

SKRIPSI

PEMBANGUNAN APLIKASI *ANDROID IFSTUDENTPORTAL*



David Christopher Sentosa

NPM: 2017730015

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2020

UNDERGRADUATE THESIS

DEVELOPMENT OF NATIVE ANDROID APPLICATION IFSTUDENTPORTAL



David Christopher Sentosa

NPM: 2017730015

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2020**

LEMBAR PENGESAHAN

PEMBANGUNAN APLIKASI ANDROID IFSTUDENTPORTAL

David Christopher Sentosa

NPM: 2017730015

Bandung, «tanggal» «bulan» 2020

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Pascal Alfadian Nugroho

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PEMBANGUNAN APLIKASI *ANDROID IFSTUDENTPORTAL*

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» 2020

Meterai Rp. 6000

David Christopher Sentosa
NPM: 2017730015

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ... »

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Bandung, «bulan» 2020

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	1
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 Portal Akademik Mahasiswa 2018	5
2.1.1 Analisis Fitur Portal Akademik Mahasiswa 2018	5
2.2 <i>Jsoup</i>	9
2.2.1 Jsoup	9
2.2.2 Connection	10
2.2.3 Response	11
2.2.4 Document	11
2.2.5 Elements	11
2.2.6 Element	12
2.3 SIA Models versi 5.0.0	12
2.3.1 Mahasiswa	12
2.3.2 Nilai	13
2.3.3 Mata Kuliah	13
2.3.4 Kelulusan	13
2.3.5 Tahun Semester	13
2.4 <i>Android Design</i>	14
2.4.1 <i>Material Design</i>	14
2.4.2 Kualitas Aplikasi	15
3 ANALISIS	19
3.1 Analisis Sistem Kini	19
3.1.1 Analisis Fitur IFStudentPortal	19
3.2 Analisis Sistem Usulan	23
3.2.1 Pemenuhan Panduan <i>Material Design</i>	23
3.2.2 Pemenuhan Panduan <i>Kualitas Aplikasi</i>	24
3.2.3 Persiapan	25

4 PERANCANGAN	27
4.1 Perancangan Kelas	27
4.1.1 Model	27
4.1.2 View	32
4.1.3 Presenter	41
4.2 Perancangan Antarmuka	44
5 IMPLEMENTASI DAN PENGUJIAN	47
5.1 Implementasi	47
5.1.1 Lingkungan Implementasi	47
5.1.2 Hasil Implementasi	48
5.2 Pengujian	52
DAFTAR REFERENSI	53
A KODE PROGRAM	55
B HASIL EKSPERIMEN	57

DAFTAR GAMBAR

2.1	Tampilan halaman awal Portal Akademik Mahasiswa	5
2.2	Tampilan Login	6
2.3	Tampilan halaman FRS/PRS	7
2.4	Tampilan halaman profil mahasiswa	7
2.5	Tampilan halaman pembayaran	8
2.6	Tampilan halaman nilai	9
2.7	Tampilan halaman indeks prestasi	9
3.1	Tampilan halaman utama IFStudentPortal	19
3.2	Tampilan Login	20
3.3	Tampilan halaman daftar prasyarat mata kuliah	21
3.4	Tampilan halaman jadwal kuliah yang tersusun	22
3.5	Tampilan halaman ringkasan data akademik	23
4.1	Diagram Kelas Keseluruhan	27
4.2	Diagram Kelas Package Model	28
4.3	Diagram Kelas Package View	33
4.4	Diagram Kelas Package Presenter	41
4.5	Rancangan Halaman <i>Login</i>	44
4.6	Rancangan Halaman Utama	45
4.7	Rancangan Halaman Prasyarat Mata Kuliah	45
4.8	Rancangan Halaman Jadwal Kuliah	46
4.9	Rancangan Halaman Syarat Kelulusan	46
5.1	Halaman <i>Login</i>	48
5.2	Halaman <i>Home</i>	49
5.3	Halaman Jadwal	50
5.4	Halaman Persiapan Perwalian	51
5.5	Halaman Syarat Kelulusan	52
B.1	Hasil 1	57
B.2	Hasil 2	57
B.3	Hasil 3	57
B.4	Hasil 4	57

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Portal Akademik Mahasiswa adalah sistem informasi berbasis *web* yang digunakan oleh mahasiswa UNPAR [1]. IFStudentPortal adalah sistem informasi berbasis *web* khusus mahasiswa Teknik Informatika UNPAR hasil pengembangan lebih lanjut dari Portal Akademik Mahasiswa [2].

Skripsi ini dibuat untuk membawa IFStudentPortal ke platform *Android* sebagai aplikasi *native Android* IFStudentPortal dengan dibantu *library* jsoup dan mengikuti pedoman *Android Design*.

Aplikasi akan dibangun dengan bahasa *native Android* yaitu Java. Fitur-fitur yang tersedia di situs IFStudentPortal akan dibawa ke platform *Android* dan dibuat agar terintegrasi dengan fitur yang dimiliki *Android*. Aplikasi *Android* IFStudentPortal akan mengambil data langsung dari Portal Akademik Mahasiswa dengan dibantu *library* jsoup. Untuk melakukan pengambilan data, jsoup akan langsung berkomunikasi dengan Portal Akademik Mahasiswa.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas di skripsi ini adalah sebagai berikut :

- Fitur apa saja yang akan tersedia di aplikasi *Android* IFStudentPortal?
- Bagaimana cara mengimplementasikan fungsi aplikasi *Android* IFStudentPortal dengan *library* jsoup?
- Bagaimana cara mengimplementasikan tampilan aplikasi *Android* IFStudentPortal dengan mengikuti pedoman *Android Design*?

1.3 Tujuan

Tujuan yang ingin dicapai dari penulisan skripsi ini sebagai berikut :

- Mengetahui fitur apa saja yang akan tersedia di IFStudentPortal.
- Mengimplementasikan fungsional aplikasi *Android* IFStudentPortal menggunakan *library* jsoup.
- Mengimplementasikan tampilan aplikasi *Android* IFStudentPortal mengikuti pedoman *Android design*.

Fokus utama dari penggerjaan skripsi ini adalah bagaimana membangun aplikasi *Android* dengan mengikuti dokumentasi dan pedoman resmi *Android*, bukan tentang aspek-aspek IFStudentPortal sebagai sebuah sistem informasi.

1.4 Batasan Masalah

Beberapa batasan yang dibuat terkait dengan penggerjaan skripsi ini adalah sebagai berikut:

1. Aplikasi akan diuji menggunakan format NPM mahasiswa angkatan 2017 ke atas.
2. Aplikasi hanya akan diuji pada perangkat jenis ponsel pintar.
3. Panduan *Material Design* difokuskan pada panduan aksesibilitas dan panduan platform.
4. Penggerjaan skripsi ini tidak mencakup pemeliharaan *library* SIA Models untuk mengikuti peraturan kurikulum yang baru tentang program peminatan kecuali jika diminta oleh pembimbing.

1.5 Metodologi

Metodologi yang dilakukan pada skripsi ini adalah sebagai berikut:

1. Melakukan studi mengenai *library* jsoup untuk mengambil data dari Portal Akademik Mahasiswa, *Android Design*, dan skripsi Herfan Heryandi untuk membangun aplikasi *Android IFStudentPortal*.
2. Menganalisis Portal Akademik Mahasiswa dan IFStudentPortal.
3. Merancang model IFStudentPortal.
4. Mengimplementasi aplikasi *Android IFStudentPortal* menggunakan *library* jsoup dengan mengikuti pedoman *Android Design*.
5. Melakukan eksperimen dan pengujian.
6. Membuat dokumentasi.

1.6 Sistematika Pembahasan

Sistematika penulisan setiap bab pada skripsi ini adalah sebagai berikut:

1. Bab Pendahuluan
Bab 1 berisi latar belakang, rumusan masalah, tujuan, metode penelitian, dan sistematika penulisan yang digunakan untuk menyusun skripsi ini.
2. Bab Dasar Teori
Bab 2 berisi teori-teori yang digunakan dalam pembuatan skripsi ini. Teori yang digunakan yaitu *library* jsoup, SIA Models, dan *Android Design*.
3. Bab Analisis
Bab 3 berisi analisis yang dilakukan pada skripsi ini, meliputi analisis sistem, analisis kebutuhan aplikasi *Android IFStudentPortal*, analisis pedoman *Android Design*, analisis Portal Akademik Mahasiswa dan situs IFStudentPortal untuk fitur IFStudentPortal, analisis *use case* meliputi diagram *use case* dan skenario, serta analisis desain kelas.
4. Bab Perancangan
Bab 4 berisi perancangan aplikasi, meliputi diagram kelas rinci beserta deskripsi kelas dan fungsinya dan perancangan antarmuka aplikasi.

5. Bab Implementasi dan Pengujian

Bab 5 berisi implementasi dan pengujian aplikasi, meliputi lingkungan implementasi, hasil implementasi, pengujian fungsional, dan pengujian eksperimental.

6. Bab Kesimpulan dan Saran

Bab 6 berisi kesimpulan dari hasil pembangunan aplikasi beserta saran untuk pengembangan berikutnya.

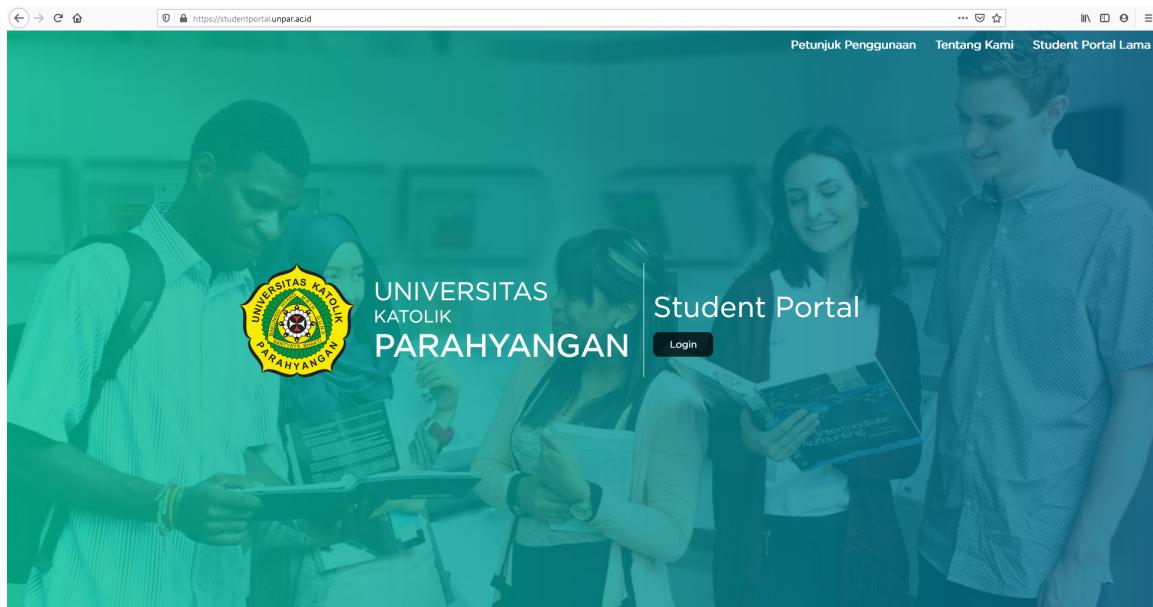
BAB 2

LANDASAN TEORI

Pada bab ini akan dijelaskan dasar teori mengenai *library* jsoup yang meliputi kelas-kelas Jsoup, Connection, Response, Document, Elements, dan Element. Akan dibahas pula Portal Akademik Mahasiswa, IFStudentPortal, SIA Models, dan panduan *Android Design* yang meliputi Panduan Aksesibilitas dan panduan *Material Design*.

