

**SISTEM INFORMASI KEGIATAN PRAKTIKUM  
DI LABORATORIUM KOMPUTER TEKNIK INFORMATIKA  
UNIVERSITAS SURYAKANCANA CIANJUR**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat Kelulusan Program Sarjana Strata- 1  
(S1) Program Studi Informatika Fakultas Teknik  
Universitas Suryakancana Cianjur**

**OLEH  
YADI PRASSETYA  
5520115044**



**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS SURYAKANCANA  
CIANJUR  
2020**

## **LEMBAR PENGESAHAN**

## **LAPORAN TUGAS AKHIR**

Nama : Yadi Prassetya  
NPM : 5520115044  
Fakultas : Teknik  
Program Studi : Informatika  
Judul : Sistem Informasi Kegiatan Praktikum di Laboratorium Komputer Teknik Informatika Universitas Suryakancana Cianjur

Telah disidangkan dan dinyatakan Lulus Sidang Tugas Akhir pada Program Sarjana Strata-1 (S1), Program Studi Informatika Fakultas Teknik Universitas Suryakancana Cianjur pada tanggal 04 Agustus 2020.

Tim Penguji

Nama Dosen Sidang Tugas Akhir

Tanda Tangan

1. Finsa Nurpandi, S.T., M.T
2. M. Kany Legiawan, S.T., M.Kom.
3. Lalan Jaelani, S.T., M.Kom.

Mengetahui :

Ketua Program Studi  
Teknik Informatika

Tarmin Abdulghani, ST., MT.

NIDN. 0401087501

## **LEMBAR PERNYATAAN KEASLIAN**

### **LAPORAN TUGAS AKHIR**

Nama : Yadi Prassetya  
NPM : 5520115044  
Fakultas : Teknik  
Prodi : Informatika  
Judul : Sistem Informasi Kegiatan Praktikum di Laboratorium Komputer Teknik Informatika Universitas Suryakancana Cianjur

Dengan ini saya menyatakan Demi Allah Yang Maha Melihat dan Maha Mengetahui bahwa dalam penelitian ini tidak terdapat karya yang pernah diajukan untuk memperoleh **kelulusan gelar kesarjanaan** di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Dan Apabila saya berdusta dan berbuat curang maka saya siap mendapatkan sanksi dengan **dibatalkannya** hasil karya dan semua yang berhubungan dengan karya ini, serta memohon maaf secara lisan dan tertulis kepada pemilik asli dan lembaga Program Studi Informatika Fakultas Teknik Universitas Suryakancana Cianjur.

Cianjur, 24 Juli 2020

Yadi Prassetya  
5520115044

**SISTEM INFORMASI KEGIATAN PRAKTIKUM DI  
LABORATORIUM KOMPUTER TEKNIK INFORMATIKA  
UNIVERSITAS SURYAKANCANA CIANJUR**

**Yadi Prasetya**

**5520115044**

**Teknik Informatika**

**ABSTRAK**

Fakultas Teknik khususnya program studi Teknik Informatika Universitas Suryakancana memiliki tiga laboratorium komputer (Lab Dasar, Lab Multimedia, dan Lab Jaringan Komputer). Semua laboratorium komputer ini di kelola oleh Koordinator Laboratorium dan Asisten Laboratorium, bagi mahasiswa/i yang ingin mengetahui informasi seperti nilai dan absensi praktikum mahasiswa/i harus menanyakan langsung ke admin lab atau asisten yang bersangkutan. Penulis berinisiatif untuk membuat sebuah sistem informasi praktikum agar mahasiswa/i lebih mudah dalam mengakses informasi seperti absensi dan nilai praktikum.

Pembuatan sistem informasi ini menggunakan bahasa pemrograman PHP dengan *framework* Codeigniter. Sistem informasi ini merupakan aplikasi berbasis web.

Metode penelitian yang digunakan yaitu *prototype*. Metode Prototype ini dapat membantu pengembang dalam memahami lebih baik apa yang akan dikembangkan saat spesifikasi kebutuhan belum jelas, sehingga pengembangan aplikasi menjadi lebih mudah karena pemakai atau pelanggan mengetahui apa yang diharapkannya. Adapun alat bantu berupa UML( *Unified Modeling Language* ).

Hasil dari perancangan pada penelitian ini diimplementasikan menjadi sebuah aplikasi berbasis web yang dapat diakses oleh mahasiswa/i dan pihak laboratorium. Dengan menggunakan aplikasi ini mahasiswa/i bisa lebih mudah untuk mengetahui informasi seputar kegiatan praktikum.

**Kata Kunci :** Praktikum, Laboratorium, Codeigniter, *Prototype*.

**SISTEM INFORMASI KEGIATAN PRAKTIKUM DI  
LABORATORIUM KOMPUTER TEKNIK INFORMATIKA  
UNIVERSITAS SURYAKANCANA CIANJUR**

**Yadi Prasetya**

**5520115044**

**Teknik Informatika**

**ABSTRACT**

*The Faculty of Engineering in particular the Informatics Engineering study program at Suryakancana University has three computer laboratories (Basic Lab, Multimedia Lab, and Computer Network Lab). All of these computer laboratories are managed by the Laboratory Coordinator and Laboratory Assistant, for students who want to know information such as grades and student practicum attendance, they must ask directly to the lab admin or assistant concerned. The author took the initiative to create a practicum information system so that students can easily access information such as attendance and practical values.*

*Making this information system using the PHP programming language with the Codeigniter framework. This information system is a web-based application.*

*The research method used is prototype. This prototype method can help developers better understand what will be developed when the specification of needs is not clear, so application development becomes easier because the user or customer knows what he is hoping for. The tools are UML (Unified Modeling Language).*

*The results of the design in this study were implemented into a web-based application that can be accessed by students and laboratories. By using this application students / i can more easily find out information about practicum activities.*

**Keyword:** *Practice, Laboratory, Codeigniter, Prototype.*

## **KATA PENGANTAR**

Puji Syukur penulis panjatkan ke hadirat Allah SWT, yang telah melimpahkan segala rahmat dan hidayah-Nya kepada penulis, sehingga tersusunlah laporan penelitian tugas akhir ini yang berjudul “Sistem Informasi Kegiatan Praktikum di Laboratorium Komputer Teknik Informatika Universitas Suryakancana Cianjur”.

Tugas Akhir ini merupakan pelengkap salah satu persyaratan yang diajukan dalam rangka menempuh ujian akhir untuk memperoleh gelar Sarjana Teknik (S.T) pada Program Sarjana Starta-1 (S1), Program Studi Informatika Fakultas Teknik Universitas Suryakancana Cianjur.

Penulis sungguh sangat menyadari, bahwa penulisan laporan Tugas Akhir ini tidak akan terwujud tanpa adanya dukungan dan bantuan dari berbagai pihak. Maka, dalam kesematan ini penulis menghaturkan penghargaan dan ucapan terima kasih yang sebesar-besarnya kepada yang terhormat :

1. Yang tercinta kedua Orang Tua saya, yang telah memberikan doa dan dorongan kepada saya baik berupa moril maupun materil dalam tugas akhir ini.
2. Bapak Finsa Nurpandi, ST., M.T., yang meluangkan waktu untuk memberikan petunjuk, arahan, motivasi serta ilmu beliau dalam menyelesaikan Laporan Tugas Akhir ini dan juga selaku Wali Kelas IF-B angkatan 2015..
3. Bapak Tarmin Abdulghani, ST., M.T, selaku Ketua Program Studi Teknik Informatika.
4. Bapak Andri Adikusumah, S.Kom., M.Kom. selaku Koordinator Tugas Akhir.
5. Teman-teman yang memberi semangat kepada penyusun dalam menyusun laporan tugas besar ini.
6. Semua pihak yang tidak dapat penyusun sebutkan satu persatu.

Akhir kata penulis mohon maaf atas kekeliruan dan kesalahan yang terdapat dalam Tugas Akhir ini dan berharap semoga Tugas Akhir ini dapat memberikan manfaat bagi khasanah pengetahuan Teknologi Informasi di Indonesia.

Cianjur, 24 Juli 2020

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN LAPORAN TUGAS AKHIR.....	i
LEMBAR PERNYATAAN KEASLIAN LAPORAN TUGAS AKHIR .....	ii
ABSTRAK .....	iii
ABSTRACT .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	x
DAFTAR TABEL.....	xv
DAFTAR SIMBOL USE CASE DIAGRAM.....	xvii
DAFTAR SIMBOL ACTIVITY DIAGRAM.....	xviii
DAFTAR SIMBOL CLASS DIAGRAM .....	xx
DAFTAR SIMBOL SEQUENCE DIAGRAM.....	xxii

### **BAB I PENDAHULUAN**

1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah .....	3
1.3. Maksud dan Tujuan.....	3
1.4. Batasan Masalah.....	3
1.5. Metode Penelitian.....	4
1.6. Waktu Penelitian .....	7
1.7. Sistematika Penulisan Laporan .....	7

### **BAB II TINJAUAN PUSTAKA**

2.1. Tinjauan Sumber Informasi.....	9
2.1.1. Nama Narasumber.....	9
2.1.2. Tempat dan Tanggal Perolehan Informasi .....	9
2.1.3. Deskripsi Perolehan Informasi .....	9
2.2. Tinjauan Organisasi .....	10
2.2.1. Nama Perusahaan .....	10

2.2.2.	Alamat Perusahaan.....	10
2.2.3.	Profil Organisasi.....	10
2.2.4.	Struktur Organisasi Laboratorium Komputer .....	10
2.2.5.	Deskripsi Kerja.....	11
2.3.	Landasan Teori.....	11
2.3.1.	Definisi Analisis.....	12
2.3.2.	Definisi Perancangan .....	12
2.3.3.	Definisi Informasi .....	12
2.3.4.	Definisi Data .....	12
2.3.5.	Definisi Aplikasi .....	13
2.3.6.	Definisi Sistem.....	13
2.3.7.	Definisi Rekayasa Perangkat Lunak .....	13
2.3.8.	Definisi Internet .....	14
2.3.9.	Definisi <i>Website</i> .....	15
2.4.	Perangkat dan <i>Tools</i> Yang Digunakan.....	16
2.4.1.	Perangkat Pembangun <i>Website</i> .....	16
2.4.2.	Tools yang Digunakan dalam Perancangan Sistem .....	22
2.4.3.	Tools yang Digunakan dalam Pembuatan Sistem.....	28
2.5.	Metode Pengujian.....	33

### **BAB III ANALISIS DAN PERANCANGAN SISTEM**

3.1.	Analisis Sistem.....	34
3.1.1.	Analisis Sistem yang Sedang Berjalan.....	34
3.1.2.	Analisis Kebutuhan Fungsional .....	35
3.1.3.	Analisis Kebutuhan Non-Fungsional .....	36
3.2.	Perancangan Sistem .....	38
3.2.1.	<i>Use Case Diagram</i> .....	38
3.2.2.	<i>Use Case Scenario</i> .....	41
3.2.2.	<i>Swimlane Diagram</i> .....	88
3.2.3.	<i>Class Diagram</i> .....	144
3.2.4.	<i>Sequence Diagram</i> .....	147

3.2.5. <i>Entity Relationship Diagram</i> .....	176
3.3. Perancangan Antarmuka .....	177

## **BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM**

4.1. Implementasi Sistem .....	202
4.1.1 Implementasi Perangkat Keras .....	202
4.1.2 Implementasi Perangkat Lunak .....	202
4.2. Implementasi Antarmuka .....	203
4.3. Pengujian Sistem.....	228

## **BAB V PENUTUP**

5.1. Kesimpulan.....	263
5.1. Saran .....	263
DAFTAR PUSTAKA .....	264
DAFTAR RIWAYAT HIDUP.....	267
DAFTAR LAMPIRAN.....	268

## DAFTAR GAMBAR

Gambar 1. 1 Model Prototype Pressman.....	5
Gambar 2. 1 Struktur Organisasi Laboratorium Teknik Informatika .....	11
Gambar 2. 2 Use Case Diagram.....	23
Gambar 2. 3 Class Diagram .....	24
Gambar 2. 4 State Machine Diagram .....	24
Gambar 2. 5 Activity Diagram.....	25
Gambar 2. 6 Sequence Diagram.....	25
Gambar 2. 7 Component Diagram .....	26
Gambar 2. 8 Deployment Diagram .....	27
Gambar 2. 9 Visual Paradigm .....	28
Gambar 2. 10 XAMPP Control Panel .....	29
Gambar 2. 11 MySQL Server phpMyAdmin.....	30
Gambar 2. 12 Visual Studio Code .....	31
Gambar 3. 1 Use Case Diagram Aplikasi Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika (Server) .....	39
Gambar 3. 2 Use Case Diagram Aplikasi Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika (Client) .....	40
Gambar 3. 3 Swimlane Diagram Login (Server) .....	88
Gambar 3. 4 Swimlane Diagram Edit Profil .....	89
Gambar 3. 5 Swimlane Diagram Ganti Password (Server).....	90
Gambar 3. 6 Swimlane Diagram Mengirim Pesan.....	91
Gambar 3. 7 Swimlane Diagram Ubah Status Pembayaran Praktikum .....	92
Gambar 3. 8 Swimlane Diagram Cek Nilai Praktikum Mahasiswa .....	93
Gambar 3. 9 Swimlane Diagram Cek Kehadiran Asisten .....	93
Gambar 3. 10 Swimlane Diagram Absensi Praktikum .....	94
Gambar 3. 11 Swimlane Diagram Penilaian Praktikum .....	95
Gambar 3. 12 Swimlane Diagram Ubah Status Pengambilan Modul Praktikum	96
Gambar 3. 13 Swimlane Diagram Cek Kehadiran.....	96
Gambar 3. 14 Swimlane Diagram Lihat Kritik & Saran.....	97
Gambar 3. 15 Swimlane Diagram Registrasi Mahasiswa .....	98
Gambar 3. 16 Swimlane Diagram Pindah Praktikum Mahasiswa .....	99
Gambar 3. 17 Swimlane Diagram Mengirim Notifikasi .....	100
Gambar 3. 18 Swimlane Diagram Ubah Status Tiket Pindah Praktikum .....	101
Gambar 3. 19 Swimlane Diagram Balas Pesan Mahasiswa.....	102
Gambar 3. 20 Swimlane Diagram Kelola Periode .....	103

Gambar 3. 21	Swimlane Diagram Ubah Status Kelas .....	104
Gambar 3. 22	Swimlane Diagram Kelola Kelas .....	105
Gambar 3. 23	Swimlane Diagram Ubah Status Kelas .....	106
Gambar 3. 24	Swimlane Diagram Kelola Mata Praktikum .....	107
Gambar 3. 25	Swimlane Diagram Ubah Status Mata Praktikum.....	108
Gambar 3. 26	Swimlane Diagram Kelola Ruangan .....	109
Gambar 3. 27	Swimlane Diagram Kelola Jam.....	110
Gambar 3. 28	Swimlane Diagram Kelola Jabatan .....	111
Gambar 3. 29	Swimlane Diagram Kelola Tugas .....	112
Gambar 3. 30	Swimlane Diagram Kelola Foto Tugas .....	113
Gambar 3. 31	Swimlane Diagram Penilaian Tugas .....	114
Gambar 3. 32	Swimlane Diagram Download & Hapus File Tugas .....	115
Gambar 3. 33	Swimlane Diagram Kelola Info Praktikum.....	116
Gambar 3. 34	Swimlane Diagram Kelola Foto Info Praktikum.....	117
Gambar 3. 35	Swimlane Diagram Tambah Jadwal.....	118
Gambar 3. 36	Swimlane Diagram Edit Jadwal .....	119
Gambar 3. 37	Swimlane Diagram Hapus Jadwal.....	120
Gambar 3. 38	Swimlane Diagram Status Jadwal .....	121
Gambar 3. 39	Swimlane Diagram Tambah Asisten.....	122
Gambar 3. 40	Swimlane Diagram Edit Asisten .....	123
Gambar 3. 41	Swimlane Diagram Hapus Asisten.....	124
Gambar 3. 42	Swimlane Diagram Reset Password Asisten.....	125
Gambar 3. 43	Swimlane Diagram Status Asisten .....	126
Gambar 3. 44	Swimlane Diagram Tambah Mahasiswa.....	127
Gambar 3. 45	Swimlane Diagram Edit Mahasiswa .....	128
Gambar 3. 46	Swimlane Diagram Hapus Mahasiswa.....	129
Gambar 3. 47	Swimlane Diagram Reset Password Mahasiswa.....	130
Gambar 3. 48	Swimlane Diagram Status Asisten .....	131
Gambar 3. 49	Swimlane Diagram Tambah Dosen.....	132
Gambar 3. 50	Swimlane Diagram Edit Dosen .....	133
Gambar 3. 51	Swimlane Diagram Hapus Dosen .....	134
Gambar 3. 52	Swimlane Diagram Reset Password Dosen .....	135
Gambar 3. 53	Swimlane Diagram Status Dosen .....	136
Gambar 3. 54	Swimlane Diagram Login .....	137
Gambar 3. 55	Swimlane Diagram Cek Nilai Tugas.....	138
Gambar 3. 56	Swimlane Diagram Cek Absensi.....	138
Gambar 3. 57	Swimlane Diagram Cek Hasil Praktikum .....	139
Gambar 3. 58	Swimlane Diagram Cek Info Praktikum .....	139
Gambar 3. 59	Swimlane Diagram Ganti Password.....	140
Gambar 3. 60	Swimlane Diagram Mengirim Kritik & Saran .....	141

Gambar 3. 61	Swimlane Diagram Buat Tiket Permintaan Pindah Praktikum ...	142
Gambar 3. 62	Swimlane Diagram Mengirim Pesan.....	143
Gambar 3. 63	Swimlane Diagram Upload Tugas .....	144
Gambar 3. 64	Class Diagram Sistem Informasi Kegiatan Praktikum (Server)..	145
Gambar 3. 65	Class Diagram Sistem Informasi Kegiatan Praktikum (Client....	146
Gambar 3. 66	Sequence Diagram Login (Server) .....	153
Gambar 3. 67	Sequence Diagram Absensi Praktikum .....	154
Gambar 3. 68	Sequence Diagram Penilaian Praktikum .....	154
Gambar 3. 69	Sequence Diagram Kelola Tugas .....	155
Gambar 3. 70	Sequence Diagram Ubah Status Modul Praktikum.....	156
Gambar 3. 71	Sequence Diagram Mengirim Pesan .....	156
Gambar 3. 72	Sequence Diagram Edit Profil.....	157
Gambar 3. 73	Sequence Diagram Ganti Password (Server) .....	157
Gambar 3. 74	Sequence Diagram Kelola Periode.....	158
Gambar 3. 75	Sequence Diagram Kelola Jadwal.....	159
Gambar 3. 76	Sequence Diagram Kelola Kelas.....	160
Gambar 3. 77	Sequence Diagram Kelola Mata Praktikum .....	161
Gambar 3. 78	Sequence Diagram Kelola Ruangan.....	162
Gambar 3. 79	Sequence Diagram Kelola Jam .....	163
Gambar 3. 80	Sequence Diagram Kelola Jabatan .....	164
Gambar 3. 81	Sequence Diagram Kelola Asisten .....	165
Gambar 3. 82	Sequence Diagram Kelola Dosen.....	166
Gambar 3. 83	Sequence Diagram Kelola Mahasiswa .....	167
Gambar 3. 84	Sequence Diagram Kelola Info Praktikum.....	168
Gambar 3. 85	Sequence Diagram Kelola E-Ticketing.....	169
Gambar 3. 86	Sequence Diagram Registrasi Praktikum .....	170
Gambar 3. 87	Sequence Diagram Ubah Status Pembayaran Praktikum.....	170
Gambar 3. 88	Sequence Diagram Pindah Praktikum.....	171
Gambar 3. 89	Sequence Diagram Login (Client).....	172
Gambar 3. 90	Sequence Diagram Ganti Password (Client) .....	172
Gambar 3. 91	Sequence Diagram Cek Nilai .....	173
Gambar 3. 92	Sequence Diagram Cek Absensi .....	173
Gambar 3. 93	Sequence Diagram Upload Tugas .....	174
Gambar 3. 94	Sequence Diagram E-Ticketing .....	175
Gambar 3. 95	Entity Relationship Diagram Sistem Informasi Kegiatan Praktikum Laboratorium Teknik informatika.....	176
Gambar 3. 96	Perancangan Antarmuka Halaman Login Server .....	177
Gambar 3. 97	Perancangan Antarmuka Halaman Dashboard Asisten.....	178
Gambar 3. 98	Perancangan Antarmuka Halaman Absensi Praktikum .....	178
Gambar 3. 99	Perancangan Antarmuka Halaman Penilaian Praktikum .....	179

Gambar 3. 100	Perancangan Antarmuka Halaman Kelola Tugas Praktikum ....	179
Gambar 3. 101	Perancangan Antarmuka Halaman Modul Praktikum.....	180
Gambar 3. 102	Perancangan Antarmuka Halaman Kehadiran Asisten .....	180
Gambar 3. 103	Perancangan Antarmuka Halaman Pesan.....	181
Gambar 3. 104	Perancangan Antarmuka Halaman Edit Profil .....	181
Gambar 3. 105	Perancangan Antarmuka Halaman Ganti Password.....	182
Gambar 3. 106	Perancangan Antarmuka Halaman Dashboard Admin.....	182
Gambar 3. 107	Perancangan Antarmuka Halaman Kelola Periode .....	183
Gambar 3. 108	Perancangan Antarmuka Halaman Kelola Jadwal .....	183
Gambar 3. 109	Perancangan Antarmuka Halaman Pembayaran Praktikum.....	184
Gambar 3. 110	Perancangan Antarmuka Halaman Registrasi Praktikum .....	184
Gambar 3. 111	Perancangan Antarmuka Halaman Pindah Praktikum .....	185
Gambar 3. 112	Perancangan Antarmuka Halaman Kelola Info Praktikum .....	185
Gambar 3. 113	Perancangan Antarmuka Halaman Kelola E - Ticketing .....	186
Gambar 3. 114	Perancangan Antarmuka Halaman Kelola Asisten .....	186
Gambar 3. 115	Perancangan Antarmuka Halaman Kelola Dosen .....	187
Gambar 3. 116	Perancangan Antarmuka Halaman Kelola Mahasiswa .....	187
Gambar 3. 117	Perancangan Antarmuka Halaman Kelola Kelas .....	188
Gambar 3. 118	Perancangan Antarmuka Halaman Kelola Ruangan .....	188
Gambar 3. 119	Perancangan Antarmuka Halaman Kelola Mata Praktikum .....	189
Gambar 3. 120	Perancangan Antarmuka Halaman Kelola Jam.....	189
Gambar 3. 121	Perancangan Antarmuka Halaman Kelola Jabatan .....	190
Gambar 3. 122	Perancangan Antarmuka Halaman Nilai Praktikum (Dosen) ...	190
Gambar 3. 123	Perancangan Antarmuka Halaman Kritik & Saran.....	191
Gambar 3. 124	Perancangan Antarmuka Halaman Login Client.....	192
Gambar 3. 125	Perancangan Antarmuka Halaman Dashboard Client .....	193
Gambar 3. 126	Perancangan Antarmuka Halaman .....	194
Gambar 3. 127	Perancangan Antarmuka Halaman .....	195
Gambar 3. 128	Perancangan Antarmuka Halaman Tugas Praktikum.....	196
Gambar 3. 129	Perancangan Antarmuka Halaman Detail Tugas .....	197
Gambar 3. 130	Perancangan Antarmuka Halaman E - Ticketing.....	198
Gambar 3. 131	Perancangan Antarmuka Halaman Hasil Praktikum .....	199
Gambar 3. 132	Perancangan Antarmuka Halaman .....	200
Gambar 3. 133	Perancangan Antarmuka Halaman Info Praktikum.....	201
Gambar 4. 1	Halaman Login Server .....	203
Gambar 4. 2	Halaman Dashboard Asisten .....	204
Gambar 4. 3	Halaman Absensi Praktikum .....	204
Gambar 4. 4	Halaman Penilaian Praktikum .....	205
Gambar 4. 5	Halaman Kelola Tugas Praktikum .....	205

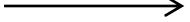
Gambar 4. 6 Halaman Modul Praktikum .....	206
Gambar 4. 7 Halaman Kehadiran Asisten.....	206
Gambar 4. 8 Halaman Pesan .....	207
Gambar 4. 9 Halaman Edit Profil.....	207
Gambar 4. 10 Halaman Ganti Password .....	208
Gambar 4. 11 Halaman Dashboard Admin .....	208
Gambar 4. 12 Halaman Kelola Periode.....	209
Gambar 4. 13 Halaman Kelola Jadwal.....	209
Gambar 4. 14 Halaman Pembayaran Praktikum .....	210
Gambar 4. 15 Halaman Registrasi Praktikum.....	210
Gambar 4. 16 Halaman Pindah Praktikum.....	211
Gambar 4. 17 Halaman Kelola Info Praktikum .....	211
Gambar 4. 18 Halaman Kelola E-Ticketing.....	212
Gambar 4. 19 Halaman Kelola Asisten.....	212
Gambar 4. 20 Halaman Kelola Dosen.....	213
Gambar 4. 21 Halaman Kelola Mahasiswa.....	213
Gambar 4. 22 Halaman Kelola Kelas.....	214
Gambar 4. 23 Halaman Kelola Ruangan.....	214
Gambar 4. 24 Halaman Kelola Mata Praktikum.....	215
Gambar 4. 25 Halaman Kelola Jam .....	215
Gambar 4. 26 Halaman Kelola Jabatan.....	216
Gambar 4. 27 Halaman Nilai Praktikum (Dosen) .....	216
Gambar 4. 28 Halaman Kritik & Saran.....	217
Gambar 4. 29 Halaman Login Client .....	218
Gambar 4. 30 Halaman Dashboard Client .....	219
Gambar 4. 31 Halaman Nilai Praktikum (Client) .....	220
Gambar 4. 32 Halaman Absensi Praktikum (Client) .....	221
Gambar 4. 33 Halaman Tugas Praktikum .....	222
Gambar 4. 34 Halaman Detail Tugas .....	223
Gambar 4. 35 Halaman E-Ticketing .....	224
Gambar 4. 36 Halaman Hasil Praktikum .....	225
Gambar 4. 37 Halaman Ganti Password (Client).....	226
Gambar 4. 38 Halaman Info Praktikum .....	227

## **DAFTAR TABEL**

Tabel 1. 1 Waktu Penelitian Tugas Akhir .....	7
Tabel 3. 1 Kebutuhan Perangkat Keras (Komputer).....	37
Tabel 3. 2 Kebutuhan Perangkat Keras (Smartphone).....	37
Tabel 3. 3 Use Case Scenario Login (Server) .....	41
Tabel 3. 4 Use Case Scenario Ganti Password (Server) .....	41
Tabel 3. 5 Use Case Scenario Edit Profil.....	42
Tabel 3. 6 Use Case Scenario Perpesanan .....	43
Tabel 3. 7 Use Case Scenario Ubah Status Pembayaran .....	44
Tabel 3. 8 Use Case Scenario Cek Nilai Praktikum Mahasiswa.....	44
Tabel 3. 9 Use Case Scenario Cek Kehadiran Asisten.....	45
Tabel 3. 10 Use Case Scenario Absensi Praktikum .....	45
Tabel 3. 11 Use Case Scenario Penilaian Praktikum .....	46
Tabel 3. 12 Use Case Scenario Kelola Tugas .....	47
Tabel 3. 13 Use Case Scenario Ubah Status Pengambilan modul .....	49
Tabel 3. 14 Use Case Scenario Cek Kehadiran.....	50
Tabel 3. 15 Use Case Scenario Lihat Kritik & Saran .....	51
Tabel 3. 16 Use Case Scenario Kelola Periode .....	51
Tabel 3. 17 Use Case Scenario Kelola Jadwal.....	54
Tabel 3. 18 Use Case Scenario Registrasi Mahasiswa.....	56
Tabel 3. 19 Use Case Scenario Pindah Praktikum Mahasiswa.....	57
Tabel 3. 20 Use Case Scenario Kelola Info Praktikum.....	58
Tabel 3. 21 Use Case Scenario Mengirim Notifikasi.....	60
Tabel 3. 22 Use Case Scenario Kelola Mahasiswa .....	63
Tabel 3. 23 Use Case Scenario Kelola Dosen.....	65
Tabel 3. 24 Use Case Scenario Kelola Kelas.....	68
Tabel 3. 25 Use Case Scenario Kelola Ruangan.....	70
Tabel 3. 26 Use Case Scenario Kelola Mata Praktikum .....	72
Tabel 3. 27 Use Case Scenario Kelola Jam.....	74
Tabel 3. 28 Use Case Scenario Kelola Jabatan .....	76
Tabel 3. 29 Use Case Scenario Ubah Status Tiket Pindah Praktikum .....	78
Tabel 3. 30 Use Case Scenario Balas Pesan Mahasiswa .....	78
Tabel 3. 31 Use Case Scenario Login (Client).....	79
Tabel 3. 32 Use Case Cek Nilai Tugas .....	80
Tabel 3. 33 Use Case Cek Absensi .....	81
Tabel 3. 34 Use Case Cek Tugas .....	81
Tabel 3. 35 Use Case Upload Tugas .....	82

Tabel 3. 36 Use Case Cek Hasil Praktikum .....	83
Tabel 3. 37 Use Case Membuat Tiket Pindah Praktikum .....	84
Tabel 3. 38 Use Case Mengirim Kritik & Saran.....	85
Tabel 3. 39 Use Case Mengirim Pesan .....	85
Tabel 3. 40 Use Case Scenario Ganti Password (Client) .....	86
Tabel 3. 41 Use Case Cek Info Praktikum.....	87
Tabel 4. 1 Hasil Pengujian Form Login (Server).....	228
Tabel 4. 2 Hasil Pengujian Absensi Praktikum.....	229
Tabel 4. 3 Hasil Pengujian Penilaian Praktikum.....	230
Tabel 4. 4 Hasil Pengujian Fungsi Kelola Tugas Praktikum .....	231
Tabel 4. 5 Hasil Pengujian Ubah Status Pengambilan Modul .....	233
Tabel 4. 6 Hasil Pengujian Perpesanan .....	233
Tabel 4. 7 Hasil Pengujian Edit Profil .....	234
Tabel 4. 8 Hasil Pengujian Form Ganti Password (Server) .....	234
Tabel 4. 9 Hasil Pengujian Fungsi Kelola Periode .....	235
Tabel 4. 10 Hasil Pengujian Fungsi Kelola Jadwal.....	237
Tabel 4. 11 Hasil Pengujian Ubah Status Pembayaran Praktikum .....	239
Tabel 4. 12 Hasil Pengujian Registrasi Praktikum .....	240
Tabel 4. 13 Hasil Pengujian Pindah Praktikum .....	240
Tabel 4. 14 Hasil Pengujian Fungsi Kelola Info Praktikum .....	240
Tabel 4. 15 Hasil Pengujian Fungsi Kelola Asisten.....	242
Tabel 4. 16 Hasil Pengujian Fungsi Kelola Dosen .....	245
Tabel 4. 17 Hasil Pengujian Fungsi Kelola Mahasiswa.....	247
Tabel 4. 18 Hasil Pengujian Fungsi Kelola Kelas.....	250
Tabel 4. 19 Hasil Pengujian Fungsi Kelola Mata Praktikum.....	252
Tabel 4. 20 Hasil Pengujian Fungsi Kelola Ruangan .....	254
Tabel 4. 21 Hasil Pengujian Fungsi Kelola Jam .....	256
Tabel 4. 22 Hasil Pengujian Fungsi Kelola Jabatan.....	257
Tabel 4. 23 Hasil Pengujian Fungsi Kelola E-Ticketing.....	259
Tabel 4. 24 Hasil Pengujian Form Login (Client) .....	259
Tabel 4. 25 Hasil Pengujian Form Upload Tugas .....	260
Tabel 4. 26 Hasil Pengujian Fungsi E-Ticketing .....	261
Tabel 4. 27 Hasil Pengujian Form Ganti Password (Client).....	262

## DAFTAR SIMBOL USE CASE DIAGRAM

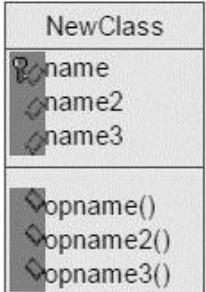
No.	Nama Simbol	Gambar	Keterangan
1	<i>Actor</i>		Menggambarkan pengguna software aplikasi ( <i>user</i> ).
2	<i>Use Case</i>		Menggambarkan fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
3	<i>Association</i>		Menggambarkan hubungan antara <i>actor</i> dengan <i>Use case</i> .
4	<i>Include</i>		Pemanggilan <i>Use case</i> oleh <i>Use case</i> lain atau untuk menggambarkan suatu <i>Use case</i> termasuk didalam <i>Use case</i> lain (diharuskan) relasi ini dapat digunakan untuk menghindari penulisan deskripsi yang berulang ulang.
5	<i>Note</i>		Menggambarkan dokumentasi dari <i>Use Case</i>

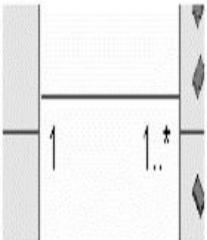
## DAFTAR SIMBOL ACTIVITY DIAGRAM

No	Nama Simbol	Gambar	Keterangan
1.	<i>Initial Node</i>		Menggambarkan awal dari rangkaian tindakan atau aktivitas
2	<i>Activity Final Node</i>		Menghentikan semua aliran kontrol dan aliran objek dalam suatu aktivitas (atau tindakan)
3.	<i>Action</i>		Menggambarkan aktivitas yang harus dilakukan pada sistem
4.	<i>Swimlane and Partition</i>		Cara untuk mengelompokkan aktivitas yang dilakukan oleh aktor yang sama pada diagram aktivitas atau mengelompokkan aktivitas dalam satu rangkaian
5.	<i>Object Flow</i>		Memperlihatkan aliran suatu objek dari satu aktivitas (atau aksi) ke aktivitas (atau aksi) lain

No	Nama Simbol	Gambar	Keterangan
6.	<i>Decision Node</i>		Mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur
7.	<i>Merge Node</i>		Menggambarkan penyatuan kembali jalur keputusan berbeda yang dibuat menggunakan <i>decision node</i>
8.	<i>Fork Node</i>		Membagi perilaku menjadi sekumpulan arus aktivitas (atau tindakan) yang paralel atau bersamaan
9.	<i>Join Node</i>		Menggambarkan penyatuan kembali serangkaian arus aktivitas (atau tindakan) yang paralel atau bersamaan

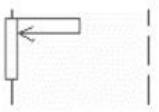
## DAFTAR SIMBOL CLASS DIAGRAM

No.	Nama Simbol	Gambar	Keterangan
1	<i>Class Diagram</i>		<ul style="list-style-type: none"> <li>• <i>Class</i> : Menggambarkan keadaan (atribut/properti) dari suatu objek. <i>Class</i> memiliki tiga area pokok, yaitu: nama, atribut, method.</li> <li>• Nama menggambarkan nama dari <i>class/objek</i>.</li> <li>• Atribut menggambarkan batasan dari nilai yang dapat dimiliki oleh properti tersebut.</li> <li>• Method menggambarkan implementasi dari layanan yang dapat diminta dari beberapa object dari <i>class</i>, yang mempengaruhi <i>behaviour</i>.</li> </ul>
2	<i>Association</i>		Menggambarkan mekanisme komunikasi suatu objek dengan objek lainnya. Atau dapat juga menggambarkan ketergantungan antarkelas.
3	<i>Aggregate</i>		Menggambarkan bahwa suatu objek secara fisik dibentuk dari objek-objek lain, atau secara logis mengandung objek lain.

No.	Nama Simbol	Gambar	Keterangan
4	<i>Multiplicity</i>		<p>Menggambarkan banyaknya object yang terhubung satu dengan yang lainnya. Contoh :</p> <ul style="list-style-type: none"> <li>1      Tepat satu</li> <li>0..*    Nol atau lebih</li> <li>1..*    Satu atau lebih</li> <li>0..1    Nol atau satu</li> <li>5..8    range 5 s.d. 8</li> <li>4..6,9   range 4 s.d. 6 dan 9</li> </ul>

## DAFTAR SIMBOL SEQUENCE DIAGRAM

No	Nama Simbol	Gambar	Keterangan
1	Aktor		Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
2	<i>Boundary</i>		Menggambarkan interaksi antara satu atau lebih <i>actor</i> dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.
3	<i>Control</i>		Menggambarkan “perilaku mengatur”, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.
4	<i>Entity</i>		Menggambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem).
5	<i>Object Message</i>		Menggambarkan pesan/hubungan antar objek, yang menunjukkan urutan kejadian yang terjadi.

No	Nama Simbol	Gambar	Keterangan
6	<i>Message to Self</i>		Menggambarkan pesan/hubungan objek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.
7	<i>Object</i>		Menggambarkan abstraksi dari sebuah entitas nyata/tidak nyata yang informasinya harus disimpan.

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Perkembangan teknologi didunia Informasi dari waktu ke waktu berkembang sangat cepat. Hal ini didukung dengan adanya teknologi baru dari hari ke hari yang semakin maju, sehingga informasi dapat diakses secara tepat, cepat dan akurat. Contoh teknologi informasi yang saat ini sedang berkembang pesat adalah *Website* dan aplikasi *Mobile*. Menurut Pressman dan Bruce (2014:9). *Website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hypertext*), baik diantara *page* yang disimpan dalam server yang sama maupun server diseluruh dunia. *Pages* diakses dan dibaca melalui *browser* seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome dan aplikasi *browser* lainnya (Hakim Lukmanul, 2004).

Fakultas Teknik khususnya program studi Teknik Informatika Universitas Suryakancana memiliki tiga laboratorium komputer (Lab Dasar, Lab Multimedia, dan Lab Jaringan Komputer). Semua laboratorium komputer ini di kelola oleh Koordinator Laboratorium dan Asisten Laboratorium. Saat ini Laboratorium Teknik Informatika (labtif) telah memiliki sistem informasi yang digunakan untuk mengelola kegiatan-kegiatan praktikum seperti registrasi mahasiswa, absensi, dan penjadwalan praktikum. Tetapi dalam sistem tersebut masih ada kekurangan, yaitu tidak adanya fungsi pengelolaan tugas praktikum, pengelolaan info praktikum, dan tidak adanya fungsi untuk menampung kritik dan saran dari mahasiswa. Untuk penilaian tugas-tugas praktikum saat ini masih dilakukan secara manual yaitu dengan mencatat nilai pada sebuah *form* kertas penilaian tugas dimana hal ini sangat tidak baik karena sewaktu-waktu kertas tersebut bisa hilang ataupun rusak. Untuk pemberian informasi atau info praktikum pihak labtif biasanya memberikan informasinya dengan cara mengunggah ke sosial media facebook dan instagram dimana hal itu tidak efektif karena tidak semua mahasiswa aktif di sosial media

tersebut. Pada sistem aplikasi yang berjalan saat ini dalam fungsi absensi praktikum masih ada *bug* atau kesalahan logika, yaitu jika keluar dari halaman absensi kemudian kembali lagi ke halaman absensi maka absensi akan langsung loncat ke pertemuan berikutnya. Hal ini dapat mengganggu kegiatan praktikum jika sewaktu-waktu asisten tidak sengaja keluar dari halaman absensi dan belum selesai melakukan absensi seluruh mahasiswa praktikum. Sistem saat ini juga hanya berjalan di pihak Laboratorium saja, artinya tidak ada sistem yang berjalan disisi mahasiswa. Untuk saat ini mahasiswa tidak bisa mengetahui secara langsung data-data praktikum seperti absensi, nilai tugas, dan nilai ujian praktikum. Serta tidak adanya fungsi mengirim kritik dan saran untuk mahasiswa informatika terhadap pihak laboratorium.

Dalam mengatasi permasalahan diatas diperlukan adanya pengembangan aplikasi pada sistem yang berjalan saat ini. Untuk mahasiswa diperlukan pengembangan aplikasi berbasis *web* sebagai sarana media informasi untuk mahasiswa yang ingin melihat data hasil kegiatan praktikum miliknya seperti data absensi dan nilai-nilai praktikum. Selain itu juga diperlukan fungsi untuk mengirim kritik dan saran, serta pengumpulan tugas praktikum. Dengan aplikasi ini mahasiswa dapat dengan mudah mengetahui informasi seputar kegiatan praktikum. Sementara itu untuk pihak laboratorium diperlukan fungsi tambahan terhadap sistem yang ada saat ini yaitu pengelolaan tugas praktikum, pengelolaan info praktikum, dan menyediakan fungsi untuk menampung kritik dan saran dari mahasiswa.

Maka dari itu penulis akan memberikan judul untuk tugas akhir ini dengan judul “**SISTEM INFORMASI KEGIATAN PRAKTIKUM DI LABORATORIUM KOMPUTER TEKNIK INFORMATIKA UNIVERSITAS SURYAKANCANA**”.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang di atas dapat diambil beberapa rumusan masalah yang harus dapat diselesaikan dengan adanya penelitian ini. Bagaimana membuat sebuah aplikasi sebagai media informasi untuk mahasiswa/i praktikum teknik informatika sehingga dapat dengan mudah mengetahui informasi seputar kegiatan praktikum dan pembuatan aplikasi disisi asisten untuk mengelola data praktikum.

## **1.3. Maksud dan Tujuan**

Maksud dari penelitian ini adalah membangun sebuah aplikasi sistem informasi berbasis *web* untuk mahasiswa teknik informatika sebagai sarana media informasi kegiatan praktikum laboratorium komputer teknik informatika.

Adapun tujuan penelitian ini adalah sebagai berikut :

- a. Memudahkan mahasiswa/i untuk mengakses informasi praktikum.
- b. Memudahkan mahasiswa/i untuk mengirimkan tugas praktikum.
- c. Memudahkan pihak Laboratorium untuk mengelola dan memberikan informasi kegiatan praktikum.
- d. Memudahkan pihak Laboratorium untuk mengelola data kegiatan praktikum.

## **1.4. Batasan Masalah**

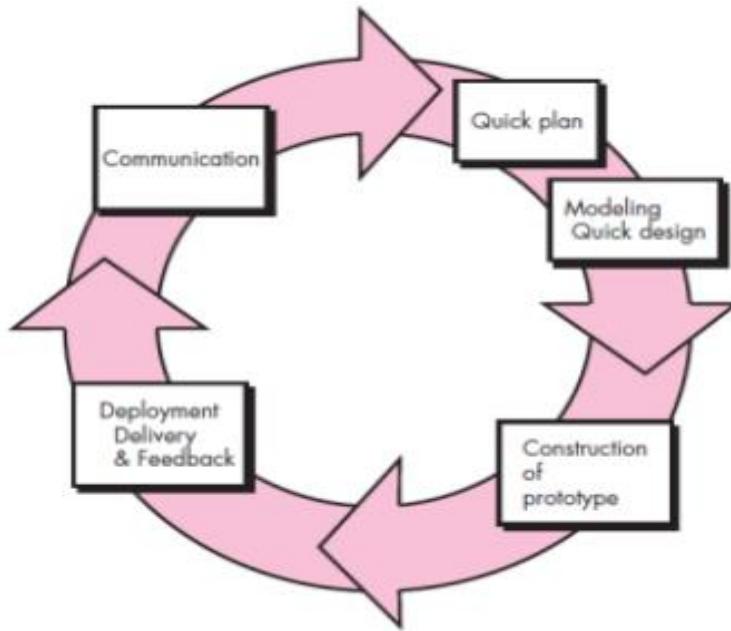
Adapun batasan – batasan masalah yang membatasi ruang lingkup dari penelitian ini adalah:

1. Membahas pembuatan aplikasi sistem informasi kegiatan praktikum laboratorium teknik informatika *server* berbasis *web*.
2. Membahas pembuatan aplikasi sistem informasi kegiatan praktikum laboratorium teknik informatika *client* berbasis *web*.
3. Kegiatan praktikum yang dibahas diantaranya : absensi praktikum, penilaian praktikum, pengelolaan tugas praktikum, pengelolaan info praktikum, pengelolaan data master.

### **1.5. Metode Penelitian**

Metode penelitian yang digunakan adalah studi kasus. Studi kasus adalah eksplorasi mendalam dari sistem terikat (misalnya, kegiatan, acara, proses, atau individu) berdasarkan pengumpulan data yang luas. Studi kasus melibatkan investigasi kasus, yang dapat didefinisikan sebagai suatu entitas atau objek studi yang dibatasi, atau terpisah untuk penelitian dalam hal waktu, tempat, atau batas-batas fisik. Penting untuk memahami bahwa kasus dapat berupa individu, program, kegiatan, sekolah, ruang kelas, atau kelompok. Setelah kasus didefinisikan dengan jelas, peneliti menyelidiki mereka secara mendalam, biasanya menggunakan beberapa metode pengumpulan data, seperti wawancara, observasi lapangan, dan dokumentasi.

Dalam pengembangan sistem diperlukan suatu metode perancangan yang digunakan untuk menghasilkan sistem yang di harapkan. metode perancangan yang digunakan adalah paradigma prototype model. Model ini mengusulkan sebuah pendekatan kepada pengembangan software yang sistematik dan iteratif yang mulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Metode Prototype ini dapat membantu pengembang dalam memahami lebih baik apa yang akan dikembangkan saat spesifikasi kebutuhan belum jelas, sehingga pengembangan aplikasi menjadi lebih mudah karena pemakai atau pelanggan mengetahui apa yang diharapkannya Paradigma rekayasa perangkat lunak ini diambil dari buku “Rekayasa Perangkat Lunak : Pendekatan Praktisi” tahun 2010 oleh R. S. Pressman.



Gambar 1. 1 Model Prototype Pressman

### 1. *Communication*

Melakukan konsultasi dengan Asisten Laboratorium dan memberikan pertanyaan mengenai bagaimana proses kegiatan praktikum berjalan, mencari informasi seperti bagaimana pihak asisten memberikan tugas dan nilai praktikum kepada mahasiswa/i. Semua data dan informasi yang di dapatkan pada tahap *communication* itu akan menjadi bahan pertimbangan di tahapan penelitian selanjutnya.

### 2. *Quick Plan*

Melakukan perencanaan penelitian yang diambil berdasarkan studi kasus memastikan hal apa yang harus dilakukan untuk dapat menyelesaikan masalah pada studi kasus dan mencari teknologi yang tepat untuk dapat diimplementasikan pada studi kasus tersebut.

### 3. *Modeling Quick Design*

Pada tahapan ini tim pengembang mulai membuat pemodelan dari rencana umum yang telah dibuat, dan berfokus pada representasi semua perangkat lunak yang akan terlihat oleh para pengguna akhir, terutama rancangan antar muka

(*user interface*). *Modelling quick design* pada penelitian ini yaitu merancang sistem informasi kegiatan praktikum.

#### 4. *Construction of Prototype*

Membangun Prototype sistem informasi kegiatan praktikum hasil dari rancangan dan modeling sebelumnya ke sebuah program aplikasi.

#### 5. *Deployment Delivery & Feedback*

Melakukan uji coba prototype aplikasi tersebut dengan melakukan pengujian menggunakan metode *black box*. Aplikasi diserahkan kepada asisten dan mahasiswa untuk diuji coba dan untuk mendengarkan pengguna terkait kekurangan maupun kelebihan dari aplikasi tersebut.

## 1.6. Waktu Penelitian

Waktu yang diperlukan dalam penelitian tugas akhir ini adalah sebagai berikut :

*Tabel 1. 1 Waktu Penelitian Tugas Akhir*

No	Uraian Langkah Penelitian	Bulan				
		Mar 2020	Apr 2019	Mei 2020	Feb 2020	Juli 2020
1	Perumusan Masalah					
2	Pengumpulan Data					
3	Analisis					
4	Desain					
5	Implementasi					
6	Testing					
7	Penyusunan Laporan					
8	Seminar Tugas Akhir					
9	Sidang Tugas Akhir					

## 1.7. Sistematika Penulisan Laporan

Berikut ini merupakan sistematika penulisan laporan penelitian Tugas Akhir yang terdiri dari lima bab, diantaranya :

### **BAB I - Pendahuluan**

Bab ini menjelaskan tentang latar belakang, perumusan masalah, maksud dan tujuan, batasan sistem, waktu pelaksanaan serta metode pengambilan data dan metode penelitian yang diambil dalam pembuatan aplikasi.

## **BAB II – Tinjauan Pustaka**

Bab ini menjelaskan mengenai landasan teori yang digunakan sebagai acuan dalam membuat laporan, definisi perancangan dan definisi-definisi lainnya, serta *tool-tool* yang digunakan untuk merancang aplikasi sistem informasi kegiatan praktikum.

## **BAB III – Analisis dan Perancangan Sistem**

Bab ini menjelaskan tentang analisis sistem yang akan dibangun dan perancangan sistem. Perancangan sistem didalamnya mencakup *Use case diagram*, *Swimlane diagram*, *Class diagram*, *Sequence diagram*, dan perancangan antar muka.

## **BAB IV – Implementasi dan Pengujian Sistem**

Bab ini menjelaskan pengujian dan analisis terhadap hasil pengujian dari aplikasi yang telah dibangun.

## **BAB V - Penutup**

Bab ini menjelaskan kesimpulan dan saran terhadap seluruh kegiatan penelitian Tugas Akhir yang telah dilakukan.

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini membahas mengenai gambaran organisasi tempat survei yang didalamnya mencakup , struktur organisasi, deskripsi kerja serta landasan teori yang dipakai dalam laporan ini.

#### **2.1. Tinjauan Sumber Informasi**

Tinjauan sumber informasi membahas tentang informasi apa saja yang diperoleh dalam penelitian dan langkah apa yang dilakukan untuk mendapatkan informasi tersebut.

##### **2.1.1. Nama Narasumber**

Reynaldy Miftahudin Fauzi (Koordinator Teknis Laboratorium Teknik Informatika), Andi Wijaya Suherman (Asisten Laboratorium Teknik Informatika) dan M. Wildan Miftahulhuda (Admin Sistem Laboratorium Teknik Informatika).

##### **2.1.2. Tempat dan Tanggal Perolehan Informasi**

Proses wawancara dalam penelitian ini dilakukan di Laboratorium Komputer pada tanggal 08 Juni 2019.

##### **2.1.3. Deskripsi Perolehan Informasi**

Perolehan informasi dilakukan dengan cara wawancara langsung dengan para narasumber. Pertanyaan yang diajukan adalah tentang sistem yang sedang berjalan di Laboratorium Komputer. Sistem yang saat ini sedang berjalan sebagian masih didominasi dengan metode manual. Mahasiswa/i harus terlebih dahulu menemui asisten atau admin sistem laboratorium komputer lalu melakukan pengajuan berbagai pelayanan data seperti mahasiswa/i yang ingin mengetahui nilai-nilai praktikum. Sesuai dengan yang diinginkan pihak Laboratorium Teknik Informatika, yaitu adanya sebuah aplikasi yang berjalan di sisi *client* dalam hal ini adalah mahasiswa/i. Maka dari itulah dibuatkannya sebuah aplikasi sistem informasi kegiatan praktikum yang berjalan disisi *client*. Tujuannya memudahkan mahasiswa/i yang ingin melihat data-data praktikum miliknya seperti absensi dan nilai praktikum . Pada sistem aplikasi yang berjalan saat ini dalam fungsi absensi

praktikum masih ada *bug* atau kesalahan logika, yaitu jika keluar dari halaman absensi kemudian kembali lagi ke halaman absensi maka absensi akan langsung loncat ke pertemuan berikutnya. Hal ini dapat mengganggu kegiatan praktikum jika sewaktu-waktu asisten tidak sengaja keluar dari halaman absensi dan belum selesai melakukan absensi seluruh mahasiswa praktikum. Sementara itu pada sistem aplikasi saat ini belum ada fungsi untuk melakukan penilaian tugas praktikum. Penilaian tugas masih dilakukan pada *form* kertas penilaian.

## **2.2. Tinjauan Organisasi**

Tinjauan umum organisasi membahas tentang gambaran organisasi tempat survei yang didalamnya mencakup sejarah singkat organisasi, struktur organisasi dan deskripsi kerja.

### **2.2.1. Nama Perusahaan**

Laboratorium Teknik Informatika, Universitas Suryakancana.

### **2.2.2. Alamat Perusahaan**

Jl. Pasir Gede Raya, Kota Cianjur, Jawa Barat, Indonesia.

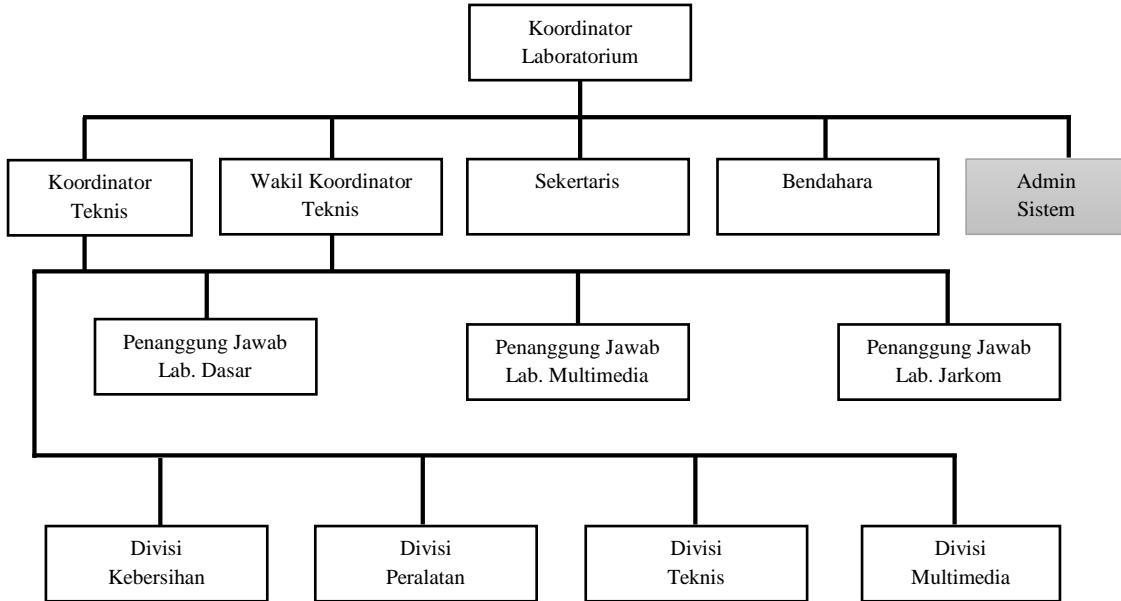
### **2.2.3. Profil Organisasi**

Laboratorium Komputer Teknik Informatika terletak di komplek Fakultas Teknik Universitas Suryakancana Cianjur. Terdiri dari tiga buah ruangan laboratorium yaitu laboratorium dasar dengan kapasitas 19 orang praktikan, laboratorium multimedia dengan kapasitas 15 orang dan laboratorium jaringan komputer dengan kapasitas 16 orang peserta.

Laboratorium ini dipergunakan untuk kegiatan penunjang perkuliahan seperti kegiatan praktikum komputer bagi mahasiswa Fakultas Teknik.

### **2.2.4. Struktur Organisasi Laboratorium Komputer**

Struktur organisasi adalah suatu bagian yang menunjukkan bentuk kerja sama dari berbagai bagian dalam suatu perusahaan atau dengan kata lain struktur organisasi adalah suatu bagan yang menunjukkan tugas-tugas untuk setiap bagian dalam suatu perusahaan untuk mencapai tujuan perusahaan. Berikut ini merupakan gambar struktur organisasi Laboratorium Teknik Informatika :



*Gambar 2. 1 Struktur Organisasi Laboratorium Teknik Informatika*

### 2.2.5. Deskripsi Kerja

- a. Admin Sistem
  - 1. Mengelola data asisten dan mahasiswa
  - 2. Membagi shift praktikum
  - 3. Mengelola data penilaian mahasiswa
  - 4. Mengelola absensi mahasiswa dan asisten
  - 5. Memperbaiki aplikasi yang ada di laboratorium
  - 6. Mengelola data inventaris laboratorium

### 2.3. Landasan Teori

Landasan teori merupakan teori yang relevan yang digunakan untuk menjelaskan *variable* yang akan diteliti dan sebagai dasar untuk memberi jawaban sementara terhadap rumusan masalah yang diajukan (hipotesis), dan penyusunan instrument penelitian. Teori yang digunakan bukan sekedar pendapat dari pengarang atau pendapat lain, tetapi teori yang benar-benar telah diuji kebenarannya.

### **2.3.1. Definisi Analisis**

Dalam Kamus Bahasa Indonesia Kontemporer karangan Peter Salim dan Yenni Salim (2002) menjabarkan pengertian analisis sebagai berikut :

Analisis adalah penyelidikan terhadap suatu peristiwa (perbuatan, karangan dan sebagainya) untuk mendapatkan fakta yang tepat (asal usul, sebab, penyebab sebenarnya, dan sebagainya).

### **2.3.2. Definisi Perancangan**

Perancangan adalah suatu kegiatan yang memiliki tujuan untuk mendesign sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik. (Al-Bahra Bin Ladjamudin, 2005).

Perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem di implementasikan. (Pressman, 2009).

### **2.3.3. Definisi Informasi**

“Informasi merupakan data yang diolah sedemikian rupa, sehingga bisa dijadikan dasar dalam mengambil sebuah keputusan yang tepat dan benar”. (George H Bodnar, 2001).

“Informasi adalah penerangan, keterangan, pemberitahuan, kabar atau berita. Informasi juga merupakan keterangan atau bahan nyata yang dapat dijadikan dasar kajian analisis atau kesimpulan”. (Anton M. Moeliono, 1990).

### **2.3.4. Definisi Data**

Data adalah suatu kejadian yang kita hadapi, data tersebut dapat berupa catatan dalam kertas, buku atau tersimpan sebagai file dalam database. Data akan menjadi bahan dalam suatu proses pengolahan data. Proses pengolahan data terbagi menjadi tiga tahapan, yang disebut dengan siklus pengolahan data yaitu :

a. Tahapan *Input*

Tahapan ini yaitu dilakukan proses pemasukan data ke dalam komputer melalui media *input*.

b. Tahapan *Processing*

Tahapan ini yaitu dilakukan proses pengolahan data yang sudah dimasukan, yang dilakukan oleh alat pemroses yang dapat berupa proses perhitungan, perbandingan, atau pencarian di *storage*.

c. Tahapan *Output*

Tahapan ini yaitu dilakukan proses menghasilkan *output* dari hasil pengolahan data ke alat *output* yaitu berupa informasi.

#### **2.3.5. Definisi Aplikasi**

Aplikasi menurut Kamus Besar Bahasa Indonesia (1998) adalah penerapan dari rancangan sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Suatu kelompok file (form, class, report) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi Payroll, aplikasi Fixed Asset. Ruang lingkup dari suatu aplikasi berbeda dari suatu perusahaan ke perusahaan lain (Santoro, 2005).

#### **2.3.6. Definisi Sistem**

Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan (McLeod et al, 2001). Sedangkan pengertian sistem menurut Sutabri (2005) adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Sistem adalah kumpulan dari komponen atau elemen yang saling berhubungan satu dengan lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu (Jogiyanto, 2005). Simpulan dari tiga definisi tersebut mengenai sistem adalah sekelompok elemen atau unsur yang saling terkait menjadi satu untuk mencapai suatu tujuan tertentu.

#### **2.3.7. Definisi Rekayasa Perangkat Lunak**

Rekayasa perangkat lunak (RPL, atau *Software Engineering*) adalah satu bidang profesi yang mendalamai cara-cara pengembangan perangkat lunak termasuk

pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak, dan sebagainya. Rekayasa perangkat lunak adalah disiplin rekayasa dengan perangkat lunak yang dikembangkan. Biasanya proses melibatkan penemuan pada keinginan klien, menyusunnya di dalam daftar kebutuhan, perancangan, pengkodean, pengujian, dan pengintegrasian bagian yang terpisah, menguji keseluruhan, penyebaran dan pemeliharaan perangkat lunak. (Sinamarta, 2010).

Rekayasa perangkat lunak adalah pengubahan perangkat lunak itu sendiri guna mengembangkan, memelihara, dan membangun kembali dengan menggunakan prinsip reakayasa untuk menghasilkan perangkat lunak yang dapat bekerja lebih efisien dan efektif untuk pengguna.

Kriteria yang dapat digunakan sebagai acuan dalam merekayasa perangkat lunak:

- a. Dapat terus dirawat dan dipelihara (*Maintainability*).
- b. Dapat mengikuti perkembangan teknologi (*Dependability*).
- c. Dapat mengikuti keinginan pengguna (*Robust*).
- d. Efektif dan efisien dalam menggunakan energi dan penggunaannya.

### 2.3.8. Definisi Internet

Internet merupakan dua komputer atau lebih yang saling berhubungan membentuk jaringan komputer hingga meliputi jutaan komputer di dunia (Internasional), yang saling berinteraksi dan bertukar informasi, sedangkan dari segi ilmu pengetahuan, internet merupakan sebuah perpustakaan besar yang di dalamnya terdapat jutaan bahkan milyaran informasi atau data yang dapat berupa teks, grafik, audio, maupun animasi dan lain-lain dalam bentuk media elektronik. Orang dapat “berkunjung” ke perpustakaan tersebut kapan saja dan dimana saja. Dari segi komunikasi, internet adalah sarana yang sangat efisien dan efektif untuk melakukan pertukaran informasi jarak jauh maupun di dalam lingkungan perkantoran.

Awalnya internet dikenal sebagai suatu wadah bagi para peneliti untuk saling bertukar informasi yang kemudian dimanfaatkan oleh perusahaan-

perusahaan komersil sebagai sarana bisnis mereka, saat ini pengguna internet tersebar di seluruh dunia dengan jumlah mencapai lebih dari 250 juta orang.

Sedangkan pengertian internet menurut Allan (2005) adalah sekumpulan jaringan komputer yang saling terhubung secara fisik dan memiliki kemampuan untuk membaca dan menguraikan protokol komunikasi tertentu yang disebut *Internet Protocol* (IP) dan *Transmission Control Protocol* (TCP). Protokol adalah spesifikasi sederhana mengenai bagaimana komputer saling bertukar informasi.

### 2.3.9. Definisi Website

*Website* merupakan fasilitas hiperteks untuk menampilkan data berupa teks, gambar, suara, animasi, dan data multimedia lainnya. Penggolongan *website* berdasarkan isinya (*website contents*) terdiri dari dua jenis yaitu (Wahana, 2006):

a. *Web Statis*

*Web* statis adalah *web* yang berisi/menampilkan informasi-informasi yang sifatnya statis (tetap). Disebut statis karena pengguna tidak dapat berinteraksi dengan *web* tersebut. Singkatnya, untuk mengetahui suatu *web* lain dan berisi suatu informasi yang tetap maka *web* tersebut disebut statis. Pada *web* statis, pengguna hanya dapat melihat isi dokumen pada halaman *web* dan apabila diklik akan berpindah ke halaman *web* yang lain. Interaksi pengguna hanya terbatas dapat melihat informasi yang ditampilkan, tetapi tidak dapat mengolah informasi yang dihasilkan. *Web* statis biasanya berupa *HyperText Markup Language* (HTML) yang ditulis pada editor teks dan disimpan dalam bentuk .html atau .htm. Pengertian HTML adalah bahasa yang menggunakan perintah sederhana dalam standar dokumen untuk menyediakan suatu tampilan *visual* yang terintegrasi. HTML terdiri atas perintah-perintah sederhana yang menjelaskan bagaimana struktur dokumen, tetapi tidak memformatnya. *Browser* yang menampilkan HTML akan memformat dan menyesuaikan tampilan HTML sehingga sesuai dengan layar komputer pengunjung. (Wahana, 2006).

b. *Web Dinamis*

*Web* dinamis adalah *web* yang menampilkan informasi serta dapat berinteraksi dengan pengguna. *Web* yang dinamis memungkinkan pengguna untuk berinteraksi

menggunakan form sehingga dapat mengolah informasi yang ditampilkan. *Web* dinamis bersifat interaktif, tidak kaku, dan terlihat lebih indah. *Web* dinamis biasanya berupa *Page Hypertext Preprocessor* (PHP) yang membuat halaman web HTML menjadi dinamis. HTML yang digabung dengan script PHP akan menghasilkan tampilan web yang dinamis, indah, dan interaktif. (Wahana, 2006).

## 2.4. Perangkat dan *Tools* Yang Digunakan

*Tool – tool* yang digunakan memaparkan perangkat apa saja yang digunakan dalam membangun aplikasi sistem informasi kegiatan praktikum. Berikut adalah perangkat tools yang digunakan untuk pembangunan dan pembuatan sistem :

### 2.4.1. Perangkat Pembangun *Website*

Untuk membangun sebuah aplikasi berbasis *web* diperlukan berbagai perangkat seperti bahasa pemrograman, dan *library*. Berikut ini adalah bahasa pemrograman dan *library* yang digunakan dalam pembuatan aplikasi sistem informasi kegiatan praktikum:

#### 1. PHP (*Hypertext Preprocessor*)

PHP adalah bahasa pemrograman *script server side* yang didesain untuk pengembangan *web*. Dimana PHP ini merupakan singkatan dari *Hypertext Preprocessor* yang digunakan sebagai script untuk memperindah tampilan *website*. Selain itu PHP juga digunakan bersamaan dengan bahasa pemrograman lainnya seperti bahasa pemrograman HTML dan javascript.

Berikut ini merupakan contoh penulisan *syntax* dasar dari bahasa pemrograman PHP :

```
<?php  
$nama = "Yadi";  
$umur = "21";  
  
echo "Hallo " . $nama . "Selamat Siang";  
echo "Usia anda sekarang " . $umur . "tahun";  
?>
```

## 2. HTML (Hypertext Markup Language)

HTML adalah singkatan dari *Hypertext Markup Language* yaitu bahasa *markup* yang digunakan untuk membuat sebuah halaman *web*, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web internet (*Browser*). HTML dapat juga digunakan sebagai link antara *file-file* dalam situs atau dalam komputer dengan menggunakan *localhost*, atau *link* yang menghubungkan antar situs dalam dunia internet.

Berikut ini merupakan contoh penulisan *syntax* dasar dari HTML :

```
<!DOCTYPE html>
<html>
<head>
    <title>Tugas Akhir 2019</title>
</head>
<body>
    <h1>Selamat Datang di TA 2019</h1>
</body>
</html>
```

## 3. CSS (Cassading Style Sheet)

CSS adalah singkatan dari *Cassading Style Sheet* yang digunakan untuk mengatur atau mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. *Cassading Style Sheet* (CSS) bukan termasuk kedalam bahasa pemrograman, namun sama halnya seperti *style* dalam aplikasi pengolahan kata Microsoft Word yang dapat mengatur beberapa *style* seperti *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya.

Berikut ini merupakan contoh penulisan *syntax* dasar dari CSS :

```
*{
    Padding : 0px;
    Margin : 0px;
}
Body{
    Background-img : url('image/foto.jpg');
```

```

        Background-repeat : no-repeat;
        Background-attachment : fixed;
    }
    h1{
        color : white;
        font-size : 30px;
        text-align : center;
    }
}

```

#### **4. Materialize CSS**

Materialize CSS merupakan *framework* CSS yang menyediakan kumpulan komponen-komponen antarmuka dasar pada web yang telah dirancang sedemikian rupa untuk digunakan bersama-sama. Selain komponen antarmuka, Materialize CSS juga menyediakan sarana untuk membangun *layout* halaman dengan mudah dan rapi, serta modifikasi pada tampilan dasar HTML untuk membuat seluruh halaman web yang dikembangkan senada dengan komponen-komponen lainnya. Materialize CSS dibuat dan didesain oleh Google.

#### **5. REST API**

REST API adalah salah satu jenis web service yang menerapkan konsep perpindahan antar state. State disini dapat digambarkan seperti jika browser meminta suatu halaman web, maka server akan mengirimkan state halaman web yang sekarang ke browser. Bernavigasi melalui link-link yang disediakan sama halnya dengan mengganti state dari halaman web. Begitu pula REST bekerja, dengan bernavigasi melalui link-link HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan state satu sama lain. Perintah HTTP yang bisa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE* (Richardson dkk, 2007).

RESTful API / REST API merupakan implementasi dari API (*Application Programming Interface*). REST (*REpresentational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan

aplikasi. Tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan (*scale*) terutama dalam pertukaran dan komunikasi data. RESTful API memiliki komponen penting di dalamnya antara lain:

- a. URL *Design*
  - b. HTTP Verbs
  - c. HTTP *Response Code*
  - d. Format *Response*
- a. URL Design

RESTful API diakses menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti developer. URL API biasa disebut *endpoint* dalam pemanggilannya. Contoh penamaan URL / *endpoint* yang baik adalah seperti berikut :

```
/users
/users/1234
/users/1234/photos
/users/1234/photos/abc
```

- b. HTTP Verbs

Setiap *request* yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang di *request* oleh *client*, diantaranya metode yang umum digunakan adalah sebagai berikut :

### 1. GET

*GET* adalah metode HTTP yang paling sederhana dari metode yang lainnya, metode *GET* ini digunakan untuk membaca atau mendapatkan data dari sumber. Berikut ini merupakan contoh penulisan dari metode *GET* :

GET/users	: Membaca atau mendapatkan daftar user.
GET/users/1234	: Membaca atau mendapatkan data user dengan id 1234.

## 2. *POST*

*POST* adalah metode HTTP *request* yang digunakan untuk membuat data baru dengan menyisipkan data dalam *body* saat *request* dilakukan. Berikut ini merupakan contoh penulisan dari metode *POST* :

POST/users : Membuat data user yang baru.

## 3. *PUT*

*PUT* adalah metode HTTP *request* yang biasanya digunakan untuk melakukan *update data resource*. Berikut ini merupakan contoh penulisan dari metode *PUT* :

PUT/users/1234 : Mengupdate data user dengan id 1234.

## 4. *PATCH*

*PATCH* adalah metode HTTP *request* yang biasanya digunakan untuk melakukan *update data resource*. Berikut ini merupakan contoh penulisan dari metode *PATCH* :

PATCH/users/1234 : Mengupdate data user dengan id 1234.

## 5. *DELETE*

*DELETE* adalah metode HTTP *request* yang digunakan untuk menghapus suatu data pada *resource*. Berikut ini merupakan contoh penulisan dari metode *DELETE* :

DELETE/users/1234 : Menghapus data user dengan id 1234.

### c. HTTP Response Code

HTTP response *code* adalah kode standarisasi dalam menginformasikan hasil *request* kepada *client*. Berikut ini merupakan jenis-jenis *response code* yang ada pada RESTful API yaitu :

#### 1. 200 OK

*Response code* ini menandakan bahwa *request* yang dilakukan berhasil.

#### 2. 201 *Created*

*Response code* ini menandakan bahwa *request* yang dilakukan berhasil dan data telah dibuat. Kode ini digunakan untuk mengkonfirmasi berhasilnya *request PUT* atau *POST*.

3. 400 *Bad Request*

*Response code* ini menandakan bahwa *request* yang dibuat salah atau data yang dikirim tidak ada.

4. 401 *Unauthorized*

*Response code* ini menandakan bahwa *request* yang dibuat membutuhkan *authentication* sebelum mengakses *resource*.

5. 404 *Not Found*

*Response code* ini menandakan bahwa *resource* yang dipanggil tidak ditemukan.

6. 405 *Method Not Allowed*

*Response code* ini menandakan bahwa *request endpoint* ada tetapi metode HTTP yang digunakan tidak diizinkan.

7. 409 *Conflict*

*Response code* ini menandakan bahwa *request* yang dibuat terdapat duplikasi, biasanya informasi yang dikirim sudah ada sebelumnya.

8. 500 *Internal Server Error*

*Response code* ini menandakan bahwa *request* yang dilakukan terdapat kesalahan pada sisi server atau *resource*.

d. *Format Response*

Setiap *request* yang dilakukan *client* akan menerima data *response* dari server, *response* tersebut biasanya berupa data XML ataupun JSON. Setelah mendapatkan data *response* tersebut barulah *client* bisa menggunakannya dengan cara memarsing data tersebut dan diolah sesuai kebutuhan. Berikut ini merupakan contoh hasil data response berupa XML dan JSON :

JSON

*GET /users/1234*

```

HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "id": "1234",
  "first_name": "jhon",
  "last_name": "doe",
  "created": "2015-05-22T14:56:29.000Z",
  "updated": "2015-05-22T14:56:29.000Z"
}

}

```

## 6. CodeIgniter

Menurut Blanco & Upton (2009:7) *CodeIgniter* adalah *powerful open source PHP framework* yang mudah dikuasai, dibangun untuk PHP *programmers* yang membutuhkan *toolkit* sederhana dan baik untuk membuat *full-featured web applications*. *CodeIgniter* adalah MVC *framework* yang di desain untuk mempermudah penggunanya.

### 2.4.2. Tools yang Digunakan dalam Perancangan Sistem

Berikut ini merupakan tools yang digunakan dalam merancang sistem yang akan dibuat, diantaranya :

#### 1. Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta – model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar – standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar – standar CORBA (*Common Object Request Broker Architecture*) (Fowler Martin, 2005).

Menurut Verdi Yasin (2012) UML adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan

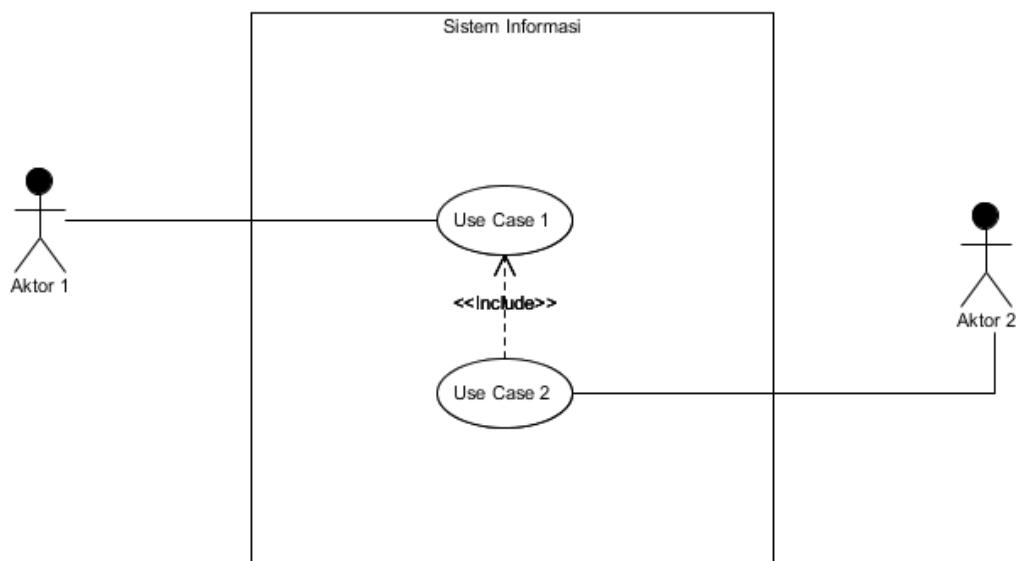
sistem piranti lunak. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. seperti bahasa - bahasa lainnya, UML mendefinisikan notasi dan *syntax/ semantik*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai *Diagram* piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: *OOD (Object-Oriented Design)*, *OMT (Object Modeling Technique)*, dan *OOSE (Object-Oriented Software Engineering)*.

Menurut Verdi Yasin (2012) di dalam UML terdapat beberapa *Diagram* yaitu:

a. *Use case Diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperlukan sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem.

Berikut ini merupakan contoh gambar *use case* :

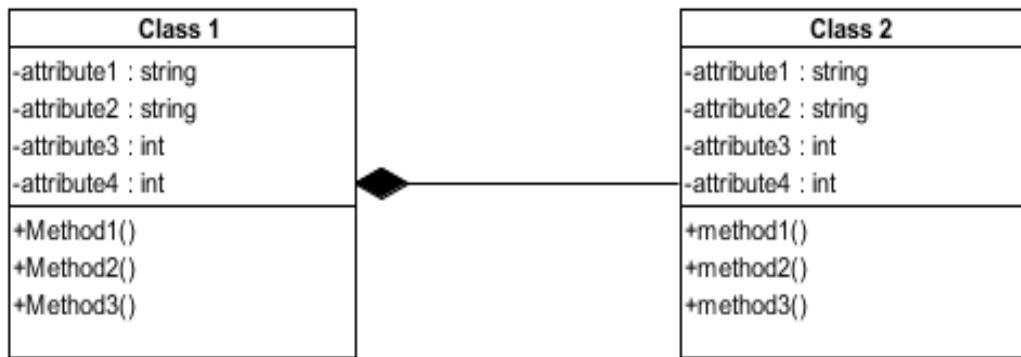


Gambar 2. 2 *Use Case Diagram*

b. *Class Diagram*

*Class diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

Berikut ini merupakan contoh gambar *class diagram* :

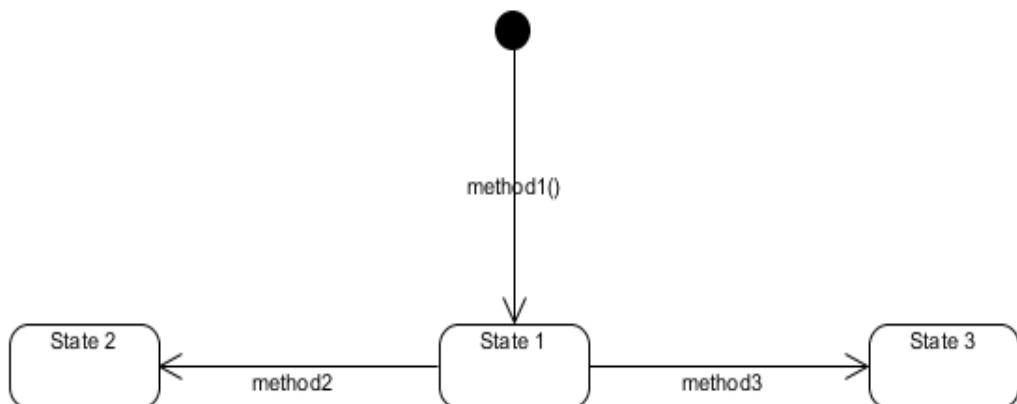


Gambar 2. 3 Class Diagram

c. *State Machine Diagram*

*State Machine Diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima.

Berikut ini merupakan contoh gambar *state machine diagram* :

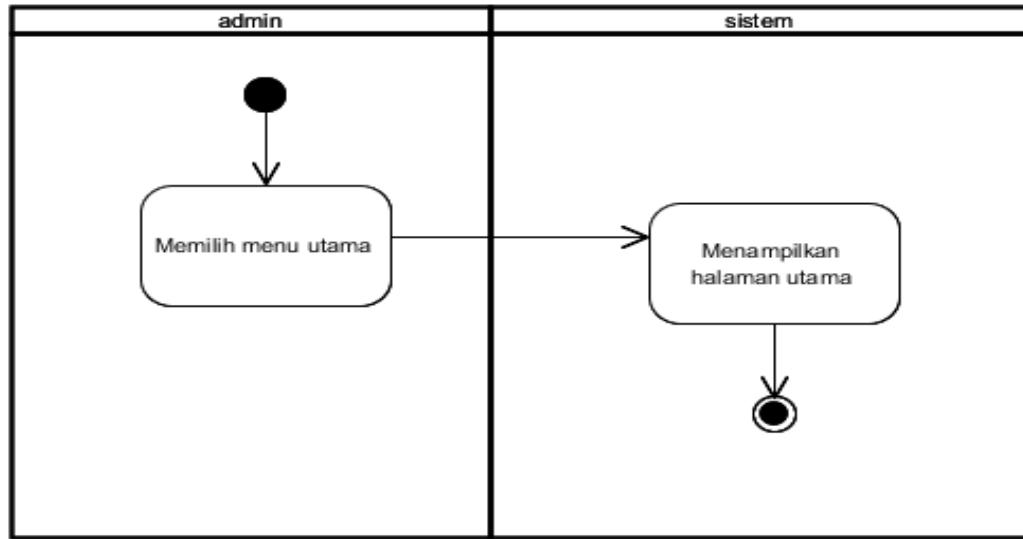


Gambar 2. 4 State Machine Diagram

d. *Swimlane Diagram*

*Swimlane Diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing – masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

Berikut ini merupakan contoh gambar *activity diagram* :

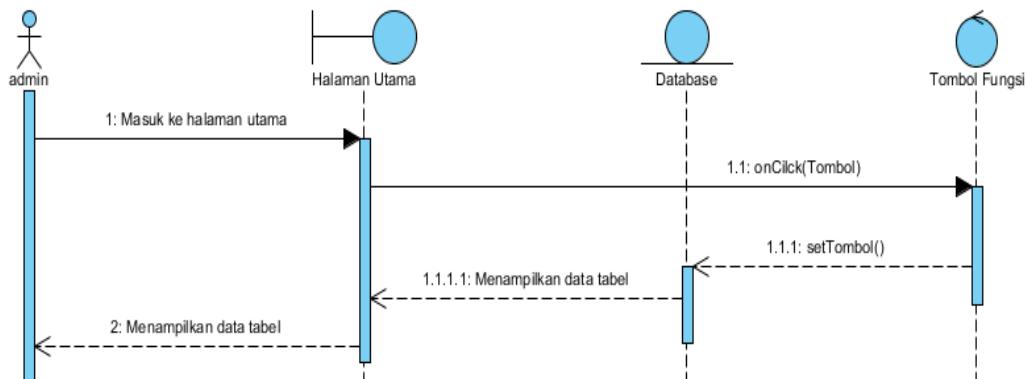


Gambar 2. 5 Activity Diagram

e. *Sequence Diagram*

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu.

Berikut ini merupakan contoh gambar *sequence diagram* :

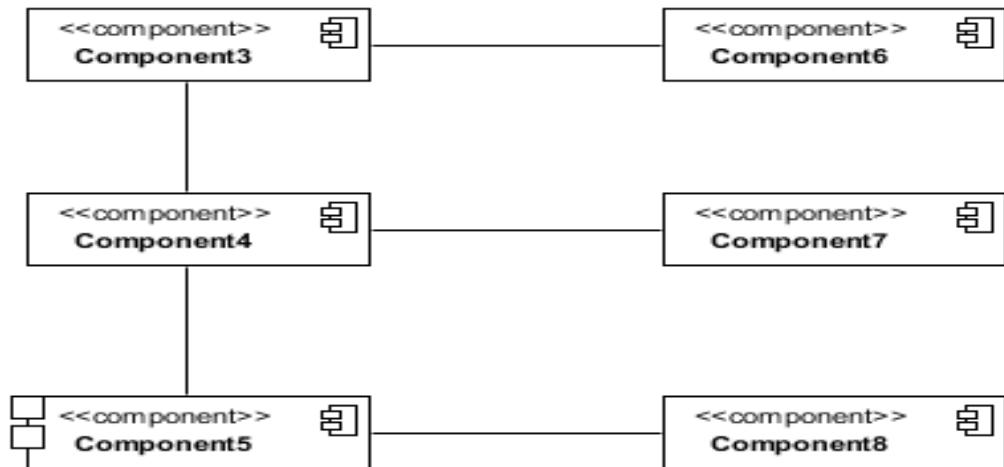


Gambar 2. 6 Sequence Diagram

f. *Component Diagram*

*Component Diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya.

Berikut ini merupakan contoh gambar *component diagram* :

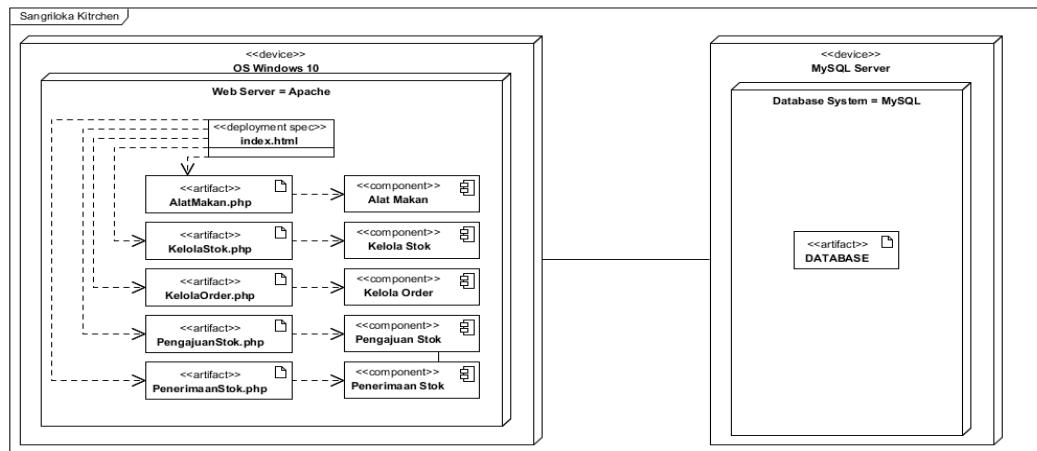


Gambar 2. 7 *Component Diagram*

g. *Deployment Diagram*

*Deployment Diagram* menggambarkan detil bagaimana komponen *di-deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal – hal lain yang bersifat fisikal.

Berikut ini merupakan contoh gambar *deployment diagram* :



Gambar 2. 8 Deployment Diagram

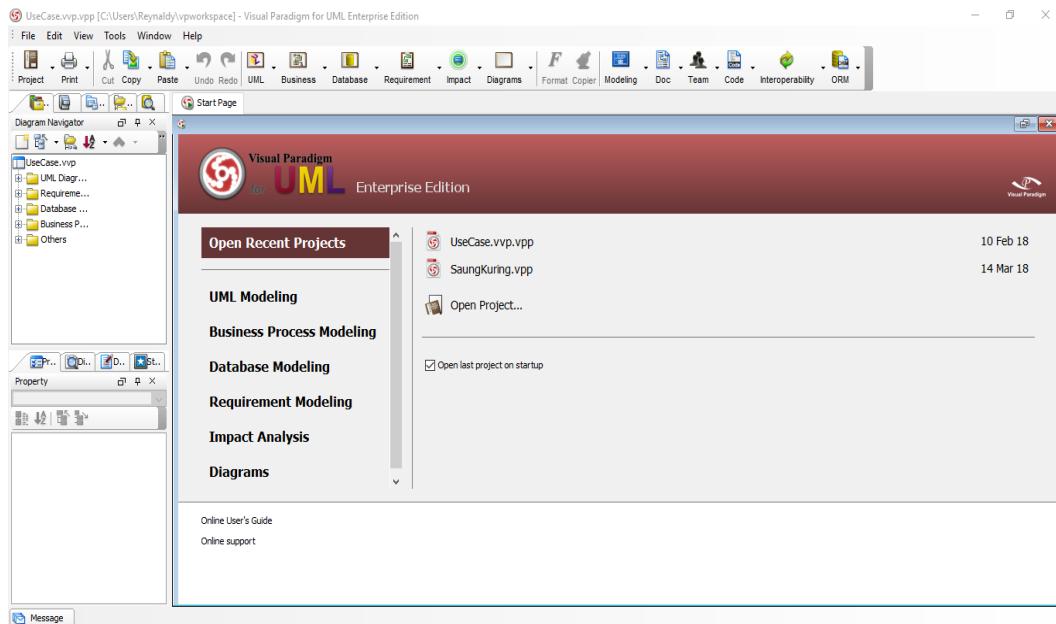
Dalam penelitian ini penulis hanya menggunakan 5 diagram yaitu *use case diagram*, *activity diagram*, *class diagram*, *sequence diagram*, dan *entity relationship diagram*.

## 2. Visual Paradigm

*Visual Paradigm* merupakan sebuah *software* model dengan sistem visualisasi yang memungkinkan model yang telah dibuat dapat digunakan sebagai representasi proyek-proyek lain dilengkapi dengan beberapa fitur yang ada didalamnya sampai pada menganalisa sebuah proyek yang akan dikerjakan. *Diagram* dapat disusun sedemikian rupa sehingga dapat dipustakakan menjadi proyek per proyek yang saling berkaitan. Hal ini dapat juga membantu memisahkan terhadap pekerjaan proyek sampai *level* terkecil.

*Visual Paradigm* dapat membandingkan perubahan antara *diagram* yang satu dengan yang lain yang saling berkaitan dan dapat digunakan untuk melacak perbedaan antara *Diagram-digran* yang ada. Dengan dukungan *design* layaknya *freehand* dapat menambahkan ke *diagram* untuk penjelasan lebih detail apa yang menjadi tujuan. Fasilitas lain dapat menggunakan mouse pen atau pena untuk sketching bentuk jalan, kita juga dapat menambahkan bentuk solid seperti bintang, dan panah *rectangles*.

Berikut ini merupakan gambar tampilan utama Visual Paradigm :



*Gambar 2. 9 Visual Paradigm*

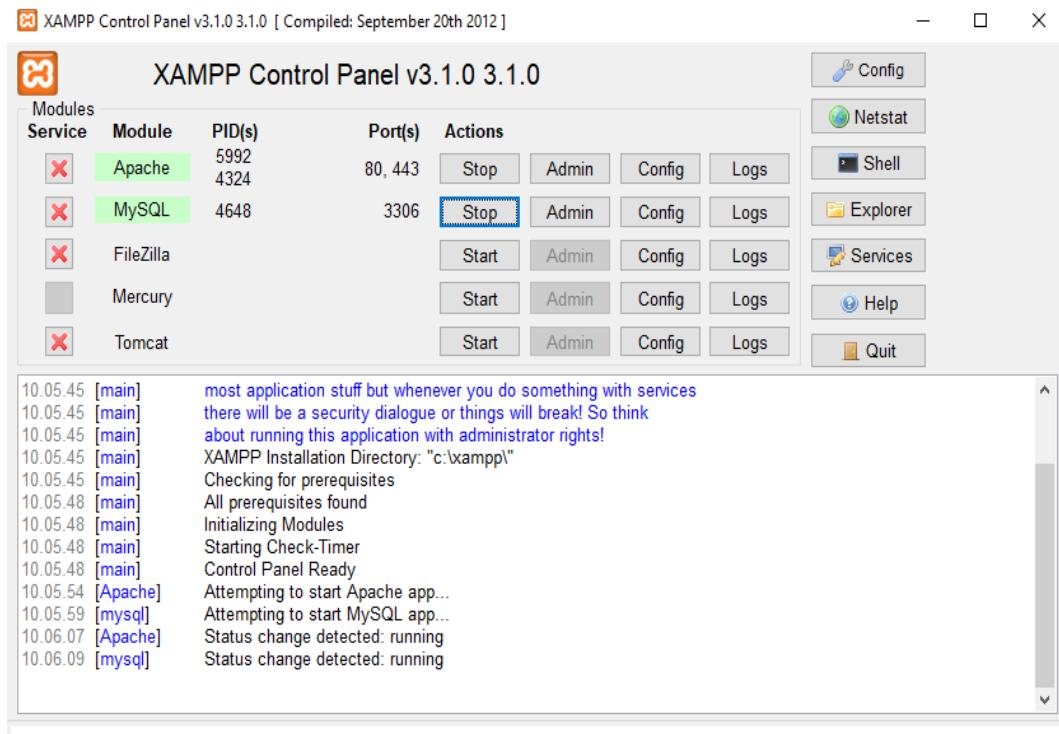
#### 2.4.3. Tools yang Digunakan dalam Pembuatan Sistem

Berikut ini merupakan *tool-tool* yang digunakan dalam pembuatan web (Admin Server), diantaranya :

##### 1. XAMMP

XAMPP adalah perangkat lunak yang menghubungkan tiga aplikasi kedalam satu paket yaitu Apache, MySQL, dan PHPMyAdmin. XAMPP ini merupakan sebuah perangkat lunak yang mudah digunakan dan mendukung instalasi di Linux dan Windows. Keuntungan menggunakan XAMPP juga adalah hanya menginstal satu kali sudah tersedia Apache Web Server, MySQL DatabaseServer, PHP Support dan beberapa modul lainnya.

Berikut ini merupakan gambar tampilan utama XAMPP :

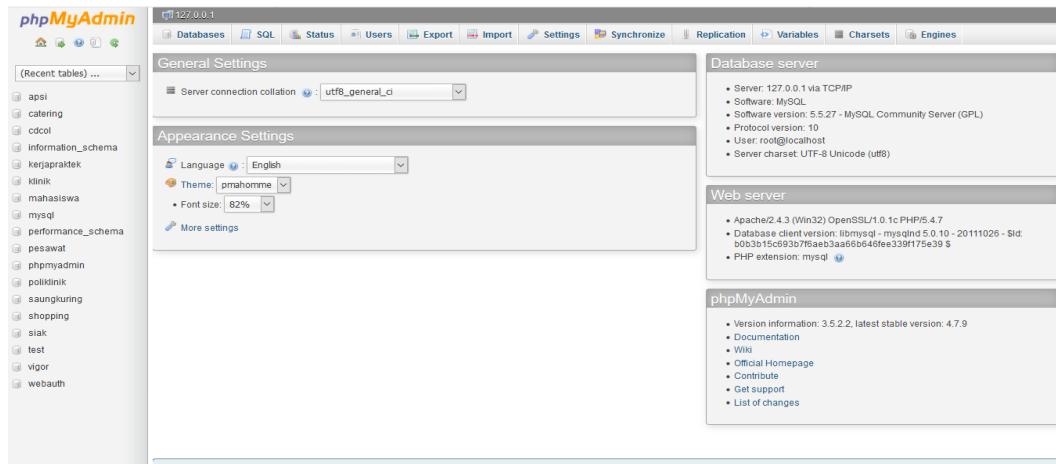


Gambar 2. 10 XAMPP Control Panel

## 2. MySQL

MySQL adalah sebuah *server database open source* yang terkenal yang digunakan berbagai aplikasi terutama untuk *server* atau membuat *web*. MySQL berfungsi sebagai SQL (*Structured Query Language*) yang dimiliki sendiri dan sudah diperluas oleh MySQL umumnya digunakan bersamaan dengan PHP untuk membuat aplikasi *server* dinamis.

Berikut ini merupakan gambar tampilan utama MySQL Server :



Gambar 2. 11 MySQL Server phpMyAdmin

### 3. Visual Studio Code

Visual Studio Code merupakan salah satu *text editor* yang dapat digunakan oleh *programmer*, khususnya *web developer*. Visual Studio Code memiliki beberapa kelebihan dari *text editor* lainnya, diantaranya :

- Terinstall otomatis fungsi EMMET

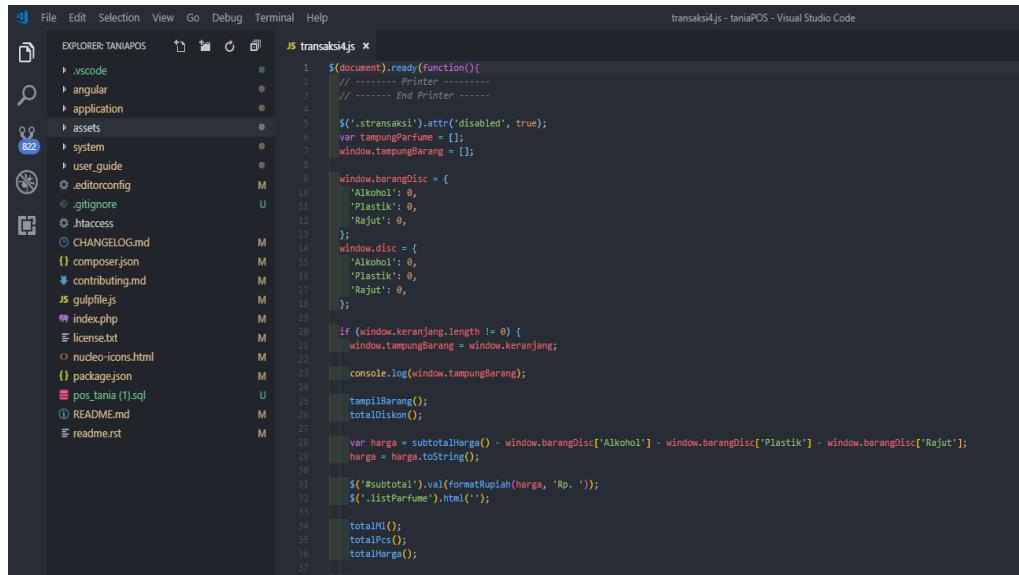
Fungsi auto complete dalam penulisan kode pemrograman.

- Linter & Intellisense*

Fungsi *error checker* yang sangat membantu ketika menulis kode pemrograman.

- Beautify*

Untuk merapikan susunan kode JavaScript, HTML, CSS, dan JSON yang sesuai standar.



```

File Edit Selection View Go Debug Terminal Help
EXPLORER: TANIAPOS
.js transaksi4.js
1 $(document).ready(function(){
2     // ----- Printer -----
3     // ----- End Printer -----
4
5     $('#stransaksi').attr('disabled', true);
6     var tumpungParfume = [];
7     window.tumpungBarang = [];
8
9     window.barangDisc = {
10        'Alkohol': 0,
11        'Plastik': 0,
12        'Rajut': 0,
13    };
14    window.disc = {
15        'Alkohol': 0,
16        'Plastik': 0,
17        'Rajut': 0,
18    };
19
20    if (window.keranjang.length != 0) {
21        window.tumpungBarang = window.keranjang;
22
23        console.log(window.tumpungBarang);
24
25        tampilBarang();
26        totalDiskon();
27
28        var harga = subtotalHarga() - window.barangDisc['Alkohol'] - window.barangDisc['Plastik'] - window.barangDisc['Rajut'];
29        harga = harga.toString();
30
31        $('#subTotal').val(formatRupiah(harga, 'Rp. '));
32        $('.listParfume').html('');
33
34        totalM();
35        totalPcs();
36        totalHarga();
37

```

*Gambar 2. 12 Visual Studio Code*

#### 4. Postman

Postman adalah aplikasi yang mendukung pengetesan API layaknya simulasi untuk melakukan testing permintaan dari *client* ke *server*.

#### 5. Web Hosting

*Web hosting* adalah *space* atau ruang *server* komputer yang digunakan sebagai penempatan data dan *file* aplikasi yang kita buat.

Berikut ini merupakan jenis-jenis hosting, diantaranya :

##### a. Shared Hosting

*Shared hosting* menggunakan server hosting bersama-sama dengan pengguna lain satu server yang dipergunakan oleh lebih dari satu nama domain. Artinya dalam satu server tersebut terdapat beberapa *account* yang dibedakan antara *account* satu dan lainnya dengan *username* dan *password*.

##### b. Virtual Private Server

*Virtual Private Server*, atau juga dikenal sebagai *Virtual Dedicated Server* merupakan proses virtualisasi dari lingkungan *software* sistem operasi yang dipergunakan oleh *server*. Karena lingkungan ini merupakan

lingkungan virtual, hal tersebut memungkinkan untuk menginstall sistem operasi yang dapat berjalan diatas sistem operasi lain.

c. *Dedicated Server*

*Dedicated server* adalah penggunaan *server* yang dikhususkan untuk aplikasi yang lebih besar dan tidak bisa dioperasikan dalam *shared hosting* atau *virtual dedicated server*. Dalam hal ini, penyediaan *server* ditanggung oleh perusahaan *hosting* yang biasanya bekerja sama dengan vendor.

d. *Colocation Server*

*Colocation server* adalah layanan penyewaan tempat untuk meletakkan *server* yang dipergunakan untuk *hosting*. *Server* disediakan oleh pelanggan yang biasanya bekerja sama dengan vendor.

## 6. Domain

Domain adalah nama unik yang diberikan untuk mengidentifikasi nama server komputer seperti web server atau email server di jaringan komputer atau internet. Nama domain berfungsi untuk mempermudah pengguna intenet pada saat melakukan akses ke server, selain itu domain juga digunakan untuk mengingat server yang dikunjungi tanpa harus mengenal deretan angka yang rumit seperti IP address. Nama domain dikenal sebagai suatu kesatuan dari sebuah situs web seperti contohnya “google.com”. nama domain kadang disebut juga dengan istilah alamat *website*.

Berikut ini merupakan jenis-jenis domain yang ada di Indonesia :

a. Domain AC.ID

AC.ID digunakan bagi lingkungan akademik/perguruan tinggi dengan ketentuan dan kebijakan yang telah ditentukan.

b. Domain OR.ID

OR.ID digunakan bagi organisasi selain organisasi yang masuk kedalam kategori domain AC.ID, CO.ID, NET.ID, GO.ID, MIL.ID, SCH.ID, dst dengan ketentuan dan kebijakan yang telah ditentukan.

## 2.5. Metode Pengujian

Pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada *output* yang dihasilkan yang merespon *input* yang dipilih dan kondisi eksekusi. (IEEE, 1990).

### 1. *Blackbox*

Metode pengujian *blackbox* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya. Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya (Ayuliana, 2009):

- a. Fungsi-fungsi yang salah atau hilang.
- b. Kesalahan *interface*.
- c. Kesalahan dalam struktur data atau akses *database* eksternal.
- d. Kesalahan performa.
- e. Kesalahan inisialisasi dan terminasi

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini menjelaskan tentang analisis sistem yang akan dibangun dan perancangan sistem. Perancangan sistem didalamnya mencakup *Use Case diagram*, *Swimlane diagram*, *Class diagram*, *Entity Relationship Diagram* dan perancangan antarmuka.

#### **3.1. Analisis Sistem**

Analisis sistem digunakan untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi serta kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan untuk suatu sistem.

##### **3.1.1. Analisis Sistem yang Sedang Berjalan**

Sistem informasi laboratorium informatika saat ini sudah cukup memadai namun masih ada beberapa kekurangan yaitu tidak adanya aplikasi untuk Mahasiswa/i (*client*) yang dapat digunakan untuk mengakses informasi seperti nilai, absensi, dan tugas. Serta tidak ada fasilitas untuk memberikan kritik dan saran. Dari sisi Asisten Laboratorium (Aslab) tidak terdapat fungsi untuk mengelola tugas praktikum di dalam sistem, saat ini untuk pengumpulan tugas yaitu dengan cara mahasiswa mengirim tugas melalui email masing-masing asisten atau dikumpulkan melalui *flashdisk*. Dan untuk penilaian tugasnya masih ditulis di kertas *form* penilaian tugas. Dari sisi Admin tidak terdapat fungsi untuk memberikan informasi seperti pengumuman jadwal tambahan, biasanya untuk mengirim informasi atau pengumuman pihak laboratorium melakukan *upload* di sosial media Facebook atau Instagram. Berdasarkan analisis inilah pengembangan sistem informasi laboratorium ini dibuat untuk memudahkan mahasiswa mengakses informasi tentang nilai-nilai praktikum, absensi, maupun jadwal. Dan mempermudah admin dalam mengkomunikasikan informasi kepada mahasiswa.

### 3.1.2. Analisis Kebutuhan Fungsional

*Requirement Analysis* atau analisis kebutuhan dibutuhkan untuk mengumpulkan dan merincikan persyaratan tingkat tinggi dari sebuah sistem serta mengidentifikasi fungsi-fungsi dasar yang akan dijalankan oleh *software*.

Aplikasi yang dibangun harus mempunyai fungsi utama yang mendukung pemecahan masalah yang ada pada sebuah studi kasus yang di ambil dalam hal ini sistem informasi manajemen kegiatan praktikum laboratorium teknik informatika. Fungsi yang dimaksud adalah apa saja yang dapat dilakukan dengan menggunakan aplikasi tersebut. Beberapa fungsi utama dalam Sistem Informasi Kegiatan Praktikum yang dibagi menjadi dua aplikasi adalah sebagai berikut :

a. Aplikasi *client* :

1. *Login*
2. Cek nilai
3. Cek absensi
4. Mengirim pengaduan atau kritik & saran
5. Membuat tiket permintaan pindah praktikum
6. Perpesanan dengan admin
7. Cek info praktikum
8. *Upload* tugas praktikum.

b. Aplikasi server :

1. *Login*
2. Kelola periode
3. Kelola jadwal
4. Registrasi mahasiswa
5. Kelola pembayaran
6. Kelola tiket pindah praktikum
7. Balas pesan mahasiswa
8. Kelola data *user*
9. Kelola data *master*
10. Penilaian praktikum

11. Absensi praktikum
12. Kelola pengambilan modul
13. Kelola tugas praktikum

### **3.1.3. Analisis Kebutuhan Non-Fungsional**

Analisis kebutuhan non-fungsional terdiri dari analisis pengguna, analisis perangkat keras, dan analisis perangkat lunak. Analisis kebutuhan non-fungsional diperlukan untuk mengetahui kebutuhan spesifikasi dari sistem yang akan dibangun.

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

Analisis Pengguna mendefinisikan siapa saja yang menggunakan sistem dan apa saja yang dapat dilakukan oleh pengguna di dalam sistem. Berikut merupakan pengguna dari Aplikasi Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika :

a. Koordinator Lab (Koorlab)

Koorlab dapat melihat data kehadiran seluruh asisten.

b. Koordinator Teknis (Koortek)

Koortek dapat melihat kritik dan saran atau pengaduan mahasiswa.

c. Admin Sistem (Admin)

Admin bertugas untuk melakukan pengelolaan seluruh data-data praktikum seperti registrasi praktikum, pembuatan jadwal, dan pengelolaan data master.

d. Asisten Laboratorium (Aslab)

Aslab bertugas untuk melakukan absensi praktikum, pengelolaan tugas dan penilaian praktikum.

e. Dosen

Dosen dapat melihat nilai praktikum mahasiswa.

f. Bagian Administrasi dan Keuangan (BAK)

BAK bertugas untuk melakukan pengelolaan pembayaran praktikum.

g. Mahasiswa

Di dalam aplikasi *client*, mahasiswa dapat melihat jadwal, nilai tugas, absensi, hasil praktikum, tugas praktikum, *upload* tugas praktikum, serta mengirim kritik dan saran.

