

**RANCANG BANGUN SISTEM INFORMASI PRAKTIKUM PROGRAM
TEKNOLOGI INFORMASI DAN ILMU KOMPUTER DENGAN
MENERAPKAN *PATTERN* MVC**

Studi Kasus: PTIIK Universitas Brawijaya

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

WAHYU MULYA CANDRAWATI

NIM. 0910680061

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
PROGRAM STUDI TEKNOLOGI INFORMASI / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER**

UNIVERSITAS BRAWIJAYA

MALANG

2014

LEMBAR PERSETUJUAN

**RANCANG BANGUN SISTEM INFORMASI PRAKTIKUM PROGRAM
TEKNOLOGI INFORMASI DAN ILMU KOMPUTER DENGAN
MENERAPKAN *PATTERN* MVC
Studi Kasus: PTIIK Universitas Brawijaya**

SKRIPSI

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh :

WAHYU MULYA CANDRAWATI

NIM. 0910680061

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Issa Arwani, S.Kom., M.Sc
NIP. 83092206110074

Agi Putra Kharisma, ST., MT

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM INFORMASI PRAKTIKUM PROGRAM
TEKNOLOGI INFORMASI DAN ILMU KOMPUTER DENGAN
MENERAPKAN *PATTERN* MVC**

SKRIPSI

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

WAHYU MULYA CANDRAWATI

NIM. 0910680061

Skripsi ini telah diuji dan dinyatakan lulus pada

Tanggal 21 MEI 2014

Penguji I

Penguji II

Denny Sagita R., S.Kom., M.Kom

NIK. 851 124 061 102 50

Adharul Muttaqin, ST.,MT

NIP. 9760121 200501 1 001

Penguji III

Eriq Muhammad Adams J ,ST.,M.Kom

NIK. 850 410 061 100 27

Mengetahui

Ketua Program Studi Teknologi Informasi / Ilmu Komputer

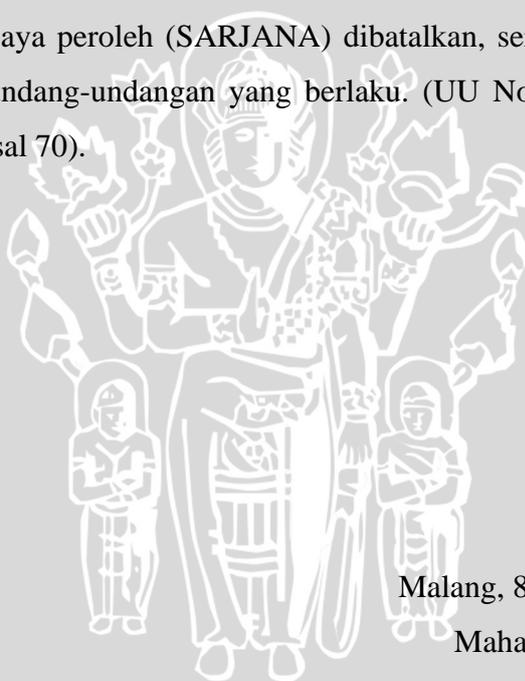
Drs. Marji, M.T.

NIP. 19670801 199203 1 001

**PERNYATAAN
ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, 8 Juni 2014

Mahasiswa,

Wahyu Mulya Candrawati

NIM 0910680061

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Penyayang. Segla puji bagi Allah SWT karena atas rahmat dan hidayah-Nya penulis dapat menyelesaikan Skripsi yang berjudul “RANCANG BANGUN SISTEM INFORMASI PRAKTIKUM PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER DENGAN MENERAPKAN *PATTERN* MVC”. Shalawat dan salam atas junjungan besar kita Nabi Muhammad S.A.W beserta keluarga dan para sahabat sekalian. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Penulis ingin menyampaikan rasa hormat dan terimakasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan – bantuan baik lahir maupun batin selama penulisan skripsi ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terimakasih penulis kepada :

1. Ibunda Retno Teguh Wijayanti, Ayahanda Yulianto, Eyang Putri, Kakak Dudy Kusuma, Kakak Yuvida Agustin, Adek Ratna Ayu dan Adek Leoreta Christina serta seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya didalam membesarkan dan mendidik penulis, serta yang senantiasa dan tiada henti – hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
2. Bapak Ir. Sutrisno MT., Bapak Ir. Heru Nurwasito M.Kom., Bapak Himawat Aryadita ST., M.Sc, dan Bapak Eddy Santoso S.Kom selaku Ketua, Wakil Ketua I, Wakil Ketua II dan Wakil Ketua III Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Drs. Marji MT., dan Bapak Issa Arwani S.Kom., M.Sc, selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
4. Bapak Issa Arwani S.Kom., M.Sc, dan Bapak Agi Putra Kharisma ST,MT selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.

5. Bapak Denny Sagita , Bapak Eriq Muhammad Adams J,ST,.M.Kom, Bapak Adharul Muttaqin ST., MT selaku dosen penguji.
6. Bapak Ir. Suprpto ST.,MT, selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
7. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan dan dedikasinya membagi ilmu kepada penulis.
8. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya.
9. Riezky Ananda Aditya S.Kom, yang selalu memberi semangat, dorongan, nasihat, kesabaran dan kasih sayang kepada penulis selama menyelesaikan penulisan skripsi ini.
10. Teman - teman angkatan 2009: Lutfi Aziz S.Kom, Dwi Arisandy S.Kom, Dara Nisa' S.Kom, Tika Rahmadian S.Kom, Dwinta Ayu Paramitha S.Kom, Alan Nur Abdan S.Kom, Annisa Aini Arifin S.Kom, dll.
11. Sahabat – sahabat: Denissa Nadya Tiffany, Anggun Yunita Sari, Dyah Ayu, Ajeng Ratna Sari, Reyna Trivianika, Lilah Amalia, dan Yolanda Lidya yang selalu menemani dan memberikan dorongan semangat dalam pengerjaan skripsi ini.
12. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesainya skripsi ini.

Hanya doa yang bisa penulis panjatkan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan jauh dari kata sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penulis maupun pihak lain yang menggunakan.

Malang 19 Maret 2014

Penulis

ABSTRAK

Wahyu Mulya Candra. 2014. : Rancang Bangun Sistem Informasi Praktikum Program Teknologi Informasi dan Ilmu Komputer dengan menerapkan *Pattern* MVC. Skripsi Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing : Issa Arwani, S.T, M.Sc dan Agi Putra Kharisma S.T, M.T.

Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya menawarkan berbagai macam mata kuliah yang bisa ditempuh oleh para mahasiswa dimana beberapa mata kuliah tersebut memiliki program praktikum. Proses dokumentasi data dan pendistribusian berkas praktikum mahasiswa masih dilakukan secara manual dan membutuhkan waktu lama. Sedikitnya informasi praktikum yang dapat diketahui oleh dosen menyebabkan dosen kurang mengetahui perkembangan praktikum yang sedang ditempuh oleh anak didiknya, apakah telah sesuai dengan materi yang diajarkan atau tidak. Sistem informasi praktikum yang dikembangkan pada penelitian ini bertujuan untuk membantu mengelola kegiatan praktikum serta memberikan wadah dalam mendistribusikan informasi dan berkas-berkas praktikum secara merata kepada mahasiswa (praktikan), asisten dosen, koordinator asisten dosen, dosen dan *administrator* (laboran). Sistem informasi dikembangkan dengan menggunakan *pattern* MVC yang diimplementasikan pada *framework CodeIgniter*. Model *waterfall* digunakan sebagai metode pengembangan perangkat lunak. Pengujian perangkat lunak dilakukan dengan metode *blackbox* (pengujian fungsionalitas) dan pengujian UAT (*User Acceptance Test*) untuk mengukur apakah fungsional sistem telah memenuhi kebutuhan pengguna. Berdasarkan hasil pengujian fungsional sistem dan pengujian UAT terhadap sistem informasi praktikum yang telah dibuat, didapatkan hasil bahwa sistem telah memenuhi kebutuhan pengguna.

Kata kunci: *CodeIgniter*, MVC, pengujian perangkat lunak, sistem informasi.

ABSTRACT

Wahyu Mulya Candra. , 2014. : The Design and Implementation of Practical Information Systems at Program Information Technology and Computer Science By Applying MVC Pattern. Thesis Informatics Engineering Program, Program Information Technology and Computer Science, University of Brawijaya.

Supervisor: Issa Arwani, ST, M.Sc and Agi Kharisma Putra ST, MT

Programs Information Technology and Computer Sciences UB offers a wide range of courses that could be taken by students in which some of these courses have practicum programs. Process documentation and distribution of datas and file practicum are still done manually and takes a long time. A little information that could be known by lecturer makes lecturer less know the development of practicum that is being doing by his protege, whether in accordance with the material being taught or not. Practicum information system developed in this study aims to help manage practicum activities as well as providing a place to distribute information and files evenly to students (practitioner), assistant lecturer, teaching assistant coordinator, lecturer and administrators. Information system developed using the MVC pattern is implemented in the framework CodeIgniter. The waterfall model is used as a method of software development. Testing was conducted using the blackbox method (testing functional) and UAT testing (User Acceptance Test) to measure whether the system has met the functional needs of the user. Based on the results of both testing, functional testing and UAT of the practicum information system that has been made, showed that the system has met users' needs.

Keywords: codeIgnitter, MVC, *software* testing, system information.

DAFTAR ISI

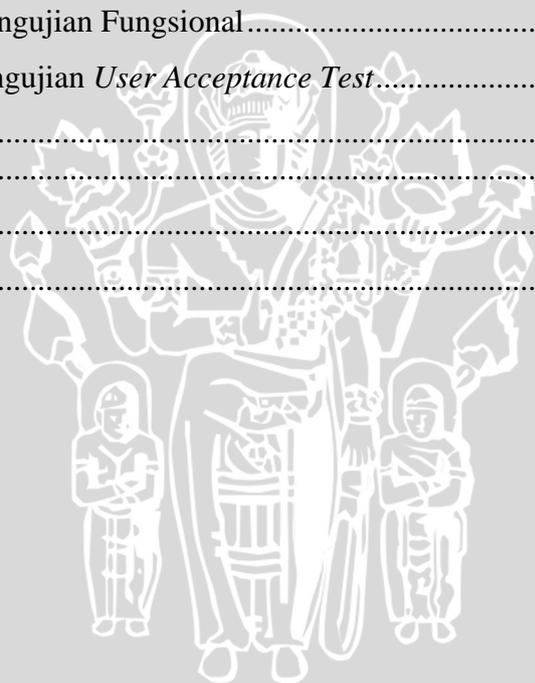
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI	iv
KATA PENGANTAR	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	5
BAB II KAJIAN PUSTAKA DAN DASAR TEORI	6
2.1 Sistem Informasi	6
2.2 <i>Unified Modelling Language</i>	6
2.2.1 Kegunaan UML	7
2.2.2 <i>Use Case Diagram</i>	7
2.2.3 Activity Diagram	8
2.3 PHP9	
2.3.1 Prinsip Kerja PHP	10
2.4 Framework	11
2.4.1 Konsep MVC	11
2.4.2 CodeIgniter	11
2.4.2.1 Struktur <i>File CodeIgniter</i>	12
2.5 Apache Web Server	12
2.6 Model Waterfall	13
2.7 Pengujian Perangkat Lunak	15
2.7.1 Desain Test Case	15

2.7.2 Blackbox Testing	15
BAB III METODE PENELITIAN DAN PERANCANGAN.....	17
3.1 Metode Penelitian	17
3.1.1 Studi dan Pengkajian Literatur.....	19
3.1.2 Analisis Pelaksanaan Praktikum di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya	19
3.1.3 Analisis Data	20
3.1.4 Analisis Kebutuhan Sistem	20
3.1.5 Arsitektur Sistem.....	21
3.1.6 Implementasi	21
3.1.7 Pengujian.....	22
3.1.8 Pengambilan Kesimpulan dan Saran.....	22
3.2 Perancangan	23
3.2.1 Analisis Kebutuhan Perangkat Lunak.....	23
3.2.2 Deskripsi Umum Sistem	24
3.2.2.1 Fungsi Utama Perangkat Lunak.....	24
3.2.3 Identifikasi Aktor	25
3.2.4 Identifikasi Kebutuhan dan Analisis <i>Use Case</i>	26
3.2.4.1 Diagram Aktivitas Sistem.....	31
3.2.5 Perancangan Basis Data	33
3.2.5.1 Tabel <i>m_user</i>	34
3.2.5.2 Tabel <i>m_tahun_ajaran</i>	34
3.2.5.3 Tabel <i>m_program_studi</i>	34
3.2.5.4 Tabel <i>m_praktikum</i>	35
3.2.5.5 Tabel <i>m_tahun_ajaran_has_m_praktikum</i>	35
3.2.5.6 Tabel <i>m_kelas</i>	35
3.2.5.7 Tabel <i>m_jadwal</i>	35
3.2.5.8 Tabel <i>m_asisten</i>	36
3.2.5.9 Tabel <i>m_materi_praktikum</i>	36
3.2.5.10 Tabel <i>mhs_has_m_kelas</i>	36
3.2.5.11 Tabel <i>m_absensi_asdos</i>	37



3.2.5.12	Tabel m_absensi_mhs.....	37
3.2.5.13	Tabel posting	37
3.2.5.14	Tabel ruang	38
3.2.5.15	Tabel ci_sessions	38
3.2.5.16	Tabel m_tugas.....	38
3.2.6	Perancangan Diagram Class.....	39
3.2.7	Perancangan <i>Interface</i>	42
3.2.8	Pengujian Perangkat Lunak.....	43
BAB IV IMPLEMENTASI		45
4.1	Spesifikasi Lingkungan Implementasi Sistem.	45
4.2	Batasan Implementasi	45
4.3	Implementasi Basis data.....	46
4.4	Implementasi Komponen	46
4.4.1	Implementasi Komponen Manajemen Praktikum.....	46
4.4.2	Implementasi Komponen Unggah Modul Praktikum	49
4.4.3	Implementasi Komponen Penjadwalan.....	52
4.5	Implementasi <i>Interface</i>	56
4.4.1	Halaman <i>Login</i>	56
4.4.2	Halaman <i>Home</i>	56
4.4.3	Halaman Kelola Profil	57
4.4.4	Halaman Kelola <i>News feed</i>	58
4.4.5	Halaman Kelola Praktikum.....	59
4.4.6	Halaman Kelola Kelas Praktikum.....	63
4.4.7	Halaman Laporan Absensi Mahasiswa dan Asisten Dosen.	66
4.4.8	Halaman Kelola <i>User</i>	68
4.4.9	Halaman Kelola Gaji Asisten.....	70
4.4.10	Halaman Import Data <i>User</i> From Excel	72
4.4.11	Halaman Kelola Modul dan Tugas Praktikum.....	72
4.4.12	Halaman Kelola Nilai Tugas Mahasiswa.....	75
4.4.13	Halaman Praktikum Mahasiswa.....	77
4.4.14	Halaman Daftar Ajar Praktikum	78

5.1 Pengujian Fungsionalitas	80
5.1.1 Skenario Pengujian	82
5.1.1.1 Home	83
5.1.1.2 Profil	83
5.1.1.4 Manajemen Jadwal	84
5.1.1.5 Manajemen Kelas Praktikum Mahasiswa	86
5.1.1.8 Manajemen Penggajian Asisten Dosen	90
5.2 <i>User Acceptance Test</i>	93
5.2.1 Hasil Pengujian <i>User Acceptance Test</i>	94
5.3 Analisis	94
5.3.1 Analisis Pengujian Fungsional	95
5.3.2 Analisis Pengujian <i>User Acceptance Test</i>	95
BAB VI PENUTUP	98
6.1 Kesimpulan	98
6.2 Saran	99
DAFTAR PUSTAKA	100



DAFTAR GAMBAR

Gambar 2. 1 Struktur File CI	12
Gambar 2. 2 Model Waterfall	13
Gambar 3. 1 Diagram blok metode penelitian	17
Gambar 3. 2 Diagram perancangan sistem	23
Gambar 3. 3 Diagram <i>Use Case</i> Sistem Informasi Praktikum.....	27
Gambar 3. 4 <i>Activity Diagram</i> Pengelolaan Praktikum.....	31
Gambar 3. 5 <i>Activity Diagram</i> Pengelolaan Jadwal Praktikum.....	32
Gambar 3. 6 <i>Activity Diagram</i> Unduh Tugas Praktikum Praktikan	32
Gambar 3. 7 Rancangan Database Sistem.....	33
Gambar 3. 8 <i>Diagram Class</i> Untuk <i>Controller</i> Pada Sistem Informasi Praktikum	40
Gambar 3. 9 <i>Diagram Class</i> Untuk Model Pada Sistem Informasi Praktikum	41
Gambar 3. 10 <i>Diagram Class</i> Untuk Subsistem Daftar Praktikum	42
Gambar 3. 11 Perancangan interface halaman login.....	42
Gambar 3. 12 Perancangan <i>interface</i> konten sistem informasi.....	43
Gambar 4. 1 Diagram ER konseptual dari sistem.....	46
Gambar 4. 2 Implementasi <i>interface login</i>	56
Gambar 4. 3 Implementasi <i>interface home</i>	57
Gambar 4. 4 Implementasi <i>interface profil</i>	57
Gambar 4. 5 Implementasi <i>interface</i> manajemen <i>news feed</i>	58
Gambar 4. 6 Implementasi <i>interface add news feed</i>	58
Gambar 4. 7 Implementasi <i>interface</i> manajemen tahun ajaran.....	59
Gambar 4. 8 Implementasi <i>interface</i> tambah tahun ajaran	59
Gambar 4. 9 Implementasi <i>interface</i> manajemen program studi	60
Gambar 4. 10 Implementasi <i>interface</i> tambah data program studi	60
Gambar 4. 11 Implementasi <i>interface</i> manajemen praktikum	61
Gambar 4. 12 Implementasi <i>interface</i> tambah data praktikum.....	61
Gambar 4. 13 Implementasi <i>interface</i> manajemen praktikum pada tahun ajaran. 62	
Gambar 4. 14 Implementasi <i>interface</i> menambahkan praktikum pada tahun ajaran	62
Gambar 4. 15 Implementasi <i>interface</i> manajemen ruang	63
Gambar 4. 16 Implementasi <i>interface</i> tambah data ruang	63
Gambar 4. 17 Implementasi <i>interface</i> manajemen kelas	64
Gambar 4. 18 Implementasi <i>interface</i> tambah data kelas praktikum.....	64
Gambar 4. 19 Implementasi <i>interface</i> manajemen jadwal.....	65
Gambar 4. 20 Implementasi <i>interface</i> tambah mahasiswa ke kelas praktikum	65
Gambar 4. 21 Implementasi <i>interface</i> form untuk memilih kelas praktikum	66

Gambar 4. 22 Implementasi <i>interface</i> laporan absensi mahasiswa (praktikan)....	66
Gambar 4. 23 Implementasi <i>interface</i> form untuk pilih laporan absensi asisten dosen berdasarkan mata kuliah	67
Gambar 4. 24 Implementasi <i>interface</i> laporan absensi asisten dosen & koordinator asisten dosen	67
Gambar 4. 25 Implementasi <i>interface</i> manajemen <i>users</i>	68
Gambar 4. 26 Implementasi <i>interface</i> untuk tambah <i>user</i> baru	68
Gambar 4. 27 Implementasi <i>interface</i> input data asisten dosen secara kelompok	69
Gambar 4. 28 Halaman <i>interface</i> daftar asisten dosen dan koordinator asisten dosen	69
Gambar 4. 29 Implementasi <i>interface</i> tambah data dosen ke kelas praktikum.....	70
Gambar 4. 30 Implementasi <i>interface</i> manajemen penggajian.....	71
Gambar 4. 31 Halaman <i>interface</i> form laporan gaji asisten dosen	71
Gambar 4. 32 Implementasi <i>interface</i> laporan gaji asisten dan koordinator asisten	71
Gambar 4. 33 Implementasi <i>interface import users from excel</i>	72
Gambar 4. 34 Implementasi <i>interface</i> manajemen modul praktikum.....	73
Gambar 4. 35 Implementasi <i>interface</i> tambah modul praktikum	73
Gambar 4. 36 Implementasi <i>interface</i> kelola tugas praktikum	74
Gambar 4. 37 Implementasi <i>interface</i> tambah tugas praktikum	74
Gambar 4. 38 Implementasi <i>interface</i> form untuk menampilkan tugas praktikum	75
Gambar 4. 39 Implementasi <i>interface</i> jawaban tugas praktikum mahasiswa (praktikan).....	75
Gambar 4. 40 Implementasi <i>interface</i> form untuk menampilkan data mahasiswa (praktikan).....	76
Gambar 4. 41 Implementasi <i>interface</i> untuk input nilai tugas praktikum.....	76
Gambar 4. 42 Implementasi <i>interface</i> form <i>update</i> nilai mahasiswa.....	77
Gambar 4. 43 Halaman <i>interface</i> daftar praktikum yang ditempuh.....	77
Gambar 4. 44 Implementasi <i>interface</i> detail praktikum.....	78
Gambar 4. 45 Implementasi <i>interface</i> daftar praktikum yang diajarkan dosen	78
Gambar 4. 46 Implementasi <i>interface</i> detail modul dan tugas praktikum	79



DAFTAR TABEL

Tabel 2. 1 Elemen-elemen <i>Use Case Diagram</i>	8
Tabel 2. 2 Elemen-elemen activity diagram	9
Tabel 3. 1 Identifikasi aktor	25
Tabel 3. 2 Spesifikasi kebutuhan fungsional aktor praktikan	28
Tabel 3. 3 Spesifikasi kebutuhan fungsional aktor <i>administrator</i>	28
Tabel 3. 4 Spesifikasi kebutuhan fungsional aktor asisten dosen dan koordinator asisten dosen	29
Tabel 3. 5 Spesifikasi kebutuhan fungsional aktor dosen.....	30
Tabel 3. 6 Detail tabel <i>m_user</i>	34
Tabel 3. 7 Detail tabel <i>m_tahun_ajaran</i>	34
Tabel 3. 8 Detail tabel <i>m_program_studi</i>	34
Tabel 3. 9 Detail tabel <i>m_praktikum</i>	35
Tabel 3. 10 Detail tabel <i>m_tahun_ajaran_has_m_praktikum</i>	35
Tabel 3. 11 Detail tabel <i>m_kelas</i>	35
Tabel 3. 12 Detail tabel <i>m_jadwal</i>	36
Tabel 3. 13 Detail tabel <i>m_asisten</i>	36
Tabel 3. 14 Detail tabel <i>m_materi_praktikum</i>	36
Tabel 3. 15 Detail tabel <i>mhs_has_m_kelas</i>	37
Tabel 3. 16 Detail tabel <i>m_absensi_asdos</i>	37
Tabel 3. 17 Detail tabel <i>m_absensi_mhs</i>	37
Tabel 3. 18 Detail tabel <i>posting</i>	38
Tabel 3. 19 Detail tabel <i>ruangan</i>	38
Tabel 3. 20 Detail tabel <i>ci_sessions</i>	38
Tabel 3. 21 Detail tabel <i>m_tugas</i>	38
Tabel 3. 22 Detail tabel <i>m_tugas_mhs</i>	39
Tabel 3. 24 Kuisisioner berdasarkan beberapa faktor.....	44
Tabel 4. 1 Tabel spesifikasi <i>hardware</i> implementasi sistem	45
Tabel 4. 2 Spesifikasi <i>software</i> implementasi sistem.....	45
Tabel 4. 3 <i>Controller c_listTAPraktikum</i>	47
Tabel 4. 4 Model <i>m_listTAPraktikum</i>	47
Tabel 4. 5 View <i>v_listTAPraktikum</i>	48
Tabel 4. 6 Controller <i>c_materi_praktikum</i>	50
Tabel 4. 7 Model <i>m_materi_praktikum</i>	51
Tabel 4. 8 View <i>v_materi_praktikum</i>	51
Tabel 4. 9 <i>Controller c_jadwal</i>	52
Tabel 4. 10 Model <i>m_jadwal</i>	54
Tabel 4. 11 View <i>v_addJadwal</i>	55

Tabel 5. 1 Daftar <i>test requirements</i>	80
Tabel 5. 2 Skenario pengujian halaman home	83
Tabel 5. 3 Skenario pengujian fitur <i>update</i> profil	83
Tabel 5. 4 Skenario pengujian fitur manajemen kelas	84
Tabel 5. 5 Skenario pengujian form tambah jadwal	85
Tabel 5. 6 Skenario pengujian form <i>update</i> jadwal	85
Tabel 5. 7 Skenario pengujian form input mahasiswa ke kelas praktikum.....	86
Tabel 5. 8 Skenario pengujian form tambah praktikum.....	86
Tabel 5. 9 Skenario pengujian form upload modul praktikum	87
Tabel 5. 10 Skenario pengujian form upload tugas praktikum	87
Tabel 5. 11 Skenario pengujian form <i>update</i> modul praktikum	88
Tabel 5. 12 Skenario pengujian input absensi mahasiswa	88
Tabel 5. 13 Skenario pengujian input data kehadiran asisten dosen.....	89
Tabel 5. 14 Skenario pengujian laporan absensi mahasiswa	89
Tabel 5. 15 Skenario pengujian laporan absensi asisten dosen.....	89
Tabel 5. 16 Skenario pengujian form input gaji asisten.....	90
Tabel 5. 17 Skenario pengujian halaman laporan gaji asisten dosen.....	90
Tabel 5. 18 Skenario pengujian form <i>add new user</i>	91
Tabel 5. 19 Skenario pengujian form <i>update user</i>	91
Tabel 5. 20 Skenario pengujian form <i>import data users from excel</i>	92
Tabel 5. 21 Skenario pengujian form <i>add news feed</i>	92
Tabel 5. 22 Skenario pengujian form <i>update news feed</i>	93
Tabel 5. 23 Hasil Pengujian UAT.....	94



DAFTAR LAMPIRAN

Halaman

Lampiran 1. Kuisisioner penelitian.....101



BAB I PENDAHULUAN

1.1 Latar Belakang

Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya memiliki berbagai macam mata kuliah yang dapat ditempuh oleh para mahasiswa dimana beberapa mata kuliah memiliki program praktikum. Kegiatan praktikum merupakan salah satu kegiatan penting untuk mendukung para mahasiswa dalam proses pembelajaran lebih lanjut untuk memahami dan mengembangkan pengetahuan mahasiswa terhadap materi mata kuliah yang telah didapatkan di dalam kelas.

Pengelolaan pelaksanaan praktikum di Program Teknologi Informasi dan Ilmu Komputer masih dilakukan secara konvensional. Hal ini ditinjau dari data / informasi praktikum yang belum didokumentasikan secara sistematis dan informatif sehingga dirasa kurang dalam memberikan pelayanan yang memadai terkait kebutuhan informasi praktikum kepada pihak - pihak yang terkait didalamnya. Permasalahan yang muncul dilapangan diantaranya pertama, bagi praktikan dan asisten dosen adalah keterbatasan pendistribusian berkas praktikum, keterbatasan memperoleh informasi jadwal praktikum, keterbatasan mengakses berita prak-tikum dan nilai praktikum. Kedua, permasalahan terkait manajemen jadwal praktikum, pengelolaan absensi mahasiswa dan absensi asisten dosen serta pengelolaan gaji asisten dosen yang masih dilakukan secara konvensional, sehingga data - datanya belum dapat didistribusikan secara merata dan *real time* kepada pihak – pihak yang terkait. Ketiga, dosen kurang mengetahui perkembangan kegiatan praktikum dikarenakan sedikitnya informasi yang dapat diakses oleh dosen sehingga dosen kurang dapat memantau apakah kegiatan pembelajaran praktikum telah sesuai dengan apa yang diajarkan atau belum.

Penelitian ini bertujuan untuk membantu menyelesaikan permasalahan yang ada dalam pelaksanaan kegiatan praktikum yaitu dengan membangun sistem yang lebih fleksibel dan membuat lingkungan belajar yang adaptif. Sistem yang

dimaksud adalah sistem yang dapat menampung dan mengolah data / informasi serta mendistribusikannya sesuai kebutuhan pengguna secara *real time* dan dapat diakses kapan saja dan dimana saja. Pengguna sistem ini adalah pihak-pihak yang berkaitan dengan penyelenggaraan praktikum, antara lain praktikan, asisten dosen, koordinator asisten dosen, dosen dan *administrator* (laboran).

MVC (*Model-View-Controller*) adalah konsep yang digunakan didalam *framework* untuk membangun aplikasi. Konsep MVC adalah konsep pemisahan antara *logic* dengan tampilan dan basis data. Manfaat konsep ini adalah membuat *coding logic* lebih sederhana. *CodeIgniter* adalah sebuah php *framework* yang mengadopsi *pattern MVC*, terdiri dari kumpulan *folder*, *file php*, *java script*, *css*, *txt* dan *file* berbasis *web* lainnya dengan *setting* tertentu untuk menggunakannya. *CodeIgniter* juga menyediakan *library* dan *helper* yang dapat dimanfaatkan didalam pemrograman php [2].

Berdasarkan latar belakang yang telah dijabarkan, penulis mengambil judul “Rancang Bangun Sistem Informasi Praktikum pada Program Teknologi Informasi dan Ilmu Komputer dengan menerapkan *Pattern MVC*”. Penelitian ini ditujukan dengan harapan sistem informasi praktikum yang akan dibuat dapat membantu memudahkan dalam mengelola serta mengontrol kegiatan praktikum sehingga dapat tersistemasi dengan lebih baik.

1.2 Rumusan Masalah

Dari uraian diatas dapat dirumuskan bahwa permasalahan yang menjadi objek sistem informasi ini adalah:

1. Bagaimana membuat sistem informasi yang *user friendly* dan handal yang memudahkan pengguna dalam mengoperasikannya serta dapat membantu mengelola kegiatan praktikum dan bagaimana sistem informasi dapat dikembangkan sesuai kebutuhan pengguna?
2. Bagaimana pengujian berhasil menjawab dan memenuhi standar kebutuhan pengguna terhadap sistem yang dibangun?
3. Bagaimana sistem dirancang untuk dapat memenuhi kebutuhan pengguna dalam membantu kegiatan praktikum?

1.3 Batasan Masalah

Batasan masalah pada sistem informasi yang akan dibangun adalah sebagai berikut:

- a. Studi kasus yang diambil bertempat di Program Teknologi Informasi dan Ilmu Komputer dengan terbatas untuk permasalahan manajemen praktikum.
- b. Sistem informasi praktikum hanya menangani masalah manajemen praktikum, pengelolaan penggajian asisten dosen, pengelolaan jadwal praktikum, rekap nilai tugas praktikum, distribusi modul dan tugas praktikum, pengumuman terkait praktikum serta sistem absensi mahasiswa dan asisten dosen.
- c. Pengelolaan jadwal praktikum hanya terbatas pada data kegiatan praktikum, tidak mencakup kondisi perkuliahan/kegiatan lainnya.
- d. Tidak ada algoritma untuk penjadwalan, jadwal praktikum dimasukkan oleh *administrator* kedalam basis data sistem secara manual dari data yang telah diolah oleh *administrator*.
- e. Bahasa pemrograman yang digunakan adalah PHP dengan pemrograman menggunakan *pattern MVC* yang diimplementasikan pada *framework CodeIgniter* dan menggunakan teknologi *AJAX* dengan *JQuery*.
- f. Aplikasi dikembangkan menggunakan model *Waterfall*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengembangkan aplikasi sistem informasi praktikum pada Program Teknologi Informasi dan Ilmu Komputer.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah :

- a. Bagi Penulis:

Dapat lebih memahami tentang pengembangan aplikasi web menggunakan *pattern MVC* yang diimplementasikan pada *framework CodeIgniter* dan dapat lebih memahami cara menemukan solusi dalam memecahkan suatu studi kasus.

b. Bagi Praktikan:

Memudahkan dalam mengakses data / informasi dan berkas praktikum, antara lain jadwal praktikum, materi praktikum, tugas praktikum, pengumpulan tugas, nilai praktikum, berita serta absensi.

c. Bagi asisten dosen:

Memberikan kemudahan bagi asisten dosen untuk:

1. Mendistribusikan berkas tugas praktikum serta melakukan *update* berkas tugas praktikum.
2. Mengakses informasi gaji asisten dosen.
3. Memberikan pengumuman terkait praktikum.
4. Rekap nilai tugas praktikum.
5. Mengelola dan mengakses absensi mahasiswa.
6. Input absensi asisten dosen.

d. Bagi koordinator asisten dosen:

Secara keseluruhan manfaat penelitian ini bagi koordinator asisten dosen sama seperti asisten dosen. Menu tambahan yang dimiliki koordinator asisten dosen antara lain adalah memiliki hak akses untuk mengunggah modul praktikum, melihat laporan absensi seluruh mahasiswa dan absensi asisten dosen serta laporan nilai praktikum mahasiswa yang dikoordinir.

e. Bagi *Administrator*:

Dapat memudahkan dalam mengontrol dan manajemen praktikum diantaranya:

1. manajemen praktikum pada tahun ajaran,
2. manajemen dosen,
3. manajemen jadwal praktikum,
4. manajemen ruang praktikum,
5. manajemen mahasiswa peserta praktikum,
6. manajemen asisten dosen dan koordinator asisten dosen,
7. manajemen penggajian asisten dosen, dan
8. akses laporan nilai mahasiswa.

f. Bagi Dosen:

Dosen dapat memantau perkembangan kegiatan praktikum yang dilaksanakan mahasiswanya, apakah sesuai dengan apa yang diajarkan atau belum.

g. Bagi Program Teknologi Informasi dan Ilmu Komputer:

Dapat meningkatkan pelayanan pelaksanaan praktikum.