2.1 Portal Akademik Mahasiswa 2018

Portal Akademik Mahasiswa adalah sistem informasi berbasis *web* yang diperuntukan bagi mahasiswa untuk memperoleh informasi kegiatan akademik [1]. Portal Akademik Mahasiswa dapat di akses melalui URL <https://www.studentportal.unpar.ac.id/>. Mahasiswa harus *login* dengan email mahasiswa agar bisa mengakses dan menggunakan fitur-fitur yang tersedia.



Gambar 2.1: Tampilan halaman awal Portal Akademik Mahasiswa

2.1.1 Analisis Fitur Portal Akademik Mahasiswa 2018

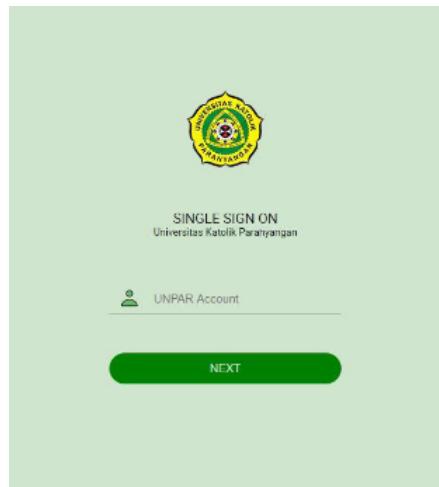
Pada sub-bab ini akan dijelaskan mengenai fitur-fitur yang ada di Portal Akademik Mahasiswa 2018

1. *Login* : Untuk dapat menggunakan situs Portal Akademik Mahasiswa, mahasiswa harus *login* menggunakan email *student* mahasiswa.

- Nama: *Login*
- Aktor: Mahasiswa

- Deskripsi: *Login* ke Portal Akademik Mahasiswa
- Kondisi awal: Mahasiswa adalah mahasiswa Universitas Katolik Parahyangan
- Kondisi akhir: Halaman utama Portal Akademik Mahasiswa ditampilkan
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa mengakses situs Portal Akademik Mahasiswa	Sistem menampilkan halaman depan
2	Mahasiswa menekan tombol <i>login</i>	Sistem menampilkan halaman <i>login</i>
3	Mahasiswa mengisi email dan kata sandi lalu menekan tombol <i>login</i>	Sistem melakukan pengecekan identitas <i>login</i>
4		Jika login berhasil maka sistem akan menampilkan halaman utama Portal Akademik Mahasiswa

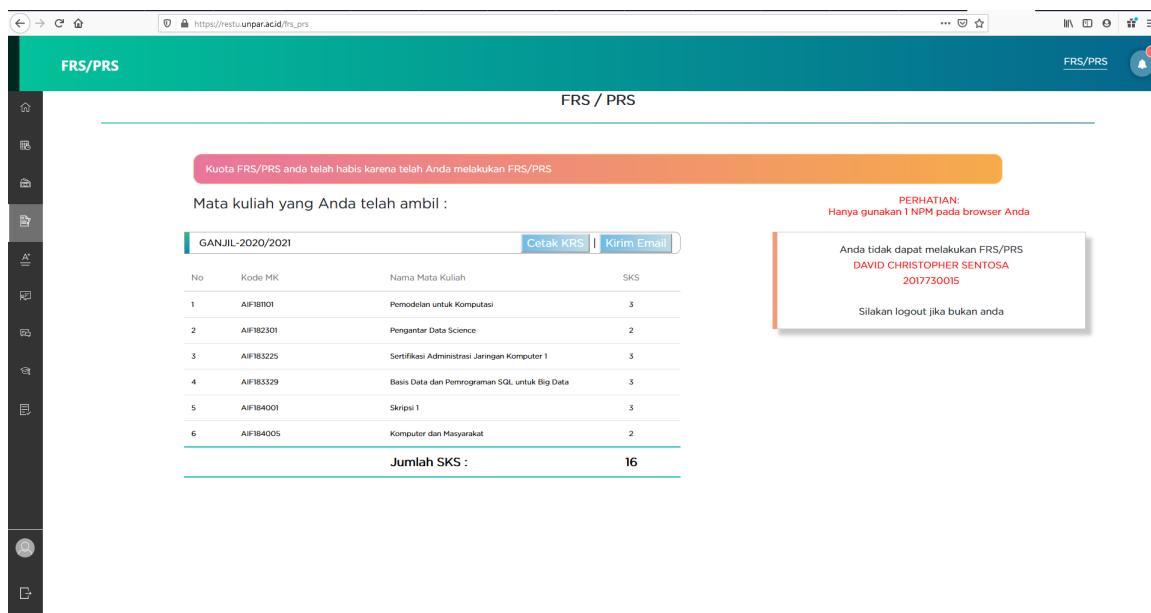


Gambar 2.2: Tampilan Login

2. *FRS/PRS Daring* : Mahasiswa dapat mengisi form rencana studi (FRS) atau melakukan perubahan rencana studi (PRS) secara daring di situs Portal Akademik Mahasiswa.

- Nama: FRS/PRS Daring
- Aktor: Mahasiswa
- Deskripsi: Mengisi form rencana studi (FRS) atau melakukan perubahan rencana studi (PRS) secara daring
- Kondisi awal: Mahasiswa sudah *login* ke Portal Akademik Mahasiswa
- Kondisi akhir: Mahasiswa berhasil mengisi form rencana studi atau merubah rencana studi
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa menekan tombol FRS/PRS	Sistem menampilkan halaman FRS/PRS
2	Mahasiswa mengisi form rencana studi atau merubah rencana studi	
3	Mahasiswa menekan tombol selesai	Sistem mencatat hasil FRS/PRS mahasiswa

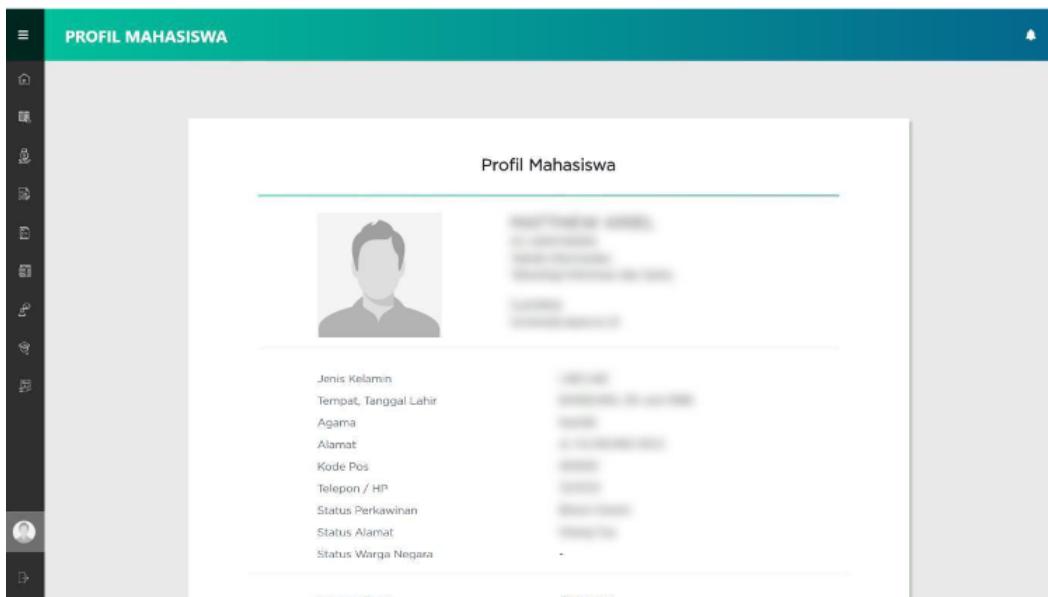


Gambar 2.3: Tampilan halaman FRS/PRS

3. *Profil Mahasiswa* : Mahasiswa dapat melihat informasi data diri di halaman profil mahasiswa.

- Nama: Profil mahasiswa
- Aktor: Mahasiswa
- Deskripsi: Melihat data diri mahasiswa
- Kondisi awal: Mahasiswa sudah *login* ke Portal Akademik Mahasiswa
- Kondisi akhir: Mahasiswa disajikan informasi data diri
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa menekan tombol profil mahasiswa	Sistem menampilkan halaman profil mahasiswa



Gambar 2.4: Tampilan halaman profil mahasiswa

4. *Pembayaran* : Mahasiswa dapat melihat informasi tentang tagihan pembayaran, riwayat pembayaran, dan keterangan tata cara pembayaran.

- Nama: Profil mahasiswa
- Aktor: Mahasiswa
- Deskripsi: Melihat informasi pembayaran
- Kondisi awal: Mahasiswa sudah *login* ke Portal Akademik Mahasiswa
- Kondisi akhir: Mahasiswa disajikan informasi pembayaran
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa menekan tombol pembayaran	Sistem menampilkan halaman pembayaran



Gambar 2.5: Tampilan halaman pembayaran

5. *Nilai* : Mahasiswa dapat melihat informasi nilai mata kuliah yang diambil mahasiswa untuk setiap semesternya. Informasi nilai yang tersedia antara lain nilai tugas, angka rata-rata tugas (ART), ujian tengah semester (UTS), ujian akhir semester (UAS), dan TOEFL. Mahasiswa juga bisa melihat grafik perkembangan indeks prestasi kumulatif (IPK) dan indeks prestasi semester (IPS).

- Nama: Profil mahasiswa
- Aktor: Mahasiswa
- Deskripsi: Melihat informasi nilai
- Kondisi awal: Mahasiswa sudah *login* ke Portal Akademik Mahasiswa
- Kondisi akhir: Mahasiswa disajikan informasi nilai
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa menekan tombol nilai	Sistem menampilkan halaman nilai

No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Kelas	Nilai	AA	NA	Statistik Hasil Nilai
1	AIF82001	Penelitian 1	3	A	Tampilkan Detail Nilai	83	A	Tampilkan Grafik Nilai
2	AIF83007	Pengantar Sistem Cerdas	3	B	Tampilkan Detail Nilai	79	A-	Tampilkan Grafik Nilai
3	AIF83117	Grafika Komputer	2	A	Tampilkan Detail Nilai	81	A	Tampilkan Grafik Nilai
4	AIF83119	Kemanan Informati	2	A	Tampilkan Detail Nilai	83	A	Tampilkan Grafik Nilai
5	AIF83209	Pemrograman pada Perangkat Bergerak	3	A	Tampilkan Detail Nilai	84	A	Tampilkan Grafik Nilai
6	AIF83303	Roketaya Perangkat Lunak	3	B	Tampilkan Detail Nilai	67	B-	Tampilkan Grafik Nilai
7	AIF83305	Manajemen Projek	2	A	Tampilkan Detail Nilai	72	B	Tampilkan Grafik Nilai

Keterangan:
\$: Nilai tidak dapat dilihat karena status pembayaran belum lunas
#: Nilai belum tersedia
%: Nilai sedang dalam proses

Gambar 2.6: Tampilan halaman nilai



Gambar 2.7: Tampilan halaman indeks prestasi

2.2 Jsoup

Jsoup adalah *library* Java yang digunakan untuk mengambil data berupa HTML dari sebuah situs. Data HTML tersebut bisa digunakan untuk memperoleh informasi yang diperlukan dari suatu situs^[3]. *Library jsoup* ini bisa digunakan saat sebuah situs tidak menyediakan API untuk memberikan datanya. Kelas-kelas yang digunakan dari *library* ini akan dijelaskan di sub bab berikut.

2.2.1 Jsoup

Kelas ini merupakan kelas utama dalam *library* jsoup. Seluruh *method* dalam kelas ini adalah **static** *method* sehingga kelas ini tidak perlu dikonstruksi terlebih dahulu sebelum menggunakan *method* dari kelas ini. Salah satu *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public static Connection connect(String url)**

Berfungsi untuk membuat koneksi baru dengan suatu situs web.

