ruangan, koordinat ruangan serta penggunaan ruangan
- d. Informasi jadwal pengguna diluar jadwal kegiatan akademik perkuliahan

3. Spesifikasi Sistem

- a. Sistem informasi jadwal yang akan dikembangkan tersebut memiliki spesifikasi sebagai berikut,
- b. Platform yang digunakan adalah web sehingga menggunakan *web browser* untuk dapat mengakses sistem
- c. Bahasa pemrograman yang digunakan untuk merancang sistem adalah PHP (Yii Framework), Javascript (Jquery), CSS3 (Twitter Bootstrap) dan HTML 5.
- d. *Database* yang digunakan adalah MySQL.
- e. *Software* yang digunakan untuk mengembangkan sistem adalah *text editor* Sublime Text 2, XAMPP (MySQL dan PHP server) dan *web browser* Google Chrome.

<i>Follow</i> Teman	Merupakan fitur untuk mengikuti teman. Notifikasi akan muncul jika teman yang diikuti melakukan perubahan, penambahan maupun penghapusan jadwal.
<i>Unfollow</i> Teman	Merupakan fitur untuk berhenti mengikuti teman (<i>unfollow</i>) sehingga tidak terdapat lagi notifikasi terhadap perubahan jadwal pada teman tersebut. <i>Unfollow</i> hanya bisa digunakan jika user sudah pernah di <i>follow</i> .
Lihat <i>Follower</i>	Merupakan fitur untuk melihat daftar teman yang mengikuti (<i>follower</i>).
Lihat <i>Following</i>	Merupakan fitur untuk melihat daftar teman yang diikuti (<i>following</i>)

Tabel 13. Definisi *Use Case Manage Kelas*

Nama Use Case	Deskripsi
Lihat Data Kelas	Merupakan fitur untuk melihat daftar kelas yang ada dalam sistem informasi jadwal.
<i>Join</i> Kelas	Merupakan fitur untuk bergabung/mengikuti (<i>join</i>) kelas. Notifikasi akan muncul jika kelas yang kita ikuti melakukan perubahan jadwal.
<i>Unjoin</i> Kelas	Merupakan fitur untuk berhenti mengikuti kelas (<i>unjoin</i>). Notifikasi perubahan jadwal tidak akan muncul lagi dari kelas tersebut.
Lihat Data Kelas	Merupakan fitur untuk melihat daftar kelas yang ada dalam sistem informasi jadwal.

Tabel 14. Definisi *Use Case Manage Jadwal*

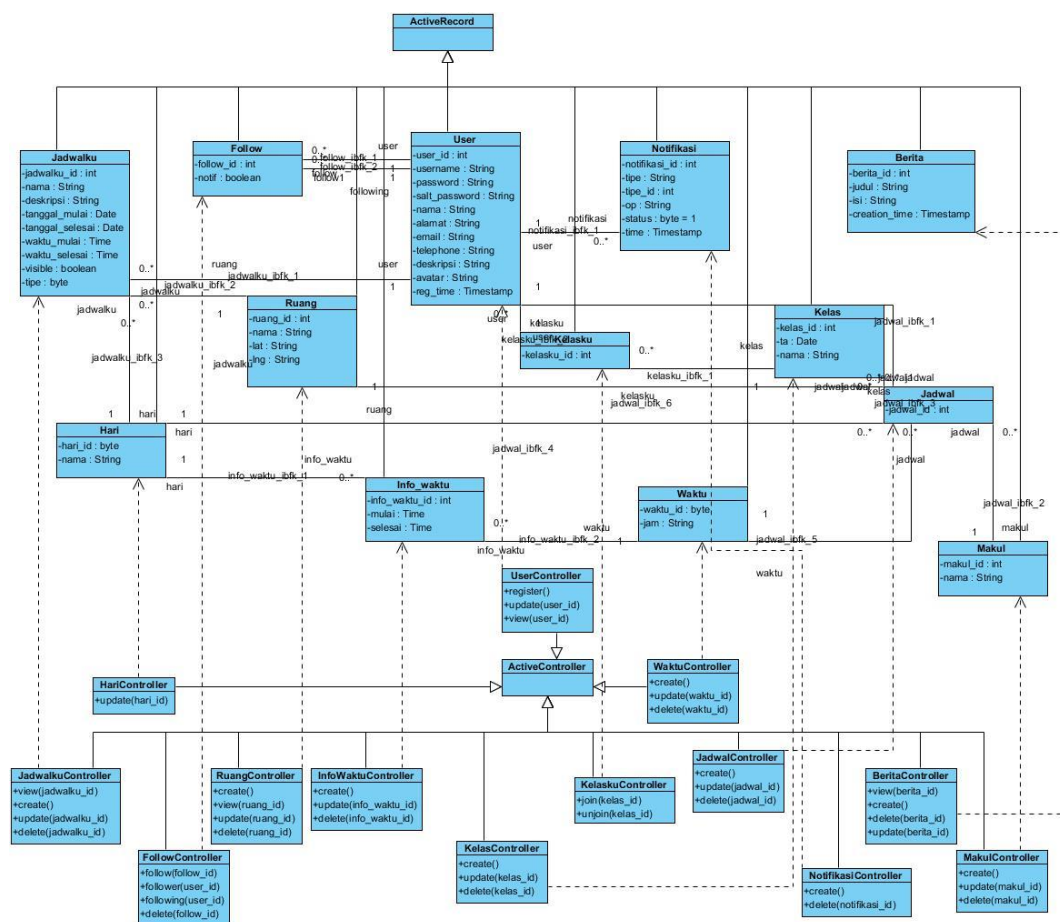
Nama Use Case	Deskripsi
Lihat Jadwal Akademik	Merupakan fitur untuk melihat seluruh data jadwal akademik

Lihat Detail Waktu	Merupakan fitur untuk melihat detail waktu pelaksanaan jadwal.
Lihat Data Jadwalku	Merupakan fitur untuk melihat semua data jadwal pribadi baik pada kelas yang kita ikuti maupun jadwal pribadi yang pernah dibuat
Tambah Jadwal Pribadi	Merupakan fitur untuk menambah jadwal pribadi diluar jadwal akademik
Ubah Jadwal Pribadi	Merupakan fitur untuk mengubah jadwal pribadi yang sudah pernah dibuat
Hapus Jadwal Pribadi	Merupakan fitur untuk menghapus jadwal pribadi dari jadwal yang sudah pernah dibuat.
Lihat Jadwal Teman	Merupakan fitur untuk melihat jadwal yang dimiliki teman
Cari Jadwal	Merupakan fitur untuk mencari jadwal ataupun mencari teman berdasarkan jadwal
Lihat Lokasi Jadwal	Merupakan fitur untuk melihat lokasi pelaksanaan jadwal pada hari ini di peta.

Tabel 15. Definisi *Use Case Manage Ruangan*

Nama Use Case	Deskripsi
Lihat Lokasi Ruang	Merupakan fitur untuk melihat lokasi ruangan di peta
Cari Ruang	Merupakan fitur untuk mencari ruangan
Lihat Data Ruang	Merupakan fitur untuk melihat seluruh data ruangan
Cari Ruang Kosong	Merupakan fitur untuk mencari ruangan yang sedang tidak digunakan pada waktu tertentu

Yii Framework mengimplementasikan *design patter Model-View-Controller* (MVC). Sehingga setiap kelas *controller* yang dibuat harus merupakan kelas turunan dari *ActiveController*. Sedangkan setiap kelas model yang dibuat harus merupakan turunan dari kelas *ActiveRecord*. Gambar 8 berikut merupakan *class diagram* dari sistem informasi jadwal akademik.

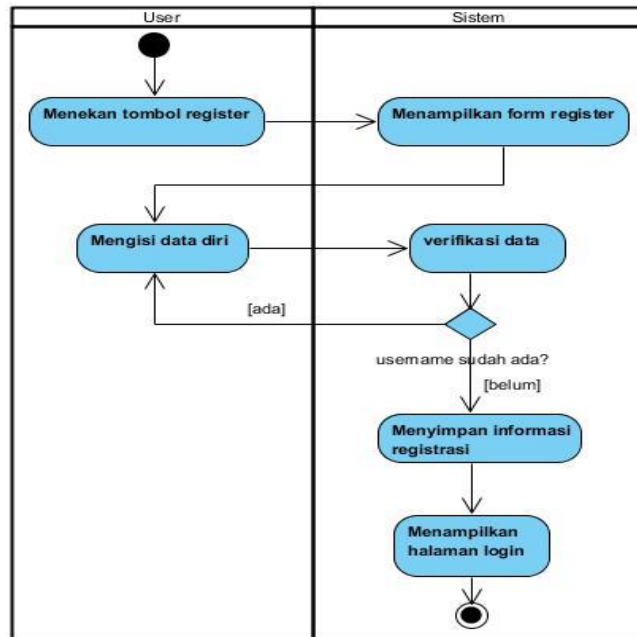


Gambar 8. *Class Diagram* Sistem Informasi Jadwal Akademik

3. Activity Diagram

a. Activity Diagram Fungsi Register

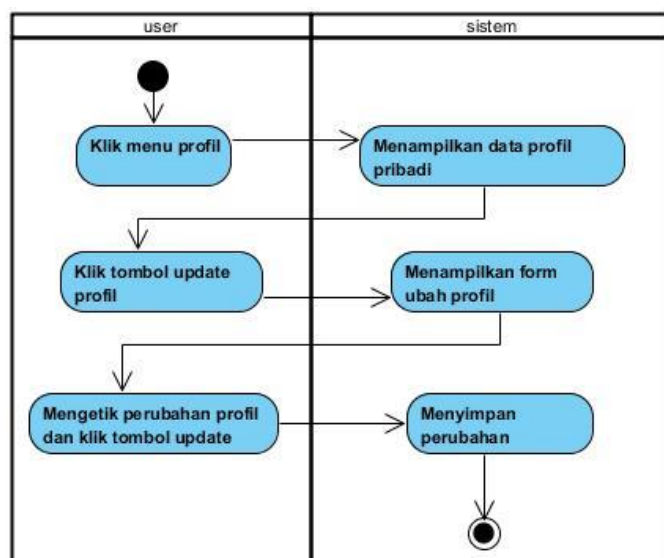
Gambar 9 berikut merupakan *activity diagram* dari fungsi *register*.



Gambar 9. Activity Diagram Register

b. Activity Diagram Fungsi Update Profil

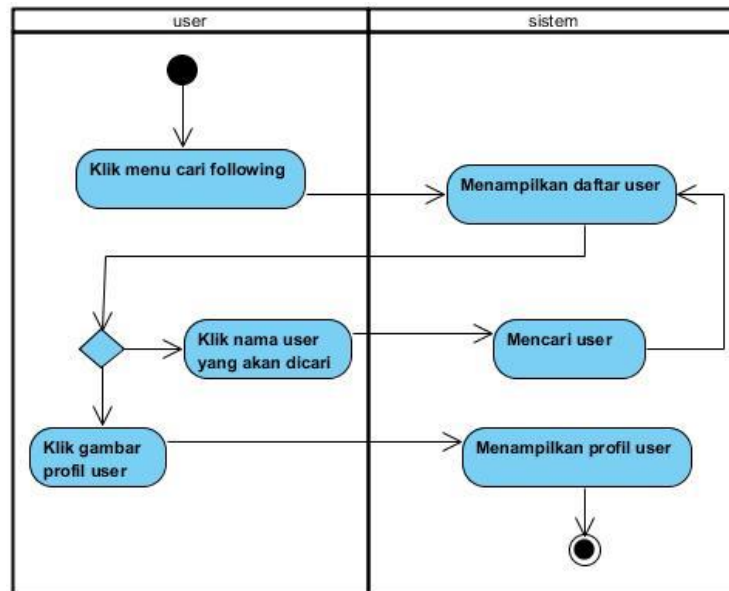
Gambar 10 berikut merupakan *activity diagram* dari fungsi *update profil*.



Gambar 10. Activity Diagram Update Profil

c. *Activity Diagram* Fungsi Lihat Profil Teman

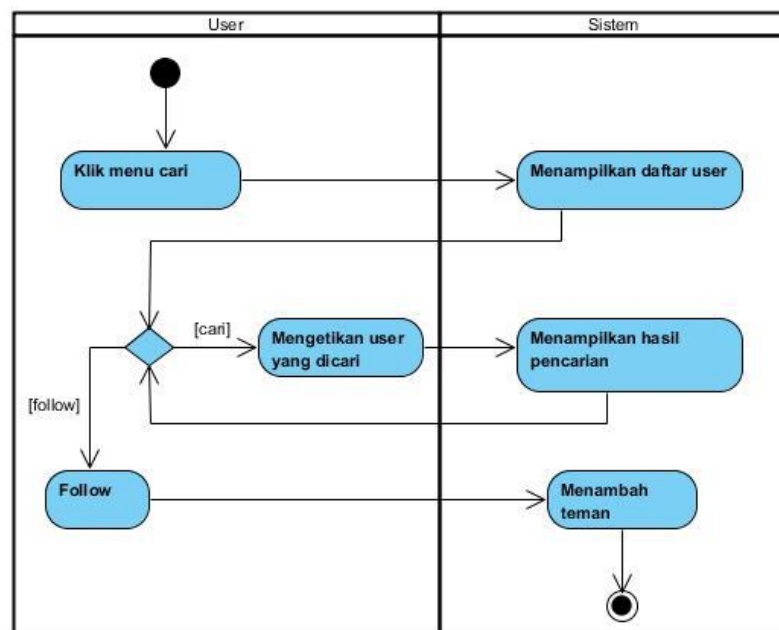
Gambar 11 berikut merupakan *activity diagram* dari fungsi lihat profil teman.



Gambar 11. *Activity Diagram* Lihat Profil User Lain

d. *Activity Diagram* Fungsi Tambah Teman

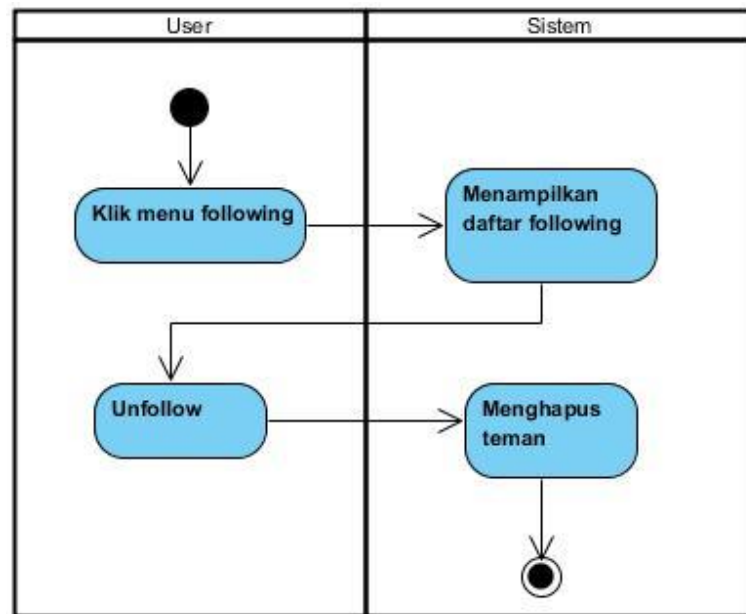
Gambar 12 berikut ini merupakan *activity diagram* dari fungsi tambah teman.



Gambar 12. *Activity Diagram* Tambah Teman

e. Activity Diagram Fungsi Unfollow

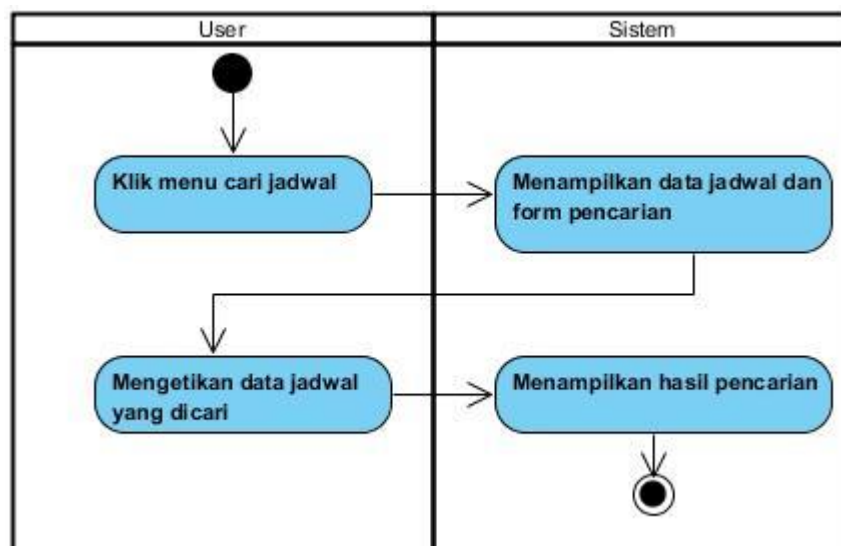
Gambar 13 berikut ini merupakan *activity diagram* dari fungsi *unfollow*.