1.6 Sistematika Penulisan

Untuk mencapai tujuan yang diharapkan maka sistematika penulisan yang disusun dalam tugas akhir ini adalah:

BAB I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

BAB II Dasar Teori

Bab ini berisi tentang tinjauan pustaka dan dasar teori.

BAB III Metodologi Penelitian dan Perancangan

Bab ini membahas tentang metode yang digunakan dalam menyelesaikan masalah yang dibahas dalam skripsi ini. Serta berisi tentang rencana pelaksanaan, alat, bahan, jalannya perancangan dan hasil yang diharapkan.

BAB IV Implementasi

Bab ini berisi tentang gambaran sistem dan deskripsi sistem hasil analisis kebutuhan, perancangan, implementasi, dan analisis pengujian.

BAB V Pengujian dan Analisis

Memuat tentang analisa dan pembahasan hasil penelitian yang telah dilakukan.

BAB VI Penutup

Memuat tentang kesimpulan dan saran-saran berdasarkan hasil pembahasan yang telah diperoleh.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini membahas dasar - dasar teori yang digunakan untuk menunjang penulisan skripsi “Rancang Bangun Sistem Informasi Praktikum di Program Teknologi Informasi dan Ilmu Komputer menggunakan *Pattern MVC*”. Dasar teori yang diperlukan berdasarkan latar belakang dan rumusan masalah adalah: Sistem informasi, *UML*, *PHP*, *MVC*, *Framework*, *CodeIgniter*, *Apache Server*, *MySQL*, *Ajax* dengan *JQuery*, *Waterfall Model* dan pengujian perangkat lunak.

2.1 Sistem Informasi

Menurut Laudon, Kenneth, Jane sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan - laporan yang diperlukan [KEN-07].

Menurut Budi Sutedjo Dharma Oetomo sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain untuk membentuk suatu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi tersebut [KEN-07].

2.2 Unified Modelling Language

Unified Modelling Language (UML) adalah notasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem berisi informasi dan fungsi tetapi secara normal digunakan untuk memodelkan sistem komputer. Bahasa permodelan UML mengijinkan deskripsi dari sistem dibuat dengan mendetail pada setiap level abstraksi. UML dimaksudkan untuk menjadi *unifying language* yang memungkinkan professional IT untuk memodelkan aplikasi/sistem komputer. Salah satu alasan UML memiliki *open standart modeling language* adalah bahwa merupakan bahasa pemrograman independen serta notasi UML adalah bahasa dan bukan metodologi. Bahasa merupakan lawan dari metodologi, dimana ia dapat

dengan mudah masuk kedalam bisnis apapun dari proses pengembangan tanpa melakukan perubahan [DBL-03].

UML memberikan kemudahan pemahaman aplikasi pada proses *development* yaitu pada tahap analisis dan desain. Sintak yang didesain bersifat independen dari bahasa target, proses perangkat lunak atau *tool*, tapi cukup umum dan fleksibel karena dapat dikustomisasi. Pada dasarnya UML berhubungan dengan pengetahuan yang ditangkap, dikomunikasikan dan dikembangkan[DBL-03].

2.2.1 Kegunaan UML

UML merupakan bahasa pemodelan *general-purpose* yang evolusioner, mudah diaplikasikan dan merupakan standar industri. UML dapat diaplikasikan pada bermacam tipe sistem, domain dan metode atau proses.

1. Sebagai bahasa pemodelan yang *general-purpose*, difokuskan pada pokok himpunan konsep yang dapat dipakai bersama, dan menggunakan pengetahuan bersama dengan mekanisme perluasan.
2. Sebagai bahasa pemodelan yang mudah diaplikasikan, dapat diaplikasikan untuk bermacam tipe sistem (*software* dan *non-software*), domain (*bisnis vs software*) dan metode atau proses.
3. Sebagai bahasa pemodelan standart industri, bukan merupakan bahasa yang tertutup atau satu-satunya, tapi bersifat terbuka dan sepenuhnya dapat diperluas.

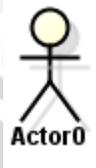
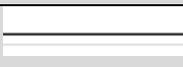
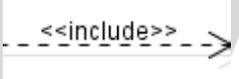
UML memungkinkan untuk menangkap, mengkomunikasikan dan mengembangkan strategi, taktik dan mengoperasikan pengetahuan untuk meningkatkan nilai tambah melalui peningkatan kualitas, mereduksi biaya, mereduksi *time-to-market*.

2.2.2 Use Case Diagram

Use Case Diagram menggambarkan interaksi yang terjadi antara sistem, sistem luar dan pengguna. *Use case diagram* menjelaskan siapa yang akan

menggunakan sistem dan dengan cara apa pengguna berinteraksi dengan sistem [JLW-04].

Tabel 2. 1 Elemen-elemen *Use Case Diagram*

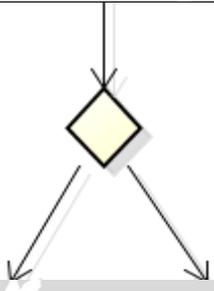
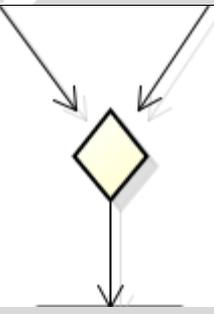
No	Nama elemen	Fungsi	Notasi
1	Aktor	Menggambarkan tokoh atau sistem yang mendapatkan hak akses sistem dan berada di luar dari sistem. aktor ditempatkan di luar <i>subject boundary</i> .	
2	<i>Use case</i>	Mewakili sebuah bagian dari fungsionalitas sistem dan ditempatkan dalam <i>system boundary</i> .	
3	<i>Association relationship</i>	Menghubungkan aktor untuk berinteraksi dengan <i>use case</i> .	
4	<i>Generalization relationship</i>	Menunjukkan generalisasi dari <i>use case</i> khusus ke yang lebih umum.	
5	<i>Include relationship</i>	Menunjukkan <i>inclusion</i> fungsionalitas dari sebuah <i>use case</i> dengan <i>use case</i> lainnya.	

Sumber – Desanti, 2010

2.2.3 Activity Diagram

Activity diagram digunakan untuk menggambarkan aliran aktivitas baik proses bisnis atau *use case*. *Activity diagram* juga digunakan untuk memodelkan aksi yang akan dilakukan saat sebuah operasi dieksekusi dan memodelkan hasil dari aksi tersebut [JLW-04]. Elemen-elemen *activity diagram* ditunjukkan pada tabel 2.2

Tabel 2. 2 Elemen-elemen activity diagram

No	Nama Elemen	Fungsi	Notasi
1	Aktivitas	Untuk mewakili kumpulan aksi	
2	<i>Control Flow</i>	Menunjukkan rangkaian pelaksanaan	
3	<i>Initial Node</i>	Menandakan awal dari kumpulan aksi atau aktivitas	
4	<i>Final-activity Node</i>	Untuk menghentikan seluruh <i>control flows</i> atau <i>object flows</i> pada sebuah aktivitas atau aksi	
5	<i>Decision Node</i>	Untuk mewakili suatu kondisi pengujian yang bertujuan untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> hanya menuju ke satu arah	
6	<i>Merge Node</i>	Untuk menyatukan kembali <i>decision path</i> yang dibuat dengan menggunakan <i>decision node</i>	

Sumber: Desanti 2010

2.3 PHP

Script PHP adalah bahasa program yang berjalan pada sebuah *web server* atau sering disebut *server-side*. PHP dapat melakukan apa saja yang bisa dilakukan program CGI lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman web yang dinamis serta menerima dan menciptakan *cookies*.

Script sever-side membutuhkan tiga program utama, yaitu *web-server* (dapat berupa IIS dari windows atau apache), modul PHP dan juga *web browser* agar dapat berjalan. PHP dapat berjalan pada beberapa jenis *web-server* antara lain Apache, Microsoft IIS, personal *web server*, Netscape dan Iplanet Server, Caudium, Xitami, Omnihttpd dan masih banyak lagi. Output yang dihasilkan PHP bukan hanya HTML, namun juga dalam bentuk gambar, *file* PDF, serta gambar animasi menggunakan Libswf dan Ming. Output yang lain dengan jenis teks dapat berupa *file* XHTML dan XML[WJG-10].

Script PHP disembunyikan dalam *server* dan *output* teks yang ditampilkan di *browser* adalah format teks HTML, XHTML atau XML. PHP juga banyak mendukung basis data populer yang sering digunakan dalam beberapa *web server*, antara lain : Adabas D, Empress, FrontBase, PostgreSQL, FilePro (*read only*), mSQL, Solid, Hyperwave, Direct MS-SQL, Sybase, IBM DB2, MySQL, Velocis, Informix, ODBC dan Unix dbm[WJG-10].

2.3.1 Prinsip Kerja PHP

Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan *url* atau alamat *website* dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *web server*, mengidentifikasi halaman yang dikehendaki dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya *web server* akan mencari *file* yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan berkas yang diminta segera menerjemahkan kode HTML dan menampilkannya. Ketika *user* melakukan *request* pada halaman yang mengandung *script* PHP maka pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *web server*, *web server* akan memeriksa tipe *file* yang diminta *user*. Jika tipe berkas yang diminta adalah PHP, maka akan memeriksa isi *script* dari halaman PHP tersebut. Apabila dalam *file* tersebut tidak mengandung *script* PHP, permintaan *user* akan langsung ditampilkan ke *browser*, namun jika dalam *file* tersebut mengandung *script* PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan *script-script* PHP dan mengolah

script tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke *browser user* [WJG-10].

2.4 Framework

Framework adalah sekumpulan perintah atau fungsi dasar yang dapat membantu dalam menyelesaikan proses-proses yang lebih kompleks, menanggapi berbagai masalah dalam pemrograman seperti koneksi basis data, pemanggilan variabel, dll. *Framework* memungkinkan *developer* lebih fokus dan lebih cepat dalam membangun aplikasi [BTS-12].

2.4.1 Konsep MVC

Konsep MVC merupakan konsep yang digunakan didalam *framework* untuk membangun aplikasi, pada konsep MVC dilakukan pemisahan antara *logic* dengan tampilan dan basis data. Manfaat konsep MVC adalah membuat *coding logic* lebih sederhana [BTS-12].

1. *Model*

Merupakan struktur data. Model berisi fungsi didalam pengolahan basis data, *script SQL* dikelola disini.

2. *View*

Merupakan kode untuk tampilan suatu program. Tampilan dapat berupa *web page, header, footer, content* dan apa saja yang berjenis tampilan.

3. *Controller*

Merupakan kode untuk *logic, algoritma* dan sebagai penghubung antara *model, view* dan sumber lain yang diperlukan untuk mengolah HTTP *request* dan *generate web page*.

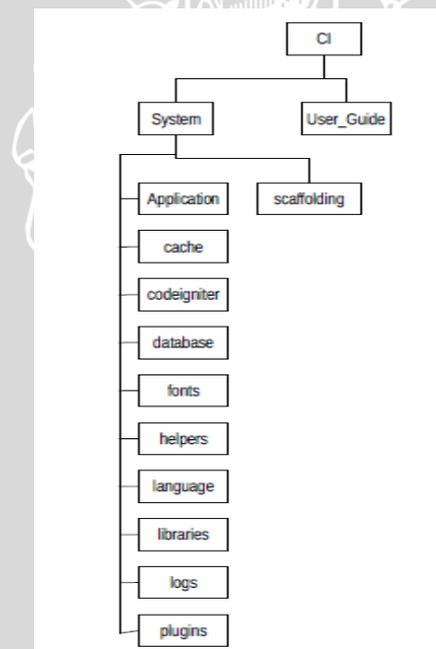
2.4.2 CodeIgniter

CodeIgniter adalah salah satu *framework PHP* yang tangguh dan populer. *CodeIgniter* menerapkan *pattern MVC* yang *flexible* karena model dapat tidak digunakan. Pengembang dapat hanya menggunakan *view* dan *controller* saja pada *CodeIgniter* apabila *model* tidak diperlukan *CodeIgniter* membuat program

dengan menggunakan PHP lebih sistematis. Pemrogram tidak perlu membuat program dari awal (*from scratch*) karena CI menyediakan sekumpulan librari yang banyak dimana librari tersebut diperlukan untuk menyelesaikan pekerjaan umum dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengakses librarinya. Pemrogram dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan [BTS-12].

2.4.2.1 Struktur File CodeIgniter

CodeIgniter terdiri dari kumpulan *folder* dan *file php*, *java script*, *css*, *txt* dan *file* berbasis web lainnya dengan *setting* tertentu untuk menggunakannya, dan menyediakan *library* serta *helper* yang dapat dimanfaatkan didalam pemrograman php. Berikut adalah struktur *file codeIgniter*:



Gambar 2. 1 Struktur File CI
Sumber komunitas eLearning ilmukomputer.com

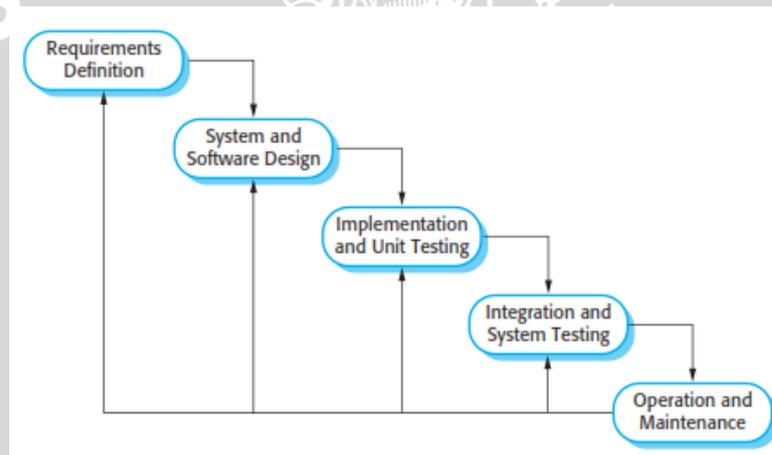
2.5 Apache Web Server

Apache Web server adalah perangkat lunak yang memberikan layanan *website*. *Web server* menggunakan protokol yang disebut dengan HTTP (*HyperText Transfer Protocol*). Banyak pilihan *web server* di dunia *open source*

yang dapat digunakan, penggunaannya bergantung pada keperluan. Apache adalah salah satu *web server* yang sangat terkenal dan menjadi standar *de facto* setiap distribusi Linux. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain [ABD-10].

2.6 Model Waterfall

Systems Development Life Cycle (SDLC) dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan suatu sistem. Salah satu model yang cukup dikenal dalam dunia rekayasa perangkat lunak adalah model *waterfall*. Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software* [ISM-11].



Gambar 2. 2 Model Waterfall
Sumber [ISM-11]

1. Requirements Analysis and Definition

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan yang harus dipenuhi oleh perangkat lunak yang akan dibangun. Hal ini sangat penting mengingat perangkat lunak harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database* dan sebagainya.

2. System and Software Design

Proses pencarian kebutuhan diintensifkan dan difokuskan pada perangkat lunak. *Software engineer* harus mengerti tentang domain informasi dari perangkat

lunak, misalnya fungsi yang dibutuhkan, *user interface* dan sebagainya untuk mengetahui sifat dari program yang akan dibuat. Pencarian kebutuhan sistem dan perangkat lunak harus didokumentasikan dan ditunjukkan kepada *user*. Proses *software design* untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam perangkat lunak berbentuk “*blueprint*” sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari perangkat lunak.

3. *Implementation and Unit Testing*

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

4. *Integration and System Testing*

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap desain yang secara teknis nantinya dikerjakan oleh *programmer*. Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).

5. *Operation and Maintenance*

Sesuatu yang dibuat haruslah diujicobakan, demikian juga dengan perangkat lunak. Semua fungsi-fungsi perangkat lunak harus diujicobakan agar perangkat lunak bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya. Pemeliharaan suatu perangkat lunak diperlukan, termasuk di dalamnya adalah pengembangan, karena kebutuhan perangkat lunak akan terus bertambah sesuai perkembangan teknologi yang ada. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada perangkat lunak tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan, seperti ketika ada pergantian sistem operasi atau perangkat lainnya.

2.7 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses atau serangkaian proses yang dirancang untuk memastikan kode computer berjalan sesuai dengan apa yang dirancang, sehingga kode dapat melakukan apa yang seharusnya dan tidak seharusnya dilakukan. Perangkat lunak harus diprediksi dan konsisten, sehingga tidak menimbulkan aksi yang diluar dugaan [GJM-04].

Dalam melakukan uji coba ada dua masalah yang penting yang akan dibahas, yaitu:

1. Teknik uji coba perangkat lunak.
2. Strategi uji coba perangkat lunak.

2.7.1 Desain Test Case

Terdapat bermacam-macam rancangan metode *test case* yang dapat digunakan, semua menyediakan pendekatan sistematis untuk uji coba, yang terpenting metode menyediakan kemungkinan yang cukup tinggi menemukan kesalahan. Terdapat dua macam *test case*:

1. Pengetahuan fungsi yang spesifik dari produk yang telah dirancang untuk diperlihatkan, test dapat dilakukan untuk menilai masing-masing fungsi apakah telah berjalan sebagaimana yang diharapkan.
2. Pengetahuan tentang cara kerja dari produk, test dapat dilakukan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah untuk melihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya.

2.7.2 Blackbox Testing

Test case ini bertujuan berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemutahirannya. *Blackbox* adalah salah satu metode pengujian penting. Metode *blackbox* digunakan dengan cara memandang sebuah program sebagai *blackbox*, tujuannya adalah untuk benar-benar melihat perilaku internal dan struktur program. *Test case* ini juga menitik beratkan untuk menemukan situasi dimana program tidak berlaku sesuai

dengan spesifikasinya. Dalam pendekatan *blackbox* data yang diuji hanya berasal dari spesifikasi (yaitu tanpa mengambil keuntungan dari pengetahuan tentang struktur internal program). Penggunaan pendekatan *blackbox* adalah dengan mencoba setiap masukan kondisi input [GJM-04].

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program.

Tujuan metode ini mencari kesalahan pada:

1. Fungsi yang salah atau hilang.
2. Kesalahan pada *interface*.
3. Kesalahan pada struktur data atau akses basis data.
4. Kesalahan performansi.
5. Kesalahan inisialisasi dan tujuan akhir.



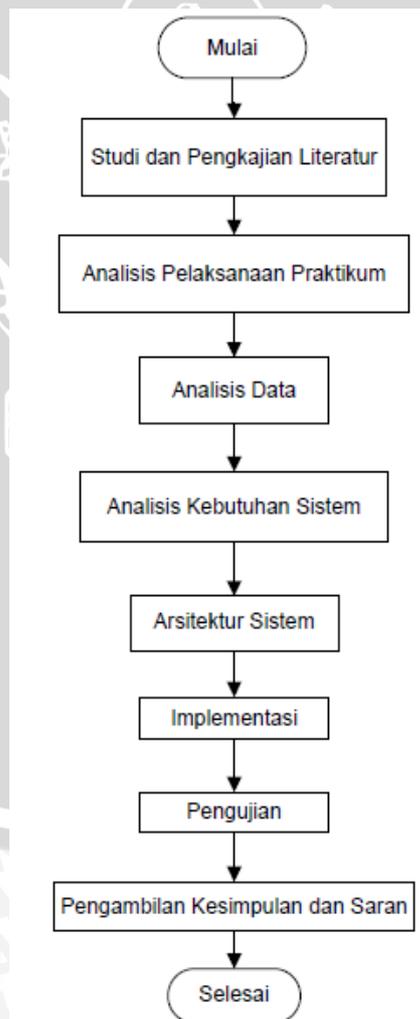
BAB III

METODE PENELITIAN DAN PERANCANGAN

Bab ini menjelaskan metode penelitian yang digunakan pada sistem informasi praktikum PTIIK. Pengembangan perangkat lunak dikembangkan dengan metode pengembangan perangkat lunak *waterfall*.

3.1 Metode Penelitian

Metode penelitian berisi langkah langkah yang digunakan dalam pembuatan sistem informasi praktikum dalam hal ini langkah-langkah yang digunakan disesuaikan dengan model *waterfall*.



Gambar 3. 1 Diagram blok metode penelitian

Pengembangan perangkat lunak dengan menggunakan model *waterfall* dibagi atas lima fase yaitu *requirements definition*, *system and software design*, *implementation and unit testing*, *integration and system testing* dan *operation and maintenance*. Pada fase *requirements definition* dilakukan pengumpulan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh sistem informasi yang akan dibangun. Fase *requirements definition* berada pada tahap analisis pelaksanaan praktikum dan analisis data pada metode penelitian, dimana pada fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

Fase *system and software design* dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap, fase ini berada pada tahap analisis kebutuhan sistem dan perancangan. Proses *software design* merupakan proses untuk mengubah kebutuhan – kebutuhan yang telah dikumpulkan pada tahap sebelumnya menjadi representasi ke dalam perangkat lunak berbentuk “*blueprint*” sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya., pada penelitian ini desain dimodelkan dengan menggunakan bahasa UML diantaranya menggunakan *use case*, *activity diagram* dan *class diagram*.

Fase *implementation and unit testing* merupakan proses dimana desain program diterjemahkan kedalam kode – kode dengan menggunakan bahasa pemrograman yang telah ditentukan, bahasa pemrograman yang digunakan pada penelitian ini adalah PHP. Fase *implementation and unit testing* berada pada tahap bab implementasi pada penelitian ini.

Fase *integration and system testing* merupakan fase penyatuan unit – unit program kemudian diuji secara keseluruhan. Pada fase ini pengujian yang dilakukan menggunakan dua metode yaitu dengan menggunakan *blackbox testing* dan *user acceptance test*.

3.1.1 Studi dan Pengkajian Literatur

Studi dan pengkajian literatur digunakan untuk mempelajari berbagai referensi yang diambil dari buku, jurnal, laporan penelitian, *website*, maupun artikel. Referensi utama yang diperlukan untuk menunjang pengembangan sistem diantaranya:

1. Model *Waterfall*
2. *CodeIgniter*
3. PHP
4. Pengolahan basis data
5. Pengujian perangkat lunak

3.1.2 Analisis Pelaksanaan Praktikum di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya

Pada tahap ini dilakukan analisis terhadap proses kegiatan praktikum yang sedang berlangsung di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Tahap analisis pelaksanaan praktikum merupakan bagian dari fase *requirements analysis and definiton* pada metode pengembangan perangkat lunak menggunakan model *waterfall*. Proses pelaksanaan praktikum pada PTIIK dilaksanakan pada setiap tahun ajaran untuk mata kuliah - mata kuliah tertentu. Kegiatan praktikum bertujuan agar mahasiswa mampu untuk mengimplementasikan dan mengembangkan ilmu yang telah diterimanya di dalam kelas. Modul kegiatan praktikum mengikuti materi mata kuliah yang telah diterima mahasiswa di kelas. Penyelenggaraan kegiatan praktikum dibimbing oleh asisten dosen dengan seorang koordinator asisten pada setiap mata kuliah yang bertugas untuk mengkoordinir para asisten dosen.

Modul praktikum diberikan dalam bentuk *hardcopy* kepada para mahasiswa praktikan, namun terkadang terdapat tambahan materi yang tidak ada pada modul *hardcopy*. Pada kasus seperti ini, biasanya asisten dosen membagikan materi tambahan tersebut kepada ketua kelas dan ketua kelas berkewajiban mendistribusikan materi tersebut kepada seluruh teman - teman di kelasnya. Pada kasus pengumpulan tugas praktikum dalam bentuk *softcopy* pun biasanya

dilakukan hal serupa, ketua kelas berkewajiban mengumpulkan tugas seluruh teman kelasnya dan kemudian mengumpulkannya kepada asisten dosen. Pada dua kasus yang dijabarkan ini dapat diambil kesimpulan bahwa perlu adanya sistem untuk mewedahi pendistribusian *file* antara praktikan dan asisten dosen.

Penilaian tugas praktikum mahasiswa pada pelaksanaan praktikum selama ini bersifat tertutup, mahasiswa hanya dapat mengetahui nilai akhir proses pembelajaran praktikum yang telah ditempuhnya melalui surat PUAS Praktikum. Surat PUAS Praktikum berisi nilai akhir praktikum mahasiswa yang diberikan pada akhir semester. Seyogyanya mahasiswa perlu tahu transparansi nilai dalam setiap tugas praktikumnya. Disamping itu, dikarenakan pelaksanaan kegiatan praktikum yang hanya melibatkan praktikan, asisten dosen dan koordinator asisten dosen sehingga dosen kurang dapat memantau dan mengetahui informasi sejauh mana praktikum berlangsung, apakah dalam pelaksanaan praktikum telah sesuai dengan apa yang diajarkan.

3.1.3 Analisis Data

Tahap analisis data merupakan fase *requirements analysis and definition* pada pengembangan perangkat lunak dengan menggunakan model *waterfall*. Pada tahap ini dilakukan pengumpulan data yang dibutuhkan untuk mencapai tujuan dari skripsi ini. Data yang diperlukan adalah data kegiatan praktikum. Data – data praktikum harus dikumpulkan secara lengkap sebelum menuju ke fase berikutnya yaitu *system and software design*. Data yang dapat diolah pada sistem yang akan dikembangkan diantaranya jadwal kegiatan praktikum, manajemen pengajian asisten dosen, file praktikum, nilai tugas praktikum, absensi praktikan, absensi asisten dosen dan koordinator asisten dosen serta berita-berita praktikum.

3.1.4 Analisis Kebutuhan Sistem

Perancangan sistem dibangun dengan menggunakan pendekatan berorientasi objek menggunakan bahasa permodelan UML berdasarkan hasil pengambilan data dan analisis kebutuhan yang telah dilakukan. Tahap analisis kebutuhan sistem merupakan bagian dari fase *system and software design* dimana

kebutuhan – kebutuhan yang telah dikumpulkan secara lengkap dimodelkan dengan menggunakan bahasa UML. *Diagram Use Case* digunakan untuk memodelkan dan menyatakan unit fungsi yang disediakan oleh sistem ke pemakai (*user*). *Activity diagram* digunakan untuk menunjukkan kegiatan yang dilakukan oleh sistem informasi yang dibangun oleh peneliti. Diagram aktivitas melibatkan sistem dan juga *user*. *Class diagram* digunakan untuk menunjukkan struktur *class* yang dibangun dengan menggunakan pola MVC.

3.1.5 Arsitektur Sistem

Arsitektur sistem pada penelitian ini menggunakan *framework CodeIgniter* yang mengadopsi *pattern MVC*. *Pattern MVC* terdiri dari tiga layer yaitu *model*, *view* dan *controller*. Layer *model* berisi fungsi untuk pengolahan basis data, *script SQL* dikelola pada layer ini. Layer *view* berisi kode untuk tampilan suatu program. Tampilan dapat berupa *web page*, *header*, *footer*, *content* dan apa saja yang berjenis tampilan. Layer *controller* berisi kode untuk *logic*, algoritma dan sebagai penghubung antara *model*, *view* dan sumber lain yang diperlukan untuk mengolah *HTTP request* dan *generate web page*.

3.1.6 Implementasi

Implementasi aplikasi akan dilakukan dengan mengacu kepada fungsionalitas sistem yang dirancang untuk membantu manajemen sistem praktikum serta akses informasi kegiatan praktikum. Tahap implementasi merupakan bagian dari fase *implementation and unit testing* pada metode pengembangan perangkat lunak dengan menggunakan model *waterfall*. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa PHP dengan bantuan *framework CodeIgniter*. Hasil yang akan diberikan dari sistem ini adalah pendokumentasian kegiatan praktikum mahasiswa yang lebih terstruktur dan terorganisir, sehingga data - data ataupun informasi yang ada dapat diakses dengan lebih fleksibel. Aplikasi didalam sistem ini dirancang untuk mengolah, menyimpan dan mendistribusikan informasi dan data praktikum. Informasi dan data praktikum diantaranya: jadwal praktikum, materi praktikum, tugas

praktikum, nilai praktikum mahasiswa serta memuat berita praktikum dan sistem penggajian asisten dosen.

3.1.7 Pengujian

Tahap pengujian merupakan fase *integration and system testing* pada metode pengembangan perangkat lunak menggunakan model *waterfall*. Pengujian perangkat lunak secara keseluruhan dilakukan dengan menggunakan metode *blackbox testing* dimana pengujian ditujukan untuk benar - benar melihat perilaku internal dan struktur sistem dan dilakukan pula pengujian tahap akhir yaitu dengan menggunakan pengujian UAT (*User Acceptance Test*). Dalam pendekatan *blackbox* yang diuji hanya berasal dari spesifikasi dengan mencoba setiap kondisi input. Pengujian dengan menggunakan metode *blackbox* memungkinkan pembuatan himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional program[8].

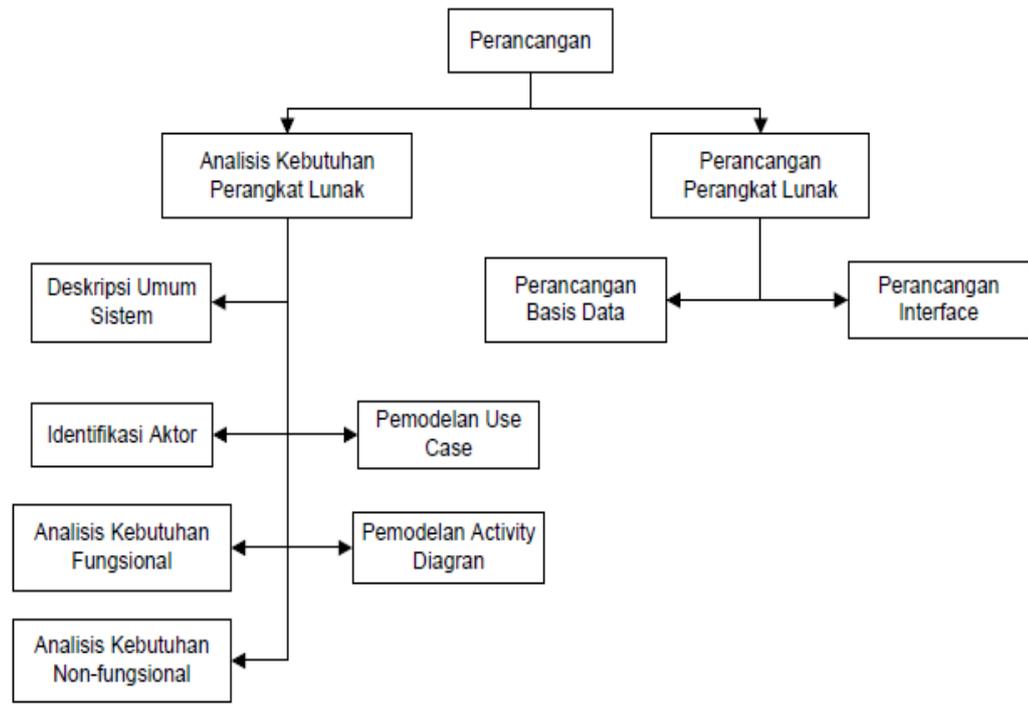
Pengujian perangkat lunak selanjutnya dilakukan melalui pengujian penerimaan pengguna menggunakan metode *User Acceptance Test* dengan pendekatan *Test Acceptance Model (TAM)*. Pendekatan TAM menerapkan sikap dan tiap-tiap perilaku pengguna dengan dua variabel, yaitu persepsi pemanfaatan (*perceived usefulness*) dan persepsi kemudahan penggunaan (*perceived ease of user*). Selain faktor persepsi pemanfaatan dan persepsi kemudahan penggunaan, pengujian juga menilai dari faktor *performance, information, satisfaction* dan *interaction*.

