

# **APLIKASI KEHADIRAN DOSEN MENGGUNAKAN *PHP OOP***

Roni Habibi

Ferdy Berliano Putra

Ida Fatrini Putri



## **APLIKASI KEHADIRAN DOSEN**

### **MENGGUNAKAN OOP PHP**

#### ***Penulis:***

Roni Habibi  
Ferdy Berliano Putra  
Ida Fatrini Putri

#### ***ISBN:***

#### ***Editor:***

Roni Habibi  
Ferdy Berliano Putra  
Ida Fatrini Putri

#### ***Penyunting:***

Roni Habibi

#### ***Desain Sampul dan Tata Letak:***

Ferdy Berliano Putra  
Ida Fatrini Putri

#### ***Penerbit:***

Kreatif Industri Nusantara

#### ***Redaksi:***

Jl. Ligar Nyawang No.2  
Bandung 40191  
Telp. 022 2045-8529  
Email : irc@poltekpos.ac.id

#### ***Distributor:***

Informatics Research Center  
Jl. Sariyah No. 54  
Bandung 40151  
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2020

Hak cipta dilindungi undang-undang  
Dilarang memperbanyak karya tulis ini dalam  
bentuk dan dengan cara apapun tanpa ijin  
tertulis dari penerbit

# KATA PENGANTAR

---

Assalamualaikum warahmatullahi wabarakatuh. Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga dapat menyelesaikan buku Aplikasi Proyek II ini, tanpa pertolongan-Nya mungkin penulis tidak akan sanggup menyelesaikannya dengan baik. Shalawat dan salam semoga terlimpah curahkan kepada Nabi Muhammad SAW beserta sahabat dan keluarga Beliau.

Buku ini disusun untuk memenuhi kelulusan matakuliah Proyek II pada Program Studi DIV Teknik Informatika. Proses Proyek II ini juga tidak terlepas dari bantuan pihak Pembimbing. Oleh karena itu, pada kata pengantar ini penulis menyampaikan terimakasih kepada :

1. Roni Habibi, S.Kom., M.T. selaku Pembimbing dalam penyusunan buki ini;
2. Rolly Maulana Awangga, S.T., M.T. selaku Koordinator Proyek II Tahun Akademik 2019/2020;
3. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika Tahun Akademik 2019/2020;

4. Dr. Ir. Agus Purnomo, M.T. selaku Direktur Politeknik Pos Indonesia Tahun Akademik 2019/2020.

Penulis telah membuat buku ini dengan sebaik-baiknya, diharapkan memberikan kritik dan saran dari semua pihak yang bersifat membangun, terimakasih.

# **DAFTAR ISI**

---

KATA PENGANTAR .....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR .....	x
DAFTAR TABEL.....	xiv
APA SAJA ISI DARI BUKU INI?.....	xv
BAB I .....	1
PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Identifikasi Masalah .....	2
1.3    Tujuan dan Manfaat.....	3
1.4    Ruang Lingkup .....	3
BAB II.....	4
APLIKASI YANG DIGUNAKAN.....	4
2. 1    Pengertian Aplikasi .....	4
2. 2 <i>XAMPP</i> .....	5
2.2.1    Pengertian Aplikasi <i>XAMPP</i> .....	5
2.2.2    Fungsi <i>XAMPP</i> .....	6
2.2.3    Sejarah Singkat <i>XAMPP</i> .....	6
2.2.4    Cara Meng- <i>install</i> <i>XAMPP</i> .....	6
2. 3 <i>Sublime Text</i> .....	9

2.3.1	Pengertian <i>Sublime Text</i> .....	9
2.3.2	Fungsi <i>Sublime Text</i> .....	10
2.3.3	Keunggulan <i>Sublime Text</i> .....	11
2. 4	<i>Google Chrome</i> .....	11
2.4.1	Pengertian <i>Google Chrome</i> .....	11
2.4.2	Sejarah <i>Google Chrome</i> .....	12
2.4.3	Kelebihan dan Kekurangan <i>Google Chrome</i> .....	13
2.4.4	Fitur <i>Google Chrome</i> .....	13
2.4.5	Cara Download <i>Google Chrome</i> .....	16
2. 5	<i>Star UML</i> .....	17
2.5.1	Pengertian <i>Star UML</i> .....	17
2.5.2	Cara Download <i>Star UML</i> .....	18
2.5.3	Fitur <i>Star UML</i> .....	19
2. 6	<i>Enterprise Architect</i> .....	21
2.6.1	Pengertian <i>Enterprise Architect</i> .....	21
2.6.2	Keunggulan Menggunakan <i>Enterprise Architect</i> .....	22
2.6.3	<i>User Interface Enterprise Architect</i> .....	27
2.6.4	Cara Download <i>Enterprise Architect</i> .....	28
BAB III .....		33
BAHASA PEMROGRAMAN.....		33
3.1	Pengertian Bahasa Pemrograman .....	33
3.2	Fungsi Pemrograman.....	34
3.3	Tingkatan Bahasa Pemrograman .....	35
3.4	Struktur Dasar Bahasa Pemrograman.....	36
3.4.1	Struktur Runtutan .....	36
3.4.2	Struktur Perulangan.....	36
3.4.3	Struktur Percabangan .....	38
3.5	Jenis Bahasa Pemrograman yang dipakai.....	39
3.5.1	Pengertian <i>PHP: Hypertext Preprocessor</i> .....	39

3.5.2	Sejarah <i>PHP</i> .....	40
3.5.3	Keunggulan Bahasa Pemrograman <i>PHP</i> .....	41
3.5.4	Dasar Pemrograman <i>PHP</i> .....	42
3.6	<i>OOP (Object Oriented Programming)</i> .....	43
BAB IV .....		46
PERANCANGAN DIAGRAM UML .....		46
4.1	Pengertian <i>UML</i> .....	46
4.2	Sejarah <i>UML</i> .....	47
4.3	Konsepsi Dasar <i>UML</i> .....	49
4.4	Diagram <i>UML</i> .....	50
4.4.1	<i>Use Case Diagram</i> .....	50
4.4.2	<i>Class Diagram</i> .....	59
4.4.3	<i>Activity Diagram</i> .....	62
4.4.4	<i>Sequence Diagram</i> .....	68
4.4.5	<i>Collaboration Diagram</i> .....	72
4.4.6	<i>Statechart Diagram</i> .....	77
4.4.7	<i>Component Diagram</i> .....	82
4.4.8	<i>Deployment Diagram</i> .....	84
4.4.9	Object Diagram .....	85
4.4.10	Struktur Menu .....	87
BAB V.....		92
TAHAPAN MEMBUAT APLIKASI KEHADIRAN DOSEN .....		92
5.1	Membuat <i>Database</i> .....	92
5.1.1	Tabel <i>User Dosen</i> .....	93
5.1.2	Tabel <i>User Mahasiswa</i> .....	94
5.1.3	Tabel <i>User Admin</i> .....	95
5.1.4	Tabel Jadwal Semester Ganjil 2018/2019 .....	97
5.1.5	Tabel Jadwal Semester Genap 2018/2019 .....	98
5.1.6	Tabel Jadwal Semester Ganjil 2019/2020.....	99

5.1.7	Tabel Jadwal Semester Genap 2019/2020 .....	101
5.2	Membuat <i>Controller</i> .....	102
5.2.1	<i>Controller Admin</i> .....	102
5.2.2	<i>Controller Dosen</i> .....	120
5.2.3	<i>Controller Mahasiswa</i> .....	135
5.2.4	<i>Controller File adminnnnn.php</i> .....	138
5.2.5	<i>Controller File Login.php</i> .....	139
5.2.6	<i>Controller File logindosen.php</i> .....	141
5.2.7	<i>Controller File loginmahasiswa.php</i> .....	143
5.2.8	<i>Controller File Welcome.php</i> .....	145
5.3	Membuat <i>Model</i> .....	146
5.3.1	File <i>D_login.php</i> .....	146
5.3.2	File <i>Dosen_model.php</i> .....	146
5.3.3	File <i>Ganjil1819_model.php</i> .....	149
5.3.4	File <i>Ganjil1920_mode.php</i> .....	154
5.3.5	File <i>Genap1819_model.php</i> .....	158
5.3.6	File <i>Hadir_model.php</i> .....	162
5.3.7	File <i>Jadwal_model.php</i> .....	167
5.3.8	File <i>Jadwalaja.php</i> .....	171
5.3.9	File <i>Kehadiran_model.php</i> .....	176
5.3.10	File <i>M_login.php</i> File.....	180
5.3.11	File <i>Mahasiswa_model.php</i> .....	181
5.3.12	File <i>W_Login</i> .....	184
5.4	Membuat <i>View</i> .....	184
5.4.1	<i>View Admin</i> .....	184
BAB VI	.....	199
IMPLEMENTASI	.....	199
6.1	<i>User Interface Admin</i> .....	200
6.1.1	Tampilan <i>Login Admin</i> .....	200

6.1.2	Tampilan Menu Jadwal .....	201
6.1.3	Tampilan Menambahkan Jadwal pada Menu Jadwal...	202
6.1.4	Tampilan Merubah Jadwal pada Menu Jadwal.....	203
6.1.5	Tampilan Menu Mahasiswa .....	204
6.1.6	Tampilan Menambahkan Mahasiswa pada Menu Mahasiswa.....	205
6.1.7	Tampilan Merubah Data Mahasiswa pada Menu Mahasiswa.....	206
6.1.8	Tampilan Menu Dosen.....	206
6.1.9	Tampilan Menambahkan Dosen pada Menu Dosen ....	207
6.1.10	Tampilan Merubah pada Menu Dosen.....	208
6.1.11	Tampilan Menghapus Data .....	208
6.2	<i>User Interface</i> Dosen.....	209
6.2.1	Tampilan Login Dosen.....	209
6.2.2	Tampilan Menu Dosen.....	210
6.2.3	Tampilan Menu Edit Dosen .....	210
6.3	<i>User Interface</i> Mahasiswa.....	211
6.3.1	Tampilan <i>Login</i> Mahasiswa .....	211
6.3.2	Tampilan Menu Mahasiswa .....	212
6.3.3	Tampilan Penggunaan Fitur <i>Searching</i> .....	212
	DAFTAR PUSTAKA .....	214

# DAFTAR GAMBAR

---

Gambar 2. 1 XAMPP.....	5
Gambar 2. 2 Install XAMPP (1).....	7
Gambar 2. 3 Install XAMPP (2).....	8
Gambar 2. 4 Install XAMPP (3).....	8
Gambar 2. 5 Sublime Text .....	10
Gambar 2. 6 Google Chrome .....	12
Gambar 2. 7 Fitur Game Dinasaurus.....	14
Gambar 2. 8 Manage Google Account.....	15
Gambar 2. 9 Pin Tab .....	15
Gambar 2. 10 Unduh Google Chrome .....	16
Gambar 2. 11 Setuju dan Pasang Google Chrome .....	17
Gambar 2. 12 Mengunduh Google Chrome .....	17
Gambar 2. 13 Download Star UML .....	18
Gambar 2. 14 Instalasi Star UML .....	19
Gambar 2. 15 Diagram pada Star UML .....	20
Gambar 2. 16 Tool Editing.....	21
Gambar 2. 17 Enterprise Architect .....	22
Gambar 2. 18 Model dan Mengelola Informasi Lebih Kompleks .....	23
Gambar 2. 19 Model, Manage, dan Trace Requirements .....	23
Gambar 2. 20 Integrasikan Tim dan Bagikan Visi.....	24

Gambar 2. 21 Desain dan Bangun Sistem Beragam menggunakan <i>UML</i> .....	25
Gambar 2. 22 Visualisasikan, Periksa dan Pahami Perangkat Lunak Kompleks .....	25
Gambar 2. 23 Menggunakan <i>Life-Cycle Modeling</i> dan <i>Project Management</i> .....	26
Gambar 2. 24 Berbagi dan Memakai Ulang <i>Information Across Tools</i>	27
Gambar 2. 25 <i>User Interface Enterprise Architect</i> .....	28
Gambar 2. 26 <i>Website Enterprise Architect</i> .....	29
Gambar 2. 27 <i>Install Enterprise Architect</i> (1).....	29
Gambar 2. 28 <i>Install Enterprise Architect</i> (2) .....	30
Gambar 2. 29 <i>Install Enterprise Architect</i> (3) .....	30
Gambar 2. 30 <i>Install Enterprise Architect</i> (4) .....	31
Gambar 2. 31 <i>Install Enterprise Architect</i> (5) .....	31
Gambar 2. 32 <i>Install Enterprise Architect</i> (6) .....	32
Gambar 2. 33 <i>Install Ensterprise Architect</i> (7).....	32
Gambar 3. 1 Bahasa Pemrograman.....	34
Gambar 3. 2 Contoh Struktur Perulangan .....	37
Gambar 3. 3 Contoh Struktur Percabangan.....	38
Gambar 3. 4 <i>PHP: Hypertext Preprocessor</i> .....	40
Gambar 3. 5 <i>Object Oriented Programming</i> .....	44
Gambar 3. 6 <i>OOP</i> .....	44
Gambar 4. 1 <i>Class Diagram</i> (1).....	59
Gambar 4. 2 <i>Class Diagram</i> (2).....	60
Gambar 4. 3 <i>Class Diagram</i> (3).....	61

Gambar 4. 4 Contoh <i>Class Diagram</i> .....	61
Gambar 4. 5 <i>Activity Diagram Login</i> .....	63
Gambar 4. 6 <i>Activity Diagram Logout</i> .....	64
Gambar 4. 7 <i>Activity Diagram Add User</i> .....	65
Gambar 4. 8 <i>Activity Diagram Mengelola Jadwal</i> .....	66
Gambar 4. 9 <i>Sequence Diagram Login</i> .....	69
Gambar 4. 10 <i>Sequence Diagram Logout</i> .....	70
Gambar 4. 11 <i>Sequence Diagram Add User</i> .....	70
Gambar 4. 12 <i>Sequence Diagram Mengelola Jadwal</i> .....	71
Gambar 4. 13 <i>Sequence Diagram Mengupdate Informasi Kehadiran</i> .	72
Gambar 4. 14 <i>Collaboration Diagram Login</i> .....	73
Gambar 4. 15 <i>Collaboration Diagram Logout</i> .....	74
Gambar 4. 16 <i>Collaboration Diagram Mengelola Jadwal</i> .....	75
Gambar 4. 17 <i>Sequence Diagram Add User</i> .....	76
Gambar 5. 1 Membuat <i>Database hadir_dosen</i> .....	92
Gambar 5. 2 Tabel <i>User Dosen</i> .....	93
Gambar 5. 3 Tabel <i>User Mahasiswa</i> .....	94
Gambar 5. 4 Tabel <i>User Admin</i> .....	96
Gambar 5. 5 Tabel Jadwal Semester Ganjil 2018/2019 .....	97
Gambar 5. 6 Tabel Jadwal Semester Genap 2018/2019 .....	98
Gambar 5. 7 Tabel Jadwal Semester Ganjil 2019/2020.....	100
Gambar 5. 8 Tabel Jadwal Semester Genap 2019/2020 .....	101
Gambar 6. 1 Tampilan <i>Login Admin</i> .....	200
Gambar 6. 2 Menu Jadwal .....	201
Gambar 6. 3 Menambahkan Jadwal .....	202

Gambar 6. 4 Mengubah Jadwal.....	203
Gambar 6. 5 Menu Mahasiswa.....	204
Gambar 6. 6 Menambahkan Mahasiswa .....	205
Gambar 6. 7 Merubah Data Mahasiswa.....	206
Gambar 6. 8 Menu Dosen .....	206
Gambar 6. 9 <i>Login</i> Mahasiswa.....	211
Gambar 6. 10 Menu Mahasiswa .....	212
Gambar 6. 11 Fitur <i>Searching</i> .....	212

# **DAFTAR TABEL**

---

Tabel 4: 1 Definisi Aktor .....	51
Tabel 4: 2 Definisi <i>Use Case</i> .....	52
Tabel 4: 3 Skenario <i>Use Case Login</i> .....	53
Tabel 4: 4 Skenario <i>Use Case Mengelola Jadwal</i> .....	54
Tabel 4: 5 Skenario <i>Use Case Add User</i> .....	56
Tabel 4: 6 Skenario <i>Use Case Meng-Update Kehadiran Dosen</i> .....	57

# **APA SAJA ISI DARI BUKU INI?**

---

Buku ini akan menjelaskan tahapan pembuatan Aplikasi Kehadiran Dosen. Tidak hanya itu saja, buku ini juga akan memberikan materi terkait aplikasi yang dirancang. Mulai dari merancang diagram, *database* yang diperlukan, *codingan* yang ada pada aplikasi akan kami tuangkan dalam buku ini.

Materi yang ada di buku ini merupakan materi yang penting sebagaimana berhubungan dengan aplikasi yang kami rancang.

# BAB I

## PENDAHULUAN

---

### 1.1 Latar Belakang

Pada zaman modern ini, teknologi sudah sangat berkembang. Salah satunya adalah pengembangan pada aplikasi. Kebanyakan aplikasi saat ini sangat membantu dan mempermudah masyarakat. Salah satu model aplikasi yang membantu saat ini adalah aplikasi pengolah data. Sudah banyak yang membuat aplikasi seperti ini, salah satunya adalah aplikasi *Microsoft Word* merupakan perangkat lunak yang berfungsi untuk mengolah dokumen berupa teks misalnya jurnal, karya ilmiah, laporan, dll. Banyak juga perangkat lunak atau software yang berfokus pada pengolah data dalam ruang lingkup yang kecil.

Adanya pengembangan aplikasi pengolah data dapat mempermudah mengolah data pada suatu organisasi atau perusahaan. Keuntungan menggunakan aplikasi pengolah data *user friendly* (mudah digunakan) dan tingkat keamanan juga cukup valid untuk setiap *user*. Aplikasi mengelolah data ini sangat berguna untuk

perusahaan atau organisasi untuk menyimpan data-data yang penting. Salah satunya mengelolah data kehadiran dosen di kampus. Dengan adanya aplikasi ini akan lebih memudahkan mahasiswa dalam melihat kehadiran dosen. Selain mahasiswa yang mendapat kemudahan, dosen juga mudah untuk memberitahu info kehadiran kepada mahasiswa

Seperti yang kita ketahui banyak dampak positif dari aplikasi pengolah data. Pada aplikasi ini membawa dampak positif bagi mahasiswa, dosen, maupun pihak kampus lainnya. Salah satu dampak positifnya adalah dosen atau mahasiswa dapat tepat waktu menghadiri perkuliahan karna adanya aplikasi ini. Dosen akan mendapat mengecek aplikasi untuk melihat mengajar perkuliahan mahasiswa, sedangkan mahasiswa mengecek aplikasi untuk melihat jadwal perkuliahan yang akan datang.

Politenik Pos Indonesia adalah salah satu kampus yang berkembang saat ini pada bidang logistik dan teknologi. Pada bidang teknologi berupa alat dan aplikasi. Sehinnga penulis berinisiatif untuk membuat aplikasi yang berjudul *Aplikasi Kehadiran Dosen* dengan harapan aplikasi ini dapat bermanfaat bagi para dosen dan mahasiswa di kampus Politeknik Pos Indonesia.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang ada, maka dapat diidentifikasi menjadi beberapa masalah sebagai berikut:

1. Banyaknya mahasiswa yang meminta informasi melalui *chatting* pribadi

### **1.3 Tujuan dan Manfaat**

Berdasarkan latar belakang dan identifikasi masalah yang ada, maka akan memberikan tujuan dan manfaat sebagai berikut:

1. Mempermudah komunikasi baik mahasiswa, maupun dosen.

### **1.4 Ruang Lingkup**

Dari data – data yang didapatkan dari proses identifikasi masalah ini maka ruang lingkup dari penelitian ini adalah sebagai berikut:

1. Info kehadiran dari Aplikasi Kehadiran Dosen
2. Mengelolah data jadwal perkuliahan mahasiswa
3. Mengelolah data jadwal mengajar dosen.
4. Mengelola *user* mahasiswa
5. Mengelola *user* dosen

## **BAB II**

# **APLIKASI YANG DIGUNAKAN**

---

### **2. 1 Pengertian Aplikasi**

Menurut *Sutabri* (2012:147), aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya. Menurut *Asropudin* (2013:6), aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Ms. Word*, *Ms. Excel*. [4]

Dari pengertian diatas disimpulkan bahwa aplikasi adalah suatu program komputer yang bertujuan untuk mengerjakan tugas dari *user*. Saat merancang aplikasi, ada beberapa *software* atau perangkat lunak pendukung untuk membuat aplikasi tersebut. Berikut ini adalah beberapa penjelasan mengenai *software* pendukung untuk membuat Aplikasi Kehadiran Dosen.

## 2. 2 XAMPP

### 2.2.1 Pengertian Aplikasi XAMPP

XAMPP merupakan perangkat lunak bebas, yang mendukung banyak *system* operasi merupakan kompilasi dari beberapa program, *xampp* adalah perangkat yang menggabungkan tiga aplikasi ke dalam satu paket yaitu *Apache MySQL*, 6 dan *PHP my admin* dengan *xampp* pekerjaan anda sangat dimudahkan karena dapat menginstalasi dan mengkonfigurasi ketiga aplikasi tersebut dengan sekaligus dan otomatis. XAMPP merupakan salah satu paket installasi *apache,php* dan *MySQL* instan yang dapat kita gunakan untuk membantu proses instalasi ketiga tersebut selain paket instalasi instant *xampp* untuk berpindah versi juga memberikan fasilitasi pilihan penggunaan php. [1]



Gambar 2. 1 XAMPP

Pada pembuatan aplikasi ini, XAMPP ini merupakan *software* pendukung yang sangat penting. Itu karena tanpa memakai XAMPP, file aplikasi yang memiliki format PHP tidak bisa terbaca. Format PHP itu

adalah mengidentifikasi Bahasa pemrograman yang dipakai. Untuk pembahasan Bahasa pemrograman akan kami bahas di bab selanjutnya.

### **2.2.2 Fungsi XAMPP**

Fungsi XAMPP sendiri adalah sebagai server yang berdiri (*localhost*) yang terdiri dari beberapa program antara lain *Apache*, *MySQL database* dan penerjemah bahasa yang di tulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama XAMPP sendiri merupakan singkatan dari *x* , program ini tersedia dalam *GNU (general public license)* dan bebas merupakan web server yang mudah untuk digunakan[1].

Sebenarnya fungsi dari XAMPP sudah dijelaskan pada pengertian XAMPP, yaitu karena tanpa memakai XAMPP, file aplikasi yang memiliki format *PHP* tidak bisa terbaca. Format *PHP* itu adalah mengidentifikasi Bahasa pemrograman yang dipakai.

### **2.2.3 Sejarah Singkat XAMPP**

XAMPP merupakan pengembangan dari *LAMPP* ,ini merupakan project *non-profit* yang di kembangkan oleh *apache frinds* yang di dirikan *Oswalad seidler dank ay Vogelgesang* pada tahun 2002 , project mereka bertujuan mempromosikan penggunaan *apache web server*[1].

### **2.2.4 Cara Meng-*install* XAMPP**

Pada umumnya untuk meng-*install* *PHP* sebaiknya mengetahui *hardware* yang dibutuhkan untuk XAMPP. Berikut ini adalah spesifikasi yang dibutuhkan.

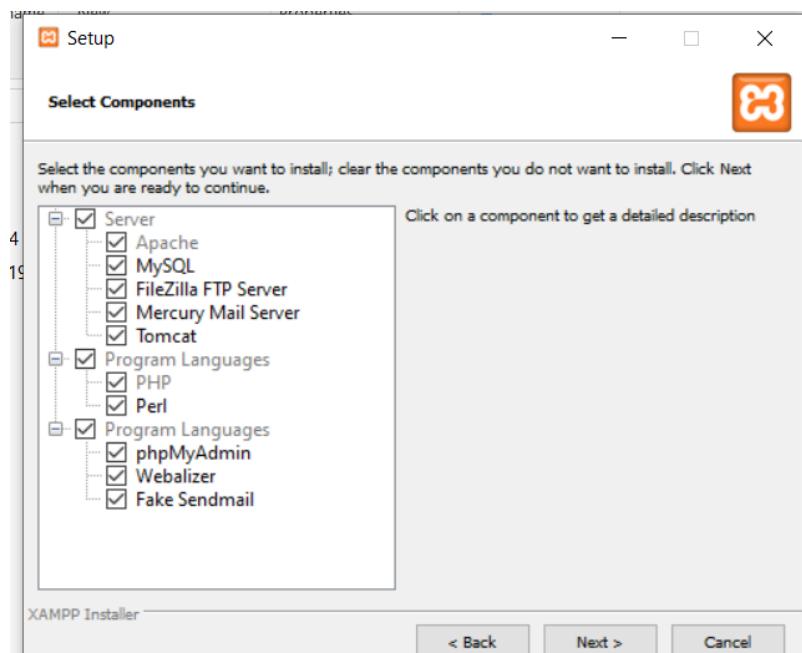
1. Sistem operasi Windows XP, Windows 7, Windows 8, atau Windows 10
2. Kapasitas *harddisk* minimal 250 mb

Setelah *hardware* laptop memenuhi syarat untuk meng-*install* XAMPP, berikut ini adalah cara instalasi XAMPP.

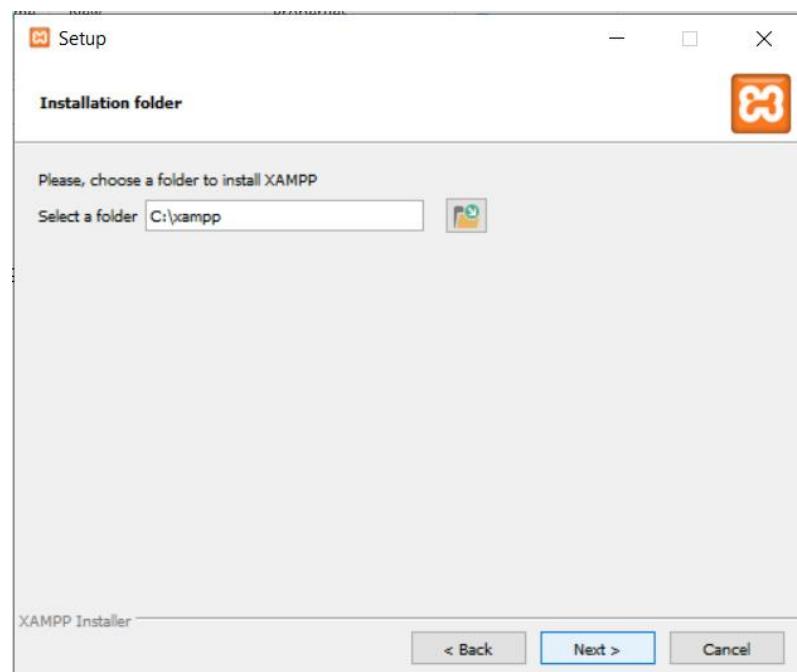
1. *Download* terlebih dahulu *software* XAMPP pada link berikut ini.

<https://www.apachefriends.org/download.html>

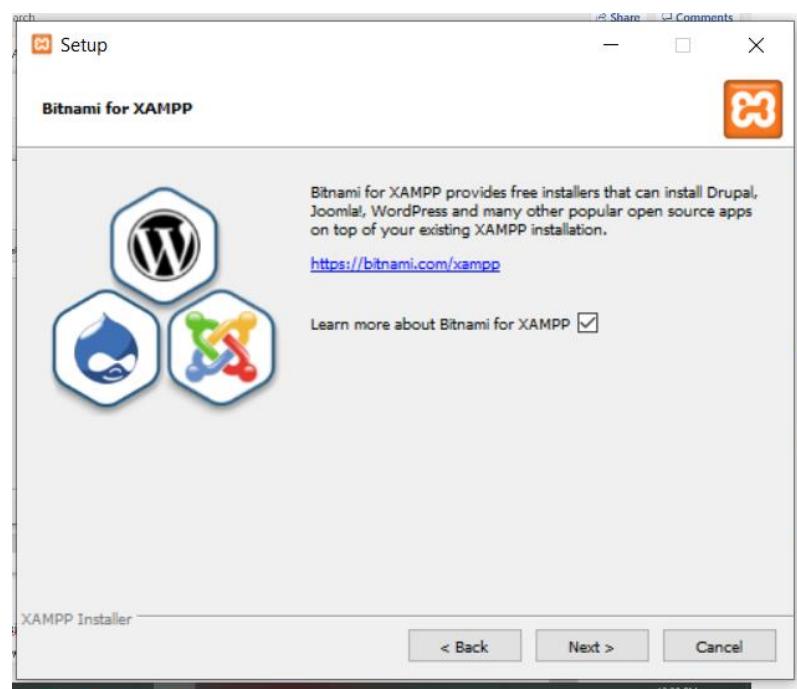
2. Setelah XAMPP terunduh, lakukan instalasi dimulai dari mengklik *double* file XAMPP tadi. Tampilannya akan seperti gambar berikut ini.



Gambar 2. 2 Install XAMPP (1)



Gambar 2. 3 Install XAMPP (2)



Gambar 2. 4 Install XAMPP (3)

8 | II Aplikasi yang digunakan

Karena sebelumnya kami sudah meng-*install XAMPP*, kami tidak bisa mengklik next. Pada tombol next di gambar 2.4 akan menuju instalasi *XAMPP*.

## 2.3 *Sublime Text*

### 2.3.1 Pengertian *Sublime Text*

*Faridl (2015:3), "Sublime Text 3 adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya fitur, cross platform, mudah dan simpel yang cukup terkenal di kalangan developer (pengembang), penulis dan desainer. Para programmer biasanya menggunakan sublime text untuk menyunting source code yang sedang ia kerjakan. Sampai saat ini sublime text sudah mencapai versi 3". "Sublime text 3 merupakan perangkat lunak text editor yang digunakan untuk membuat atau meng-edit suatu aplikasi" (Supono dan Putratama 2018:14). Jadi, dapat penulis simpulkan bahwa Sublime Text 3 adalah teks editor yang digunakan untuk meng-edit aplikasi.* [2]



Gambar 2. 5 Sublime Text

Menurut *Eric Haughee* (2013), bahwa *Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai *platform operating system* dengan menggunakan teknologi *Python API*. Aplikasi ini sangatlah fleksibel dan *powerfull*. [3]

### 2.3.2 Fungsi *Sublime Text*

Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *open source*, yang artinya aplikasi ini membutuhkan lisensi (*license*) yang harus dibeli. Akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi (*license*) aplikasi gratis. *Sublime Text* mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas.

### **2.3.3 Keunggulan *Sublime Text***

Banyak keunggulan yang dimiliki oleh *Sublime Text* ini, dan berikut ini kami jelaskan beberapa keunggulannya.

1. *Software* yang ringan dan tidak banyak memakan RAM pada komputer kita saat menggunakannya.
2. *Support Platform Operation System*, seperti *Windows*, dan yang lainnya.
3. *Mini Map* padadisisi kanan atasnya untuk mempermudahkan kita dalam menemukan kode maupun penulisan kode.
4. Mengganti *background* satu ini jika kalian bosan dengan tampilan warnanya yaitu hitam dengan cara pilih menu *preferences* pada menu bar dan pilih *color scheme*.

## **2.4 *Google Chrome***

### **2.4.1 Pengertian *Google Chrome***

*Google Chrome* adalah sebuah penjelajah web sumber terbuka yang dikembangkan oleh *Google* dengan menggunakan mesin *rendering Webkit*. Proyek sumber terbukanya sendiri dinamakan *Chromium*[5].



Gambar 2. 6 Google Chrome

*Google Chrome* merupakan browser yang dikeluarkan oleh *Google*, sebuah perusahaan *search engine* terkemuka di dunia. *Google Chrome* juga dirancang agar berjalan secepat mungkin. Cepat dimulai dari desktop, memuat halaman web dalam sekejap, dan menjalankan aplikasi web yang rumit dengan sangat cepat[6].

*Google Chrome* merupakan browser yang mudah digunakan dan biasanya sudah tersedia pada laptop kita.