Parameter:

- **url** : URL situs web dengan protokol HTTP atau HTTPS.

Kembalian: koneksi dengan situs web.

2.2.2 Connection

Kelas ini merupakan **interface** yang berguna untuk pengambilan data dari situs web. Beberapa **method** yang dimiliki kelas ini adalah sebagai berikut:

- **Connection cookies(Map<String, String> cookies)**

Berfungsi untuk menambahkan *cookie*.

Parameter:

- **cookies** : Map dari *cookie*.

Kembalian: koneksi yang sama tetapi sudah diberi *cookies*.

- **Connection data(String key, String value)**

Berfungsi untuk menambahkan parameter data yang bisa dikirim melalui metode HTTP GET atau POST.

Parameter:

- **key** : kunci data.
 - **value** : nilai data.

Kembalian: koneksi yang sama tetapi sudah diberi parameter data.

- **Connection method(Connection.Method method)**

Berfungsi untuk mengatur metode permintaan HTTP, GET atau POST. Metode pengiriman secara *default* adalah GET

Parameter:

- **method** : metode pengiriman permintaan HTTP.

Kembalian: koneksi yang sama tetapi sudah diatur metode permintaannya.

- **Connection timeout(int millis)**

Berfungsi untuk mengatur batas waktu *request*. Batas waktu nol akan dianggap sebagai batas waktu yang tak terhingga.

Parameter:

- **millis** : batas waktu dalam milidetik.

Kembalian: koneksi yang sama tetapi sudah diberi batas waktu untuk *request*.

- **Connection validateTLCertificates(boolean value)**

Berfungsi untuk mengatur pemeriksaan sertifikat TLS untuk permintaan HTTPS. Nilai **true** untuk memeriksa dan nilai **false** untuk tidak memeriksa.

Parameter:

- **value** : status pemeriksaan sertifikat TLS.

Kembalian: koneksi yang sama tetapi sudah diatur pemeriksaan sertifikatnya.

- **Connection.Response execute()**
Berfungsi untuk mengirim permintaan HTTP.
Kembalian: objek Response.

2.2.3 Response

Kelas ini merepresentasikan permintaan HTTP. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **Map<String, String> cookies()**
Method ini berfungsi untuk mendapatkan seluruh *cookies*.
Kembalian: seluruh *cookies*.
- **Document parse()**
Berfungsi untuk merubah *body* jawaban menjadi dokumen.
Kembalian: dokumen yang sudah dirubah.
- **String body()**
Berfungsi untuk mendapatkan *body* jawaban dalam bentuk *string*.
Kembalian: *body* jawaban dalam bentuk *string*.

2.2.4 Document

Kelas ini merepresentasikan dokumen HTML. Salah satu *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public Elements select(String cssQuery)**
Method ini diturunkan dari kelas Element, berfungsi untuk menemukan elemen HTML yang sesuai dengan kueri CSS.
Parameter:
 - **cssQuery** : kueri CSS berupa CSS Selector.**Kembalian:** elemen-elemen HTML yang sesuai dengan kueri CSS.

2.2.5 Elements

Kelas ini merepresentasikan kumpulan elemen HTML. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public Elements select(String query)**
Berfungsi untuk menemukan elemen-elemen yang sesuai dalam *list* elemen.
Parameter:
 - **query** : kueri CSS berupa CSS Selector.**Kembalian:** elemen-elemen yang sudah diseleksi sesuai kueri.
- **public String val()**
Berfungsi untuk mendapatkan nilai dari elemen pertama.
Kembalian: nilai elemen.
- **public String text()**
Method Berfungsi untuk mendapatkan kombinasi teks dari seluruh elemen yang sesuai.
Kembalian: seluruh teks dalam *string*.

2.2.6 Element

Kelas ini merepresentasikan sebuah elemen HTML yang berisikan *tag*, atribut, dan anak elemen. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public Element child(int index)**

Berfungsi untuk mendapatkan anak elemen berdasarkan nomor index.

Parameter:

- **index** : nomor index.

Kembalian: anak elemen.

- **public Element children()**

Berfungsi untuk mendapatkan seluruh anak elemen.

Kembalian: seluruh anak elemen.

- **public String className()**

Berfungsi untuk mendapatkan nama kelas elemen.

Kembalian: nama kelas elemen.

- **public String text()**

Berfungsi untuk mendapatkan teks dari elemen.

Kembalian: teks dalam *string*.

2.3 SIA Models versi 5.0.0

SIA Models merupakan *library* Java yang merepresentasikan Sistem Informasi Akademik Teknik Informatika UNPAR [4]. Saat ini SIA Models mendukung kurikulum 2018. Kelas-kelas yang digunakan dari *library* ini akan dijelaskan di sub bab berikut.

2.3.1 Mahasiswa

Kelas ini merepresentasikan seorang mahasiswa Universitas Katolik Parahyangan. Kelas ini memiliki atribut antara lain nama, npm, tanggal lahir, jenis kelamin, riwayat nilai, foto, dan jadwal kuliah. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public byte[] getPhotoImage() throws IOException, MalformedURLException**

Berfungsi untuk mendapatkan foto profil mahasiswa dalam bentuk *byte array*.

Kembalian: *byte array* foto profil mahasiswa.

- **public double calculateIPK() throws ArrayIndexOutOfBoundsException**

Berfungsi untuk menghitung nilai index Prestasi Kumulatif mahasiswa.

Kembalian: nilai IPK mahasiswa.

- **public double calculateIPS(TahunSemester tahunSemester) throws ArrayIndexOutOfBoundsException**

Berfungsi untuk menghitung nilai index Prestasi Semester mahasiswa.

Parameter:

- **tahunSemester** : tahun dan semester yang ingin dihitung IPS-nya.

Kembalian: nilai IPK mahasiswa pada tahun dan semester yang ditentukan.

- **public int calculateSKSTempuh(boolean lulusSaja) throws ArrayIndexOutOfBoundsException**

Berfungsi untuk Menghitung jumlah SKS tempuh mahasiswa saat ini.

Parameter:

- **lulusSaja** : set *true* untuk menghitung hanya SKS yang lulus.

Kembalian: jumlah SKS yang sudah ditempuh mahasiswa.

2.3.2 Nilai

Kelas ini merepresentasikan nilai sebuah mata kuliah mahasiswa Universitas Katolik Parahyangan. Kelas ini memiliki atribut antara lain mata kuliah, kelas, nilai UTS, nilai UAS, dan nilai akhir. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public Double getAngkaAkhir()**

Berfungsi untuk mendapatkan angka akhir dari index nilai.

Kembalian: angka akhir dari nilai.

2.3.3 Mata Kuliah

Kelas ini merepresentasikan sebuah mata kuliah. Kelas ini memiliki atribut antara lain nama, kode, dan sks.

2.3.4 Kelulusan

Kelas ini merepresentasikan kelulusan mahasiswa Universitas Katolik Parahyangan. Kelas ini memiliki atribut daftar mata kuliah wajib lulus sebagai syarat kelulusan. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public boolean checkPrasyarat(Mahasiswa mahasiswa, List<String> reasonsContainer)**

Berfungsi untuk memeriksa apakah mahasiswa sudah memenuhi syarat kelulusan.

Parameter:

- **mahasiswa** : mahasiswa yang ingin diperiksa prasyarat kelulusannya.
- **reasonsContainer** : daftar alasan-alasan yang membuat tidak lulus.

Kembalian: nilai *boolean true* atau *false* yang merepresentasikan status kelulusan prasyarat mahasiswa.

- **public Map<String, String> getMkEkivalensi()**

Berfungsi menambahkan data mata kuliah kurikulum 2013 yang ekivalen dengan mata kuliah kurikulum 2018.

Kembalian: daftar mata kuliah yang ekivalen dengan kurikulum 2018 dalam bentuk *map*.

2.3.5 Tahun Semester

Kelas ini merepresentasikan tahun dan semester masa studi mahasiswa di Universitas Katolik Parahyangan. Kelas ini memiliki atribut kode tahun semester yang memiliki 3 digit dimana 2 digit pertama adalah tahun dan digit terakhir adalah kode semester (1 untuk ganjil, 2 untuk genap, 4 untuk pendek). Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **private static void validateKodeSemester(String kodeTahunSemester) throws IllegalArgumentException**

Berfungsi untuk memvalidasi kode tahun semester.

Parameter:

- **kodeTahunSemester** : kode tahun semester yang akan divalidasi.

Eksepsi: kode semester tidak valid.

2.4 *Android Design*

Dalam mengembangkan aplikasi *Android*, ada pedoman desain yang diberikan. Desain tampilan aplikasi *Android* mengacu pada *Android Material Design*. Pedoman kualitas aplikasi juga diberikan untuk aspek kompatibilitas, keamanan, performa, dll. Kedua pedoman ini diberikan agar aplikasi yang dihasilkan bisa memiliki tampilan dan perilaku yang konsisten dengan platform *Android*.

2.4.1 *Material Design*

Material Design terdiri dari panduan, komponen, dan alat-alat untuk mendukung pembuatan tampilan antarmuka yang baik. *Material Design* bertujuan mempermudah kolaborasi antara desainer dan pengembang aplikasi untuk membuat produk yang cantik dengan cepat[5]. Ada 2 panduan utama yang harus diikuti dari *Material Design* yaitu panduan aksesibilitas dan panduan platform.

Panduan Aksesibilitas

Aksesibilitas dapat diartikan sebagai tingkat kemudahan saat pengguna mempelajari dan menggunakan tampilan antarmuka aplikasi. Meningkatkan aksesibilitas akan meningkatkan *usability* atau tingkat kebergunaan aplikasi kepada berbagai kalangan pengguna [5]. Hal penting yang harus diperhatikan agar aplikasi memiliki aksesibilitas yang baik :

- (HR-01) Hierarki : Elemen-elemen harus terlihat dengan jelas baik dari ukuran, kontras, dan infomasi yang terkandung di dalamnya. Elemen-elemen juga harus terurut berdasarkan kepentingan dan elemen dengan fungsi tertentu diletakan di tempat yang mudah dijangkau.
- (WK-01) Warna dan Kontras : Kontras yang cukup tinggi akan membuat elemen terlihat jelas, namun jika terlalu tinggi juga akan kurang nyaman dilihat terlalu lama, jika terlalu rendah maka elemen-elemen akan sulit dibedakan satu sama lain (misal warna teks dengan latar belakangnya). Jika warna digunakan sebagai suatu indikator, maka perlu ada tambahan keterangan lain agar pengguna yang menderita buta warna tetap bisa menerima informasi dengan baik.
- (TL-01) Tata Letak dan Tipografi : Menggunakan *layout* yang fleksibel dan resposif akan membantu isi konten menyesuaikan dengan skala layar agar konten tidak ada yang terpotong dengan tidak sengaja. Pastikan menggunakan format sp untuk ukuran *font* agar ukuran teks ikut terskala dengan baik jika pengguna merubah ukuran *font* dari pengaturan sistem perangkat. Pastikan juga ada ruang yang cukup untuk menampung ukuran teks yang diperbesar.
- (GB-01) Gambar : Gunakan gambar untuk memperjelas informasi yang disajikan. Gambar logo boleh tidak mematuhi panduan warna, kontras, dan ukuran teks, namun sebaiknya tetap memiliki fungsi (misalnya logo sebagai tombol ke halaman utama).