## 2. Analisis Perangkat Keras

Analisis perangkat keras bertujuan untuk memenuhi kebutuhan minimal dari perangkat lunak yang akan menjalankan sistem. Berikut ini merupakan perangkat keras minimal yang dibutuhkan untuk menjalankan Sistem Informasi Kegiatan Praktikum :

*Tabel 3. 1 Kebutuhan Perangkat Keras (Komputer)*

No.	Perangkat	Spesifikasi
1	Processor	Intel Core 2 Duo
2	Memory RAM	2 GB
3	Harddisk	500GB (Min 3GB free space)
4	Mouse	Standar
5	Keyboard	Standar
6	Monitor	Standar

*Tabel 3. 2 Kebutuhan Perangkat Keras (Smartphone)*

No.	Perangkat	Spesifikasi
1	Processor	Dual Core
2	Memory RAM	1 GB
3	Penyimpanan Internal	8 GB

## 3. Analisis Perangkat Lunak

Analisis perangkat lunak merupakan mendefinisikan perangkat lunak apa saja yang digunakan oleh komputer atau *smartphone* untuk menjadi pendukung agar sistem dapat berjalan dengan semestinya. Berikut ini merupakan perangkat

lunak yang dibutuhkan untuk menjalankan Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika :

- a. Web Browser (direkomendasikan Google Chrome)

### **3.2. Perancangan Sistem**

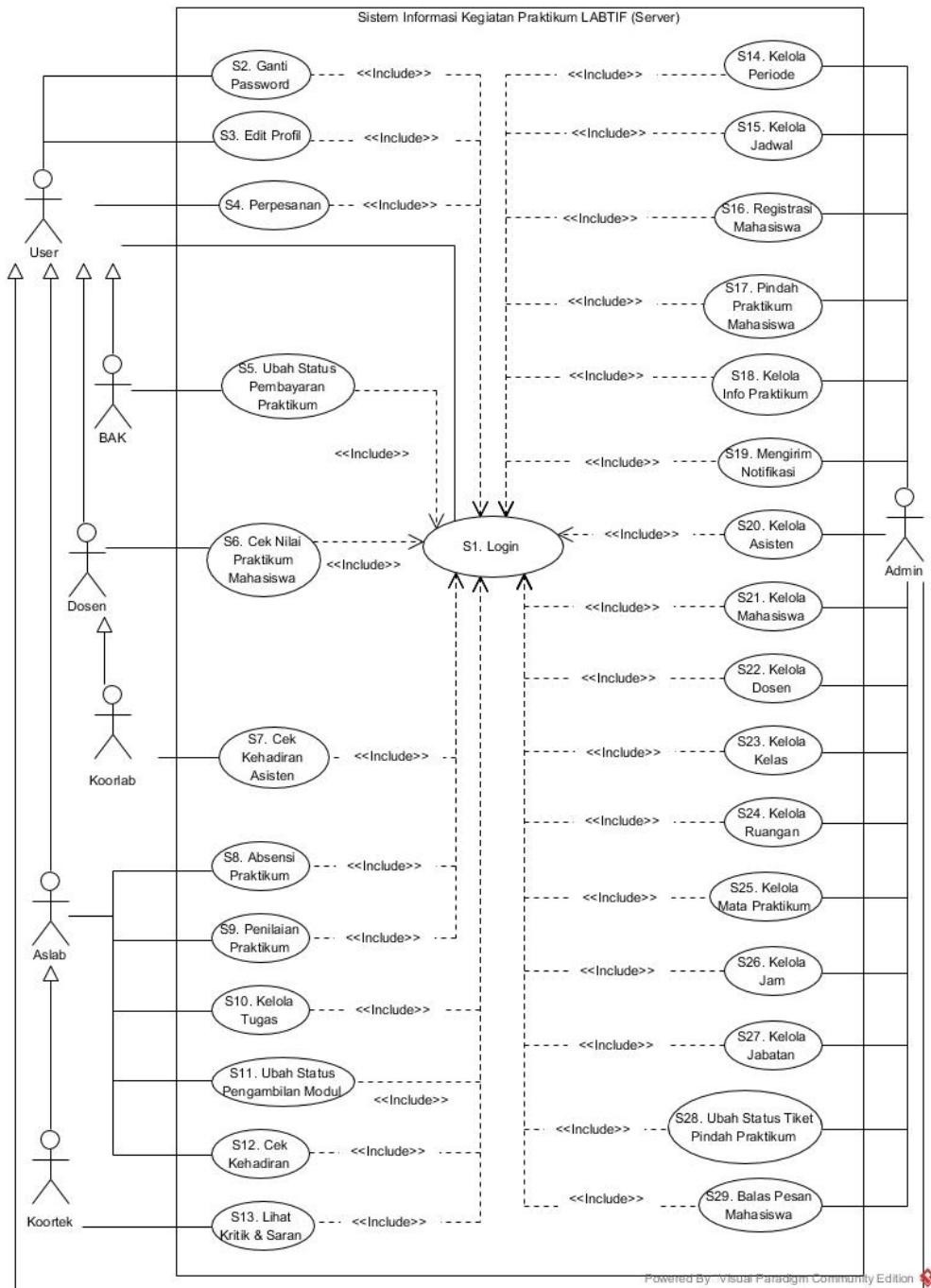
Perancangan sistem merupakan pengembangan sistem baru dari sistem yang telah ada, dimana pengembangan ini bertujuan untuk mengatasi permasalahan yang ditemui pada sistem yang lama. Pada tahap ini tentunya sudah diketahui permasalahan yang terjadi, sehingga pengembangan langsung dilakukan pada sistem yang baru.

Perancangan sistem dilakukan dengan menggunakan UML (*Unified Modeling Language*) sebagai software untuk menggambarkan sistem yang akan dibangun. Adapun tahapan analisis sistem menggunakan UML adalah *Use Case Diagram*, *Use Case Scenario*, *Swimlane Diagram*, *Class Diagram*, dan *Entity Relationship Diagram*.

#### **3.2.1. Use Case Diagram**

*Use Case Diagram* mendeskripsikan kelakuan sistem dari sudut pandang pengguna, berguna untuk membantu memahami kebutuhan. *Use Case* adalah dasar dari diagram lain. *Use Case* abstraksi dari interaksi antara sistem dan aktor. *Use Case* bekerja dengan mendeskripsikan tipe interaksi antara aktor sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Berikut merupakan *Use Case Diagram* dari Aplikasi Sistem Informasi Kegiatan Praktikum Laboratorium Komputer Teknik Informatika.

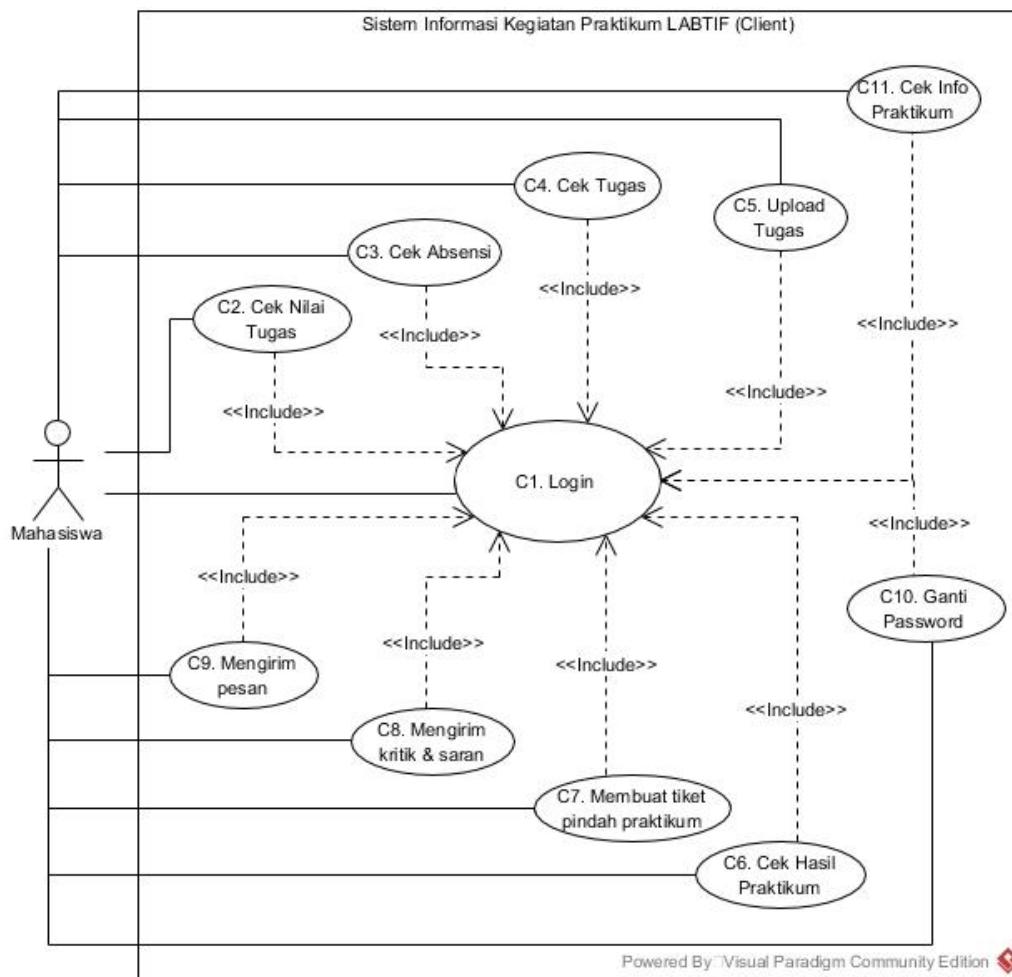
### 1. Use Case Diagram Aplikasi Server



Gambar 3. 1 Use Case Diagram Aplikasi Sistem Informasi Kegiatan Praktikum  
Laboratorium Teknik Informatika (Server)

Berdasarkan Gambar 3.1 *use case* aplikasi sistem informasi kegiatan praktikum laboratorium teknik informatika (server) terdapat 29 *use case*. Semua aktor harus melakukan *login* terlebih dahulu sebelum melakukan fungsi lain.

## 2. Use Case Diagram Aplikasi Client



*Gambar 3. 2 Use Case Diagram Aplikasi Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika (Client)*

Berdasarkan Gambar 3.2 *use case* aplikasi sistem informasi kegiatan praktikum laboratorium teknik informatika (*client*) terdapat 11 *use case*. Mahasiswa harus melakukan login terlebih dahulu sebelum melakukan fungsi lain.

### 3.2.2. Use Case Scenario

Berikut ini merupakan *Use Case Scenario* dari Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika:

#### 1. Use Case Scenario Login (Server)

Berikut ini merupakan *Use Case Scenario Login*.

*Tabel 3. 3 Use Case Scenario Login (Server)*

<i>Use Case ID</i>	S1
<i>Use Case Name</i>	Login
<i>Description</i>	<i>Use case</i> ini memungkinkan user untuk login ke dalam sistem agar dapat menjalankan fungsi sistem lainnya.
<i>Primary Actor</i>	User
<i>Secondary Actor</i>	-
<i>Included use cases</i>	-
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. User harus sudah mempunyai akun
<i>Postconditions</i>	1. Sistem mengalihkan user ke halaman dashboard
<i>Main Flow</i>	1. User mengisi <i>username &amp; password</i> 2. User menekan tombol login 3. Sistem memverifikasi <i>username &amp; password</i> 4. Sistem menampilkan halaman dashboard

#### 2. Use Case Scenario Ganti Password (Server)

Berikut ini merupakan *Use Case Scenario Ganti Password*.

*Tabel 3. 4 Use Case Scenario Ganti Password (Server)*

<i>Use Case ID</i>	S2
<i>Use Case Name</i>	Ganti Password

<i>Description</i>	<i>Use case</i> ini memungkinkan user untuk mengganti password akun masing-masing
<i>Primary Actor</i>	User
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. User harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem mengganti password user
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. User memilih menu ganti password</li> <li>2. Sistem menampilkan halaman ganti password</li> <li>3. User mengisi form password</li> <li>4. User menekan tombol ganti password</li> <li>5. Sistem mengganti password user</li> </ol>

### 3. Use Case Scenario Edit Profil

Berikut ini merupakan *Use Case Scenario Edit Profil*.

*Tabel 3. 5 Use Case Scenario Edit Profil*

<i>Use Case ID</i>	S3
<i>Use Case Name</i>	Edit Profil
<i>Description</i>	<i>Use case</i> ini memungkinkan user untuk mengganti data profil akun masing-masing
<i>Primary Actor</i>	User
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. User harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem mengganti data profil user
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. User memilih menu Edit Profil</li> <li>2. Sistem menampilkan halaman profil</li> </ol>

	3. User menekan tombol edit 4. User mengisi form profil 5. User menekan tombol simpan 6. Sistem menyimpan profil user
--	--

#### 4. Use Case Scenario Perpesanan

Berikut ini merupakan *Use Case Scenario Perpesanan..*

*Tabel 3. 6 Use Case Scenario Perpesanan*

<i>Use Case ID</i>	S4
<i>Use Case Name</i>	Perpesanan
<i>Description</i>	<i>Use case</i> ini memungkinkan user untuk melakukan percakapan dengan user lainnya.
<i>Primary Actor</i>	User
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. User harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menyimpan pesan user
<i>Main Flow</i>	1. User memilih menu Pesan 2. Sistem menampilkan halaman pesan 3. User memilih kontak 4. User mengisi kolom pesan 5. User menekan tombol kirim 6. Sistem menyimpan pesan user

#### 5. Use Case Scenario Ubah Status Pembayaran Praktikum

Berikut ini merupakan *Use Case Scenario Ubah Status Pembayaran Praktikum.*

*Tabel 3. 7 Use Case Scenario Ubah Status Pembayaran*

<i>Use Case ID</i>	S5
<i>Use Case Name</i>	Ubah Status Pembayaran Praktikum
<i>Description</i>	<i>Use case ini memungkinkan BAK untuk mengubah status pembayaran praktikum mahasiswa</i>
<i>Primary Actor</i>	BAK
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. BAK harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menyimpan status pembayaran praktikum mahasiswa
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. BAK memilih menu pembayaran praktikum</li> <li>2. Sistem menampilkan halaman pembayaran praktikum</li> <li>3. BAK menekan tombol status</li> <li>4. Sistem mengubah status pembayaran</li> </ol>

#### 6. Use Case Scenario Cek Nilai Praktikum Mahasiswa

Berikut ini merupakan *Use Case Scenario Cek Nilai Praktikum Mahasiswa*.

*Tabel 3. 8 Use Case Scenario Cek Nilai Praktikum Mahasiswa*

<i>Use Case ID</i>	S6
<i>Use Case Name</i>	Cek Nilai Praktikum Mahasiswa
<i>Description</i>	<i>Use case ini memungkinkan dosen untuk melihat nilai atau hasil praktikum mahasiswa</i>
<i>Primary Actor</i>	Dosen
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-

<i>Preconditions</i>	1. Dosen harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data nilai mahasiswa
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Dosen memilih menu Nilai Praktikum</li> <li>2. Dosen memilih mata praktikum dan kelas</li> <li>3. Sistem menampilkan halaman nilai praktikum</li> </ol>

#### 7. Use Case Scenario Cek Kehadiran Asisten

Berikut ini merupakan *Use Case Scenario Cek Kehadiran Asisten*.

*Tabel 3. 9 Use Case Scenario Cek Kehadiran Asisten*

<i>Use Case ID</i>	S7
<i>Use Case Name</i>	Cek Kehadiran Asisten
<i>Description</i>	<i>Use case</i> ini memungkinkan koorlab untuk melihat data kehadiran seluruh asisten
<i>Primary Actor</i>	Koorlab
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Koorlab harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data kehadiran asisten
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Koorlab memilih menu kehadiran asisten</li> <li>2. Koorlab memilih asisten</li> <li>3. Sistem menampilkan halaman data kehadiran asisten</li> </ol>

#### 8. Use Case Scenario Absensi Praktikum

Berikut ini merupakan *Use Case Scenario Absensi Praktikum*.

*Tabel 3. 10 Use Case Scenario Absensi Praktikum*

<i>Use Case ID</i>	S8
--------------------	----

<i>Use Case Name</i>	Absensi Praktikum
<i>Description</i>	<i>Use case ini memungkinkan asisten untuk mengabsen mahasiswa praktikum</i>
<i>Primary Actor</i>	Asisten
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Asisten harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menyimpan absensi mahasiswa praktikum
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Asisten memilih menu absensi praktikum</li> <li>2. Asisten memilih ID praktikum</li> <li>3. Sistem menampilkan halaman absensi praktikum</li> <li>4. Asisten menekan tombol kehadiran</li> <li>5. Sistem menyimpan data absen</li> </ol>

#### 9. Use Case Scenario Penilaian Praktikum

Berikut ini merupakan *Use Case Scenario Penilaian Praktikum*.

*Tabel 3. 11 Use Case Scenario Penilaian Praktikum*

<i>Use Case ID</i>	S9
<i>Use Case Name</i>	Absensi Praktikum
<i>Description</i>	<i>Use case ini memungkinkan asisten untuk melakukan penilaian praktikum. Nilai praktikum terdiri dari nilai kehadiran, nilai tugas dan nilai ujian.</i>
<i>Primary Actor</i>	Asisten
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Asisten harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menyimpan nilai mahasiswa praktikum

<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Asisten memilih menu penilaian praktikum</li> <li>2. Asisten memilih ID praktikum</li> <li>3. Sistem menampilkan halaman penilaian praktikum</li> <li>4. Asisten memilih nilai mahasiswa</li> <li>5. Sistem menampilkan kolom nilai</li> <li>6. Asisten mengisi kolom nilai</li> <li>7. Sistem menyimpan nilai praktikum</li> </ol>
------------------	---

#### 10. Use Case Scenario Kelola Tugas

Berikut ini merupakan *Use Case Scenario Kelola Tugas*.

*Tabel 3. 12 Use Case Scenario Kelola Tugas*

<i>Use Case ID</i>	S10
<i>Use Case Name</i>	Kelola Tugas
<i>Description</i>	<i>Use case</i> ini memungkinkan asisten untuk mengelola tugas-tugas praktikum. Mengelola di sini adalah, asisten dapat menambah, mengubah, dan menghapus tugas praktikum serta melakukan penilaian tugas
<i>Primary Actor</i>	Asisten
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<p>a. Buat Tugas</p> <ol style="list-style-type: none"> <li>1. Asisten harus sudah login ke dalam sistem</li> </ol> <p>b. Edit Tugas</p> <ol style="list-style-type: none"> <li>1. Asisten harus sudah login ke dalam sistem</li> <li>2. Sudah ada tugas yang dibuat</li> </ol> <p>c. Hapus Tugas</p> <ol style="list-style-type: none"> <li>1. Asisten harus sudah login ke dalam sistem</li> </ol>

	<ul style="list-style-type: none"> <li>2. Sudah ada tugas yang dibuat</li> <li>d. Penilaian Tugas           <ul style="list-style-type: none"> <li>1. Asisten harus sudah login ke dalam sistem</li> <li>2. Sudah ada tugas yang dibuat</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Buat Tugas           <ul style="list-style-type: none"> <li>1. Sistem menyimpan tugas yang dibuat asisten</li> </ul> </li> <li>b. Edit Tugas           <ul style="list-style-type: none"> <li>1. Sistem menyimpan data tugas yang diedit</li> </ul> </li> <li>c. Hapus Tugas           <ul style="list-style-type: none"> <li>1. Sistem menghapus data tugas</li> </ul> </li> <li>d. Penilaian Tugas           <ul style="list-style-type: none"> <li>1. Sistem menyimpan nilai tugas</li> </ul> </li> </ul>
<i>Main Flow</i>	<ul style="list-style-type: none"> <li>a. Buat Tugas           <ul style="list-style-type: none"> <li>1. Asisten memilih menu tugas praktikum</li> <li>2. Asisten memilih ID praktikum</li> <li>3. Sistem menampilkan halaman tugas praktikum</li> <li>4. Asistem menekan tombol buat tugas</li> <li>5. Sistem menampilkan form tambah tugas</li> <li>6. Asisten mengisi form</li> <li>7. Asisten menekan tombol simpan</li> <li>8. Sistem menyimpan tugas praktikum</li> </ul> </li> <li>b. Edit tugas           <ul style="list-style-type: none"> <li>1. Asisten memilih menu tugas praktikum</li> <li>2. Asisten memilih ID praktikum</li> <li>3. Sistem menampilkan halaman tugas praktikum</li> <li>4. Asistem menekan tombol edit tugas</li> <li>5. Sistem menampilkan form edit tugas</li> <li>6. Asisten mengubah isi form</li> </ul> </li> </ul>

	<p>7. Asisten menekan tombol simpan</p> <p>8. Sistem menyimpan tugas praktikum</p> <p>c. Hapus Tugas</p> <ol style="list-style-type: none"> <li>1. Asisten memilih menu tugas praktikum</li> <li>2. Asisten memilih ID praktikum</li> <li>3. Sistem menampilkan halaman tugas praktikum</li> <li>4. Asistem menekan tombol hapus tugas</li> <li>5. Sistem menampilkan dialog konfirmasi hapus tugas</li> <li>6. Asisten menekan tombol hapus</li> <li>7. Sistem menghapus tugas praktikum</li> </ol> <p>d. Penilaian Tugas</p> <ol style="list-style-type: none"> <li>1. Asisten memilih menu tugas praktikum</li> <li>2. Asisten memilih ID praktikum</li> <li>3. Sistem menampilkan halaman tugas praktikum</li> <li>4. Asistem memilih tugas praktikum</li> <li>5. Sistem menampilkan halaman penilaian tugas</li> <li>6. Asisten memilih mahasiswa</li> <li>7. Sistem menampilkan kolom nilai</li> <li>8. Asisten mengisi kolom nilai</li> <li>9. Sistem menyimpan nilai tugas</li> </ol>
--	---

### 11. Use Case Scenario Ubah Status Pengambilan Modul

Berikut ini merupakan *Use Case Scenario Ubah Status Pengambilan Modul*.

*Tabel 3. 13 Use Case Scenario Ubah Status Pengambilan modul*

<i>Use Case ID</i>	S11
<i>Use Case Name</i>	Ubah Status Pengambilan Modul

<i>Description</i>	<i>Use case</i> ini memungkinkan asisten untuk mengubah status pengambilan modul praktikum
<i>Primary Actor</i>	Asisten
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Asisten harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menyimpan status pengambilan modul praktikum
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Asisten memilih menu modul praktikum</li> <li>2. Sistem menampilkan halaman modul praktikum</li> <li>3. Asisten menekan tombol status</li> <li>4. Sistem mengubah status pengambilan modul praktikum</li> </ol>

## 12. Use Case Scenario Cek Kehadiran

Berikut ini merupakan *Use Case Scenario Cek Kehadiran*.

*Tabel 3. 14 Use Case Scenario Cek Kehadiran*

<i>Use Case ID</i>	S12
<i>Use Case Name</i>	Cek Kehadiran
<i>Description</i>	<i>Use case</i> ini memungkinkan asisten untuk melihat data kehadiran praktikum masing-masing
<i>Primary Actor</i>	Asisten
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Asisten harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data kehadiran asisten
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Asisten memilih menu kehadiran asisten</li> </ol>

	2. Sistem menampilkan halaman data kehadiran asisten
--	--

#### 13. Use Case Scenario Lihat Kritik & Saran

Berikut ini merupakan *Use Case Scenario Lihat Kritik & Saran*.

*Tabel 3. 15 Use Case Scenario Lihat Kritik & Saran*

<i>Use Case ID</i>	S13
<i>Use Case Name</i>	Lihat Kritik & Saran
<i>Description</i>	<i>Use case</i> ini memungkinkan koortek untuk melihat pengaduan atau kritik & saran dari mahasiswa terhadap Laboratorium Komputer Teknik Informatika
<i>Primary Actor</i>	Koortek
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Koortek harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data kritik & saran
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Koortek memilih menu kritik &amp; saran</li> <li>2. Sistem menampilkan halaman data kritik &amp; saran</li> </ol>

#### 14. Use Case Scenario Kelola Periode

Berikut ini merupakan *Use Case Scenario Kelola Periode*.

*Tabel 3. 16 Use Case Scenario Kelola Periode*

<i>Use Case ID</i>	S14
<i>Use Case Name</i>	Kelola Periode
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola periode praktikum. Mengelola di sini adalah, admin

	dapat menambah, mengubah, dan menghapus periode praktikum serta mengubah status periode
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Periode <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ul> </li> <li>b. Edit Periode <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada periode praktikum</li> </ul> </li> <li>c. Hapus Periode <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada periode praktikum</li> </ul> </li> <li>d. Status Periode <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada periode praktikum</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Periode <ul style="list-style-type: none"> <li>1. Sistem menyimpan periode praktikum</li> </ul> </li> <li>b. Edit Periode <ul style="list-style-type: none"> <li>1. Sistem menyimpan periode praktikum</li> </ul> </li> <li>c. Hapus Periode <ul style="list-style-type: none"> <li>1. Sistem menghapus periode praktikum</li> </ul> </li> <li>d. Status Periode <ul style="list-style-type: none"> <li>1. Sistem menyimpan status periode praktikum</li> </ul> </li> </ul>
<i>Main Flow</i>	<ul style="list-style-type: none"> <li>a. Tambah Periode <ul style="list-style-type: none"> <li>1. Admin memilih menu penjadwalan</li> <li>2. Sistem menampilkan halaman periode praktikum</li> <li>3. Admin menekan tombol tambah periode</li> </ul> </li> </ul>

	<ol style="list-style-type: none"><li>4. Sistem menampilkan form tambah periode</li><li>5. Admin mengisi form</li><li>6. Admin menekan tombol simpan</li><li>7. Sistem menyimpan periode praktikum</li></ol> <p>b. Edit Periode</p> <ol style="list-style-type: none"><li>1. Admin memilih menu penjadwalan</li><li>2. Sistem menampilkan halaman periode praktikum</li><li>3. Admin menekan tombol edit periode</li><li>4. Sistem menampilkan form edit periode</li><li>5. Admin mengubah isi form</li><li>6. Admin menekan tombol simpan</li><li>7. Sistem menyimpan periode praktikum</li></ol> <p>c. Hapus Periode</p> <ol style="list-style-type: none"><li>1. Admin memilih menu penjadwalan</li><li>2. Sistem menampilkan halaman periode praktikum</li><li>3. Admin menekan tombol hapus periode</li><li>4. Sistem menampilkan dialog konfirmasi hapus periode</li><li>5. Admin menekan tombol hapus</li><li>6. Sistem menghapus periode praktikum</li></ol> <p>d. Status Periode</p> <ol style="list-style-type: none"><li>1. Admin memilih menu penjadwalan</li><li>2. Sistem menampilkan halaman periode praktikum</li><li>3. Admin menekan tombol status periode</li><li>4. Sistem menyimpan status periode</li></ol>
--	---

### 15. Use Case Scenario Kelola Jadwal

Berikut ini merupakan *Use Case Scenario Kelola Jadwal*.

*Tabel 3. 17 Use Case Scenario Kelola Jadwal*

<i>Use Case ID</i>	S15
<i>Use Case Name</i>	Kelola Jadwal
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola jadwal praktikum. Mengelola di sini adalah, admin dapat menambah, mengubah, dan menghapus jadwal praktikum serta mengubah status jadwal
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Jadwal <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada periode praktikum</li> </ul> </li> <li>b. Edit Jadwal <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada jadwal praktikum</li> </ul> </li> <li>c. Hapus Jadwal <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada jadwal praktikum</li> </ul> </li> <li>d. Status Jadwal <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada jadwal praktikum</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Jadwal <ul style="list-style-type: none"> <li>1. Sistem menyimpan jadwal praktikum</li> </ul> </li> <li>b. Edit Jadwal <ul style="list-style-type: none"> <li>1. Sistem menyimpan jadwal praktikum</li> </ul> </li> <li>c. Hapus Jadwal</li> </ul>

	<ul style="list-style-type: none"> <li>1. Sistem menghapus jadwal praktikum</li> <li>d. Status Jadwal</li> <li>1. Sistem menyimpan status jadwal praktikum</li> </ul>
<i>Main Flow</i>	<ul style="list-style-type: none"> <li>a. Tambah Jadwal <ul style="list-style-type: none"> <li>1. Admin memilih menu penjadwalan</li> <li>2. Sistem menampilkan halaman periode praktikum</li> <li>3. Admin memilih periode praktikum</li> <li>4. Sistem menampilkan halaman penjadwalan</li> <li>5. Admin menekan tombol tambah jadwal</li> <li>6. Sistem menampilkan form tambah jadwal</li> <li>7. Admin mengisi form</li> <li>8. Admin menekan tombol simpan</li> <li>9. Sistem menyimpan jadwal praktikum</li> </ul> </li> <li>b. Edit Jadwal <ul style="list-style-type: none"> <li>1. Admin memilih menu penjadwalan</li> <li>2. Sistem menampilkan halaman periode praktikum</li> <li>3. Admin memilih periode praktikum</li> <li>4. Admin menampilkan halaman penjadwalan</li> <li>5. Admin menekan tombol edit jadwal</li> <li>6. Sistem menampilkan form edit jadwal</li> <li>7. Admin mengubah isi form</li> <li>8. Admin menekan tombol simpan</li> <li>9. Sistem menyimpan periode praktikum</li> </ul> </li> <li>c. Hapus Jadwal <ul style="list-style-type: none"> <li>1. Admin memilih menu penjadwalan</li> <li>2. Sistem menampilkan halaman periode praktikum</li> <li>3. Admin memilih periode praktikum</li> <li>4. Sistem menampilkan halaman penjadwalan</li> </ul> </li> </ul>

	<ol style="list-style-type: none"> <li>5. Admin menekan tombol hapus jadwal</li> <li>6. Sistem menampilkan dialog konfirmasi hapus jadwal</li> <li>7. Admin menekan tombol hapus</li> <li>8. Sistem menghapus jadwal praktikum</li> </ol> <p>d. Status Jadwal</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu penjadwalan</li> <li>2. Sistem menampilkan halaman periode praktikum</li> <li>3. Admin memilih periode praktikum</li> <li>4. Sistem menampilkan halaman penjadwalan</li> <li>5. Admin menekan tombol status jadwal</li> <li>6. Sistem menyimpan status jadwal</li> </ol>
--	---

#### 16. Use Case Scenario Registrasi Mahasiswa

Berikut ini merupakan *Use Case Scenario Registrasi Mahasiswa*.

*Tabel 3. 18 Use Case Scenario Registrasi Mahasiswa*

<i>Use Case ID</i>	S16
<i>Use Case Name</i>	Registrasi Mahasiswa
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk meregistrasi mahasiswa pada jadwal praktikum yang sudah dibuat
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada jadwal praktikum</li> </ol>

<i>Postconditions</i>	1. Sistem menyimpan data registrasi mahasiswa praktikum
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Admin memilih menu registrasi</li> <li>2. Admin memilih ID praktikum</li> <li>3. Sistem menampilkan halaman registrasi praktikum</li> <li>4. Admin memilih mahasiswa</li> <li>5. Sistem menyimpan data registrasi</li> </ol>

### 17. Use Case Scenario Pindah Praktikum Mahasiswa

Berikut ini merupakan *Use Case Scenario Pindah Praktikum Mahasiswa*.

*Tabel 3. 19 Use Case Scenario Pindah Praktikum Mahasiswa*

<i>Use Case ID</i>	S17
<i>Use Case Name</i>	Pindah Praktikum Mahasiswa
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk memproses pemindahan praktikum mahasiswa. Admin memproses permintaan mahasiswa yang ingin pindah jadwal praktikum dikarenakan bentrok dengan jadwal kuliah atau hal lainnya
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada jadwal praktikum</li> </ol>
<i>Postconditions</i>	1. Sistem menyimpan data pindah praktikum mahasiswa
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Admin memilih menu pindah praktikum</li> <li>2. Admin memilih ID praktikum awal</li> </ol>

	<ul style="list-style-type: none"> <li>3. Sistem menampilkan halaman pindah praktikum</li> <li>4. Admin memilih ID praktikum tujuan</li> <li>5. Admin memilih mahasiswa</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data pindah praktikum</li> </ul>
--	---

### 18. Use Case Scenario Kelola Info Praktikum

Berikut ini merupakan *Use Case Scenario Kelola Info Praktikum*.

*Tabel 3. 20 Use Case Scenario Kelola Info Praktikum*

<i>Use Case ID</i>	S18
<i>Use Case Name</i>	Kelola Info Praktikum
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola info praktikum. Mengelola di sini adalah, admin dapat menambah, mengubah, dan menghapus info praktikum.
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Info <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ul> </li> <li>b. Edit Info <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada info praktikum</li> </ul> </li> <li>c. Hapus Info <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada info praktikum</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Info <ul style="list-style-type: none"> <li>1. Sistem menyimpan info praktikum</li> </ul> </li> </ul>

	<p>b. Edit Info</p> <ol style="list-style-type: none"> <li>1. Sistem menyimpan info praktikum</li> </ol> <p>c. Hapus Info</p> <ol style="list-style-type: none"> <li>1. Sistem menghapus info praktikum</li> </ol>
<i>Main Flow</i>	<p>a. Tambah Info</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu info praktikum</li> <li>2. Sistem menampilkan halaman info praktikum</li> <li>3. Admin menekan tombol buat info</li> <li>4. Sistem menampilkan form tambah info</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan info praktikum</li> </ol> <p>b. Edit Info</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu info praktikum</li> <li>2. Sistem menampilkan halaman info praktikum</li> <li>3. Admin menekan tombol edit info</li> <li>4. Sistem menampilkan form edit info</li> <li>5. Admin mengubah isi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan periode praktikum</li> </ol> <p>c. Hapus Info</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu info praktikum</li> <li>2. Sistem menampilkan halaman info praktikum</li> <li>3. Admin menekan tombol hapus info</li> <li>4. Sistem menampilkan dialog konfirmasi hapus info</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus info praktikum</li> </ol>

### 19. Use Case Scenario Mengirim Notifikasi

Berikut ini merupakan *Use Case Scenario Mengirim Notifikasi*.

*Tabel 3. 21 Use Case Scenario Mengirim Notifikasi*

<i>Use Case ID</i>	S19
<i>Use Case Name</i>	Mengirim Notifikasi
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengirim notifikasi praktikum
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Admin harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem mengirim notifikasi praktikum
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Admin memilih menu info praktikum</li> <li>2. Sistem menampilkan halaman info praktikum</li> <li>3. Admin menekan tombol kirim notifikasi</li> <li>4. Sistem menampilkan form kirim notifikasi</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol kirim</li> <li>7. Sistem mengirim notifikasi</li> </ol>

### 20. Use Case Scenario Kelola Asisten

Berikut ini merupakan *Use Case Scenario Kelola Asisten*.

*Tabel 3.22 Use Case Scenario Kelola Asisten*

<i>Use Case ID</i>	S20
<i>Use Case Name</i>	Kelola Asisten
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola asisten. Mengelola di sini adalah, admin dapat

	menambah, mengubah, menghapus asisten serta mengubah status asisten dan reset password asisten
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Asisten <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ul> </li> <li>b. Edit Asisten <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data asisten</li> </ul> </li> <li>c. Hapus Asisten <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data asisten</li> </ul> </li> <li>d. Reset Password <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data asisten</li> </ul> </li> <li>e. Status Asisten <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data asisten</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Asisten <ul style="list-style-type: none"> <li>1. Sistem menyimpan data asisten</li> </ul> </li> <li>b. Edit Asisten <ul style="list-style-type: none"> <li>1. Sistem menyimpan data asisten</li> </ul> </li> <li>c. Hapus Asisten <ul style="list-style-type: none"> <li>1. Sistem menghapus data asisten</li> </ul> </li> <li>d. Reset Password <ul style="list-style-type: none"> <li>1. Sistem mereset password asisten</li> </ul> </li> <li>d. Status Asisten <ul style="list-style-type: none"> <li>1. Sistem menyimpan status asisten</li> </ul> </li> </ul>

<i>Main Flow</i>	<ul style="list-style-type: none"> <li>a. Tambah Asisten           <ul style="list-style-type: none"> <li>1. Admin memilih menu asisten</li> <li>2. Sistem menampilkan halaman asisten</li> <li>3. Admin menekan tombol tambah asisten</li> <li>4. Sistem menampilkan form tambah asisten</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data asisten</li> </ul> </li>   <li>b. Edit Asisten           <ul style="list-style-type: none"> <li>1. Admin memilih menu asisten</li> <li>2. Sistem menampilkan halaman asisten</li> <li>3. Admin menekan tombol edit asisten</li> <li>4. Sistem menampilkan form edit asisten</li> <li>5. Admin mengubah isi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data asisten</li> </ul> </li>   <li>c. Hapus Asisten           <ul style="list-style-type: none"> <li>1. Admin memilih menu asisten</li> <li>2. Sistem menampilkan halaman asisten</li> <li>3. Admin menekan tombol hapus asisten</li> <li>4. Sistem menampilkan dialog konfirmasi hapus asisten</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus data asisten</li> </ul> </li>   <li>d. Reset Password           <ul style="list-style-type: none"> <li>1. Admin memilih menu asisten</li> <li>2. Sistem menampilkan halaman asisten</li> <li>3. Admin menekan tombol reset password</li> <li>4. Sistem menampilkan dialog konfirmasi reset password</li> <li>5. Admin menekan tombol reset</li> </ul> </li> </ul>
------------------	--

	<ul style="list-style-type: none"> <li>6. Sistem mereset password asisten</li> <li>e. Status Asisten</li> <li>5. Admin memilih menu asisten</li> <li>6. Sistem menampilkan halaman asisten</li> <li>7. Admin menekan tombol status asisten</li> <li>8. Sistem menyimpan status asisten</li> </ul>
--	---

### 21. Use Case Scenario Kelola Mahasiswa

Berikut ini merupakan *Use Case Scenario Kelola Mahasiswa*.

*Tabel 3. 22 Use Case Scenario Kelola Mahasiswa*

<i>Use Case ID</i>	S21
<i>Use Case Name</i>	Kelola Mahasiswa
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola mahasiswa. Mengelola di sini adalah, admin dapat menambah, mengubah, menghapus mahasiswa serta mengubah status mahasiswa dan reset password mahasiswa
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Mahasiswa <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ul> </li> <li>b. Edit Mahasiswa <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data mahasiswa</li> </ul> </li> <li>c. Hapus Mahasiswa <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data mahasiswa</li> </ul> </li> </ul>

	<p>d. Reset Password</p> <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data mahasiswa</li> </ol> <p>e. Status Mahasiswa</p> <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data mahasiswa</li> </ol>
<i>Postconditions</i>	<p>a. Tambah Mahasiswa</p> <ol style="list-style-type: none"> <li>1. Sistem menyimpan data mahasiswa</li> </ol> <p>b. Edit Mahasiswa</p> <ol style="list-style-type: none"> <li>1. Sistem menyimpan data mahasiswa</li> </ol> <p>c. Hapus Asisten</p> <ol style="list-style-type: none"> <li>2. Sistem menghapus data mahasiswa</li> </ol> <p>d. Reset Password</p> <ol style="list-style-type: none"> <li>2. Sistem mereset password mahasiswa</li> </ol> <p>d. Status Asisten</p> <ol style="list-style-type: none"> <li>2. Sistem menyimpan status mahasiswa</li> </ol>
<i>Main Flow</i>	<p>a. Tambah Mahasiswa</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu mahasiswa</li> <li>2. Sistem menampilkan halaman mahasiswa</li> <li>3. Admin menekan tombol tambah mahasiswa</li> <li>4. Sistem menampilkan form tambah mahasiswa</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data mahasiswa</li> </ol> <p>b. Edit Mahasiswa</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu mahasiswa</li> <li>2. Sistem menampilkan halaman mahasiswa</li> <li>3. Admin menekan tombol edit mahasiswa</li> <li>4. Sistem menampilkan form edit mahasiswa</li> <li>5. Admin mengubah isi form</li> </ol>

	<p>6. Admin menekan tombol simpan</p> <p>7. Sistem menyimpan data mahasiswa</p> <p>c. Hapus Asisten</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu mahasiswa</li> <li>2. Sistem menampilkan halaman mahasiswa</li> <li>3. Admin menekan tombol hapus mahasiswa</li> <li>4. Sistem menampilkan dialog konfirmasi hapus mahasiswa</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus data mahasiswa</li> </ol> <p>d. Reset Password</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu mahasiswa</li> <li>2. Sistem menampilkan halaman mahasiswa</li> <li>3. Admin menekan tombol reset password</li> <li>4. Sistem menampilkan dialog konfirmasi reset password</li> <li>5. Admin menekan tombol reset</li> <li>6. Sistem mereset password mahasiswa</li> </ol> <p>e. Status Asisten</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu mahasiswa</li> <li>2. Sistem menampilkan halaman mahasiswa</li> <li>3. Admin menekan tombol status mahasiswa</li> <li>4. Sistem menyimpan status mahasiswa</li> </ol>
--	--

## 22. Use Case Scenario Kelola Dosen

Berikut ini merupakan *Use Case Scenario Kelola Dosen*.

*Tabel 3. 23 Use Case Scenario Kelola Dosen*

<i>Use Case ID</i>	S22
<i>Use Case Name</i>	Kelola Dosen

<i>Description</i>	<i>Use case ini memungkinkan admin untuk mengelola dosen. Mengelola di sini adalah, admin dapat menambah, mengubah, menghapus dosen serta mengubah status dosen dan reset password dosen</i>
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Dosen <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ul> </li> <li>b. Edit Dosen <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data dosen</li> </ul> </li> <li>c. Hapus Dosen <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data dosen</li> </ul> </li> <li>d. Reset Dosen <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data dosen</li> </ul> </li> <li>e. Status Dosen <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data dosen</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Dosen <ul style="list-style-type: none"> <li>1. Sistem menyimpan data dosen</li> </ul> </li> <li>b. Edit Dosen <ul style="list-style-type: none"> <li>1. Sistem menyimpan data dosen</li> </ul> </li> <li>c. Hapus Asisten <ul style="list-style-type: none"> <li>1. Sistem menghapus data dosen</li> </ul> </li> <li>d. Reset Password <ul style="list-style-type: none"> <li>1. Sistem mereset password dosen</li> </ul> </li> </ul>

	<p>d. Status Dosen</p> <ol style="list-style-type: none"> <li>1. Sistem menyimpan status dosen</li> </ol>
<i>Main Flow</i>	<p>a. Tambah Dosen</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu dosen</li> <li>2. Sistem menampilkan halaman dosen</li> <li>3. Admin menekan tombol tambah dosen</li> <li>4. Sistem menampilkan form tambah dosen</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data dosen</li> </ol> <p>b. Edit Dosen</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu dosen</li> <li>2. Sistem menampilkan halaman dosen</li> <li>3. Admin menekan tombol edit dosen</li> <li>4. Sistem menampilkan form edit dosen</li> <li>5. Admin mengubah isi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data dosen</li> </ol> <p>c. Hapus Dosen</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu dosen</li> <li>2. Sistem menampilkan halaman dosen</li> <li>3. Admin menekan tombol hapus dosen</li> <li>4. Sistem menampilkan dialog konfirmasi hapus dosen</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus data dosen</li> </ol> <p>d. Reset Password</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu dosen</li> <li>2. Sistem menampilkan halaman dosen</li> <li>3. Admin menekan tombol reset password</li> </ol>

	<ol style="list-style-type: none"> <li>4. Sistem menampilkan dialog konfirmasi reset password</li> <li>5. Admin menekan tombol reset</li> <li>6. Sistem mereset password dosen</li> </ol> <p>e. Status Dosen</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu dosen</li> <li>2. Sistem menampilkan halaman dosen</li> <li>3. Admin menekan tombol status dosen</li> <li>4. Sistem menyimpan status dosen</li> </ol>
--	---

### 23. Use Case Scenario Kelola Kelas

Berikut ini merupakan *Use Case Scenario Kelola Kelas*.

*Tabel 3. 24 Use Case Scenario Kelola Kelas*

<i>Use Case ID</i>	S23
<i>Use Case Name</i>	Kelola Kelas
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola kelas. Mengelola di sini adalah, admin dapat menambah, mengubah, dan menghapus kelas serta mengubah status kelas
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<p>a. Tambah Kelas</p> <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ol> <p>b. Edit Kelas</p> <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data kelas</li> </ol> <p>c. Hapus Kelas</p>

	<ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data kelas</li> <li>d. Status Kelas           <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data kelas</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Kelas           <ul style="list-style-type: none"> <li>1. Sistem menyimpan data kelas</li> </ul> </li> <li>b. Edit Kelas           <ul style="list-style-type: none"> <li>1. Sistem menyimpan data kelas</li> </ul> </li> <li>c. Hapus Kelas           <ul style="list-style-type: none"> <li>1. Sistem menghapus data kelas</li> </ul> </li> <li>d. Status Kelas           <ul style="list-style-type: none"> <li>1. Sistem menyimpan status kelas</li> </ul> </li> </ul>
<i>Main Flow</i>	<ul style="list-style-type: none"> <li>a. Tambah Kelas           <ul style="list-style-type: none"> <li>1. Admin memilih menu kelas</li> <li>2. Sistem menampilkan halaman kelas</li> <li>3. Admin menekan tombol tambah kelas</li> <li>4. Sistem menampilkan form tambah kelas</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data kelas</li> </ul> </li> <li>b. Edit Kelas           <ul style="list-style-type: none"> <li>1. Admin memilih menu kelas</li> <li>2. Sistem menampilkan halaman kelas</li> <li>3. Admin menekan tombol edit kelas</li> <li>4. Sistem menampilkan form edit kelas</li> <li>5. Admin mengubah isi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data kelas</li> </ul> </li> <li>c. Hapus Kelas</li> </ul>

	<ol style="list-style-type: none"> <li>1. Admin memilih menu kelas</li> <li>2. Sistem menampilkan halaman kelas</li> <li>3. Admin menekan tombol hapus kelas</li> <li>4. Sistem menampilkan dialog konfirmasi hapus kelas</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus data kelas</li> </ol> <p>d. Status Kelas</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu kelas</li> <li>2. Sistem menampilkan halaman kelas</li> <li>3. Admin menekan tombol status kelas</li> <li>4. Sistem menyimpan status kelas</li> </ol>
--	---

#### 24. Use Case Scenario Kelola Ruangan

Berikut ini merupakan *Use Case Scenario Kelola Ruangan*.

*Tabel 3. 25 Use Case Scenario Kelola Ruangan*

<i>Use Case ID</i>	S24
<i>Use Case Name</i>	Kelola Ruangan
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola ruangan praktikum. Mengelola di sini adalah, admin dapat menambah, mengubah, dan menghapus data ruangan praktikum
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<p>a. Tambah Ruangan</p> <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ol> <p>b. Edit Ruangan</p>

	<ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data ruangan</li> <li>c. Hapus Ruangan           <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data ruangan</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Ruangan           <ul style="list-style-type: none"> <li>1. Sistem menyimpan data ruangan</li> </ul> </li> <li>b. Edit Ruangan           <ul style="list-style-type: none"> <li>1. Sistem menyimpan data ruangan</li> </ul> </li> <li>c. Hapus Ruangan           <ul style="list-style-type: none"> <li>1. Sistem menghapus data ruangan</li> </ul> </li> </ul>
<i>Main Flow</i>	<ul style="list-style-type: none"> <li>a. Tambah Ruangan           <ul style="list-style-type: none"> <li>1. Admin memilih menu ruangan</li> <li>2. Sistem menampilkan halaman ruangan</li> <li>3. Admin menekan tombol tambah ruangan</li> <li>4. Sistem menampilkan form tambah ruangan</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data ruangan</li> </ul> </li> <li>b. Edit Ruangan           <ul style="list-style-type: none"> <li>1. Admin memilih menu ruangan</li> <li>2. Sistem menampilkan halaman ruangan</li> <li>3. Admin menekan tombol edit ruangan</li> <li>4. Sistem menampilkan form edit ruangan</li> <li>5. Admin mengubah isi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data ruangan</li> </ul> </li> <li>c. Hapus Ruangan           <ul style="list-style-type: none"> <li>1. Admin memilih menu ruangan</li> <li>2. Sistem menampilkan halaman ruangan</li> </ul> </li> </ul>

	<ol style="list-style-type: none"> <li>3. Admin menekan tombol hapus ruangan</li> <li>4. Sistem menampilkan dialog konfirmasi hapus ruangan</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus data ruangan</li> </ol>
--	---

## 25. Use Case Scenario Kelola Mata Praktikum

Berikut ini merupakan *Use Case Scenario Kelola Mata Praktikum*.

*Tabel 3. 26 Use Case Scenario Kelola Mata Praktikum*

<i>Use Case ID</i>	S25
<i>Use Case Name</i>	Kelola Kelas
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola mata praktikum (matkum). Mengelola di sini adalah, admin dapat menambah, mengubah, dan menghapus mata praktikum serta mengubah status mata praktikum
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>a. Tambah Matkum <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ol> </li> <li>b. Edit Matkum <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data mata praktikum</li> </ol> </li> <li>c. Hapus Matkum <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data mata praktikum</li> </ol> </li> <li>d. Status Matkum</li> </ol>

	<ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data mata praktikum</li> </ol>
<i>Postconditions</i>	<ol style="list-style-type: none"> <li>a. Tambah Matkum             <ol style="list-style-type: none"> <li>1. Sistem menyimpan data mata praktikum</li> </ol> </li> <li>b. Edit Matkum             <ol style="list-style-type: none"> <li>1. Sistem menyimpan data mata praktikum</li> </ol> </li> <li>c. Hapus Matkum             <ol style="list-style-type: none"> <li>1. Sistem menghapus data mata praktikum</li> </ol> </li> <li>d. Status Matkum             <ol style="list-style-type: none"> <li>1. Sistem menyimpan status mata praktikum</li> </ol> </li> </ol>
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>a. Tambah Matkum             <ol style="list-style-type: none"> <li>1. Admin memilih menu mata praktikum</li> <li>2. Sistem menampilkan halaman mata praktikum</li> <li>3. Admin menekan tombol tambah matkum</li> <li>4. Sistem menampilkan form tambah matkum</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data mata praktikum</li> </ol> </li> <li>b. Edit Matkum             <ol style="list-style-type: none"> <li>1. Admin memilih menu mata praktikum</li> <li>2. Sistem menampilkan halaman mata praktikum</li> <li>3. Admin menekan tombol edit matkum</li> <li>4. Sistem menampilkan form edit matkum</li> <li>5. Admin mengubah isi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data mata praktikum</li> </ol> </li> <li>c. Hapus Matkum             <ol style="list-style-type: none"> <li>1. Admin memilih menu mata praktikum</li> </ol> </li> </ol>

	<ol style="list-style-type: none"> <li>2. Sistem menampilkan halaman mata praktikum</li> <li>3. Admin menekan tombol hapus matkul</li> <li>4. Sistem menampilkan dialog konfirmasi hapus matkul</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus data mata praktikum</li> </ol> <p>d. Status Matkul</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu mata praktikum</li> <li>2. Sistem menampilkan halaman mata praktikum</li> <li>3. Admin menekan tombol status matkul</li> <li>4. Sistem menyimpan status mata praktikum</li> </ol>
--	---

## 26. Use Case Scenario Kelola Jam

Berikut ini merupakan *Use Case Scenario Kelola Jam*.

*Tabel 3. 27 Use Case Scenario Kelola Jam*

<i>Use Case ID</i>	S26
<i>Use Case Name</i>	Kelola Jam
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola jam praktikum. Mengelola di sini adalah, admin dapat menambah, mengubah, dan menghapus data jam praktikum
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<p>a. Tambah Jam</p> <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ol>

	<p>b. Edit Jam</p> <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data jam</li> </ol> <p>c. Hapus Jam</p> <ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data jam</li> </ol>
<i>Postconditions</i>	<p>a. Tambah Jam</p> <ol style="list-style-type: none"> <li>1. Sistem menyimpan data jam</li> </ol> <p>b. Edit Jam</p> <ol style="list-style-type: none"> <li>1. Sistem menyimpan data jam</li> </ol> <p>c. Hapus Jam</p> <ol style="list-style-type: none"> <li>1. Sistem menghapus data jam</li> </ol>
<i>Main Flow</i>	<p>a. Tambah Jam</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu jam</li> <li>2. Sistem menampilkan halaman jam</li> <li>3. Admin menekan tombol tambah jam</li> <li>4. Sistem menampilkan form tambah jam</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data jam</li> </ol> <p>b. Edit Jam</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu jam</li> <li>2. Sistem menampilkan halaman jam</li> <li>3. Admin menekan tombol edit jam</li> <li>4. Sistem menampilkan form edit jam</li> <li>5. Admin mengubah isi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data jam</li> </ol> <p>c. Hapus Jam</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu jam</li> </ol>

	<ol style="list-style-type: none"> <li>2. Sistem menampilkan halaman jam</li> <li>3. Admin menekan tombol hapus jam</li> <li>4. Sistem menampilkan dialog konfirmasi hapus jam</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus data jam</li> </ol>
--	--

### 27. Use Case Scenario Kelola Jabatan

Berikut ini merupakan *Use Case Scenario Kelola Jabatan*.

*Tabel 3. 28 Use Case Scenario Kelola Jabatan*

<i>Use Case ID</i>	S27
<i>Use Case Name</i>	Kelola Jabatan
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengelola jabatan. Mengelola di sini adalah, admin dapat menambah, mengubah, dan menghapus jabatan
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Jabatan <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> </ul> </li> <li>b. Edit Jabatan <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data jabatan</li> </ul> </li> <li>c. Hapus Jam <ul style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada data jabatan</li> </ul> </li> </ul>
<i>Postconditions</i>	<ul style="list-style-type: none"> <li>a. Tambah Jabatan <ul style="list-style-type: none"> <li>1. Sistem menyimpan data jabatan</li> </ul> </li> </ul>

	<p>b. Edit Jabatan</p> <ol style="list-style-type: none"> <li>1. Sistem menyimpan data jabatan</li> </ol> <p>c. Hapus Jabatan</p> <ol style="list-style-type: none"> <li>1. Sistem menghapus data jabatan</li> </ol>
<i>Main Flow</i>	<p>a. Tambah Jabatan</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu jabatan</li> <li>2. Sistem menampilkan halaman jabatan</li> <li>3. Admin menekan tombol tambah jabatan</li> <li>4. Sistem menampilkan form tambah jabatan</li> <li>5. Admin mengisi form</li> <li>6. Admin menekan tombol simpan</li> <li>7. Sistem menyimpan data jabatan</li> </ol> <p>b. Edit Jabatan</p> <ol style="list-style-type: none"> <li>8. Admin memilih menu jabatan</li> <li>9. Sistem menampilkan halaman jabatan</li> <li>10. Admin menekan tombol edit jabatan</li> <li>11. Sistem menampilkan form edit jabatan</li> <li>12. Admin mengubah isi form</li> <li>13. Admin menekan tombol simpan</li> <li>14. Sistem menyimpan data jabatan</li> </ol> <p>c. Hapus Jabatan</p> <ol style="list-style-type: none"> <li>1. Admin memilih menu jabatan</li> <li>2. Sistem menampilkan halaman jabatan</li> <li>3. Admin menekan tombol hapus jabatan</li> <li>4. Sistem menampilkan dialog konfirmasi hapus jabatan</li> <li>5. Admin menekan tombol hapus</li> <li>6. Sistem menghapus data jabatan</li> </ol>

### 28. Use Case Scenario Ubah Status Tiket Pindah Praktikum

Berikut ini merupakan *Use Case Scenario Ubah Status Tiket Pindah Praktikum*.

*Tabel 3. 29 Use Case Scenario Ubah Status Tiket Pindah Praktikum*

<i>Use Case ID</i>	S28
<i>Use Case Name</i>	Ubah Status Tiket Pindah Praktikum
<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk mengubah status tiket permintaan pindah praktikum yang dibuat oleh mahasiswa
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada tiket masuk</li> </ol>
<i>Postconditions</i>	<ol style="list-style-type: none"> <li>1. Sistem mengubah status tiket</li> </ol>
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Admin memilih menu e-ticketing</li> <li>2. Sistem menampilkan halaman e-ticketing</li> <li>3. Admin menekan tombol lihat tiket</li> <li>4. Sistem menampilkan detail tiket</li> <li>5. Admin mengubah status tiket</li> <li>6. Sistem mengubah status tiket</li> </ol>

### 29. Use Case Scenario Balas Pesan Mahasiswa

Berikut ini merupakan *Use Case Scenario Balas Pesan Mahasiswa*.

*Tabel 3. 30 Use Case Scenario Balas Pesan Mahasiswa*

<i>Use Case ID</i>	S29
<i>Use Case Name</i>	Balas Pesan Mahasiswa

<i>Description</i>	<i>Use case</i> ini memungkinkan admin untuk membalas pesan atau pertanyaan yang dikirim oleh mahasiswa
<i>Primary Actor</i>	Admin
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. Admin harus sudah login ke dalam sistem</li> <li>2. Sudah ada pesan masuk</li> </ol>
<i>Postconditions</i>	<ol style="list-style-type: none"> <li>1. Sistem menyimpan pesan admin</li> </ol>
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Admin memilih menu e-ticketing</li> <li>2. Sistem menampilkan halaman e-ticketing</li> <li>3. Admin memilih pesan mahasiswa</li> <li>4. Sistem menampilkan kolom pesan</li> <li>5. Admin mengisi kolom pesan</li> <li>6. Admin menekan tombol kirim</li> <li>7. Sistem menyimpan pesan admin</li> </ol>

### 30. Use Case Scenario Login (Client)

Berikut ini merupakan *Use Case Scenario Login*.

*Tabel 3. 31 Use Case Scenario Login (Client)*

<i>Use Case ID</i>	C1
<i>Use Case Name</i>	Login
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk login ke dalam sistem agar dapat menjalankan fungsi sistem lainnya.
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	-
<i>Extends use cases</i>	-
<i>Preconditions</i>	<ol style="list-style-type: none"> <li>1. Mahasiswa harus sudah mempunyai akun</li> </ol>

<i>Postconditions</i>	1. Sistem mengalihkan mahasiswa ke halaman utama aplikasi
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. User mengisi NPM &amp; password</li> <li>2. User menekan tombol login</li> <li>3. Sistem memverifikasi NPM &amp; password</li> <li>4. Sistem menampilkan halaman utama aplikasi</li> </ol>

### 31. Use Case Scenario Cek Nilai Tugas

Berikut ini merupakan *Use Case Cek Nilai Tugas*.

*Tabel 3. 32 Use Case Cek Nilai Tugas*

<i>Use Case ID</i>	C2
<i>Use Case Name</i>	Cek Nilai Tugas
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk melihat nilai tugas semua praktikum yang diambil.
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data nilai tugas
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih menu nilai</li> <li>2. Sistem menampilkan halaman nilai tugas</li> <li>3. Mahasiswa memilih ID praktikum</li> <li>4. Sistem menampilkan data nilai tugas praktikum yang dipilih</li> </ol>

### 32. Use Case Scenario Cek Absensi

Berikut ini merupakan *Use Case Cek Absensi*.

*Tabel 3. 33 Use Case Cek Absensi*

<i>Use Case ID</i>	C3
<i>Use Case Name</i>	Cek Absensi
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk melihat data absensi semua praktikum yang diambil.
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data absensi
<i>Main Flow</i>	5. Mahasiswa memilih menu absensi 6. Sistem menampilkan halaman absensi 7. Mahasiswa memilih ID praktikum 8. Sistem menampilkan data absensi praktikum yang dipilih

### 33. Use Case Scenario Cek Tugas

Berikut ini merupakan *Use Case Cek Tugas*.

*Tabel 3. 34 Use Case Cek Tugas*

<i>Use Case ID</i>	C4
<i>Use Case Name</i>	Cek Tugas
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk melihat tugas praktikum yang tersedia
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login

<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data tugas
<i>Main Flow</i>	1. Mahasiswa memilih menu tugas 2. Sistem menampilkan halaman tugas

#### 34. Use Case Scenario Upload Tugas

Berikut ini merupakan *Use Case Upload Tugas*.

*Tabel 3. 35 Use Case Upload Tugas*

<i>Use Case ID</i>	C5
<i>Use Case Name</i>	Upload Tugas
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk mengirim file tugas praktikum.
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem mengupload file tugas mahasiswa ke server
<i>Main Flow</i>	1. Mahasiswa memilih menu tugas 2. Sistem menampilkan halaman tugas 3. Mahasiswa memilih tugas 4. Sistem menampilkan halaman detail tugas 5. Mahasiswa menekan tombol kumpulkan tugas 6. Sistem menampilkan form upload 7. Mahasiswa menekan tombol pilih file 8. OS menampilkan file manager 9. Mahasiswa memilih file tugas yang akan dikirim

	10. Sistem menyampaikan file tugas yang dipilih 11. Mahasiswa menekan tombol upload 12. Sistem mengupload file tugas ke server
--	--

### 35. Use Case Scenario Cek Hasil Praktikum

Berikut ini merupakan *Use Case Cek Hasil Praktikum*.

*Tabel 3. 36 Use Case Cek Hasil Praktikum*

<i>Use Case ID</i>	C6
<i>Use Case Name</i>	Cek Hasil Praktikum
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk melihat total nilai praktikum yang didapatkan. Total nilai praktikum ini adalah hasil perhitungan dari nilai kehadiran, nilai tugas, dan nilai ujian praktikum
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data nilai tugas
<i>Main Flow</i>	1. Mahasiswa memilih menu hasil akhir 2. Sistem menampilkan halaman hasil akhir 3. Mahasiswa memilih ID praktikum 4. Sistem menampilkan data nilai praktikum yang dipilih

### 36. Use Case Scenario Membuat Tiket Pindah Praktikum

Berikut ini merupakan *Use Case Membuat Tiket Pindah Praktikum*.

*Tabel 3. 37 Use Case Membuat Tiket Pindah Praktikum*

<i>Use Case ID</i>	C7
<i>Use Case Name</i>	Membuat Tiket Pindah Praktikum
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk membuat tiket pindah praktikum. Mahasiswa dapat melakukan permintaan pindah praktikum karena jadwal yang didapatkan bentrok dengan jadwal kuliah atau lainnya
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem mengirim tiket pindah praktikum ke server
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih menu e-ticketing</li> <li>2. Sistem menampilkan halaman e-ticketing</li> <li>3. Mahasiswa memilih menu permintaan pindah praktikum</li> <li>4. Sistem menampilkan form tiket pindah praktikum</li> <li>5. Mahasiswa mengisi form tiket</li> <li>6. Mahasiswa menekan tombol buat tiket</li> <li>7. Sistem menyimpan tiket</li> </ol>

### 37. Use Case Scenario Mengirim Kritik & Saran

Berikut ini merupakan *Use Case Mengirim Kritik & Saran*.

*Tabel 3. 38 Use Case Mengirim Kritik & Saran*

<i>Use Case ID</i>	C8
<i>Use Case Name</i>	Mengirim Kritik & Saran
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk mengirim pengaduan atau kritik & saran kepada pihak laboratorium.
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem mengirim data kritik & saran ke server
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih menu e-ticketing</li> <li>2. Sistem menampilkan halaman e-ticketing</li> <li>3. Mahasiswa memilih menu kritik &amp; saran</li> <li>4. Sistem menampilkan form kritik &amp; saran</li> <li>5. Mahasiswa mengisi form</li> <li>6. Mahasiswa menekan tombol kirim</li> <li>7. Sistem mengirim data kritik &amp; saran</li> </ol>

### 38. Use Case Scenario Mengirim Pesan

Berikut ini merupakan *Use Case Mengirim Pesan*.

*Tabel 3. 39 Use Case Mengirim Pesan*

<i>Use Case ID</i>	C9
<i>Use Case Name</i>	Mengirim Pesan

<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk mengirim pesan kepada admin atau melakukan percakapan dengan admin.
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem mengirim pesan ke server
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih menu e-ticketing</li> <li>2. Sistem menampilkan halaman e-ticketing</li> <li>3. Mahasiswa menekan tombol chat</li> <li>4. Mahasiswa form pesan</li> <li>5. Mahasiswa mengisi kolom pesan</li> <li>6. Mahasiswa menekan tombol kirim</li> <li>7. Sistem mengirim pesan</li> </ol>

### 39. Use Case Scenario Ganti Password (Client)

Berikut ini merupakan *Use Case Scenario Ganti Password*.

*Tabel 3. 40 Use Case Scenario Ganti Password (Client)*

<i>Use Case ID</i>	C10
<i>Use Case Name</i>	Ganti Password
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk mengganti password akun masing-masing
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem

<i>Postconditions</i>	1. Sistem mengganti password mahasiswa
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Mahasiswa menekan tombol menu</li> <li>2. Sistem menampilkan menu</li> <li>3. Mahasiswa memilih menu ganti password</li> <li>4. Sistem menampilkan halaman ganti password</li> <li>5. Mahasiswa mengisi form password</li> <li>6. Mahasiswa menekan tombol ganti password</li> </ol>

#### 40. Use Case Scenario Cek Info Praktikum

Berikut ini merupakan *Use Case Cek Info Praktikum*.

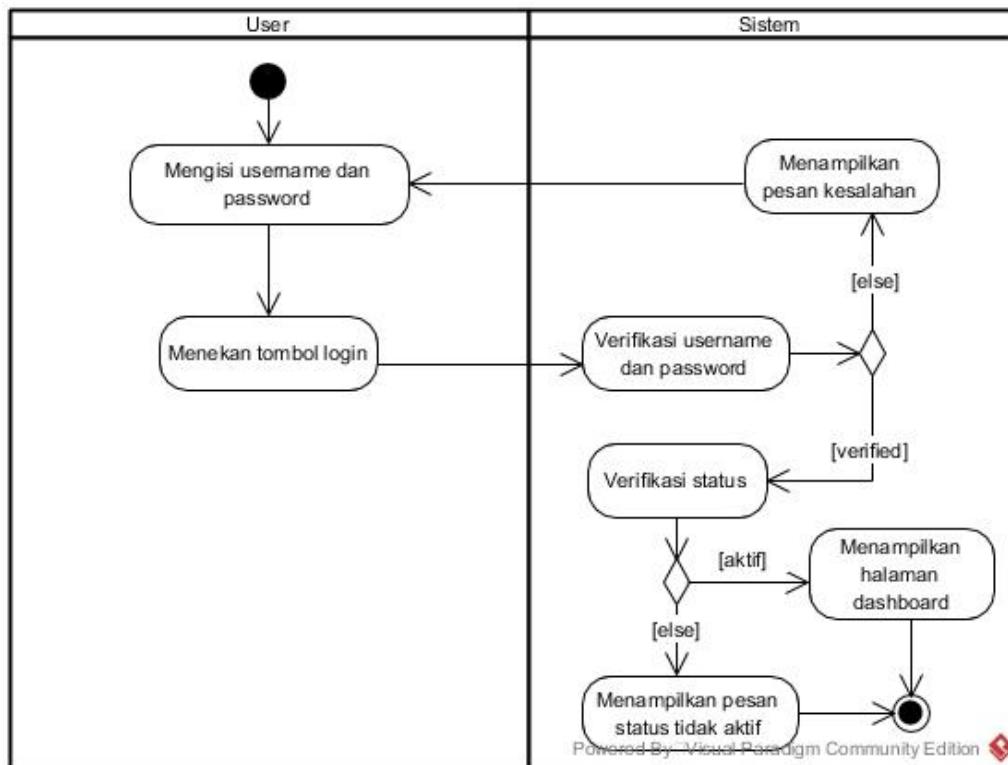
*Tabel 3. 41 Use Case Cek Info Praktikum*

<i>Use Case ID</i>	C11
<i>Use Case Name</i>	Cek Info Praktikum
<i>Description</i>	<i>Use case</i> ini memungkinkan mahasiswa untuk melihat info praktikum
<i>Primary Actor</i>	Mahasiswa
<i>Secondary Actor</i>	-
<i>Included use cases</i>	Login
<i>Extends use cases</i>	-
<i>Preconditions</i>	1. Mahasiswa harus sudah login ke dalam sistem
<i>Postconditions</i>	1. Sistem menampilkan data tugas
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Mahasiswa menekan tombol info</li> <li>2. Sistem menampilkan halaman info praktikum</li> </ol>

### 3.2.2. *Swimlane Diagram*

Berikut ini merupakan *Swimlane Diagram* dari Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika:

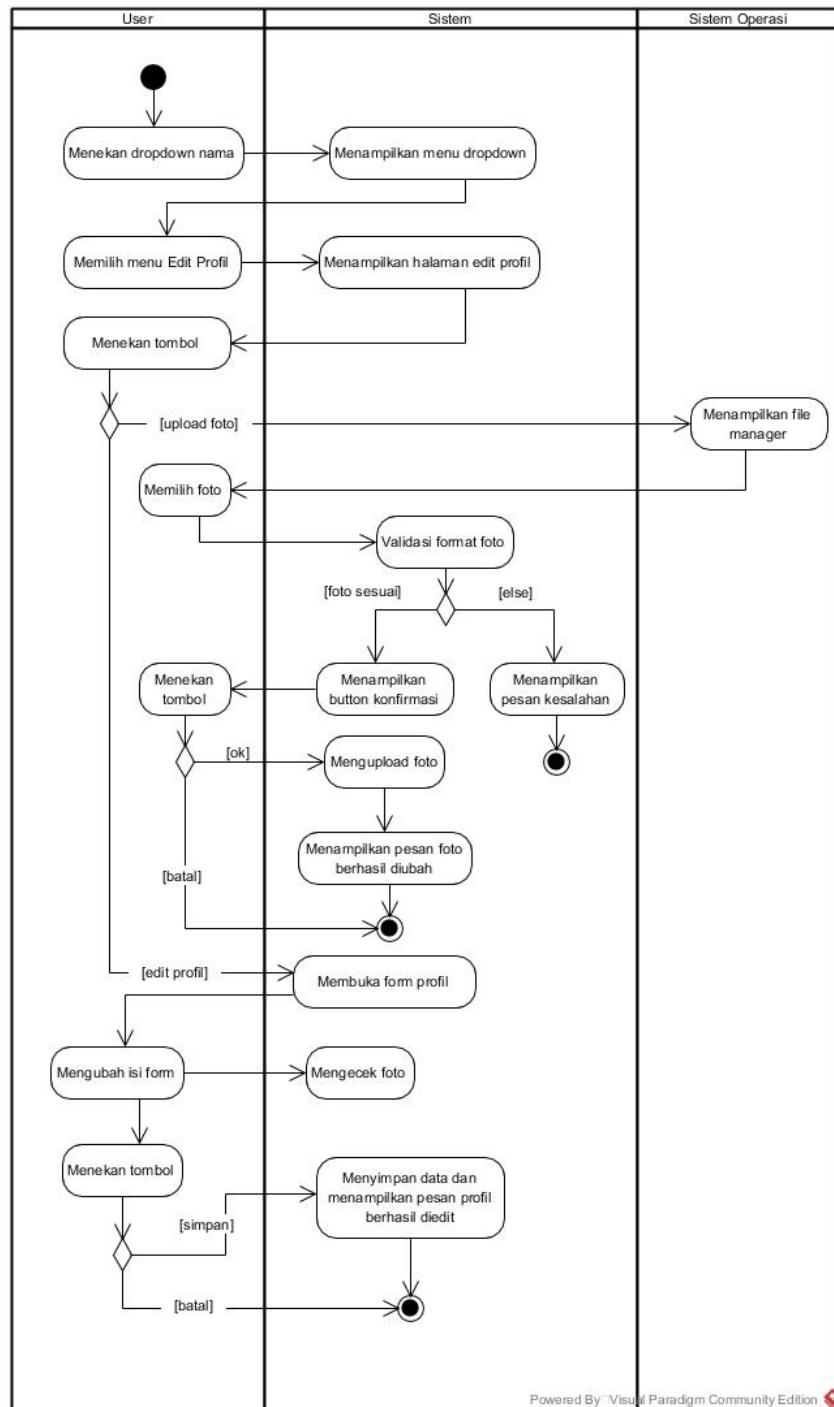
#### 1. *Swimlane Diagram Login (Server)*



Gambar 3. 3 *Swimlane Diagram Login (Server)*

Keterangan : Berdasarkan Gambar 3.3 *Swimlane Diagram Login* menunjukan langkah-langkah user untuk masuk ke aplikasi server. *User* dengan jabatan asisten *login* menggunakan *username*. *User* dengan jabatan dosen *login* menggunakan NIDN.

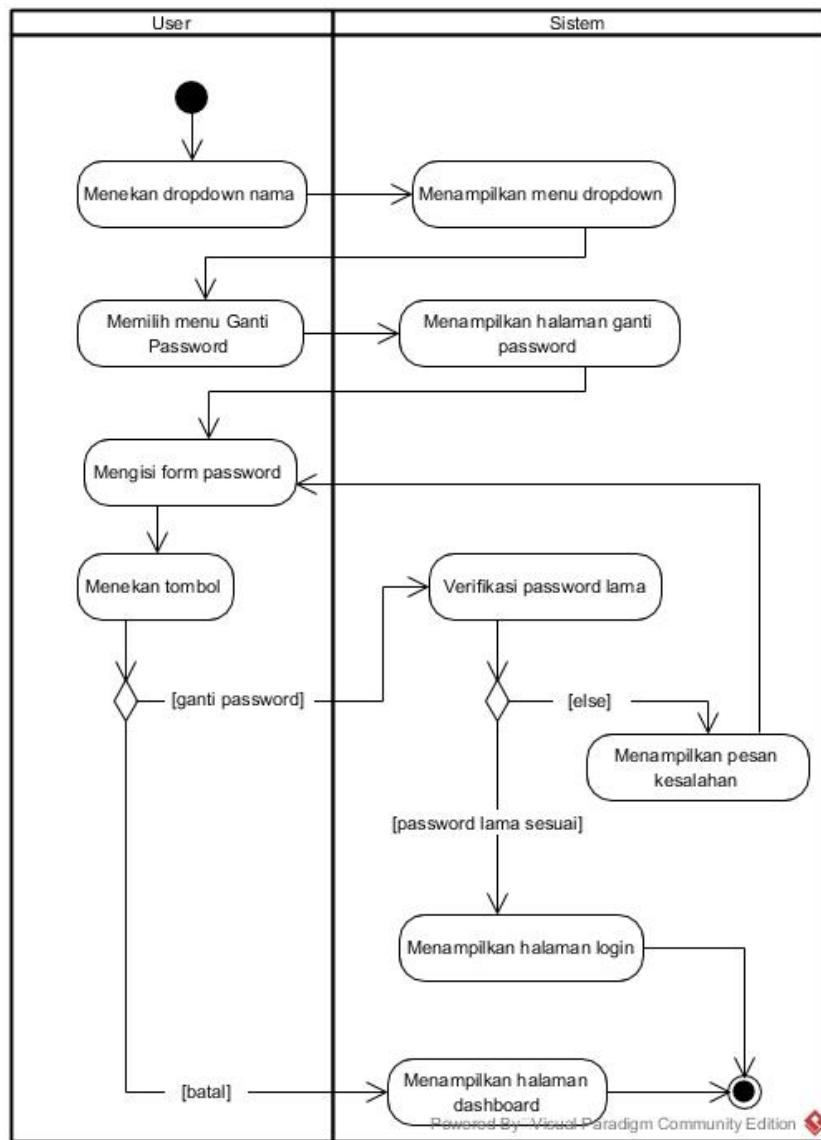
## 2. Swimlane Diagram Edit Profil



Gambar 3. 4 Swimlane Diagram Edit Profil

Keterangan : Berdasarkan Gambar 3.4 *Swimlane Diagram Edit Profil* menunjukan langkah-langkah user untuk mengubah data profil dan foto profil.

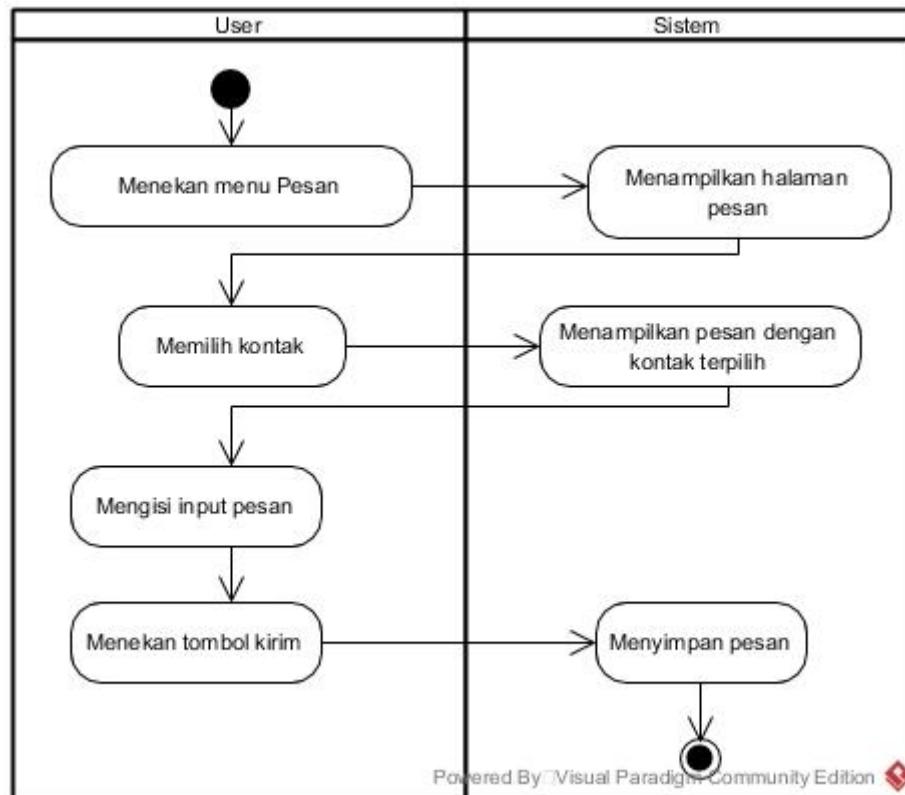
### 3. Swimlane Diagram Ganti Password (Server)



Gambar 3. 5 Swimlane Diagram Ganti Password (Server)

Keterangan : Berdasarkan Gambar 3.5 *Swimlane Diagram Ganti Password* menunjukkan langkah-langkah user untuk mengganti password akun.

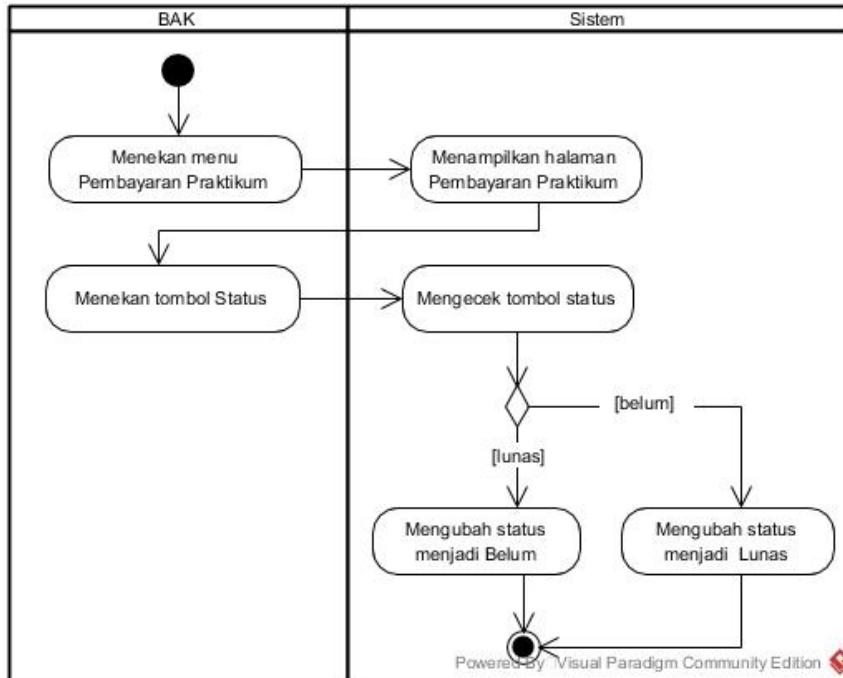
#### 4. *Swimlane Diagram Mengirim Pesan*



Gambar 3. 6 *Swimlane Diagram Mengirim Pesan*

Keterangan : Berdasarkan Gambar 3.6 *Swimlane Diagram Mengirim Pesan* menunjukan langkah-langkah user untuk mengirim pesan di dalam aplikasi.

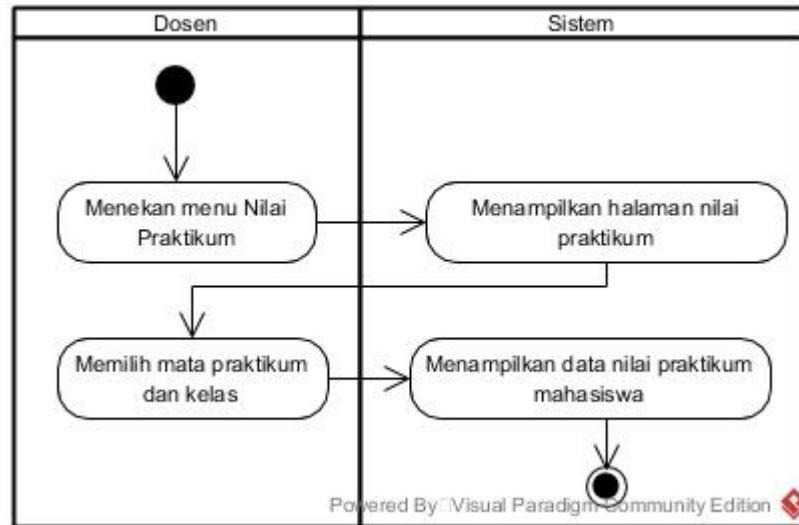
### 5. Swimlane Diagram Ubah Status Pembayaran Praktikum



Gambar 3. 7 Swimlane Diagram Ubah Status Pembayaran Praktikum

Keterangan : Berdasarkan Gambar 3.7 *Swimlane Diagram Ubah Status Pembayaran Praktikum* menunjukan langkah-langkah BAK untuk mengubah status pembayaran praktikum mahasiswa.

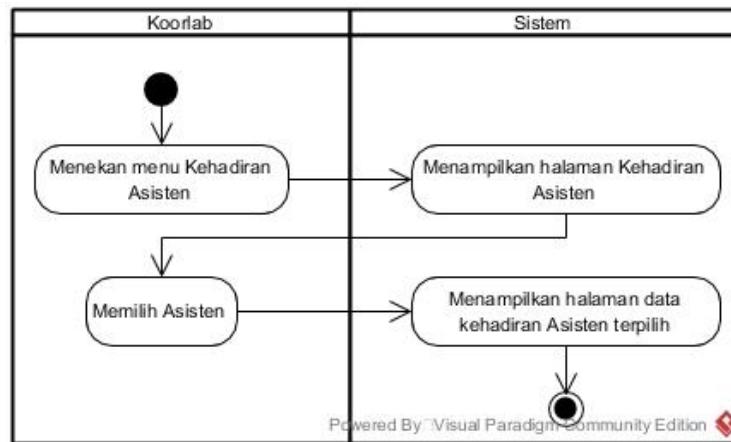
## 6. Swimlane Diagram Cek Nilai Praktikum Mahasiswa



Gambar 3. 8 Swimlane Diagram Cek Nilai Praktikum Mahasiswa

Keterangan : Berdasarkan Gambar 3.8 *Swimlane Diagram Cek Nilai Praktikum Mahasiswa* menunjukan langkah-langkah dosen untuk melihat nilai praktikum mahasiswa.

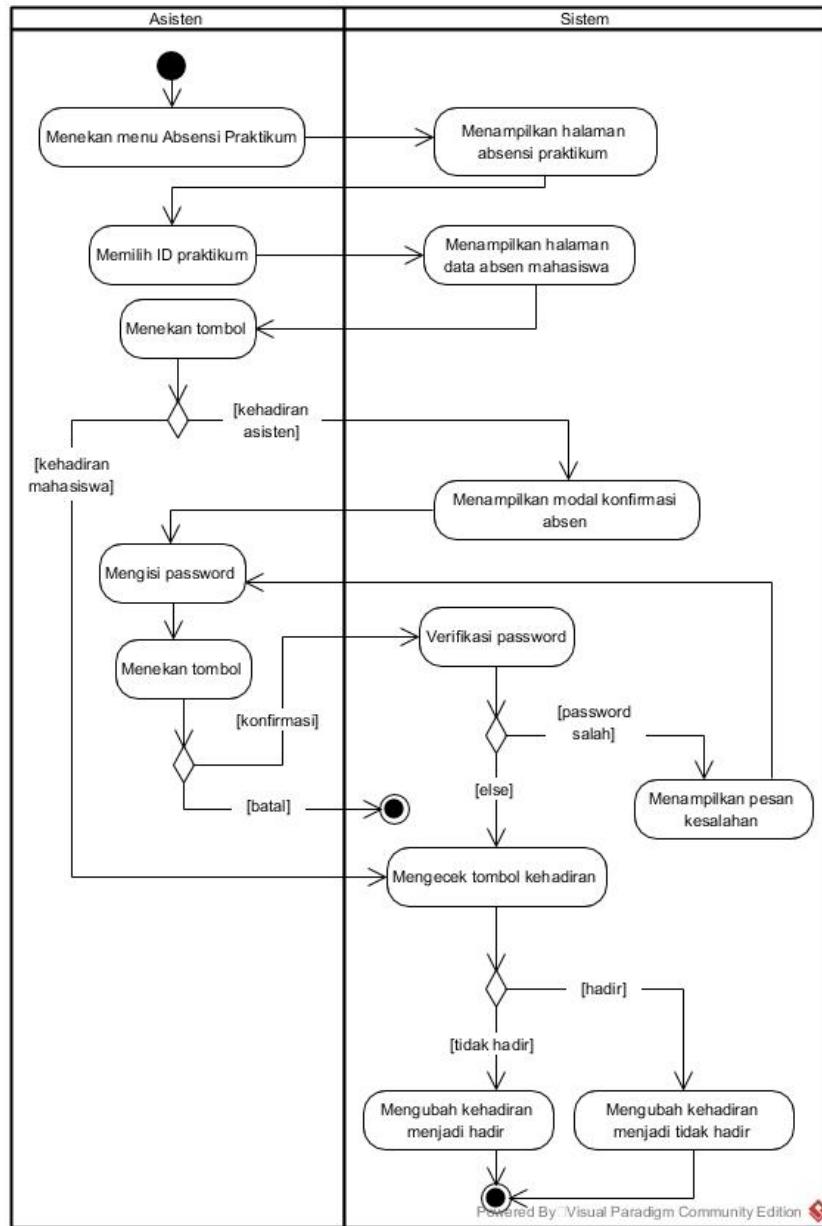
## 7. Swimlane Diagram Cek Kehadiran Asisten



Gambar 3. 9 Swimlane Diagram Cek Kehadiran Asisten

Keterangan : Berdasarkan Gambar 3.9 *Swimlane Diagram Cek Kehadiran Asisten* menunjukan langkah-langkah dosen untuk melihat data kehadiran praktikum asisten.

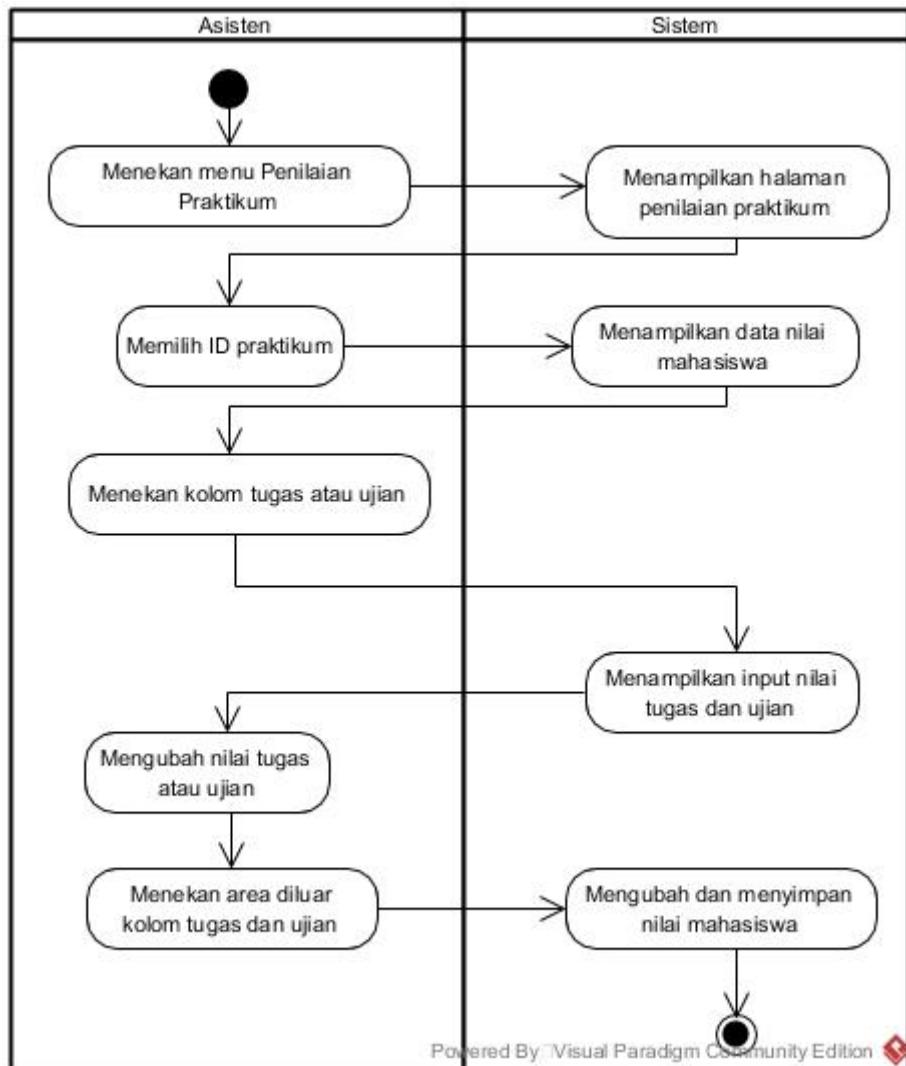
### 8. Swimlane Diagram Absensi Praktikum



Gambar 3. 10 Swimlane Diagram Absensi Praktikum

Keterangan : Berdasarkan Gambar 3.10 *Swimlane Diagram Absensi Praktikum* menunjukan langkah-langkah asisten untuk melakukan absensi praktikum.

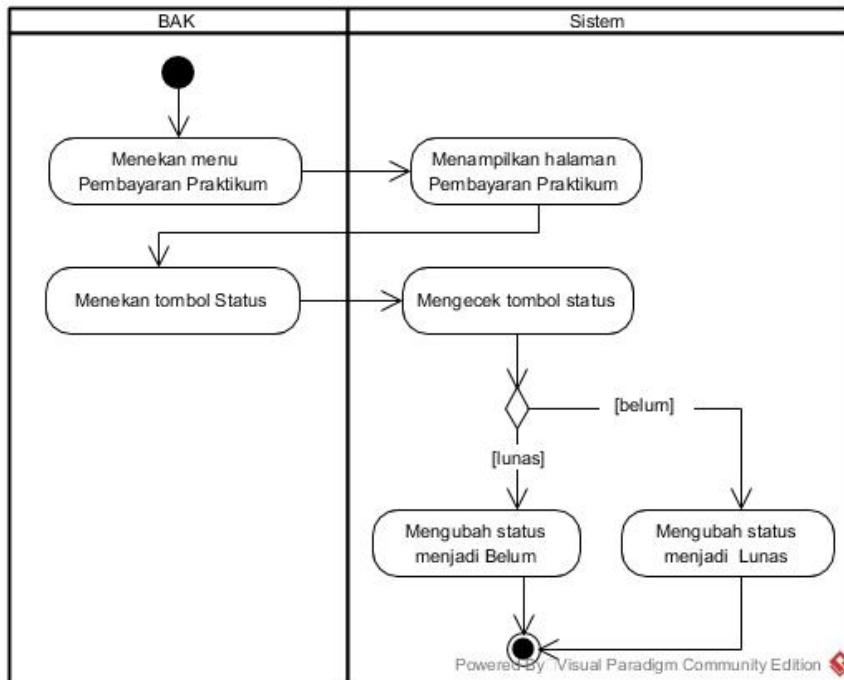
### 9. Swimlane Diagram Penilaian Praktikum



Gambar 3. 11 *Swimlane Diagram Penilaian Praktikum*

Keterangan : Berdasarkan Gambar 3.11 *Swimlane Diagram Penilaian Praktikum* menunjukan langkah-langkah asisten untuk melakukan penilaian praktikum.

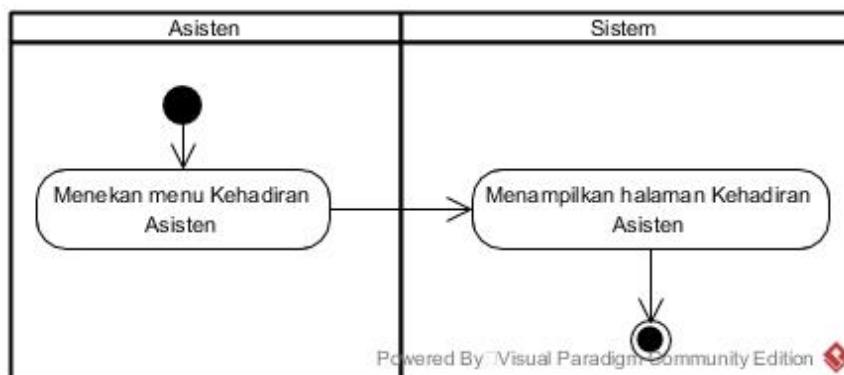
### 10. Swimlane Diagram Ubah Status Pengambilan Modul Praktikum



*Gambar 3. 12 Swimlane Diagram Ubah Status Pengambilan Modul Praktikum*

Keterangan : Berdasarkan Gambar 3.12 *Swimlane Diagram Ubah Status Pengambilan Modul Praktikum* menunjukan langkah-langkah asisten untuk mengubah status pengambilan modul praktikum mahasiswa

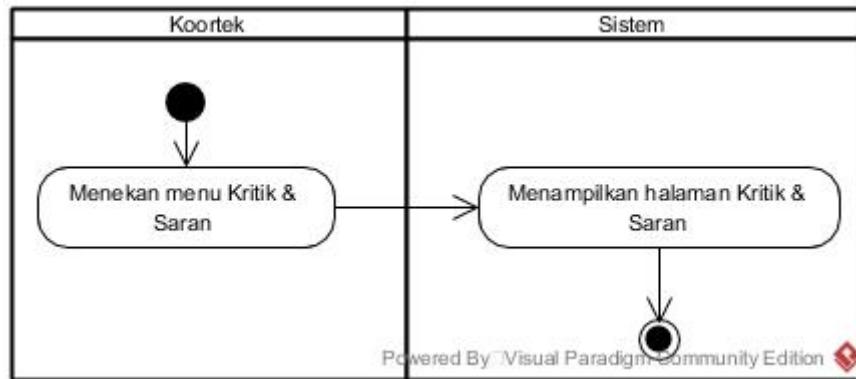
### 11. Swimlane Diagram Cek Kehadiran



*Gambar 3. 13 Swimlane Diagram Cek Kehadiran*

Keterangan : Berdasarkan Gambar 3.13 *Swimlane Diagram Cek Kehadiran* menunjukan langkah-langkah asisten untuk melihat data kehadiran praktikum masing-masing.

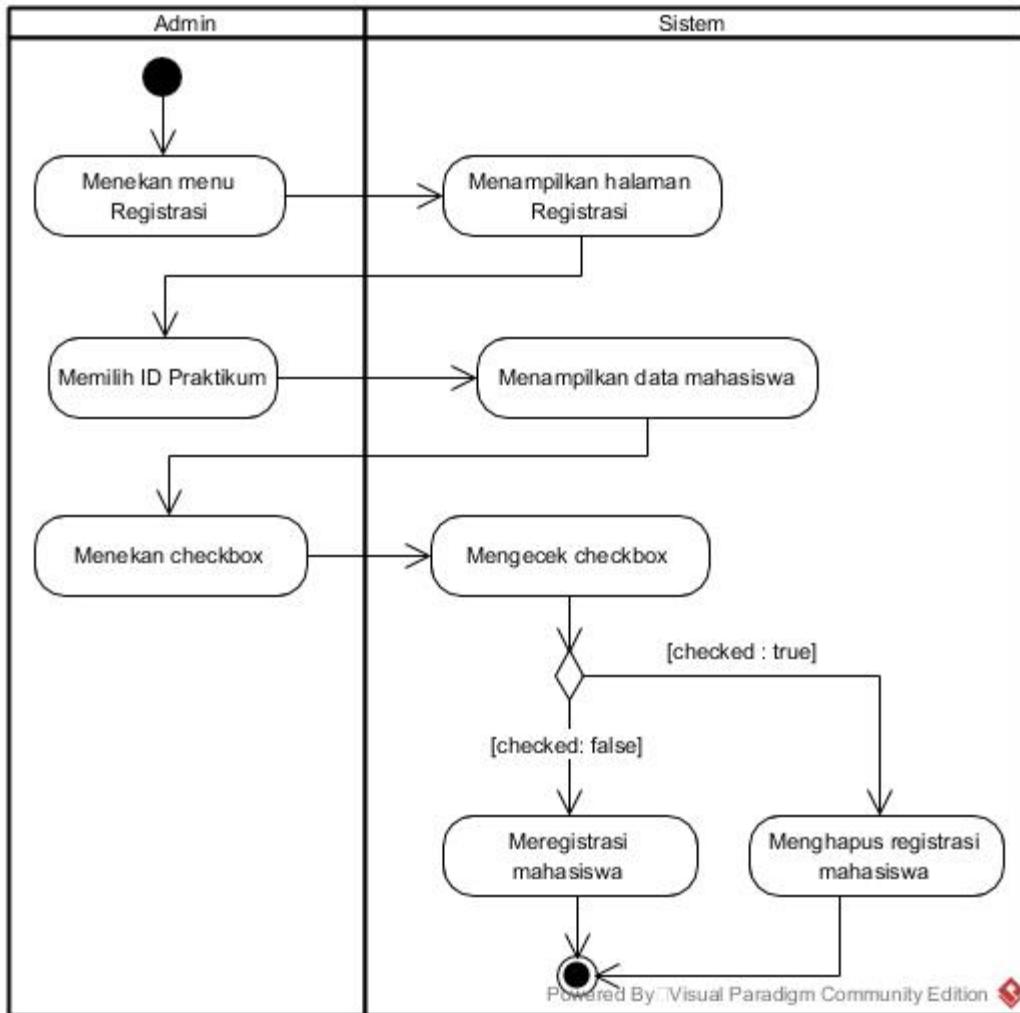
#### 12. *Swimlane Diagram Lihat Kritik & Saran*



Gambar 3. 14 *Swimlane Diagram Lihat Kritik & Saran*

Keterangan : Berdasarkan Gambar 3.14 *Swimlane Diagram Lihat Kritik & Saran* menunjukan langkah-langkah koortek untuk melihat data kritik & saran yang dikirim oleh mahasiswa.

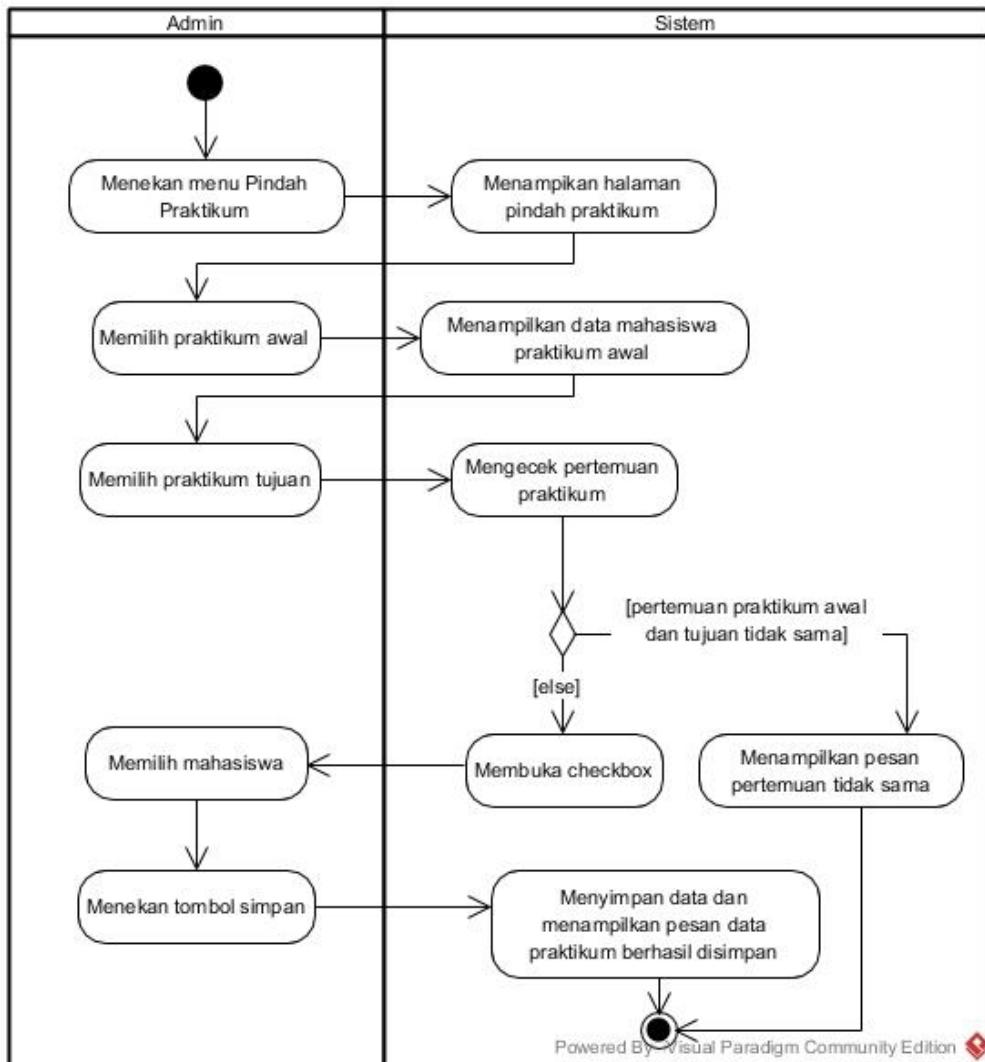
### 13. Swimlane Diagram Registrasi Mahasiswa



Gambar 3. 15 Swimlane Diagram Registrasi Mahasiswa

Keterangan : Berdasarkan Gambar 3.15 *Swimlane Diagram Registrasi Mahasiswa* menunjukan langkah-langkah admin meregistrasi mahasiswa ke dalam jadwal praktikum.

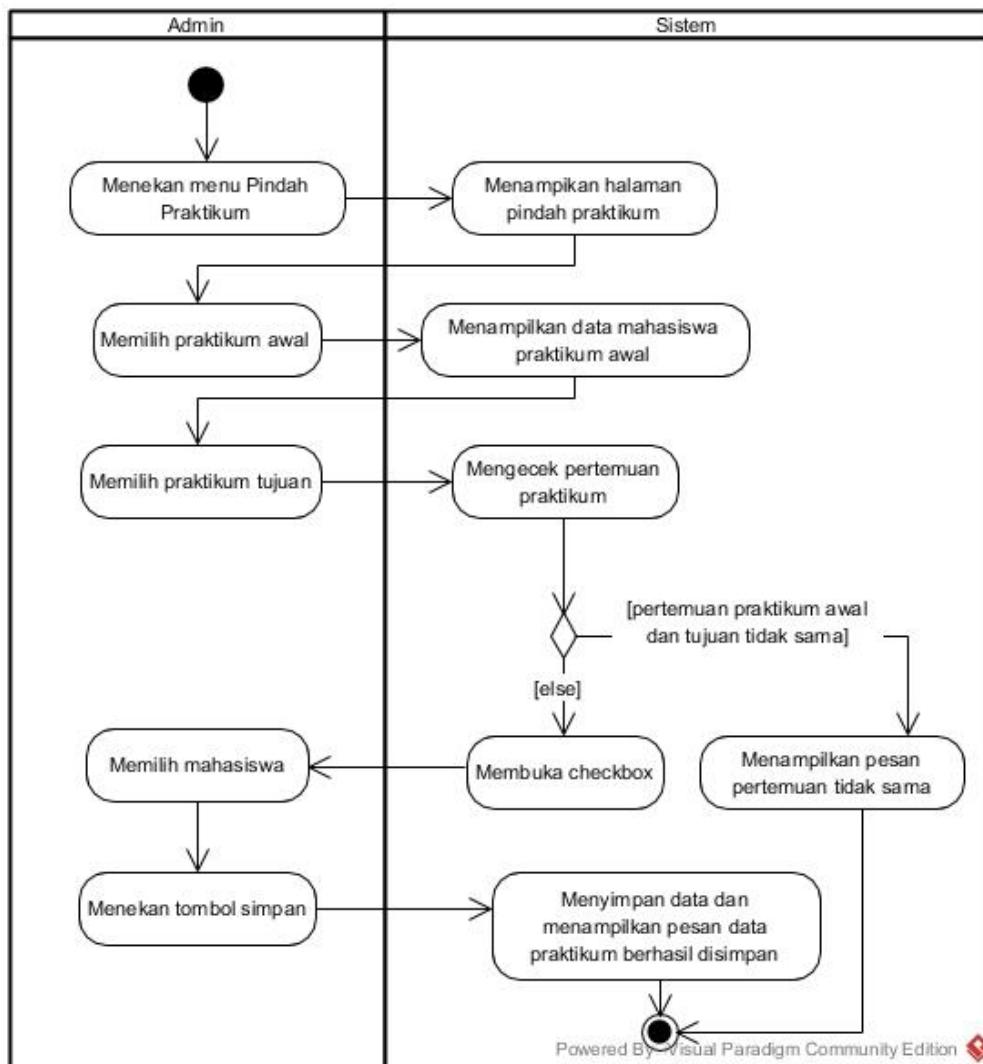
#### 14. Swimlane Diagram Pindah Praktikum Mahasiswa



Gambar 3. 16 Swinlane Diagram Pindah Praktikum Mahasiswa

Keterangan : Berdasarkan Gambar 3.16 *Swimlane Diagram Pindah Praktikum Mahasiswa* menunjukan langkah-langkah admin dalam memproses pemindahan praktikum mahasiswa dari praktikum asal ke praktikum tujuan.