#### 2.4.2 Sejarah *Google Chrome*

*Google Chrome* pertama kali di rilis oleh *Google* pada tanggal 2 September 2008. Saat itu *Chrome* hanya untuk *Microsoft Windows* karena masih dalam status beta. Lalu, pada 11 Desember di tahun yang sama *Google Chrome* diluncurkan untuk semua sistem operasi karena telah mencapai versi stabil. Selanjutnya pada bulan Januari 2012, *Google Chrome* diperkirakan telah berhasil meraih presentase 25-28% dari keseluruhan pengguna browser dunia, membuatnya sebagai browser

kedua atau juga ketiga paling banyak di gunakan setelah *Mozilla Firefox*[6].

#### **2.4.3 Kelebihan dan Kekurangan *Google Chrome***

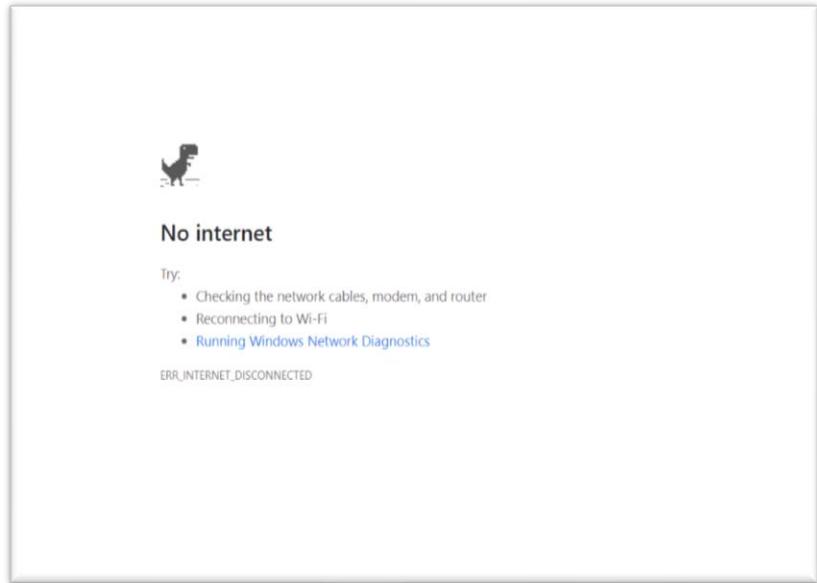
Browser ini harus di *install* dengan menggunakan koneksi internet, karena file instalasinya yang hanya sebesar kurang lebih 600 KB memerlukan *download* file-file yang diperlukan langsung ke situs *Google*. Berikut ini adalah kelebihan dan kekurangan dari *Google Chrome*[7].

- Kelebihan:
  1. *Bug*-nya sulit dieksplorasi
  2. Browser yang paling ringan
  3. *Design simple* dan minimalis
  4. *Update*-nya mudah
  5. *Security* atau keamanannya paling kuat, menjadi browser paling tahan terhadap serangan *hacker*
- Kekurangan:
  1. Fitur yang kurang
  2. Saat melakukan installnya harus ada koneksi internet

#### **2.4.4 Fitur *Google Chrome***

Google Chrome merupakan salah satu web browser yang memiliki market share pengguna terbesar yaitu berkisar antara 44%. Sebagai web browser dengan market share terbesar pastinya Google Chrome telah dilengkapi dengan fitur dan desain yang sangat memanjakan penggunanya. Hal ini berbanding lurus dengan kepuasan dirasakan oleh pengguna Google Chrome[8]. Berikut ini adalah beberapa fitur *Google Chrome* yang harus diketahui.

## **1. Game Dinasaurus**



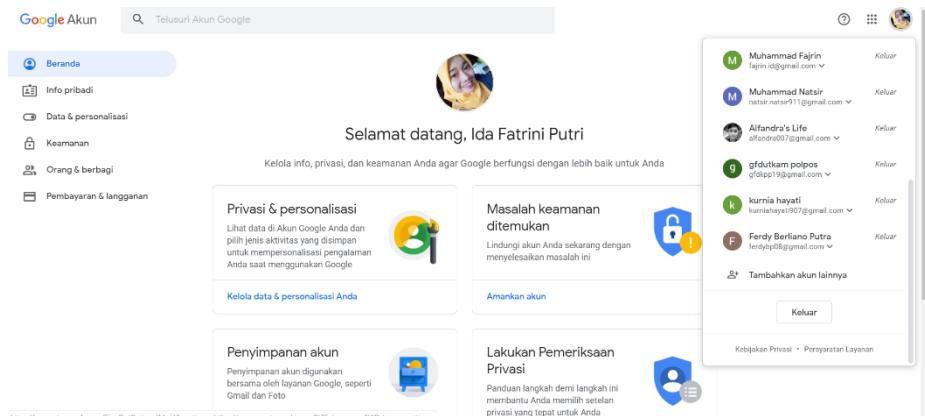
*Gambar 2. 7 Fitur Game Dinasaurus*

*Game* ini dapat dimainkan jika koneksi internet computer kita tidak ada. Fitur ini cocok bila kamu merasa bosan.

## **2. Manage Google Account**

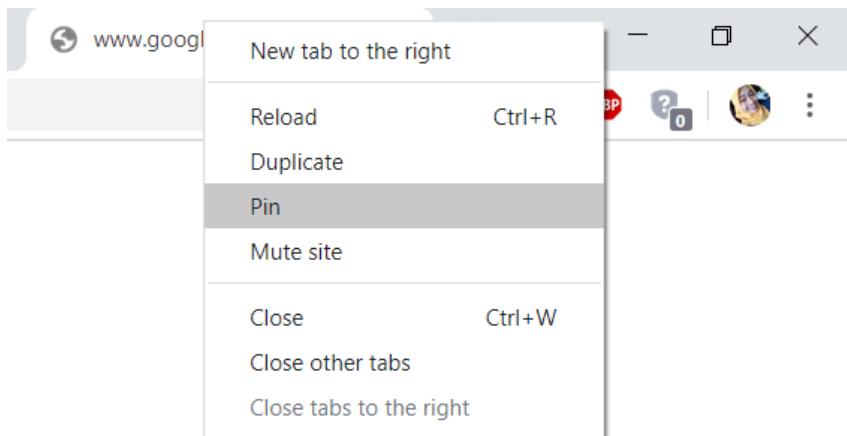
Fitur *manage google account* berguna untuk membuat beberapa akun pengguna yang menggunakan *Google Chrome*. Kamu dapat menyimpan *bookmark* dan *history* *Google Chrome* sesuai akun pengguna.

Cara menggunakannya adalah buka *Setting > Manage Your Google Account > Add More Account*.



Gambar 2. 8 Manage Google Account

### 3. Pin Tab



Gambar 2. 9 Pin Tab

Fungsi dari fitur pin adalah untuk menandakan halaman yang tidak ingin di *close*. Cara menggunakannya dengan mengklik kanan pada *tab* halaman dan pilih pin. *Tab* akan otomatis pindah ke kiri.

#### **2.4.5 Cara Download Google Chrome**

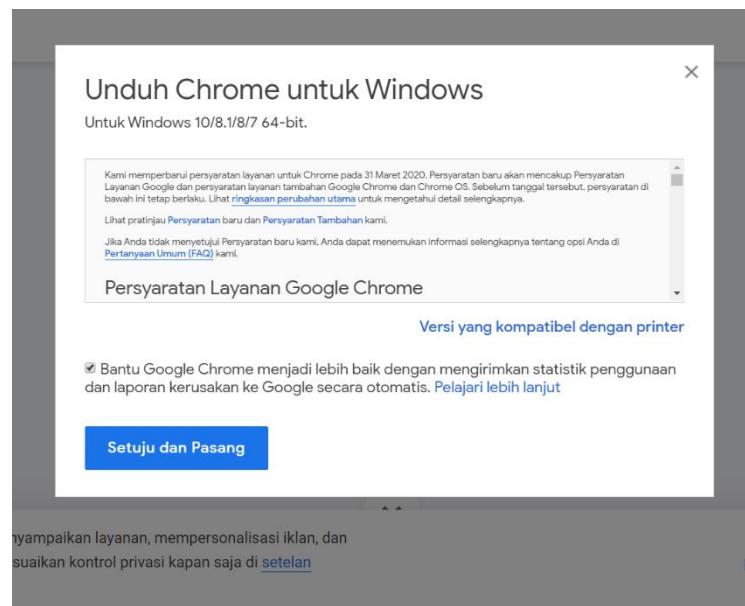
Berikut ini adalah langkah-langkah untuk men-download Aplikasi *Google Chrome*.

1. Pergi ke halaman <https://www.google.com/chrome/>
2. Klik unduh *Chrome*



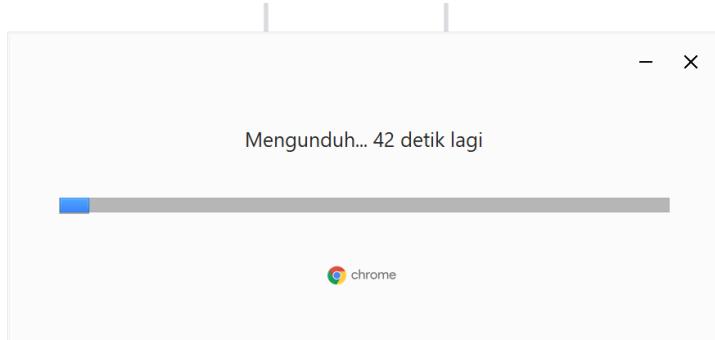
*Gambar 2. 10 Unduh Google Chrome*

3. Klik setuju dan pasang *Google Chrome*



Gambar 2. 11 Setuju dan Pasang Google Chrome

4. Tunggu proses pengunduhan dan lakukan instalasi.



Gambar 2. 12 Mengunduh Google Chrome

## 2. 5 Star UML

### 2.5.1 Pengertian Star UML

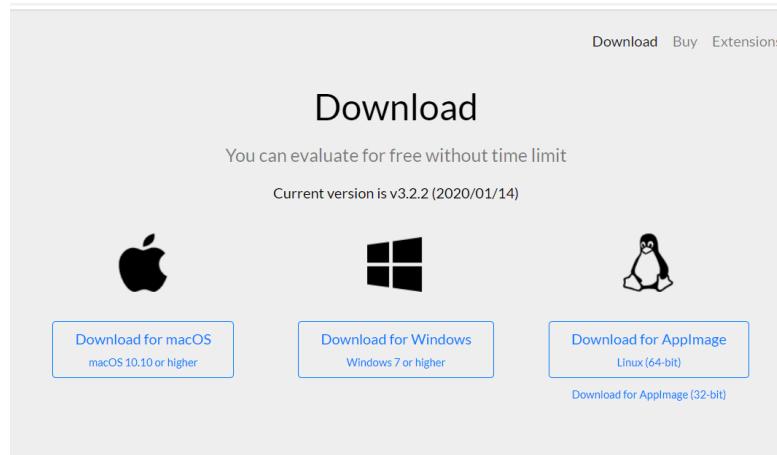
*StarUML* adalah *software* permodelan yang mendukung *UML* (*Unified Modeling Language*). Berdasarkan pada *UML version 1.4* dan dilengkapi 11 macam diagram yang berbeda, mendukung notasi *UML*

2.0 dan juga mendukung pendekatan *MDA* (*Model Driven Architecture*) dengan dukungan konsep *UML*. *Star UML* dapat memaksimalkan produktivitas dan kualitas dari suatu *software project* (Bayu Aji, 2012)[10].

### 2.5.2 Cara *Download Star UML*

Berikut ini adalah langkah-langkah untuk men-*download* *StarUML*.

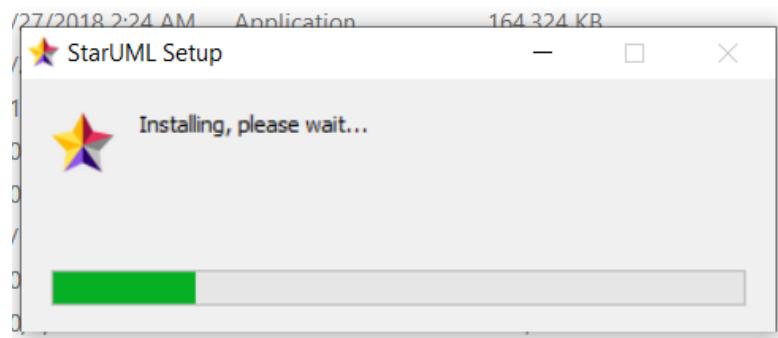
1. Pergi ke *website* resmi *Star UML* <http://staruml.io/download>
2. Klik *download* sesuai sistem operasi yang sesuai dengan komputermu.



### Release Notes

Gambar 2. 13 Download *Star UML*

3. Setelah *software* terunduh, lakukan instalasi seperti biasa.



Gambar 2. 14 Instalasi Star UML

### 2.5.3 Fitur *Star UML*

Pada aplikasi *Star UML* terdapat beberapa fitur yang bisa dipakai, berikut ini adalah fitur yang dapat dilakukan pada aplikasi ini.

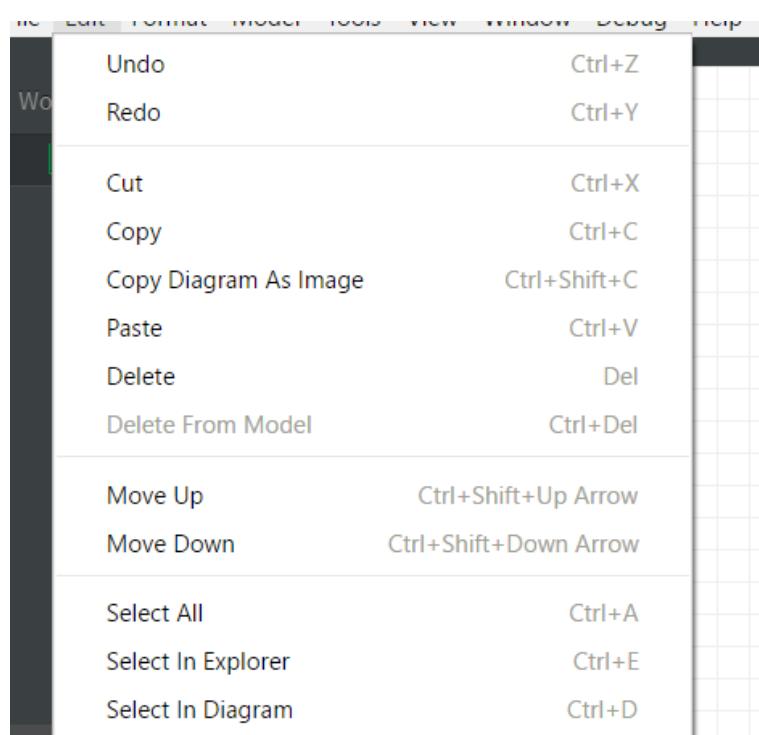
1. Dapat membuat Diagram yang dibutuhkan

Diagram yang dimaksud adalah sebagai berikut.



Gambar 2. 15 Diagram pada Star UML

2. Medukung beberapa Bahasa pemrograman, yaitu sebagai berikut.
  - a. *C++ Profile*
  - b. *C# Profile*
  - c. *Java Profile*
3. *Tool editing* yang mendukung



Gambar 2. 16 Tool Editing

## 2. 6 *Enterprise Architect*

### 2.6.1 Pengertian *Enterprise Architect*

*Enterprise Architect* adalah *platform* visual untuk merancang dan membangun sistem perangkat lunak, untuk pemodelan proses bisnis, dan untuk tujuan pemodelan yang lebih umum. *Enterprise Architect* didasarkan pada spesifikasi *UML® 2.5* terbaru. *UML* mendefinisikan bahasa visual digunakan untuk memodelkan domain atau sistem tertentu (baik yang diusulkan atau yang sudah ada)[11].



*Gambar 2. 17 Enterprise Architect*

*Enterprise Architect* adalah alat progresif yang mencakup semua aspek dari siklus pengembangan, menyediakan penuh ketertelusuran dari tahap desain awal hingga penerapan, pemeliharaan, pengujian, dan kontrol perubahan[11].

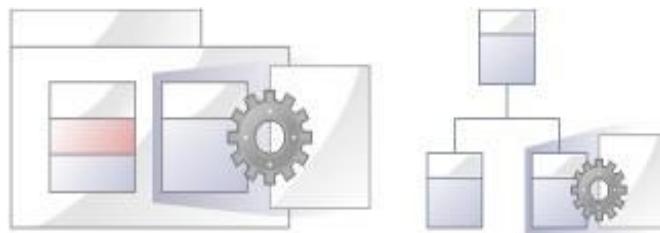
*Software* ini kami gunakan untuk membuat diagram-diagram yang diperlukan dalam perancangan aplikasi.

### **2.6.2 Keunggulan Menggunakan *Enterprise Architect***

Ada beberapa keunggulan dalam menggunakan *software* ini, diantaranya sebagai berikut.

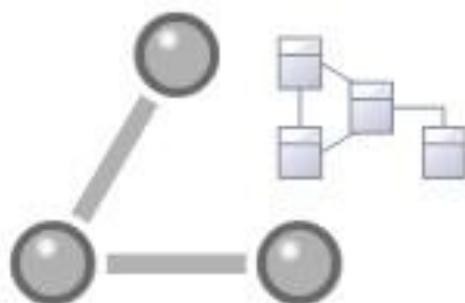
1. Model dan Mengelola Informasi Lebih Kompleks

*Software* ini membantu individu, kelompok dan organisasi besar dan mengelola kompleks informasi. Sering sekali berkaitan dengan pengembangan perangkat lunak dan desain dan penyebaran sistem TI, tetapi juga bisa berhubungan dengan analisis bisnis dan pemodelan proses bisnis. *Enterprise Architect* mengintegrasikan dan menghubungkan secara luas berbagai informasi struktural dan perilaku, membantu membangun model arsitektur dapat diverifikasi[11].



Gambar 2. 18 Model dan Mengelola Informasi Lebih Kompleks

## 2. Model, Manage dan Trace Requirements



Gambar 2. 19 Model, Manage, dan Trace Requirements

Ambil persyaratan dan gunakan *trace* penuh dari persyaratan dasar hingga desain, bangun, penempatan, dan seterusnya. Gunakan analisis dampak untuk melacak dari perubahan yang diajukan ke persyaratan asli. Serta bangun sistem yang benar[11].

### 3. Integrasikan Tim dan Bagikan Visi



Gambar 2. 20 Integrasikan Tim dan Bagikan Visi

*Enterprise Architect* mengintegrasikan anggota tim dari semua bagian dan semua fase siklus pengembangan dan pemeliharaan produk (atau sistem), memberikan manfaat signifikan dari kolaborasi internal dan berbagi informasi yang melekat. Tempat penyimpanan tunggal untuk analis bisnis, arsitek perangkat lunak, pengembang, manajer proyek, penguji, peluncuran dan staf pendukung. Model bersama dapat diakses dengan mudah dan aman oleh anggota tim jarak jauh dengan *Server Cloud Enterprise Architect*[11].

### 4. Desain dan Bangun Sistem Beragam menggunakan *UML*

*UML* 2.5, standar terbuka, menyediakan bahasa yang bertujuan untuk menggambarkan, mendokumentasikan dan merancang perangkat lunak, bisnis, dan sistem TI secara umum.



Gambar 2. 21 Desain dan Bangun Sistem Beragam menggunakan UML

*Enterprise Architect* memungkinkan Anda memanfaatkan kekuatan ekspresif penuh *UML* 2.5 untuk memodelkan, merancang, dan membangun beragam sistem secara terbuka dan dipahami dengan baik[11].

#### 5. Visualisasikan, Periksa dan Pahami Perangkat Lunak Kompleks

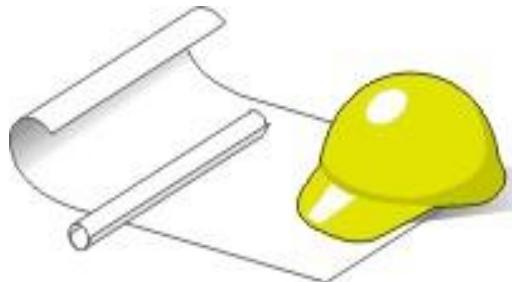
Perangkat lunak atau *software* itu kompleks dan seringkali sulit dipahami. Gunakanlah *Enterprise Architect* untuk merekayasa balik beragam kode sumber untuk memahami struktur statis[11].



Gambar 2. 22 Visualisasikan, Periksa dan Pahami Perangkat Lunak Kompleks

Untuk melengkapi gambar, gunakan alat bantu pembuatan profil dan *debugging* bawaan yang unik untuk menangkap dan menggambarkan perangkat lunak pelaksana. Membuat *instance run-time* elemen model dan memanggil metode menggunakan *Object Workbench* bawaan. Mengintegrasikan model data yang ada dengan skema database rekayasa terbalik untuk berbagai sistem[11].

#### 6. Menggunakan *Life-Cycle Modeling* dan *Project Management*



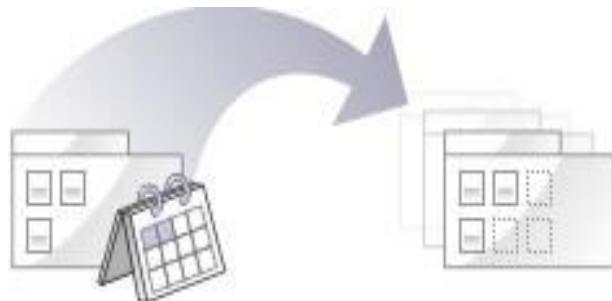
Gambar 2. 23 Menggunakan *Life-Cycle Modeling* dan *Project Management*

Ambil dan lacak informasi tentang elemen-elemen model yang penting bagi kesuksesan. Misalnya pengujian, manajemen proyek dan rincian pemeliharaan. Gunakan informasi ini untuk mendorong dan melacak pengembangan dan pengiriman produk[11].

#### 7. Berbagi dan Memakai Ulang *Information Across Tools*

*Enterprise Architect* mendukung sejumlah mekanisme untuk mengeksport dan mengimport model menggunakan standar industri *XMI*. Hal ini memungkinkan pemodelan untuk menggunakan informasi yang dibuat dalam alat lain, untuk menyalin informasi antara model *Enterprise Architect*

dan bahkan untuk menulis dan menggunakan alat kustom yang mengambil XMI secara langsung sebagai input[11].

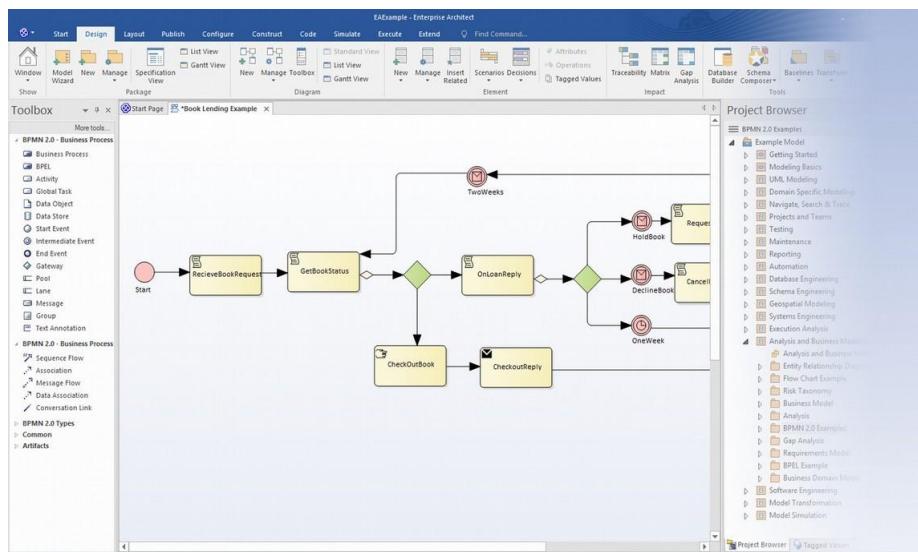


*Gambar 2. 24 Berbagi dan Memakai Ulang Information Across Tools*

*Enterprise Architect* juga dapat digunakan sebagai Open Services for Lifecycle Collaboration (*OSLC*), yang memungkinkan alat lain untuk mengidentifikasi dan mengakses persyaratan yang dimodelkan melalui URL unik[11].

### **2.6.3 User Interface *Enterprise Architect***

Antarmuka pengguna *Enterprise Architect* adalah sebagai berikut.



Gambar 2. 25 User Interface Enterprise Architect

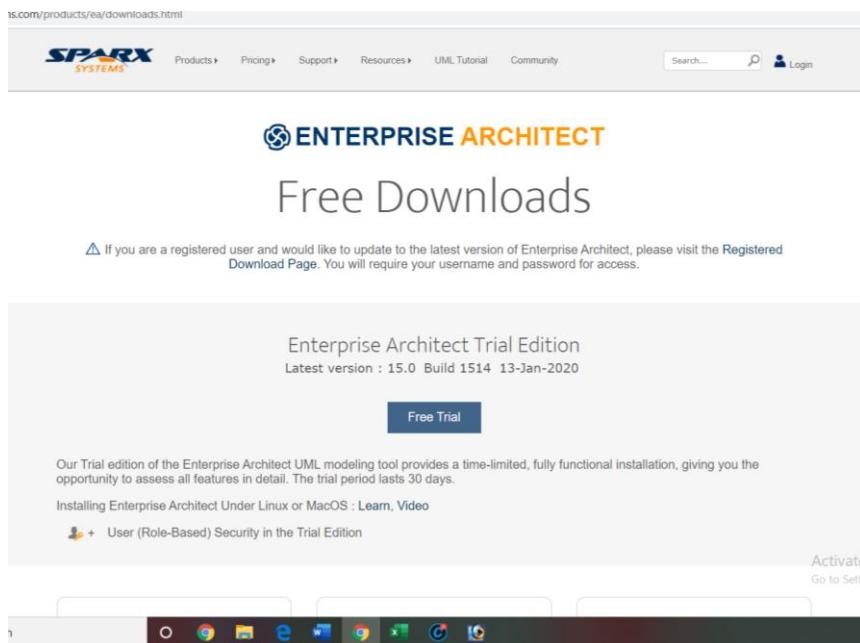
1. *The Project Browser* yang menampilkan model proyek anda dan memfasilitasi penambahan, pemilihan, penataan ulang, atau penghapusan paket, diagram, dan elemen[11].
2. *The Diagram Toolbox* merupakan konteks terhadap diagram yang sedang diedit, dan menyediakan cara yang efisien untuk membuat elemen atau konektor model[11].
3. *The Diagram View* merupakan konektor dan elemen warna untuk presentasi dan identifikasi yang ditingkatkan[11].
4. *Edit multiple views concurrently* dengan mudah mentransfer elemen antara beberapa diagram terbuka[11].

#### 2.6.4 Cara Download Enterprise Architect

Berikut ini adalah langkah-langkah men-download *Enterprise Architect*.

1. Pergi ke *website*

<https://sparxsystems.com/products/ea/downloads.html>



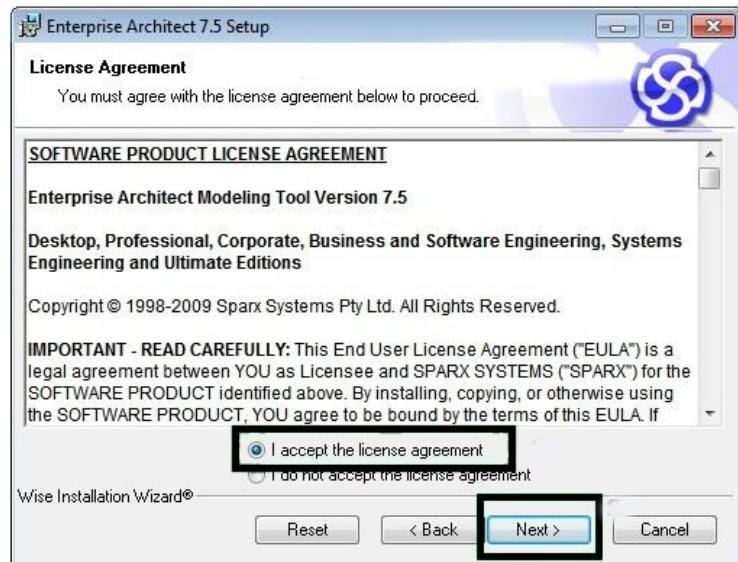
Gambar 2. 26 Website Enterprise Architect

2. Setelah *software* diunduh, lakukan instalasi dengan mengklik *double* pada file tersebut.



Gambar 2. 27 Install Enterprise Architect (1)

3. Pilih terima atau *accept* dan klik *next*



Gambar 2. 28 Install Enterprise Architect (2)

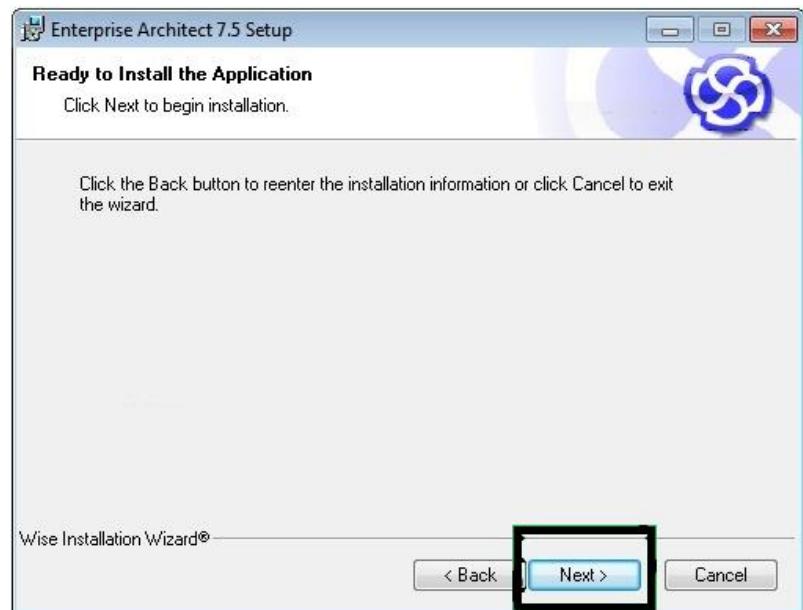
4. Selanjutnya klik *next* lagi seperti gambar berikut ini



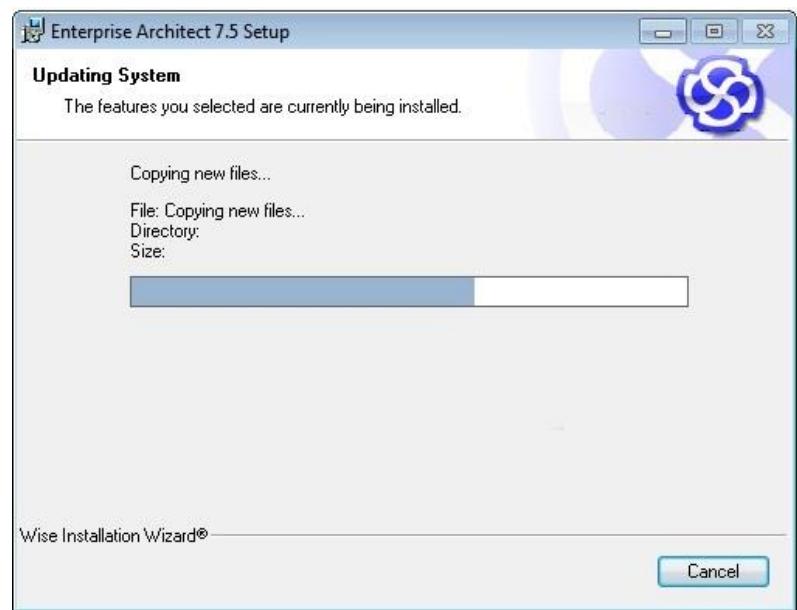
Gambar 2. 29 Install Enterprise Architect (3)



Gambar 2. 30 Install Enterprise Architect (4)



Gambar 2. 31 Install Enterprise Architect (5)



Gambar 2. 32 Install Enterprise Architect (6)

5. Software telah sudah terinstall



Gambar 2. 33 Install Ensterprise Architect (7)

## BAB III

# BAHASA PEMROGRAMAN

---

### 3.1 Pengertian Bahasa Pemrograman

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang *programmer* dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi[12].