Panduan Platform

Panduan platform membantu menentukan bagaimana ketentuan yang akan digunakan untuk setiap platform. Platform yang dimaksud pada bagian ini adalah bagian-bagian dari komponen aplikasi Android. Platform yang dimaksud adalah :

- (NT-01) Notifikasi : notifikasi digunakan untuk memberi tahu komunikasi dari pengguna lain dan mengingatkan hal yang perlu dilakukan. Notifikasi bisa ditampilkan di halaman terkunci, di *status bar*, dengan kedipan lampu LED, dan dengan suara dan getaran. Sebaiknya notifikasi tidak digunakan untuk promosi, meminta rating aplikasi, dan memberi tahu proses yang tidak berhubungan dengan pengguna. Pada saat layar terkunci sebaiknya notifikasi tidak menampilkan informasi yang sensitif. Notifikasi harus punya bagian *header*, bagian konten, dan bagian aksi.
- (IA-01) Izin Akses (*permission*) : Secara normal, aplikasi punya izin akses ke beberapa hal tanpa perlu memintanya kepada pengguna. Namun beberapa hal berikut membutuhkan konfirmasi dari pengguna agar aplikasi memiliki izin akses :
 - Kalender : Mengakses dan mengatur kalender.
 - Kamera : Mengambil foto dan merekam video.
 - Kontak : Membaca dan mengatur kontak.
 - Lokasi : Membaca lokasi perangkat saat ini.
 - Mikrofon : Merekam suara.
 - Telepon : Membuat dan mengatur panggilan telepon.
 - Sensor tubuh : Membaca detak jantung dan hal sejenis lainnya.
 - Media penyimpanan : Mengakses dan menggunakan media penyimpanan.
 - SMS : Mengirim dan membaca SMS (pesan singkat).

2.4.2 Kualitas Aplikasi

Pengguna tentu menginginkan aplikasi berkualitas. Kualitas aplikasi akan menentukan keberhasilan aplikasi dalam hal instalasi, ulasan, loyalitas dan keterlibatan pengguna untuk jangka panjang [6]. Dokumentasi *Android Developers* memberikan kriteria untuk mengukur kualitas aplikasi sebagai berikut :

- Desain visual dan interaksi pengguna
- Fungsionalitas
- Kompatibilitas, performa, dan stabilitas
- Keamanan
- *Google Play*

Untuk keterangan lebih lanjut dari setiap poin diatas akan dijelaskan di bagian berikutnya.

Desain Visual dan Interaksi Pengguna

Kriteria ini bertujuan memastikan aplikasi akan memiliki desain visual dan pola interaksi standar agar pengalaman pengguna konsisten dan intuitif[7]. Kriteria ini dapat dijabarkan sebagai berikut :

- (UX-B1) Aplikasi tidak boleh merubah definisi ikon sistem dengan fungsinya, jika aplikasi menyediakan ikon yang disesuaikan, maka tampilannya harus mirip ikon standar dan memicu perilaku sesuai fungsi standarnya.

- (UX-S2) Aplikasi hanya menggunakan notifikasi untuk memberitahu perubahan yang terjadi dengan konteks yang berkaitan dengan pengguna pribadi dan untuk memberi tahu informasi/kontrol terhadap kejadian yang sedang berlangsung.

Fungsionalitas

Kriteria ini bertujuan memastikan aplikasi memberikan perilaku fungsional yang diharapkan, dengan tingkat izin yang sesuai [7]. Kriteria ini dapat dijabarkan sebagai berikut :

- (FN-P1) Aplikasi hanya meminta izin untuk mendukung fungsionalitas aplikasi tersebut. Aplikasi tidak boleh meminta izin untuk mengakses data sensitif atau menggunakan layanan yang bisa membebani pengguna kecuali jika fitur inti aplikasi memerlukan izin tersebut.
- (FN-L1) Aplikasi harus berfungsi normal jika dipasang di kartu *SD*.
- (FN-A1) Audio tidak boleh diputar di layar utama, saat layar mati, dibalik layar, atau saat layar dikunci kecuali memutar audio adalah fitur utama.
- (FN-U1) Jika memungkinkan aplikasi mendukung orientasi *landscape* dan *portrait*, dan menggunakan seluruh layar untuk kedua orientasi.
- (FN-S1) Aplikasi tidak boleh membiarkan layanan tetap aktif saat di latar belakang layar, kecuali jika diperlukan fitur utama.
- (FN-S2) Aplikasi mempertahankan status pengguna atau aplikasi saat meninggalkan latar depan dan mencegah kehilangan data tanpa sengaja akibat navigasi mundur dan perubahan status lainnya.

Kompatibilitas, Performa, dan Stabilitas

Kriteria ini bertujuan memastikan aplikasi memberikan kompatibilitas, performa, stabilitas, dan daya respons yang diharapkan oleh pengguna [7]. Kriteria ini dapat dijabarkan sebagai berikut :

- (PS-S1) Aplikasi diharapkan tidak macet, berfungsi tidak normal, menutup sendiri di perangkat yang menjalankannya.
- (PS-P1) Aplikasi dimuat dengan cepat atau memberikan indikasi kepada pengguna tentang kapan aplikasi selesai dimuat.
- (PS-T1) Aplikasi dibuat dengan *SDK* terbaru dan berjalan di *Android* versi terbaru tanpa kendala.
- (PS-M1) Aplikasi memutar video dan audio dengan lancar, tidak tersendat, suara dan gambar tidak pecah, atau cacat lainnya.
- (PS-V1) Aplikasi menyediakan grafik berkualitas tinggi untuk semua ukuran layar yang ditargetkan dan menampilkan elemen antarmuka tanpa pikselasi, distorsi, dan tidak bergerigi pada tepian.
- (PS-B1) Aplikasi mendukung fitur pengelolaan daya baterai (*Android 6.0+*).

Keamanan

Kriteria ini bertujuan memastikan aplikasi menangani dan mengamankan data pengguna dan informasi pribadi dengan benar[7]. Kriteria ini dapat dijabarkan sebagai berikut :

- (SC-D1) Aplikasi harus menyimpan data pribadi di penyimpanan internal aplikasi dan tidak boleh mencatat data pribadi di log.

- (SC-D2) Aplikasi harus memverifikasi data eksternal sebelum digunakan.
- (SC-P1) Aplikasi hanya boleh mengekspor komponen aplikasi yang membagikan data dengan aplikasi lain, atau komponen yang harus dipanggil oleh aplikasi lain.
- (SC-P2) Semua komponen aplikasi yang berbagi konten dengan aplikasi lain menetapkan (dan memberlakukan) izin yang sesuai, termasuk aktivitas, layanan, penerima siaran, dan khususnya penyedia konten.
- (SC-N1) Aplikasi harus menyatakan konfigurasi keamanan jaringan dan semua lalu lintas jaringan dilakukan melalui *SSL*.
- (SC-N2) Jika aplikasi menggunakan layanan *Google Play*, inisialisasi keamanan dilakukan saat aplikasi dimulai.
- (SC-U1) Aplikasi harus menggunakan dependensi, *library* dan *SDK* terbaru.
- (SC-E1) Aplikasi tidak boleh menjalankan kode dari luar aplikasi secara dinamis.
- (SC-C1) Aplikasi harus menggunakan algoritma kriptografi kuat yang disediakan oleh platform.

Google Play

Kriteria ini bertujuan memastikan aplikasi yang dibuat sudah layak, memenuhi standar dan syarat untuk dipublikasikan di layanan *Google Play*[7]. Kriteria ini dapat dijabarkan sebagai berikut :

- (GP-P1) Aplikasi mematuhi Kebijakan Materi Pengembang *Google Play* (tidak menawarkan materi tidak pantas, tidak menggunakan hak kekayaan intelektual atau merk orang lain, dll).
- (GP-D1) Aplikasi sudah memenuhi kriteria yang sudah diuraikan sebelum bagian ini.
- (GP-X1) Pengembang aplikasi harus mengatasi *bug* yang disampaikan di halaman ulasan di layanan *Google Play* jika *bug* tersebut ditemukan di banyak perangkat dan berulang kali atau ditemukan di perangkat terbaru atau perangkat paling populer.

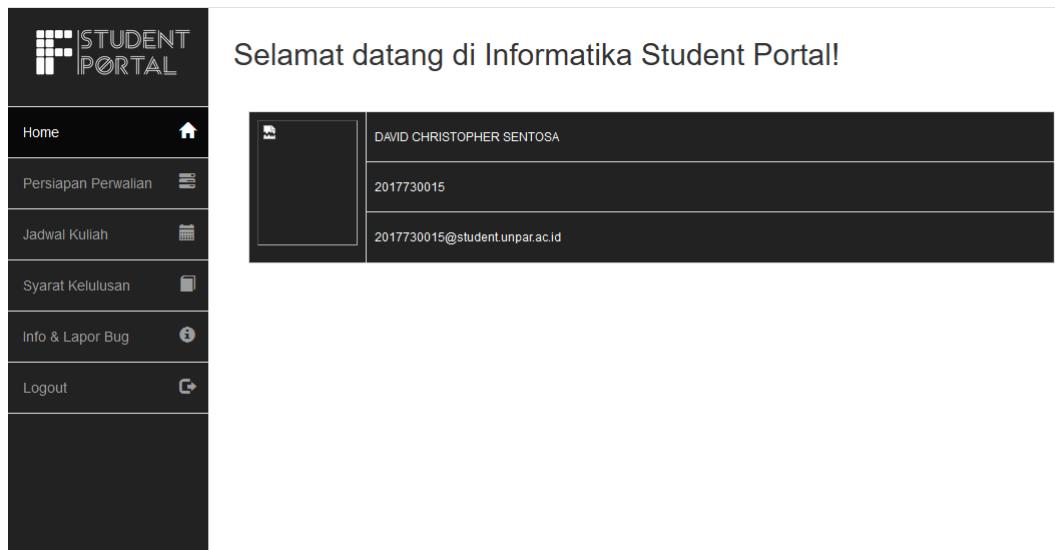
BAB 3

ANALISIS

Pada bab ini akan dijelaskan analisis sistem yang sudah berjalan dan sistem yang akan dibangun. Analisis yang akan dibahas meliputi analisis *use case*, analisis kebutuhan fungsional dan non fungsional, dan analisis pemenuhan panduan pembangunan aplikasi *Android*.

3.1 Analisis Sistem Kini

IFStudentPortal adalah situs yang diperuntukan bagi mahasiswa Teknik Informatika UNPAR^[2]. IFStudentPortal dibuat dengan tujuan menjadi pengembangan lebih lanjut dari Portal Akademik Mahasiswa pada masanya. IFStudentPortal adalah hasil skripsi Herfan Heryandi [2] dan kontributor lainnya. Saat ini IFStudentPortal telah mendukung kurikulum 2018 berkat kontribusi skripsi Andrianto Sugiarto [8]¹. Untuk mengakses Portal Akademik Mahasiswa, mahasiswa harus *login* menggunakan akun email *student*. Di halaman utama setelah login, terdapat tampilan profil mahasiswa, di sebelah kiri ada kolom menu untuk mengakses fitur-fitur IFStudentPortal seperti melihat ringkasan data akademik, melihat jadwal yang tersusun, dan melihat prasyarat mata kuliah.