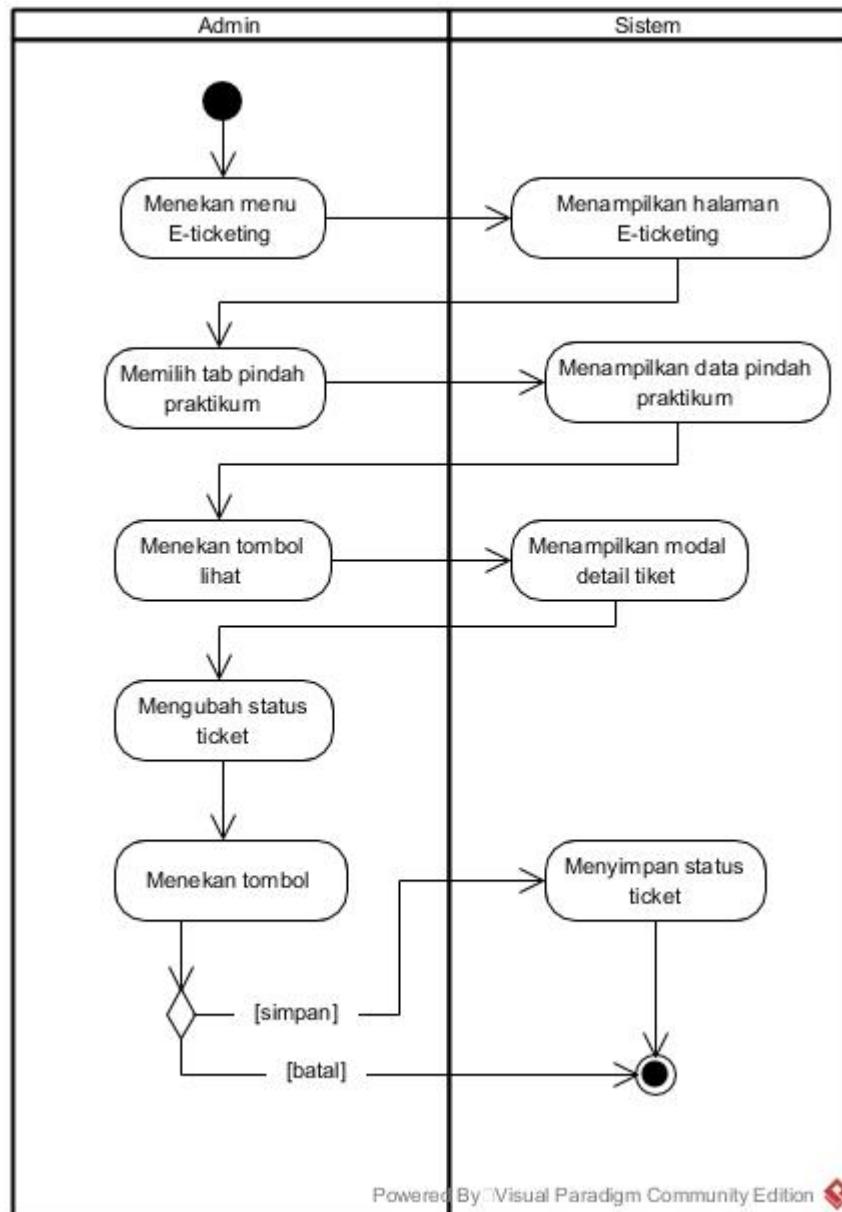
### 15. Swimlane Diagram Mengirim Notifikasi



Gambar 3. 17 Swimlane Diagram Mengirim Notifikasi

Keterangan : Berdasarkan Gambar 3.17 *Swimlane Diagram Mengirim Notifikasi* menunjukan langkah-langkah admin untuk mengirim notifikasi info praktikum.

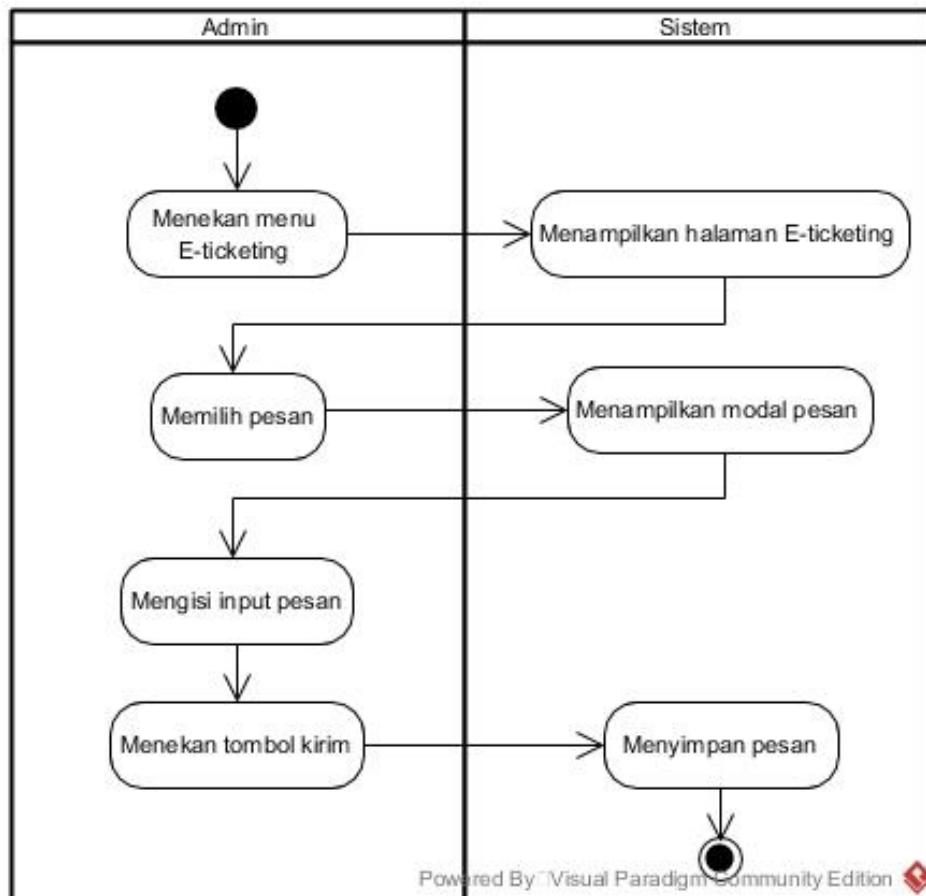
16. *Swimlane Diagram Ubah Status Tiket Pindah Praktikum*



Gambar 3. 18 *Swimlane Diagram Ubah Status Tiket Pindah Praktikum*

Keterangan : Berdasarkan Gambar 3.18 *Swimlane Diagram Ubah Status Tiket Pindah Praktikum* menunjukkan langkah-langkah admin untuk mengubah status tiket permintaan pindah praktikum.

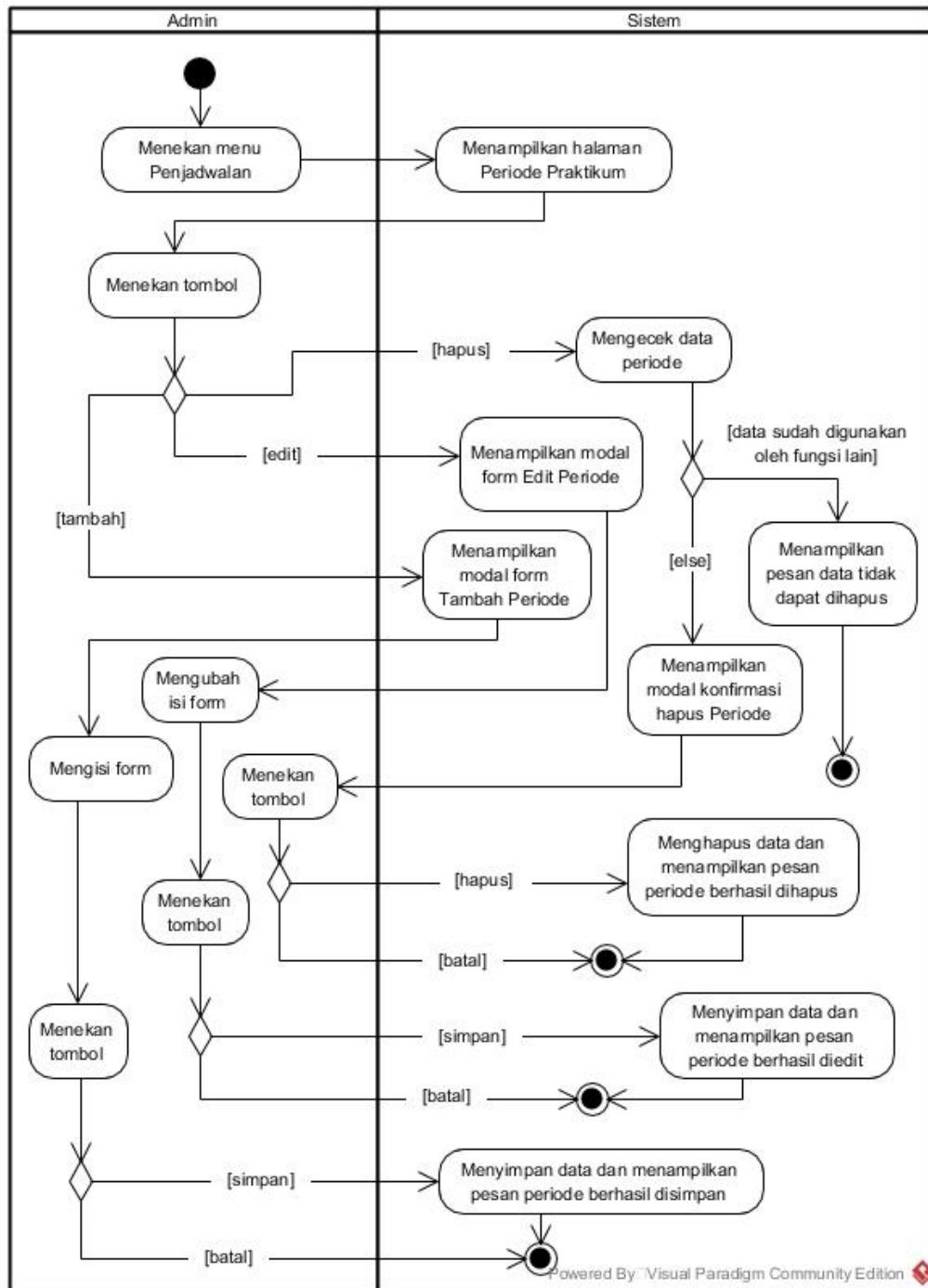
17. *Swimlane Diagram Balas Pesan Mahasiswa*



*Gambar 3. 19 Swimlane Diagram Balas Pesan Mahasiswa*

Keterangan : Berdasarkan Gambar 3.19 *Swimlane Diagram Balas Pesan Mahasiswa* menunjukan langkah-langkah admin dalam membalas pesan atau pertanyaan yang dikirim oleh mahasiswa.

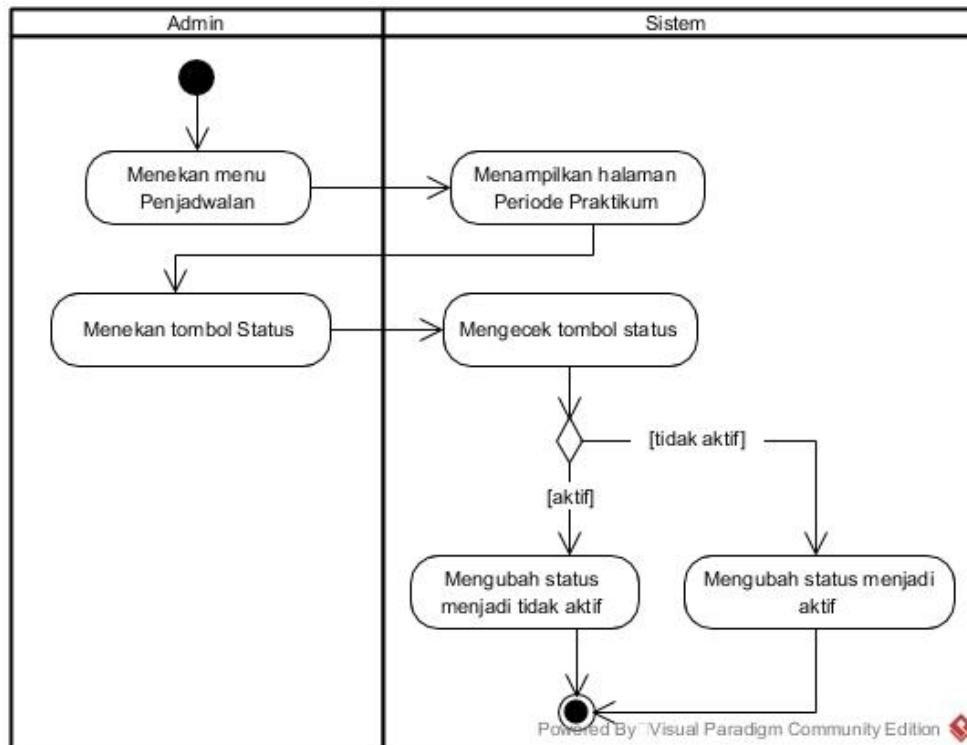
### 18. Swimlane Diagram Kelola Periode



Gambar 3. 20 Swinlane Diagram Kelola Periode

Keterangan : Berdasarkan Gambar 3.20 Swinlane Diagram Kelola Periode menunjukan langkah-langkah admin dalam mengelola data periode yaitu menambah, mengubah, dan menghapus periode praktikum.

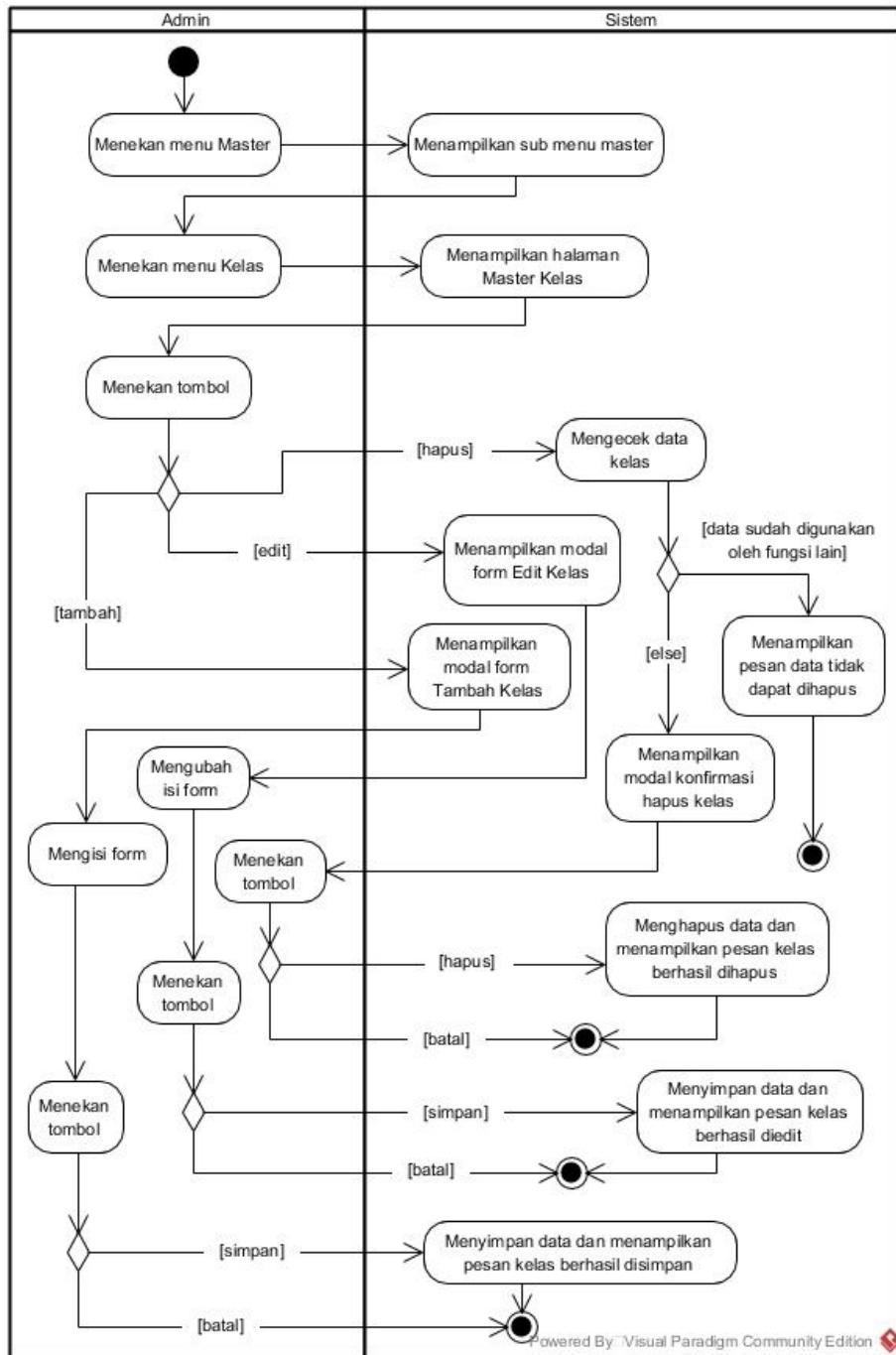
19. *Swimlane Diagram Ubah Status Periode*



Gambar 3. 21 *Swimlane Diagram Ubah Status Kelas*

Keterangan : Berdasarkan Gambar 3.21 *Swimlane Diagram Ubah Status Kelas* menunjukan langkah-langkah admin untuk mengubah status periode menjadi aktif atau tidak aktif.

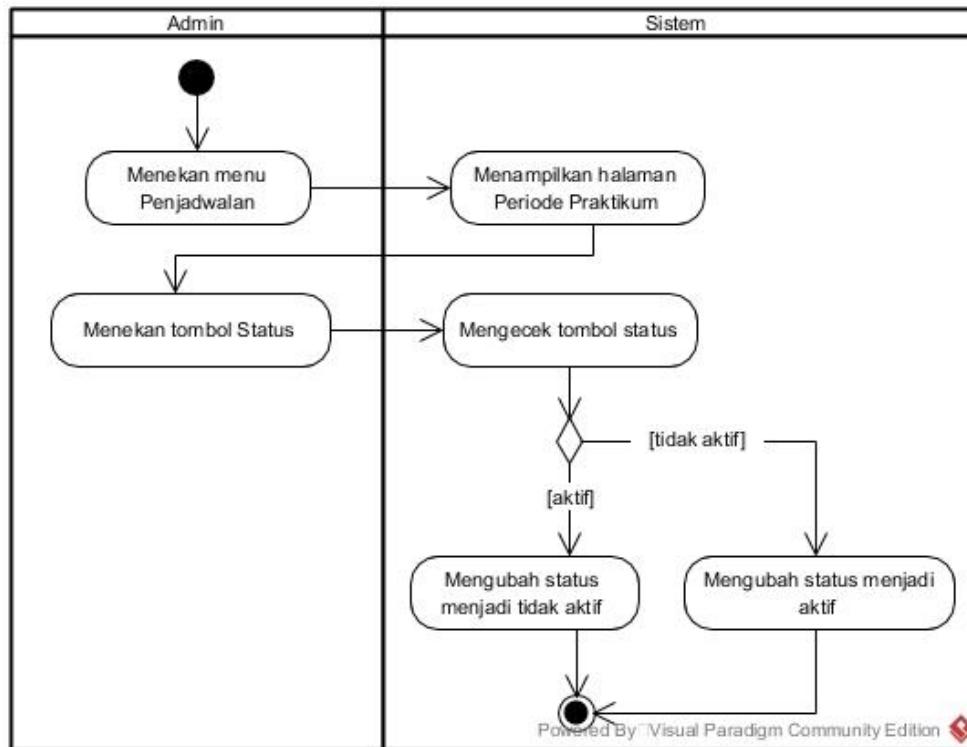
## 20. Swimlane Diagram Kelola Kelas



Gambar 3. 22 Swinlane Diagram Kelola Kelas

Keterangan : Berdasarkan Gambar 3.22 Swinlane Diagram Kelola Kelas menunjukan langkah-langkah admin dalam mengelola data kelas yaitu menambah, mengubah, dan menghapus kelas.

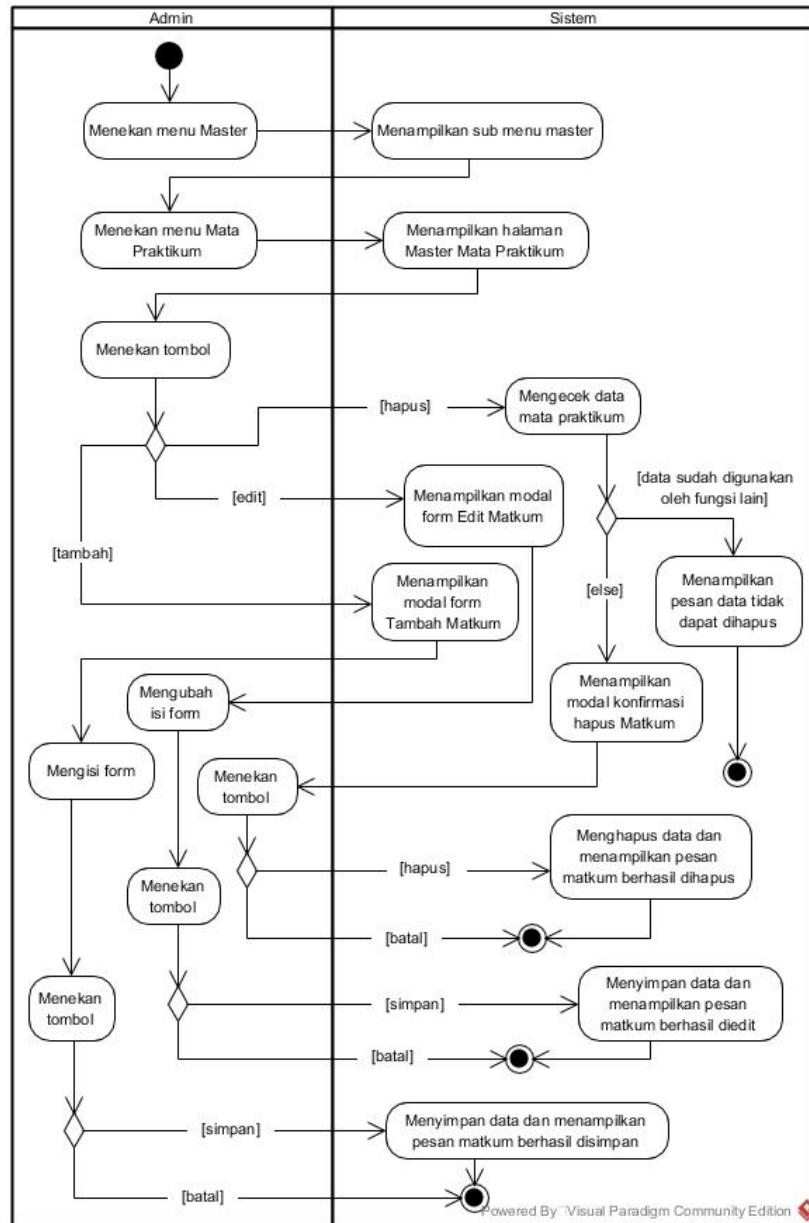
### 21. Swimlane Diagram Ubah Status Kelas



Gambar 3. 23 Swimlane Diagram Ubah Status Kelas

Keterangan : Berdasarkan Gambar 3.24 *Swimlane Diagram Ubah Status Kelas* menunjukan langkah-langkah admin untuk mengubah status periode menjadi aktif atau tidak aktif.

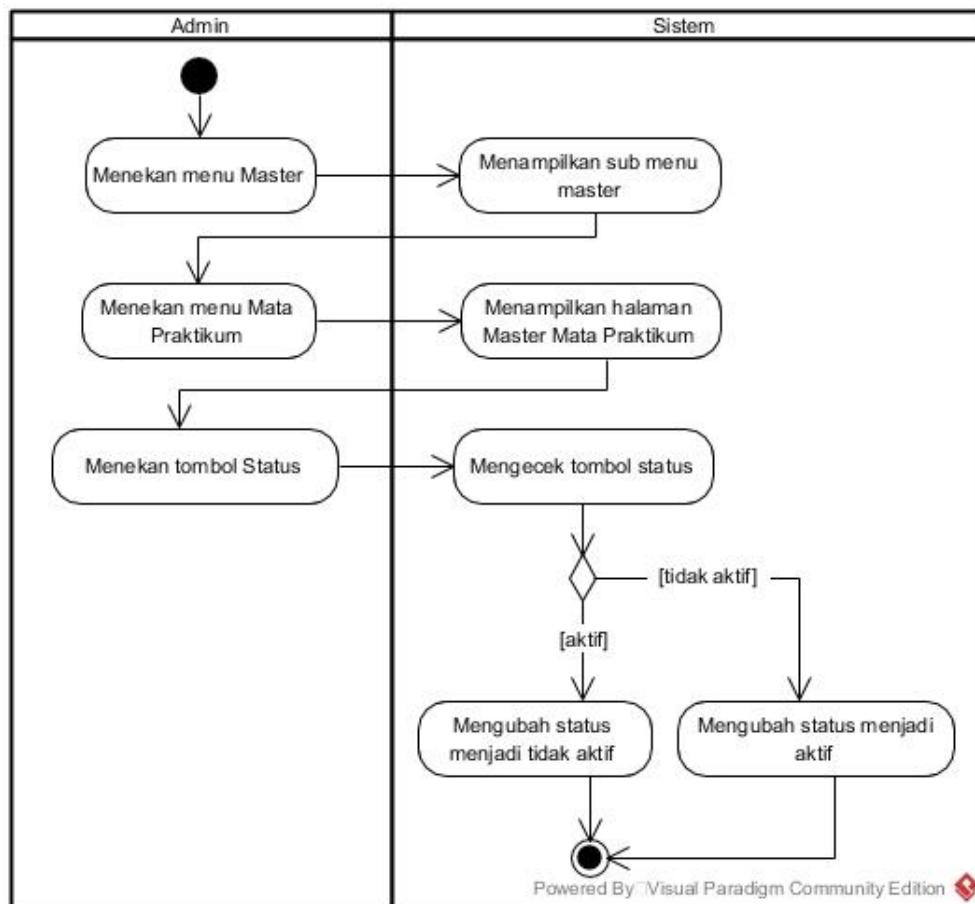
## 22. Swimlane Diagram Kelola Mata Praktikum



Gambar 3. 24 Swimlane Diagram Kelola Mata Praktikum

Keterangan : Berdasarkan Gambar 3.24 Swimlane Diagram Kelola Mata Praktikum menunjukkan langkah-langkah admin dalam mengelola data mata praktikum yaitu menambah, mengubah, dan menghapus mata praktikum.

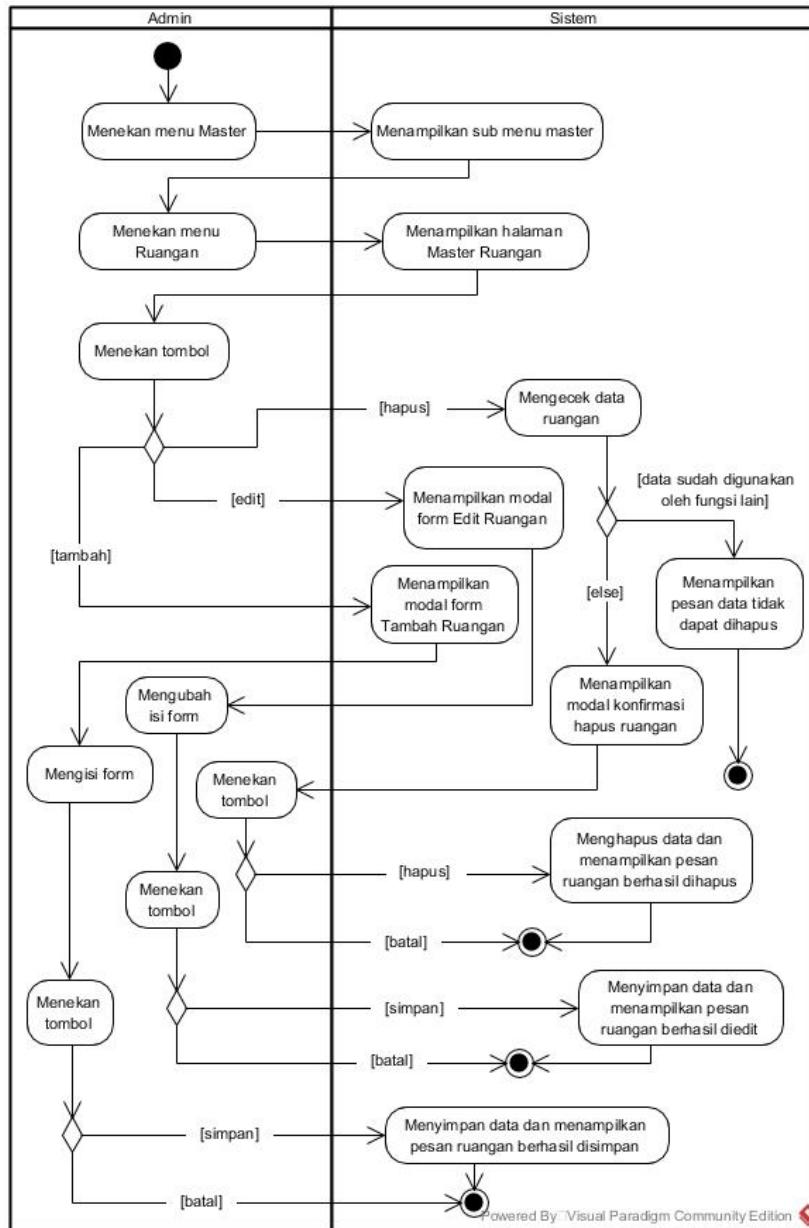
23. *Swimlane Diagram Ubah Status Mata Praktikum*



Gambar 3. 25 *Swimlane Diagram Ubah Status Mata Praktikum*

Keterangan : Berdasarkan Gambar 3.25 *Swimlane Diagram Ubah Status Mata Praktikum* menunjukan langkah-langkah admin untuk mengubah status mata praktikum menjadi aktif atau tidak aktif.

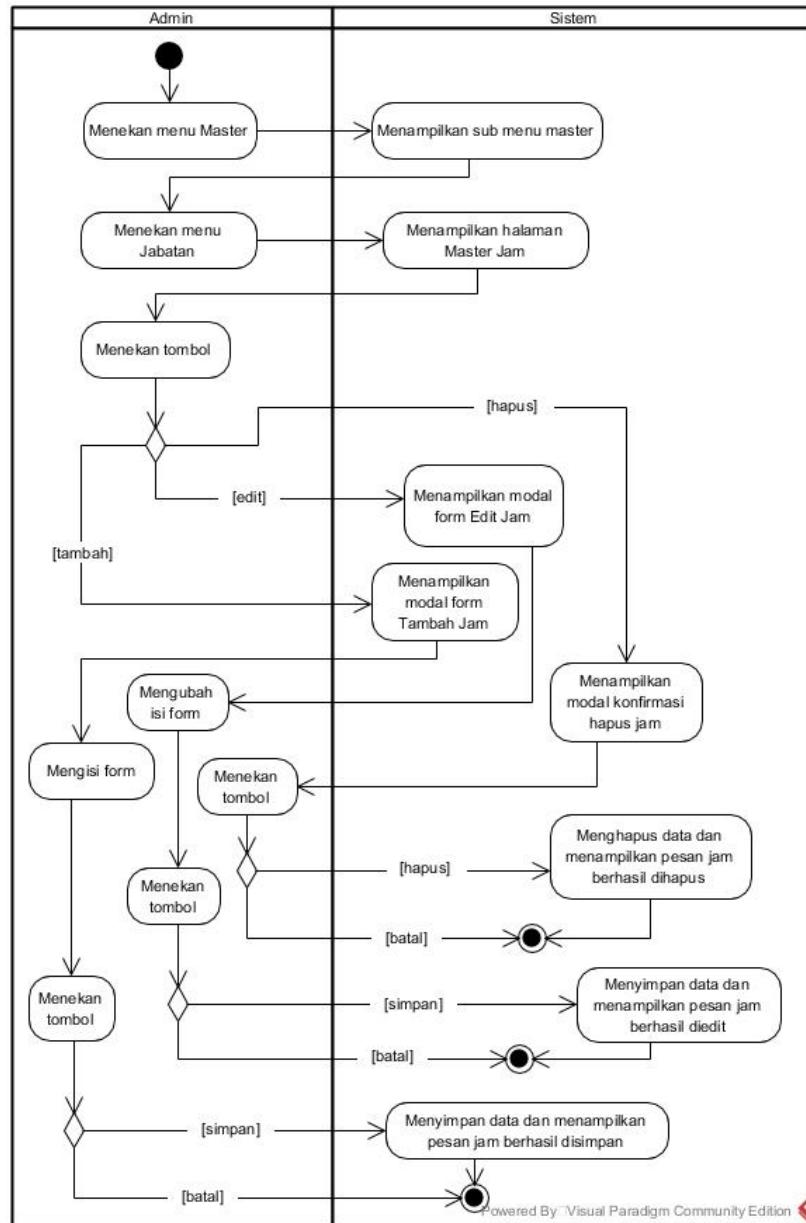
## 24. Swimlane Diagram Kelola Ruangan



Gambar 3. 26 Swimlane Diagram Kelola Ruangan

Keterangan : Berdasarkan Gambar 3.26 *Swimlane Diagram Kelola Ruangan* menunjukan langkah-langkah admin dalam mengelola data ruangan yaitu menambah, mengubah, dan menghapus ruangan.

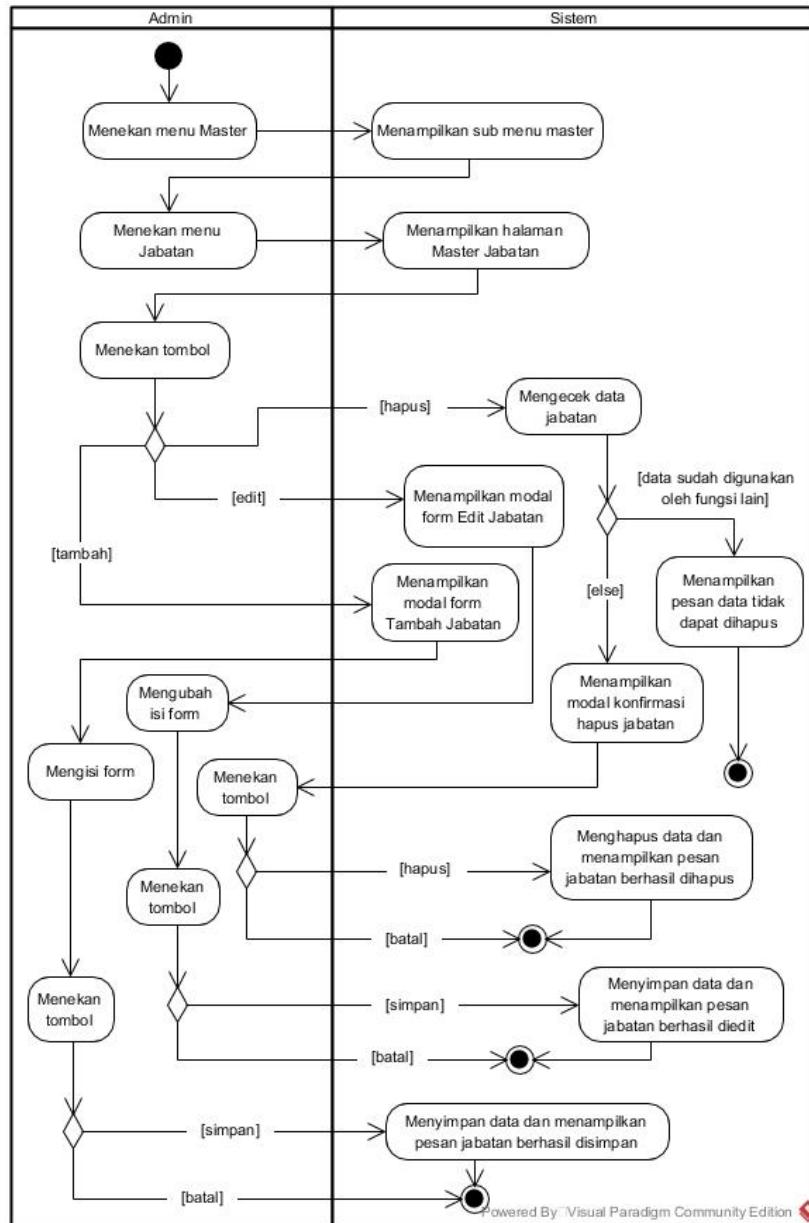
### 25. Swimlane Diagram Kelola Jam



Gambar 3. 27 Swinlane Diagram Kelola Jam

Keterangan : Berdasarkan Gambar 3.27 Swinlane Diagram Kelola Jam menunjukan langkah-langkah admin dalam mengelola data jam yaitu menambah, mengubah, dan menghapus jam praktikum.

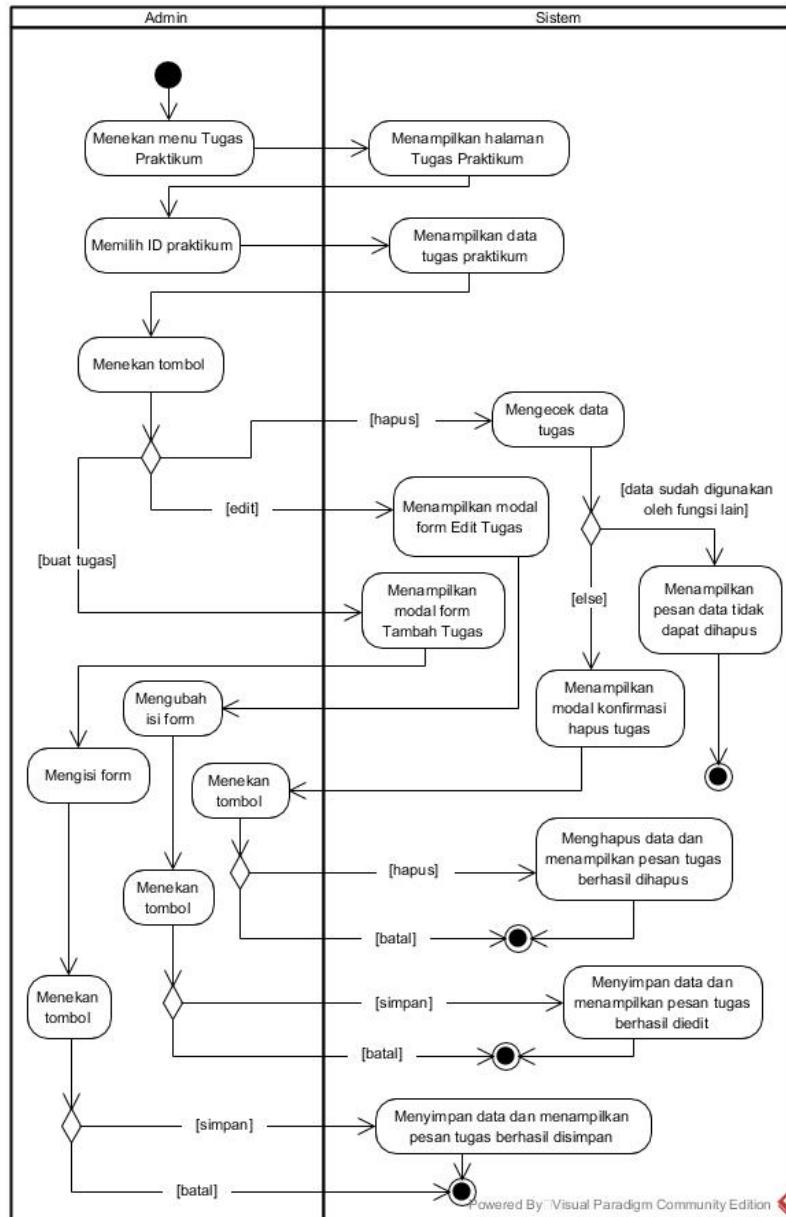
## 26. Swimlane Diagram Kelola Jabatan



Gambar 3. 28 Swimlane Diagram Kelola Jabatan

Keterangan : Berdasarkan Gambar 3.28 *Swimlane Diagram Kelola Jabatan* menunjukkan langkah-langkah admin dalam mengelola data jabatan yaitu menambah, mengubah, dan menghapus jabatan.

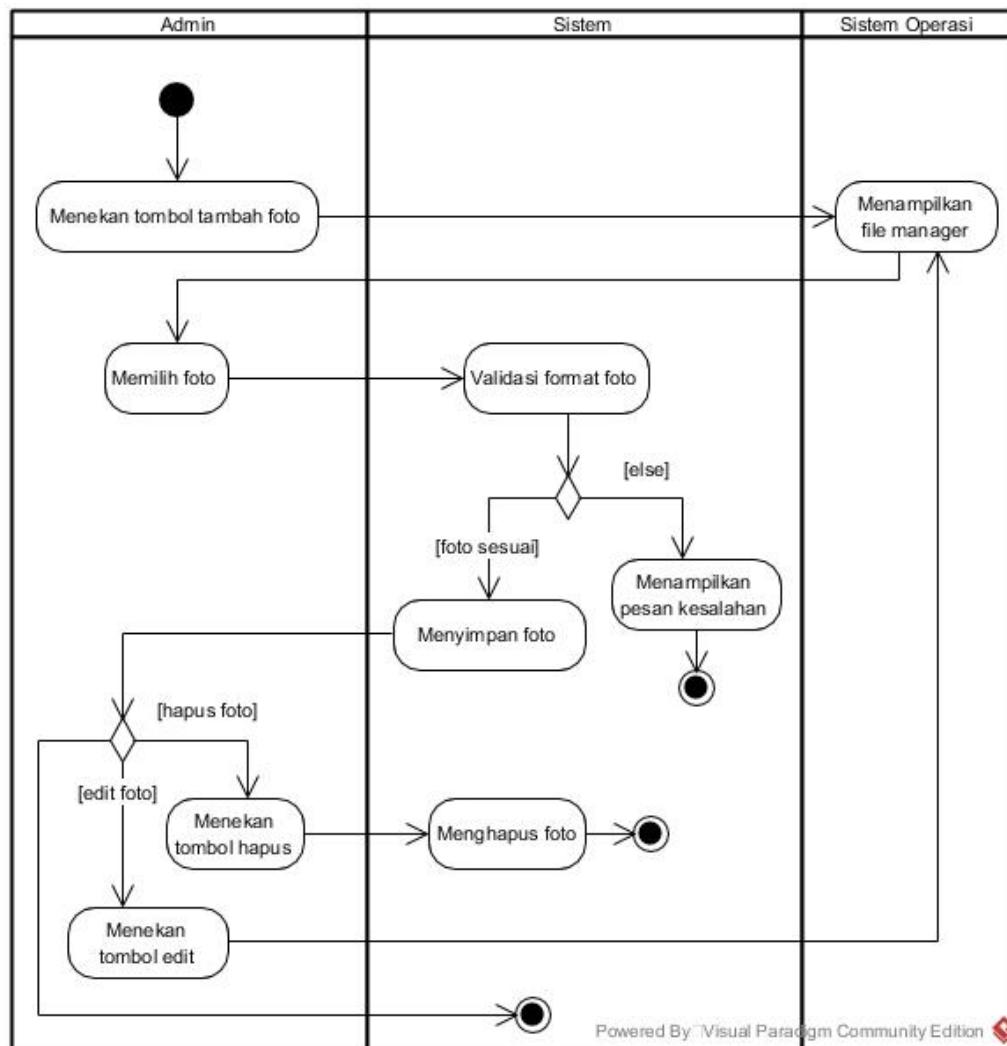
### 27. Swimlane Diagram Kelola Tugas



Gambar 3. 29 Swinlane Diagram Kelola Tugas

Keterangan : Berdasarkan Gambar 3.29 *Swimlane Diagram Kelola Tugas* menunjukan langkah-langkah asisten dalam mengelola data tugas yaitu menambah, mengubah, dan menghapus tugas praktikum.

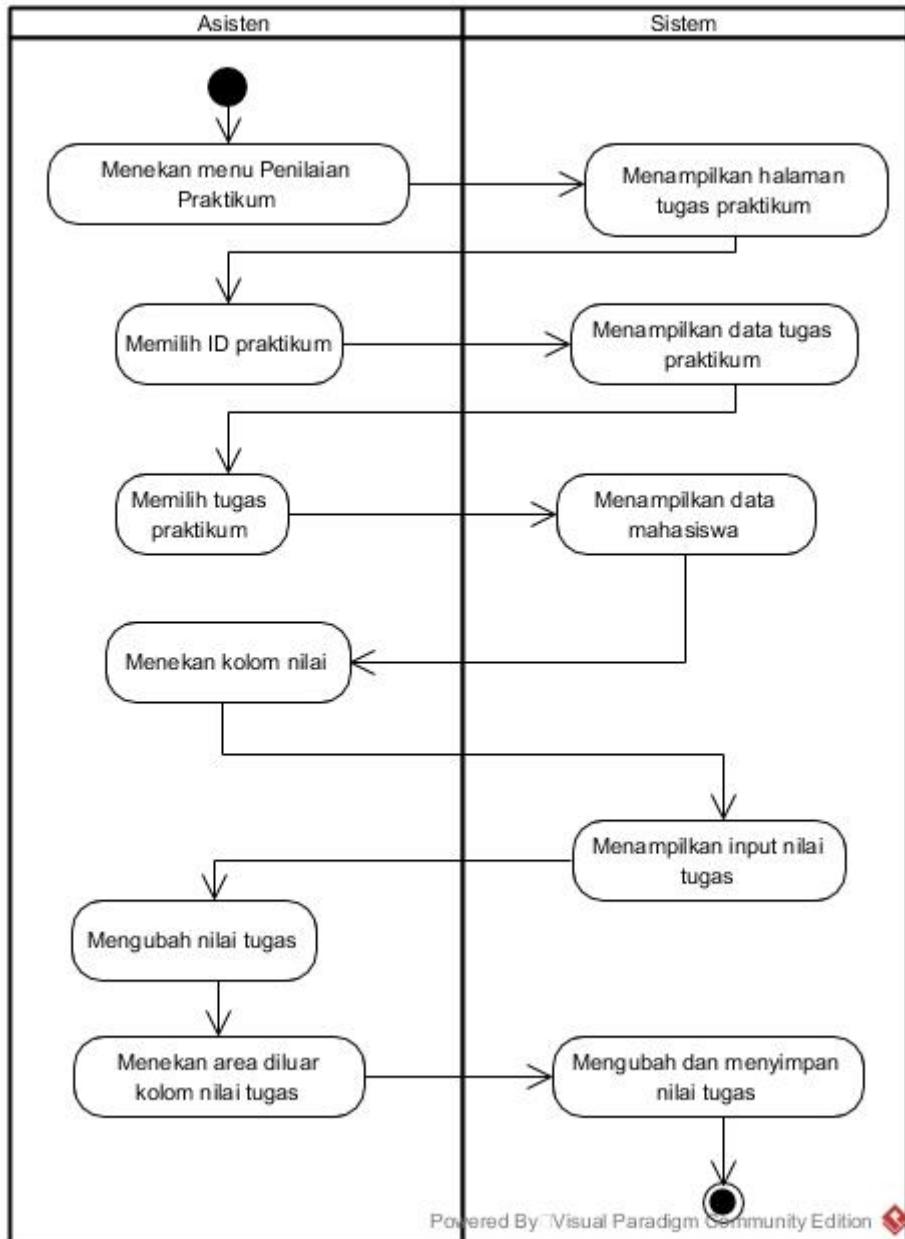
28. *Swimlane Diagram Kelola Foto Tugas*



Gambar 3. 30 *Swimlane Diagram Kelola Foto Tugas*

Keterangan : Berdasarkan Gambar 3.30 *Swimlane Diagram Kelola Foto Tugas* menunjukan langkah-langkah asisten dalam mengelola foto tugas yaitu menambah, mengubah, dan menghapus foto tugas praktikum. Kelola foto tugas dilakukan di dalam modal tambah atau edit tugas.

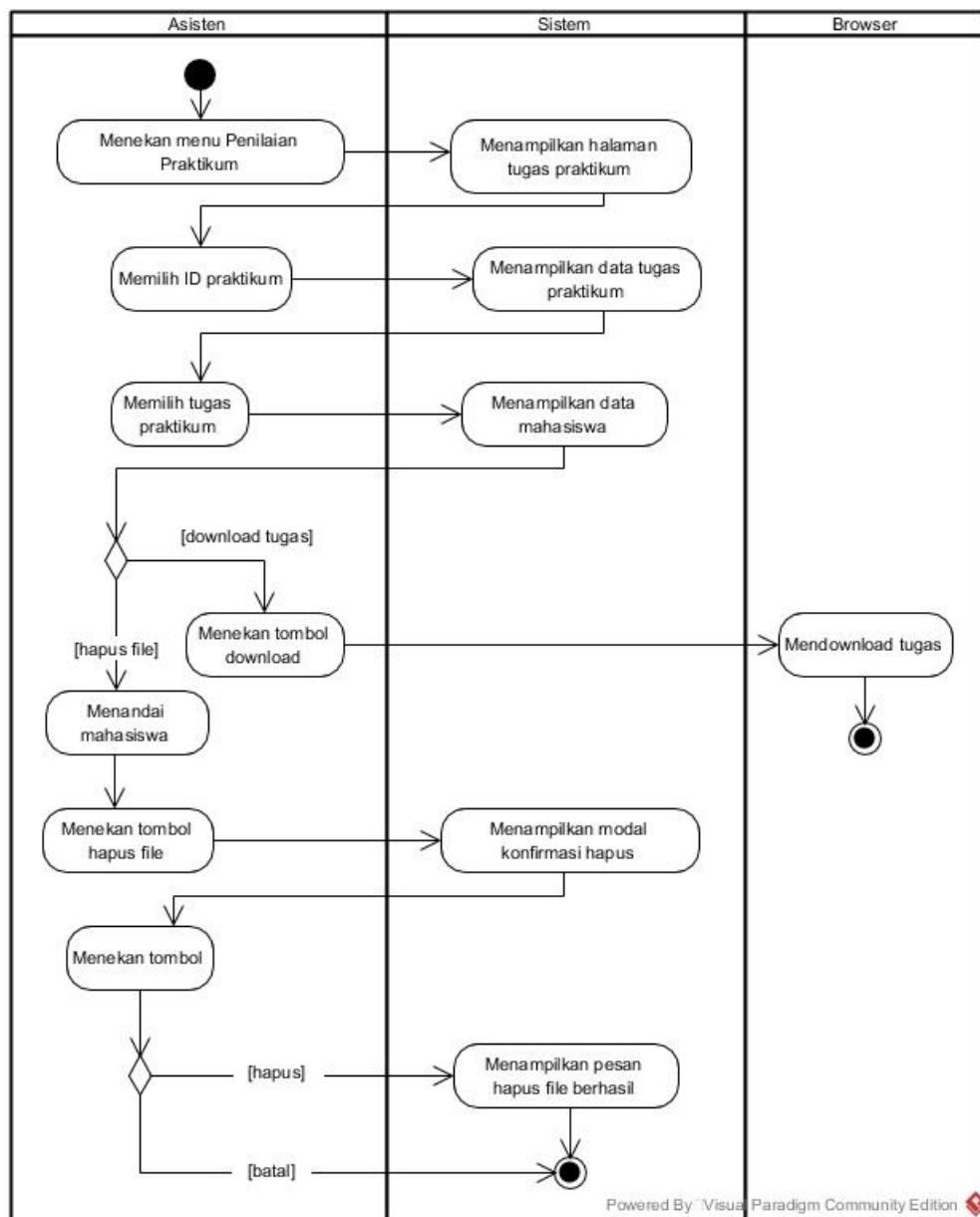
29. *Swimlane Diagram Penilaian Tugas*



Gambar 3. 31 *Swimlane Diagram Penilaian Tugas*

Keterangan : Berdasarkan Gambar 3.31 *Swimlane Diagram Penilaian Tugas* menunjukkan langkah-langkah asisten dalam menilai tugas praktikum mahasiswa.

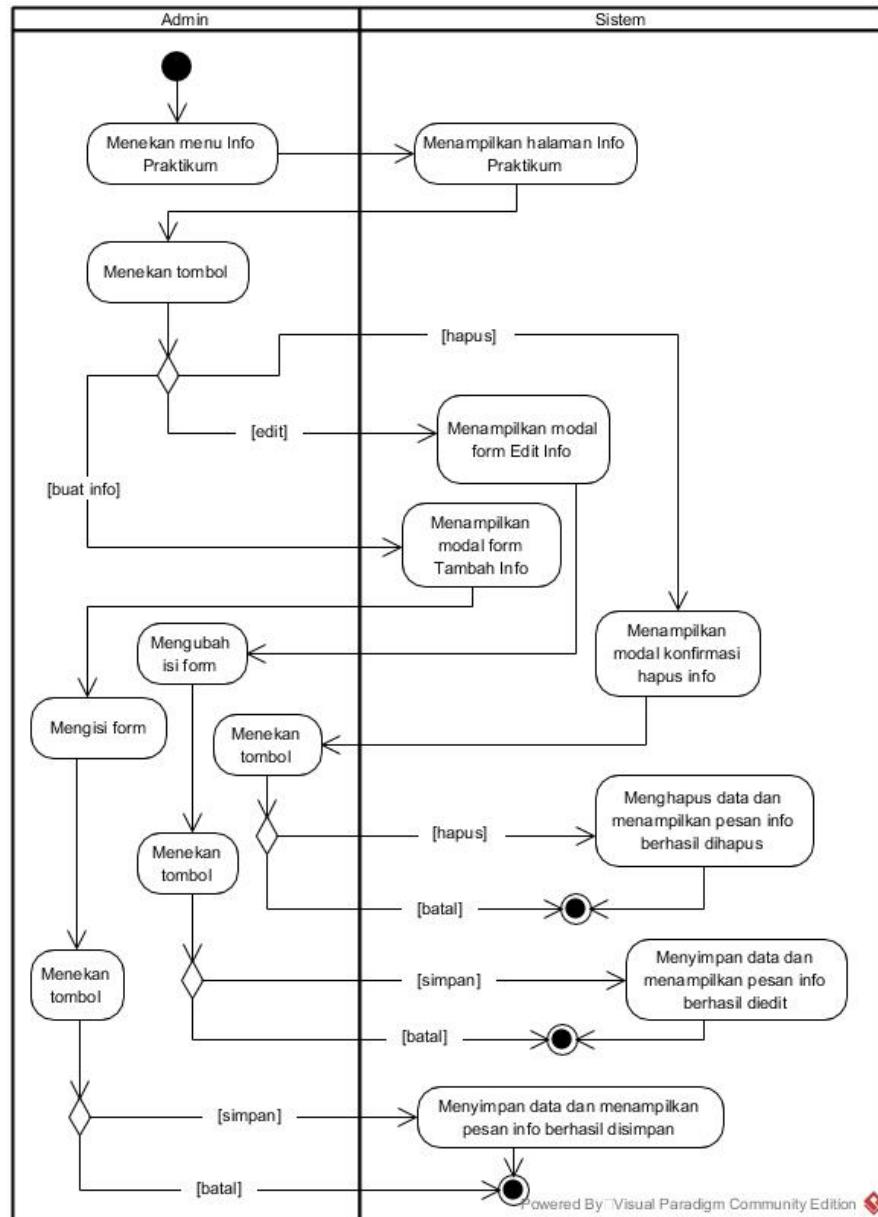
### 30. Swimlane Diagram Download & Hapus File Tugas



Gambar 3. 32 Swimlane Diagram Download & Hapus File Tugas

Keterangan : Berdasarkan Gambar 3.32 *Swimlane Diagram Download & Hapus File Tugas* menunjukan langkah-langkah asisten dalam mendownload dan menghapus file tugas yang dikirim oleh mahasiswa.

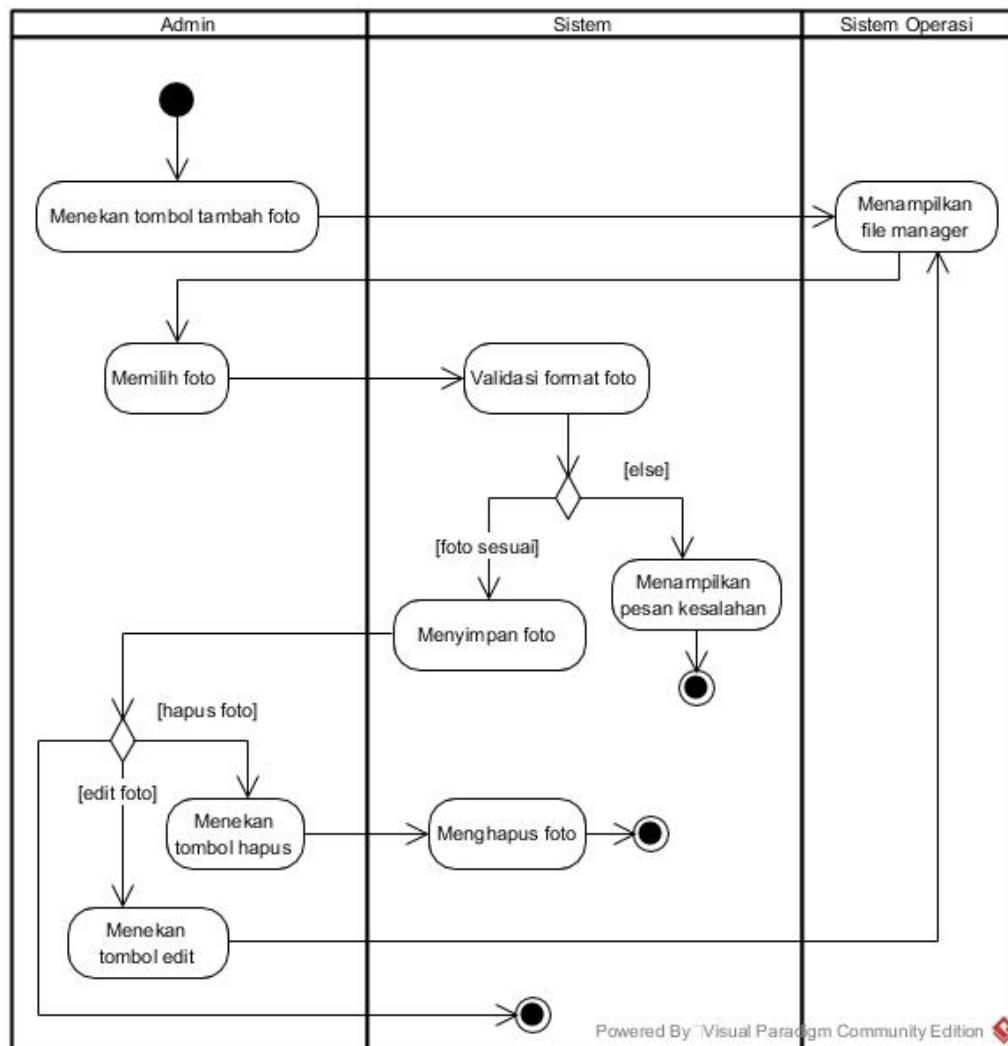
### 31. Swimlane Diagram Kelola Info Praktikum



Gambar 3. 33 Swimlane Diagram Kelola Info Praktikum

Keterangan : Berdasarkan Gambar 3.33 *Swimlane Diagram Kelola Info Praktikum* menunjukkan langkah-langkah admin dalam mengelola data info praktikum yaitu menambah, mengubah, dan menghapus info praktikum.

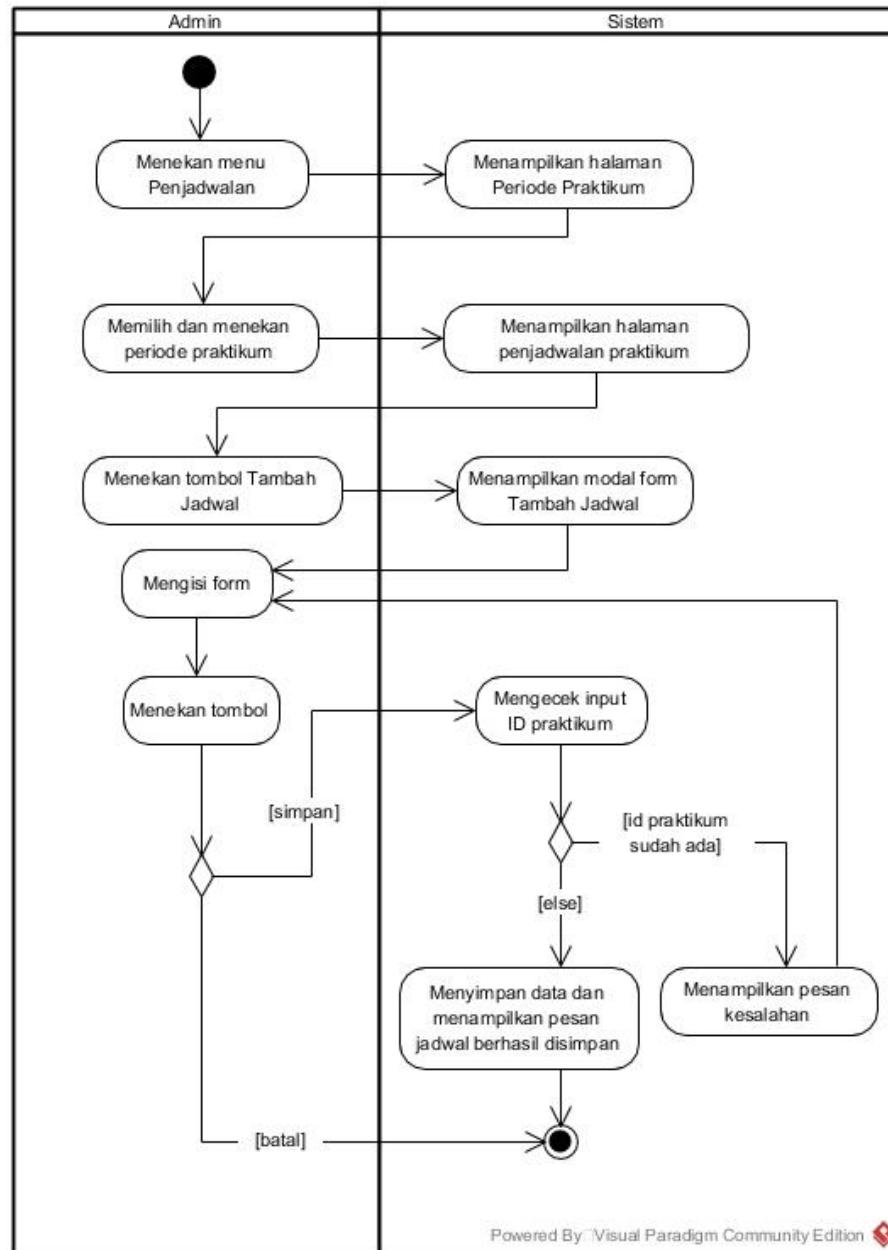
### 32. Swimlane Diagram Kelola Foto Info Praktikum



Gambar 3. 34 Swimlane Diagram Kelola Foto Info Praktikum

Keterangan : Berdasarkan Gambar 3.34 *Swimlane Diagram Kelola Foto Info Praktikum* menunjukan langkah-langkah admin dalam mengelola foto tugas yaitu menambah, mengubah, dan menghapus foto info praktikum. Kelola foto info praktikum dilakukan di dalam modal tambah atau edit info praktikum.

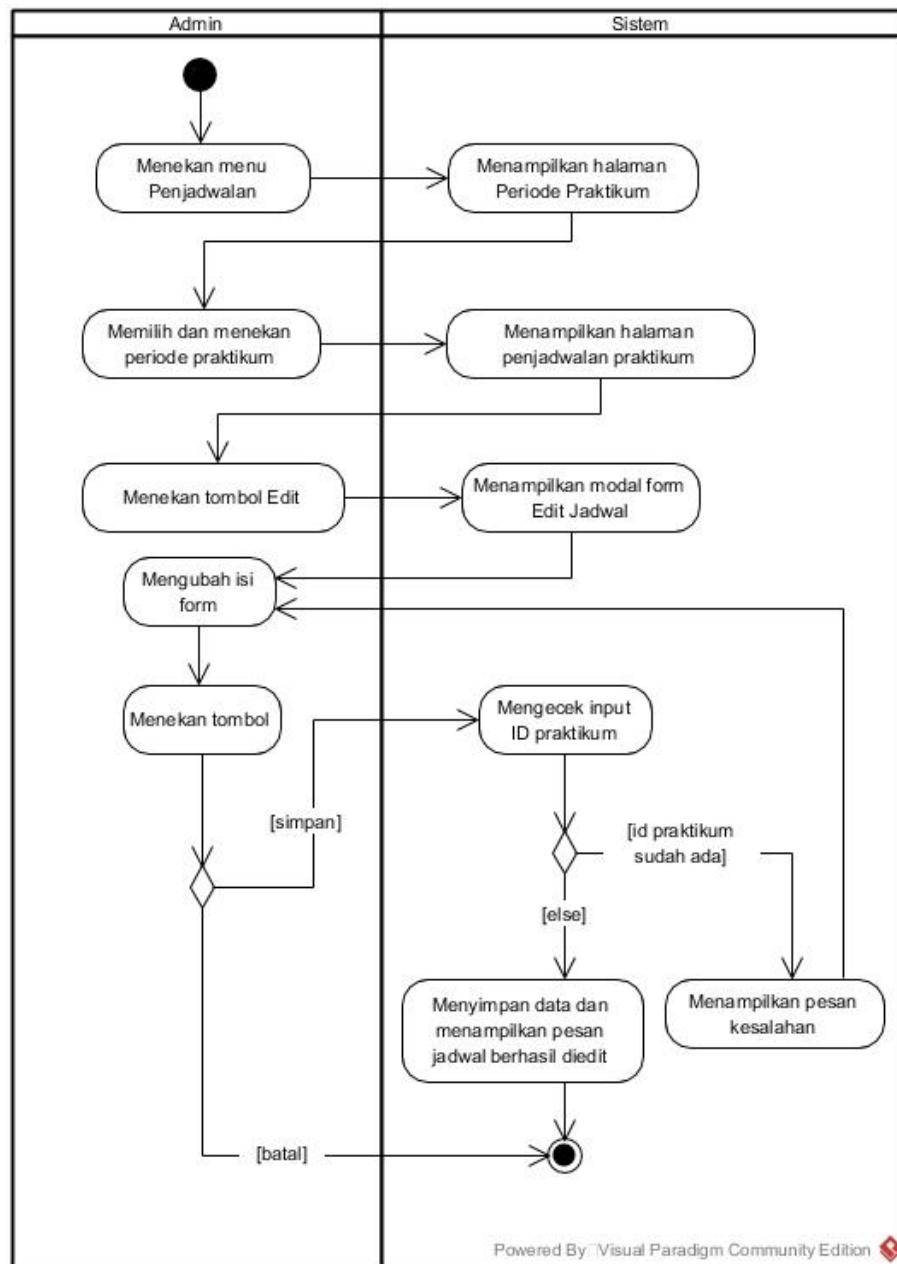
### 33. Swimlane Diagram Tambah Jadwal



Gambar 3. 35 Swimlane Diagram Tambah Jadwal

Keterangan : Berdasarkan Gambar 3.35 *Swimlane Diagram Tambah Jadwal* menunjukan langkah-langkah admin dalam menambah jadwal praktikum.

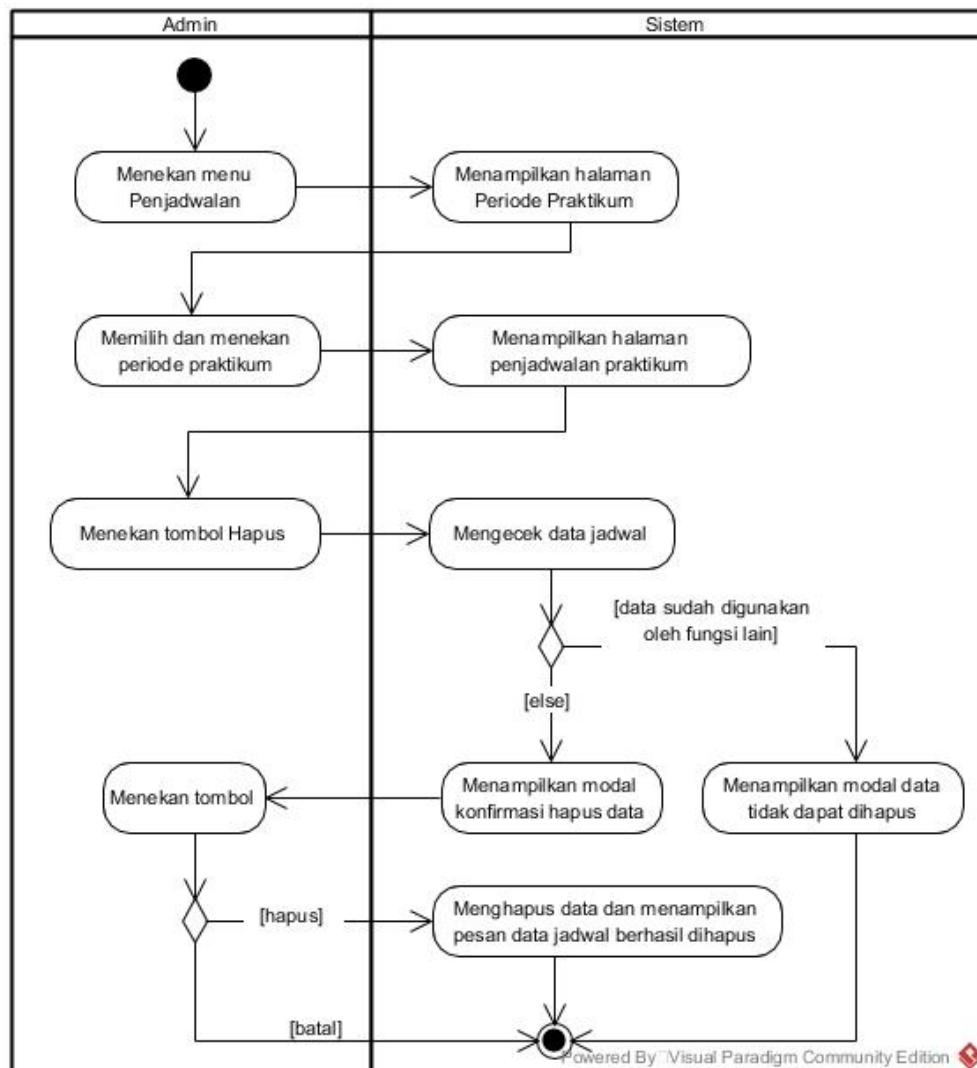
### 34. Swimlane Diagram Edit Jadwal



Gambar 3. 36 Swimlane Diagram Edit Jadwal

Keterangan : Berdasarkan Gambar 3.36 *Swimlane Diagram Edit Jadwal* menunjukan langkah-langkah admin dalam mengubah jadwal praktikum.

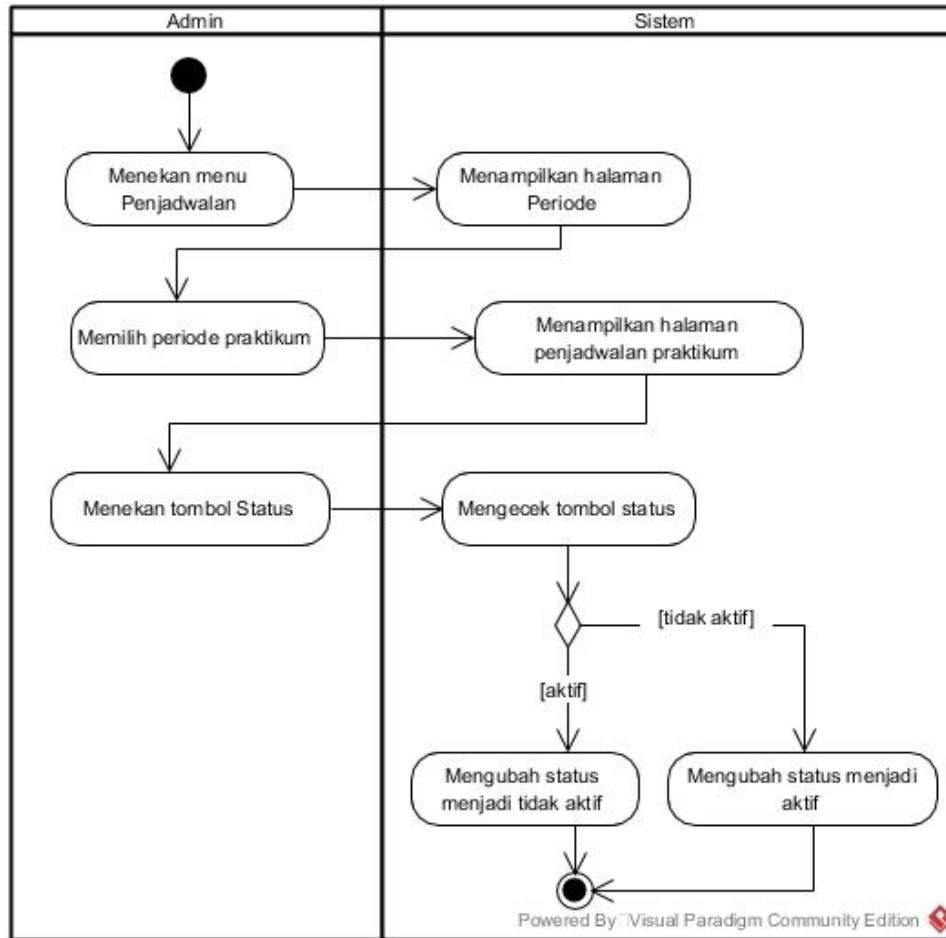
### 35. Swimlane Diagram Hapus Jadwal



Gambar 3. 37 Swimlane Diagram Hapus Jadwal

Keterangan : Berdasarkan Gambar 3.37 *Swimlane Diagram Hapus Jadwal* menunjukkan langkah-langkah admin dalam menghapus jadwal praktikum.

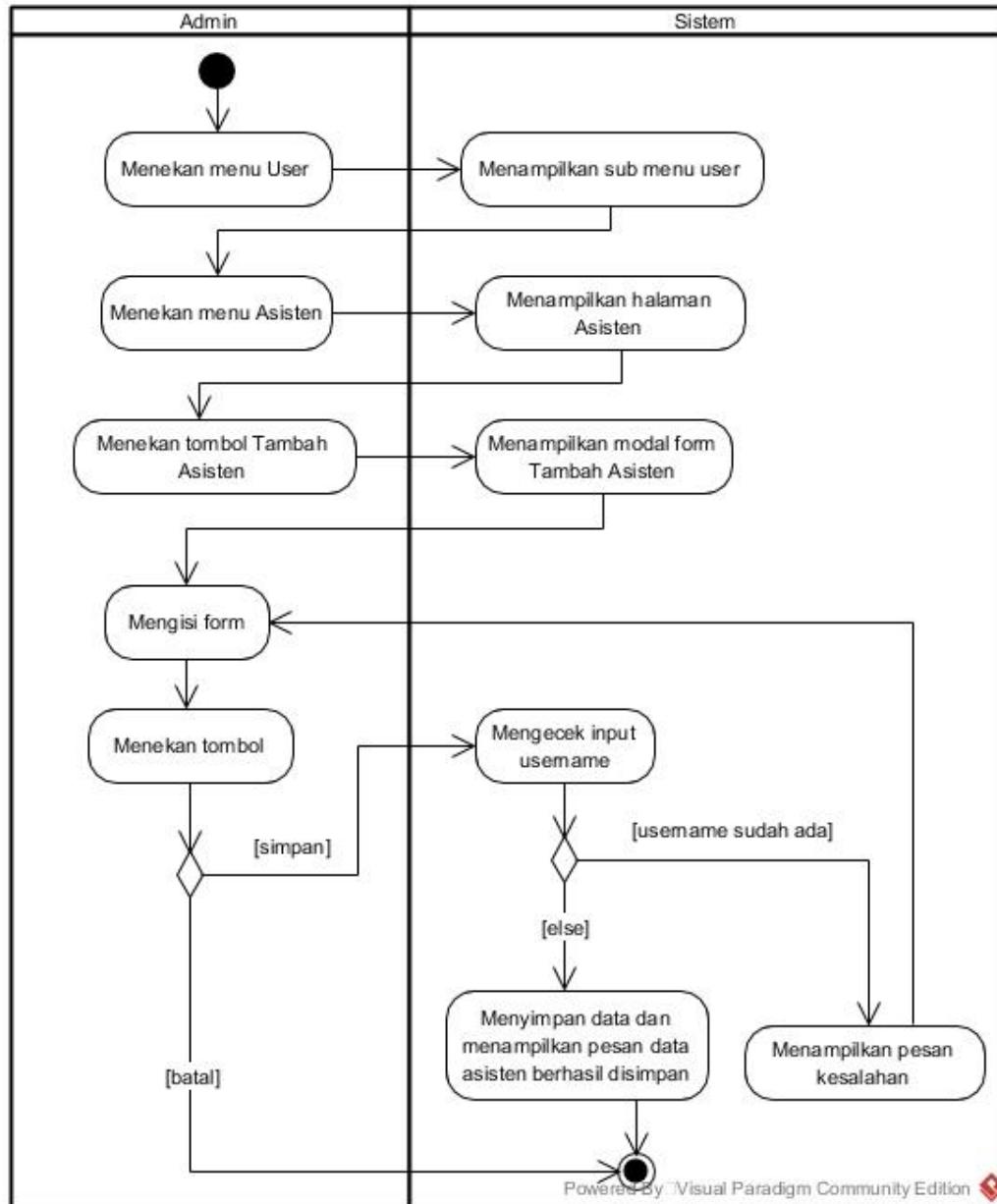
### 36. Swimlane Diagram Status Jadwal



Gambar 3. 38 Swimlane Diagram Status Jadwal

Keterangan : Berdasarkan Gambar 3.38 *Swimlane Diagram Status Jadwal* menunjukan langkah-langkah admin dalam mengubah status jadwal praktikum menjadi aktif atau tidak aktif.

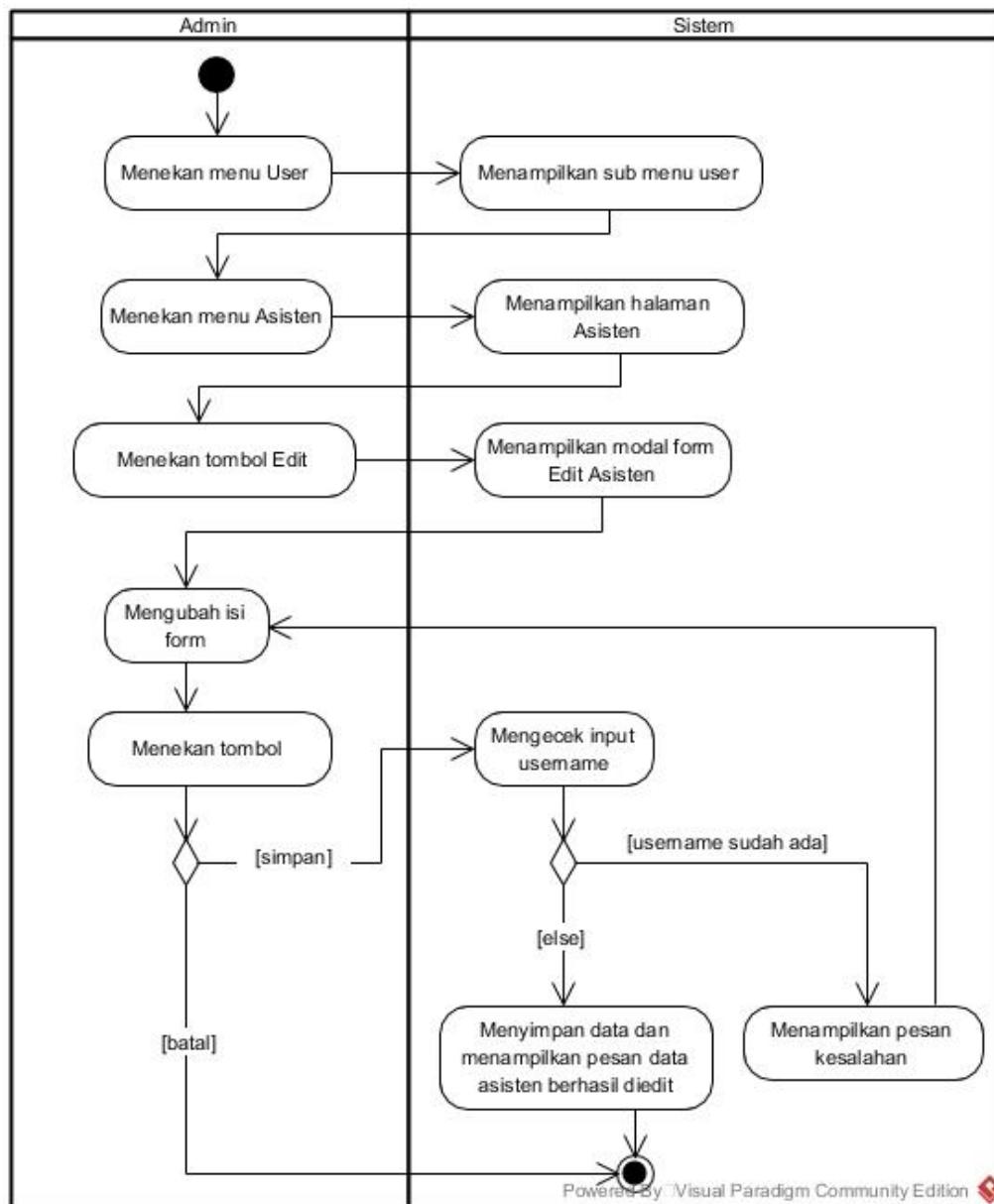
37. *Swimlane Diagram Tambah Asisten*



Gambar 3. 39 *Swimlane Diagram Tambah Asisten*

Keterangan : Berdasarkan Gambar 3.39 *Swimlane Diagram Tambah Asisten* menunjukan langkah-langkah admin dalam menambah asisten.

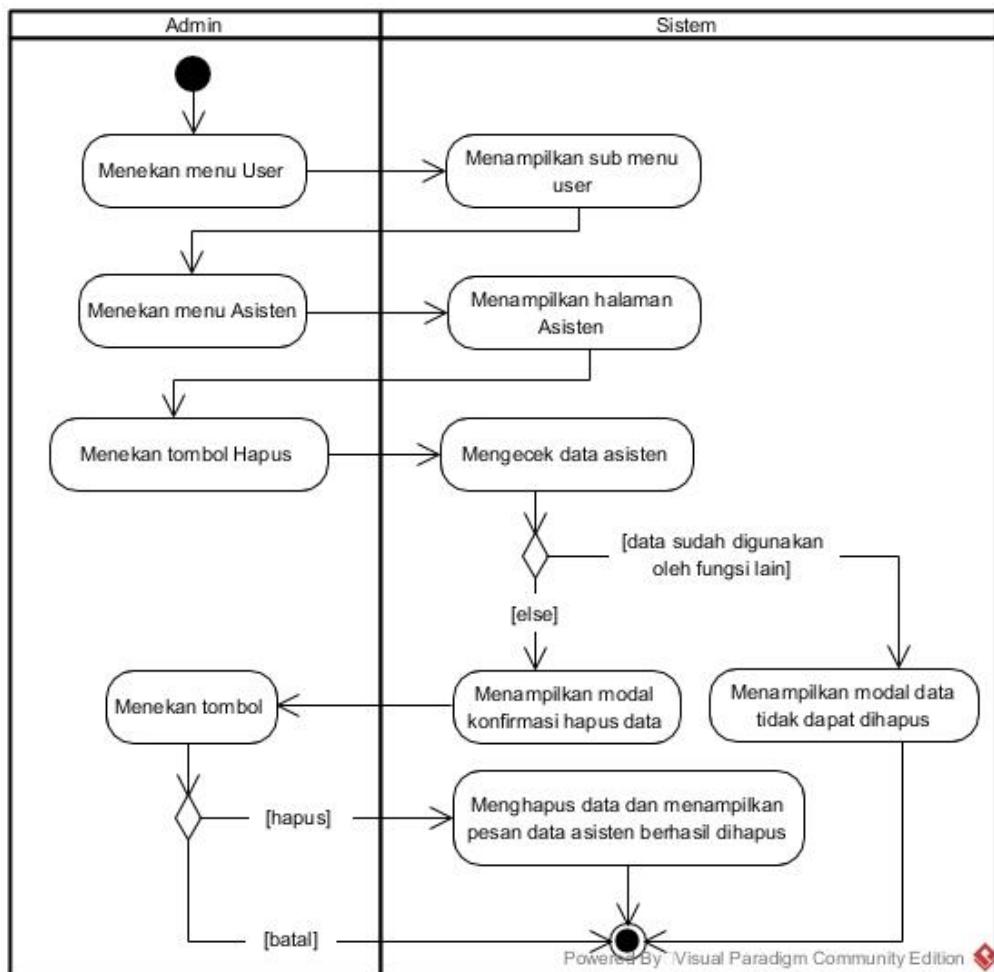
38. *Swimlane Diagram Edit Asisten*



*Gambar 3. 40 Swimlane Diagram Edit Asisten*

Keterangan : Berdasarkan Gambar 3.40 *Swimlane Diagram Edit Asisten* menunjukan langkah-langkah admin dalam mengubah data asisten.

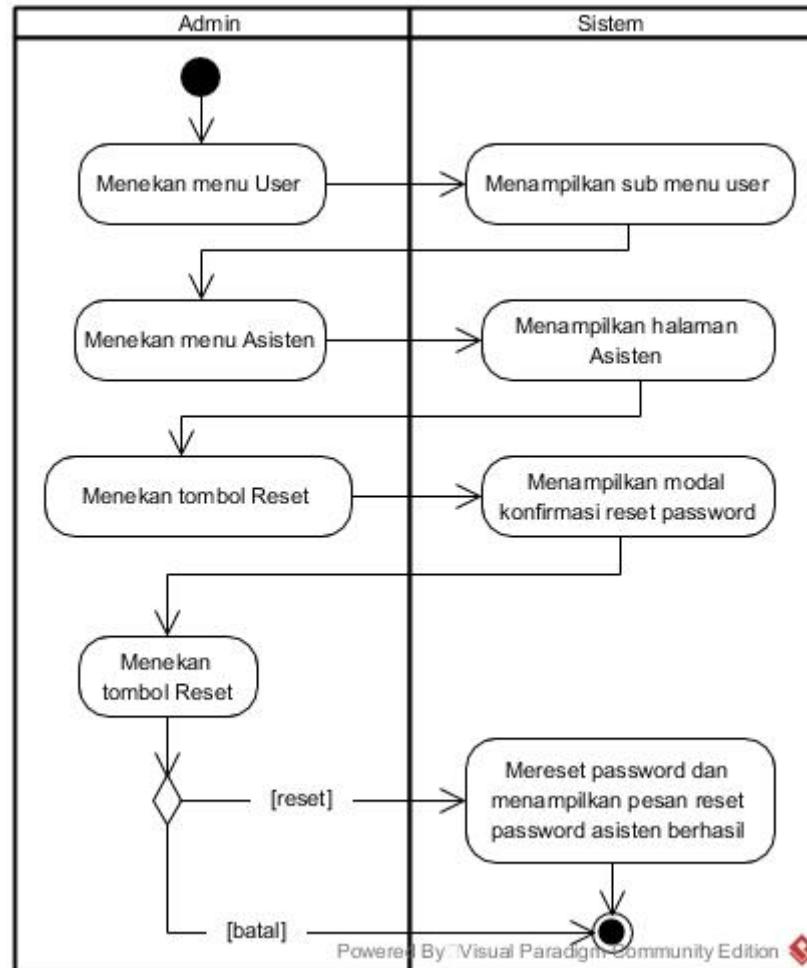
### 39. Swimlane Diagram Hapus Asisten



Gambar 3. 41 Swimlane Diagram Hapus Asisten

Keterangan : Berdasarkan Gambar 3.41 *Swimlane Diagram Hapus Asisten* menunjukan langkah-langkah admin dalam menghapus data asisten.

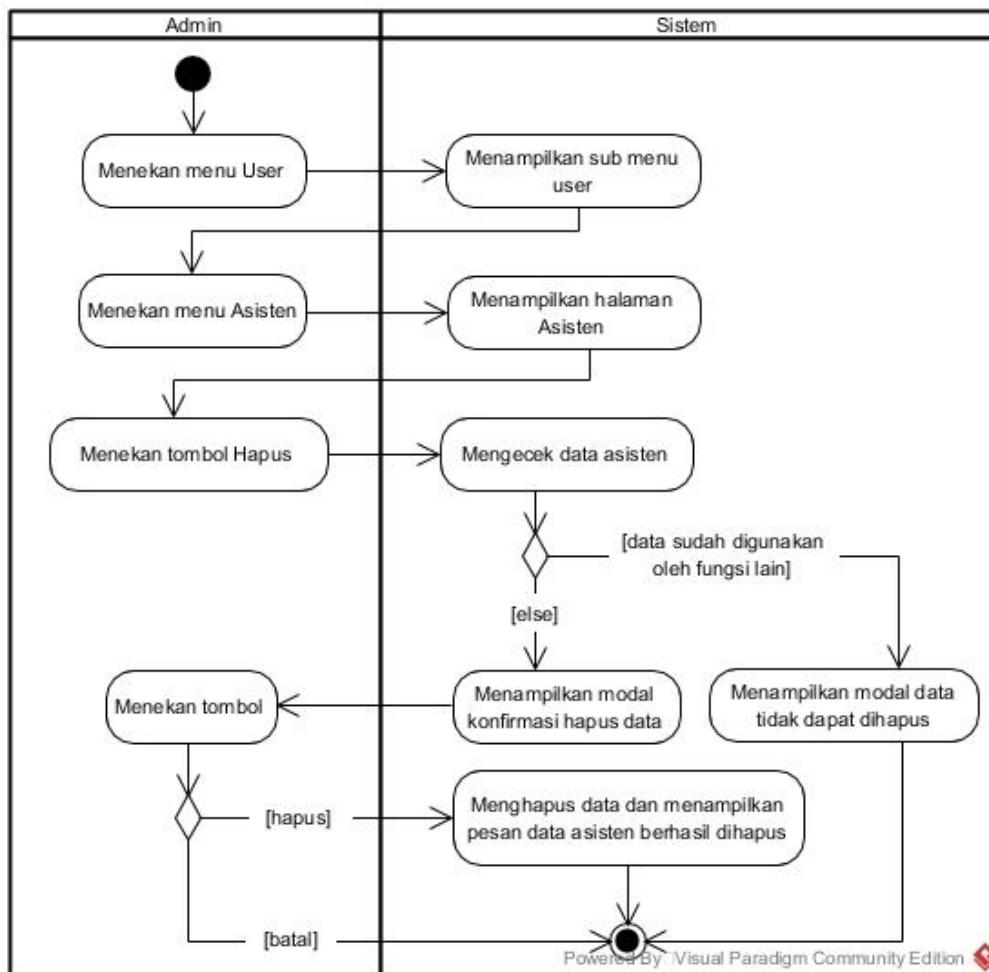
40. *Swimlane Diagram Reset Password Asisten*



*Gambar 3. 42 Swimlane Diagram Reset Password Asisten*

Keterangan : Berdasarkan Gambar 3.42 *Swimlane Diagram Reset Password Asisten* menunjukan langkah-langkah admin dalam mereset password asisten.

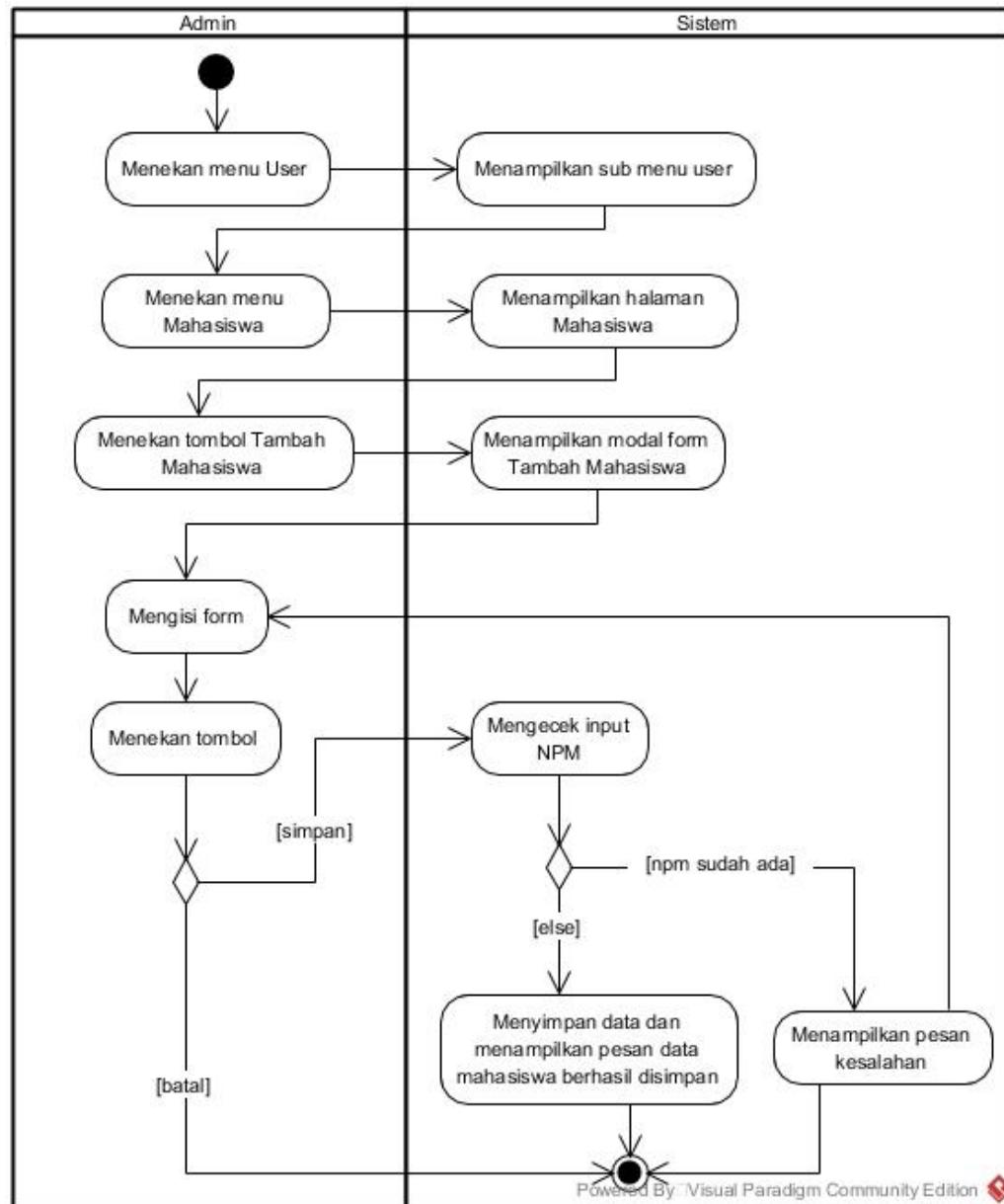
#### 41. Swimlane Diagram Status Asisten



Gambar 3. 43 Swinlane Diagram Status Asisten

Keterangan : Berdasarkan Gambar 3.43 *Swimlane Diagram Status Asisten* menunjukan langkah-langkah admin dalam mengubah status mengajar asisten menjadi aktif atau tidak aktif.

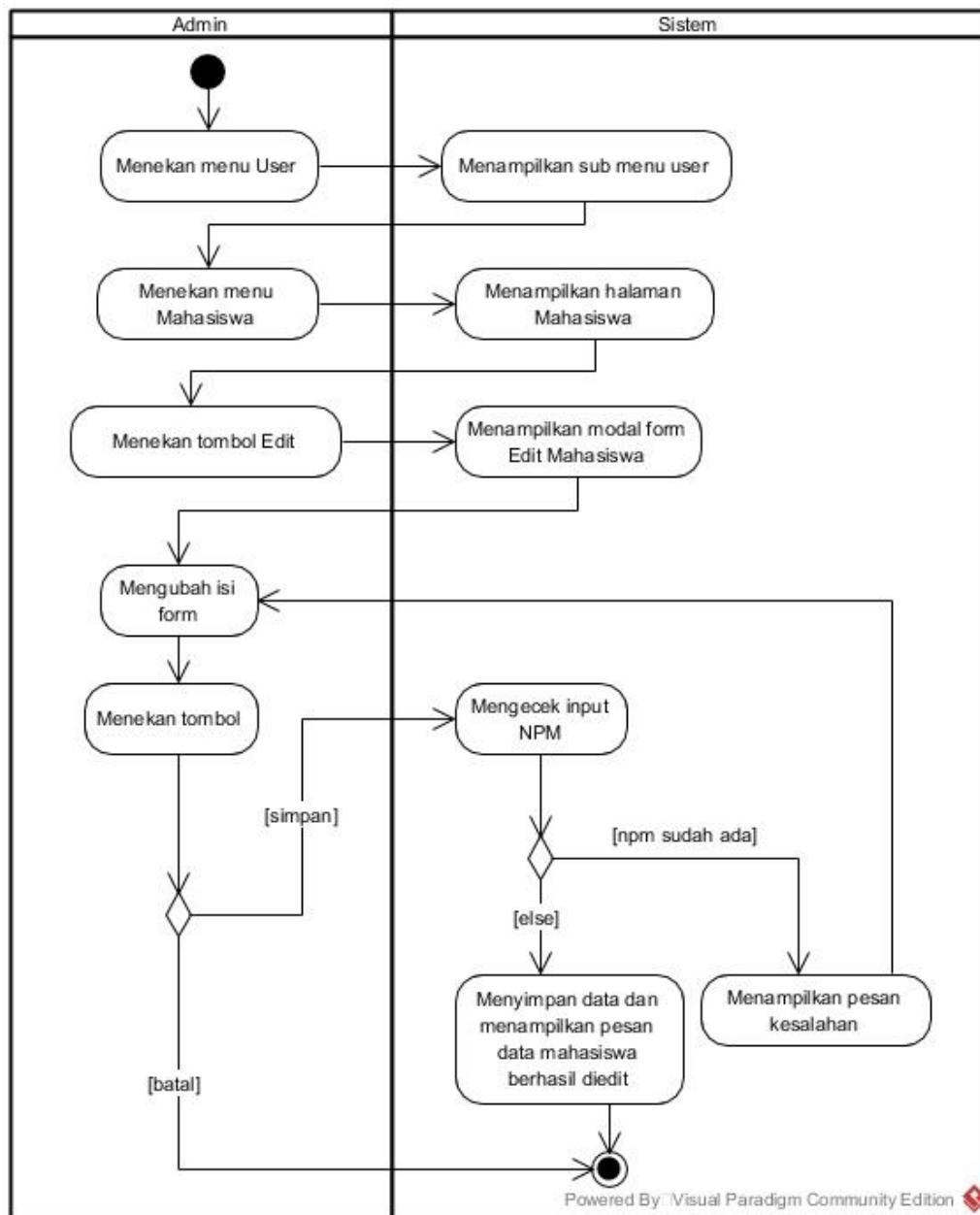
42. Swimlane Diagram Tambah Mahasiswa



Gambar 3. 44 Swinlane Diagram Tambah Mahasiswa

Keterangan : Berdasarkan Gambar 3.44 Swinlane Diagram Tambah Mahasiswa menunjukan langkah-langkah admin dalam menambah mahasiswa.

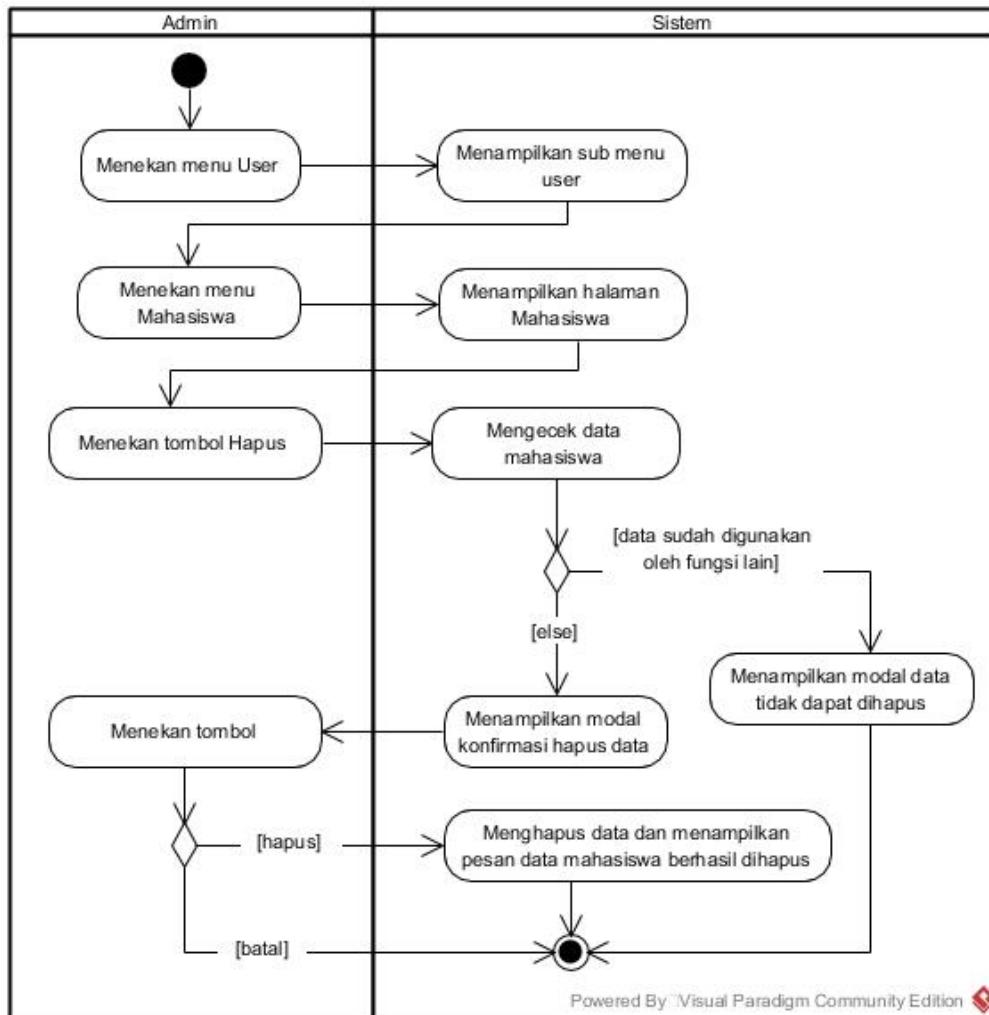
43. *Swimlane Diagram Edit Mahasiswa*



Gambar 3. 45 *Swimlane Diagram Edit Mahasiswa*

Keterangan : Berdasarkan Gambar 3.45 *Swimlane Diagram Edit Mahasiswa* menunjukan langkah-langkah admin dalam mengubah data mahasiswa.

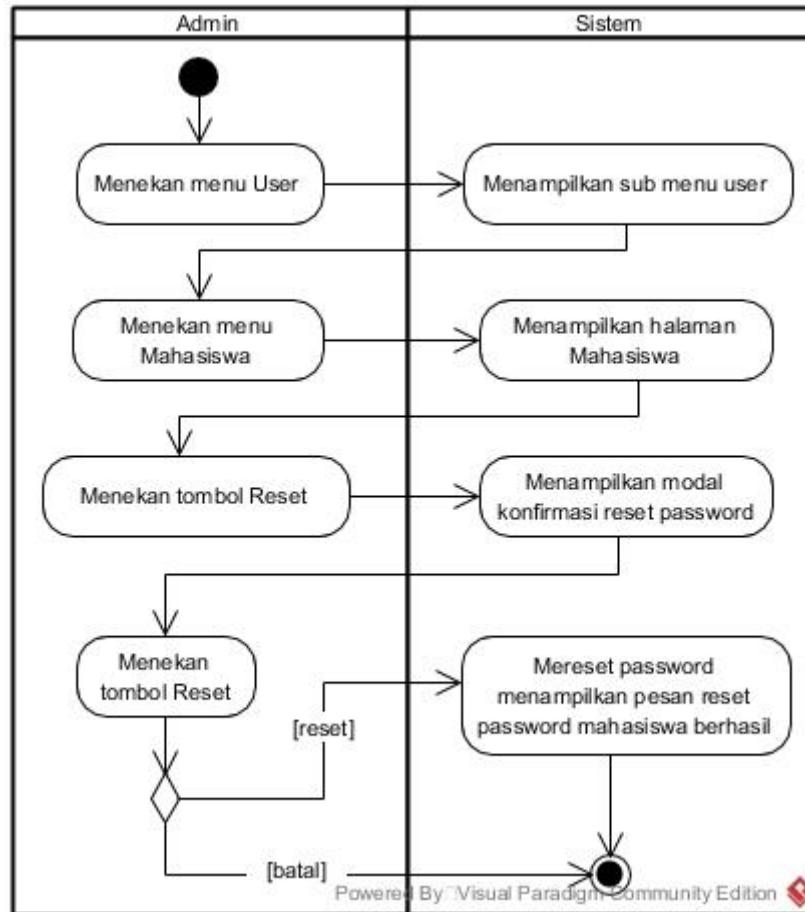
44. *Swimlane Diagram Hapus Mahasiswa*



Gambar 3. 46 *Swimlane Diagram Hapus Mahasiswa*

Keterangan : Berdasarkan Gambar 3.46 *Swimlane Diagram Hapus Mahasiswa* menunjukan langkah-langkah admin dalam menghapus data mahasiswa.