3.1.8 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan - kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

3.2 Perancangan

Perancangan perangkat lunak dilakukan setelah semua kebutuhan telah didapatkan dari proses analisis kebutuhan. Proses perancangan terdiri dari analisa kebutuhan perangkat lunak (*software*) / perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*) / perangkat keras (*hardware*).



Gambar 3. 2 Diagram perancangan sistem

3.2.1 Analisis Kebutuhan Perangkat Lunak

Proses analisis kebutuhan mengacu pada gambaran umum sistem perangkat lunak. Proses analisis kebutuhan ini diawali dengan penjabaran gambaran umum sistem, identifikasi aktor yang terlibat dalam sistem, analisis data yang akan disimpan, penjabaran tentang daftar kebutuhan dan kemudian memodelkannya ke dalam diagram *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan - kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna. Analisis kebutuhan harus dikumpulkan secara lengkap sebelum menuju pada tahap selanjutnya guna menghasilkan desain yang lengkap.

3.2.2 Deskripsi Umum Sistem

Aplikasi sistem informasi praktikum yang akan dibangun dikembangkan dengan bahasa pemrograman PHP dengan menggunakan teknologi *codeIgniter* yang mengadopsi *pattern* MVC. Aplikasi sistem informasi praktikum dibagi menjadi lima bagian yang didasarkan pada hak akses yang berbeda pada tiap aktor. Aktor dibagi menjadi lima yaitu praktikan, asisten dosen, koordinator asisten dosen, dosen dan *administrator*.

3.2.2.1 Fungsi Utama Perangkat Lunak

Sistem perangkat lunak yang akan dikembangkan pada penelitian ini bertujuan untuk membantu pengelolaan data dan pendistribusian berkas praktikum. Data dan informasi yang dapat dikelola dan diakses dalam sistem informasi ini secara umum adalah:

1. jadwal praktikum,
2. absensi mahasiswa dan asisten dosen,
3. modul praktikum,
4. tugas praktikum,
5. berita praktikum,
6. nilai tugas praktikum mahasiswa,
7. manajemen *user*, dan
8. pengelolaan gaji asisten dosen.

Pengembangan sistem informasi praktikum ditujukan untuk memudahkan semua pihak yang terlibat dalam kegiatan praktikum untuk mendapatkan data dan atau mengakses informasi serta mengolah data tersebut dengan lebih mudah. Informasi dan data yang dapat diakses melalui sistem informasi praktikum dibedakan atas hak akses tiap - tiap aktor. Aktor yang memiliki hak akses ke sistem antara lain *administrator*, praktikan, asisten dosen, koordinator asisten dosen dan dosen mata kuliah.

3.2.3 Identifikasi Aktor

Tahap ini adalah tahap untuk mengidentifikasi aktor-aktor yang akan berinteraksi dengan sistem informasi yang akan dibangun. Tabel 3.1 melihat aktor-aktor yang terlibat beserta penjelasannya masing-masing yang merupakan hasil dari identifikasi aktor.

Tabel 3. 1 Identifikasi aktor

Aktor	Deskripsi
Praktikan	Praktikan adalah mahasiswa yang sedang menempuh praktikum. Praktikan memiliki hak akses untuk mendapatkan informasi jadwal praktikum, nilai tugas, berita praktikum dan absensi mahasiswa serta hak akses untuk <i>update</i> profil, mengunduh modul dan tugas praktikum serta mengunggah hasil tugas praktikumnya.
Asisten Dosen	Asisten dosen memiliki hak akses ke sistem antara lain unggah tugas praktikum, unduh hasil tugas praktikum mahasiswa, manajemen profil, akses absensi praktikan, akses absensi asisten dosen, penilaian tugas praktikum mahasiswa, unggah berita praktikum dan akses informasi penggajian. Asisten dosen juga memiliki hak akses yang sama dengan praktikan, apabila asisten dosen di tahun ajaran yang sama juga berstatus sebagai praktikan pada praktikum yang lain.
Koordinator Asisten Dosen	Koordinator asisten dosen memiliki hak akses yang sama secara keseluruhan dengan asisten dosen namun memiliki tambahan hak akses untuk unggah modul praktikum, hak akses untuk melihat laporan nilai dan absensi seluruh praktikan yang mengikuti praktikum yang diempunya serta mengakses data kehadiran asisten dosen yang dikoordinir. Koordinator asisten dosen juga memiliki hak akses yang sama dengan praktikan, apabila ia di tahun ajaran yang sama

	berstatus sebagai praktikan dalam kegiatan praktikum yang lain.
<i>Administrator</i>	<i>Administrator</i> memiliki hak akses untuk mengunggah berita praktikum, mengelola akun pengguna, mengelola praktikum, mengelola jadwal praktikum, mengelola mahasiswa peserta praktikum, mengelola gaji asisten dosen dan mengelola dosen yang mengajar mata kuliah praktikum.
Dosen	Dosen memiliki hak akses untuk melihat informasi dan laporan praktikum yang ada pada sistem informasi praktikum, tanpa melakukan perubahan terhadap data-data tersebut.

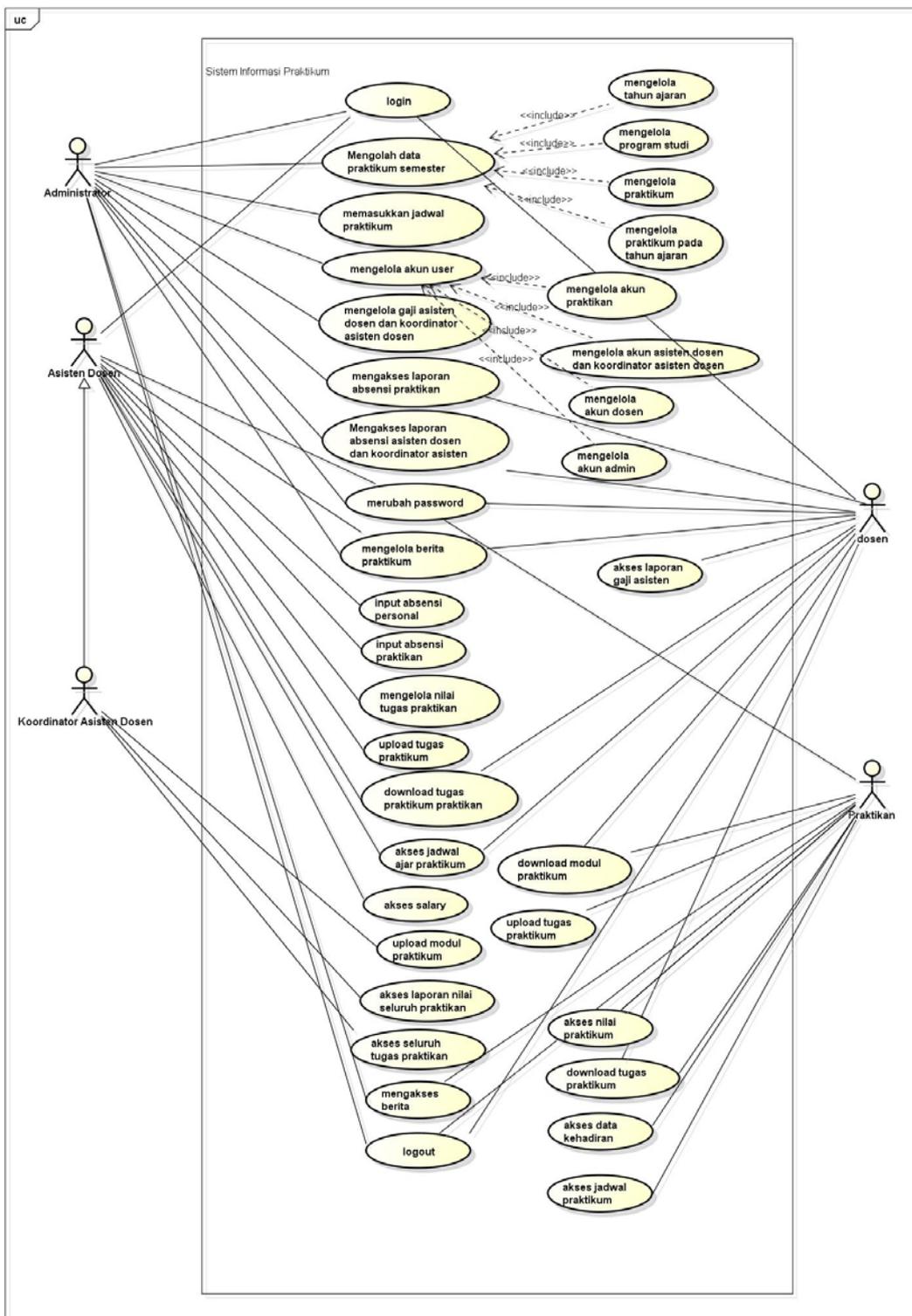
3.2.4 Identifikasi Kebutuhan dan Analisis *Use Case*

Pada tahap ini dilakukan analisis dan perancangan kebutuhan sistem. Daftar kebutuhan sistem terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Pada daftar kebutuhan akan dispesifikasikan menjadi lima bagian dimana spesifikasi kebutuhan akan dijabarkan per aktor sistem.

Diagram *use case* adalah salah satu diagram untuk memodelkan aspek perilaku sistem. Diagram *use case* menunjukkan sekumpulan *use case*, aktor dan hubungannya. *Use case* merupakan fungsionalitas dari sistem yang diinisialisasi oleh aktor.

Tahap pemodelan dengan menggunakan diagram *use case* merupakan bagian dari fase *system and software design* pada metode pengembangan perangkat lunak menggunakan model *waterfall*, fase ini dikerjakan setelah semua kebutuhan telah dikumpulkan secara lengkap serta dianalisa dan didefinisikan kebutuhan yang harus dipenuhi oleh sistem informasi yang akan dibangun.

Gambar 3.3 adalah diagram *use case* sistem informasi praktikum.



Gambar 3. 3 Diagram Use Case Sistem Informasi Praktikum

Spesifikasi kebutuhan fungsional sistem informasi praktikum untuk praktikan ditunjukkan pada Tabel 3.2.

Tabel 3. 2 Spesifikasi kebutuhan fungsional aktor praktikan

Nomor SRS	Kebutuhan	Use Case
SRS_001_01	Sistem harus menyediakan fasilitas untuk <i>login</i> dimana berfungsi untuk membedakan hak akses tiap <i>aktor</i> .	<i>Login</i>
SRS_001_02	Sistem harus memberikan daftar praktikum apa saja yang sedang ditempuh <i>user</i> (pengguna) tersebut.	Akses Praktikum
SRS_001_03	Sistem harus menyediakan fasilitas agar <i>user</i> dapat mengakses jadwal praktikum	Akses jadwal praktikum
SRS_001_04	Sistem harus menyediakan fasilitas untuk mengunduh modul praktikum.	Akses dan unduh modul praktikum
SRS_001_05	Sistem harus menyediakan fasilitas untuk melihat informasi absensi mahasiswa.	Akses Absensi
SRS_001_06	Sistem harus menyediakan fasilitas untuk <i>update password</i> .	Manajemen akun pengguna
SRS_001_07	Sistem harus menyediakan fasilitas untuk mengunggah hasil tugas praktikum.	Unggah tugas
SRS_001_08	Sistem harus menyediakan fasilitas untuk akses informasi nilai mahasiswa.	Akses nilai
SRS_001_09	Sistem harus menyediakan fasilitas untuk akses berita praktikum.	Akses berita praktikum
SRS_001_10	Sistem harus menyediakan fasilitas untuk keluar dari hak akses <i>user</i> .	<i>Logout</i>

Spesifikasi kebutuhan fungsional sistem informasi praktikum untuk *administrator* ditunjukkan pada Tabel 3.3.

Tabel 3. 3 Spesifikasi kebutuhan fungsional aktor *administrator*

Nomor SRS	Kebutuhan	Use Case
SRS_002_01	Sistem harus menyediakan fasilitas untuk <i>login</i> yang berfungsi untuk membedakan hak akses tiap aktor.	<i>Login</i>
SRS_002_02	Sistem harus menyediakan fasilitas untuk mengolah data <i>administrator</i> dan data semua pengguna sistem.	Mengelola <i>user account</i>
SRS_002_03	Sistem harus menyediakan fasilitas untuk mengelola praktikum.	Mengelola data praktikum
SRS_002_04	Sistem harus menyediakan fasilitas untuk mengelola jadwal praktikum.	Mengelola jadwal praktikum
SRS_002_05	Sistem harus menyediakan fasilitas untuk mengelola gaji asisten dosen.	Mengelola gaji asisten dosen

SRS_002_06	Sistem harus menyediakan fasilitas untuk unggah berita / informasi praktikum.	<i>News feed</i>
SRS_002_07	Sistem harus menyediakan fasilitas untuk <i>update password user</i> .	Manajemen akun pengguna
SRS_002_08	Sistem harus menyediakan fasilitas untuk mengakses laporan absensi mahasiswa.	Mengakses absensi mahasiswa
SRS_002_09	Sistem harus menyediakan fasilitas untuk mengakses laporan absensi asisten dosen dan koordinator asisten dosen.	Mengakses absensi asisten dosen dan koordinator asisten dosen
SRS_002_10	Sistem harus menyediakan fasilitas untuk menambah dan menghapus <i>administrator</i> .	Mengolah data <i>administrator</i>
SRS_002_11	Sistem harus menyediakan operasi untuk keluar dari hak akses <i>user</i> .	<i>Logout</i>

Spesifikasi kebutuhan fungsional sistem informasi praktikum untuk asisten dosen dan koordinator asisten dosen ditunjukkan pada Tabel 3.4.

Tabel 3. 4 Spesifikasi kebutuhan fungsional aktor asisten dosen dan koordinator asisten dosen

Nomor SRS	Kebutuhan	<i>Use Case</i>
SRS_003_01	Sistem harus menyediakan fasilitas untuk <i>login</i> yang berfungsi untuk masuk kedalam sistem dan membedakan hak akses tiap aktor.	<i>Login</i>
SRS_003_02	Sistem harus menyediakan fasilitas untuk akses informasi jadwal mengajar asisten dosen.	Mengakses jadwal mengajar
SRS_003_03	Sistem harus menyediakan fasilitas kepada asisten dosen untuk mengunggah dan mengolah berita yang telah diunggah.	Mengelola berita praktikum
SRS_003_04	Sistem harus menyediakan fasilitas kepada asisten dosen untuk mengunduh modul praktikum.	Mengunduh modul praktikum.
SRS_003_05	Sistem harus menyediakan fasilitas kepada asisten dosen untuk input absensi mahasiswa.	Mengelola absensi MHS
SRS_003_06	Sistem harus menyediakan fasilitas untuk input nilai tugas praktikum mahasiswa.	Input nilai praktikum mahasiswa
SRS_003_07	Sistem harus menyediakan fasilitas kepada asisten dosen untuk mengunduh tugas praktikum mahasiswa.	Mengunduh tugas praktikum MHS

SRS_003_08	Sistem harus menyediakan fasilitas untuk input absensi asisten dosen.	<i>Input absensi</i>
SRS_003_09	Sistem harus menyediakan fasilitas kepada koordinator asisten dosen untuk mengakses laporan nilai tugas mahasiswa.	Mengakses laporan nilai praktikum MHS
SRS_003_10	Sistem harus memberikan fasilitas kepada koordinator asisten dosen untuk mengunggah modul praktikum.	Mengunggah modul praktikum
SRS_003_11	Sistem harus memberikan fasilitas kepada koordinator asisten dosen untuk mengakses laporan absensi mahasiswa.	Akses laporan absensi mahasiswa
SRS_003_12	Sistem harus memberikan fasilitas kepada koordinator asisten dosen untuk mengakses laporan absensi asisten dosen.	Akses laporan absensi asisten dosen.
SRS_003_13	Sistem harus memberikan fasilitas untuk asisten dosen dan koordinator asisten untuk mengakses informasi gaji.	Akses <i>Penggajian</i> .
SRS_003_14	Perangkat lunak harus memberikan fasilitas untuk keluar dari hak akses pengguna	<i>Logout</i>

Spesifikasi kebutuhan fungsional sistem informasi praktikum untuk dosen praktikan ditunjukkan pada Tabel 3.5.

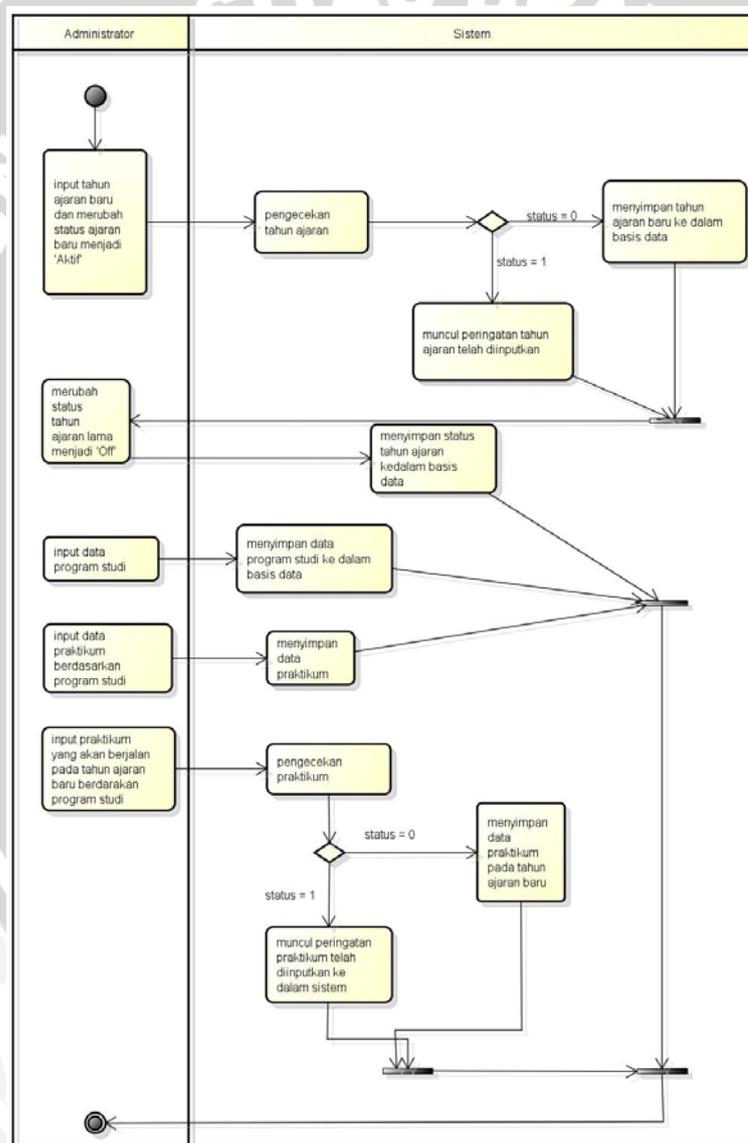
Tabel 3. 5 Spesifikasi kebutuhan fungsional aktor dosen

Nomor SRS	Kebutuhan	<i>Use Case</i>
SRS_004_01	Sistem harus menyediakan fasilitas untuk <i>login</i> yang berfungsi untuk masuk kedalam sistem dan membedakan hak akses tiap aktor.	<i>Login</i>
SRS_004_02	Sistem harus menyediakan fasilitas untuk mengakses jadwal praktikum	Mengakses jadwal praktikum
SRS_004_03	Sistem harus menyediakan fasilitas untuk mengakses materi (modul) yang diajarkan asisten dosen.	Mengakses informasi materi
SRS_004_04	Sistem harus menyediakan fasilitas untuk mengakses laporan absensi mahasiswa dan asisten dosen.	Mengakses data kehadiran mahasiswa dan asisten dosen
SRS_004_05	Sistem harus menyediakan fasilitas untuk mengakses laporan nilai tugas praktikum mahasiswa.	Mengakses nilai mahasiswa
SRS_004_06	Sistem harus menyediakan fasilitas untuk mengakses berita praktikum.	Mengakses <i>News feed</i>
SRS_004_07	Sistem harus menyediakan fasilitas untuk keluar dari hak akses pengguna.	<i>Logout</i>

3.2.4.1 Diagram Aktivitas Sistem

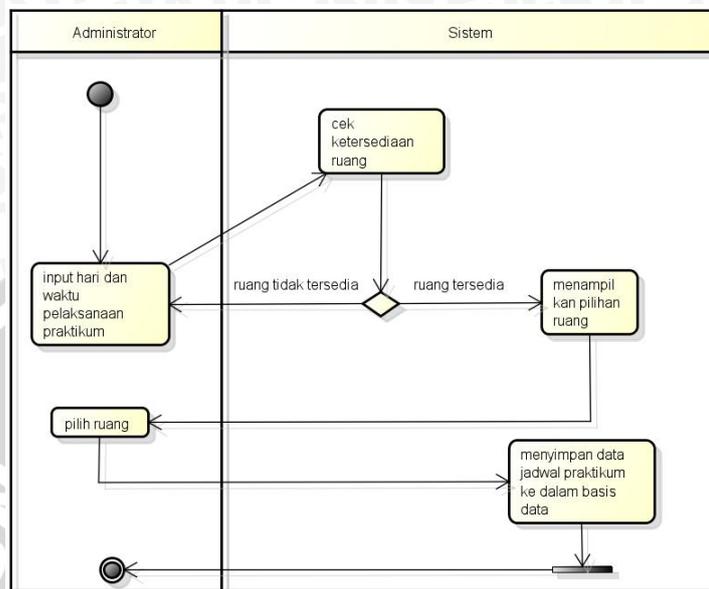
Diagram aktivitas sistem merupakan diagram yang menunjukkan kegiatan yang dilakukan oleh sistem informasi yang dibangun oleh peneliti. Diagram aktivitas melibatkan sistem dan juga *user* yang terbagi atas 5 aktor. *User* yang terlibat dalam aktivitas sistem antara lain *administrator*, asisten dosen, koordinator asisten dosen, dosen, dan praktikan.

Gambar 3.4 menunjukkan aktifitas administrator mengelola praktikum yang akan berlangsung.



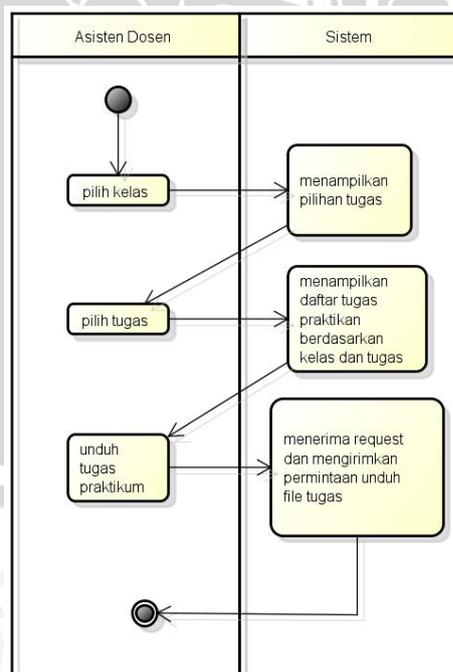
Gambar 3. 4 Activity Diagram Pengelolaan Praktikum

Gambar 3.5 menunjukkan aktifitas administrator dalam mengelola jadwal praktikum.



Gambar 3.5 Activity Diagram Pengelolaan Jadwal Praktikum

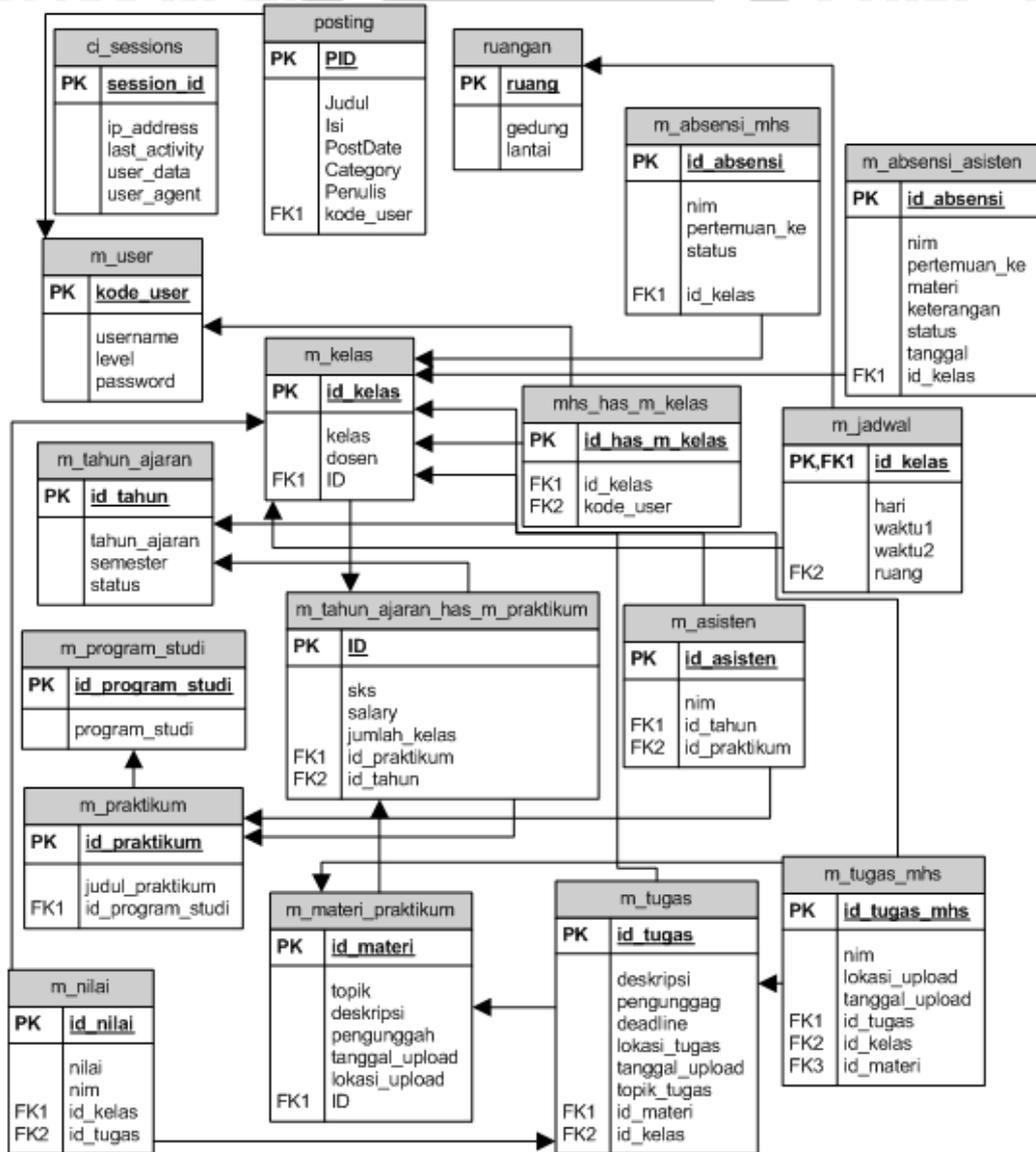
Gambar 3.6 menunjukkan aktifitas asisten dosen dalam mengunduh tugas praktikum praktikan.



Gambar 3.6 Activity Diagram Unduh Tugas Praktikum Praktikan

3.2.5 Perancangan Basis Data

Perancangan basis data sistem digunakan untuk memodelkan struktur basis data yang akan digunakan dalam pengembangan sistem informasi yang akan dibangun. Perancangan basis data disesuaikan dengan kebutuhan akan data – data yang akan disimpan dan diolah oleh sistem.



Gambar 3. 7 Rancangan Database System.

3.2.5.1 Tabel m_user

Tabel m_user digunakan untuk menyimpan *username*, *password* dan level *user* (aktor) dari data pengguna yang nantinya memiliki hak akses ke sistem. Atribut dan tipe data yang digunakan pada tabel m_user ditunjukkan Tabel 3.6.

Tabel 3. 6 Detail tabel m_user

Name	Type
kode_user	varchar(15)
username	varchar(30)
level	varchar(15)
password	varchar(15)
status	int(1)

3.2.5.2 Tabel m_tahun_ajaran

Tabel m_tahun_ajaran digunakan untuk menyimpan data tentang tahun ajaran. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada Tabel 3.7.

Tabel 3. 7 Detail tabel m_tahun_ajaran

Name	Type
id_tahun	varchar(255)
semester	varchar(255)
tahun_ajaran	varchar(255)
status	varchar(10)

3.2.5.3 Tabel m_program_studi

Tabel m_program_studi berisi data program studi apa saja yang dimiliki Program Teknologi Informasi dan Ilmu Komputer. Tabel m_program_studi digunakan untuk mengelompokkan mata kuliah berdasarkan pada program studinya. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada Tabel 3.8.

Tabel 3. 8 Detail tabel m_program_studi

Name	Type
id_program	varchar(255)
program_studi	varchar(255)

3.2.5.4 Tabel m_praktikum

Tabel m_praktikum berisi data praktikum (mata kuliah). Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada Tabel 3.9.

Tabel 3. 9 Detail tabel m_praktikum

Name	Type
id_praktikum	varchar(255)
judul_praktikum	varchar(255)
id_program_studi	varchar(255)

3.2.5.5 Tabel m_tahun_ajaran_has_m_praktikum

Tabel m_tahun_ajaran_has_m_praktikum digunakan untuk menyimpan dan mendapatkan data tentang praktikum(mata kuliah) apa saja yang sedang berlangsung pada tahun ajaran tertentu. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada Tabel 3.10.

Tabel 3. 10 Detail tabel m_tahun_ajaran_has_m_praktikum

Name	Type
id	int(11)
id_tahun	varchar(255)
id_praktikum	varchar(255)
sks	int(255)
salary	double

3.2.5.6 Tabel m_kelas

Tabel m_kelas berisi data tentang kelas praktikum yang didasarkan pada mata kuliah di tahun ajaran tertentu. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada tabel 3.11.

Tabel 3. 11 Detail tabel m_kelas

Name	Type
id_kelas	int(255)
id_tahun_praktikum	int(255)
kelas	varchar(255)

3.2.5.7 Tabel m_jadwal

Tabel m_jadwal berisi data jadwal praktikum yang didasarkan pada kelas praktikum. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada Tabel 3.12.

Tabel 3. 12 Detail tabel m_jadwal

Name	Type
id_kelas	int(255)
hari	varchar(255)
waktu1	time
waktu2	time

3.2.5.8 Tabel m_asisten

Tabel m_asisten berisi data mahasiswa yang bertindak sebagai asisten dosen pada praktikum mata kuliah di tahun ajaran tertentu. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada Tabel 3.13.

Tabel 3. 13 Detail tabel m_asisten

Name	Type
nim	varchar(255)
id_tahun_ajaran	int(255)
id_kelas	int(255)
id_praktikum	varchar(255)

3.2.5.9 Tabel m_materi_praktikum

Tabel m_materi_praktikum berisi data materi (modul) praktikum yang disimpan berdasarkan mata kuliah dan tahun ajaran. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada Tabel 3.14.

Tabel 3. 14 Detail tabel m_materi_praktikum

Name	Type
id_materi	varchar(255)
id_tahun_ajaran_praktikum	int(255)
Topik	varchar(255)
deskripsi	text
pengunggah	varchar(255)
tanggal_upload	datetime
lokasi_upload	varchar(255)

3.2.5.10 Tabel mhs_has_m_kelas

Tabel mhs_has_m_kelas berisi data mahasiswa yang berada pada kelas praktikum berdasarkan mata kuliah dan tahun ajaran. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada Tabel 3.15.

Tabel 3. 15 Detail tabel mhs_has_m_kelas

Name	Type
id_has_m_kelas	int(11)
nim	varchar(255)
id_kelas	int(255)

3.2.5.11 Tabel m_absensi_asdos

Tabel m_absensi_asdos digunakan untuk menyimpan data kehadiran asisten dosen yang didasarkan pada mata kuliah dan tahun ajaran. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada tabel 3.16.