Gambar 3.1: Tampilan halaman utama IFStudentPortal

3.1.1 Analisis Fitur IFStudentPortal

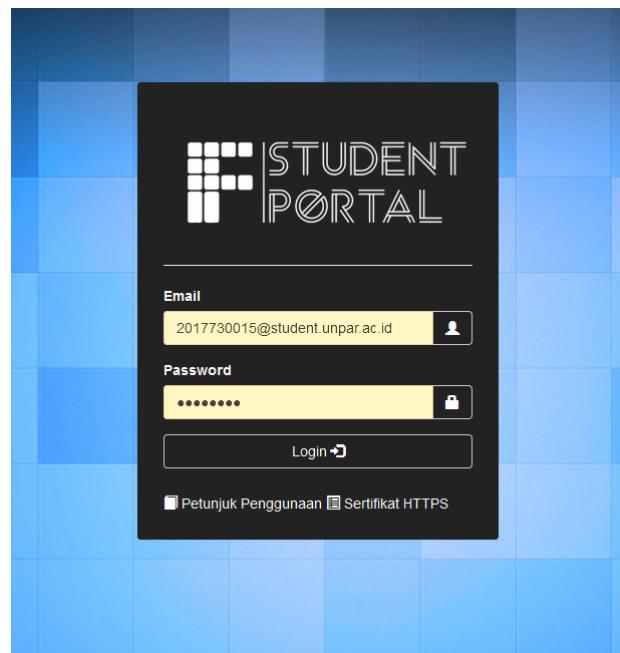
IFStudentPortal pada awalnya dibuat untuk melengkapi informasi yang belum ada di Portal Akademik Mahasiswa pada tahun 2015. Fitur-fitur yang diimplementasikan di IFStudentPortal adalah sebagai berikut :

¹Mahasiswa Teknik Informatika dapat mengakses IFStudentPortal melalui URL <https://ifstudentportal.herokuapp.com/>

1. *Login* : Untuk dapat menggunakan situs IFStudentPortal, mahasiswa Teknik Informatika UNPAR harus *login* menggunakan akun dan kata sandi yang sama dengan yang digunakan untuk *login* ke Portal Akademik Mahasiswa.

- Nama: *Login*
- Aktor: Mahasiswa
- Deskripsi: *Login* ke IFStudentPortal dengan membuka koneksi ke Portal Akademik Mahasiswa
- Kondisi awal: Mahasiswa adalah mahasiswa program studi Teknik Informatika UNPAR
- Kondisi akhir: Halaman utama IFStudentPortal ditampilkan
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa mengakses situs IFStudentPortal	Sistem menampilkan halaman <i>login</i>
2	Mahasiswa mengisi akun dan kata sandi lalu menekan tombol <i>login</i>	Sistem membuka koneksi ke Portal Akademik Mahasiswa untuk melakukan pengecekan identitas <i>login</i>
3		Jika login berhasil maka sistem akan menampilkan halaman utama IFStudentPortal



Gambar 3.2: Tampilan Login

2. Prasyarat Mata Kuliah Mahasiswa bisa melihat mata kuliah apa saja yang dibuka di semester berjalan dan memeriksa prasyarat untuk mata kuliah tersebut.

- Nama: Memeriksa prasyarat mata kuliah
- Aktor: Mahasiswa
- Deskripsi: Memeriksa prasyarat mata kuliah yang dibuka pada semester terkini
- Kondisi awal: Mahasiswa telah *login*

- Kondisi akhir: Halaman prasyarat mata kuliah ditampilkan dan berisi mata kuliah yang dibuka pada semester terkini beserta status prasyaratnya
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa memilih menu prasyarat mata kuliah.	Sistem mendapatkan data mahasiswa kemudian menampilkan halaman prasyarat mata kuliah

- Eksepsi: Mahasiswa sedang menempuh semester 1

The screenshot shows the IFStudentPortal interface. On the left is a sidebar with navigation links: Home, Persiapan Perwalian, Jadwal Kuliah, Syarat Kuliah, Info & Lapor Usg, and Logout. The main content area has a title 'PERSIAPAN PERWALIAN SEMESTER GANJIL 2018/2019' and a sub-section 'DATA AKADEMIK'. Below this is a box containing academic statistics: IPS (GPNAP) 2017/2018: 4.00 dari 4 SKR, IP Kumulatif: 2.90, IP Kuliah: 2.90, IP II Terbaik: 2.90, Rata-Rata: 1.90, Nilai TOEFL: [54]. The main table lists courses with their prerequisites and status:

Kode Mata Kuliah	Nama Mata Kuliah	Keterangan
AII 101100	Dasar Pemrograman	memenuhi syarat ✓
AIF151101	Pemrograman untuk Komputasi	tidak memenuhi prasyarat ✗
AII 101102	Matematika Dasar	sudah lulus ✅
AIF151104	Logika Informatika	sudah lulus ✅
AII 101105	Pengantar Informatika	tidak memenuhi prasyarat ✗
AIF151106	Matris dan Ruang Vektor	tidak memenuhi prasyarat ✗
AII 101107	Matematika Diskret	sudah lulus ✅
AIF152007	Teknik Presentasi	tidak memenuhi prasyarat ✗
AIF152100	Analisis dan Desain Perangkat Lunak	sudah lulus ✅
AIF152101	Algoritma dan Struktur Data	tidak memenuhi prasyarat kelas AIF151100 ✗
AIF152103	Struktur Diskret	sudah lulus ✅
AIF152105	Pemrograman Berorientasi Objek	tidak memenuhi prasyarat kelas AIF151100 ✗
AIF152106	Desain dan Analisis Algoritma	tidak memenuhi prasyarat kelas AIF152101 ✗
AIF152109	Statistik untuk Komputasi	sudah lulus ✅
AIF152111	Pemrograman Kompetitif I	tidak memenuhi prasyarat kelas AIF152101 ✗
AIF152112	Pemrograman Kompetitif II	tidak memenuhi prasyarat kelas AIF152111 ✗

Gambar 3.3: Tampilan halaman daftar prasyarat mata kuliah

3. Jadwal kuliah yang tersusun Pada saat IFStudentPortal pertama kali dibangun, Portal Akademik Mahasiswa saat itu belum memiliki jadwal dengan tampilan grafik yang terurut berdasarkan hari dan jamnya. Fitur ini dibuat untuk mempermudah melihat susunan jadwal.

- Nama: Melihat jadwal kuliah
- Aktor: Mahasiswa
- Deskripsi: Melihat jadwal kuliah yang sudah tersusun dan terurut berdasarkan hari dan jam
- Kondisi awal: Mahasiswa telah *login*
- Kondisi akhir: Halaman jadwal ditampilkan dan berisi jadwal kuliah yang sudah tersusun dan terurut berdasarkan hari dan jam
- Eksepsi: Jadwal kuliah belum keluar
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa memilih menu jadwal.	Sistem menyusun dan mengurutkan jadwal mahasiswa berdasarkan hari kemudian menampilkan halaman jadwal

JADWAL KULIAH						
SEMESTER GANJIL 2020/2021						
Pukul	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
7.00	AIF183329	-	-	AIF182301	-	-
	AIF183329	-	-	AIF182301	-	-
8.00	AIF183329	-	-	AIF182301	-	-
	AIF183329	-	-	AIF182301	-	-
9.00	AIF184001	-	-	AIF184005	AIF183225	-
	AIF184001	-	-	AIF184005	AIF183225	-
10.00	-	-	-	AIF184005	AIF183225	-
	-	-	-	AIF184005	AIF183225	-
11.00	-	-	-	-	-	-
	-	-	-	-	-	-
12.00	-	-	-	-	-	-
	-	-	-	-	-	-
13.00	-	AIF181101	AIF183225	AIF181101	-	-
	-	AIF181101	AIF183225	AIF181101	-	-
14.00	-	AIF181101	AIF183225	AIF181101	-	-
	-	AIF181101	AIF183225	AIF181101	-	-
15.00	-	AIF181101	AIF183225	-	-	-
	-	AIF181101	AIF183225	-	-	-
16.00	-	-	-	-	-	-
	-	-	-	-	-	-
17.00	-	-	-	-	-	-

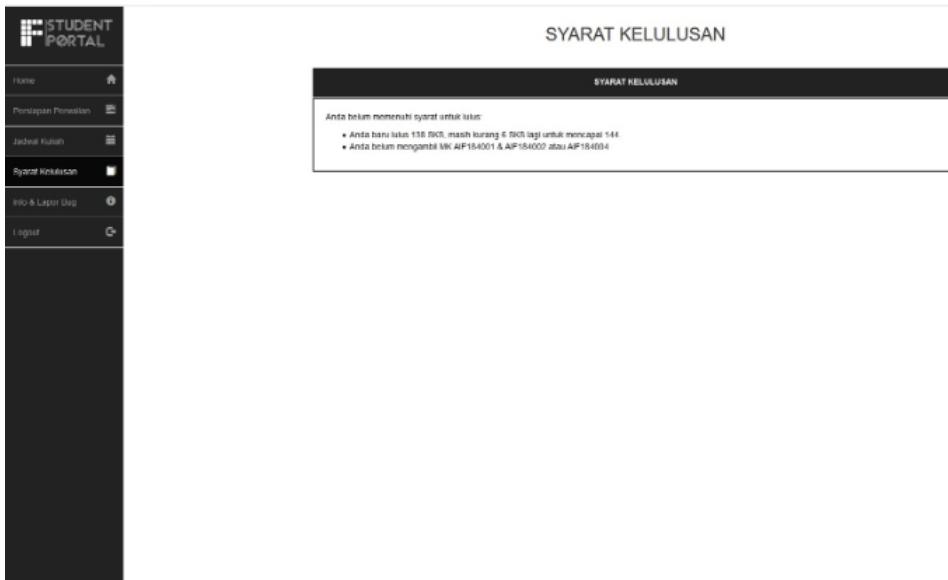
Gambar 3.4: Tampilan halaman jadwal kuliah yang tersusun

4. Melihat ringkasan data akademik Pada saat IFStudentPortal pertama kali dibangun, Portal Akademik Mahasiswa saat itu belum memberikan rincian data akademik yang detil. Fitur ini dibuat untuk melengkapi informasi data akademik tersebut.

5. Melihat Ringkasan Data Akademik

- Nama: Melihat ringkasan data akademik
- Aktor: Mahasiswa
- Deskripsi: melihat data mengenai mata kuliah apa saja yang sudah lulus beserta jenis mata kuliahnya(wajib atau pilihan), sisa SKS untuk mencapai kelulusan, dan mata kuliah wajib yang belum ditempuh. Mahasiswa juga dapat melihat IPS dan IPK yang lebih ter-update
- Kondisi awal: Mahasiswa telah *login*
- Kondisi akhir: Halaman yang berisi jadwal ringkasan data akademik ditampilkan
- Skenario utama:

No	Aksi Aktor	Reaksi Sistem
1	Mahasiswa memilih menu ringkasan data akademik.	Sistem meringkas data akademik mahasiswa kemudian menampilkan halaman ringkasan data akademik



Gambar 3.5: Tampilan halaman ringkasan data akademik

3.2 Analisis Sistem Usulan

IFStudentPortal akan dibangun untuk platform *Android*. Dokumentasi *Android* menyediakan panduan dalam membangun aplikasi agar aplikasi yang dibangun memiliki tampilan dan perilaku yang konsisten dengan platform *Android*. Panduan tersebut sudah dijelaskan di sub-bab ???. Pada sub-bab ini akan dijelaskan cara mencapai standar yang diberikan oleh panduan tersebut.

3.2.1 Pemenuhan Panduan *Material Design*

Panduan aksesibilitas bertujuan memberikan arahan agar aplikasi yang dibangun mudah untuk dipelajari dan digunakan. Meningkatkan aksesibilitas akan meningkatkan kebergunaan aplikasi kepada berbagai kalangan pengguna. Berikut akan dipaparkan langkah-langkah yang dilakukan untuk meningkatkan aksesibilitas aplikasi *Android* IFStudentPortal :

1. (HR-01) Hierarki :

- Warna latar belakang akan memiliki kontras yang cukup dengan warna teks.
- Ukuran teks akan menyesuaikan dengan ukuran layar sehingga tidak ada teks yang terlalu kecil sehingga sulit dibaca
- Pengisian *form* akan mengalir dari atas ke bawah, dan tombol untuk *submit form* tersebut akan berada di bawah form.
- Penyampaian informasi akan mengalir terurut dari atas ke bawah
- Tombol diletakan di tempat yang mudah dijangkau dengan memberikan area tekan yang besar atau meletakannya di dekat ujung layar.