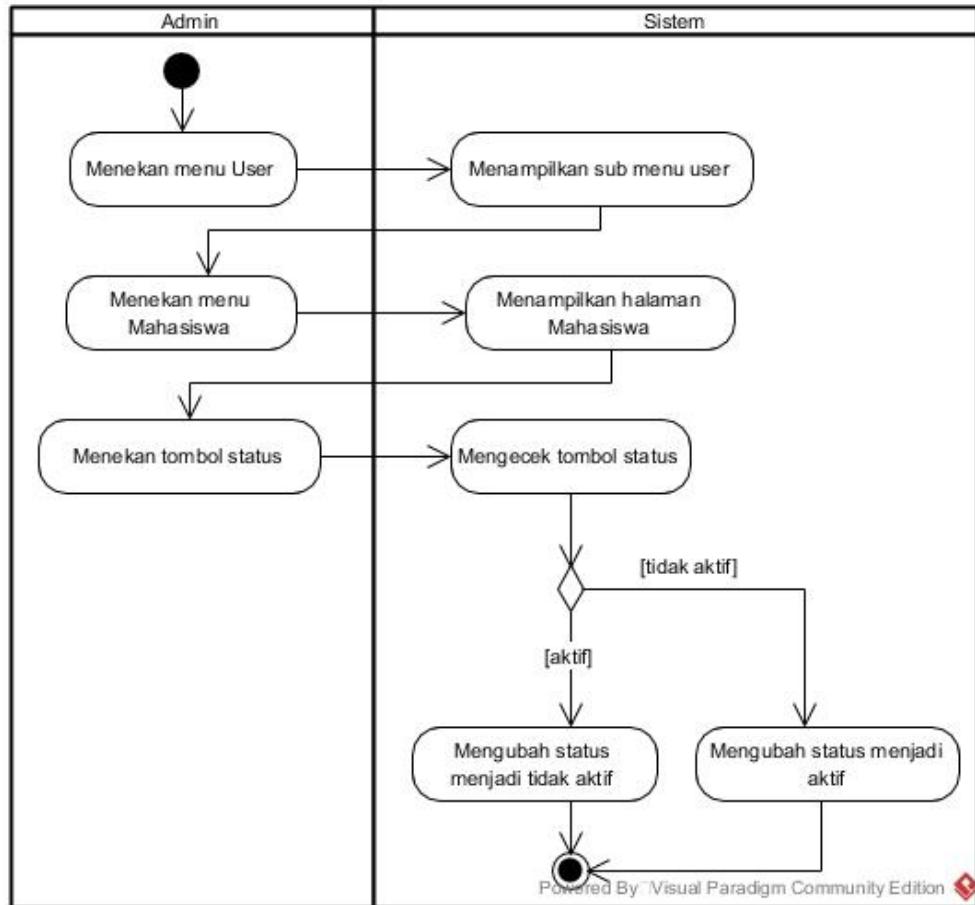
45. *Swimlane Diagram Reset Password Mahasiswa*



*Gambar 3. 47 Swimlane Diagram Reset Password Mahasiswa*

Keterangan : Berdasarkan Gambar 3.47 *Swimlane Diagram Reset Password Mahasiswa* menunjukan langkah-langkah admin dalam mereset password mahasiswa.

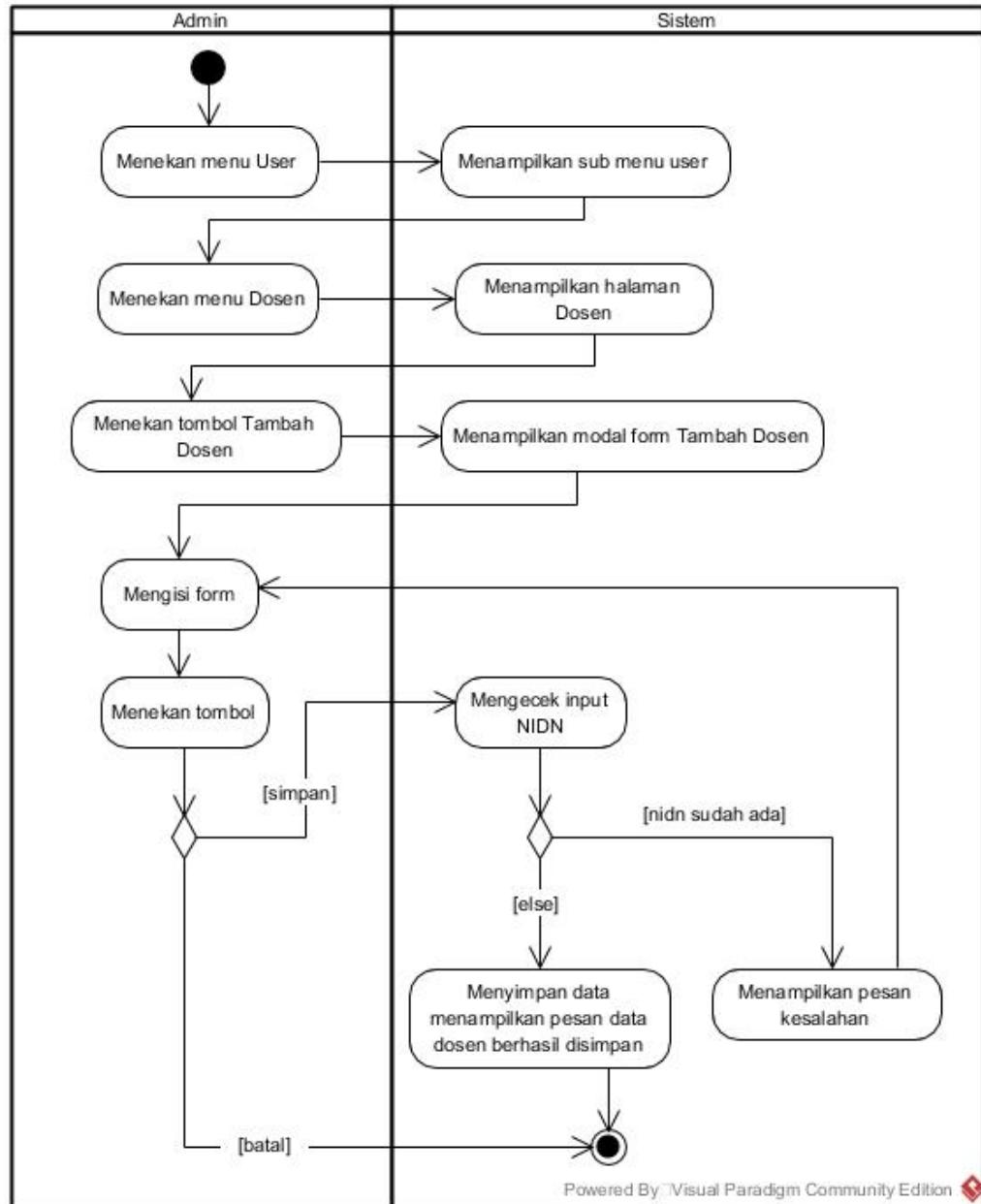
46. *Swimlane Diagram Status Mahasiswa*



Gambar 3. 48 *Swimlane Diagram Status Asisten*

Keterangan : Berdasarkan Gambar 3.48 *Swimlane Diagram Status Mahasiswa* menunjukan langkah-langkah admin dalam mengubah status mahasiswa menjadi aktif atau tidak aktif.

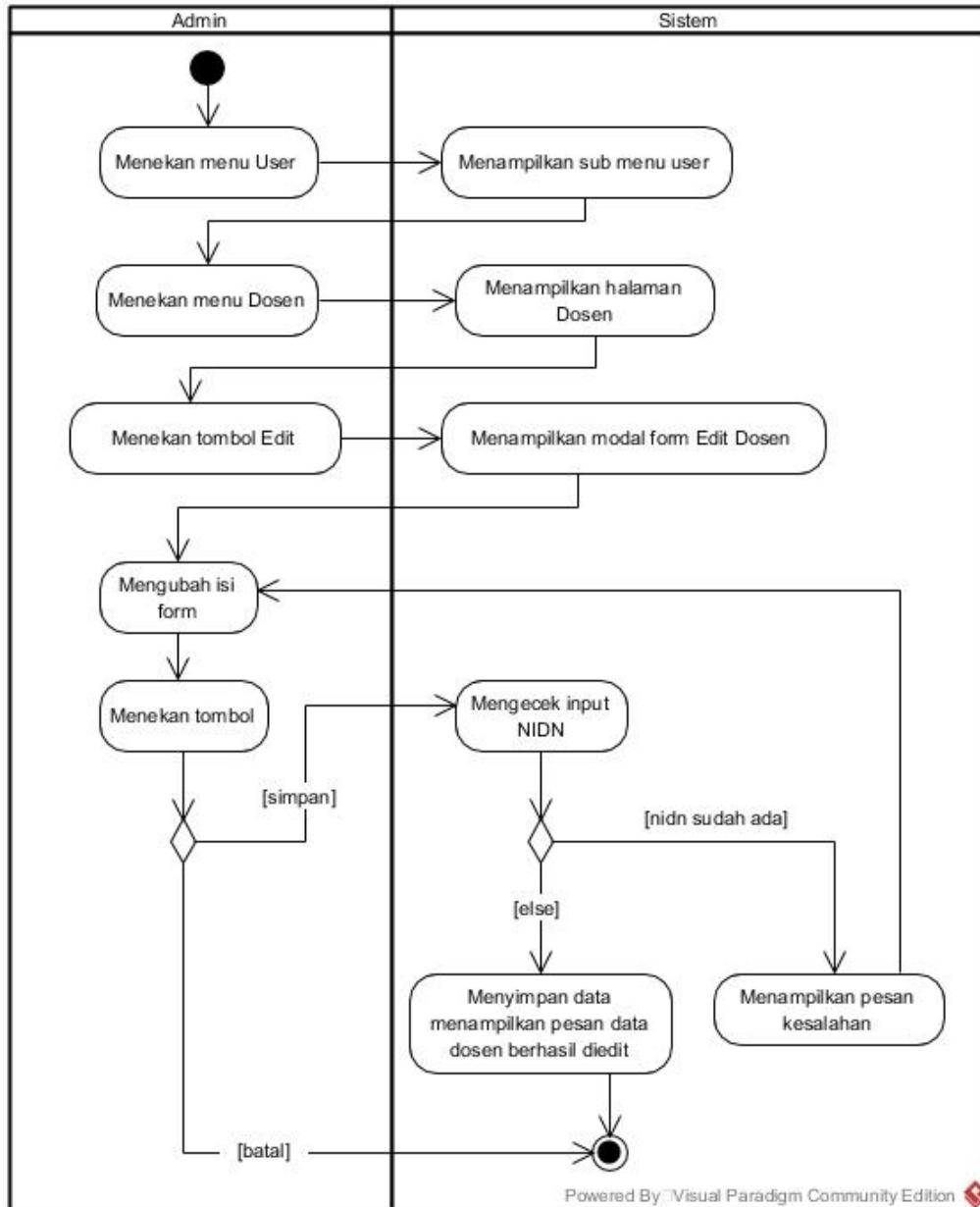
47. *Swimlane Diagram Tambah Dosen*



Gambar 3. 49 *Swimlane Diagram Tambah Dosen*

Keterangan : Berdasarkan Gambar 3.49 *Swimlane Diagram Tambah Dosen* menunjukkan langkah-langkah admin dalam menambah dosen.

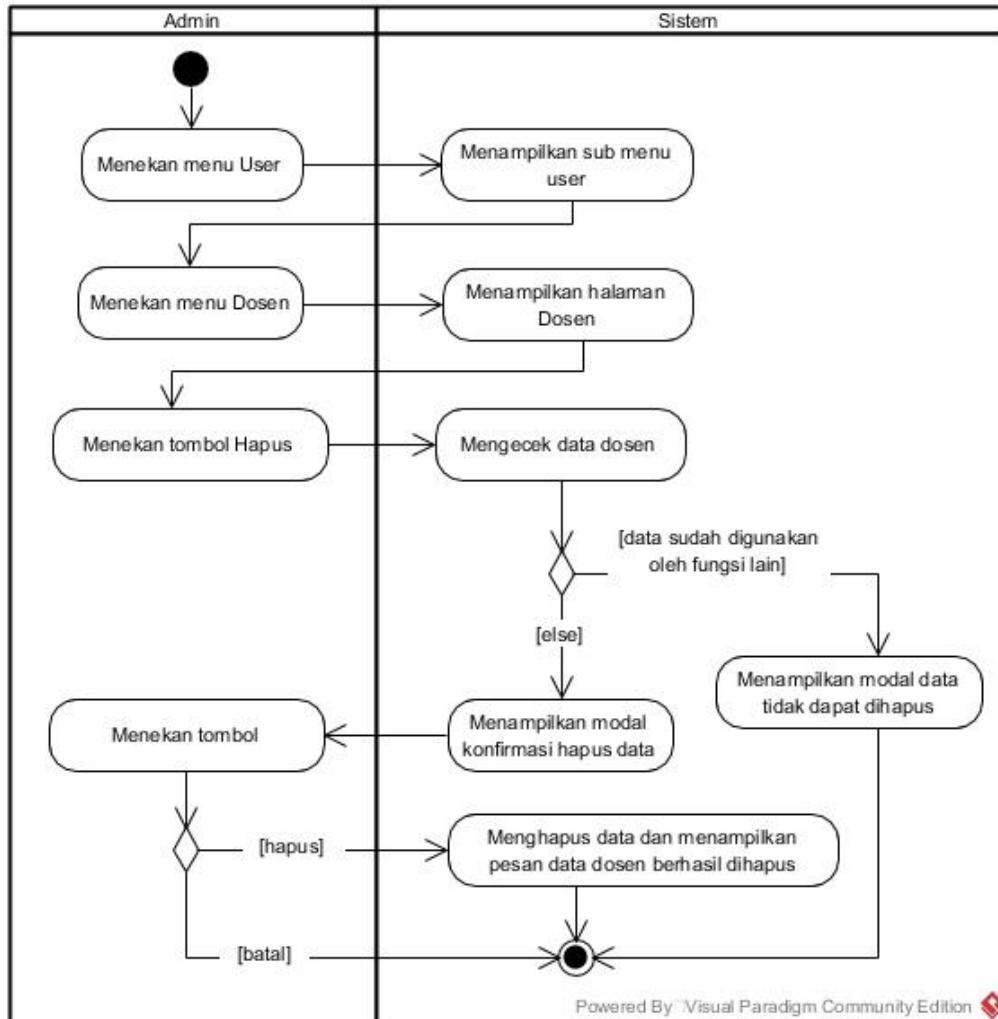
48. *Swimlane Diagram Edit Dosen*



*Gambar 3. 50 Swimlane Diagram Edit Dosen*

Keterangan : Berdasarkan Gambar 3.50 *Swimlane Diagram Edit Dosen* menunjukkan langkah-langkah admin dalam mengubah data dosen.

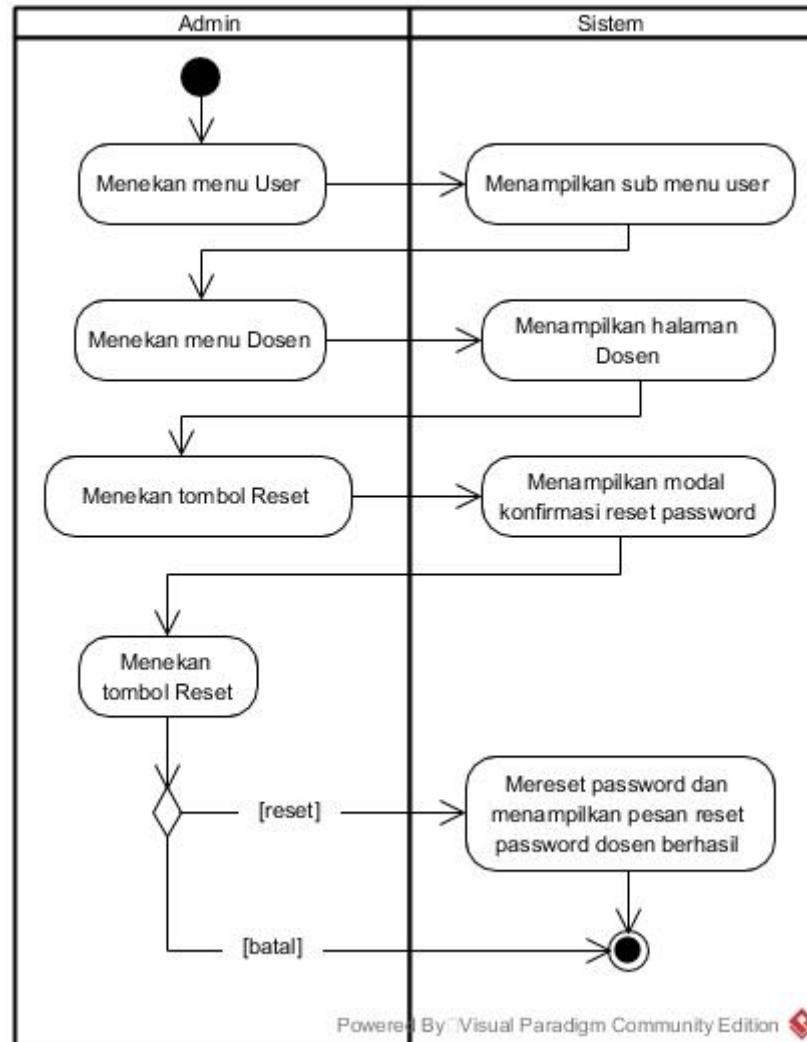
49. *Swimlane Diagram Hapus Dosen*



Gambar 3. 51 *Swimlane Diagram Hapus Dosen*

Keterangan : Berdasarkan Gambar 3.51 *Swimlane Diagram Hapus Dosen* menunjukkan langkah-langkah admin dalam menghapus data dosen.

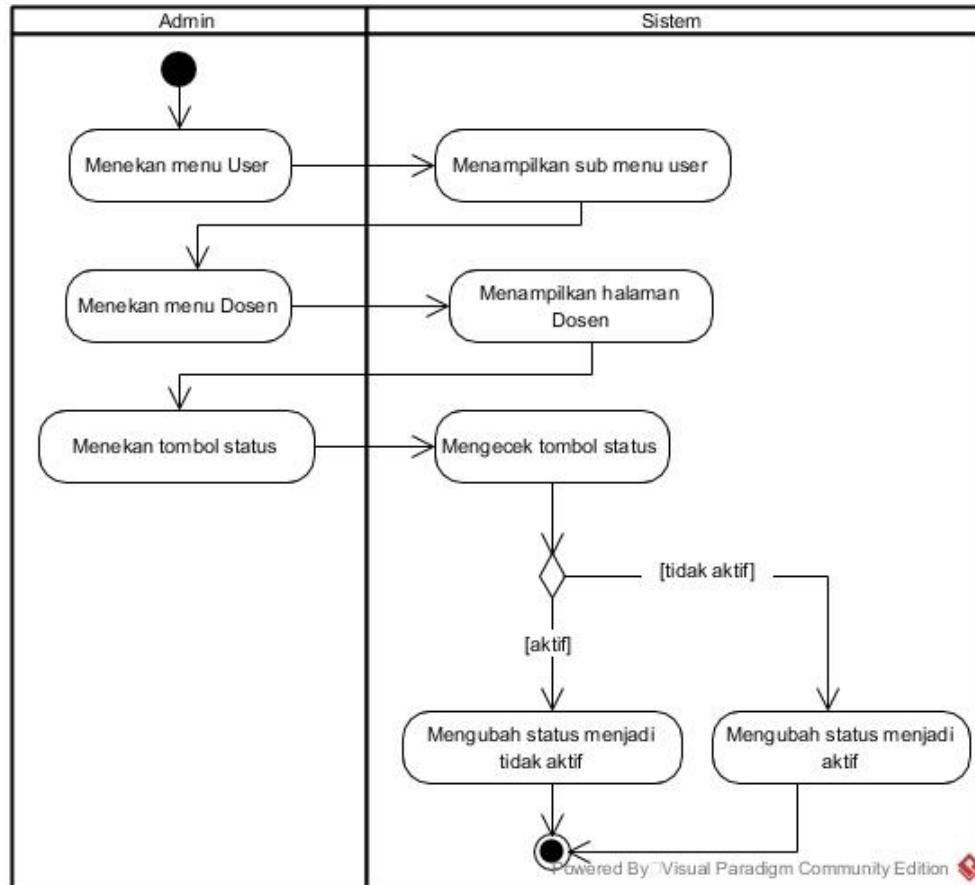
50. *Swimlane Diagram Reset Password Dosen*



*Gambar 3. 52 Swimlane Diagram Reset Password Dosen*

Keterangan : Berdasarkan Gambar 3.52 *Swimlane Diagram Reset Password Dosen* menunjukan langkah-langkah admin dalam mereset password dosen.

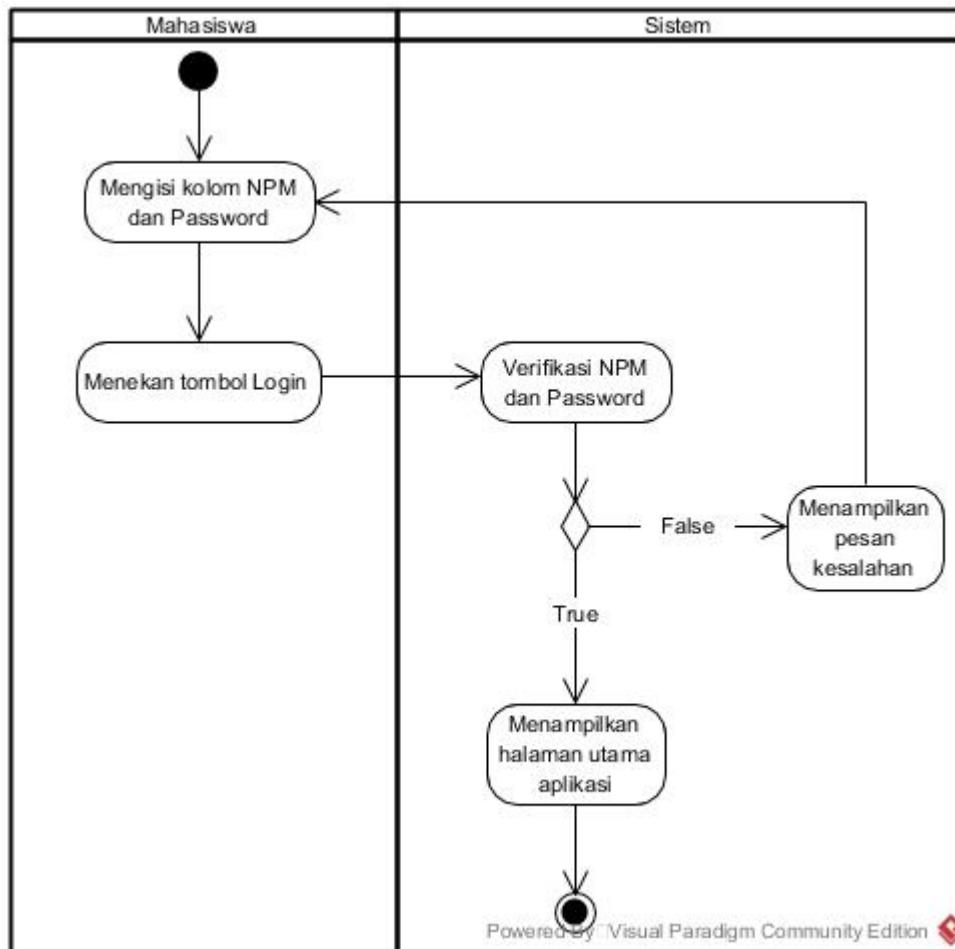
51. *Swimlane Diagram Status Dosen*



Gambar 3. 53 *Swimlane Diagram Status Dosen*

Keterangan : Berdasarkan Gambar 3.53 *Swimlane Diagram Status Dosen* menunjukan langkah-langkah admin dalam mengubah status mengajar dosen menjadi aktif atau tidak aktif.

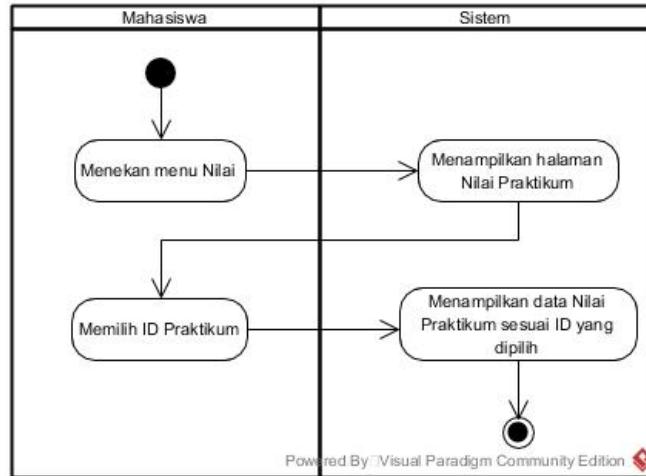
52. *Swimlane Diagram Login (Client)*



Gambar 3. 54 *Swimlane Diagram Login*

Keterangan : Berdasarkan Gambar 3.54 *Swimlane Diagram Login Aplikasi Client* menunjukan langkah-langkah mahasiswa untuk masuk ke dalam aplikasi *client*.

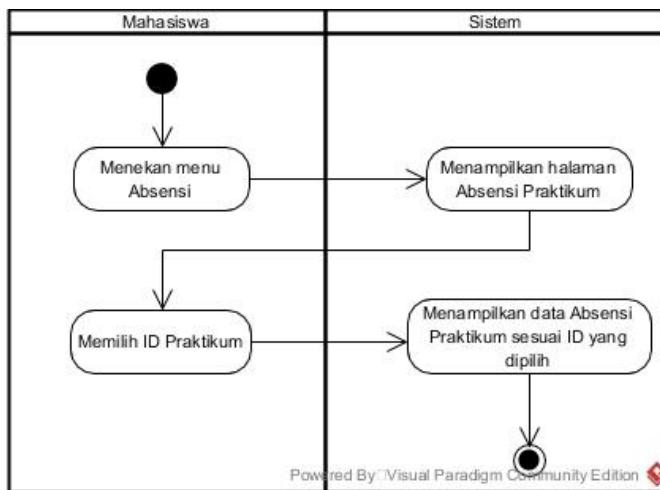
### 53. Swimlane Diagram Cek Nilai Tugas



Gambar 3. 55 Swimlane Diagram Cek Nilai Tugas

Keterangan : Berdasarkan Gambar 3.55 *Swimlane Diagram Cek Nilai Tugas* menunjukan langkah-langkah mahasiswa untuk melihat seluruh nilai tugas praktikum.

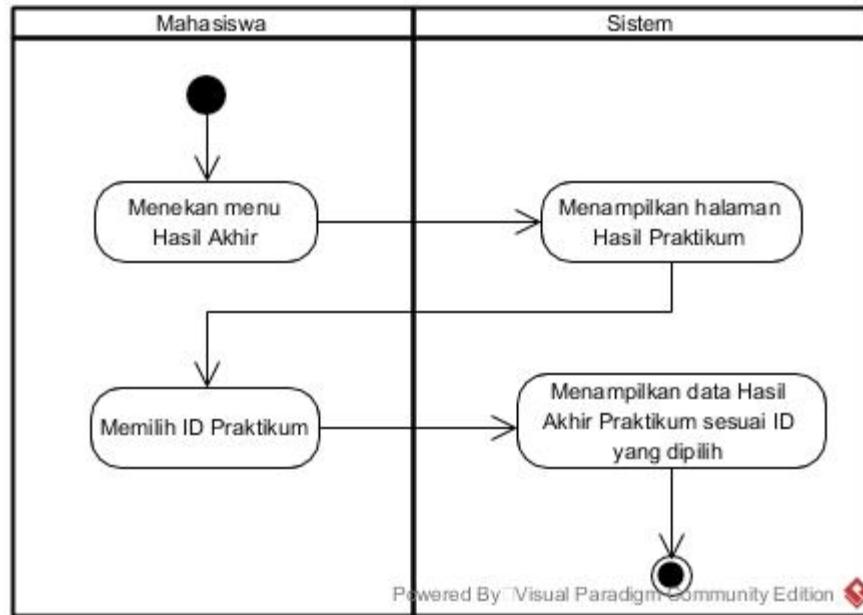
### 54. Swimlane Diagram Cek Absensi



Gambar 3. 56 Swimlane Diagram Cek Absensi

Keterangan : Berdasarkan Gambar 3.56 *Swimlane Diagram Cek Absensi* menunjukan langkah-langkah mahasiswa untuk melihat seluruh data absensi praktikum.

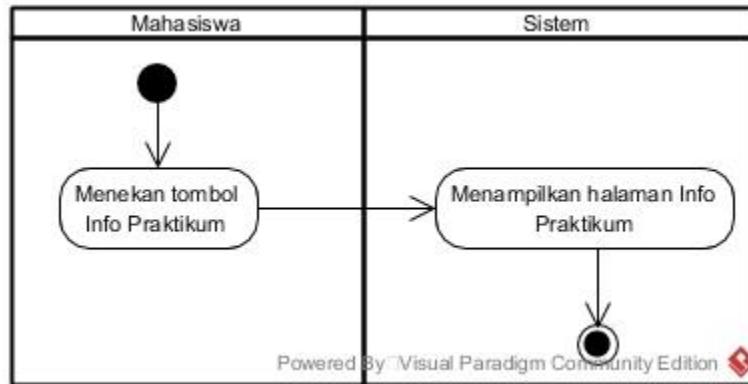
### 55. Swimlane Diagram Cek Hasil Praktikum



Gambar 3. 57 Swimlane Diagram Cek Hasil Praktikum

Keterangan : Berdasarkan Gambar 3.57 *Swimlane Diagram Cek Hasil Praktikum* menunjukan langkah-langkah mahasiswa untuk melihat hasil praktikum.

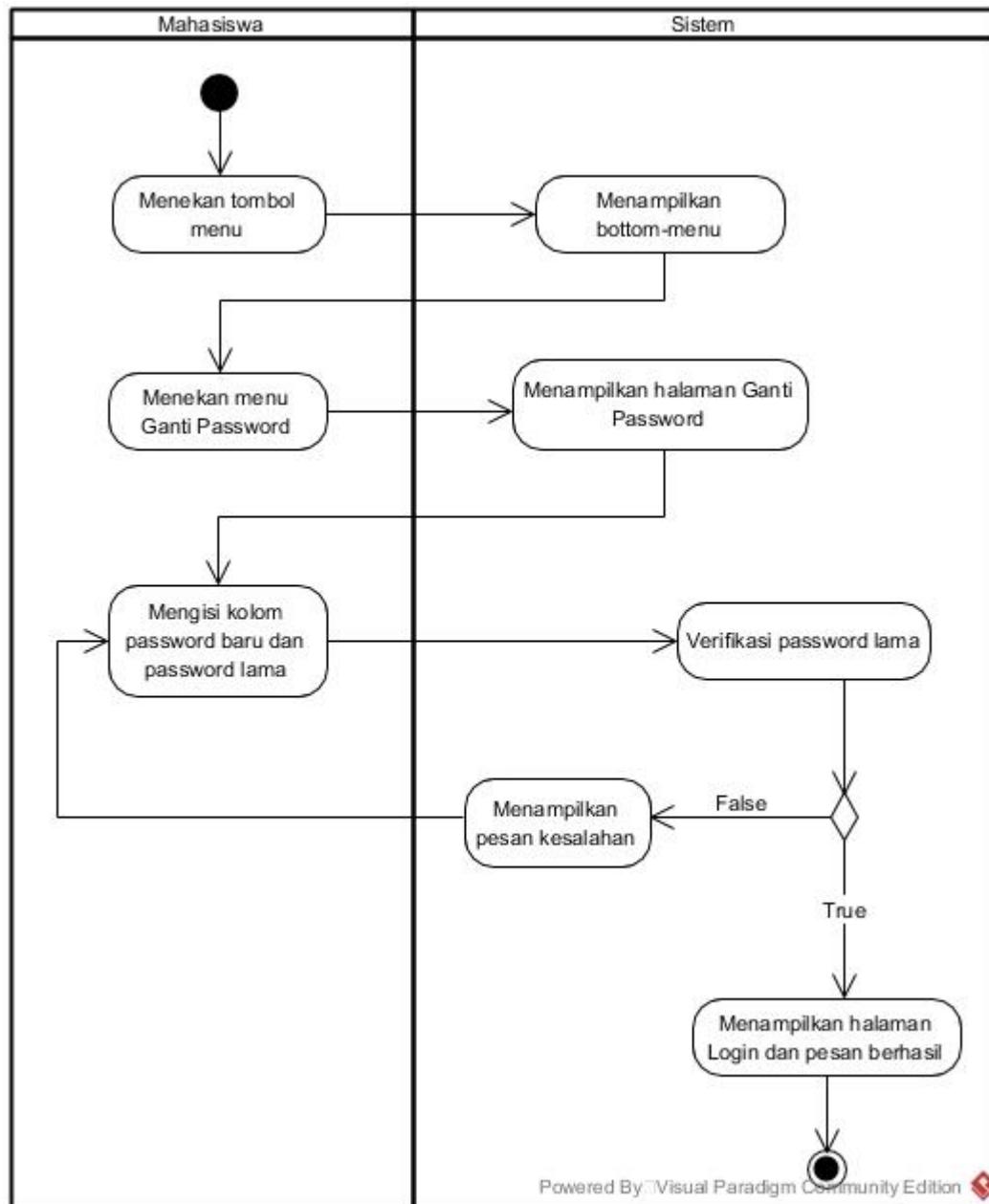
### 56. Swimlane Diagram Cek Info Praktikum



Gambar 3. 58 Swimlane Diagram Cek Info Praktikum

Keterangan : Berdasarkan Gambar 3.58 *Swimlane Diagram Cek Info Praktikum* menunjukan langkah-langkah mahasiswa untuk melihat info praktikum.

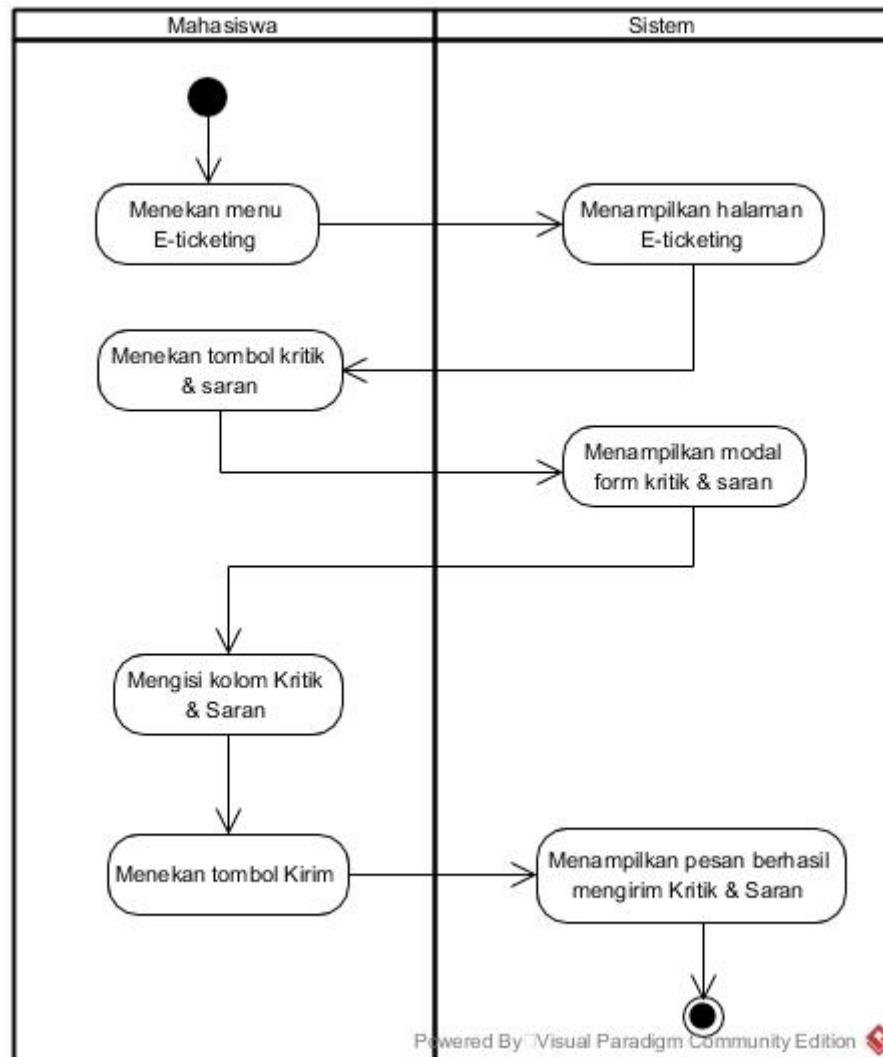
57. *Swimlane Diagram Ganti Password*



Gambar 3. 59 *Swimlane Diagram Ganti Password*

Keterangan : Berdasarkan Gambar 3.59 *Swimlane Diagram Ganti Password* menunjukan langkah-langkah mahasiswa untuk mengganti password akun.

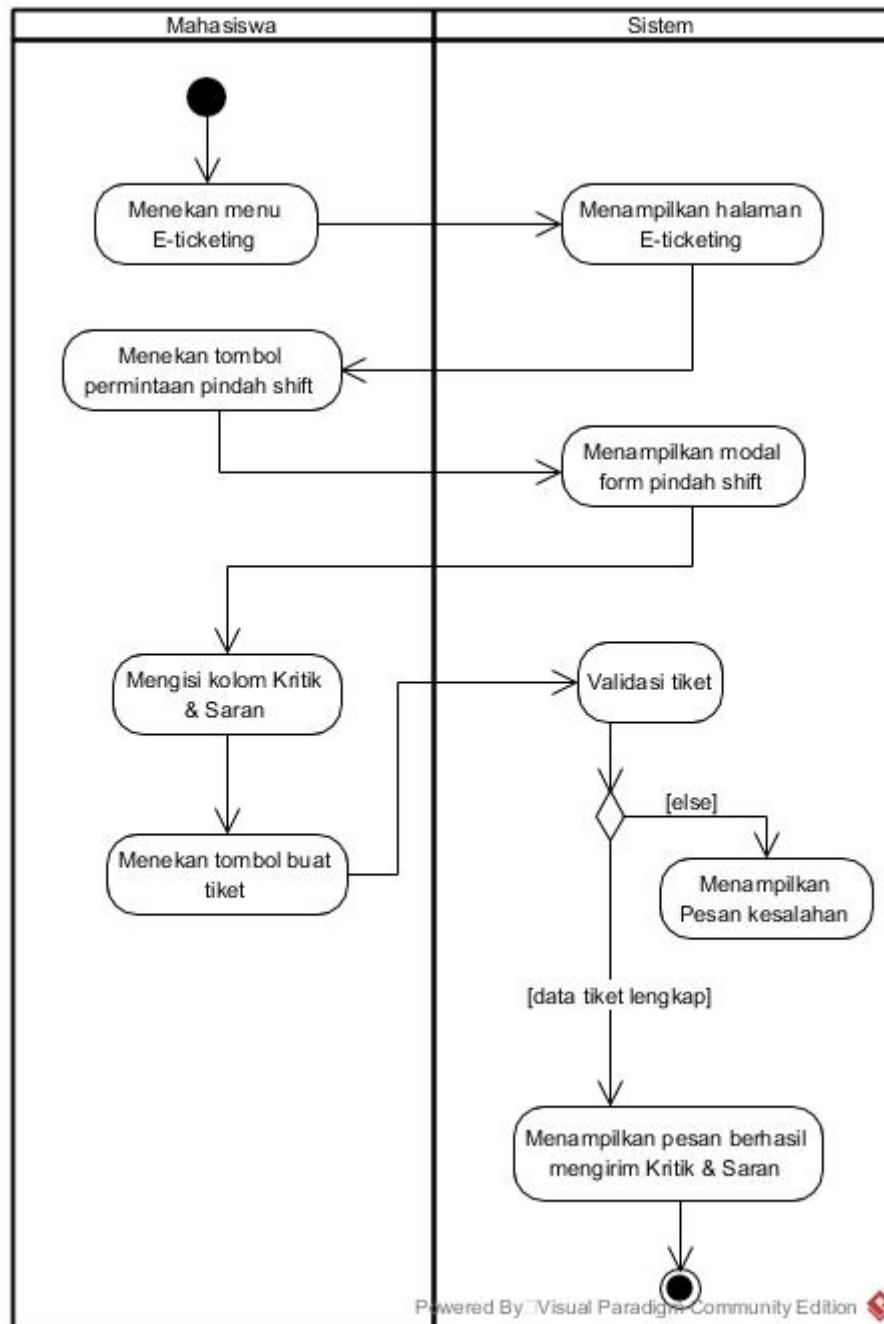
58. *Swimlane Diagram Mengirim Kritik & Saran*



*Gambar 3. 60 Swimlane Diagram Mengirim Kritik & Saran*

Keterangan : Berdasarkan Gambar 3.60 *Swimlane Diagram Mengirim Kritik & Saran* menunjukan langkah-langkah mahasiswa untuk mengirim kritik & saran.

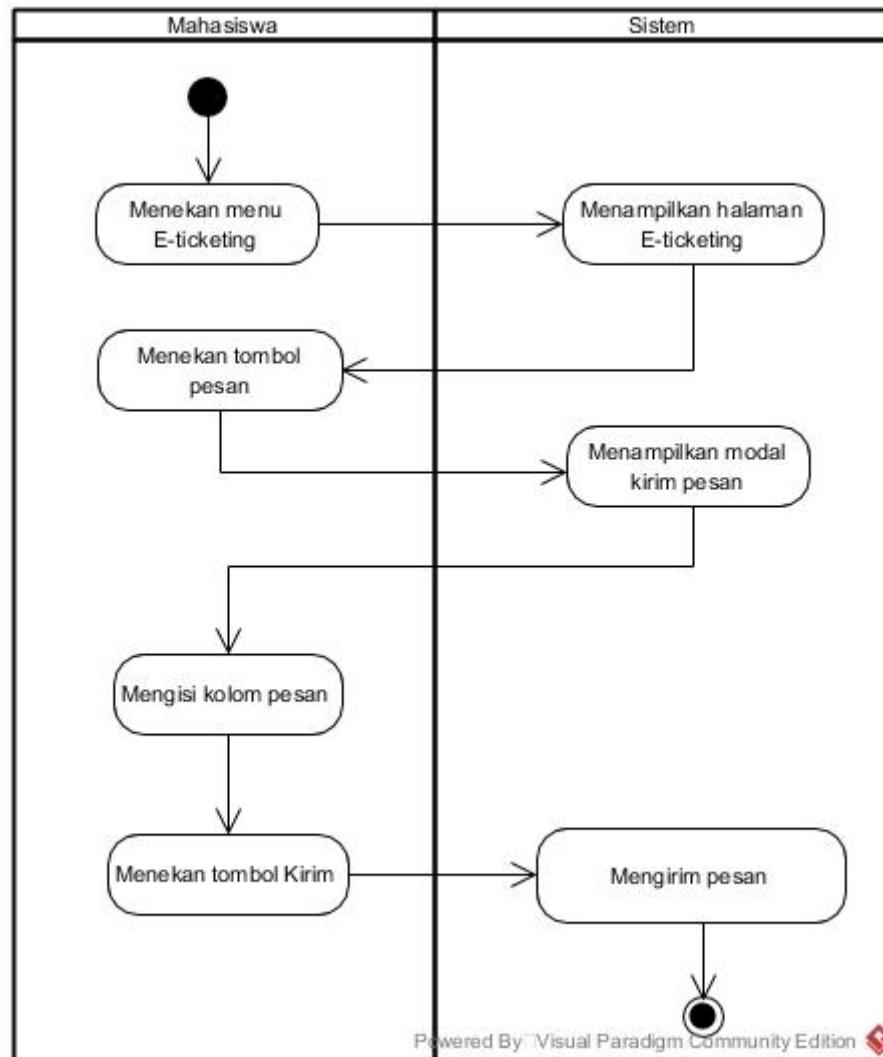
59. Swimlane Diagram Buat Tiket Permintaan Pindah Praktikum



Gambar 3. 61 Swinlane Diagram Buat Tiket Permintaan Pindah Praktikum

Keterangan : Berdasarkan Gambar 3.61 Swinlane Diagram Buat Tiket Permintaan Pindah Praktikum menunjukan langkah-langkah mahasiswa untuk membuat tiket pindah praktikum.

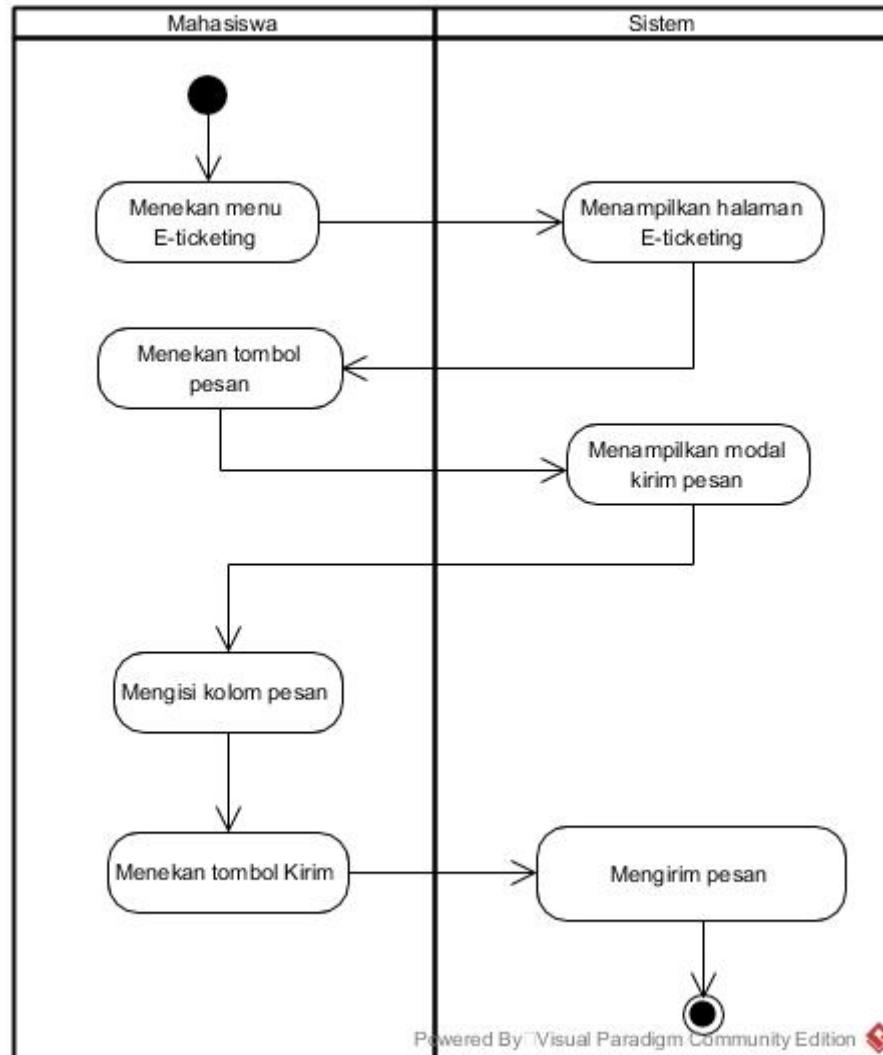
### 60. Swimlane Diagram Mengirim Pesan



Gambar 3. 62 Swimlane Diagram Mengirim Pesan

Keterangan : Berdasarkan Gambar 3.62 *Swimlane Diagram Mengirim Pesan* menunjukkan langkah-langkah mahasiswa untuk mengirim pesan ke admin.

### 61. Swimlane Diagram Upload Tugas



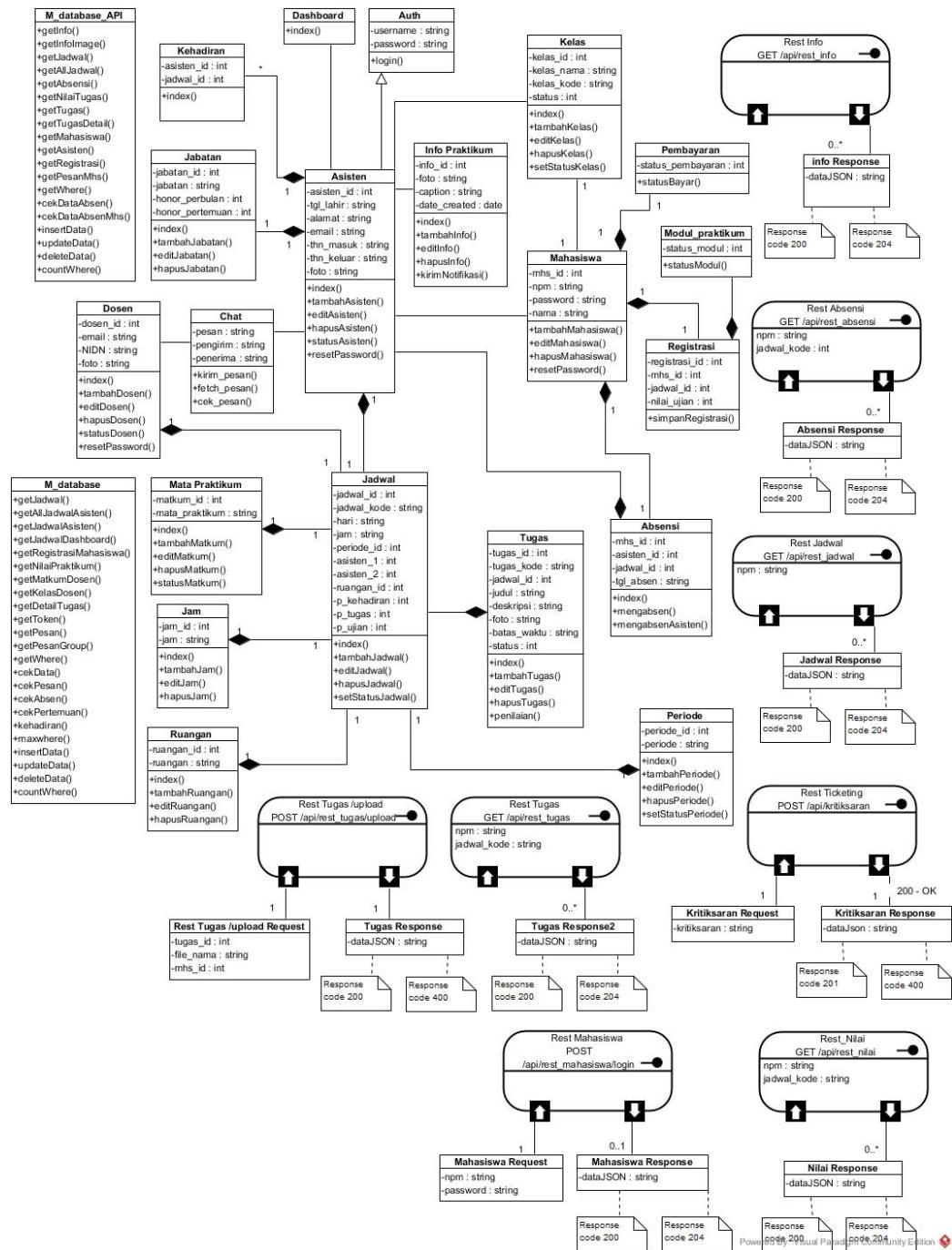
Gambar 3. 63 *Swimlane Diagram Upload Tugas*

Keterangan : Berdasarkan Gambar 3.63 *Swimlane Diagram Upload Tugas* menunjukan langkah-langkah mahasiswa untuk mengupload tugas praktikum.

#### 3.2.3. Class Diagram

Berikut ini merupakan *class diagram* dari Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika:

## 1. Class Diagram Aplikasi Server

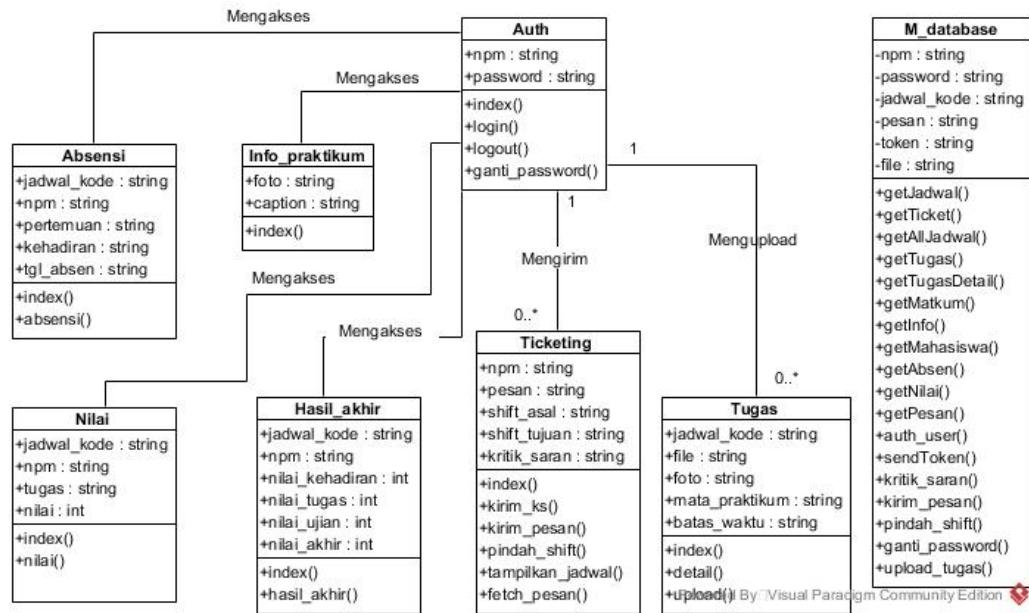


Gambar 3. 64 Class Diagram Sistem Informasi Kegiatan Praktikum (Server)

Berdasarkan Gambar 3.64 *Class Diagram* aplikasi sistem informasi kegiatan praktikum server terdapat 24 *class*. 7 diantaranya merupakan *class* yang

merepresentasikan *Rest Service Class* yaitu Rest Info, Rest Jadwal, Rest Absensi, Rest Nilai, Rest Mahasiswa, Rest Kritiksaran dan Rest Tugas.

## 2. Class Diagram Aplikasi Client



Gambar 3. 65 Class Diagram Sistem Informasi Kegiatan Praktikum (Client)

Berdasarkan Gambar 3.65 *Class Diagram* aplikasi sistem informasi kegiatan praktikum client terdapat 8 *class*.

### **3.2.4. Sequence Diagram**

*Sequence Diagram* merupakan bentuk kedua dari behavioral model. *Sequence Diagram* menunjukkan bagaimana sebuah kejadian (*event*) menyebabkan transisi dari objek ke objek. Berikut ini merupakan *Sequence Diagram* Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika :

1. *Sequence Diagram Login (Server)*

*Sequence Diagram Login (Server)* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat *user* akan *login* ke dalam aplikasi. *Sequence Diagram Login (Server)* dapat dilihat pada **Gambar 3.66**.

2. *Sequence Diagram Absensi Praktikum*

*Sequence Diagram Absensi Praktikum* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat asisten melakukan absensi praktikum di dalam aplikasi. *Sequence Diagram Absensi Praktikum* dapat dilihat pada **Gambar 3.67**.

3. *Sequence Diagram Penilaian Praktikum*

*Sequence Diagram Penilaian Praktikum* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat asisten melakukan penilaian praktikum di dalam aplikasi. *Sequence Diagram Penilaian Praktikum* dapat dilihat pada **Gambar 3.68**.

4. *Sequence Diagram Kelola Tugas*

*Sequence Diagram Kelola Tugas* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat asisten mengelola data tugas praktikum di dalam aplikasi. *Sequence Diagram Kelola Tugas* dapat dilihat pada **Gambar 3.69**.

5. *Sequence Diagram Ubah Status Modul Praktikum*

*Sequence Diagram Ubah Status Modul Praktikum* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat asisten mengubah status

pengambilan modul praktikum. *Sequence Diagram Ubah Status Modul Praktikum* dapat dilihat pada **Gambar 3.70**.

6. *Sequence Diagram Mengirim Pesan*

*Sequence Diagram Mengirim Pesan* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat asisten melakukan perpesanan di dalam aplikasi. *Sequence Diagram Mengirim Pesan* dapat dilihat pada **Gambar 3.71**.

7. *Sequence Diagram Edit Profil*

*Sequence Diagram Edit Profil* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat *user* mengubah data profil akun miliknya. *Sequence Diagram Edit Profil* dapat dilihat pada **Gambar 3.72**.

8. *Sequence Diagram Ganti Password (Server)*

*Sequence Diagram Ganti Password (Server)* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat *user* mengubah *password* akun miliknya. *Sequence Diagram Ganti Password (Server)* dapat dilihat pada **Gambar 3.73**.

9. *Sequence Diagram Kelola Periode*

*Sequence Diagram Kelola Periode* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data periode praktikum di dalam aplikasi. *Sequence Diagram Kelola Periode* dapat dilihat pada **Gambar 3.74**.

10. *Sequence Diagram Kelola Jadwal*

*Sequence Diagram Kelola Jadwal* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data jadwal praktikum di dalam aplikasi. *Sequence Diagram Kelola Jadwal* dapat dilihat pada **Gambar 3.75**.

#### 11. *Sequence Diagram Kelola Kelas*

*Sequence Diagram Kelola Kelas* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data kelas di dalam aplikasi.

*Sequence Diagram Kelola Kelas* dapat dilihat pada **Gambar 3.76**.

#### 12. *Sequence Diagram Kelola Mata Praktikum*

*Sequence Diagram Kelola Mata Praktikum* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data mata praktikum di dalam aplikasi. *Sequence Diagram Kelola Mata Praktikum* dapat dilihat pada **Gambar 3.77**.

#### 13. *Sequence Diagram Kelola Ruangan*

*Sequence Diagram Kelola Ruangan* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data mata praktikum di dalam aplikasi. *Sequence Diagram Kelola Ruangan* dapat dilihat pada **Gambar 3.78**.

#### 14. *Sequence Diagram Kelola Jam*

*Sequence Diagram Kelola Jam* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data jam praktikum di dalam aplikasi. *Sequence Diagram Kelola Jam* dapat dilihat pada **Gambar 3.79**.

#### 15. *Sequence Diagram Kelola Jabatan*

*Sequence Diagram Kelola Jabatan* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data jabatan di dalam aplikasi. *Sequence Diagram Kelola Jabatan* dapat dilihat pada **Gambar 3.80**.

16. *Sequence Diagram Kelola Asisten*

*Sequence Diagram Kelola Asisten* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data asisten di dalam aplikasi. *Sequence Diagram Kelola Asisten* dapat dilihat pada **Gambar 3.81.**

17. *Sequence Diagram Kelola Dosen*

*Sequence Diagram Kelola Dosen* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data dosen di dalam aplikasi. *Sequence Diagram Kelola Dosen* dapat dilihat pada **Gambar 3.82.**

18. *Sequence Diagram Kelola Mahasiswa*

*Sequence Diagram Kelola Mahasiswa* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data mahasiswa di dalam aplikasi. *Sequence Diagram Kelola Mahasiswa* dapat dilihat pada **Gambar 3.83.**

19. *Sequence Diagram Kelola Info Praktikum*

*Sequence Diagram Kelola Info Praktikum* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data info praktikum di dalam aplikasi. *Sequence Diagram Kelola Info Praktikum* dapat dilihat pada **Gambar 3.84.**

20. *Sequence Diagram Kelola E-Ticketing*

*Sequence Diagram Kelola E-Ticketing* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data tiket permintaan pindah shift dan membalas pesan mahasiswa. *Sequence Diagram Kelola E-Ticketing* dapat dilihat pada **Gambar 3.85.**

#### 21. *Sequence Diagram Registrasi Praktikum*

*Sequence Diagram Registrasi Praktikum* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin meregistrasi mahasiswa ke dalam jadwal praktikum. *Sequence Diagram Registrasi Praktikum* dapat dilihat pada **Gambar 3.86**.

#### 22. *Sequence Diagram Kelola Asisten*

*Sequence Diagram Kelola Asisten* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola data asisten di dalam aplikasi. *Sequence Diagram Kelola Asisten* dapat dilihat pada **Gambar 3.87**.

#### 23. *Sequence Diagram Pindah Praktikum*

*Sequence Diagram Kelola Pindah Praktikum* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat admin mengelola mahasiswa yang ingin pindah praktikum. *Sequence Diagram Pindah Praktikum* dapat dilihat pada **Gambar 3.88**.

#### 24. *Sequence Diagram Login (Client)*

*Sequence Diagram Login (Client)* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat mahasiswa *login* ke dalam aplikasi. *Sequence Diagram Login (Client)* dapat dilihat pada **Gambar 3.89**.

#### 25. *Sequence Diagram Ganti Password (Client)*

*Sequence Diagram Ganti Password (Client)* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat mahasiswa mengganti *password* akun. *Sequence Diagram Ganti Password (Client)* dapat dilihat pada **Gambar 3.90**.

#### 26. *Sequence Diagram Cek Nilai*

*Sequence Diagram Cek Nilai* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat mahasiswa melakukan pengecekan data nilai praktikum. *Sequence Diagram Cek Nilai* dapat dilihat pada **Gambar 3.91**.

#### 27. *Sequence Diagram Cek Absensi*

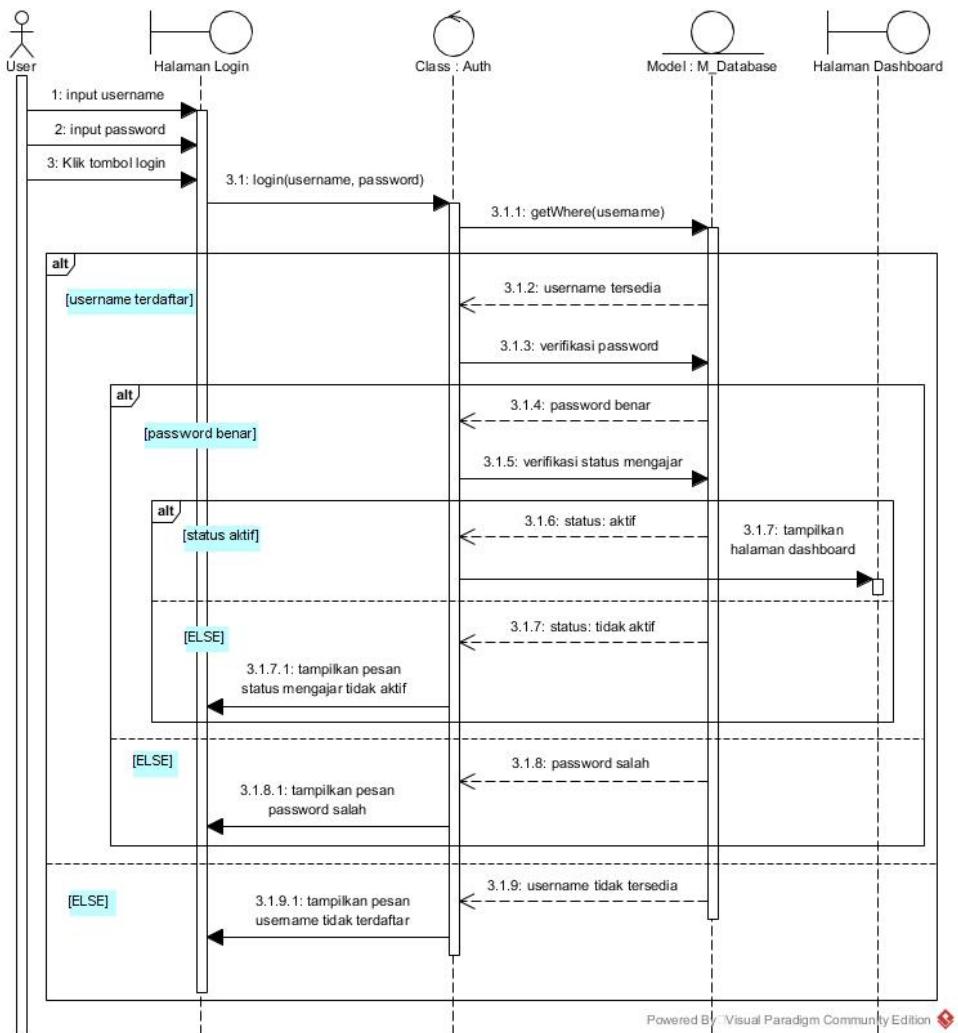
*Sequence Diagram Cek Absensi* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat mahasiswa melakukan pengecekan data absensi praktikum. *Sequence Diagram Cek Absensi* dapat dilihat pada **Gambar 3.92**.

#### 28. *Sequence Diagram Upload Tugas*

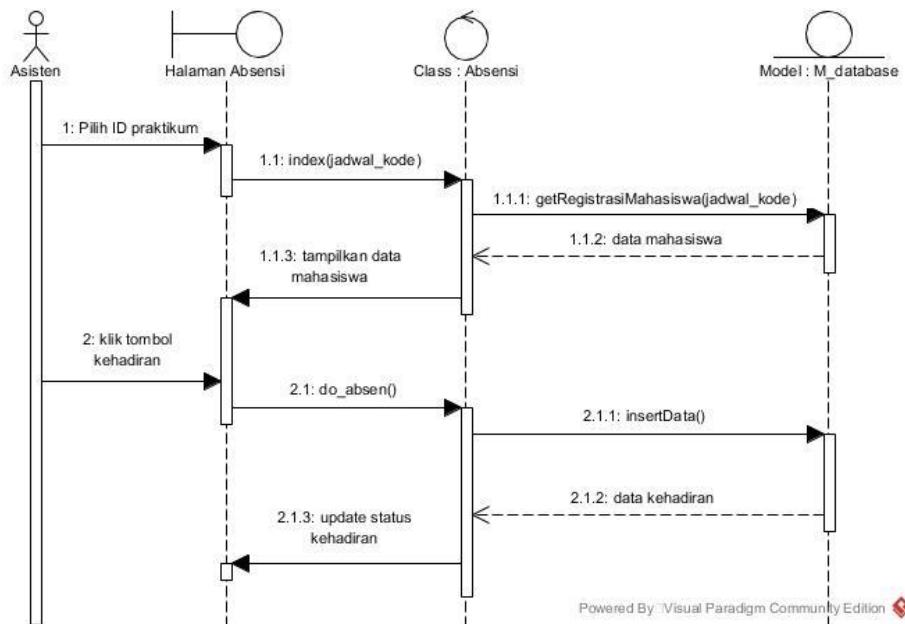
*Sequence Diagram Upload Tugas* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat mahasiswa mengupload tugas praktikum. *Sequence Diagram Upload Tugas* dapat dilihat pada **Gambar 3.93**.

#### 29. *Sequence Diagram E-Ticketing*

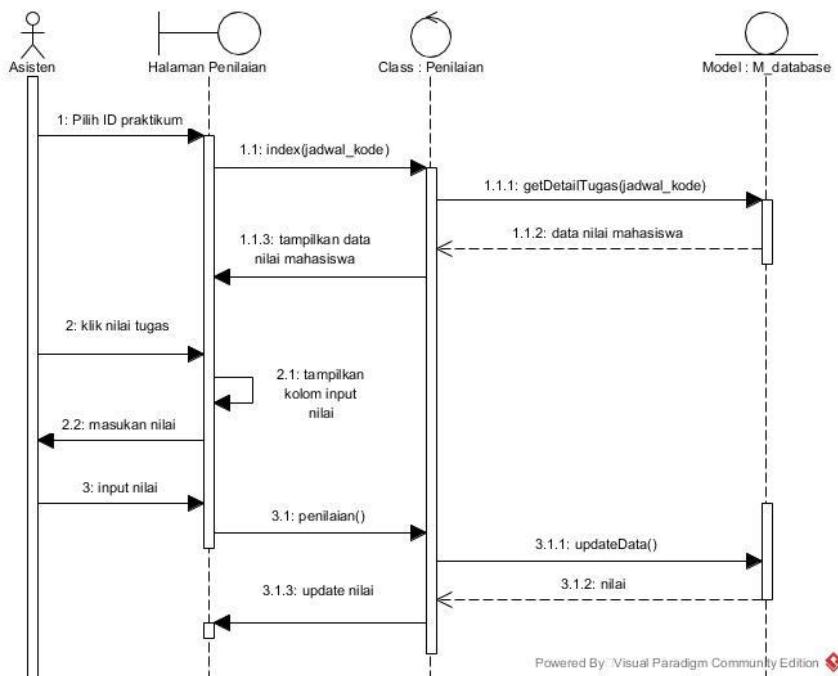
*Sequence Diagram E-Ticketing* merupakan rangkaian pesan yang terdapat di dalam sistem pada saat mahasiswa melakukan *e-ticketing* yaitu membuat tiket pindah praktikum, mengirim kritik & saran, dan mengirim pesan. *Sequence Diagram E-ticketing* dapat dilihat pada **Gambar 3.94**.



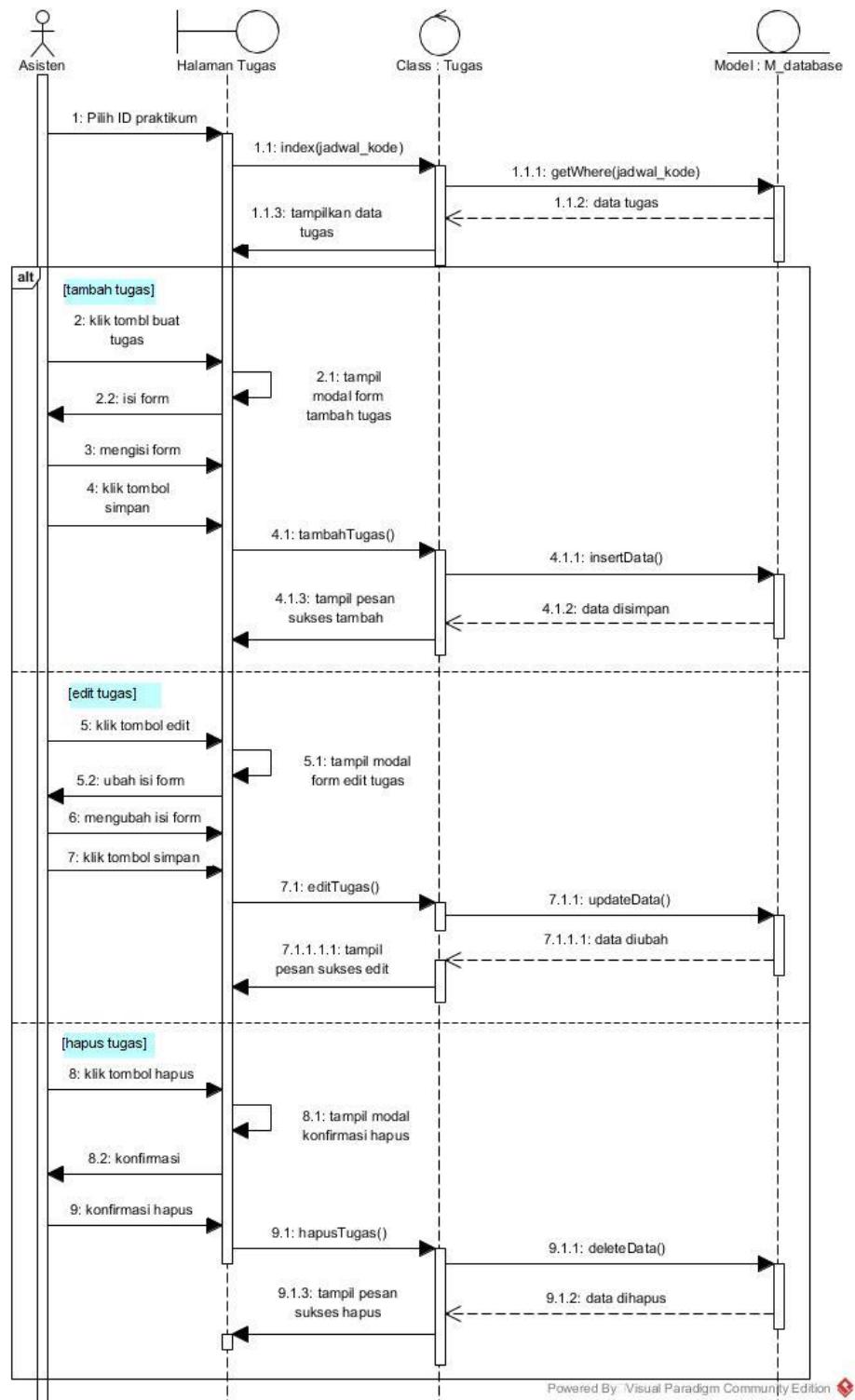
Gambar 3. 66 Sequence Diagram Login (Server)



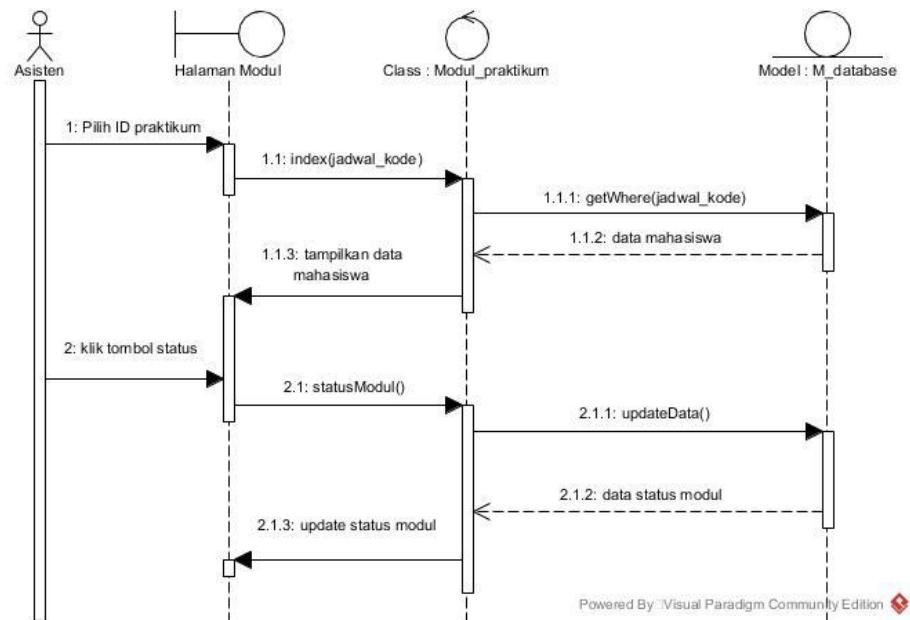
Gambar 3. 67 Sequence Diagram Absensi Praktikum



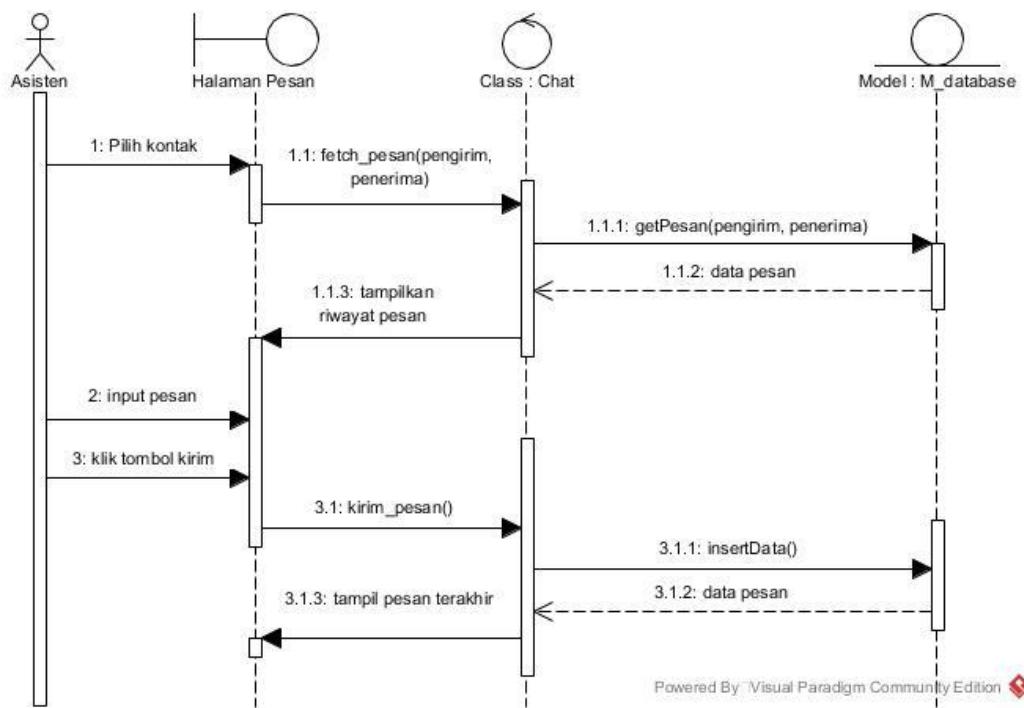
Gambar 3. 68 Sequence Diagram Penilaian Praktikum



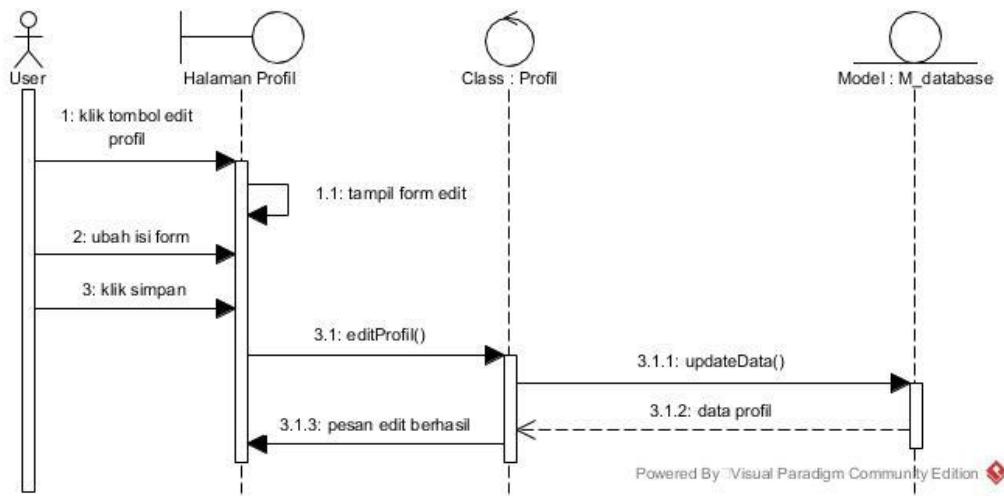
Gambar 3. 69 Sequence Diagram Kelola Tugas



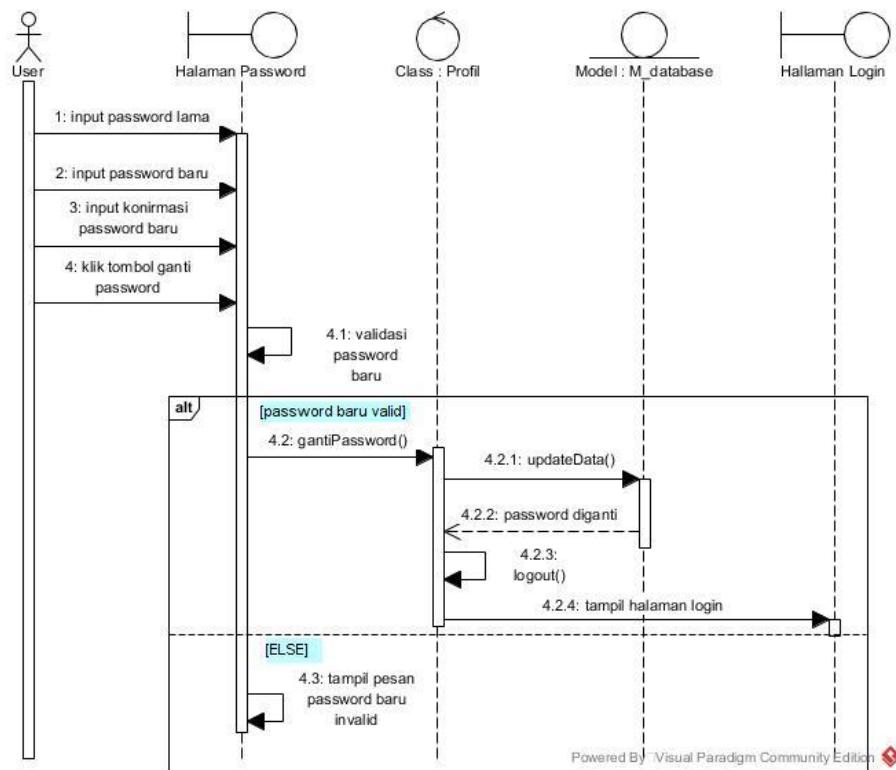
Gambar 3. 70 Sequence Diagram Ubah Status Modul Praktikum



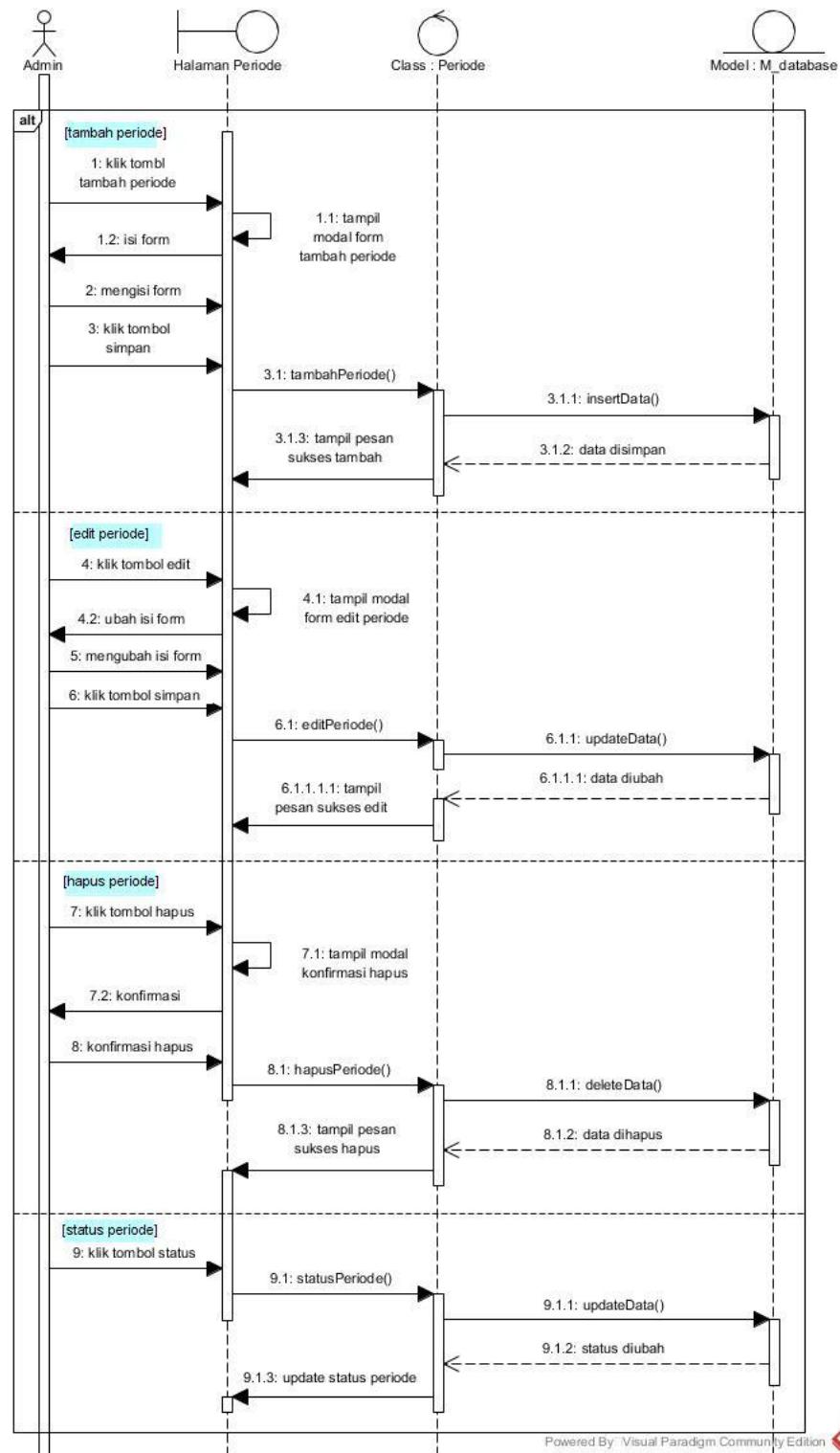
Gambar 3. 71 Sequence Diagram Mengirim Pesan



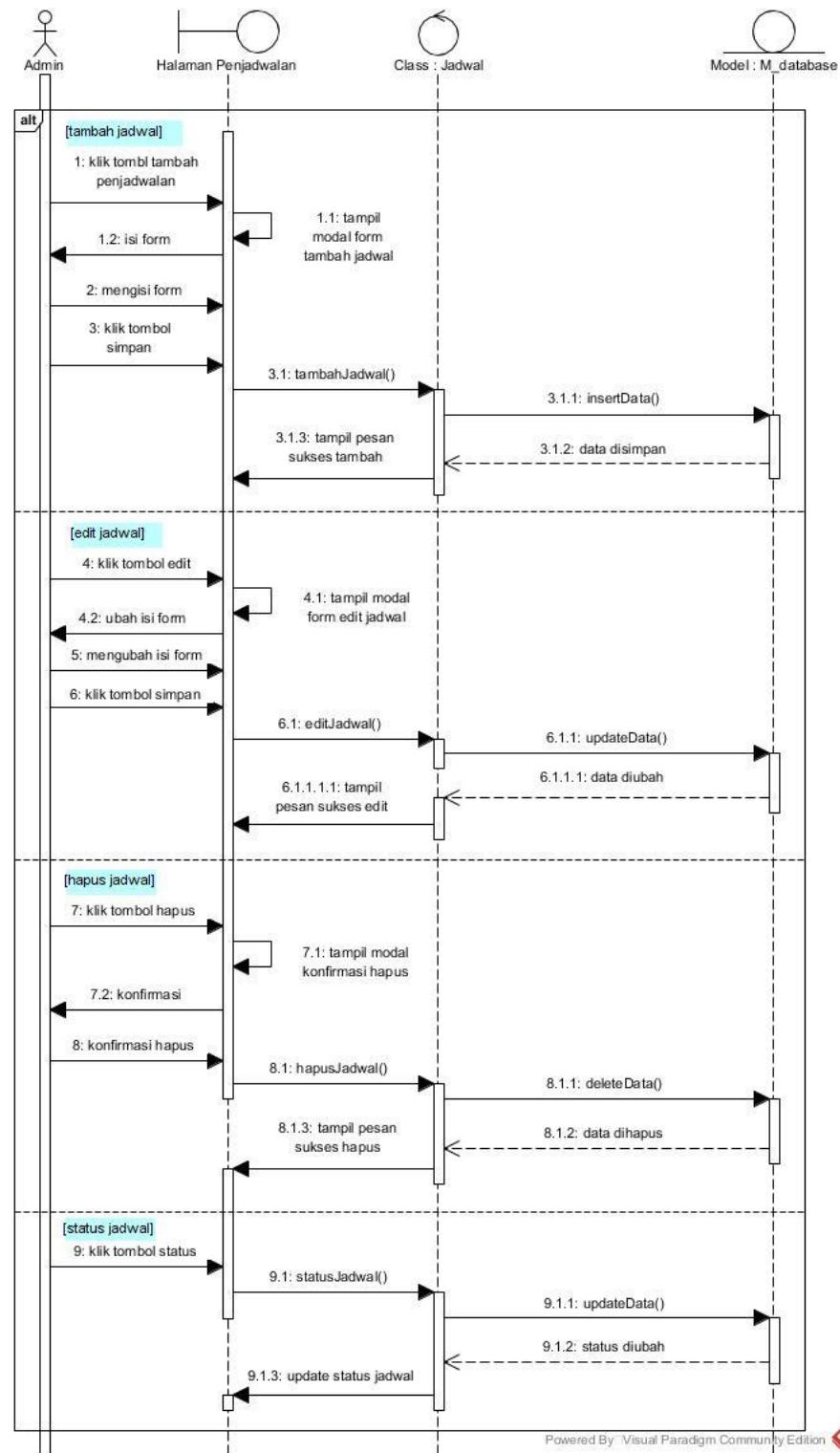
Gambar 3. 72 Sequence Diagram Edit Profil



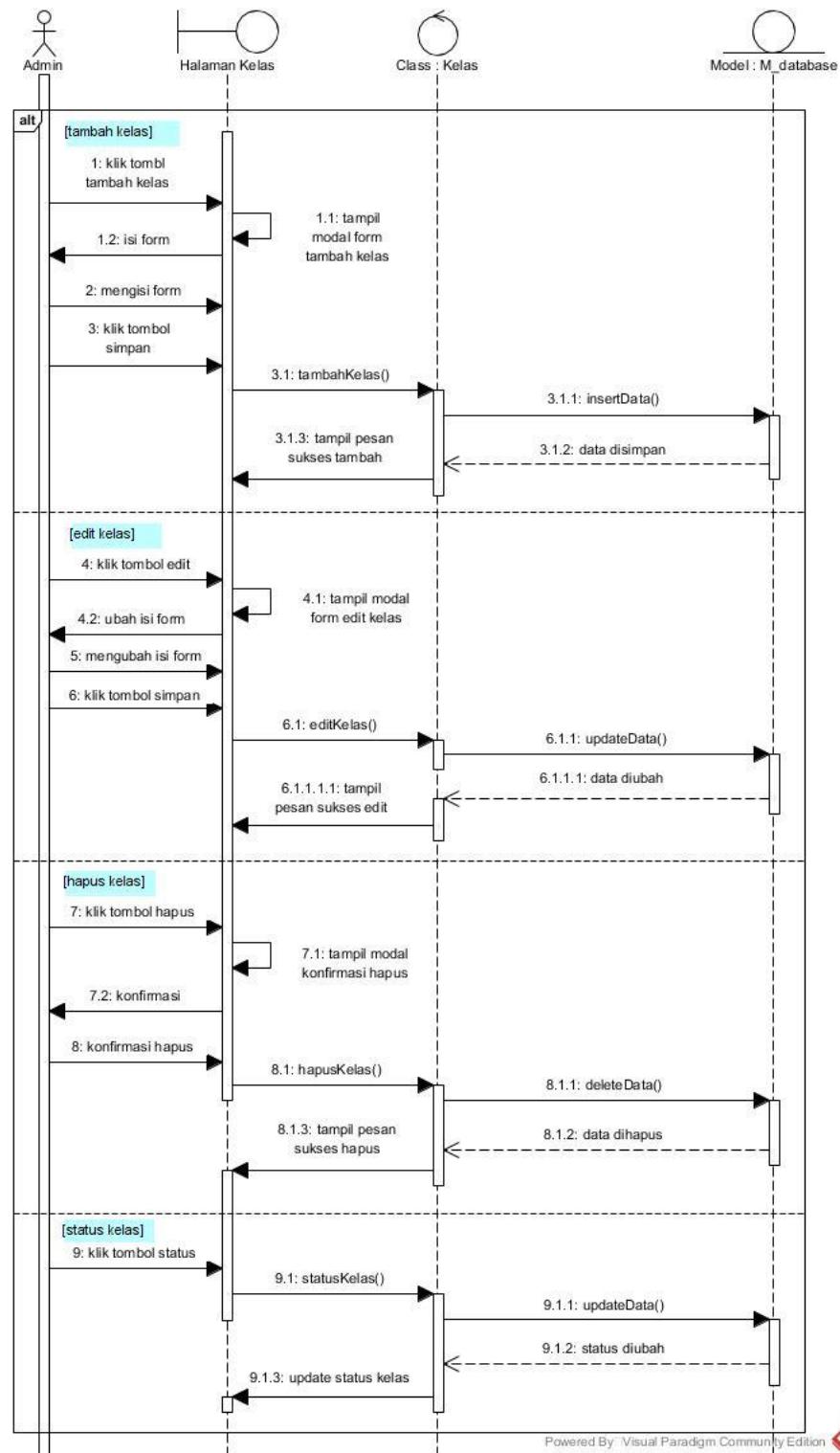
Gambar 3. 73 Sequence Diagram Ganti Password (Server)



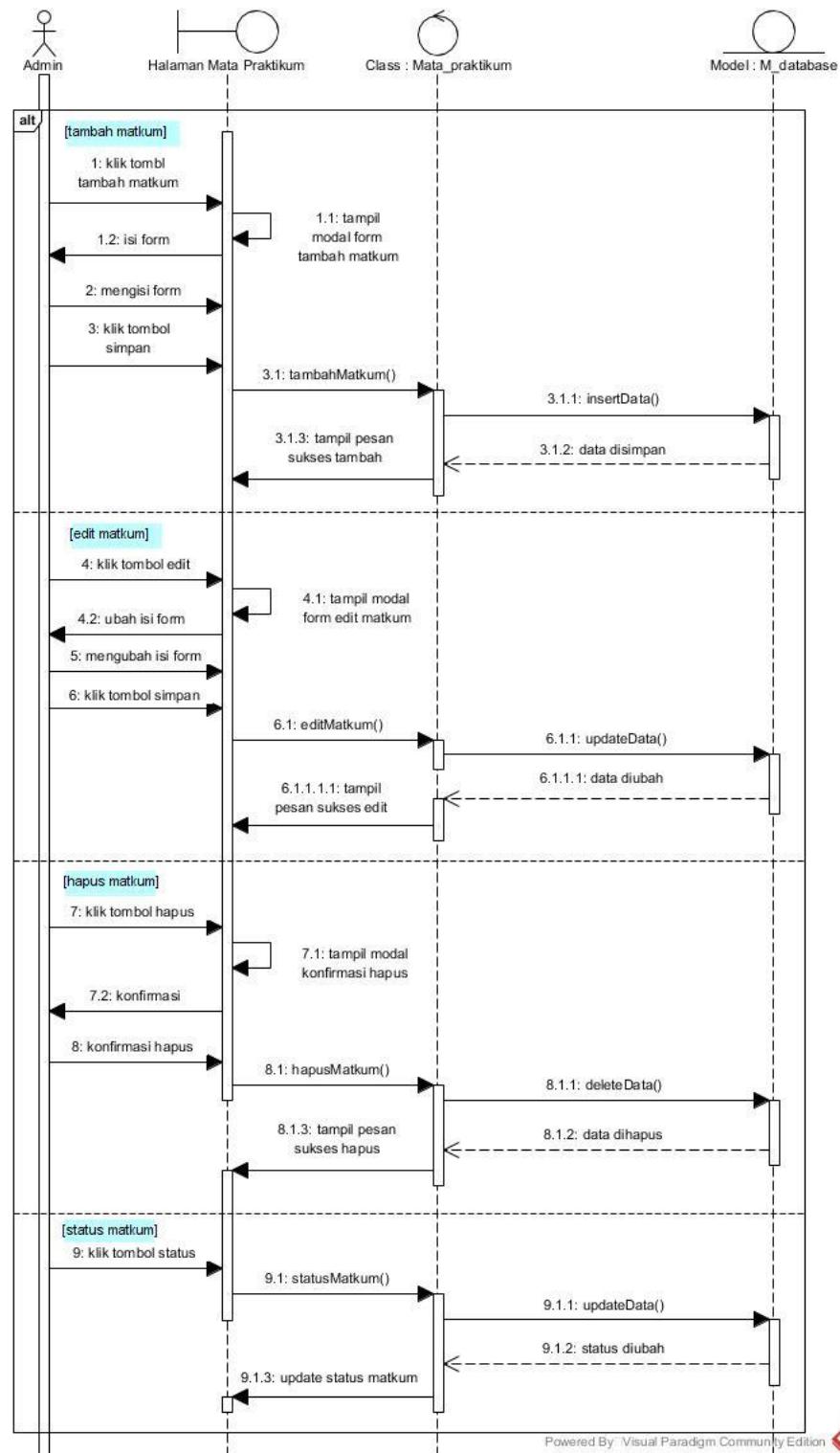
Gambar 3. 74 Sequence Diagram Kelola Periode



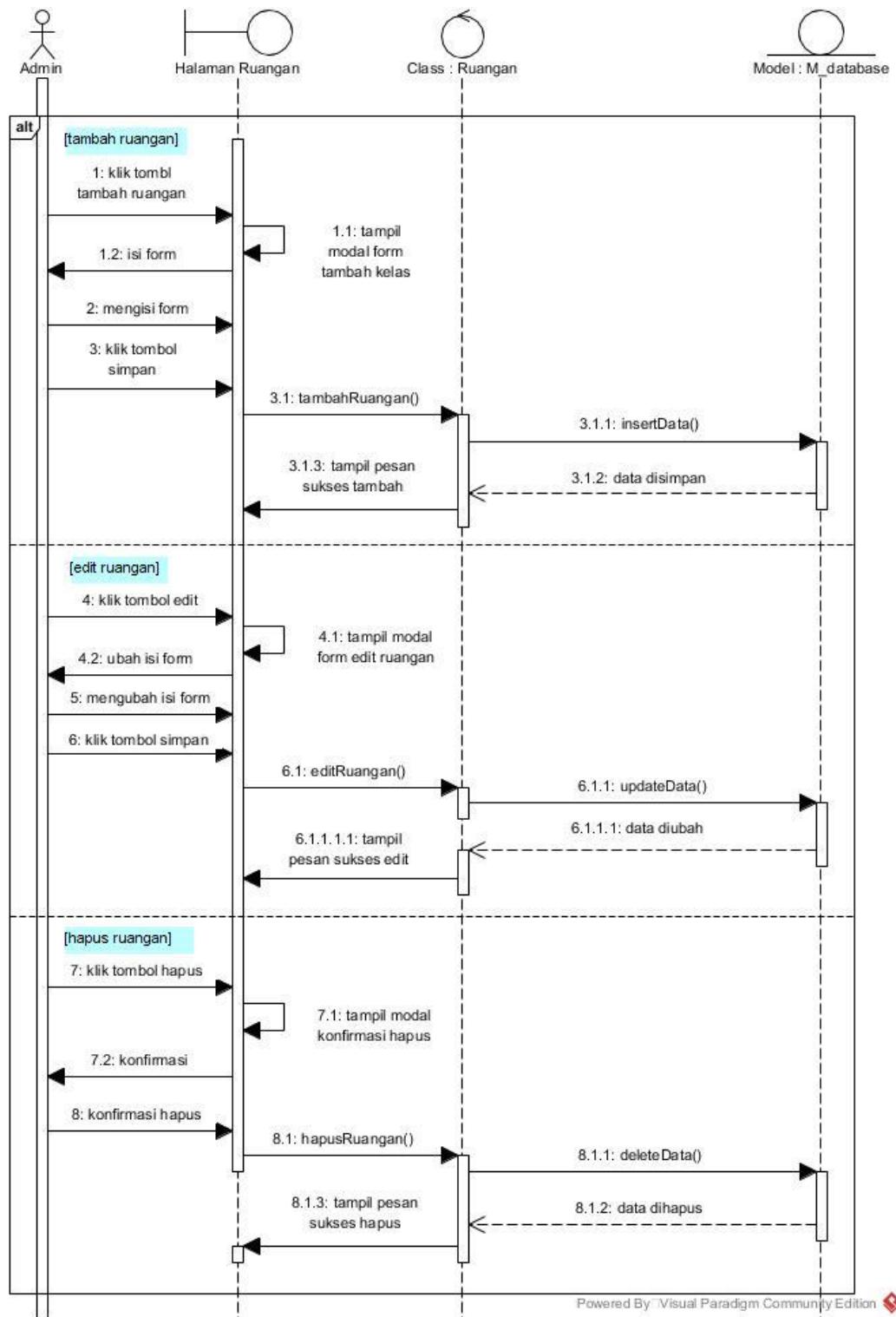
Gambar 3. 75 Sequence Diagram Kelola Jadwal



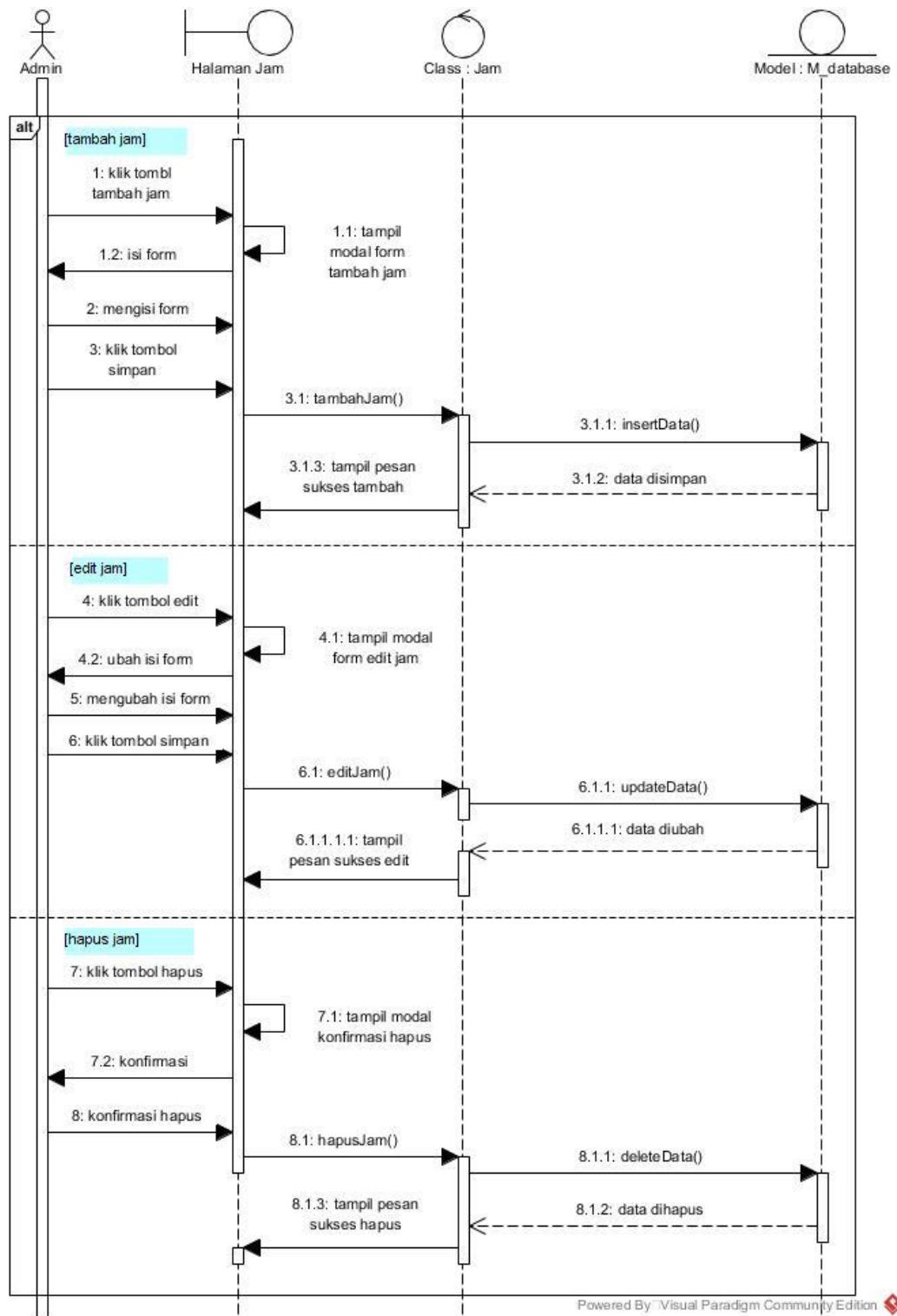
Gambar 3. 76 Sequence Diagram Kelola Kelas



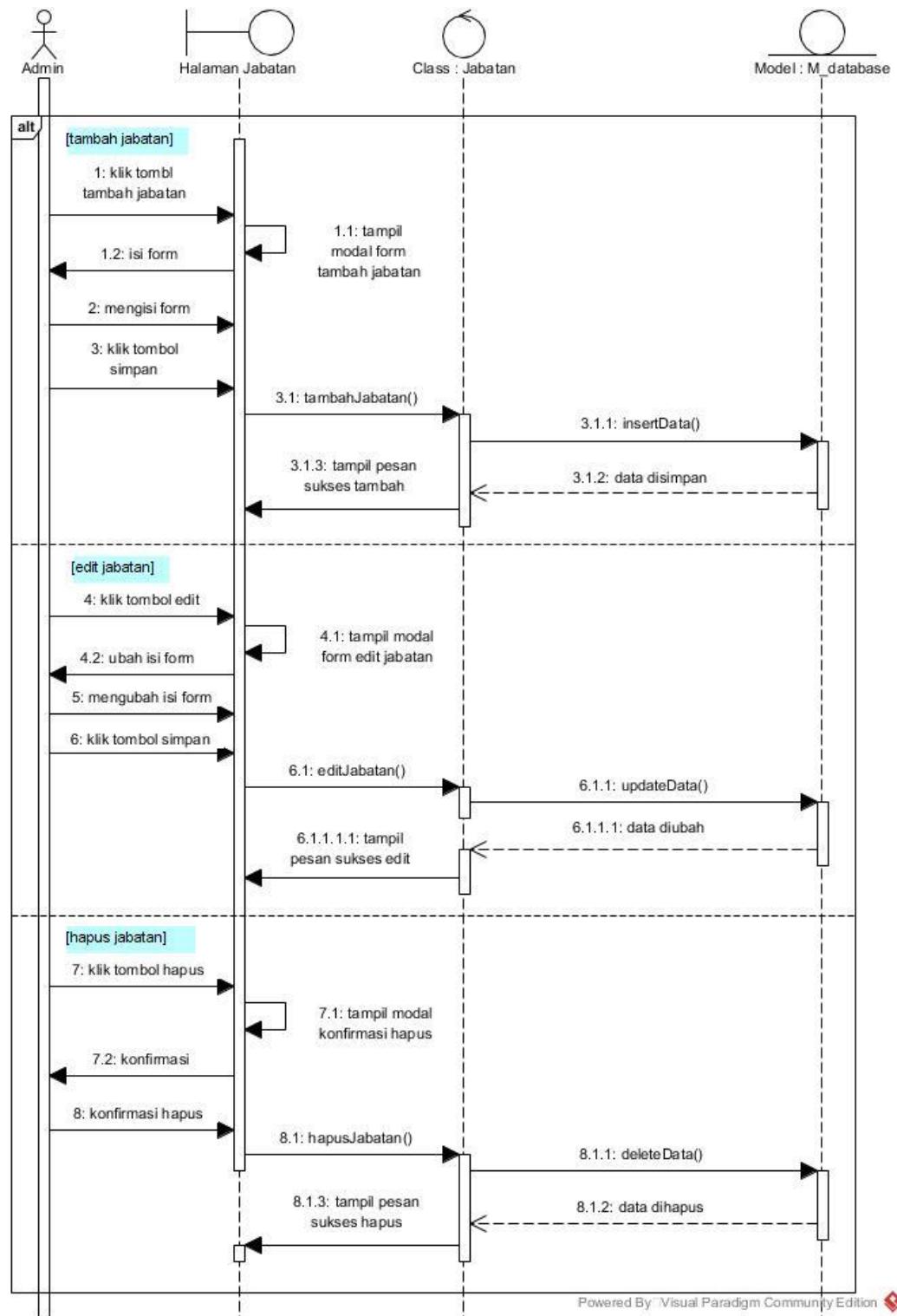
Gambar 3. 77 Sequence Diagram Kelola Mata Praktikum