Gambar 13. Activity Diagram Unfollow

f. Activity Diagram Fungsi Cari Jadwal

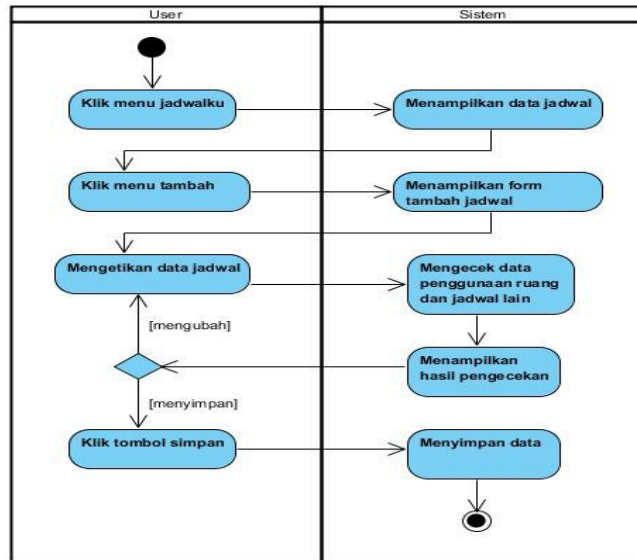
Gambar 14 berikut ini merupakan *activity diagram* mencari fungsi jadwal.



Gambar 14. Activity Diagram Mencari Jadwal

g. Activity Diagram Fungsi Tambah Jadwal

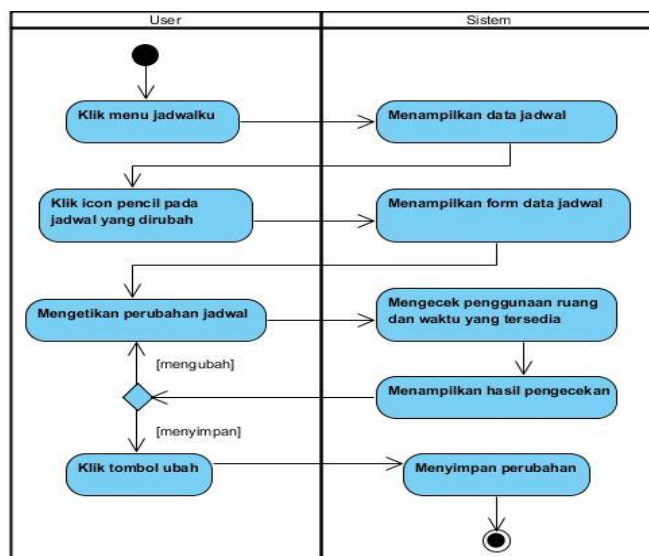
Gambar 15 berikut ini merupakan *activity diagram* dari fungsi menambah jadwal.



Gambar 15. Activity Diagram Menambah Jadwal

h. Activity Diagram Fungsi Ubah Jadwal

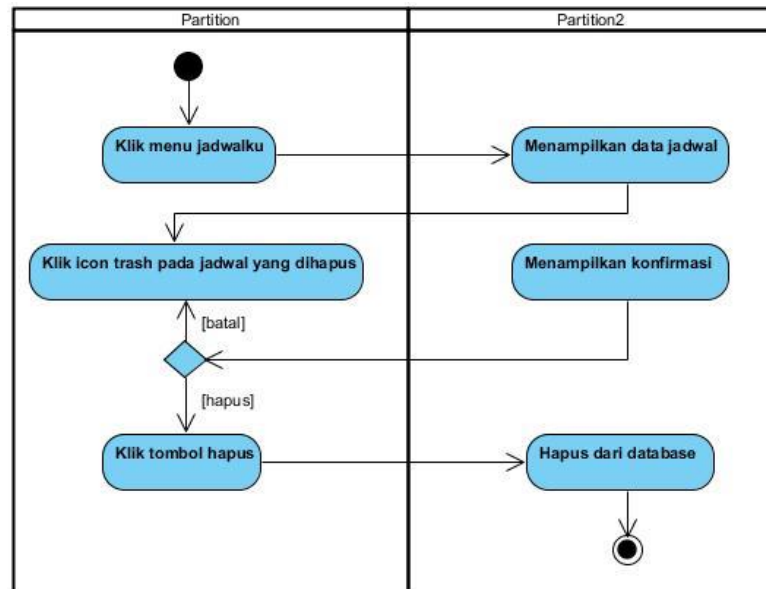
Gambar 16 berikut ini merupakan *activity diagram* dari fungsi mengubah jadwal.



Gambar 16. Activity Diagram Mengubah Jadwal

i. *Activity Diagram* Fungsi Hapus Jadwal

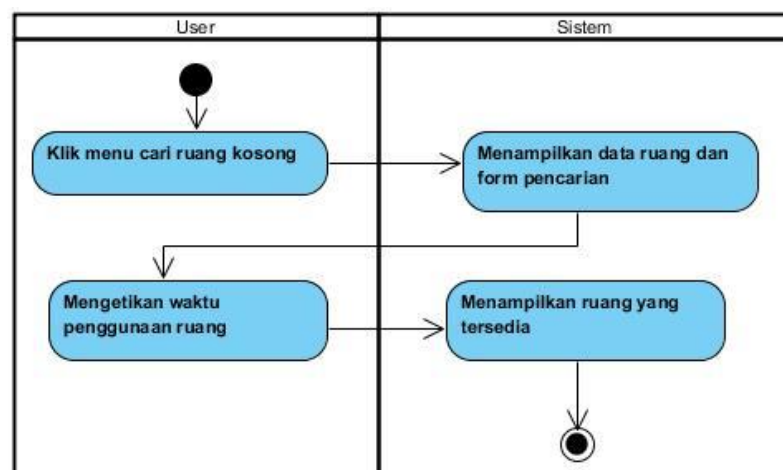
Gambar 17 berikut ini merupakan *activity diagram* dari fungsi menghapus jadwal.



Gambar 17. *Activity Diagram* Menghapus Jadwal

j. *Activity Diagram* Fungsi Cari Ruang Kosong

Gambar 18 berikut ini merupakan *activity diagram* dari fungsi mencari ruang kosong.

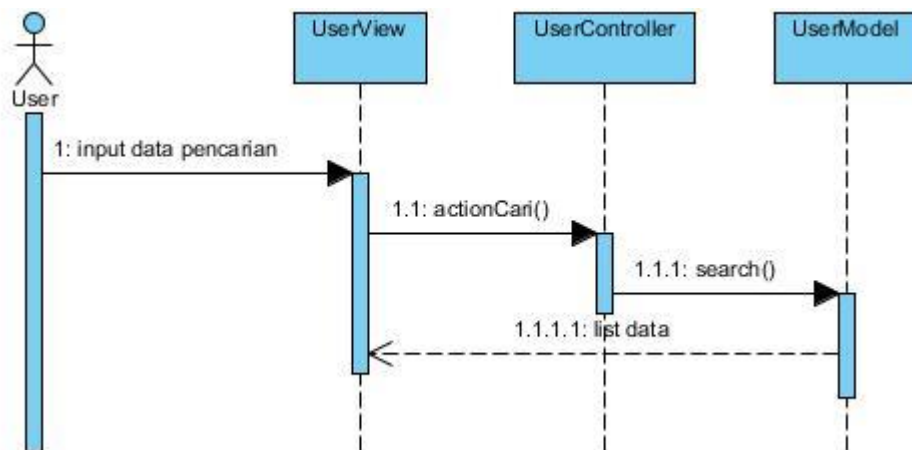


Gambar 18. *Activity Diagram* Mencari Ruang Kosong

4. Sequence Diagram

a. Sequence Diagram Fungsi Cari Teman

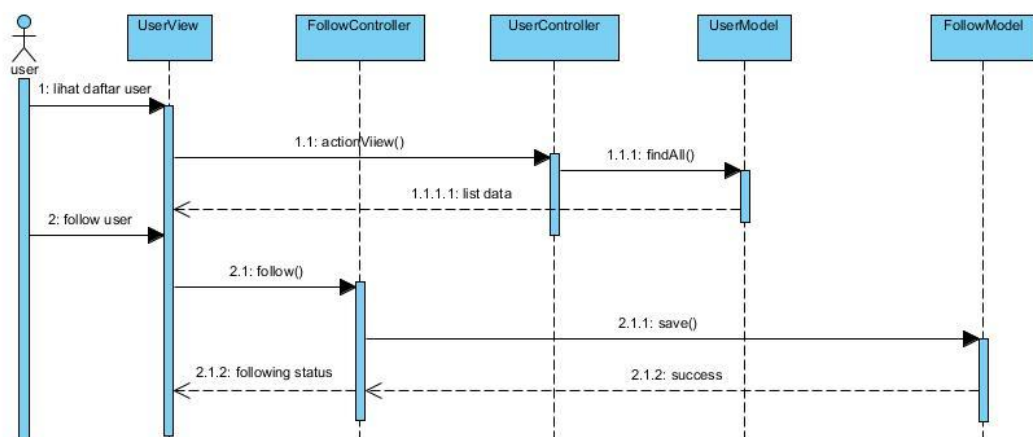
Gambar 19 berikut ini merupakan *sequence diagram* dari fungsi mencari teman.



Gambar 19. Sequence Diagram Cari Teman

b. Sequence Diagram Fungsi Follow Teman

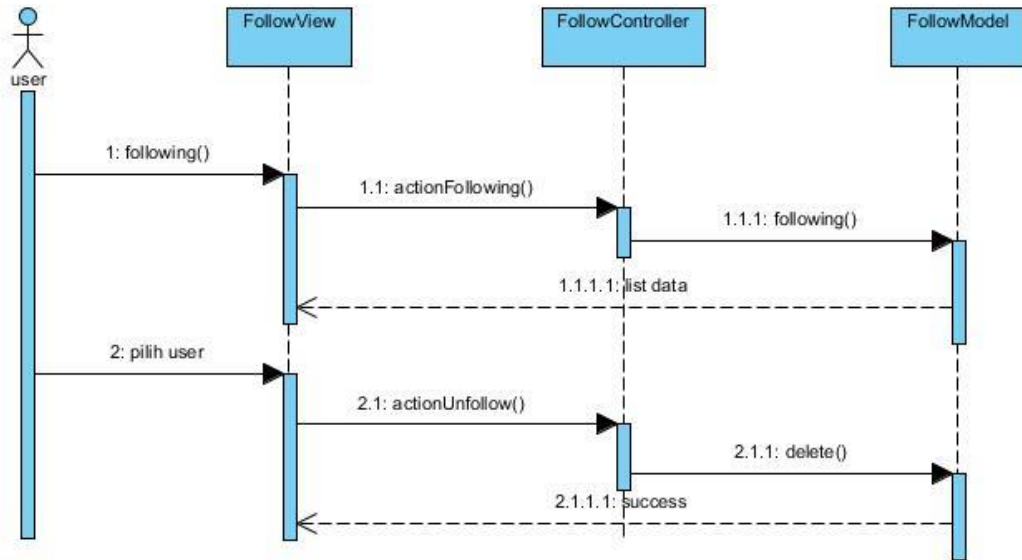
Gambar 20 berikut ini merupakan *sequence diagram* dari fungsi follow teman.



Gambar 20. Sequence Diagram Follow Teman

c. *Sequence Diagram Fungsi Unfollow Teman*

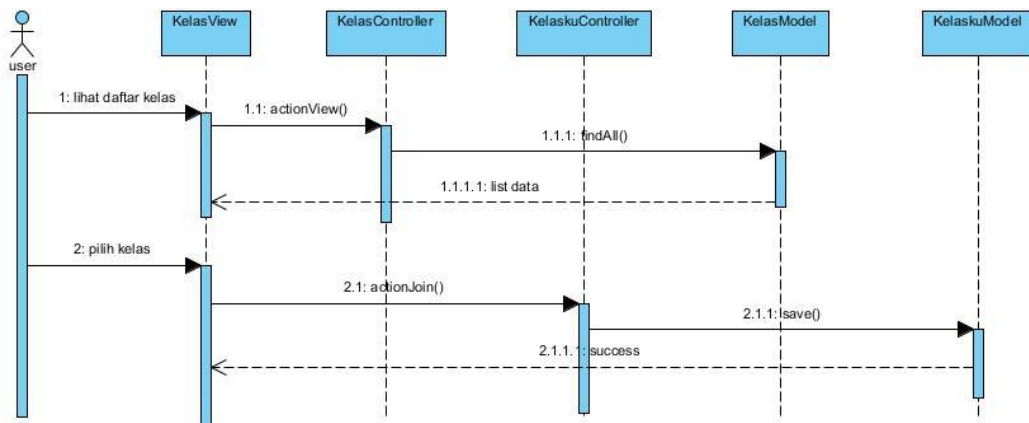
Gambar 21 berikut ini merupakan *sequence diagram* dari fungsi *unfollow* teman.



Gambar 21. *Sequence Diagram Unfollow Teman*

d. *Sequence Diagram Fungsi Join Kelas*

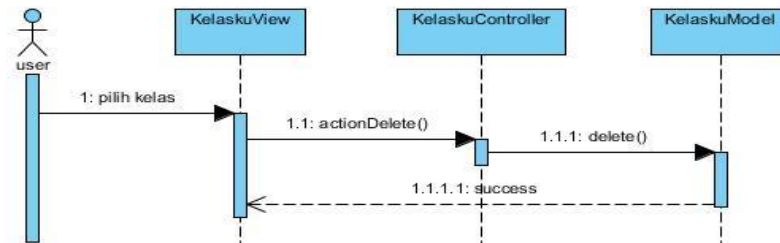
Gambar 22 berikut ini merupakan *sequence diagram* join kelas.



Gambar 22. *Sequence Diagram Join Kelas*

e. *Sequence Diagram Fungsi Unjoin Kelas*

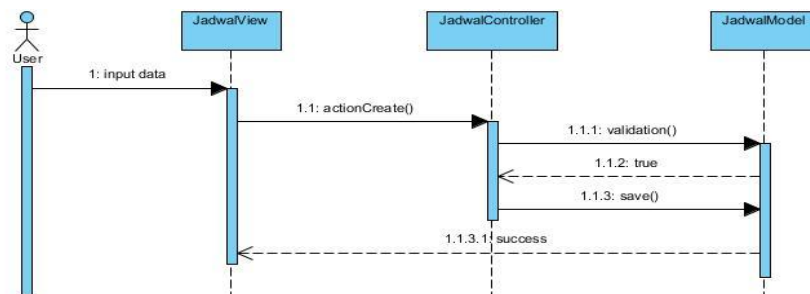
Gambar 23 berikut ini merupakan *sequence diagram* dari fungsi *unjoin* kelas.



Gambar 23. *Sequence Diagram Unjoin Kelas*

f. *Sequence Diagram Fungsi Tambah Jadwal*

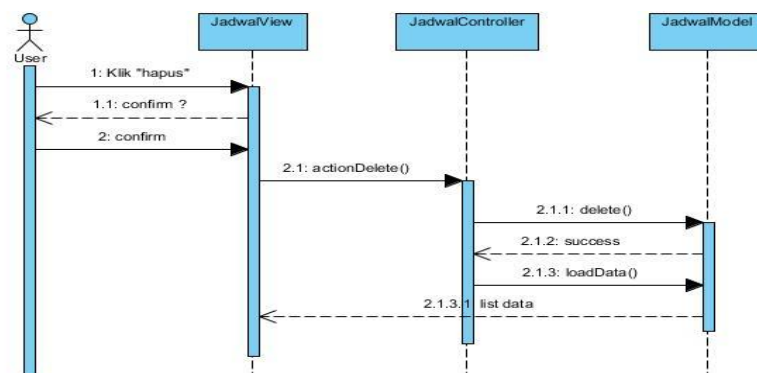
Gambar 24 berikut ini merupakan *sequence diagram* dari fungsi tambah jadwal.



Gambar 24. *Sequence Diagram Tambah Jadwal*

g. *Sequence Diagram Hapus Jadwal*

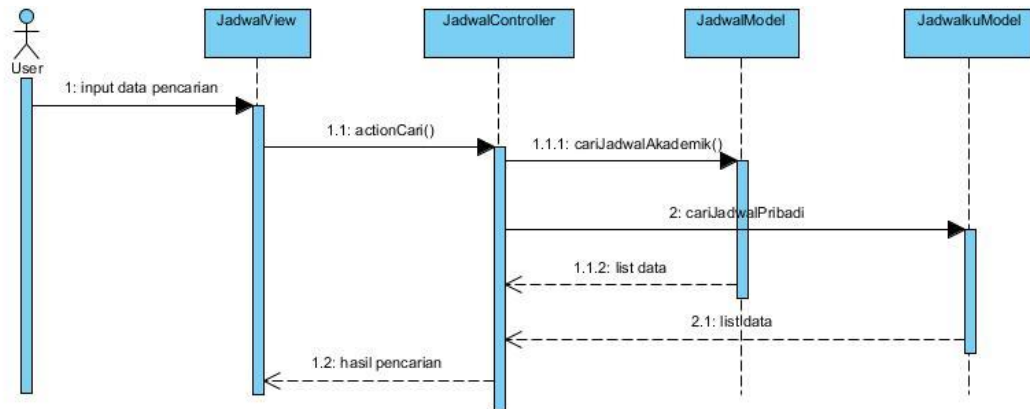
Gambar 25 berikut ini merupakan *sequence diagram* dari fungsi hapus jadwal.