Tabel 3. 16 Detail tabel m_absensi_asdos

Name	Type
id_absensi	int(255)
nim	varchar(255)
pertemuan_ke	int(20)
materi	varchar(255)
keterangan	varvhar(255)
status	tinyint(1)
id_kelas	int(255)
tanggal	datetime

3.2.5.12 Tabel m_absensi_mhs

Tabel m_absensi_mhs digunakan untuk menyimpan data kehadiran asisten dosen yang didasarkan pada kelas, mata kuliah dan tahun ajaran. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada tabel 3.17.

Tabel 3. 17 Detail tabel m_absensi_mhs

Name	Type
id_absensi	int(11)
nim	varchar(255)
pertemuan_ke	int(255)
status	tinyint(1)
id_kelas	int(255)

3.2.5.13 Tabel posting

Tabel posting digunakan untuk menyimpan berita praktikum yang di *post*-kan oleh asisten dosen atau *administrator*. Atribut dan tipe data yang digunakan pada tabel m_tahun_ajaran ditunjukkan pada tabel 3.18.

Tabel 3. 18 Detail tabel posting

Name	Type
pid	int(5)
judul	varchar(255)
isi	text
penulis	varchar(10)
postdate	Datetime

3.2.5.14 Tabel ruang

Tabel ruang berisi data ruang kelas yang dimiliki Program Teknologi Informasi dan Ilmu Komputer. Atribut dan tipe data yang digunakan pada tabel `m_tahun_ajaran` ditunjukkan pada tabel 3.19.

Tabel 3. 19 Detail tabel ruangan

Name	Type
ruang	varchar(255)
gedung	varchar(255)
lantai	varchar(255)

3.2.5.15 Tabel ci_sessions

Tabel `ci_sessions` berisi data tentang *sessions* / aktivitas *user*. Atribut dan tipe data yang digunakan pada tabel `m_tahun_ajaran` ditunjukkan pada Tabel 3.20.

Tabel 3. 20 Detail tabel ci_sessions

Name	Type
session_id	varchar(40)
ip_address	varchar(16)
last_activity	int(10)
userdata	text
user_agent	varchar(120)

3.2.5.16 Tabel m_tugas

Tabel `m_tugas` berisi data tentang tugas mahasiswa. Atribut dan tipe data yang digunakan pada tabel `m_tugas` ditunjukkan pada Tabel 3.21.

Tabel 3. 21 Detail tabel m_tugas

Name	Type
id_tugas	int(11)
id_kelas	int(255)
id_materi	int(11)
deskripsi	varchar(255)
pengunggah	varchar(255)
deadline	datetime

lokasi_tugas	varchar(255)
tanggal_upload	datetime

3.2.5.17 Tabel m_tugas_mhs

Tabel m_tugas_mhs berisi data - data pengumpulan tugas praktikum mahasiswa. Atribut dan tipe data yang digunakan pada tabel m_tugas_mhs ditunjukkan pada Tabel 3.22.

Tabel 3. 22 Detail tabel m_tugas_mhs

Name	Type
id_tugas_mhs	int(11)
id_kelas	int(255)
nim	varchar(255)
id_materi	int(255)
lokasi_upload	varchar(255)
id_tugas	int(11)
tanggal_upload	datetime

3.2.5.18 Tabel m_nilai

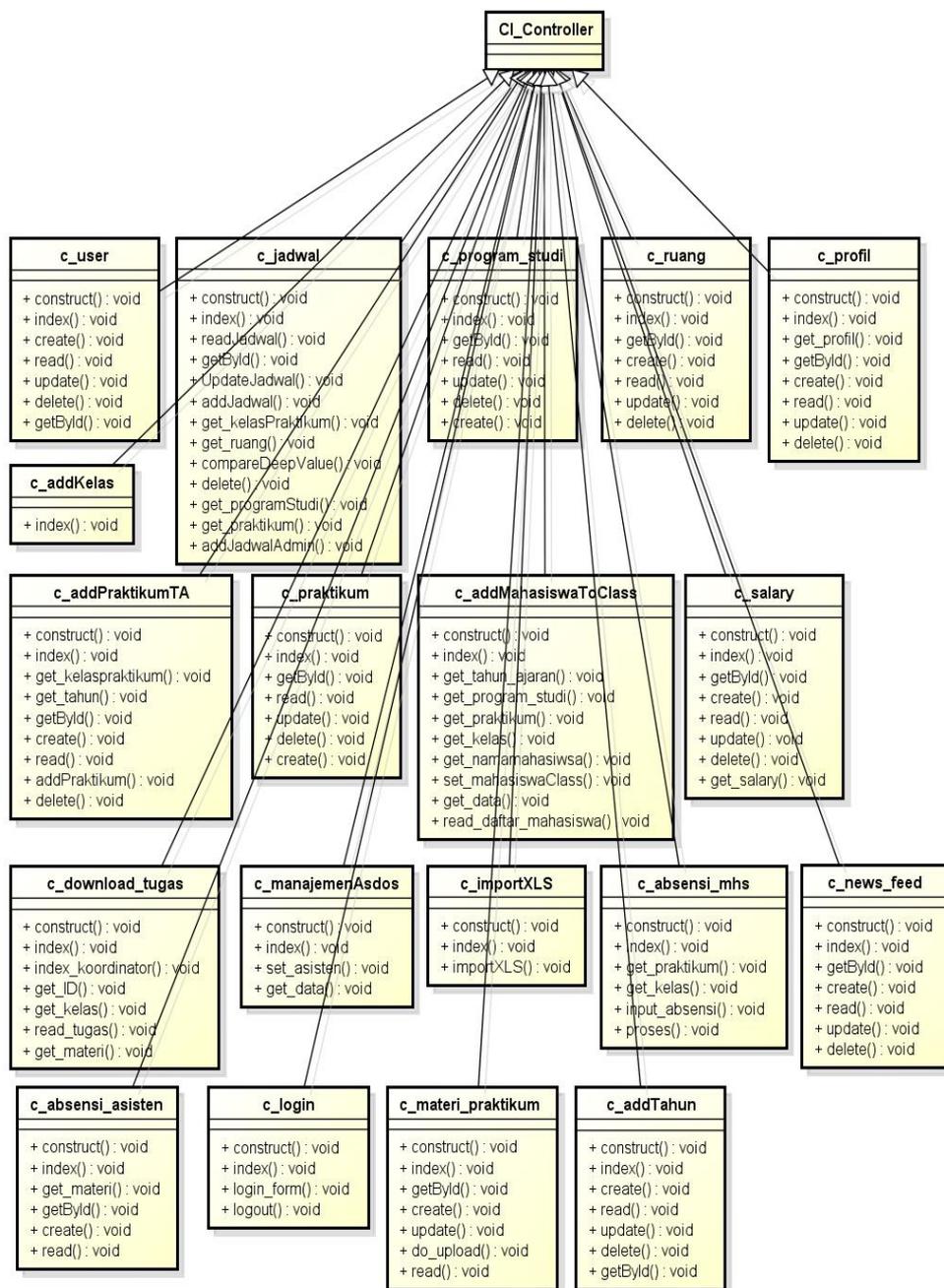
Tabel m_nilai berisi nilai tugas praktikum mahasiswa. Atribut dan tipe data yang digunakan pada tabel m_nilai ditunjukkan pada Tabel 3.23.

Tabel 3. 23 Detail tabel m_nilai

Name	Type
id_nilai	int(11)
id_kelas	int(255)
id_tugas	int(11)
nilai	float
nim	varchar(255)

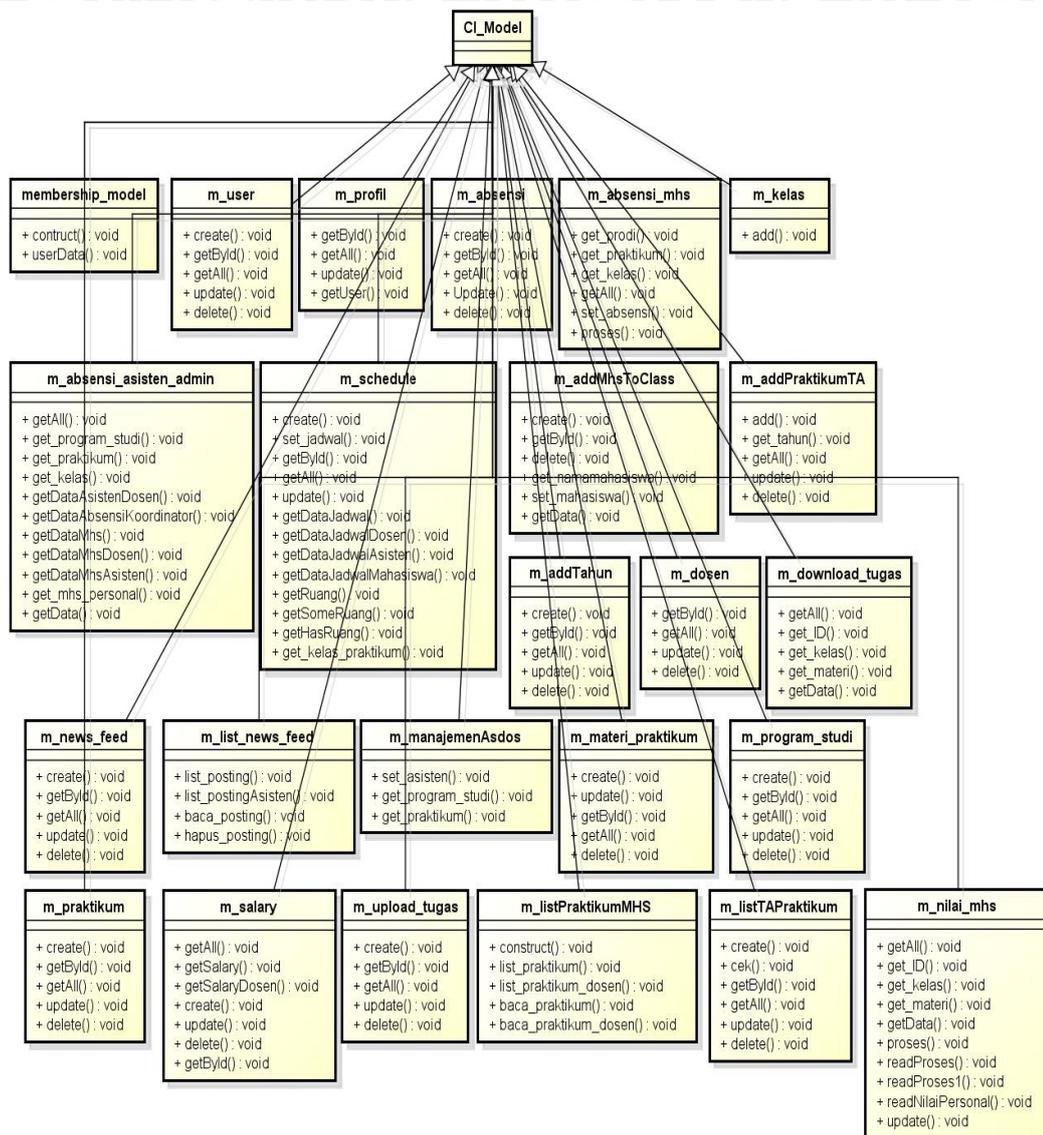
3.2.6 Perancangan Diagram Class

Diagram kelas memberikan gambaran tentang sistem dan relasi - relasi yang terdapat di dalamnya. Kelas - kelas yang telah teridentifikasi dapat memiliki hubungan antar kelas. Kelas-kelas juga dapat memiliki pewarisan dan ketergantungan terhadap komponen - komponen yang telah ditentukan. Gambar 3.8 menunjukkan diagram kelas untuk *controller* pada sistem informasi praktikum.



Gambar 3. 8 Diagram Class Untuk Controller Pada Sistem Informasi Praktikum

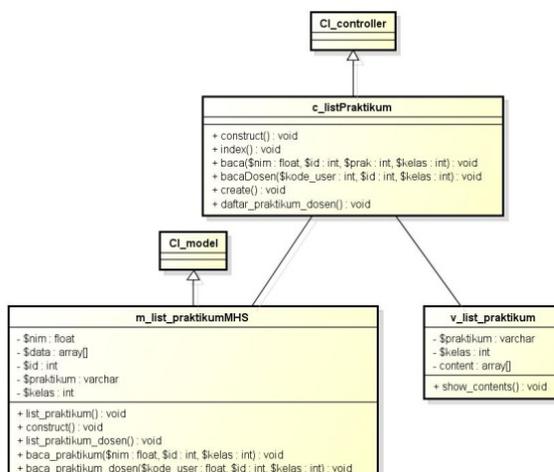
Gambar 3.9 menunjukkan diagram kelas untuk *model* pada sistem informasi praktikum.



Gambar 3. 9 Diagram Class Untuk Model Pada Sistem Informasi Praktikum

Semua kelas pada sistem mengalami pewarisan dari kelas Controller pada *Code Igniter framework*. Pewarisan pada tiap kelas dikarenakan kelas-kelas tersebut merupakan penambahan dari *controller-controller* pada sistem agar dapat beroperasi sesuai dengan spesifikasi kebutuhan.

Gambar 3.10 menunjukkan diagram kelas untuk subsistem daftar praktikum yang ditempuh praktikan.



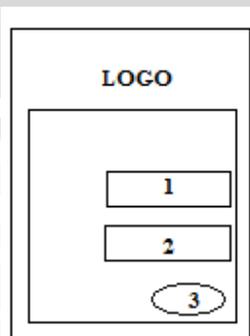
Gambar 3. 10 Diagram Class Untuk Subsistem Daftar Praktikum

3.2.7 Perancangan Interface

Interface atau view berada pada sisi *client*, yaitu berupa *web application*. Interface dirancang sesuai *platform client* dengan luas layar 1024 x 680 pixel. *User Interface* merupakan hal penting dalam pembuatan aplikasi untuk menampilkan tatap muka aplikasi yang dikembangkan. Perancangan *interface* dibuat sebaik-baiknya untuk memenuhi aspek *interaction* sehingga sistem menjadi *user friendly* dan pengguna (*user*) dapat dengan mudah mengoperasikan sistem.

Rancangan *interface* yang dibuat pada sistem informasi praktikum di Program Teknologi Informasi dan Ilmu Komputer universitas Brawijaya ditunjukkan oleh Gambar 3.11 dan 3.12. Gambar 3.11 menunjukkan rancangan *interface* halaman login dan Gambar 3.12 menunjukkan rancangan *interface* untuk konten sistem informasi praktikum.

a. Desain halaman Login

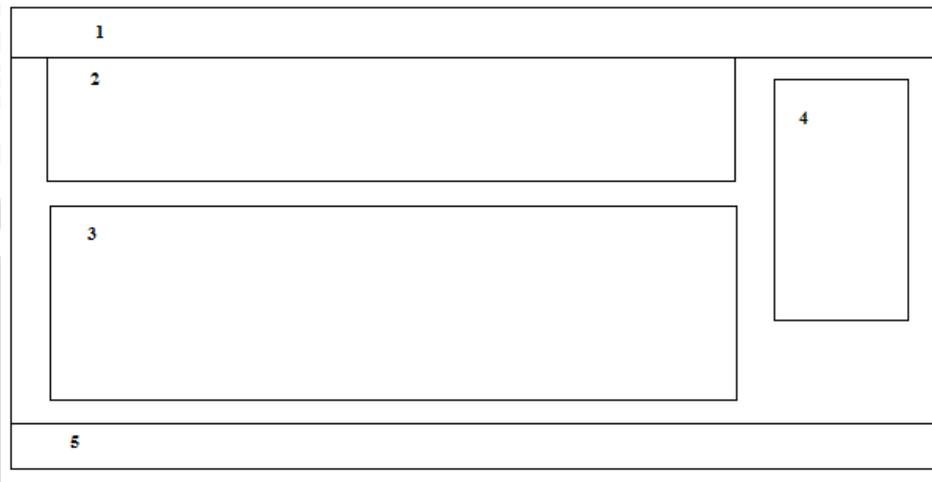


Gambar 3. 11 Perancangan interface halaman login

Keterangan:

1. *Textfield username*
2. *Textfield password*
3. Button submit

b. Desain halaman aplikasi



Gambar 3. 12 Perancangan *interface* konten sistem informasi

Keterangan :

1. Navigasi atas.
2. *Header*.
3. Konten.
4. Navigasi menu.
5. *Footer*.

3.2.8 Pengujian Perangkat Lunak

Pada tahap ini dilakukan uji coba sistem baru yang telah dibuat yang terdiri dari *black-box testing* (pengujian validitas) dan pengujian menggunakan metode *user acceptance test*. *User acceptance test* menggunakan pendekatan TAM (*Technology Acceptance Model*) dimana menerapkan sikap dan tiap-tiap perilaku pengguna dengan dua variabel, yaitu persepsi pemanfaatan (*perceived usefulness*) dan persepsi kemudahan penggunaan (*perceived ease of user*). Selain faktor persepsi pemanfaatan dan persepsi kemudahan penggunaan, pengujian juga menilai dari faktor *performance*, *information*, *satisfaction* dan *interaction*.

Beberapa pertanyaan yang mewakili faktor-faktor tersebut ditunjukkan pada Tabel 3.24.

Tabel 3. 24 Kuisisioner berdasarkan beberapa faktor

No	Pertanyaan	Aspek
1	Sistem informasi praktikum mudah untuk dioperasikan?	<i>Interaction</i>
2	Susunan menu pada sistem ini mudah untuk dipahami?	<i>Interaction</i>
3	Sistem memiliki tampilan yang menarik?	<i>Interaction</i>
4	Sistem memiliki navigasi cepat untuk pindah ke halaman lain?	<i>Performance</i>
5	Sistem menyediakan akses cepat dan mudah untuk mencari informasi?	Persepsi kemudahan penggunaan (<i>perceive ease of use</i>)
6	Sistem menyediakan informasi yang relevan?	<i>Information</i>
7	Sistem menyediakan konten informasi yang mudah dibaca?	<i>Information</i>
8	Sistem dapat mengelola data praktikum dengan baik?	Persepsi pemanfaatan (<i>perceived usefulness</i>)
9	Sistem dapat membantu pengguna dalam kegiatan praktikum?	Persepsi pemanfaatan (<i>perceived usefulness</i>)
10	Pengguna puas dengan kinerja sistem yang telah dibangun?	<i>Satisfaction</i>

BAB IV IMPLEMENTASI

Bab ini membahas mengenai tahapan implementasi perangkat lunak Sistem Informasi Praktikum berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak. Pembahasan terdiri atas penjelasan tentang spesifikasi sistem, batasan – batasan dalam implementasi, implementasi basis data dan implementasi antarmuka.

4.1 Spesifikasi Lingkungan Implementasi Sistem.

Aplikasi ini dibangun berdasarkan spesifikasi *hardware* dan *software* berikut:

Tabel 4. 1 Tabel spesifikasi *hardware* implementasi sistem

Notebook Toshiba Portege R-705	
Proessor	Intel i-5 2.53ghz
RAM	4 Gb
Hardisk	500 Gb

Tabel 4. 2 Spesifikasi *software* implementasi sistem

Operating System Windows 7	
SDK server	<i>CodeIgnitter</i> <i>Language: php, Ajax, JQuery</i>
DBMS	MySQL
Server HTTP	Apache

4.2 Batasan Implementasi

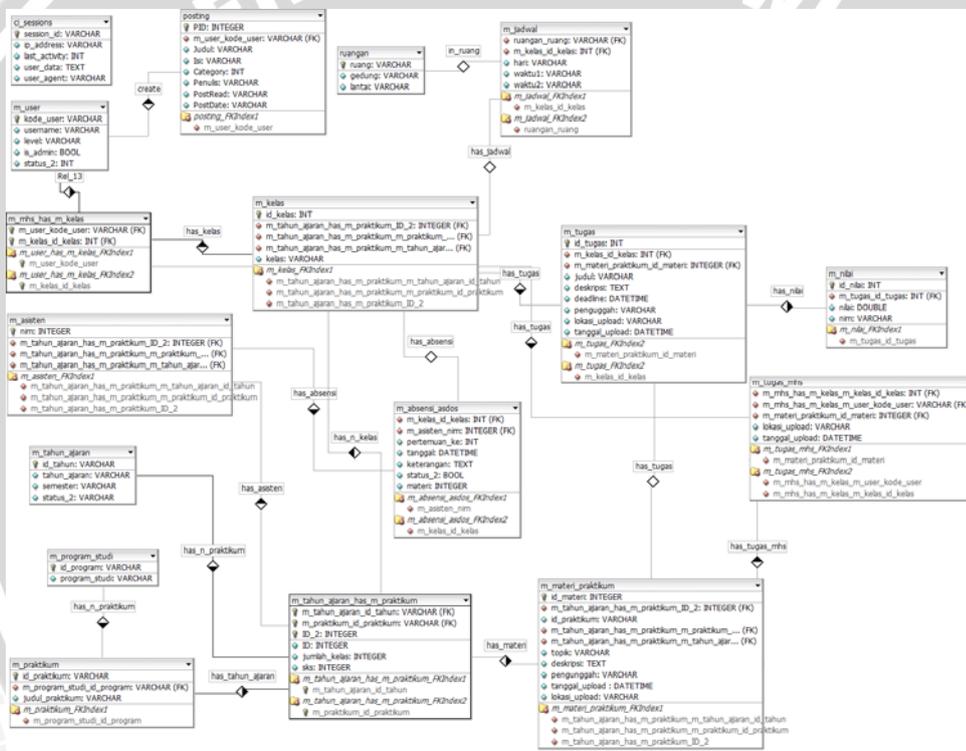
Batasan dalam implementasi aplikasi sebagai berikut :

- *Database Manajement System* yang digunakan adalah MySQL.
- Perangkat lunak ini dibangun dengan menggunakan teknologi *Framework CodeIgnitter*
- Aplikasi berbasis web dan dijalankan pada *web browser* Chrome, Opera, Internet Explorer atau Mozilla Firefox.

- Server yang digunakan dalam pengembangan aplikasi adalah Apache Server.

4.3 Implementasi Basis data

Implementasi penyimpanan data dilakukan dengan *database management system* MySQL. Hasil implementasi penyimpanan data ini berupa *script – script* SQL. Hasil implementasi SQL pada *database* ini dimodelkan dalam diagram *physical*. Gambar 4.1 menggambarkan diagram ER konseptual dari aplikasi sistem informasi praktikum.



Gambar 4. 1 Diagram ER konseptual dari sistem

4.4 Implementasi Komponen

4.4.1 Implementasi Komponen Manajemen Praktikum

Halaman manajemen praktikum digunakan untuk mengelola praktikum yang akan dilaksanakan pada tahun ajaran baru. Tabel berikut merupakan cuplikan *code* dari *Controller* `c_listTAPraktikum` yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4.3 *Controller c_listTAPraktikum*

```

1 class c_listTAPraktikum extends CI_Controller {
2   ...
3
4   public function index()
5   {
6     $user = $this->session->userdata('kode_user');
7     $data['user'] = $this->membership_model->userData($user);
8     $this->load->view( 'admin/v_listTAPraktikum', $data );
9   }
10
11   public function create() {
12     if(!empty( $_POST)){
13       $this->m_listTAPraktikum->create();
14     }
15   }
16   public function get_kelas_praktikum() {
17
18     $posts = $this->M_listTAPraktikum->get_kelas_praktikum();
19     if(isset($posts[0])){
20       echo "error";
21       return;
22     }else{
23       echo "success";
24       return;
25     }
26   }
27 }

```

Penjelasan implementasi dari Tabel 4.3 adalah :

1. Baris 4-9 adalah fungsi `index`, dimana merupakan fungsi pertama yang akan dieksekusi ketika *controller* dipanggil.
2. Baris 6-7 berfungsi untuk mengambil session berupa kode user (nim) dari pengguna.
3. Baris 8 berfungsi untuk memanggil *view* `v_listTahunAjaranPraktikum`.
4. Baris 11-15 berfungsi untuk menambahkan data praktikum yang akan berlangsung pada tahun ajaran dengan status “Aktif” ke dalam basis data.
5. Baris 16-25 berfungsi untuk memanggil fungsi `get_kelas_praktikum` pada model `m_listTAPraktikum`.

Tabel 4.4 berikut merupakan cuplikan *code* dari *Model* `m_listTAPraktikum` yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4.4 Model `m_listTAPraktikum`

```

1 class m_listTAjaranPraktikum extends CI_Model {
2   ...
3   public function create() {
4     $data = array(

```

```

5      'id_tahun' => $this->input->post( 'cTahunAjaran', true ),
6
7      'id_praktikum' => $this->input->post( 'praktikum', true ),
8
9      'sks' => $this->input->post( 'cSks', true ), );
10
11     $this->db->insert( 'm_tahun_ajaran_has_m_praktikum', $data );
12     return $this->db->insert_id();
13     }
14
15     function get_kelas_praktikum ( )
16     {
17
18         $prodi = $this->input->post( 'prodi', true );
19         $matkul = $this->input->post( 'praktikum', true );
20         $result_data = $this->db->query("SELECT * from
21 m_tahun_ajaran_has_m_praktikum , m_tahun_ajaran
22 WHERE          id_praktikum=' $matkul'          and
23 m_tahun_ajaran_has_m_praktikum.id_tahun=
24 m_tahun_ajaran.id_tahun and m_tahun_ajaran.status='Aktif'");
25         $hasil = $result_data->result_array();
26         // $this->db->close();
27         return $hasil;
28     }
29     ...
30 }

```

Penjelasan implementasi dari Tabel 4.4 adalah :

1. Baris 3-13 adalah fungsi **create** yang digunakan untuk memasukkan data praktikum ke dalam basis data.
2. Baris 15-28 adalah fungsi **get_kelas_praktikum** yang digunakan untuk melakukan pengecekan data praktikum pada tabel `m_tahun_ajaran_has_m_praktikum` untuk mengetahui apakah data praktikum yang akan diinputkan tidak sama dengan data sebelumnya yang telah diinputkan.

Tabel 4.5 berikut merupakan cuplikan *code* dari *view* `v_listTAPraktikum` yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4. 5 View `v_listTAPraktikum`

```

1      ...
2      function fungpraktikum( )
3      {
4      var formDataa = "&prodi="+document.getElementById
5      ('prodi').value+"&praktikum="+document.getElementById('prakti
6      kum').value;
7      $.ajax({
8      url:'<?php          echo          base_url()>index.php/admin/
9      c_listTAPraktikum/get_kelas_praktikum',
10     type: 'POST',
11     data: formDataa,
12     success:function(data) {

```

```

13         if(data == "error"){
14     $( '#msgDialog > p' ).html("Warning : Praktikum sudah di tambahkan
15     kelasnya");
16     $( '#msgDialog' ).dialog( 'option', 'title', 'Success' ).dialog(
17     'open' );
18     } else{
19         $('#cSks').prop('disabled', false);
20         }} //end success
21     }); }
22 </script>
23 ...
24     <div id="create">
25         <form action="" method="post">
26             Tahun Ajaran:
27             Program Studi
28                 <select onChange="get_praktikum()" id="prodi"
29     name="prodi">
30                 <option>--Pilih Prodi--</option>
31                 </select>
32             Matakuliah
33                 <select onChange="fungpraktikum()" id="praktikum"
34     name="praktikum">
35                 <option>--Pilih Matakuliah--</option>
36                 </select>
37             SKS
38                 <input type="text" name="cSks" id="cSks" disabled>
39                 <input type="submit" name="createSubmit" value="Submit" />
40             </form>
41         </div>
42     ...

```

Penjelasan implementasi dari tabel 4.5 adalah :

1. Baris 2-22 adalah *javascript* yang akan dipanggil ketika *user* selesai menginputkan kode mata kuliah untuk melakukan pengecekan bahwa data yang diinputkan tidak sama dengan data yang tersimpan di dalam basis data.
2. Baris 24-41 adalah kode *form* untuk menginputkan data mata kuliah (praktikum) yang akan disimpan ke dalam basis data sistem.
3. Baris 33 digunakan untuk memanggil fungsi *javascript fungpraktikum* ketika *user* telah menginputkan kode mata kuliah.

4.4.2 Implementasi Komponen Unggah Modul Praktikum

Halaman unggah modul praktikum digunakan untuk mengunggah modul praktikum, halaman ini dapat diakses oleh koordinator asisten dosen. Tabel berikut merupakan cuplikan *code* dari *Controller c_materi_praktikum* yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4. 6 Controller `c_materi_praktikum`

```

1 class c_materi_praktikum extends CI_Controller {
2   ...
3   public function __construct()
4     {
5       parent::__construct();
6       $this->load->model( 'M_materi_praktikum' );
7       $this->load->model( 'membership_model1' );
8       $this->load->helper(array("html", "form", "url", "text"));
9       $this->load->library('upload');
10    }
11   public function index()
12     {
13       $user = $this->session->userdata('kode_user');
14       $data['user'] = $this->membership_model1->userData($user);
15       $this->load->view( 'asisten/v_materi_praktikum'
16 , $data , array('error' => ' ' ));
17     }
18   public function create() {
19     $config['upload_path'] = './modul/';
20     $config['allowed_types'] = 'docx|doc|pdf';
21     $config['max_size']     = '20000000';
22     $config['overwrite']=TRUE;
23     $this->load->library('upload', $config);
24     $this->upload->initialize($config);
25
26     if ( ! $this->upload->do_upload() )
27     {
28       $error = array('error' => $this->upload->display_errors());
29     }
30     else
31     {
32       $data['array']=$this->upload->data();
33       $filename=$data['array']['file_name'];
34       $link="modul/".$filename;
35       if( !empty( $_POST ) ) {
36         $this->M_materi_praktikum->create($link);
37         C_materi_praktikum::index();
38       } } }
39   ...
40 }

```

Penjelasan implementasi dari Tabel 4.6 adalah:

1. Baris 3-10 adalah fungsi **construct** yang memanggil / menginialisasi *model* dan *helper* yang dibutuhkan didalam *class controller* `c_materi_praktikum`.
2. Baris 11-17 adalah fungsi **index** yang akan pertama kali dieksekusi ketika *controller* dipanggil.
3. Baris 18-39 adalah fungsi **create** yang digunakan untuk mengunggah data dan file modul praktikum ke dalam basis data sistem.
4. Baris 19-22 digunakan untuk konfigurasi library CI.
5. Baris 28 digunakan untuk apabila terjadi error ketika file akan diunggah.

6. Baris 32-34 dieksekusi ketika data dan file modul dapat diunggah. Alamat dari file modul disimpan pada variabel \$link pada baris 34 dan kemudian akan disimpan bersama data-data lain dengan memanggil fungsi **create** pada baris 36.

Tabel 4.6 berikut merupakan cuplikan *code* dari *model* `m_materi_praktikum` yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4. 7 Model `m_materi_praktikum`

```

1 class m_materi_praktikum extends CI_Model {
2   ...
3   public function create($link) {
4
5       $modul_ke           = $_POST['cModulKe'];
6       $id                 = $_POST['ID'];
7       $deskripsi          =
8   $_POST['cDeskripsi'];
9       $pengunggah        = $_POST['cPengunggah'];
10      $topik               =
11  $_POST['cTopikMateri'];
12
13      $link= $link;
14      $query               = "INSERT INTO m_materi_praktikum ( modul_ke,
15 id_tahun_ajaran_praktikum, topik, deskripsi, pengunggah,
16 tanggal_upload, lokasi_upload ) VALUES ('$modul_ke', '$id',
17 '$topik', '$deskripsi', '$pengunggah', NOW(), '$link' ) ";
18      $sqlquery            = $this->db->query($query);
19      return $this->db->insert_id();
20   }
21   ...
22 }

```

Penjelasan implementasi dari Tabel 4.7 adalah:

1. Baris 3-20 adalah fungsi **create** untuk menyimpan data-data dan alamat file modul ke dalam basis data.
2. Baris 5-13 digunakan untuk menyimpan data kedalam variabel baru.
3. Baris 14-19 adalah query sql untuk menginputkan data-data ke dalam tabel `m_materi_praktikum` di dalam basis data.

Tabel 4.7 berikut merupakan cuplikan *code* dari view `v_materi_praktikum` yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4. 8 View `v_materi_praktikum`

```

1   ...
2   <div id="create">
3     <form enctype="multipart/form-data" action="<?php echo base_url();
4     ?>index.php/asisten/C_materi_praktikum/create" method="post">
5
6     Modul Ke:

```

```

7 <select name="cModulKe" id="cModulKe">
8     <?php for($i=1; $i<=15; $i++){
9         echo         "<option         value=".$i.">
10    ".$i."</option>";
11        }
12        ?>
13    </select>
14    Topik:
15    <input type="text" id="cTopikMateri" name="cTopikMateri" />
16        Deskripsi:
17    <textarea name="cDeskripsi" id="cDeskripsi" rows="5" cols="20"
18    ></textarea>
19    Browse:
20    <input type="file" name="userfile" size="20" required/>
21    <input         type="hidden"         id="cPengunggah"         name="cPengunggah"
22    value="<?php         echo         $this->session->userdata('kode_user');         ?>"
23    readonly="readonly" />
24        <input type="submit" value="Upload" />
25    </form>
26    </div>
27    ...