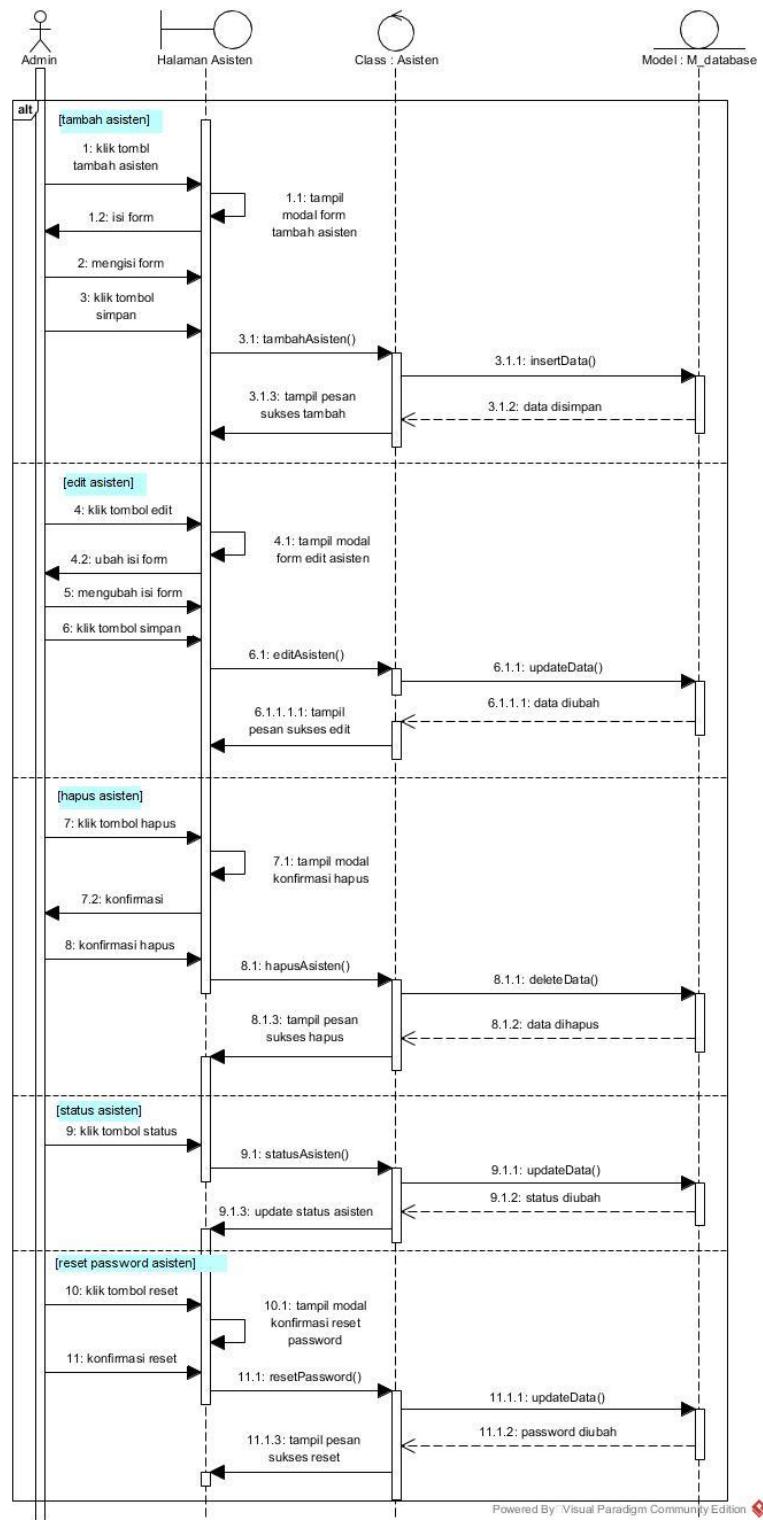
Gambar 3. 78 Sequence Diagram Kelola Ruangan



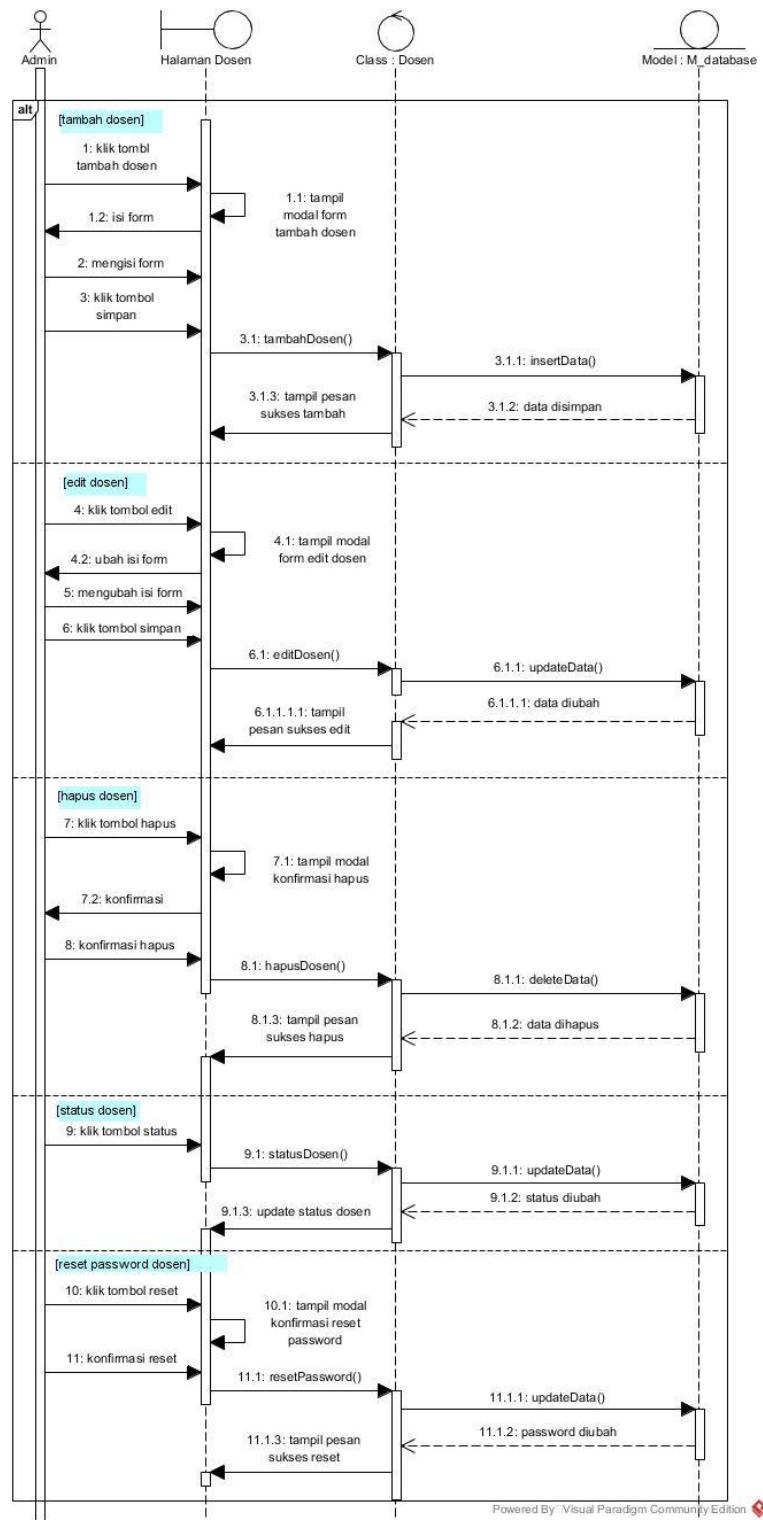
Gambar 3. 79 Sequence Diagram Kelola Jam



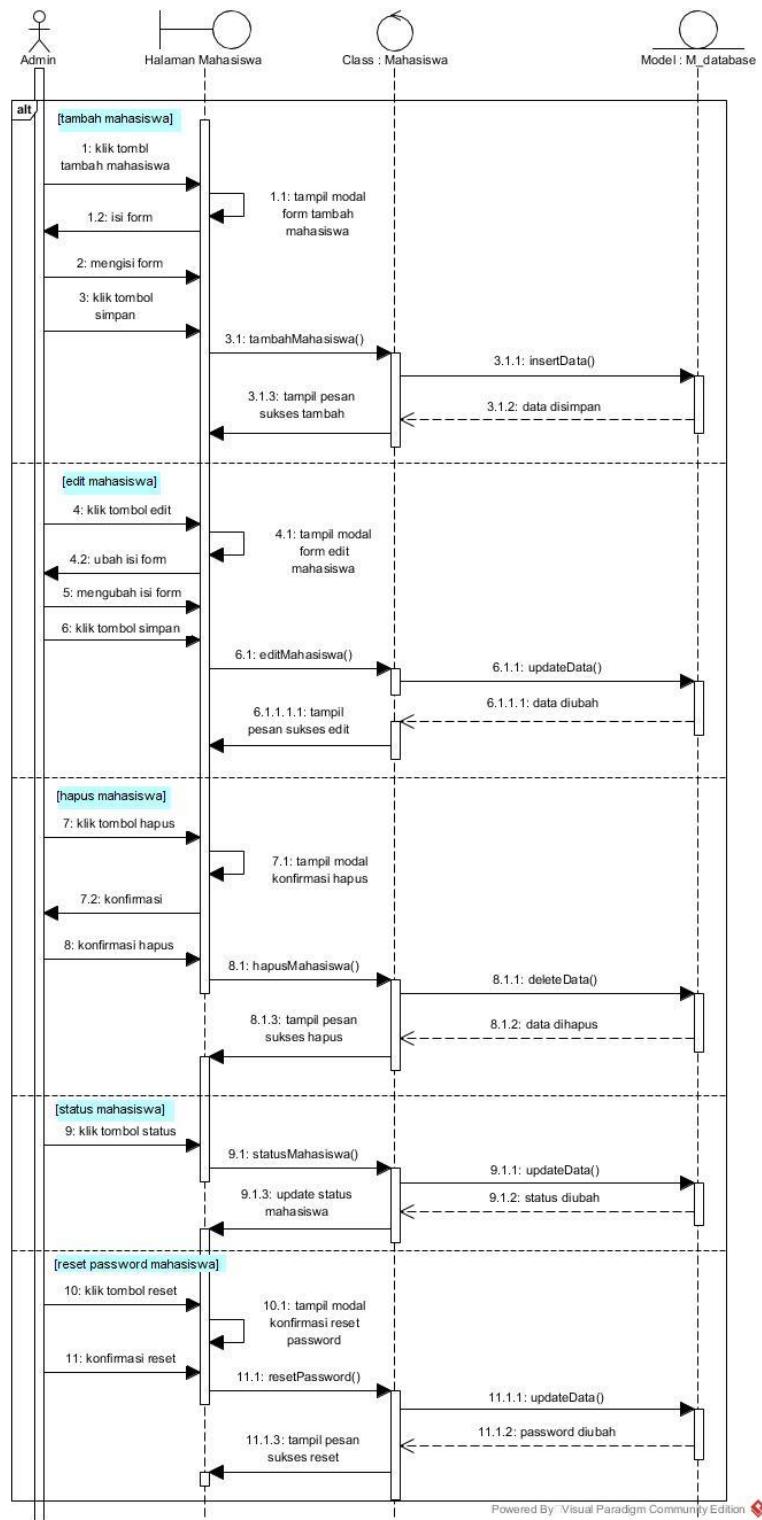
Gambar 3. 80 Sequence Diagram Kelola Jabatan



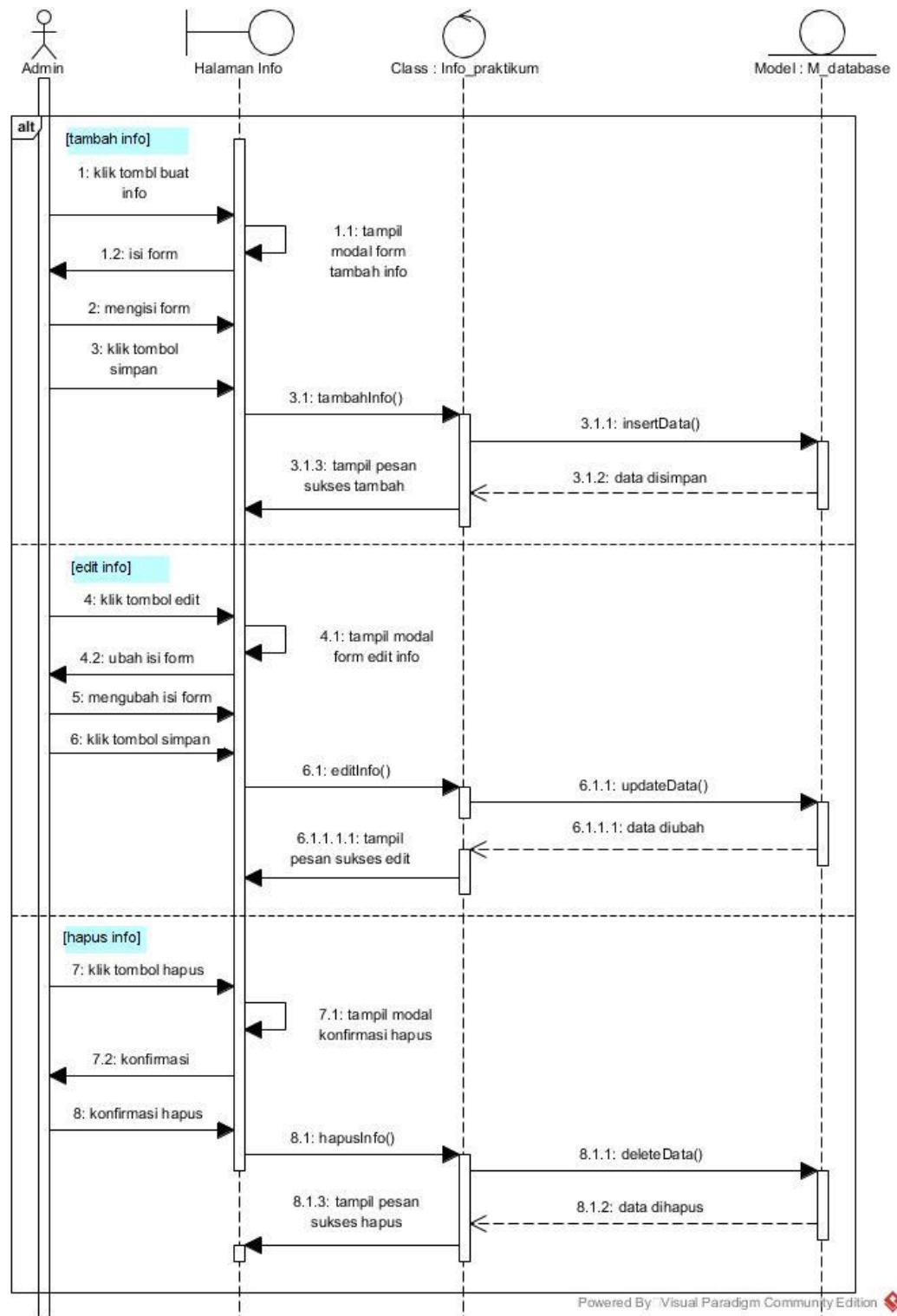
Gambar 3. 81 Sequence Diagram Kelola Asisten



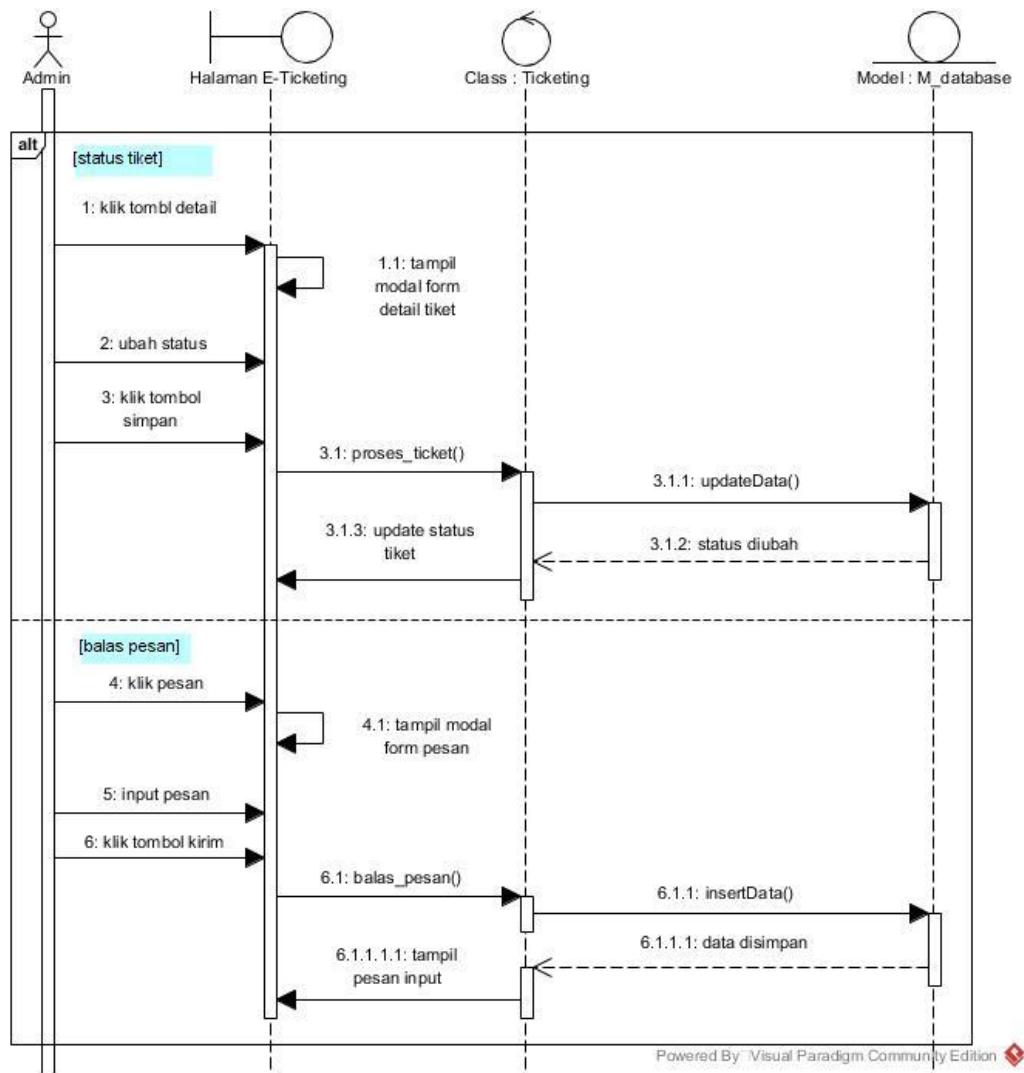
Gambar 3. 82 Sequence Diagram Kelola Dosen



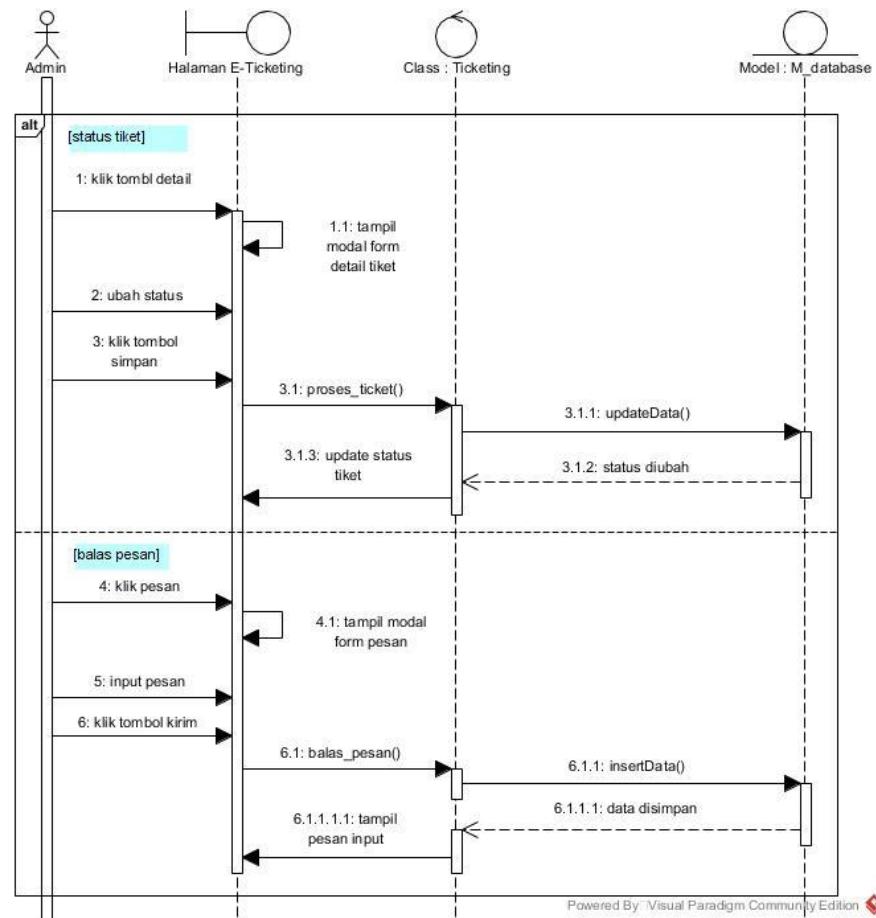
Gambar 3. 83 Sequence Diagram Kelola Mahasiswa



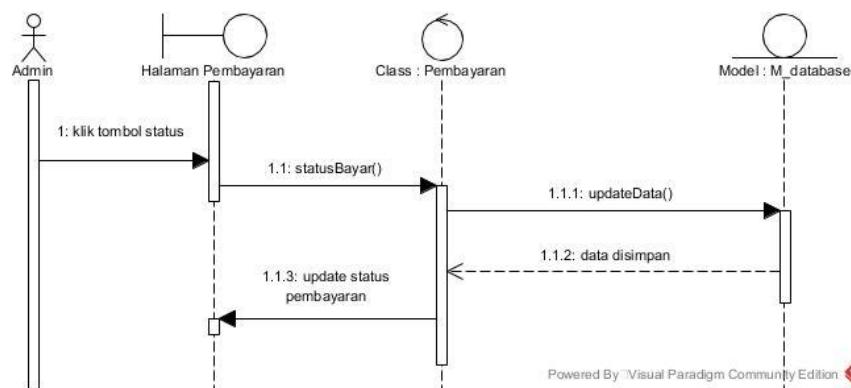
Gambar 3. 84 Sequence Diagram Kelola Info Praktikum



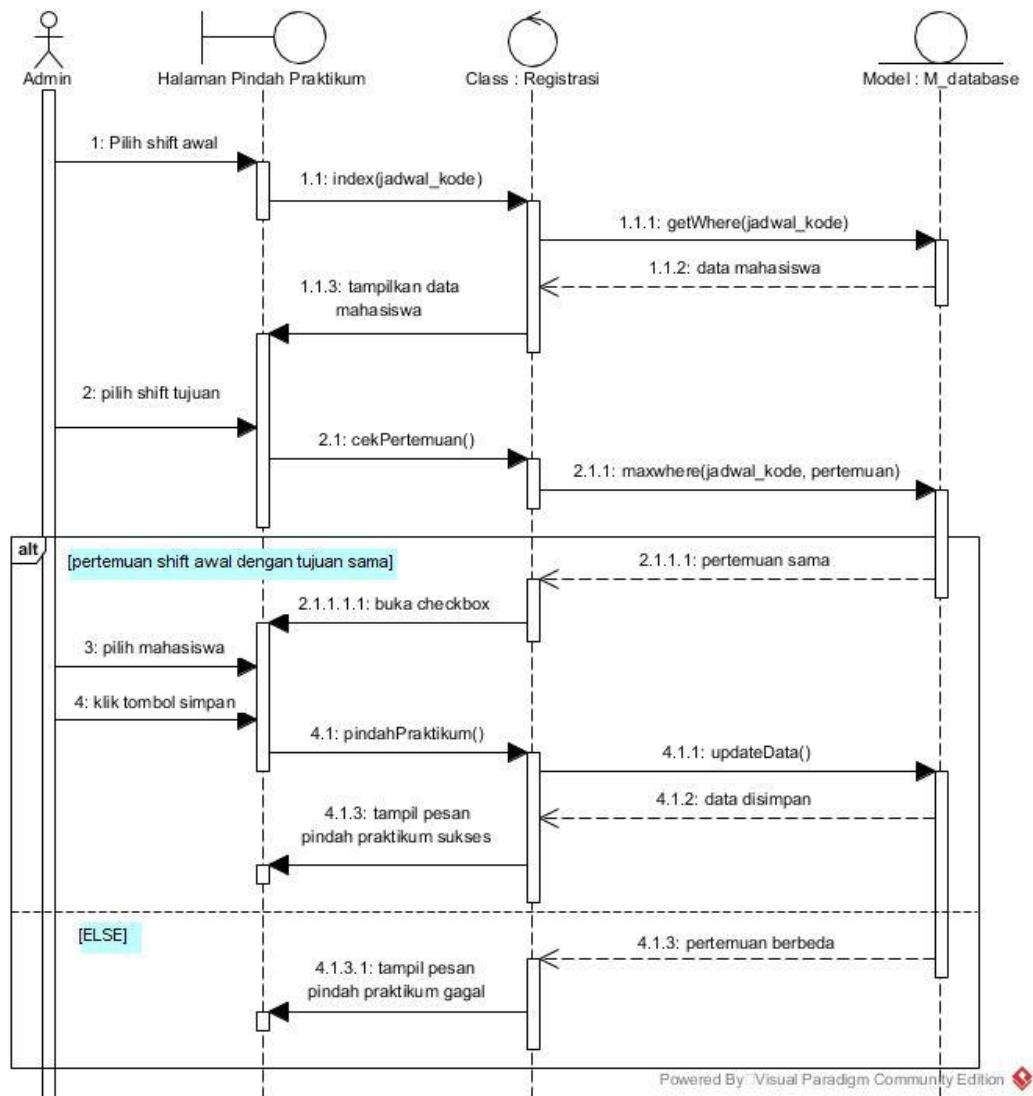
Gambar 3. 85 Sequence Diagram Kelola E-Ticketing



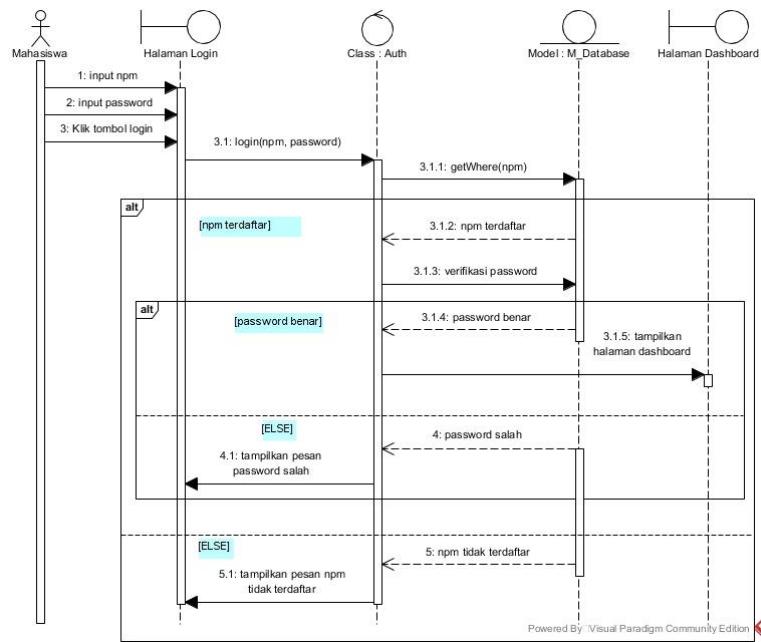
Gambar 3. 86 Sequence Diagram Registrasi Praktikum



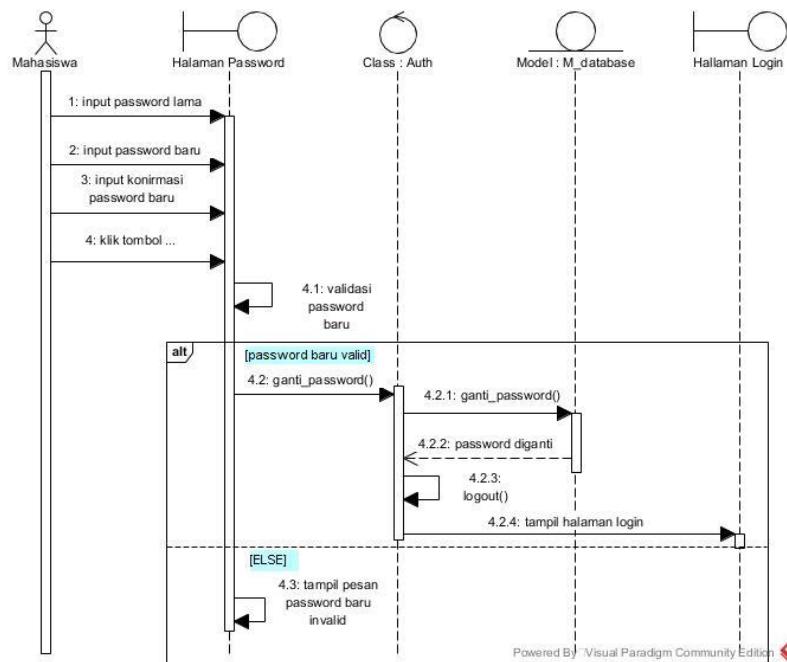
Gambar 3. 87 Sequence Diagram Ubah Status Pembayaran Praktikum



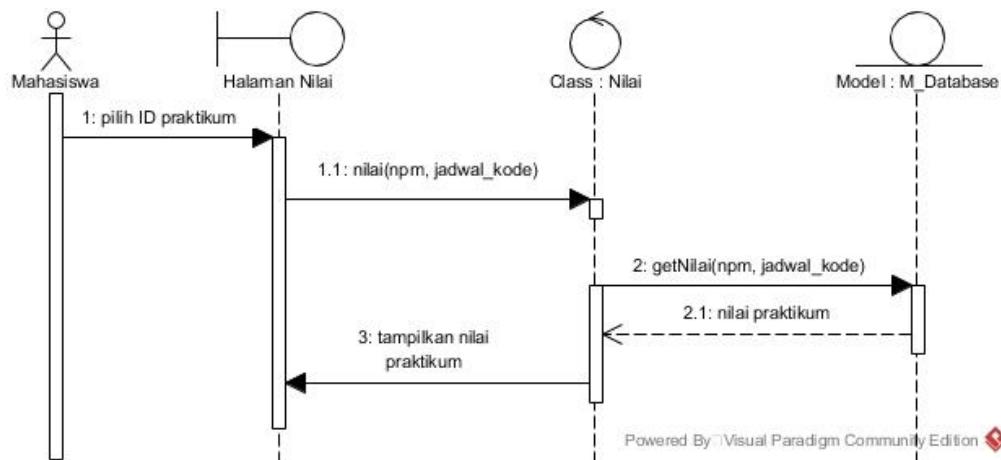
Gambar 3. 88 Sequence Diagram Pindah Praktikum



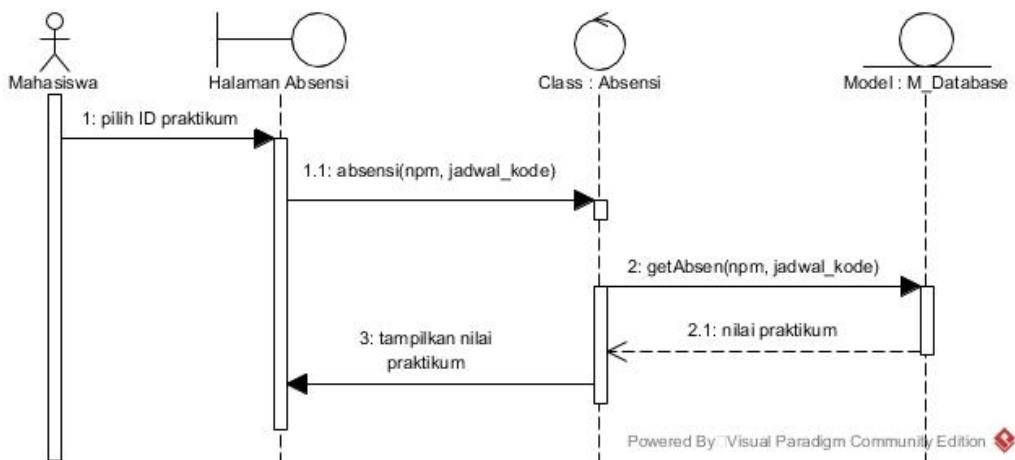
Gambar 3. 89 Sequence Diagram Login (Client)



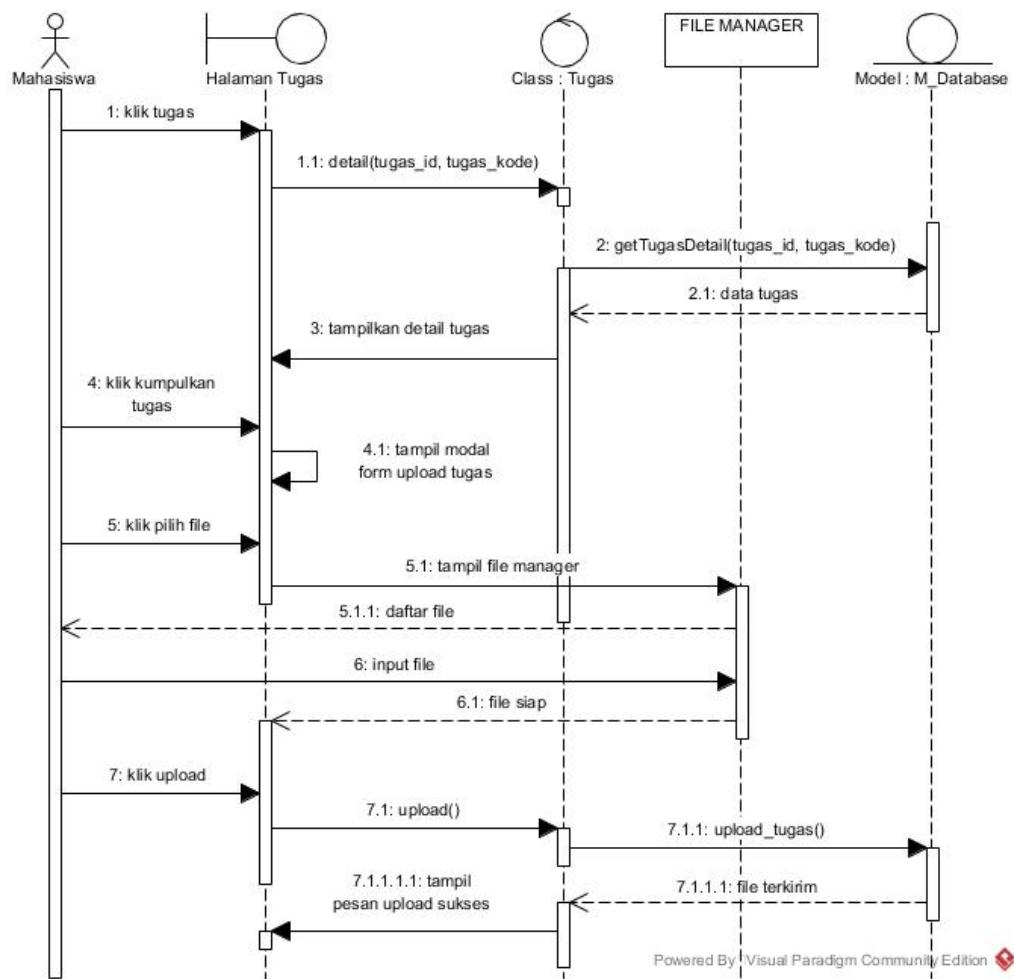
Gambar 3. 90 Sequence Diagram Ganti Password (Client)



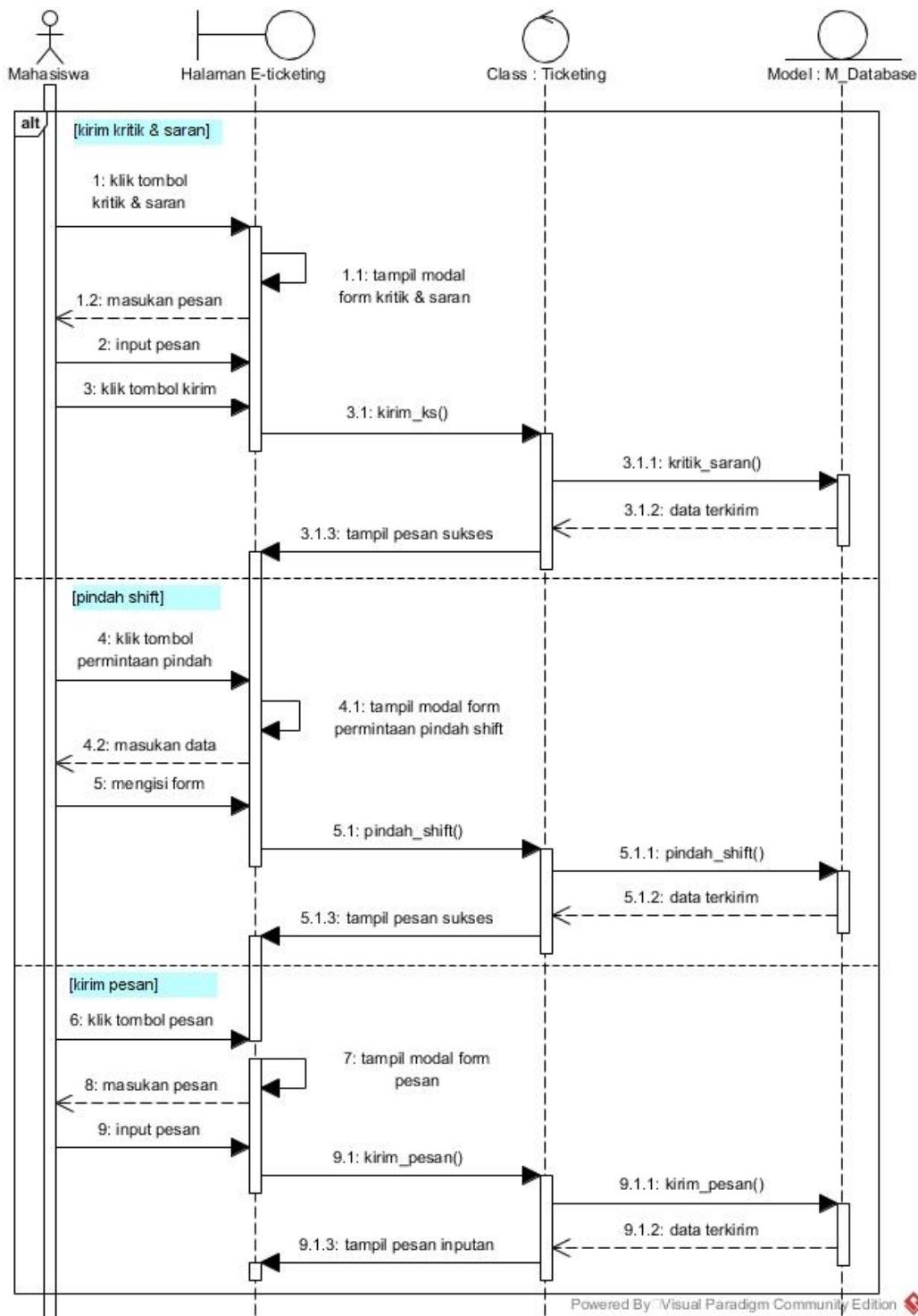
Gambar 3. 91 Sequence Diagram Cek Nilai



Gambar 3. 92 Sequence Diagram Cek Absensi



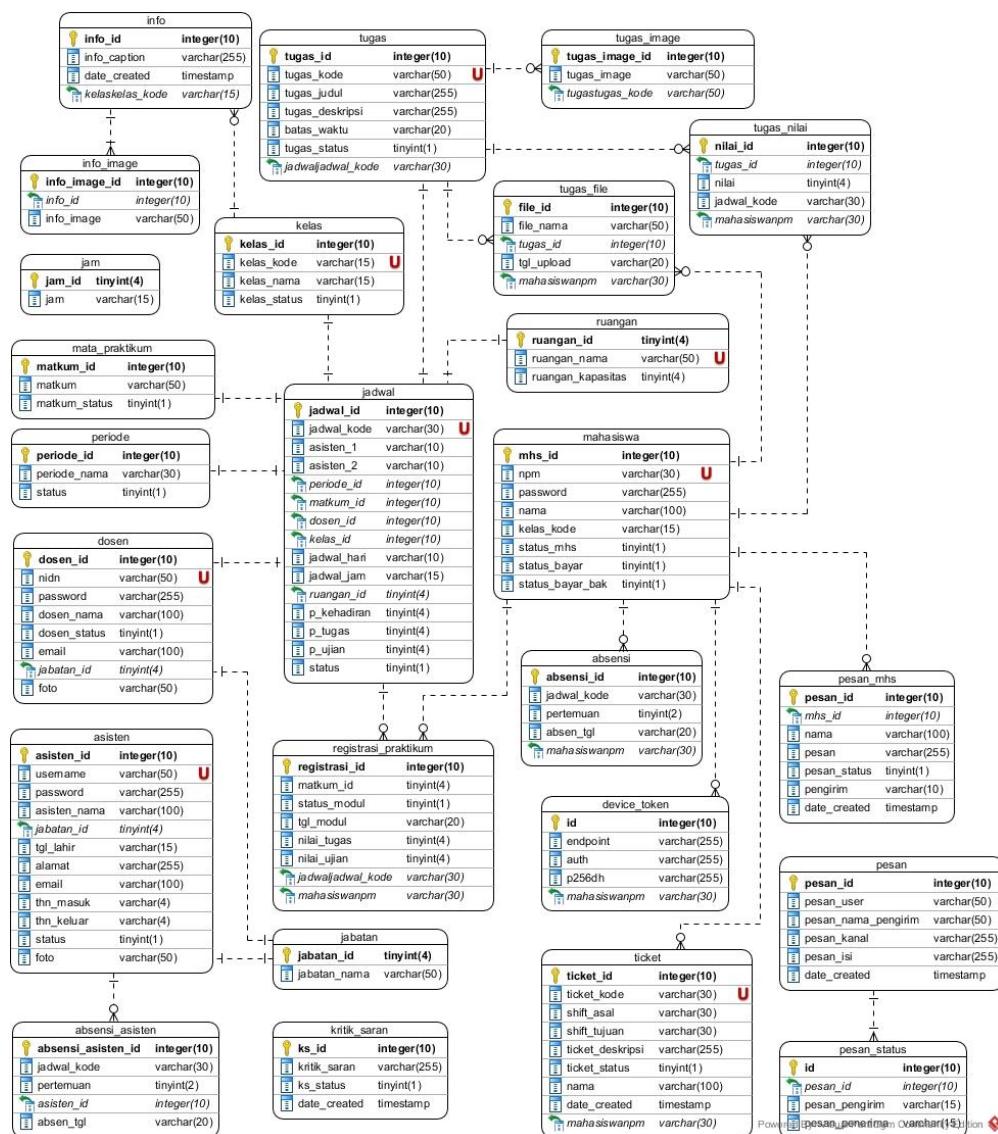
Gambar 3. 93 Sequence Diagram Upload Tugas



Gambar 3. 94 Sequence Diagram E-Ticketing

### 3.2.5. Entity Relationship Diagram

*Entity Relationship Diagram* adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas (*entity*), atribut dan hubungan antar entitas (*relationship*). Suatu entitas bersifat unik dan memiliki atribut sebagai pembeda dengan entitas lainnya. Berikut merupakan *Entity Realationship Diagram* dari Sistem Informasi Manajemen Kegiatan Praktikum Laboratorium Komputer Teknik informatika :



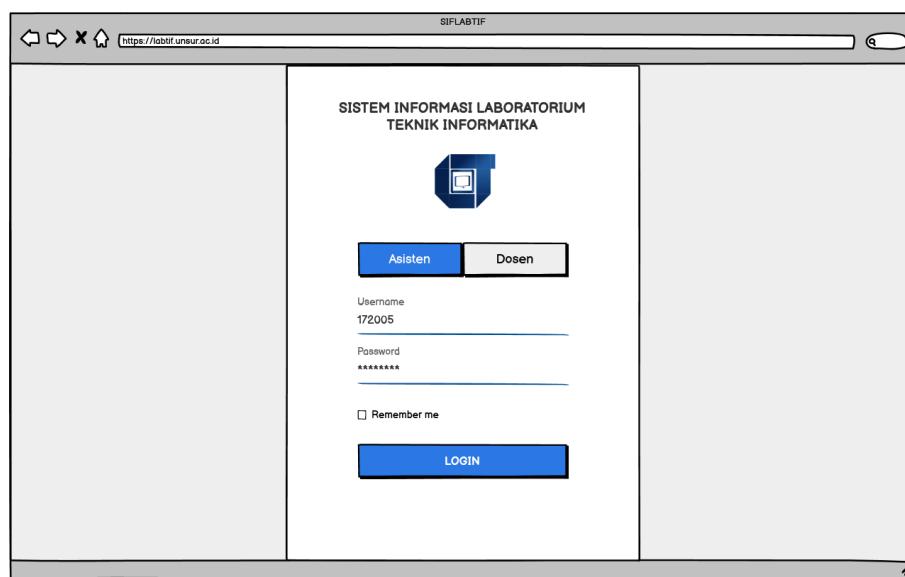
Gambar 3. 95 Entity Relationship Diagram Sistem Informasi Kegiatan

Praktikum Laboratorium Teknik informatika

### 3.3. Perancangan Antarmuka

Perancangan antarmuka dibuat menggunakan aplikasi Balsamiq Wireframes. Berikut ini merupakan perancangan antarmuka dari pembuatan aplikasi Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika :

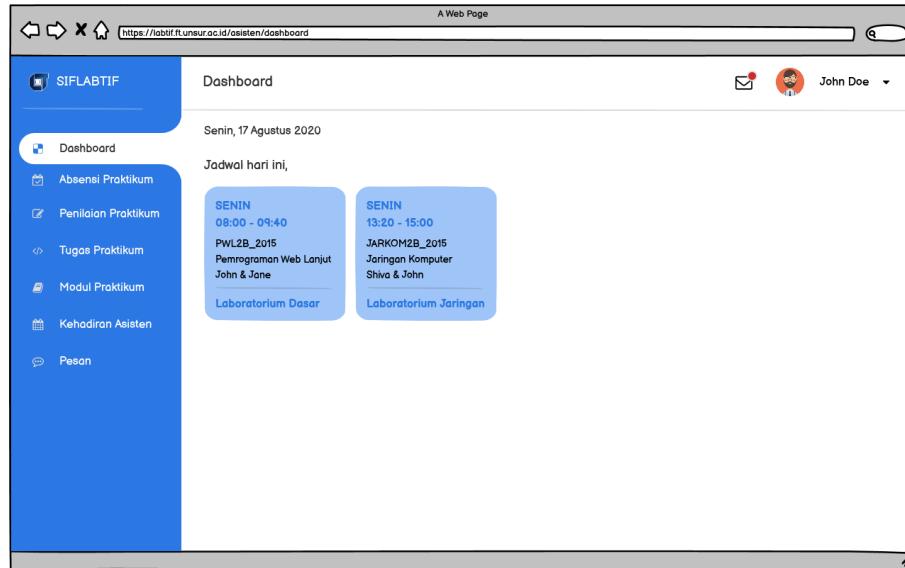
#### 1. Halaman Login Server



*Gambar 3. 96 Perancangan Antarmuka Halaman Login Server*

Keterangan : Halaman ini digunakan admin dan dosen untuk masuk ke dalam aplikasi server.

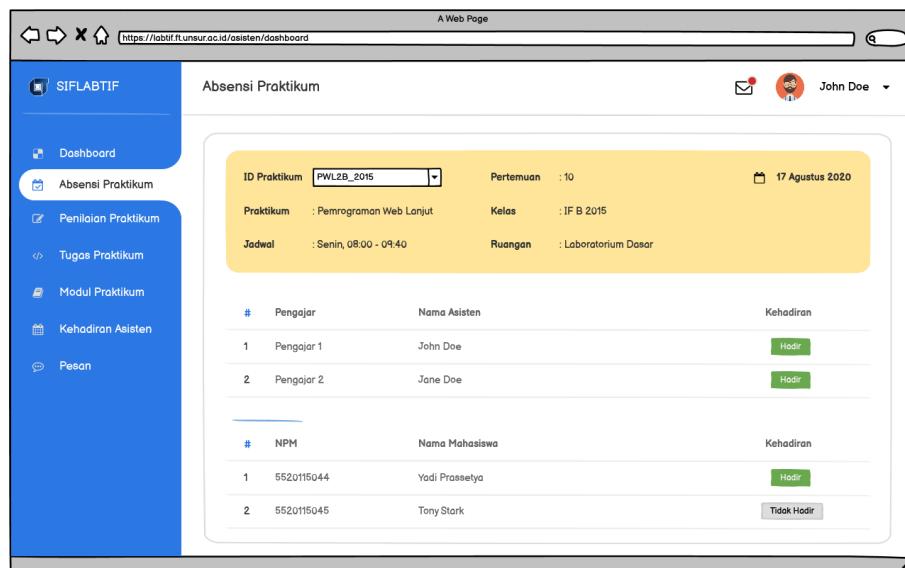
## 2. Halaman Dashboard Asisten



Gambar 3. 97 Perancangan Antarmuka Halaman Dashboard Asisten

Keterangan : Halaman ini dapat digunakan oleh asisten untuk melihat jadwal mengajar miliknya.

## 3. Halaman Absensi Praktikum



Gambar 3. 98 Perancangan Antarmuka Halaman Absensi Praktikum

Keterangan : Halaman ini dapat digunakan oleh asisten untuk melakukan absensi praktikum.

#### 4. Halaman Penilaian Praktikum

The screenshot shows a web-based application for practical examination scoring. The left sidebar has a blue header 'SIFLABTIF' and a list of navigation items: Dashboard, Absensi Praktikum, Penilaian Praktikum (which is selected and highlighted in blue), Tugas Praktikum, Modul Praktikum, Kehadiran Asisten, and Pesan. The main content area is titled 'Penilaian Praktikum' and displays a table of student scores for 'Pemrograman Web Lanjut' in 'IF B 2015'. The table includes columns for #, NPM, Name, Total Tugas, Kehadiran (30%), Tugas (30%), Ujian (40%), and Total. The data is as follows:

#	NPM	Nama Mahasiswa	Total Tugas	Kehadiran (30%)	Tugas (30%)	Ujian (40%)	Total
1	5520115044	Yodi Prasetyo	70	30	25	35	90
2	5520115045	Tony Stark	70	30	25	35	90
3	5520115046	Steve Rogers	70	30	25	35	90
4	5520115047	Thor Odinson	70	30	25	35	90
5	5520115048	Peter Parker	70	30	25	35	90

Gambar 3. 99 Perancangan Antarmuka Halaman Penilaian Praktikum

Keterangan : Halaman ini dapat digunakan oleh asisten untuk melakukan penilaian praktikum.

#### 5. Halaman Kelola Tugas Praktikum

The screenshot shows a web-based application for managing practical assignments. The left sidebar has a blue header 'SIFLABTIF' and a list of navigation items: Dashboard, Absensi Praktikum, Penilaian Praktikum (selected), Tugas Praktikum (highlighted in blue), Modul Praktikum, Kehadiran Asisten, and Pesan. The main content area is titled 'Tugas Praktikum' and displays a table of assignments for 'Pemrograman Web Lanjut' in 'IF B 2015'. The table includes columns for #, ID Tugas, Tugas, Status, and Aksi. The data is as follows:

#	ID Tugas	Tugas	Status	Aksi
1	PWL2B_2015-1	Membuat fungsi login	Tidak aktif	
2	PWL2B_2015-2	Membuat aplikasi CRUD	Aktif	

Gambar 3. 100 Perancangan Antarmuka Halaman Kelola Tugas Praktikum

Keterangan : Halaman ini dapat digunakan oleh asisten untuk mengelola tugas-tugas praktikum seperti menambah, mengubah, dan menghapus tugas.

## 6. Halaman Modul Praktikum

#	NPM	Nama Mahasiswa	Status Pengambilan
1	5520115044	Yadi Prossetyo	Belum
2	5520115045	Tony Stark	Sudah
3	5520115046	Thor Odinson	Sudah
4	5520115047	Steve Rogers	Sudah
5	5520115048	Peter Parker	Sudah

*Gambar 3. 101 Perancangan Antarmuka Halaman Modul Praktikum*

Keterangan : Halaman ini dapat digunakan oleh asisten untuk mengubah status pengambilan modul praktikum.

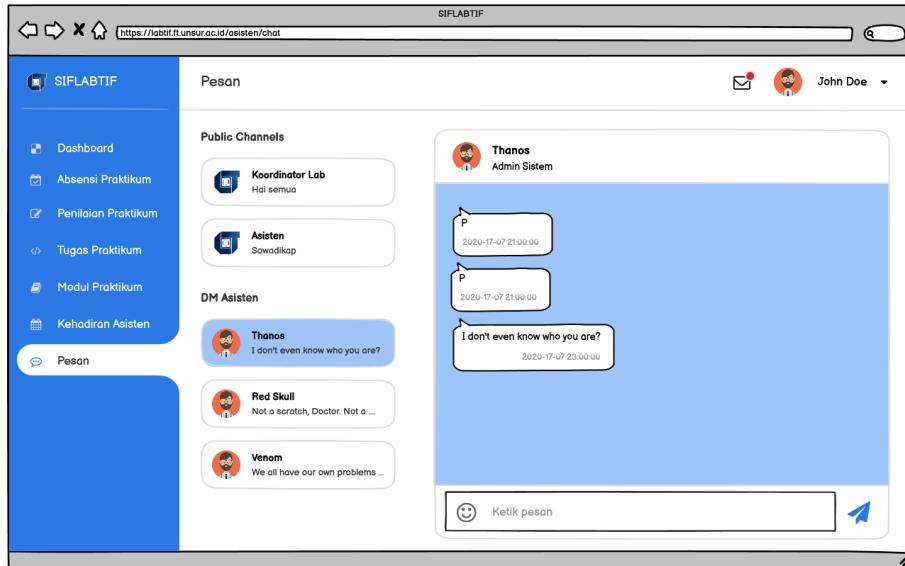
## 7. Halaman Kehadiran Asisten

ID Praktikum	Pertemuan									
	1	2	3	4	5	6	7	8	9	10
PWL2B_2015	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
JARKOM2B_2015	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
PWD1B_2015	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
BD2B_2015	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

*Gambar 3. 102 Perancangan Antarmuka Halaman Kehadiran Asisten*

Keterangan : Halaman ini dapat digunakan oleh asisten untuk mengubah status pengambilan modul praktikum.

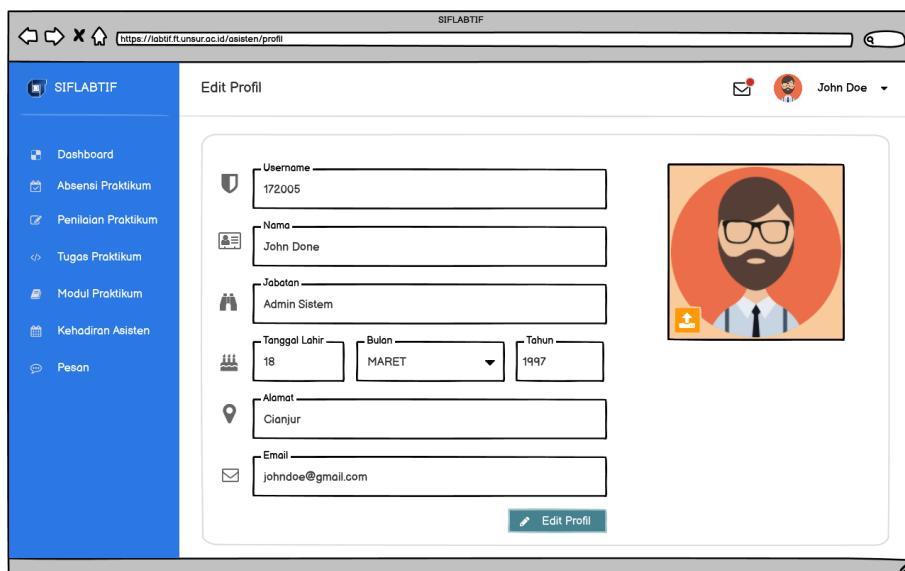
## 8. Halaman Pesan



*Gambar 3. 103 Perancangan Antarmuka Halaman Pesan*

Keterangan : Halaman ini dapat digunakan oleh asisten dan dosen untuk melakukan perpesanan.

## 9. Halaman Edit Profil



*Gambar 3. 104 Perancangan Antarmuka Halaman Edit Profil*

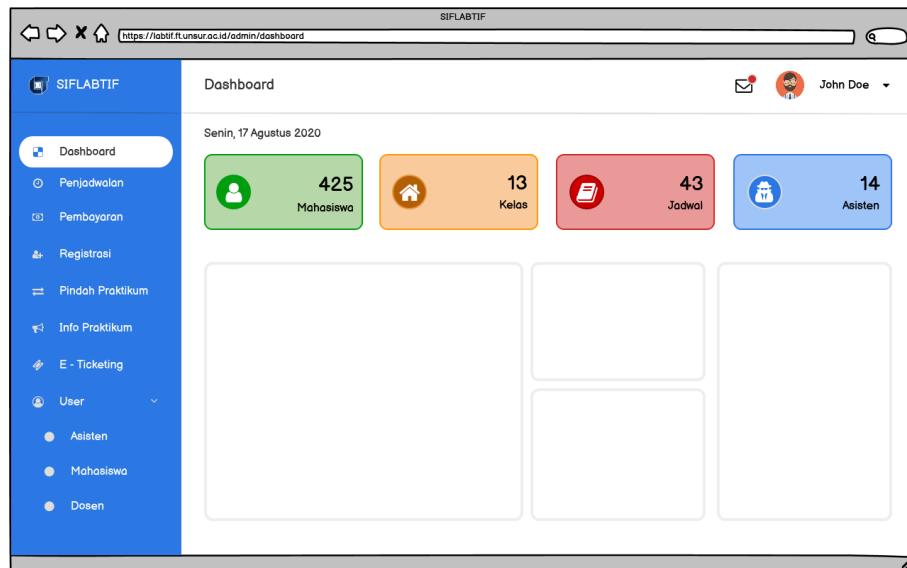
Keterangan : Halaman ini dapat digunakan oleh asisten dan dosen untuk mengubah data dan foto profil.

## 10. Halaman Ganti Password

*Gambar 3. 105 Perancangan Antarmuka Halaman Ganti Password*

Keterangan : Halaman ini dapat digunakan oleh asisten dan dosen untuk mengganti password akun miliknya.

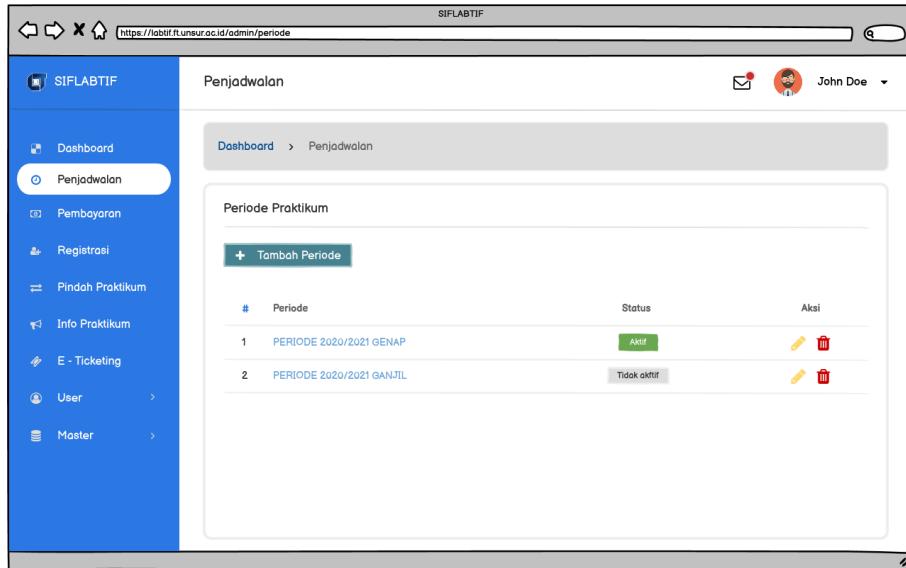
## 11. Halaman Dashboard Admin



*Gambar 3. 106 Perancangan Antarmuka Halaman Dashboard Admin*

Keterangan : Halaman ini dapat digunakan oleh admin untuk melihat ringkasan data-praktikum.

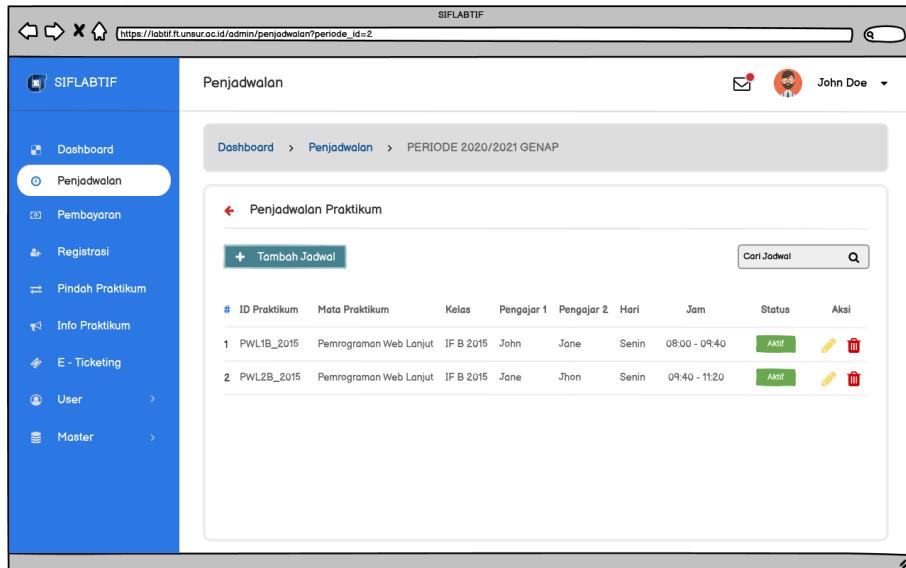
## 12. Halaman Kelola Periode



Gambar 3. 107 Perancangan Antarmuka Halaman Kelola Periode

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data periode praktikum seperti menambah, mengubah, dan menghapus periode.

## 13. Halaman Kelola Jadwal



Gambar 3. 108 Perancangan Antarmuka Halaman Kelola Jadwal

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola jadwal praktikum seperti menambah, mengubah, dan menghapus jadwal.

## 14. Halaman Pembayaran Praktikum

#	NPM	Nama Mahasiswa	Kelas	Status Pembayaran
1	5520115044	Yadi Prasetyo	IF B 2015	Belum
2	5520115045	Tony Stark	IF B 2015	Lunas
3	5520115046	Thor Odinson	IF B 2015	Lunas
4	5520115047	Steve Rogers	IF B 2015	Lunas
5	5520115048	Peter Parker	IF B 2015	Lunas

*Gambar 3. 109 Perancangan Antarmuka Halaman Pembayaran Praktikum*

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengubah status pembayaran praktikum.

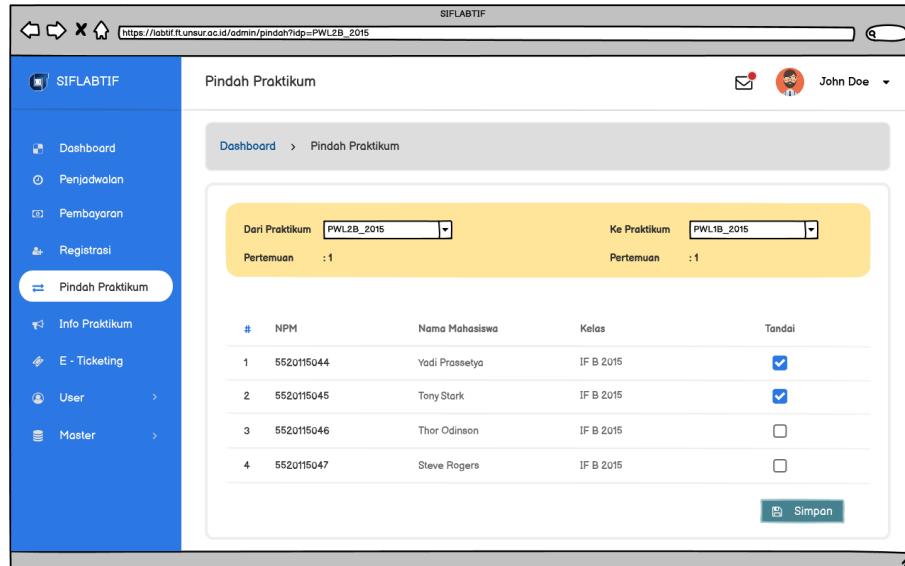
## 15. Halaman Registrasi Praktikum

#	NPM	Nama Mahasiswa	Kelas	Registrasi
1	5520115044	Yadi Prasetyo	IF B 2015	<input checked="" type="checkbox"/>
2	5520115045	Tony Stark	IF B 2015	<input checked="" type="checkbox"/>
3	5520115046	Thor Odinson	IF B 2015	<input checked="" type="checkbox"/>
4	5520115047	Steve Rogers	IF B 2015	<input type="checkbox"/>
5	5520115048	Peter Parker	IF B 2015	<input type="checkbox"/>

*Gambar 3. 110 Perancangan Antarmuka Halaman Registrasi Praktikum*

Keterangan : Halaman ini dapat digunakan oleh admin untuk meregistrasi mahasiswa praktikum.

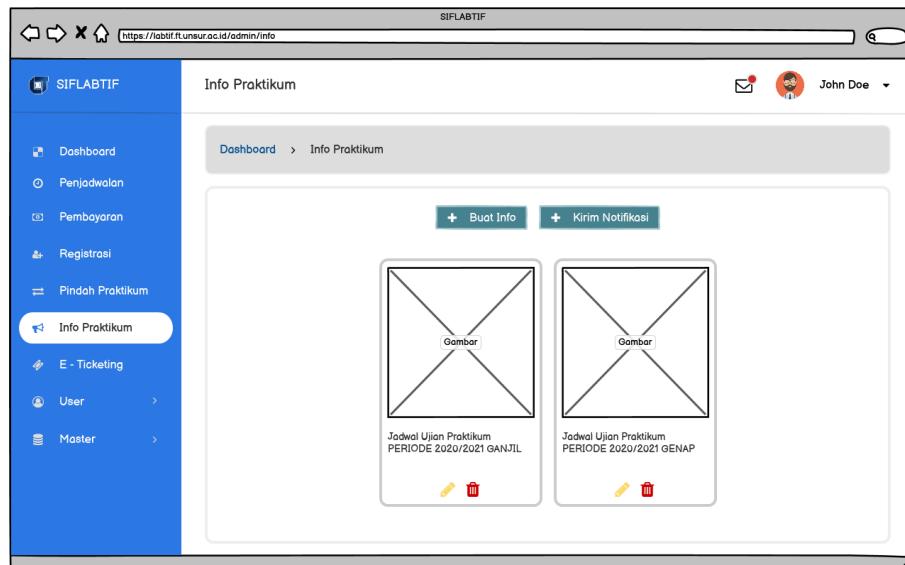
## 16. Halaman Pindah Praktikum



Gambar 3. 111 Perancangan Antarmuka Halaman Pindah Praktikum

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola perpindahan praktikum mahasiswa.

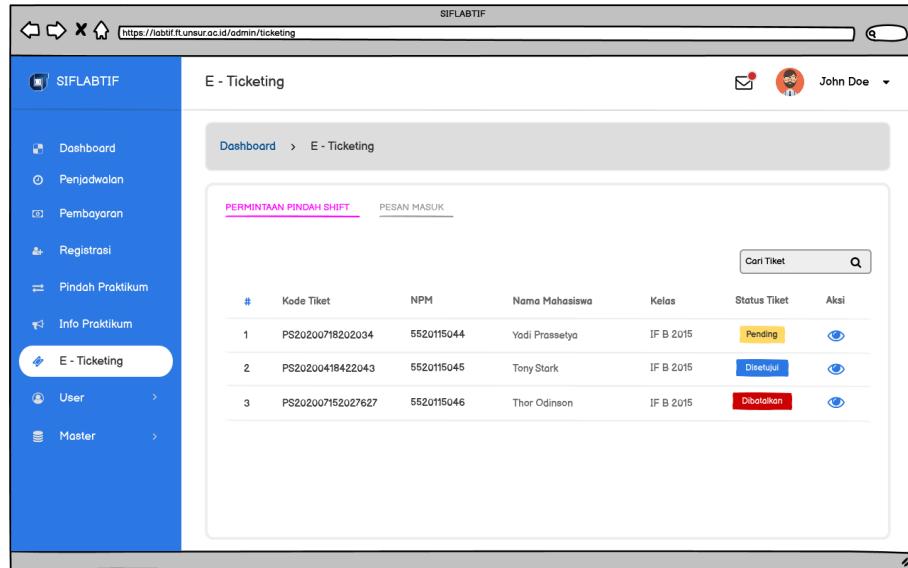
## 17. Halaman Kelola Info Praktikum



Gambar 3. 112 Perancangan Antarmuka Halaman Kelola Info Praktikum

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola info praktikum seperti menambah, mengubah, dan menghapus info praktikum.

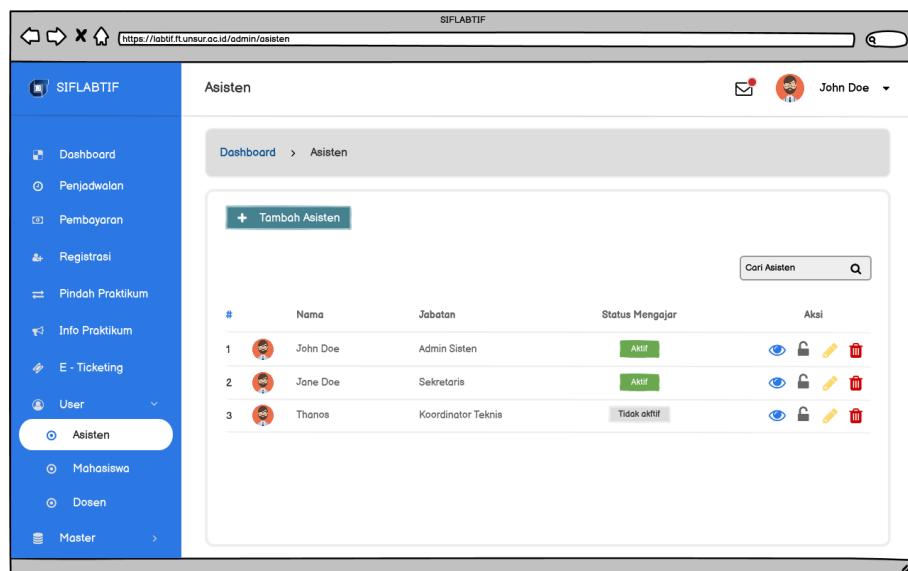
## 18. Halaman Kelola E - Ticketing



Gambar 3. 113 Perancangan Antarmuka Halaman Kelola E - Ticketing

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola layanan E-Ticketing seperti membatal pesan mahasiswa dan mengubah status tiket.

## 19. Halaman Kelola Asisten



Gambar 3. 114 Perancangan Antarmuka Halaman Kelola Asisten

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data asisten seperti menambah, mengubah, dan menghapus asisten.

## 20. Halaman Kelola Dosen

#	Nama	Jabatan	NIDN	Email	Status Mengajar	Aksi
1	Finsa Nurpandi, ST, MT.	Dosen	0411118503	finsa@unsur.ac.id	Aktif	
2	M. Kary Legijawan, ST, M.kom.	Koordinator Lab	0415018402	nama@unsur.ac.id	Aktif	
3	Tormin Abdulghoni, ST, MT.	Dosen	0401087501	nama@unsur.ac.id	Tidak aktif	

Gambar 3. 115 Perancangan Antarmuka Halaman Kelola Dosen

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data dosen seperti menambah, mengubah, dan menghapus dosen.

## 21. Halaman Kelola Mahasiswa

#	NPM	Nama Mahasiswa	Kelas	Status	Aksi
1	5520115044	Yodi Prasetyo	IF B 2015	Aktif	
2	5520115045	Tony Stark	IF B 2015	Aktif	
3	5520115046	Thor Odinson	IF B 2015	Aktif	
4	5520115047	Steve Rogers	IF B 2015	Aktif	
5	5520115048	Peter Parker	IF B 2015	Tidak aktif	

Gambar 3. 116 Perancangan Antarmuka Halaman Kelola Mahasiswa

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data mahasiswa seperti menambah, mengubah, dan menghapus mahasiswa.

## 22. Halaman Kelola Kelas

#	ID Kelas	Kelas	Status Kelas	Aksi
1	ifa2015	IF A 2015	Aktif	
2	ifb2015	IF B 2015	Aktif	
3	ifc2015	IF C 2015	Aktif	
4	ifd2015	IF D 2015	Aktif	
5	ifa2014	IF A 2014	Tidak aktif	

Gambar 3. 117 Perancangan Antarmuka Halaman Kelola Kelas

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data kelas seperti menambah, mengubah, dan menghapus kelas.

## 23. Halaman Kelola Ruangan

#	Ruangan	Jumlah PC	Aksi
1	Laboratorium Dasar	20	
2	Laboratorium Multimedia	15	
3	Laboratorium Jaringan	20	

Gambar 3. 118 Perancangan Antarmuka Halaman Kelola Ruangan

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data ruangan praktikum seperti menambah, mengubah, dan menghapus ruangan.

## 24. Halaman Kelola Mata Praktikum

#	Mata Praktikum	Status	Aksi
1	Analisa dan Perancangan Sistem Informasi	Aktif	
2	Basica Data	Aktif	
3	Pemrograman Web Lanjut	Aktif	
4	Paket Aplikasi Komputer	Tidak aktif	
5	Pemrograman Desktop	Tidak aktif	

Gambar 3. 119 Perancangan Antarmuka Halaman Kelola Mata Praktikum

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data mata praktikum seperti menambah, mengubah, dan menghapus mata praktikum.

## 25. Halaman Kelola Jam

#	Nama	Jam	Aksi
1	Jam ke - 1	08:00 - 09:40	
2	Jam ke - 2	08:50 - 10:30	
3	Jam ke - 3	09:40 - 11:20	
4	Jam ke - 4	10:30 - 12:10	
5	Jam ke - 5	11:20 - 13:00	

Gambar 3. 120 Perancangan Antarmuka Halaman Kelola Jam

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data jam praktikum seperti menambah, mengubah, dan menghapus jam.

## 26. Halaman Kelola Jabatan

#	Nama	Aksi
1	Koordinator Lab	
2	Koordinator Teknis	
3	Wakil Koordinator Teknis	
4	Sekretaris	
5	Bendahara	
6	Admin Sistem	

Gambar 3. 121 Perancangan Antarmuka Halaman Kelola Jabatan

Keterangan : Halaman ini dapat digunakan oleh admin untuk mengelola data jabatan seperti menambah, mengubah, dan menghapus jabatan.

## 27. Halaman Nilai Praktikum (Dosen)

#	NPM	Nama Mahasiswa	Kehadiran (30%)	Tugas (30%)	Ujian (40%)	Total
1	5520115044	Yedi Prossetya	30	25	35	90
2	5520115045	Tony Stark	30	25	35	90
3	5520115046	Steve Rogers	30	25	35	90
4	5520115047	Thor Odinson	30	25	35	90

Gambar 3. 122 Perancangan Antarmuka Halaman Nilai Praktikum (Dosen)

Keterangan : Halaman ini dapat digunakan oleh dosen untuk melihat data nilai praktikum mahasiswa.

## 28. Halaman Kritik & Saran

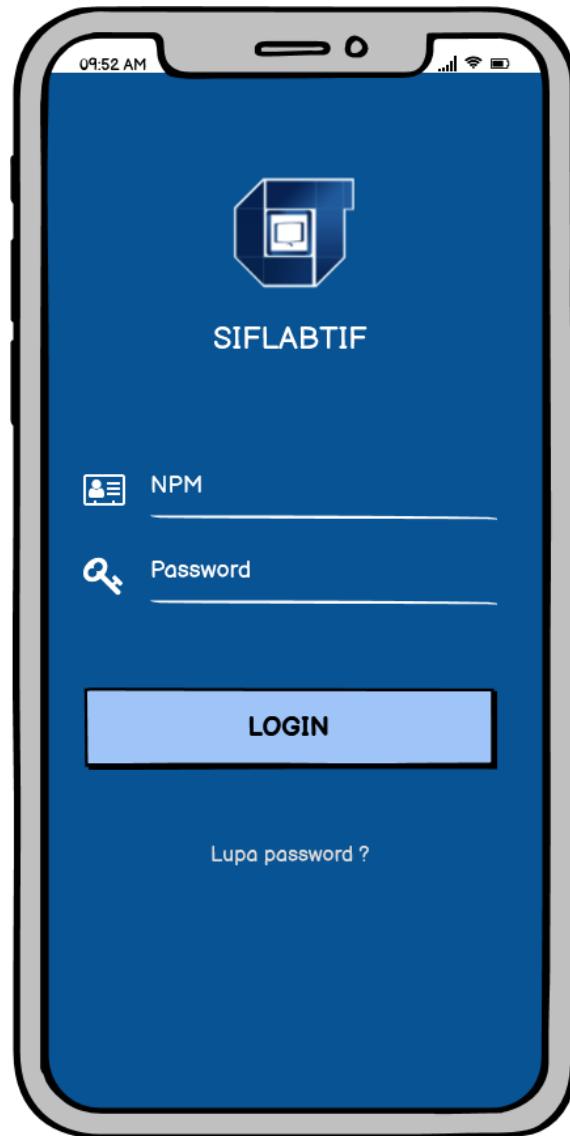
The screenshot shows a web-based application titled 'SIFLABTIF' with a URL of [https://siflabtif.umsur.ac.id/asisten/kritik\\_saran](https://siflabtif.umsur.ac.id/asisten/kritik_saran). The interface includes a sidebar on the left with icons and labels for 'Dashboard', 'Absensi Praktikum', 'Penilaian Praktikum', 'Tugas Praktikum', 'Modul Praktikum', 'Kehadiran Asisten', 'Kritik & Saran', and 'Pesanan'. The main content area is titled 'Kritik & Saran' and displays a table of messages. The table has columns for '#', 'Tanggal' (Date), 'Pesan' (Message), and 'Status'. There are two entries:

#	Tanggal	Pesan	Status
1	2020-08-17	Merdeka !!!	Diterima
2	2020-08-20	Pembelajaran terlalu fokus ke modul tanpa penjelasan lebih lanjut. Sebaiknya penjelasan dilakukan dengan presentasi	Belum

Gambar 3. 123 Perancangan Antarmuka Halaman Kritik & Saran

Keterangan : Halaman ini dapat digunakan oleh koordinator teknis untuk melihat data kritik & saran yang dikirim oleh mahasiswa.

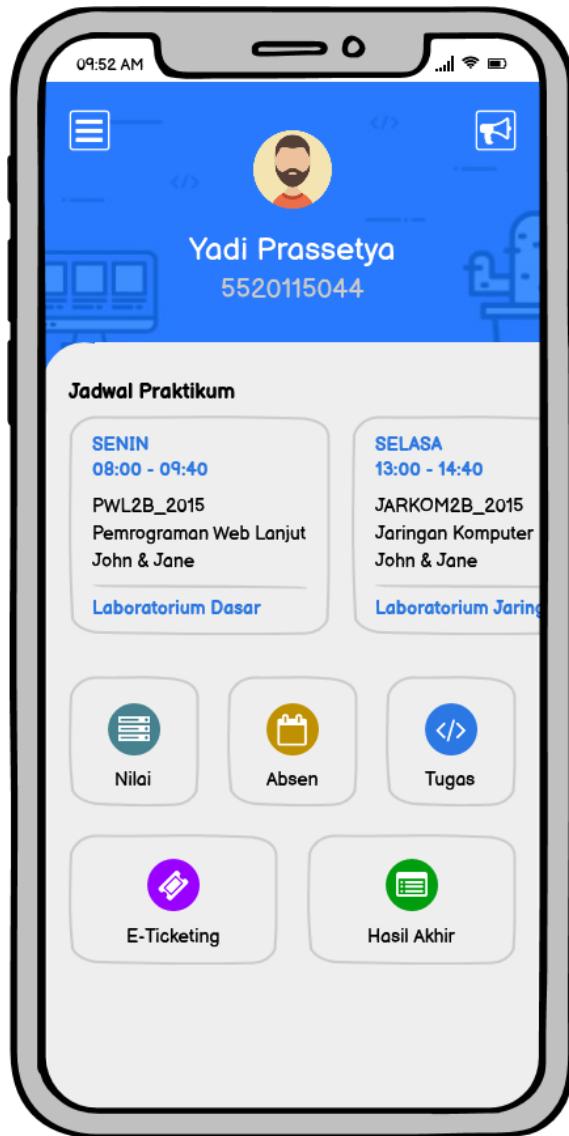
## 29. Halaman Login Client



*Gambar 3. 124 Perancangan Antarmuka Halaman Login Client*

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk masuk ke dalam aplikasi client.

### 30. Halaman Dashboard Client



Gambar 3. 125 Perancangan Antarmuka Halaman Dashboard Client

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk melihat jadwal praktikum miliknya dan juga sebagai halaman menu.

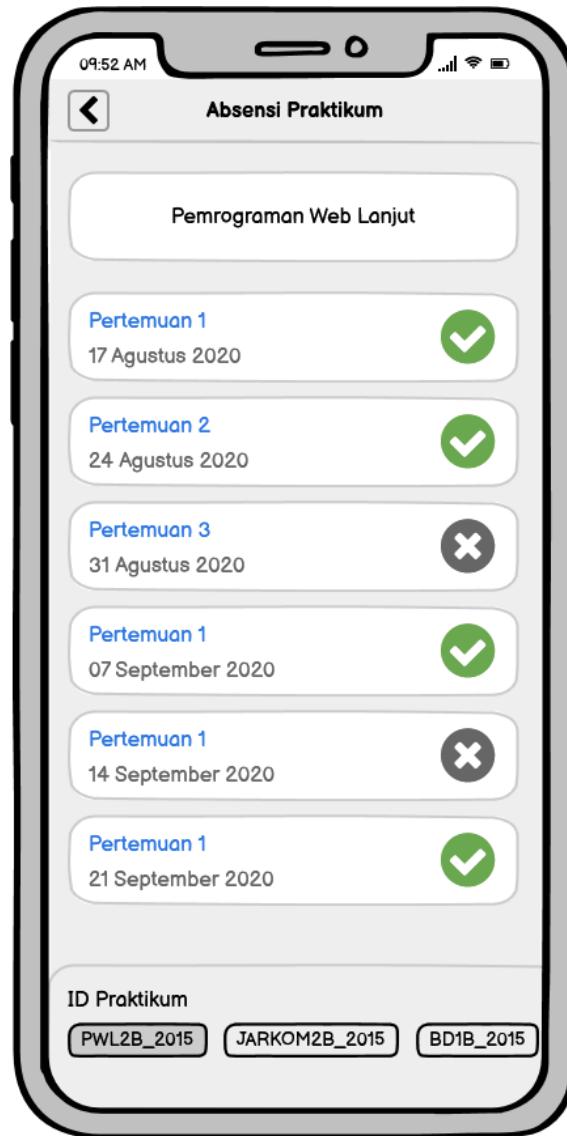
### 31. Halaman Nilai Praktikum (Client)



*Gambar 3. 126 Perancangan Antarmuka Halaman  
Nilai Praktikum (Client)*

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk melihat nilai-nilai tugas praktikum miliknya.

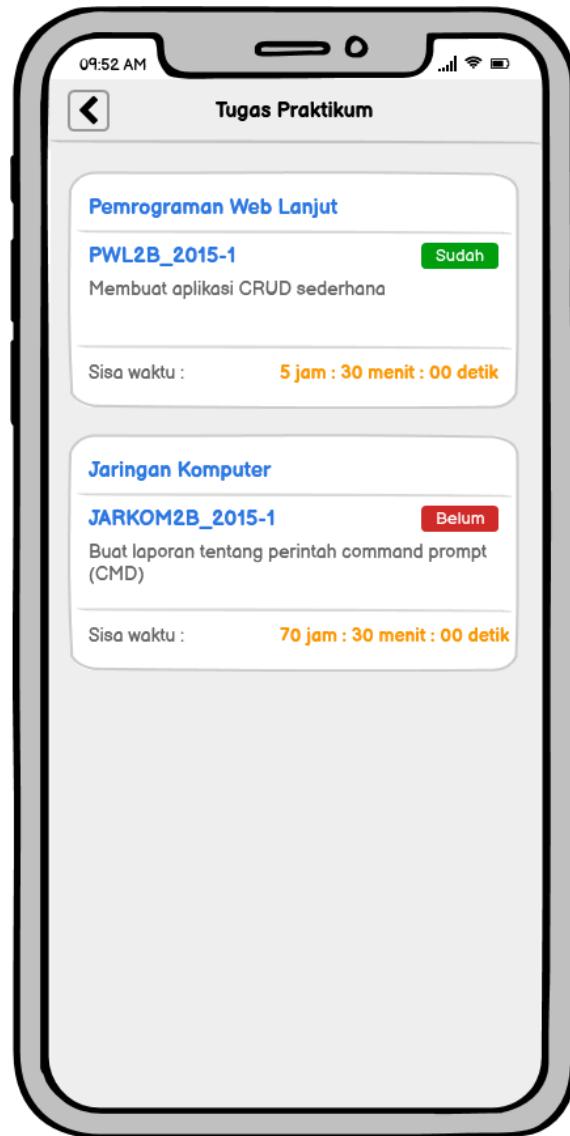
## 32. Halaman Absensi Praktikum (Client)



*Gambar 3. 127 Perancangan Antarmuka Halaman  
Absensi Praktikum (Client)*

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk melihat data absensi praktikum miliknya.

### 33. Halaman Tugas Praktikum



*Gambar 3. 128 Perancangan Antarmuka Halaman Tugas Praktikum*

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk melihat daftar tugas praktikum.

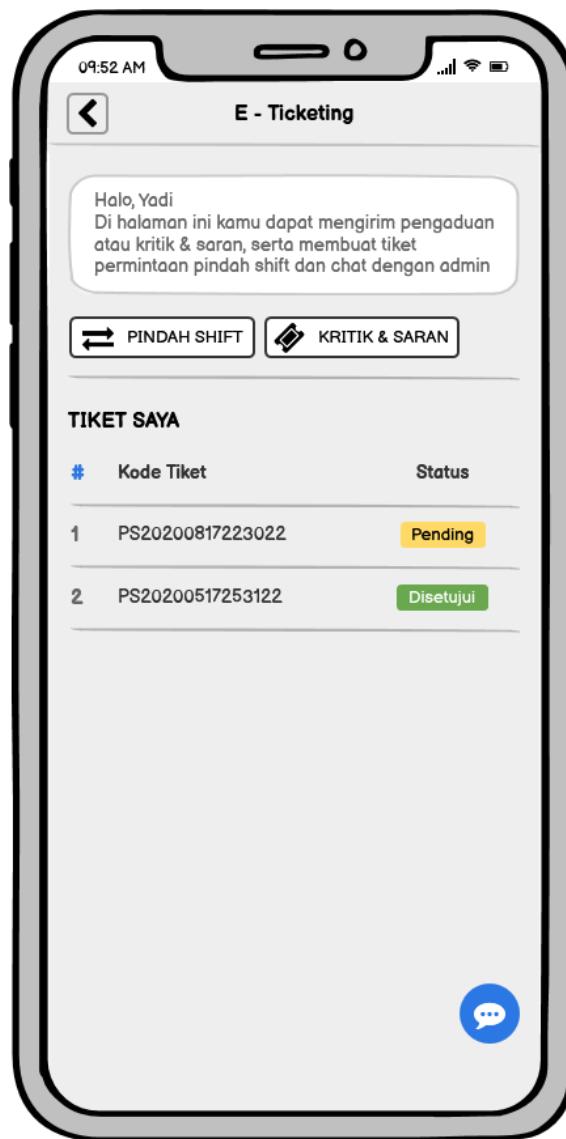
## 34. Halaman Detail Tugas



Gambar 3. 129 Perancangan Antarmuka Halaman Detail Tugas

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk melihat detail tugas dan *upload* tugas praktikum.

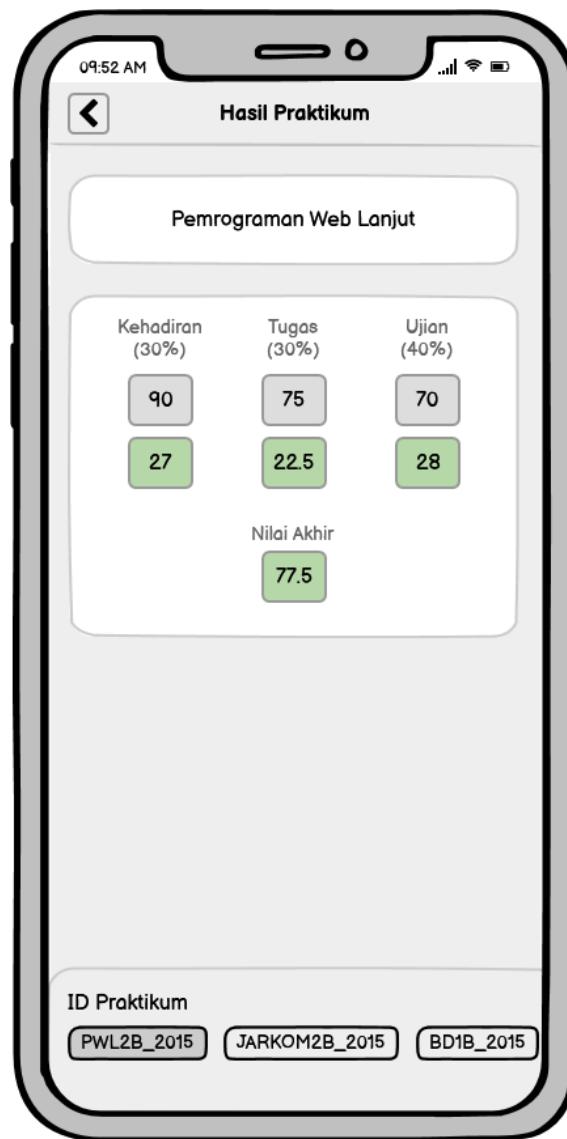
## 35. Halaman E - Ticketing



Gambar 3. 130 Perancangan Antarmuka Halaman E - Ticketing

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk melakukan permintaan pindah shift praktikum, perpesanan dengan admin, dan mengirim pengaduan atau kritik & saran.

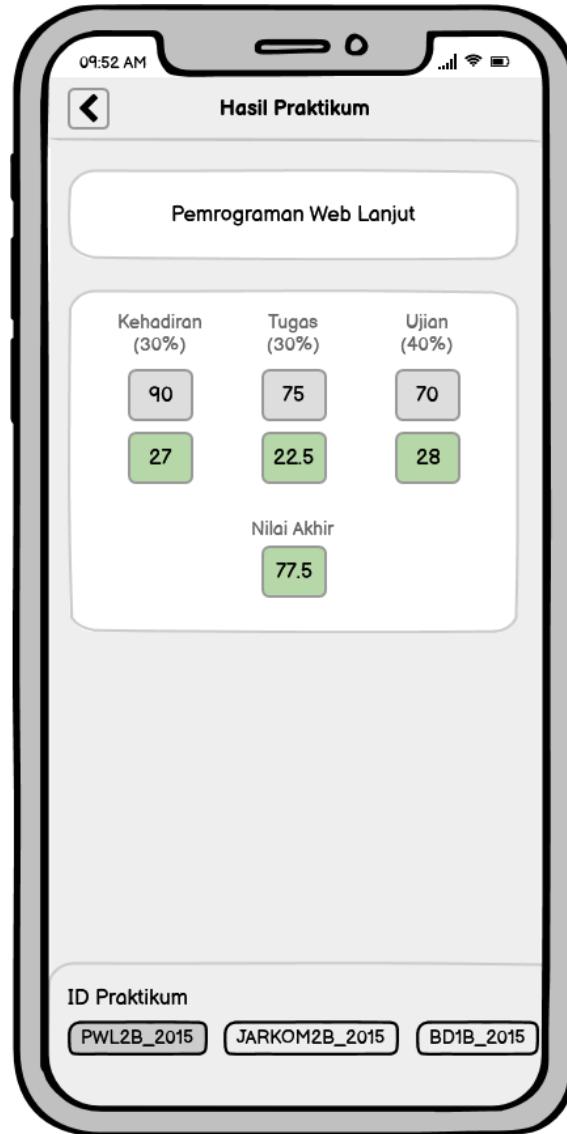
### 36. Halaman Hasil Praktikum



Gambar 3. 131 Perancangan Antarmuka Halaman Hasil Praktikum

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk melihat rangkuman nilai hasil praktikum.

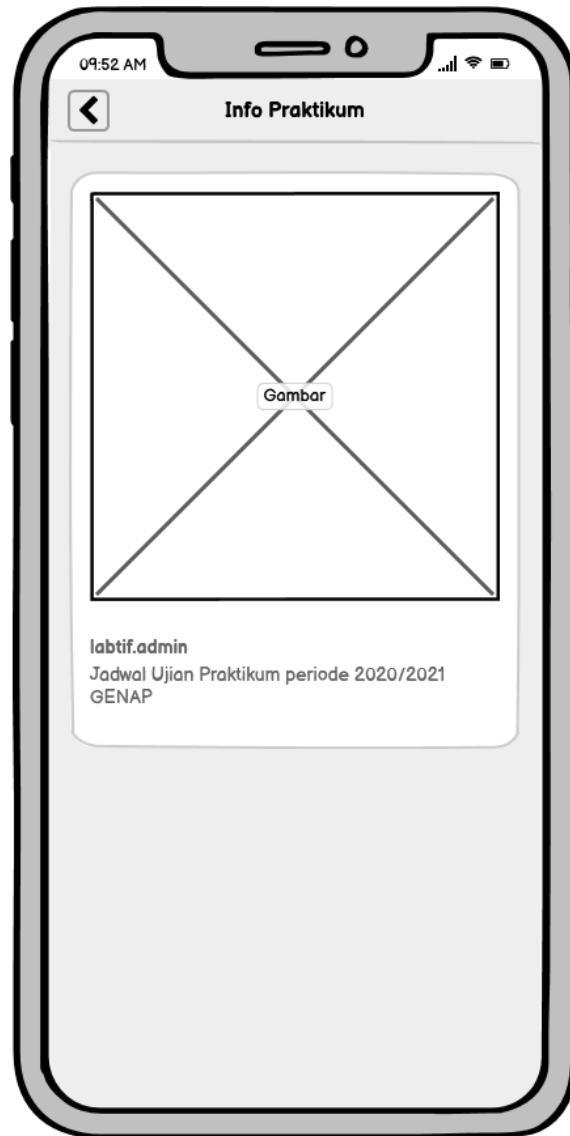
## 37. Halaman Ganti Password (Client)



*Gambar 3. 132 Perancangan Antarmuka Halaman  
Ganti Password (Client)*

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk mengganti password akun miliknya.

### 38. Halaman Info Praktikum



*Gambar 3. 133 Perancangan Antarmuka Halaman Info Praktikum*

Keterangan : Halaman ini dapat digunakan oleh mahasiswa untuk melihat info-praktikum.

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN SISTEM**

Bab ini menjelaskan pengujian dan implementasi sistem aplikasi yang telah dibangun.

#### **4.1. Implementasi Sistem**

Tahap implementasi merupakan proses lanjutan dari hasil analisis dan perancangan yang telah dilakukan sebelumnya.

Adapun untuk proses penerapan Aplikasi Kegiatan Praktikum agar dapat diakses oleh pengunjung situs, yaitu dengan cara sebagai berikut:

1. Menyewa *hosting* dan *domain* kepada perusahaan yang menyewakan *hosting* dan *domain*.
2. Melakukan konfigurasi *file* dan *database* serta mengupload *file Website* pada *hosting* yang telah disewa.

##### **4.1.1 Implementasi Perangkat Keras**

Kebutuhan minimal perangkat keras yang dibutuhkan oleh sistem dalam implementasi perangkat keras adalah sebagai berikut:

1. Processor Intel core 2 duo.
2. Memori RAM 2 GB.
3. *Mouse*, *Keyboard*, dan Monitor sebagai peralatan antarmuka.

##### **4.1.2 Implementasi Perangkat Lunak**

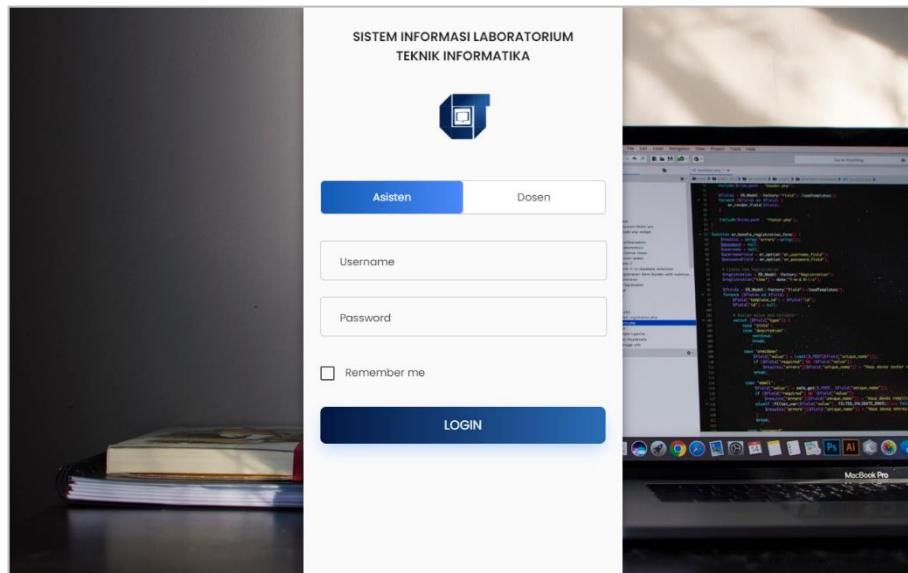
Selain perangkat keras, dalam implementasi pembuatan Aplikasi Kegiatan Praktikum dibutuhkan juga perangkat lunak. Berikut ini merupakan spesifikasi minimal perangkat lunak yang digunakan untuk menjalankan dan mengimplementasikan *Website*, diantaranya :

1. Browser (direkomendasikan Google Chrome).

## 4.2. Implementasi Antarmuka

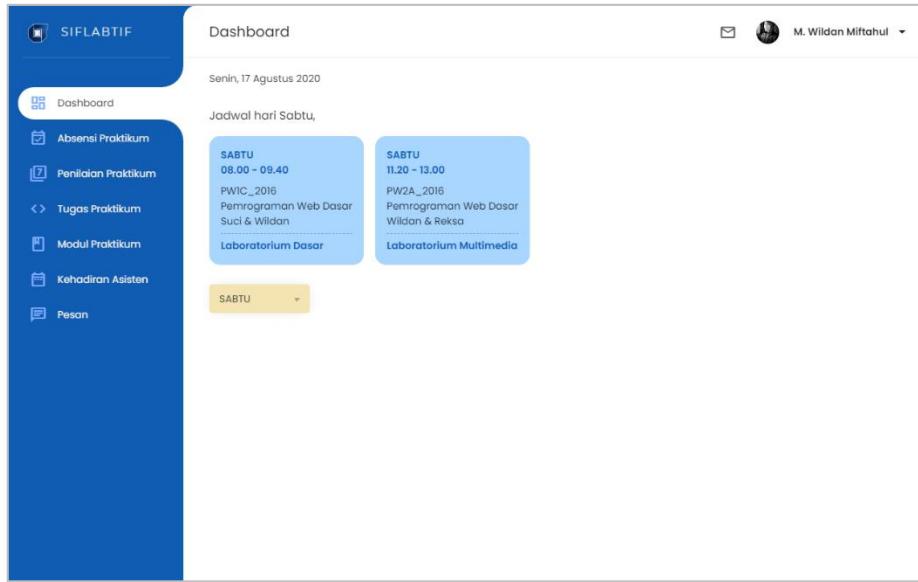
Berikut ini merupakan implementasi antarmuka dari pembuatan Sistem Informasi Kegiatan Praktikum Laboratorium Teknik Informatika:

### 1. Halaman Login Server



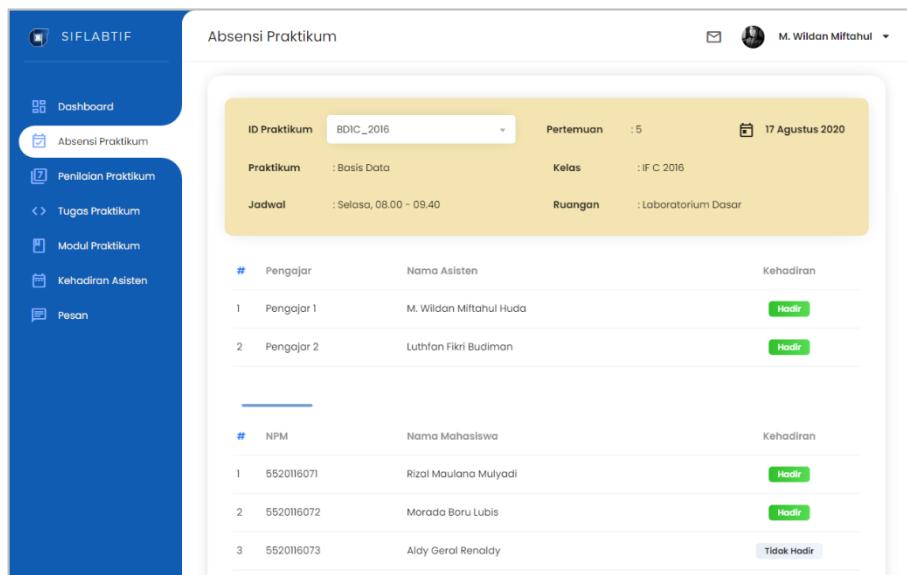
Gambar 4. 1 Halaman Login Server

## 2. Halaman Dashboard Asisten



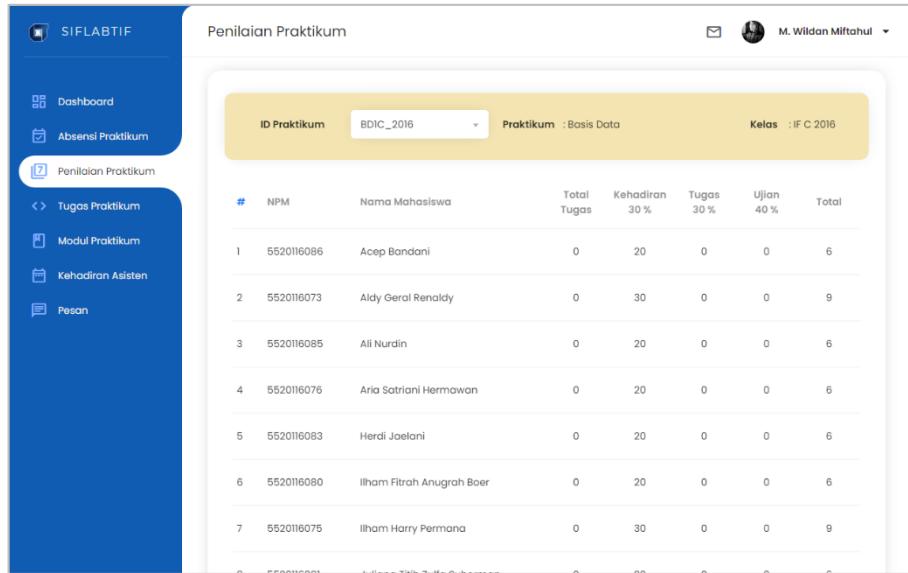
Gambar 4. 2 Halaman Dashboard Asisten

## 3. Halaman Absensi Praktikum



Gambar 4. 3 Halaman Absensi Praktikum

#### 4. Halaman Penilaian Praktikum

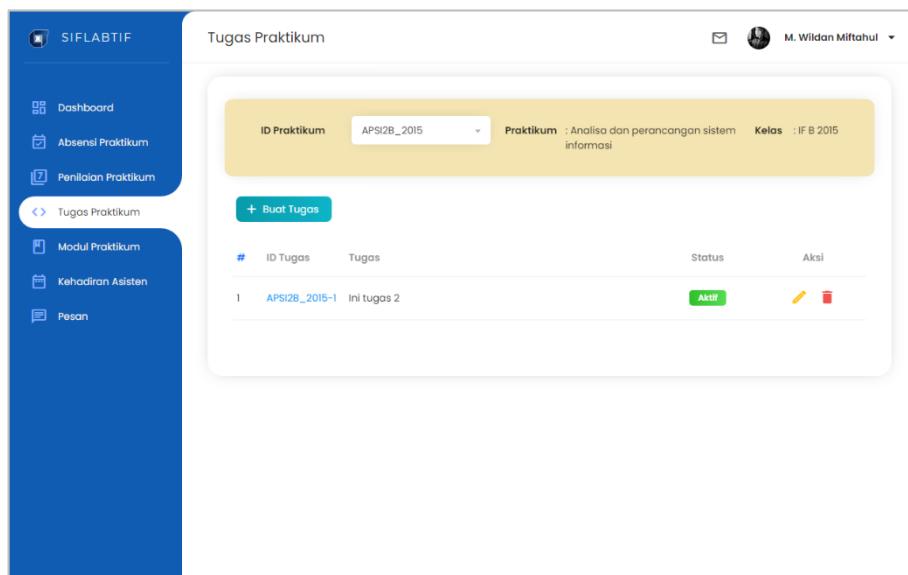


The screenshot shows a table of student grades for the 'Basis Data' practical course. The columns represent NPM, Name, Total Tugas (30%), Kehadiran (30%), Tugas (30%), Ujian (40%), and Total grade.