Bahasa Pemrograman (*programming language*) adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang *programmer* dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi[12].



Gambar 3. 1 Bahasa Pemrograman

### 3.2 Fungsi Pemrograman

Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. *Output* atau keluaran dari bahasa pemrograman tersebut berupa program/aplikasi. Contohnya adalah program yang digunakan oleh kasir di mal atau swalayan, penggunaan lampu lalu lintas di jalan raya, dll. Bahasa Pemrograman yang kita kenal ada banyak sekali di belahan dunia, tentang ilmu komputer dan teknologi dewasa ini[12].

Perkembangannya mengikuti tingginya inovasi yang dilakukan dalam dunia teknologi. Contoh bahasa pemrograman yang kita kenal antara lain adalah untuk membuat aplikasi *game*, antivirus, web, dan teknologi lainnya. Bahasa pemrograman komputer yang kita kenal antara lain adalah *Java*, *Visual Basic*, *C++*, *C*, *Cobol*, *PHP*, *.Net*, dan ratusan bahasa lainnya[12].

Namun tentu saja kebutuhan bahasa ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya. Secara umum bahasa pemrograman terbagi menjadi 4 kelompok, yaitu :

1. *Object Oriented Language* (contohnya *Visual dBase*, *Visual FoxPro*, *Delphi*, *Visual C*)
2. *High Level Language* (contohnya *Pascal* dan *Basic*)
3. *Middle Level Language* (contohnya bahasa *C*)
4. *Low Level Language* (seperti bahasa *Assembly*)[12]

### **3.3 Tingkatan Bahasa Pemrograman**

Bahasa pemrograman memiliki tingkatan bahasa yang dibedakan menjadi beberapa jenis yaitu:

1. Bahasa mesin, yaitu memberikan perintah kepada *computer* dengan memakai kode bahasa biner, contohnya 01100101100110[12].
2. Bahasa tingkat rendah, atau dikenal dengan istilah bahasa rakitan (bah. Inggris *Assembly*), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode mnemonic), contohnya kode\_mesin *MOV*, *SUB*, *CMP*, *JMP*, *JGE*, *JL*, *LOOP*, dsb[12].
3. Bahasa tingkat menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <>, &&, //, dsb[12].
4. Bahasa tingkat tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia,

contohnya *begin*, *end*, *if*, *for*, *while*, *and*, *or*, dsb. Komputer dapat mengerti bahasa manusia itu diperlukan program *compiler* atau *interpreter*. Sebagian besar bahasa pemrograman digolongkan sebagai bahasa tingkat tinggi, hanya bahasa *C* yang digolongkan sebagai bahasa tingkat menengah dan *Assembly* yang merupakan bahasa tingkat rendah[12].

### **3.4 Struktur Dasar Bahasa Pemrograman**

Setiap bahasa pemrograman memiliki struktur dasar bahasa pemrograman yang berbeda-beda, tetapi struktur Bahasa pemrograman dapat masuk dalam beberapa kategori. Beberapa kategori struktur dasar bahasa pemrograman yang dapat dipelajari oleh pemula adalah sebagai berikut.

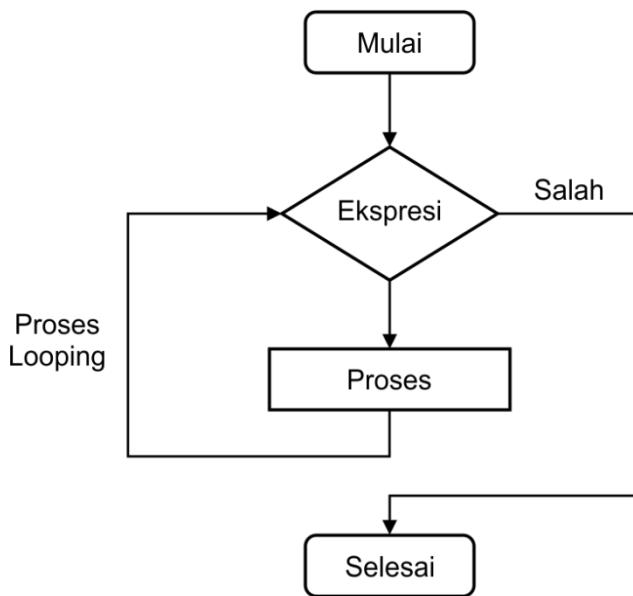
#### **3.4.1 Struktur Runtutan**

Struktur Runtutan merupakan struktur algoritma paling dasar yang berisi rangkaian intruksi yang diproses secara sekuensial, yang artinya dikerjakan satu per satu , mulai dari intruksi pertama sampai intruksi terakhir artinya dimulai dari langkah pertama, kemudian kedua, ketiga, dan seterusnya hingga akhir. Pada dasarnya suatu program memang menjalankan suatu proses dari yang dasar seperti struktur seperti ini[12].

#### **3.4.2 Struktur Perulangan**

Dengan memakai struktur perulangan saat menjalankan sebuah program, kita harus menjalankan sebuah perintah secara berulang-ulang, agar program dapat berjalan secara berkelanjutan. Struktur ini dapat

menjadi pilihan yang tepat untuk membuat sebuah program menjalankan baris-baris perintah secara berulang-ulang[12].

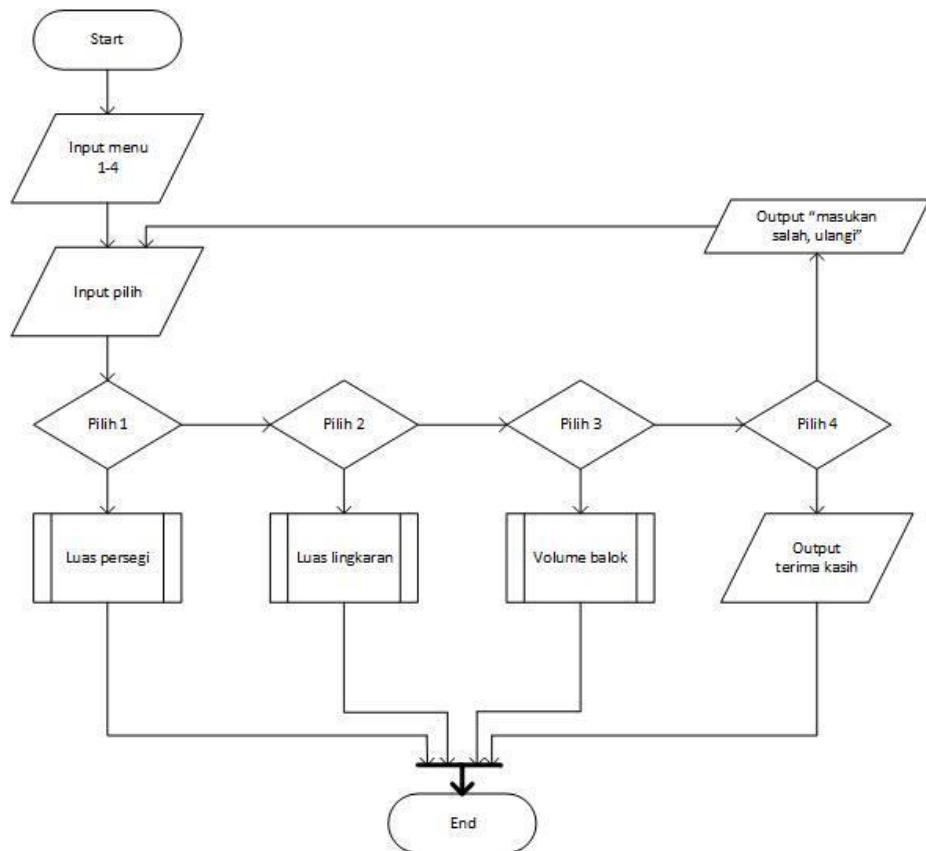


Gambar 3. 2 Contoh Struktur Perulangan

Dengan menggunakan struktur dasar bahasa pemrograman ini, seorang pembuat program dapat mengatur perintah-perintah dalam programnya agar berjalan secara berulang-ulang hingga pada suatu kondisi dimana syarat untuk perulangannya sudah tidak terpenuhi kembali. Terdapat beberapa fungsi perulangan yang dapat digunakan untuk membuat struktur program ini, seperti *For*, *While*, dan *While-do* pada bahasa *C++*. Ketiga fungsi ini dapat digunakan untuk membuat sebuah program untuk menjalankan baris-baris perintah dalam syarat tertentu[12].

### 3.4.3 Struktur Percabangan

Membuat program juga tidak akan terlepas dari membuat sebuah pilihan dalam menjalankan perintah sesuai dengan syarat tertentu. Pembuat program dapat menggunakan struktur ini untuk membuat sebuah program dengan beberapa pilihan keputusan yang disesuaikan dengan syarat atau kondisi yang dikehendaki. Sebagai contoh, saat kita ingin melihat apakah lulus atau tidak dalam sebuah mata pelajaran, logikanya adalah jika nilai kita lebih dari 70 maka kita lulus, dan jika kurang dari 70 maka kita tidak lulus. Dalam hal ini, maka kita dapat menggunakan struktur percabangan ini[12].



Gambar 3. 3 Contoh Struktur Percabangan

Dalam kondisi di atas, kita dapat menggunakan fungsi *If-Else* atau *Switch Case* dalam bahasa pemrograman *C* untuk *board Arduino*. Sebagai contoh berikut ini contoh struktur percabangan dengan menggunakan fungsi *If-Else* untuk kasus yang telah disebutkan. Membuat sebuah program merupakan aktivitas memilih struktur mana yang tepat untuk program tersebut. Sehingga seorang pembuat program harus mengetahui berbagai macam jenis struktur dasar bahasa pemrograman, sebagaimana yang telah dijelaskan di atas[12].

Setelah mengetahui berbagai macam struktur dan fungsi yang dapat digunakan, pembuat program dapat memilih dan menentukan struktur mana yang tepat untuk program yang akan dibuat. Dengan struktur yang tepat maka program yang dibuat akan sesuai dengan yang diinginkan[12].

### **3.5 Jenis Bahasa Pemrograman yang dipakai**

Pada aplikasi Kehadiran Dosen menggunakan bahasa pemrograman *PHP: Hypertext Preprocessor*. Berikut ini adalah penjelasan dari bahasa pemrograman *PHP* yang digunakan pada aplikasi Kehadiran Dosen.

#### **3.5.1 Pengertian *PHP: Hypertext Preprocessor***

*PHP* adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. *PHP* banyak dipakai untuk memprogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. *PHP* juga dapat dilihat sebagai pilihan lain dari *ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems*, dan *CGI/Perl*. Contoh aplikasi lain yang lebih

kompleks berupa *CMS* yang dibangun menggunakan *PHP* adalah *Mambo*, *Joomla!*, *Postnuke*, *Xaraya*, dan lain-lain[13].



Gambar 3. 4 PHP: Hypertext Preprocessor

*PHP* merupakan salah satu bahasa pemrograman berbasis web. *PHP* adalah bahasa scripting yang menyatu dengan *HTML* dan dijalankan pada *server side*. Artinya semua sintaks sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke browser hanya hasilnya saja[14].

### 3.5.2 Sejarah *PHP*

*PHP* (*PHP: Hypertext Preprocessor*) adalah sebuah bahasa pemrograman yang perintahnya dilaksanakan server dan kemudian hasilnya ditampilkan pada komputer *client*. *PHP* juga merupakan *HTML embedded*, yaitu sintaks *PHP* yang dituliskan bersamaan dengan sintaks *HTML*. Jadi *PHP* dan *HTML* adalah sinergi dua bahasa pemrograman

yang saling menguatkan. Walaupun sebagian orang berpendapat *HTML* bukan sebuah bahasa pemrograman[14].

PHP diperkenalkan oleh Rasmus Lerdorf pada tahun 1995. Pada awalnya PHP memiliki kepanjangan Personal Homepage. Hal ini karena PHP merupakan sebuah aplikasi kecil yang digunakan untuk melengkapi situs personal Rasmus di Internet (<http://www.php.net>).

PHP juga merupakan bahasa pemrograman bersifat Open Source yang bisa didownload gratis. Hingga saat ini PHP versi terakhir adalah versi 7.0.8, dimana bisa didownload pada situs resmi PHP <http://php.net/downloads.php>[14].

### **3.5.3 Keunggulan Bahasa Pemrograman *PHP***

Berikut ini adalah beberapa keunggulan jika menggunakan Bahasa pemrograman *PHP*.

#### **1. Keamanan**

Keamanan sebuah program selain sistem operasi menjadi sangat penting. *PHP* menyediakan 3 jenis authentikasi *user*, yaitu *http* authentikasi, penggunaan *cookies* dan penggunaan *session* [14].

#### **2. Integritas dengan *Database***

*PHP* mendukung integritas, kecepatan dan efisiensi akses ke *database* yang kebanyakan menggunakan *database* berjenis relational seperti *MySQL*, *PostgreSQL*, *Oracle*, *SQLite* dan lain-lain[14].

#### **3. *Cross-platform***

*PHP* mendukung berbagai jenis sistem operasi seperti semua varian *Linux*, *Microsoft Windows*, *Mac OS* dan lain-lain[14].

#### 4. Reliabilitas

*PHP* merupakan salah satu bahasa pemrograman yang bebasis web. Alasan utama adalah dukungan dokumentasi yang lengkap, aman dan banyak komunitas helpdesk untuk membantu para pengembang web sistem yang menggunakan *PHP*[14].

#### 5. Harga

*PHP* berada dalam lisensi *GPL (GNU Public Lisence)*. Hal ini berarti bahwa *PHP* bebas digunakan dan didistribusikan serta gratis. Saat ini juga banyak *hosting* gratis dan *unlimited men-support PHP*[14].

#### 6. Kemudahan Bermigrasi

Tujuannya adalah memperbaiki kinerja dan menambah fitur-fitur baru. Kelebihan ini karena banyaknya dukungan terhadap *PHP* sehingga berdampak *PHP* terus menerus dikembangkan[14].

### **3.5.4 Dasar Pemrograman *PHP***

Untuk pengguna yang baru pertama kali mencoba *PHP*, tuliskan sintaks bahasa pemrograman *PHP* dan berikan nama *testing.php*, kemudian file ini disimpan ke dalam folder *C:\xampp\htdocs*. Untuk memudahkan manajemen file sebaiknya dibuatkan folder tersendiri. Dalam buku ini folder latihan *PHP* diberi nama *test* menjadi *C:\xampp\htdocs\test*.

Untuk melihat hasil program *PHP*, cobalah buka browser *Google Chrome* dan ketik <http://localhost/test/testing.php>. Berikut sintaks dasar bahasa pemrograman *PHP* dengan nama *testing.php*

File *testing.php*:

```
<html>

<head>

<title> PHP Sederhana </title>

</head>

<body>

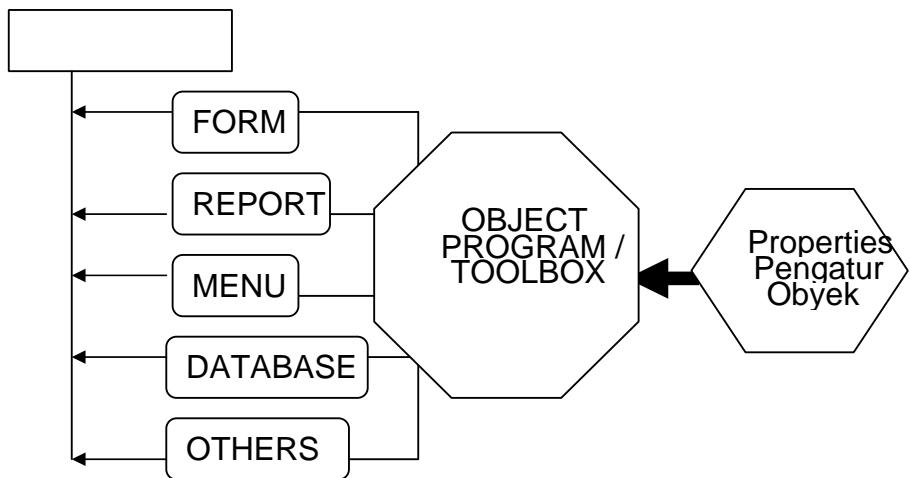
<?php echo("Hallo apa kabar? Nama saya PHP
script"); ?>

</body>

</html>
```

### 3.6 *OOP (Object Oriented Programming)*

*OOP (Object Oriented Programming)* adalah sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan teknik berorientasi atau berbasis pada sebuah obyek dalam pembangunan program aplikasi, maksudnya bahwa orientasi pembuatan program tidak lagi menggunakan orientasi linear melainkan berorientasi pada objek-objek yang terpisah-pisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah Obyek yang didalamnya berisi beberapa perintah-perintah standar sederhana. Obyek ini dikumpulkan dalam Modul *form* atau *Report* atau modul lain dan disusun didalam sebuah *project*[15]. Gambaran tentang pemrograman ini seperti dibawah ini :



*Gambar 3. 5 Object Oriented Programming*

*Object Oriented Programming* (OOP) adalah paradigma pemrograman yang memandang perangkat lunak sebagai kumpulan objek yang saling berinteraksi didalam suatu sistem. (Azis, 2005, p2). Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Masing-masing objek harus berisikan informasi mengenai dirinya sendiri (*encapsulation*) dan objek yang dapat dikaitkan (*inheritance*). (Febrian, 2004)[16].

# OOP

*Gambar 3. 6 OOP*

Pada OOP, terdapat juga yang disebut dengan *inheritance* (pewarisan), yaitu kepemilikan yang bersifat implicit dari *fitur subclass* yang didefinisikan dalam *superclass*. Fitur tersebut mencakup *variables* dan *method*[16].

## BAB IV

# PERANCANGAN DIAGRAM UML

---

### 4.1 Pengertian *UML*

*Unified Modelling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem[17].

Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena *UML* juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan dalam Bahasa-bahasa berorientasi objek seperti *C++*, *Java*, *C#* atau *VB.NET*. Walaupun demikian, *UML* tetap dapat digunakan untuk modeling aplikasi prosedural dalam *VB* atau *C*[17].



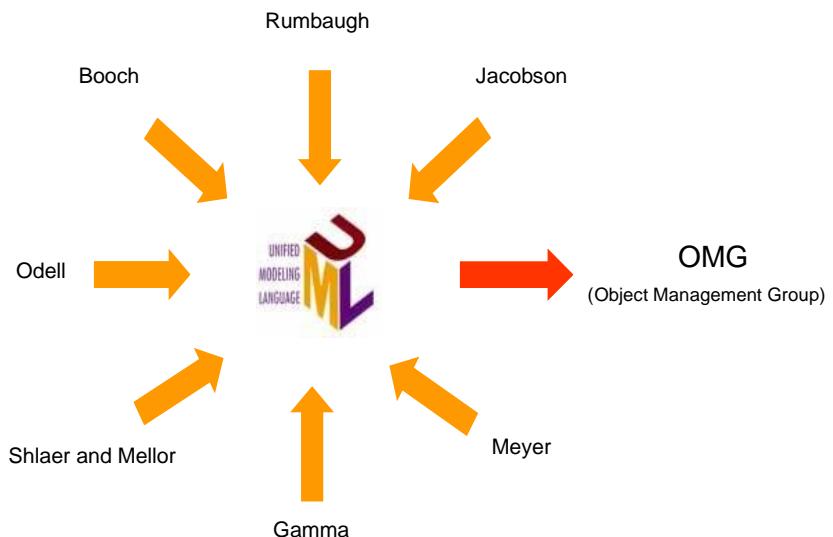
#### 4. 1 Unified Modelling Language (UML)

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[17].

#### 4.2 Sejarah *UML*

Sejarah *UML* sendiri cukup panjang. Sampai era tahun 1990 seperti yang kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi *booch* [1], metodologi *coad* [2], metodologi *OOSE* [3], metodologi *OMT* [4], metodologi *shlaer-mellor* [5], metodologi *wirfs-brock* [6], dsb. Masa itu terkenal dengan masa perang metodologi (*method war*)

dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.



4. 2 *Unified Modelling Language (UML) (2)*

Dimulai pada bulan Oktober 1994 *Booch*, *Rumbaugh* dan *Jacobson*, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 di-release draft pertama dari *UML* (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group (OMG* – <http://www.omg.org>). Tahun 1997 *UML* versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. *Booch*, *Rumbaugh* dan *Jacobson* menyusun tiga buku serial tentang *UML* pada

tahun 1999 [7] [8] [9]. Sejak saat itu UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek[17].

### 4.3 Konsepsi Dasar *UML*

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku *UML*. Sebenarnya konsepsi dasar *UML* bisa kita rangkumkan dalam gambar dibawah[17].

<i>Major Area</i>	<i>View</i>	<i>Diagrams</i>	<i>Main Concepts</i>
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
	deployment view	deployment diagram	node, component, dependency, location
dynamic	state machine view	statechart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
		collaboration diagram	collaboration, interaction, collaboration role, message
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

#### 4. 3 Konsepsi Dasar *UML*

Abstraksi konsep dasar *UML* yang terdiri dari structural classification, dynamic behavior, dan model management, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams*. *Main concepts* bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram. Dan view adalah kategori dari diagaram tersebut[17].

#### **4.4 Diagram UML**

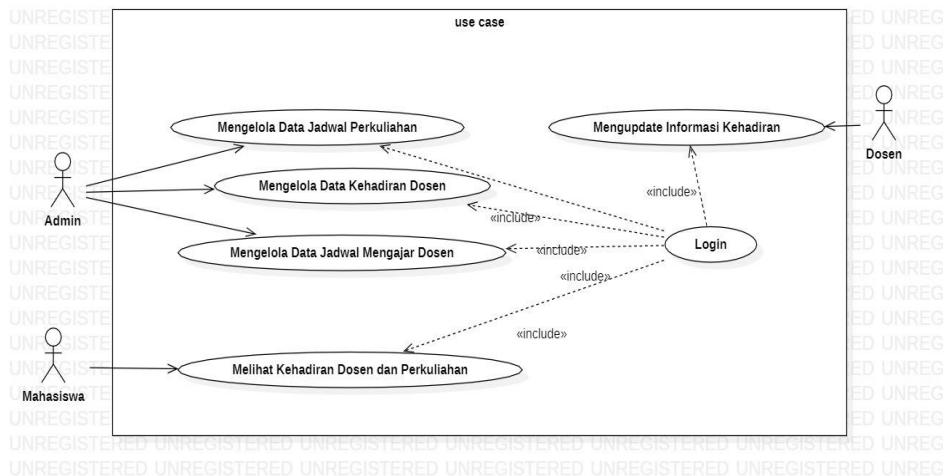
Berikut ini adalah diagram-diagram yang bisa digambar dengan *UML*

##### **4.4.1 Use Case Diagram**

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

*Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Berikut ini adalah *use case diagram* yang dibuat untuk merancang aplikasi Kehadiran Dosen.



4. 4 Use Case Diagram

#### 4.4.1.1 Definisi Use Case

##### 1. Definisi Aktor

Tabel 4: 1 Definisi Aktor

No	Aktor	Keterangan
1	Admin	Aktor yang dapat mengelola data jadwal perkuliahan, jadwal

		mengajar, dan data informasi kehadiran dosen
2	Dosen	Aktor dapat melihat jadwal mengajar dan dapat meng- <i>update</i> kehadiran dosen
3	Mahasiswa	Aktor yang mendapat notifikasi perkuliahan dan dapat melihat info kehadiran dosen mengajar

## 2. Definisi Use Case

Tabel 4: 2 Definisi Use Case

No	Use Case	Keterangan
1	Login	Merupakan proses untuk melakukan <i>login</i> admin, dosen, dan mahasiswa
2	Mengelola Data Jadwal Perkuliahan	Mengelola jadwal perkuliahan merupakan proses pengelolahan data jadwal kuliah
3	Mengelola Data Kehadiran Dosen	Mengelola data kehadiran dosen merupakan proses pengelolahan data informasi kehadiran
4	Melihat Kehadiran Dosen	Mahasiswa dapat melihat info kehadiran dosen yang mengajar serta melihat jadwal kuliah.

5	Meng-update Informasi Kehadiran	Dosen dapat meng-update informasi kehadiran untuk mengajar
---	---------------------------------------	--

#### 4.4.1.2 Skenario Use Case

Skenario *use case* menjelaskan langkah-langkah dalam sistem yang berjalan pada aplikasi, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan sistem terhadap aktor. Berikut ini adalah penjelasan dari masing-masing skenario tersebut.

##### 1. Skenario *Use Case Login*

Tabel 4: 3 Skenario *Use Case Login*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memasukkan <i>username</i> dan <i>password</i>	2. Memeriksa validasi data dengan memeriksa ke tabel aktor
	3. Masuk ke Aplikasi Kehadiran Dosen
Skenario Alternatif	
1. Memasukkan <i>username</i> dan <i>Password</i>	

	2. Memeriksa validasi data dengan memeriksa ke tabel aktor
	3. Menampilkan pesan <i>login</i> tidak <i>valid</i>
4. Memasukkan <i>username</i> dan <i>password</i> yang <i>valid</i>	
	5. Memeriksa validasi data dengan memeriksa ke tabel aktor
	6. Masuk ke Aplikasi Kehadiran Dosen

## 2. Skenario Use Case Mengelola Jadwal

*Tabel 4: 4 Skenario Use Case Mengelola Jadwal*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Login menggunakan <i>username</i> dan <i>password</i>	
	2. Memeriksa validasi data dengan memeriksa ke tabel aktor
	3. Menampilkan Halaman Utama

4. Memilih menu tambah jadwal	
5. Menginputkan jadwal dan menyimpan jadwal	
	6. Jadwal tersimpan
Skenario Alternatif	
1. Login menggunakan <i>username</i> dan <i>password</i>	
	2. Memeriksa validasi data dengan memeriksa ke tabel actor
	3. Menampilkan Halaman Utama
4. Memilih menu tambah jadwal	
5. Menginputkan jadwal dan menyimpan jadwal	
	6. Menampilkan kolom yang belum terisi
7. Menginput ulang jadwal menyimpan jadwal	
	8. Data jadwal tersimpan

### 3. Skenario Use Case Add User

Tabel 4: 5 Skenario Use Case Add User

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Login menggunakan <i>username</i> dan <i>password</i>	
	2. Memeriksa validasi data dengan memeriksa ke tabel aktor
	3. Menampilkan Halaman Utama
4. Memilih menu <i>user</i>	
5. Menginputkan data <i>user</i> dan menyimpan data	
	6. Data tersimpan
Skenario Alternatif	
1. Login menggunakan <i>username</i> dan <i>password</i>	
	2. Memeriksa validasi data dengan memeriksa ke tabel aktor

	3. Menampilkan Halaman Utama
4. Memilih menu <i>user</i>	
5. Menginputkan data <i>user</i> dan menyimpan data	
	6. Menampilkan kolom yang belum terisi
7. Menginput ulang data menyimpan data	
	8. Data <i>user</i> tersimpan

#### 4. Skenario Use Case Meng-Update Kehadiran Dosen

Tabel 4: 6 Skenario Use Case Meng-Update Kehadiran Dosen

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Input kehadiran dosen	
	2. Memberi dua pilihan, yaitu hadir atau tidak
3. Jika tidak hadir, dosen menmberi alasan ketidakhadiran	

4. Melihat informasi kehadiran dosen	
Skenario Alternatif	
1. Input kehadiran dosen	
	2. Memberi dua pilihan, yaitu hadir atau tidak
3. Jika tidak hadir, dosen memberi alasan ketidakhadiran	
	4. Memeriksa valid tidaknya data dimasukkan
	5. Menampilkan pesan bahwa data yang dimasukkan tidak valid
6. Memperbaiki data masukan yang di <i>update</i> dan tidak valid	
7. Melihat informasi kehadiran dosen	

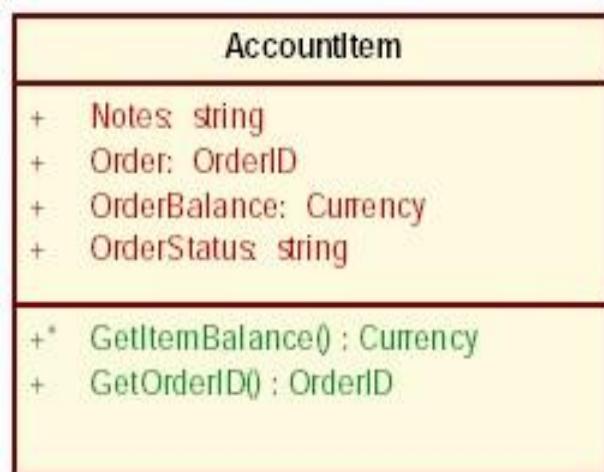
#### 4.4.2 Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi)[17].

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain[17].

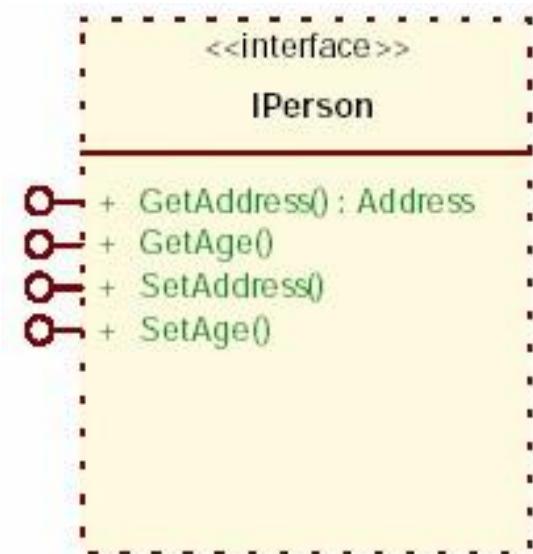
Atribut dan metoda dapat memiliki salah satu sifat berikut :

- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- *Public*, dapat dipanggil oleh siapa saja[17].



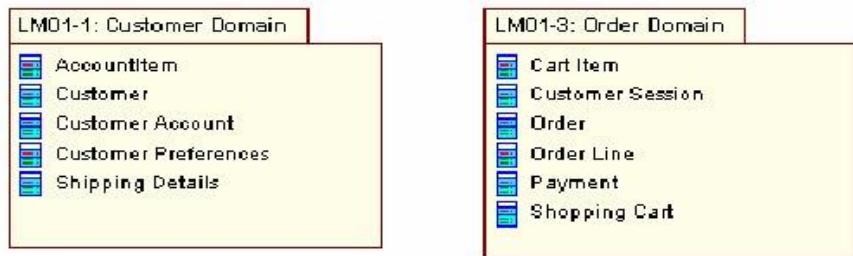
Gambar 4. 1 Class Diagram (1)

*Class* dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.



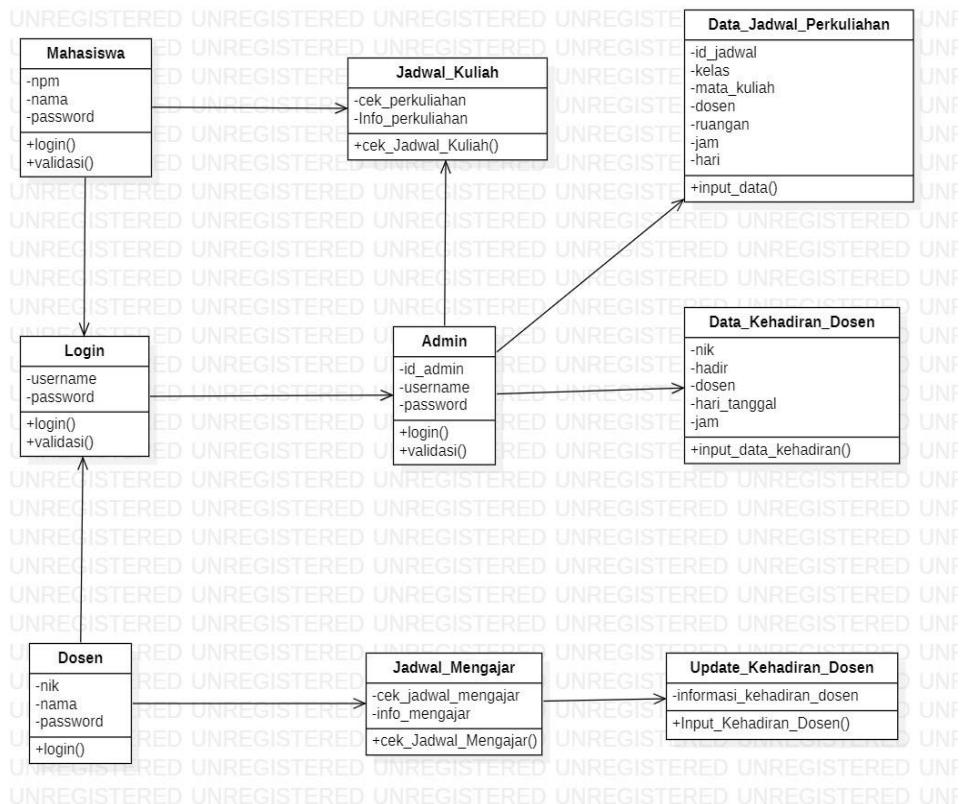
Gambar 4. 2 Class Diagram (2)

Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*[17].