```

Penjelasan implementasi dari Tabel 4.8 adalah:

1. Baris 3-4 digunakan untuk menginisialisasi bahwa *action* dari *form create* ini adalah *controller c_materi_praktikum* pada fungsi *create*.
2. Baris 6-23 digunakan untuk menyimpan data inputan sementara kedalam variabel.
3. Baris 24 digunakan untuk *submit* data.

3.4.3 Implementasi Komponen Penjadwalan

Halaman set jadwal digunakan untuk menambahkan jadwal praktikum, hak akses halaman ini dimiliki oleh *administrator*. Tabel berikut merupakan cuplikan *code* dari *Controller c_jadwal* yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4. 9 *Controller c_jadwal*

```

1 class c_jadwal extends CI_Controller {
2     ...
3     public function addJadwalAdmin()
4     {
5         $user = $this->session->userdata('kode_user');
6         $data['user'] = $this->membership_model->userData($user);
7         $data['ruang'] = $this->M_jadwal->getSomeRuang();
8         $this->load->view( 'admin/v_addJadwal', $data);
9     }
10    public function get_ruang() {
11
12        function fungsiRuang($array, $key, $val) {
13            foreach ($array as $item)
14                if (isset($item[$key]) && $item[$key] == $val)
15                    return true;
16            return false;

```

```

17     }
18     $posts = $this->M_jadwal->getRuang();
19     $postsa = $this->M_jadwal->getHasRuang();
20     function compareDeepValue($val1, $val2)
21     {
22         return strcmp($val1['ruang'], $val2['ruang']);
23     }
24     $result = array_uintersect($posts,$postsa,
25 'compareDeepValue');
26     if(isset($posts)) :
27
28         echo "<select id='ruang' name='ruang'>";
29         foreach ( $posts as $post ) :
30     if(fungsiRuang($result, "ruang",$post['ruang'])){
31     echo "<option value=" . $post['ruang'] . " disabled" .
32 $post['ruang'] . "</option>";
33     }else{
34     echo "<option value=" . $post['ruang'] . ">" . $post['ruang'] .
35 "</option>";
36     }
37     endforeach;
38     echo "</select>";
39     endif;
40     ;
41     }
42     public function addJadwal() {
43         if( !empty( $_POST ) ) {
44             $this->M_jadwal->addJadwal();
45             //echo 'Record added successfully!';
46         }
47     }
48     ...
49 }

```

Penjelasan implementasi dari Tabel 4.9 adalah:

1. Baris 3-9 adalah fungsi **addJadwalAdmin** fungsi ini adalah fungsi yang pertama dieksekusi ketika *user* meminta halaman tambah jadwal praktikum.
2. Baris 10-41 adalah fungsi **get_ruang** yang akan dipanggil untuk melakukan pengecekan ruang yang tersedia pada hari dan jam yang diinputkan oleh *user (administrator)*. Ruang yang memiliki jadwal yang sama dengan jadwal yang diinputkan *administrator* statusnya akan di *disabled*.
3. Baris 42-47 adalah fungsi **addJadwal** yang akan dipanggil ketika *user (administrator)* akan menambahkan data jadwal praktikum, dimana *controller* akan memanggil fungsi **addJadwal** pada *model m_jadwal*.

Tabel 4.10 berikut merupakan cuplikan *code* dari *model m_jadwal* yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4. 10 Model m_jadwal

```

1 class m_jadwal extends CI_Controller {
2   ...
3   public function addJadwal() {
4     $time1 = $this->input->post( 'waktu1', true );+":00";
5     $time2 = $this->input->post( 'waktu2', true );+":00";
6     $data = array(
7     id_kelas' => $this->input->post( 'id_kelas', true ),
8     'hari'     => $this->input->post( 'hari', true ),
9     'waktu1'  => $time1,
10    'waktu2'  => $time2,
11    'ruang'   => $this->input->post( 'ruang', true ) );
12    $this->db->insert( 'm_jadwal', $data );
13    return $this->db->insert_id(); }
14   ...
15   function getRuang()
16   {
17     $this->db->from('ruangan');
18     $posts = $this->db->get()->result_array();
19     return $posts;
20   }
21   function getSomeRuang()
22   {
23     $result_data = $this->db->query("SELECT distinct
24 id_kelas AS kelas FROM m_jadwal ");
25     $hasil = $result_data->result_array();
26     if(isset($hasil)) {
27       return $hasil;}
28   }
29   function getHasRuang()
30   {
31     $time1 = $this->input->post( 'waktu1', true );+":00";
32     $time2 = $this->input->post( 'waktu2', true );+":00";
33     $hari = $this->input->post( 'hari', true );
34     $result_data = $this->db->query("SELECT distinct ruang
35 FROM m_jadwal WHERE hari = '$hari' AND ('$time1' BETWEEN waktu1 AND
36 waktu2 OR '$time2' BETWEEN waktu1 AND waktu2 OR waktu1 BETWEEN
37 '$time1' AND '$time2' OR waktu2 BETWEEN '$time1' AND '$time2')");
38     $hasil = $result_data->result_array();
39     if(isset($hasil)) {
40       return $hasil;
41     }else{
42       $hasil[0]['ruang'] = "null";
43     }
44     return $hasil;}
45   ...
46 }

```

Penjelasan implementasi dari Tabel 4.10 adalah:

1. Baris 3-13 adalah fungsi **addJadwal** yang digunakan untuk menginputkan data jadwal praktikum ke dalam basis data.
2. Baris 15-20 adalah fungsi **getRuang** yang digunakan untuk mengambil *records* data ruang yang ada di dalam basis data.
3. Baris 21-28 adalah fungsi **getSomeRuang** yang digunakan untuk mengambil data *id_kelas* yang ada pada tabel *m_jadwal*.

4. Baris 29-44 adalah fungsi **getHasRuang** yang digunakan untuk mendapatkan data ruang yang telah digunakan pada hari dan waktu yang diinputkan oleh *administrator*.

Tabel 4.11 berikut merupakan cuplikan *code* dari *view* `v_addJadwal` yang diimplementasikan pada bahasa pemrograman PHP.

Tabel 4. 11 View `v_addJadwal`

```

1  ...
2  <script>
3      function enable()
4      {
5          $('#ruang').prop('disabled', false);
6          var formData1 = "hari="+document.getElementById
7          ('hari').value+"&waktu1="+document.getElementById('waktu1').value+"&
8          waktu2="+document.getElementById('waktu2').value;
9          $.ajax({
10             url: '=php echo
11             base_url()&gt;index.php/admin/C_jadwal/get_ruang',
12             type: 'POST',
13             data: formData1,
14             success: function(data) {
15                 $('#ruangan').html(data);
16             } //end success
17             });
18         }
19     &lt;/script&gt;
20 ...
21 &lt;div&gt;
22     &lt;form action="" method="post"&gt;
23     &lt;input type="hidden" id="id_kelas" name="id_kelas" /&gt;
24     Hari:
25     &lt;select id="hari" name="hari"&gt;
26     &lt;option value="Senin"&gt;Senin&lt;/option&gt;
27     &lt;option value="Selasa"&gt;Selasa&lt;/option&gt;
28     &lt;option value="Rabu"&gt;Rabu&lt;/option&gt;
29     &lt;option value="Kamis"&gt;Kamis&lt;/option&gt;
30     &lt;option value="Jumat"&gt;Jumat&lt;/option&gt;
31     &lt;/select&gt;
32     Waktu:
33     &lt;input type="text" id="waktu1" name="waktu1" size="5" /&gt;
34     &lt;input type="text" id="waktu2" name="waktu2" size="5"
35     onKeyUp="enable();" onChange="enable();" /&gt;
36
37     Ruang:
38     &lt;select id="ruangan" name="ruang" disabled&gt;
39 &lt;/select&gt;
40     &lt;input type="hidden" id="userId" name="id" /&gt;
41 &lt;/form&gt;
42 &lt;/div&gt;
43 ...
</pre

```

Penjelasan implemmentasi dari Tabel 4.11 adalah:

1. Baris 2-19 adalah *code javascript* yang digunakan untuk mengelola ruang yang tersedia (*available*) dan tidak tersedia (*non-available*) pada jadwal

yang diinputkan oleh *administrator*. *Code javascript* ini dieksekusi ketika *administrator* selesai mnginputkan data jam praktikum (pada baris 36-37).

2. Baris 22-41 adalah *code form* untuk menginputkan jadwal praktikum.

4.5 Implementasi *Interface*

Implementasi *interface* dibangun berdasarkan pada perancangan *interface* pada bab perancangan. Implementasi *interface* terdiri atas *interface login*, *home*, profil, *news feed*, manajemen praktikum, manajemen kelas, manajemen absensi, manajemen penggajian dan modul praktikum.

4.4.1 Halaman *Login*

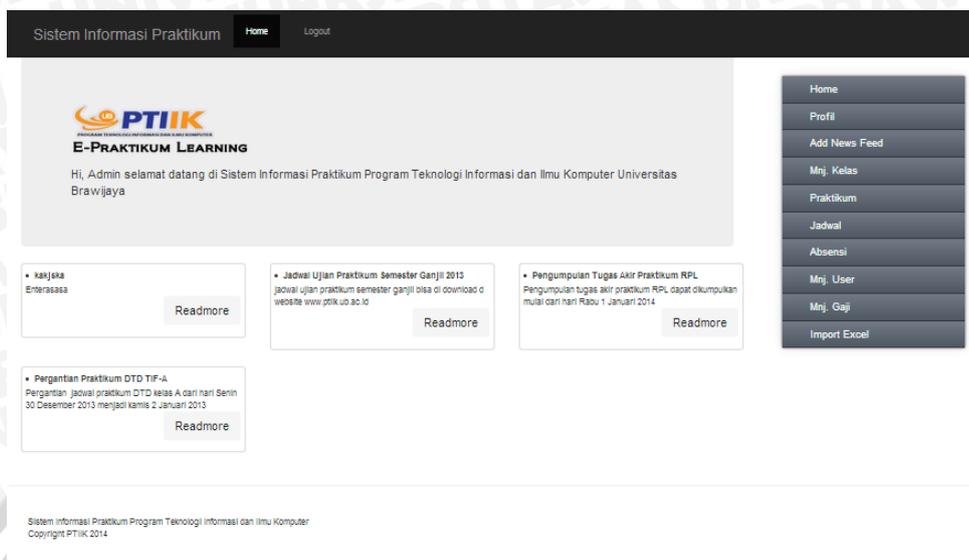
Halaman *login* merupakan halaman yang pertama kali diakses ketika aplikasi web dijalankan. Pengguna harus memasukkan *username* dan *password*. Masing-masing aktor memiliki hak akses yang berbeda dalam mengoperasikan sistem. Halaman *login* ditunjukkan pada gambar 4.2.



Gambar 4. 2 Implementasi *interface login*

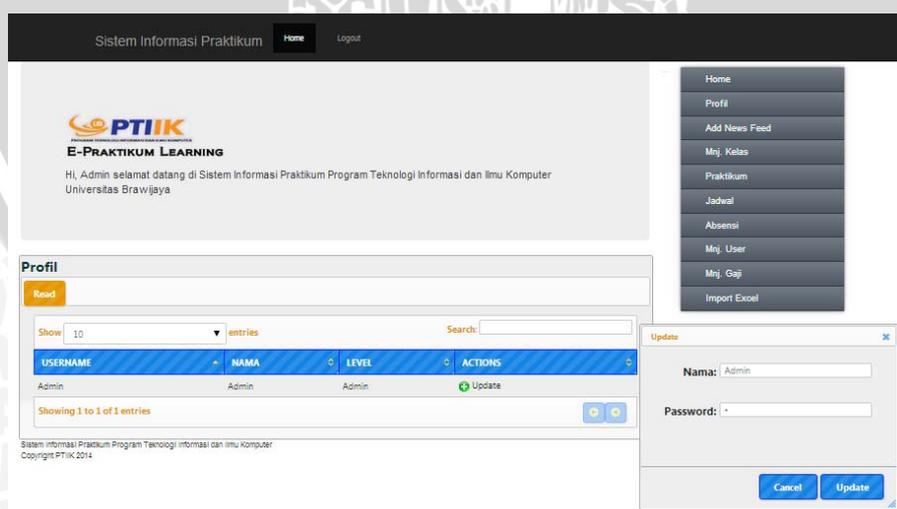
4.4.2 Halaman *Home*

Halaman *home* merupakan halaman yang pertama kali ditampilkan ketika *user* telah melakukan *login*. Pada halaman ini terdapat beberapa menu, dimana menu yang ditampilkan bergantung pada aktor yang mengakses sistem. Gambar 4.3 menunjukkan halaman *home* beserta menu aktor *administrator*.

Gambar 4. 3 Implementasi *interface home*

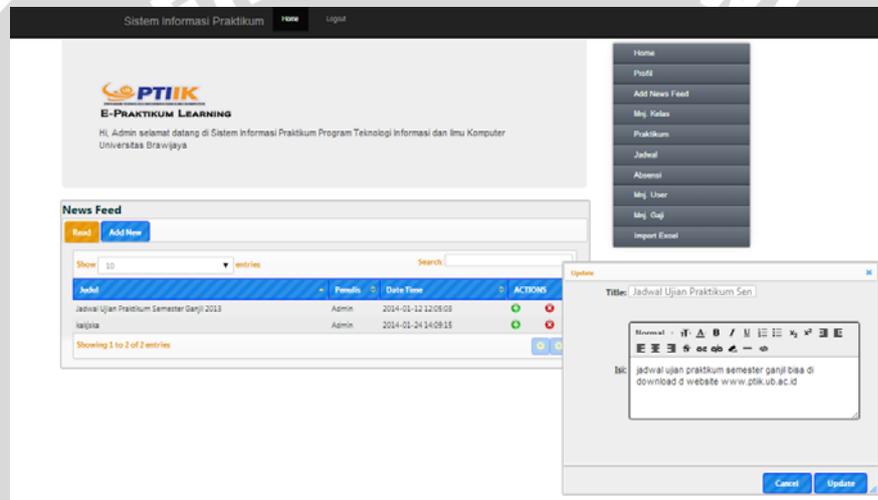
4.4.3 Halaman Kelola Profil

Halaman ini berisi tentang profil pengguna serta submenu untuk melakukan *update* profil pengguna. Halaman ini dapat diakses oleh semua aktor. Pada dasarnya tampilan halaman ini sama pada setiap aktor, namun menu yang mengikutinya bergantung pada aktor pengguna. Gambar 4.4 menunjukkan halaman kelola profil aktor *administrator*.

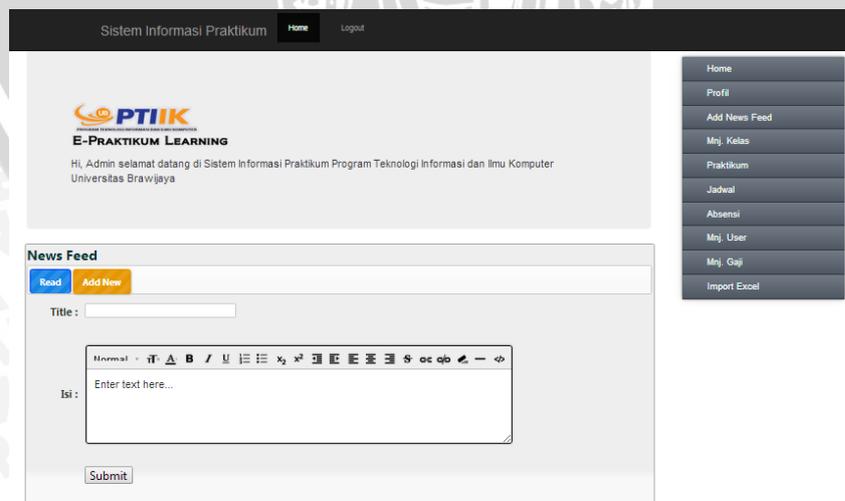
Gambar 4. 4 Implementasi *interface profil*

4.4.4 Halaman Kelola *News feed*

Halaman kelola *news feed* berisi tentang pengelolaan berita praktikum. Hak akses halaman kelola *news feed* dimiliki oleh aktor *administrator*, asisten dosen, koordinator asisten dosen dan dosen. Aktor-aktor tersebut memiliki hak akses untuk mengunggah berita praktikum. Tampilan halaman kelola *news feed* pada dasarnya sama untuk *administrator*, asisten dosen, koordinator asisten dosen dan dosen. Gambar 4.5 menunjukkan halaman kelola *news feed* untuk aktor *administrator*, Gambar 4.6 menunjukkan halaman untuk mengunggah berita (*news*) aktor *administrator*.



Gambar 4. 5 Implementasi *interface* manajemen *news feed*

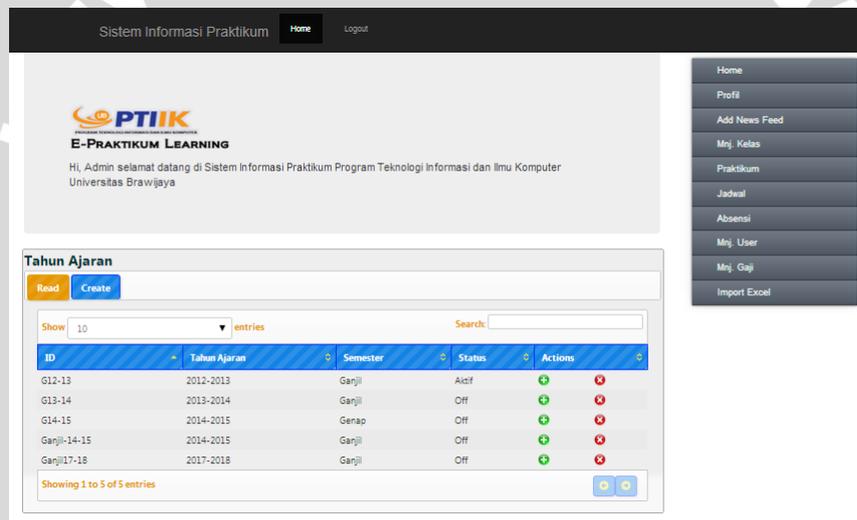


Gambar 4. 6 Implementasi *interface* add *news feed*

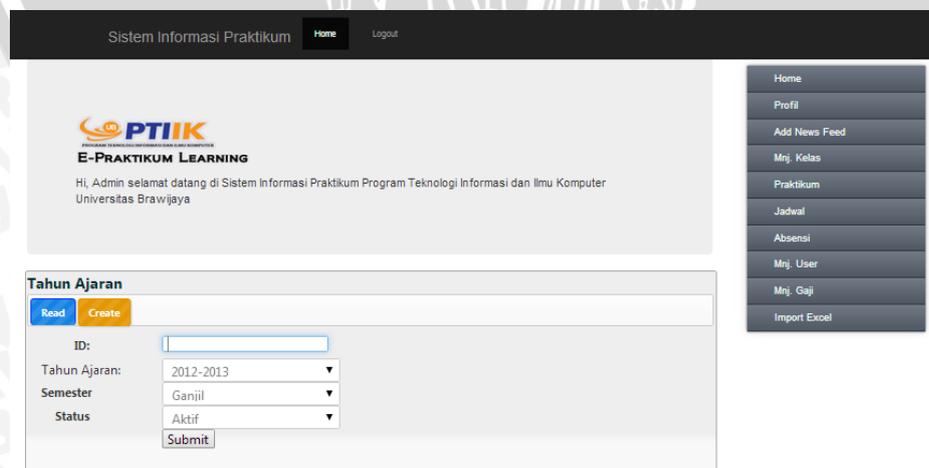
4.4.5 Halaman Kelola Praktikum

Hak akses halaman kelola praktikum dimiliki oleh aktor *administrator*. Pada halaman kelola praktikum terdapat beberapa menu diantaranya untuk mengelola tahun ajaran, mengelola program studi, mengelola praktikum dan mengelola praktikum apa saja yang berlangsung pada tahun ajaran tertentu.

Halaman kelola tahun ajaran digunakan untuk manajemen tahun ajaran yang akan atau sedang berlangsung. Pada halaman ini terdapat menu untuk melakukan *add*, *update* dan *delete* tahun ajaran serta melakukan *setting* kondisi tahun ajaran. Sistem akan mengolah data pada tahun ajaran dengan kondisi 'Aktif'.

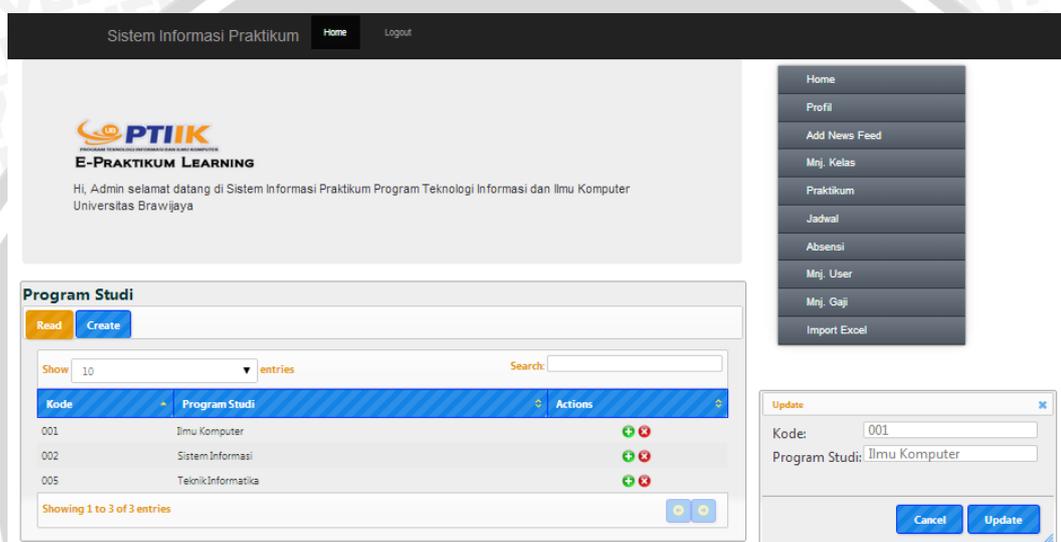


Gambar 4. 7 Implementasi *interface* manajemen tahun ajaran

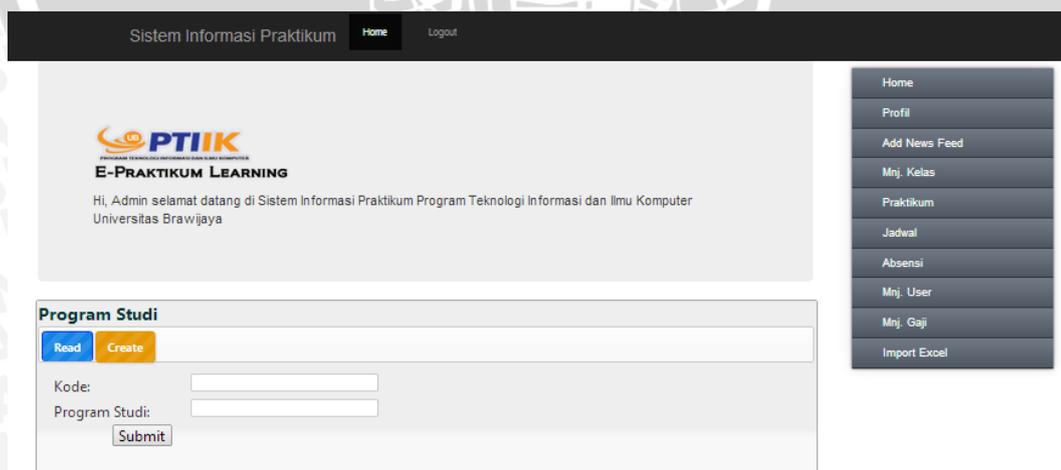


Gambar 4. 8 Implementasi *interface* tambah tahun ajaran

Pada halaman *interface* kelola program studi terdapat menu untuk melakukan *insert*, *update* dan *delete* data program studi yang ada pada PTIIK. Fungsi halaman manajemen program studi adalah untuk mengelompokkan praktikum berdasarkan program studinya masing-masing. Form untuk insert program studi memiliki dua inputan yaitu inputan untuk id program studi dan nama program studi. Tampilan halaman untuk mengelola program studi yang ada pada PTIIK ditunjukkan pada gambar 4.9 dan Gambar 4.10.



Gambar 4. 9 Implementasi *interface* manajemen program studi



Gambar 4. 10 Implementasi *interface* tambah data program studi

Halaman *interface* manajemen praktikum digunakan untuk mengelola praktikum yang ada pada PTIIK yang dikelompokkan berdasarkan program studi. Pada halaman manajemen praktikum pada form untuk menambahkan praktikum, data - data yang diinputkan adalah kode praktikum (kode matakuliah), nama praktikum serta program studi praktikum tersebut. Tampilan halaman manajemen praktikum ditunjukkan pada Gambar 4.11 dan tampilan form untuk menambahkan praktikum baru ditunjukkan pada gambar 4.12.

The screenshot shows the 'Sistem Informasi Praktikum' interface. At the top, there is a navigation bar with 'Home' and 'Logout' links. Below the navigation bar, the PTIIK logo and 'E-PRAKTIKUM LEARNING' are displayed. A welcome message reads: 'Hi, Admin selamat datang di Sistem Informasi Praktikum Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya'. The main content area is titled 'Manajemen Praktikum' and contains two buttons: 'ListPraktikum' and 'Add Praktikum'. Below these buttons is a search bar with 'Show: 10 entries' and a search input field. A table lists the following practices:

Kode	Praktikum	Program Studi	Actions
IFK15041	Pengembangan Aplikasi Perangkat Bergerak	Sistem Informasi	+ -
SI41111	Desain & Pemrograman Web	Sistem Informasi	+ -
TIF4001	Basis Data	Teknik Informatika	+ -
TIF4002	Rekayasa Perangkat Lunak	Teknik Informatika	+ -

At the bottom of the table, it says 'Showing 1 to 4 of 4 entries' and there are two pagination buttons.

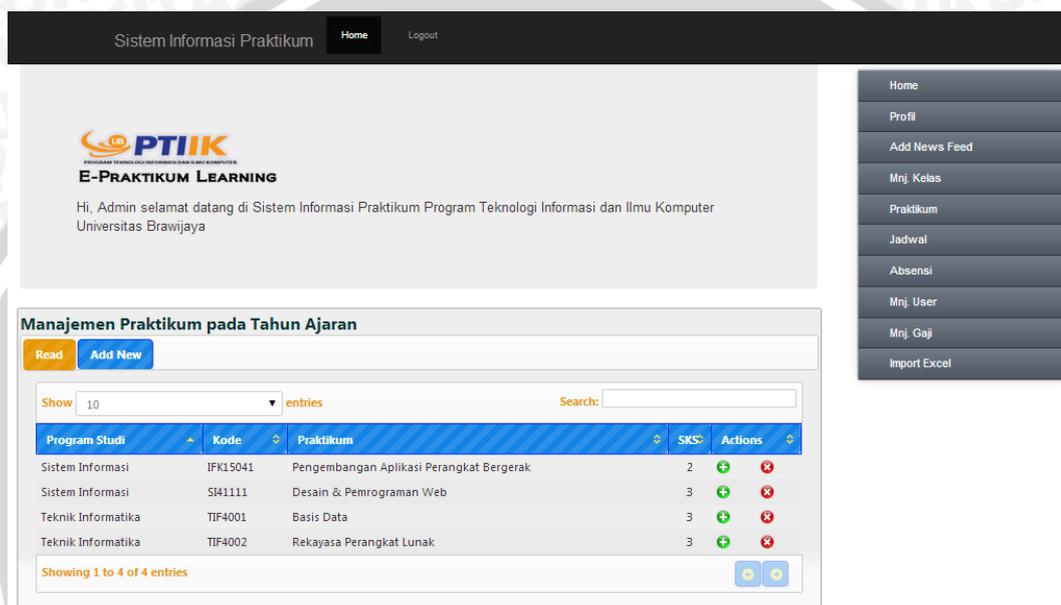
Gambar 4. 11 Implementasi *interface* manajemen praktikum

The screenshot shows the 'Sistem Informasi Praktikum' interface for adding a new practice. It features the same navigation bar and header as Gambar 4.11. The 'Manajemen Praktikum' section has the 'Add Praktikum' button highlighted. Below the buttons is a form with the following fields:

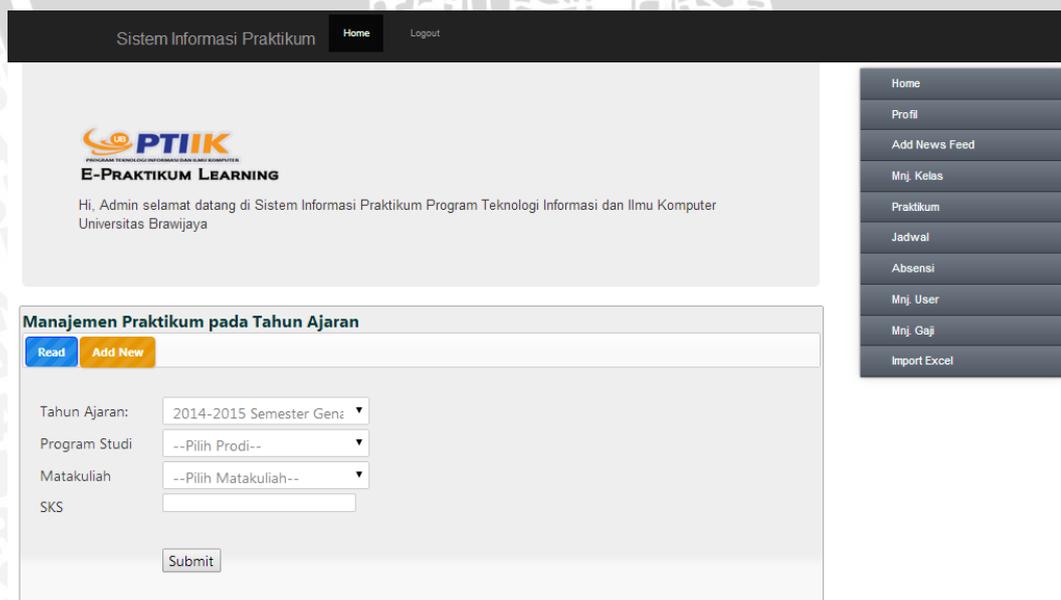
- Kode Praktikum:
- Praktikum:
- Program Studi:
- Submit:

Gambar 4. 12 Implementasi *interface* tambah data praktikum

Halaman kelola praktikum pada tahun ajaran digunakan untuk mengelola praktikum apa saja yang akan dilangsungkan pada tahun ajaran yang akan dimulai. Pada halaman ini terdapat menu untuk *insert*, *update* dan *delete* data praktikum. Tampilan halaman untuk mengelola tahun ajaran praktikum ditunjukkan pada gambar 4.13 dan tampilan halaman untuk memasukkan data praktikum pada tahun ajaran ditunjukkan pada Gambar 4.14.