Gambar 25. *Sequence Diagram Hapus Jadwal*

h. *Sequence Diagram* Fungsi Cari Jadwal

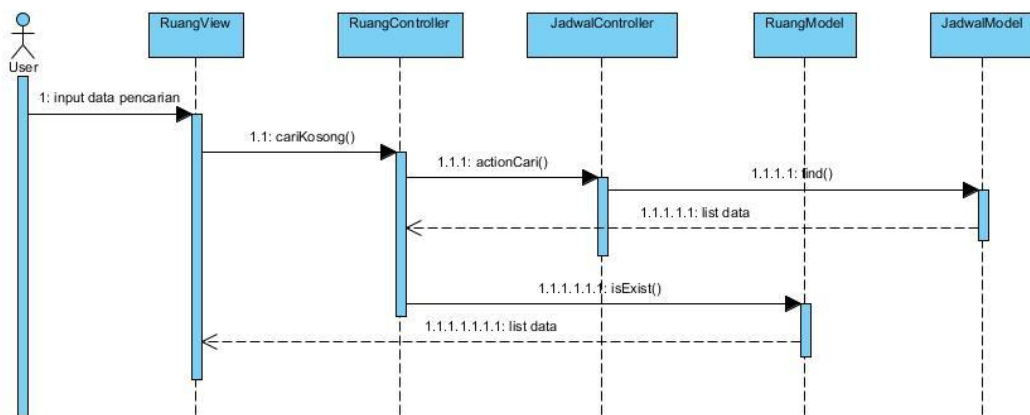
Gambar 26 berikut ini merupakan *sequence diagram* dari fungsi cari jadwal.



Gambar 26. *Sequence Diagram* Cari Jadwal

i. *Sequence Diagram* Fungsi Cari Ruang Kosong

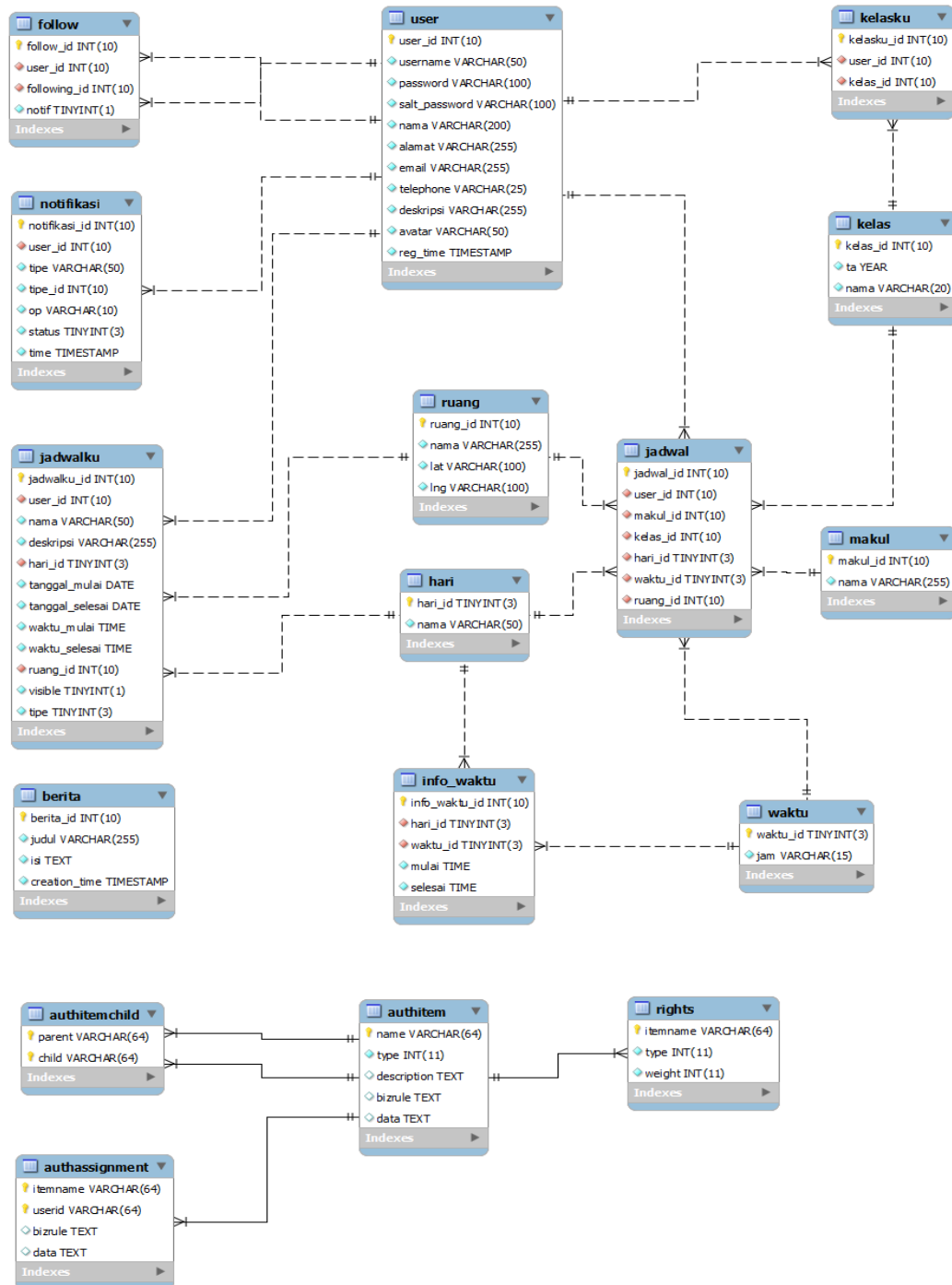
Gambar 27 berikut ini merupakan *sequence diagram* dari fungsi mencari ruang kosong.



Gambar 27. *Sequence Diagram* Cari Ruang Kosong

5. Entity Relational Diagram (ERD)

Gambar 28 berikut ini merupakan *Entity Relational Diagram (ERD)* dari sistem informasi jadwal akademik.



Gambar 28. *Entity Relational Diagram (ERD)*

6. Desain Tampilan *User Interface* (UI)

a. Desain *User Interface* Halaman Home

Gambar 29 berikut ini merupakan desain *user interface* dari halaman *home*.

Sistem Informasi Jadwal Akademik		Login	Register
Menu	Berita terbaru		

Gambar 29. Desai UI Halaman Home

b. Desain *User Interface* Halaman Teman

Gambar 30 berikut merupakan desain *user interface* dari halaman teman.

Sistem Informasi Jadwal Akademik		task	notif	Welcome, user
Menu	Teman >> cari teman			
	Klik untuk mencari			
	user1	user2	user3	user4
	user5	user6		

Gambar 30. Desain UI Halaman Teman

c. Desain *User Interface* Halaman Jadwal Akademik

Gambar 31 berikut ini merupakan desain halaman *user interface* dari halaman jadwal akademik.

Sistem Informasi Jadwal Akademik			task	notif	Welcome, user
Menu	Jadwal Akademik				

Gambar 31. Desain UI Halaman Jadwal Akademik

d. Desain *User Interface* Halaman Jadwal Pribadi

Gambar 32 berikut ini merupakan desain *user interface* dari halaman jadwal pribadi.

Sistem Informasi Jadwal Akademik			task	notif	Welcome, user
Menu	Jadwal Pribadi				

submenu

Gambar 32. Desain UI Halaman Jadwal Pribadi

e. Desain *User Interface* Halaman Cari Jadwal

Gambar 33 berikut ini merupakan desain *user interface* dari halaman cari jadwal.

Sistem Informasi Jadwal Akademik		task	notif	Welcome, user									
Menu	Jadwal >> cari jadwal												
	<table border="1"> <tr><td colspan="3"></td></tr> <tr> <td></td> <td></td> <td></td> </tr> <tr><td colspan="3"></td></tr> </table>											<div>Form cari</div> <div> <input type="text"/> <input type="text"/> <input type="text"/> </div>	

Gambar 33. Desain UI Halaman Cari Jadwal

f. Desain *User Interface* Halaman Informasi Ruangan

Gambar 34 berikut ini merupakan user interface dari halaman informasi ruangan.

Sistem Informasi Jadwal Akademik		task	notif	Welcome, user															
Menu	Ruangan																		
	<table border="1"> <tr><td colspan="5"></td></tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>																		
<div>peta</div>																			

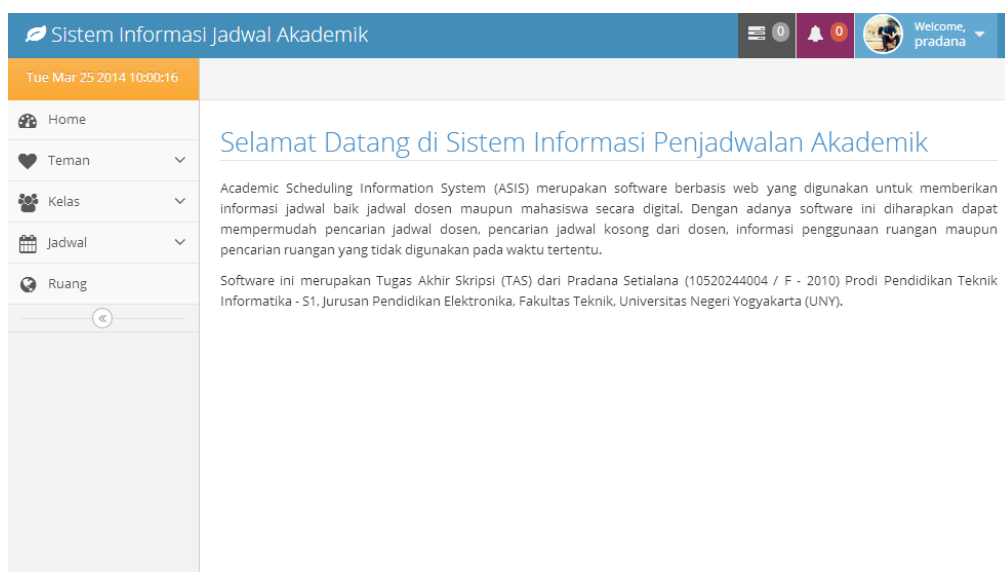
Gambar 34. Desain UI Halaman Informasi Ruangan

C. Tahap Implementasi

Yii Framework kelebihan dibandingkan framework lain yaitu adanya fitur *software reusability* dalam bentuk *extensions* dan modul. *Extension* dan modul pada prinsipnya adalah dapat memanfaatkan kode program yang sudah pernah dikembangkan oleh orang lain ke dalam *software* lain. Dalam implementasi pengembangan sistem informasi jadwal akademik, digunakan *extensions* dan modul untuk mempercepat pengembangan *software* diantaranya yaitu YiiBooster, EGMap, Rights, AjaxUploader dan PHPExcel. Berikut merupakan hasil implementasi pengembangan sistem informasi jadwal akademik pada setiap halaman.

1. Implementasi Halaman Utama

Halaman utama dari sistem menampilkan berita maupun informasi terbaru dari sistem informasi jadwal akademik. Sedangkan pada bagian atas terdiri dari notifikasi dan informasi user yang telah login. Pada bagian samping kiri terdapat menu terhadap fungsi-fungsi yang ada di dalam sistem. Gambar 35 berikut merupakan implementasi dari halaman utama.



Gambar 35. Halaman Utama

2. Implementasi Halaman Profil

Halaman profil menampilkan informasi data user baik data sendiri maupun informasi profil orang lain. Informasi yang ada pada halaman profil mencakup foto profil, *username*, alamat, telephone, email, tanggal registrasi, deskripsi diri, jumlah *following* dan *follower* serta informasi jadwal baik jadwal kuliah yang diikuti maupun jadwal yang dibuat sendiri. Gambar 36 berikut merupakan implementasi dari halaman profil.



Gambar 36. Halaman Profil

3. Implementasi Halaman Data Kelas

Halaman data kelas berisi informasi mengenai daftar kelas yang ada di dalam sistem yang mencakup nama kelas, semester serta kategori kelas. User dapat mengikuti suatu kelas dan mendapatkan informasi jadwal dari kelas tersebut dengan cara *join* kelas. Sedangkan untuk berhenti mendapatkan informasi jadwal dari kelas dapat dilakukan dengan cara *unjoin* kelas. Gambar 37 berikut merupakan implementasi dari halaman data kelas.

#	Nama Kelas	Semester	Kategori Kelas	Tindakan
1	F	1	Teori	JOIN
2	F	2	Teori	JOIN
3	F	3	Teori	JOIN
4	F	4	Teori	JOIN
5	F	5	Teori	JOIN
6	F	6	Teori	UNJOIN

Gambar 37. Halaman Data Kelas

4. Implementasi Halaman Kelas yang Diikuti

Halaman data kelas yang diikuti (kelasku) adalah halaman yang menampilkan data kelas mana yang sedang diikuti dan mendapatkan jadwal dari kelas tersebut. Dalam halaman ini, user juga dapat berhenti mengikuti kelas. Gambar 38 berikut merupakan implementasi dari halaman kelas yang diikuti.

#	Nama Kelas	Semester	Tindakan
1	F	6	👁️
2	Sistem Informasi	6	👁️

Gambar 38. Halaman Data Kelas yang Diikuti

5. Implementasi Halaman Jadwal Akademik

Halaman jadwal akademik berisi informasi mengenai semua data jadwal akademik. Data jadwal akademik bisa dicari berdasarkan kelas, hari, jam kuliah,

mata kuliah, dosen serta ruangan. Gambar 39 berikut merupakan implementasi halaman jadwal akademik.

Sistem Informasi Jadwal Akademik

4

0

Welcome, pradana

Tue Mar 25 2014 09:48:00

Home / Jadwal

Home

Teman

Kelas

Jadwal

Jadwal Akademik

Detail Waktu Kuliah

Data Jadwalku

Lihat Jadwal Teman

Cari Jadwal

Ruang

Akademik » jadwal kuliah

#	Kelas Semester	Hari	Jam	Mata Kuliah	Dosen	Ruangan
	F E					
1	F 6	Selasa	7	Pendidikan Teknologi dan Kejuruan	Prof. Drs. Soenarto, M.Sc., MA., Ph.D.	RE 4
2			8			
3		Rabu	1	Sosio- antropologi Pendidikan	Siti	Teater 2
4			2			
5			5	Manajemen Industri	Ponco Wali	LPTK 5
6			6			

Gambar 39. Halaman Jadwal Akademik

6. Implementasi Halaman Cari Jadwal

Halaman cari jadwal adalah halaman yang berfungsi untuk mencari jadwal lebih detail. Halaman cari jadwal dapat mencari jadwal berdasarkan username, nama jadwal, tanggal jadwal, waktu jadwal maupun nama ruang. Informasi hasil pencarian sudah meliputi jadwal dari kelas yang diikuti oleh user, jadwal mengajar serta jadwal pribadi. Gambar 40 berikut merupakan implementasi halaman cari jadwal.

Data Jadwal

Jadwal Event

Username	Kegiatan	Tanggal Mulai	Tanggal Selesai	Waktu Mulai	Waktu Selesai	Nama
No results found.						

Jadwal Rutin

Username	Kegiatan	Hari	Waktu Mulai	Waktu Selesai	Nama
----------	----------	------	-------------	---------------	------

Cari Jadwal

User: 195804221984031002 | Drs.

Nama Jadwal: *

Tanggal Jadwal: 25/03/2014

Waktu Jadwal: *

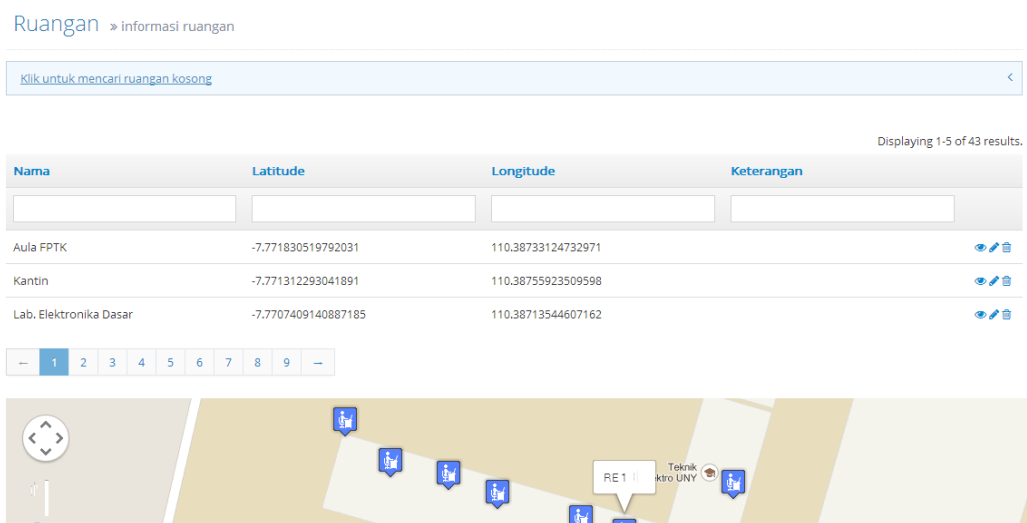
Nama Ruang: *

Gambar 40. Halaman Cari Jadwal

7. Implementasi Halaman Data Ruang

Halaman ruang memberikan informasi daftar ruang serta lokasi ruangan tersebut di peta. Halaman ini juga terdapat fungsi pencarian ruangan kosong atau ruangan yang sedang tidak digunakan berdasarkan tanggal dan waktu. Hasil pencarian akan ditampilkan dalam bentuk nama ruang dan lokasinya dalam peta.