Gambar 4. 3 Class Diagram (3)

Berikut ini adalah *class diagram* yang dibuat pada perancangan aplikasi Kehadiran Dosen.



Gambar 4. 4 Contoh Class Diagram

#### 4.4.3 *Activity Diagram*

*Activity diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi[17].

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum[17].

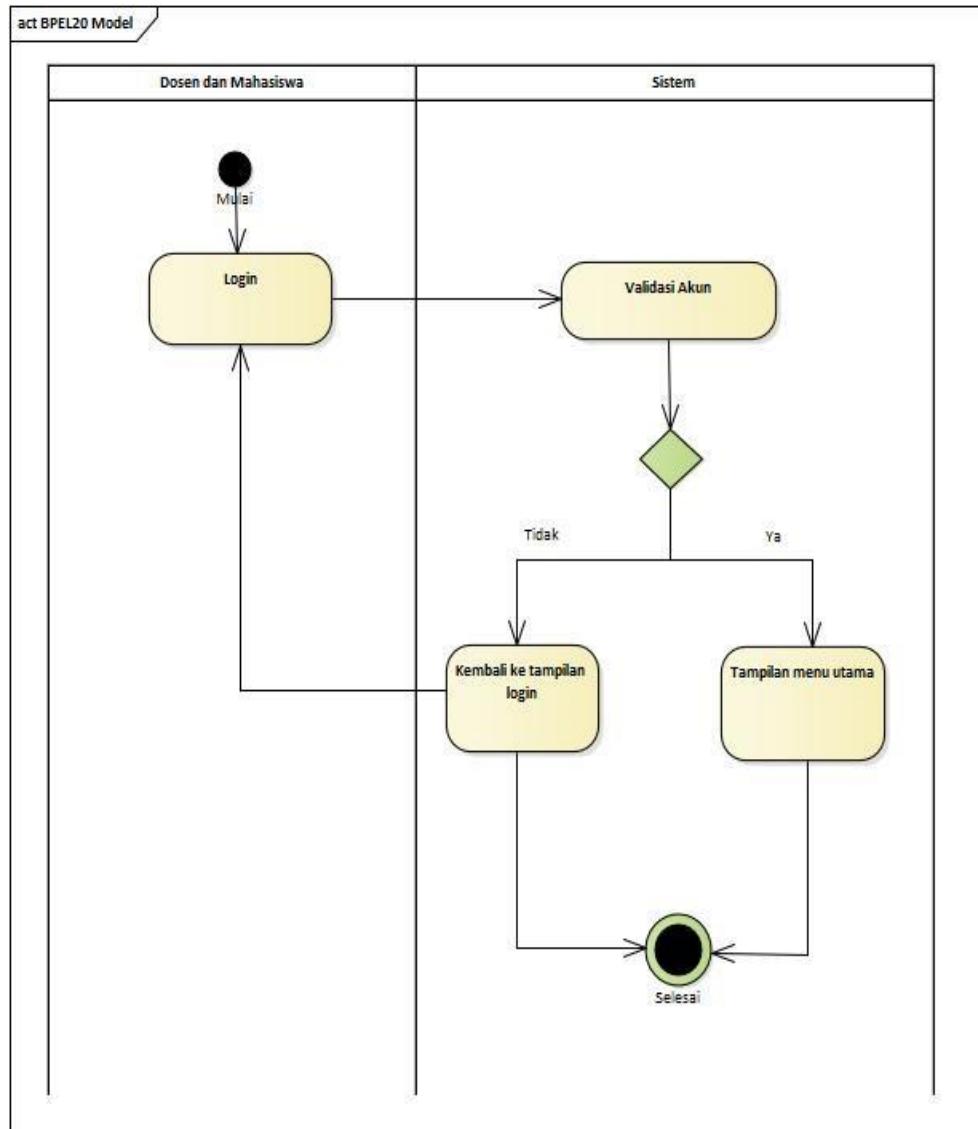
Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas[17].

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertical[17].

*Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu[17].

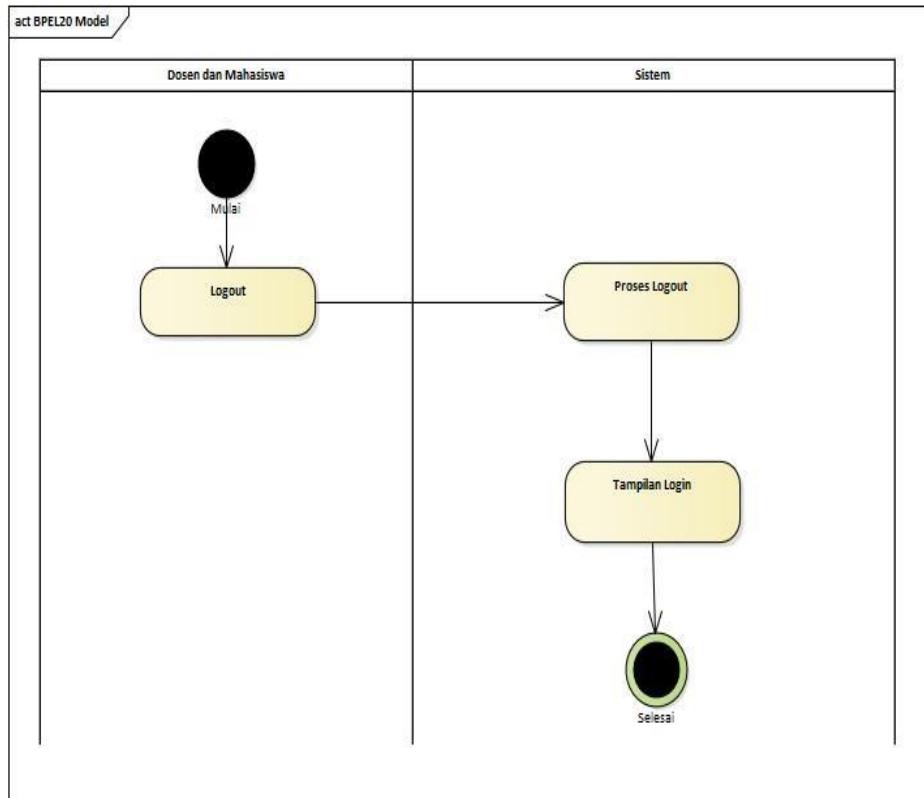
Berikut ini adalah *class diagram* yang dibuat pada perancangan aplikasi Kehadiran Dosen.

### 1. *Activity Diagram Login*



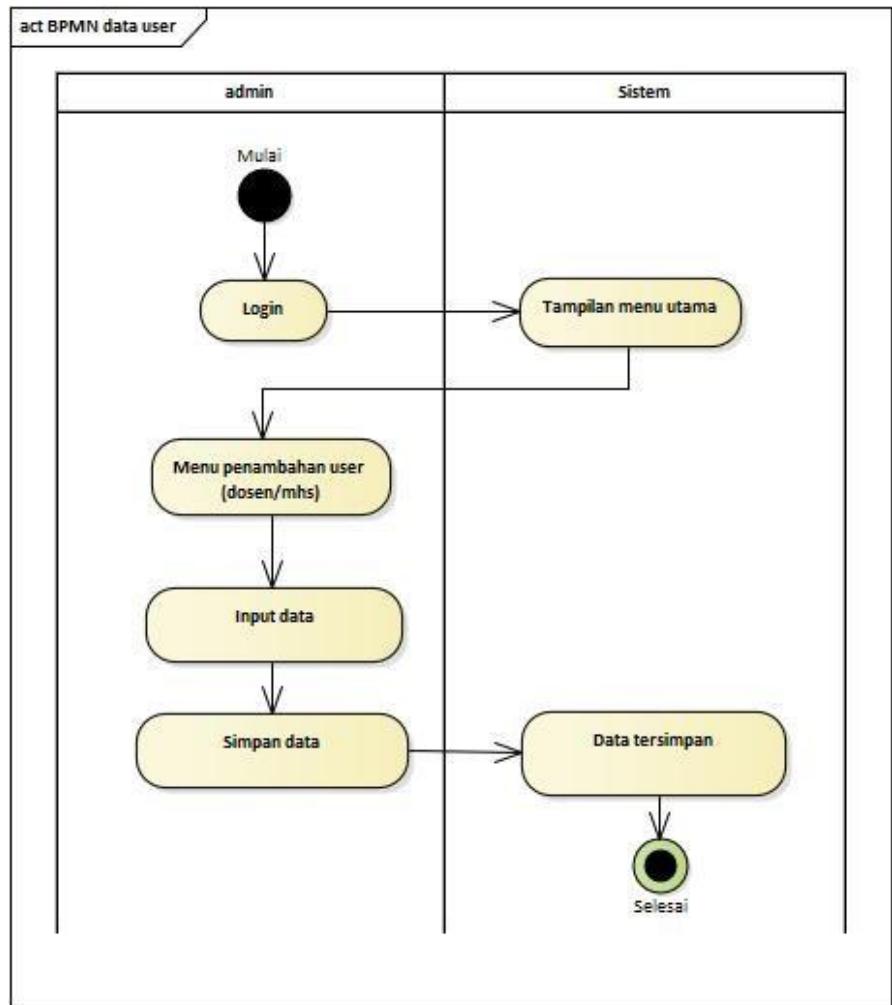
Gambar 4. 5 Activity Diagram Login

## 2. Activity Diagram Logout



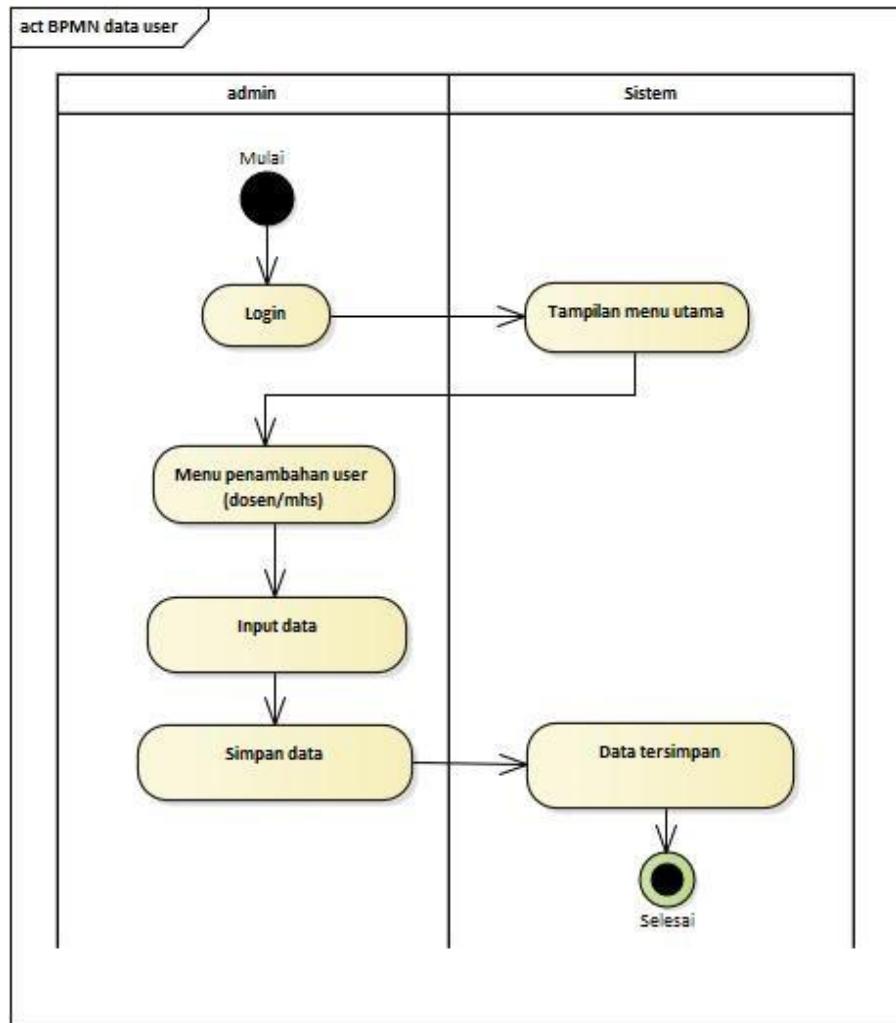
Gambar 4. 6 Activity Diagram Logout

### 3. Activity Diagram Add User



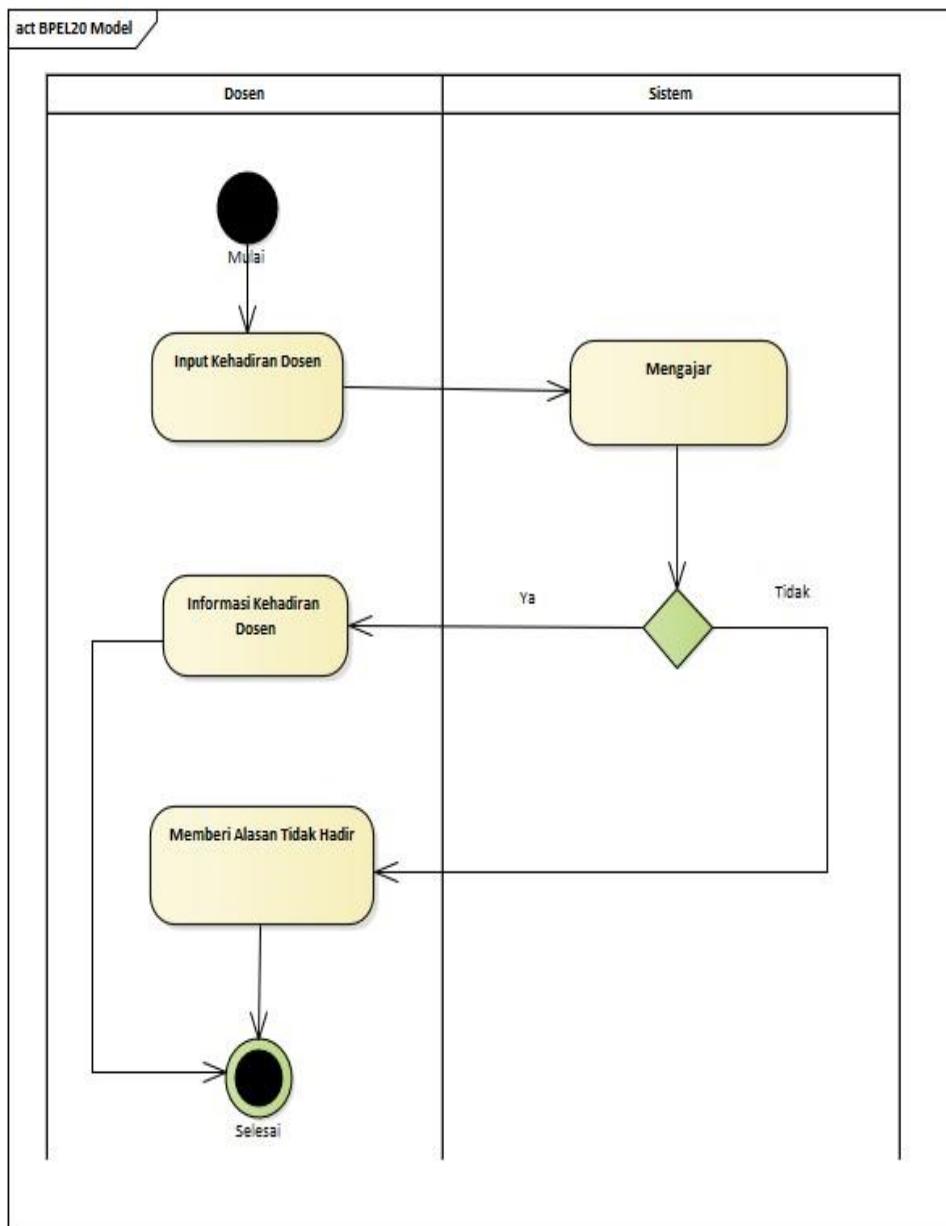
Gambar 4. 7 Activity Diagram Add User

#### 4. Activity Diagram Mengelola Jadwal



Gambar 4. 8 Activity Diagram Mengelola Jadwal

## 5. Activity Diagram Meng-update Informasi Kehadiran



#### 4.4.4 Sequence Diagram

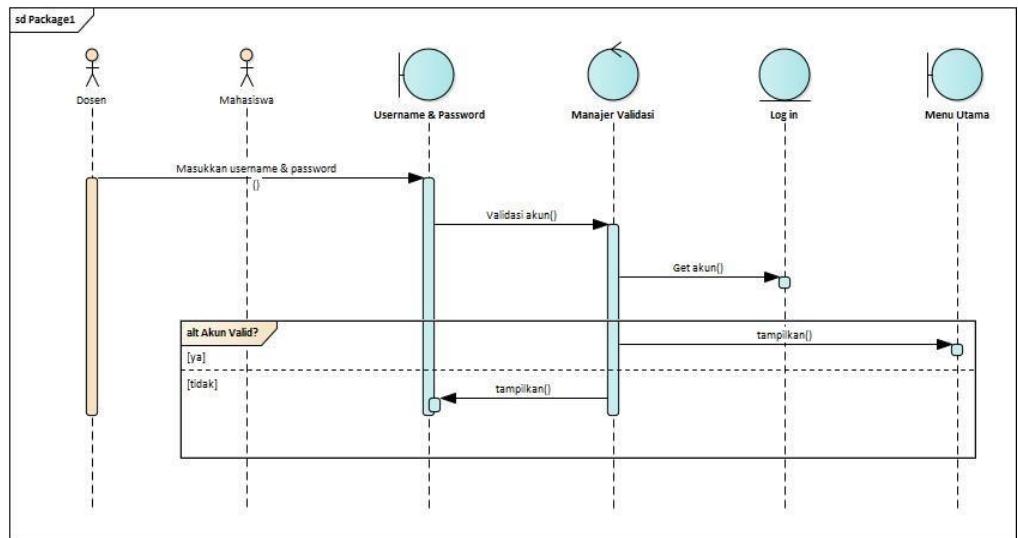
*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait)[17].

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal[17].

*Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar *UML* mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*[17].

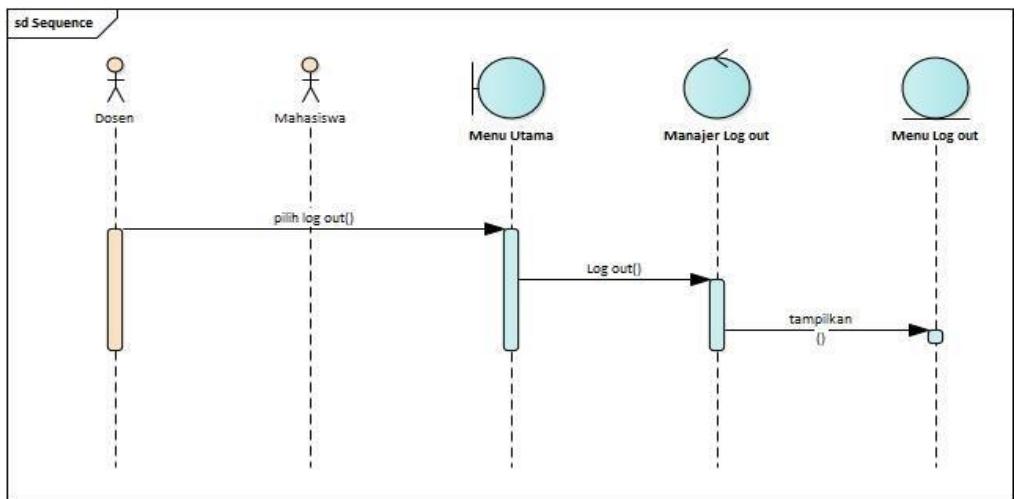
Berikut ini adalah *sequence diagram* yang dibuat pada perancangan aplikasi Kehadiran Dosen.

## 1. Sequence Diagram Login



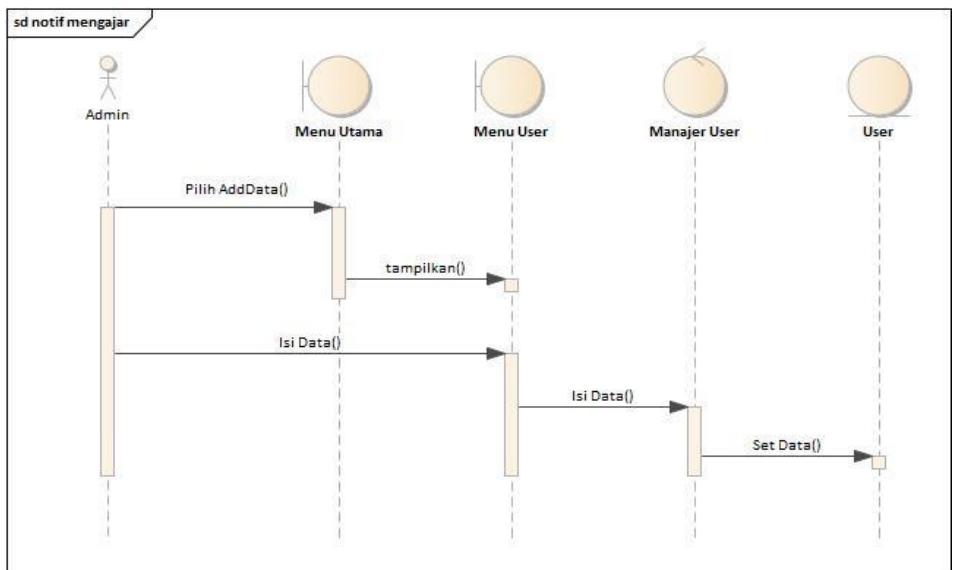
Gambar 4. 9 Sequence Diagram Login

## 2. Sequence Diagram Logout



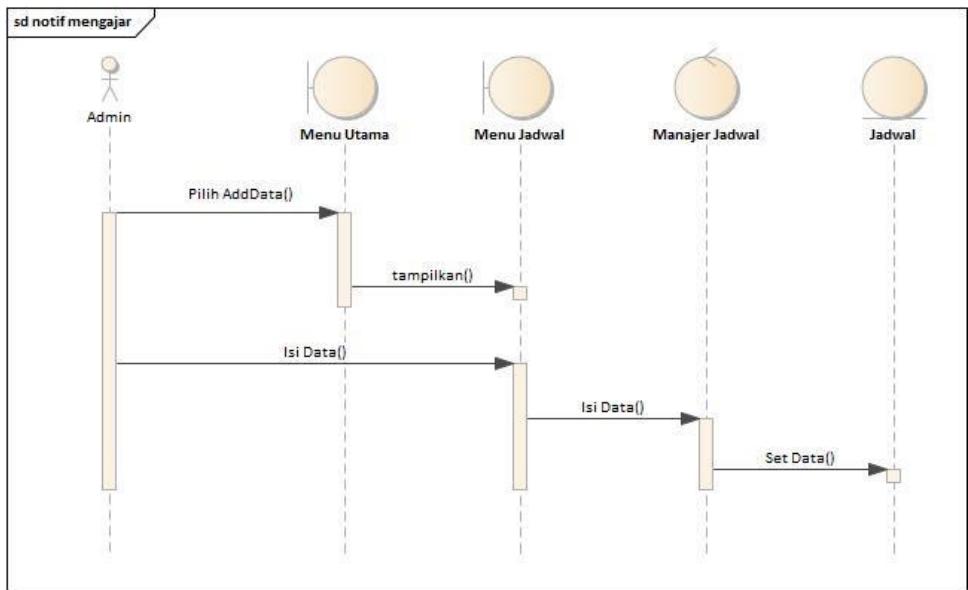
Gambar 4. 10 Sequence Diagram Logout

### 3. Sequence Diagram Add User



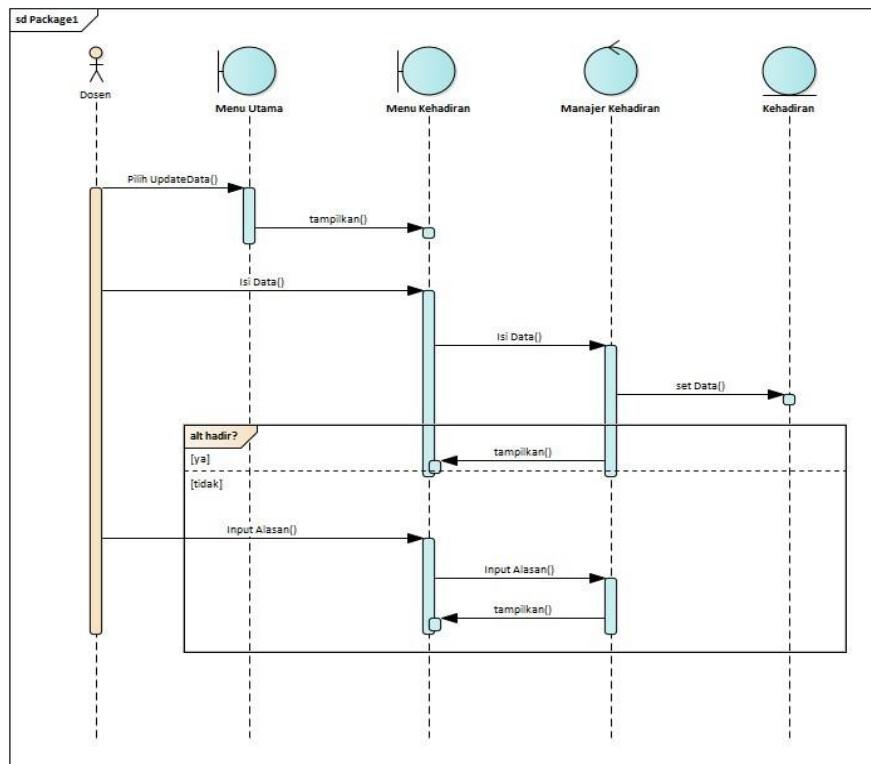
Gambar 4. 11 Sequence Diagram Add User

### 4. Sequence Diagram Mengelola Jadwal



Gambar 4. 12 Sequence Diagram Mengelola Jadwal

##### 5. Sequence Diagram Meng-update Informasi Kehadiran



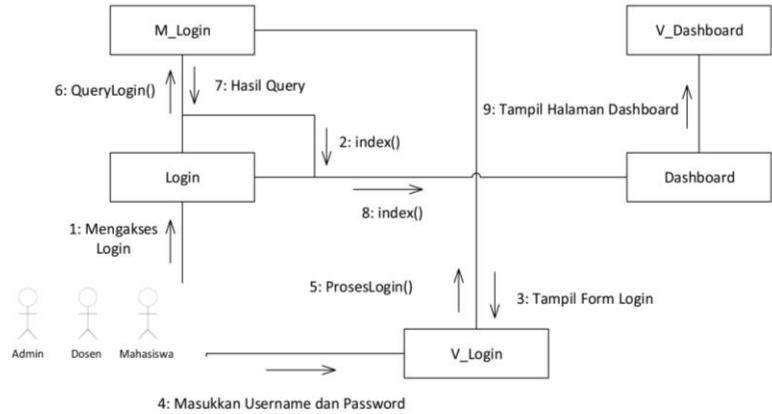
Gambar 4. 13 Sequence Diagram Mengupdate Informasi Kehadiran

#### 4.4.5 Collaboration Diagram

*Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.

Berikut ini adalah *collaboration diagram* yang dibuat pada perancangan aplikasi Kehadiran Dosen.

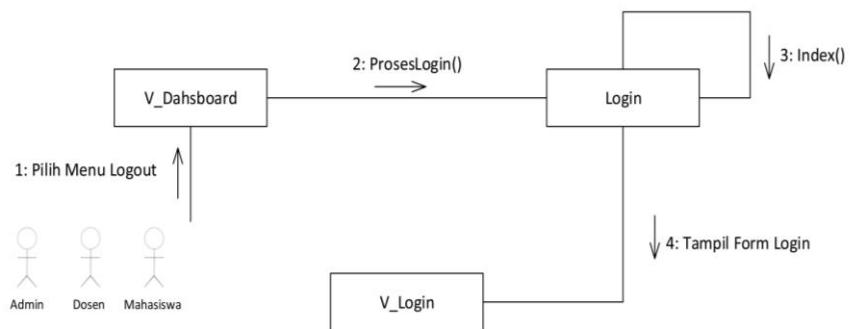
## 1. Collaboration Diagram Login



Gambar 4. 14 Collaboration Diagram Login

Pada *collaboration diagram login* menjelaskan bagaimana proses *login*. Ketiga aktor, yaitu admin, dosen, dan mahasiswa akan mengakses *form login* dan masuk ke dalam *dashboard* dengan memasukkan *username* dan *password* pada *user interface login* akan mengecek data yang dimasukkan oleh aktor.

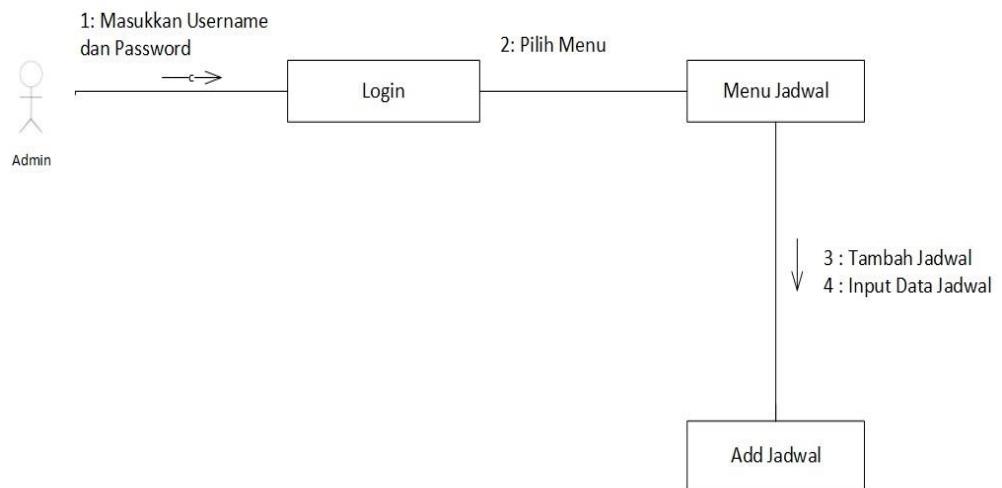
## 2. Collaboration Diagram Logout



Gambar 4. 15 Collaboration Diagram Logout

Pada *collaboration diagram logout* ini menjelaskan bagaimana proses atau alur *logout*. Ketiga aktor, yaitu admin, dosen, dan mahasiswa akan mengakses menu *logout* dalam *dashboard* sistem dan kemudian akan keluar dari sistem, serta akan kembali ke halaman *login*.