Gambar 4. 13 Implementasi *interface* manajemen praktikum pada tahun ajaran

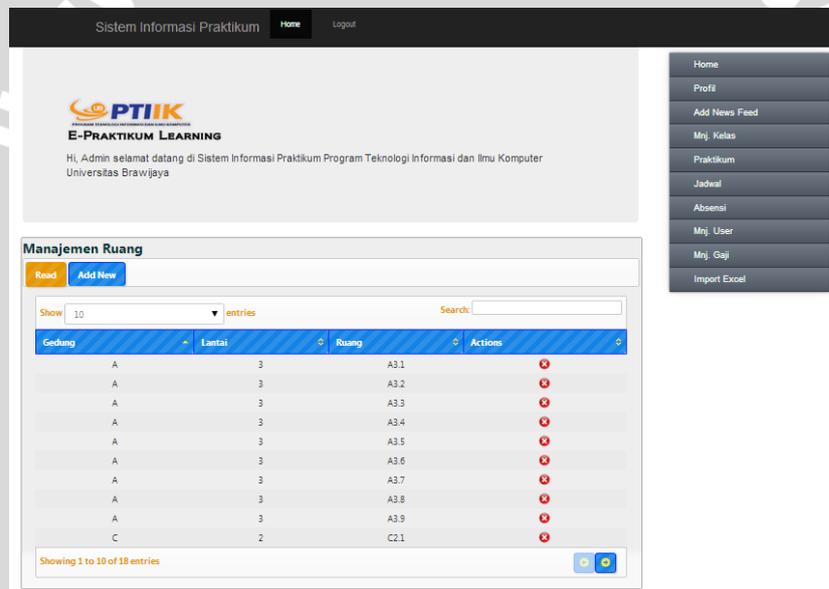


Gambar 4. 14 Implementasi *interface* menambahkan praktikum pada tahun ajaran

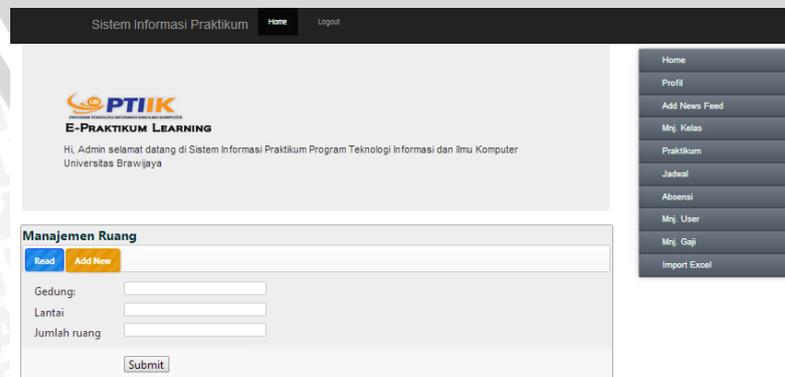
4.4.6 Halaman Kelola Kelas Praktikum

Pada halaman kelola kelas praktikum terdapat beberapa menu untuk mengelola ruang praktikum, mengelola jumlah kelas praktikum, mengelola jadwal praktikum dan untuk memasukkan data mahasiswa yang mengikuti kegiatan praktikum pada kelas praktikumnya. Hak akses halaman ini hanya dimiliki oleh aktor *administrator*.

Halaman manajemen ruang digunakan untuk mengelola ruang (kelas) yang ada pada gedung PTIIK. Pengelolaan ruang ini nantinya digunakan dalam penjadwalan. Tampilan halaman kelola ruang ditunjukkan pada Gambar 4.15 dan tampilan untuk menambahkan ruang baru ditunjukkan pada gambar 4.16.

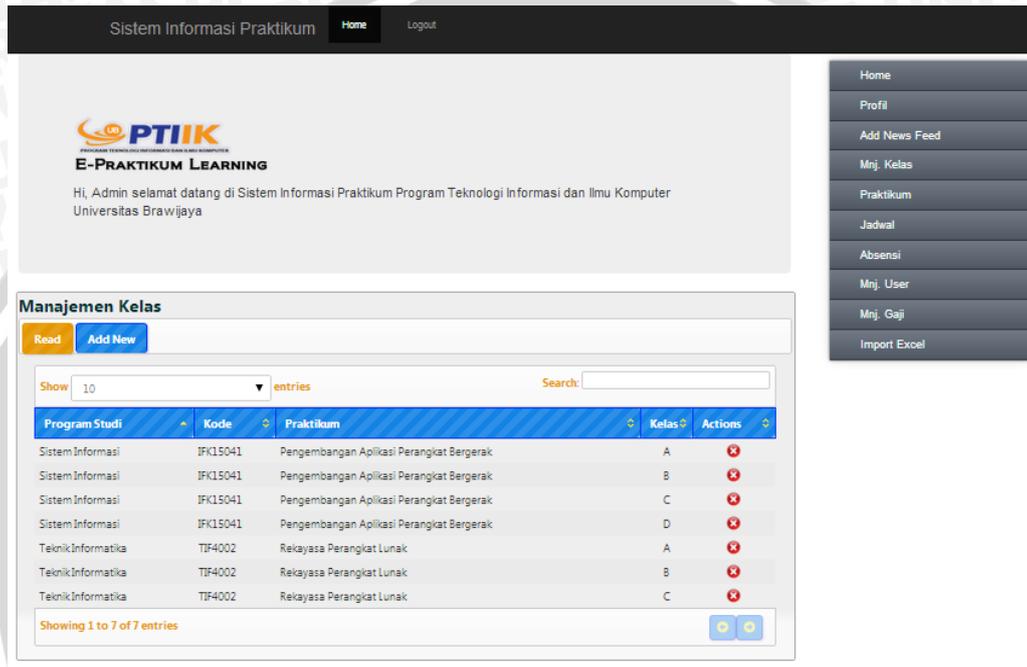


Gambar 4. 15 Implementasi *interface* manajemen ruang

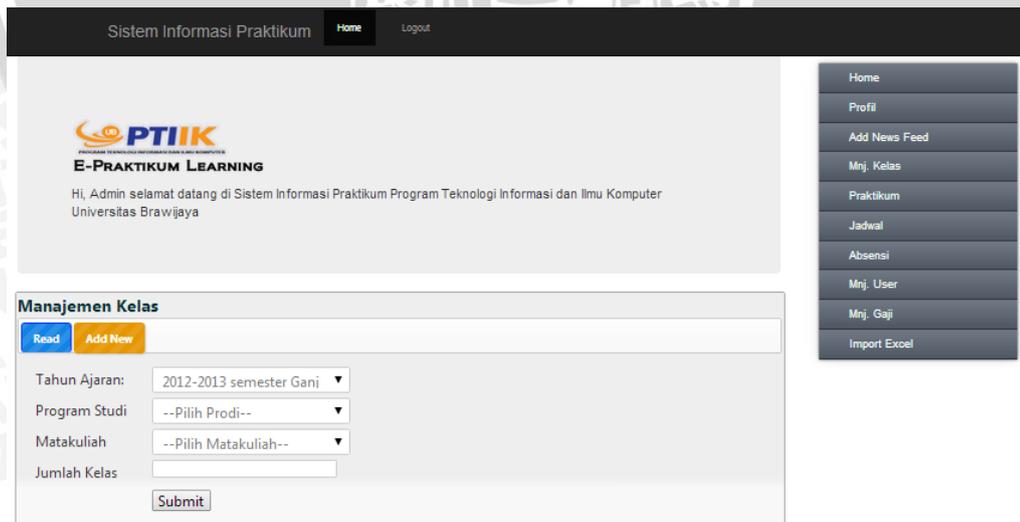


Gambar 4. 16 Implementasi *interface* tambah data ruang

Halaman kelola kelas digunakan untuk mengelola jumlah kelas praktikum yang dibuka pada setiap mata kuliah. Pada halaman kelola kelas terdapat submenu untuk memasukkan jadwal praktikum dan juga untuk menghapus kelas praktikum. Tampilan halaman kelola kelas ditunjukkan pada Gambar 4.17 dan tampilan halaman untuk memasukkan jumlah kelas praktikum ditunjukkan pada Gambar 4.18.

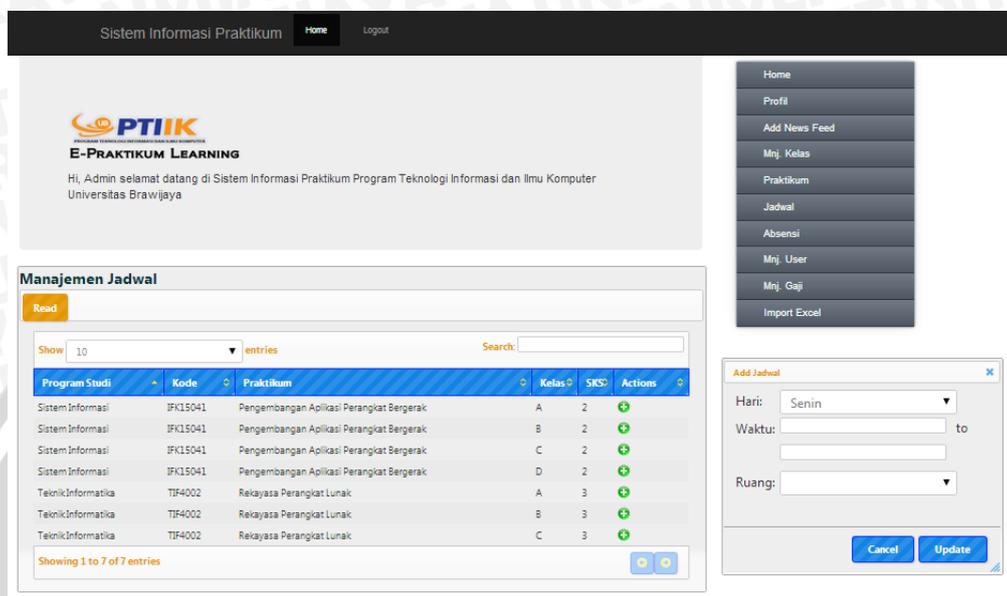


Gambar 4. 17 Implementasi *interface* manajemen kelas



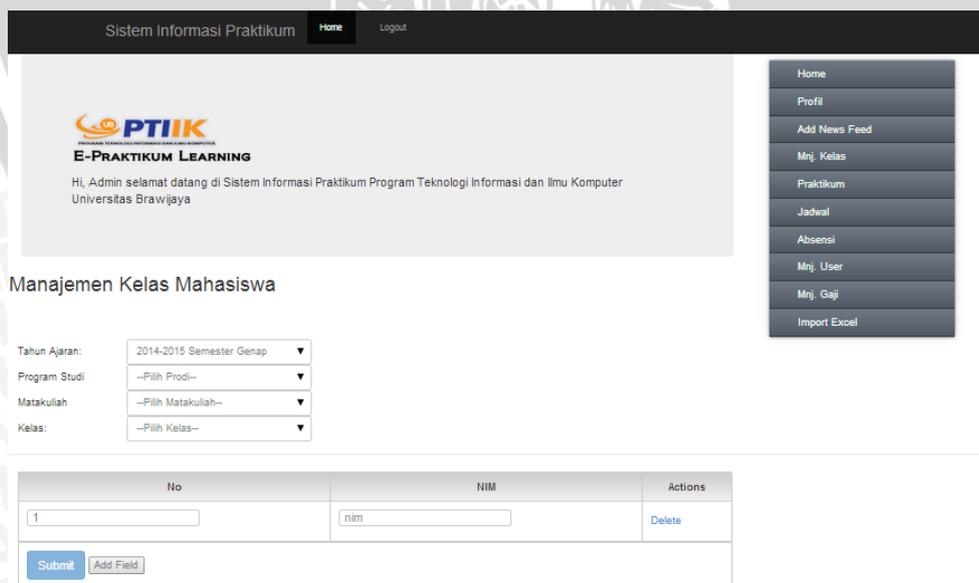
Gambar 4. 18 Implementasi *interface* tambah data kelas praktikum

Halaman kelola jadwal berisi data jadwal praktikum. Pada halaman kelola jadwal juga terdapat submenu untuk melakukan *update* dan *delete* jadwal. Tampilan halaman kelola jadwal ditunjukkan pada gambar 4.19.



Gambar 4. 19 Implementasi *interface* manajemen jadwal

Halaman kelola kelas praktikum mahasiswa berisi form untuk memasukkan data mahasiswa yang mengikuti praktikum. Tampilan halaman kelola kelas praktikum mahasiswa ditunjukkan pada Gambar 4.20.



Gambar 4. 20 Implementasi *interface* tambah mahasiswa ke kelas praktikum

4.4.7 Halaman Laporan Absensi Mahasiswa dan Asisten Dosen.

Halaman laporan absensi mahasiswa dan asisten dosen digunakan untuk mengakses informasi absensi mahasiswa dan asisten dosen. Pengaksesan informasi absensi mahasiswa (praktikan) dikelompokkan berdasarkan program kelas mahasiswa. *Administrator* terlebih dahulu mengisi form daftar absensi mahasiswa, kemudian sistem akan memproses permintaan *administrator* dan menampilkan data sesuai permintaan *administrator*. Tampilan form ditunjukkan pada Gambar 4.21 dan tampilan laporan absensi yang berisi data kehadiran mahasiswa (praktikan) ditunjukkan pada Gambar 4.22

Gambar 4. 21 Implementasi *interface* form untuk memilih kelas praktikum

Kode praktikum	Praktikum	Nim	Nama	Alpha	Jml Hadir	Presentase Kehadiran
TIF4002	Rekayasa Perangkat Lunak	0910690086	Widya Hanif	2	2	50%
TIF4002	Rekayasa Perangkat Lunak	0910681001	Dara Nisa'	0	4	100%

Gambar 4. 22 Implementasi *interface* laporan absensi mahasiswa (praktikan)

Halaman laporan absensi asisten dosen digunakan untuk menampilkan informasi absensi asisten dosen dan koordinator asisten dosen. Laporan absensi asisten dosen ditampilkan setelah *administrator* mengisi form, laporan absensi asisten dosen dan koordinator asisten dosen dikelompokkan berdasarkan mata kuliah. Halaman *interface* form untuk memilih laporan absensi berdasarkan mata kuliah ditunjukkan pada Gambar 4.23 dan laporan data kehadiran asisten dosen dan koordinator asisten dosen ditunjukkan pada Gambar 4.24.

The screenshot shows the PTIIK E-Praktikum Learning interface. At the top, there is a navigation bar with 'Sistem Informasi Praktikum', 'Home', and 'Logout'. Below this is a header section with the PTIIK logo and the text 'E-PRAKTIKUM LEARNING'. A welcome message reads: 'Hi, Admin selamat datang di Sistem Informasi Praktikum Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya'. The main content area contains a form with two dropdown menus: 'Program Studi' with the value '--Pilih Prodi--' and 'Matakuliah' with the value '--Pilih Matakuliah--'. Below the dropdowns is a 'Submit' button. On the right side, there is a vertical sidebar menu with the following items: Home, Profil, Add News Feed, Mnj. Kelas, Praktikum, Jadwal, Absensi, Mnj. User, Mnj. Gaji, and Import Excel.

Gambar 4. 23 Implementasi *interface* form untuk pilih laporan absensi asisten dosen berdasarkan mata kuliah

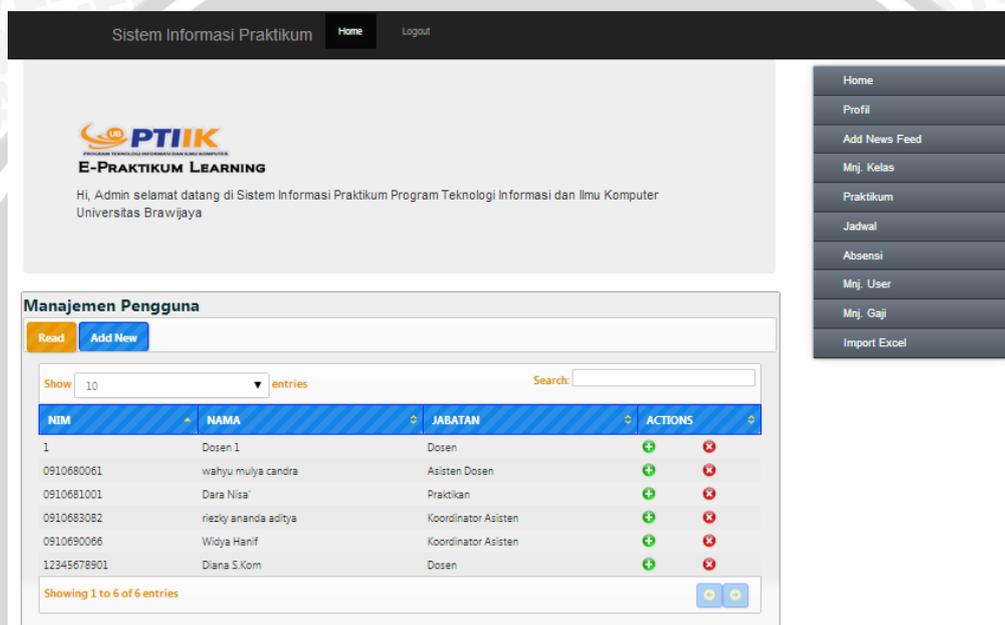
The screenshot shows the PTIIK E-Praktikum Learning interface displaying a table of absence reports. The table has the following columns: Kode Praktikum, Praktikum, Nim, Nama, Jabatan, and Jml. Kehadiran. The table contains two rows of data. Below the table is a 'Back' button. The sidebar menu on the right is identical to the one in Gambar 4.23.

Kode Praktikum	Praktikum	Nim	Nama	Jabatan	Jml. Kehadiran
TIF4002	Rekayasa Perangkat Lunak	0910683082	riezky ananda aditya	Koordinator Asisten	1
TIF4002	Rekayasa Perangkat Lunak	0910680061	wahyu mulya candra	Asisten Dosen	1

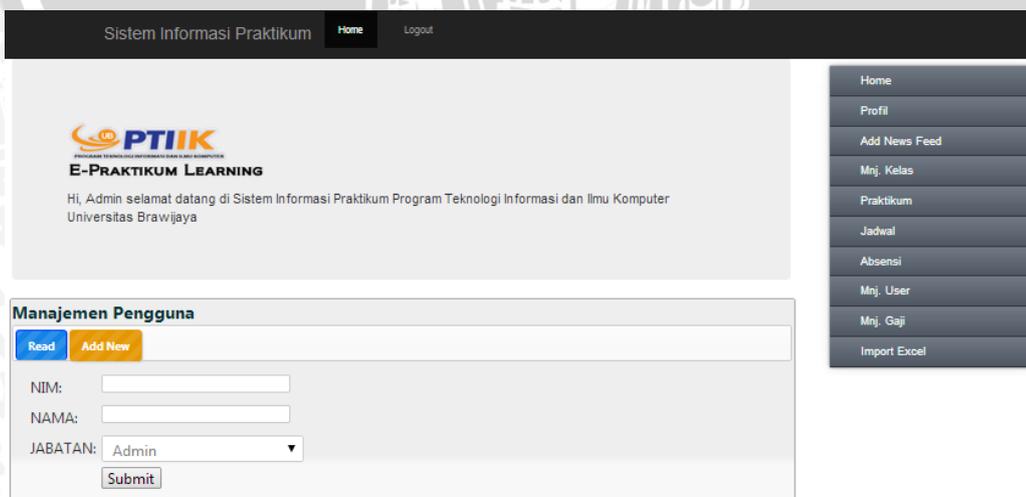
Gambar 4. 24 Implementasi *interface* laporan absensi asisten dosen & koordinator asisten dosen

4.4.8 Halaman Kelola User

Halaman ini terdiri dari beberapa submenu yang digunakan untuk mengelola pengguna sistem. Halaman kelola *user* terdiri dari beberapa halaman diantaranya halaman untuk mengelola seluruh pengguna (*user*), halaman untuk mengelola asisten dosen, dan halaman yang berisi daftar asisten dosen. Tampilan untuk mengelola *user* ditunjukkan pada Gambar 4.25 dan halaman untuk *input* pengguna baru ditunjukkan pada Gambar 4.26.

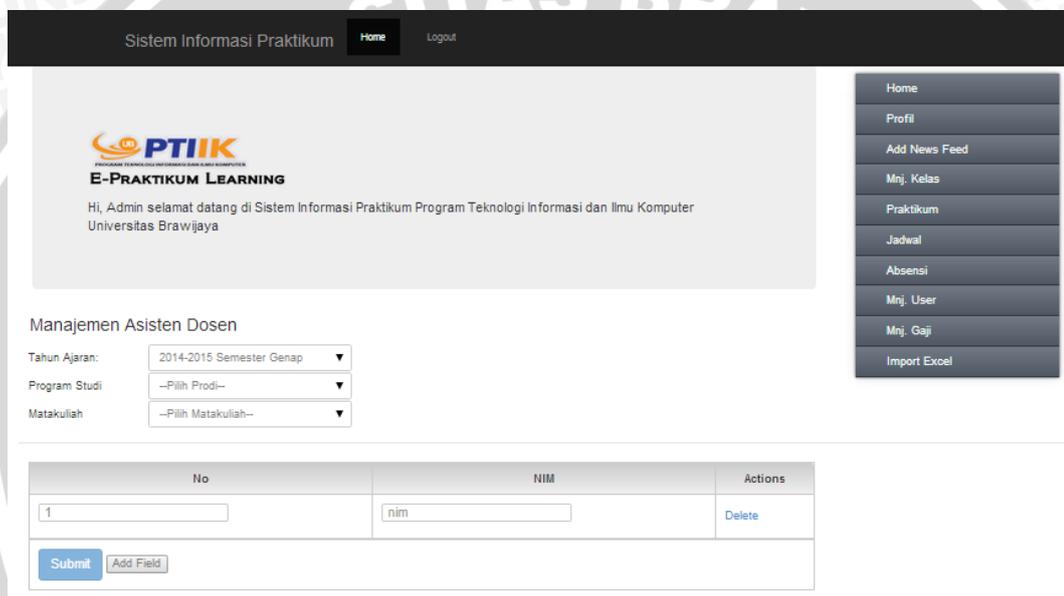


Gambar 4. 25 Implementasi *interface* manajemen *users*

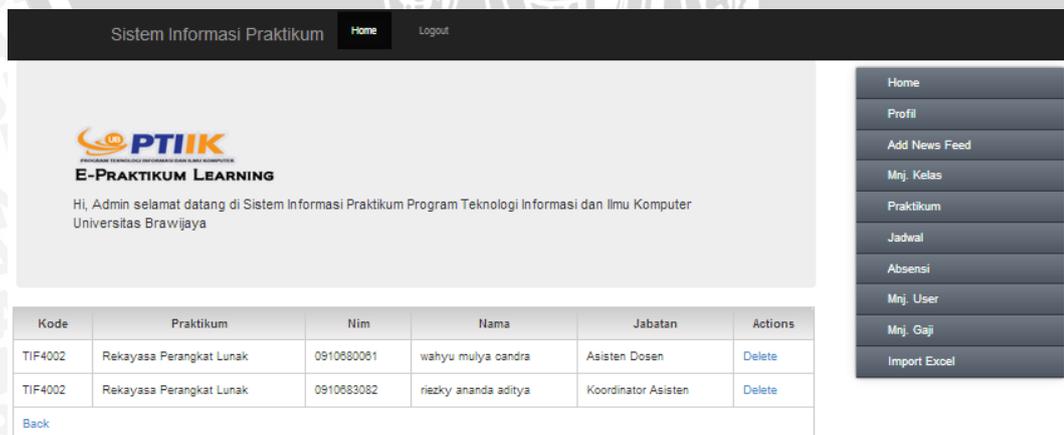


Gambar 4. 26 Implementasi *interface* untuk tambah *user* baru

Pada halaman untuk tambah *user* baru yang ditunjukkan pada Gambar 4.26 *default password* di-*setting* sama dengan *nim user* yang ditambahkan. Halaman kelola asisten dosen digunakan untuk memasukkan data secara kelompok berdasarkan pada mata kuliah. Tampilan halaman kelola asisten dosen untuk memasukkan data asisten dosen secara kelompok berdasarkan program mata kuliah ditunjukkan pada gambar 4.27. Halaman daftar asisten dosen yang dikelompokkan berdasarkan mata kuliah (praktikum) ditunjukkan pada gambar 4.28.



Gambar 4. 27 Implementasi *interface* input data asisten dosen secara kelompok



Gambar 4. 28 Halaman *interface* daftar asisten dosen dan koordinator asisten dosen

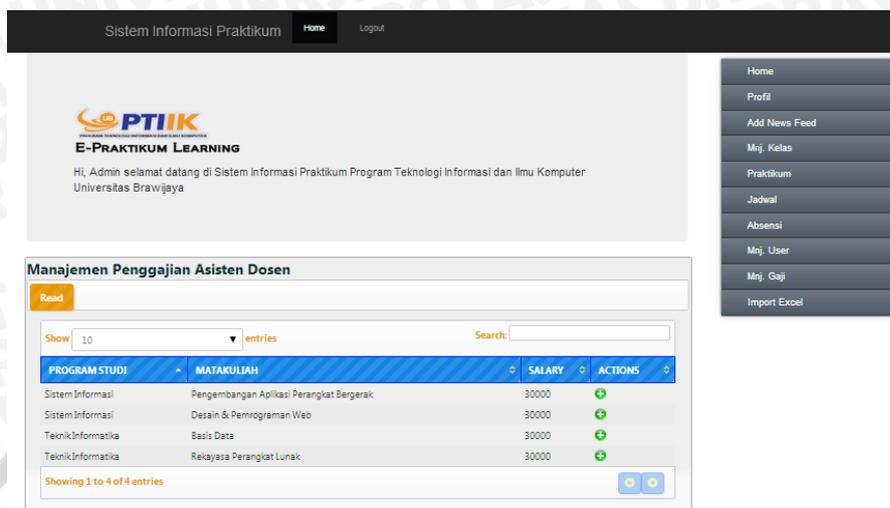
Halaman kelola dosen digunakan untuk memasukkan data dosen ke dalam kelas yang diajarnya, data yang dimasukkan adalah NIK dosen (ID). Setelah melakukan insert data dosen pada halaman kelola dosen, selanjutnya dosen memiliki hak akses untuk memantau perkembangan praktikum pada mata kuliah yang sedang diajarnya. Tampilan halaman manajemen dosen ditunjukkan pada Gambar 4.29.

Program Studi	Kode	Mata Kuliah	Kelas	Dosen	ACTIONS
Sistem Informasi	IFK15041	Pengembangan Aplikasi Perangkat Bergerak	A	Diana S.Kom	+
Sistem Informasi	IFK15041	Pengembangan Aplikasi Perangkat Bergerak	B	Diana S.Kom	+
Sistem Informasi	IFK15041	Pengembangan Aplikasi Perangkat Bergerak	C	Dosen 1	+
Sistem Informasi	IFK15041	Pengembangan Aplikasi Perangkat Bergerak	D	Dosen 1	+
Teknik Informatika	TIF4002	Rekayasa Perangkat Lunak	A	Dosen 1	+
Teknik Informatika	TIF4002	Rekayasa Perangkat Lunak	B	Dosen 1	+
Teknik Informatika	TIF4002	Rekayasa Perangkat Lunak	C	Dosen 1	+

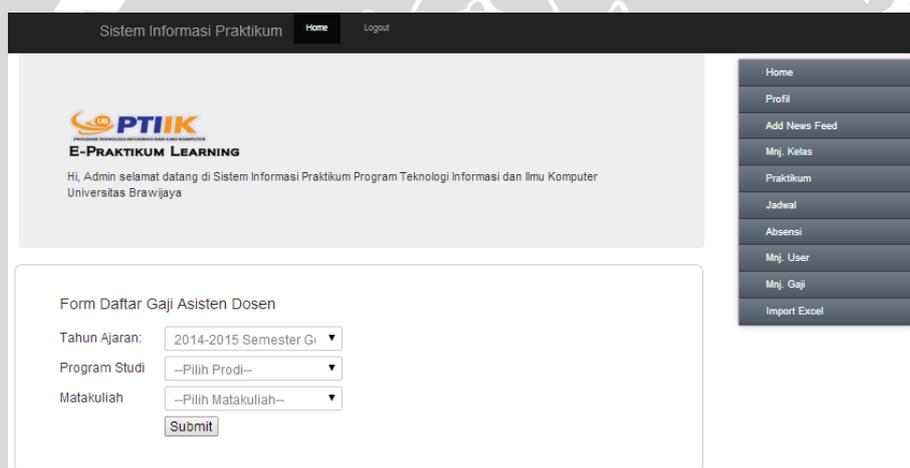
Gambar 4. 29 Implementasi *interface* tambah data dosen ke kelas praktikum

4.4.9 Halaman Kelola Gaji Asisten

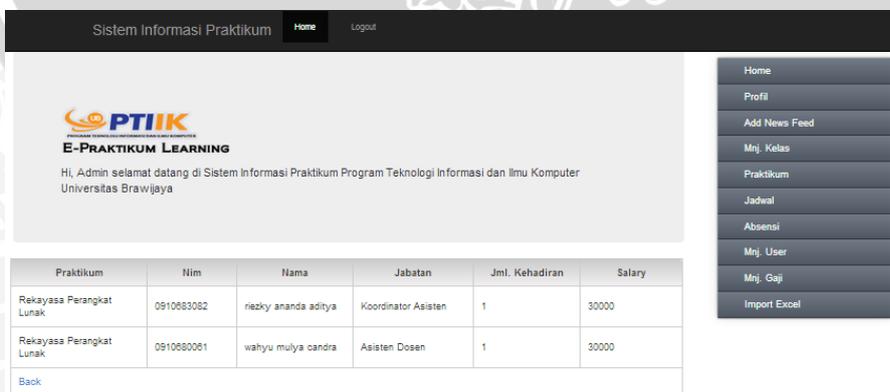
Halaman kelola gaji asisten dosen terdiri dari dua halaman, satu halaman digunakan untuk mengelola (*insert, update*) gaji dan satu halaman lagi digunakan untuk menampilkan hasil pengolahan gaji. Hak akses halaman untuk mengelola (*insert* dan *update*) gaji asisten dosen dimiliki oleh *administrator*, sedangkan hak akses halaman untuk melihat laporan gaji asisten dosen dimiliki oleh *administrator* dan dosen. Tampilan halaman untuk mengelola gaji asisten dosen ditunjukkan pada Gambar 4.30 dan halaman untuk menampilkan gaji asisten dosen ditunjukkan pada gambar 4.31 dan 4.32.



Gambar 4. 30 Implementasi *interface* manajemen penggajian.