2. (WK-01) Warna dan Kontras :

- Warna latar belakang akan memiliki kontras yang cukup dengan warna teks.
- Peringatan atau pesan *error* akan berwarna merah disertai dengan keterangan .

3. (TL-01) Tata Letak dan Tipografi :

- Menggunakan *layout* yang fleksibel dan responsif.

- Menggunakan format sp untuk satuan ukuran *font* agar ukuran teks menyesuaikan dengan ukuran layar.

4. (GB-01) Gambar :

- Melengkapi informasi yang disajikan dengan menampilkan foto profil mahasiswa.

5. (IA-01) Izin akses :

- Menampilkan kotak dialog dan tombol untuk meminta izin pengguna sebelum mengakses dan mengatur kalender.
- Hal lain di bagian izin akses [2.4.1](#) tidak diimplementasikan/digunakan.

3.2.2 Pemenuhan Panduan *Kualitas Aplikasi*

Aplikasi yang berkualitas tentu saja diinginkan oleh pengguna *Android*. Kualitas akan berpengaruh terhadap keberhasilan aplikasi dalam instalasi, ulasan, loyalitas, dan keterlibatan pengguna dalam jangka panjang. Berikut akan dipaparkan langkah-langkah yang dilakukan untuk memastikan aplikasi *Android* IFStudentPortal memiliki kualitas yang baik :

1. Desain visual dan interaksi Pengguna :

- (UX-B1) Aplikasi tidak akan menggunakan ikon yang ambigu sehingga pengguna tidak kebingungan dalam mengartikan fungsi tombol dengan ikon tersebut.
- (UX-S2) Bagian ini belum memerlukan perhatian khusus

2. Fungsionalitas :

- (FN-P1) Aplikasi akan meminta izin akses dari pengguna sebelum menggunakan kalender.
- (FN-L1) Bagian ini belum memerlukan perhatian khusus.
- (FN-A1) Aplikasi tidak menggunakan audio sebagai fitur utama.
- (FN-U1) Aplikasi tidak menggunakan orientasi *landscape*.
- (FN-S1) Aplikasi akan memberhentikan layanan saat pengguna menutup aplikasi sehingga aplikasi tidak aktif saat di belakang layar.
- (FN-S2) Aplikasi akan mempertahankan *state login* selama ada aktifitas atau sampai batas waktu yang ditentukan.

3. Kompatibilitas, Performa, dan Stabilitas :

- (PS-S1) Aplikasi akan dibangun dengan *library Android* versi terbaru yang sudah stabil untuk meminimalisir kemungkinan aplikasi macet. Aplikasi akan diuji sebelum dirilis untuk memperbaiki *bug* yang ditemukan.
- (PS-P1) Aplikasi tidak menggunakan algoritma yang kompleks dan melakukan proses yang berat sehingga tidak perlu waktu lama untuk memuat aplikasi.
- (PS-T1) Aplikasi dibangun dengan SDK terbaru dan diuji di perangkat terbaru dan di perangkat yang populer.
- (PS-M1) Aplikasi tidak memutar video dan audio sebagai fitur utama sehingga tidak memerlukan perhatian khusus.
- (PS-B1) Aplikasi sederhana sehingga bagian ini belum memerlukan perhatian khusus.
- (PS-V1) Grafik yang ditampilkan adalah gambar dengan resolusi tinggi

4. Keamanan :

- (SC-D1) Aplikasi tidak menyimpan data pengguna baik di internal maupun di log.
- (SC-D2) Aplikasi menggunakan data yang valid dan aman dari Portal Akademik Mahasiswa.
- (SC-P1) Aplikasi tidak mengekspor komponen aplikasi dengan aplikasi lain.
- (SC-P2) Aplikasi tidak berbagi konten dengan aplikasi lain.
- (SC-N1) Aplikasi membuka koneksi dengan Portal Akademik Mahasiswa yang sudah diamankan dengan SSL.
- (SC-N2) Bagian ini belum memerlukan perhatian khusus.
- (SC-U1) Aplikasi dibangun dengan *dependency*, *library*, dan *SDK* terbaru.
- (SC-E1) Aplikasi tidak mengeksekusi kode external secara dinamis.
- (SC-C1) Aplikasi tidak perlu mengenkripsi apapun.

5. *Google Play* :

- (GP-P1) Aplikasi tidak mengandung materi yang tidak pantas dan tidak menggunakan hak kekayaan intelektual atau merk orang lain.
- (GP-D1) Aplikasi akan dibangun dengan mengikuti panduan yang sudah diuraikan sebelum bagian ini.
- (GP-X1) Pengembang akan memperbaiki *bug* yang ditemukan.

3.2.3 Persiapan

Sebelum dan selama proses pengembangan aplikasi *Android IFStudentPortal* berjalan, ada beberapa pekerjaan yang dilakukan sebagai berikut :

- Perawatan situs IFStudentPortal
IFStudentPortal beberapa kali mengalami gangguan sehingga tidak bisa digunakan sebagaimana mestinya, sehingga dilakukan perbaikan-perbaikan yang diperlukan. Perubahan yang dilakukan adalah sebagai berikut :
 - Memperbaiki *bug* alamat foto profil dari Portal Akademik Mahasiswa².
 - Memperbaiki *bug* mengambil kode semester dari Portal Akademik Mahasiswa dengan memindahkan halaman sumber kode semester dari halaman frs ke halaman nilai³.
 - Merubah Mahasiswaan SIA Models dari *submodule* ke *Maven*⁴.
 - Memperbaiki *bug* akibat perubahan format tanggal di halaman nilai TOEFL Portal Akademik Mahasiswa⁵.
- Perawatan SIA Models
SIA Models versi v3.1.0 belum cocok untuk digunakan di platform *Android* sehingga perlu sedikit modifikasi agar bisa digunakan untuk pembangunan *Android IFStudentPortal*. Perubahan yang dilakukan adalah sebagai berikut :
 - Merubah nilai kembalian dari *method getPhotoImage* menjadi *byte*⁶.

²Kode dapat dilihat di <https://github.com/ftisunpar/IFStudentPortal/commit/2534764>

³Kode dapat dilihat di <https://github.com/ftisunpar/IFStudentPortal/commit/4775b17>

⁴Kode dapat dilihat di <https://github.com/ftisunpar/IFStudentPortal/commit/d88bba>

⁵Kode dapat dilihat di <https://github.com/ftisunpar/IFStudentPortal/commit/856fd77>

⁶Kode dapat dilihat di <https://www.github.com/pascalalfadian/siamodels/commit/0df064f>.

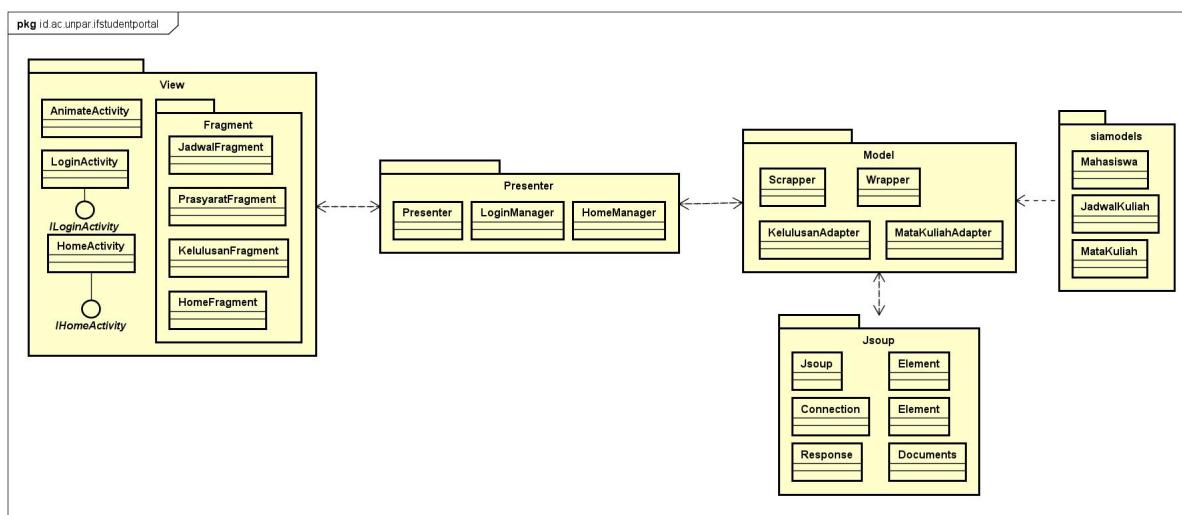
BAB 4

PERANCANGAN

Pada bab ini akan dijelaskan mengenai perancangan aplikasi yang akan dibangun meliputi diagram kelas rinci beserta deskripsi dan fungsinya, dan perancangan antarmuka.

4.1 Perancangan Kelas

Diagram kelas secara keseluruhan dapat dilihat di gambar 4.1. Terdapat 3 *package* dalam aplikasi IFStudentPortal ini yaitu *Model*, *View*, *Presenter*. *Package* IFStudentPortal ini terhubung dengan *library* SIA Models dan *library* Jsoup. Penjelasan mengenai *library* Jsoup sudah dijelaskan di bagian 2.2. Penjelasan mengenai SIA Models sudah dijelaskan di bagian 2.3.



Gambar 4.1: Diagram Kelas Keseluruhan

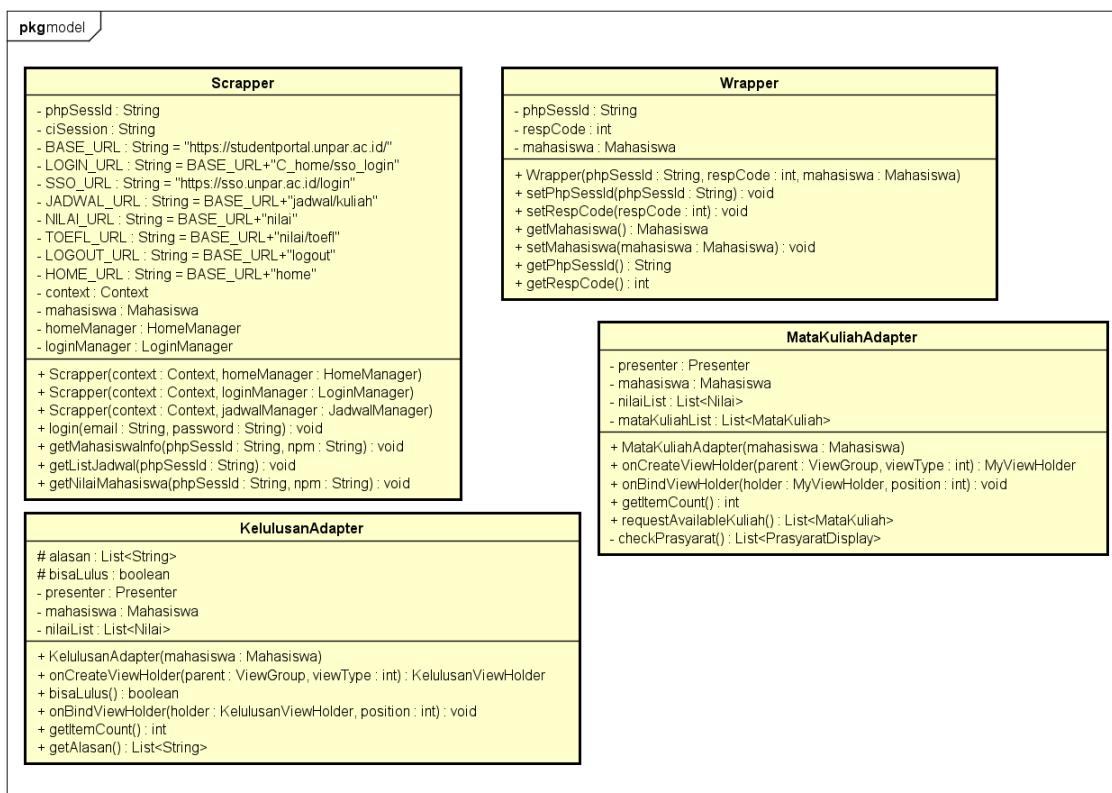
4.1.1 Model

Kelas-kelas dalam *package* ini merepresentasikan objek-objek yang ada dalam IFStudentPortal. Diagram kelas rinci untuk *package* *Model* dapat di lihat di gambar 4.2. Penjelasan mengenai kelas dan *method* yang ada di dalam *package* *Model* adalah sebagai berikut :

1. Scrapper

Kelas ini berfungsi untuk mengambil data dari Portal Akademik Mahasiswa untuk diolah dan ditampilkan. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **String phpSessId:** menyimpan *php session id* yang dibuat saat terjadi koneksi ke Portal Akademik Mahasiswa.