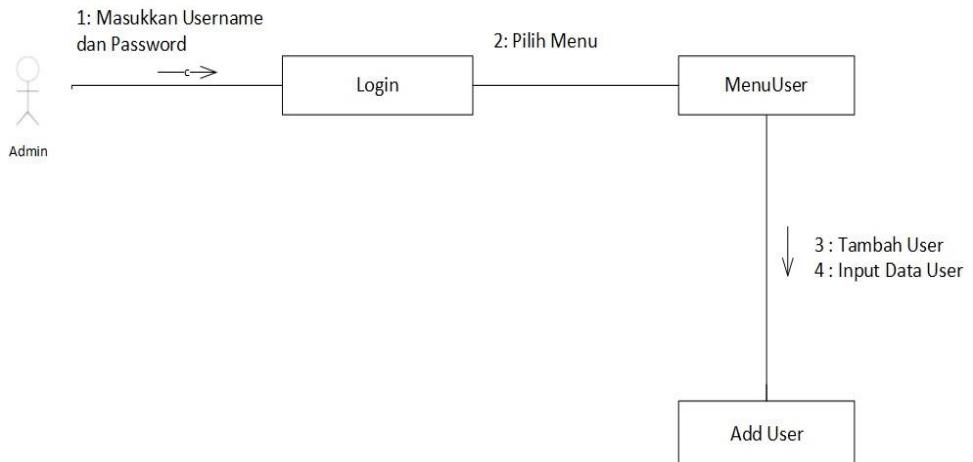
### 3. Collaboration Diagram Mengelola Jadwal



*Gambar 4. 16 Collaboration Diagram Mengelola Jadwal*

Pada *collaboration diagram* mengelola data jadwal mengajar menjelaskan alur atau proses mengelola data tersebut. Aktor yang akan mengelola data adalah admin.

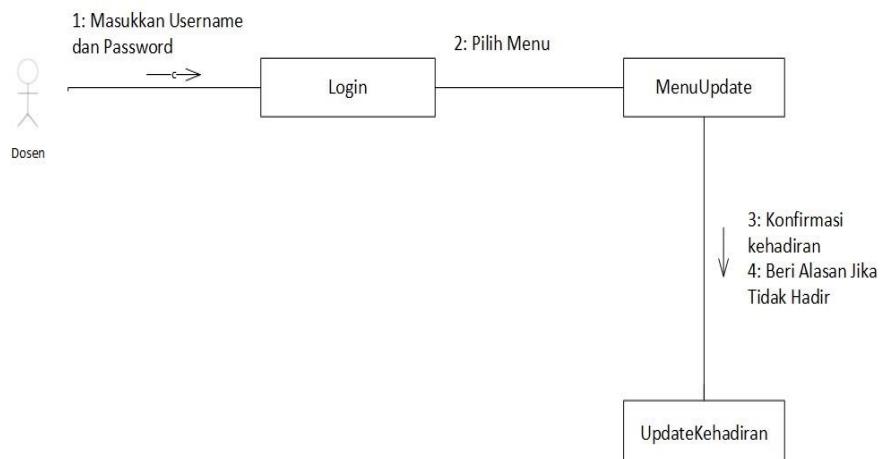
#### 4. *Collaboration Diagram Add User*



Gambar 4. 17 Sequence Diagram Add User

Pada *collaboration diagram add user* ini menjelaskan bagaimana proses menambahkan *user*. Aktor yang akan mengelola data adalah admin.

##### 5. Collaboration Diagram: Meng-update Kehadiran Dosen



Gambar 4. 18 Collaboration Diagram Mengupdate Kehadiran Dosen

Pada *collaboration diagram* meng-update kehadiran dosen ini menjelaskan bagaimana proses *update* kehadiran. Saat proses penginputan kehadiran, terdapat dua pilihan, yaitu hadir dan tidak hadir. Jika tidak hadir, dosen dapat memberikan alasan tidak dapat mengajar perkuliahan,

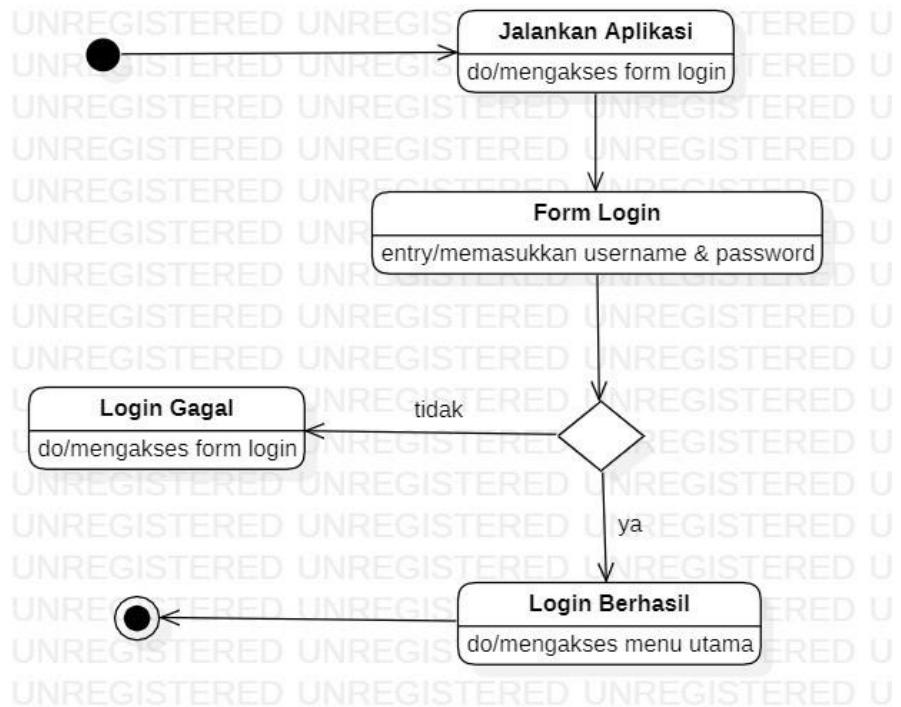
#### 4.4.6 *Statechart Diagram*

*Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*)[17].

Dalam *UML*, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah[17].

Berikut ini adalah beberapa *statechart diagram* dari Aplikasi Kehadiran Dosen.

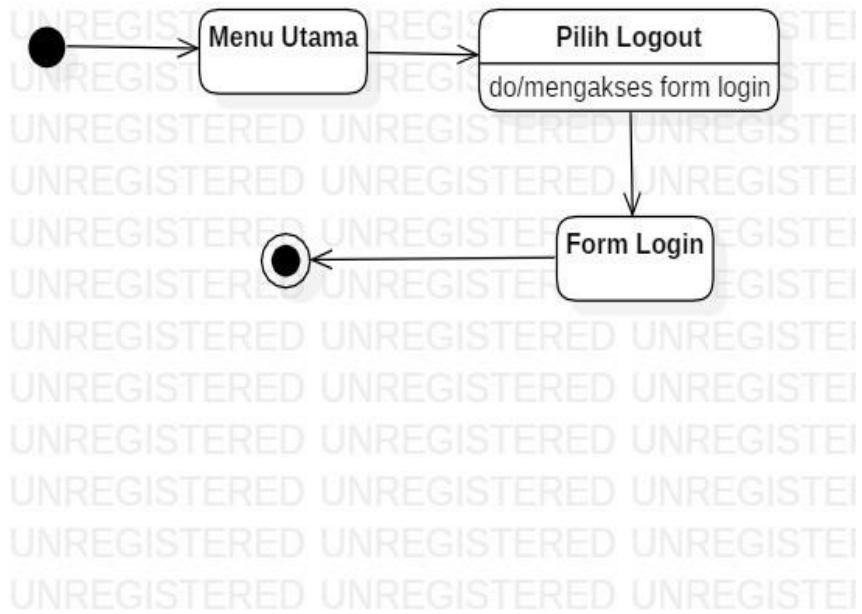
##### 1. *Statechart Diagram Login*



Gambar 4. 19 Statechart Diagram: Login

Pada menu *login* akan menjalankan aplikasi dan mengakses *form login*. Apabila data valid maka *login* berhasil, jika data tidak valid maka *login* gagal.

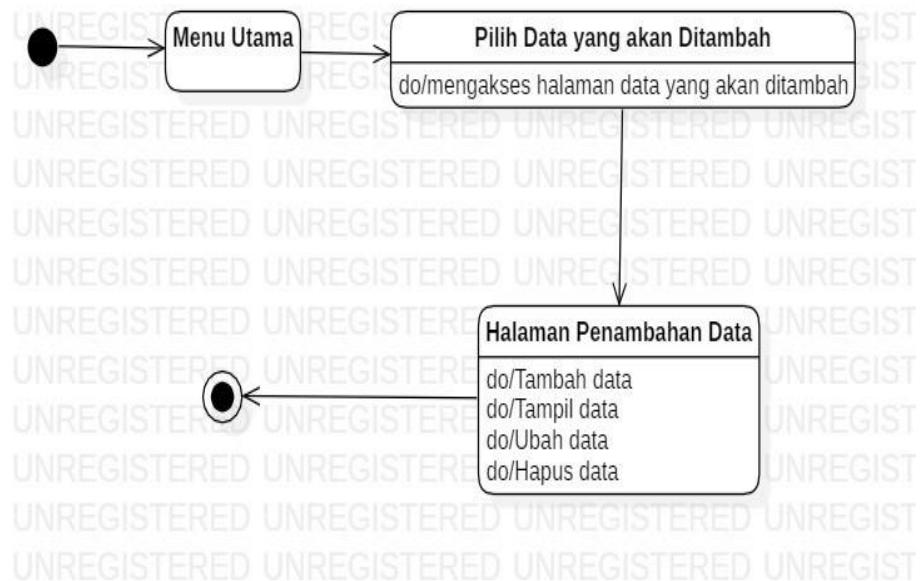
## 2. Statechart Diagram: Logout



Gambar 4. 20 Statechart Diagram: Logout

Pada menu *logout* aktor akan memilih menu *logout* dan kemudian mengakses *form login*.

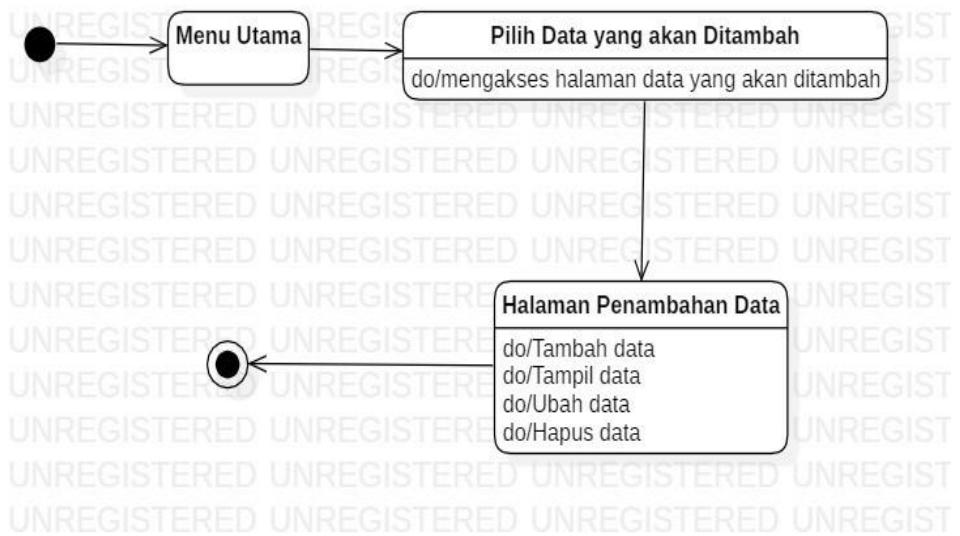
### 3. Statechart Diagram: Mengelola Data Jadwal



*Gambar 4. 21 Statechart Diagram Add User*

Pada menu mengelola data jadwal, aktor akan masuk ke menu utama dan memilih menu jadwal, lalu aktor dapat menambahkan data jadwal pada menu jadwal.

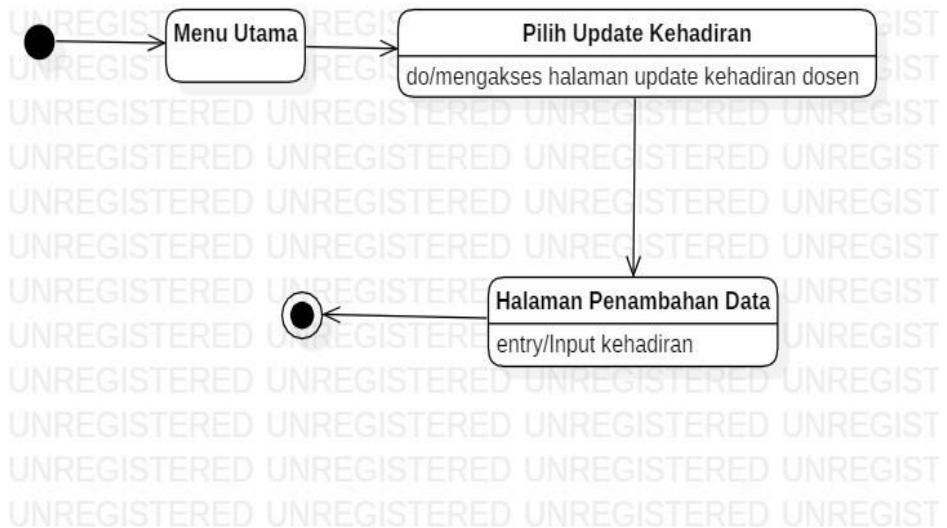
#### 4. Statechart Diagram Add User



Gambar 4. 22 Statechart Diagram Add User

Pada menu *add user*, aktor akan masuk ke menu utama dan memilih menu *user*, lalu aktor dapat menambahkan data *user* pada menu *user*.

##### 5. Statechart Diagram Update Kehadiran



Gambar 4. 23 Statechart Diagram Update Kehadiran

Pada menu *update* kehadiran dosen, aktor akan masuk ke menu utama dan memilih menu *update*, lalu aktor akan masuk ke halaman menu *update* kehadiran dosen.

#### 4.4.7 Component Diagram

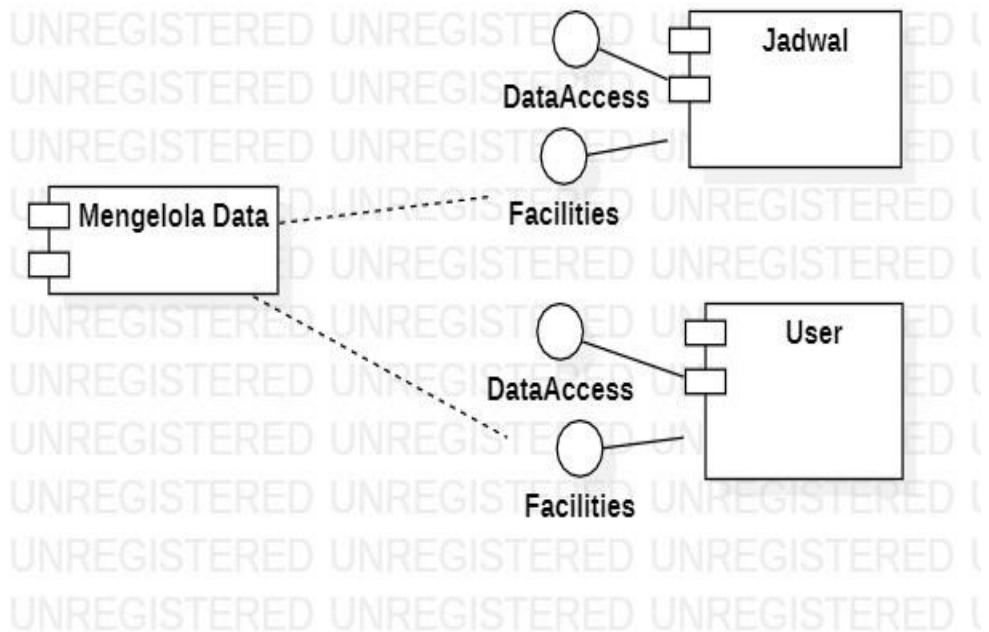
*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*[17].

Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain[17].

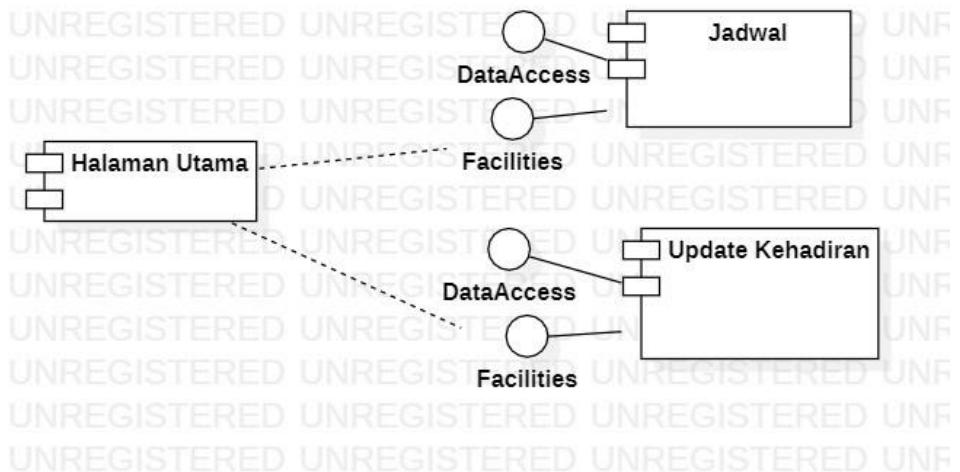
Diagram komponen fokus pada komponen sistem yang dibutuhkan disuatu sistem, berikut ini adalah gambar *component diagram* dari aplikasi Kehadiran Dosen.

### 1. Component Diagram Admin



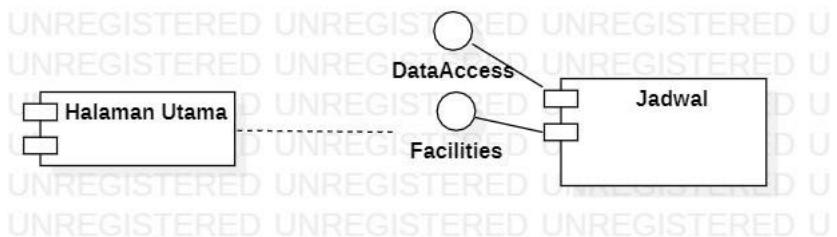
Gambar 4. 24 Component Diagram Admin

### 2. Component Diagram Dosen



Gambar 4. 25 Component Diagram Dosen

### 3. Component Diagram: Mahasiswa

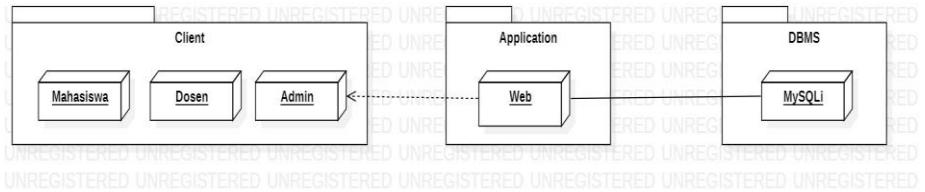


Gambar 4. 26 Component Diagram Mahasiswa

### 4.4.8 Deployment Diagram

*Deployment/physical diagram* menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau peranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server,

dan hal-hal lain yang bersifat fisikal. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar *node*(misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini. Berikut ini adalah *deployment diagram* yang menggambarkan susunan fisik dan perangkat lunak dan sistem.

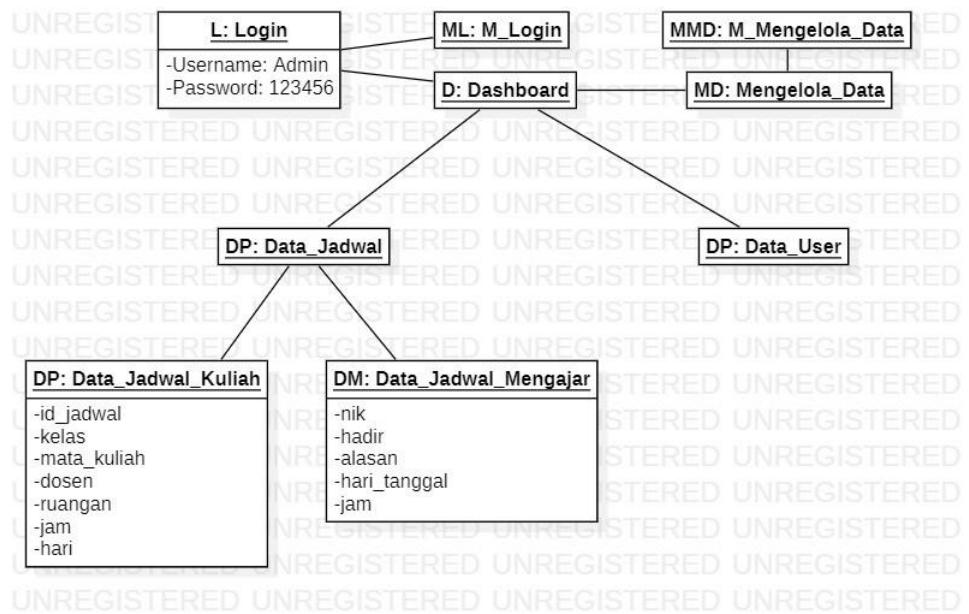


Gambar 4. 27 Deployment Diagram

#### 4.4.9 Object Diagram

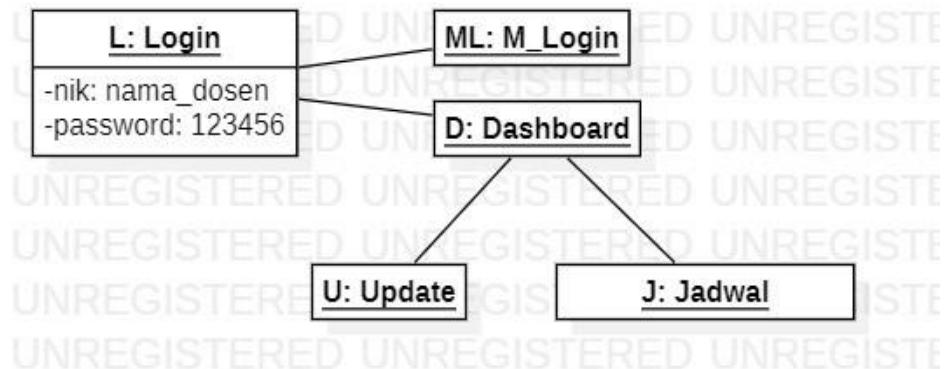
Berikut ini adalah *object diagram* atau diagram objek yang sudah dibuat dari aplikasi Kehadiran Dosen.

##### 1. Object Diagram Admin



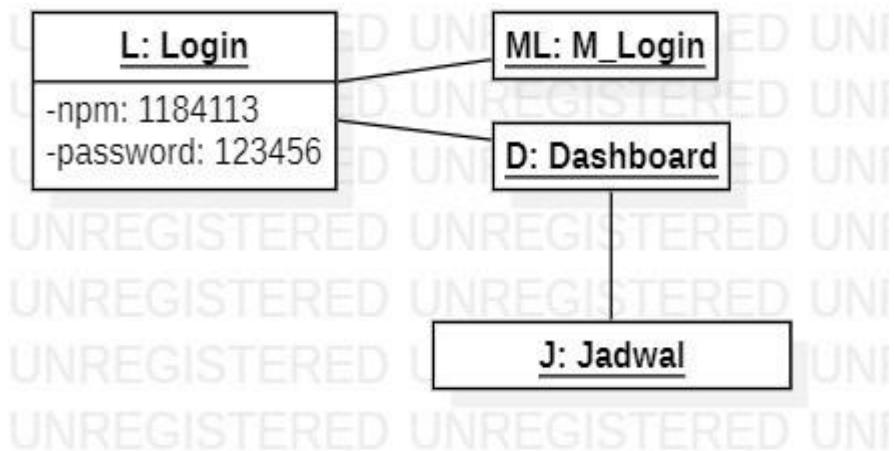
Gambar 4. 28 Object Diagram Admin

## 2. Object Diagram Dosen



Gambar 4. 29 Object Diagram Dosen

### 3. Object Diagram Mahasiswa

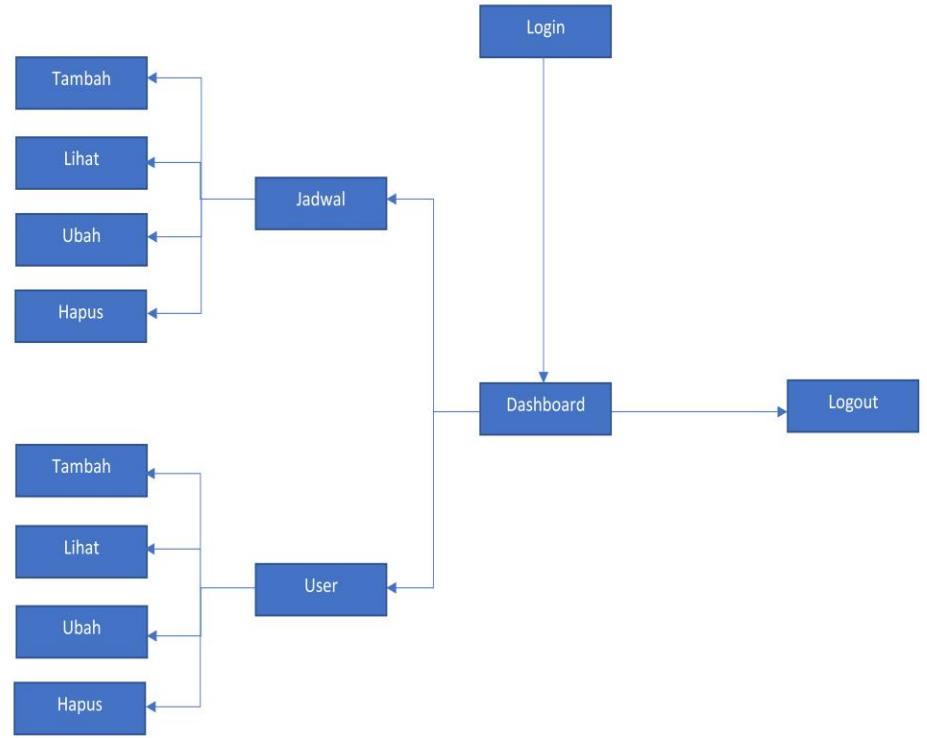


Gambar 4. 30 Object Diagram Mahasiswa

#### 4.4.10 Struktur Menu

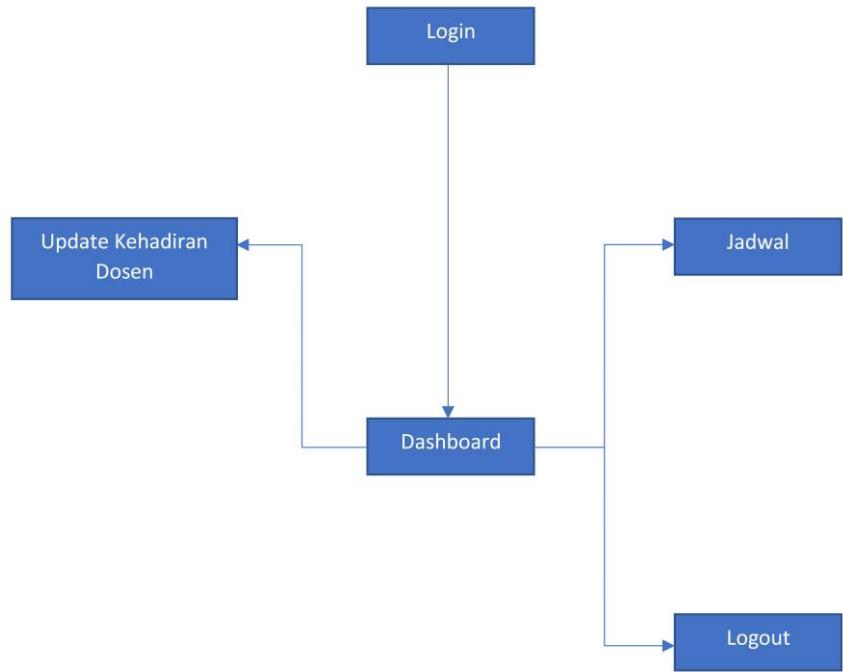
Struktur menu dibuat sebagai gambaran mengenai skema program yang akan dirancang. Berikut ini adalah struktur menu yang digunakan pada perancangan aplikasi Kehadiran Dosen.

##### 1. Struktur Menu Admin



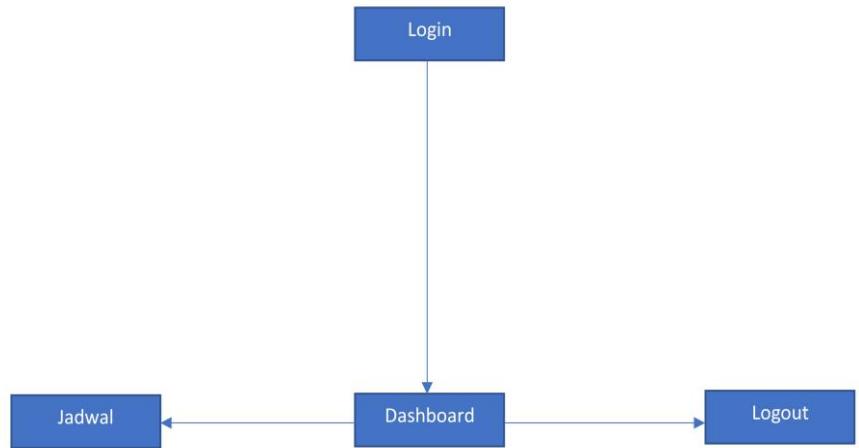
Gambar 4. 31 Struktur Menu Admin

## 2. Struktur Menu Dosen



Gambar 4. 32 Struktur Menu Dosen

### 3. Struktur Menu Mahasiswa



Gambar 4. 33 Struktur Menu Mahasiswa

# BAB V

## TAHAPAN MEMBUAT APLIKASI KEHADIRAN DOSEN

### 5.1 Membuat *Database*

Langkah pertama yang dilakukan dalam pembuatan aplikasi adalah membuat *database* terlebih dahulu. Caranya adalah pergi ke browser dan buka *localhost/phpMyAdmin*. Lalu, klik *new* dan *create database*. Kami membuat *database* dengan nama *hadir\_dosen*.



Gambar 5. 1 Membuat Database *hadir\_dosen*

Setelah membuat *database*, kami membuat beberapa *table* yang dibutuhkan. Berikut ini adalah table-tabel yang dibuat.

### 5.1.1 Tabel User Dosen

Berikut ini adalah *table* untuk mengelola *user* dosen. Fungsi dari mengelola *user* adalah bertujuan agar *user* dapat melakukan *login*.

The screenshot shows the MySQL Workbench interface with the 'dosens' table selected. The table has columns: id\_dosen, nik, nama, password, and hak\_akses. The data includes various records such as Rolly Maulana Awangga, M. Yusril Helmi Setyawan, Mohamad Nurkamal Fauzan, Roni Habibi, Roni Andarsyah, Cahyo Prianto, Syafrial Fachri Pane, M. Harry K Saputra, Rd. Nuraini Siti Fatonah, Nisa Hanum Harani, Woro Isti Rahayu, and Noviana Riza. The 'password' column contains the value 'sariash54' for most users. The 'hak\_akses' column contains values 0 or 1. Navigation buttons like 'Edit', 'Copy', and 'Delete' are visible at the bottom of the table view.

	<u>id_dosen</u>	<u>nik</u>	<u>nama</u>	<u>password</u>	<u>hak_akses</u>
e	5e428ca9e2c3f	11786219	Rolly Maulana Awangga, S.T., M.T.	sariash54	0
e	5e428d56e4245	11374163	M. Yusril Helmi Setyawan, S.Kom., M.Kom.	sariash54	0
e	5e428d7a6a72b	11380159	Mohamad Nurkamal Fauzan, S.T., M.T.	sariash54	0
e	5e428da0b344a	10378069	Roni Habibi, S.Kom., M.T.	sariash54	0
e	5e428dc5a5322	11588193	Roni Andarsyah, S.T., M.Kom.	sariash54	0
e	5e428de475f98	11784222	Cahyo Prianto, S.Pd., M.T.	sariash54	0
e	5e428dfecaa1c	11788233	Syafrial Fachri Pane, S.T., M.T.I.,EBDP	sariash54	0
e	5e428e2725068	21388109	M. Harry K Saputra, S.T., M.T.I	sariash54	0
e	5e428e3e2d63a	21772187	Rd. Nuraini Siti Fatonah, S.S., M.Hum.	sariash54	0
e	5e428e5641d42	11789223	Nisa Hanum Harani, S.Kom., M.T.	sariash54	0
e	5e428e72e8839	10579081	Woro Isti Rahayu, S.T., M.T.	sariash54	0
e	5e428e8962a19	10378065	Noviana Riza, S.Si., M.T.	sariash54	0

Gambar 5. 2 Tabel User Dosen

Pada *table* ini memiliki beberapa atribut, yaitu sebagai berikut.