Gambar 4. 31 Halaman *interface* form laporan gaji asisten dosen

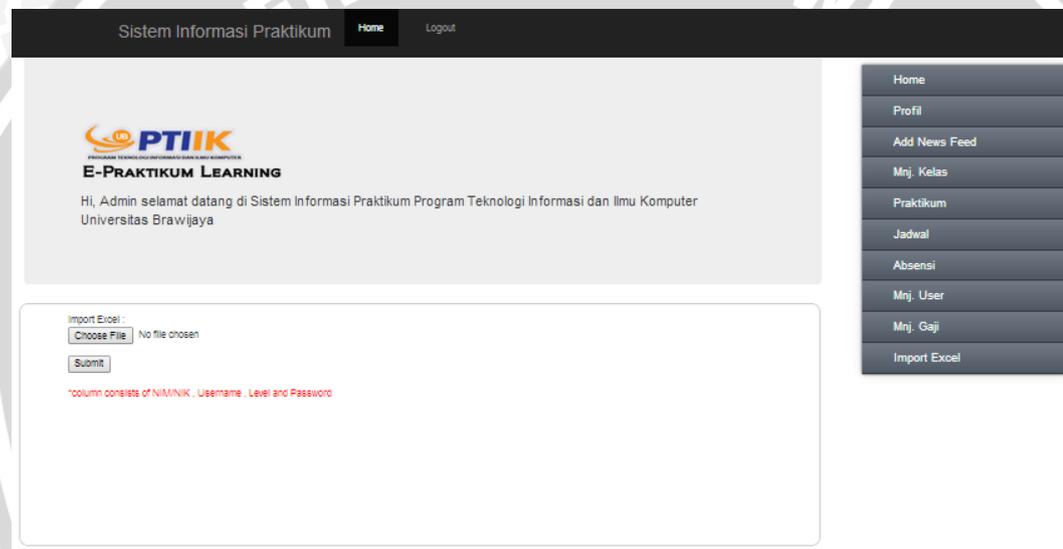


Gambar 4. 32 Implementasi *interface* laporan gaji asisten dan koordinator asisten

4.4.10 Halaman Import Data *User From Excel*

Halaman ini digunakan untuk melakukan *import data user* dari *excel* kedalam basis data. Hak akses halaman ini dimiliki oleh aktor *administrator*. Struktur kolom masukan data yang diperbolehkan adalah:

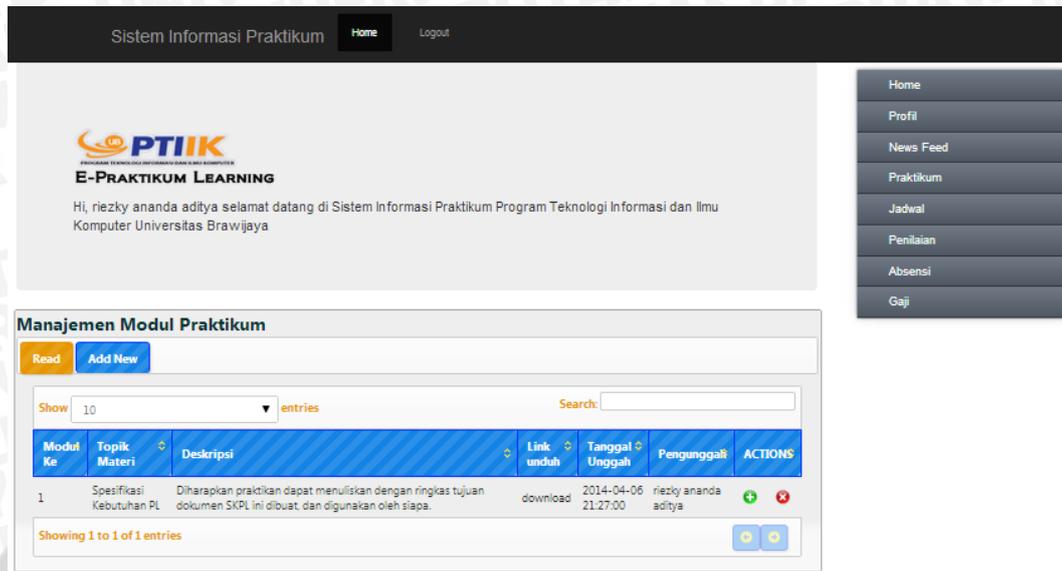
1. Kolom 1: *username* (nim, nik, dsb)
2. Kolom 2: nama
3. Kolom 3: jenis aktor antara lain asisten dosen, praktikan, koordinator asisten dosen, *administrator* dan dosen.
4. Kolom 4: *password*



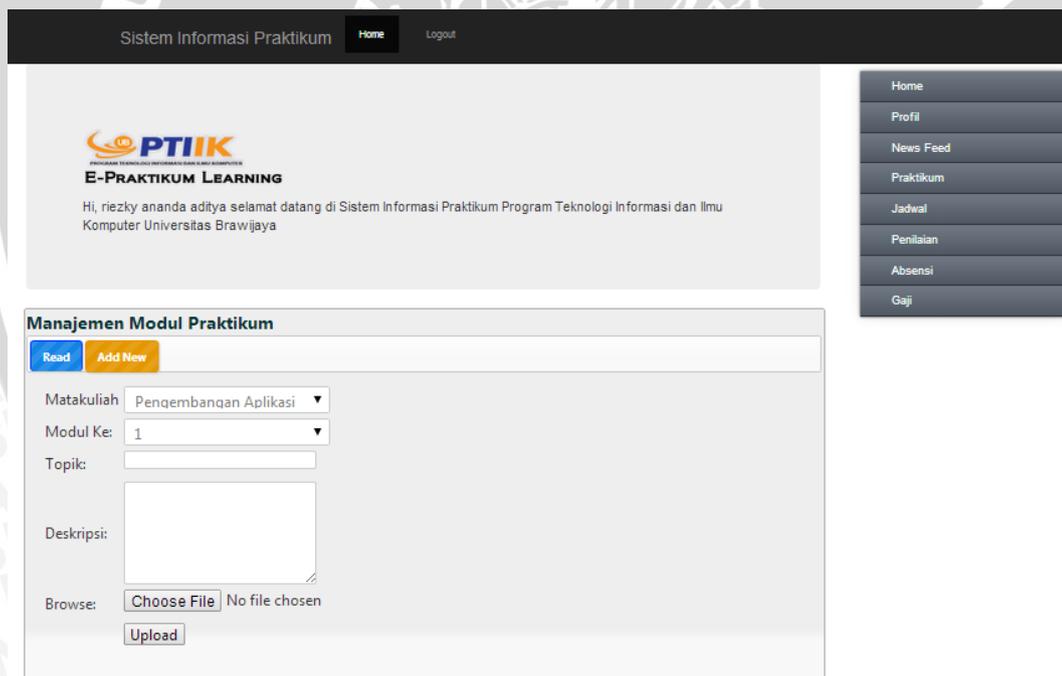
Gambar 4. 33 Implementasi *interface import users from excel*

4.4.11 Halaman Kelola Modul dan Tugas Praktikum

Halaman kelola modul dan tugas praktikum terdiri dari beberapa halaman, diantaranya halaman untuk *upload* modul praktikum, *upload* tugas praktikum dan *download* hasil tugas praktikum mahasiswa. Tampilan halaman untuk mengelola modul praktikum (*delete*, *update* modul) ditunjukkan pada Gambar 4.34. Tampilan halaman untuk menambahkan modul praktikum ditunjukkan pada Gambar 4.35. Hak akses untuk mengelola dan menambahkan modul praktikum dimiliki oleh aktor koordinator asisten dosen.



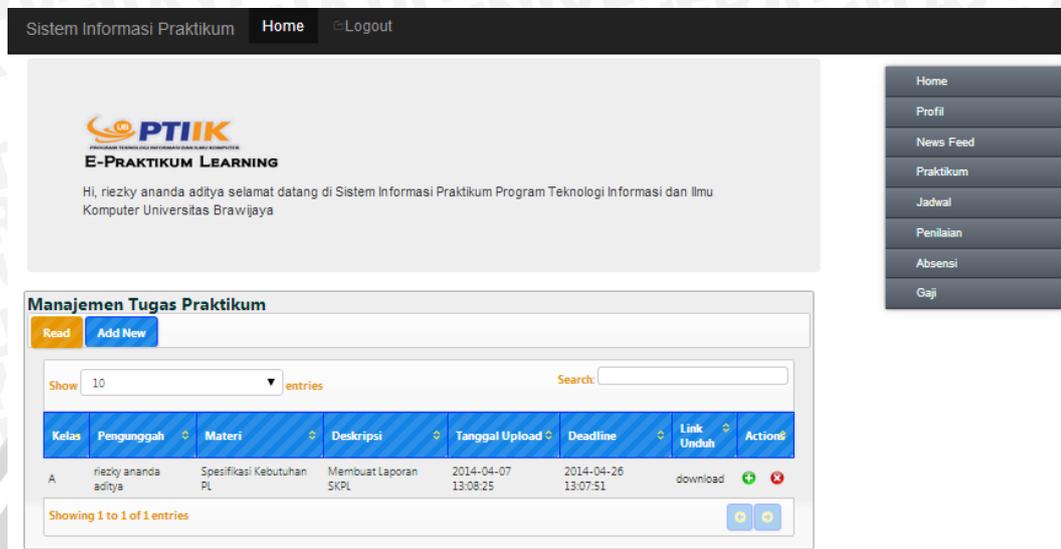
Gambar 4. 34 Implementasi *interface* manajemen modul praktikum



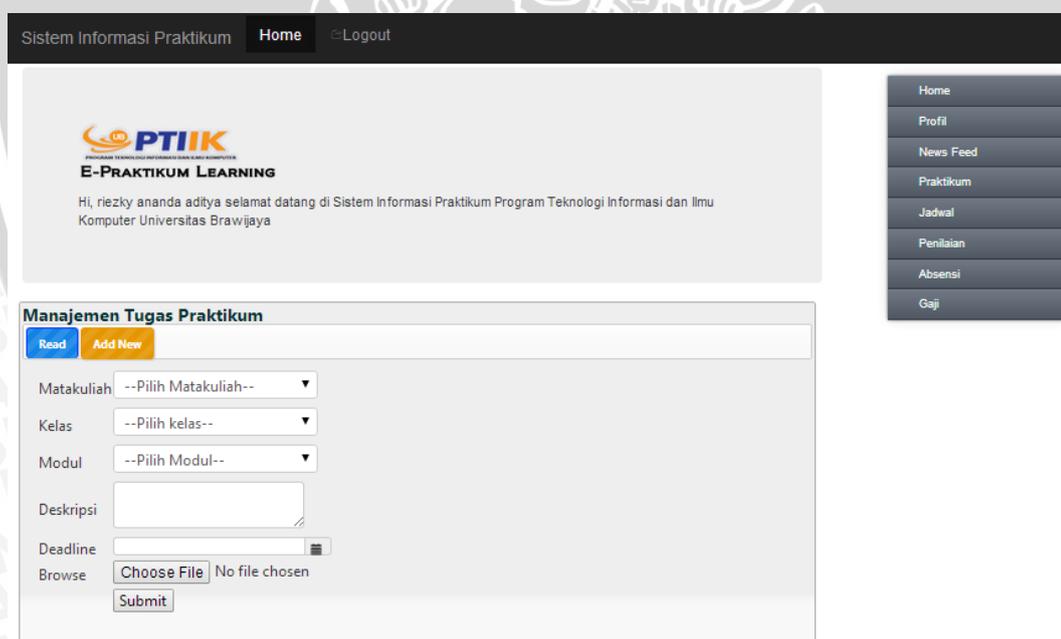
Gambar 4. 35 Implementasi *interface* tambah modul praktikum

Halaman kelola tugas praktikum digunakan untuk mengelola dan menambahkan tugas praktikum mahasiswa. Hak akses halaman ini dimiliki oleh aktor asisten dosen dan koordinator asisten dosen. Tampilan halaman kelola tugas praktikum (*update, delete* tugas) ditunjukkan pada Gambar 4.36. Tampilan

halaman untuk menambahkan (*upload*) tugas praktikum ditunjukkan pada Gambar 4.37.



Gambar 4. 36 Implementasi *interface* kelola tugas praktikum

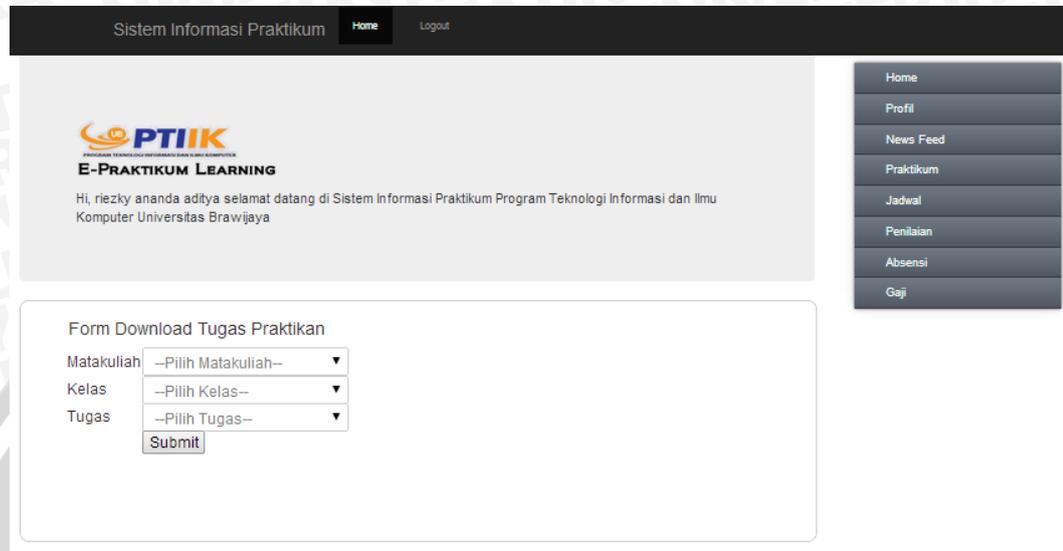


Gambar 4. 37 Implementasi *interface* tambah tugas praktikum

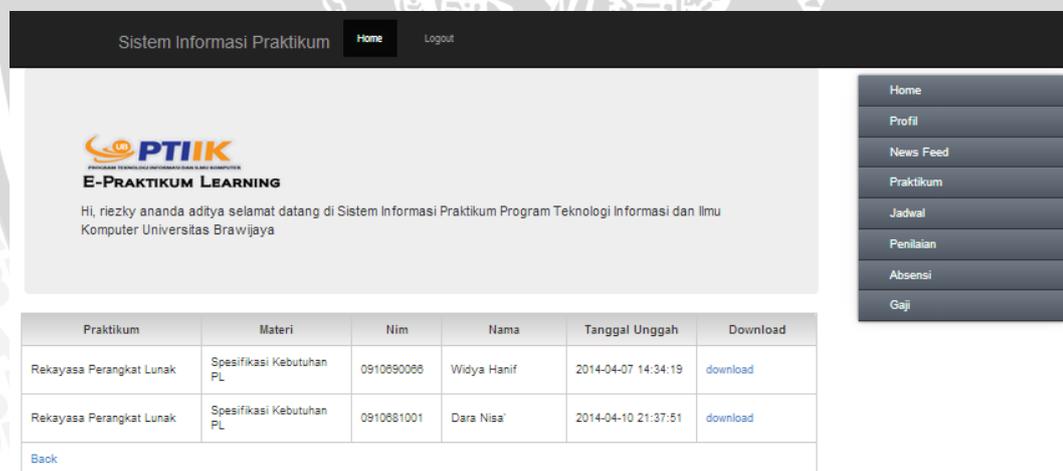
Halaman unduh tugas praktikum digunakan untuk mengunduh hasil tugas praktikum mahasiswa (praktikan). Hak akses halaman ini dimiliki oleh aktor asisten dosen, koordinator asisten dosen dan dosen. Tampilan halaman form untuk



mengunduh hasil tugas praktikum mahasiswa berdasarkan pada kelas dan tugas praktikumnya ditunjukkan pada Gambar 4.38 dan halaman untuk mengunduh tugas ditunjukkan pada Gambar 4.39.



Gambar 4. 38 Implementasi *interface* form untuk menampilkan tugas praktikum



Gambar 4. 39 Implementasi *interface* jawaban tugas praktikum mahasiswa (praktikan)

4.4.12 Halaman Kelola Nilai Tugas Mahasiswa

Halaman kelola nilai tugas mahasiswa digunakan untuk mengelola nilai tugas. Hak akses untuk menginputkan nilai praktikum mahasiswa dimiliki oleh aktor asisten dosen dan atau koordinator asisten dosen yang mengajar pada materi

di kelas tersebut. Tampilan form input nilai praktikum ditunjukkan pada Gambar 4.40 dan tampilan halaman untuk menginputkan nilai mahasiswa berdasarkan kelas dan tugas praktikumnya ditunjukkan pada Gambar 4.41.

Sistem Informasi Praktikum Home Logout

PTIK
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
E-PRAKTIKUM LEARNING

Hi, riezky ananda aditya selamat datang di Sistem Informasi Praktikum Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya

Home
Profil
News Feed
Praktikum
Jadwal
Penilaian
Absensi
Gaji

Form Input Nilai Praktikan

Matakuliah --Pilih Matakuliah--
Kelas --Pilih Kelas--
Tugas --Pilih Tugas--
Submit

Gambar 4. 40 Implementasi *interface* form untuk menampilkan data mahasiswa (praktikan)

Sistem Informasi Praktikum Home Logout

PTIK
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
E-PRAKTIKUM LEARNING

Hi, riezky ananda aditya selamat datang di Sistem Informasi Praktikum Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya

Modul	Nim	Nama	Nilai
Spesifikasi Kebutuhan PL	0910690066	Widya Hanif	<input type="text"/>
Spesifikasi Kebutuhan PL	0910681001	Dara Nisa'	<input type="text"/>

SUBMIT Back

Home
Profil
News Feed
Praktikum
Jadwal
Penilaian
Absensi
Gaji

Gambar 4. 41 Implementasi *interface* untuk input nilai tugas praktikum

Halaman untuk melakukan *update* nilai praktikum terdiri dari dua halaman. Halaman pertama adalah form untuk masuk ke halaman nilai mahasiswa (praktikan) yang akan di-*update*, halaman kedua adalah halaman yang berisi laporan nilai tugas praktikum mahasiswa yang akan di-*update*. Hak akses halaman ini dimiliki oleh aktor asisten dosen dan koordinator asisten dosen. Tampilan halaman form ditunjukkan oleh Gambar 4.42 dan tampilan halaman *update* nilai ditunjukkan oleh Gambar 4.43.

Sistem Informasi Praktikum Home Logout

PTIIK
E-PRAKTIKUM LEARNING

Hi, riezky ananda aditya selamat datang di Sistem Informasi Praktikum Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya

Home
Profil
News Feed
Praktikum
Jadwal
Penilaian
Absensi
Gaji

Form Update Nilai Tugas Praktikum Mahasiswa

Matakuliah --Pilih Matakuliah--
Kelas --Pilih Kelas--
Tugas --Pilih Tugas--
Submit

Gambar 4. 42 Implementasi *interface* form *update* nilai mahasiswa

4.4.13 Halaman Praktikum Mahasiswa

Halaman praktikum mahasiswa (praktikan) berisi daftar praktikum yang ditempuh mahasiswa pada tahun ajaran yang sedang berlangsung. Hak akses yang dimiliki mahasiswa pada halaman ini adalah unduh modul praktikum, unduh tugas praktikum dan unggah hasil tugas praktikum mahasiswa. Tampilan halaman daftar praktikum ditunjukkan pada Gambar 4.43 dan tampilan halaman detail praktikum ditunjukkan pada halaman Gambar 4.44.

Sistem Informasi Praktikum Home Logout

PTIIK
E-PRAKTIKUM LEARNING

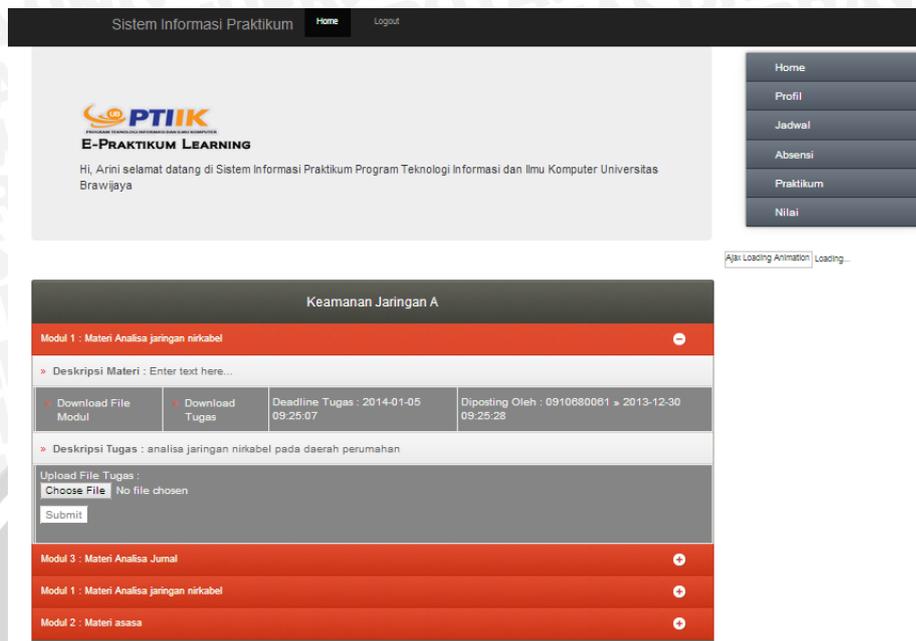
Hi, Arini selamat datang di Sistem Informasi Praktikum Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya

Home
Profil
Jadwal
Absensi
Praktikum
Nilai

Ajax Loading Animation Loading...

Pemrograman Web A	Read More
Keamanan Jaringan A	Read More
BASIS DATA LANJUT A	Read More
Logika Informatika B	Read More

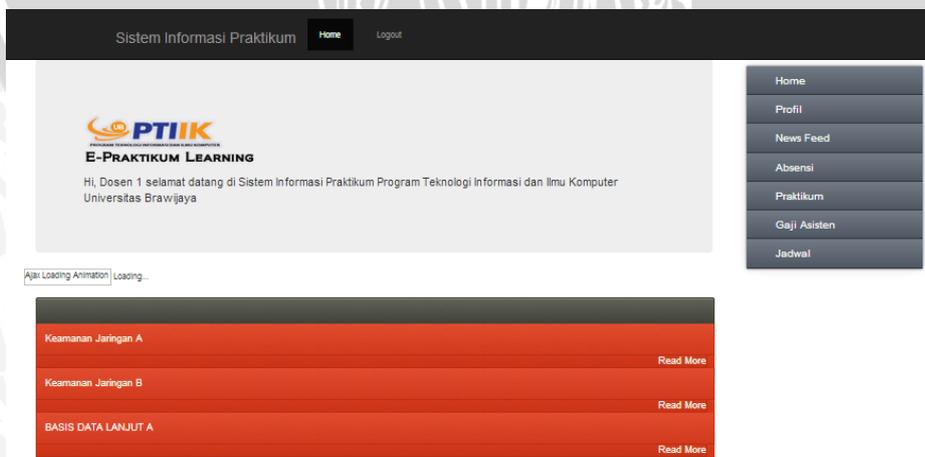
Gambar 4. 43 Halaman *interface* daftar praktikum yang ditempuh



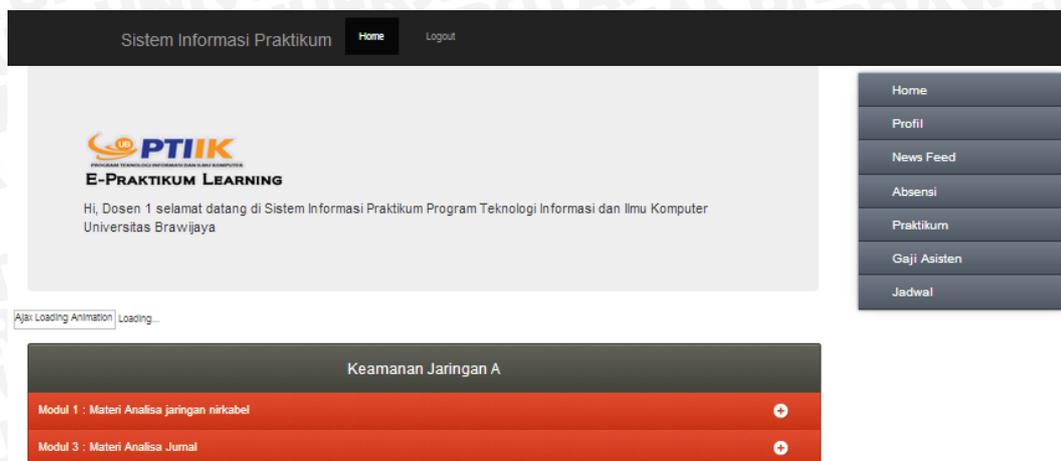
Gambar 4. 44 Implementasi *interface* detail praktikum

4.4.14 Halaman Daftar Ajar Praktikum

Halaman daftar ajar praktikum digunakan untuk memantau kegiatan praktikum pada tahun ajaran yang sedang berlangsung. Hak akses halaman ini dimiliki oleh aktor dosen. Dosen dapat melihat modul yang sedang diajarkan oleh asisten dosen dan dapat mengunduh modul dan tugas yang diberikan oleh asisten dosen kepada mahasiswa. Tampilan halaman daftar praktikum ditunjukkan pada Gambar 4.45 dan halaman detail praktikum ditunjukkan pada Gambar 4.46.



Gambar 4. 45 Implementasi *interface* daftar praktikum yang diajarkan dosen



Gambar 4. 46 Implementasi *interface* detail modul dan tugas praktikum



BAB V PENGUJIAN DAN ANALISIS

Bab pengujian akan membahas prosedur pengujian aplikasi. Teknik yang digunakan pada penelitian ini adalah *blackbox testing* yaitu dengan menggunakan metode *functional testing* dan pengujian UAT (*User Acceptance Test*) dengan menggunakan pendekatan TAM (*Test Acceptance Model*).

5.1 Pengujian Fungsionalitas

Secara umum sistem informasi praktikum yang dikembangkan dalam penelitian ini membagi fungsionalitasnya sesuai dengan tingkatan *user* yang mengakses sistem. Pada bagian pengujian fungsional dengan metode *blackbox* peneliti terlebih dahulu menyusun daftar *test requirement* pada setiap fitur yang ada. Hal ini dilakukan agar pada pembuatan *test case* bisa mengacu pada susunan *test requirement* yang telah dibuat, berikut daftar *test requirement*:

Tabel 5. 1 Daftar *test requirements*

No	Fungsi	Task	Transaction
1	Home	Validasi halaman <i>home</i>	Apakah halaman <i>home</i> dapat ditampilkan?
2	Profil	Validasi form <i>update password</i>	Apakah halaman profil dapat menampilkan data <i>user</i> ? Apakah <i>button submit</i> pada <i>update form</i> dapat berfungsi dan <i>password</i> berhasil di- <i>update</i> ?
3	Manajemen Kelas	Validasi form <i>create new class</i>	Apakah <i>user</i> dapat menampilkan form untuk menambahkan kelas baru? Apakah form untuk menambahkan kelas baru dapat berfungsi untuk menyimpan data kelas ke dalam basis data? Apakah dapat menampilkan data kelas yang telah ditambahkan? Apakah <i>button update</i> dan <i>delete</i> berfungsi?
4.	Manajemen Jadwal	Validasi form tambah jadwal	Apakah form untuk tambah jadwal berhasil ditampilkan? Apakah ketika diinputkan data hari, waktu dan ruang yang sama, data akan tetap bisa disimpan



			ke dalam basis data?
		Validasi form <i>update schedule</i>	Apakah form dapat berfungsi untuk melakukan <i>update</i> jadwal?
			Apakah ketika diinputkan data hari, waktu dan ruang yang sama data akan tetap berhasil disimpan ke dalam basis data?
5	Manajemen Kelas Praktikum MHS	Validasi form input data MHS	Apakah form untuk input data MHS berhasil ditampilkan?
			Apakah form dapat berfungsi dan data berhasil disimpan ke dalam basis data?
6	Managemen Praktikum	Validasi form <i>add new</i> praktikum pada tahun ajaran tertentu	Apakah form tambah data praktikum baru berhasil ditampilkan?
			Apakah <i>textfield</i> dapat diinputkan data dan <i>button submit</i> dapat berfungsi sehingga data berhasil disimpan ke dalam basis data?
		Validasi form <i>upload</i> modul praktikum	Apakah form <i>upload</i> modul praktikum berhasil ditampilkan?
			Apakah <i>textfield</i> dapat diinputkan <i>text</i> dan <i>button browse</i> dapat berfungsi untuk mencari letak modul praktikum yang akan diunggah dan apakah modul berhasil disimpan ke dalam basis data?
		Validasi form <i>upload</i> tugas praktikum	Apakah form <i>upload</i> modul praktikum berhasil ditampilkan?
			Apakah <i>textfield</i> dapat diinputkan <i>text</i> dan <i>button browse</i> dapat berfungsi untuk mencari letak <i>file</i> tugas praktikum yang akan diunggah dan apakah <i>file</i> tugas berhasil disimpan ke dalam basis data?
		Validasi form <i>update</i> modul praktikum	Apakah form <i>update</i> modul praktikum berhasil ditampilkan?
			Apakah form <i>update</i> berhasil merubah dan data berhasil disimpan ke dalam basis data?
7	Manajemen Absensi	Validasi form <i>input</i> data absensi MHS	Apakah form untuk input data absensi berhasil ditampilkan dan apakah form berhasil menampilkan data mahasiswa?
			Apakah <i>checkbox</i> dapat berfungsi?
			Apakah <i>button submit</i> dapat berfungsi?
		Validasi form <i>input</i> absensi asisten	Apakah form untuk input absensi asisten berhasil ditampilkan?
			Apakah form input absensi berhasil menyimpan data ke dalam basis data?
		Validasi halaman laporan absensi MHS	Apakah berhasil menampilkan data absensi mahasiswa?
		Validasi halaman laporan absensi asisten	Apakah berhasil menampilkan data absensi asisten?
8	Manajemen Penggajian Asisten Dosen	Validasi form <i>add salary</i>	Apakah data praktikum yang akan ditambahkan nominal <i>salary</i> -nya berhasil ditampilkan?

			Apakah form tambah gaji berhasil menyimpan data nominal gaji asisten dosen ke dalam basis data?
		Validasi halaman <i>read salary</i>	Apakah berhasil menampilkan data gaji asisten mahasiswa?
9	Manajemen <i>User</i>	Validasi form <i>add new user</i>	Apakah form tambah <i>user</i> baru berhasil ditampilkan? Apakah <i>textfield</i> dapat diinputkan <i>char</i> ? Apakah <i>button submit</i> dapat berfungsi?
		Validasi form <i>update praktikan(MHS)</i>	Apakah form <i>update</i> dapat ditampilkan? Apakah form <i>update</i> berhasil menyimpan data baru ke dalam basis data?
		Validasi form <i>add new asisten</i>	Apakah form untuk tambah asisten dosen berhasil ditampilkan? Apakah <i>textfield</i> dapat diinputkan <i>char</i> ? Apakah <i>button submit</i> dapat berfungsi?
		Validasi form <i>update asisten</i>	Apakah form <i>update</i> dapat ditampilkan? Apakah form <i>update</i> berhasil menyimpan data baru ke dalam basis data?
		Validasi form <i>add dosen</i>	Apakah form untuk menambahkan dosen berhasil ditampilkan? Apakah form untuk menambahkan data dosen berhasil menyimpan data ke dalam basis data?
		Validasi form <i>import data from Excel</i>	Apakah form berhasil ditampilkan? Apakah <i>button browse</i> dapat berfungsi? Apakah form <i>import data</i> berhasil menyimpan data ke dalam basis data?
10	Managemen <i>News feed</i>	Validasi form <i>create news feed</i>	Apakah form untuk unggah berita berhasil ditampilkan? Apakah <i>textfield</i> dapat diinputkan <i>text</i> ? Apakah <i>button submit</i> dapat berfungsi?
		Validasi form <i>update news feed</i>	Apakah form untuk <i>update</i> berita berhasil ditampilkan? Apakah <i>textfield</i> dapat diinputkan <i>text</i> ? Apakah <i>button submit</i> dapat berfungsi?

5.1.1 Skenario Pengujian

Pada pengerjaan *functional test* terdapat beberapa parameter dalam pengujian berikut penjelasannya:

- Fitur kode : sesuai dengan penomoran fitur di tabel fitur *test requirement*.
- Case #* : urutan *case*.
- Test case* : keterangan kasus yang dicoba.

- d. *Data value* : detail data yang dicoba.
- e. *Expected res.* : hasil yang seharusnya terjadi.
- f. *Actual res.* : hasil yang sebenarnya terjadi.
- g. *Pass/failure* : kesesuaian antara *expected result* dengan *actual result*. Jika sesuai maka *pass* kalau tidak sama maka *failure*.

Berikut ini adalah scenario pengujian pada fitur-fitur yang dijabarkan pada tabel 5.1.

5.1.1.1 Home

Skenario 1.1

Skenario ini digunakan untuk menguji bahwa halaman *Home* berhasil diakses oleh pengguna, baik meliputi *content* yang berisi tentang berita-berita sistem informasi praktikum maupun menu yang menyertainya. Kesalahan sistem pada fungsi ini menyebabkan menu tidak dapat diakses oleh pengguna dan karena itu sistem tidak dapat bekerja sebagaimana mestinya.

Tabel 5. 2 Skenario pengujian halaman home

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS / FAILURE
1.1.1	A	Membuka halaman HOME	Konten HOME	Menampilkan konten <i>home</i> yang berisi berita-berita praktikum dan menu navigasi sistem	Menampilkan konten <i>home</i> yang berisi berita-berita praktikum dan menu navigasi sistem	<i>Pass</i>

5.1.1.2 Profil

Skenario 2.1

Skenario ini digunakan untuk menguji bahwa halaman profil berhasil menampilkan profil pengguna dan dapat melakukan *update* data pada profil pengguna. Kesalahan sistem pada fungsi ini menyebabkan sistem tidak dapat memberikan fitur untuk *update profil*.

Tabel 5. 3 Skenario pengujian fitur *update* profil

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS / FAILURE
2.1.1	A	Membuka halaman Profil	Data profil pengguna	Menampilkan data profil pengguna	Menampilkan data profil pengguna	<i>Pass</i>

2.1.2	B	Submit form update profil	Data baru pengguna yang diinputkan pada form	Button submit berfungsi dan data berhasil diinputkan ke dalam basis data	Button submit berfungsi dan data berhasil diinputkan ke dalam basis data	Pass
-------	---	---------------------------	--	--	--	------

5.1.1.3 Manajemen Kelas

Skenario 3.1

Skenario ini digunakan untuk menguji bahwa sistem berhasil menampilkan form untuk menambahkan kelas baru dan menguji bahwa form dapat berfungsi dengan benar, dimana data yang diinputkan pada form berhasil disimpan ke dalam basis data. Kesalahan sistem pada fungsi ini mengakibatkan proses manajemen maha-siswa untuk menginputkan data mahasiswa ke dalam kelas praktikum tidak dapat dilakukan dan sistem gagal berfungsi sebagaimana mestinya.