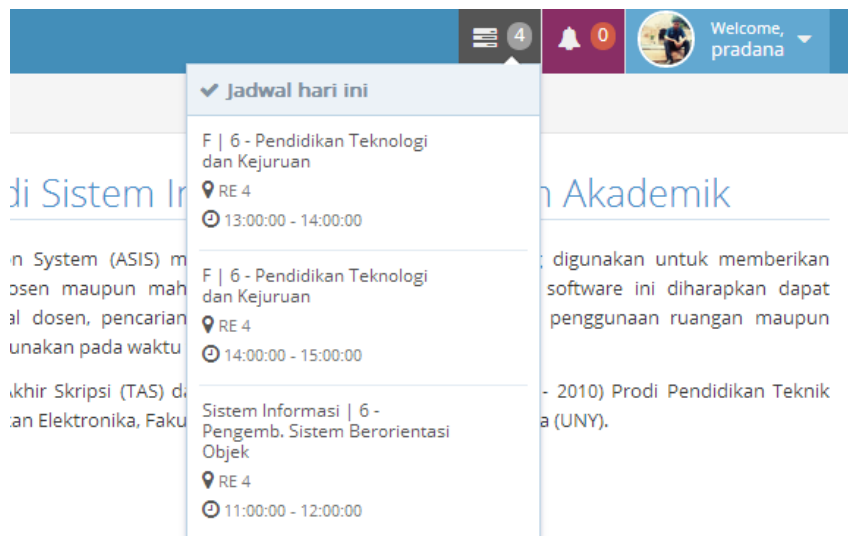
Gambar 41 berikut merupakan implementasi dari halaman data ruang.



Gambar 41. Halaman Data Ruang

8. Implementasi Notifikasi

Notifikasi merupakan fitur pemberitahuan dari sistem yang terdapat dibagian atas semua halaman dari sistem informasi jadwal akademik. Terdapat 2 macam notifikasi yaitu notifikasi jadwal yang berlangsung hari ini dan notifikasi adanya follower baru serta notifikasi perubahan, penambahan maupun penghapusan jadwal dari user yang diikuti maupun kelas yang diikuti. Gambar 42 implementasi dari notifikasi.



Gambar 42. Tampilan Notifikasi Jadwal

D. Tahap Pengujian

1. *Functionality*

Berdasarkan hasil dari pengujian *functionality* yang dilakukan oleh 3 orang yang terdiri dari *programmer*, *web developer* dan *software developer* dengan menggunakan *test case*, maka didapatkan hasil berikut ini. Tabel 16 berikut merupakan hasil pengujian *functionality* yang dilakukan oleh ahli dalam pengembangan *software* berbasis web.

Tabel 16. Hasil Pengujian *Functionality*

No	Fungsi	Pernyataan	Jumlah Penjawab	
			Lolos	Gagal
User				
1	Login	Fungsi login sudah berfungsi dengan baik	3	0
2	Register	Fungsi register sudah berfungsi dengan baik	3	0
3	Lihat profile	Fungsi lihat profile sudah berfungsi dengan biak	3	0
4	Ubah profile	Fungsi ubah data profile sudah berfungsi dengan baik	3	0

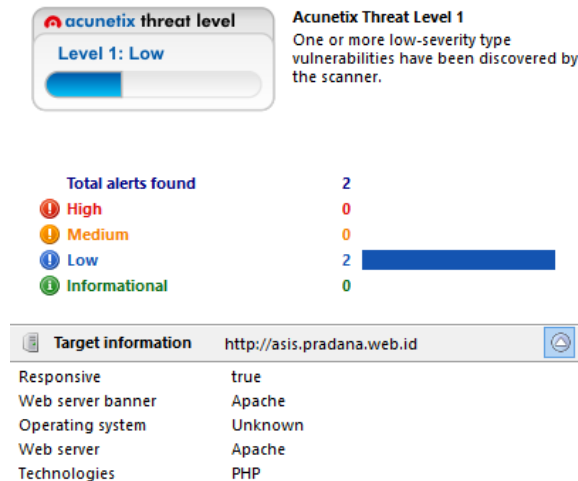
5	Akses user	Fungsi manajemen akses user sudah berfungsi dengan baik	3	0
6	Import data user	Fungsi import data user dari <i>file Excel (x/s)</i> sudah berfungsi dengan baik	3	0
Berita				
7	Tambah berita	Fungsi tambah berita sudah berfungsi dengan baik	3	0
8	Ubah berita	Fungsi ubah berita sudah berfungsi dengan baik	3	0
9	Hapus berita	Fungsi hapus berita sudah berfungsi dengan baik	3	0
Teman				
10	Lihat daftar user	Fungsi lihat daftar semua teman sudah berfungsi dengan baik	3	0
11	Cari teman	Fungsi cari teman sudah berfungsi dengan baik	3	0
12	Follow teman	Fungsi <i>follow</i> teman sudah berfungsi dengan baik	3	0
13	Unfollow Teman	Fungsi <i>unfollow</i> teman sudah berfungsi dengan baik	3	0
14	Lihat following	Fungsi lihat teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	3	0
15	Lihat follower	Fungsi lihat teman yang mengikuti (<i>follower</i>) sudah berfungsi dengan baik	3	0
16	Ubah notifikasi following	Fungsi ubah notifikasi (notifikasi aktif/nonaktif) dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	3	0
Kelas				
17	Lihat daftar kelas	Fungsi melihat semua daftar kelas sudah berfungsi dengan baik	3	0
18	Cari Kelas	Fungsi mencari kelas sudah berfungsi dengan baik	3	0
19	Join Kelas	Fungsi mengikuti kelas (<i>join</i>) sudah berfungsi dengan baik	3	0
20	Unjoin Kelas	Fungsi keluar dari kelas yang diikuti (<i>unjoin</i>) sudah berfungsi dengan baik	3	0
21	Tambah kelas	Fungsi tambah data kelas sudah berfungsi dengan baik	3	0
22	Ubah kelas	Fungsi ubah data kelas sudah berfungsi dengan baik	3	0
23	Hapus kelas	Fungsi hapus data kelas sudah berfungsi dengan baik	3	0

24	Data kelas yang diikuti	Fungsi melihat daftar kelas yang diikuti sudah berfungsi dengan baik	3	0
Jadwal				
25	Lihat data jadwal akademik	Fungsi lihat data jadwal akademik sudah berfungsi dengan baik	3	0
26	Cari data jadwal akademik	Fungsi mencari data jadwal akademik sudah berfungsi dengan baik	3	0
27	Tambah jadwal akademik	Fungsi menambah jadwal akademik sudah berfungsi dengan baik	3	0
28	Ubah jadwal akademik	Fungsi mengubah jadwal akademik sudah berfungsi dengan baik	3	0
29	Hapus jadwal akademik	Fungsi hapus jadwal akademik sudah berfungsi dengan baik	3	0
30	Lihat jadwal pribadi	Fungsi lihat data jadwal pribadi sudah berfungsi dengan baik	3	0
31	Tambah jadwal pribadi	Fungsi tambah jadwal pribadi sudah berfungsi dengan baik	3	0
32	Ubah jadwal pribadi	Fungsi ubah data jadwal pribadi sudah berfungsi dengan baik	3	0
33	Hapus jadwal pribadi	Fungsi hapus data jadwal pribadi sudah berfungsi dengan baik	3	0
34	Lihat jadwal saat ini	Fungsi lihat data jadwal yang berlangsung hari ini sudah berfungsi dengan baik	3	0
35	Cari jadwal	Fungsi cari jadwal sudah berfungsi dengan baik	3	0
36	Lihat lokasi pelaksanaan jadwal	Fungsi lihat lokasi ruangan dari jadwal yang berlangsung hari ini sudah berfungsi dengan baik	3	0
Ruang				
37	Lihat daftar ruangan	Fungsi lihat daftar ruangan sudah berfungsi dengan baik	3	0
38	Lihat ruangan di peta	Fungsi lihat lokasi ruangan di peta sudah berfungsi dengan baik	3	0
39	Cari ruangan	Fungsi cari ruangan sudah berfungsi dengan baik	3	0
40	Cari ruang kosong	Fungsi cari ruangan yang tidak digunakan sudah berfungsi dengan baik	3	0

41	Tambah ruangan	Fungsi tambah data ruangan sudah berfungsi dengan baik	3	0
42	Ubah ruangan	Fungsi ubah data ruangan sudah berfungsi dengan baik	3	0
43	Hapus ruangan	Fungsi hapus data ruangan sudah berfungsi dengan baik	3	0
Notifikasi				
44	Notifikasi tambah jadwal dari following	Fungsi notifikasi adanya penambahan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	3	0
45	Notifikasi pengubahan jadwal dari following	Fungsi notifikasi adanya pengubahan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	3	0
46	Notifikasi hapus jadwal dari following	Fungsi notifikasi adanya penghapusan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	3	0
47	Notifikasi follower baru	Fungsi notifikasi adanya teman baru yang mengikuti (<i>follower</i>) sudah berfungsi dengan baik	3	0
48	Notifikasi tambah jadwal dari kelas	Fungsi notifikasi adanya penambahan jadwal dari kelas yang diikuti (<i>join</i>) sudah berfungsi dengan baik	3	0
49	Notifikasi ubah jadwal dari kelas	Fungsi notifikasi adanya pengubahan jadwal dari kelas yang diikuti (<i>join</i>) sudah berfungsi dengan baik	3	0
50	Lihat semua notifikasi	Fungsi lihat semua notifikasi baik yang sudah dibaca maupun belum, sudah berfungsi dengan baik	3	0
51	Notifikasi jadwal hari ini	Fungsi notifikasi jadwal yang berlangsung hari ini sudah berfungsi dengan baik	3	0
52	Lihat lokasi pelaksanaan jadwal	Fungsi lihat lokasi ruangan dari jadwal yang berlangsung hari ini sudah berfungsi dengan baik	3	0

Dari data tersebut dapat diketahui bahwa semua penguji *functionality* menyatakan lolos terhadap fungsi yang ada pada sistem informasi jadwal akademik.

Sedangkan untuk pengujian *security*, gambar 43 berikut ini merupakan hasil pemindaian sistem informasi jadwal akademik dengan menggunakan *software Acunetix Web Vulnerability Scanner*.

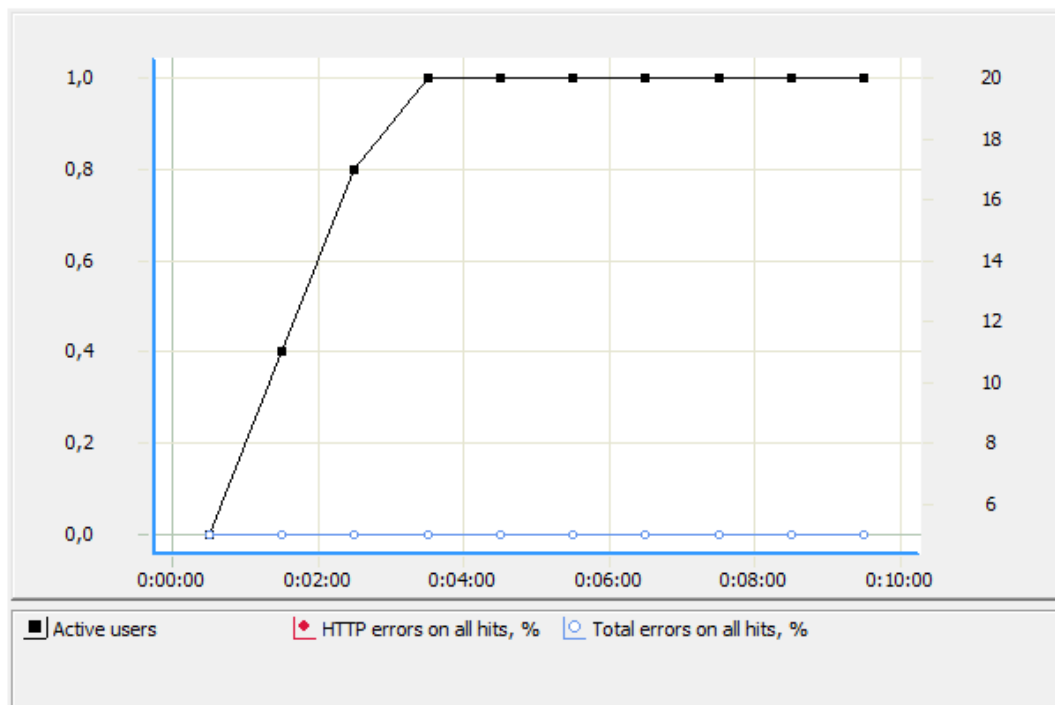


Gambar 43. Hasil Pengujian Keamanan Sistem

Hasil pengujian tersebut menghasilkan *Level 1: Low* yang berarti bahwa website rendah terhadap kerentanan (*vulnerability*) terdapat serangan *SQL Injection* maupun *Cross Site Scripting (XSS)*.

2. Reliability

Pengujian *reliability* dari *software* dilakukan dengan menguji *stress testing* dengan menggunakan *Web Application Load, Stress and Performance Testing (WAPT)*. WAPT menguji sistem dengan cara menjalankan 20 *virtual user* secara bertahap yaitu setiap 10 detik sekali *user* akan bertambah 1 hingga mencapai jumlah maksimal yaitu 20 *user* selama 10 menit. Gambar 44 berikut ini merupakan hasil pengujian *stress testing* dari sistem informasi jadwal akademik.



Gambar 44. Grafik Hasil Stress Testing

Selain dalam bentuk grafik, WAPT juga menghasilkan data pengujian dalam bentuk tabel yang berisi jumlah session, pages dan hits dalam setiap menit. Gambar 45 berikut merupakan jumlah *page*, *session* dan *hits* dari pengujian.

Successful sessions (Failed sessions)											
Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
local	0(0)	11(0)	17(0)	27(0)	26(0)	30(0)	29(0)	29(0)	30(0)	28(0)	227(0)
Total	0(0)	11(0)	17(0)	27(0)	26(0)	30(0)	29(0)	29(0)	30(0)	28(0)	227(0)

Successful pages (Failed pages)											
Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
local	32(0)	132(0)	228(0)	295(0)	317(0)	318(0)	319(0)	314(0)	314(0)	318(0)	2587(0)
Total	32(0)	132(0)	228(0)	295(0)	317(0)	318(0)	319(0)	314(0)	314(0)	318(0)	2587(0)

Successful hits (Failed hits)											
Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
local	215(0)	874(0)	1459(0)	1912(0)	2004(0)	2015(0)	2050(0)	1976(0)	2042(0)	1972(0)	16519(0)
Total	215(0)	874(0)	1459(0)	1912(0)	2004(0)	2015(0)	2050(0)	1976(0)	2042(0)	1972(0)	16519(0)

Gambar 45. Jumlah Session, Pages dan Hits Setiap Menit

Dari gambar 45 tersebut maka dapat dijelaskan dalam bentuk tabel 17 berikut ini.

Tabel 17. Jumlah Total Success dan Failed

No	Nama	Total Success	Total Failed	Jumlah
1	Session	227	0	227
2	Pages	2587	0	2587
3	Hits	16519	0	16519

3. Usability

Hasil pengujian reliabilitas pada kuesioner USE dengan menggunakan *software* IBM SPSS versi 20 didapatkan hasil sesuai pada gambar 46 berikut ini.

Case Processing Summary

		N	%
Cases	Valid	27	90,0
	Excluded ^a	3	10,0
	Total	30	100,0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	N of Items
,976	30

Gambar 46. Hasil Pengujian Reliabilitas Kuesioner dengan SPSS

Sedangkan hasil dari pengisian kuesioner oleh responden maka didapatkan hasil pada tabel 18 berikut ini.

Tabel 18. Jumlah Jawaban Poin Skala Likert

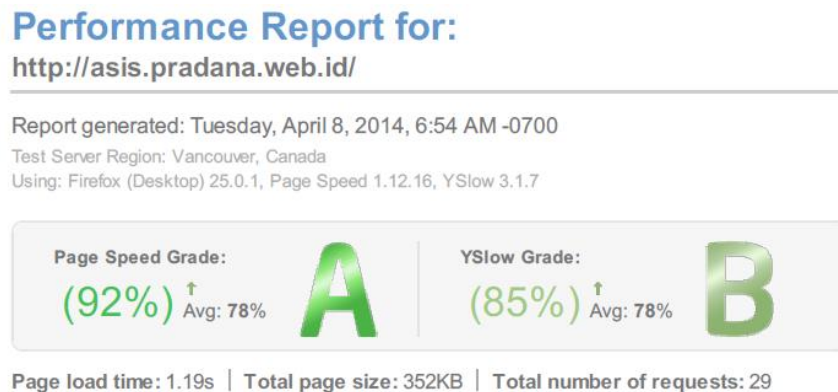
Poin	Keterangan	Jumlah Jawaban
1	Sangat tidak setuju	1
2	Tidak setuju	10
3	Sedikit tidak setuju	35
4	Ragu – ragu	130
5	Sedikit setuju	232
6	Setuju	315
7	Sangat setuju	173

4. Efficiency

Berikut ini merupakan hasil dari GTMatrix pada setiap halaman fungsi utama dari sistem informasi jadwal akademik.

a. Hasil Pengujian *Efficiency* Halaman Utama

Gambar 47 berikut ini merupakan hasil pengujian dengan GTMatrix pada halaman utama.