Gambar 4.2: Diagram Kelas Package Model

- **String ciSession:** menyimpan *Code Igniter* session yang dibuat saat terjadi koneksi ke Portal Akademik Mahasiswa.
- **Mahasiswa mahasiswa:** merepresentasikan mahasiswa yang akan diolah dan ditampilkan datanya.
- **Context context:** menyimpan *context* tujuan pengiriman data setelah diambil dan diolah dari Portal Akademik Mahasiswa.
- **HomeManager homeManager:** menyimpan objek yang mengatur halaman utama aplikasi.
- **LoginManager loginManager:** menyimpan objek yang mengatur halaman *login* aplikasi.
- **JadwalManager jadwalManager:** menyimpan objek yang mengatur halaman jadwal mahasiswa.
- **ProgressDialog dialog:** menyimpan objek yang mengatur *progress dialog*.
- **final String BASE_URL:** menyimpan *url* ke Portal Akademik Mahasiswa.
- **final String LOGIN_URL:** menyimpan *url* ke halaman *login* Portal Akademik Mahasiswa.
- **final String SSO_URL:** menyimpan *url* ke halaman login *Single Sign On* UNPAR.
- **final String JADWAL_URL:** menyimpan *url* ke halaman jadwal Portal Akademik Mahasiswa.
- **final String NILAI_URL:** menyimpan *url* ke halaman nilai Portal Akademik Mahasiswa.
- **final String TOEFL_URL:** menyimpan *url* ke halaman nilai TOEFL Portal Akademik Mahasiswa.
- **final String HOME_URL:** menyimpan *url* ke halaman utama Portal Akademik Mahasiswa.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public Scrapper(Context context, HomeManager homeManager)**
Parameter:

- **context:** *context activity* kelas Scrapper dibuat.
 - **homeManager:** objek homeManager.

Berfungsi sebagai *constructor* kelas Scrapper.

Kembalian: objek kelas scrapper untuk digunakan di halaman utama setelah pengguna *login*.

- **public Scrapper(Context context, LoginManager loginManager)**
Parameter:

- **context:** *context activity* kelas Scrapper dibuat.
 - **loginManager:** objek loginManager.

Berfungsi sebagai *constructor* kelas Scrapper.

Kembalian: objek kelas scrapper untuk digunakan di halaman *login* saat pengguna akan *login*.

- **public Scrapper(Context context, JadwalManager jadwalManager)**
Parameter:

- **context:** *context activity* kelas Scrapper dibuat.
 - **jadwalManager:** objek jadwalManager.

Berfungsi sebagai *constructor* kelas Scrapper.

Kembalian: objek kelas scrapper untuk digunakan di halaman jadwal mahasiswa.

- **public void login(String email, String password)**
Parameter:

- **email:** *email* mahasiswa yang dipakai *login*.
 - **password:** *password* mahasiswa yang dipakai *login*.

Berfungsi untuk *login* ke Portal Akademik Mahasiswa agar bisa mengambil data mahasiswa.

Kembalian: tidak ada.

- **public void getMahasiswaInfo(String phpSessId, String npm)**
Parameter:

- **phpSessId:** *php session id* dari koneksi ke Portal Akademik Mahasiswa.
 - **npm:** nomor pokok mahasiswa yang akan diambil datanya.

Berfungsi untuk mengambil data mahasiswa untuk ditampilkan di halaman utama.

Kembalian: tidak ada.

- **public void getListJadwal(String phpSessId, String npm)**
Parameter:

- **phpSessId:** *php session id* dari koneksi ke Portal Akademik Mahasiswa.
 - **npm:** nomor pokok mahasiswa yang akan diambil datanya.

Berfungsi untuk mengambil data jadwal mahasiswa untuk ditampilkan di halaman jadwal.

Kembalian: tidak ada.

- **public void getNilaiMahasiswa(String phpSessId, String npm)**
Parameter:

- **phpSessId:** *php session id* dari koneksi ke Portal Akademik Mahasiswa.
 - **npm:** nomor pokok mahasiswa yang akan diambil datanya.

Berfungsi untuk mengambil data nilai mahasiswa untuk ditampilkan di halaman jadwal.

Kembalian: tidak ada.

Terdapat 4 *private class* di dalam kelas Scrapper ini. Kelas ini merupakan perluasan dari kelas *AsyncTask*. Hal ini diperlukan karena untuk mengakses internet tidak bisa dilakukan di *thread* utama *Android*. Semua kelas ini memiliki *method-method* yang harus diimplementasi dari kelas *AsyncTask* yaitu **onPreExecute**, **doInBackground**, dan **onPostExecute**. Berikut penjelasan dari *method-method* yang ada di dalam beberapa *private class* di dalam kelas Scrapper.

(a) Login

Kelas ini digunakan untuk *login* ke Portal Akademik Mahasiswa untuk mengambil data. *Method onPreExecute* di kelas ini berfungsi menampilkan *progress dialog* dengan teks keterangan sedang mencoba *login*. *Method doInBackground* di kelas ini melakukan *login* ke Portal Akademik Mahasiswa dengan *email* dan kata sandi mahasiswa. *Method onPostExecute* di kelas ini berfungsi memindahkan aplikasi dari halaman *login* ke halaman utama setelah berhasil *login*.

(b) GetMahasiswaInfo

Kelas ini digunakan untuk mengambil data mahasiswa dari Portal Akademik Mahasiswa untuk diolah dan ditampilkan kemudian. Tidak ada *method onPreExecute* di kelas ini. *Method doInBackground* di kelas ini melakukan pengambilan data nama, npm, dan foto profil mahasiswa dari Portal Akademik Mahasiswa. *Method onPostExecute* di kelas ini berfungsi mengirimkan data ke kelas lain yang berfungsi menampilkan data tersebut di halaman utama.

(c) RequestJadwal

Kelas ini digunakan untuk mengambil data jadwal mahasiswa dari Portal Akademik Mahasiswa untuk diolah dan ditampilkan kemudian. Tidak ada *method onPreExecute* di kelas ini. *Method doInBackground* di kelas ini melakukan pengambilan data mahasiswa seperti nama mata kuliah, kode mata kuliah, hari dan jam kelas, dan nama dosen dan dari Portal Akademik Mahasiswa. *Method onPostExecute* di kelas ini berfungsi mengirimkan data ke kelas lain yang berfungsi menampilkan data tersebut di halaman jadwal.

(d) GetNilaiMahasiswa

Kelas ini digunakan untuk mengambil data riwayat nilai akademik termasuk nilai TOEFL mahasiswa dari Portal Akademik Mahasiswa untuk diolah dan ditampilkan di halaman persiapan perwalian dan syarat kelulusan. *Method onPreExecute* di kelas ini berfungsi menampilkan *progress dialog* dengan teks keterangan sedang mengambil data nilai mahasiswa. *Method doInBackground* di kelas ini mengambil data nilai mahasiswa per semester dari Portal Akademik Mahasiswa. *Method onPostExecute* di kelas ini berfungsi mengirimkan data mahasiswa yang sudah diambil ke kelas Adapter untuk diolah dan kemudian ditampilkan.

2. KelulusanAdapter

Kelas ini berfungsi untuk mengolah data yang sudah diambil dari Portal Akademik Mahasiswa untuk ditampilkan di halaman syarat kelulusan. Kelas ini merupakan ekstensi dari kelas *RecyclerView.Adapter*. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **List<String> alasan:** menyimpan daftar syarat kelulusan yang belum dipenuhi.
- **List<Mahasiswa.Nilai> nilaiList:** menyimpan daftar nilai mahasiswa.
- **Mahasiswa mahasiswa:** merepresentasikan mahasiswa yang akan diolah dan ditampilkan datanya.

- **Presenter presenter:** menyimpan objek presenter yang mengirim data mahasiswa ke kelas ini.
- **boolean bisaLulus:** merepresentasikan apakah mahasiswa sudah memenuhi semua syarat kelulusan.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public KelulusanAdapter(Mahasiswa mahasiswa)**
Parameter:
 - **mahasiswa:** objek mahasiswa yang datanya akan diolah.
 Berfungsi sebagai *constructor* kelas KelulusanAdapter.
Kembalian: tidak ada.
- **public KelulusanViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)**
Parameter:
 - **parent:** parent dari ViewGroup
 - **viewType:** parameter ini tidak dipakai.
 Berfungsi untuk memasangkan ViewHolder ke layout.
Kembalian: objek ViewHolder.
- **public boolean bisaLulus()**
 Berfungsi untuk mengembalikan atribut bisaLulus.
Kembalian: boolean bisaLulus.
- **public void onBindViewHolder(@NonNull KelulusanViewHolder holder, int position)**
Parameter:
 - **holder:** objek ViewHolder yang akan diolah dan ditampilkan sekarang.
 - **position:** menunjukkan posisi tampilan holder.
 Berfungsi untuk menulis syarat kelulusan yang belum dipenuhi ke holder.
Kembalian: tidak ada.
- **public int getItemCount()**
 Berfungsi mengembalikan ukuran List<String> alasan.
Kembalian: ukuran List<String> alasan.
- **public List<String> getAlasan()**
 Berfungsi mengembalikan List<String> alasan.
Kembalian: List<String> alasan.

Terdapat *private class* KelulusanViewHolder yang merupakan ekstensi dari kelas RecyclerView.ViewHolder di dalam kelas KelulusanAdapter ini. Kelas KelulusanViewHolder berfungsi merepresentasikan 1 baris alasan belum bisa lulus yang ada di halaman syarat kelulusan.

3. MataKuliahAdapter

Kelas ini berfungsi untuk mengolah data yang sudah diambil dari Portal Akademik Mahasiswa untuk ditampilkan di halaman persiapan perwalian. Kelas ini merupakan ekstensi dari kelas RecyclerView.Adapter. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **List<PrasyaratDisplay> itemDisplayList:** menyimpan daftar mata kuliah dan statusnya.
- **List<Mahasiswa.Nilai> nilaiList:** menyimpan daftar nilai mahasiswa.
- **List<MataKuliah> mkList:** menyimpan daftar mata kuliah yang tersedia.

- **Mahasiswa mahasiswa:** merepresentasikan mahasiswa yang akan diolah dan ditampilkan datanya.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public MataKuliahAdapter(Mahasiswa mahasiswa)**
Parameter:
 - **mahasiswa:** objek mahasiswa yang datanya akan diolah.

Berfungsi sebagai *constructor* kelas MataKuliahAdapter.
Kembalian: tidak ada.
- **public MataKuliahAdapter.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType)**
Parameter:
 - **parent:** parent dari ViewGroup
 - **viewType:** parameter ini tidak dipakai.