Tabel 5. 4 Skenario pengujian fitur manajemen kelas

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
3.1.1	A	Membuka halaman <i>create new</i> kelas	Form untuk menambah kan kelas baru	Menampilkan form untuk menambahkan kelas baru	Menampilkan form untuk menambahkan kelas baru	Pass
3.1.2	B	Mengisi form <i>create new class</i>	Memilih program studi, mata kuliah dan menginput kan jumlah kelas	Data berhasil tersimpan ke dalam basis data	Data berhasil tersimpan ke dalam basis data	Pass

5.1.1.4 Manajemen Jadwal

Skenario 4.1

Skenario ini digunakan untuk melakukan validasi form yang digunakan untuk memasukkan jadwal praktikum ke dalam basis data. Pengujian form input jadwal praktikum juga dilengkapi dengan pengujian fitur cek kondisi ketika data yang dimasukkan sama dengan data yang sudah tersimpan di dalam basis data. Kesalahan sistem pada fungsi ini mengakibatkan terdapat duplikasi data.

Tabel 5. 5 Skenario pengujian form tambah jadwal

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
4.1.1	A	Input jadwal praktikum ke dalam form	Data Mata kuliah: DTD Kelas: A Hari: Senin Waktu: 17:00 – 18:00 Ruang: A2.12	Data berhasil disimpan ke dalam basis data	Data berhasil disimpan ke dalam basis data	Pass
4.1.2	B	Input jadwal praktikum dengan kondisi waktu dan ruang yang sama dengan case A	Data Mata kuliah: RPL Kelas: A Hari: Senin Waktu: 17:30 – 18:30	Data tidak dapat disimpan.	Ruang A2.12 tidak ada pada pilihan ruang yang tersedia. Data tidak dapat disimpan.	Pass

Skenario 4.2

Skenario ini digunakan untuk melakukan validasi form *update schedule*. Pengujian pada skenario ini juga dilengkapi pengujian terhadap kondisi input data hari, waktu dan ruang yang sama dengan data lain yang tersimpan di dalam basis data. Kegagalan pada fungsi ini mengakibatkan data tidak dapat diubah atau terjadi duplikasi data hari, ruang dan waktu yang sama di dalam basis data.

Tabel 5. 6 Skenario pengujian form *update* jadwal

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
4.2.1	A	Memasukkan data jadwal praktikum ke dalam form <i>update</i>	Data lama Mata kuliah: DTD Kelas: A Data baru: Hari: Senin Waktu: 18:00 – 19:00 Ruang: A2.13	Data berhasil disimpan ke dalam basis data	Data berhasil disimpan ke dalam basis data	Pass
4.2.2		Memasukkan data jadwal praktikum yang sama dgn data yang tersimpan di basis data ke dalam form <i>update</i>	Mata kuliah: DTD Kelas: B Data <i>update</i> : Hari: Senin Waktu: 18:20 – 19:20 Ruang: A2.13	Data tidak dapat disimpan.	Ruang A2.13 tidak ada pada pilihan ruang yang tersedia. Data tidak dapat disimpan.	Pass

5.1.1.5 Manajemen Kelas Praktikum Mahasiswa

Skenario 5.1

Skenario pengujian ini dilakukan untuk melakukan validasi form input data mahasiswa ke dalam kelas praktikum. Kegagalan pada fungsi ini mengakibatkan data mahasiswa tidak dapat tersimpan ke dalam basis data dan berpengaruh kepada pengelolaan data lain pada proses selanjutnya.

Tabel 5. 7 Skenario pengujian form input mahasiswa ke kelas praktikum

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
5.1.1	A	Menampilkan halaman untuk input data mahasiswa ke kelas praktikum	Klik kiri menu “Tambah Mahasiswa ke Kelas Praktikum”	Menampilkan form untuk input data mahasiswa.	Menampilkan form untuk input data mahasiswa	Pass
5.1.2		Mengisi form <i>input</i> mahasiswa lalu menekan tombol <i>submit</i>	Memilih program studi, mata kuliah dan kelas pada <i>combobox</i> dan memasukkan data nim mahasiswa lalu menekan tombol <i>submit</i>	Data mahasiswa yang baru ditambahkan berhasil tersimpan ke dalam basis data	Data mahasiswa yang baru ditambahkan berhasil tersimpan ke dalam basis data	Pass

5.1.1.6 Manajemen Praktikum

Skenario 6.1

Skenario ini digunakan untuk menguji fitur untuk tambah data praktikum. Kegagalan pada fitur ini mengakibatkan data praktikum tidak dapat disimpan ke dalam basidata dan aplikasi tidak dapat digunakan.

Tabel 5. 8 Skenario pengujian form tambah praktikum

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
6.1.1	A	Menampilkan halaman untuk input data praktikum	Klik kiri menu “Add New”	Menampilkan form untuk input praktikum	Menampilkan form untuk input praktikum	Pass
6.1.2	B	Mengisi form tambah praktikum baru dengan data praktikum dan menekan tombol <i>submit</i>	Mengisi <i>text-field</i> kode praktikum dan praktikum serta memilih program studi pada <i>combobox</i>	Data praktikum yang baru ditambahkan berhasil tersimpan ke dalam basis data	Data praktikum yang baru ditambahkan berhasil tersimpan ke dalam basis data	Pass

Skenario 6.2

Skenario pengujian ini digunakan untuk melakukan validasi fitur *upload* modul praktikum. Kesalahan sistem pada fitur ini mengakibatkan modul praktikum tidak dapat diunggah ke dalam basis data.

Tabel 5. 9 Skenario pengujian form upload modul praktikum

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
6.2.1	A	Menampilkan halaman untuk <i>upload</i> modul praktikum	Klik kiri menu “Upload Modul Praktikum”	Menampilkan form untuk <i>upload</i> modul praktikum	Menampilkan form untuk <i>upload</i> modul praktikum	Pass
6.2.2	B	Mengisi form <i>upload</i> modul praktikum dan menekan tombol <i>submit</i>	Mengisi <i>textfield</i> topik dan deskripsi serta memilih <i>combobox</i> modul dan <i>browse</i> lokasi <i>file</i> yang akan diunggah	Modul praktikum yang baru ditambahkan berhasil tersimpan ke dalam basis data	Modul praktikum yang baru ditambahkan berhasil tersimpan ke dalam basis data	Pass

Skenario 6.3

Skenario pengujian ini digunakan untuk melakukan validasi fitur *upload* tugas praktikum. Kesalahan sistem pada fitur ini mengakibatkan tugas praktikum tidak dapat diunggah ke dalam basis data.

Tabel 5. 10 Skenario pengujian form upload tugas praktikum

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
6.3.1	A	Menampilkan halaman untuk <i>upload</i> tugas praktikum	Klik kiri menu “Upload Tugas Praktikum”	Menampilkan form untuk <i>upload</i> tugas praktikum	Menampilkan form untuk <i>upload</i> tugas praktikum	Pass
6.3.2	B	Mengisi form <i>upload</i> tugas praktikum dan menekan tombol <i>submit</i>	Mengisi <i>textfield</i> pada form serta memilih <i>combobox</i> modul dan <i>browse</i> lokasi <i>file</i> yang akan diunggah	Tugas praktikum yang baru ditambahkan berhasil tersimpan ke dalam basis data	Tugas praktikum yang baru ditambahkan berhasil tersimpan ke dalam basis data	Pass

Skenario 6.4

Skenario pengujian ini digunakan untuk melakukan validasi fitur *update* modul praktikum. Kesalahan sistem pada fitur ini mengakibatkan modul praktikum tidak dapat di *update*.

Tabel 5. 11 Skenario pengujian form *update* modul praktikum

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/FAILURE
6.4.1	A	Menampilkan kotak dialog untuk <i>update</i> modul praktikum	Klik kiri <i>button</i> " <i>update</i> "	Menampilkan form untuk <i>update</i> modul praktikum	Menampilkan form untuk <i>upload</i> modul praktikum	Pass
6.4.2	B	Mengisi form <i>upload</i> tugas praktikum dan menekan tombol <i>submit</i>	Mengisi <i>text-field</i> pada form serta memilih <i>combobox</i> modul dan <i>browse</i> lokasi <i>file</i> yang akan diunggah	Modul praktikum yang baru di <i>update</i> berhasil tersimpan ke dalam basis data	Modul praktikum yang baru di <i>update</i> berhasil tersimpan ke dalam basis data	Pass

5.1.1.7 Manajemen Absensi

Skenario 7.1

Skenario pengujian ini digunakan untuk melakukan validasi form *input* absensi mahasiswa. Kesalahan sistem pada fitur ini mengakibatkan data absensi tidak dapat disimpan ke dalam basis data.

Tabel 5. 12 Skenario pengujian input absensi mahasiswa

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/FAILURE
7.1.1	A	Menampilkan form <i>input</i> absensi mahasiswa	Memilih <i>combobox</i> kelas praktikum dan menekan tombol <i>submit</i> .	Menampilkan form <i>input</i> absensi mahasiswa.	Menampilkan form <i>input</i> absensi mahasiswa.	Pass
7.1.2	B	Cek fungsi <i>checkbox</i>	Cek pada <i>radio button</i> kehadiran.	<i>Checkbox</i> dapat berfungsi.	<i>Checkbox</i> dapat berfungsi.	Pass
7.1.3	C	Cek fungsi <i>button submit</i>	<i>Checklist checkbox</i> lalu menekan <i>button submit</i>	Data absensi mahasiswa berhasil disimpan ke dalam basis data.	Data absensi mahasiswa berhasil disimpan ke dalam basis data.	Pass

Skenario 7.2

Skenario pengujian ini digunakan untuk melakukan validasi form *input* absensi asisten. Kesalahan sistem pada fitur ini mengakibatkan data tidak dapat disimpan ke dalam basis data.

Tabel 5. 13 Skenario pengujian input data kehadiran asisten dosen

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
7.2.1	A	Menampilkan form <i>input</i> absensi asisten	Klik kiri menu “ <i>Entry Absensi</i> ”	Menampilkan form <i>input</i> absensi asisten data menampilkan data absensi	Menampilkan form <i>input</i> absensi asisten data menampilkan data absensi	<i>Pass</i>
7.2.2	B	Mengisi form <i>input</i> absensi asisten	Mengisi data kehadiran asisten pada form lalu menekan <i>button submit</i>	Data absensi asisten berhasil disimpan ke dalam basis data	Data absensi asisten berhasil disimpan ke dalam basis data	<i>Pass</i>

Skenario 7.3

Skenario pengujian ini digunakan untuk melakukan validasi laporan data absensi mahasiswa yang ditampilkan. Kesalahan sistem pada fitur ini mengakibatkan laporan tidak dapat ditampilkan atau data pada laporan yang ditampilkan salah.

Tabel 5. 14 Skenario pengujian laporan absensi mahasiswa

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
7.3.1	A	Memilih <i>combobox</i> mata kuliah dan kelas untuk melihat laporan absensi mahasiswa	Memilih Mata kuliah DTD kelas A	Menampilkan laporan absensi mahasiswa pada mata kuliah DTD kelas A	Menampilkan laporan absensi mahasiswa pada mata kuliah DTD kelas A	<i>Pass</i>

Skenario 7.4

Skenario pengujian ini digunakan untuk melakukan validasi laporan data absensi asisten dosen yang ditampilkan. Kesalahan sistem pada fitur ini mengakibatkan laporan tidak dapat ditampilkan atau data pada laporan yang ditampilkan salah.

Tabel 5. 15 Skenario pengujian laporan absensi asisten dosen

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
7.4.1	A	Memilih <i>combobox</i> mata kuliah untuk melihat laporan absensi mhs	Memilih Mata kuliah DTD	Menampilkan laporan absensi asisten pada mata kuliah DTD	Menampilkan laporan absensi asisten pada mata kuliah DTD	<i>Pass</i>

5.1.1.8 Manajemen Penggajian Asisten Dosen

Skenario 8.1

Skenario pengujian ini digunakan untuk melakukan validasi form untuk menginputkan nilai gaji asisten ke dalam basis data. Kesalahan sistem pada fitur ini mengakibatkan data tidak dapat disimpan ke dalam basis data sehingga asisten dosen tidak dapat mengakses informasi gaji yang mereka dapatkan.

Tabel 5. 16 Skenario pengujian form input gaji asisten

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
8.1.1	A	Menampilkan halaman untuk <i>setting</i> gaji asisten	Klik kiri menu "Set Gaji"	Menampilkan data praktikum yang akan diinputkan besaran nilai gajinya.	Menampilkan praktikum yang akan diinputkan besaran nilai gajinya.	Pass
8.1.2	B	Mengisi form untuk <i>setting</i> besaran nilai gaji asisten	Mengisi <i>textfield</i> besaran gaji pada form lalu menekan tombol <i>submit</i>	Data berhasil tersimpan ke dalam basis data	Data berhasil tersimpan ke dalam basis data	Pass

Skenario 8.2

Skenario pengujian ini digunakan untuk melakukan validasi laporan gaji asisten dosen. Kesalahan sistem pada fungsi ini mengakibatkan laporan tidak dapat ditampilkan atau data laporan yang ditampilkan salah.

Tabel 5. 17 Skenario pengujian halaman laporan gaji asisten dosen

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
8.2.1	A	Menampilka n laporan gaji asisten dosen	Pilih <i>combobox</i> mata kuliah DTD lalu tekan tombol <i>submit</i>	Menampilkan laporan gaji asisten dosen DTD	Menampilkan laporan gaji asisten dosen DTD	Pass

5.1.1.9 Manajemen User

Skenario 9.1

Skenario pengujian ini digunakan untuk melakukan validasi form tambah *user* (praktikan, dosen, asisten dosen, koordinator asisten dosen dan administrator). Kesalahan pada sistem ini mengakibatkan data *user* tidak dapat disimpan ke dalam basis data dan mengakibatkan sistem tidak dapat berjalan sebagaimana mestinya.

Tabel 5. 18 Skenario pengujian form *add new user*

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
9.1.1	A	Menampilkan form <i>create new user</i>	Pilih menu "Management All Users"	Menampilkan form untuk menambahkan <i>user</i> baru	Menampilkan form untuk menambahkan <i>user</i> baru	Pass
	B	Cek <i>textfield</i> dengan inputan char	Input data kode <i>user</i> (NIM, NIK) , nama dan level <i>user</i> (aktor) ke dalam <i>textfield</i>	<i>Textfield</i> dapat diinputkan data	<i>Textfield</i> dapat diinputkan data	Pass
	C	Simpan data pengguna baru ke dalam basis data	Input data kode <i>user</i> (NIM, NIK) , nama dan level <i>user</i> ke dalam <i>textfield</i> dan tekan tombol <i>submit</i>	Data pengguna baru berhasil tersimpan ke dalam basis data	Data pengguna baru berhasil tersimpan ke dalam basis data	Pass

Skenario 9.2

Skenario pengujian ini digunakan untuk melakukan validasi form *update users*. Kesalahan sistem pada fungsi ini mengakibatkan data *user* tidak dapat di-*update*.

Tabel 5. 19 Skenario pengujian form *update user*

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
9.2.1	A	Menampilkan form <i>update</i>	Klik kiri <i>button update</i>	Menampilkan kotak dialog yang berisi form untuk <i>update</i> data <i>user</i>	Menampilkan kotak dialog yang berisi form untuk <i>update</i> data <i>user</i>	Pass
	B	Cek <i>textfield</i> dengan inputan char	Input data kode <i>user</i> (NIM, NIK) , nama dan level <i>user</i> ke dalam <i>textfield</i>	<i>Textfield</i> dapat diinputkan data	<i>Textfield</i> dapat diinputkan data	Pass
	C	Simpan data baru pengguna ke dalam basis data	Input data kode <i>user</i> (NIM, NIK) , nama dan level <i>user</i> ke dalam <i>textfield</i> dan tekan tombol <i>submit</i>	Berhasil melakukan <i>update</i> data pengguna	Data pengguna baru berhasil tersimpan ke dalam basis data	Pass

Skenario 9.6

Skenario pengujian ini berfungsi untuk melakukan validasi fitur *import* data dari excel ke basis data. Kesalahan sistem pada fungsi ini mengakibatkan data tidak berhasil disimpan ke dalam basis data.

Tabel 5. 20 Skenario pengujian form *import data users from excel*

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
9.6.1	A	Menampilkan halaman form <i>import data from Excel</i>	Menampilkan halaman form	Menampilkan halaman form	Menampilkan halaman form	Pass
9.6.2	B	Cek fungsi <i>button browse</i>	Klik kiri <i>button browse</i> dan mencari lokasi file excel yang akan diimport ke dalam basidata	<i>Button browse</i> berhasil menampilkan kotak dialog pencarian <i>file</i>	<i>Button browse</i> berhasil menampilkan kotak dialog pencarian <i>file</i>	Pass
9.6.3	C	Menambahkan data <i>user</i> melalui file excel ke dalam basis data	Klik kiri <i>button browse</i> dan mencari lokasi file excel yang akan diimport ke dalam basidata lalu menekan tombol <i>submit</i>	Data berhasil disimpan kedalam basis data	Data berhasil disimpan kedalam basis data	Pass

5.1.1.10 Manajemen *News feed*

Skenario 10.1

Skenario pengujian ini digunakan untuk melakukan validasi form untuk menambahkan berita. Kesalahan sistem pada fungsi ini mengakibatkan informasi berita tidak dapat disimpan dan di-*publish* ke sistem informasi.

Tabel 5. 21 Skenario pengujian form *add news feed*

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
10.1.1	A	Menampilkan halaman form <i>create news feed</i>	Klik menu “tambah baru”	Menampilkan halaman form untuk menambahkan berita	Menampilkan halaman form untuk menambahkan berita	Pass

10.1.2	B	Cek <i>textfield</i> dengan inputan char	Input data <i>title</i> dan isi ke dalam <i>textfield</i>	<i>Textfield</i> dapat diinputkan data	<i>Textfield</i> dapat diinputkan data	Pass
10.1.3	C	Cek fungsi <i>button submit</i>	Input data <i>title</i> dan isi ke dalam <i>textfield</i> dan tekan <i>button submit</i>	Berhasil menyimpan berita ke dalam basis data	Berhasil menyimpan berita ke dalam basis data	Pass

Skenario 10.2

Skenario pengujian ini digunakan untuk melakukan validasi form *update news feed*. Kesalahan sistem pada fungsi ini mengakibatkan berita yang telah tersimpan di dalam basis data tidak dapat dilakukan *update* data.

Tabel 5. 22 Skenario pengujian form *update news feed*

FITUR CODE	CASE #	TEST CASE	DATA VALUE	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAILURE
10.2.1	A	Menampilkan halaman form <i>update news feed</i>	Klik menu “tambah baru” dan klik kiri <i>button update</i>	Menampilkan form pada kotak dialog <i>update</i>	Menampilkan form pada kotak dialog <i>update</i>	Pass
10.2.2	B	Cek <i>textfield</i> dengan inputan char	Input data baru : <i>title</i> dan isi ke dalam <i>textfield</i>	<i>Textfield</i> dapat diinputkan data	<i>Textfield</i> dapat diinputkan data	Pass
10.2.3	C	Cek fungsi <i>button submit</i>	Input data baru: <i>title</i> dan isi ke dalam <i>textfield</i> dan tekan <i>button submit</i>	Data berhasil di <i>update</i>	Data berhasil di <i>update</i>	Pass

5.2 User Acceptance Test

User acceptance test diuji cobakan pada sistem informasi untuk mengukur apakah fungsional sistem telah memenuhi kebutuhan pengguna, ditinjau dari sisi pengguna. *User acceptance test* diuji cobakan kepada 30 orang di lingkungan PTIIK yang dipilih secara acak. Pemilihan jumlah 30 orang ini dikarenakan sudah mencukupi karena eksperimen yang dilakukan adalah eksperimen kecil, dari jumlah 30 orang yang dipilih tersebut diantaranya adalah laboran, mahasiswa dan dosen. Data pengujian adalah persentase jumlah skor yang diperoleh dari

jawaban responden atas pertanyaan mengenai kinerja sistem yang telah dibuat oleh peneliti.

5.2.1 Hasil Pengujian *User Acceptance Test*

Tabel 5. 23 Hasil Pengujian UAT

No	Aspek	Pertanyaan	1	2	3	4	5
			(%)				
1	<i>Interaction</i>	Sistem informasi praktikum mudah untuk dioperasikan?	0	0	10	90	0
2	<i>Interaction</i>	Susunan menu pada sistem ini mudah untuk dipahami?	0	0	20	80	0
3	<i>Interaction</i>	Sistem memiliki tampilan yang menarik?	0	0	13	87	0
4	<i>Performance</i>	Sistem memiliki navigasi cepat untuk pindah ke halaman lain?	0	0	17	83	0
5	Persepsi kemudahan penggunaan (<i>perceive ease of use</i>)	Sistem menyediakan akses cepat dan mudah untuk mencari informasi?	0	0	20	80	0
6	<i>Information</i>	Sistem menyediakan informasi yang relevan?	0	0	10	90	0
7	<i>Information</i>	Sistem menyediakan konten informasi yang mudah dibaca?	0	0	0	100	0
8	Persepsi pemanfaatan (<i>perceived usefulness</i>)	Sistem dapat mengelola data praktikum dengan baik?	0	0	13	87	0
9	Persepsi pemanfaatan (<i>perceived usefulness</i>)	Sistem dapat membantu pengguna dalam kegiatan praktikum?	0	0	0	97	3
10	<i>Satisfaction</i>	Pengguna puas dengan kinerja sistem yang telah dibangun?	0	0	0	100	0

5.3 Analisis

Analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian sistem informasi praktikum yang telah dilakukan. Analisis didapat dari hasil pengujian fungsional dan pengujian unit.

5.3.1 Analisis Pengujian Fungsional

Berdasarkan hasil pengujian *blackbox* (pengujian fungsional) didapatkan hasil bahwa semua fungsi telah berjalan sesuai dengan kebutuhan yang diharapkan. Dari hasil yang diperoleh dari pengujian *blackbox* dapat disimpulkan bahwa implementasi sistem sesuai dengan kebutuhan awal sistem.

5.3.2 Analisis Pengujian *User Acceptance Test*

Pengujian *user acceptance test* dengan pendekatan TAM (*Technology Acceptance Model*) menerapkan sikap dan tiap-tiap perilaku pengguna yang ditinjau dari persepsi pemanfaatan dan kemudahan penggunaan. Pengujian *user acceptance test* juga menilai dari beberapa faktor lain diantaranya *interaction*, *satisfaction*, *performance* dan *information* dari sistem informasi yang telah dibangun.

Dari hasil pengujian yang telah dilakukan didapatkan hasil:

1. Aspek *Interaction*

Ditunjukkan hasil pengujian *User Acceptance Test* untuk aspek *interaction* pada Tabel 5.23 didapatkan hasil untuk pertanyaan no.1 dari kuisisioner yang telah dibuat bahwa 90% responden memilih setuju bahwa sistem informasi praktikum mudah untuk dioperasikan dan 10% lainnya memilih netral. Pada pertanyaan no.2 yang ditunjukkan oleh Tabel 5.23 didapatkan hasil 80% responden memilih setuju bahwa susunan menu dari sistem informasi yang telah dibuat mudah untuk dipahami dan 20% lainnya memilih netral. Pada pertanyaan no.3 yang ditunjukkan oleh Tabel 5.23 didapatkan hasil 87% responden memilih setuju bahwa sistem informasi praktikum memiliki tampilan yang menarik dan 13% responden lainnya memilih netral.

Dari hasil yang didapatkan untuk aspek *interaction* pada pertanyaan no.1, 2 dan 3 bahwa lebih dari 80% jawaban responden menyatakan setuju sehingga dapat ditarik kesimpulan bahwa sistem informasi praktikum yang telah dibangun telah memenuhi standar kebutuhan pengguna dari segi *interaction*. Peninjauan lebih lanjut untuk

jawaban responden yang memilih *option* netral bahwa dimungkinkan sistem informasi praktikum yang telah dibangun kurang memenuhi standar penerimaan sebagian kecil pengguna dari sisi *interaction*, namun juga tidak sepenuhnya pengguna kurang setuju sehingga mereka lebih memilih netral.

2. Aspek *Performance*

Data yang diperoleh dari hasil kuisioner pada Tabel 5.23 untuk aspek *performance* pada pertanyaan no.4 menunjukkan 83% responden memilih setuju bahwa sistem informasi praktikum memiliki navigasi yang cepat dan 17% responden lainnya memilih netral. Dari data yang ditunjukkan pada Tabel 5.23 bahwa lebih dari 80% responden memilih setuju sehingga dapat disimpulkan bahwa sistem telah memenuhi standar kebutuhan pengguna ditinjau dari aspek *performance*. Peninjauan lebih lanjut untuk jawaban 17% responden yang memilih *option* netral bahwa responden menginginkan navigasi yang dapat berjalan lebih cepat namun responden juga tidak menyangkal bahwa navigasi sistem telah berjalan cukup cepat sehingga mereka memilih *option* netral.

3. Aspek Persepsi Kemudahan Penggunaan (*Perceive Ease of Use*)

Data yang ditunjukkan pada Tabel 5.23 untuk aspek persepsi kemudahan penggunaan (*perceive ease of use*) pada pertanyaan no. menunjukkan 80% responden menyatakan setuju bahwa sistem menyediakan akses cepat dan mudah untuk mencari informasi dan 20% lainnya memilih netral. Dari data yang diperoleh atas jawaban responden bahwa 80% memilih setuju sehingga dapat disimpulkan bahwa sistem informasi yang dibangun telah memenuhi standar kebutuhan pengguna dari aspek kemudahan penggunaan. Peninjauan lebih lanjut analisa 20% responden yang memilih *option* netral bahwa ada perbedaan penerimaan relatif antara pengguna terhadap sistem. Sebagian pengguna kurang merasa tercukupi dari segi kecepatan dan kemudahan sistem dalam mencari informasi yang dibutuhkan namun mereka juga tidak menyangkal

bahwa sistem telah memenuhi syarat kebutuhan tersebut sehingga mereka memilih *option* netral.

4. Aspek *Information*

Data yang ditunjukkan pada Tabel 5.23 untuk aspek *information* pertanyaan no.6 90% responden menyatakan setuju bahwa sistem menyediakan informasi yang relevan dan 10% lainnya memilih netral. Data yang ditunjukkan oleh Tabel 5.23 untuk pertanyaan no.10 menunjukkan bahwa 100% responden menyatakan setuju bahwa sistem menyediakan konten yang mudah dibaca. Dari data yang diperoleh atas jawaban responden bahwa lebih dari 80% memilih setuju sehingga dapat disimpulkan bahwa sistem informasi yang dibangun telah memenuhi standar kebutuhan pengguna ditinjau dari aspek *information*.

5. Aspek Persepsi Pemanfaatan (*Perceive Usefulness*)

Data yang ditunjukkan pada Tabel 5.23 untuk aspek *information* pertanyaan no.6 menunjukkan 90% responden menyatakan setuju bahwa sistem menyediakan informasi yang relevan dan 10% lainnya memilih netral. Data yang ditunjukkan oleh Tabel 5.23 untuk pertanyaan no.10 menunjukkan bahwa 100% responden menyatakan setuju bahwa sistem menyediakan konten yang mudah dibaca. Dari data yang diperoleh atas jawaban responden bahwa lebih dari 80% memilih setuju sehingga dapat disimpulkan bahwa sistem informasi yang dibangun telah memenuhi standar kebutuhan pengguna ditinjau dari aspek *information*.

6. Aspek *Satisfaction*

Data yang ditunjukkan pada Tabel 5.23 untuk aspek *satisfaction* pertanyaan no.10 menunjukkan 100% responden menyatakan setuju bahwa mereka puas terhadap kinerja sistem yang telah dibangun. Dari data yang diperoleh atas jawaban responden bahwa 100% memilih setuju sehingga dapat disimpulkan bahwa sistem informasi yang dibangun telah memenuhi standar kebutuhan pengguna ditinjau dari aspek *satisfaction*.

BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut:

1. Aplikasi sistem informasi praktikum berbasis *web* berhasil diimplementasikan menggunakan bahasa pemrograman PHP dengan *framework Code Igniter* dengan metode pengembangan perangkat lunak menggunakan model *waterfall*. Aplikasi sistem informasi praktikum menggunakan desain *interface* dari *bootstrap* untuk memudahkan *user* dalam memahami dan mengoperasikan sistem.
2. Pengujian dilakukan dengan dua cara, yaitu pengujian *blackbox* (fungsional) dan pengujian UAT (*User Acceptance Test*) dengan pendekatan TAM. Pengujian fungsional sistem perangkat lunak sistem informasi praktikum di PTIIK menunjukkan bahwa aplikasi dapat berjalan sesuai rancangan yang telah dibuat. Pada pengujian UAT (*User Acceptance Test*) dengan pendekatan TAM didapatkan hasil bahwa sistem informasi praktikum telah memenuhi kebutuhan pengguna ditinjau dari faktor persepsi pemanfaatan (*perceive usefulness*), persepsi kemudahan (*perceive ease of use*), faktor *performance*, faktor *information*, faktor *satisfaction* dan faktor *interaction*.
3. Sistem informasi praktikum yang dibangun dapat membantu pihak-pihak yang terlibat kegiatan praktikum dalam melaksanakan perannya. Secara garis besar sistem memberikan kemudahan bagi asisten dosen, koordinator asisten dosen, praktikan, dan dosen serta laboran dalam mengakses informasi - informasi praktikum, mengelola serta mendistribusikan data-data dan file praktikum.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan perangkat lunak ini antara lain :

1. Sistem dapat diintegrasikan dengan SIAM, sehingga *administrator* tidak perlu untuk memasukkan data mahasiswa dan data mata kuliah yang ditempuh mahasiswa pada setiap awal semester.
2. Membenahi fitur penjadwalan dengan menggunakan algoritma penjadwalan.
3. Untuk pengembangan lebih lanjut sebaiknya ditambahkan fitur *bulk download* berkas pada subsistem pengunduhan tugas praktikum mahasiswa, sehingga tidak perlu lagi melakukan proses unduh secara satu per satu.
4. Membenahi proses alur pelaksanaan praktikum untuk studi kasus perijinan permohonan pelaksanaan praktikum yang dilakukan administrator dengan menambahkan aktor Ketua Laboraturim.



DAFTAR PUSTAKA

- [ABD-10] Abidin, Jenal. 2010, “Aplikasi Penggajian Karyawan dan Manajemen Transaksi SPA Menggunakan Code Igniter Framework”, Politeknik Telkom Bandung.
- [ANN-10] Anonim. 2010, “Modul Praktikum Pemrograman Internet Teknik Informatika”, Malang:Universitas Brawijaya
- [DBL-03] Bell, Donald.2003, “*An Introduction to the Unified Modelling Language*”, *Rational Software*.
- [GJM-04] J.Myers, Glendford. 2004, “*The Art of Software Testing, Second Edition*”, John Wiley & Sons,Inc.,Hoboken,New Jersey
- [JLW-04] Whitten, J.L., Lonnie D.B., dan Kevin C.D. 2004, “*Metode Desain dan Analisis Sistem*”, Edisi 6, Mcgraw Hill Education.
- [ISM-11] Sommerville, Ian. 2011, *Software Engineering* (9th ed.). United States of America: Addison-Wesley Pearson Education Inc.
- [KEN-07] Laudon, Kenneth C.; Laudon, Jane P. (2007:42), “Sistem Informasi Manajemen”, Palgrave, Basingstoke.
- [RPE-10] Pressman, R. 2010, *Software Engineering : A Practitioner’s Approach, Fifth Edition*. McGraw Hill.
- [BTS-12] Sidik, Betha.2012. “Framework CodeIgniter” , Penerbit : Informatika Bandung. Hal 1-6.
- [WPR-08] Prianto, W. 2008, “Perancangan Arsitektur Enterprise untuk pengembangan E-government pada Pemerintah Kabupaten Barito Utara Studi Kasus: Pelayanan Perizinan”, *Thesis*, Institut Pertanian Bogor, Bogor.
- [WJG-10] Gilmore, W.Jason. 2010, “*Beginning PHP and MySQL*”, Edisi 4, Apress.
- [MER-11] Ehmer, Mohd.2011.”*Different Approaches to Whitebox Testing Technique for Finding Errors*”, Department of Information Technology Al Musanna College of Technology, Sultanate of Oman.