#	NPM	Nama Mahasiswa	Total Tugas 30 %	Kehadiran 30 %	Tugas 30 %	Ujian 40 %	Total
1	5520116086	Acep Bandani	0	20	0	0	6
2	5520116073	Aldy Geral Renaldi	0	30	0	0	9
3	5520116085	All Nurdin	0	20	0	0	6
4	5520116076	Aria Satriani Hermawon	0	20	0	0	6
5	5520116083	Herdi Jeloni	0	20	0	0	6
6	5520116080	Ilham Fitrah Anugrah Boer	0	20	0	0	6
7	5520116075	Ilham Harry Permana	0	30	0	0	9

Gambar 4. 4 Halaman Penilaian Praktikum

#### 5. Halaman Kelola Tugas Praktikum



The screenshot shows a list of assignments for the 'Analisa dan perancangan sistem informasi' practical course. The columns represent ID Tugas, Tugas, Status, and Aksi (Action).

#	ID Tugas	Tugas	Status	Aksi
1	APSI2B_2015-1	Ini tugas 2	Aktif	

Gambar 4. 5 Halaman Kelola Tugas Praktikum

## 6. Halaman Modul Praktikum

The screenshot shows a list of students with their NPM numbers, names, and status of completion (Sudah or Belum). The status column uses green buttons for 'Sudah' and grey buttons for 'Belum'.

#	NPM	Nama	Status Pengambilan
1	5520116086	Acep Bandoni	Sudah
2	5520116085	Ali Nurdin	Sudah
3	5520116084	Taufik Usman	Sudah
4	5520116083	Herdii Jaelani	Belum
5	5520116082	Selviani Kusnadi	Belum
6	5520116081	Juliana Titih Zulfa Suherman	Belum
7	5520116080	Iliham Fitrah Anugrah Boer	Belum
8	5520116079	Muhammad Ardi Kurniawan	Belum
9	5520116078	Muhammad Irpan Supardi	Belum

Gambar 4. 6 Halaman Modul Praktikum

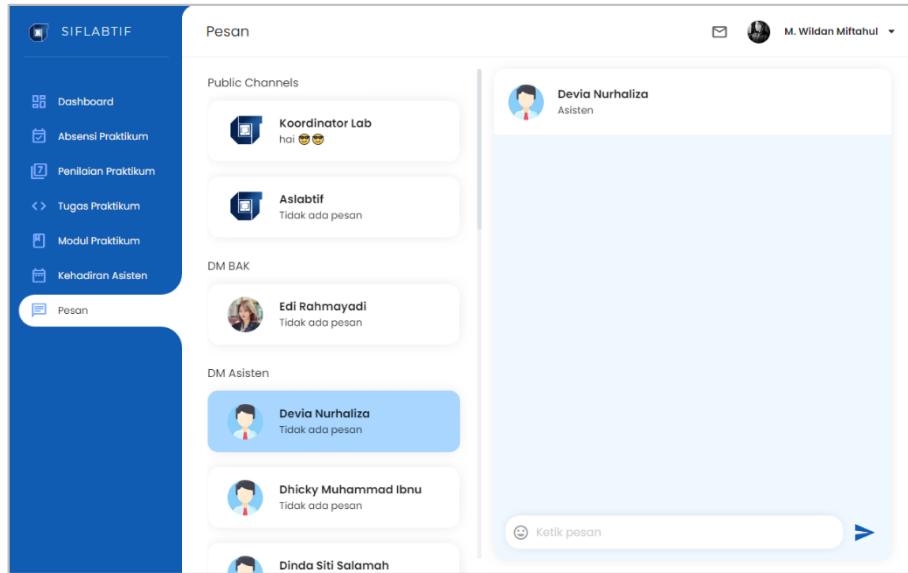
## 7. Halaman Kehadiran Asisten

The screenshot displays a grid showing student attendance across 10 sessions for six different practical modules. A legend indicates that green checkmarks represent present (hadir) and red 'X's represent absent (tidak hadir).

ID Praktikum	Pertemuan									
	1	2	3	4	5	6	7	8	9	10
PW2C_2016	X	X	X	X	X	X	X	X	X	X
BDIC_2016	✓	✓	✓	X	✓	X	X	X	X	X
PWIC_2016	X	X	X	X	X	X	X	X	X	X
PDNR_2017	X	X	X	X	X	X	X	X	X	X
APSI2B_2015	X	X	X	X	X	X	X	X	X	X
PW2A_2016	X	X	X	X	X	X	X	X	X	X

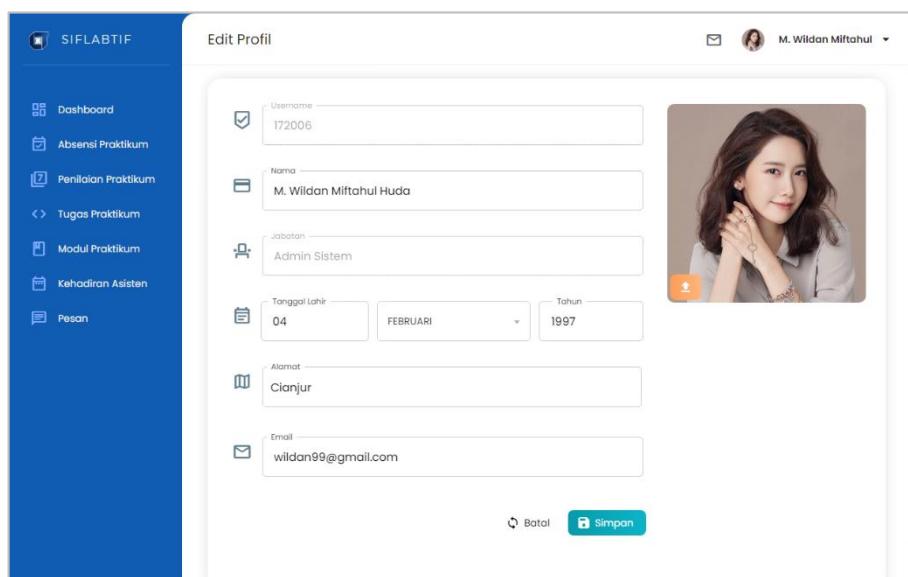
Gambar 4. 7 Halaman Kehadiran Asisten

## 8. Halaman Pesan



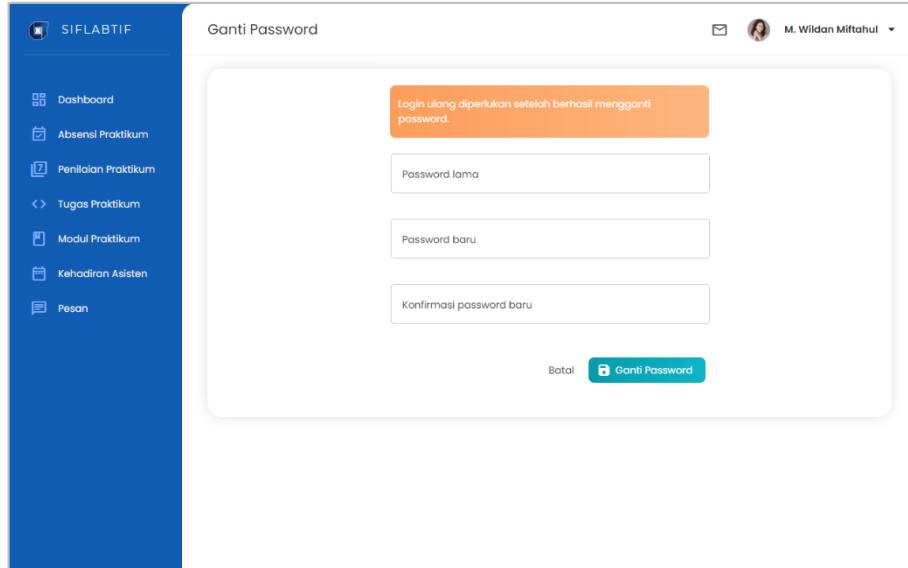
Gambar 4. 8 Halaman Pesan

## 9. Halaman Edit Profil



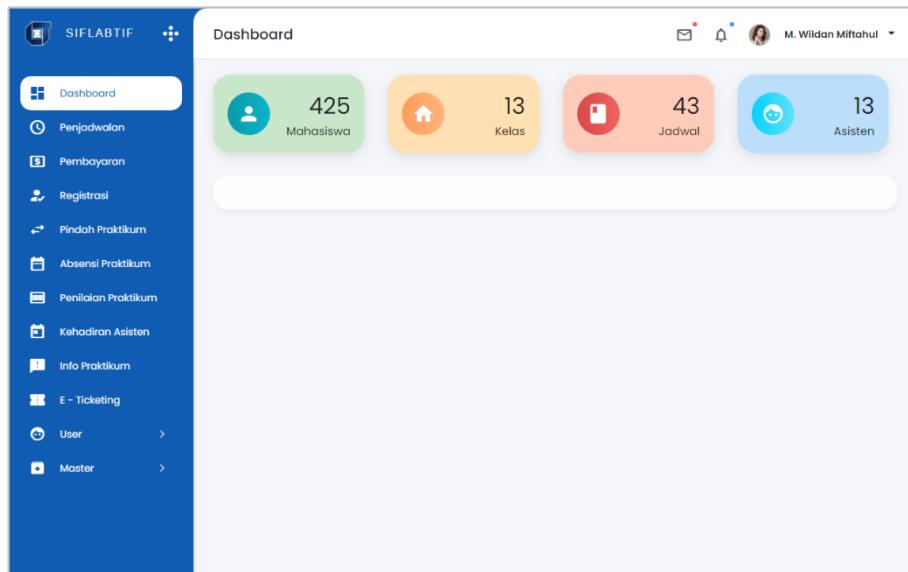
Gambar 4. 9 Halaman Edit Profil

## 10. Halaman Ganti Password



Gambar 4. 10 Halaman Ganti Password

## 11. Halaman Dashboard Admin



Gambar 4. 11 Halaman Dashboard Admin

## 12. Halaman Kelola Periode

The screenshot shows a list of academic periods:

#	Periode	Status Periode	Aksi
1	PERIODE 2017/2018 GANJIL	Aktif	
2	PERIODE 2016/2017 GENAP	Tidak aktif	
3	PERIODE 2016/2017 GANJIL	Tidak aktif	
4	PERIODE 2015/2016 GENAP	Tidak aktif	
5	PERIODE 2015/2016 GANJIL	Tidak aktif	
6	PERIODE 2014/2015 GENAP	Tidak aktif	

Gambar 4. 12 Halaman Kelola Periode

## 13. Halaman Kelola Jadwal

The screenshot shows a list of practical sessions:

#	ID Jadwal	Mata Praktikum	Kelas	Pengajar 1	Pengajar 2	Hari	Jam	Ruangan	Status	Aksi
1	APSI2B_2015	Analisa dan perancangan sistem informasi	IF B 2015	Agung Gumelar	M. Wildan Miftahul Huda	Kamis	13:00 – 14:40	Laboratorium Dasar	Aktif	
2	BD2D_2016	Basis Data	IF D 2016	Reksa Mulia Fitrananda	Dhicky Muhammad Ibnu Zamil	Jumat	14:40 – 16:20	Laboratorium Multimedia	Aktif	
3	PW2D_2016	Pemrograman Web Dasar	IF D 2016	Agung Gumelar	Suci Lestioni Putri	Rabu	14:40 – 16:20	Laboratorium Multimedia	Aktif	
4	PWID_2016	Pemrograman Web Dasar	IF D 2016	Dzirki Haqiqi	Agung Gumelar	Selasa	12:10 – 13:50	Laboratorium Multimedia	Aktif	

Gambar 4. 13 Halaman Kelola Jadwal

#### 14. Halaman Pembayaran Praktikum

The screenshot shows the 'Pembayaran Praktikum' (Practical Payment) page. The left sidebar has a blue header 'SIFLABTIF' and a list of navigation items: Dashboard, Penjadwalan, Pembayaran (selected), Registrasi, Pindah Praktikum, Absensi Praktikum, Penilaian Praktikum, Kehadiran Asisten, Info Praktikum, E - Ticketing, User, and Master. The main content area has a title 'Pembayaran Praktikum' and a sub-section 'Status Pembayaran Praktikum'. It displays a table with columns: #, NPM, Nama, Kelas, and Status Pembayaran. The table shows 8 rows of student data, all marked as 'Lunas' (Paid). A search bar 'Cari Mahasiswa' and a magnifying glass icon are at the top right of the table.

#	NPM	Nama	Kelas	Status Pembayaran
1	5520117120	Amirul Fadillah	IF NR 2017	Lunas
2	5520117119	Muldani	IF NR 2017	Lunas
3	5520117118	Heru Dwi Permana	IF NR 2017	Lunas
4	5520117117	Yoga Muhamad Rimba A.	IF NR 2017	Lunas
5	5520117116	Nabila Suryana	IF NR 2017	Lunas
6	5520117115	Ilyas Muzakir Harahap	IF NR 2017	Lunas
7	5520117114	Dondi Rustandi	IF NR 2017	Lunas
8	5520117113	Niman Maulana	IF NR 2017	Lunas

Gambar 4. 14 Halaman Pembayaran Praktikum

#### 15. Halaman Registrasi Praktikum

The screenshot shows the 'Registrasi Praktikum' (Practical Registration) page. The left sidebar has a blue header 'SIFLABTIF' and a list of navigation items: Dashboard, Penjadwalan, Pembayaran, Registrasi (selected), Pindah Praktikum, Absensi Praktikum, Penilaian Praktikum, Kehadiran Asisten, Info Praktikum, E - Ticketing, User, and Master. The main content area has a title 'Registrasi Praktikum' and a sub-section 'Dashboard > Registrasi'. It displays a table with columns: #, NPM, Nama, Kelas, and Tanda. The table shows 7 rows of student data, all marked with a checked checkbox in the 'Tanda' column. A search bar 'Cari Mahasiswa' and a magnifying glass icon are at the top right of the table. At the top of the table, there are dropdown menus for 'ID Praktikum' (BDIC\_2016), 'Praktikum' (Basis Data), and 'Kelas' (IF C 2016).

#	NPM	Nama	Kelas	Tanda
1	5520116071	Rizal Maulana Mulyadi	IF C 2016	<input checked="" type="checkbox"/>
2	5520116072	Morada Boru Lubis	IF C 2016	<input checked="" type="checkbox"/>
3	5520116073	Aldy Geral Renaldi	IF C 2016	<input checked="" type="checkbox"/>
4	5520116074	Tito Setiawati	IF C 2016	<input checked="" type="checkbox"/>
5	5520116075	Ilham Harry Permana	IF C 2016	<input checked="" type="checkbox"/>
6	5520116076	Aria Satriani Hermawan	IF C 2016	<input checked="" type="checkbox"/>
7	5520116077	M. Faridly Habibi Maulana	IF C 2016	<input checked="" type="checkbox"/>

Gambar 4. 15 Halaman Registrasi Praktikum

## 16. Halaman Pindah Praktikum

#	NPM	Nama Mahasiswa	Kelas	Tandai
1	5520116086	Acep Bandani	IF C 2016	<input type="checkbox"/>
2	5520116073	Aldy Geral Renaldi	IF C 2016	<input type="checkbox"/>
3	5520116085	Ali Nurdin	IF C 2016	<input type="checkbox"/>
4	5520116076	Aria Satriani Hermawan	IF C 2016	<input type="checkbox"/>
5	5520116083	Herdi Jaelani	IF C 2016	<input type="checkbox"/>
6	5520116080	Ilham Fitrah Anugrah Boer	IF C 2016	<input type="checkbox"/>
7	5520116075	Ilham Harry Permono	IF C 2016	<input type="checkbox"/>

Gambar 4. 16 Halaman Pindah Praktikum

## 17. Halaman Kelola Info Praktikum

JADWAL PRAKTIKUM  
Tipe: Internship & Praktik  
JADWAL PRAKTIKUM MINGGU KE - 1

No	Pelajaran	Ruang	Periode
1	10.00 - 11.45	IFC-1A	Praktik & Diskusi
2	10.00 - 11.45	IFC-1D	Praktik & Diskusi
3	10.00 - 11.45	IFC-1D	Praktik & Diskusi
4	10.00 - 11.45	IFC-1D	Praktik & Diskusi
5	10.00 - 11.45	IFC-1D	Praktik & Diskusi
6	10.00 - 11.45	IFC-1D	Praktik & Diskusi
7	10.00 - 11.45	IFC-1D	Praktik & Diskusi

Jadwal Ujian Praktikum PERIODE  
2019/2020 GENAP

Gambar 4. 17 Halaman Kelola Info Praktikum

## 18. Halaman Kelola E-Ticketing

The screenshot shows the 'E - Ticketing' section of the SIFLABTIF application. On the left sidebar, under 'User', the 'E - Ticketing' option is selected. The main content area displays a table titled 'PERMINTAAN PINDAH SHIFT' with two rows of data. The columns are '#', 'Kode Tiket', 'NPM', 'Nama', 'Status', and 'Aksi'. Row 1: Kode Tiket PS20200719112435, NPM 5520115044, Nama Yadi Prasetya, Status Diproses, Aksi button. Row 2: Kode Tiket PS20200804035139, NPM 5520115044, Nama Yadi Prasetya, Status Pending, Aksi button. Navigation buttons at the bottom include 'PREVIOUS', '1', and 'NEXT'.

Gambar 4. 18 Halaman Kelola E-Ticketing

## 19. Halaman Kelola Asisten

The screenshot shows the 'Asisten' section of the SIFLABTIF application. On the left sidebar, under 'User', the 'Asisten' option is selected. The main content area displays a table titled 'Dashboard > Asisten' with five rows of data. The columns are '#', 'Nama', 'Jabatan', 'Status Mengajar', and 'Aksi'. Row 1: Edi Rahmayadi, Bagian Administrasi Dan Keuangan (BAK), Aktif, Aksi button. Row 2: M. Firaz F. Nurjaman, Wakil Koordinator Teknis, Aktif, Aksi button. Row 3: Dinda Siti Salomah, Asisten, Aktif, Aksi button. Row 4: Devia Nurhaliza, Asisten, Aktif, Aksi button. Row 5: Reynoldy Miftahuddin Fouzi, Koordinator Teknis, Aktif, Aksi button. Each row has a small profile picture and a set of icons for edit, lock, and delete actions.

Gambar 4. 19 Halaman Kelola Asisten

## 20. Halaman Kelola Dosen

The screenshot shows the 'Master Dosen' (Faculty Management) page. On the left is a sidebar with various menu items. The main area displays a table with four rows of faculty information:

#	Nama	Jabatan	Username	Email	Status Mengajar	Aksi
1	Tarmin Abdulghani, ST., MT.	Dosen	D202002	tarmin@gmail.com	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
2	Siti Sarah Abdullah, ST., MT.	Dosen	D202001	sitisarah@ft.unsur.ac.id	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
3	Agus Suheri S.T., M.Kom	Dosen	I62001	asuheri@gmail.com	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
4	Sutono, S.Si, M.Kom	Dosen	I62002	sutono@gmail.com	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>

Gambar 4. 20 Halaman Kelola Dosen

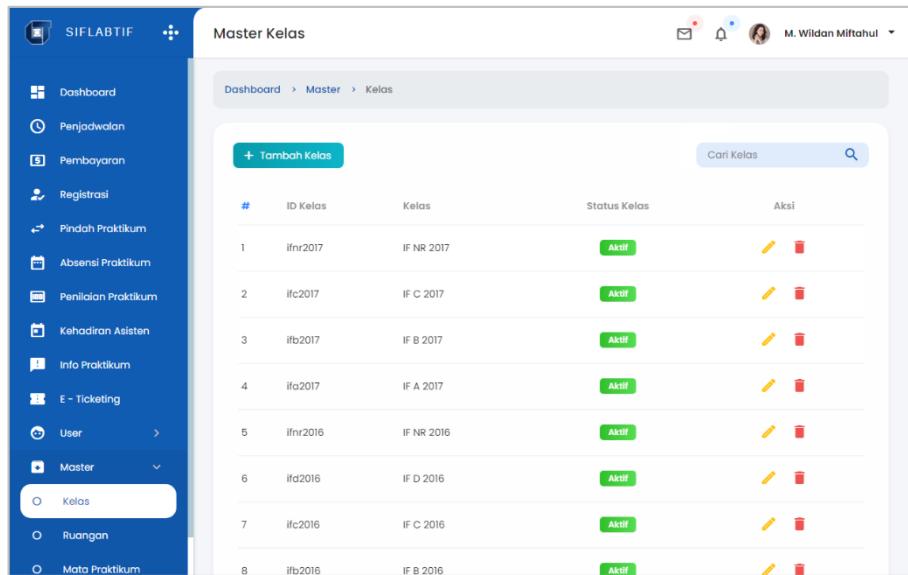
## 21. Halaman Kelola Mahasiswa

The screenshot shows the 'Master Mahasiswa' (Student Management) page. On the left is a sidebar with various menu items. The main area displays a table with eight rows of student information:

#	NPM	Nama	Kelas	Status	Aksi
1	5520117120	Amirul Fadillah	IF NR 2017	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
2	5520117119	Muldani	IF NR 2017	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
3	5520117118	Heru Dwi Permana	IF NR 2017	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
4	5520117117	Yoga Muhamad Rimba A.	IF NR 2017	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
5	5520117116	Nabela Suryana	IF NR 2017	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
6	5520117115	Ilyas Muzakir Harahap	IF NR 2017	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
7	5520117114	Dandi Rustandi	IF NR 2017	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>
8	5520117113	Ni'man Maulana	IF NR 2017	<span>Aktif</span>	<span>Lock</span> <span>Edit</span> <span>Delete</span>

Gambar 4. 21 Halaman Kelola Mahasiswa

## 22. Halaman Kelola Kelas

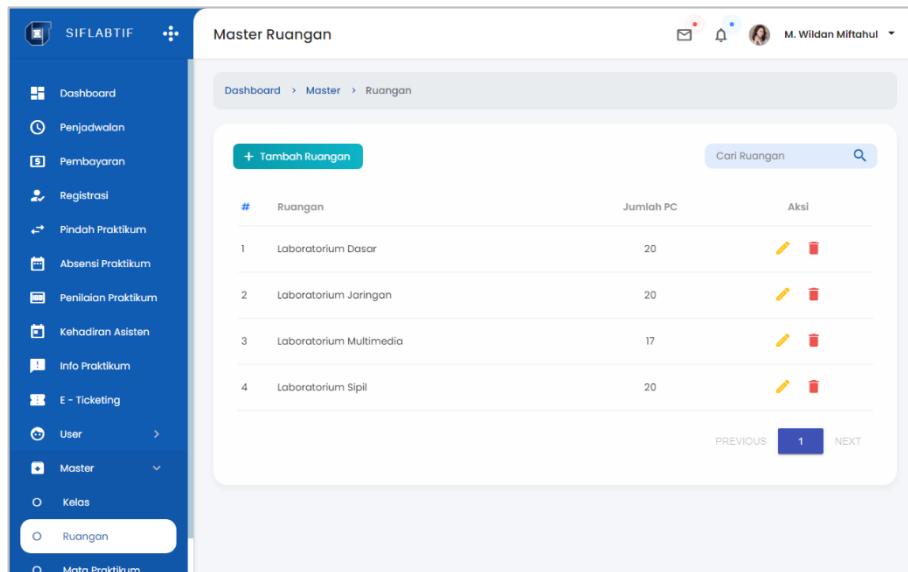


The screenshot shows the 'Master Kelas' (Class Management) page. On the left is a sidebar with various menu items. The main area has a header 'Master Kelas' and a breadcrumb 'Dashboard > Master > Kelas'. Below is a table with the following data:

#	ID Kelas	Kelas	Status Kelas	Aksi
1	ifnr2017	IF NR 2017	Aktif	
2	ifc2017	IF C 2017	Aktif	
3	ifb2017	IF B 2017	Aktif	
4	ifa2017	IF A 2017	Aktif	
5	ifnr2016	IF NR 2016	Aktif	
6	ifd2016	IF D 2016	Aktif	
7	ifc2016	IF C 2016	Aktif	
8	ifb2016	IF B 2016	Aktif	

Gambar 4. 22 Halaman Kelola Kelas

## 23. Halaman Kelola Ruangan

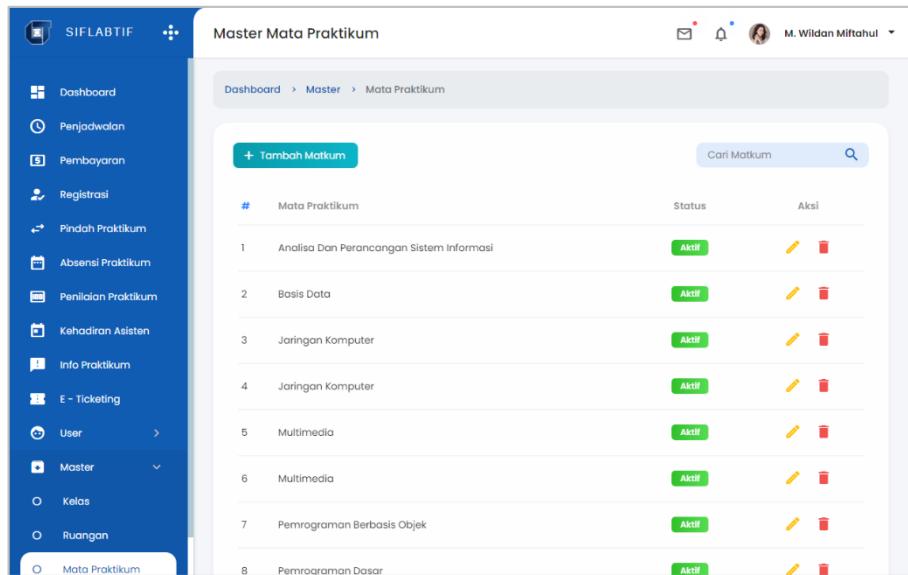


The screenshot shows the 'Master Ruangan' (Room Management) page. On the left is a sidebar with various menu items. The main area has a header 'Master Ruangan' and a breadcrumb 'Dashboard > Master > Ruangan'. Below is a table with the following data:

#	Ruangan	Jumlah PC	Aksi
1	Laboratorium Dasar	20	
2	Laboratorium Jaringan	20	
3	Laboratorium Multimedia	17	
4	Laboratorium Sipil	20	

Gambar 4. 23 Halaman Kelola Ruangan

## 24. Halaman Kelola Mata Praktikum

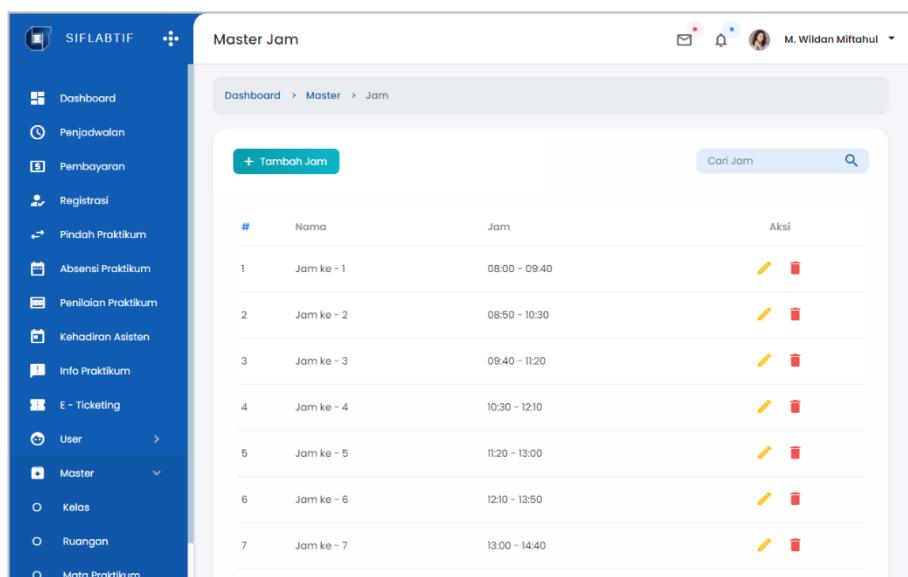


The screenshot shows the 'Master Mata Praktikum' (Practical Subject Management) page. The left sidebar has a dark blue background with white icons and text. The main area has a light gray header with the title 'Master Mata Praktikum'. Below the header is a breadcrumb navigation: 'Dashboard > Master > Mata Praktikum'. A teal button labeled '+ Tambah Matkul' is at the top left. To its right is a search bar with the placeholder 'Cari Matkul' and a magnifying glass icon. The main content is a table with columns: '#', 'Mata Praktikum', 'Status', and 'Aksi'. The table lists eight practical subjects, all marked as 'Aktif' (Active). Each row has edit and delete icons.

#	Mata Praktikum	Status	Aksi
1	Analisa Dan Perancangan Sistem Informasi	Aktif	 
2	Basis Data	Aktif	 
3	Jaringan Komputer	Aktif	 
4	Jaringan Komputer	Aktif	 
5	Multimedia	Aktif	 
6	Multimedia	Aktif	 
7	Pemrograman Berbasis Objek	Aktif	 
8	Pemrograman Dasar	Aktif	 

Gambar 4. 24 Halaman Kelola Mata Praktikum

## 25. Halaman Kelola Jam



The screenshot shows the 'Master Jam' (Time Slot Management) page. The left sidebar has a dark blue background with white icons and text. The main area has a light gray header with the title 'Master Jam'. Below the header is a breadcrumb navigation: 'Dashboard > Master > Jam'. A teal button labeled '+ Tambah Jam' is at the top left. To its right is a search bar with the placeholder 'Cari Jam' and a magnifying glass icon. The main content is a table with columns: '#', 'Nama', 'Jam', and 'Aksi'. The table lists seven time slots, each starting with 'Jam ke -'. The times are: 08:00 - 08:40, 08:50 - 10:30, 09:40 - 11:20, 10:30 - 12:10, 11:20 - 13:00, 12:10 - 13:50, and 13:00 - 14:40. Each row has edit and delete icons.

#	Nama	Jam	Aksi
1	Jam ke - 1	08:00 - 08:40	 
2	Jam ke - 2	08:50 - 10:30	 
3	Jam ke - 3	09:40 - 11:20	 
4	Jam ke - 4	10:30 - 12:10	 
5	Jam ke - 5	11:20 - 13:00	 
6	Jam ke - 6	12:10 - 13:50	 
7	Jam ke - 7	13:00 - 14:40	 

Gambar 4. 25 Halaman Kelola Jam

## 26. Halaman Kelola Jabatan

#	Jabatan	Aksi
1	Koordinator Lab	
2	Koordinator Teknis	
3	Wakil Koordinator Teknis	
4	Sekretaris	
5	Bendahara	
6	Admin Sistem	
7	PJ Lab Dasar	

Gambar 4. 26 Halaman Kelola Jabatan

## 27. Halaman Nilai Praktikum (Dosen)

#	NPM	Nama Mahasiswa	Kehadiran 30 %	Tugas 30 %	Ujian 40 %	Total
1	5520115047	Agni Fodhil Abdurrahim	80	85	80	81,5
2	5520115046	Andi Wijaya Suherman	90	80	70	79
3	5520115039	Andri Mayanti Wahyudi	90	80	75	81
4	5520115054	Asep Jomaludin	100	95	100	98,5
5	5520115053	Billy Mulyana	0	0	0	0
6	5520115051	Cecen Rahman	90	70	70	73

Gambar 4. 27 Halaman Nilai Praktikum (Dosen)

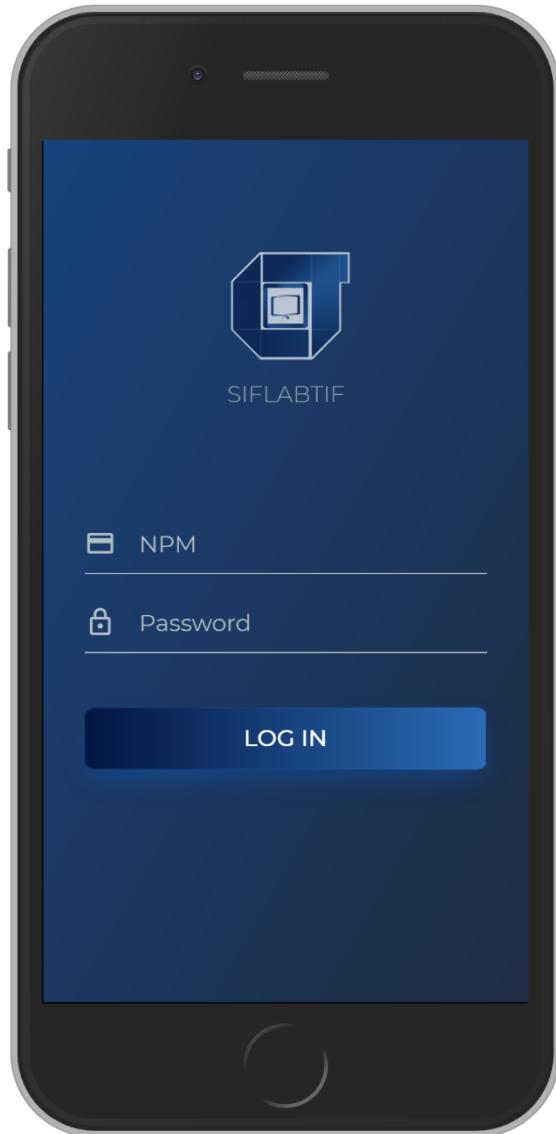
## 28. Halaman Kritik & Saran

The screenshot shows the 'Kritik & Saran' (Feedback & Suggestions) page of the SIFLABTIF application. The left sidebar has a blue background with white icons and text: Dashboard, Absensi Praktikum, Penilaian Praktikum, Tugas Praktikum, Modul Praktikum, Kehadiran Asisten, Kritik & Saran (highlighted with a white circle), and Pesan. The main content area has a white background with a header 'Kritik & Saran' and a user profile for 'Reynaldy Miftahuddin Fauzi'. Below is a table with two rows of feedback. Row 1: #1, Tanggal: 2020-08-17 19:16:39, Pesan: Pembelajaran terlalu fokus ke modul tanpa penjelasan lebih lanjut. Sebaiknya penjelasan dilakukan dengan presentasi, Status: Diterima. Row 2: #2, Tanggal: 2020-08-17 19:16:02, Pesan: Aplikasinya bagus sangat membantu, Status: Belum. Navigation buttons at the bottom include PREVIOUS, a page number '1' in a blue box, and NEXT.

#	Tanggal	Pesan	Status
1	2020-08-17 19:16:39	Pembelajaran terlalu fokus ke modul tanpa penjelasan lebih lanjut. Sebaiknya penjelasan dilakukan dengan presentasi	Diterima
2	2020-08-17 19:16:02	Aplikasinya bagus sangat membantu	Belum

Gambar 4. 28 Halaman Kritik & Saran

## 29. Halaman Login Client



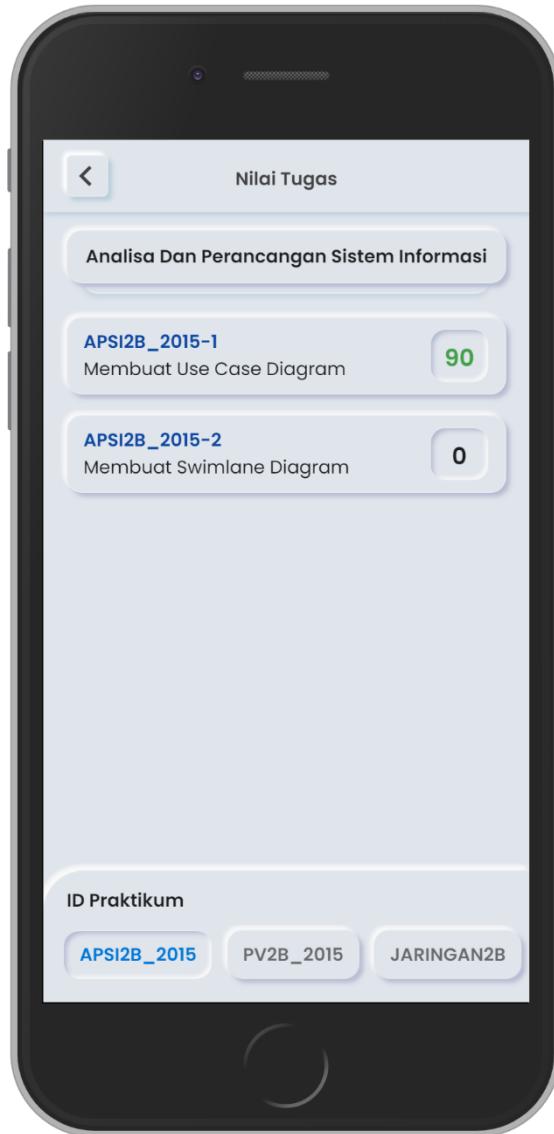
Gambar 4. 29 Halaman Login Client

### 30. Halaman Dashboard Client



Gambar 4. 30 Halaman Dashboard Client

### 31. Halaman Nilai Praktikum (Client)



Gambar 4. 31 Halaman Nilai Praktikum (Client)

### 32. Halaman Absensi Praktikum (Client)



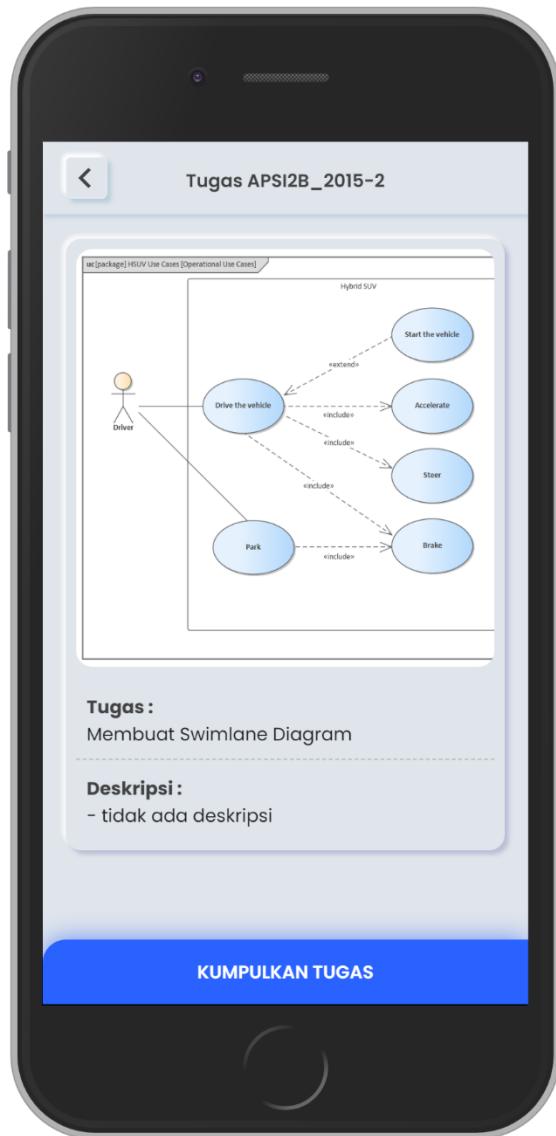
Gambar 4. 32 Halaman Absensi Praktikum (Client)

### 33. Halaman Tugas Praktikum



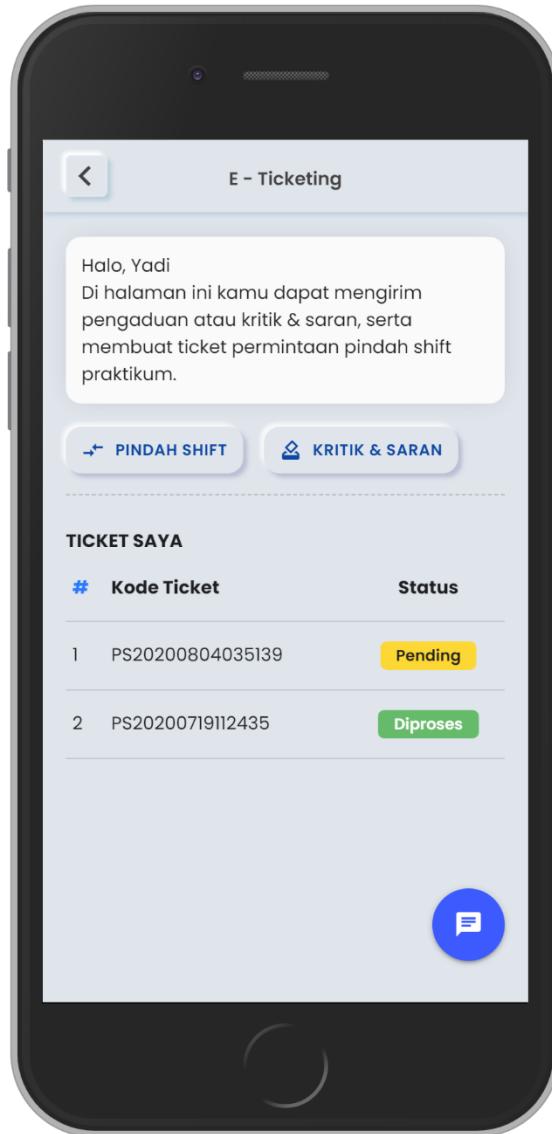
Gambar 4. 33 Halaman Tugas Praktikum

## 34. Halaman Detail Tugas



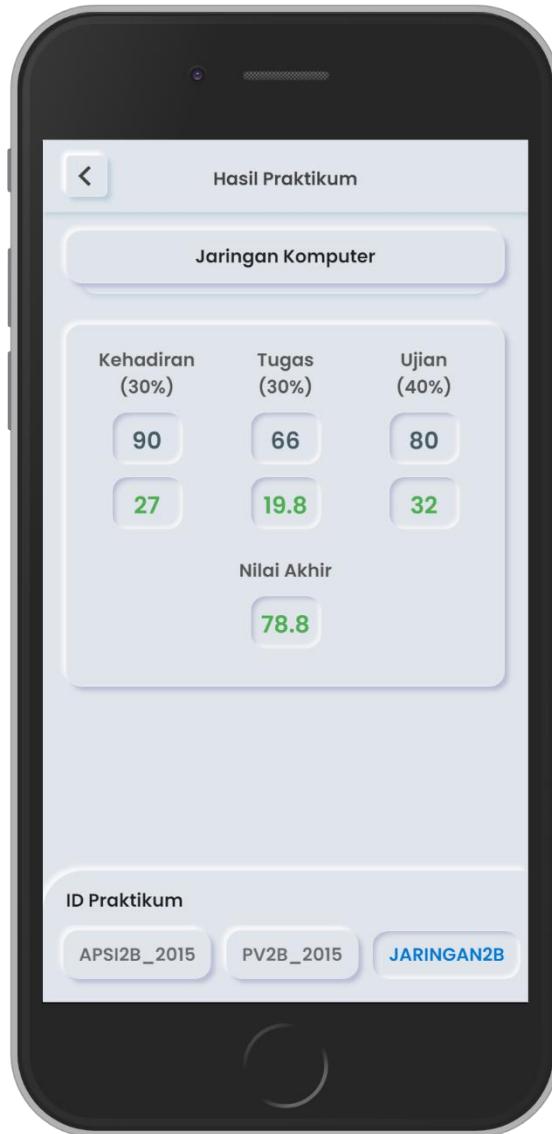
Gambar 4. 34 Halaman Detail Tugas

## 35. Halaman E-Ticketing



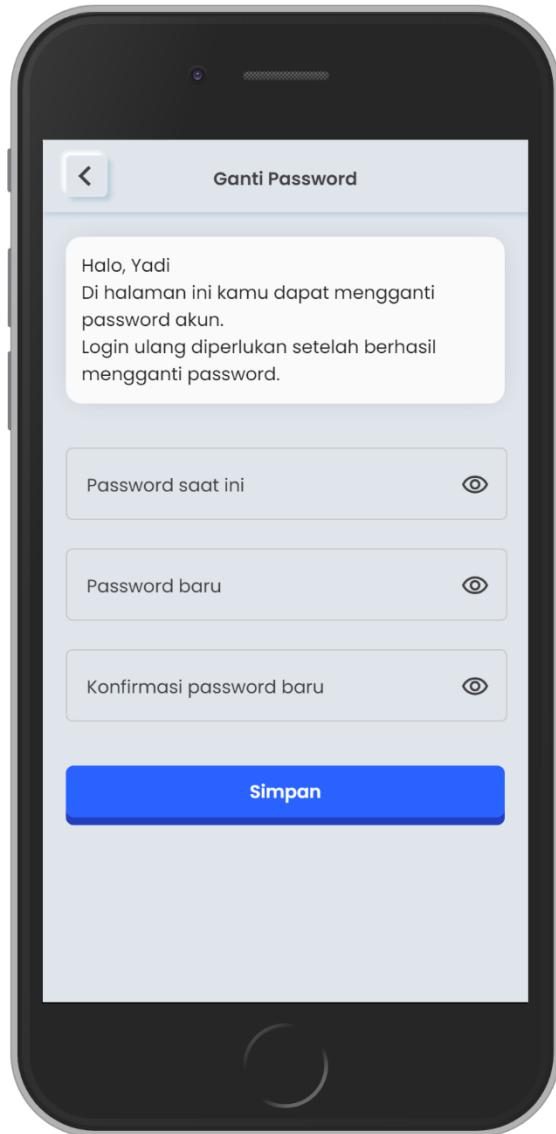
Gambar 4. 35 Halaman E-Ticketing

### 36. Halaman Hasil Praktikum



Gambar 4. 36 Halaman Hasil Praktikum

### 37. Halaman Ganti Password (Client)



Gambar 4. 37 Halaman Ganti Password (Client)

## 38. Halaman Info Praktikum



Gambar 4. 38 Halaman Info Praktikum

### 4.3. Pengujian Sistem

Pengujian dilakukan dengan menggunakan metode *Black Box Testing*. Yaitu dengan menguji antarmuka, fungsi-fungsi dan *form* yang tersedia pada setiap halaman aplikasi.

#### 1. Pengujian Form Login (Server)

*Tabel 4. 1 Hasil Pengujian Form Login (Server)*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	<i>Username atau Password</i> tidak diisi dan menekan tombol Login	Username: (kosong)	Sistem menampilkan pesan “Username harus diisi”	Sesuai
		Password: (kosong)	Sistem menampilkan pesan “Password harus diisi”	Sesuai
2.	Mengisi <i>username</i> dengan ID yang tidak terdaftar	Username: 171771 Password: 12345	Sistem menampilkan pesan “Username tidak terdaftar”	Sesuai
3.	Mengisi <i>username</i> dengan ID yang terdaftar dengan <i>password</i> yang salah	Username: 172005 Password: qwerty12	Sistem menampilkan pesan “Password salah”	Sesuai
4.	<i>Login</i> dengan akun yang sudah tidak aktif	Username: 172005	Sistem menampilkan pesan “Maaf status mengajar anda sudah tidak aktif”	Sesuai

		Password: 12345  Status mengajar: tidak aktif		
5.	<i>Login</i> dengan akun yang masih aktif	Username: 172005  Password: 12345  Status mengajar: aktif	Sistem mengalihkan <i>user</i> ke halaman dashboard	Sesuai

## 2. Pengujian Absensi Praktikum

*Tabel 4. 2 Hasil Pengujian Absensi Praktikum*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Menekan tombol Hadir pada kolom kehadiran mahasiswa	-	Mengubah status kehadiran menjadi Tidak Hadir	Sesuai
2.	Menekan tombol Tidak Hadir kolom kehadiran mahasiswa	-	Mengubah status kehadiran menjadi Hadir	Sesuai

3.	Menekan tombol Hadir atau Tidak Hadir pada kolom kehadiran asisten	-	Menampilkan modal <i>form</i> konfirmasi absensi asisten	Sesuai
4.	Mengisi <i>password</i> dengan <i>password</i> yang salah	Password: qwertty123	Menampilkan pesan password salah	Sesuai
5.	Mengisi <i>password</i> dengan <i>password</i> yang benar	Password: 12345	Mengubah status kehadiran	Sesuai

### 3. Pengujian Penilaian Praktikum

*Tabel 4. 3 Hasil Pengujian Penilaian Praktikum*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Menekan nilai tugas atau ujian	-	Menampilkan input nilai tugas dan ujian	Sesuai
2.	Mengisi input nilai atau ujian kemudian menekan area diluar tabel	-	Sistem menghitung nilai keseluruhan dan mengubah total nilai	Sesuai

#### 4. Pengujian Fungsi Kelola Tugas Praktikum

*Tabel 4. 4 Hasil Pengujian Fungsi Kelola Tugas Praktikum*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Tugas</i>				
1.	Menekan tombol Buat Tugas	-	Menampilkan modal tambah tugas	Sesuai
2.	Mengisi semua kolom inputan tugas kemudian menekan tombol simpan	-	Sistem menyimpan data tugas dan menampilkan pesan tambah tugas berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan tugas kemudian menekan tombol simpan	Jam: (kosong)	Menampilkan pesan data harus diisi	Sesuai
<i>Edit Tugas</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit tugas	Sesuai
2.	Mengubah isi kolom tugas kemudian menekan tombol simpan	-	Sistem menyimpan data tugas terbaru dan menampilkan pesan edit tugas berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan	Jam: (kosong)	Menampilkan pesan data harus diisi	Sesuai

	tugas kemudian menekan tombol simpan			
<i>Hapus Tugas</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus tugas	Sesuai
2.	Menekan tombol hapus tugas pada modal konfirmasi hapus	-	Sistem menghapus data tugas dan menampilkan pesan hapus tugas berhasil	Sesuai
<i>Download &amp; Hapus File Tugas</i>				
1.	Menekan tombol download	-	Browser mengunduh file tugas praktikum	Sesuai
2.	Memilih checkbox dan menekan tombol hapus file	-	Menampilkan konfirmasi hapus file	Sesuai
3.	Menekan tombol hapus pada modal hapus file	-	Sistem menghapus file tugas yang dipilih	Sesuai
<i>Penilaian Tugas</i>				
1.	Menekan kolom nilai tugas	-	Menampilkan kolom input nilai tugas	Sesuai

2.	Mengisi nilai tugas dan menekan area diluar nilai tugas	-	Sistem menyimpan nilai tugas	Sesuai
----	---	---	------------------------------	--------

## 5. Pengujian Ubah Status Pengambilan Modul

*Tabel 4. 5 Hasil Pengujian Ubah Status Pengambilan Modul*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Menekan tombol Sudah	-	Mengubah status pengambilan menjadi Belum	Sesuai
2.	Menekan tombol Belum	-	Mengubah status pengambilan menjadi Hadir	Sesuai

## 6. Pengujian Perpesanan

*Tabel 4. 6 Hasil Pengujian Perpesanan*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Memilih kontak	-	Menampilkan riwayat percakapan dengan kontak terpilih	Sesuai
2.	Mengisi kolom pesan dan menekan tombol kirim	-	Sistem menyimpan pesan	Sesuai

## 7. Pengujian Form Edit Profil

*Tabel 4. 7 Hasil Pengujian Edit Profil*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Mengubah data profil dan menekan tombol simpan	-	Sistem menyimpan profil dan menampilkan pesan profil berhasil diedit	Sesuai
2.	Mengosongkan salah satu kolom inputan kemudian menekan tombol simpan	-	Menampilkan pesan data harus diisi	Sesuai

## 8. Pengujian Form Ganti Password (Server)

*Tabel 4. 8 Hasil Pengujian Form Ganti Password (Server)*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Mengisi kolom input password lama dengan password yang salah kemudian menekan tombol Ganti Password	-	Sistem menampilkan pesan password lama tidak sesuai	Sesuai
2.	Mengisi kolom input password lama dengan	-	Sistem mengalihkan user ke halaman login	Sesuai

	password yang benar kemudian menekan tombol Ganti Password			
3.	Membedakan input password baru dengan konfirmasi password	-	Menampilkan pesan konfirmasi password tidak sesuai	Sesuai
4.	Mengosongkan salah satu kolom inputan kemudian menekan tombol Ganti Password	-	Menampilkan pesan data harus diisi	Sesuai

## 9. Pengujian Fungsi Kelola Periode

*Tabel 4. 9 Hasil Pengujian Fungsi Kelola Periode*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Periode</i>				
1.	Menekan tombol Tambah Periode	-	Menampilkan modal tambah periode	Sesuai
2.	Mengisi semua kolom inputan periode	-	Sistem menyimpan data jadwal dan menampilkan pesan	Sesuai

	kemudian menekan tombol simpan		tambah periode berhasil	
3.	Mengosongkan salah satu kolom inputan tugas kemudian menekan tombol simpan	periode: (kosong)	Menampilkan pesan data harus diisi	Sesuai
<i>Edit Periode</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit periode	Sesuai
2.	Mengubah isi kolom periode kemudian menekan tombol simpan	-	Sistem menyimpan data periode terbaru dan menampilkan pesan edit periode berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan periode kemudian menekan tombol simpan	periode: (kosong)	Menampilkan pesan data harus diisi	Sesuai
<i>Hapus Periode</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus periode	Sesuai
2.	Menekan tombol hapus tugas pada	-	Sistem menghapus data periode dan menampilkan pesan	Sesuai

	modal konfirmasi hapus		hapus periode berhasil	
<i>Status Periode</i>				
1.	Menekan tombol Aktif	-	Mengubah status periode dan jadwal dengan periode terkait menjadi Tidak Aktif	Sesuai
2.	Menekan tombol Tidak Aktif	-	Mengubah status periode dan jadwal dengan periode terkait menjadi Aktif	Sesuai

## 10. Pengujian Fungsi Kelola Jadwal

*Tabel 4. 10 Hasil Pengujian Fungsi Kelola Jadwal*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Jadwal</i>				
1.	Menekan tombol Tambah Jadwal	-	Menampilkan modal tambah jadwal	Sesuai
2.	Mengisi semua kolom inputan jadwal kemudian menekan tombol simpan	-	Sistem menyimpan data jadwal dan menampilkan pesan tambah jadwal berhasil	Sesuai
3.	Mengosongkan salah satu ruangan: (kosong)		Menampilkan pesan data harus diisi	Sesuai

	kolom inputan jadwal kemudian menekan tombol simpan			
4.	Mengisi ID praktikum dengan ID yang sudah ada atau sama	-	Sistem menampilkan pesan ID praktikum sudah ada	Sesuai
<i>Edit Jadwal</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit jadwal	Sesuai
2.	Mengubah isi kolom jadwal kemudian menekan tombol simpan	-	Sistem menyimpan data jadwal terbaru dan menampilkan pesan edit jadwal berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan jadwal kemudian menekan tombol simpan	ruangan: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengubah ID praktikum dengan ID yang sudah ada atau sama	-	Sistem menampilkan pesan ID praktikum sudah ada	Sesuai

<i>Hapus Jadwal</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus jadwal	Sesuai
2.	Menekan tombol hapus jadwal pada modal konfirmasi hapus	-	Sistem menghapus data jadwal dan menampilkan pesan hapus jadwal berhasil	Sesuai
<i>Status Jadwal</i>				
1.	Menekan tombol Aktif	-	Mengubah status jadwal menjadi Tidak Aktif	Sesuai
2.	Menekan tombol Tidak Aktif	-	Mengubah status jadwal menjadi Aktif	Sesuai

## 11. Pengujian Ubah Status Pembayaran Praktikum

*Tabel 4. 11 Hasil Pengujian Ubah Status Pembayaran Praktikum*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Menekan tombol Lunas	-	Mengubah status pembayaran menjadi Belum	Sesuai
2.	Menekan tombol Belum	-	Mengubah status pembayaran menjadi Lunas	Sesuai

## 12. Pengujian Registrasi Praktikum

*Tabel 4. 12 Hasil Pengujian Registrasi Praktikum*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Mengisi checkbox	checkbox: checked	Sistem menyimpan registrasi mahasiswa	Sesuai
2.	Mengosongkan checkbox	checkbox: unchecked	Sistem menghapus registrasi mahasiswa	Sesuai

## 13. Pengujian Pindah Praktikum

*Tabel 4. 13 Hasil Pengujian Pindah Praktikum*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Memilih praktikum dengan pertemuan yang berbeda	-	Menampilkan pesan pertemuan tidak sama	Sesuai
2.	Mengisi checkbox kemudian menekan tombol simpan	checkbox: checked	Sistem mengganti registrasi mahasiswa dan menampilkan pesan pindah praktikum berhasil	Sesuai

## 14. Pengujian Fungsi Kelola Info Praktikum

*Tabel 4. 14 Hasil Pengujian Fungsi Kelola Info Praktikum*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Info</i>				

1.	Menekan tombol Tambah info	-	Menampilkan modal tambah info	Sesuai
2.	Mengisi semua kolom inputan info kemudian menekan tombol simpan	-	Sistem menyimpan data info dan menampilkan pesan tambah info berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan info kemudian menekan tombol simpan	caption: (kosong)	Menampilkan pesan data harus diisi	Sesuai
<i>Edit Info</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit info	Sesuai
2.	Mengubah isi kolom caption kemudian menekan tombol simpan	-	Sistem menyimpan data info terbaru dan menampilkan pesan edit info berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan info kemudian menekan tombol simpan	caption: (kosong)	Menampilkan pesan data harus diisi	Sesuai
<i>Hapus Info</i>				

1.	Menekan tombol Hapus	-	Menampilkan modal hapus info	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data info dan menampilkan pesan hapus info berhasil	Sesuai

### 15. Pengujian Fungsi Kelola Asisten

*Tabel 4. 15 Hasil Pengujian Fungsi Kelola Asisten*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Asisten</i>				
1.	Menekan tombol Tambah asisten	-	Menampilkan modal tambah asisten	Sesuai
2.	Mengisi semua kolom inputan asisten kemudian menekan tombol simpan	-	Sistem menyimpan data asisten dan menampilkan pesan tambah asisten berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan asisten kemudian menekan tombol simpan	username: (kosong)	Menampilkan pesan data harus diisi	Sesuai

4.	Mengisi kolom <i>username</i> dengan <i>username</i> yang sama atau sudah ada	-	Menampilkan pesan <i>username</i> sudah ada	Sesuai
<i>Edit Asisten</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit asisten	Sesuai
2.	Mengubah isi kolom asisten kemudian menekan tombol simpan	-	Sistem menyimpan data asisten terbaru dan menampilkan pesan edit asisten berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan dosen kemudian menekan tombol simpan	username: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengubah kolom <i>username</i> dengan <i>username</i> yang sama atau sudah ada	-	Menampilkan pesan <i>username</i> sudah ada	Sesuai
<i>Hapus Asisten</i>				

1.	Menekan tombol Hapus	-	Menampilkan modal hapus asisten	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data asisten dan menampilkan pesan hapus asisten berhasil	Sesuai
<i>Reset Password Asisten</i>				
1.	Menekan tombol Reset	-	Menampilkan modal konfirmasi reset password asisten	Sesuai
2.	Menekan tombol Reset pada modal konfirmasi reset	-	Sistem mengubah <i>password</i> asisten menjadi 12345 dan menampilkan pesan reset <i>password</i> asisten berhasil	Sesuai
<i>Status Asisten</i>				
1.	Menekan tombol Aktif	-	Mengubah status mengajar menjadi Tidak Aktif	Sesuai
2.	Menekan tombol Tidak Aktif	-	Mengubah status mengajar menjadi Aktif	Sesuai

## 16. Pengujian Fungsi Kelola Dosen

*Tabel 4. 16 Hasil Pengujian Fungsi Kelola Dosen*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Dosen</i>				
1.	Menekan tombol Tambah dosen	-	Menampilkan modal tambah dosen	Sesuai
2.	Mengisi semua kolom inputan dosen kemudian menekan tombol simpan	-	Sistem menyimpan data dosen dan menampilkan pesan tambah dosen berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan dosen kemudian menekan tombol simpan	NIDN: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengisi kolom NIDN dengan NIDN yang sama atau sudah ada	-	Menampilkan pesan NIDN sudah ada	Sesuai
<i>Edit Dosen</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit dosen	Sesuai

2.	Mengubah isi kolom dosen kemudian menekan tombol simpan	-	Sistem menyimpan data dosen terbaru dan menampilkan pesan edit dosen berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan dosen kemudian menekan tombol simpan	NIDN: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengubah kolom NIDN dengan NIDN yang sama atau sudah ada	-	Menampilkan pesan NIDN sudah ada	Sesuai
<i>Hapus Dosen</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus asisten	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data dosen dan menampilkan pesan hapus dosen berhasil	Sesuai
<i>Reset Password Dosen</i>				
1.	Menekan tombol Reset	-	Menampilkan modal konfirmasi reset password dosen	Sesuai

2.	Menekan tombol Reset pada modal konfirmasi reset	-	Sistem mengubah <i>password</i> dosen menjadi 12345 dan menampilkan pesan reset <i>password</i> dosen berhasil	Sesuai
<i>Status Dosen</i>				
1.	Menekan tombol Aktif	-	Mengubah status mengajar menjadi Tidak Aktif	Sesuai
2.	Menekan tombol Tidak Aktif	-	Mengubah status mengajar menjadi Aktif	Sesuai

## 17. Pengujian Fungsi Kelola Mahasiswa

*Tabel 4. 17 Hasil Pengujian Fungsi Kelola Mahasiswa*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Mahasiswa</i>				
1.	Menekan tombol Tambah mahasiswa	-	Menampilkan modal tambah mahasiswa	Sesuai
2.	Mengisi semua kolom inputan mahasiswa kemudian menekan tombol simpan	-	Sistem menyimpan data mahasiswa dan menampilkan pesan tambah mahasiswa berhasil	Sesuai
3.	Mengosongkan salah satu	NPM: (kosong)	Menampilkan pesan data harus diisi	Sesuai

	kolom inputan mahasiswa kemudian menekan tombol simpan			
4.	Mengisi kolom NPM dengan NPM yang sama atau sudah ada	-	Menampilkan pesan NPM sudah ada	Sesuai
<i>Edit Mahasiswa</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit mahasiswa	Sesuai
2.	Mengubah isi kolom mahasiswa kemudian menekan tombol simpan	-	Sistem menyimpan data mahasiswa terbaru dan menampilkan pesan edit mahasiswa berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan mahasiswa kemudian menekan tombol simpan	NPM: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengubah kolom NPM dengan NPM	-	Menampilkan pesan NPM sudah ada	Sesuai

	yang sama atau sudah ada			
<i>Hapus Mahasiswa</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus mahasiswa	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data mahasiswa dan menampilkan pesan hapus mahasiswa berhasil	Sesuai
<i>Reset Password Mahasiswa</i>				
1.	Menekan tombol Reset	-	Menampilkan modal konfirmasi reset password mahasiswa	Sesuai
2.	Menekan tombol Reset pada modal konfirmasi reset	-	Sistem mengubah <i>password</i> mahasiswa menjadi 12345 dan menampilkan pesan reset <i>password</i> mahasiswa berhasil	Sesuai
<i>Status Mahasiswa</i>				
1.	Menekan tombol Aktif	-	Mengubah status menjadi Tidak Aktif	Sesuai
2.	Menekan tombol Tidak Aktif	-	Mengubah status menjadi Aktif	Sesuai

## 18. Pengujian Fungsi Kelola Kelas

*Tabel 4. 18 Hasil Pengujian Fungsi Kelola Kelas*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Kelas</i>				
1.	Menekan tombol Tambah Kelas	-	Menampilkan modal tambah kelas	Sesuai
2.	Mengisi semua kolom inputan kelas kemudian menekan tombol simpan	-	Sistem menyimpan data kelas dan menampilkan pesan tambah kelas berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan kelas kemudian menekan tombol simpan	nama: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengisi kolom nama kelas dengan nama kelas yang sama atau sudah ada	-	Menampilkan pesan kelas sudah ada	Sesuai
<i>Edit Kelas</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit kelas	Sesuai
2.	Mengubah isi kolom kelas	-	Sistem menyimpan data kelas terbaru	Sesuai

	kemudian menekan tombol simpan		dan menampilkan pesan edit kelas berhasil	
3.	Mengosongkan salah satu kolom inputan kelas kemudian menekan tombol simpan	nama: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengubah kolom nama kelas dengan nama kelas yang sama atau sudah ada	-	Menampilkan pesan kelas sudah ada	Sesuai
<i>Hapus Kelas</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus kelas	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data kelas dan menampilkan pesan hapus kelas berhasil	Sesuai
<i>Status Kelas</i>				
1.	Menekan tombol Aktif	-	Mengubah status kelas menjadi Tidak Aktif	Sesuai
2.	Menekan tombol Tidak Aktif	-	Mengubah status kelas menjadi Aktif	Sesuai

## 19. Pengujian Fungsi Kelola Mata Praktikum

*Tabel 4. 19 Hasil Pengujian Fungsi Kelola Mata Praktikum*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Matkum</i>				
1.	Menekan tombol Tambah matkum	-	Menampilkan modal tambah matkum	Sesuai
2.	Mengisi semua kolom matkum kemudian menekan tombol simpan	-	Sistem menyimpan data matkum dan menampilkan pesan tambah matkum berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan matkum kemudian menekan tombol simpan	matkum: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengisi kolom matkum dengan matkum yang sama atau sudah ada	-	Menampilkan pesan matkum sudah ada	Sesuai
<i>Edit Matkum</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit matkum	Sesuai

2.	Mengubah isi kolom matkum kemudian menekan tombol simpan	-	Sistem menyimpan data matkum terbaru dan menampilkan pesan edit matkum berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan matkum kemudian menekan tombol simpan	matkum: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengubah kolom matkum dengan matkum yang sama atau sudah ada	-	Menampilkan pesan matkum sudah ada	Sesuai
<i>Hapus Matkum</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus matkum	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data matkum dan menampilkan pesan hapus matkum berhasil	Sesuai
<i>Status Matkum</i>				
1.	Menekan tombol Aktif	-	Mengubah status menjadi Tidak Aktif	Sesuai

2.	Menekan tombol Tidak Aktif	-	Mengubah status menjadi Aktif	Sesuai
----	----------------------------	---	-------------------------------	--------

## 20. Pengujian Fungsi Kelola Ruangan

*Tabel 4. 20 Hasil Pengujian Fungsi Kelola Ruangan*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Ruangan</i>				
1.	Menekan tombolTambah ruangan	-	Menampilkan modal tambah ruangan	Sesuai
2.	Mengisi semua kolom ruangan kemudian menekan tombol simpan	-	Sistem menyimpan data ruangan dan menampilkan pesan tambah ruangan berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan matkum kemudian menekan tombol simpan	ruangan: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengisi kolom ruangan dengan ruangan yang sama atau sudah ada	-	Menampilkan pesan ruangan sudah ada	Sesuai
<i>Edit Ruangan</i>				

1.	Menekan tombol Edit	-	Menampilkan modal edit ruangan	Sesuai
2.	Mengubah isi kolom ruangan kemudian menekan tombol simpan	-	Sistem menyimpan data ruangan terbaru dan menampilkan pesan edit ruangan berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan ruangan kemudian menekan tombol simpan	ruangan: (kosong)	Menampilkan pesan data harus diisi	Sesuai
4.	Mengubah kolom ruangan dengan ruangan yang sama atau sudah ada	-	Menampilkan pesan ruangan sudah ada	Sesuai
<i>Hapus Ruangan</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus ruangan	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data ruangan dan menampilkan pesan hapus ruangan berhasil	Sesuai

## 21. Pengujian Fungsi Kelola Jam

*Tabel 4. 21 Hasil Pengujian Fungsi Kelola Jam*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Jam</i>				
1.	Menekan tombol Tambah jam	-	Menampilkan modal tambah jam	Sesuai
2.	Mengisi semua kolom jam kemudian menekan tombol simpan	-	Sistem menyimpan data jam dan menampilkan pesan tambah jam berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan jam kemudian menekan tombol simpan	jam_mulai: (kosong)	Menampilkan pesan data harus diisi	Sesuai
<i>Edit Jam</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit jam	Sesuai
2.	Mengubah isi kolom jam kemudian menekan tombol simpan	-	Sistem menyimpan data jam terbaru dan menampilkan pesan edit jam berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan	jam_mulai: (kosong)	Menampilkan pesan data harus diisi	Sesuai

	jam kemudian menekan tombol simpan			
<i>Hapus Jam</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus jam	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data jam dan menampilkan pesan hapus jam berhasil	Sesuai

## 22. Pengujian Fungsi Kelola Jabatan

*Tabel 4. 22 Hasil Pengujian Fungsi Kelola Jabatan*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Tambah Jabatan</i>				
1.	Menekan tombol Tambah jabatan	-	Menampilkan modal tambah jabatan	Sesuai
2.	Mengisi semua kolom jabatan kemudian menekan tombol simpan	-	Sistem menyimpan data jabatan dan menampilkan pesan tambah jabatan berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan jabatan kemudian	jabatan: (kosong)	Menampilkan pesan data harus diisi	Sesuai

	menekan tombol simpan			
<i>Edit Jam</i>				
1.	Menekan tombol Edit	-	Menampilkan modal edit jabatan	Sesuai
2.	Mengubah isi kolom jam kemudian menekan tombol simpan	-	Sistem menyimpan data jabatan terbaru dan menampilkan pesan edit jam berhasil	Sesuai
3.	Mengosongkan salah satu kolom inputan jabatan kemudian menekan tombol simpan	jabatan: (kosong)	Menampilkan pesan data harus diisi	Sesuai
<i>Hapus Jabatan</i>				
1.	Menekan tombol Hapus	-	Menampilkan modal hapus jabatan	Sesuai
2.	Menekan tombol hapus pada modal konfirmasi hapus	-	Sistem menghapus data jabatan dan menampilkan pesan hapus jabatan berhasil	Sesuai

### 23. Pengujian Fungsi Kelola E-Ticketing

*Tabel 4. 23 Hasil Pengujian Fungsi Kelola E-Ticketing*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Status Ticket</i>				
1.	Menekan tombol Detail	-	Menampilkan modal Detail tiket	Sesuai
2.	Mengubah status tiket kemudian menekan tombol simpan	-	Sistem menyimpan status tiket terbaru	Sesuai
<i>Balas Pesan</i>				
1.	Memilih pesan	-	Menampilkan modal riwayat perpesanan	Sesuai
2.	Mengisi kolom pesan kemudian menekan tombol kirim	-	Sistem menyimpan pesan	Sesuai

### 24. Pengujian Form Login (Client)

*Tabel 4. 24 Hasil Pengujian Form Login (Client)*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	NPM atau <i>Password</i> tidak diisi dan	NPM: (kosong)	Sistem menampilkan pesan “NPM harus diisi”	Sesuai

	menekan tombol Login	Password: (kosong)	Sistem menampilkan pesan “Password harus diisi”	Sesuai
2.	Mengisi NPM dengan ID yang tidak terdaftar	NPM: 171771  Password: 12345	Sistem menampilkan pesan “NPM tidak terdaftar”	Sesuai
3.	Mengisi NPM dengan ID yang terdaftar dengan <i>password</i> yang salah	NPM: 5520115044  Password: qwerty12	Sistem menampilkan pesan “Password salah”	Sesuai
3.	Mengisi NPM dengan NPM yang terdaftar dengan <i>password</i> yang benar	NPM: 5520115044  Password: 12345	Sistem mengalihkan mahasiswa ke halaman menu	Sesuai

## 25. Pengujian Form Upload Tugas

*Tabel 4. 25 Hasil Pengujian Form Upload Tugas*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Memilih file yang tidak sesuai ketentuan	-	Sistem menampilkan pesan file tidak didukung	Sesuai

2.	Memilih file yang sesuai ketentuan kemudian menekan tombol upload	-	Sistem mengupload file tugas ke server dan menampilkan pesan upload berhasil	Sesuai
----	---	---	--	--------

## 26. Pengujian Fungsi E-Ticketing

*Tabel 4. 26 Hasil Pengujian Fungsi E-Ticketing*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
<i>Permintaan Pindah Shift</i>				
1.	Tidak mengisi semua kolom inputan tiket	-	Sistem menampilkan pesan data harus diisi	Sesuai
<i>Kirim Kritik &amp; Saran</i>				
2.	Mengisi kolom pesan kemudian menekan tombol kirim	-	Sistem mengirim pesan ke server	Sesuai
<i>Chat Admin</i>				
2.	Mengisi kolom pesan kemudian menekan tombol kirim	-	Sistem mengirim pesan ke server	Sesuai

27. Pengujian Form Ganti Password (Client)

*Tabel 4. 27 Hasil Pengujian Form Ganti Password (Client)*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1.	Mengisi kolom input password lama dengan password yang salah kemudian menekan tombol Simpan	-	Sistem menampilkan pesan password lama tidak sesuai	Sesuai
2.	Mengisi kolom input password lama dengan password yang benar kemudian menekan tombol Simpan	-	Sistem mengalihkan <i>user</i> ke halaman login	Sesuai
3.	Membedakan input password baru dengan konfirmasi password	-	Menampilkan pesan konfirmasi password tidak sesuai	Sesuai
4.	Mengosongkan salah satu kolom inputan kemudian menekan tombol Simpan	-	Menampilkan pesan data harus diisi	Sesuai

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil analisis selama melaksanakan Tugas Akhir ini, dapat disimpulkan bahwa :

1. Aplikasi dibuat untuk mempermudah pihak laboratorium dalam mengelola data praktikum.
2. Mempermudah mahasiswa untuk melihat data-data selama kegiatan praktikum.
3. Mempermudah mahasiswa dalam mengumpulkan tugas praktikum.

#### **5.1. Saran**

Untuk meningkatkan pengelolaan dari Aplikasi Kegiatan Praktikum ini, diataranya :

1. Sebagai penunjang kegiatan pengelolaan *Website* maka pengguna yang berperan dalam aplikasi ini minimal dapat mengoperasikan komputer dan bisa menggunakan internet.
2. Diintegrasikan dengan sistem informasi akademik.
3. Diadakan pelatihan asisten dalam menggunakan aplikasi untuk mengelola keseluruhan data.

## DAFTAR PUSTAKA

- Al -Babra Bin Ladjamudin. (2008). *Analisa dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Allan. (2005). *Pengertian Internet dan Asal Usul dari Kata Internet*. Surabaya: Indah.
- Andi. (2009). *Global Positioning System*. Yogyakarta: ANDI.
- Anhar. (2010). *Panduan Menguasai PHP dan MYSQL Secara Otodidak*. Jakarta: Mediakata.
- Arief, M. R. (2011). *Pemrograman Web Dinamis Menggunakan PHP & MYSQL*. Yogyakarta: Andi Offset.
- Arief, R. (2009). *Web Dinamis*.
- Arikunto, S. (2002). *Prosedur Penelitian Suatu Pendekatan Praktek*. Jakarta: PT. Rineka Cipta.
- Ayuliana. (2009). *Testing dan Implementasi*.
- Badan Pusat Statistik*. (2015). Retrieved from Website Badan Pusat Statistik:  
<https://www.bps.go.id/linkTableDinamis/view/id/1133>
- Buyens, J. (2001). *Web Database Development*. Jakarta: Elex Media Komputindo.
- Dhanada, V. (2013). *Analisis Dan Perancangan Sistem Informasi Rawat Jalan Di Rumah Sakit Hewan Universitas Airlangga Surabaya Dengan Metode Berorientasi Objek*. Surabaya: Universitas Airlangga.
- Dhanta, R. (2009). *Pengantar Ilmu Komputer*. Surabaya: INDAH.
- Duncan, P. M. (1959). *The Development and Status of American*. Chicago: University of Chicago Press.
- Ernan Rustiadi, e. a. (2006). Perencanaan dan Pengembangan Wilayah. *Fakultas Pertanian*. IPB, Bogor.
- Jogiyanto. (2009). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi.
- Kadir, A. (2012). *Panduan Praktis Mempelajari Aplikasi Mikokontroler dan*. Yogyakarta: ANDI.

- Ladjamudin, A. B. (2008). *Analisis dan Desain Sistem Informasi*. Klaten: Gava Media.
- Lukmanul, H. (2010). *Membangun Web Berbasis PHP dengan Framework CodeIgniter*. Yogyakarta: Lokomedia.
- Martin, F. (2005). *UML Distilled : A Brief Guide to the Standard Object Modelling Language*. Addison Wesley.
- Mulyanto, A. (2009). *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta: Pustaka Pelajar.
- Murad, D. F. (2013). *Aplikasi Intelligence Website Untuk Penunjang Laporan PAUD Pada Himpaudi Kota Tangerang*. Tangerang: Perguruan Tinggi Raharja.
- Nugroho, A. (2010). *Rekayasa Perangkat Lunak Menggunakan UML dan JAVA* . Yogyakarta: Andi Offset.
- Peter , S., & Yenni , S. (2002). *Kamus Bahasa Indonesia Kontemporer*. Jakarta: Erlangga.
- Peter Salim, Y. S. (2002). *Kamus Besar Bahasa Indonseia*.
- Pipiapioh. (2010). *Kiat Praktis Menjadi Desainer Web Professional*. Jakarta: PT. Elex Media Computindo.
- Politik\_Hukum\_Pertahanan\_dan\_Keamanan*. (2014). Retrieved from  
[www.bappenas.go.id:](http://www.bappenas.go.id/)  
[https://www.bappenas.go.id/files/data/Politik\\_Hukum\\_Pertahanan\\_dan\\_Keamanan/Statistik%20Kriminal%202014.pdf](https://www.bappenas.go.id/files/data/Politik_Hukum_Pertahanan_dan_Keamanan/Statistik%20Kriminal%202014.pdf)
- Pressman, R. ( 2010). Software Engineering. In *Software Engineering* (p. 7). Newyork: McGraw Hill.
- Pressman, R. S. (Software Engineering: A Practitioner's Approach, Fifth Ed.).  
*2001*. New York: McGraw-Hill Book Company.
- Santoro. (2005). *Kamus Besar Bahasa Indonesia*.
- Sinamarta, J. (2010). *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Sutarman. (2009). *Pengantar teknologi Informasi*. Jakarta: Bumi Aksara.
- Turban, M. W. (1999). *Pengenalan Sistem Informasi*. Yogyakarta: Andi.

- Wijayanto, T. (2013). *Analisis Dan Perancangan Sistem Informasi Pemesanan Dan Penjualan Barang Dengan Metode Berorientasi Objek DI U.D Aneka Jaya Surabaya*. Surabaya: Universitas Airlangga.
- Yasin, V. (2012). *Rekayasa Perangkat Lunak Berorientasi Objek*. Jakarta: Mitra Wacana Media.

## **DAFTAR RIWAYAT HIDUP**



#### **a. Data Personal**

NPM : 5520115044  
Nama : Yadi Prasetya  
Tempat / Tgl. Lahir : Cianjur, 11 Juli 1997  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Program Studi : Teknik Informatika  
Alamat Rumah : Kp. Margaluyu RT 03/21 kelurahan Muka,  
kabupaten Cianjur  
Email : yadiprasetya11@gmail.com

### **b. Pendidikan**

Jenjang	Nama Lembaga	Program Studi	Tahun Lulus
SD	SDN Pawenang	-	2009
SMP	SMPN 2 Karangtengah	-	2012
SMA	MAN Cianjur	Ilmu Pengetahuan Alam	2015
PT	Universitas Suryakancana	Teknik Informatika	

Demikian daftar riwayat hidup ini dibuat dengan sebenarnya.

Cianjur, 04 September 2020

Mahasiswa Ybs.

Yadi Prasetya

## **DAFTAR LAMPIRAN**

1. SOURCE CODE APLIKASI SERVER

2. SOURCE CODE APLIKASI CLIENT

## SOURCE CODE APLIKASI SERVER

### Auth.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Auth extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.    }
11.    public function index()
12.    {
13.        $this->load->view('v-login');
14.    }
15.
16.    function login()
17.    {
18.        $role = $this->input->post('role');
19.        $username = $this->input->post('username');
20.        $password = $this->input->post('password');
21.
22.        if ($role == 'asisten') {
23.            $user = $this->db-
>get_where('t_asisten', array('username' => $username))-
>row_array();
24.            if ($user) {
25.                // status aktif
26.                if ($user['status'] == 1)
27.                {
28.                    // password benar
29.                    if (password_verify($password, $user['
password'])) {
30.                        $data = array(
31.                            'asisten_id' => $user[
'asisten_id'],
32.                            'jabatan_id' => $user[
'jabatan_id']
33.                        );
34.
35.                        $this->session-
>set_userdata($data);
36.                            // admin
37.                            if ($user['jabatan_id'] == 6) {
38.                                echo 'sukses|admin/dashboard';
39.                            }
40.                            else if ($user['jabatan_id'] == 12
   ) {
41.                                // BAK
42.                                echo 'sukses|bak/dashboard';
43.                            }

```

```

44.                     else {
45.                         // asisten
46.                         echo 'sukses|asisten/dashboard
47.                     }
48.
49.                 } else {
50.                     // password salah
51.                     echo 'gagal|Password yang anda mas
52.                        ukan salah';
53.                     }
54.                 } else {
55.                     // status tidak aktif
56.                     echo 'status|Maaf, status mengajar and
57.                         a sudah tidak aktif';
58.                     }
59.                 } else {
60.                     // user tidak ada
61.                     echo 'gagal|Username tidak terdaftar';
62.                 }
63.             } else {
64.                 $dosen = $this->db-
65. >get_where('t_dosen', array('username' => $username))-
66. >row_array();
67.             if ($dosen) {
68.                 // status aktif
69.                 if ($dosen['dosen_status'] == 1) {
70.                     // password benar
71.                     if (password_verify($password, $dosen[
72.                         'password'])) {
73.                         $data = array(
74.                             'dosen_id' => $dosen['
75.                               dosen_id'],
76.                             'jabatan_id' => $dosen[
77.                               'jabatan_id']
78.                         );
79.                     } else {
80.                         // status tidak aktif
81.                         echo 'status|Maaf, status mengajar and
82.                             a sudah tidak aktif';
83.                     }
84.                 } else {
85.                     echo 'gagal|NIDN tidak terdaftar';
86.                 }

```

```

87.
88.    }
89.
90.    function logout()
91.    {
92.        $this->session->sess_destroy();
93.        redirect('auth');
94.    }
95.
96. }

```

### Absensi.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Absensi extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $jadwal_kode = $this->input->get('jadwal_kode');
16.        $data['jadwalkosong'] = $this->input-
>get('jadwal_kode');
17.
18.        $data['mahasiswa'] = $this->M_database-
>getRegistrasiMahasiswa(array('registrasi_praktikum.jadwal_k
ode' => $jadwal_kode));
19.        $data['jadwal'] = $this->M_database-
>getWhere('t_jadwal', null, null, array('status' => 1), 'jad
wal_kode ASC');
20.        $data['halaman_title'] = "Absensi Praktikum";
21.        $data['pertemuan'] = $this->M_database-
>cekPertemuan($jadwal_kode);
22.        $data['absen_tgl'] = $this->M_database-
>cekTanggal($jadwal_kode);
23.        // data hide table
24.        $data['visibility'] = '';
25.
26.        if ($jadwal_kode == null) {
27.            $data['visibility'] = 'hide';
28.        } else {
29.            $data['visibility'] = '';
30.        }
31.

```

```

32.           $this->template->load('admin/template', 'admin/v-
absensi', $data);
33.
34.       }
35.
36.       function do_absen()
37.       {
38.           $jadwal_kode = $this->input->post('jadwal_kode');
39.           $npm = $this->input->post('npm');
40.           $kehadiran = $this->input->post('kehadiran');
41.           $pertemuan = $this->input->post('pertemuan');
42.           $tgl = $this->input->post('tgl');
43.
44.           $data = array(
45.               'jadwal_kode' => $jadwal_kode,
46.               'pertemuan' => $pertemuan,
47.               'npm' => $npm,
48.               'absen_tgl' => $tgl
49.           );
50.
51.           // cek absensi
52.           $cek = $this->M_database-
>cekAbsen('t_absensi', $jadwal_kode, $npm, $pertemuan);
53.
54.           if($kehadiran == 'Tidak Hadir' && $cek-
>hadir == 0)
55.           {
56.               $this->M_database-
>insertData('t_absensi', $data);
57.               $badge = '<span class="badge-status badge-
ok">Hadir</span>';
58.           }
59.           else if($kehadiran == 'Hadir' && $cek->hadir == 1)
60.           {
61.               $this->M_database-
>hapusAbsen('t_absensi', 'jadwal_kode', 'npm', 'pertemuan',
$jadwal_kode, $npm, $pertemuan);
62.               $badge = '<span class="badge-status badge-
not">Tidak Hadir</span>';
63.           }
64.           else
65.           {
66.               $badge = '<span class="badge-status badge-
not">Tidak Hadir</span>';
67.           }
68.           echo $badge;
69.       }
70.   }
71.
72.   function do_absen_asisten()
73.   {
74.       $jadwal_kode = $this->input->post('jadwal_kode');
75.       $asisten_id = $this->input->post('asisten_id');
76.       $kehadiran = $this->input->post('kehadiran');
77.       $pertemuan = $this->input->post('pertemuan');
78.       $tgl = $this->input->post('tgl');

```

```

79.      $data = array(
80.          'jadwal_kode' => $jadwal_kode,
81.          'pertemuan' => $pertemuan,
82.          'asisten_id' => $asisten_id,
83.          'absen_tgl' => $tgl
84.      );
85.
86.
87.      // cek absensi
88.      $cek = $this->M_database-
>cekAbsenAsisten('t_absensi_asisten', $jadwal_kode, $asisten_
_id, $pertemuan);
89.
90.      if($kehadiran == 'Tidak Hadir' && $cek-
>hadir == 0) {
91.          $this->M_database-
>insertData('t_absensi_asisten', $data);
92.          $badge = '<span class="badge-status badge-
ok">Hadir</span>';
93.      }
94.      else if($kehadiran == 'Hadir' && $cek-
>hadir == 1) {
95.          $this->M_database-
>hapusAbsen('t_absensi_asisten', 'jadwal_kode', 'asisten_id'
, 'pertemuan', $jadwal_kode, $asisten_id, $pertemuan);
96.          $badge = '<span class="badge-status badge-
not">Tidak Hadir</span>';
97.      }
98.      else{
99.          $badge = '<span class="badge-status badge-
not">Tidak Hadir</span>';
100.     }
101.     echo "$badge";
102. }
103.
104. }

```

### Asisten.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Asisten extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {

```

```

15.      $jumlah_data = $this->M_database-
    >jumlah_data('t_asisten');
16.      $offset = $this->uri->segment(4);
17.
18.      $this->load->library('pagination');
19.      $config['base_url'] = base_url().'admin/asisten/in
    dex/';
20.      $config['total_rows'] = $jumlah_data;
21.      $config['per_page'] = 50;
22.      $config['first_link'] = '...';
23.      $config['last_link'] = '...';
24.      $config['next_link'] = 'NEXT';
25.      $config['prev_link'] = 'PREV';
26.      $config['full_tag_open'] = '<div class="custom-
    pagination">';
27.      $config['full_tag_close'] = '</div>';
28.      $config['num_tag_open'] = '<button class="btn
    transparent z-depth-0" style="padding: 0">';
29.      $config['num_tag_close'] = '</button>';
30.      $config['cur_tag_open'] = '<button class="btn
    indigo" style="padding: 0 24px">';
31.      $config['cur_tag_close'] = '</button>';
32.      $config['next_tag_open'] = '<button class="btn
    transparent z-depth-0 grey-text" style="padding: 0">';
33.      $config['next_tagl_close'] = '</button>';
34.      $config['prev_tag_open'] = '<button class="btn
    transparent z-depth-0 grey-text" style="padding: 0">';
35.      $config['prev_tagl_close'] = '</button>';
36.      $config['first_tagl_open'] = '<button class="btn
    transparent z-depth-0" style="padding: 0">';
37.      $config['first_tagl_close'] = '</button>';
38.      $config['last_tagl_open'] = '<button class="btn
    transparent z-depth-0" style="padding: 0">';
39.      $config['last_tagl_close'] = '</button>';
40.      $this->pagination->initialize($config);
41.      $data['pagination'] = $this->pagination-
    >create_links();
42.
43.      $data['asisten'] = $this->M_database-
    >getAllData('t_asisten', 't_jabatan', 't_asisten.jabatan_id
    = t_jabatan.jabatan_id', 'status DESC', 'asisten_id DESC', $
    config['per_page'], $offset);
44.      $data['jabatan'] = $this->db-
    >select('jabatan_id, jabatan_nama')-
    >where('jabatan_id !=', 1)->where('jabatan_id !=', 13)-
    >get('t_jabatan');
45.      $data['halaman_title'] = "Asisten";
46.      $this->template->load('admin/template', 'admin/v-
    asisten', $data);
47.  }
48.
49.  function view_tambah()
50.  {
51.      $data['asisten'] = $this->M_database-
    >getAllData('t_asisten', 't_jabatan', 't_asisten.jabatan_id
    = t_jabatan.jabatan_id', null);

```

```

52.         $data['jabatan'] = $this->M_database-
>getAllData('t_jabatan', null, null, 'jabatan_nama ASC');
53.         $this->template->load('admin/template', 'admin/v-
asisten-tambah', $data);
54.     }
55.
56.     function view_edit()
57.     {
58.         $asisten_id = $this->uri->segment(4);
59.         $asisten = $this->M_database-
>getWhere('t_asisten', 't_jabatan', 't_asisten.jabatan_id =
t_jabatan.jabatan_id', array('asisten_id' => $asisten_id))->result();
60.         $data['jabatan'] = $this->M_database-
>getAllData('t_jabatan', null, null, 'jabatan_nama ASC');
61.
62.         foreach ($asisten as $key => $value) {
63.             $data['asisten_id'] = $value->asisten_id;
64.             $data['username'] = $value->username;
65.             $data['asisten_nama'] = $value->asisten_nama;
66.             $data['jabatan_id'] = $value->jabatan_id;
67.             $data['jabatan_nama'] = $value->jabatan_nama;
68.             $format = date_create($value->tgl_lahir);
69.             $date = date_format($format, 'm');
70.             $bulan = 'Pilih Bulan';
71.             switch ($date) {
72.                 case '01':
73.                     $bulan = 'Januari';
74.                     break;
75.                 case '02':
76.                     $bulan = 'Februari';
77.                     break;
78.                 case '03':
79.                     $bulan = 'Maret';
80.                     break;
81.                 case '04':
82.                     $bulan = 'April';
83.                     break;
84.                 case '05':
85.                     $bulan = 'Mei';
86.                     break;
87.                 case '06':
88.                     $bulan = 'Juni';
89.                     break;
90.                 case '07':
91.                     $bulan = 'Juli';
92.                     break;
93.                 case '08':
94.                     $bulan = 'Agustus';
95.                     break;
96.                 case '09':
97.                     $bulan = 'September';
98.                     break;
99.                 case '10':
100.                    $bulan = 'Oktober';
101.                   break;

```

```

102.         case '11':
103.             $bulan = 'November';
104.             break;
105.         case '12':
106.             $bulan = 'Desember';
107.             break;
108.         default:
109.             $bulan = 'Pilih Bulan';
110.             break;
111.     }
112.     $data['bulan'] = $bulan;
113.     $tgl_lahir = explode('-', $value->tgl_lahir);
114.     $data['tanggal'] = $tgl_lahir[2];
115.     $data['n_bulan'] = $tgl_lahir[1];
116.     $data['tahun'] = $tgl_lahir[0];
117.     $data['alamat'] = $value->alamat;
118.     $data['email'] = $value->email;
119.     $data['tahun_masuk'] = $value->thn_masuk;
120.     $data['tahun_keluar'] = $value->thn_keluar;
121.     $data['foto'] = $value->foto;
122. }
123.
124.     $this->template->load('admin/template', 'admin/v-
    asisten-edit', $data);
125. }
126.
127. function tambahAsisten() {
128.
129.     $username = $this->input->post('username');
130.     $password = password_hash('12345', PASSWORD_DEFAULT);
131.     $asisten_nama = ucwords($this->input-
    >post('asisten_nama'));
132.     $asisten_foto = $this->input-
    >post('asisten_foto');
133.     $tgl_lahir = $this->input->post('tgl_lahir');
134.     $bulan_lahir = $this->input->post('bulan_lahir');
135.     $tahun_lahir = $this->input->post('tahun_lahir');
136.     $jabatan_id = $this->input->post('jabatan_id');
137.     $alamat = $this->input->post('alamat');
138.     $email = $this->input->post('email');
139.     $tahun_masuk = $this->input->post('tahun_masuk');
140.     $tahun_keluar = $this->input-
    >post('tahun_keluar');
141.
142.     $data = array(
143.         'username' => $username,
144.         'password' => $password,
145.         'asisten_nama' => $asisten_nama,
146.         'tgl_lahir' => $tahun_lahir.'-
    '.$bulan_lahir.'-'.$tgl_lahir,
147.         'jabatan_id' => $jabatan_id,
148.         'alamat' => $alamat,
149.         'email' => $email,
150.         'thn_masuk' => $tahun_masuk,
151.         'thn_keluar' => $tahun_keluar,

```

```

152.           'status' => 1,
153.           'foto' => $asisten_foto
154.       );
155.
156.       $cek_asisten = $this->db-
>where('username', $username)->get('t_asisten')-
>row();
157.       if ($cek_asisten) {
158.           echo "gagal|Username <b>$username</b> sudah ad-
a, data gagal disimpan";
159.       } else {
160.           $this->M_database-
>insertData('t_asisten', $data);
161.           $asisten_akhir = $this->M_database-
>getAllData('t_asisten', 't_jabatan', 't_asisten.jabatan_id
= t_jabatan.jabatan_id', 'asisten_id DESC', null, 1, null)-
>row();
162.           $asisten_1 = $this->M_database-
>cekdata('t_jadwal', 'asisten_1', $asisten_akhir-
>asisten_id);
163.           $asisten_2 = $this->M_database-
>cekdata('t_jadwal', 'asisten_2', $asisten_akhir-
>asisten_id);
164.
165.           $title = 'hapus';
166.           $modal = 'modalHapus';
167.
168.           if ($asisten_1 != 0 || $asisten_2 != 0) {
169.               $title = 'tidak bisa hapus';
170.               $modal = 'modalAlert';
171.           }
172.
173.           $status = '';
174.           if ($asisten_akhir->status == 0){
175.               $status = '<span class="badge-
status badge-not">Tidak aktif</span>';
176.           } else {
177.               $status = '<span class="badge-
status badge-ok">Aktif</span>';
178.           }
179.           $table = '<tr class="green lighten-5">
<td></td>
180.           <td></td>
181.           <td>'.$asisten_akhir-
>asisten_nama.'</td>
182.           <td>'.$asisten_akhir-
>jabatan_nama.'</td>
183.           <td class="status-asisten td-
data center" id="'.$asisten_akhir-
>asisten_id.'">'>'.$status.'</td>
184.
185.           <td class="center">
```

```

186.                               <button class="waves-
    effect mybtn transparent modal-trigger btn-lihat-
    asisten" data-target="modalLihat"
187.                               data-
    asisten_id="'.\$asisten_akhir->asisten_id.'"
188.                               data-
    asisten_nama="'.\$asisten_akhir->asisten_nama.'"
189.                               data-
    jabatan_nama="'.\$asisten_akhir->jabatan_nama.'"
190.                               data-
    username="'.\$asisten_akhir->username.'"
191.                               data-
    tgl_lahir="'.\$asisten_akhir->tgl_lahir.'"
192.                               data-
    alamat="'.\$asisten_akhir->alamat.'"
193.                               data-
    email="'.\$asisten_akhir->email.'"
194.                               data-
    foto="'.base_url().'assets/images/profil/'.\$asisten_akhir-
    >foto.'"
195.                               data-
    tahun_masuk="'.\$asisten_akhir->thn_masuk.'"
196.                               data-
    tahun_keluar="'.\$asisten_akhir-
    >thn_keluar.'"><i class="material-icons-outlined blue-
    text text-accent-1">remove_red_eye</i>
197.                               </button>
198.                               <button class="waves-
    effect mybtn transparent modal-trigger btn-password-
    asisten" data-target="modalPassword"
199.                               data-
    asisten_id="'.\$asisten_akhir->asisten_id.'"
200.                               data-
    asisten_nama="'.\$asisten_akhir-
    >asisten_nama.'"><i class="material-icons-outlined grey-
    text">lock_open</i>
201.                               </button>
202.                               <button class="waves-
    effect mybtn transparent modal-trigger btn-edit-
    asisten" data-target="modalEdit"
203.                               data-
    asisten_id="'.\$asisten_akhir->asisten_id.'"
204.                               data-
    asisten_nama="'.\$asisten_akhir->asisten_nama.'"
205.                               data-
    jabatan_nama="'.\$asisten_akhir->jabatan_nama.'"
206.                               data-
    jabatan_id="'.\$asisten_akhir->jabatan_id.'"
207.                               data-
    username="'.\$asisten_akhir->username.'"
208.                               data-
    tgl_lahir="'.\$asisten_akhir->tgl_lahir.'"
209.                               data-
    alamat="'.\$asisten_akhir->alamat.'"

```

```

210.          data-
  email="'. $asisten_akhir-
 >email.'"
211.          data-
  tahun_masuk="'. $asisten_akhir->thn_masuk.'"
212.          data-
  tahun_keluar="'. $asisten_akhir-
 >thn_keluar.'"><i class="material-icons amber-text">edit</i>
213.          </button>

214.          <button title="'. $title.'" cla-
 ss="waves-effect mybtn transparent modal-trigger btn-hapus-
 asisten" data-target="'. $modal.'"
215.          data-
  asisten_id="'. $asisten_akhir->asisten_id.'"
216.          data-
  asisten_nama="'. $asisten_akhir->asisten_nama.'"
217.          data-
  foto_nama="'. $asisten_akhir->foto.'"><i class="material-
 icons red-text text-lighten-1">delete</i>
218.          </button>
219.          </td>
220.          </tr>';
221.      echo "sukses|Data asisten <b>$asisten_nama</b>
 berhasil disimpan|$table";
222.  }
223.
224. }
225.
226. function editAsisten() {
227.
228.     $asisten_id = $this->input->post('asisten_id');
229.     $username = $this->input->post('username');
230.     $asisten_nama = ucwords($this->input-
 >post('asisten_nama'));
231.     $tgl_lahir = $this->input->post('tgl_lahir');
232.     $bulan_lahir = $this->input->post('bulan_lahir');
233.     $tahun_lahir = $this->input->post('tahun_lahir');
234.     $jabatan_id = $this->input->post('jabatan_id');
235.     $alamat = $this->input->post('alamat');
236.     $email = $this->input->post('email');
237.     $tahun_masuk = $this->input->post('tahun_masuk');
238.     $tahun_keluar = $this->input-
 >post('tahun_keluar');
239.
240.     $data = array(
241.         'username' => $username,
242.         'asisten_nama' => $asisten_nama,
243.         'tgl_lahir' => $tahun_lahir.'-
 '.$bulan_lahir.'-'.$tgl_lahir,
244.         'jabatan_id' => $jabatan_id,
245.         'alamat' => $alamat,
246.         'email' => $email,
247.         'thn_masuk' => $tahun_masuk,
248.         'thn_keluar' => $tahun_keluar
249.     );

```

```

250.
251.         $this->session-
>set_flashdata('edit', "Data asisten <b>$asisten_nama</b> berhasil diedit");
252.         $this->M_database-
>updateData('t_asisten', array('asisten_id' => $asisten_id),
$data);
253.
254.         redirect('admin/asisten', 'refresh');
255.
256.     }
257.
258.     function hapusAsisten()
259.     {
260.         $asisten_id = $this->input->post('asisten_id');
261.         $asisten_nama = $this->input-
>post('asisten_nama');
262.         $foto = $this->input->post('foto');
263.
264.         if ($foto == 'default-profil-
1.jpg' || $foto == 'default-profil-p.jpg') {
265.             // no action
266.         } else {
267.             unlink('assets/images/profil/'.$foto);
268.         }
269.
270.         $this->session-
>set_flashdata('hapus', "Data asisten <b>$asisten_nama</b> berhasil dihapus");
271.         $this->M_database-
>deleteData('t_asisten', array('asisten_id' => $asisten_id));
;
272.
273.         redirect('admin/asisten', 'refresh');
274.     }
275.
276.     function statusAsisten()
277.     {
278.         $asisten_id = $this->input->post('asisten_id');
279.         $status = $this->input->post('status');
280.
281.         if ($status == 'Aktif') {
282.             $data = array('status' => 0);
283.             $badge = '<span class="badge-status badge-
not">Tidak aktif</span>';
284.         } else {
285.             $data = array('status' => 1);
286.             $badge = '<span class="badge-status badge-
ok">Aktif</span>';
287.         }
288.
289.         $this->M_database-
>updateData('t_asisten', array('asisten_id' => $asisten_id),
$data);
290.
291.         echo $badge;

```

```

292.     }
293.
294.     function resetPassword()
295.     {
296.         $asisten_id = $this->input->post('asisten_id');
297.         $asisten_nama = $this->input-
298.             >post('asisten_nama');
299.         $password = password_hash('12345', PASSWORD_DEFAULT);
300.     }
301.     $data = array(
302.         'password' => $password
303.     );
304.     $this->session-
305.         >set_flashdata('edit', "Reset password asisten <b>$asisten_n
ama</b> berhasil");
306.         $this->M_database-
307.             >updateData('t_asisten', array('asisten_id' => $asisten_id),
308.             $data);
309.
310. }

```

## Dashboard.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Dashboard extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model("M_database");
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $data['jumlah_mhs'] = $this->M_database-
16.            >countWhere('t_mahasiswa', 'status_mhs = 1');
17.        $data['jumlah_kelas'] = $this->M_database-
18.            >countWhere('t_kelas', 'kelas_status = 1');
19.        $data['jumlah_jadwal'] = $this->M_database-
20.            >countWhere('t_jadwal', 'status = 1');
21.        $data['jumlah_asisten'] = $this->M_database-
22.            >countWhere('t_asisten', array('status' => 1, 'jabatan_id !=
' => 12));
23.        $data['halaman_title'] = "Dashboard";

```

```

20.         $this->template->load('admin/template', 'admin/v-
    dashboard', $data);
21.     }
22.
23. }

```

## Dosen.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Dosen extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $jumlah_data = $this->M_database-
>jumlah_data('t_dosen');
16.        $offset = $this->uri->segment(4);
17.
18.        $this->load->library('pagination');
19.        $config['base_url'] = base_url().'admin/dosen/inde
x/';
20.        $config['total_rows'] = $jumlah_data;
21.        $config['per_page'] = 50;
22.        $config['first_link']      = '...';
23.        $config['last_link']       = '...';
24.        $config['next_link']       = 'NEXT';
25.        $config['prev_link']       = 'PREV';
26.        $config['full_tag_open']   = '<div class="custom-
pagination">';
27.        $config['full_tag_close']  = '</div>';
28.        $config['num_tag_open']    = '<button class="btn
transparent z-depth-0" style="padding: 0">';
29.        $config['num_tag_close']   = '</button>';
30.        $config['cur_tag_open']    = '<button class="btn
indigo" style="padding: 0 24px">';
31.        $config['cur_tag_close']   = '</button>';
32.        $config['next_tag_open']   = '<button class="btn
transparent z-depth-0 grey-text" style="padding: 0">';
33.        $config['next_tagl_close'] = '</button>';
34.        $config['prev_tag_open']   = '<button class="btn
transparent z-depth-0 grey-text" style="padding: 0">';
35.        $config['prev_tagl_close'] = '</button>';
36.        $config['first_tagl_open'] = '<button class="btn
transparent z-depth-0" style="padding: 0">';
37.        $config['first_tagl_close'] = '</button>';

```

```

38.      $config['last_tag_open']    = '<button class="btn
transparent z-depth-0" style="padding: 0">';
39.      $config['last_tagl_close']  = '</button>';
40.      $this->pagination->initialize($config);
41.      $data['pagination'] = $this->pagination-
>create_links();
42.
43.      $data['dosen'] = $this->M_database-
>getAllData('t_dosen', 't_jabatan', 't_dosen.jabatan_id = t_
jabatan.jabatan_id', 'dosen_status DESC', 'dosen_id DESC', $_
config['per_page'], $offset);
44.      $data['jabatan'] = $this->db-
>select('jabatan_id, jabatan_nama')->where('jabatan_id', 1)-
>or_where(array('jabatan_id' => 13))->get('t_jabatan');
45.      $data['halaman_title'] = "Master Dosen";
46.      $this->template->load('admin/template', 'admin/v-
dosen', $data);
47. }
48.
49. function tambahDosen() {
50.
51.     $username = $this->input->post('username');
52.     $password = password_hash('12345', PASSWORD_DEFAULT);
53.     $dosen_nama = ucwords($this->input-
>post('dosen_nama'));
54.     $dosen_foto = $this->input->post('dosen_foto');
55.     $jabatan_id = $this->input->post('jabatan_id');
56.     $email = $this->input->post('email');
57.
58.     $data = array(
59.             'username' => $username,
60.             'password' => $password,
61.             'dosen_nama' => $dosen_nama,
62.             'jabatan_id' => $jabatan_id,
63.             'email' => $email,
64.             'dosen_status' => 1,
65.             'foto' => $dosen_foto
66.         );
67.
68.     $cek_dosen = $this->db-
>where('username', $username)->get('t_dosen')-
>row();
69.     if ($cek_dosen) {
70.         echo "gagal|Username <b>$username</b> sudah ad-
a, data gagal disimpan";
71.     } else {
72.         $this->M_database-
>insertData('t_dosen', $data);
73.         $dosen_akhir = $this->M_database-
>getAllData('t_dosen', 't_jabatan', 't_dosen.jabatan_id = t_
jabatan.jabatan_id', 'dosen_id DESC', null, 1, null)-
>row();
74.         $dosen = $this->M_database-
>cekdata('t_jadwal', 'dosen_id', $dosen_akhir->dosen_id);
75.         $title = 'hapus';

```

```

76.          $modal = 'modalHapus';
77.
78.          if ($dosen) {
79.              $title = 'tidak bisa hapus';
80.              $modal = 'modalAlert';
81.          }
82.          $status = '';
83.          if ($dosen_akhir->dosen_status == 0) {
84.              $status = '<span class="badge-
    status badge-not">Tidak aktif</span>';
85.          } else {
86.              $status = '<span class="badge-
    status badge-ok">Aktif</span>';
87.          }
88.          $table = '<tr class="green lighten-5">
89.                      <td></td>

90.                      <td></td>
91.                      <td>'.$dosen_akhir-
    >dosen_nama.'</td>
92.                      <td>'.$dosen_akhir-
    >jabatan_nama.'</td>
93.                      <td>'.$dosen_akhir-
    >username.'</td>
94.                      <td>'.$dosen_akhir->email.'</td>
95.                      <td class="status-dosen td-
    data_center" id="'.$dosen_akhir-
    >dosen_id.'">'.$status.'</td>

96.          <td class="center">
97.              <button class="waves-
    effect mybtn transparent modal-trigger btn-password-
    dosen" data-target="modalPassword"
98.                  data-dosen_id="'.$dosen_akhir-
    >dosen_id.'"
99.                  data-
    dosen_nama="'.$dosen_akhir-
    >dosen_nama.'"><i class="material-icons-outlined grey-
    text">lock_open</i>
100.                 </button>
101.                 <button class="waves-
    effect mybtn transparent modal-trigger btn-edit-dosen" data-
    target="modalEdit"
102.                     data-
    dosen_id="'.$dosen_akhir->dosen_id.'"
103.                     data-
    dosen_nama="'.$dosen_akhir->dosen_nama.'"
104.                     data-
    jabatan_nama="'.$dosen_akhir->jabatan_nama.'"
105.                     data-
    jabatan_id="'.$dosen_akhir->jabatan_id.'"
106.                     data-
    username="'.$dosen_akhir->username.'"

```

```

107.           data-
email="'.\$dosen_akhir->email.'"><i class="material-
icons amber-text">edit</i>
108.         </button>

109.           <button title="'.\$title.'" cla-
ss="waves-effect mybtn transparent modal-trigger btn-hapus-
dosen" data-target="'.\$modal.'"
110.           data-
dosen_id="'.\$dosen_akhir->dosen_id.'"
111.           data-
dosen_nama="'.\$dosen_akhir->dosen_nama.'"
112.           data-
foto_nama="'.\$dosen_akhir->foto.'"><i class="material-
icons red-text text-lighten-1">delete</i>
113.         </button>
114.       </td>
115.     </tr>';
116.   echo "sukses|Data dosen <b>\$dosen_nama</b> ber-
hasil disimpan|\$table";
117. }
118.
119. }
120.
121. function editdosen() {
122.
123.   \$dosen_id = \$this->input->post('dosen_id');
124.   \$username = \$this->input->post('username');
125.   \$dosen_nama = ucwords(\$this->input-
>post('dosen_nama'));
126.   \$jabatan_id = \$this->input->post('jabatan_id');
127.   \$email = \$this->input->post('email');
128.
129.   \$data = array(
130.     'username' => \$username,
131.     'dosen_nama' => \$dosen_nama,
132.     'jabatan_id' => \$jabatan_id,
133.     'email' => \$email,
134.   );
135.
136.   \$this->session-
>set_flashdata('edit', "Profil <b>\$dosen_nama</b> berhasil d
iedit");
137.   \$this->M_database-
>updateData('t_dosen', array('dosen_id' => \$dosen_id), \$data
);
138.
139.   redirect('admin/dosen','refresh');
140.
141. }
142.
143. function hapusdosen()
144. {
145.   \$dosen_id = \$this->input->post('dosen_id');
146.   \$dosen_nama = ucwords(\$this->input-
>post('dosen_nama'));

```

```

147.         $foto = $this->input->post('foto');
148.
149.         if ($foto == 'default-profil-
150.             1.jpg' || $foto == 'default-profil-p.jpg') {
151.             // no action
152.         } else {
153.             unlink('assets/images/profil/'.$foto);
154.         }
155.         $this->session-
156.         >set_flashdata('hapus', "Data dosen <b>$dosen_nama</b> berha
157.             sil dihapus");
158.         $this->M_database-
159.         >deleteData('t_dosen', array('dosen_id' => $dosen_id));
160.
161.         function statusdosen()
162.     {
163.         $dosen_id = $this->input->post('dosen_id');
164.         $status = $this->input->post('status');
165.
166.         if ($status == 'Aktif') {
167.             $data = array('dosen_status' => 0);
168.             $badge = '<span class="badge-status badge-
169.                 not">Tidak aktif</span>';
170.         } else {
171.             $data = array('dosen_status' => 1);
172.             $badge = '<span class="badge-status badge-
173.                 ok">Aktif</span>';
174.         }
175.         $this->M_database-
176.         >updateData('t_dosen', array('dosen_id' => $dosen_id), $data
177.     );
178.
179.         function resetPassword()
180.     {
181.         $dosen_id = $this->input->post('dosen_id');
182.         $dosen_nama = $this->input->post('dosen_nama');
183.         $password = password_hash('12345', PASSWORD_DEFAULT);
184.
185.         $data = array(
186.             'password' => $password
187.         );
188.
189.         $this->session-
190.         >set_flashdata('edit', "Reset password dosen <b>$dosen_nama<
191.             /b> berhasil");

```

```

190.         $this->M_database-
    >updateData('t_dosen', array('dosen_id' => $dosen_id), $data
    );
191.
192.         redirect('admin/dosen', 'refresh');
193.     }
194.
195. }

```

### Jabatan.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Jabatan extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $data['jabatan'] = $this->M_database-
    >getAllData('t_jabatan', null, null, 'jabatan_id ASC');
16.        $data['halaman_title'] = "Master Jabatan";
17.        $this->template->load('admin/template', 'admin/v-
    jabatan', $data);
18.    }
19.
20.    function tambahJabatan()
21.    {
22.        $jabatan_nama = ucwords($this->input-
    >post('jabatan_nama'));
23.        $honor_pertemuan = $this->input-
    >post('honor_pertemuan');
24.        $honor_perbulan = $this->input-
    >post('honor_perbulan');
25.
26.        $data = array(
27.            'jabatan_nama' => $jabatan_nama,
28.            'honor_pertemuan' => $honor_pertemuan,
29.            'honor_perbulan' => $honor_perbulan
30.        );
31.
32.        $this->session-
    >set_flashdata('tambah', "Jabatan <b>$jabatan_nama</b> berha
    sil disimpan");
33.        $this->M_database->insertData('t_jabatan', $data);
34.
35.        redirect('admin/jabatan', 'refresh');

```

```

36.      }
37.
38.      function editjabatan()
39.      {
40.          $jabatan_id = $this->input->post('jabatan_id');
41.          $jabatan_nama = ucwords($this->input-
>post('jabatan_nama'));
42.          $honor_pertemuan = $this->input-
>post('honor_pertemuan');
43.          $honor_perbulan = $this->input-
>post('honor_perbulan');
44.
45.          $data = array(
46.              'jabatan_nama' => $jabatan_nama,
47.              'honor_pertemuan' => $honor_pertemuan,
48.              'honor_perbulan' => $honor_perbulan
49.          );
50.
51.          $this->session-
>set_flashdata('edit', "Data jabatan <b>$jabatan_nama</b> berhasil diedit");
52.          $this->M_database-
>updateData('t_jabatan', array('jabatan_id' => $jabatan_id),
$data);
53.
54.          redirect('admin/jabatan', 'refresh');
55.      }
56.
57.      function hapusJabatan()
58.      {
59.          $jabatan_id = $this->input->post('jabatan_id');
60.          $jabatan_nama = ucwords($this->input-
>post('jabatan_nama'));
61.
62.          $this->session-
>set_flashdata('hapus', "Data jabatan <b>$jabatan_nama</b> berhasil dihapus");
63.          $this->M_database-
>deleteData('t_jabatan', array('jabatan_id' => $jabatan_id));
;
64.
65.          redirect('admin/jabatan', 'refresh');
66.      }
67.
68.  }

```

## Jam.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');

```

```

3.
4. class Jam extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $data['jam'] = $this->M_database-
>getAllData('t_jam', null, null, 'jam ASC');
16.        $data['halaman_title'] = "Master Jam";
17.        $this->template->load('admin/template', 'admin/v-
jam', $data);
18.    }
19.
20.    function tambahJam()
21.    {
22.        $jam_mulai = $this->input->post('jam_mulai');
23.        $jam_selesai = $this->input->post('jam_selesai');
24.
25.        $data = array(
26.            'jam' => $jam_mulai.' - '.$jam_selesai
27.        );
28.
29.        $this->session-
>set_flashdata('tambah', "Data jam berhasil ditambahkan");
30.        $this->M_database->insertData('t_jam', $data);
31.
32.        redirect('admin/jam', 'refresh');
33.    }
34.
35.    function editJam()
36.    {
37.        $jam_id = $this->input->post('jam_id');
38.        $jam_nama = $this->input->post('jam_nama');
39.        $jam_mulai = $this->input->post('jam_mulai');
40.        $jam_selesai = $this->input->post('jam_selesai');
41.
42.        $data = array(
43.            'jam' => $jam_mulai.' - '.$jam_selesai
44.        );
45.
46.        $this->session-
>set_flashdata('edit', "Data <b>$jam_nama</b> berhasil diedi-
t");
47.        $this->M_database-
>updateData('t_jam', array('jam_id' => $jam_id), $data);
48.
49.        redirect('admin/jam', 'refresh');
50.    }
51.
52.    function hapusJam()

```

```

53.      {
54.          $jam_id = $this->input->post('jam_id');
55.          $jam_lengkap = $this->input->post('jam_lengkap');
56.
57.          $this->session-
>set_flashdata('hapus', "Data jam <b>$jam_lengkap</b> berhasil dihapus");
58.          $this->M_database-
>deleteData('t_jam', array('jam_id' => $jam_id));
59.
60.          redirect('admin/jam', 'refresh');
61.      }
62.
63.  }

```

### Kehadiran.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Kehadiran extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $asisten_id = $this->input->get('asisten_id');
16.        $data['asistenkosong'] = $this->input-
>get('asisten_id');
17.
18.        $data['jadwal'] = $this->db-
>where('asisten_1', $asisten_id)-
>or_where('asisten_2', $asisten_id)->get('t_jadwal');
19.        $data['asisten'] = $this->M_database-
>getWhere('t_asisten', null, null, array('status' => 1), 'as
isten_nama ASC');
20.        $data['halaman_title'] = "Kehadiran Asisten";
21.        $data['visibility'] = '';
22.
23.        if ($asisten_id == null) {
24.            $data['visibility'] = 'hide';
25.        } else {
26.            $data['visibility'] = '';
27.        }
28.
29.        $this->template->load('admin/template', 'admin/v-
kehadiran', $data);
30.