1. *Id Dosen*
2. *Nik*
3. *Nama*

4. *Password*
5. *Hak Akses*

### 5.1.2 Tabel *User Mahasiswa*

Berikut ini adalah *table* untuk mengelola *user* dosen. Fungsi dari mengelola *user* adalah bertujuan agar *user* dapat melakukan *login*.

	<a href="#">id_mahasiswa</a>	<a href="#">npm</a>	<a href="#">nama</a>	<a href="#">password</a>	<a href="#">hak_akses</a>
1	5e3fc537e20e7	1184112	Ferdy Berliano Putra	sariasih54	0
2	5e429153328cf	1184113	Ida Fatrini Putri	sariasih54	0

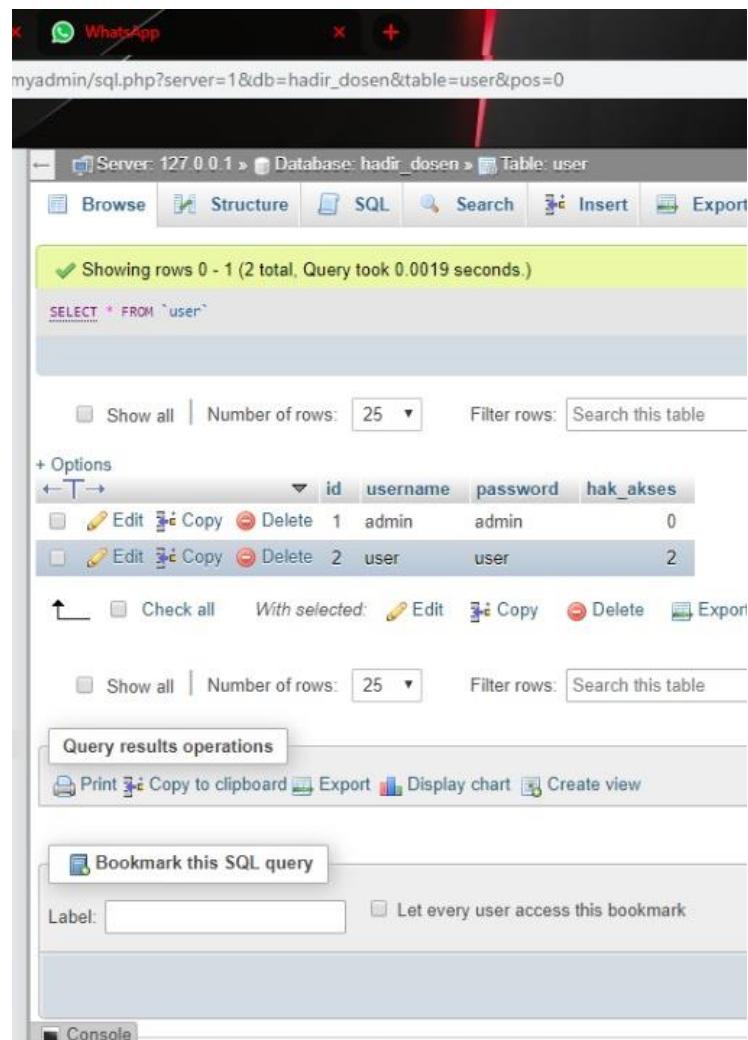
Gambar 5. 3 Tabel User Mahasiswa

Pada *table* ini memiliki beberapa atribut, yaitu sebagai berikut.

1. *Id Mahasiswa*
2. *Npm*
3. Nama
4. *Password*
5. Hak Akses

### 5.1.3 Tabel *User Admin*

Berikut ini adalah *table* untuk mengelola *user* admin. Fungsi dari mengelola *user* adalah bertujuan agar *user* dapat melakukan *login*.



Gambar 5. 4 Tabel User Admin

Pada *table* ini memiliki beberapa atribut, yaitu sebagai berikut.

1. *Id*
2. *Username*
3. *Password*
4. *Hak Akses*

### 5.1.4 Tabel Jadwal Semester Ganjil 2018/2019

Berikut ini adalah *table* untuk mengelola jadwal semester ganjil tahun 2018/2019. Fungsi dari mengelola jadwal ini adalah bertujuan agar *user* dapat melihat jadwal perkuliahan.

The screenshot shows a MySQL Workbench interface with the following details:

- URL:** hadir\_dosen&table=ganjil1819&pos=0
- Toolbar:** SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, Triggers.
- Message Bar:** Query took 0.0020 seconds.
- Table View:** A table named 'jadwal\_dosen' is displayed with the following columns and data:

id_jadwal	pengajar	matkul	ruang	hari	jam	kelas	tanggal	status	alasan
5e4d9dac6e1d2	Mohamad Nurkamal Fauzan, S.T., M.T.	Algoritma I	310	Selasa	07.00-10.20	D4 TI 1A	2019-01-22	Tidak	Saya sakit, cari pengganti
- Action Buttons:** selected, Edit, Copy, Delete, Export.
- Row Selection:** rows: 25, Filter rows: Search this table.
- Bottom Buttons:** Export, Display chart, Create view.
- Bookmark Area:** Let every user access this bookmark, Bookmark this SQL query.

Gambar 5. 5 Tabel Jadwal Semester Ganjil 2018/2019

Pada *table* ini memiliki beberapa atribut, yaitu sebagai berikut.

1. *Id\_jadwal*
2. Pengajar
3. Matkul
4. Ruang

5. Hari
6. Jam
7. Kelas
8. Tanggal
9. Status
10. Keterangan

### 5.1.5 Tabel Jadwal Semester Genap 2018/2019

Berikut ini adalah *table* untuk mengelola jadwal semester genap tahun 2018/2019. Fungsi dari mengelola jadwal ini adalah bertujuan agar *user* dapat melihat jadwal perkuliahan.

The screenshot shows the MySQL Workbench interface with the database 'hadir\_dosen' and table 'genap1819'. The table structure includes columns: id\_jadwal, pengajar, matkul, ruang, hari, jam, kelas, tanggal, status, and alasan. The data consists of eight rows, each representing a student's schedule entry. The 'status' column indicates attendance ('Ya') and the 'alasan' column indicates the reason for attendance ('Hadir').

	id_jadwal	pengajar	matkul	ruang	hari	jam	kelas	tanggal	status	alasan
e	5e4df1c87a5f	Rd. Nuraini Siti Fatonah, S.S., M.Hum.	GENERAL ENGLISH II	117	Senin	13.00- 16.20	D4 TI 2A	2019-06- 24	Ya	Hadir
e	5e4dfd419a7b08	Cahyo Prianto, S.Pd., M.T.	Aljabar Linear	108	Selasa	07.00- 08.40	D4 TI 2A	2019-06- 25	Ya	Hadir
e	5e4dfd65f348f	Mohamad Nurkamal Fauzan, S.T., M.T.	ALGORITMA II	309	Selasa	08.40- 10.20	D4 TI 2A	2019-06- 25	Ya	Hadir
e	5e4dfd95e80c0	Syafrial Fachri Pane, S.T., M.T.I_EBDP	BASIS DATA I	311	Rabu	07.00- 10.20	D4 TI 2A	2019-06- 26	Ya	Hadir
e	5e4dfdd32429b	M. Harry K Saputra, S.T., M.T.I	SISTEM OPRASI	318	Kamis	08.40- 10.20	D4 TI 2A	2019-06- 27	Ya	Hadir
e	5e4dfe18e684a	Woro Isti Rahayu, S.T., M.T.	ANALISIS DAN PERANCANGAN SISTEM INFORMASI	105	Kamis	13.00- 14.40	D4 TI 2A	2019-06- 27	Ya	Hadir

Gambar 5. 6 Tabel Jadwal Semester Genap 2018/2019

Pada *table* ini memiliki beberapa atribut, yaitu sebagai berikut.

1. *Id\_jadwal*
2. Pengajar
3. Matkul
4. Ruang
5. Hari
6. Jam
7. Kelas
8. Tanggal
9. Status
10. Keterangan

### **5.1.6 Tabel Jadwal Semester Ganjil 2019/2020**

Berikut ini adalah *table* untuk mengelola jadwal semester ganjil tahun 2019/2020. Fungsi dari mengelola jadwal ini adalah bertujuan agar *user* dapat melihat jadwal perkuliahan.

The screenshot shows the MySQL Workbench interface with the following table data:

	<u>id_jadwal</u>	<u>pengajar</u>	<u>matkul</u>	<u>ruang</u>	<u>hari</u>	<u>jam</u>	<u>kelas</u>	<u>tanggal</u>	<u>status</u>	<u>alasan</u>
1	5e4df888d81f6	M. Yusril Helmi Setyawan, S.Kom., M.Kom.	JARINGAN KOMPUTER	318	Senin	07.00-10.20	D4 TI 2A	2020-01-20	Tidak	Saya Rapat, Cari Hari Pengantin
2	5e4df8bb2c8f9	Mohamad Nurkamal Fauzan, S.T., M.T.	PEMROGRAMAN II	312	Senin	13.00-18.00	D4 TI 2A	2020-01-20	Ya	Hadir
3	5e4df8e9755a1	Noviana Riza, S.Si., M.T.	DESAIN INTERAKSI	103	Selasa	13.00-14.40	D4 TI 2A	2020-01-21	Ya	Hadir
4	5e4df926bb107	Rd. Nuraini Siti Fatonah, S.S., M.Hum.	GENERAL ENGLISH III	117	Rabu	09.30-12.00	D4 TI 2A	2020-01-22	Ya	Hadir
5	5e4df95376e2b	Woro Isti Rahayu, S.T., M.T.	REKAYASA PERANGKAT LUNAK	117	Jumat	08.40-12.00	D4 TI 2A	2020-01-24	Ya	Hadir
6	5e4df9af24f67	Syafrial Fachri Pane, S.T., M.T.I., EBDP	BASIS DATA II	311	Kamis	13.00-18.00	D4 TI 2A	2020-01-23	Ya	Hadir
7	5e4df9d4c67f2	Woro Isti Rahayu, S.T., M.T.	METODOLOGI PENELITIAN	117	Jumat	13.00-14.40	D4 TI 2A	2020-01-24	Ya	Hadir

Buttons at the bottom: selected, Edit, Copy, Delete, Export.

Gambar 5. 7 Tabel Jadwal Semester Ganjil 2019/2020

Pada *table* ini memiliki beberapa atribut, yaitu sebagai berikut.

1. *Id\_jadwal*
2. Pengajar
3. Matkul
4. Ruang
5. Hari
6. Jam
7. Kelas
8. Tanggal
9. Status

## 10. Keterangan

### 5.1.7 Tabel Jadwal Semester Genap 2019/2020

Berikut ini adalah *table* untuk mengelola jadwal semester genap tahun 2019/2020. Fungsi dari mengelola jadwal ini adalah bertujuan agar *user* dapat melihat jadwal perkuliahan.

The screenshot shows a MySQL database interface with the following details:

- Database: hadir\_dosen
- Table: jadwalaja
- Query took 0.0019 seconds.
- Table structure:

id_jadwal	nik	pengajar	matkul	ruang	hari	jam	kelas	tanggal	status	alasan
5e4df6712492b	10579081	Woro Isti Rahayu, S.T., M.T.	Rekayasa Perangkat Lunak I	311	Jumat	13.00-16.20	D4 TI 2A	2020-02-28	Ya	Hadir
5e4df6bf3bf4c	21388109	M. Harry K Saputra, S.T., M.T.I	Pemrograman III	315	Kamis	13.00-18.00	D4 TI 2A	2020-02-27	Tidak	Saya Rapat, Cari Hari Pengganti
5e4df7037e6a1	10378065	Noviana Riza, S.Si., M.T.	ETIKA DAN MANAJEMEN PROFESI IT	111	Rabu	13.00-14.40	D4 TI 2A	2020-02-26	Ya	Hadir
5e4df74be4e05	10378065	Noviana Riza, S.Si., M.T.	CYBERPREENEURSHIP	106	Selasa	13.00-14.40	D4 TI 2A	2020-02-25	Ya	Hadir
5e4df783f2e50	11789223	Nisa Hanum Harani, S.Kom., M.T.	SISTEM ERP 1	317	Selasa	08.40-12.00	D4 TI 2A	2020-02-25	Ya	Hadir
5e4df7e2073f5	11374163	M. Yusril Helmi Setyawan, S.Kom., M.Kom.	ADMINISTRASI JARINGAN KOMPUTER	318	Senin	13.00-16.20	D4 TI 2A	2020-03-02	Tidak	Keluar Kota
5e4df8167cf01	10378069	Roni Habibi, S.Kom., M.T.	MANAJEMEN PROYEK	320	Senin	10.20-12.00	D4 TI 2A	2020-02-24	Ya	Hadir

- Buttons at the bottom: selected, Edit, Copy, Delete, Export.
- Bottom navigation: rows: 25, Filter rows: Search this table, Sort by key: None.

Gambar 5. 8 Tabel Jadwal Semester Genap 2019/2020

Pada *table* ini memiliki beberapa atribut, yaitu sebagai berikut.

1. *Id\_jadwal*

2. Pengajar
3. Matkul
4. Ruang
5. Hari
6. Jam
7. Kelas
8. Tanggal
9. Status
10. Keterangan

## 5.2 Membuat *Controller*

Pada *controller* dibagi menjadi 3, yaitu admin, dosen, dan mahasiswa. Berikut ini adalah *codingan* dari masing-masing *controller*.

### 5.2.1 *Controller Admin*

#### 5.2.1.1 File *Dosens.php*

```
<?php  
  
defined('BASEPATH') OR exit('No direct script ac  
cess allowed');  
  
class Dosens extends CI_Controller  
{  
    public function __construct()  
    {  
        parent::__construct();  
    }  
}
```

```

        $this->load->model("Dosen_model");
        $this->load->library('form_validation');

    }

public function index()
{
    $data["dosens"] = $this->Dosen_model-
>getAll();

    $this->load-
>view("admin/dosen/list", $data);
}

public function add()
{
    $dosen = $this->Dosen_model;
    $validation = $this->form_validation;
    $validation->set_rules($dosen->rules());

    if ($validation->run()) {
        $dosen->save();
        $this->session-
>set_flashdata('success', 'Berhasil disimpan');
    }

    $this->load-
>view("admin/dosen/new_form");
}

```

```

public function edit($id = null)
{
    if (!isset($id)) redirect('admin/dosens');
}

$dosen = $this->Dosen_model;
$validation = $this->form_validation;
$validation->set_rules($dosen->rules());

if ($validation->run()) {
    $dosen->update();
    $this->session->set_flashdata('success', 'Berhasil disimpan');
}

$data["dosen"] = $dosen->getById($id);
if (!$data["dosen"]) show_404();

$this->load-
>view("admin/dosen/edit_form", $data);
}

public function delete($id=null)
{
    if (!isset($id)) show_404();
}

```

```

        if ($this->Dosen_model->delete($id)) {
            redirect(site_url('admin/dosens'));
        }
    }
}

```

### **5.2.1.2 File *Ganjil1819s.php***

```

<?php

defined('BASEPATH') OR exit('No direct script ac
cess allowed');

class Ganjil1819s extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("Ganjil1819_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["ganjil1819s"] = $this-
>Ganjil1819_model->getAll();
        $this->load-
>view("admin/ganjil1819/list", $data);
    }
}

```

```

    }

public function add()
{
    $ganjill1819 = $this->Ganjill1819_model;
    $validation = $this->form_validation;
    $validation->set_rules($ganjill1819-
>rules());

    if ($validation->run ()) {
        $ganjill1819->save();
        $this->session-
>set_flashdata('success', 'Berhasil disimpan');
    }

    $this->load-
>view("admin/ganjill1819/new_form");
}

public function edit($id = null)
{
    if (!isset($id)) redirect('admin/ganjill
819s');

    $ganjill1819 = $this->Ganjill1819_model;
    $validation = $this->form_validation;
}

```

```

        $validation->set_rules($ganjill1819-
>rules());

        if ($validation->run()) {
            $ganjill1819->update();
            $this->session-
>set_flashdata('success', 'Berhasil disimpan');
        }

        $data["ganjill1819"] = $ganjill1819-
>getById($id);
        if (!$data["ganjill1819"]) show_404();

        $this->load-
>view("admin/ganjill1819/edit_form", $data);
    }

    public function delete($id=null)
    {
        if (!isset($id)) show_404();

        if ($this->Ganjill1819_model-
>delete($id)) {
            redirect(site_url('admin/ganjill1819s
'));
        }
    }
}

```

### **5.2.1.3 File *Ganjil1920s.php***

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Ganjil1920s extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("Ganjil1920_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["ganjil1920s"] = $this->Ganjil1920_model->getAll();
        $this->load->view("admin/ganjil1920/list", $data);
    }

    public function add()
    {
```

```

$ganjill1920 = $this->Ganjill1920_model;
$validation = $this->form_validation;
$validation->set_rules($ganjill1920-
>rules());

if ($validation->run()) {
    $ganjill1920->save();
    $this->session-
>set_flashdata('success', 'Berhasil disimpan');
}

$this->load-
>view("admin/ganjill1920/new_form");
}

public function edit($id = null)
{
    if (!isset($id)) redirect('admin/ganjill
920s');

$ganjill1920 = $this->Ganjill1920_model;
$validation = $this->form_validation;
$validation->set_rules($ganjill1920-
>rules());

if ($validation->run()) {
    $ganjill1920->update();
}

```

```

        $this->session-
>set_flashdata('success', 'Berhasil disimpan');

    }

    $data["ganjill1920"] = $ganjill1920-
>getById($id);

    if (!$data["ganjill1920"]) show_404();

    $this->load-
>view("admin/ganjill1920/edit_form", $data);

}

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->Ganjill1920_model-
>delete($id)) {
        redirect(site_url('admin/ganjill1920s
'));
    }
}
}

```

#### **5.2.1.4 File *Genap1819s.php***

<?php  
**110 | V Tahapan Membuat Aplikasi Kehadiran Dosen**

```

defined('BASEPATH') OR exit('No direct script ac-
cess allowed');

class Genap1819s extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("Genap1819_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["genap1819s"] = $this-
>Genap1819_model->getAll();
        $this->load-
>view("admin/genap1819/list", $data);
    }

    public function add()
    {
        $genap1819 = $this->Genap1819_model;
        $validation = $this->form_validation;
        $validation->set_rules($genap1819-
>rules());
    }
}

```

```

        if ($validation->run()) {
            $genap1819->save();
            $this->session-
>set_flashdata('success', 'Berhasil disimpan');
        }

        $this->load-
>view("admin/genap1819/new_form");
    }

    public function edit($id = null)
    {
        if (!isset($id)) redirect('admin/genap18
19s');

        $genap1819 = $this->Genap1819_model;
        $validation = $this->form_validation;
        $validation->set_rules($genap1819-
>rules());
    }

    if ($validation->run()) {
        $genap1819->update();
        $this->session-
>set_flashdata('success', 'Berhasil disimpan');
    }
}

```

```

        $data["genap1819"] = $genap1819-
>getById($id);

        if (!$data["genap1819"]) show_404();

        $this->load-
>view("admin/genap1819/edit_form", $data);

    }

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->Genap1819_model-
>delete($id)) {
        redirect(site_url('admin/genap1819s'
));
    }
}

```

File Jadwalajas.php

```

<?php

defined('BASEPATH') OR exit('No direct script ac
cess allowed');

```

```

class Jadwalajas extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("Jadwalaja_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["jadwalajas"] = $this-
>Jadwalaja_model->getAll();
        $this->load-
>view("admin/jadwalaja/list", $data);
    }

    public function add()
    {
        $jadwalaja = $this->Jadwalaja_model;
        $validation = $this->form_validation;
        $validation->set_rules($jadwalaja-
>rules());
        if ($validation->run()) {
            $jadwalaja->save();
        }
    }
}

```

```

        $this->session-
>set_flashdata('success', 'Berhasil disimpan');

    }

    $this->load-
>view("admin/jadwalaja/new_form");

}

public function edit($id = null)
{
    if (!isset($id)) redirect('admin/jadwala
jas');

    $jadwalaja = $this->Jadwalaja_model;
    $validation = $this->form_validation;
    $validation->set_rules($jadwalaja-
>rules());

    if ($validation->run()) {
        $jadwalaja->update();
        $this->session-
>set_flashdata('success', 'Berhasil disimpan');

    }

    $data["jadwalaja"] = $jadwalaja-
>getById($id);
    if (!$data["jadwalaja"]) show_404();
}

```

```

        $this->load-
>view("admin/jadwalaja/edit_form", $data);
    }

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->Jadwalaja_model-
>delete($id)) {
        redirect(site_url('admin/jadwalajas'));
    }
}

```

### **5.2.1.5 File Mahasiswa.php**

```

<?php

defined('BASEPATH') OR exit('No direct script ac-
cess allowed');

class Mahasiswa extends CI_Controller
{
    public function __construct()
    {

```

```

parent::__construct();

$this->load->model("Mahasiswa_model");

$this->load->library('form_validation');

}

public function index()
{
    $data["mahasiswas"] = $this-
>Mahasiswa_model->getAll();

    $this->load-
>view("admin/mahasiswa/list", $data);
}

public function add()
{
    $mahasiswa = $this->Mahasiswa_model;
    $validation = $this->form_validation;
    $validation->set_rules($mahasiswa-
>rules());
}

if ($validation->run()) {
    $mahasiswa->save();
    $this->session-
>set_flashdata('success', 'Berhasil disimpan');
}

```

```

        $this->load-
>view("admin/mahasiswa/new_form");
}

public function edit($id = null)
{
    if (!isset($id)) redirect('admin/mahasiswa');

    $mahasiswa = $this->Mahasiswa_model;
    $validation = $this->form_validation;
    $validation->set_rules($mahasiswa-
>rules());

    if ($validation->run()) {
        $mahasiswa->update();
        $this->session-
>set_flashdata('success', 'Berhasil disimpan');
    }

    $data["mahasiswa"] = $mahasiswa-
>getById($id);
    if (!$data["mahasiswa"]) show_404();

    $this->load-
>view("admin/mahasiswa/edit_form", $data);
}

```

```

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->Mahasiswa_model->delete($id)) {
        redirect(site_url('admin/mahasiswas'));
    }
}

```

### **5.2.1.6 File *Overview.php***

```

<?php

class Overview extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
    }

    public function index()
    {
        // load view admin/overview.php

```

```
        $this->load->view("admin/overview");

    }

}
```

## 5.2.2 Controller Dosen

### 5.2.2.1 File *Ganjil1819s.php*

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Ganjil1819s extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("Ganjil1819_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["ganjil1819s"] = $this->Ganjil1819_model->getAll();
        $this->load->view("dosen/ganjil1819/list", $data);
    }
}

120 | V Tahapan Membuat Aplikasi Kehadiran Dosen
```

```

}

public function add()
{
    $ganjill1819 = $this->Ganjill1819_model;
    $validation = $this->form_validation;
    $validation->set_rules($ganjill1819-
>rules());
}

if ($validation->run ()) {
    $ganjill1819->save();
    $this->session-
>set_flashdata('success', 'Berhasil disimpan');
}

$this->load-
>view("dosen/ganjill1819/new_form");
}

public function edit($id = null)
{
    if (!isset($id)) redirect ('dosen/ganjill
819s');

    $ganjill1819 = $this->Ganjill1819_model;
    $validation = $this->form_validation;
}

```

```

        $validation->set_rules($ganjill1819-
>rules());

        if ($validation->run()) {
            $ganjill1819->update();
            $this->session-
>set_flashdata('success', 'Berhasil disimpan');
        }

        $data["ganjill1819"] = $ganjill1819-
>getById($id);
        if (!$data["ganjill1819"]) show_404();

        $this->load-
>view("dosen/ganjill1819/edit_form", $data);
    }

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->Ganjill1819_model-
>delete($id)) {
        redirect(site_url('dosen/ganjill1819s
'));
    }
}

```

### **5.2.2.2 File *Ganjil1920s.php***

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Ganjil1920s extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("Ganjil1920_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["ganjil1920s"] = $this->Ganjil1920_model->getAll();
        $this->load->view("dosen/ganjil1920/list", $data);
    }

    public function add()
    {
```

```

$ganjill1920 = $this->Ganjill1920_model;
$validation = $this->form_validation;
$validation->set_rules($ganjill1920-
>rules());

if ($validation->run()) {
    $ganjill1920->save();
    $this->session-
>set_flashdata('success', 'Berhasil disimpan');
}

$this->load-
>view("dosen/ganjill1920/new_form");
}

public function edit($id = null)
{
    if (!isset($id)) redirect('dosen/ganjill
920s');

$ganjill1920 = $this->Ganjill1920_model;
$validation = $this->form_validation;
$validation->set_rules($ganjill1920-
>rules());

if ($validation->run()) {
    $ganjill1920->update();
}

```

```

        $this->session-
>set_flashdata('success', 'Berhasil disimpan');

    }

    $data["ganjill1920"] = $ganjill1920-
>getById($id);

    if (!$data["ganjill1920"]) show_404();

    $this->load-
>view("dosen/ganjill1920/edit_form", $data);

}

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->Ganjill1920_model-
>delete($id)) {
        redirect(site_url('dosen/ganjill1920s
'));
    }
}

```

### **5.2.2.3 File *Genap1819s.php***

<?php

```

defined('BASEPATH') OR exit('No direct script access allowed');

class Genap1819s extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("Genap1819_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["genap1819s"] = $this->Genap1819_model->getAll();
        $this->load->view("dosen/genap1819/list", $data);
    }

    public function add()
    {
        $genap1819 = $this->Genap1819_model;
        $validation = $this->form_validation;
        $validation->set_rules($genap1819->rules());
    }
}

```

```

        if ($validation->run()) {
            $genap1819->save();
            $this->session->set_flashdata('success', 'Berhasil disimpan');
        }

        $this->load->view("dosen/genap1819/new_form");
    }

    public function edit($id = null)
    {
        if (!isset($id)) redirect('dosen/genap1819s');

        $genap1819 = $this->Genap1819_model;
        $validation = $this->form_validation;
        $validation->set_rules($genap1819->rules());

        if ($validation->run()) {
            $genap1819->update();
            $this->session->set_flashdata('success', 'Berhasil disimpan');
        }

        $data["genap1819"] = $genap1819->getById($id);
    }
}

127 | V Tahapan Membuat Aplikasi Kehadiran Dosen

```

```

        if (!$_data["genap1819"]) show_404();

        $this->load-
>view("dosen/genap1819/edit_form", $_data);
    }

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->Genap1819_model-
>delete($id)) {
        redirect(site_url('dosen/genap1819s'
));
    }
}

```

#### **5.2.2.4 File *Hadirs.php***

```

<?php

defined('BASEPATH') OR exit('No direct script ac
cess allowed');

class Hadirs extends CI_Controller
{
    public function __construct()

```

```

    {

        parent::__construct();

        $this->load->model("Jadwalaja_model");

        $this->load->library('form_validation');

    }

public function index()

{

    $id = $this->session->userdata("nama");

    $data["jadwalajas"] = $this-
>Jadwalaja_model->getByNik($id);

    $this->load-
>view("dosen/jadwalaja/list", $data);

}

public function add()

{

    $jadwalaja = $this->Jadwalaja_model;

    $validation = $this->form_validation;

    $validation->set_rules($jadwalaja-
>rules());




    if ($validation->run()) {

        $jadwalaja->save();

        $this->session-
>set_flashdata('success', 'Berhasil disimpan');

    }

}

```

```

        $this->load-
>view("dosen/jadwalaja/new_form");
}

public function edit($id = null)
{
    if (!isset($id)) redirect('dosen/jadwala
jas');

    $jadwalaja = $this->Jadwalaja_model;
    $validation = $this->form_validation;
    $validation->set_rules($jadwalaja-
>rules());

    if ($validation->run()) {
        $jadwalaja->update();
        $this->session-
>set_flashdata('success', 'Berhasil disimpan');
    }

    $data["jadwalaja"] = $jadwalaja-
>getById($id);
    if (!$data["jadwalaja"]) show_404();

    $this->load-
>view("dosen/jadwalaja/edit_form", $data);
}

```

```

public function delete($id=null)
{
    if (!isset($id)) show_404();

    if ($this->Jadwalaja_model->delete($id)) {
        redirect(site_url('dosen/jadwalajas'));
    }
}

```

#### **5.2.2.5 File Jadwalajas.php**

```

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Jadwalajas extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("Jadwalaja_model");
        $this->load->library('form_validation');
    }
}

```

```

}

public function index()
{
    $data["jadwalajas"] = $this->Jadwalaja_model->getAll();
    $this->load->view("dosen/jadwalaja/list", $data);
}

public function add()
{
    $jadwalaja = $this->Jadwalaja_model;
    $validation = $this->form_validation;
    $validation->set_rules($jadwalaja->rules());
    if ($validation->run()) {
        $jadwalaja->save();
        $this->session->set_flashdata('success', 'Berhasil disimpan');
    }
    $this->load->view("dosen/jadwalaja/new_form");
}

```

```

public function edit($id = null)
{
    if (!isset($id)) redirect('dosen/jadwala
jas');

    $jadwalaja = $this->Jadwalaja_model;
    $validation = $this->form_validation;
    $validation->set_rules($jadwalaja-
>rules());

    if ($validation->run()) {
        $jadwalaja->update();
        $this->session-
>set_flashdata('success', 'Berhasil disimpan');
    }

    $data["jadwalaja"] = $jadwalaja-
>getById($id);
    if (!$data["jadwalaja"]) show_404();

    $this->load-
>view("dosen/jadwalaja/edit_form", $data);
}

public function delete($id=null)
{

```

```

        if (!isset($id)) show_404();

        if ($this->Jadwalaja_model-
>delete($id)) {
            redirect(site_url('dosen/jadwalajas'
));
        }
    }
}

```

#### **5.2.2.6 File *Overview.php***

```

<?php

class Overview extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
    }

    public function index()
    {
        // load view dsoen/overview.php
        $this->load->view("dosen/overview");
    }
}

```

### **5.2.3 Controller Mahasiswa**

#### **5.2.3.1 File *Kehadirans.php***

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Kehadirans extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model("kehadiran_model");
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data["kehadirans"] = $this->kehadiran_model->getAll();
        $this->load->view("mahasiswa/kehadiran/list", $data);
    }

    public function add()
```

```

    {

        $kehadiran = $this->kehadiran_model;

        $validation = $this->form_validation;

        $validation->set_rules($kehadiran-
>rules()) ;



        if ($validation->run()) {

            $kehadiran->save();

            $this->session-
>set_flashdata('success', 'Berhasil disimpan');

        }





        $this->load-
>view("mahasiswa/kehadiran/new_form");

    }







public function edit($id = null)

{

    if (!isset($id)) redirect('mahasiswa/Keh
adirans');





    $kehadiran = $this->kehadiran_model;

    $validation = $this->form_validation;

    $validation->set_rules($kehadiran-
>rules()) ;





    if ($validation->run()) {


```

```

        $kehadiran->update();
        $this->session-
>set_flashdata('success', 'Berhasil disimpan');
    }

    $data["kehadiran"] = $kehadiran-
>getById($id);
    if (!$data["kehadiran"])
        show_404();

    $this->load-
>view("mahasiswa/kehadiran/edit_form", $data);
}

public function delete($id=null)
{
    if (!isset($id))
        show_404();

    if ($this->kehadiran_model-
>delete($id)) {
        redirect(site_url('mahasiswa/Kehadirans'));
    }
}