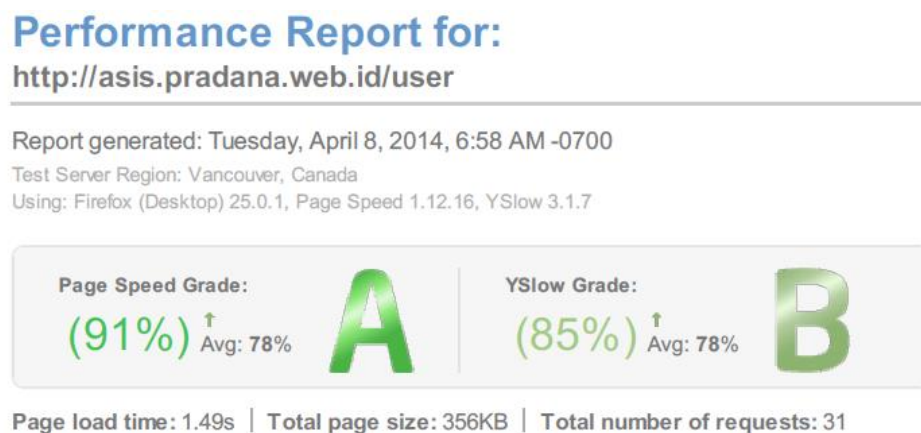


Gambar 47. Hasil GTMatrix Halaman Utama

Dari hasil pengujian pada gambar 47 tersebut dapat terlihat bahwa halaman utama memiliki *page load time* adalah 1,19 detik, *total page size* adalah 352KB, *total number of request* adalah 29, *Page Speed Grade* adalah A (92%) dan *YSlow Grade* adalah B (85%)

b. Hasil Pengujian *Efficiency* Halaman Cari Teman

Gambar 48 berikut ini merupakan hasil pengujian dengan GTMatrix pada halaman cari teman.



Gambar 48. Hasil GTMatrix Halaman Cari Teman

Dari hasil pengujian pada gambar 48 tersebut dapat terlihat bahwa halaman cari teman memiliki *page load time* adalah 1,49 detik, *total page size* adalah 356KB, *total number of request* adalah 31, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (85%).

c. Hasil Pengujian *Efficiency* Halaman *Following*

Gambar 49 berikut ini merupakan hasil pengujian dengan GTMatrix pada halaman *following*.



Gambar 49. Hasil GTMatrix Halaman *Following*

Dari hasil pengujian pada gambar 49 tersebut dapat terlihat bahwa halaman *following* memiliki *page load time* adalah 1,92 detik, *total page size* adalah 356KB, *total number of request* adalah 30, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (85%).

d. Hasil Pengujian *Efficiency* Halaman *Follower*

Gambar 50 merupakan hasil pengujian dengan GTMatrix pada halaman *follower*.

Performance Report for: <http://asis.pradana.web.id/follow/follower>

Report generated: Tuesday, April 8, 2014, 7:34 AM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.7



Page load time: 1.55s | Total page size: 356KB | Total number of requests: 31

Gambar 50. Hasil GTMatrix Halaman *Follower*

Dari hasil pengujian pada gambar 50 tersebut dapat terlihat bahwa halaman *follower* memiliki *page load time* adalah 1,55 detik, *total page size* adalah 356KB, *total number of request* adalah 31, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (85%).

e. Hasil Pengujian *Efficiency* Halaman Data Kelas

Gambar 51 berikut ini merupakan hasil pengujian dengan GTMatrix pada halaman data kelas.

Performance Report for: <http://asis.pradana.web.id/kelas>

Report generated: Tuesday, April 8, 2014, 7:43 AM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.7



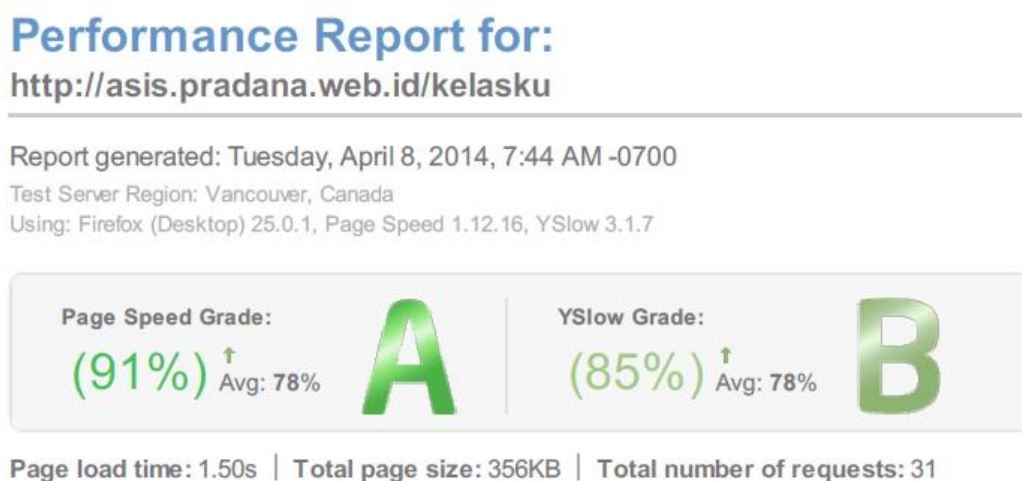
Page load time: 1.52s | Total page size: 385KB | Total number of requests: 31

Gambar 51. Hasil GTMatrix Halaman Data Kelas

Dari hasil pengujian pada gambar 51 tersebut dapat terlihat bahwa halaman data kelas memiliki *page load time* adalah 1,52 detik, *total page size* adalah 385KB, *total number of request* adalah 31, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (85%).

f. Hasil Pengujian *Efficiency* Halaman Kelasku

Gambar 52 berikut ini merupakan hasil pengujian dengan GTMatrix pada halaman kelasku.



Gambar 52. Hasil GTMatrix Halaman Kelasku

Dari hasil pengujian pada gambar 51 tersebut dapat terlihat bahwa halaman kelasku memiliki *page load time* adalah 1,50 detik, *total page size* adalah 356KB, *total number of request* adalah 31, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (85%).

g. Hasil Pengujian *Efficiency* Halaman Jadwal Akademik

Gambar 53 merupakan hasil pengujian dengan GTMatrix pada halaman jadwal akademik.

Performance Report for:

<http://asis.pradana.web.id/jadwal>

Report generated: Tuesday, April 8, 2014, 7:45 AM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.7

Page Speed Grade: (91%) ↑ Avg: 78%	A	YSlow Grade: (85%) ↑ Avg: 78%	B
---------------------------------------	---	----------------------------------	---

Page load time: 1.54s | Total page size: 373KB | Total number of requests: 31

Gambar 53. Hasil GTMatrix Halaman Jadwal Akademik

Dari hasil pengujian pada gambar 53 tersebut dapat terlihat bahwa halaman jadwal akademik memiliki *page load time* adalah 1,54 detik, *total page size* adalah 373KB, *total number of request* adalah 31, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (85%).

h. Hasil Pengujian *Efficiency* Halaman Waktu Kuliah

Gambar 54 berikut ini merupakan hasil pengujian dengan GTMatrix pada halaman waktu kuliah.

Performance Report for:

<http://asis.pradana.web.id/infoWaktu>

Report generated: Tuesday, April 8, 2014, 7:47 AM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.7

Page Speed Grade: (91%) ↑ Avg: 78%	A	YSlow Grade: (85%) ↑ Avg: 78%	B
---------------------------------------	---	----------------------------------	---

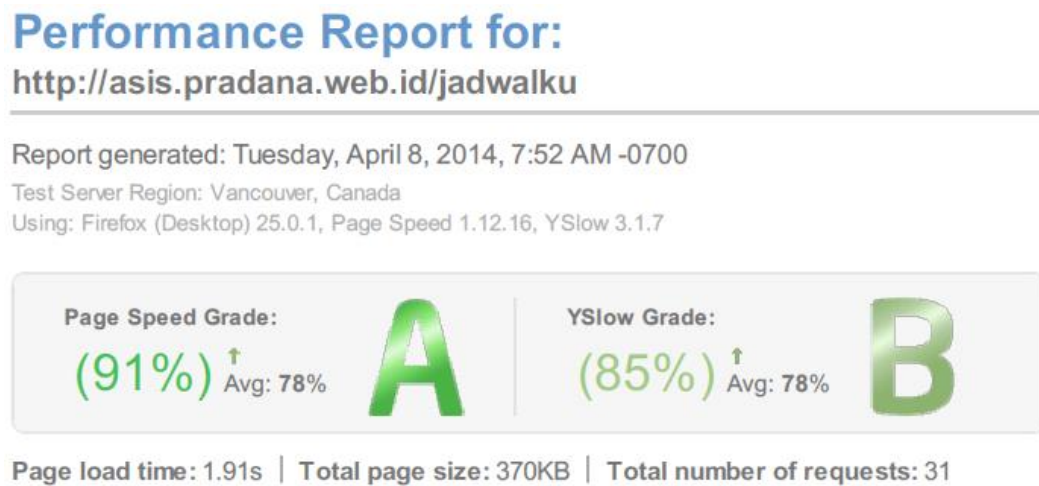
Page load time: 1.59s | Total page size: 371KB | Total number of requests: 31

Gambar 54. Hasil GTMatrix Halaman Waktu Kuliah

Dari hasil pengujian pada gambar 54 tersebut dapat terlihat bahwa halaman waktu kuliah memiliki *page load time* adalah 1,59 detik, *total page size* adalah 371KB, *total number of request* adalah 31, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (85%).

i. Hasil Pengujian *Efficiency* Halaman Jadwal Pribadi

Gambar 55 berikut ini merupakan hasil pengujian dengan GTMatrix pada halaman jadwal pribadi.



Gambar 55. Hasil GTMatrix Halaman Jadwal Pribadi

Dari hasil pengujian pada gambar 55 tersebut dapat terlihat bahwa halaman jadwal pribadi memiliki *page load time* adalah 1,91 detik, *total page size* adalah 370KB, *total number of request* adalah 31, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (85%).

j. Hasil Pengujian *Efficiency* Halaman Cari Jadwal

Gambar 56 merupakan hasil pengujian dengan GTMatrix pada halaman cari jadwal.

Performance Report for: <http://asis.pradana.web.id/jadwalku/cari>

Report generated: Tuesday, April 8, 2014, 7:53 AM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.7

Page Speed Grade: (91%) ↑ Avg: 78%	A	YSlow Grade: (83%) ↓ Avg: 78%	B
---------------------------------------	---	----------------------------------	---

Page load time: 2.18s | Total page size: 388KB | Total number of requests: 38

Gambar 56. Hasil GTMatrix Halaman Cari Jadwal

Dari hasil pengujian pada gambar 56 tersebut dapat terlihat bahwa halaman cari jadwal memiliki *page load time* adalah 2,18 detik, *total page size* adalah 388KB, *total number of request* adalah 38, *Page Speed Grade* adalah A (91%) dan *YSlow Grade* adalah B (83%).

k. Hasil Pengujian *Efficiency* Halaman Data Ruang

Gambar 57 merupakan hasil pengujian dengan GTMatrix pada halaman data ruang.

Performance Report for: <http://asis.pradana.web.id/ruang>

Report generated: Tuesday, April 8, 2014, 7:55 AM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.7

Page Speed Grade: (92%) ↑ Avg: 78%	A	YSlow Grade: (80%) ↓ Avg: 78%	B
---------------------------------------	---	----------------------------------	---

Page load time: 1.96s | Total page size: 474KB | Total number of requests: 40

Gambar 57. Hasil GTMatrix Halaman Data Ruang

Dari hasil pengujian pada gambar 57 tersebut dapat terlihat bahwa halaman data ruang memiliki *page load time* adalah 1,96 detik, *total page size* adalah 474KB, *total number of request* adalah 40, *Page Speed Grade* adalah A (92%) dan *YSlow Grade* adalah B (80%).

5. Maintainability

Tabel 19 berikut merupakan hasil dari Source Code SearchEngine dari sistem informasi jadwal akademik.

Tabel 19. Hasil Pengujian dengan Source Code SearchEngine

No	Modul	Code Lines (LOC)	Cyclomatic Complexity (g)	Halstead Volume (V)
1	Berita	31	7	483,2492
2	ContactForm	23	3	105,41563
3	Follow	59	13	1331,3685
4	Hari	28	7	387,61963
5	Jadwal	116	13	1962,422
6	Kelas	43	8	804
7	Kelasku	46	10	1017,46094
8	LoginForm	47	9	581,3172
9	Makul	28	7	355,89297
10	Notifikasi	39	8	774,04553
11	Ruang	103	14	912,644
12	Semester	42	7	355,89297
13	User	82	11	1912,8667
14	Waktu	44	7	425,31427
Rata - rata		52,214	8,857	814,965

6. Portability

Gambar 58 merupakan hasil pengujian BrowseEmAll terhadap sistem informasi jadwal akademik pada *web browser desktop*.

Tested: http://localhost/skripsi/	
No errors found	
Internet Explorer 11	0 Errors, 0 warning(s)
Internet Explorer 10	0 Errors, 0 warning(s)
Firefox 27	0 Errors, 0 warning(s)
Firefox 26	0 Errors, 0 warning(s)
Chrome 32	0 Errors, 0 warning(s)
Chrome 31	0 Errors, 0 warning(s)
Opera 19	0 Errors, 0 warning(s)
Opera 17	0 Errors, 0 warning(s)

Gambar 58. Hasil Pengujian *Portability* pada Perangkat *Desktop*

Gambar 59 berikut merupakan hasil pengujian BrowseEmAll terhadap sistem informasi jadwal akademik pada *web browser mobile*.

Tested: http://localhost/skripsi/	
No errors found	
iPad 4 (iOS 7)	0 Errors, 0 warning(s)
iPad 3 (iOS 7)	0 Errors, 0 warning(s)
iPhone 5S (iOS 7)	0 Errors, 0 warning(s)
iPhone 4S (iOS 7)	0 Errors, 0 warning(s)
Android 4.4	0 Errors, 0 warning(s)
Android 4.3	0 Errors, 0 warning(s)
Android 4.4 tablet	0 Errors, 0 warning(s)
Android 4.3 tablet	0 Errors, 0 warning(s)

Gambar 59. Hasil Pengujian *Portability* pada Perangkat *Mobile*

E. Pembahasan Hasil Penelitian

1. *Functionality*

Dari hasil pengujian *functionality* dengan menggunakan *test case* terhadap 3 orang yang ahli dibidangnya maka didapatkan hasil yang dapat dilihat pada tabel 20 berikut ini.

Tabel 20. Hasil Uji Fungsi *Software*

Penguji	Lolos	Gagal	Total Fungsi
1	52	0	52
2	52	0	52
3	52	0	52
Rata - Rata	52	0	52

Hasil tersebut menyatakan bahwa semua pengujian menyatakan bahwa semua fungsi dalam sistem dapat berjalan dengan baik. Hasil tersebut kemudian dihitung dengan persamaan ISO metric berikut ini.

$$X = 1 - \frac{A}{B}$$

$$X = 1 - \frac{0}{52}$$

$$X = 1 - 0$$

$$X = 1$$

Dari hasil tersebut didapatkan nilai *functionality* akhir yaitu 1 sehingga sesuai dengan ISO Metrics maka sistem informasi jadwal akademik memenuhi aspek *functionality* karena memiliki nilai *functionality* maksimal yaitu 1.

Sedangkan untuk aspek security dalam ISO 9126, berdasarkan hasil dari pengujian menggunakan *Acunetix Web Vulnerability Scanner* didapatkan hasil bahwa sistem informasi jadwal akademik berada pada Level 1 – Low atau memiliki kerentanan yang rendah (*Low Vulnerability*) sehingga dapat dikatakan bahwa

sistem informasi jadwal akademik memenuhi aspek *security* karena tingkat kerentanannya dibawah Level 2 - Medium. Tabel 21 berikut ini merupakan perbandingan hasil uji *functionality* dengan standar kualitas *software*.

Tabel 21. Perbandingan Hasil dengan Standar *Functionality*

Aspek Functionality	Hasil	Standar Kualitas	Kesimpulan
Fungsi	1	Lebih besar dari 0,5. Semakin mendekati 1 semakin baik	Memenuhi Standar
Keamanan	Level 1 - Low	Lebih rendah dari Level 2 - Medium	Memenuhi Standar

2. Reliability

Hasil dari pengujian dari *stress testing* dengan menggunakan WAPT kemudian diubah dalam bentuk persentase dalam tabel 22 berikut ini.