```

```

31.     }
32.
33. }
```

## Kelas.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Kelas extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model ('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $jumlah_data = $this->M_database-
>jumlah_data('t_kelas');
16.        $offset = $this->uri->segment(4);
17.
18.        $this->load->library('pagination');
19.        $config['base_url'] = base_url().'admin/kelas/inde
x/';
20.        $config['total_rows'] = $jumlah_data;
21.        $config['per_page'] = 50;
22.        $config['first_link'] = '...';
23.        $config['last_link'] = '...';
24.        $config['next_link'] = 'NEXT';
25.        $config['prev_link'] = 'PREV';
26.        $config['full_tag_open'] = '<div class="custom-
pagination">';
27.        $config['full_tag_close'] = '</div>';
28.        $config['num_tag_open'] = '<button class="btn
transparent z-depth-0" style="padding: 0">';
29.        $config['num_tag_close'] = '</button>';
30.        $config['cur_tag_open'] = '<button class="btn
indigo" style="padding: 0 24px">';
31.        $config['cur_tag_close'] = '</button>';
32.        $config['next_tag_open'] = '<button class="btn
transparent z-depth-0 grey-text" style="padding: 0">';
33.        $config['next_tagl_close'] = '</button>';
34.        $config['prev_tag_open'] = '<button class="btn
transparent z-depth-0 grey-text" style="padding: 0">';
35.        $config['prev_tagl_close'] = '</button>';
36.        $config['first_tagl_open'] = '<button class="btn
transparent z-depth-0" style="padding: 0">';
37.        $config['first_tagl_close'] = '</button>';
38.        $config['last_tagl_open'] = '<button class="btn
transparent z-depth-0" style="padding: 0">';
```

```

39.         $config['last_tagl_close'] = '</button>';
40.         $this->pagination->initialize($config);
41.         $data['pagination'] = $this->pagination-
    >create_links();
42.
43.         $data['kelas'] = $this->M_database-
    >getAllData('t_kelas', null, null, 'kelas_id DESC');
44.         $data['halaman_title'] = "Master Kelas";
45.         $this->template->load('admin/template', 'admin/v-
    kelas', $data);
46.     }
47.
48.     function tambahKelas()
49.     {
50.         $kelas_kode = $this->input->post('kelas_kode');
51.         $kelas_nama = $this->input->post('kelas_nama');
52.
53.         $data = array(
54.             'kelas_kode' => $kelas_kode,
55.             'kelas_nama' => $kelas_nama,
56.             'kelas_status' => 1
57.         );
58.
59.         $this->session-
    >set_flashdata('tambah', "Data kelas <b>$kelas_nama</b> berh
    asil disimpan");
60.         $this->M_database->insertData('t_kelas', $data);
61.
62.         redirect('admin/kelas', 'refresh');
63.     }
64.
65.     function editKelas()
66.     {
67.         $kelas_id = $this->input->post('kelas_id');
68.         $kelas_kode = $this->input->post('kelas_kode');
69.         $kelas_nama = $this->input->post('kelas_nama');
70.
71.         $data = array(
72.             'kelas_kode' => $kelas_kode,
73.             'kelas_nama' => $kelas_nama
74.         );
75.
76.         $this->session-
    >set_flashdata('edit', "Data kelas <b>$kelas_nama</b> berhas
    il diedit");
77.         $this->M_database-
    >updateData('t_kelas', array('kelas_id' => $kelas_id), $data
    );
78.
79.         redirect('admin/kelas', 'refresh');
80.     }
81.
82.     function hapusKelas()
83.     {
84.         $kelas_id = $this->input->post('kelas_id');
85.         $kelas_nama = $this->input->post('kelas_nama');

```

```

86.
87.         $this->session-
>set_flashdata('hapus', "Data kelas <b>$kelas_nama</b> berhasil dihapus");
88.         $this->M_database-
>deleteData('t_kelas', array('kelas_id' => $kelas_id));
89.
90.         redirect('admin/kelas', 'refresh');
91.     }
92.
93.     function statusKelas()
94.     {
95.         $kelas_id = $this->input->post('kelas_id');
96.         $status = $this->input->post('status');
97.         $kelas = $this->db->select('kelas_kode')-
>where('kelas_id', $kelas_id)->get('t_kelas')->row();
98.         $kelas_kode = $kelas->kelas_kode;
99.
100.        if ($status == 'Aktif') {
101.            $data = array('kelas_status' => 0);
102.            $data_mhs = array('status_mhs' => 0);
103.            $badge = '<span class="badge-status badge-not">Tidak aktif</span>';
104.        } else {
105.            $data = array('kelas_status' => 1);
106.            $data_mhs = array('status_mhs' => 1);
107.            $badge = '<span class="badge-status badge-ok">Aktif</span>';
108.        }
109.
110.        $this->M_database-
>updateData('t_kelas', array('kelas_id' => $kelas_id), $data);
111.        $this->M_database-
>updateData('t_mahasiswa', array('kelas_kode' => $kelas_kode),
$data_mhs);
112.
113.        echo $badge;
114.    }
115.
116. }

```

### Mahasiswa.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Mahasiswa extends CI_Controller {
5.

```

```

6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $data['mahasiswa'] = $this->M_database-
>getAllData('t_mahasiswa', 't_kelas', 't_mahasiswa.kelas_kod
e = t_kelas.kelas_kode', 'npm DESC');
16.        $data['kelas'] = $this->M_database-
>getWhere('t_kelas', null, null, array('kelas_status' => 1),
'kelas_id DESC');
17.        $data['halaman_title'] = "Master Mahasiswa";
18.        $this->template->load('admin/template', 'admin/v-
mahasiswa', $data);
19.    }
20.
21.    function tambahMahasiswa()
22.    {
23.        $npm = $this->input->post('npm');
24.        $password = password_hash('12345', PASSWORD_DEFAULT);
25.        $nama = ucwords($this->input->post('nama'));
26.        $kelas_kode = $this->input->post('kelas_kode');
27.
28.        $data = array(
29.            'npm' => $npm,
30.            'password' => $password,
31.            'nama' => $nama,
32.            'kelas_kode' => $kelas_kode,
33.            'status_bayar' => 0,
34.            'status_mhs' => 1
35.        );
36.
37.        $cek_npm = $this->db->where('npm', $npm)-
>get('t_mahasiswa')->row();
38.
39.        if ($cek_npm) {
40.            echo "gagal|Mahasiswa dengan NPM : <b>$npm</b>
sudah ada, data gagal disimpan";
41.        } else {
42.
43.            $this->M_database-
>insertData('t_mahasiswa', $data);
44.            echo "sukses|Data mahasiswa <b>$npm</b> berhas
il disimpan";
45.        }
46.    }
47.
48.    function editMahasiswa()
49.    {
50.        $mhs_id = $this->input->post('mhs_id');
51.        $npm = $this->input->post('npm');

```

```

52.         $nama = ucwords($this->input->post('nama'));
53.         $kelas_kode = $this->input-
>post('kelas_kode');
54.
55.         $data = array(
56.                         'npm' => $npm,
57.                         'nama' => $nama,
58.                         'kelas_kode' => $kelas_kode
59.                     );
60.
61.         $this->session-
>set_flashdata('edit', "Data mahasiswa <b>$nama</b> berhasil
diedit");
62.         $this->M_database-
>updateData('t_mahasiswa', array('mhs_id' => $mhs_id), $data
);
63.
64.         redirect('admin/mahasiswa', 'refresh');
65.     }
66.
67.     function hapusMahasiswa()
68.     {
69.         $mhs_id = $this->input->post('mhs_id');
70.         $nama = $this->input->post('nama');
71.
72.         $delete = $this->M_database-
>deleteData('t_mahasiswa', array('mhs_id' => $mhs_id));
73.         $this->session-
>set_flashdata('hapus', "Data mahasiswa <b>$nama</b> berhasil
1 dihapus");
74.         redirect('admin/mahasiswa', 'refresh');
75.     }
76.
77.     function resetPassword()
78.     {
79.         $mhs_id = $this->input->post('mhs_id');
80.         $nama = $this->input->post('nama');
81.         $password = password_hash('12345', PASSWORD_DEFAULT);
82.
83.         $data = array(
84.                         'password' => $password
85.                     );
86.
87.         $this->M_database-
>updateData('t_mahasiswa', array('mhs_id' => $mhs_id), $data
);
88.         $this->session-
>set_flashdata('edit', "Reset password mahasiswa <b>$nama</b>
> berhasil");
89.         redirect('admin/mahasiswa', 'refresh');
90.     }
91.
92.     function statusMhs()
93.     {
94.         $npm = $this->input->post('npm');

```



```

139.                               data-nama="'. $value-
   >nama.'"                                     data-kelas_nama="'. $value-
140.                               data-kelas_nama.'"                                     data-kelas_kode="'. $value-
141.                               data-kelas_kode.'">                                <i class="material-icons-
   >kelas_nama.'"                                     outlined grey-text">lock_open</i>
142.                                         </button>
143.                                         <button class="waves-
144.                                         effect mybtn transparent modal-trigger btn-edit-mhs" data-
   target="modalEdit"                                     data-mhs_id="'. $value-
145.                                     >mhs_id.'"
146.                                     data-npm="'. $value->npm.'"
147.                                     data-nama="'. $value-
   >nama.'"                                     data-kelas_nama="'. $value-
148.                                     data-kelas_nama.'"                                     data-kelas_kode="'. $value-
149.                                     data-kelas_kode.'">                                <i class="material-icons-
   >kelas_kode.'"                                     outlined amber-text">edit</i>
150.                                         </button>
151.                                         <button title="'. $title.'" cla-
   ss="waves-effect mybtn transparent modal-trigger btn-hapus-
   mhs" data-target="'. $modal.'"                                     data-mhs_id="'. $value-
153.                                     >mhs_id.'"
154.                                     data-npm="'. $value->npm.'"
155.                                     data-nama="'. $value-
   >nama.'"><i class="material-icons red-text text-lighten-
   1">delete</i>
156.                                         </button></div>';
157.             $data[] = $sub_array;
158.         }
159.         $output = array(
160.             "draw"          => intval($_POST["draw"]),
161.             "recordsTotal"  => $this->M_database-
   >get_all_data(),
162.             "recordsFiltered" => $this->M_database-
   >get_filtered_data(),
163.             "data"           => $data
164.         );
165.         echo json_encode($output);
166.     }
167.
168. }

```

### Matkum.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');

```

```

3.
4. class Matkum extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $data['matkum'] = $this->M_database-
>getAllData('t_mata_praktikum', null, null, 'matkum_status D
ESC', 'matkum ASC');
16.        $data['halaman_title'] = "Master Mata Praktikum";
17.        $this->template->load('admin/template', 'admin/v-
matkum', $data);
18.    }
19.
20.    function tambahMatkum()
21.    {
22.        $matkum = ucwords($this->input->post('matkum'));
23.
24.        $data = array(
25.            'matkum' => $matkum,
26.            'matkum_status' => 1
27.        );
28.
29.        $this->session-
>set_flashdata('tambah', "Mata praktikum <b>$matkum</b> berh
asil disimpan");
30.        $this->M_database-
>insertData('t_mata_praktikum', $data);
31.
32.        redirect('admin/matkum', 'refresh');
33.    }
34.
35.    function editMatkum()
36.    {
37.        $matkum_id = $this->input->post('matkum_id');
38.        $matkum = $this->input->post('matkum');
39.
40.        $data = array(
41.            'matkum' => $matkum
42.        );
43.
44.        $this->session-
>set_flashdata('edit', "Mata praktikum <b>$matkum</b> berhas
il diedit");
45.        $this->M_database-
>updateData('t_mata_praktikum', array('matkum_id' => $matkum_
_id), $data);
46.
47.        redirect('admin/matkum', 'refresh');
48.    }

```

```

49.
50.     function hapusMatkum()
51.     {
52.         $matkum_id = $this->input->post('matkum_id');
53.         $matkum = $this->input->post('matkum');
54.
55.         $this->session-
56.             >set_flashdata('hapus', "Mata praktikum <b>$matkum</b> berhasil dihapus");
57.         $this->M_database-
58.             >deleteData('t_mata_praktikum', array('matkum_id' => $matkum_id));
59.     }
60.
61.     function statusMatkum()
62.     {
63.         $matkum_id = $this->input->post('matkum_id');
64.         $status = $this->input->post('status');
65.
66.         if ($status == 'Aktif') {
67.             $data = array('matkum_status' => 0);
68.             $badge = '<span class="badge-status badge-not">Tidak aktif</span>';
69.         } else {
70.             $data = array('matkum_status' => 1);
71.             $badge = '<span class="badge-status badge-ok">Aktif</span>';
72.         }
73.
74.         $this->M_database-
75.             >updateData('t_mata_praktikum', array('matkum_id' => $matkum_id), $data);
76.         echo $badge;
77.     }
78.
79. }
```

### Pembayaran.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Pembayaran extends CI_Controller {
5.
6.     function __construct()
7.     {
```

```

8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $data['jumlah_mhs'] = $this->M_database-
>countWhere('t_mahasiswa', 'status_mhs => 1');
16.        $data['jumlah_lunas'] = $this->M_database-
>countWhere('t_mahasiswa', array('status_mhs' => 1, 'status_
bayar' => 1));
17.        $data['jumlah_belum'] = $this->M_database-
>countWhere('t_mahasiswa', array('status_mhs' => 1, 'status_
bayar' => 0));
18.        $data['mahasiswa'] = $this->M_database-
>getWhere('t_mahasiswa', 't_kelas', 't_mahasiswa.kelas_kode
= t_kelas.kelas_kode', array('status_mhs' => 1), 'npm DESC')
;
19.        $data['halaman_title'] = "Pembayaran Praktikum";
20.        $this->template->load('admin/template', 'admin/v-
pembayaran', $data);
21.    }
22.
23.    function statusBayar()
24.    {
25.        $npm = $this->input->post('npm');
26.        $status = $this->input->post('status');
27.
28.        if ($status == 'Lunas') {
29.            $data = array('status_bayar' => 1);
30.            $badge = '<span class="badge-status badge-
not">Belum</span>';
31.            $pembayaran = 'belum';
32.        } else {
33.            $data = array('status_bayar' => 0);
34.            $badge = '<span class="badge-status badge-
ok">Lunas</span>';
35.            $pembayaran = 'lunas';
36.        }
37.
38.        $this->M_database-
>updateData('t_mahasiswa', array('npm' => $npm), $data);
39.
40.        echo "$pembayaran|$badge";
41.    }
42.
43. }

```

## Pengumuman.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');

```

```

3.
4. use Minishlink\WebPush\WebPush;
5. use Minishlink\WebPush\Subscription;
6.
7. class Pengumuman extends CI_Controller {
8.
9.     function __construct()
10.    {
11.        parent::__construct();
12.        $this->load->model('M_database');
13.        cek_session();
14.    }
15.
16.    public function index()
17.    {
18.        $info_id = $this->db->select('info_id')-
>order_by('info_id DESC')->limit(1)->get('t_info')->row();
19.        $id = 0;
20.        if ($info_id == null) {
21.            $id = 1;
22.        } else {
23.            $id = $info_id->info_id + 1;
24.        }
25.        $data['kode_info'] = $id;
26.        $data['info'] = $this->M_database-
>getAllData('t_info', null, null, 'info_id DESC');
27.        $data['kelas'] = $this->M_database-
>getWhere('t_kelas', null, null, array('kelas_status' => 1,
'kelas_id DESC'));
28.        $data['foto_info'] = $this->M_database-
>getWhere('t_info_image', null, null, array('info_id' => $id
), null);
29.        $data['halaman_title'] = "Info Praktikum";
30.        $this->template->load('admin/template', 'admin/v-
pengumuman', $data);
31.    }
32.
33.    function tambahInfo() {
34.
35.        $info_caption = $this->input-
>post('info_caption');
36.        $date_created = date('Y-m-
d H:i:s');
37.
38.        $data = array(
39.            'info_caption' => $info_caption,
40.            'date_created' => $date_created
41.        );
42.
43.        $this->session-
>set_flashdata('tambah', 'Info Praktikum berhasil disimpan')
;
44.        $this->M_database->insertData('t_info', $data);
45.
46.        redirect('admin/pengumuman', 'refresh');
47.

```

```

48.      }
49.
50.      function editInfo() {
51.
52.          $info_id = $this->input->post('info_id');
53.          $info_caption = $this->input-
54.              >post('info_caption');
55.          $data = array(
56.              'info_caption' => $info_caption,
57.          );
58.
59.          $this->session-
60.              >set_flashdata('edit', 'Info Praktikum berhasil diedit');
61.          $this->M_database-
62.              >updateData('t_info', array('info_id' => $info_id) , $data);
63.
64.      }
65.
66.      function hapusInfo()
67.      {
68.          $info_id = $this->input->post('info_id');
69.
70.          $info_image = $this->db-
71.              >where('info_id', $info_id)->get('t_info_image');
72.              foreach ($info_image-
73.                  >result() as $key => $value) {
74.                      unlink('./assets/images/info/'.$value-
75.                          >info_image);
76.
77.          $this->M_database-
78.              >deleteData('t_info', array('info_id' => $info_id));
79.          $this->M_database-
80.              >deleteData('t_info_image', array('info_id' => $info_id));
81.          $this->session-
82.              >set_flashdata('hapus', 'Info Praktikum berhasil dihapus');
83.              redirect('admin/pengumuman', 'refresh');
84.      }
85.
86.      function kirimNotifikasi()
87.      {
88.          $judul = $this->input->post('judul');
89.          $url = 'http://localhost/clientsilabtif/info_prakt
90.              ikum';
91.          $token = $this->db->get('t_device_token')-
92.              >result();
93.          $notifications= [];
94.
95.          foreach ($token as $key => $value) {
96.              array_push($notifications,

```

```

92.          [
93.              'subscription' => Subscription::create
94.              (
95.                  [
96.                      "endpoint" => $value->endpoint,
97.                      "keys" => [
98.                          'p256dh' => $value->p256dh,
99.                          'auth' => $value->auth
100.                     ],
101.                ],
102.            );
103.        }
104.        $auth = array(
105.            'VAPID' => array(
106.                'subject' => '*****',
107.                'publicKey' => '*****',
108.                'privateKey' => '*****' // in the real
      world, this would be in a secret file
109.            ),
110.        );
111.
112.        $webPush = new WebPush($auth);
113.
114.        // send multiple notifications with payload
115.        foreach ($notifications as $notification) {
116.            $webPush->sendNotification(
117.                $notification['subscription'],
118.                $notification['payload'] // optional (defa
      ults null)
119.            );
120.        }
121.
122.        /**
123.         * Check sent results
124.         * @var MessageSentReport $report
125.         */
126.        foreach ($webPush->flush() as $report) {
127.            $endpoint = $report->getRequest()->getUri()->
      toString();
128.
129.            // if ($report->isSuccess()) {
130.                // echo "[v] Message sent successfully for su
      bscription {$endpoint}.";
131.            // } else {
132.                // echo "[x] Message failed to sent for subsc
      ription {$endpoint}: {$report->getReason()}";
133.            // }
134.        }
135.
136.        /**
137.         * send one notification and flush directly
138.         * @var \Generator<MessageSentReport> $sent
139.         */
140.        $sent = $webPush->sendNotification(

```

```

141.         $notifications[0]['subscription'],
142.         $notifications[0]['payload'], // optional (def
143.             aults null)
144.             true // optional (defaults false)
145.         );
146.         redirect('admin/pengumuman', 'refresh');
147.     }
148.
149.     private function _notification($token, $caption)
150.     {
151.
152.         $msg = array
153.         (
154.             'body' => $caption,
155.             'title' => 'Info Praktikum',
156.             'sound' => 1,
157.             'vibrate' => 1
158.         );
159.
160.         $fields = array
161.         (
162.             'to' => $token,
163.             'notification' => $msg
164.         );
165.
166.
167.         //kunci server
168.         $headers = array
169.         (
170.             'Authorization: key=' . ****,
171.             'Content-Type: application/json'
172.         );
173.
174.         $ch = curl_init();
175.         curl_setopt( $ch,CURLOPT_URL, 'https://fcm.googleapis.com/fcm/send' );
176.         curl_setopt( $ch,CURLOPT_POST, true );
177.         curl_setopt( $ch,CURLOPT_HTTPHEADER, $headers );
178.         curl_setopt( $ch,CURLOPT_RETURNTRANSFER, true );
179.         curl_setopt( $ch,CURLOPT_SSL_VERIFYPEER, false );
180.         curl_setopt( $ch,CURLOPT_POSTFIELDS, json_encode(
181.             $fields ) );
182.         $result = curl_exec($ch );
183.         curl_close( $ch );
184.         // echo $result;
185.     }
186.
187.     function fetch_image()
188.     {
189.         $info_id = $this->input->get('info_id');
190.         $image = $this->M_database-
191.             >getWhere('t_info_image', null, null, array('info_id' => $in
fo_id));
foreach ($image->result() as $key => $value) {

```

```

192.           echo '<div class="img-wrapper" id="img-
    wrapper-' . $value -> info_image_id . '">
193.                         <div class="preloader-
    wrapper small active hide" id="myloader-' . $value-
    > info_image_id . '" style="position: absolute; margin: auto; t
    op:0;bottom:0;left:0;right:0; z-index:10">
194.                             <div class="spinner-
    layer spinner-teal-only">
195.                               <div class="circle-
    clipper left">
196.                                 <div class="circle"></
    div>
197.                               <div class="circle"></
    div><div class="gap-
    patch">
198.                                 <div class="circle"></
    div>
199.                               <div class="circle-
    clipper right">
200.                                 <div class="circle"></
    div>
201.                               </div>
202.                         </div>
203.                       </div>

204.            info_image_id . '" class="responsive-
img img materialboxed">
205.           <div id="action-foto-' . $value-
    > info_image_id . '" style="position: absolute; bottom:0; width:
    100%">
206.             <span class="material-
    icons red-text text-lighten-1 left delete-foto-
    info" id="hapus-' . $value -> info_image_id . '" data-
    gambar="'. $value -> info_image . '" data-id="'. $value-
    > info_image_id . '" style="cursor: pointer;">close</span>
207.             <span class="material-icons-
    outlined amber-text text-lighten-1 right edit-foto-
    info" data-gambar="'. $value -> info_image . '" data-
    id="'. $value -> info_image_id . '" id="edit-' . $value-
    > info_image_id . '" style="cursor: pointer;">edit</span>
208.           </div>
209.         </div>';
210.       }
211.     }
212.   }
213.   function uploadFoto()
214.   {
215.     $image_id = $this->input-
    >post('image_id');
216.     $image_edit = $this->input-
    >post('image_edit');
217.     $info_id = $this->input-
    >post('info_id');
218.     $config['upload_path'] = './assets/images/info/';
219.

```

```

220.      $config['allowed_types'] = 'jpg|png|jpeg|webp';
221.      $config['file_name'] = 'info-img-' . time();
222.      $config['max_size'] = 2000;
223.      $config['max_width'] = 1080;
224.      $config['max_height'] = 1080;
225.      $config['overwrite'] = true;
226.
227.      $this->load->library('upload', $config);
228.
229.      if ($this->upload->do_upload('foto')) {
230.
231.          $foto = $this->upload->data();
232.          $data_foto = $foto['file_name'];
233.          $this->compressImage($data_foto);
234.
235.          if ($image_id != '-') {
236.              unlink('./assets/images/info/' . $image_edit);
237.              $this->M_database-
238.                  >updateData('t_info_image', array('info_image_id' => $image_id),
239.                             array('info_image' => $data_foto));
240.                  echo "edit|$image_id|".base_url()."assets/
241.                      images/info/$data_foto|$data_foto";
242.          } else {
243.              $this->M_database-
244.                  >insertData('t_info_image', array('info_image' => $data_foto,
245.                               'info_id' => $info_id));
246.                  $id = $this->db->insert_id();
247.                  echo 'tambah|<div class="img-
248.                      wrapper" id="img-wrapper-'.$id.'">
249.                          <div class="preloader-
250.                              wrapper small active hide" id="myloader-
251.                                '.$id.'" style="position: absolute; margin: auto; top:0;bott
252.                                om:0;left:0;right:0; z-index:10">
253.                                  <div class="spinner-
254.                                      layer spinner-teal-only">
255.                                      <div class="circle-
256.                                          clipper left">
257.                                          <div class="circle
258.                                              patch">
259.                                              </div><div class="gap-
260.                                              <div class="circle
261.                                              <div class="circle-
262.                                                  le-clipper right">
263.                                                  <div class="circle
264.                                              </div><div class="circ
265.                                              <div class="circle
266.                                              </div>
267.                                              </div>
268.                                              </div>
269.                                              
```

```

255.                               <div id="action-foto-
  '. $id.'" style="position:absolute; bottom:0; width:100%">
256.                               <span class="material-
  icons red-text text-lighten-1 left delete-foto-info" data-
  gambar="'. $data_foto.'" data-id="'. $id.'" id="hapus-
  '$. $id.'" style="cursor: pointer;">close</span>
257.                               <span class="material-
  icons-outlined amber-text text-lighten-1 right edit-foto-
  info" data-gambar="'. $data_foto.'" data-
  id="'. $id.'" id="edit-
  '$. $id.'" style="cursor: pointer;">edit</span>
258.                           </div>
259.                         </div>';
260.                     }
261.                 } else {
262.                     echo "gagal|$image_id";
263.                 }
264.
265.             }
266.
267.         function hapusFoto()
268.         {
269.             $id = $this->input->post('id');
270.             $gambar = $this->input->post('gambar');
271.             $this->M_database-
  >deleteData('t_info_image', array('info_image_id' => $id));
272.             if ($this->db->affected_rows() > 0) {
273.                 unlink('./assets/images/info/'. $gambar);
274.                 echo "berhasil";
275.             } else {
276.                 echo "gagal";
277.             }
278.         }
279.
280.         function compressImage($path)
281.         {
282.             $config['image_library']='gd2';
283.             $config['source_image']= './assets/images/info/'. $p
  ath;
284.             $config['create_thumb']= FALSE;
285.             $config['maintain_ratio']= FALSE;
286.             $config['quality']= '50%';
287.             $config['new_image']= './assets/images/info/'. $pat
  h;
288.             $this->load->library('image_lib', $config);
289.             $this->image_lib->resize();
290.         }
291.
292.     }

```

### Penilaian.php

1. <?php
----------

```

2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Penilaian extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $jadwal_kode = $this->input->get('jadwal_kode');
16.        $data['jadwalkosong'] = $this->input-
>get('jadwal_kode');
17.
18.        $data['mahasiswa'] = $this->M_database-
>getDetailTugas(array('jadwal_kode' => $jadwal_kode));
19.        $data['jadwal'] = $this->M_database-
>getWhere('t_jadwal', null, null, array('status' => 1), 'jad
wal_kode ASC');
20.        $data['halaman_title'] = "Penilaian Praktikum";
21.        $data['visibility'] = '';
22.
23.        if ($jadwal_kode == null) {
24.            $data['visibility'] = 'hide';
25.        } else {
26.            $data['visibility'] = '';
27.        }
28.
29.        $this->template->load('admin/template', 'admin/v-
penilaian', $data);
30.
31.    }
32.
33.    function editPersentase() {
34.        $jadwal_kode = $this->input->post('jadwal_kode');
35.        $p_kehadiran = $this->input->post('p_kehadiran');
36.        $p_tugas = $this->input->post('p_tugas');
37.        $p_ujian = $this->input->post('p_ujian');
38.
39.        $data = array(
40.            'p_kehadiran' => $p_kehadiran,
41.            'p_tugas' => $p_tugas,
42.            'p_ujian' => $p_ujian
43.        );
44.
45.        $this->M_database-
>updateData('t_jadwal', array('jadwal_kode' => $jadwal_kode)
, $data);
46.
47.        $this->session-
>set_flashdata('edit', "Persentase penilaian <b>$jadwal_kode
</b> berhasil diedit");

```

```

48.         redirect("admin/penilaian?jadwal_kode=$jadwal_kode
  &id_praktikum=$jadwal_kode", 'refresh');
49.
50.     }
51.
52.     function penilaian() {
53.         $npm = $this->input->post('npm');
54.         $jadwal_kode = $this->input->post('jadwal_kode');
55.         $nilai_tugas = $this->input-
>post('nilai_tugas');
56.         $nilai_ujian = $this->input->post('nilai_ujian');
57.         $total = $nilai_tugas + $nilai_ujian;
58.         $this->M_database-
>updateData('registrasi_praktikum', array('jadwal_kode' => $jadwal_kode, 'npm' => $npm), array('nilai_tugas' => $nilai_tugas, 'nilai_ujian' => $nilai_ujian));
59.         echo "$nilai_tugas|$nilai_ujian|$total";
60.     }
61.
62. }

```

### Penjadwalan.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Penjadwalan extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $periode_id = $this->input->get('periode_id');
16.        $data['periode'] = $this->uri->segment(4);
17.        $data['matkum'] = $this->M_database-
>getWhere('t_mata_praktikum', null, null, array('matkum_status' => 1), 'matkum ASC');
18.        $data['kelas'] = $this->M_database-
>getWhere('t_kelas', null, null, array('kelas_status' => 1), 'kelas_id DESC');
19.        $data['asisten'] = $this->M_database-
>getWhere('t_asisten', null, null, array('status' => 1), 'asisten_nama ASC');
20.        $data['dosen'] = $this->M_database-
>getWhere('t_dosen', null, null, array('dosen_status' => 1), 'dosen_nama ASC');
21.        $data['jam'] = $this->M_database-
>getAllData('t_jam');

```

```

22.      $data['ruangan'] = $this->M_database-
    >getAllData('t_ruangan');
23.      $data['halaman_title'] = "Penjadwalan";
24.      $data['jadwal'] = $this->M_database-
    >getJadwal(array('periode_id' => $periode_id), 'jadwal_id DE
    SC');
25.      $this->template->load('admin/template', 'admin/v-
    penjadwalan', $data);
26.  }
27.
28.  function view_tambah()
29.  {
30.      $data['halaman_title'] = "Penjadwalan";
31.      $data['periode'] = $this->uri->segment(4);
32.      $data['matkum'] = $this->M_database-
    >getAllData('t_mata_praktikum', null, null, 'matkum ASC');
33.      $data['kelas'] = $this->M_database-
    >getAllData('t_kelas');
34.      $data['asisten'] = $this->M_database-
    >getWhere('t_asisten', null, null, array('status' => 1), 'as
    isten_nama ASC');
35.      $data['jam'] = $this->M_database-
    >getAllData('t_jam');
36.      $data['ruangan'] = $this->M_database-
    >getAllData('t_ruangan');
37.      $this->template->load('admin/template', 'admin/v-
    penjadwalan-tambah', $data);
38.  }
39.
40.  function view_edit()
41.  {
42.      $jadwal_id = $this->uri->segment(4);
43.      $jadwal = $this->M_database-
    >getJadwal(array('jadwal_id' => $jadwal_id))->result();
44.      $data['matkum'] = $this->M_database-
    >getAllData('t_mata_praktikum', null, null, 'matkum ASC');
45.      $data['kelas'] = $this->M_database-
    >getAllData('t_kelas');
46.      $data['asisten'] = $this->M_database-
    >getWhere('t_asisten', null, null, array('status' => 1), 'as
    isten_nama ASC');
47.      $data['jam'] = $this->M_database-
    >getAllData('t_jam');
48.      $data['ruangan'] = $this->M_database-
    >getAllData('t_ruangan');
49.
50.      foreach ($jadwal as $key => $value) {
51.
52.          $data['jadwal_id'] = $value->jadwal_id;
53.          $data['jadwal_kode'] = $value->jadwal_kode;
54.          $data['periode_id'] = $value->periode_id;
55.          $data['matkum_id'] = $value->matkum_id;
56.          $data['matkum_nama'] = $value->matkum;
57.          $data['kelas_id'] = $value->kelas_id;
58.          $data['kelas_nama'] = $value->kelas_nama;
59.          $data['asisten_1'] = $value->asisten_1;

```

```

60.          $data['asisten_2'] = $value->asisten_2;
61.          $data['jadwal_hari'] = $value->jadwal_hari;
62.          $data['jadwal_jam'] = $value->jadwal_jam;
63.          $data['ruangan_id'] = $value->ruangan_id;
64.          $data['ruangan_nama'] = $value->ruangan_nama;
65.      }
66.
67.      $this->template->load('admin/template', 'admin/v-
  penjadwalan-edit', $data);
68.  }
69.
70.  function tambahJadwal()
71.  {
72.      $periode_id = $this->input->post('periode_id');
73.      $jadwal_kode = $this->input->post('jadwal_kode');
74.      $matkum_id = $this->input->post('matkum_id');
75.      $periode_id = $this->input->post('periode_id');
76.      $kelas_kode = $this->input->post('kelas_kode');
77.      $dosen_id = $this->input->post('dosen_id');
78.      $asisten_1 = $this->input->post('asisten_1');
79.      $asisten_2 = $this->input->post('asisten_2');
80.      $jadwal_hari = $this->input->post('jadwal_hari');
81.      $jadwal_jam = $this->input->post('jadwal_jam');
82.      $ruangan_id = $this->input->post('ruangan_id');
83.
84.      $data = array(
85.          'jadwal_kode' => $jadwal_kode,
86.          'matkum_id' => $matkum_id,
87.          'kelas_kode' => $kelas_kode,
88.          'dosen_id' => $dosen_id,
89.          'asisten_1' => $asisten_1,
90.          'asisten_2' => $asisten_2,
91.          'periode_id' => $periode_id,
92.          'jadwal_hari' => $jadwal_hari,
93.          'jadwal_jam' => $jadwal_jam,
94.          'ruangan_id' => $ruangan_id,
95.          'periode_id' => $periode_id,
96.          'p_kehadiran' => 30,
97.          'p_tugas' => 30,
98.          'p_ujian' => 40,
99.          'status' => 0
100.     );
101.
102.     $cek_jadwal = $this->db-
  >where('jadwal_kode', $jadwal_kode)->get('t_jadwal')-
  >row();
103.     if ($cek_jadwal) {
104.         echo "gagal|Jadwal praktikum <b>$jadwal_kode</
  b> sudah ada, data gagal disimpan";
105.     } else {
106.         $this->M_database-
  >insertData('t_jadwal', $data);
107.         $jadwal_akhir = $this->M_database-
  >getJadwal(array('periode_id' => $periode_id), 'jadwal_id DE
  SC', 1)->row();

```

```

108.           $asisten_1 = $this->db-
>where('asisten_id', $jadwal_akhir->asisten_1)-
>get('t_asisten')->row();
109.           $asisten_2 = $this->db-
>where('asisten_id', $jadwal_akhir->asisten_2)-
>get('t_asisten')->row();
110.           $jadwal = $this->M_database-
>cekdata('registrasi_praktikum', 'jadwal_kode', $jadwal_akhir-
>jadwal_kode);
111.           $title = 'hapus';
112.           $modal = 'modalHapus';
113.
114.           if ($jadwal != 0) {
115.               $title = 'tidak bisa hapus';
116.               $modal = 'modalAlert';
117.           }
118.           $status = '';
119.           if ($jadwal_akhir->status == 0) {
120.               $status = '<span class="badge-
status badge-not">Tidak aktif</span>';
121.           } else {
122.               $status = '<span class="badge-
status badge-ok">Aktif</span>';
123.           }
124.           $table = '<tr class="green lighten-5">
<td></td>
125.
126.           <td>'.$jadwal_akhir-
>jadwal_kode.'</td>
127.           <td>'.$jadwal_akhir->matkum.'</td>
128.           <td>'.$jadwal_akhir-
>kelas_nama.'</td>
129.           <td>'.$asisten_1-
>asisten_nama.'</td>
130.           <td>'.$asisten_2-
>asisten_nama.'</td>
131.           <td>'.$jadwal_akhir-
>jadwal_hari.'</td>
132.           <td class="center">'.$jadwal_akhir-
->jadwal_jam.'</td>
133.           <td>'.$jadwal_akhir-
>ruangan_nama.'</td>
134.           <td class="status-jadwal td-
data_center" id="'.$jadwal_akhir-
>jadwal_id.'">'.$status.'</td>
135.
136.           <td class="center">
<button class="waves-
effect mybtn transparent modal-trigger btn-edit-
jadwal" data-target="modalEdit"
137.               data-
jadwal_id="'.$jadwal_akhir->jadwal_id.'"
138.               data-
jadwal_kode="'.$jadwal_akhir->jadwal_kode.'"
139.               data-
matkum_id="'.$jadwal_akhir->matkum_id.'"

```

```

140.          data-
141.          dosen_id="'. $jadwal_akhir->dosen_id.'"
142.          data-
143.          kelas_kode="'. $jadwal_akhir->kelas_kode.'"
144.          data-
145.          asisten_1="'. $jadwal_akhir->asisten_1.'"
146.          data-
147.          asisten_2="'. $jadwal_akhir->asisten_2.'"
148.          data-
149.          jadwal_hari="'. $jadwal_akhir->jadwal_hari.'"
150.          data-
151.          jadwal_jam="'. $jadwal_akhir->jadwal_jam.'"
152.          data-
153.          ruangan_id="'. $jadwal_akhir->ruangan_id.'"
154.          ><i class="material-
155.          icons amber-text">edit</i></button>
156.          <button title="'. $title.'" cla-
157.          ss="waves-effect mybtn transparent modal-trigger btn-hapus-
158.          jadwal" data-target="'. $modal.'"
159.          data-
160.          jadwal_id="'. $jadwal_akhir->jadwal_id.'"
161.          data-
162.          periode_id="'. $jadwal_akhir->periode_id.'"><i class="material-icons red-text text-
163.          lighten-1 ">delete</i>
164.          </button>
165.          </td>
166.          </tr>';
167.          echo "sukses|Jadwal praktikum <b>$jadwal_kode<
168.          /b> berhasil disimpan|$table";
169.      }
170.  }
171.
172.  function editJadwal()
173.  {
174.      $kode_asal = $this->input->post('kode_asal');
175.      $periode_id = $this->input->post('periode_id');
176.      $jadwal_id = $this->input->post('jadwal_id');
177.      $jadwal_kode = $this->input->post('jadwal_kode');
178.      $matkum_id = $this->input->post('matkum_id');
179.      $dosen_id = $this->input->post('dosen_id');
180.      $kelas_kode = $this->input->post('kelas_kode');
181.      $asisten_1 = $this->input->post('asisten_1');
182.      $asisten_2 = $this->input->post('asisten_2');
183.      $jadwal_hari = $this->input->post('jadwal_hari');
184.      $jadwal_jam = $this->input->post('jadwal_jam');
185.      $ruangan_id = $this->input->post('ruangan_id');
186.
187.      $cek_jadwal = $this->db-
188.          >where('jadwal_kode', $jadwal_kode)->get('t_jadwal')->row();
189.
190.      $data = array(
191.          'jadwal_kode' => $jadwal_kode,
192.          'matkum_id' => $matkum_id,
193.          'kelas_kode' => $kelas_kode,
194.          'asisten_1' => $asisten_1,
195.          'periode_id' => $periode_id,
196.          'dosen_id' => $dosen_id,
197.          'jadwal_id' => $jadwal_id,
198.          'jadwal_hari' => $jadwal_hari,
199.          'jadwal_jam' => $jadwal_jam,
200.          'ruangan_id' => $ruangan_id);
201.
202.      $this->model('jadwal_model')
203.          ->update($data);
204.
205.      $this->session->setFlashdata('success', 'Jadwal berhasil diubah');
206.
207.      redirect('jadwal');
208.
```

```

180.           'asisten_2' => $asisten_2,
181.           'jadwal_hari' => $jadwal_hari,
182.           'jadwal_jam' => $jadwal_jam,
183.           'dosen_id' => $dosen_id,
184.           'ruangan_id' => $ruangan_id

185.       );
186.       if ($kode_asal == $jadwal_kode) {
187.           $this->M_database-
188.           >updateData('t_jadwal', array('jadwal_id' => $jadwal_id), $d
ata);
189.           $this->session-
190.           >set_flashdata('edit', 'Jadwal <b>' . $jadwal_kode . '</b> berha
sil diedit');
191.           echo "sukses|admin/penjadwalan?periode_id=$per
iode_id";
192.       } else if($cek_jadwal) {
193.           echo "gagal|Jadwal praktikum <b>$jadwal_kode</
b> sudah ada, data gagal disimpan";
194.       } else {
195.           $this->M_database-
196.           >updateData('t_jadwal', array('jadwal_id' => $jadwal_id), $d
ata);
197.           $this->session-
198.           >set_flashdata('edit', 'Jadwal <b>' . $jadwal_kode . '</b> berha
sil diedit');
199.           echo "sukses|admin/penjadwalan?periode_id=$per
iode_id";
200.       }
201.   }
202.   $jadwal_id = $this->input->post('jadwal_id');
203.   $jadwal_kode = $this->input->post('jadwal_kode');
204.   $periode_id = $this->input->post('periode_id');
205.   $this->M_database-
206.   >deleteData('t_jadwal', array('jadwal_id' => $jadwal_id));
207.   $this->session-
208.   >set_flashdata('hapus', 'Jadwal <b>' . $jadwal_kode . '</b> berh
asil dihapus');
209.   redirect("admin/penjadwalan?periode_id=$periode_id
");
210. }
211. function statusJadwal()
212. {
213.     $jadwal_id = $this->input->post('jadwal_id');
214.     $status = $this->input->post('status');
215.
216.     if ($status == 'Aktif') {
217.         $data = array('status' => 0);
218.         $badge = '<span class="badge-status badge-
not">Tidak aktif</span>';

```

```

219.         } else {
220.             $data = array('status' => 1);
221.             $badge = '<span class="badge-status badge-
ok">Aktif</span>';
222.         }
223.
224.         $this->M_database-
>updateData('t_jadwal', array('jadwal_id' => $jadwal_id), $d
ata);
225.
226.         echo $badge;
227.     }
228.
229. }

```

### Periode.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Periode extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $data['halaman_title'] = "Penjadwalan";
16.        $data['periode'] = $this->M_database-
>getAllData('t_periode', null, null, 'periode_id DESC');
17.        $this->template->load('admin/template', 'admin/v-
periode', $data);
18.    }
19.
20.    function tambahPeriode()
21.    {
22.        $periode_nama = $this->input-
>post('periode_nama');
23.
24.        $data = array(
25.            'periode_nama' => $periode_nama,
26.            'status' => 1
27.        );
28.
29.        $this->M_database->insertData('t_periode', $data);
30.        $this->session-
>set_flashdata('tambah', "Data <b>$periode_nama</b> berhasil
ditambahkan");
31.

```

```

32.         redirect('admin/periode', 'refresh');
33.     }
34.
35.     function editPeriode()
36.     {
37.         $periode_id = $this->input->post('periode_id');
38.         $periode_nama = $this->input-
39.             >post('periode_nama');
40.
41.         $data = array(
42.             'periode_nama' => $periode_nama
43.         );
44.
45.         $this->M_database-
46.             >updateData('t_periode', array('periode_id' => $periode_id),
47.             $data);
48.
49.         $this->session-
50.             >set_flashdata('edit', "Data periode berhasil diedit");
51.
52.         redirect('admin/periode', 'refresh');
53.     }
54.
55.     function hapusPeriode()
56.     {
57.         $periode_id = $this->input->post('periode_id');
58.         $periode_nama = $this->input-
59.             >post('periode_nama');
60.
61.         $this->M_database-
62.             >deleteData('t_periode', array('periode_id' => $periode_id))
63. ;
64.
65.         $this->session-
66.             >set_flashdata('hapus', "Data <b>$periode_nama</b> berhasil
67. dihapus");
68.
69.         redirect('admin/periode', 'refresh');
70.     }
71.
72.
73.     function statusPeriode()
74.     {
75.         $periode_id = $this->input->post('periode_id');
76.         $status = $this->input->post('status');
77.
78.         if ($status == 'Aktif') {
79.             $data = array('status' => 0);
80.             $badge = '<span class="badge-status badge-
81. not">Tidak aktif</span>';
82.         } else {
83.             $data = array('status' => 1);
84.             $badge = '<span class="badge-status badge-
85. ok">Aktif</span>';
86.         }
87.
88.         $this->M_database-
89.             >updateData('t_periode', array('periode_id' => $periode_id),
90.             $data);

```

```

74.         $this->M_database-
    >updateData('t_jadwal', array('periode_id' => $periode_id),
    $data);
75.         // $this->db-
    >set(array('status_bayar' => 0, 'status_bayar_bak' => 0))->
    update('t_mahasiswa');
76.
77.         echo $badge;
78.     }
79.
80. }

```

### Registrasi.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Registrasi extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $jadwal_kode = $this->input->get('jadwal_kode');
16.        $data['jadwalkosong'] = $this->input-
>get('jadwal_kode');
17.
18.        $data['mahasiswa'] = $this->M_database-
>getWhere('t_mahasiswa', 't_kelas', 't_mahasiswa.kelas_kode
= t_kelas.kelas_kode', array('status_mhs' => 1, 'status_baya
r' => 1), 'npm DESC');
19.        $data['jadwal'] = $this->M_database-
>getWhere('t_jadwal', null, null, array('status' => 1), 'jad
wal_kode ASC');
20.        $data['registrasi'] = $this->M_database-
>getRegistrasiMahasiswa(array('jadwal_kode' => $jadwal_kode)
);
21.        $data['halaman_title'] = "Registrasi Praktikum";
22.        // data hide table
23.        $data['visibility'] = '';
24.
25.        if ($jadwal_kode == null) {
26.            $data['visibility'] = 'hide';
27.        } else {
28.            $data['visibility'] = '';
29.        }
30.

```

```

31.         $this->template->load('admin/template', 'admin/v-
    registrasi', $data);
32.
33.     }
34.
35.     function simpanRegistrasi()
36.     {
37.         $jadwal_kode = $this->input->post('jadwal_kode');
38.         $npm = $this->input->post('npm');
39.         $status = $this->input->post('status');
40.         $id_praktikum = $this->input-
    >post('id_praktikum');
41.
42.         $data = array(
43.             'npm'=> $npm,
44.             'jadwal_kode'=> $jadwal_kode,
45.             'status_modul'=> 0,
46.             'tgl_modul'=> "-",
47.             'nilai_tugas'=> 0,
48.             'nilai_ujian'=> 0
49.         );
50.
51.         if ($status == "add") {
52.             $this->M_database-
    >insertData('registrasi_praktikum', $data);
53.             $message = "add";
54.             $checkbox = '<label>
55.                             <input type="checkbox" class="'
filled-in" data-
    npm="'. $npm .'" value="'. $npm .'"?>" name="npm" checked>
56.                             <span class="checkbox" style="
    left: 10px"></span>
57.                         </label>';
58.         } else {
59.             $this->M_database-
    >deleteData('registrasi_praktikum', array('jadwal_kode' => $
    jadwal_kode, 'npm' => $npm));
60.             $message = "delete";
61.             $checkbox = '<label>
62.                             <input type="checkbox" class="'
filled-in" data-
    npm="'. $npm .'" value="'. $npm .'"?>" name="npm">
63.                             <span class="checkbox" style="
    left: 10px"></span>
64.                         </label>';
65.         }
66.
67.         echo "$message|$checkbox";
68.     }
69.
70.     function view_pindah()
71.     {
72.         $jadwal_kode = $this->input->get('jadwal_kode');
73.         $data['jadwalkosong'] = $this->input-
    >get('jadwal_kode');
74.

```

```

75.      $data['mahasiswa'] = $this->M_database-
    >getDetailTugas(array('jadwal_kode' => $jadwal_kode));
76.      $data['jadwal'] = $this->M_database-
    >getWhere('t_jadwal', null, null, array('status' => 1), 'jad-
    wal_kode ASC');
77.      $data['halaman_title'] = "Pindah Praktikum";
78.      // $data['praktikum1'] = $this->M_database-
    >maxwhere('t_absensi','jadwal_kode', $jadwal_kode, 'pertemua-
    n')->pertemuan;
79.      $praktikum = $this->M_database-
    >maxwhere('t_absensi','jadwal_kode', $jadwal_kode, 'pertemua-
    n')->pertemuan;
80.      if ($praktikum == '') {
81.          $data['praktikum1'] = 'Belum ada pertemuan';
82.          $data['pertemuan1'] = 0;
83.      } else {
84.          $data['praktikum1'] = $praktikum;
85.          $data['pertemuan1'] = $praktikum;
86.      }
87.      // data hide table
88.      $data['visibility'] = '';
89.
90.      if ($jadwal_kode == null) {
91.          $data['visibility'] = 'hide';
92.      } else {
93.          $data['visibility'] = '';
94.      }
95.
96.      $this->template->load('admin/template', 'admin/v-
    pindah-praktikum', $data);
97.  }
98.
99.  function pindahPraktikum()
100.  {
101.      $terpilih = $this->input->post('terpilih');
102.      $id_praktikum = $this->input-
    >post('id_praktikum');
103.      $praktikum_awal = $this->input-
    >post('praktikum_awal');
104.      $praktikum_tujuan = $this->input-
    >post('praktikum_tujuan');
105.
106.      $praktikum1 = $this->M_database-
    >maxwhere('t_absensi','jadwal_kode', $praktikum_awal, 'per-
    temuan')->pertemuan;
107.      $praktikum2 = $this->M_database-
    >maxwhere('t_absensi','jadwal_kode', $praktikum_tujuan, 'per-
    temuan')->pertemuan;
108.
109.      if ($terpilih != null) {
110.          if($praktikum1 == $praktikum2){
111.
112.              foreach($terpilih as $npm){
113.
114.                  $data = array(

```

```

115.                               'jadwal_kode' => $praktikum_tu
   juan,
116.                               'npm' => $npm
117. );
118.
119.             $this->M_database-
   >updateData('registrasi_praktikum', array('jadwal_kode' => $praktikum_awal, 'npm' => $npm) , $data);
120.             $this->M_database-
   >updateData('t_absensi', array('jadwal_kode' => $praktikum_awal, 'npm' => $npm) , $data);
121. }
122.
123.             $this->session-
   >set_flashdata('edit', 'Data pindah praktikum berhasil disimpan');
124.
125.             redirect("admin/registrasi/view_pindah?jadwal_kode=$praktikum_awal&id_praktikum=$id_praktikum", 'refresh');
126. }
127.
128. else{
129.             $this->session-
   >set_flashdata('warning', 'Pindah praktikum gagal, jumlah pertemuan tidak sama');
130.             redirect("admin/registrasi/view_pindah?jadwal_kode=$praktikum_awal&id_praktikum=$id_praktikum", 'refresh');
131. }
132. } else {
133.             $this->session-
   >set_flashdata('edit', 'Tidak ada mahasiswa yang dipilih');
134.             redirect("admin/registrasi/view_pindah?jadwal_kode=$praktikum_awal&id_praktikum=$id_praktikum", 'refresh')
   ;
135. }
136. }
137.
138. function hapusRegistrasi()
139. {
140.     $npm = $this->input->post('npm');
141.     $jadwal_kode = $this->input->post('jadwal_kode');
142.     $id_praktikum = $this->input-
   >post('id_praktikum');
143.
144.     $this->M_database-
   >deleteData('registrasi_praktikum', array('npm' => $npm, 'jadwal_kode' => $jadwal_kode));
145.     $this->M_database-
   >deleteData('t_absensi', array('npm' => $npm, 'jadwal_kode' => $jadwal_kode));
146.
147.     redirect("admin/registrasi/view_pindah?jadwal_kode=$jadwal_kode&id_praktikum=$id_praktikum", 'refresh');
148.

```

```

149.     }
150.
151.     function cekPertemuan()
152.     {
153.         $jadwal_kode = $this->input->post('jadwal_kode');
154.         $pertemuan1 = $this->input->post('pertemuan1');
155.         $pertemuan2 = $this->M_database-
>maxwhere('t_absensi','jadwal_kode', $jadwal_kode, 'pertemua-
n')->pertemuan;
156.
157.         if ($pertemuan2 == '') {
158.             $pertemuan = 'Belum ada pertemuan';
159.             $msg = 'gagal';
160.             $pertemuan2 = 0;
161.         }
162.
163.         if ($pertemuan1 == $pertemuan2) {
164.             $msg = 'sukses';
165.             if ($pertemuan2 == 0) {
166.                 $pertemuan = 'Belum ada pertemuan';
167.             } else {
168.                 $pertemuan = $pertemuan2;
169.             }
170.         } else {
171.             $msg = 'gagal';
172.             if ($pertemuan2 == 0) {
173.                 $pertemuan = 'Belum ada pertemuan';
174.             } else {
175.                 $pertemuan = $pertemuan2;
176.             }
177.         }
178.
179.         echo "$msg|$pertemuan";
180.     }
181.
182.     function fetch_mahasiswa() {
183.         $fetch_data = $this->M_database-
>render_datatable();
184.         $data = array();
185.         foreach ($fetch_data as $row => $value) {
186.             $cek = $this->db-
>query("SELECT COUNT(registrasi_id) AS exist FROM registrasi-
_praktikum WHERE jadwal_kode='pw12b' AND npm='$value-
>npm'")->row();
187.
188.             $checked = '';
189.             $marker = '';
190.             $val = '1';
191.
192.             if ($cek->exist == 1) {
193.                 $checked = 'checked';
194.                 $marker = 'hide';
195.                 $val = '1';
196.             }
197.             else
198.             {

```

```

199.                     $checked = '';
200.                     $marker = '';
201.                     $val = '2';
202.                 }
203.
204.                     $status = '';
205.
206.                     if ($value->status_bayar == 0) {
207.                         $status = '<span class="badge-
    status badge-not">Belum Lunas</span>';
208.                         $hide_checkbox = "disabled";
209.                     } else {
210.                         $status = '<span class="badge-
    status badge-ok">Lunas</span>';
211.                         $hide_checkbox = "";
212.                     }
213.                     $sub_array = array();
214.                     $sub_array[] = $row+1;
215.                     $sub_array[] = $value->npm;
216.                     $sub_array[] = $value->nama;
217.                     $sub_array[] = $value->kelas_nama;
218.                     $sub_array[] = '<div class="center-
    align">' . $status . '</div>';
219.                     $sub_array[] = '<div class="center-
    align"><label>
220.                                     <input type="checkbox" cla-
    ss="filled-in checkboxes2". $value->npm.' data-npm="'. $value-
    >npm.'"'. $hide_checkbox.' value="'. $value-
    >npm.'" name="npm[]' . $checked .'>
221.                                     <span class="checkbox2" st-
    yle="left: 10px"></span>
222.                                 </label></div>';
223.                     $data[] = $sub_array;
224.                 }
225.                 $output = array(
226.                     "draw"          => intval($_POST["draw"]),
227.                     "recordsTotal"  => $this->M_database-
    >get_all_data(),
228.                     "recordsFiltered" => $this->M_database-
    >get_filtered_data(),
229.                     "data"           => $data
230.                 );
231.                 echo json_encode($output);
232.             }
233.
234.         }

```

### Ruangan.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Ruangan extends CI_Controller {

```

```

5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session();
11.    }
12.
13.    public function index()
14.    {
15.        $data['ruangan'] = $this->M_database-
>getAllData('t_ruangan', null, null);
16.        $data['halaman_title'] = "Master Ruangan";
17.        $this->template->load('admin/template', 'admin/v-
ruangan', $data);
18.    }
19.
20.    function tambahRuangan()
21.    {
22.        $ruangan_nama = $this->input-
>post('ruangan_nama');
23.        $kapasitas = $this->input->post('kapasitas');
24.
25.        $data = array(
26.            'ruangan_nama' => $ruangan_nama,
27.            'ruangan_kapasitas' => $kapasitas
28.        );
29.
30.        $this->session-
>set_flashdata('tambah', "Data ruangan <b>$ruangan_nama</b>
berhasil disimpan");
31.        $this->M_database->insertData('t_ruangan', $data);
32.
33.        redirect('admin/ruangan', 'refresh');
34.    }
35.
36.    function editRuangan()
37.    {
38.        $ruangan_id = $this->input->post('ruangan_id');
39.        $ruangan_nama = $this->input-
>post('ruangan_nama');
40.        $kapasitas = $this->input->post('kapasitas');
41.
42.        $data = array(
43.            'ruangan_nama' => $ruangan_nama,
44.            'ruangan_kapasitas' => $kapasitas
45.        );
46.
47.        $this->session-
>set_flashdata('edit', "Data ruangan <b>$ruangan_nama</b> be-
rhasil diedit");
48.        $this->M_database-
>updateData('t_ruangan', array('ruangan_id' => $ruangan_id),
$data);
49.
50.        redirect('admin/ruangan', 'refresh');

```

```

51.      }
52.
53.      function hapusRuangan()
54.      {
55.          $ruangan_id = $this->input->post('ruangan_id');
56.          $ruangan_nama = $this->input-
>post('ruangan_nama');
57.
58.          $this->session-
>set_flashdata('hapus', "Data ruangan <b>$ruangan_nama</b> b
erhasil dihapus");
59.          $this->M_database-
>deleteData('t_ruangan', array('ruangan_id' => $ruangan_id ))
);
60.
61.          redirect('admin/ruangan', 'refresh');
62.      }
63.
64.  }

```

## Ticketing.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. use Minishlink\WebPush\WebPush;
5. use Minishlink\WebPush\Subscription;
6.
7. class Ticketing extends CI_Controller {
8.
9.     function __construct()
10.    {
11.        parent::__construct();
12.        $this->load->model("M_database");
13.        cek_session();
14.    }
15.
16.    public function index()
17.    {
18.        $data['jumlah_mhs'] = $this->M_database-
>countWhere('t_mahasiswa', 'status_mhs = 1');
19.        $data['jumlah_kelas'] = $this->M_database-
>countWhere('t_kelas', 'kelas_status = 1');
20.        $data['jumlah_jadwal'] = $this->M_database-
>countWhere('t_jadwal', 'status = 1');
21.        $data['jumlah_asisten'] = $this->M_database-
>countWhere('t_asisten', 'status = 1');
22.        $data['jadwal'] = $this->M_database-
>getWhere('t_jadwal', null, null, array('status' => 1), 'jad
wal_kode ASC');
23.        $data['pindah_shift'] = $this->M_database-
>getAllData('t_ticket', 't_jadwal', 't_ticket.shift_asal=t_j
adwal.jadwal_kode', 'ticket_status DESC', 'ticket_id ASC');

```

```

24.      $data['pesan_mhs'] = $this->M_database-
>getPesanMhs();
25.      $data['halaman_title'] = "E - Ticketing";
26.      $this->template->load('admin/template', 'admin/v-
ticketing', $data);
27.  }
28.
29.  public function fetch_jadwal()
30.  {
31.      $matkum_id = $this->input->get('matkum_id');
32.      $shift_asal = $this->input->get('shift_asal');
33.
34.      $jadwal = $this->M_database-
>getWhere('t_jadwal', 't_ruangan', 't_jadwal.ruangan_id=t_ru
angan.ruangan_id', array('status' => 1, 'matkum_id' => $matk
um_id, 'jadwal_kode !=' => $shift_asal), 'jadwal_kode ASC');
35.
36.      foreach ($jadwal->result() as $key => $value) {
37.          $mhs = $this->M_database-
>countWhere('registrasi_praktikum', array('jadwal_kode' => $
value->jadwal_kode));
38.          $opt .= '<option value="'. $value-
>jadwal_kode.'">'.strtoupper($value-
>jadwal_kode). '</option>';
39.          $item .= '<div class="jadwal-
ticket" id="'. $value->jadwal_kode.'">
<span>'. $value-
>jadwal_kode. '</span>
<span style="margin-
top:8px">'. $value->jadwal_hari. '</span>
<span>'. $value-
>jadwal_jam. '</span>
<div><span class="badge-
status orange-
gradient" style="margin:8px auto 4px">Jumlah PC: '. $value-
>ruangan_kapasitas.'</span><span class="badge-status badge-
not">Mahasiswa: '. $mhs. '</span></div>
</div>';
40.      }
41.
42.      echo "$opt|$item";
43.  }
44.
45.  public function balas_pesan()
46.  {
47.      $npm = $this->input->post('npm');
48.      $nama = $this->input->post('nama');
49.      $pesan = $this->input->post('pesan');
50.
51.      $data = array (
52.          'npm' => $npm,
53.          'nama' => $nama,
54.          'pesan' => $pesan,
55.          'pesan_status' => 0,
56.          'pengirim' => 'admin',
57.          'date_created' => date('Y-m-d H:i:s')
58.      );
59.
60.
61.
62.

```

```

63.          );
64.
65.          $this->M_database-
66.              >insertData('t_pesanan_mhs', $data);
67.              $token = $this->db->where('npm', $npm)-
68.                  >get('t_device_token')->row();
69.                  $this->kirimNotifikasi($token->endpoint, $token-
70.                      >p256dh, $token->auth, $pesan);
71.          }
72.
73.      public function proses_ticket()
74.      {
75.          $ticket_id = $this->input->post('ticket_id');
76.          $ticket_status = $this->input-
77.              >post('ticket_status');
78.          $shift_tujuan = $this->input-
79.              >post('shift_tujuan');
80.
81.          $status = '';
82.          if ($ticket_status == 0) {
83.              $status = '<span class="badge-
84. status yellow darken-1">Pending</span>';
85.          } else if ($ticket_status == 1) {
86.              $status = '<span class="badge-
87. status green accent-4 white-text">Diproses</span>';
88.          } else if ($ticket_status == 2) {
89.              $status = '<span class="badge-
90. status blue white-text">Disetujui</span>';
91.          } else {
92.              $status = '<span class="badge-
93. status red darken-1 white-text">Dibatalkan</span>';
94.          }
95.
96.          $data = array (
97.              'ticket_status' => $ticket_status,
98.              'shift_tujuan' => $shift_tujuan
99.          );
100.
101.         $this->M_database-
102.             >updateData('t_ticket', array('ticket_id' => $ticket_id), $d
ata);
103.
104.         echo "$status|$ticket_status|$shift_tujuan";
105.     }
106.
107.     function fetch_pesan()
108.     {
109.         $npm = $this->input->get('npm');
110.         $this->M_database-
111.             >updateData('t_pesanan_mhs', array('npm' => $npm, 'pengirim' =
112.                 > 'mahasiswa'), array('pesan_status' => 1));
113.         $pesan = $this->M_database-
114.             >getwhere('t_pesanan_mhs', 't_mahasiswa', 't_pesanan_mhs.npm=t_
115. mahasiswa.npm', array('t_pesanan_mhs.npm' => $npm), 'pesan_id A
SC');
116.
117.         foreach ($pesan->result() as $key => $value) {

```

```

103.          $pos = "";
104.          if ($value->pengirim == 'admin') {
105.              $pos = "chat-text-right";
106.              $color = "grey-text text-lighten-1";
107.          } else {
108.              $pos = "chat-text-left";
109.              $color = "grey-text text-darken-1";
110.          }
111.          echo '<div class="'. $pos .'"><span>' . $value-
>pesan . '<br><small class="right ' . $color . '">' . $value-
>date_created . '</small></span></div>';
112.      }
113.      echo '<div id="end-content"></div>';
114.  }
115.
116.  function kirimNotifikasi($endpoint, $p256dh, $auth, $p
esan)
117.  {
118.      $url = 'http://localhost/clientsilabtif/info_prakt
ikum';
119.
120.      // array of notifications
121.      $notifications = [
122.          [
123.              'subscription' => Subscription::create([
124.                  "endpoint" => $endpoint,
125.                  "keys" => [
126.                      'p256dh' => $p256dh,
127.                      'auth' => $auth
128.                  ],
129.              ]),
130.              'payload' => "Admin Chat|$pesan|$url",
131.          ]
132.      ];
133.
134.      $auth = array(
135.          'VAPID' => array(
136.              'subject' => '*****',
137.              'publicKey' => '*****',
138.              'privateKey' => '*****' // in the real
world, this would be in a secret file
139.          ),
140.      );
141.
142.      $webPush = new WebPush($auth);
143.
144.      // send multiple notifications with payload
145.      foreach ($notifications as $notification) {
146.          $webPush->sendNotification(
147.              $notification['subscription'],
148.              $notification['payload'] // optional (defa
ults null)
149.          );
150.      }
151.
152.      /**

```

```

153.          * Check sent results
154.          * @var MessageSentReport $report
155.          */
156.          foreach ($webPush->flush() as $report) {
157.              $endpoint = $report->getRequest()->getUri()-
>__toString();
158.
159.              // if ($report->isSuccess()) {
160.                  // echo "[v] Message sent successfully for su
bscription {$endpoint}.";
161.                  // } else {
162.                      // echo "[x] Message failed to sent for subsc
ription {$endpoint}: {$report->getReason()}";
163.                  // }
164.          }
165.
166.          /**
167.             * send one notification and flush directly
168.             * @var \Generator<MessageSentReport> $sent
169.             */
170.             $sent = $webPush->sendNotification(
171.                 $notifications[0]['subscription'],
172.                 $notifications[0]['payload'], // optional (def
aults null)
173.                 true // optional (defaults false)
174.             );
175.
176.             // redirect('admin/pengumuman', 'refresh');
177.         }
178.
179.     }

```

## Asisten\_absensi.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Asisten_absensi extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {

```

```

15.      $asisten_id = $this->session-
    >userdata('asisten_id');
16.      $jadwal_kode = $this->input->get('idp');
17.
18.      // data hide table
19.      $data['visibility'] = '';
20.      if ($jadwal_kode == null) {
21.          $data['visibility'] = 'hide';
22.      } else {
23.          $data['visibility'] = '';
24.          $jadwal = $this->db-
    >where('jadwal_kode', $jadwal_kode)->get('t_jadwal')->row();
25.          if ($jadwal-
    >asisten_1 != $asisten_id && $jadwal-
    >asisten_2 != $asisten_id) {
26.              redirect('akses_ditolak');
27.          }
28.      }
29.
30.      $data['jadwalkosong'] = $jadwal_kode;
31.      $data['mahasiswa'] = $this->M_database-
    >getRegistrasiMahasiswa(array('jadwal_kode' => $jadwal_kode)
    );
32.      $data['jadwal'] = $this->M_database-
    >getJadwalAsisten("t_jadwal.asisten_1 = $asisten_id OR t_jad-
    wal.asisten_2 = $asisten_id");
33.      $data['halaman_title'] = ucfirst($this->uri-
    >segment(2). " ". $this->uri->segment(3));
34.      $data['pertemuan'] = $this->M_database-
    >cekPertemuan($jadwal_kode);
35.      $data['absen_tgl'] = $this->M_database-
    >cekTanggal($jadwal_kode);
36.
37.      $this->template->load('asisten/template-
    asisten', 'asisten/v-absensi', $data);
38.
39.  }
40.
41.  function do_absen()
42.  {
43.      $jadwal_kode = $this->input->post('jadwal_kode');
44.      $npm = $this->input->post('npm');
45.      $kehadiran = $this->input->post('kehadiran');
46.      $pertemuan = $this->input->post('pertemuan');
47.      $absen_tgl = $this->input->post('absen_tgl');
48.
49.      $data = array(
50.          'jadwal_kode'=>$jadwal_kode,
51.          'pertemuan'=>$pertemuan,
52.          'npm'=>$npm,
53.          'absen_tgl'=> $absen_tgl
54.      );
55.
56.      // cek absensi
57.      $cek = $this->M_database-
    >cekAbsen('t_absensi', $jadwal_kode, $npm, $pertemuan);

```

```

58.
59.         if($kehadiran == 'Tidak Hadir' && $cek-
   >hadir == 0)
60.         {
61.             $this->M_database-
   >insertData('t_absensi',$data);
62.             $badge = '<span class="badge-status badge-
   ok">Hadir</span>';
63.         }
64.         else if($kehadiran == 'Hadir' && $cek->hadir == 1)
65.         {
66.             $this->M_database-
   >hapusAbsen('t_absensi', 'jadwal_kode', 'npm', 'pertemuan',
   $jadwal_kode, $npm, $pertemuan);
67.             $badge = '<span class="badge-status badge-
   not">Tidak Hadir</span>';
68.         }
69.         else
70.         {
71.             $badge = '<span class="badge-status badge-
   not">Tidak Hadir</span>';
72.         }
73.     }
74.
75.     echo $badge;
76. }
77.
78. function do_absen_asisten()
79. {
80.     $jadwal_kode = $this->input->post('jadwal_kode');
81.     $asisten_id = $this->input->post('asisten_id');
82.     $password = $this->input->post('password');
83.     $kehadiran = $this->input->post('kehadiran');
84.     $pertemuan = $this->input->post('pertemuan');
85.     $absen_tgl = $this->input->post('absen_tgl');
86.
87.     $data = array(
88.         'jadwal_kode'=>$jadwal_kode,
89.         'pertemuan'=>$pertemuan,
90.         'asisten_id'=>$asisten_id,
91.         'absen_tgl'=>$absen_tgl
92.     );
93.
94.     // cek absensi
95.     $cek = $this->M_database-
   >cekAbsenAsisten('t_absensi_asisten', $jadwal_kode, $asisten_
   _id, $pertemuan);
96.     // get asisten
97.     $asisten = $this->db-
   >get_where('t_asisten', array('asisten_id' => $asisten_id))-
   >row();
98.
99.     if (password_verify($password, $asisten-
   >password)) {
100.         $pass = 'benar';

```

```

101.           if($kehadiran == 'Tidak Hadir' && $cek-
>hadir == 0) {
102.               $this->M_database-
>insertData('t_absensi_asisten',$data);
103.               $badge = '<span class="badge-status badge-
ok">Hadir</span>';
104.           }
105.           else if($kehadiran == 'Hadir' && $cek-
>hadir == 1) {
106.               $this->M_database-
>hapusAbsen('t_absensi_asisten', 'jadwal_kode', 'asisten_id'
, 'pertemuan', $jadwal_kode, $asisten_id, $pertemuan);
107.               $badge = '<span class="badge-status badge-
not">Tidak Hadir</span>';
108.           }
109.           else{
110.               $badge = '<span class="badge-status badge-
not">Tidak Hadir</span>';
111.           }
112.       } else {
113.           $pass = 'salah';
114.           $badge = '<span class="badge-status badge-
not">Tidak Hadir</span>';
115.       }
116.       echo "$pass|$badge";
117.   }
118.
119. }

```

## Asisten dashboard.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Asisten_dashboard extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $asisten_id = $this->session-
>userdata('asisten_id');
16.        $hari = date('D');
17.        switch ($hari) {
18.            case 'Sun':
19.                $hari = 'Minggu';
20.                break;
21.            case 'Mon':

```

```

22.             $hari = 'Senin';
23.             break;
24.         case 'Tue':
25.             $hari = 'Selasa';
26.             break;
27.         case 'Wed':
28.             $hari = 'Rabu';
29.             break;
30.         case 'Thu':
31.             $hari = 'Kamis';
32.             break;
33.         case 'Fri':
34.             $hari = 'Jumat';
35.             break;
36.         case 'Sat':
37.             $hari = 'Sabtu';
38.             break;
39.         default:
40.             $hari = 'Tidak ada';
41.             break;
42.     }
43.     $data['hari_ini'] = $hari;
44.     $data['halaman_title'] = ucfirst($this->uri-
>segment(2) . " " . $this->uri->segment(3));
45.     $data['jadwal'] = $this->M_database-
>getJadwalDashboard("t_jadwal.asisten_1 = $asisten_id OR t_j
adwal.asisten_2 = $asisten_id", 'jadwal_jam ASC', array('jad
wal_hari' => $hari))->result();
46.     $data['hari_ini_title'] = "Jadwal hari ini,";

47.     // $now = date('Y-m-d H:i:s');
48.     // $tugas = $this->db-
>where("batas_waktu < '$now')->get('t_tugas')->result();
49.     // foreach ($tugas as $key => $value) {
50.     //   $this->M_database-
>updateData('t_tugas', array('tugas_id' => $value-
>tugas_id), array('tugas_status' => 0));
51.     // }
52.     $this->template->load('asisten/template-
asisten', 'asisten/v-dashboard', $data);
53.   }
54.
55.   public function lihat_jadwal()
56.   {
57.     $asisten_id = $this->session-
>userdata('asisten_id');
58.     $jadwal_hari = $this->input->get('jadwal_hari');
59.     $jadwal = $data['jadwal'] = $this->M_database-
>getJadwalDashboard("t_jadwal.asisten_1 = $asisten_id OR t_j
adwal.asisten_2 = $asisten_id", 'jadwal_jam ASC', array('jad
wal_hari' => $jadwal_hari))->result();
60.     foreach ($jadwal as $key => $value) {
61.       $asisten_1 = $this->db-
>where('asisten_id', $value->asisten_1)->get('t_asisten')-
>row();

```

```

62.           $asisten_2 = $this->db-
>where('asisten_id', $value->asisten_2)->get('t_asisten')-
>row();
63.
64.           $namal = explode(' ', $asisten_1-
>asisten_nama);
65.           if (sizeof($namal) > 1) {
66.               if ($namal[0] == 'M' || $namal[0] == 'M.' ||
$namal[0] == 'Muhamad' || $namal[0] == 'Muhammad' || $namal[0] ==
= 'Mohamed' || $namal[0] == 'Mohammed' || $namal[0] ==
= 'Moch' || $namal[0] == 'Moch.' || $namal[0] == 'Mochamad' ||
$namal[0] == 'Mochammad' || $namal[0] == 'Much' || $namal[0] ==
= 'Much.' || $namal[0] == 'Muchamad' || $namal[0] == 'M
uchhammad') {
67.                   $panggilan_1 = $namal[1];
68.               } else {
69.                   $panggilan_1 = $namal[0];
70.               }
71.           } else {
72.               $panggilan_1 = $asisten_1->asisten_nama;
73.           }
74.
75.
76.           $nama2 = explode(' ', $asisten_2-
>asisten_nama);
77.           if (sizeof($nama2) > 1) {
78.               if ($nama2[0] == 'M' || $nama2[0] == 'M.' ||
$nama2[0] == 'Muhamad' || $nama2[0] == 'Muhammad' || $nam
a2[0] == 'Mohamed' || $nama2[0] == 'Mohammed' || $nama2[0] ==
= 'Moch' || $nama2[0] == 'Moch.' || $nama2[0] == 'Mochamad' ||
$nama2[0] == 'Mochammad' || $nama2[0] == 'Much' || $nam
a2[0] == 'Much.' || $nama2[0] == 'Muchamad' || $nama2[0] == 'M
uchhammad') {
79.                   $panggilan_2 = $nama2[1];
80.               } else {
81.                   $panggilan_2 = $nama2[0];
82.               }
83.           } else {
84.               $panggilan_2 = $asisten_2->asisten_nama;
85.           }
86.
87.           echo '<div class="jadwal-item">
88.                     <span class="txt-hari">'.$value-
>jadwal_hari.'</span>
89.                     <span class="txt-jam">'.$value-
>jadwal_jam.'</span>
90.
91.                     <span class="txt-
kode">'.strtoupper($value-
>jadwal_kode).'
92.                     <span class="txt-
asisten">'.$panggilan_1.' & '.$panggilan_2.'

```

```

93.                                     <div class="line-dashed"></div>
94.                                     <span class="txt-
ruangan">' . $value->ruangan_nama . '</span>
95.                                     </div>';
96.                                     }
97.                                     if ($jadwal == null) {
98.                                         echo "kosong|<div class='jadwal-item center-
align'>Tidak ada jadwal</div>";
99.                                     }
100.                                     }
101.                                     }
102.                                     public function akses()
103.                                     {
104.                                         $this->template->load('asisten/template-
asisten', 'asisten/v-akses');
105.                                     }
106.                                     }
107.                                     }

```

### Asisten kehadiran.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Asisten_kehadiran extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $asisten_id = $this->session-
>userdata('asisten_id');
16.
17.        $data['jadwal'] = $this->M_database-
>getAllJadwalAsisten("t_jadwal.asisten_1 = $asisten_id OR t_
jadwal.asisten_2 = $asisten_id", 'status DESC');
18.
19.        $this->template->load('asisten/template-
asisten', 'asisten/v-kehadiran', $data);
20.
21.    }
22.
23. }

```

### Asisten penilaian.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Asisten_penilaian extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $asisten_id = $this->session-
>userdata('asisten_id');
16.        $jadwal_kode = $this->input->get('idp');
17.
18.        // data hide table
19.        $data['visibility'] = '';
20.        if ($jadwal_kode == null) {
21.            $data['visibility'] = 'hide';
22.        } else {
23.            $data['visibility'] = '';
24.            $jadwal = $this->db-
>where('jadwal_kode', $jadwal_kode)->get('t_jadwal')->row();
25.            if ($jadwal-
>asisten_1 != $asisten_id && $jadwal-
>asisten_2 != $asisten_id) {
26.                redirect('akses_ditolak');
27.            }
28.        }
29.
30.        $data['jadwalkosong'] = $jadwal_kode;
31.        $data['mahasiswa'] = $this->M_database-
>getDetailTugas(array('jadwal_kode' => $jadwal_kode));
32.        $data['jadwal'] = $this->M_database-
>getJadwalAsisten("t_jadwal.asisten_1 = $asisten_id OR t_jad-
wal.asisten_2 = $asisten_id");
33.
34.        $this->template->load('asisten/template-
asisten', 'asisten/v-penilaian', $data);
35.
36.    }
37.
38.    function penilaian() {
39.        $npm = $this->input->post('npm');
40.        $jadwal_kode = $this->input->post('jadwal_kode');
41.        $nilai_tugas = $this->input-
>post('nilai_tugas');
42.        $nilai_ujian = $this->input->post('nilai_ujian');
43.        if ($nilai_tugas == '') {
44.            $nilai_tugas = 0;
45.        }
46.        if ($nilai_ujian == '') {

```

```

47.             $nilai_ujian = 0;
48.         }
49.         $total = $nilai_tugas + $nilai_ujian;
50.         $this->M_database-
>updateData('registrasi_praktikum', array('jadwal_kode' => $jadwal_kode, 'npm' => $npm), array('nilai_tugas' => $nilai_tugas, 'nilai_ujian' => $nilai_ujian));
51.         echo "$nilai_tugas|$nilai_ujian|$total";
52.     }
53.
54. }

```

### Asisten\_tugas.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. use Minishlink\WebPush\WebPush;
5. use Minishlink\WebPush\Subscription;
6.
7. class Asisten_tugas extends CI_Controller {
8.
9.     function __construct()
10.    {
11.        parent::__construct();
12.        $this->load->model('M_database');
13.        cek_session_null();
14.    }
15.
16.
17.    public function index()
18.    {
19.        $asisten_id = $this->session-
>userdata('asisten_id');
20.        $jadwal_kode = $this->input->get('idp');
21.        $nomor_tugas = $this->M_database-
>countWhere('t_tugas', array('jadwal_kode' => $jadwal_kode))
+ 1;
22.        $kode_tugas = "$jadwal_kode-$nomor_tugas";
23.
24.        // data hide table
25.        $data['visibility'] = '';
26.        if ($jadwal_kode == null) {
27.            $data['visibility'] = 'hide';
28.        } else {
29.            $data['visibility'] = '';
30.            $jadwal = $this->db-
>where('jadwal_kode', $jadwal_kode)->get('t_jadwal')->row();
31.            if ($jadwal-
>asisten_1 != $asisten_id && $jadwal-
>asisten_2 != $asisten_id) {
32.                redirect('akses_ditolak');
33.            }

```

```

34.         }
35.
36.         $data['jadwalkosong'] = $jadwal_kode;
37.         $data['tugas'] = $this->M_database-
>getWhere('t_tugas', null, null, array('jadwal_kode' => $jad-
    wal_kode), null);
38.         $data['foto_tugas'] = $this->M_database-
>getWhere('t_tugas_image', null, null, array('tugas_kode' =>
    $kode_tugas), null);
39.         $data['jadwal'] = $this->M_database-
>getJadwalAsisten("t_jadwal.asisten_1 = $asisten_id OR t_jad-
    wal.asisten_2 = $asisten_id");
40.         $data['halaman_title'] = ucfirst($this->uri-
    >segment(2). " ". $this->uri->segment(3));
41.         $data['pertemuan'] = $this->M_database-
    >cekPertemuan($jadwal_kode);
42.         $data['absen_tgl'] = $this->M_database-
    >cekTanggal($jadwal_kode);
43.
44.         $this->template->load('asisten/template-
    asisten', 'asisten/v-tugas', $data);
45.
46.     }
47.
48.     function view_penilaian()
49.     {
50.         $jadwal_kode = $this->input->get('idp');
51.         $tugas_id = $this->input->get('t');
52.         $data['mahasiswa'] = $this->M_database-
    >getDetailTugas(array('jadwal_kode' => $jadwal_kode));
53.         $this->template->load('asisten/template-
    asisten', 'asisten/v-tugas-penilaian', $data);
54.     }
55.
56.     function penilaian() {
57.         $npm = $this->input->post('npm');
58.         $tugas_id = $this->input->post('tugas_id');
59.         $jadwal_kode = $this->input->post('jadwal_kode');
60.         $nilai = $this->input->post('nilai');
61.         $nilai_asal = $this->input->post('nilai_asal');
62.
63.         if ($nilai == null) {
64.             $nilai = 0;
65.         }
66.
67.         $n_tugas = $nilai/10;
68.
69.         $cek_nilai = $this->db-
>where(array('npm' => $npm, 'tugas_id' => $tugas_id))->
    get('t_tugas_nilai')->row();
70.         $nilai_tugas = $this->db-
>where(array('npm' => $npm, 'jadwal_kode' => $jadwal_kode))-
    >get('registrasi_praktikum')->row();
71.
72.
73.         $data = array(

```

```

74.          'npm' => $npm,
75.          'tugas_id' => $tugas_id,
76.          'nilai' => $nilai,
77.          'jadwal_kode' => $jadwal_kode
78.      );
79.
80.      if ($cek_nilai) {
81.          $n_akhir = $nilai_asal /10;
82.          $nilai_fix = ($nilai_tugas-
>nilai_tugas - $n_akhir) + $n_tugas;
83.          if ($nilai > 0) {
84.              $this->M_database-
>updateData('t_tugas_nilai', array('npm' => $npm, 'tugas_id' => $tugas_id), array('nilai' => $nilai));
85.              $this->M_database-
>updateData('registrasi_praktikum', array('npm' => $npm, 'ja-
dwal_kode' => $jadwal_kode), array('nilai_tugas' => $nilai_f-
ix));
86.              $penilaian = $nilai;
87.          } else {
88.              $this->M_database-
>deleteData('t_tugas_nilai', array('npm' => $npm, 'tugas_id' => $tugas_id));
89.              $this->M_database-
>updateData('registrasi_praktikum', array('npm' => $npm, 'ja-
dwal_kode' => $jadwal_kode), array('nilai_tugas' => $nilai_f-
ix));
90.              $penilaian = 0;
91.          }
92.      } else {
93.          $this->M_database-
>insertData('t_tugas_nilai', $data);
94.          $this->M_database-
>updateData('registrasi_praktikum', array('npm' => $npm, 'ja-
dwal_kode' => $jadwal_kode), array('nilai_tugas' => $nilai_t-
ugas->nilai_tugas + $n_tugas));
95.          $penilaian = $nilai;
96.      }
97.      echo $penilaian;
98.  }
99.
100. function tambahTugas() {
101.
102.     $tugas_kode = $this->input->post('tugas_kode');
103.     $tugas_judul = $this->input->post('tugas_judul');
104.     $tugas_deskripsi = $this->input-
>post('tugas_deskripsi');
105.     $tugas_judul = $this->input->post('tugas_judul');
106.     $tugas_status = $this->input-
>post('tugas_status');
107.     $jadwal_kode = $this->input->post('jadwal_kode');
108.     $batas_waktu = $this->input-
>post('tanggal')." ".$this->input->post('jam').":00";
109.
110.     $l = explode(" ", $batas_waktu);
111.     $waktu = date_create_from_format("d/m/Y", $l[1]);

```

```

112.         $tanggal = date_format($waktu, 'Y-m-d');
113.         $jam = $1[2];
114.
115.         $data = array(
116.             'tugas_kode' => $tugas_kode,
117.             'tugas_judul' => $tugas_judul,
118.             'tugas_deskripsi' => $tugas_deskripsi,
119.             'batas_waktu' => $tanggal." ".$jam,
120.             'jadwal_kode' => $jadwal_kode,
121.
122.             'tugas_status' => $tugas_status
123.
124.         );
125.         $this->session-
126.             >set_flashdata('tambah', 'Tugas praktikum berhasil ditambahkan');
127.         $this->M_database->insertData('t_tugas', $data);
128.
129.         redirect("asisten/tugas?idp=$jadwal_kode", 'refresh');
130.
131.     }
132.
133.     function editTugas() {
134.         $tugas_id = $this->input->post('tugas_id');
135.         $tugas_kode = $this->input->post('tugas_kode');
136.         $tugas_judul = $this->input->post('tugas_judul');
137.         $tugas_deskripsi = $this->input-
138.             >post('tugas_deskripsi');
139.         $total_file = $this->input->post('total_file');
140.         $tugas_status = $this->input-
141.             >post('tugas_status');
142.         $jadwal_kode = $this->input->post('jadwal_kode');
143.         $batas_waktu = $this->input-
144.             >post('tanggal')." ".$this->input->post('jam').":00";
145.
146.         $1 = explode(" ", $batas_waktu);
147.         $waktu = date_create_from_format("d/m/Y", $1[1]);
148.         $tanggal = date_format($waktu, 'Y-m-d');
149.         $jam = $1[2];
150.
151.         $data = array(
152.             'tugas_judul' => $tugas_judul,
153.             'tugas_deskripsi' => $tugas_deskripsi,
154.             'batas_waktu' => $tanggal." ".$jam,
155.             'tugas_status' => $tugas_status
156.         );
157.
158.         $this->session-
159.             >set_flashdata('edit', "Tugas praktikum <b>$tugas_kode</b> berhasil diedit");

```

```

156.      $this->M_database-
    >updateData('t_tugas', array('tugas_id' => $tugas_id) , $data);
157.
158.      redirect("asisten/tugas?idp=$jadwal_kode", 'refresh');
159.
160. }
161.
162. function hapusTugas()
163. {
164.     $jadwal_kode = $this->input->post('jadwal_kode');
165.     $tugas_id = $this->input->post('tugas_id');
166.     $tugas_kode = $this->input->post('tugas_kode');
167.
168.     $tugas_image = $this->db-
        >where('tugas_kode', $tugas_kode)->get('t_tugas_image');
169.
170.     foreach ($tugas_image-
        >result() as $key => $value) {
171.         unlink('./assets/images/tugas/'.$value-
        >tugas_image);
172.     }
173.
174.     $this->M_database-
        >deleteData('t_tugas', array('tugas_id' => $tugas_id));
175.     $this->M_database-
        >deleteData('t_tugas_image', array('tugas_kode' => $tugas_kode));
176.     $this->session-
        >set_flashdata('hapus', 'Tugas praktikum berhasil dihapus');
177.
178.     redirect("asisten/tugas?idp=$jadwal_kode", 'refresh');
179. }
180.
181. function hapusFileTugas()
182. {
183.     $file = $this->input->post('file_tugas');
184.     $tugas_id = $this->input->post('tugas_id');
185.     $tugas_kode = $this->input->post('tugas_kode');
186.     $jadwal_kode = $this->input->post('jadwal_kode');
187.
188.     foreach ($file as $tugas) {
189.         $this->M_database-
            >deleteData('t_tugas_file', array('file_nama' => $tugas));
190.         unlink('assets/tugas_file/'.$tugas);

191.     }
192.
193.     $this->session-
        >set_flashdata('hapus', 'File tugas berhasil dihapus');
194.     redirect("asisten/asisten_tugas/view_penilaian?tug
        as_id=$tugas_id&tugas_kode=$tugas_kode&jadwal_kode=$jadwal_kode&id_praktikum=$jadwal_kode", 'refresh');
195. }

```

```

196.
197.     function fetch_image()
198.     {
199.         $tugas_kode = $this->input->get('tugas_kode');
200.         $image = $this->M_database-
>getWhere('t_tugas_image', null, null, array('tugas_kode' =>
$tugas_kode));
201.         foreach ($image->result() as $key => $value) {
202.             echo '<div class="img-wrapper" id="img-
wrapper-'.$value->tugas_image_id.'">
203.                 <div class="preloader-
wrapper small active hide" id="myloader-'.$value-
>tugas_image_id.'" style="position: absolute; margin: auto;
top:0;bottom:0;left:0;right:0; z-index:10">
204.                     <div class="spinner-
layer spinner-teal-only">
205.                         <div class="circle-
clipper left">
206.                             <div>
207.                             <div class="circle"></
patch">
208.                             <div>
209.                             <div class="circle-
clipper right">
210.                             <div>
211.                             </div>
212.                         </div>
213.                     </div>

214.                     tugas_image_id.'" class="responsive-
img img materialboxed">
215.                     <div id="action-foto-'.$value-
>tugas_image_id.'" style="position:absolute; bottom:0; width
:100%">
216.                         <span class="material-
icons red-text text-lighten-1 left delete-foto-
tugas" id="hapus-'.$value->tugas_image_id.'" data-
gambar="'.$value->tugas_image.'" data-id="'.$value-
>tugas_image_id.'" style="cursor: pointer;">close</span>
217.                         <span class="material-icons-
outlined amber-text text-lighten-1 right edit-foto-
tugas" data-gambar="'.$value->tugas_image.'" data-
id="'.$value->tugas_image_id.'" id="edit-'.$value-
>tugas_image_id.'" style="cursor: pointer;">edit</span>
218.                     </div>
219.                 </div>';
220.             }
221.         }
222.
223.     function uploadFoto()
224.     {

```

```

225.      $image_id = $this->input-
>post('image_id');
226.      $image_edit = $this->input-
>post('image_edit');
227.      $kode_t = $this->input-
>post('kode_t');
228.
229.      $config['upload_path'] = './assets/images/tugas/';
230.      $config['allowed_types'] = 'jpg|png|jpeg|webp';
231.      $config['file_name'] = 'tugas-img-' . time();
232.      $config['max_size'] = 2000;
233.      $config['max_width'] = 1080;
234.      $config['max_height'] = 1080;
235.      $config['overwrite'] = true;
236.
237.      $this->load->library('upload', $config);
238.
239.      if ($this->upload->do_upload('foto')) {
240.
241.          $foto = $this->upload->data();
242.          $data_foto = $foto['file_name'];
243.          $this->compressImage($data_foto);
244.
245.          if ($image_id != '-') {
246.              unlink('./assets/images/tugas/' . $image_ed-
i-
t);
247.              $this->M_database-
>updateData('t_tugas_image', array('tugas_image_id' => $im-
age_id), array('tugas_image' => $data_foto));
248.              echo "edit|$image_id|" . base_url() . "assets/
images/tugas/$data_foto|$data_foto";
249.          } else {
250.              $this->M_database-
>insertData('t_tugas_image', array('tugas_image' => $data_fo-
to, 'tugas_kode' => $kode_t));
251.              $id = $this->db->insert_id();
252.              echo 'tambah|<div class="img-
wrapper" id="img-wrapper-' . $id . '">
253.                  <div class="preloader-
wrapper small active hide" id="myloader-
' . $id . '" style="position: absolute; margin: auto; top:0;bott-
om:0;left:0;right:0; z-index:10">
254.                      <div class="spinner-
layer spinner-teal-only">
255.                          <div class="circle-
clipper left">
256.                          <div class="circle-
clipper right">
257.                              <div class="gap-
patch">
258.                          <div class="circle-
clipper right">
259.                              <div class="circle-
le-clipper right">
260.                          <div class="circle-
"></div>
```

```

261.                                     </div>
262.                                 </div>
263.                               </div>

264.                               
265.                               <div id="action-foto-
   '.$id.'" style="position:absolute; bottom:0; width:100%">
266.                                   <span class="material-
   icons red-text text-lighten-1 left delete-foto-
   tugas" id="hapus-".$id.'" data-gambar="'.$data_foto.'" data-
   id="'.$id.'" style="cursor: pointer;">close</span>
267.                                   <span class="material-
   icons-outlined amber-text text-lighten-1 right edit-foto-
   tugas" data-gambar="'.$data_foto.'" data-
   id="'.$id.'" id="edit-
   '.$id.'" style="cursor: pointer;">edit</span>
268.                               </div>
269.                             </div>';
270.                         }
271.                     } else {
272.                         echo "gagal|$image_id";
273.                     }
274.
275.                 }
276.
277.             function hapusFoto()
278.             {
279.                 $id = $this->input->post('id');
280.                 $gambar = $this->input->post('gambar');
281.                 $this->M_database-
                   >deleteData('t_tugas_image', array('tugas_image_id' => $id));
282.                 if ($this->db->affected_rows() > 0) {
283.                     unlink('./assets/images/tugas/'.$gambar);
284.                     echo "berhasil";
285.                 } else {
286.                     echo "gagal";
287.                 }
288.             }
289.
290.             function compressImage($path)
291.             {
292.                 $config['image_library']='gd2';
293.                 $config['source_image']= './assets/images/tugas/'.$
   path;
294.                 $config['create_thumb']= FALSE;
295.                 $config['maintain_ratio']= FALSE;
296.                 $config['quality']= '50%';
297.                 $config['new_image']= './assets/images/tugas/'.$pa
   th;
298.                 $this->load->library('image_lib', $config);
299.                 $this->image_lib->resize();
300.             }

```

```

301.
302.     function kirimNotifikasi($tugas_kode)
303.     {
304.         // $judul = $this->input->post('judul');
305.         $url = 'http://localhost/clientsilabtif/tugas';
306.
307.         $token = $this->db->get('t_device_token')-
308.             >result();
309.         $notifications= [];
310.
311.         foreach ($token as $key => $value) {
312.             array_push($notifications,
313.                         [
314.                             'subscription' => Subscription::create
315.                                 (
316.                                     [
317.                                         'endpoint' => $value->endpoint,
318.                                         'keys' => [
319.                                             'p256dh' => $value->p256dh,
320.                                             'auth' => $value->auth
321.                                         ]
322.                                     ],
323.                                     [
324.                                         'payload' => "Tugas Praktikum|Tugas pr
325.                 aktikum baru $tugas_kode. Yuk kerjakan sekarang!|$url",
326.                                         ]
327.                                     );
328.                                 );
329.             $auth = array(
330.                 'VAPID' => array(
331.                     'subject' => '*****',
332.                     'publicKey' => '*****',
333.                     'privateKey' => '*****' // in the rea
334.                         l world, this would be in a secret file
335.                     ),
336.                 );
337.             );
338.             $webPush = new WebPush($auth);
339.
340.             // send multiple notifications with payload
341.             foreach ($notifications as $notification) {
342.                 $webPush->sendNotification(
343.                     $notification['subscription'],
344.                     $notification['payload'] // optional (defa
345.                         ults null)
346.                     );
347.                 }
348.             }
349.             /**
350.             * Check sent results
351.             * @var MessageSentReport $report
352.             */
353.             foreach ($webPush->flush() as $report) {
354.                 $endpoint = $report->getRequest()->getUri()-
355.                     >__toString();
356.                 // if ($report->isSuccess()) {

```