```

### **5.2.3.2 File *Overview.php***

<?php

```
class Overview extends CI_Controller {  
    public function __construct()  
    {  
        parent::__construct();  
    }  
  
    public function index()  
    {  
        // load view mahasiswa/overview.php  
        $this->load->view("mahasiswa/overview");  
    }  
}
```

#### 5.2.4 Controller File *adminnnnnn.php*

```
<?php  
  
class Admin extends CI_Controller{  
  
    function __construct()  
    {  
        parent::__construct();  
  
        if($this->session->userdata('status') != "login"){  
            redirect(base_url("login"));  
        }  
    }  
}
```

```
        }

    }

function index() {
    $this->load->view('v_admin');
}

}
```

### 5.2.5 Controller File *Login.php*

```
<?php

class Login extends CI_Controller{

    function __construct() {
        parent::__construct();
        $this->load->model('m_login');

    }

    function index() {
        $this->load->view('v_login');
    }

    function aksi_login() {
```

```

        $username = $this->input-
>post('username');

        $password = $this->input-
>post('password');

        $where = array(
            'username' => $username,
            'password' => $password
        );

        $cek = $this->m_login-
>cek_login("user", $where)->num_rows();

        if($cek > 0){

            $data_session = array(
                'nama' => $username,
                'status' => "login"
            );

            $this->session-
>set_userdata($data_session);

            redirect(base_url("index.php/admin"));
        };

    }else{
        echo "Username dan password salah !";
    }
}

```

```
function logout() {  
    $this->session->sess_destroy();  
    redirect(base_url('login'));  
}  
}
```

### 5.2.6 Controller File *logindosen.php*

```
<?php  
  
class LoginDosen extends CI_Controller{  
  
    function __construct() {  
        parent::__construct();  
        $this->load->model('d_login');  
  
    }  
  
    function index() {  
        $this->load->view('d_login');  
    }  
  
    function aksi_login2() {
```

```

        $nik = $this->input->post('nik');

        $password = $this->input-
>post('password');

        $where = array(
            'nik' => $nik,
            'password' => $password
        );

        $cek = $this->d_login-
>cek_login("dosens", $where)->num_rows();

        if($cek > 0) {

            $data_session = array(
                'nama' => $nik,
                'status' => "login"
            );

            $this->session-
>set_userdata($data_session);

            redirect(base_url("index.php/dosen"));
        };

    } else{
        echo "Username dan password salah !";
    }
}

```

```
function logout() {  
    $this->session->sess_destroy();  
    redirect(base_url('logindsn'));  
}  
}
```

### 5.2.7 Controller File *loginmahasiswa.php*

```
<?php  
  
class LoginDosen extends CI_Controller{  
  
    function __construct(){  
        parent::__construct();  
        $this->load->model('d_login');  
    }  
  
    function index(){  
        $this->load->view('d_login');  
    }  
  
    function aksi_login2(){  
        $nik = $this->input->post('nik');
```

```

        $password = $this->input-
>post('password');

        $where = array(
            'nik' => $nik,
            'password' => $password
        );

        $cek = $this->d_login-
>cek_login("dosens", $where)->num_rows();

        if($cek > 0) {

            $data_session = array(
                'nama' => $nik,
                'status' => "login"
            );

            $this->session-
>set_userdata($data_session);

            redirect(base_url("index.php/dosen"))
        };

    } else{
        echo "Username dan password salah !"
    }
}

```

```

        function logout() {
            $this->session->sess_destroy();
            redirect(base_url('logindsn'));
        }
    }
}

```

### **5.2.8 Controller File *Welcome.php***

```

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

class Welcome extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *      http://example.com/index.php/welcome
     *      - or -
     *      http://example.com/index.php/welcome
     *      /index
     *
     * Since this controller is set as the default controller in
     * config/routes.php, it's displayed at http://example.com/
     *

```

```

        * So any other public methods not prefixed
with an underscore will

        * map to /index.php/welcome/<method_name>

        * @see https://codeigniter.com/user_guide/g
eneral/urls.html

        */

public function index()
{
    $this->load->view('welcome_message');

}

}

```

### **5.3 Membuat Model**

#### **5.3.1 File *D\_login.php***

```

<?php

class D_login extends CI_Model{
    function cek_login($table,$where) {
        return $this->db-
    >get_where($table,$where);
    }
}

```

#### **5.3.2 File *Dosen\_model.php***

```

<?php defined('BASEPATH') OR exit('No direct scr
ipt access allowed');

```

```
class Dosen_model extends CI_Model
{
    private $_table = "dosesns";

    public $id_dosen;
    public $nik;
    public $nama;
    public $password;

    public function rules()
    {
        return [
            ['field' => 'nik',
             'label' => 'NIK',
             'rules' => 'required'],

            ['field' => 'nama',
             'label' => 'Nama',
             'rules' => 'required'],

            ['field' => 'password',
             'label' => 'Password',
             'rules' => 'required']
        ];
    }
}
```

```

public function getAll()
{
    return $this->db->get($this->_table)-
>result();
}

public function getById($id)
{
    return $this->db->get_where($this-
>_table, ["id_dosen" => $id])->row();
}

public function save()
{
    $post = $this->input->post();
    $this->id_dosen = uniqid();
    $this->nik = $post["nik"];
    $this->nama = $post["nama"];
    $this->password = $post["password"];
    $this->db->insert($this->_table, $this);
}

public function update()
{
    $post = $this->input->post();

```

```

        $this->id_dosen = $post["id"];
        $this->nik = $post["nik"];
        $this->nama = $post["nama"];
        $this->password = $post["password"];
        $this->db->update($this-
>_table, $this, array('id_dosen' => $post['id']))
);

}

public function delete($id)
{
    return $this->db->delete($this-
>_table, array("id_dosen" => $id));
}

```

### 5.3.3 File *Ganjil1819\_model.php*

```

<?php defined('BASEPATH') OR exit('No direct scr
ipt access allowed');

class Ganjil1819_model extends CI_Model
{
    private $_table = "ganjil1819";

    public $id_jadwal;
    public $pengajar;

```

```
public $matkul;  
public $ruang;  
public $hari;  
public $jam;  
public $kelas;  
public $stanggal;  
public $status;  
public $alasan;  
  
public function rules()  
{  
    return [  
        ['field' => 'pengajar',  
         'label' => 'pengajar',  
         'rules' => 'required'],  
  
        ['field' => 'matkul',  
         'label' => 'matkul',  
         'rules' => 'required'],  
  
        ['field' => 'ruang',  
         'label' => 'ruang',  
         'rules' => 'required'],  
  
        ['field' => 'hari',
```

```
        'label' => 'hari',
        'rules' => 'required'],

        ['field' => 'jam',
         'label' => 'jam',
         'rules' => 'required'],

        ['field' => 'kelas',
         'label' => 'kelas',
         'rules' => 'required'],

        ['field' => 'tanggal',
         'label' => 'tanggal',
         'rules' => 'required'],

        ['field' => 'status',
         'label' => 'status',
         'rules' => 'required'],

        ['field' => 'alasan',
         'label' => 'alasan',
         'rules' => 'required']
    ];
}
```

```

public function getAll()
{
    return $this->db->get($this->_table)-
>result();
}

public function getById($id)
{
    return $this->db->get_where($this-
>_table, ["id_jadwal" => $id])->row();
}

public function save()
{
    $post = $this->input->post();
    $this->id_jadwal = uniqid();
    $this->pengajar = $post["pengajar"];
    $this->matkul = $post["matkul"];
    $this->ruang = $post["ruang"];
    $this->hari = $post["hari"];
    $this->jam = $post["jam"];
    $this->kelas = $post["kelas"];
    $this->tanggal = $post["tanggal"];
    $this->status = $post["status"];
    $this->alasan = $post["alasan"];
    $this->db->insert($this->_table, $this);
}

```

```

}

public function update()
{
    $post = $this->input->post();
    $this->id_jadwal = $post["id"];
    $this->pengajar = $post["pengajar"];
    $this->matkul = $post["matkul"];
    $this->ruang = $post["ruang"];
    $this->hari = $post["hari"];
    $this->jam = $post["jam"];
    $this->kelas = $post["kelas"];
    $this->tanggal = $post["tanggal"];
    $this->status = $post["status"];
    $this->alasan = $post["alasan"];
    $this->db->update($this-
>_table, $this, array('id_jadwal' => $post['id']
));
}

public function delete($id)
{
    return $this->db->delete($this-
>_table, array("id_jadwal" => $id));
}

```

### 5.3.4 File *Ganjil1920\_mode.php*

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

class Ganjil1920_model extends CI_Model
{
    private $_table = "ganjil1920";

    public $id_jadwal;
    public $pengajar;
    public $matkul;
    public $ruang;
    public $hari;
    public $jam;
    public $kelas;
    public $stanggal;
    public $status;
    public $alasan;

    public function rules()
    {
        return [
            ['field' => 'pengajar',
             'label' => 'pengajar',
             'rules' => 'required'],

```

```
[ 'field' => 'matkul',
  'label' => 'matkul',
  'rules' => 'required'] ,  
  
[ 'field' => 'ruang',
  'label' => 'ruang',
  'rules' => 'required'] ,  
  
[ 'field' => 'hari',
  'label' => 'hari',
  'rules' => 'required'] ,  
  
[ 'field' => 'jam',
  'label' => 'jam',
  'rules' => 'required'] ,  
  
[ 'field' => 'kelas',
  'label' => 'kelas',
  'rules' => 'required'] ,  
  
[ 'field' => 'tanggal',
  'label' => 'tanggal',
  'rules' => 'required'] ,
```

```

        ['field' => 'status',
         'label' => 'status',
         'rules' => 'required'] ,

        ['field' => 'alasan',
         'label' => 'alasan',
         'rules' => 'required']]

    ];

}

public function getAll()
{
    return $this->db->get($this->_table)-
>result();
}

public function getById($id)
{
    return $this->db->get_where($this-
>_table, ["id_jadwal" => $id])->row();
}

public function save()
{
    $post = $this->input->post();
    $this->id_jadwal = uniqid();
}

```

```

        $this->pengajar = $post["pengajar"];
        $this->matkul = $post["matkul"];
        $this->ruang = $post["ruang"];
        $this->hari = $post["hari"];
        $this->jam = $post["jam"];
        $this->kelas = $post["kelas"];
        $this->tanggal = $post["tanggal"];
        $this->status = $post["status"];
        $this->alasan = $post["alasan"];
        $this->db->insert($this->_table, $this);
    }


```

```

public function update()
{
    $post = $this->input->post();
    $this->id_jadwal = $post["id"];
    $this->pengajar = $post["pengajar"];
    $this->matkul = $post["matkul"];
    $this->ruang = $post["ruang"];
    $this->hari = $post["hari"];
    $this->jam = $post["jam"];
    $this->kelas = $post["kelas"];
    $this->tanggal = $post["tanggal"];
    $this->status = $post["status"];
    $this->alasan = $post["alasan"];
}


```

```

        $this->db->update($this-
>_table, $this, array('id_jadwal' => $post['id']
));
}

public function delete($id)
{
    return $this->db->delete($this-
>_table, array("id_jadwal" => $id));
}

```

### 5.3.5 File *Genap1819\_model.php*

```

<?php defined('BASEPATH') OR exit('No direct scr
ipt access allowed');

class Genap1819_model extends CI_Model
{
    private $_table = "genap1819";

    public $id_jadwal;
    public $pengajar;
    public $matkul;
    public $ruang;
    public $hari;
    public $jam;
}

```

```
public $kelas;  
public $stanggal;  
public $status;  
public $alasan;  
  
public function rules()  
{  
    return [  
        ['field' => 'pengajar',  
         'label' => 'pengajar',  
         'rules' => 'required'],  
  
        ['field' => 'matkul',  
         'label' => 'matkul',  
         'rules' => 'required'],  
  
        ['field' => 'ruang',  
         'label' => 'ruang',  
         'rules' => 'required'],  
  
        ['field' => 'hari',  
         'label' => 'hari',  
         'rules' => 'required'],  
  
        ['field' => 'jam',  
         'label' => 'jam',  
         'rules' => 'required'],  
    ];  
}
```

```

        'label' => 'jam',
        'rules' => 'required'],
    ],
    [
        'field' => 'kelas',
        'label' => 'kelas',
        'rules' => 'required'],
    ],
    [
        'field' => 'tanggal',
        'label' => 'tanggal',
        'rules' => 'required'],
    ],
    [
        'field' => 'status',
        'label' => 'status',
        'rules' => 'required'],
    ],
    [
        'field' => 'alasan',
        'label' => 'alasan',
        'rules' => 'required']
];
}

public function getAll()
{
    return $this->db->get($this->_table)-
>result();
}

```

```

public function getById($id)
{
    return $this->db->get_where($this-
>_table, ["id_jadwal" => $id])->row();
}

public function save()
{
    $post = $this->input->post();
    $this->id_jadwal = uniqid();
    $this->pengajar = $post["pengajar"];
    $this->matkul = $post["matkul"];
    $this->ruang = $post["ruang"];
    $this->hari = $post["hari"];
    $this->jam = $post["jam"];
    $this->kelas = $post["kelas"];
    $this->tanggal = $post["tanggal"];
    $this->status = $post["status"];
    $this->alasan = $post["alasan"];
    $this->db->insert($this->_table, $this);
}

public function update()
{
    $post = $this->input->post();

```

```

        $this->id_jadwal = $post["id"];
        $this->pengajar = $post["pengajar"];
        $this->matkul = $post["matkul"];
        $this->ruang = $post["ruang"];
        $this->hari = $post["hari"];
        $this->jam = $post["jam"];
        $this->kelas = $post["kelas"];
        $this->tanggal = $post["tanggal"];
        $this->status = $post["status"];
        $this->alasan = $post["alasan"];
        $this->db->update($this-
>_table, $this, array('id_jadwal' => $post['id'])
);
    }

public function delete($id)
{
    return $this->db->delete($this-
>_table, array("id_jadwal" => $id));
}

```

### 5.3.6 File *Hadir\_model.php*

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

```
class Hadir_model extends CI_Model
{
    private $_table = "jadwalaja";

    public $id_jadwal;
    public $pengajar;
    public $matkul;
    public $ruang;
    public $hari;
    public $jam;
    public $kelas;
    public $stanggal;
    public $status;
    public $alasan;

    public function rules()
    {
        return [
            ['field' => 'pengajar',
             'label' => 'pengajar',
             'rules' => 'required'],
            ['field' => 'matkul',
             'label' => 'matkul',
             'rules' => 'required'],
        ];
    }
}
```

```
[ 'field' => 'ruang',
  'label' => 'ruang',
  'rules' => 'required'] ,  
  
[ 'field' => 'hari',
  'label' => 'hari',
  'rules' => 'required'] ,  
  
[ 'field' => 'jam',
  'label' => 'jam',
  'rules' => 'required'] ,  
  
[ 'field' => 'kelas',
  'label' => 'kelas',
  'rules' => 'required'] ,  
  
[ 'field' => 'tanggal',
  'label' => 'tanggal',
  'rules' => 'required'] ,  
  
[ 'field' => 'status',
  'label' => 'status',
  'rules' => 'required'] ,
```

```

        ['field' => 'alasan',
         'label' => 'alasan',
         'rules' => 'required']
     ] ;
}

public function getAll()
{
    return $this->db->get($this->_table)-
>result();
}

public function getById($id)
{
    return $this->db->get_where($this-
>_table, ["id_jadwal" => $id])->row();
}

public function save()
{
    $post = $this->input->post();
    $this->id_jadwal = uniqid();
    $this->pengajar = $post["pengajar"];
    $this->matkul = $post["matkul"];
    $this->ruang = $post["ruang"];
    $this->hari = $post["hari"];
}

```

```

        $this->jam = $post["jam"];
        $this->kelas = $post["kelas"];
        $this->tanggal = $post["tanggal"];
        $this->status = $post["status"];
        $this->alasan = $post["alasan"];
        $this->db->insert($this->_table, $this);
    }

public function update()
{
    $post = $this->input->post();
    $this->id_jadwal = $post["id"];
    $this->pengajar = $post["pengajar"];
    $this->matkul = $post["matkul"];
    $this->ruang = $post["ruang"];
    $this->hari = $post["hari"];
    $this->jam = $post["jam"];
    $this->kelas = $post["kelas"];
    $this->tanggal = $post["tanggal"];
    $this->status = $post["status"];
    $this->alasan = $post["alasan"];
    $this->db->update($this-
>_table, $this, array('id_jadwal' => $post['id'])
);
}

```

```
    public function delete($id)
    {
        return $this->db->delete($this-
>_table, array("id_jadwal" => $id));
    }
}
```

### 5.3.7 File Jadwal\_model.php

```
<?php defined('BASEPATH') OR exit('No direct scr
ipt access allowed');

class Jadwal_model extends CI_Model
{
    private $_table = "jadwal";

    public $id_jadwal;
    public $id_pengajar;
    public $id_matkul;
    public $id_ruang;
    public $id_hari;
    public $id_jam;
    public $status;
    public $alasan;

    public function rules()
    {
```

```
return [  
    {'field' => 'id_pengajar',  
     'label' => 'pengajar',  
     'rules' => 'required'},  
  
    {'field' => 'id_matkul',  
     'label' => 'matkul',  
     'rules' => 'required'},  
  
    {'field' => 'id_ruang',  
     'label' => 'ruang',  
     'rules' => 'required'},  
  
    {'field' => 'id_hari',  
     'label' => 'hari',  
     'rules' => 'required'},  
  
    {'field' => 'id_jam',  
     'label' => 'id_jam',  
     'rules' => 'required'},  
  
    {'field' => 'status',  
     'label' => 'status',  
     'rules' => 'required'}]
```

```

        [ 'field' => 'alasan',
          'label' => 'alasan',
          'rules' => 'required' ]
    ] ;
}

public function getAll()
{
    return $this->db->get($this->_table)-
>result();
}

public function getById($id)
{
    return $this->db->get_where($this-
>_table, ["id_jadwal" => $id])->row();
}

function ambil()
{
    $this->db-
>join('pengajar p', 'j.id_pengajar = p.id_pengaj
ar');
    $this->db-
>join('matkul m', 'j.id_matkul = m.id_matkul');
    $this->db-
>join('ruang r', 'j.id_ruang = r.id_ruang');
}

```

```

        $this->db-
>join('hari h', 'j.id_hari = h.id_hari');

        $this->db-
>join('jam j', 'j.id_pengajar = j.id_jam');

        $this->db->order_by('id_jadwal');

        return $this->db->get('jadwal')-
>result();

}

function get() {
    $query = $this->db-
>query('SELECT * FROM tb_pengajar');

    return $query->result();

}

public function save()
{
    $post = $this->input->post();

    $this->id_jadwal = uniqid();

    $this-
>id_pengajar = $post["id_pengajar"];

    $this->id_matkul = $post["id_matkul"];
    $this->id_ruang = $post["id_ruang"];
    $this->id_hari = $post["id_hari"];
    $this->id_jam = $post["id_jam"];
    $this->db->insert($this->_table, $this);

}

```

```

public function update()
{
    $post = $this->input->post();
    $this->id_jadwal = $post["id"];
    $this-
>id_pengajar = $post["id_pengajar"];
    $this->id_matkul = $post["id_matkul"];
    $this->id_ruang = $post["id_ruang"];
    $this->id_hari = $post["id_hari"];
    $this->id_jam = $post["id_jam"];
    $this->status = $post["status"];
    $this->alasan = $post["alasan"];
    $this->db->update($this-
>_table, $this, array('id_jadwal' => $post['id']
));
}

public function delete($id)
{
    return $this->db->delete($this-
>_table, array("id_jadwal" => $id));
}

```

### 5.3.8 File Jadwalaja.php

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

class Jadwalaja_model extends CI_Model
{
    private $_table = "jadwalaja";

    public $id_jadwal;
    public $pengajar;
    public $matkul;
    public $ruang;
    public $hari;
    public $jam;
    public $kelas;
    public $stanggal;
    public $status;
    public $alasan;

    public function rules()
    {
        return [
            ['field' => 'pengajar',
            'label' => 'pengajar',
            'rules' => 'required'],
            ['field' => 'matkul',

```

```
'label' => 'matkul',
'rules' => 'required'] ,  
  
[ 'field' => 'ruang',
'label' => 'ruang',
'rules' => 'required'] ,  
  
[ 'field' => 'hari',
'label' => 'hari',
'rules' => 'required'] ,  
  
[ 'field' => 'jam',
'label' => 'jam',
'rules' => 'required'] ,  
  
[ 'field' => 'kelas',
'label' => 'kelas',
'rules' => 'required'] ,  
  
[ 'field' => 'tanggal',
'label' => 'tanggal',
'rules' => 'required'] ,  
  
[ 'field' => 'status',
'label' => 'status',
```

```
        'rules' => 'required'] ,  
  
        ['field' => 'alasan',  
         'label' => 'alasan',  
         'rules' => 'required']  
    ] ;  
}  
  
public function getAll()  
{  
    return $this->db->get($this->_table)->result();  
}  
  
public function getByNik($id)  
{  
    return $this->db->get_where($this->_table, ["nik"=>$id])->result();  
}  
  
public function getById($id)  
{  
    return $this->db->get_where($this->_table, ["id_jadwal" => $id])->row();  
}  
  
public function save()  
174 | V Tahapan Membuat Aplikasi Kehadiran Dosen
```

```

    {

        $post = $this->input->post();

        $this->id_jadwal = uniqid();

        $this->pengajar = $post["pengajar"];

        $this->matkul = $post["matkul"];

        $this->ruang = $post["ruang"];

        $this->hari = $post["hari"];

        $this->jam = $post["jam"];

        $this->kelas = $post["kelas"];

        $this->tanggal = $post["tanggal"];

        $this->status = $post["status"];

        $this->alasan = $post["alasan"];

        $this->db->insert($this->_table, $this);

    }

```

```

public function update()
{
    $post = $this->input->post();

    $this->id_jadwal = $post["id"];

    $this->pengajar = $post["pengajar"];

    $this->matkul = $post["matkul"];

    $this->ruang = $post["ruang"];

    $this->hari = $post["hari"];

    $this->jam = $post["jam"];

    $this->kelas = $post["kelas"];

```

```

        $this->tanggal = $post["tanggal"];
        $this->status = $post["status"];
        $this->alasan = $post["alasan"];
        $this->db->update($this-
>_table, $this, array('id_jadwal' => $post['id'])
));
}

public function delete($id)
{
    return $this->db->delete($this-
>_table, array("id_jadwal" => $id));
}

```

### 5.3.9 File *Kehadiran\_model.php*

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

```

class Kehadiran_model extends CI_Model
{
    private $_table = "jadwalaja";

    public $id_jadwal;
    public $pengajar;
    public $matkul;

```

```
public $ruang;
public $hari;
public $jam;
public $kelas;
public $tanggal;
public $status;
public $alasan;

public function rules()
{
    return [
        ['field' => 'pengajar',
         'label' => 'pengajar',
         'rules' => 'required'],
        ['field' => 'matkul',
         'label' => 'matkul',
         'rules' => 'required'],
        ['field' => 'ruang',
         'label' => 'ruang',
         'rules' => 'required'],
        ['field' => 'hari',
         'label' => 'hari',
```

```
'rules' => 'required'],  
  
        ['field' => 'jam',  
         'label' => 'jam',  
         'rules' => 'required'],  
  
        ['field' => 'kelas',  
         'label' => 'kelas',  
         'rules' => 'required'],  
  
        ['field' => 'tanggal',  
         'label' => 'tanggal',  
         'rules' => 'required'],  
  
        ['field' => 'status',  
         'label' => 'status',  
         'rules' => 'required'],  
  
        ['field' => 'alasan',  
         'label' => 'alasan',  
         'rules' => 'required']  
    ];  
}  
  
public function getAll()
```

```

{

    return $this->db->get($this->_table)->result();
}

public function getById($id)
{
    return $this->db->get_where($this->_table, ["id_jadwal" => $id])->row();
}

public function save()
{
    $post = $this->input->post();
    $this->id_jadwal = uniqid();
    $this->pengajar = $post["pengajar"];
    $this->matkul = $post["matkul"];
    $this->ruang = $post["ruang"];
    $this->hari = $post["hari"];
    $this->jam = $post["jam"];
    $this->kelas = $post["kelas"];
    $this->tanggal = $post["tanggal"];
    $this->status = $post["status"];
    $this->alasan = $post["alasan"];
    $this->db->insert($this->_table, $this);
}

```

```
public function update()
{
    $post = $this->input->post();
    $this->id_jadwal = $post["id"];
    $this->pengajar = $post["pengajar"];
    $this->matkul = $post["matkul"];
    $this->ruang = $post["ruang"];
    $this->hari = $post["hari"];
    $this->jam = $post["jam"];
    $this->kelas = $post["kelas"];
    $this->tanggal = $post["tanggal"];
    $this->status = $post["status"];
    $this->alasan = $post["alasan"];
    $this->db->update($this-
    >_table, $this, array('id_jadwal' => $post['id']));
}
```

```
public function delete($id)
{
    return $this->db->delete($this-
    >_table, array("id_jadwal" => $id));
}
```

### 5.3.10 File *M\_login.php* File

```
<?php

class M_login extends CI_Model{

    function cek_login($table,$where) {
        return $this->db-
>get_where($table,$where);
    }
}
```

### 5.3.11 File Mahasiswa\_model.php

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

class Mahasiswa_model extends CI_Model
{
    private $_table = "mahasiswas";

    public $id_mahasiswa;
    public $npm;
    public $nama;
    public $password;

    public function rules()
    {
        return [
            ['field' => 'npm',
```

```

        'label' => 'NPM',
        'rules' => 'required'],
    ],
    [
        'field' => 'nama',
        'label' => 'Nama',
        'rules' => 'required'],
    ],
    [
        'field' => 'password',
        'label' => 'Password',
        'rules' => 'required']
];
}

public function getAll()
{
    return $this->db->get($this->_table)-
>result();
}

public function getById($id)
{
    return $this->db->get_where($this-
>_table, ["id_mahasiswa" => $id])->row();
}

public function save()

```

```

    {

        $post = $this->input->post();

        $this->id_mahasiswa = uniqid();

        $this->npm = $post["npm"];

        $this->nama = $post["nama"];

        $this->password = $post["password"];

        $this->db->insert($this->_table, $this);

    }

public function update()

{

    $post = $this->input->post();

    $this->id_mahasiswa = $post["id"];

    $this->npm = $post["npm"];

    $this->nama = $post["nama"];

    $this->password = $post["password"];

    $this->db->update($this-
>_table, $this, array('id_mahasiswa' => $post['i
d']));

}

public function delete($id)

{
    return $this->db->delete($this-
>_table, array("id_mahasiswa" => $id));
}
}

```

### **5.3.12 File *W\_Login***

```
<?php  
  
class W_login extends CI_Model{  
    function cek_login($table,$where) {  
        return $this->db->get_where($table,$where);  
    }  
}
```

## **5.4 Membuat *View***

Pada *view*, kami membuat beberapa *view* yaitu admin, dosen, dan mahasiswa, serta beberapa *view* tambahan lainnya. Berikut ini adalah beberapa *view* yang telah kami buat.

### **5.4.1 *View Admin***

*View admin* mempunyai banyak *view*, karena admin bertugas mengelola semua data yang ada di aplikasi. Berikut ini adalah beberapa *view* yang ada di *view admin*.

1. *User* dosen
2. *User* mahasiswa
3. Jadwal Semester

Dibawah ini adalah contoh codingan dari *view user* dosen.

#### **5.4.1.1 File *edit\_form.php***

```
<!DOCTYPE html>  
<html lang="en">
```

```
<head>
    <?php $this->load-
>view("admin/_partials/head.php") ?>
</head>

<body id="page-top">

    <?php $this->load-
>view("admin/_partials/navbar.php") ?>
    <div id="wrapper">

        <?php $this->load-
>view("admin/_partials/sidebar.php") ?>

        <div id="content-wrapper">

            <div class="container-fluid">

                <?php $this->load-
>view("admin/_partials/breadcrumb.php") ?>

                <?php if ($this->session-
>flashdata('success')): ?>
                    <div class="alert alert-
success" role="alert">
                        <?php echo $this->session-
>flashdata('success'); ?>

```

```

        </div>
        <?php endif; ?>

<div class="card mb-3">
    <div class="card-header">
        <a href="<?php echo site
_url('admin/dosens/') ?>"><i class="fas fa-
arrow-left"></i>
        Back</a>
    </div>
    <div class="card-body">
        <form action="<? echo si
te_url('admin/dosens/edit') ?>" method="post" en
ctype="multipart/form-data" >
            <input type="hidden"
name="id" value="<?php echo $dosen-
>id_dosen?>" />
        <div class="form-
group">
            <label for="name
">NIK*</label>
            <input class="fo
rm-control <?php echo form_error('nik') ? 'is-
invalid':'' ?>"
```

```

                type="text" name="nik" placeholder="NIK" value="php echo $dosen-&gt;nik ?&gt;" /&gt;

                &lt;div class="invalid-feedback"&gt;
                    &lt;?php echo form_error('nik') ?&gt;
                &lt;/div&gt;
            &lt;/div&gt;

            &lt;div class="form-group"&gt;
                &lt;label for="name"&gt;Nama*&lt;/label&gt;
                &lt;input class="form-control" type="text" name="nama" placeholder="NIK" value="<?php echo $dosen-&gt;nama ?&gt;" /&gt;

                &lt;div class="invalid-feedback"&gt;
                    &lt;?php echo form_error('nama') ?&gt;
                &lt;/div&gt;
            &lt;/div&gt;

            &lt;div class="form-group"&gt;
</pre

```

```

        <label for="name">
    " > Password* </label>

        <input class="form-control" type="text" name="password" placeholder="NPM" value="<?php echo $dosen->password ?>" />

        <div class="invalid-feedback">
            <?php echo form_error('password') ?>
        </div>
    </div>

        <input class="btn btn-success" type="submit" name="btn" value="Save" />
    </form>

</div>

<div class="card-footer small text-muted">
    * required fields
</div>

</div>

```

```
<!-- /.container-fluid -->

<!-- Sticky Footer -->
<?php $this->load-
>view("admin/_partials/footer.php") ?>

</div>
<!-- /.content-wrapper -->

</div>
<!-- /#wrapper -->

<?php $this->load-
>view("admin/_partials/scrolltop.php") ?>

<?php $this->load-
>view("admin/_partials/js.php") ?>

</body>

</html>
```

#### **5.4.1.1 File *list.php***

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
    <?php $this->load-
>view("admin/_partials/head.php") ?>
</head>

<body id="page-top">

    <?php $this->load-
>view("admin/_partials/navbar.php") ?>
    <div id="wrapper">

        <?php $this->load-
>view("admin/_partials/sidebar.php") ?>

        <div id="content-wrapper">

            <div class="container-fluid">

                <?php $this->load-
>view("admin/_partials/breadcrumb.php") ?>

                <!-- DataTables -->
                <div class="card mb-3">
                    <div class="card-header">
                        <a href="<?php echo site
_url('admin/dosens/add') ?>"><i class="fas fa-
plus"></i> Add New</a>
                    </div>

```

```

<div class="card-body">

    <div class="table-
responsive">
        <table class="table
table-
hover" id="dataTable" width="100%" cellspacing="0">
            <thead>
                <tr>
                    <th>NIK</th>
                    <th>Nama</th>
                    <th>Pass
word</th>
                    <th>Acti
on</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach
($dosens as $dosen) : ?>
                <tr>
                    <td widt
h="150">
                        <?ph
p echo $dosen->nik ?>
                    </td>

```

```

        <td width="150">
            <?php
                echo $dosen->nama ?>
            </td>
        <td width="150">
            <?php
                echo $dosen->password ?>
            </td>
        <td width="250">
            <a href="<?php echo site_url('admin/dosens/edit/'.$dosen->id_dosen) ?>" class="btn btn-small"><i class="fas fa-edit"></i> Edit</a>
            <a onclick="deleteConfirm('<?php echo site_url('admin/dosens/delete/'.$dosen->id_dosen) ?>')" href="#" class="btn btn-small text-danger"><i class="fas fa-trash"></i> Hapus</a>
        </td>
    </tr>
<?php endforeach;
each; ?>

        </tbody>