Tabel 22. Persentase Keberhasilan dalam *Stress Testing*

No	Nama	Jumlah Berhasil	Jumlah Gagal	Jumlah Skenario	Persentase Berhasil
1	Session	227	0	227	100%
2	Pages	2587	0	2587	100%
3	Hits	16519	0	16519	100%

Dari tabel 19 tersebut dapat disimpulkan bahwa persentase keberhasilan *session* adalah 100%, keberhasilan *pages* adalah 100% dan keberhasilan *hits* juga 100%. Jika dibandingkan dengan standar Telcordia maka sistem informasi jadwal akademik sudah memenuhi standar *reliability* karena tingkat keberhasilan telah diatas 95%.

3. Usability

Dari hasil pengujian reliabilitas kuesioner USE maka didapatkan nilai *Aplha Cronbach* sebesar 0,976. Hasil tersebut jika dibandingkan dengan standar *internal consistency* maka didapatkan bahwa kuesioner USE memiliki nilai *internal*

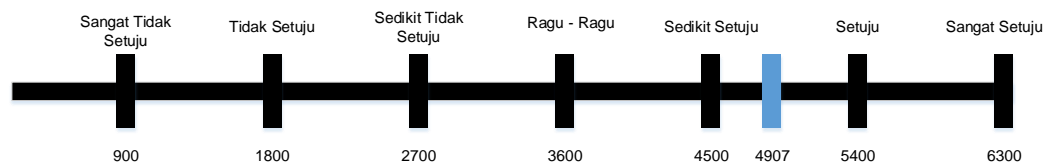
consistency sempurna (*excellent*) karena memiliki nilai *aplha cronbach* lebih dari 0,9.

Selain itu, data hasil jawaban kuesioner dari responden dapat juga dihitung dengan cara memberikan skoring pada setiap jawaban. Tabel 23 berikut merupakan hasil skoring jawaban kuesioner dari responden.

Tabel 23. Jumlah Skor Skala Likert

Jawaban	Skor Likert	Skor Maksimal	Jumlah Jawaban	Jumlah x Skor Likert
Sangat tidak setuju	1	900	1	1
Tidak setuju	2	1800	10	20
Sedikit tidak setuju	3	2700	35	105
Ragu – ragu	4	3600	130	520
Sedikit setuju	5	4500	232	1160
Setuju	6	5400	315	1890
Sangat setuju	7	6300	173	1211
Jumlah Total				4907

Jumlah skor maksimal dari seluruh item adalah $7 \times 30 \times 30 = 6300$. Sedangkan jumlah total dari data yang didapatkan pada tabel diatas adalah 4907 yang secara kontinum dapat digambarkan sesuai dengan gambar berikut ini.



Gambar 60. Kontinum Skor Jawaban

Sehingga dari data tersebut dapat diketahui persentase jawaban dari responden sebagai berikut.

$$\text{Persetujuan terhadap sistem} = \frac{\text{Jumlah skor yang diperoleh}}{\text{Jumlah skor maksimal}} \times 100\%$$

$$\text{Persentase persetujuan terhadap sistem} = \frac{4907}{6300} \times 100\%$$

$$\text{Persentase persetujuan terhadap sistem} = 77,89\%$$

Dari hasil tersebut dapat diketahui bahwa jawaban responden dikisaran setuju dan memiliki persentase persetujuan 77,89%. Jadi sistem informasi jadwal akademik memenuhi aspek *usability*.

4. Efficiency

Dari hasil pengujian sistem informasi jadwal akademik dengan menggunakan GTMatrix maka dapat dijelaskan lebih detail dalam tabel 24 berikut ini.

Tabel 24. Hasil Pengujian GTMatrix

No	Nama Halaman	Load Time (detik)	Page Size	Request	Page Speed		Yslow	
					Score	Grade	Score	Grade
1	Halaman Utama	1,19	352KB	29	92%	A	85%	B
2	Cari Teman	1,49	356KB	31	91%	A	85%	B
3	Following	1,92	356KB	30	91%	A	85%	B
4	Follower	1,55	356KB	31	91%	A	85%	B
5	Data Kelas	1,52	385KB	31	91%	A	85%	B
6	Kelasku	1,5	356KB	31	91%	A	85%	B
7	Jadwal Akademik	1,54	373KB	31	91%	A	85%	B
8	Waktu Kuliah	1,59	371KB	31	91%	A	85%	B
9	Jadwal Pribadi	1,61	370KB	31	91%	A	85%	B
10	Cari Jadwal	2,18	388KB	38	91%	A	83%	B
11	Data Ruang	1,96	474KB	40	92%	A	80%	B
Rata - Rata		1,64	376KB	32,18	91%	A	84%	B

Hasil dari pengujian tingkat kecepatan website dengan menggunakan GTMatrix didapatkan bahwa sistem informasi jadwal akademik memiliki *load time* rata-rata sebesar 1.64 detik sehingga jika dibandingkan dengan penelitian Subraya (2006) maka sistem informasi jadwal akademik memenuhi aspek

efficiency karena memiliki *load time* kurang dari 10 detik. Sedangkan jika dibandingkan dengan data Fortune (2010) maka sistem informasi jadwal akademik sudah di atas rata-rata *load time* website lokal yaitu kurang dari 7 detik.

5. Maintainability

Pada pengujian *maintainability* sistem informasi jadwal akademik dengan menggunakan Source Code SearchEngine didapatkan nilai berikut ini.

Rata-rata *Hastead Volume* (*aveV*) = 814,965

Rata-rata *Cyclomatic Complexity* (*aveV(g)*) = 8,857

Rata-rata *Line of Code* (*aveLOC*) = 52,214

Dari hasil tersebut kemudian dihitung dengan persamaan *Maintainability Index* (MI) yang dikemukakan Ganpati (2012) berikut ini.

$$MI = 171 - 5.2 * \ln(aveV) - 0.23 * aveV(g) - 16.2 * \ln(aveLOC)$$

$$MI = 171 - 5.2 * \ln(814,965) - 0.23 * 8,857 - 16.2 * \ln(52,214)$$

$$MI = 70,030$$

Hasil perhitungan tersebut menghasilkan nilai *Maintainability Index* (MI) sebesar 70,030. Nilai tersebut dibandingkan dengan standar nilai *Maintability Index* (MI) yang dikemukakan oleh Subraya (2006) maka didapatkan bahwa sistem informasi jadwal akademik layak atau sedang dalam kemudahan perawatan (*Moderate Maintainable*) karena memiliki nilai MI diatas 65 dan diantara 66 – 85. Tabel 25 berikut ini merupakan perbandingan hasil dengan standar *maintainability*.

Tabel 25. Perbandingan Hasil dengan Standar Kualitas *Maintainability*

Aspek	Hasil	Standar Kualitas	Kesimpulan
<i>Maintainability Index</i>	70,03	Lebih besar dari 65	Memenuhi standar

6. Portability

Pengujian portability dilakukan menjalankan *software* berbasis web di web browser dengan *software* BrowseEmAll. Web browser yang digunakan tersebut merupakan pilihan yang berikan oleh BrowseEmAll. *Software* BrowseEmAll bekerja dengan menguji *software* berbasis web dengan menggunakan *web browser* yang paling banyak digunakan pada saat ini baik pada *web browser* perangkat *desktop* maupun *web browser* perangkat *mobile*. Berdasarkan hasil pengujian yang dilakukan dengan menggunakan *software* BrowseEmAll maka dapat diambil kesimpulan sesuai dengan tabel 26 dibawah ini.

Tabel 26. Tabel Hasil Pengujian *Portability*

No	Web Browser	Platform	Hasil
1	Internet Explorer 11	Desktop	Sukses
2	Internet Explorer 10	Dekstop	Sukses
3	Mozilla Firefox 27	Dekstop	Sukses
4	Mozilla Firefox 26	Desktop	Sukses
5	Google Chrome 32	Desktop	Sukses
6	Google Chrome 31	Desktop	Sukses
7	Opera 19	Desktop	Sukses
8	Opera 17	Desktop	Sukses
9	iPad 4 (iOS 7)	Mobile	Sukses
10	iPad 3 (iOS 7)	Mobile	Sukses
11	iPhone 5S (iOS 7)	Mobile	Sukses
12	iPhone 4S (iOS 7)	Mobile	Sukses
13	Android 4.4	Mobile	Sukses
14	Android 4.3	Mobile	Sukses
15	Android 4.4 tablet	Mobile	Sukses
16	Android 4.3 tablet	Mobile	Sukses

Hasil pengujian *portability* sistem informasi jadwal akademik dengan menggunakan BrowseEmAll tersebut menyatakan bahwa sistem informasi jadwal akademik berjalan sukses di 8 *web browser desktop* dan 8 *web browser mobile* tanpa terjadi kesalahan (*error*) pada masing-masing *web browser*. Hal tersebut berarti bahwa sistem informasi jadwal akademik memenuhi standar kualitas pada aspek *portability* karena sesuai dengan penelitian Salonen (2012) jika *software* dapat dijalankan minimal menggunakan 7 *web browser desktop* dan 5 *web browser mobile*. Tabel 27 berikut ini merupakan perbandingan dari hasil pengujian *portability* menggunakan *software* BrowseEmAll dengan standar kualitas pada aspek *portability* dari hasil penelitian Salonen (2012).

Tabel 27. Perbandingan Hasil Pengujian dengan Standar *Portability*

Jenis Browser	Hasil	Standar	Kesimpulan
<i>Desktop</i>	Berjalan sukses di 8 <i>web browser desktop</i>	Minimal berjalan sukses di 7 <i>web browser desktop</i>	Memenuhi standar kualitas
<i>Mobile</i>	Berjalan sukses di 8 <i>web browser mobile</i>	Minimal berjalan sukses di 5 <i>web browser mobile</i>	Memenuhi standar kualitas

BAB V

SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan penelitian analisis dan pengembangan sistem informasi jadwal akademik berbasis *Yii Framework* di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta maka dapat diambil kesimpulan berikut ini.

1. Penelitian ini menghasilkan sistem informasi jadwal akademik Jurusan Pendidikan Teknik Elektronika UNY berbasis *Yii Framework* yang mempermudah pengolahan informasi jadwal akademik yang meliputi penambahan jadwal, pengubahan jadwal, penghapusan jadwal dan pencarian jadwal maupun ruangan kuliah. Pengembangan *software* dilakukan dengan beberapa tahapan. Tahapan pertama pengembangan perangkat lunak adalah analisis kebutuhan. Tahapan kedua adalah desain yang mencakup pemodelan UML, ERD dan desain *user interface*. Tahap ketiga adalah implementasi yaitu dengan penulisan kode program menggunakan bahasa pemrograman utama PHP dengan menggunakan *Yii Framework*. Tahap keempat adalah pengujian kualitas *software* sesuai dengan standar ISO 9126. Proses pengembangan *software* dapat berjalan dengan cepat karena adanya fitur *software reusable* dalam *Yii Framework* sehingga dapat menggunakan kode yang sudah ada sebelumnya dalam bentuk *extensions* dan modul.
2. *Software* sistem informasi jadwal akademik telah memenuhi standar kualitas *software* berdasarkan ISO 9126 pada aspek *functionality*, *reliability*, *usability*, *efficiency*, *maintainability* dan *portability*. Kesimpulan tersebut didukung dengan

hasil pengujian setiap aspek yaitu pengujian *functionality* didapatkan bahwa 100% fungsi *software* dapat berfungsi dan *low vulnerability* (rendah kerentanan) yang berarti telah memenuhi aspek *functionality*. Pengujian *reliability* dengan WAPT didapatkan nilai 100% yang berarti telah memenuhi aspek *reliability*. Pengujian *usability* dengan didapatkan nilai reliabilitas sebesar 0,976 yang berarti memiliki nilai *internal consistency* sempurna (*excellent*) dan persentase jawaban setuju sebesar 77,89% sehingga telah memenuhi aspek *usability*. Pengujian *efficiency* didapatkan peringkat (*grade*) A untuk Page Speed dan B untuk YSlow dan *page load time* rata-rata membutuhkan 1,64 detik sehingga telah memenuhi aspek *efficiency*. Pengujian *maintainability* menghasilkan nilai *Maintainability Index* (MI) sebesar 70,03 yang berarti bahwa *software* layak dalam kemudahan perawatan (*Moderate Maintable*) sehingga memenuhi telah memenuhi aspek *maintainability*. Pengujian *portability* didapatkan bahwa *software* dapat berjalan dengan lancar pada 8 *web browser* desktop dan 8 *web browser* mobile tanpa terdapat *error* sehingga telah memenuhi aspek *portability*.

B. Keterbatasan Produk

Sistem informasi jadwal akademik merupakan *software* berbasis web yang memberikan informasi jadwal sehingga mempermudah dilakukan perubahan maupun penambahan maupun pencarian jadwal. Keterbatasan dari sistem ini adalah tidak terdapatnya fitur membuat jadwal secara otomatis. Sehingga masih diperlukan *software* lain untuk melakukan pembuatan jadwal secara otomatis berdasarkan tingkat prioritas, penggunaan waktu maupun penggunaan ruang. Selain itu, sistem informasi ini masih berbasis web sehingga untuk mengetahui informasi jadwal, pengguna (*user*) harus mengakses website sistem informasi jadwal menggunakan perangkat desktop.

C. Pengembangan Produk Lebih Lanjut

Software sistem informasi jadwal akademik dapat dikembangkan lebih lanjut dengan menambahkan fitur untuk membuat jadwal secara otomatis dengan menggunakan algoritma genetika. Dengan adanya fitur tersebut jadwal kegiatan kuliah dapat dengan cepat dibuat dan disebarluaskan. Selain itu, sistem ini juga dapat dikembangkan dalam versi mobile. Dengan adanya sistem informasi dalam bentuk aplikasi *mobile* maka informasi jadwal dapat lebih cepat diterima maupun diakses oleh pengguna (*user*).

D. Saran

Dalam penelitian ini masih terdapat berbagai macam kekurangan sehingga masih perlu dikembangkan lagi. Terdapat berbagai saran dari peneliti untuk mengembangkan penelitian ini diantaranya adalah

1. Sistem informasi jadwal akademik dapat ditambahkan fitur pembuatan jadwal secara otomatis dengan algoritma genetika dan dikembangkan agar dapat diakses pada perangkat mobile.
2. Teknik pengujian kualitas perangkat lunak yang lebih kompleks dan dengan berbagai macam metode.

DAFTAR PUSTAKA

- Alhir, Sinan Si. (2003). *Learning UML*. Sebastopol, CA: O'Reilly Media, Inc.
- Andhika Lady Maharsi. (2013). Sistem Penjadwalan Mata Pelajaran Sekolah Menggunakan Algoritma Genetika. *Skripsi*. Program Studi Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta.
- Asthana, A., Olivieri, J. (2009). Quantifying Software Reliability and Readiness. *Communications Quality and Reliability, 2009*. CQR 2009. IEEE International Workshop Technical Committee on , vol., no., pp.1,6, 12-14 May 2009
- Brown, P.J. (2003). *Encyclopedia of Computer Science 4th Edition*. Chichester: John Wiley and Sons Ltd.
- Butzon, Toby. (2002). *PHP By Example*. Indiana: Que
- Chua, B.B. & Dyson, L.E. (2004). Applying the ISO9126 model to the evaluation of an elearning system. In R. Atkinson, C. McBeath, D. Jonas-Dwyer & R. Phillips (Eds). *Proceedings. Beyond the comfort zone: Proceedings of the 21st. ASCILITE Conference* (pp. 184 -190). Perth, 5-8 December. <http://www.ascilite.org.au/conferences/perth04/procs/chua.html>
- Cote, Marc-Alexis. (2006). *Software Quality Model Requirements for Software Quality Engineering*. http://profs.etsmtl.ca/wsurn/research/SQE-Publ/Quality%20model_requirements.%20SQM2006.pdf
- Dougherty, Chad. (2012). *Practical Identification of SQL Injection Vulnerabilities*. Carnegie Mellon University.
- Doupe, Adam., Cova, Marco., & Vigna, Giovanni. (2010). Why Johnny can't pentest: an analysis of black-box web vulnerability scanners. *Proceedings. DIMVA'10 Proceedings of the 7th international conference on Detection of intrusions and malware, and vulnerability assessment*. Halaman 111-131
- Fahmy, Syahrul. 2012. Evaluating the Quality of Software in e-Book using the ISO 9126 Model. *International Journal of Control and Automation* (Vol 5, No 2 June 2012).Halaman 115-122.
- Fajar Kurniawan. (2012). Sistem Penyusunan Jadwal Pelajaran Berbasis Web. *E-Jurnal Skripsi*. Program Studi Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta.
- Ganpati, Anita., Kalia, Arvind., Singh, Hardeep. (2012). Maintainability Index over Multiple Releases: A Case Study PHP Open Source Software. *International Journal of Engineering Research & Technology (IJERT)* (Vol 1 issue 6). Halaman 1 - 3.