```

351.           // echo "[v] Message sent successfully for su
   bscription {$endpoint}.";
352.           // } else {
353.           // echo "[x] Message failed to sent for subsc
   ription {$endpoint}: {$report->getReason()}";
354.           //
355.       }
356.
357.       /**
358.        * send one notification and flush directly
359.        * @var \Generator<MessageSentReport> $sent
360.        */
361.       $sent = $webPush->sendNotification(
362.           $notifications[0]['subscription'],
363.           $notifications[0]['payload'], // optional (def
   aults null)
364.           true // optional (defaults false)
365.       );
366.
367.       // redirect('admin/pengumuman', 'refresh');
368.   }
369.
370. }

```

## Chat.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Chat extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $asisten_id = $this->session-
   >userdata('asisten_id');
16.        $data['jml_notif'] = $this->M_database-
   >countWhere('t_pesan_status', array('pesan_penerima' => "asi
   sten_$asisten_id"));
17.        $data['asisten'] = $this->M_database-
   >getWhere('t_asisten', 't_jabatan', 't_asisten.jabatan_id=t_

```

```

    jabatan.jabatan_id', array('status' => 1, 't_asisten.jabatan
    _id !=' => 12), 'asisten_nama ASC');
18.          $data['dosen'] = $this->M_database-
    >getWhere('t_dosen', 't_jabatan', 't_dosen.jabatan_id=t_jaba
    tan.jabatan_id', array('dosen_status' => 1), 'dosen_nama ASC
    ');
19.          $data['bak'] = $this->M_database-
    >getWhere('t_asisten', 't_jabatan', 't_asisten.jabatan_id=t_
    jabatan.jabatan_id', array('status' => 1, 't_asisten.jabatan_
    _id' => 12), 'asisten_nama ASC');
20.
21.          $this->template->load('asisten/template-
    asisten', 'asisten/v-chat', $data);
22.
23.      }
24.
25.      public function kirim_pesan()
26.      {
27.          $jabatan_id = $this->session-
    >userdata('jabatan_id');
28.          $asisten_id = $this->session-
    >userdata('asisten_id');
29.          $pesan = $this->input->post('pesan');
30.          $kanal = $this->input->post('kanal');
31.          $penerima = $this->input->post('penerima');
32.          $pengirim = $this->input->post('pengirim');
33.          $jabatan = "";
34.          if ($jabatan_id == 1 || $jabatan_id == 13) {
35.              $jabatan = "dosen";
36.          } else {
37.              $jabatan = "asisten";
38.          }
39.
40.          $data = array(
41.              'pesan_isi' => $pesan,
42.              'pesan_user' => $jabatan.'_'.$asisten_id.'-
    '.$penerima,
43.              'pesan_nama_pengirim' => $pengirim,
44.              'pesan_kanal' => $kanal
45.          );
46.
47.          $this->M_database->insertData('t_pesan', $data);
48.          $last_id = $this->db->insert_id();
49.
50.          $read = "";
51.          $mt = "";
52.          if ($kanal == '-') {
53.              $read = '<i class="material-icons right grey-
    text text-lighten-1" style="font-size:20px">done_all</i>';
54.              $mt = "margin-top:4px";
55.              $data2 = array(
56.                  'pesan_id' => $last_id,
57.                  'pesan_pengirim' => $jabatan.'_'.$asisten_
    id,
58.                  'pesan_penerima' => $penerima
59.              );

```

```

60.           $this->M_database-
61.           >insertData('t_pesan_status', $data2);
62.       } else {
63.           $asisten = $this->db->select('asisten_id')-
64.           >where(array('asisten_id !=' => $asisten_id, 'status' => 1,
65.           'jabatan_id !=' => 12))->get('t_asisten');
66.           $read = '<i class="material-icons right grey-
67.           text text-lighten-1" style="font-size:20px; margin-
68.           top:4px">done_all</i>';
69.           $mt = "margin-top:8px";
70.           foreach ($asisten-
71.           >result() as $key => $value) {
72.               $data2 = array(
73.                   'pesan_id' => $last_id,
74.                   'pesan_penerima' => 'asisten_'. $value-
75.                   >asisten_id,
76.                   'pesan_pengirim' => $kanal
77.               );
78.               $this->M_database-
79.               >insertData('t_pesan_status', $data2);
80.           }
81.       }
82.   }
83.
84.   echo "$penerima|$pesan|". '<div class="chat-text-
85.   right"><span class="pesan">'.$pesan.'<br>'.$read.'<small sty-
86.   le="'. $mt .'" class="right grey-text text-lighten-
87.   1">'.date('Y-m-d H:i:s').'</small></span></div>';
88.
89.   $options = array(
90.       'cluster' => 'ap1',
91.       'useTLS' => true
92.   );
93.   $pusher = new Pusher\Pusher(
94.       'bf535588d7d38b92fa01',
95.       '0477973f39c9a381253d',
96.       '1029875',
97.       $options
98.   );
99.   $data3['message'] = 'hello world';
100.  $pusher->trigger('my-channel', 'my-
101.  event', $data3);
102. }
```

```

100.
101.     function fetch_pesanan()
102.     {
103.         $jabatan_id = $this->session-
104.             >userdata('jabatan_id');
105.         $asisten_id = $this->session-
106.             >userdata('asisten_id');
107.         $jabatan = "";
108.         if ($jabatan_id == 1 || $jabatan_id == 13) {
109.             $jabatan = "dosen";
110.         } else {
111.             $jabatan = "asisten";
112.         }
113.         $pengirim = $jabatan.'_'.$this->session-
114.             >userdata('asisten_id');
115.         $penerima = $this->input->get('penerima');
116.         $user1 = $pengirim.'-'.$penerima;
117.         $user2 = $penerima.'-'.$pengirim;
118.         $pesan = $this->M_database-
119.             >getPesan($user1, $user2);
120.         $notif = $this->db->order_by('id ASC')-
121.             >where('pesan_penerima', "asisten_".$asisten_id)-
122.             >join('t_pesanan', 't_pesanan_status.pesan_id=t_pesanan.pesan_id')-
123.             >get('t_pesanan_status');
124.         $user_in = "";
125.         $pesan_in = "";
126.         $jumlah = "";
127.         if ($notif) {
128.             foreach ($notif->result() as $key => $value) {
129.                 $jml_pesan = $this->M_database-
130.                     >countWhere('t_pesanan_status', array('pesan_pengirim' => $value-
131.                         >penerima, 'pesan_penerima' => "$pengirim"));
132.                 echo "$msg|".<a class="hide" href="#end-
133.                     content"><button class="to-
134.                         bottom></button></a>";
135.                 foreach ($pesan as $key => $value) {
136.                     $pengirim_pesan = explode('-', $value-
137.                         >pesan_user);
138.                     $pos = "";
139.                     $style = "";
140.                     $read = "";
141.                     $pesan_status = $this->M_database-
142.                         >countWhere('t_pesanan_status', array('pesan_id' => $value-
143.                             >pesan_id, 'pesan_penerima' => $penerima));
144.                     if ($pengirim_pesan[0] == $pengirim) {
145.                         $msg .= $user_in.$pesan_in.$jumlah;
146.                     } else {
147.                         $msg .= "0|0|0";
148.                     }
149.                 }
150.             }
151.         }
152.     }
153. 
```

```

140.           $pos = "chat-text-right";
141.           $color = "grey-text text-lighten-2";
142.           if ($pesan_status > 0 ) {
143.               $read = '<i class="material-
144. icons right grey-text text-lighten-1" style="font-
145. size:20px">done_all</i>';
146.           } else {
147.               $read = '<i class="material-
148. icons right light-blue-text text-lighten-2" style="font-
149. size:20px">done_all</i>';
150.           }
151.       }
152.
153.           echo '<div class="'. $pos .' '. $pengirim_pesan[0
154. ].'" style="'. $style .'">'><span class="pesan">' . $value-
155. >pesan_isi.'<br>' . $read. '<small style="margin-
156. top:4px" class="right ' . $color .'">' . $value-
157. >date_created.'</small></span></div>';
158.       }
159.   function fetch_pesanan_grup()
160.   {
161.       $asisten_id = $this->session-
162. >userdata('asisten_id');
163.       $jabatan_id = $this->session-
164. >userdata('jabatan_id');
165.       $jabatan = "";
166.       if ($jabatan_id == 1 || $jabatan_id == 13) {
167.           $jabatan = "dosen";
168.       } else {
169.           $jabatan = "asisten";
170.       }
171.       $pengirim = $jabatan. '_' . $this->session-
172. >userdata('asisten_id');
173.       $kanal = $this->input->get('kanal');
174.       $pesan = $this->M_database->getPesanGrup($kanal);
175.       $last_msg = $this->db->order_by('id DESC')-
176. >limit(1)->where('pesan_penerima', "asisten_$asisten_id")-
177. >join('t_pesanan', 't_pesanan_status.pesan_id=t_pesanan.pesan_id')-
178. >get('t_pesanan_status')->row();
179.       if ($last_msg) {
180.           $jml_pesan = $this->M_database-
181. >countWhere('t_pesanan_status', array('pesan_pengirim' => $last_
182. msg->pesan_pengirim, 'pesan_penerima' => "$pengirim"));
183.           $msg = "$last_msg->pesan_pengirim|$last_msg-
184. >pesan_isi|$jml_pesan";
185.       } else {
186.           $msg = "0|0|0";
187.       }
188.   }

```

```

179.      $this->M_database-
    >deleteData('t_pesanan_status', array('pesan_pengirim' => $kan-
        al, 'pesan_penerima' => $pengirim));
180.      echo "$msg|".<a class="hide" href="#end-
    content"><button class=""to-
    bottom></button></a>';
181.      foreach ($pesan as $key => $value) {
182.          $pengirim_pesan = explode('-', $value-
            >pesan_user);
183.          $pos = "";
184.          $style = "";
185.          $read = "";
186.          $hide = "";
187.          $pesan_status = $this->M_database-
            >countWhere('t_pesanan_status', array('pesan_pengirim' => $kan-
                al, 'pesan_penerima !=' => "$pengirim"));
188.          if ($pengirim_pesan[0] == $pengirim) {
189.              $hide = 'hide';
190.              $pos = "chat-text-right";
191.              $color = "grey-text text-lighten-2";
192.              if ($pesan_status > 0 ) {
193.                  $read = '<i class="material-
                    icons right grey-text text-lighten-1" style="font-
                    size:20px; margin-top:4px">done_all</i>';
194.              } else {
195.                  $read = '<i class="material-
                    icons right light-blue-text text-lighten-2" style="font-
                    size:20px; margin-top:4px">done_all</i>';
196.              }
197.          } else {
198.              $hide = '';
199.              $read = "";
200.              $pos = "chat-text-left";
201.              $color = "grey-text text-darken-1";
202.          }
203.
204.          echo '<div class="'. $pos .' '. $pengirim_pesan[0
            ].'" style="'. $style .'"><span class="pesan"><div class="pink-
            text '$.hide.'" style="font-weight:600; font-
            size: 12px">'$.value->pesan_nama_pengirim.'</div>'$.value-
            >pesan_isi.'<br>'$.read.'<small style="margin-
            top:8px" class="right '$.color.'">'$.value-
            >date_created.'</small></span></div>';
205.      }
206.      echo '<div id="end-content"></div>';
207.  }
208.
209.  function cek_pesanan()
210.  {
211.      $asisten_id = $this->session-
        >userdata('asisten_id');
212.      $jabatan_id = $this->session-
        >userdata('jabatan_id');
213.      $jabatan = "";
214.      if ($jabatan_id == 1 || $jabatan_id == 13) {
215.          $jabatan = "dosen";

```

```

216.         } else {
217.             $jabatan = "asisten";
218.         }
219.         $last_msg = $this->db->order_by('id DESC')-
>limit(1)->where('pesan_penerima', "asisten_$asisten_id")-
>join('t_pesanan', 't_pesanan_status.pesan_id=t_pesanan.pesan_id')-
->get('t_pesanan_status')->row();
220.         $user = $jabatan.'_'.$asisten_id;
221.         $pesan_d = "";
222.         if ($last_msg) {
223.             $dpesan = $this->db->order_by('id DESC')-
>group_by('pesan_pengirim')->limit(4)-
>where('pesan_penerima', "asisten_$asisten_id")-
>join('t_pesanan', 't_pesanan_status.pesan_id=t_pesanan.pesan_id')-
->get('t_pesanan_status');
224.             foreach ($dpesan-
>result() as $key => $value) {
225.                 if ($value->pesan_pengirim == 'asisten') {
226.                     $jumlah_d = $this->M_database-
>countWhere('t_pesanan_status', array('pesan_penerima' => $use
r, 'pesan_pengirim' => 'asisten'));
227.                     $ini_pengirim = 'Asisten';
228.                     $ini_foto = 'icons/icon-96x96.png';
229.                 } else if ($value-
>pesan_pengirim == 'koorlab') {
230.                     $jumlah_d = $this->M_database-
>countWhere('t_pesanan_status', array('pesan_penerima' => $use
r, 'pesan_pengirim' => 'koorlab'));
231.                     $ini_pengirim = 'Koordinator Lab';
232.                     $ini_foto = 'icons/icon-96x96.png';
233.                 } else {
234.                     $jumlah_d = $this->M_database-
>countWhere('t_pesanan_status', array('pesan_penerima' => $use
r, 'pesan_pengirim' => $value->pesan_pengirim));
235.                     $pengirim_d = explode('_', $value-
>pesan_pengirim);
236.                     if ($pengirim_d[0] == 'asisten') {
237.                         $u_user = $this->db-
>select('foto')->where('asisten_id', $pengirim_d[1])-
>get('t_asisten')->row();
238.                     } else {
239.                         $u_user = $this->db-
>select('foto')->where('dosen_id', $pengirim_d[1])-
>get('t_dosen')->row();
240.                     }
241.                     $ini_pengirim = $value-
>pesan_nama_pengirim;
242.                     $ini_foto = "profil/$u_user->foto";
243.                 }
244.                 $pesan_d .= '
245.                     <div class="p-pesan">
246.                         
247.                         <div style="margin-
left:16px; width:100%">

```

```

248.                                     <a class="p-link" data-
    user="'. $value-
    >pesan_pengirim.'" href="'.base_url().'asisten/chat?'. $value
    ->pesan_pengirim.'">
249.                                         <span class="p-
    pengirim">'.$ini_pengirim.'</span>
250.                                         <span class="p-jumlah-
    pesan">'.$jumlah_d.' pesan masuk</span>
251.                                         </a>
252.                                         </div>
253.                                         </div>';
254.                                         }
255.             $total_pesan = $this->M_database-
    >countWhere('t_pesan_status', array('pesan_penerima' => $use
    r));
256.             if ($last_msg->pesan_kanal == 'asisten') {
257.                 $pesan = $this->M_database-
    >countWhere('t_pesan_status', array('pesan_penerima' => $use
    r, 'pesan_pengirim' => 'asisten'));
258.                 echo "$total_pesan|$pesan|Asisten|".base_u
    rl()."assets/images/icons/icon-
    96x96.png|".base_url()."asisten/chat?$last_msg-
    >pesan_pengirim|$pesan_d|$last_msg->pesan_pengirim";
259.             } else if ($last_msg-
    >pesan_kanal == 'koorlab') {
260.                 $pesan = $this->M_database-
    >countWhere('t_pesan_status', array('pesan_penerima' => $use
    r, 'pesan_pengirim' => 'koorlab'));
261.                 echo "$total_pesan|$pesan|Koordinator Lab|"
    ".base_url()."assets/images/icons/icon-
    96x96.png|".base_url()."asisten/chat?$last_msg-
    >pesan_pengirim|$pesan_d|$last_msg->pesan_pengirim";
262.             } else {
263.                 $pesan = $this->M_database-
    >countWhere('t_pesan_status', array('pesan_penerima' => $use
    r, 'pesan_pengirim' => $last_msg->pesan_pengirim));
264.                 $pengirim = explode('_', $last_msg-
    >pesan_pengirim);
265.                 if ($pengirim[0] == 'asisten') {
266.                     $u_user = $this->db->select('foto')-
    >where('asisten_id', $pengirim[1])->get('t_asisten')->row();
267.                 } else {
268.                     $u_user = $this->db->select('foto')-
    >where('dosen_id', $pengirim[1])->get('t_dosen')->row();
269.                 }
270.                 echo "$total_pesan|$pesan|$last_msg-
    >pesan_nama_pengirim|".base_url()."assets/images/profil/$u_u
    ser->foto|".base_url()."asisten/chat?$last_msg-
    >pesan_pengirim|$pesan_d|$last_msg->pesan_pengirim";
271.             }
272.         } else {
273.             echo "0|0|none|none|none|Tidak ada pesan|user"
    ;
274.         }
275.     }
276.

```

```

277.     function cek_notifikasi()
278.     {
279.         $asisten_id = $this->session-
280.             >userdata('asisten_id');
281.         $last_msg = $this->db->order_by('id DESC')-
282.             >limit(1)->where('pesan_penerima', "asisten_$asisten_id")-
283.             >join('t_pesan', 't_pesan_status.pesan_id=t_pesan.pesan_id')-
284.             >get('t_pesan_status')->row();
285.         if ($last_msg) {
286.             $jml_pesan = $this->M_database-
287.                 >countWhere('t_pesan_status', array('pesan_pengirim' => $las-
288.                     t_msg-
289.                     >pesan_pengirim, 'pesan_penerima' => "asisten_$asisten_id"))
290.         ;
291.     }

```

### Kritik\_saran.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Kritik_saran extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.
16.        $data['krisar'] = $this->M_database-
17.            >getAllData('t_kritik_saran', null, null, 'date_created DESC
   ');
18.        $this->template->load('asisten/template-
   asisten', 'asisten/v-kritik-saran', $data);
19.    }

```

```

20.
21.     function statusKrisar()
22.     {
23.         $krisar_id = $this->input->post('krisar_id');
24.         $status = $this->input->post('status');
25.
26.         if ($status == 'Diterima') {
27.             $data = array('ks_status' => 0);
28.             $badge = '<span class="badge-status badge-
not">Belum</span>';
29.         } else {
30.             $data = array('ks_status' => 1);
31.             $badge= '<span class="badge-status badge-
ok">Diterima</span>';
32.         }
33.
34.         $this->M_database-
            >updateData('t_kritik_saran', array('ks_id' => $krisar_id),
            $data);
35.
36.         echo $badge;
37.     }
38.
39. }
```

## Modul\_praktikum.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Modul_praktikum extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $asisten_id = $this->session-
           >userdata('asisten_id');
16.        $jadwal_kode = $this->input->get('id_praktikum');
17.
18.        $data['jadwalkosong'] = $this->input-
           >get('id_praktikum');
19.        $data['mahasiswa'] = $this->M_database-
           >getWhere('registrasi_praktikum', 't_mahasiswa', 'registrasi')
```

```

1. _praktikum.npm = t_mahasiswa.npm', array('jadwal_kode' => $j
adwal_kode), 'registrasi_praktikum.npm DESC');
20.         $data['jadwal'] = $this->M_database-
>getJadwalAsisten("t_jadwal.asisten_1 = $asisten_id OR t_jad
wal.asisten_2 = $asisten_id");
21.         // data hide table
22.         $data['visibility'] = '';
23.
24.         if ($jadwal_kode == null) {
25.             $data['visibility'] = 'hide';
26.         } else {
27.             $data['visibility'] = '';
28.         }
29.
30.         $this->template->load('asisten/template-
asisten', 'asisten/v-modul', $data);
31.
32.     }
33.
34.     function statusModul()
35.     {
36.         $registrasi_id = $this->input-
>post('registrasi_id');
37.         $status = $this->input->post('status');
38.
39.         if ($status == 'Sudah') {
40.             $data = array('status_modul' => 0, 'tgl_modul'
=> '-');
41.             $tgl = '-';
42.             $badge = '<span class="badge-status badge-
not">Belum</span>';
43.         } else {
44.             $data = array('status_modul' => 1, 'tgl_modul'
=> date('Y-m-d'));
45.             $tgl = date('Y-m-d');
46.             $badge= '<span class="badge-status badge-
ok">Sudah</span>';
47.         }
48.
49.         $this->M_database-
>updateData('registrasi_praktikum', array('registrasi_id' =>
$registrasi_id), $data);
50.
51.         echo "$badge|$tgl";
52.     }
53.
54. }

```

## Profil.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.

```

```

4. class Profil extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.
14.    function index()
15.    {
16.        $asisten_id = $this->session-
>userdata('asisten_id');
17.        $asisten = $this->M_database-
>getWhere('t_asisten', 't_jabatan', 't_asisten.jabatan_id =
t_jabatan.jabatan_id', array('asisten_id' => $asisten_id))->row();
18.        $data['jabatan'] = $this->M_database-
>getAllData('t_jabatan', null, null, null);
19.
20.        $data['asisten_id'] = $asisten->asisten_id;
21.        $data['username'] = $asisten->username;
22.        $data['asisten_nama'] = $asisten->asisten_nama;
23.        $data['jabatan_id'] = $asisten->jabatan_id;
24.        $data['jabatan_nama'] = $asisten->jabatan_nama;
25.        $format = date_create($asisten->tgl_lahir);
26.        $date = date_format($format, 'm');
27.        $bulan = 'Pilih Bulan';
28.        switch ($date) {
29.            case '01':
30.                $bulan = 'Januari';
31.                break;
32.            case '02':
33.                $bulan = 'Februari';
34.                break;
35.            case '03':
36.                $bulan = 'Maret';
37.                break;
38.            case '04':
39.                $bulan = 'April';
40.                break;
41.            case '05':
42.                $bulan = 'Mei';
43.                break;
44.            case '06':
45.                $bulan = 'Juni';
46.                break;
47.            case '07':
48.                $bulan = 'Juli';
49.                break;
50.            case '08':
51.                $bulan = 'Agustus';
52.                break;
53.            case '09':
54.                $bulan = 'September';

```

```

55.          break;
56.      case '10':
57.          $bulan = 'Oktober';
58.          break;
59.      case '11':
60.          $bulan = 'November';
61.          break;
62.      case '12':
63.          $bulan = 'Desember';
64.          break;
65.      default:
66.          $bulan = 'Pilih Bulan';
67.          break;
68.      }
69.     $data['bulan'] = $bulan;
70.     $tgl_lahir = explode('-', $asisten->tgl_lahir);
71.     $data['tanggal'] = $tgl_lahir[2];
72.     $data['n_bulan'] = $tgl_lahir[1];
73.     $data['tahun'] = $tgl_lahir[0];
74.     $data['alamat'] = $asisten->alamat;
75.     $data['email'] = $asisten->email;
76.     $data['tahun_masuk'] = $asisten->thn_masuk;
77.     $data['tahun_keluar'] = $asisten->thn_keluar;
78.     $data['foto'] = $asisten->foto;
79.
80.     $this->template->load('asisten/template-
    asisten', 'asisten/v-profil', $data);
81.   }
82.
83.   function editProfil() {
84.
85.     $asisten_id = $this->session-
    >userdata('asisten_id');
86.     $asisten_nama = $this->input-
    >post('asisten_nama');
87.     $tgl_lahir = $this->input->post('tgl_lahir');
88.     $bulan_lahir = $this->input->post('bulan_lahir');
89.     $tahun_lahir = $this->input->post('tahun_lahir');
90.     $alamat = $this->input->post('alamat');
91.     $email = $this->input->post('email');
92.
93.     $data = array(
94.       'asisten_nama' => $asisten_nama,
95.       'tgl_lahir' => $tahun_lahir . '-'
    . $bulan_lahir . '-' . $tgl_lahir,
96.       'email' => $email,
97.       'alamat' => $alamat
98.     );
99.
100.    $this->session-
    >set_flashdata('edit', 'Profil berhasil diedit');
101.    $this->M_database-
    >updateData('t_asisten', array('asisten_id' => $asisten_id),
    $data);
102.
103.    redirect("asisten/profil", 'refresh');

```

```

104.
105.    }
106.
107.    function uploadFoto()
108.    {
109.        $asisten_id = $this->session-
>userdata('asisten_id');
110.        $foto_nama = $this->input-
>post('foto_nama');
111.
112.        $config['upload_path'] = './assets/images/profil/'
;
113.        $config['allowed_types'] = 'jpg|png|jpeg|webp';
114.        $config['file_name'] = 'asisten'.time();
115.        $config['max_size'] = 2000;
116.        $config['max_width'] = 1080;
117.        $config['max_height'] = 1080;
118.        $config['overwrite'] = true;
119.
120.        $this->load->library('upload', $config);
121.
122.        if ($this->upload->do_upload('foto')) {
123.
124.            if ($foto_nama == 'default-profil-
1.jpg' || $foto_nama == 'default-profil-p.jpg') {
125.                $foto = $this->upload-
>data();
126.            } else {
127.                unlink('./assets/images/profil/'.$foto_nam
a);
128.            $foto = $this->upload->data();
129.        }
130.
131.        $data_foto = $foto['file_name'];
132.        $this->compressImage($data_foto);
133.        echo "$data_foto";
134.    } else {
135.        $data_foto = $foto_nama;
136.        echo "gagal";
137.    }
138.
139.    $this->M_database-
>updateData('t_asisten', array('asisten_id' => $asisten_id),
array('foto' => $data_foto));
140.
141.    }
142.
143.    function view_password()
144.    {
145.        $this->template->load('asisten/template-
asisten', 'asisten/v-password');
146.    }
147.
148.    function gantiPassword()
149.    {

```

```

150.      $asisten_id = $this->session-
>userdata('asisten_id');
151.      $old_pass = $this->input->post('old_pass');
152.      $password = $this->input->post('password');
153.      $asisten = $this->db->select('password')-
>where('asisten_id', $asisten_id)->get('t_asisten')->row();
154.
155.      if (password_verify($old_pass, $asisten-
>password)) {
156.          $data = array('password' => password_hash($pas-
sword, PASSWORD_DEFAULT));
157.          $this->M_database-
>updateData('t_asisten', array('asisten_id' => $asisten_id),
$data);
158.          $this->session->sess_destroy();
159.          echo 'sukses|'.base_url().'auth';
160.      } else {
161.          echo 'gagal|asisten';
162.      }
163.  }
164.
165.  function compressImage($path)
166.  {
167.      $config['image_library']='gd2';
168.      $config['source_image']= './assets/images/profil/' .
$path;
169.      $config['create_thumb']= FALSE;
170.      $config['maintain_ratio']= FALSE;
171.      $config['quality']= '50%';
172.      $config['width']= 300;
173.      $config['height']= 300;
174.      $config['new_image']= './assets/images/profil/'.$p-
ath;
175.      $this->load->library('image_lib', $config);
176.      $this->image_lib->resize();
177.  }
178.
179. }

```

### Nilai\_praktikum.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Nilai_praktikum extends CI_Controller {
5.
6.     function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_dosen();
11.    }
12.

```

```

13.     public function index()
14.     {
15.         $dosen_id = $this->session->userdata('dosen_id');
16.         $matkum_id = $this->input->get('mp');
17.         $kelas_kode = $this->input->get('kelas');
18.
19.         $data['jadwalkosong'] = $this->input->get('mp');
20.         $data['kelas'] = $this->M_database-
>getNilaiPraktikum(array('matkum_id' => $matkum_id, 'kelas_kode' => $kelas_kode));
21.         $data['jadwal'] = $this->M_database-
>getMatkumDosen("t_jadwal.dosen_id = $dosen_id");
22.         $data['kls'] = $this->M_database-
>getKelasDosen("t_jadwal.dosen_id = $dosen_id");
23.         // data hide table
24.         $data['visibility'] = '';
25.
26.         if ($matkum_id == null) {
27.             $data['visibility'] = 'hide';
28.         } else {
29.             $data['visibility'] = '';
30.         }
31.
32.         $this->template->load('dosen/template-dosen', 'dosen/v-penilaian', $data);
33.
34.     }
35.
36. }

```

### Rest\_absensi.php

```

1. <?php
2. use Restserver\Libraries\REST_Controller;
3. defined('BASEPATH') OR exit('No direct script access allowed');
4.
5. require APPPATH . 'libraries/REST_Controller.php';
6. require APPPATH . 'libraries/Format.php';
7.
8. class Rest_absensi extends CI_Controller {
9.
10.     use REST_Controller {
11.         REST_Controller::__construct as private __resTraitConstruct;
12.     }
13.
14.     function __construct()
15.     {
16.         // Construct the parent class
17.         // parent::__construct();
18.         $this->__resTraitConstruct();
19.         $this->load->model('M_database_API');
20.

```

```

21.      }
22.
23.      public function index_get()
24.      {
25.
26.          $npm = $this->get('npm');
27.          $jadwal_kode = $this->get('jadwal_kode');
28.
29.          $absen = $this->M_database_API-
>getAbsensi('t_absensi', array('npm' => $npm, 't_absensi.jad
wal_kode' => $jadwal_kode), 'pertemuan ASC');
30.          $matkum = $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_prakt
ikum.matkum_id')->where('jadwal_kode', $jadwal_kode)-
>get('t_jadwal')->row();
31.          $pertemuan = $this->db-
>where('jadwal_kode', $jadwal_kode)->limit(1)-
>order_by('pertemuan DESC')->get('t_absensi')->row();
32.
33.          $grup = $this->db-
>where('jadwal_kode', $jadwal_kode)->group_by('absen_tgl')-
>get('t_absensi')->result_array();
34.
35.          if ($absen != null) {
36.
37.              $arr = [];
38.              $i = 1;
39.              $j = 0;
40.
41.              $tgl_absen = $absen[0]['absen_tgl'];
42.
43.              while ($i <= $pertemuan->pertemuan) {
44.                  array_push($arr, ['pertemuan' => $i, 'abse
n_tgl' => $tgl_absen, 'kehadiran' => 'tidak hadir']);
45.                  $i++;
46.              }
47.
48.              foreach ($absen as $key => $value) {
49.                  // cek absensi
50.                  $cek_absen = $this->M_database_API-
>cekdataAbsenMhs('t_absensi', 'npm', 'jadwal_kode', 'absen_t
gl', $npm, $jadwal_kode, $value['absen_tgl']);
51.
52.                  $kehadiran = '';
53.
54.                  if ($cek_absen) {
55.                      $arr[$j]['kehadiran'] = 'hadir';
56.
57.                  }
58.                  else
59.                  {
60.                      $arr[$j]['kehadiran'] = 'tidak hadir';
61.                  }

```

```

62.                 $j++;
63.             }
64.
65.             foreach ($grup as $key => $tgl) {
66.                 $cek_absen = $this->M_database_API-
>cekdataAbsenMhs('t_absensi', 'npm', 'jadwal_kode', 'absen_t
gl', $npm, $jadwal_kode, $tgl['absen_tgl']);
67.                 $arr[$key]['absen_tgl'] = $tgl['absen_tgl'
];
68.
69.                 if ($cek_absen)
70.                 {
71.                     $arr[$key]['kehadiran'] = 'hadir';
72.                 }
73.                 else
74.                 {
75.                     $arr[$key]['kehadiran'] = 'tidak hadir
';
76.                 }
77.             }
78.
79.         }
80.
81.
82.         if ($absen) {
83.             $this->response([
84.                 'status' => true,
85.                 'npm' => $npm,
86.                 'jadwal_kode' => $matkum->jadwal_kode,
87.                 'matkum' => $matkum->matkum,
88.                 'absensi' => $arr
89.             ], 200);
90.         } else {
91.             $this->response([
92.                 'status' => false,
93.                 'npm' => $npm,
94.                 'jadwal_kode' => $matkum-
>jadwal_kode,
95.                 'matkum' => $matkum->matkum,
96.                 'absensi' => $absen
97.             ], 200);
98.         }
99.     }
100.
101. }

```

### Rest info.php

```

1. <?php
2. use Restserver\Libraries\REST_Controller;
3. defined('BASEPATH') OR exit('No direct script access allowed
');
4.

```

```

5. require APPPATH . 'libraries/REST_Controller.php';
6. require APPPATH . 'libraries/Format.php';
7.
8. class Rest_info extends CI_Controller {
9.
10.    use REST_Controller;
11.    REST_Controller::__construct as private __resTrait
12.    Construct;
13. }
14. function __construct()
15. {
16.     // Construct the parent class
17.     // parent::__construct();
18.     $this->__resTraitConstruct();
19.     $this->load->model('M_database_API');
20.
21. }
22.
23. public function index_get()
24. {
25.
26.     $info = $this->M_database_API->getInfo()-
>result_array();
27.
28.     foreach ($info as $key => $value) {
29.         $img = [];
30.         $info_image = $this->M_database_API-
>getInfoImage(array('info_id' => $value['info_id']))-
>result_array();
31.         foreach ($info_image as $image) {
32.             array_push($img, base_url()."assets/images
/info/".$image['info_image']);
33.         }
34.         $info[$key]['info_image'] = $img;
35.     }
36.
37.     if ($info > 0) {
38.         $this->response([
39.             'status' => true,
40.             'info' => $info
41.         ], 200);
42.     } else {
43.         $this->response([
44.             'status' => false,
45.             'info' => $info
46.         ], 200);
47.     }
48. }
49. ?>
50.
51. ?>
```

## Rest\_jadwal.php

```

1. <?php
2. use Restserver\Libraries\REST_Controller;
3. defined('BASEPATH') OR exit('No direct script access allowed
   ');
4.
5. require APPPATH . 'libraries/REST_Controller.php';
6. require APPPATH . 'libraries/Format.php';
7.
8. class Rest_jadwal extends CI_Controller {
9.
10.     use REST_Controller {
11.         REST_Controller::__construct as private __resTrait
12.         Construct;
13.     }
14.     function __construct()
15.     {
16.         // Construct the parent class
17.         // parent::__construct();
18.         $this->__resTraitConstruct();
19.         $this->load->model('M_database_API');
20.         $this->load->model('M_database');
21.
22.     }
23.
24.     public function index_get()
25.     {
26.
27.         $npm = $this->get('npm');
28.
29.         $jadwal = $this->M_database_API-
>getJadwal('registrasi_praktikum', array('npm' => $npm, 't_j
adwal.status' => 1), 'jadwal_jam ASC');
30.         $pembayaran = $this->db-
>select('status_bayar_bak')->where('npm', $npm)-
>get('t_mahasiswa')->row()->status_bayar_bak;
31.
32.         $i = 0;
33.
34.         foreach ($jadwal as $key => $value) {
35.             $asisten_1 = $this->db-
>where('asisten_id', $jadwal[$i]['pengajar1'])-
>get('t_asisten')->row();
36.             $asisten_2 = $this->db-
>where('asisten_id', $jadwal[$i]['pengajar2'])-
>get('t_asisten')->row();
37.
38.             $praktikum = $this->M_database-
>maxwhere('t_absensi', 'jadwal_kode', $value['jadwal_kode'],
'pertemuan')->pertemuan;
39.
40.             $kehadiran = $this->M_database-
>kehadiran($value['jadwal_kode'], $npm);

```

```

41.             $total_kehadiran = $kehadiran->kehadiran * 10;
42.
43.             $nama1 = explode(' ', $asisten_1-
44.                         >asisten_nama);
45.             if (sizeof($nama1) > 1) {
46.                 if ($nama1[0] == 'M' || $nama1[0] == 'M.' ||
47.                     $nama1[0] == 'Muhamad' || $nama1[0] == 'Muhammad' || $nam-
48.                     a1[0] == 'Mohamed' || $nama1[0] == 'Mohammed' || $nama1[0] =
49.                     = 'Moch' || $nama1[0] == 'Moch.' || $nama1[0] == 'Mochamad' ||
50.                     $nama1[0] == 'Mochammad' || $nama1[0] == 'Much' || $nam-
51.                     a1[0] == 'Much.' || $nama1[0] == 'Muchamad' || $nama1[0] == 'M-
52.                     uchammad') {
53.
54.                 $panggilan_1 = $nama1[1];
55.             } else {
56.                 $panggilan_1 = $nama1[0];
57.             }
58.         } else {
59.             $panggilan_1 = $asisten_1->asisten_nama;
60.         }
61.
62.         $panggilan_2 = $asisten_2->asisten_nama;
63.     }
64.
65.     $jadwal[$i]['pengajar1'] = $panggilan_1;
66.     $jadwal[$i]['pengajar2'] = $panggilan_2;
67.     $jadwal[$i]['kehadiran'] = $total_kehadiran;
68.     $jadwal[$i]['pertemuan'] = $praktikum;
69.
70.
71.
72.     $i++;
73. }
74.
75. if ($jadwal) {
76.     $this->response([
77.         'status' => true,
78.         'pembayaran' => $pembayaran,
79.         'jadwal' => $jadwal
80.     ], 200);
81. } else {
82.     $this->response([

```

```

83.                     'status' => false,
84.                     'jadwal' => $jadwal
85.                 ], 200);
86.             }
87.         }
88.
89.     public function jadwal_get()
90.     {
91.
92.         $periode_id = $this->get('periode_id');
93.         $matkum_id = $this->get('matkum_id');
94.
95.         $jadwal = $this->M_database_API-
>getAllJadwal('t_jadwal', array('periode_id' => $periode_id,
'matkum_id' => $matkum_id), 'jadwal_hari DESC');
96.
97.         if ($jadwal) {
98.             $this->response([
99.                 'status' => true,
100.                'jadwal' => $jadwal
101.            ], 200);
102.        } else {
103.            $this->response([
104.                'status' => false,
105.                'jadwal' => $jadwal
106.            ], 200);
107.        }
108.    }
109.
110. }
111.
112. ?>

```

### Rest\_mahasiswa.php

```

1. <?php
2. use Restserver\Libraries\REST_Controller;
3. defined('BASEPATH') OR exit('No direct script access allowed
');
4.
5. require APPPATH . 'libraries/REST_Controller.php';
6. require APPPATH . 'libraries/Format.php';
7.
8. class Rest_mahasiswa extends CI_Controller {
9.
10.     use REST_Controller {
11.         REST_Controller::__construct as private __resTrait
Construct;
12.     }
13.
14.     function __construct()
15.     {
16.         // Construct the parent class

```

```

17.         // parent::__construct();
18.         $this->_resTraitConstruct();
19.         $this->load->model('M_database_API');
20.
21.     }
22.
23.     // public function index_get()
24.     //
25.     //     $mhs_id = $this->get('mhs_id');
26.
27.     //     $mhs = $this->db->select('npm')-
>where('mhs_id', $mhs_id)->get('t_mahasiswa')->row();
28.
29.     //     if ($mhs) {
30.     //         $this->response([
31.     //             'status' => true,
32.     //             'npm' => $mhs
33.     //         ], 200);
34.     //     } else {
35.     //         $this->response([
36.     //             'status' => false,
37.     //             'npm' => $mhs
38.     //         ], 204);
39.     //     }
40.     //
41.
42.     public function index_get()
43.     {
44.         $mhs_id = $this->get('mhs_id');
45.
46.         $mhs = $this->db-
>get_where('t_mahasiswa', array('mhs_id' => $mhs_id))->row();
47.         $mhs_profil = $this->M_database_API-
>getMahasiswa('t_mahasiswa', array('mhs_id' => $mhs_id));
48.
49.         $nama = explode(' ', $mhs_profil->nama);
50.
51.         if (sizeof($nama) > 3) {
52.             $nama_p = $nama[0]." ".$nama[1]." ".$nama[2];
53.         } else {
54.             $nama_p = $mhs_profil->nama;
55.         }
56.
57.         // nama panggilan
58.         if (sizeof($nama) > 1) {
59.             if ($nama[0] == 'M' || $nama[0] == 'M.' || $na-
ma[0] == 'Muhamad' || $nama[0] == 'Muhammad' || $nama[0] ==
'Mohamed' || $nama[0] == 'Mohammed' || $nama[0] == 'Moch' ||
$nama[0] == 'Moch.' || $nama[0] == 'Mochamad' || $nama[0] ==
'Mochammad' || $nama[0] == 'Much' || $nama[0] == 'Much.' ||
$nama[0] == 'Muchamad' || $nama[0] == 'Muchhammad') {
60.                 $panggilan = $nama[1];
61.             } else {
62.                 $panggilan = $nama[0];
63.             }

```

```

64.         } else {
65.             $panggilan = $mhs_profil->nama;
66.         }
67.
68.         if ($mhs) {
69.             $this->response([
70.                 'status' => true,
71.                 'mahasiswa' => [
72.                     'mhs_id' => $mhs_profil->mhs_id,
73.                     'npm' => $mhs_profil->npm,
74.                     'nama' => $mhs_profil->nama,
75.                     'nama_pendek' => $nama_p,
76.                     'panggilan' => $panggilan,
77.                     'kelas' => $mhs_profil->kelas_nama
78.                 ],
79.             ], 200);
80.         } else {
81.             $this->response([
82.                 'status' => false,
83.                 'mahasiswa' => $mhs
84.             ], 204);
85.         }
86.     }
87.
88.     public function login_post()
89.     {
90.         $npm = $this->post('npm');
91.         $password = $this->post('password');
92.
93.         $mhs = $this->db-
94.             >get_where('t_mahasiswa', array('npm' => $npm))->row();
95.         $mhs_profil = $this->M_database_API-
96.             >getMahasiswa('t_mahasiswa', array('npm' => $npm));
97.
98.         $nama = explode(' ', $mhs_profil->nama);
99.
100.        if (sizeof($nama) > 3) {
101.            $nama_p = $nama[0] . " " . $nama[1] . " " . $nama[2];
102.        } else {
103.            $nama_p = $mhs_profil->nama;
104.        }
105.        // nama panggilan
106.        if (sizeof($nama) > 1) {
107.            if ($nama[0] == 'M' || $nama[0] == 'M.' || $na-
108.                ma[0] == 'Muhamad' || $nama[0] == 'Muhammad' || $nama[0] ==
109.                'Mohamed' || $nama[0] == 'Mohammed' || $nama[0] == 'Moch' ||
110.                $nama[0] == 'Moch.' || $nama[0] == 'Mochamad' || $nama[0] =
111.                = 'Mochammad' || $nama[0] == 'Much' || $nama[0] == 'Much.' |
112.                | $nama[0] == 'Muchamad' || $nama[0] == 'Muchammad') {
113.                    $panggilan = $nama[1];
114.                } else {
115.                    $panggilan = $nama[0];
116.                }
117.            } else {
118.                $panggilan = $mhs_profil->nama;

```

```

113.         }
114.
115.         if ($mhs) {
116.             if (password_verify($password, $mhs-
>password)) {
117.                 $this->response([
118.                     'login' => true,
119.                     'kode' => 1,
120.                     'message' => "login berhasil",
121.                     'mahasiswa' => [
122.                         'mhs_id' => $mhs_profil->mhs_id,
123.                         'npm' => $mhs_profil->npm,
124.                         'nama' => $mhs_profil->nama,
125.                         'nama_pendek' => $nama_p,
126.                         'panggilan' => $panggilan,
127.                         'kelas' => $mhs_profil-
>kelas_nama
128.                         ]
129.                     ], 200);
130.             } else {
131.                 $this->response([
132.                     'login' => false,
133.                     'kode' => 2,
134.                     'message' => "password salah"
135.                 ], 200);
136.             }
137.         } else {
138.             $this->response([
139.                 'login' => false,
140.                 'kode' => 3,
141.                 'message' => "npm tidak diketahui"
142.             ], 200);
143.         }
144.     }
145.
146.     public function pass_post()
147.     {
148.         $mhs_id = $this->post('mhs_id');
149.         $old_pass = $this->post('old_pass');
150.         $new_pass = password_hash($this-
>post('new_pass'), PASSWORD_DEFAULT);
151.
152.         $mhs = $this->M_database_API-
>getMahasiswa('t_mahasiswa', array('mhs_id' => $mhs_id));
153.
154.         $data = array('password' => $new_pass);
155.
156.         if (password_verify($old_pass, $mhs->password)) {
157.             $this->M_database_API-
>updateData('t_mahasiswa', array('mhs_id' => $mhs_id), $data
);
158.             $this->response([
159.                 'changed' => true,
160.                 'message' => "Password diganti"
161.             ], 200);
162.         } else {

```

```

163.         $this->response([
164.             'changed' => false,
165.             'message' => "Password gagal"
166.         ], 200);
167.     }
168.
169. }
170.
171. public function token_post()
172. {
173.     $npm = $this->post('npm');
174.     $token = $this->post('token');
175.
176.     $obj = json_decode($token);
177.     $endpoint = $obj->endpoint;
178.     $auth = $obj->keys->auth;
179.     $p256dh = $obj->keys->p256dh;
180.
181.     $cek_token = $this->db->where('npm', $npm)-
>get('t_device_token')->row();
182.
183.     if ($cek_token) {
184.         $data = array('endpoint' => $endpoint, 'auth'-
=> $auth, 'p256dh' => $p256dh, 'npm' => $npm);
185.         $this->M_database_API-
>updateData('t_device_token', array('npm' => $npm), $data);
186.         $this->response([
187.             'token' => true,
188.             'message' => $token
189.         ], 200);
190.     } else {
191.         $data = array('endpoint' => $endpoint, 'auth'-
=> $auth, 'p256dh' => $p256dh, 'npm' => $npm);
192.         $this->M_database_API-
>insertData('t_device_token', $data);
193.         $this->response([
194.             'token' => true,
195.             'message' => $token
196.         ], 201);
197.     }
198. }
199.
200. }
201.
202. ?>
```

### Rest matkum.php

```

1. <?php
2. use Restserver\Libraries\REST_Controller;
3. defined('BASEPATH') OR exit('No direct script access allowed
');
4.
```

```

5. require APPPATH . 'libraries/REST_Controller.php';
6. require APPPATH . 'libraries/Format.php';
7.
8. class Rest_matkum extends CI_Controller {
9.
10.     use REST_Controller;
11.     REST_Controller::__construct as private __resTrait
12.     Construct;
13. }
14. function __construct()
15. {
16.     // Construct the parent class
17.     // parent::__construct();
18.     $this->__resTraitConstruct();
19.     $this->load->model('M_database_API');
20.
21. }
22.
23. public function index_get()
24. {
25.
26.     $npm = $this->get('npm');
27.
28.     // Get Registrasi terakhir
29.     $matkum_akhir = $this->db->limit(1)-
>order_by('registrasi_id DESC')-
>join('t_jadwal', 'registrasi_praktikum.jadwal_kode = t_jadw
al.jadwal_kode')-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_prakt
ikum.matkum_id')->where('npm', $npm)-
>get('registrasi_praktikum')->row();
30.     // Get All Registrasi
31.     $jadwal = $this->M_database_API-
>getJadwal('registrasi_praktikum', array('npm' => $npm), 're
gistrasi_id DESC');
32.
33.     if ($matkum_akhir) {
34.         $this->response([
35.             'status' => true,
36.             'id_akhir' => $matkum_akhir-
>jadwal_id,
37.             'kode_akhir' => $matkum_akhir-
>jadwal_kode,
38.             'matkum_akhir' => $matkum_akhir-
>matkum,
39.             'jadwal' => $jadwal
40.         ], 200);
41.     } else {
42.         $this->response([
43.             'status' => false,
44.             'message' => "Data tidak ditemukan"
45.         ], 204);
46.     }
47. }
48.

```

```

49. }
50.
51. ?>

```

### Rest\_nilai.php

```

1. <?php
2. use Restserver\Libraries\REST_Controller;
3. defined('BASEPATH') OR exit('No direct script access allowed
   ');
4.
5. require APPPATH . 'libraries/REST_Controller.php';
6. require APPPATH . 'libraries/Format.php';
7.
8. class Rest_nilai extends CI_Controller {
9.
10.    use REST_Controller {
11.        REST_Controller::__construct as private __resTrait
12.        Construct;
13.    }
14.    function __construct()
15.    {
16.        // Construct the parent class
17.        // parent::__construct();
18.        $this->__resTraitConstruct();
19.        $this->load->model('M_database_API');
20.        $this->load->model('M_database');
21.
22.    }
23.
24.    public function index_get()
25.    {
26.
27.        $npm = $this->get('npm');
28.        $jadwal_kode = $this->get('jadwal_kode');
29.
30.        $jadwal = $this->db-
>select('p_kehadiran, p_tugas, p_ujian')-
>where('jadwal_kode', $jadwal_kode)->get('t_jadwal')->row();
31.        $ujian = $this->db-
>select('nilai_tugas, nilai_ujian')-
>where(array('jadwal_kode' => $jadwal_kode, 'npm' => $npm))-
>get('registrasi_praktikum')->row_array();
32.        $tugas = $this->db-
>select('t_tugas.tugas_judul, t_tugas.tugas_kode')-
>where('jadwal_kode', $jadwal_kode)->get('t_tugas')-
>result_array();
33.        $nilai = $this->db-
>select('t_tugas_nilai.nilai, t_tugas_nilai.npm')-
>where(array('npm' => $npm, 't_tugas_nilai.jadwal_kode' => $
jadwal_kode))->get('t_tugas_nilai')->result_array();

```

```

34.         $kehadiran = $this->M_database-
>kehadiran($jadwal_kode, $npm);
35.         $matkum = $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_prakt
ikum.matkum_id')->where('jadwal_kode', $jadwal_kode)-
>get('t_jadwal')->row();
36.         $pertemuan = $this->M_database_API-
>countWhere('t_tugas', array('jadwal_kode' => $jadwal_kode))
;
37.
38.         $arr = [];
39.         $i = 0;
40.         while ($i < $pertemuan) {
41.             array_push($arr, ['tugas_kode' => "", 'tugas_j
udul' => "", 'nilai' => 0]);
42.             $tugas[$i]['nilai'] = "0";
43.             $i++;
44.         }
45.
46.
47.         foreach ($tugas as $key => $value) {
48.             $arr[$key]['tugas_kode'] = $value['tugas_kode'
];
49.             $arr[$key]['tugas_judul'] = $value['tugas_judu
l'];
50.         }
51.
52.         foreach ($nilai as $key => $value) {
53.             $arr[$key]['nilai'] = $value['nilai'];
54.         }
55.
56.         $nilai_ujian = $ujian['nilai_ujian'];
57.         $nilai_tugas = $ujian['nilai_tugas'];
58.
59.         $total_kehadiran = $kehadiran->kehadiran * 10;
60.         $total_tugas = $nilai_tugas*$jadwal->p_tugas/100;
61.
62.         $n_kehadiran = $total_kehadiran*$jadwal-
>p_kehadiran/100;
63.         $n_tugas = $total_tugas;
64.         $n_ujian = $nilai_ujian*$jadwal->p_ujian/100;
65.
66.         $hasil = ($total_kehadiran*$jadwal-
>p_kehadiran/100) + $total_tugas + ($nilai_ujian*$jadwal-
>p_ujian/100);
67.
68.         $this->response([
69.             'npm' => $npm,
70.             'jadwal_kode' => $matkum->jadwal_kode,
71.             'matkum' => $matkum->matkum,
72.             'p_kehadiran' => $jadwal->p_kehadiran,
73.             'p_tugas' => $jadwal->p_tugas,
74.             'p_ujian' => $jadwal->p_ujian,
75.             'total_kehadiran' => "$total_kehadiran",
76.             'total_tugas' => "$nilai_tugas",
77.             'total_ujian' => $nilai_ujian,

```

```

78.             'kehadiran' => "$n_kehadiran",
79.             'tugas' => "$n_tugas",
80.             'ujian' => "$n_ujian",
81.             'hasil' => "$hasil",
82.             'nilai' => $arr
83.         ], 200);
84.     }
85. }
86. ?>
87.
88. ?>

```

### Rest\_ticketing.php

```

1. <?php
2. use Restserver\Libraries\REST_Controller;
3. defined('BASEPATH') OR exit('No direct script access allowed
   ');
4.
5. require APPPATH . 'libraries/REST_Controller.php';
6. require APPPATH . 'libraries/Format.php';
7.
8. class Rest_ticketing extends CI_Controller {
9.
10.     use REST_Controller {
11.         REST_Controller::__construct as private __resTrait
Construct;
12.     }
13.
14.     function __construct()
15.     {
16.         // Construct the parent class
17.         // parent::__construct();
18.         $this->__resTraitConstruct();
19.         $this->load->model('M_database_API');
20.
21.     }
22.
23.     public function index_get()
24.     {
25.         $npm = $this->get('npm');
26.
27.         $ticket = $this->M_database_API-
>getWhere('t_ticket', null, null, array('npm' => $npm), 'tic
ket_id DESC');
28.         if ($ticket) {
29.             $this->response([
30.                 'status' => true,
31.                 'ticket' => $ticket
32.             ], 200);
33.         } else {
34.             $this->response([
35.                 'status' => false,

```

```

36.             'ticket' => $ticket
37.         ], 200);
38.     }
39. }
40.
41. public function pesan_get()
42. {
43.     $npm = $this->get('npm');
44.     $this->M_database_API-
>updateData('t_pesan_mhs', array('npm' => $npm, 'pengirim' =
> 'admin'), array('pesan_status' => 1));
45.
46.     $pesan = $this->M_database_API->getPesanMhs($npm);
47.     if ($pesan) {
48.         $this->response([
49.             'status' => true,
50.             'pesan' => $pesan
51.         ], 200);
52.     } else {
53.         $this->response([
54.             'status' => false,
55.             'pesan' => $pesan
56.         ], 200);
57.     }
58. }
59.
60. public function kritiksaran_post()
61. {
62.     $kritik_saran = $this-
>post('kritik_saran');
63.     $ks = $this->M_database_API-
>insertData('t_kritik_saran', array('kritik_saran' => $kriti
k_saran, 'ks_status' => 0));
64.     if ($ks) {
65.         $this->response([
66.             'ks' => true,
67.             'message' => 'sukses'
68.         ], 201);
69.     } else {
70.         $this->response([
71.             'ks' => false,
72.             'message' => 'gagal'
73.         ], 204);
74.     }
75. }
76.
77. public function kirimpesan_post()
78. {
79.     $npm = $this->post('npm');
80.     $nama = $this->post('nama');
81.     $isi_pesan = $this->post('pesan');
82.
83.     $data = array(
84.         'npm' => $npm,
85.         'nama' => $nama,
86.         'pesan' => $isi_pesan,

```

```

87.          'pengirim' => 'mahasiswa',
88.          'pesan_status' => 0,
89.          'date_created' => date('Y-m-d H:i:s')
90.      );
91.
92.      $pesan = $this->M_database_API-
93.          >insertData('t_pesan_mhs', $data);
94.      if ($pesan) {
95.          $this->response([
96.              'kirim' => true,
97.              'message' => 'sukses',
98.              'pesan' => $data
99.          ], 201);
100.     } else {
101.         $this->response([
102.             'kirim' => false,
103.             'message' => 'gagal'
104.         ], 204);
105.     }
106.
107.     public function pindahshift_post()
108.     {
109.         $npm = $this->post('npm');
110.         $nama = $this->post('nama');
111.         $shift_asal = $this->post('shift_asal');
112.         $shift_tujuan = $this-
113.             >post('shift_tujuan');
114.         $deskripsi = $this->post('deskripsi');
115.
116.         $data = array(
117.             'ticket_kode' => "PS".date('Ymdhis'),
118.             'shift_asal' => $shift_asal,
119.             'shift_tujuan' => $shift_tujuan,
120.             'ticket_deskripsi' => $deskripsi,
121.             'npm' => $npm,
122.             'nama' => $nama,
123.             'ticket_status' => 0,
124.             'date_created' => date('Y-m-d H:i:s')
125.         );
126.
127.         $data2 = array(
128.             'npm' => $npm,
129.             'nama' => $nama,
130.             'pesan' => 'Permintaan pindah shift dari '.$sh-
131.                 ift_asal.' ke '.$shift_tujuan.', kode ticket PS'.date('Ymdhi-
132.                     s'),
133.             'pengirim' => 'mahasiswa',
134.             'pesan_status' => 0,
135.             'date_created' => date('Y-m-d H:i:s')
136.         );
137.
138.         $pesan = $this->M_database_API-
139.             >insertData('t_pesan_mhs', $data2);
140.         $ticket = $this->M_database_API-
141.             >insertData('t_ticket', $data);

```

```

137.         if ($ticket) {
138.             $this->response([
139.                 'status' => true,
140.                 'message' => 'sukses',
141.                 'ticket' => $data,
142.                 'pesan' => $data2
143.             ], 201);
144.         } else {
145.             $this->response([
146.                 'status' => false,
147.                 'message' => 'gagal'
148.             ], 204);
149.         }
150.     }
151.
152. }
153.
154. ?>

```

### Rest\_tugas.php

```

1. <?php
2. use Restserver\Libraries\REST_Controller;
3. defined('BASEPATH') OR exit('No direct script access allowed
   ');
4.
5. require APPPATH . 'libraries/REST_Controller.php';
6. require APPPATH . 'libraries/Format.php';
7.
8. class Rest_tugas extends CI_Controller {
9.
10.     use REST_Controller {
11.         REST_Controller::__construct as private __resTrait
12.         Construct;
13.     }
14.     function __construct()
15.     {
16.         // Construct the parent class
17.         // parent::__construct();
18.         $this->__resTraitConstruct();
19.         $this->load->model('M_database_API');
20.
21.     }
22.
23.     public function index_get()

```

```

24.      {
25.
26.          $npm = $this->get('npm');
27.          $jadwal_kode = $this->get('jadwal_kode');
28.
29.          $registrasi = $this->M_database_API-
>getRegistrasi('registrasi_praktikum', array('npm' => $npm,
't_jadwal.status' => 1), null);
30.
31.          $jadwal = [];
32.
33.          if ($registrasi != null) {
34.              foreach ($registrasi as $reg) {
35.                  array_push($jadwal, "t_tugas.jadwal_kode =
'$reg[jadwal_kode]' OR");
36.              }
37.
38.          $jadwal_mhs = rtrim(implode(" ", $jadwal), ' O
R');
39.
40.      } else {
41.          $jadwal_mhs = "t_tugas.jadwal_kode = 'kosong'";
;
42.      }
43.
44.      $tugas = $this->M_database_API-
>getTugas('t_tugas', array('t_tugas.tugas_status' => 1), $ja
dwal_mhs, 'batas_waktu ASC');
45.
46.
47.      $matkum = $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_prakt
ikum.matkum_id')->where('jadwal_kode', $jadwal_kode)->
get('t_jadwal')->row();
48.
        $pertemuan = $this->db-
>where('jadwal_kode', $jadwal_kode)->limit(1)->
order_by('pertemuan DESC')->get('t_absensi')->row();
49.
50.      $j = 0;
51.
52.      foreach ($tugas as $key => $value) {
53.
54.          $file_tugas = $this->db-
>query("SELECT COUNT(file_id) AS exist FROM t_tugas_file WHE
RE tugas_id='".$value[tugas_id]' AND npm='".$npm"'")->row();
55.
        $nilai_tugas = $this->db-
>query("SELECT COUNT(nilai_id) AS exist FROM t_tugas_nilai W
HERE tugas_id='".$value[tugas_id]' AND npm='".$npm"'")->row();
56.
57.          if ($file_tugas->exist == 1 || $nilai_tugas-
>exist == 1) {
58.
            $tugas[$j]['mengumpulkan'] = 'sudah';
59.        } else {
59.
            $tugas[$j]['mengumpulkan'] = 'belum';
60.
        }
    }
}

```

```

61.         }
62.
63.         $waktu = strtotime($value['batas_waktu']) - st
64.             rtotime('now');
65.
66.
67.         if ($menit < 1) {
68.             $menit = 0;
69.         }
70.
71.         $hours = floor($menit / 60);
72.         $minutes = ($menit % 60);
73.
74.         $sisa_waktu = sprintf('%02d:%02d', $hours, $mi
75.             nutes);
76.
77.         $tugas[$j]['sisa_waktu'] = $menit;
78.
79.         $j++;
80.     }
81.
82.     if ($tugas) {
83.         $this->response([
84.             'status' => true,
85.             'npm' => $npm,
86.             'tugas' => $tugas
87.         ], 200);
88.     } else {
89.         $this->response([
90.             'status' => false,
91.             'npm' => $npm,
92.             'tugas' => $tugas
93.         ], 200);
94.     }
95. }
96.
97. public function detail_get()
98. {
99.
100.     $tugas_id = $this->get('tugas_id');
101.     $tugas_kode = $this->get('tugas_kode');
102.
103.     $tugas = $this->M_database_API-
104.         >getTugasDetail('t_tugas', array('t_tugas.tugas_id' => $tuga
105.             s_id));
106.     $image = $this->db-
107.         >where('tugas_kode', $tugas_kode)->get('t_tugas_image')-
108.             >result_array();
109.     $j = 0;
110.
111.     foreach ($image as $key => $img) {
112.         $image[$key]['tugas_image'] = base_url()."asse
113.             ts/images/tugas/$img[tugas_image]";
114.     }
115.
```

```

110.         }
111.
112.         if ($tugas) {
113.             $this->response([
114.                 'status' => true,
115.                 'id_tugas' => $tugas[0]['tugas_kode'],
116.                 'tugas' => $tugas,
117.                 't_image' => $image
118.             ], 200);
119.         } else {
120.             $this->response([
121.                 'status' => false,
122.                 'tugas' => $tugas,
123.                 't_image' => $image
124.             ], 200);
125.         }
126.     }
127.
128.     public function upload_post()
129.     {
130.
131.         $tugas_id = $this->post('tugas_id');
132.         $file_nama = $this->post('file_nama');
133.         $npm = $this->post('npm');
134.
135.         // CEK FILE
136.         $cek_file = $this->db-
>where(array('tugas_id' => $tugas_id, 'npm' => $npm))-
>get('t_tugas_file')->row();
137.
138.         $config['upload_path']          = './assets/tugas_
file/';
139.         $config['allowed_types']        = 'jpg|png|jpeg|wo
rd|pdf|doc|docx|zip|rar|7z|7zip';
140.         $config['file_name']           = $file_nama;
141.         $config['overwrite']           = false;
142.         $config['max_size']            = 0;
143.
144.         $this->load->library('upload', $config);
145.
146.         // FILE ADA
147.         if ($cek_file) {
148.             if ($this->upload->do_upload('file')) {
149.                 unlink('assets/tugas_file/'.$cek_file-
>file_nama);
150.                 $file = $this->upload->data();
151.
152.                 $data = array(
153.                     'tugas_id' => $tugas_id,
154.                     'npm' => $npm,
155.                     'file_nama' => $file['file_name'],
156.                     'tgl_upload' => date("Y-m-d H:i")
157.                 );
158.             }

```

```

159.           $this->M_database_API-
160.           >updateData('t_tugas_file', array('npm' => $npm, 'tugas_id'
161.           => $tugas_id), $data);
162.           $this->response([
163.               'upload' => true,
164.               'message' => "Upload berhasil"
165.           ], 200);
166.       } else {
167.           $this->response([
168.               'upload' => false,
169.               'message' => $this->upload-
170.               >display_errors()
171.           ], 200);
172.
173.       } else {
174.           // FILE KOSONG
175.           if ($this->upload-
176.               >do_upload('file')) {
177.               $file = $this->upload->data();
178.
179.               $data = array(
180.                   'tugas_id' => $tugas_id,
181.                   'npm' => $npm,
182.                   'file_nama' => $file['file_name'],
183.                   'tgl_upload' => date("Y-m-d H:i")
184.               );
185.
186.           $this->M_database_API-
187.           >insertData('t_tugas_file', $data);
188.
189.           $this->response([
190.               'upload' => true,
191.               'message' => "Upload berhasil"
192.           ], 200);
193.       } else {
194.           $this->response([
195.               'upload' => false,
196.               'message' => $this->upload-
197.               >display_errors()
198.           ], 200);
199.
200.       }
201.
202.   }
203.
204. ?>

```

## M\_database.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. /**
5.  *
6. */
7. class M_database extends CI_Model
8. {
9.
10.    function __construct()
11.    {
12.        parent::__construct();
13.    }
14.
15.    // GET DATA JADWAL PERIODE
16.    public function getJadwal($where, $order = null, $limi
t = null)
17.    {
18.        $this->db->order_by($order);
19.        $this->db->limit($limit);
20.        $this->db->where($where);
21.        $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id=t_mata_praktik
um.matkum_id');
22.        $this->db-
>join('t_kelas', 't_jadwal.kelas_kode=t_kelas.kelas_kode');
23.        $this->db-
>join('t_ruangan', 't_jadwal.ruangan_id=t_ruangan.ruangan_id
');
24.        return $this->db->get('t_jadwal');
25.    }
26.
27.    public function getAllJadwalAsisten($orwhere, $order =
null)
28.    {
29.        $this->db->order_by($order);
30.        $this->db-
>where("($orwhere)", NULL, FALSE);
31.        $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_prakt
ikum.matkum_id');
32.        $this->db-
>join('t_ruangan', 't_jadwal.ruangan_id = t_ruangan.ruangan_
id');
33.        return $this->db->get('t_jadwal');
34.    }
35.
36.    public function getJadwalAsisten($orwhere, $order = nu
ll)
37.    {
38.        $this->db->order_by($order);
39.        $this->db->where('status = 1');

```

```

40.      $this->db-
>where("($orwhere)", NULL, FALSE);
41.      $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_praktikum.matkum_id');
42.      $this->db-
>join('t_ruangan', 't_jadwal.ruangan_id = t_ruangan.ruangan_id');
43.      return $this->db->get('t_jadwal');
44. }
45.
46. public function getJadwalDashboard($orwhere, $order =
null, $where = null)
47. {
48.     $this->db->order_by($order);
49.     $this->db->where('status = 1');
50.     $this->db->where($where);
51.     $this->db-
>where("($orwhere)", NULL, FALSE);
52.     $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_praktikum.matkum_id');
53.     $this->db-
>join('t_kelas', 't_jadwal.kelas_kode = t_kelas.kelas_kode')
;
54.     $this->db-
>join('t_ruangan', 't_jadwal.ruangan_id = t_ruangan.ruangan_id');
55.     return $this->db->get('t_jadwal');
56. }
57.
58. // GET DATA REGISTRASI MAHASISWA
59.
60. public function getRegistrasiMahasiswa($where)
61. {
62.     $this->db->order_by('t_mahasiswa.npm ASC');
63.     $this->db->where($where);
64.     $this->db-
>join('t_mahasiswa', 'registrasi_praktikum.npm=t_mahasiswa.npm');
65.     $this->db-
>join('t_kelas', 't_mahasiswa.kelas_kode=t_kelas.kelas_kode'
);
66.     return $this->db->get('registrasi_praktikum');
67. }
68.
69. // GET NILAI PRAKTIKUM
70. public function getNilaiPraktikum($where)
71. {
72.     $this->db-
>select('jadwal_kode, matkum_id, kelas_kode, p_kehadiran, p_tugas, p_ujian, asisten_1, asisten_2');
73.     $this->db->order_by('jadwal_kode ASC');
74.     $this->db->where($where);
75.     return $this->db->get('t_jadwal');
76. }

```

```

77.
78.      // GET MATKUM DOSEN
79.      public function getMatkumDosen($where)
80.      {
81.          $this->db->select('t_jadwal.matkum_id, matkum');
82.          $this->db->order_by('matkum ASC');
83.          $this->db->group_by('t_jadwal.matkum_id');
84.          $this->db->where($where);
85.          $this->db->where('status', 1);
86.          $this->db-
87.              >join('t_mata_praktikum', 't_jadwal.matkum_id=t_mata_praktik
     um.matkum_id');
88.              return $this->db->get('t_jadwal');
89.      }
90.      // GET KELAS DOSEN
91.      public function getKelasDosen($where)
92.      {
93.          $this->db-
94.              >select('t_jadwal.kelas_kode, kelas_nama');
95.              $this->db->order_by('kelas_nama ASC');
96.              $this->db->group_by('t_jadwal.kelas_kode');
97.              $this->db->where($where);
98.              $this->db-
99.                  >join('t_kelas', 't_jadwal.kelas_kode=t_kelas.kelas_kode');
100.                 return $this->db->get('t_jadwal');
101.             }
102.
103.             // GET DETAIL TUGAS
104.             public function getDetailTugas($where)
105.             {
106.                 $this->db->order_by('t_mahasiswa.nama ASC');
107.                 $this->db-
108.                     >join('t_mahasiswa', 'registrasi_praktikum.npm=t_mahasiswa.n
     pm');
109.                     $this->db-
110.                         >join('t_kelas', 't_mahasiswa.kelas_kode=t_kelas.kelas_kode'
     );
111.                         return $this->db->get('registrasi_praktikum');
112.                     }
113.                     // GET TOKEN AKTIF
114.                     public function getToken()
115.                     {
116.                         $this->db->where('kelas_status = 1');
117.                         $this->db-
118.                             >join('t_mahasiswa', 't_device_token.npm=t_mahasiswa.npm');
119.                             $this->db-
120.                                 >join('t_kelas', 't_mahasiswa.kelas_kode=t_kelas.kelas_kode'
     );
121.                                 return $this->db->get('t_device_token');
122.                             }
123.                             // GET PESAN

```

```

122.
123.     public function getPesananMhs()
124.     {
125.         $this->db->select('*');
126.         $this->db->select_max('pesan_id' , 'max_ur_time');
127.         $this->db->order_by('pesan_status', 'DESC');
128.         $this->db->group_by('t_pesan_mhs.npm');
129.         return $this->db->get('t_pesanan_mhs');
130.     }
131.
132.     public function getPesan($user1, $user2)
133.     {
134.         $this->db->where('pesan_user', $user1);
135.         $this->db->or_where('pesan_user', $user2);
136.         return $this->db->get('t_pesanan')->result();
137.     }
138.
139.     public function getPesanGrup($kanal)
140.     {
141.         $this->db->where('pesan_kanal', $kanal);
142.         return $this->db->get('t_pesanan')->result();
143.     }
144.
145. // GET ALL DATA
146.     public function getAllData($table, $join = null, $onjoin
147.     in = null, $order1 = null, $order2 = null, $limit = null, $o
148.     ffset = null)
149.     {
150.         $this->db->order_by($order1);
151.         $this->db->order_by($order2);
152.         $this->db->limit($limit, $offset);
153.         if ($join == null) {
154.             return $this->db->get($table);
155.         } else {
156.             $this->db->join($join, $onjoin);
157.             return $this->db->get($table);
158.         }
159.
160. // GET ALL DATA WHERE
161.     public function getWhere($table, $join = null, $onjoin
162.     = null, $where = null, $order = null)
163.     {
164.         $this->db->order_by($order);
165.         if ($join == null) {
166.             $this->db->where($where);
167.             return $this->db->get($table);
168.         } else {
169.             $this->db->join($join, $onjoin);
170.             $this->db->where($where);
171.             return $this->db->get($table);
172.         }
173.     }
174.

```

```

175.     public function cekdata($table, $att, $isi){
176.         return $this->db-
>query("SELECT COUNT($att) AS jumlah FROM (SELECT * FROM $table WHERE $att='$isi' LIMIT 1) tabel")->row()->jumlah;
177.     }
178.
179.     public function cekPesan($pesan_id, $penerima){
180.         return $this->db-
>query("SELECT COUNT($pesan_id) AS jumlah FROM (SELECT * FROM t_pesan_status WHERE pesan_id = '$pesan_id' AND pesan_penerima = '$penerima' LIMIT 1) tabel")->row()->jumlah;
181.     }
182.
183.     public function cekdataAbsen($table, $att1, $att2, $att3, $isi1, $isi2, $isi3){
184.         return $this->db-
>query("SELECT COUNT($att1) AS jumlah FROM (SELECT * FROM $table WHERE $att1='$isi1' AND $att2='$isi2' AND $att3='$isi3' LIMIT 1) tabel")->row()->jumlah;
185.     }
186.
187. // cek absensi mahasiswa
188. function cekAbsen($table, $jadwal_kode, $npm, $pertemuan){
189.     return $this->db-
>query("SELECT COUNT(absensi_id) AS hadir FROM $table WHERE jadwal_kode='$jadwal_kode' AND npm='$npm' AND pertemuan='$pertemuan'")->row();
190. }
191.
192. function cekAbsenAsisten($table, $jadwal_kode, $asisten_id, $pertemuan){
193.     return $this->db-
>query("SELECT COUNT(absensi_asisten_id) AS hadir FROM $table WHERE jadwal_kode='$jadwal_kode' AND asisten_id='$asisten_id' AND pertemuan='$pertemuan'")->row();
194. }
195.
196. // hapus absen
197. function hapusAbsen($table, $att1, $att2, $att3, $isi1, $isi2, $isi3){
198.     $this->db->where($att1, $isi1)->where($att2, $isi2)->where($att3, $isi3);
199.     $this->db->delete($table);
200. }
201.
202. // cek pertemuan
203. function cekPertemuan($jadwal_kode){
204.     return $this->db-
>query("SELECT MAX(pertemuan) AS pertemuan FROM t_absensi WHERE jadwal_kode='$jadwal_kode'")->row();
205. }
206.
207. // cek tanggal absen
208. function cekTanggal($jadwal_kode)
209. {

```

```

210.         return $this->db-
>query("SELECT MAX(absen_tgl) AS absen_tgl FROM t_absensi WHERE jadwal_kode='$jadwal_kode'")->row();
211.     }
212.
213.     // get total kehadiran mahasiswa
214.
215.     function kehadiran($jadwal_kode, $npm) {
216.         return $this->db-
>query("SELECT COUNT(npm) AS kehadiran FROM t_absensi WHERE jadwal_kode='$jadwal_kode' AND npm='$npm'")->row();
217.     }
218.
219.     function nilai_tugas($jadwal_kode, $npm) {
220.         $this->db-
>where(array('jadwal_kode' => $jadwal_kode, 'npm' => $npm));
221.         return $this->db->get('t_tugas_nilai');
222.     }
223.
224.     // INSERT MULTIPLE DATA
225.
226.     function insertBatch($table, $data) {
227.         return $this->db->insert_batch($table, $data);
228.     }
229.
230.     function maxwhere($table, $att1, $isi, $att2) {
231.         return $this->db->where($att1, $isi)-
>select_max($att2)->get($table)->row();
232.     }
233.
234.     // BASIC FUNCTION
235.     public function insertData($table, $data)
236.     {
237.         $this->db->insert($table, $data);
238.     }
239.
240.     public function updateData($table, $where, $data)
241.     {
242.         $this->db->where($where);
243.         $this->db->update($table, $data);
244.     }
245.
246.     public function deleteData($table, $where)
247.     {
248.         $this->db->where($where);
249.         $this->db->delete($table);
250.     }
251.
252.     public function count($table = null)
253.     {
254.         return $this->db->count_all($table);
255.     }
256.
257.     public function countWhere($table = null, $where = null)
258.     {

```

```

259.         $this->db->where($where);
260.         $this->db->from($table);
261.         return $this->db->count_all_results();
262.     }
263.
264.     function jumlah_data($table) {
265.         return $this->db->get($table)->num_rows();
266.     }
267.
268.     // DATATABLE SERVER SIDE
269.
270.     // var $select_column = array('npm', 'nama', 'kelas_na
271.     // ma', 'kelas_status');
272.     // var $order_column = array('npm' => 'DESC');
273.
274.     function get_all_mahasiswa() {
275.         $this->db-
276.             >select('mhs_id, npm, nama, t_kelas.kelas_kode, kelas_nama,
277.                     status_mhs, status_bayar');
278.             $this->db->from('t_mahasiswa');
279.             $this->db-
280.                 >join('t_kelas', 't_kelas.kelas_kode=t_mahasiswa.kelas_kode'
281. );
282.             if (isset($_POST["search"]["value"])) {
283.                 $this->db-
284.                     >like("npm", $_POST["search"]["value"]);
285.                     $this->db-
286.                         >or_like("nama", $_POST["search"]["value"]);
287.                         $this->db-
288.                             >or_like("kelas_nama", $_POST["search"]["value"]);
289.                             }
290.                             function render_datatable() {
291.                                 $this->get_all_mahasiswa();
292.                                 if ($_POST["length"] != -1) {
293.                                     $this->db-
294.                                         >limit($_POST['length'], $_POST["start"]);
295.                                         }
296.                                         $query = $this->db->get();
297.                                         return $query->result();
298.                                         }
299.                                         function get_filtered_data() {
300.                                             $this->get_all_mahasiswa();
301.                                             $query = $this->db->get();
302.                                             return $query->num_rows();
303.                                         }
304.                                         function get_all_data()
305.                                         {

```

```

306.         $this->db->select("*");
307.         $this->db->from("t_mahasiswa");
308.         return $this->db->count_all_results();
309.     }
310.
311. }
312.
313. ?>

```

### M\_database\_API.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. /**
5. *
6. */
7. class M_database_API extends CI_Model
8. {
9.
10.     function __construct()
11.     {
12.         parent::__construct();
13.     }
14.
15.     public function getInfo()
16.     {
17.         $this->db->order_by('info_id DESC');
18.         return $this->db->get('t_info');
19.     }
20.
21.     public function getInfoImage($where)
22.     {
23.         $this->db->order_by('info_image_id ASC');
24.         $this->db->where($where);

```

```

25.         return $this->db->get('t_info_image');
26.     }
27.
28.     public function getJadwal($table, $where = null, $order = null)
29.     {
30.         $this->db->order_by($order);
31.         $this->db-
>select('t_jadwal.jadwal_kode, t_jadwal.periode_id, t_mata_praktikum.matkul_id, t_mata_praktikum.matkul, t_jadwal.asisten_1 AS pengajar1, t_jadwal.asisten_2 AS pengajar2, t_jadwal.jadwal_hari, t_jadwal.jadwal_jam, t_jadwal.jadwal_kode, t_ruangan.ruangan_nama');
32.         $this->db-
>join('t_jadwal', 'registrasi_praktikum.jadwal_kode = t_jadwal.jadwal_kode');
33.         $this->db-
>join('t_asisten', 't_jadwal.asisten_1 = t_asisten.asisten_id');
34.         $this->db-
>join('t_mata_praktikum', 't_jadwal.matkul_id = t_mata_praktikum.matkul_id');
35.         $this->db-
>join('t_ruangan', 't_jadwal.ruangan_id = t_ruangan.ruangan_id');
36.         return $this->db->get_where($table, $where)-
>result_array();
37.     }
38.
39.     public function getAllJadwal($table, $where = null, $order = null)
40.     {
41.         $this->db->order_by($order);
42.         $this->db-
>select('t_jadwal.jadwal_kode, t_jadwal.matkul_id, t_jadwal.periode_id, t_jadwal.jadwal_hari, t_jadwal.jadwal_jam, kelas_nama');
43.         $this->db-
>join('t_kelas', 't_jadwal.kelas_kode = t_kelas.kelas_kode');
44.         return $this->db->get_where($table, $where)-
>result_array();
45.     }
46.
47.     public function getAbsensi($table, $where = null, $order = null)
48.     {
49.         $this->db->order_by($order);
50.         $this->db->group_by('pertemuan');
51.         $this->db-
>select('t_absensi.pertemuan, t_absensi.absen_tgl');
52.         $this->db-
>join('t_jadwal', 't_absensi.jadwal_kode = t_jadwal.jadwal_kode');

```

```

53.      $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_prakt
ikum.matkum_id');
54.      return $this->db->get_where($table, $where)-
>result_array();
55.  }
56.
57.  public function getNilaiTugas($table, $where = null, $order =
null)
58.  {
59.      $this->db->order_by($order);
60.      $this->db-
>select('t_tugas.tugas_judul, t_tugas.tugas_kode, t_tugas_ni
lai.nilai');
61.      $this->db-
>join('t_tugas', 't_tugas.nilai.tugas_id = t_tugas.tugas_id'
);
62.      $this->db-
>join('t_jadwal', 't_tugas.jadwal_kode = t_jadwal.jadwal_kod
e');
63.      return $this->db->get_where($table, $where)-
>result_array();
64.  }
65.
66.  public function getTugas($table, $where = null, $orwhe
re, $order = null)
67.  {
68.      $this->db->order_by($order);
69.      $this->db->where($where);
70.      $this->db-
>where("($orwhere)", NULL, FALSE);
71.      $this->db-
>select('t_tugas.tugas_id, t_tugas.tugas_judul, t_tugas.tuga
s_kode, t_mata_praktikum.matkum, t_tugas.batas_waktu, t_tuga
s.tugas_deskripsi');
72.      $this->db-
>join('t_jadwal', 't_tugas.jadwal_kode = t_jadwal.jadwal_kod
e');
73.      $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_prakt
ikum.matkum_id');
74.      return $this->db->get($table)-
>result_array();
75.  }
76.
77.  public function getTugasDetail($table, $where = null)
78.  {
79.      $this->db-
>select('t_tugas.tugas_id, t_tugas.tugas_judul, t_tugas.tuga
s_kode, t_mata_praktikum.matkum, t_tugas.tugas_deskripsi');
80.      $this->db-
>join('t_jadwal', 't_tugas.jadwal_kode = t_jadwal.jadwal_kod
e');
81.      $this->db-
>join('t_mata_praktikum', 't_jadwal.matkum_id = t_mata_prakt
ikum.matkum_id');

```

```

82.         return $this->db->get_where($table, $where)-
>result_array();
83.     }
84.
85.     public function getMahasiswa($table, $where)
86.     {
87.         $this->db-
>join('t_kelas', 't_mahasiswa.kelas_kode = t_kelas.kelas_kod
e');
88.         return $this->db->get_where($table, $where)-
>row();
89.
90.     }
91.
92.     public function getAsisten($table, $where)
93.     {
94.         return $this->db->get_where($table, $where)-
>result_array();
95.     }
96.
97.     public function getRegistrasi($table, $where = null, $o
rder = null)
98.     {
99.         $this->db->order_by($order);
100.        $this->db->select('t_jadwal.jadwal_kode');
101.        $this->db-
>join('t_jadwal', 'registrasi_praktikum.jadwal_kode = t_jadw
al.jadwal_kode');
102.        return $this->db->get_where($table, $where)-
>result_array();
103.
104.    }
105.
106.    // GET PESAN
107.
108.    public function getPesanMhs($npm)
109.    {
110.        $this->db->order_by('pesan_id ASC');
111.        $this->db-
>join('t_mahasiswa', 't_pesan_mhs.npm=t_mahasiswa.npm');
112.        $this->db->where('t_pesan_mhs.npm', $npm);
113.        return $this->db->get('t_pesan_mhs')-
>result_array();
114.    }
115.
116.    // GET ALL DATA WHERE
117.    public function getWhere($table, $join = null, $onjoin
= null, $where = null, $order = null)
118.    {
119.        $this->db->order_by($order);
120.
121.        if ($join == null) {
122.            $this->db->where($where);
123.            return $this->db->get($table)-
>result_array();
124.        } else {

```

```

125.         $this->db->join($join, $onjoin);
126.         $this->db->where($where);
127.         return $this->db->get($table)-
>result_array();
128.     }
129. }
130.
131. // Cek absen
132. public function cekdataAbsen($table, $att1, $att2, $is
i1, $isi2){
133.     return $this->db-
>query("SELECT COUNT($att1) AS jumlah FROM (SELECT * FROM $t
able WHERE $att1='$isi1' AND $att2='$isi2' LIMIT 1) tabel")-
>row()->jumlah;
134. }
135.
136. public function cekdataAbsenMhs($table, $att1, $att2,
$att3, $isi1, $isi2, $isi3){
137.     return $this->db-
>query("SELECT COUNT($att1) AS jumlah FROM (SELECT * FROM $t
able WHERE $att1='$isi1' AND $att2='$isi2' AND $att3='$isi3'
LIMIT 1) tabel")->row()->jumlah;
138. }
139.
140. // BASIC FUNCTION
141. public function insertData($table, $data)
142. {
143.     $this->db->insert($table, $data);
144.     return $this->db->affected_rows();
145. }
146.
147. public function updateData($table, $where, $data)
148. {
149.     $this->db->where($where);
150.     $this->db->update($table, $data);
151. }
152.
153. public function deleteData($table, $where)
154. {
155.     $this->db->where($where);
156.     $this->db->delete($table);
157. }
158.
159. public function countWhere($table = null, $where = nul
l)
160. {
161.     $this->db->where($where);
162.     $this->db->from($table);
163.     return $this->db->count_all_results();
164. }
165.
166. }
167.
168. ?>

```



## SOURCE CODE APLIKASI CLIENT

### Absensi.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Absensi extends CI_Controller {
5.
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $npm = $this->session->userdata('npm');
16.        $jadwal_kode = $this->input->get('idp');
17.        $data['absen'] = $this->M_database-
>getAbsen($npm, $jadwal_kode);
18.
19.        $data['matkum'] = $this->M_database-
>getMatkum($npm);
20.        // $this->load->view('v-absensi', $data);
21.        $this->template->load('template', 'v-
absensi', $data);
22.    }
23.
24.    public function absensi()
25.    {
26.        $npm = $this->session->userdata('npm');
27.        $jadwal_kode = $this->input->get('idp');
28.        $data['absen'] = $this->M_database-
>getAbsen($npm, $jadwal_kode);
29.
30.        $data['matkum'] = $this->M_database-
>getMatkum($npm);
31.        $this->load->view('v-absensi', $data);
32.    }
33. }
```

### Auth.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Auth extends CI_Controller {
5.
```

```

6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        $this->load->helper('cookie');
11.    }
12.
13.    public function index()
14.    {
15.
16.        if (isset($_COOKIE['mhs']) && isset($_COOKIE['npm'])
17. )) {
18.             $mhs_id = $_COOKIE['mhs'];
19.             $npm = $_COOKIE['npm'];
20.             $mhs = $this->M_database-
>getMahasiswa($mhs_id);
21.             if ($npm === hash('sha256', $mhs['mahasiswa'][
'npm'])) {
22.                 $data_user = [
23.                     'mhs_id' => $mhs['mahasiswa']['mhs_id'],
24.                     'nama' => $mhs['mahasiswa']['nama'],
25.                     'nama_pendek' => $mhs['mahasiswa']['na-
ma_pendek'],
26.                     'panggilan' => $mhs['mahasiswa']['pang-
gilan'],
27.                     'kelas' => $mhs['mahasiswa']['kelas'],
28.                     'npm' => $mhs['mahasiswa']['npm']
29.                 ];
30.                 $this->session->set_userdata($data_user);
31.                 redirect('dashboard');
32.             }
33.         } else {
34.             $this->template->load('template', 'v-
login');
35.         }
36.
37.        public function do_login()
38.        {
39.            $npm = $this->input->post('npm');
40.            $password = $this->input->post('password');
41.            $auth = $this->M_database-
>auth_user($npm, $password);
42.
43.            if ($auth['kode'] == 1) {
44.                $data_user = [
45.                    'mhs_id' => $auth['mahasiswa']['mhs_id'],
46.                    'nama' => $auth['mahasiswa']['nama'],
47.                    'nama_pendek' => $auth['mahasiswa']['na-
ma_pendek'],
48.                    'panggilan' => $auth['mahasiswa']['panggil-
an'],
49.                    'kelas' => $auth['mahasiswa']['kelas'],
50.                    'npm' => $auth['mahasiswa']['npm']
51.                ];

```

```

52.           setcookie('mhs', $auth['mahasiswa']['mhs_id'],
53.             time()+(86400 * 180), "/");
54.           setcookie('npm', hash('sha256', $auth['mahasiswa']['npm']), time()+(86400 * 180), "/");
55.           $this->session->set_userdata($data_user);
56.           $user_id = $this->session->userdata('mhs_id');
57.           echo "sukses|dashboard|$user_id";
58.       } else if ($auth['kode'] == 2) {
59.           echo "gagal|Password salah";
60.       } else {
61.           echo "gagal|NPM tidak terdaftar";
62.       }
63.
64.   public function logout()
65.   {
66.       delete_cookie("npm");
67.       delete_cookie("mhs");
68.       $this->session->unset_userdata('npm');
69.       $this->session->unset_userdata('nama');
70.       $this->session->unset_userdata('nama_pendek');
71.       $this->session->unset_userdata('panggilan');
72.       $this->session->unset_userdata('kelas');
73.       $this->session->sess_destroy();
74.       redirect('auth');
75.   }
76.
77.   public function password_view()
78.   {
79.       $this->template->load('template', 'v-password');
80.   }
81.
82.   public function ganti_password()
83.   {
84.       $mhs_id = $this->input->post("mhs_id");
85.       $old_pass = $this->input->post("old_pass");
86.       $new_pass = $this->input->post("new_pass");
87.
88.       $password = $this->M_database-
89.           >ganti_password($mhs_id, $old_pass, $new_pass);
90.           if ($password['changed'] == true) {
91.               echo "sukses|auth/logout";
92.           } else {
93.               echo "gagal|Password salah";
94.           }
95.   }

```

## Dashboard.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');

```

```

3.
4. class Dashboard extends CI_Controller {
5.
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $npm = $this->session->userdata('npm');
16.        $data['kode_akhir'] = $this->M_database-
>getMatkum($npm);
17.        $data['info'] = $this->M_database->getInfo();
18.        $data['matkum'] = $this->M_database-
>getMatkum($npm);
19.        $data['jadwal'] = $this->M_database-
>getJadwal($npm);
20.        $praktikum = $this->M_database->getJadwal($npm);
21.        $pertemuan = 0;
22.        foreach ($praktikum['jadwal'] as $key => $value) {
23.            $pertemuan += $value['pertemuan'];
24.        }
25.        if ($pertemuan >= 20) {
26.            $data['pertemuan'] = '';
27.        } else {
28.            $data['pertemuan'] = 'hide';
29.        }
30.        $this->template->load('template', 'v-
dashboard', $data);
31.    }
32.
33.    public function info_praktikum()
34.    {
35.        $data['info'] = $this->M_database->getInfo();
36.        $this->template->load('template', 'v-
info', $data);
37.    }
38.
39.    public function tentang_lab()
40.    {
41.        $data['info'] = $this->M_database->getInfo();
42.        $this->template->load('template', 'v-
tentang', $data);
43.    }
44.
45.    public function profil()
46.    {
47.        $data['info'] = $this->M_database->getInfo();
48.        $this->template->load('template', 'v-
profil', $data);
49.    }
50.
51.    public function login_view()

```

```

52.      {
53.          $this->template->load('template', 'v-login');
54.      }
55.  }

```

### Hasil\_akhir.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Hasil_akhir extends CI_Controller {
5.
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $npm = $this->session->userdata('npm');
16.        $jadwal_kode = $this->input->get('idp');
17.        $data['nilai'] = $this->M_database-
>getNilai($npm, $jadwal_kode);
18.
19.        $data['matkum'] = $this->M_database-
>getMatkum($npm);
20.        $this->template->load('template', 'v-hasil-
   akhir', $data);
21.    }
22.
23.    public function hasil_akhir()
24.    {
25.        $npm = $this->session->userdata('npm');
26.        $jadwal_kode = $this->input->get('idp');
27.        $data['nilai'] = $this->M_database-
>getNilai($npm, $jadwal_kode);
28.
29.        $data['matkum'] = $this->M_database-
>getMatkum($npm);
30.        $this->load->view('v-hasil-akhir', $data);
31.    }
32. }

```

### Info\_praktikum.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Info_praktikum extends CI_Controller {
5.
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $data['info'] = $this->M_database->getInfo();
16.        $this->template->load('template', 'v-
   info', $data);
17.    }
18. }
```

### Nilai.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Nilai extends CI_Controller {
5.
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $npm = $this->session->userdata('npm');
16.        $jadwal_kode = $this->input->get('idp');
17.        $data['nilai'] = $this->M_database-
   >getNilai($npm, $jadwal_kode);
18.
19.        $data['matkum'] = $this->M_database-
   >getMatkum($npm);
20.        $this->template->load('template', 'v-
   nilai', $data);
21.    }
22.
23.    public function nilai()
24.    {
```

```

25.      $npm = $this->session->userdata('npm');
26.      $jadwal_kode = $this->input->get('idp');
27.      $data['nilai'] = $this->M_database-
>getNilai($npm, $jadwal_kode);
28.
29.      $data['matkum'] = $this->M_database-
>getMatkum($npm);
30.      $this->load->view('v-nilai', $data);
31.  }
32. }
```

## Pengaturan.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Pengaturan extends CI_Controller {
5.
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $this->template->load('template', 'v-pengaturan');
16.    }
17.
18.    public function subscribe()
19.    {
20.        $npm = $this->session->userdata('npm');
21.        $token = $this->input->post('token');
22.        $this->M_database-
>sendToken($npm, $token);
23.    }
24. }
```

## Ticketing.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. class Ticketing extends CI_Controller {
5.
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $npm = $this->session->userdata('npm');
16.        $data['jadwal'] = $this->M_database-
>getJadwal($npm);
17.        $data['ticket'] = $this->M_database-
>getTicket($npm);
18.        $this->template->load('template', 'v-
ticketing', $data);
19.    }
20.
21.    public function kirim_ks()
22.    {
23.        $ks = $this->input->post('kritik_saran');
24.        $this->M_database->kritik_saran($ks);
25.    }
26.
27.    public function kirim_pesan()
28.    {
29.        $npm = $this->session->userdata('npm');
30.        $nama = $this->session->userdata('nama');
31.        $isi_pesan = $this->input->post('pesan');
32.        $pesan = $this->M_database-
>kirim_pesan($npm, $nama, $isi_pesan);
33.        echo '<div class="chat-text-
right"><span>'.$pesan['pesan']['pesan'].'<br><small class="r
ight grey-text text-lighten-
1">'.$pesan['pesan']['date_created'].'</small></span></div>';
34.    }
35.
36.    public function pindah_shift()
37.    {
38.        $npm = $this->session->userdata('npm');
39.        $nama = $this->session->userdata('nama');
40.        $shift_asal = $this->input->post('shift_asal');
41.        $shift_tujuan = $this->input-
>post('shift_tujuan');
42.        $deskripsi = $this->input->post('deskripsi');

```

```

43.         $ticket = $this->M_database-
>pindah_shift($npm, $nama, $shift_asal, $shift_tujuan, $desk-
ripsi);
44.         echo '<tr>
45.                 <td></td>
46.                 <td class="modal-trigger pointer det-
ticket" data-target="modalTicket"
47.                         data-
ticket_kode="'. $ticket['ticket']['ticket_kode'] .'"
48.                         data-
ticket_deskripsi="'. $ticket['ticket']['ticket_deskripsi'] .'"
49.                         data-
shift_asal="'. $ticket['ticket']['shift_asal'] .'"
50.                         data-
shift_tujuan="'. $ticket['ticket']['shift_tujuan'] .'"
51.                         data-
date_created="'. $ticket['ticket']['date_created'] .'"
52.                     >'. $ticket['ticket']['ticket_kode'] . '</td>
53.                     <td class="center status"><span class="bad-
ge-status yellow darken-1 grey-text text-darken-
4">Pending</span></td>
54.                 </tr>';
55.             }
56.
57.     public function tampilkan_jadwal()
58.     {
59.         $periode_id = $this->input->get('periode_id');
60.         $matkum_id = $this->input->get('matkum_id');
61.         $jadwal = $this->M_database-
>getAllJadwal($periode_id, $matkum_id);

62.
63.         echo '<span>Pindah ke Shift</span><table class="st-
ripped">';
64.         foreach ($jadwal['jadwal'] as $key => $value) {
65.             echo '<tr>
66.                     <td class="hide"></td>
67.                     <td><button class="mini-jadwal-kode shift-
tujuan" data-
jadwal_kode="'. $value['jadwal_kode'] .'">'. strtoupper($value[-
'jadwal_kode']) . '</button></td>
68.                     <td><span>'. $value['kelas_nama'] . '</span><
/td>
69.                     <td class="center"><span>'. $value['jadwal_-
hari'] . '<br>'. $value['jadwal_jam'] . '</span></td>
70.                 </tr>';
71.             }
72.             echo '</table>';
73.         }
74.
75.     function fetch_pesan()
76.     {
77.         $npm = $this->session->userdata('npm');
78.         $pesan = $this->M_database->getPesan($npm);

```

```

79.         echo '<a class="hide" href="#end-
content"><button class="to-
bottom></button></a>';
80.         foreach ($pesan['pesan'] as $key => $value) {
81.             $pos = "";
82.             $style = "";
83.             if ($value['pengirim'] == 'mahasiswa') {
84.                 $pos = "chat-text-right";
85.                 $color = "grey-text text-lighten-1";
86.             } else {
87.                 $pos = "chat-text-left";
88.                 $color = "grey-text text-darken-1";
89.             }
90.
91.             echo '<div class="'.$pos.' '.$value['pengirim']
92. ].'" style="'.$style.'"><span>'.$value['pesan'].'
93. <br><small class="right '.$color.'">' . $value['date_created'] . '</small><
94. /span></div>';
95.         }

```

## Tugas.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
');
3.
4. class Tugas extends CI_Controller {
5.
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('M_database');
10.        cek_session_null();
11.    }
12.
13.    public function index()
14.    {
15.        $npm = $this->session->userdata('npm');
16.        $data['tugas'] = $this->M_database-
>getTugas($npm);
17.        $this->template->load('template', 'v-
tugas', $data);
18.    }
19.
20.    public function detail()
21.    {
22.        $tugas_id = $this->input->get('idt');
23.        $tugas_kode = $this->input->get('tk');
24.        $data['tugas'] = $this->M_database-
>getTugasDetail($tugas_id, $tugas_kode);

```

```

25.         $this->template->load('template', 'v-tugas-
   detail', $data);
26.     }
27.
28.     public function upload()
29.     {
30.         $tugas_id = $this->input->post("tugas_id");
31.         $npm = $this->input->post("npm");
32.         $file_nama = $this->input->post("file_nama");
33.         $file = $_FILES['file']['name'];
34.
35.         $upload = $this->M_database-
           >upload_tugas($tugas_id, $npm, $file_nama, $file);
36.         $error_msg = $upload['message'];
37.         if ($upload['upload'] == true) {
38.             echo "berhasil";
39.         } else {
40.             echo "$error_msg";
41.         }
42.     }
43. }
```

### M\_database.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed
   ');
3.
4. use GuzzleHttp\Client;
5.
6. class M_database extends CI_Model {
7.
8.     private $_client;
9.
10.    public function __construct()
11.    {
12.        $this->_client = new Client([
13.            'base_uri' => 'http://localhost/adminsilabtif/
   api/'
14.        ]);
15.    }
16.
17.    public function getJadwal($npm)
18.    {
19.        $response = $this->_client-
           >request('GET', 'rest_jadwal', [
20.            'query' => [
21.                'npm' => $npm
22.            ]
23.        ]);
24.
25.        $result = json_decode($response->getBody(), true);
26.    }
```

```

27.         return $result;
28.     }
29.
30.     public function getTicket($npm)
31.     {
32.         $response = $this->_client-
33.             >request('GET', 'rest_ticketing', [
34.                 'query' => [
35.                     'npm' => $npm
36.                 ]
37.             ]);
38.         $result = json_decode($response->getBody(), true);
39.
40.         return $result;
41.     }
42.
43.     public function getAllJadwal($periode_id, $matkum_id)
44.     {
45.         $response = $this->_client-
46.             >request('GET', 'rest_jadwal/jadwal', [
47.                 'query' => [
48.                     'periode_id' => $periode_id,
49.                     'matkum_id' => $matkum_id
50.                 ]
51.             ]);
52.         $result = json_decode($response->getBody(), true);
53.
54.         return $result;
55.     }
56.
57.     public function getTugas($npm)
58.     {
59.         $response = $this->_client-
60.             >request('GET', 'rest_tugas', [
61.                 'query' => [
62.                     'npm' => $npm
63.                 ]
64.             ]);
65.         $result = json_decode($response->getBody(), true);
66.
67.         return $result;
68.     }
69.
70.     public function getTugasDetail($tugas_id, $tugas_kode)
71.     {
72.         $response = $this->_client-
73.             >request('GET', 'rest_tugas/detail', [
74.                 'query' => [
75.                     'tugas_id' => $tugas_id,
76.                     'tugas_kode' => $tugas_kode
77.                 ]
78.             ]);

```

```

79.         $result = json_decode($response->getBody(), true);
80.
81.         return $result;
82.     }
83.
84.     public function getMatkum($npm)
85.     {
86.         $response = $this->_client-
87.             >request('GET', 'rest_matkum', [
88.                 'query' => [
89.                     'npm' => $npm
90.                 ]
91.             ]);
92.
93.         $result = json_decode($response->getBody(), true);
94.
95.         return $result;
96.     }
97.
98.     public function getInfo()
99.     {
100.         $response = $this->_client-
101.             >request('GET', 'rest_info');
102.
103.         $result = json_decode($response->getBody(), true);
104.
105.
106.     public function getMahasiswa($mhs_id)
107.     {
108.         $response = $this->_client-
109.             >request('GET', 'rest_mahasiswa', [
110.                 'query' => [
111.                     'mhs_id' => $mhs_id
112.                 ]
113.             ]);
114.
115.         $result = json_decode($response->getBody(), true);
116.
117.         return $result;
118.     }
119.     public function getAbsen($npm, $jadwal_kode)
120.     {
121.         $response = $this->_client-
122.             >request('GET', 'rest_absensi', [
123.                 'query' => [
124.                     'npm' => $npm,
125.                     'jadwal_kode' => $jadwal_kode
126.                 ]
127.             ]);
128.
129.         $result = json_decode($response->getBody(), true);
130.         return $result;

```

```

131.     }
132.
133.     public function getNilai($npm, $jadwal_kode)
134.     {
135.         $response = $this->_client-
136.             >request('GET', 'rest_nilai', [
137.                 'query' => [
138.                     'npm' => $npm,
139.                     'jadwal_kode' => $jadwal_kode
140.                 ]
141.             ]);
142.         $result = json_decode($response->getBody(), true);
143.
144.         return $result;
145.     }
146.
147.     public function getPesan($npm)
148.     {
149.         $response = $this->_client-
150.             >request('GET', 'rest_ticketing/pesan', [
151.                 'query' => [
152.                     'npm' => $npm
153.                 ]
154.             ]);
155.         $result = json_decode($response->getBody(), true);
156.
157.         return $result;
158.     }
159.
160.     public function auth_user($npm, $password)
161.     {
162.
163.         $response = $this->_client-
164.             >request('POST', 'rest_mahasiswa/login', [
165.                 'form_params' => [
166.                     'npm' => $npm,
167.                     'password' => $password
168.                 ]
169.             ]);
170.         $result = json_decode($response->getBody(), true);
171.
172.         return $result;
173.     }
174.
175.     public function sendToken($npm, $token)
176.     {
177.
178.         $response = $this->_client-
179.             >request('POST', 'rest_mahasiswa/token', [
180.                 'form_params' => [
181.                     'npm' => $npm,
182.                     'token' => $token
183.                 ]
184.             ]);
185.         $result = json_decode($response->getBody(), true);
186.
187.         return $result;
188.     }
189.
```

```

183.         ]);
184.
185.         $result = json_decode($response->getBody(), true);
186.
187.         return $result;
188.     }
189.
190.     public function kritik_saran($ks)
191.     {
192.
193.         $response = $this->_client-
194.             >request('POST', 'rest_ticketing/kritiksaran', [
195.                 'form_params' => [
196.                     'kritik_saran' => $ks
197.                 ]
198.             ]);
199.
200.         $result = json_decode($response->getBody(), true);
201.
202.         return $result;
203.     }
204.     public function kirim_pesan($npm, $nama, $pesan)
205.     {
206.
207.         $response = $this->_client-
208.             >request('POST', 'rest_ticketing/kirimpesan', [
209.                 'form_params' => [
210.                     'npm' => $npm,
211.                     'nama' => $nama,
212.                     'pesan' => $pesan
213.                 ]
214.             ]);
215.
216.         $result = json_decode($response->getBody(), true);
217.
218.         return $result;
219.     }
220.     public function pindah_shift($npm, $nama, $shift_asal,
221.         $shift_tujuan, $deskripsi)
222.     {
223.
224.         $response = $this->_client-
225.             >request('POST', 'rest_ticketing/pindahshift', [
226.                 'form_params' => [
227.                     'npm' => $npm,
228.                     'nama' => $nama,
229.                     'shift_asal' => $shift_asal,
230.                     'shift_tujuan' => $shift_tujuan,
231.                     'deskripsi' => $deskripsi,
232.                 ]
233.             ]);
234.         $result = json_decode($response->getBody(), true);

```

```

235.         return $result;
236.     }
237.
238.     public function ganti_password($mhs_id, $old_pass, $ne
239.     {
240.
241.         $response = $this->_client-
242.             >request('POST', 'rest_mahasiswa/pass', [
243.                 'form_params' => [
244.                     'mhs_id' => $mhs_id,
245.                     'old_pass' => $old_pass,
246.                     'new_pass' => $new_pass
247.                 ]
248.             ]);
249.
250.         $result = json_decode($response->getBody(), true);
251.
252.         return $result;
253.
254.     public function upload_tugas($tugas_id, $npm, $file_na
255.     ma, $file)
256.     {
257.         $response = $this->_client-
258.             >request('POST', 'rest_tugas/upload', [
259.                 'multipart' => [
260.                     [
261.                         'name' => "file",
262.                         'contents' => fopen($_FILES['file']['tmp_n
263.                         ame'], 'r'),
264.                         'filename' => $_FILES['file']['name'],
265.                         'Content-type' => 'multipart/form-data'
266.                     ],
267.                     [
268.                         'name' => "file_nama",
269.                         'contents' => $file_nama
270.                     ],
271.                     [
272.                         'name' => "npm",
273.                         'contents' => $npm
274.                     ],
275.                     [
276.                         'name' => "tugas_id",
277.                         'contents' => $tugas_id
278.                     ]
279.                 ],
280.             ]);
281.
282.         $result = json_decode($response->getBody(), true);
283.
284.     }

```