```

```
        </table>

    </div>
</div>

</div>
<!-- /.container-fluid -->

<!-- Sticky Footer -->
<?php $this->load-
>view("admin/_partials/footer.php") ?>

</div>
<!-- /.content-wrapper -->

</div>
<!-- /#wrapper -->

<?php $this->load-
>view("admin/_partials/scrolltop.php") ?>
<?php $this->load-
>view("admin/_partials/modal.php") ?>

<?php $this->load-
>view("admin/_partials/js.php") ?>

<script>
```

```
function deleteConfirm(url) {  
    $('#btn-delete').attr('href', url);  
    $('#deleteModal').modal();  
}  
</script>  
</body>  
  
</html>
```

#### 5.4.1.1 File *new\_form.php*

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
    <?php $this->load-  
>view("admin/_partials/head.php") ?>  
</head>  
  
<body id="page-top">  
  
    <?php $this->load-  
>view("admin/_partials/navbar.php") ?>  
    <div id="wrapper">
```

```

        <?php $this->load-
>view("admin/_partials/sidebar.php") ?>

<div id="content-wrapper">

    <div class="container-fluid">

        <?php $this->load-
>view("admin/_partials/breadcrumb.php") ?>

        <?php if ($this->session-
>flashdata('success')): ?>
            <div class="alert alert-
success" role="alert">
                <?php echo $this->session-
>flashdata('success'); ?>
            </div>
        <?php endif; ?>

        <div class="card mb-3">
            <div class="card-header">
                <a href="<?php echo site-
_url('admin/dosens/') ?>"><i class="fas fa-
arrow-left"></i> Back</a>
            </div>
            <div class="card-body">

```

```

                <form action="<?php echo
site_url('admin/dosens/add') ?>" method="post"
enctype="multipart/form-data" >

                <div class="form-
group">

                    <label for="name
">NIK*</label>

                    <input class="fo
rm-control <?php echo form_error('nik') ? 'is-
invalid': '' ?>"

                        type="text" nam
e="nik" placeholder="NIK" />

                    <div class="inva
lid-feedback">

                        <?php echo f
orm_error('nik') ?>

                    </div>

                </div>

                <div class="form-
group">

                    <label for="name
">Nama*</label>

                    <input class="fo
rm-control <?php echo form_error('nama') ? 'is-
invalid': '' ?>"

                        type="text" nam
e="nama" placeholder="Nama" />

                    <div class="inva
lid-feedback">

```

```
<?php echo f
orm_error('nama') ?>
</div>

</div>

<div class="form-
group">
    <label for="pass
word">Password*</label>
    <input class="fo
rm-
control <?php echo form_error('password') ? 'is-
invalid':'' ?>" type="text" nam
e="password" placeholder="Password" />
    <div class="inva
lid-feedback">
        <?php echo f
orm_error('password') ?>
    </div>
</div>

<input class="btn bt
n-
success" type="submit" name="btn" value="Save" /
>
</form>

</div>
```

```
        <div class="card-
footer small text-muted">
            * required fields
        </div>

        </div>
        <!-- /.container-fluid -->

        <!-- Sticky Footer -->
        <?php $this->load-
>view("admin/_partials/footer.php") ?>

        </div>
        <!-- /.content-wrapper -->

        </div>
        <!-- /#wrapper -->

        <?php $this->load-
>view("admin/_partials/scrolltop.php") ?>

        <?php $this->load-
>view("admin/_partials/js.php") ?>

    </body>

</html>
```

# BAB VI

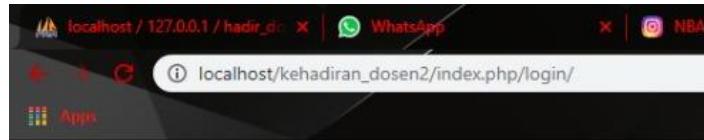
# IMPLEMENTASI

---

Berikut ini adalah hasil implementasi aplikasi Kehadiran Dosen seperti *user interface* admin, *user interface* dosen, dan *user interface* mahasiswa.

## 6.1 User Interface Admin

### 6.1.1 Tampilan *Login* Admin



**Selamat datang**

Username	<input type="text" value="admin"/>
Password	<input type="password" value="....."/>
<input type="button" value="Login"/>	

---

Gambar 6. 1 Tampilan *Login* Admin

Tampilan *login* admin merupakan halaman *login user* dengan memasukkan *username* dan *password*.

## 6.1.2 Tampilan Menu Jadwal

The screenshot shows a web application interface for managing student attendance. The top navigation bar includes links for 'Dashboard', 'Attendance', 'Photo', 'Report', 'Attendance Record', and 'Logout'. The main header says 'kehadiran dosen Admin - Ganjil 1819s'. The date '29-02-2020 10:10' and user 'Admin' are displayed. On the left, a sidebar has 'Overview' and 'Jadwal' selected. Under 'Jadwal', there are four options: 'Ganjil 2018/2019', 'Genap 2018/2019', 'Ganjil 2019/2020', and 'Genap 2019/2020'. Below this are 'Mahasiswa' and 'Dosen' sections. The main content area shows a table titled 'Add New' with columns: Pengajar, Matkul, Ruangan, Hari, Jam, Kelas, Tanggal Mengajar, Status, Keterangan, and Action. One row is listed: Mohamad Nurkamal Fauzan, S.T., M.T., Algoritma I, 310, Selasa, 07.00-10.20, D4 TI 1A, 2019-01-22, Tidak, Saya sakit, cari hari pengganti, with edit and delete buttons. At the bottom, it says 'Showing 1 to 1 of 1 entries' and 'Copyright © kehadiran dosen 2020'.

Gambar 6. 2 Menu Jadwal

Pada halaman ini kita bisa memilih tahun ajaran yg telah ditempuh, jika kita tekan salah satu akan muncul jadwal pada tahun ajaran yang kita tempuh tersebut dan kita bisa menambahkan pada tahun ajaran yang akan berlangsung, merubah dan menghapus.

### 6.1.3 Tampilan Menambahkan Jadwal pada Menu Jadwal

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'localhost/127.0.0.1/hadr...' and displays a form for adding a new schedule ('jadwalajas/add'). The left sidebar lists categories: Mahasiswa, Dosen, Pengajar\*, Matakul\*, Ruang\*, Hari\*, Jam\*, Kelas\*, Tanggal Mengajar\*, Status\*, and Keterangan\*. Each category has a corresponding input field. A 'Save' button is located at the bottom of the form.

Gambar 6. 3 Menambahkan Jadwal

Halaman ini berfungsi untuk menambahkan jadwal baru, dimana yang bisa menambahkan jadwal baru hanyalah admin. Admin dapat mengisi pengajar, mata kuliah, ruangannya, hari, jam, kelas, tanggal mengajar, status, dan keterangan.

### 6.1.4 Tampilan Merubah Jadwal pada Menu Jadwal

The screenshot shows a web application window with multiple tabs open. The active tab is titled 'localhost / 127.0.0.1 / hadr... / index.php/admin/jadwalajas/edit/Se4df6bf3bf4c'. The page displays a form for editing a schedule entry. The form fields are as follows:

- M. Harry K Saputra, S.T., M.T.I.
- Matkul\*
- Penprograman III
- Ruang\*
- 315
- Hari\*
- Kamis
- Jam\*
- 13.00-18.00
- Kelas\*
- D4 TI 2A
- Tanggal\*
- 27/02/2020
- Status\*
- Tidak
- Keterangan\*
- Saya Rapat; Cari Hari Pengganti

A green 'Save' button is located at the bottom left of the form.

Gambar 6. 4 Mengubah Jadwal

Di sini adalah proses *edit* yaitu jika terjadi kesalahan saat kita mengisikan jadwalnya bisa di ubah.

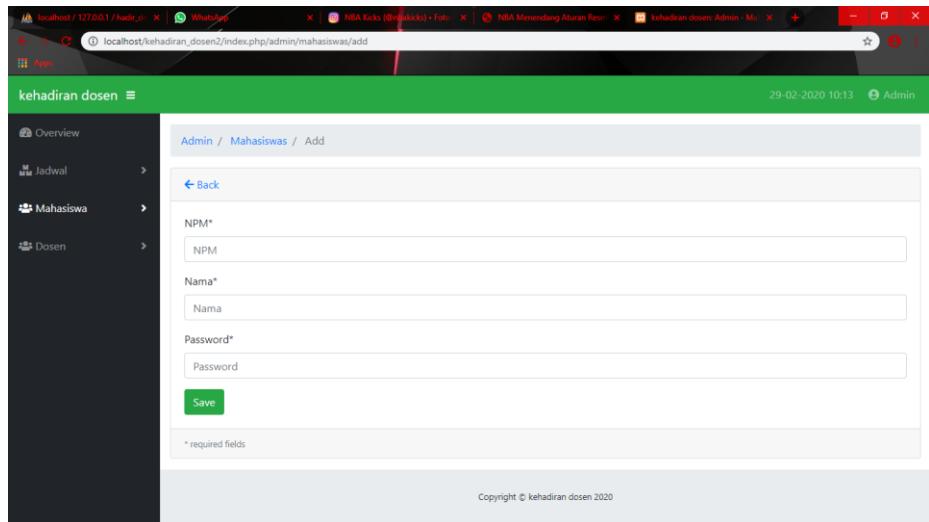
### 6.1.5 Tampilan Menu Mahasiswa

NPM	Nama	Password	Action
1184112	Ferdy Berliano Putra	sariash54	<a href="#">Edit</a> <a href="#">Hapus</a>
1184113	Ida Fatrini Putri	sariash54	<a href="#">Edit</a> <a href="#">Hapus</a>

Gambar 6. 5 Menu Mahasiswa

Pada tampilan menu mahasiswa, yaitu bisa menambahkan mahasiswa baru, merubahnya, dan menghapusnya.

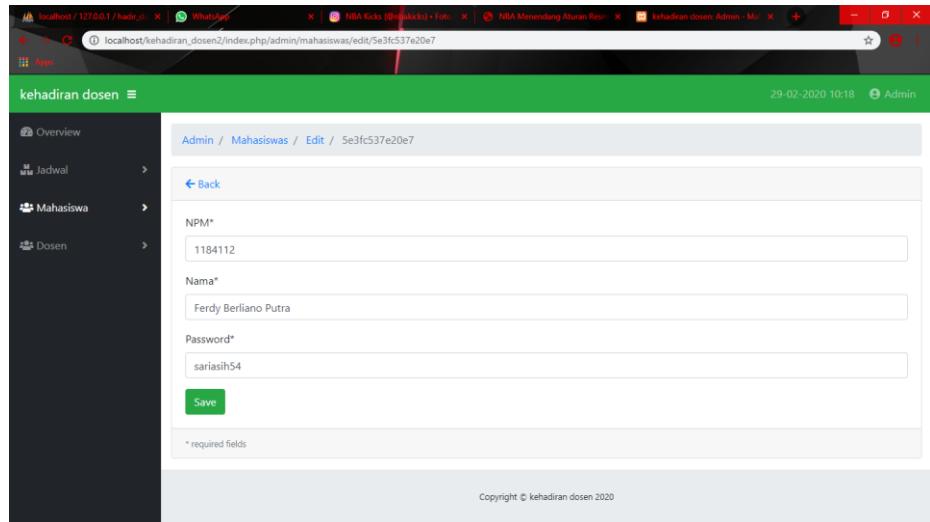
## 6.1.6 Tampilan Menambahkan Mahasiswa pada Menu Mahasiswa



Gambar 6. 6 Menambahkan Mahasiswa

Di sini kita bisa menambahkan *NPM*, *Nama* dan *Password*-nya yang sudah di tetapkan oleh institusi bersangkutan.

### 6.1.7 Tampilan Merubah Data Mahasiswa pada Menu Mahasiswa



Gambar 6. 7 Merubah Data Mahasiswa

Pada tampilan merubah data mahasiswa jika terjadi kesalahan pada memasukkannya.

### 6.1.8 Tampilan Menu Dosen

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'Admin / Dosen'. The page has a green header bar with the text 'kehadiran dosen' and the date '29-02-2020 10:18' along with a user icon. On the left, there's a sidebar with navigation links: 'Overview', 'Jadwal', 'Mahasiswa', and 'Dosen' (which is highlighted in blue). The main content area displays a table of faculty members. The table has columns for 'NIK', 'Nama', 'Password', and 'Action'. Each row contains a 'Edit' link and a 'Hapus' link. The table shows the following data:

NIK	Nama	Password	Action
10378065	Noviana Riza, S.Si, M.T.	sariasih54	<a href="#">Edit</a> <a href="#">Hapus</a>
10378069	Roni Habibi, S.Kom., M.T.	sariasih54	<a href="#">Edit</a> <a href="#">Hapus</a>
10579081	Woro Isti Rahayu, S.T., M.T.	sariasih54	<a href="#">Edit</a> <a href="#">Hapus</a>
11374163	M. Yusril Helmi Setyawan, S.Kom., M.Kom.	sariasih54	<a href="#">Edit</a> <a href="#">Hapus</a>
11380159	Mohamad Nurkamal Fauzan, S.T., M.T.	sariasih54	<a href="#">Edit</a> <a href="#">Hapus</a>
11508102	Ronal Andriyansyah, S.T., M.Kom.	sariasih54	<a href="#">Edit</a> <a href="#">Hapus</a>

Gambar 6. 8 Menu Dosen

Pada halaman ini adalah menu dosen.

### 6.1.9 Tampilan Menambahkan Dosen pada Menu Dosen

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'kehadiran\_dosen Admin - Dosen'. The page content is a form for adding a teacher ('Dosen'). The sidebar on the left lists 'Overview', 'Jadwal', 'Mahasiswa', and 'Dosen'. The main form area has a header 'Admin / Dosen / Add' and a 'Back' button. It contains three input fields: 'NIK\*' (with placeholder 'NIK'), 'Nama\*' (with placeholder 'Nama'), and 'Password\*' (with placeholder 'Password'). Below the fields is a green 'Save' button. At the bottom of the form, a note says '\* required fields'. The footer of the page includes the text 'Copyright © kehadiran dosen 2020'.

Di sini adalah untuk menambahkan data dosen yaitu *NIK*, *Nama*, dan *Password*.

### **6.1.10 Tampilan Merubah pada Menu Dosen**

localhost / 127.0.0.1 / hadir\_dosen / WhatsApp / NBA Kickz (@nba\_kicks) • Foto / NBA Memendang Aturan Resmi / kehadiran dosen Admin - Dokumen

localhost: kehadiran\_dosen2/index.php/admin/dosen/edit/5e428e8962a19

Apps

## kehadiran dosen

29-02-2020 10:23 Admin

Overview

Jadwal

Mahasiswa

Dosen

Admin / Dosen / Edit / 5e428e8962a19

[Back](#)

NIK\*

10378065

Nama\*

Noviana Riza, S.Si., M.T.

Password\*

sariashish54

[Save](#)

\* required fields

Di sini adalah tampilan merubah data-data dosen jika terjadi kesalahan pada saat memasukkan pada sebelumnya.

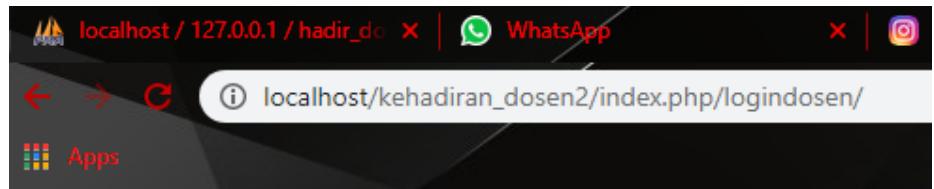
### **6.1.11 Tampilan Menghapus Data**

NIK	Nama	Password	Action
10378065	Noviana Riza, S.Si., M.T.	sariasih54	<input checked="" type="checkbox"/> Edit <span style="color: red;">Hapus</span>
10378069	Roni Habibi, S.Kom., M.T.	sariasih54	<input checked="" type="checkbox"/> Edit <span style="color: red;">Hapus</span>
10579081	Woro Isti Rahayu, S.T., M.T.	sariasih54	<input checked="" type="checkbox"/> Edit <span style="color: red;">Hapus</span>
11374163	M. Yusril Helmi Setyawan, S.Kom., M.Kom.	sariasih54	<input checked="" type="checkbox"/> Edit <span style="color: red;">Hapus</span>
11380159	Mohamad Nurkamal Fauzan, S.T., M.T.	sariasih54	<input checked="" type="checkbox"/> Edit <span style="color: red;">Hapus</span>

Di sini adalah jika kita ingin menghapus data jadwal atau mahasiswa atau dosen maka akan muncul sebuah tampilan *java script* jika kita yakin untuk menghapus data tersebut

## 6.2 User Interface Dosen

### 6.2.1 Tampilan Login Dosen



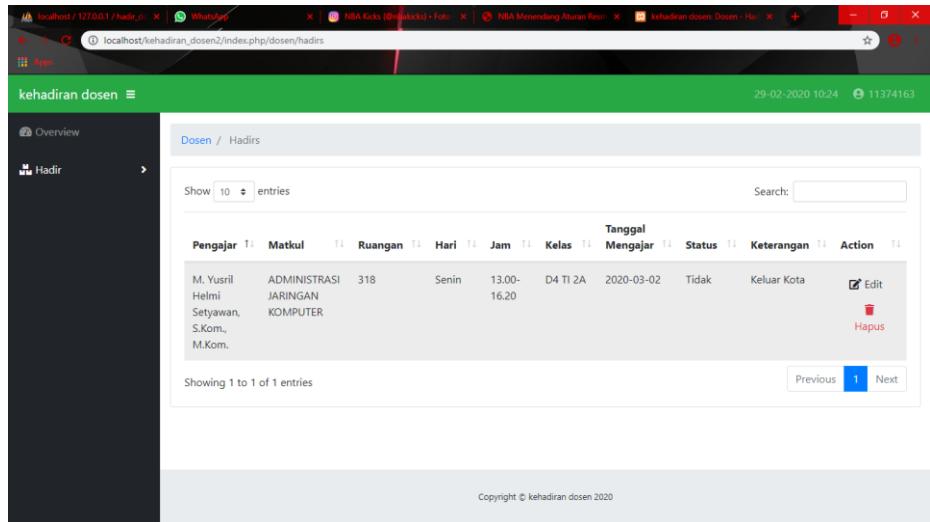
# Selamat datang

NIK

Password

Di sini adalah tampilan login untuk dosen sesuai *NIK* dosen dan *password*.

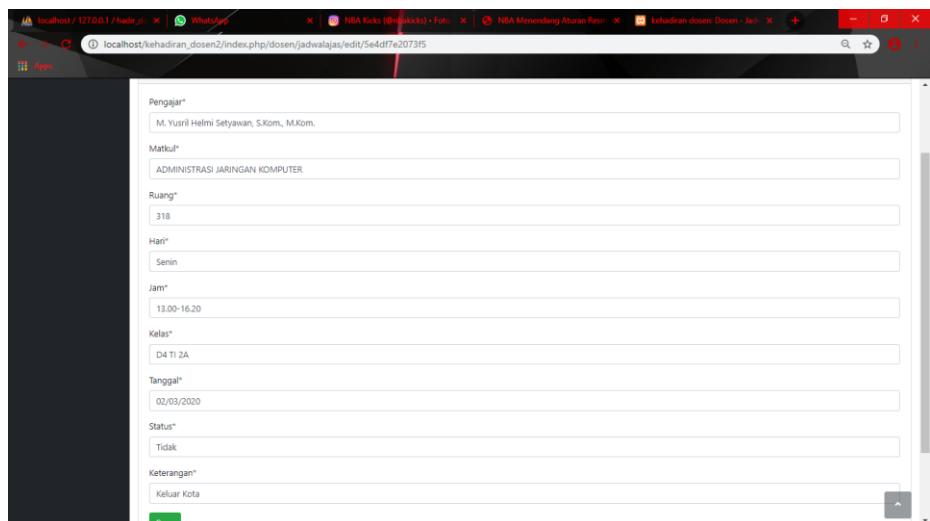
## 6.2.2 Tampilan Menu Dosen



The screenshot shows a web application interface titled "kehadiran dosen". On the left, there's a sidebar with "Overview" and "Hadir" options. The main content area is titled "Dosen / Hadirs" and displays a table with one row of data. The columns are labeled: Pengajar, Matkul, Ruangan, Hari, Jam, Kelas, Tanggal Mengajar, Status, Keterangan, and Action. The data row shows: M. Yusril Helmri Setyawan, S.Kom., M.Kom. as the teacher; ADMINISTRASI JARINGAN KOMPUTER as the subject; Ruangan 318; Hari Senin; Jam 13.00-16.20; Kelas D4 TI 2A; Tanggal Mengajar 2020-03-02; Status Tidak; Keterangan Keluar Kota; and Action buttons for Edit and Hapus. At the bottom, it says "Showing 1 to 1 of 1 entries" and has "Previous" and "Next" buttons. The footer says "Copyright © kehadiran dosen 2020".

Di sini adalah tampilan dosen untuk merubah jadwal dosen mengajar. Tetapi di sini hanya tampil bagian si dosen yang sedang *login* saja. Tidak bisa melihat semua jadwal.

## 6.2.3 Tampilan Menu Edit Dosen

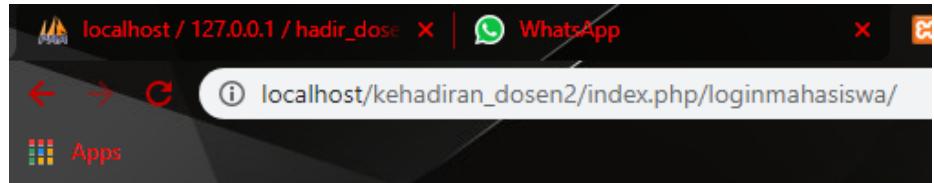


The screenshot shows a web application interface titled "kehadiran dosen: Dosen - Jadwal". The page contains a form with fields for "Pengajar" (M. Yusril Helmri Setyawan, S.Kom., M.Kom.), "Matkul" (ADMINISTRASI JARINGAN KOMPUTER), "Ruang" (318), "Hari" (Senin), "Jam" (13.00-16.20), "Kelas" (D4 TI 2A), "Tanggal" (02/03/2020), "Status" (Tidak), and "Keterangan" (Keluar Kota). A "Save" button is at the bottom right. The background shows other browser tabs like WhatsApp, NBA Kick, and NBA Membanding Alasan Resi.

Di sini dosen bisa mengubah jadwalnya mengajar, tetapi hanya bisa merubah tanggal, status, dan keterangan.

## 6.3 User Interface Mahasiswa

### 6.3.1 Tampilan Login Mahasiswa



# Selamat datang

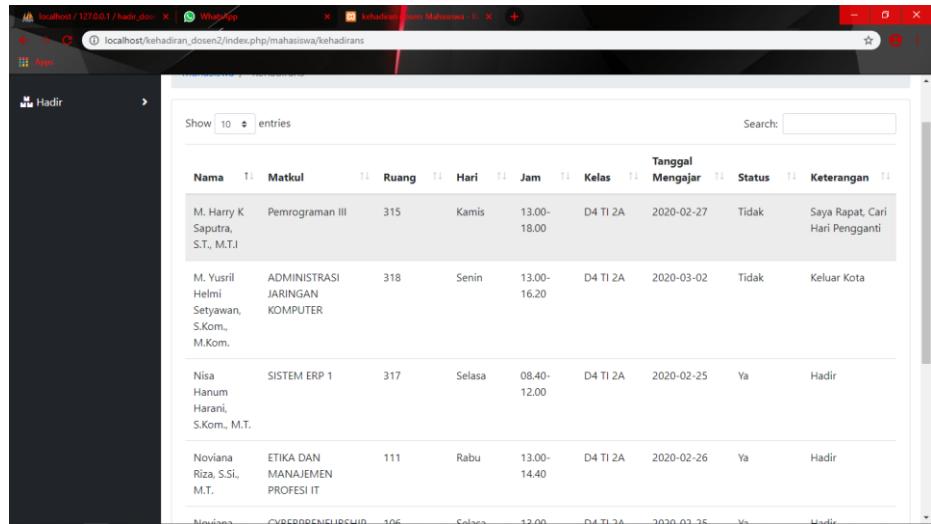
NPM

Password

Gambar 6. 9 Login Mahasiswa

Di sini adalah tampilan login mahasiswa sesuai *NPM* mahasiswa dan *password*.

### 6.3.2 Tampilan Menu Mahasiswa



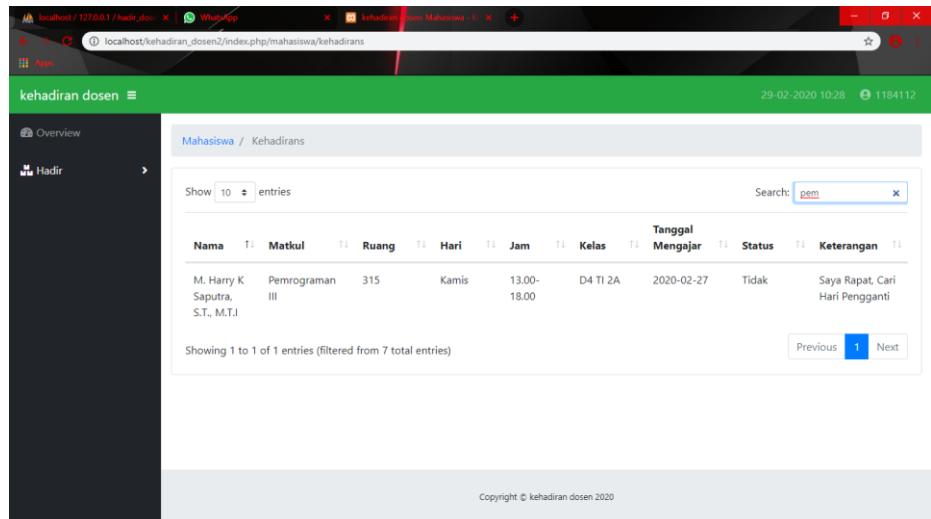
The screenshot shows a web application interface for managing student attendance. The main content area displays a table of student records with the following columns: Nama, Matkul, Ruang, Hari, Jam, Kelas, Tanggal Mengajar, Status, and Keterangan. The data includes:

Nama	Matkul	Ruang	Hari	Jam	Kelas	Tanggal Mengajar	Status	Keterangan
M. Harry K Saputra, S.T., M.T.I	Pemrograman III	315	Kamis	13.00-18.00	D4 TI 2A	2020-02-27	Tidak	Saya Rapat, Cari Hari Pengganti
M. Yusril Helmi Setyawan, S.Kom., M.Kom.	ADMINISTRASI JARINGAN KOMPUTER	318	Senin	13.00-16.20	D4 TI 2A	2020-03-02	Tidak	Keluar Kota
Nisa Hanum Harani, S.Kom., M.T.	SISTEM ERP 1	317	Selasa	08.40-12.00	D4 TI 2A	2020-02-25	Ya	Hadir
Noviana Riza, S.Si., M.T.	ETIKA DAN MANAJEMEN PROFESI IT	111	Rabu	13.00-14.40	D4 TI 2A	2020-02-26	Ya	Hadir
Mousaa	CYBERSECURITY	106	Seneca	12.00	D4 TI 2A	2020-02-25	Ya	Hadir

Gambar 6. 10 Menu Mahasiswa

Di sini adalah yang akan tampil data jadwal sesuai mata kuliah yang diikuti oleh mahasiswa.

### 6.3.3 Tampilan Penggunaan Fitur Searching



The screenshot shows a search results page for the term "pem". The search bar at the top contains "pem". The table of results is identical to the one in Figure 6.10, showing student attendance data for "Pemrograman III".

Nama	Matkul	Ruang	Hari	Jam	Kelas	Tanggal Mengajar	Status	Keterangan
M. Harry K Saputra, S.T., M.T.I	Pemrograman III	315	Kamis	13.00-18.00	D4 TI 2A	2020-02-27	Tidak	Saya Rapat, Cari Hari Pengganti

Gambar 6. 11 Fitur Searching

Di sini adalah kegunaan fitur searching yaitu kita bisa mencari apapun itu yg terkait dengan mata pelajaran kita, yaitu nama pengajar, mata kuliah, ruang, tanggal, hari jam, status, dan keterangan.

# DAFTAR PUSTAKA

---

1. Rais, Nendy Akbar Rozaq. 2016. E-Commerce Batik di Sentra Batik Desa Pilang Masaran. *Tesis*. Surakarta. Universitas Muhammadiyah.
2. Sujatmi, Sri. 2019. Aplikasi Program Kemitraan dan Bina Lingkungan (PKBL) Kantor Cabang PT. ASABRI (PERSERO) Palembang. *Tesis*. Palembang. Politeknik Negeri.
3. Suprianto, Andi dan Asri Amaliza Fathia Matsea. Rancang Bangun Aplikasi Pendaftaran Pasien Online dan Pemeriksaan Dokter di Klinik Pengobatan Berbasis Web. *Jurnal Rekayasa Informasi*, 7(1),2018
4. Utama, Weno Veringga. 2014. Aplikasi Pengolahan Data Alat Tulis Kantor pada Staff Operasi Komando Daerah Militer II Sriwijaya. *Tesis*. Palembang. Politeknik Negeri
5. Meilinda, Tri. 2017. Sistem Informasi Geografis Pemetaan Penyebaran Penyakit Endemik Berbasi Web di Kota Palembang. *Tesis*. Palembang. Politeknik Negeri.
6. Arlina, Adi. Pemanfaatan History pada Browser Komputer untuk Menentukan Pemilihan Website atau Blog pada Seindonesia. *Jurnal Compiler*, 3(2),2014.

7. Setiawan, Daniel and Setiawan, Rendra and Karunia, Rizka dan Rory, Rory dan Wicaksana, I Wayan Simri. Membandingkan Kinerja Web Browser. *Skripsi*. Depok. Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Gunadarma
8. Pratiti, Prima Hening dan Achmad Zaini. Aplikasi Bisnis Pengaruh Fitru dan Desain Terhadap Kepuasan Penggunaan Web Browser Google Chrome Versi 42.0.2311.90 dev-m. *Jurnal Aplikasi Bisnis*. 1(1), 2015.
9. Triandini, Evi. 2012. Step by Step Desain Proyek Menggunakan UML. Yogyakarta. CV. ANDI OFFSET
10. Rosyadi, Imam, dan Wanti Kholiani. Media Pembelajaran Tenses Bahasa Inggris pada English Club Berbasis Adobe Flash CS 3. *Jurnal Surya Informatika*. 2(1), 2016.
11. Sparx Systems. 2019.  
<https://www.sparxsystems.com/resources/user-guides/15.1/index.html>. (24 Februari 2020)
12. Richy Rotuahta Saragih. 2018.  
[https://www.researchgate.net/profile/Richy\\_Saragih/publication/329885312 PEMROGRAMAN DAN BAHASA PEMROGRAMAN/links/5c209488458515a4c7f4b123/PEMROGRAMAN-DAN-BAHASA-PEMROGRAMAN.pdf](https://www.researchgate.net/profile/Richy_Saragih/publication/329885312 PEMROGRAMAN DAN BAHASA PEMROGRAMAN/links/5c209488458515a4c7f4b123/PEMROGRAMAN-DAN-BAHASA-PEMROGRAMAN.pdf). (24 Februari 2020)
13. Masruro, Siti Tazkiyatul, Syaifudin Ramadhani, dan Urifatun Anis. Rancang Bangun Sistem Informasi Geografis Layanan Kesehatan Di Kecamatan Lamongan Dengan PHP MySQL. *Jurnal Teknika*. 5(2), 2013.

14. Haviluddin, Agus Tri Haryono, dan Dwi Rahmawati Mulawarman. 2016. Aplikasi program PHP dan MySQL. Samarinda. Mulawarman University PRESS
15. Danuri, Muhamad. Object Oriented Programming (OOP) Pembangunan Program Aplikasi Berbasis Windows. *Jurnal*. 5(1), 2009
16. Ropianto, Muhammad. Pemahaman Penggunaan Unified Modelling Language. *Jurnal Teknik Ibnu Sina*. 01(01), 2016
17. Wahono, Romi Satria, dan Sri Dhawiyanti. 2003. Pengantar Unified Modeling Language (UML). *Jurnal*