- George, D., & Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference. 11.0 update (4th ed.)*. Boston: Allyn & Bacon
- Gorman, Jason. (2006). *Metric Definition*. <http://www.codemanship.co.uk/parlezuml/metrics/Metrics%20Definitions.pdf>
- Halfond, William G.J., Jeremy Viegas, Alessandro Osro. (2006). A Classification of SQL Injection Attacks and Countermeasures. *Proceedings of the IEEE International Symposium on Secure Software Engineering (ISSSE 2006)*.
- Hamilton, Kim., Miles, Russell. (2006). *Learning UML 2.0*. Sebastopol, CA: O'Reilly Media, Inc.
- Haverbeke, Marijn. (2011). *Eloquent JavaScript: A Modern Introduction to Programming*. San Francisco: No Starch Press, Inc.
- Heitlager, Ilja. (2007). A Practical Model for Measuring Maintainability. *Quality of Information and Communications Technology, 2007. QUATIC 2007. 6th International Conference*. Halaman 30 – 39.
- Huang, Yao-Wen, Shih-Kun Huang & Tsung-Po Lin. (2003). Web Application Security Assessment by Fault Injection and Behavior Monitoring. *Association for Computing Machinery (ACM)* 1-58113-680-3/03/0005. Halaman 148-159
- ISO. (2003). *ISO/IEC TR 9126-2: Software engineering-product quality-part 2: External metrics*. International Organization for Standardization, Geneva, Switzerland.
- Jenkins, Sue. (2009). *Web Design All In One For Dummies*. Indiana: Wiley Publishing, Inc.
- Joo, Soohyung., Lin, Suyu., Lu, Kun. (2011). A Usability Evaluation Model for Academic Library Websites: Efficiency, Effectiveness and Learnability. *Journal of Library and Information Studies*. (Volume 9 number 2, December 2011). Halaman 11 – 26.
- Larsen, Rob. (2013). *Beginning HTML & CSS*. Indianapolis: John Wiley & Sons, Inc.
- Laudon, Kenneth C. & Laudon, Jane P. (2012). *Management Information Systems: Managing The Digital Firm 12th Edition*. New Jersey: Pearson Education Inc.
- Loka Dwiartara. (2010). *Menyelam dan Menaklukan Samudra PHP*. Bogor: IW Ilmuwebsite
- Lund, Arnold M. (2001). *Measuring Usability with the USE Questionnaire*. STC Usability SIG Newsletter Usability Interface October 2001 issue (Vol 8, No. 2)

- Maheshwari, Shikha., Jain, Denish Ch. (2012). Comparative Analysis of Different types of Models in Software Development Life Cycle. *International Journal of Advanced Research in Computer Science and Software Engineering*. (Volume 2, Issue 5, May 2012). Halaman 285 – 290.
- Mulugeta, Mesfin. (2008). *Performance Testing and Optimization in Web-Service Based Applications*. Blackboard Developer Convergence.
- Munassar, N. M. A., & Govardhan, A. (2010). A Comparison Between Five Models of Software Engineering. *International Journal of Computer Science Issues (IJCSI)*, 7(5).
- Nielsen, Jakob. (2012). *How Many Test Users in a Usability Study?*. Nielsen Norman Group. <http://www.nngroup.com/articles/how-many-test-users/>
- Olsina, L., Covella, G., & Rossi, G. (2006). *Web quality*. Web Engineering (pp. 109-142). Springer Berlin Heidelberg.
- Rainer, R. Kelly & Cegielski, Casey G. (2011). *Introduction to Information Systems: Supporting and Transforming Business 3rd Edition*. Hoboken: John Wiley & Sons, Inc.
- Rina. & Tyagi, Sanjay. (2013). A Comparative Study of Performance Testing Tools. *International Journal of Advanced Research in Computer Science and Software Engineering*. (Volume 3, Issue 5, May 2013). Halaman 1300 – 1307.
- Robbins, Jennifer Niederst . (2007). *Learning Web Design, Third Edition*. Sebastopol: O'Reilly Media, Inc.
- Salonen, Ville. (2012). Automatic Portability Testing. *Master's Thesis*. Information Technology, Department of Mathematical Information Technology, University of Jyväskylä.
- Sam'ani. (2012). Rancang Bangun Sistem Penjadwalan Perkuliahan dan Ujian Akhir Semester dengan Pendekatan Algoritma Genetika. *Thesis*. Program Pasca Sarjana Universitas Negeri Yogyakarta.
- Seffah, Ahmed., Donyaee, Mohammad. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Journal*. (Volume 16). Halaman 159 - 178.
- Sigit Suryadi. (2008). Perbedaan Insomnia pada Mahasiswa yang Sedang Mengerjakan Skripsi dan Belum Mengerjakan Skripsi. *Skripsi*. Fakultas Muhammadiyah Surakarta. <http://v2.eprints.ums.ac.id/archive/etd/847/2/1>
- Spett, Kevin. (2005). *Cross-Site Scripting: Are your web applications vulnerable?*. SPI Dynamics, Inc
- Subraya, B.M. (2006). *Integrated Approach to Web Performance Testing: A Practitioner's Guide*. London: IRM Press.

- Sugiyono. (2012). *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta
- Vagias, Wade M. (2006). *Likert-type scale response anchors*. Clemson International Institute for Tourism & Research Development, Department of Parks, Recreation and Tourism Management. Clemson University.
- Watson, Arthur H., Thomas J. McCabe. (1996). *Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric*. Gaithersburg: National Institute of Standards and Technology.
- Winesett, Jeffrey. (2012). *Web Application Development with Yii and PHP, Second Edition*. Birmingham : Packt Publishing Minneapolis, Minnesota
- York, Richard. (2009). *Beginning JavaScript and CSS Development with jQuery*. Indiana: Wiley Publishing, Inc

LAMPIRAN

Lampiran 1. Surat Keputusan Pembimbing

**9KEPUTUSAN DEKAN
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
NOMOR : 226/ELK/Q-I/XI/2013
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	: Handaru Jati, Ph.D
Bagi mahasiswa	:
Nama/No. Mahasiswa	: Pradana Setialana / 010520244004
Jurusan/ Prodi	: Pendidikan Teknik Elektronika / Pendidikan Teknik Informatika
Judul Skripsi	: <i>Analisis Pengembangan Sistem Informasi Penjadwalan Akademik dan Petunjuk Lokasi Dosen Di Lingkungan Universitas Negeri Yogyakarta (UNY)</i>

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.

Ditetapkan : di Yogyakarta
Pada tanggal : 7 November 2013
Dekan

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

Tembusan Yth :

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

Lampiran 2. Surat Permohonan Ijin Observasi



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI YOGYAKARTA
FAKULTAS TEKNIK**

Alamat : Kampus Karangmalang, Yogyakarta, 55281
Telp. (0274) 586168 psw. 276,289,292 (0274) 586734 Fax. (0274) 586734
website : <http://ft.uny.ac.id> e-mail: ft@uny.ac.id ; teknik@uny.ac.id



Certificate No. QSC 00592

Nomor : 4209/UN34.15/PL/ 2013

19 Desember 2013

Hal : Permohonan Ijin Observasi/Survey--
Lamp. :

Yth. Pimpinan /Direktur /Kepala /Ketua *) : Universitas Negeri Yogyakarta
JL. COLOMBO NO.1, KARANGMALANG, DEPOK, SLEMAN
YOGYAKARTA

Dalam rangka pelaksanaan Mata Kuliah Tugas Akhir Skripsi, kami mohon dengan hormat bantuan Saudara memberikan ijin untuk melaksanakan observasi/Survey dengan fokus permasalahan " Analisis Pengembangan Sistem Informasi Penjadwalan Akademik dan Petunjuk Lokasi Dosen di Lingkungan Universitas Negeri Yogyakarta (UNY) ", bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta sebagai berikut:

No.	Nama	NIM	Jurusan/Program Studi
1	Pradana Setialana	10520244004	Pend. Teknik Informatika - S1

Dosen Pembimbing/Dosen Pengampu:

Nama : Handaru Jati, Ph.D.

NIP : 19740511 199903 1 002

Demikian permohonan kami, atas bantuan dan kerjasama yang baik selama ini, kami mengucapkan terima kasih.

Dekan,
Wakil Dekan I,

Dr. Suharyo Soenarto
NIP. 19580630 198601 1 0014

Tembusan:
Ketua Jurusan

*) Coret yang tidak perlu
10520244004 No. 2114

Lampiran 3. Surat Permohonan Ijin Penelitian



PEMERINTAH DAERAH DAERAH ISTIMEWA YOGYAKARTA
SEKRETARIAT DAERAH
 Kompleks Kepatihan, Danurejan, Telepon (0274) 562811 - 562814 (Hunting)
 YOGYAKARTA 55213

operator1@yahoo.com

SURAT KETERANGAN / IJIN
 070/REG/562/3/2014

Membaca Surat : **WAKIL DEKAN I FAKULTAS TEKNIK** Nomor : **--/UN34.15/PL/--**
UNIVERSITAS NEGERI YOGYAKARTA
 Tanggal : **13 MARET 2014** Perihal : **IJIN PENELITIAN/RISET**

- Mengingat :
1. Peraturan Pemerintah Nomor 41 Tahun 2006, tentang Perizinan bagi Perguruan Tinggi Asing, Lembaga Penelitian dan Pengembangan Asing, Badan Usaha Asing dan Orang Asing dalam melakukan Kegiatan Penelitian dan Pengembangan di Indonesia;
 2. Peraturan Menteri Dalam Negeri Nomor 20 Tahun 2011, tentang Pedoman Penelitian dan Pengembangan di Lingkungan Kementerian Dalam Negeri dan Pemerintah Daerah;
 3. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor 37 Tahun 2008, tentang Rincian Tugas dan Fungsi Satuan Organisasi di Lingkungan Sekretariat Daerah dan Sekretariat Dewan Perwakilan Rakyat Daerah.
 4. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor 18 Tahun 2009 tentang Pedoman Pelayanan Perizinan, Rekomendasi Pelaksanaan Survei, Penelitian, Pendataan, Pengembangan, Pengkajian, dan Studi Lapangan di Daerah Istimewa Yogyakarta.

DIIJINKAN untuk melakukan kegiatan survei/penelitian/pendataan/pengembangan/pengkajian/studi lapangan kepada:

Nama : **PRADANA SETIALANA** NIP/NIM : **10520244004**
 Alamat : **FAKULTAS TEKNIK, PENDIDIKAN TEKNIK ELEKTRONIKA, UNIVERSITAS NEGERI YOGYAKARTA**
 Judul : **ANALISIS DAN PENGEMBANGAN SISTEM INFORMASI JADWAL AKADEMIK BERBASIS YII FRAMEWORK DI JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**
 Lokasi : **UNIVERSITAS NEGERI YOGYAKARTA**
 Waktu : **20 MARET 2014 s/d 20 JUNI 2014**

Dengan Ketentuan

1. Menyerahkan surat keterangan/ijin survei/penelitian/pendataan/pengembangan/pengkajian/studi lapangan *) dari Pemerintah Daerah DIY kepada Bupati/Walikota melalui institusi yang berwenang mengeluarkan ijin dimaksud;
2. Menyerahkan soft copy hasil penelitiannya baik kepada Gubernur Daerah Istimewa Yogyakarta melalui Biro Administrasi Pembangunan Setda DIY dalam compact disk (CD) maupun mengunggah (upload) melalui website adbang.jogjapro.go.id dan menunjukkan cetakan asli yang sudah disahkan dan dibubuhi cap institusi;
3. Ijin ini hanya dipergunakan untuk keperluan ilmiah, dan pemegang ijin wajib mentaati ketentuan yang berlaku di lokasi kegiatan;
4. Ijin penelitian dapat diperpanjang maksimal 2 (dua) kali dengan menunjukkan surat ini kembali sebelum berakhir waktunya setelah mengajukan perpanjangan melalui website adbang.jogjapro.go.id;
5. Ijin yang diberikan dapat dibatalkan sewaktu-waktu apabila pemegang ijin ini tidak memenuhi ketentuan yang berlaku.

Dikeluarkan di Yogyakarta
 Pada tanggal **20 MARET 2014**
 An Sekretaris Daerah
 Asisten Perekonomian dan Pembangunan
 Kepala Biro Administrasi Pembangunan



Tembusan

1. GUBERNUR DAERAH ISTIMEWA YOGYAKARTA (SEBAGAI LAPORAN)
2. REKTOR UNIVERSITAS NEGERI YOGYAKARTA
3. WAKIL DEKAN I FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA
4. YANG BERSANGKUTAN

Notice: Undefined variable: rsmstembusan in /var/www/izin/application/modules/pzn/controllers/izinController.php on line 180

Lampiran 4. Instrumen *Functionality*

No	Fungsi	Pernyataan	Lolos	
			Ya	Tidak
User				
1	Login	Fungsi <i>login</i> sudah berfungsi dengan baik		
2	Register	Fungsi <i>register</i> sudah berfungsi dengan baik		
3	Lihat <i>profile</i>	Fungsi lihat <i>profile</i> sudah berfungsi dengan baik		
4	Ubah <i>profile</i>	Fungsi ubah data <i>profile</i> sudah berfungsi dengan baik		
5	Akses <i>user</i>	Fungsi manajemen akses <i>user</i> sudah berfungsi dengan baik		
6	Import data <i>user</i>	Fungsi import data <i>user</i> dari <i>file Excel</i> (xls) sudah berfungsi dengan baik		
Berita				
7	Tambah berita	Fungsi tambah berita sudah berfungsi dengan baik		
8	Ubah berita	Fungsi ubah berita sudah berfungsi dengan baik		
9	Hapus berita	Fungsi hapus berita sudah berfungsi dengan baik		
Teman				
10	Lihat daftar <i>user</i>	Fungsi lihat daftar semua teman sudah berfungsi dengan baik		
11	Cari teman	Fungsi cari teman sudah berfungsi dengan baik		
12	Follow teman	Fungsi <i>follow</i> teman sudah berfungsi dengan baik		
13	Unfollow Teman	Fungsi <i>unfollow</i> teman sudah berfungsi dengan baik		
14	Lihat <i>following</i>	Fungsi lihat teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik		
15	Lihat <i>follower</i>	Fungsi lihat teman yang mengikuti (<i>follower</i>) sudah berfungsi dengan baik		
16	Ubah notifikasi <i>following</i>	Fungsi ubah notifikasi (notifikasi aktif/nonaktif) dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik		
Kelas				
17	Lihat daftar kelas	Fungsi melihat semua daftar kelas sudah berfungsi dengan baik		

18	Cari Kelas	Fungsi mencari kelas sudah berfungsi dengan baik		
19	Join Kelas	Fungsi mengikuti kelas (<i>join</i>) sudah berfungsi dengan baik		
20	Unjoin Kelas	Fungsi keluar dari kelas yang diikuti (<i>unjoin</i>) sudah berfungsi dengan baik		
21	Tambah kelas	Fungsi tambah data kelas sudah berfungsi dengan baik		
22	Ubah kelas	Fungsi ubah data kelas sudah berfungsi dengan baik		
23	Hapus kelas	Fungsi hapus data kelas sudah berfungsi dengan baik		
24	Data kelas yang diikuti	Fungsi melihat daftar kelas yang diikuti sudah berfungsi dengan baik		
Jadwal				
25	Lihat data jadwal akademik	Fungsi lihat data jadwal akademik sudah berfungsi dengan baik		
26	Cari data jadwal akademik	Fungsi mencari data jadwal akademik sudah berfungsi dengan baik		
27	Tambah jadwal akademik	Fungsi menambah jadwal akademik sudah berfungsi dengan baik		
28	Ubah jadwal akademik	Fungsi mengubah jadwal akademik sudah berfungsi dengan baik		
29	Hapus jadwal akademik	Fungsi hapus jadwal akademik sudah berfungsi dengan baik		
30	Lihat jadwal pribadi	Fungsi lihat data jadwal pribadi sudah berfungsi dengan baik		
31	Tambah jadwal pribadi	Fungsi tambah jadwal pribadi sudah berfungsi dengan baik		
32	Ubah jadwal pribadi	Fungsi ubah data jadwal pribadi sudah berfungsi dengan baik		
33	Hapus jadwal pribadi	Fungsi hapus data jadwal pribadi sudah berfungsi dengan baik		
34	Lihat jadwal saat ini	Fungsi lihat data jadwal yang berlangsung hari ini sudah berfungsi dengan baik		
35	Cari jadwal	Fungsi cari jadwal sudah berfungsi dengan baik		
36	Lihat lokasi pelaksanaan jadwal	Fungsi lihat lokasi ruangan dari jadwal yang berlangsung hari ini sudah berfungsi dengan baik		

Ruang				
37	Lihat daftar ruangan	Fungsi lihat daftar ruangan sudah berfungsi dengan baik		
38	Lihat ruangan di peta	Fungsi lihat lokasi ruangan di peta sudah berfungsi dengan baik		
39	Cari ruangan	Fungsi cari ruangan sudah berfungsi dengan baik		
40	Cari ruang kosong	Fungsi cari ruangan yang tidak digunakan sudah berfungsi dengan baik		
41	Tambah ruangan	Fungsi tambah data ruangan sudah berfungsi dengan baik		
42	Ubah ruangan	Fungsi ubah data ruangan sudah berfungsi dengan baik		
43	Hapus ruangan	Fungsi hapus data ruangan sudah berfungsi dengan baik		
Notifikasi				
44	Notifikasi tambah jadwal dari <i>following</i>	Fungsi notifikasi adanya penambahan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik		
45	Notifikasi pengubahan jadwal <i>following</i>	Fungsi notifikasi adanya pengubahan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik		
46	Notifikasi hapus jadwal dari <i>following</i>	Fungsi notifikasi adanya penghapusan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik		
47	Notifikasi follower baru	Fungsi notifikasi adanya teman baru yang mengikuti (<i>follower</i>) sudah berfungsi dengan baik		
48	Notifikasi tambah jadwal dari kelas	Fungsi notifikasi adanya penambahan jadwal dari kelas yang diikuti (<i>join</i>) sudah berfungsi dengan baik		
49	Notifikasi ubah jadwal dari kelas	Fungsi notifikasi adanya pengubahan jadwal dari kelas yang diikuti (<i>join</i>) sudah berfungsi dengan baik		
50	Lihat semua notifikasi	Fungsi lihat semua notifikasi baik yang sudah dibaca maupun belum, sudah berfungsi dengan baik		
51	Notifikasi jadwal hari ini	Fungsi notifikasi jadwal yang berlangsung hari ini sudah berfungsi dengan baik		
52	Lihat lokasi pelaksanaan jadwal	Fungsi lihat lokasi ruangan dari jadwal yang berlangsung hari ini sudah berfungsi dengan baik		

Lampiran 5. Hasil Pengujian *Functionality*

INSTRUMEN FUNCTIONALITY

Analisis dan Pengembangan Sistem Informasi Jadwal Akademik Berbasis Yii Framework di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta

A. Identitas Responden

Nama : YANUAR ARIFIN
Pekerjaan : WEB DEVELOPER

B. Petunjuk Umum

1. Sebelum mengisi angket ini, pastikan Anda sudah menggunakan Sistem Informasi Jadwal Akademik.
2. Isilah data identitas responden yaitu nama dan pekerjaan.
3. Isilah jawaban sesuai dengan pernyataan dengan cara **memberikan tanda cek (✓)**

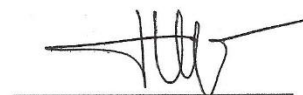
C. Pernyataan

No	Fungsi	Pernyataan	Lolos	
			Ya	Tidak
User				
1	Login	Fungsi login sudah berfungsi dengan baik	✓	
2	Register	Fungsi register sudah berfungsi dengan baik	✓	
3	Lihat profile	Fungsi lihat profile sudah berfungsi dengan baik	✓	
4	Ubah profile	Fungsi ubah data profile sudah berfungsi dengan baik	✓	
5	Akses user	Fungsi manajemen akses user sudah berfungsi dengan baik	✓	
6	Import data user	Fungsi import data user dari <i>file Excel (xls)</i> sudah berfungsi dengan baik	✓	
Berita				
7	Tambah berita	Fungsi tambah berita sudah berfungsi dengan baik	✓	
8	Ubah berita	Fungsi ubah berita sudah berfungsi dengan baik	✓	
9	Hapus berita	Fungsi hapus berita sudah berfungsi dengan baik	✓	
Teman				
10	Lihat daftar user	Fungsi lihat daftar semua teman sudah berfungsi dengan baik	✓	
11	Cari teman	Fungsi cari teman sudah berfungsi dengan baik	✓	
12	Follow teman	Fungsi <i>follow</i> teman sudah berfungsi dengan baik	✓	
13	Unfollow Teman	Fungsi <i>unfollow</i> teman sudah berfungsi dengan baik	✓	
14	Lihat following	Fungsi lihat teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	✓	
15	Lihat follower	Fungsi lihat teman yang mengikuti (<i>follower</i>) sudah berfungsi dengan baik	✓	

16	Ubah notifikasi following	Fungsi ubah notifikasi (notifikasi aktif/nonaktif) dari teman yang diikuti (following) sudah berfungsi dengan baik	✓	
Kelas				
17	Lihat daftar kelas	Fungsi melihat semua daftar kelas sudah berfungsi dengan baik	✓	
18	Cari Kelas	Fungsi mencari kelas sudah berfungsi dengan baik	✓	
19	Join Kelas	Fungsi mengikuti kelas (<i>join</i>) sudah berfungsi dengan baik	✓	
20	Unjoin Kelas	Fungsi keluar dari kelas yang diikuti (<i>unjoin</i>) sudah berfungsi dengan baik	✓	
21	Tambah kelas	Fungsi tambah data kelas sudah berfungsi dengan baik	✓	
22	Ubah kelas	Fungsi ubah data kelas sudah berfungsi dengan baik	✓	
23	Hapus kelas	Fungsi hapus data kelas sudah berfungsi dengan baik	✓	
24	Data kelas yang diikuti	Fungsi melihat daftar kelas yang diikuti sudah berfungsi dengan baik	✓	
Jadwal				
25	Lihat data jadwal akademik	Fungsi lihat data jadwal akademik sudah berfungsi dengan baik	✓	
26	Cari data jadwal akademik	Fungsi mencari data jadwal akademik sudah berfungsi dengan baik	✓	
27	Tambah jadwal akademik	Fungsi menambah jadwal akademik sudah berfungsi dengan baik	✓	
28	Ubah jadwal akademik	Fungsi mengubah jadwal akademik sudah berfungsi dengan baik	✓	
29	Hapus jadwal akademik	Fungsi hapus jadwal akademik sudah berfungsi dengan baik	✓	
30	Lihat jadwal pribadi	Fungsi lihat data jadwal pribadi sudah berfungsi dengan baik	✓	
31	Tambah jadwal pribadi	Fungsi tambah jadwal pribadi sudah berfungsi dengan baik	✓	
32	Ubah jadwal pribadi	Fungsi ubah data jadwal pribadi sudah berfungsi dengan baik	✓	
33	Hapus jadwal pribadi	Fungsi hapus data jadwal pribadi sudah berfungsi dengan baik	✓	
34	Lihat jadwal saat ini	Fungsi lihat data jadwal yang berlangsung hari ini sudah berfungsi dengan baik	✓	
35	Cari jadwal	Fungsi cari jadwal sudah berfungsi dengan baik	✓	
36	Lihat lokasi pelaksanaan jadwal	Fungsi lihat lokasi ruangan dari jadwal yang berlangsung hari ini sudah berfungsi dengan baik	✓	

Ruang				
37	Lihat daftar ruangan	Fungsi lihat daftar ruangan sudah berfungsi dengan baik	✓	
38	Lihat ruangan di peta	Fungsi lihat lokasi ruangan di peta sudah berfungsi dengan baik	✓	
39	Cari ruangan	Fungsi cari ruangan sudah berfungsi dengan baik	✓	
40	Cari ruang kosong	Fungsi cari ruangan yang tidak digunakan sudah berfungsi dengan baik	✓	
41	Tambah ruangan	Fungsi tambah data ruangan sudah berfungsi dengan baik	✓	
42	Ubah ruangan	Fungsi ubah data ruangan sudah berfungsi dengan baik	✓	
43	Hapus ruangan	Fungsi hapus data ruangan sudah berfungsi dengan baik	✓	
Notifikasi				
44	Notifikasi tambah jadwal dari following	Fungsi notifikasi adanya penambahan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	✓	
45	Notifikasi perubahan jadwal following	Fungsi notifikasi adanya perubahan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	✓	
46	Notifikasi hapus jadwal dari following	Fungsi notifikasi adanya penghapusan jadwal dari teman yang diikuti (<i>following</i>) sudah berfungsi dengan baik	✓	
47	Notifikasi follower baru	Fungsi notifikasi adanya teman baru yang mengikuti (<i>follower</i>) sudah berfungsi dengan baik	✓	
48	Notifikasi tambah jadwal dari kelas	Fungsi notifikasi adanya penambahan jadwal dari kelas yang diikuti (<i>join</i>) sudah berfungsi dengan baik	✓	
49	Notifikasi ubah jadwal dari kelas	Fungsi notifikasi adanya perubahan jadwal dari kelas yang diikuti (<i>join</i>) sudah berfungsi dengan baik	✓	
50	Lihat semua notifikasi	Fungsi lihat semua notifikasi baik yang sudah dibaca maupun belum, sudah berfungsi dengan baik	✓	
51	Notifikasi jadwal hari ini	Fungsi notifikasi jadwal yang berlangsung hari ini sudah berfungsi dengan baik	✓	
52	Lihat lokasi pelaksanaan jadwal	Fungsi lihat lokasi ruangan dari jadwal yang berlangsung hari ini sudah berfungsi dengan baik	✓	

Tanda Tangan



Lampiran 6. Validasi Bahasa Instrumen *Usability*

VALIDASI BAHASA INSTRUMEN USABILITY

Analisis dan Pengembangan Sistem Informasi Jadwal Akademik Berbasis Yii Framework di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta

A. Identitas Validator

Nama : Astri Ollivia Kuncanya
 Bidang Keahlian : Bahasa Inggris (Linguistik)

B. Pernyataan

No	Pernyataan	Revisi
Usefulness		
1	It helps me be more effective	
	Sistem ini membantu saya menjadi lebih efektif	
2	It helps me be more productive	
	Sistem ini membantu saya menjadi lebih produktif	
3	It is useful	
	Sistem ini bermanfaat <u>bagi saya</u>	
4	It gives me more control over the activities in my life.	
	Sistem ini memberikan kontrol lebih besar terhadap aktivitas saya	
5	It makes the things I want to accomplish easier to get done.	
	Sistem ini membuat hal-hal yang ingin saya lakukan menjadi lebih mudah	
6	It saves me time when I use it.	
	Sistem ini menghemat waktu saya ketika saya gunakan	
7	It meets my needs. ^{memenuhi}	
	Sistem ini <u>sesuai dengan</u> apa yang saya butuhkan	
8	It does everything I would expect it to do.	
	Sistem ini melakukan segala sesuatu yang saya harapkan untuk dilakukan	
Ease of Use		
9	It is easy to use.	
	Sistem ini mudah digunakan	

	<i>It is simple to use.</i>	
10	Sistem ini <u>seederhana</u> untuk digunakan <i>pentis</i>	
	<i>It is user friendly.</i>	
11	Sistem ini user friendly <i>(mudah dipahami)</i>	
	<i>It requires the fewest steps possible to accomplish what I want to do with it.</i>	
12	Sistem ini hanya membutuhkan sedikit langkah-langkah untuk mencapai apa yang saya ingin lakukan	
	<i>It is flexible.</i>	
13	Sistem ini fleksibel <i>Dapat disesuaikan dengan kebutuhan</i>	
	<i>Using it is effortless.</i>	<i>tidak ke</i>
14	Saya menggunakan sistem ini tanpa kesulitan	
	<i>I can use it without written instructions.</i>	
15	Saya dapat menggunakan sistem ini tanpa panduan tertulis	
	<i>I don't notice any inconsistencies as I use it.</i>	
16	Saya tidak melihat adanya inkonsistensi ketika saya menggunakan sistem ini <i>ketidakkonsistengn</i>	
	<i>Both occasional and regular users would like it.</i>	<i>Pengguna yg jarang dan rutin</i>
17	Pengguna (user) teratur akan menyukai sistem ini	
	<i>I can recover from mistakes quickly and easily.</i>	
18	Saya dapat pulih dari kesalahan dengan cepat dan mudah	<i>Sistem ini dapat mengetahui kesalahan saya</i>
	<i>I can use it successfully every time.</i>	
19	Saya dapat menggunakan sistem ini dengan sukses setiap kali saya menggunakannya	
Ease of Learning		
	<i>I learned to use it quickly.</i>	
20	Saya belajar menggunakan sistem ini dengan cepat	
	<i>I easily remember how to use it.</i>	
21	Saya mudah mengingat bagaimana cara menggunakan sistem ini	
	<i>It is easy to learn to use it.</i>	
22	Sistem ini mudah untuk dipelajari penggunaannya	

23	<i>I quickly became skillful with it.</i>	
	Saya dengan cepat menjadi terampil menggunakan sistem ini <i>mahir</i>	
Satisfaction		
24	<i>I am satisfied with it.</i>	
	Saya puas menggunakan sistem ini	
25	<i>I would recommend it to a friend.</i>	
	Saya akan merekomendasikan sistem ini kepada teman saya	
26	<i>It is fun to use.</i>	→ sistem ini menyenangkan untuk digunakan
	Saya merasa senang menggunakan sistem ini	
27	<i>It works the way I want it to work.</i>	
	Sistem ini bekerja seperti apa yang saya inginkan	
28	<i>It is wonderful.</i>	
	Sistem ini <u>indah</u> <i>rasanya bagus</i>	
29	<i>I feel I need to have it.</i>	
	Saya merasa <u>harus memiliki</u> <i>memerlukan</i> sistem ini	
30	<i>It is pleasant to use.</i>	
	Sistem ini <u>menyenangkan</u> <i>nyaman</i> untuk digunakan	

Tanda Tangan Validator



ASTRI OLLIVIA KUNCANYA

Lampiran 7. Hasil Kuesioner *Usability*

Responden	Kuesioner																													
	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	5	5	6	5	7	7	5	4	6	6	5	5	6	4	5	4	6	4	5	4	6	7	5	6	7	5	4	6	5	5
2	6	6	6	4	6	6	6	6	4	4	5	4	4	3	3	6	6	5	6	5	5	6	5	6	6	6	6	5	5	5
3	4	4	4	4	4	4	4	4	3	3	3	3	3	3	4	3	4	3	4	3	3	4	4	3	4	4	4	4	3	4
4	7	7	7	5	6	7	7	6	5	5	6	4	6	6	4	6	7	6	4	5	4	4	5	7	7	7	6	6	7	7
5	5	5	5	5	5	6	5	5	6	6	6	6	5	5	6	6	5	6	6	7	7	7	7	7	6	6	6	5	5	5
6	6	7	6	7	6	7	6	6	6	6	6	5	6	7	6	5	6	5	5	7	7	6	6	7	7	7	7	6	7	7
7	6	5	6	4	6	7	5	4	6	5	6	4	5	5	3	5	5	4	6	5	5	5	5	6	7	6	6	5	7	6
8	7	6	6	7	6	6	7	6	6	5	4	5	6	6	6	5	6	6	6	6	6	6	6	2	6	6	6	7	6	6
9	5	4	4	5	4	6	5	4	4	3	4	3	3	4	4	3	2	3	4	6	5	5	4	3	6	4	4	3	6	3
10	7	7	7	5	6	6	7	5	5	5	6	5	6	5	5	5	5	5	4	5	4	5	5	6	4	5	5	5	5	6
11	6	6	7	5	7	7	7	6	7	6	6	6	5	6	5	5	6	6	7	7	7	7	6	6	6	6	6	6	6	6
12	5	4	5	4	4	6	5	4	4	4	4	5	5	5	5	5	5	4	5	6	6	6	5	4	4	4	4	4	4	4
13	5	5	5	5	5	5	5	5	5	4	4	5	5	5	5	4	4	4	4	5	5	5	5	5	5	5	5	4	5	5
14	6	5	5	2	5	5	4	3	4	4	4	2	4	2	3	1	6	4	4	5	5	5	5	4	6	6	4	6	5	6
15	5	6	5	4	5	6	6	6	5	5	5	5	5	4	4	5	5	4	4	4	4	4	4	5	5	5	5	5	5	5
16	7	7	7	7	6	7	7	6	7	7	7	7	7	7	6	6	6	6	7	7	7	7	6	7	7	7	7	6	7	7
17	6	5	5	5	5	5	5	4	7	7	7	6	5	6	7	6	6	7	7	7	7	7	6	6	5	5	5	5	5	5
18	7	7	7	7	6	7	6	4	4	4	4	5	6	5	5	3	4	4	5	5	4	5	5	6	6	5	5	5	7	5
19	4	4	4	4	4	4	4	4	2	2	2	3	3	2	2	6	4	4	6	4	4	3	3	4	4	4	3	3	3	3
20	6	6	6	6	6	7	6	5	6	5	5	5	6	4	4	7	7	6	6	6	6	7	6	6	6	6	6	6	6	6
21	6	6	6	6	6	5	6	5	6	6	6	5	5	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
22	5	6	7	6	5	6	6	5	6	5	6	4	6	6	5	5	5	4	6	5	6	6	5	6	6	5	6	5	6	6
23	6	7	7	5	6	7	7	7	7	7	7	6	7	6	7	7	7	7	7	7	6	7	7	7	7	7	6	7	7	7
24	6	6	5	6	6	7	6	5	6	6	7	5	6	6	5	6	7	6	7	6	6	7	6	6	7	6	6	5	6	6
25	7	7	7	6	7	7	7	6	7	6	7	6	5	5	6	7	7	6	7	7	7	7	6	6	7	7	6	6	7	6
26	7	7	6	5	6	7	6	5	7	7	6	5	5	5	5	6	7	5	6	7	6	6	5	6	6	6	6	5	6	5
27	6	6	5	5	6	6	6	5	6	5	6	4	4	3	5	4	5	4	6	6	5	6	5	5	6	5	5	4	5	5
28	7	4	7	4	7	7	6	4	7	5	6	6	5	7	7	6	5	4	5	6	6	6	7	6	6	6	6	6	7	6
29	6	6	5	5	5	6	6	5	5	5	6	6	5	5	5	5	6	5	6	6	6	6	6	6	6	6	6	6	6	6
30	7	7	6	6	6	7	7	6	7	7	7	6	6	6	6	6	7	6	7	6	7	7	6	7	7	6	7	7	7	7