Berfungsi untuk memasangkan ViewHolder ke layout.
Kembalian: objek ViewHolder.
- **public void onBindViewHolder(@NonNull MataKuliahAdapter.MyViewHolder holder, int position)**
Parameter:
 - **holder:** objek ViewHolder yang akan diolah dan ditampilkan sekarang.
 - **position:** menunjukan posisi tampilan holder.

Berfungsi untuk menulis syarat kelulusan yang belum dipenuhi ke holder.
Kembalian: tidak ada.
- **public int getItemCount()**
 Berfungsi mengembalikan ukuran List<PrasyaratDisplay> itemDisplayList.
Kembalian: ukuran List<PrasyaratDisplay> itemDisplayList.
- **public List<MataKuliah> requestAvailableKuliah()**
 Berfungsi membuat daftar mata kuliah yang tersedia.
Kembalian: daftar mata kuliah yang tersedia.
- **public List<PrasyaratDisplay> checkPrasyarat()**
 Berfungsi memeriksa status untuk semua mata kuliah yang tersedia.
Kembalian: salah 1 dari status (sudah lulus, memenuhi syarat, tidak memiliki syarat, tidak memenuhi prasyarat).

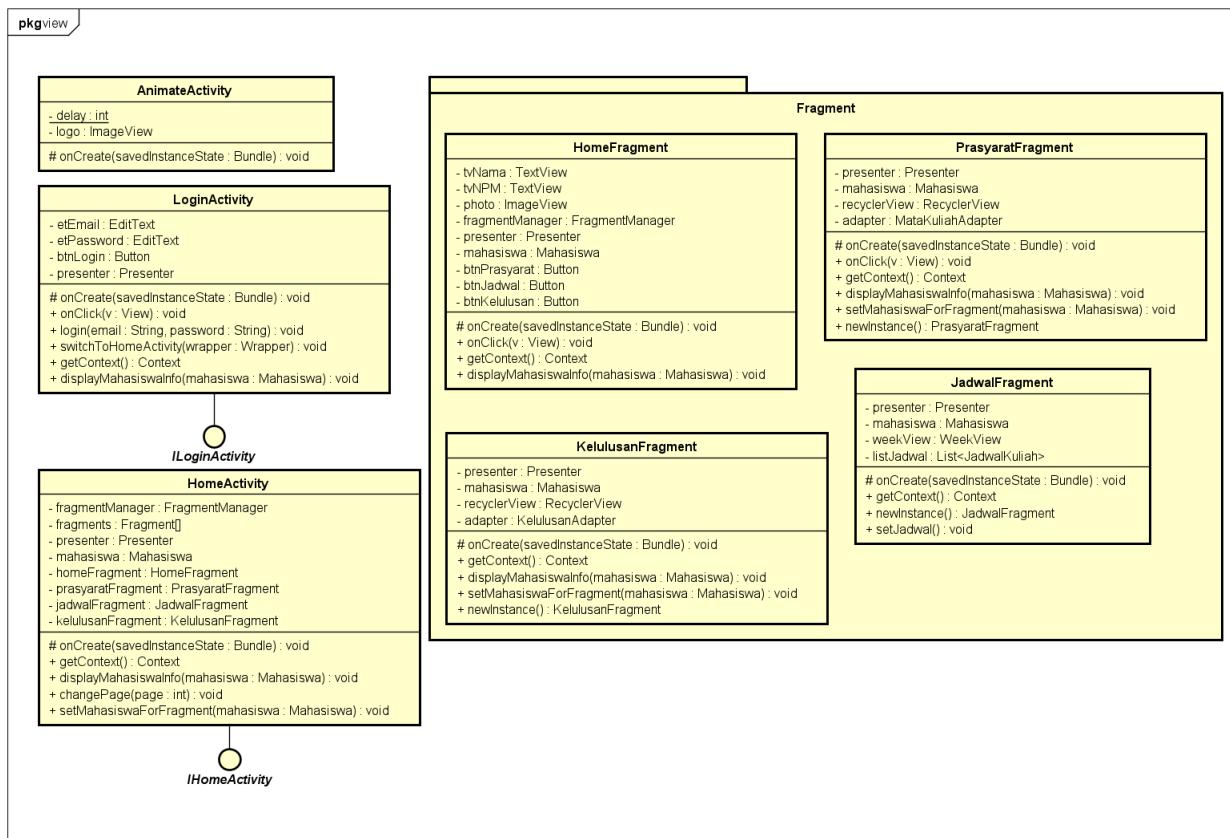
Terdapat *private class* MyViewHolder yang merupakan ekstensi dari kelas RecyclerView.ViewHolder di dalam kelas MataKuliahAdapter ini. Kelas MyViewHolder berfungsi merepresentasikan 1 baris data mata kuliah dan statusnya yang ada di halaman persiapan perwalian. Terdapat juga *private class* PrasyaratDisplay yang berfungsi menyimpan mata kuliah dan statusnya.

4. Wrapper

Kelas ini digunakan untuk membungkus objek mahasiswa, phpSessId, dan respCode saat terjadi perpindahan data dari *activity login* ke *activity home*. Method yang ada di kelas ini hanya *setter* dan *getter* dari masing-masing atribut yang ada.

4.1.2 View

Kelas-kelas dalam *package* ini merepresentasikan tampilan antarmuka yang ada di aplikasi IFS-StudentPortal. Diagram kelas rinci untuk *package* View dapat di lihat di gambar 4.3. Penjelasan mengenai kelas dan *method* yang ada di dalam *package* View adalah sebagai berikut :



Gambar 4.3: Diagram Kelas Package View

1. ILoginActivity

Kelas ini adalah kelas abstrak sebagai *interface* untuk kelas *LoginActivity*. Kelas ini tidak memiliki atribut, namun memiliki beberapa *method* yang harus diimplementasi oleh kelas yang mengimplementasi *interface* ini. *Method* yang dimiliki kelas ini sebagai berikut :

- **public void login(String email, String password)**

Parameter:

- **email:** *email* mahasiswa yang dipakai *login*.
- **password:** *password* mahasiswa yang dipakai *login*.

Berfungsi untuk *login* ke Portal Akademik Mahasiswa agar bisa mengambil data mahasiswa.

Kembalian: tidak ada.

- **public void switchToHomeActivity(Wrapper wrapper)**

Parameter:

- **wrapper:** objek yang berfungsi membungkus objek dan data agar mudah dikirim objek *Intent* ke *activity Android* utama.

Berfungsi untuk memindahkan aplikasi ke halaman utama setelah pengguna berhasil *login*.

Kembalian: tidak ada.

- **public Context getContext()**

Berfungsi untuk mengembalikan *context activity Android* ini.

Kembalian: *context*.

2. LoginActivity

Kelas ini merepresentasikan halaman *login* dari IFStudentPortal. Kelas ini merupakan ekstensi dari kelas *AppCompatActivity* dan mengimplementasi *interface iLoginActivity* dan *View.OnClickListener*. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **EditText etPassword:** menyimpan tempat menulis kata sandi mahasiswa di halaman *login*.
- **EditText etEmail:** menyimpan tempat menulis email mahasiswa di halaman *login*.
- **AppCompatCheckBox checkbox:** menyimpan *checkbox* untuk menampilkan atau menyembunyikan kata sandi mahasiswa di halaman *login*.
- **MaterialButton btnLogin:** menyimpan tombol untuk melakukan *login*.
- **Presenter presenter:** menyimpan objek presenter yang mengatur alur data dalam aplikasi.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public void onCreate(Bundle savedInstanceState)**

Parameter:

- **savedInstanceState:** menyimpan informasi status instansi dari *activity Android*.

Berfungsi menginisiasi elemen *Android* dasar dan atribut-atribut kelas.

Kembalian: tidak ada.

- **public void onClick(View v)**

Parameter:

- **v:** menyimpan elemen tampilan yang fungsinya saat ditekan akan diatur di *method* ini.

Berfungsi mengatur fungsi elemen tampilan saat ditekan. **Kembalian:** tidak ada.

- **public void login(String email, String password)**

Parameter:

- **email:** *email* mahasiswa yang dipakai *login*.

- **password:** *password* mahasiswa yang dipakai *login*.

Berfungsi untuk *login* ke Portal Akademik Mahasiswa agar bisa mengambil data mahasiswa.

Kembalian: tidak ada.

- **public void switchToHomeActivity(Wrapper wrapper)**

Parameter:

- **wrapper:** objek yang berfungsi membungkus objek dan data agar mudah dikirim objek *Intent* ke *activity Android* utama.

Berfungsi untuk memindahkan aplikasi ke halaman utama setelah pengguna berhasil *login*.

Kembalian: tidak ada.

- **public Context getContext()**

Berfungsi untuk mengembalikan *context activity Android* ini.

Kembalian: *context*.

3. IHomeActivity

Kelas ini adalah kelas abstrak sebagai *interface* untuk kelas *HomeActivity*. Kelas ini tidak memiliki atribut, namun memiliki beberapa *method* yang harus diimplementasi oleh kelas yang mengimplementasi *interface* ini. *Method* yang dimiliki kelas ini sebagai berikut :

- **void displayMahasiswaInfo(Mahasiswa mahasiswa)**
Parameter:
 - **mahasiswa:** menyimpan objek mahasiswa pengguna aplikasi.Berfungsi menampilkan data-data mahasiswa di halaman utama.
Kembalian: tidak ada.
- **void changePage(int page)**
Parameter:
 - **page:** nomor halaman *fragment* yang dituju.Berfungsi merubah tampilan ke halaman *fragment* yang dituju. **Kembalian:** tidak ada.
- **Context getContext()**
Berfungsi untuk mengembalikan *context activity Android* ini.
Kembalian: *context*.
- **void setMahasiswaForFragment(Mahasiswa mahasiswa);**
Berfungsi mengirimkan objek mahasiswa ke *Fragment*.
Kembalian: tidak ada.

4. HomeActivity

Kelas ini merepresentasikan halaman utama dari IFStudentPortal. Kelas ini merupakan ekstensi dari kelas *AppCompatActivity* dan mengimplementasi *interface IHomeActivity*. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **FragmentManager fm:** menyimpan *fragment manager* untuk pergantian tampilan halaman.
- **Fragment[] fragments:** menyimpan kumpulan *fragment* halaman jadwal, persiapan perwalian, dan syarat kelulusan.
- **Presenter presenter:** menyimpan objek presenter yang mengatur alur data dalam aplikasi.
- **HomeFragment homeFragment:** menyimpan objek *fragment* untuk tampilan halaman utama.
- **JadwalFragment jadwalFragment:** menyimpan objek *fragment* untuk tampilan halaman jadwal.
- **PrasyaratFragment prasyaratFragment:** menyimpan objek *fragment* untuk tampilan halaman persiapan perwalian.
- **KelulusanFragment kelulusanFragment:** menyimpan objek *fragment* untuk tampilan halaman kelulusan.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **protected void onCreate(Bundle savedInstanceState)**
Parameter:
 - **savedInstanceState:** menyimpan informasi status instansi dari *activity Android*.Berfungsi menginisiasi *activity home*.
Kembalian: tidak ada.
- **public void displayMahasiswaInfo(Mahasiswa mahasiswa)**
Parameter:
 - **mahasiswa:** menyimpan objek mahasiswa pengguna aplikasi.Berfungsi menampilkan data-data mahasiswa di halaman utama.
Kembalian: tidak ada.

- **public void changePage(int page)**

Parameter:

- **page:** nomor halaman *fragment* yang dituju.

Berfungsi merubah tampilan ke halaman *fragment* yang dituju. **Kembalian:** tidak ada.

- **public Context getContext()**

Berfungsi untuk mengembalikan *context activity Android* ini.

Kembalian: *context*.

- **public setMahasiswaForFragment(Mahasiswa mahasiswa)**

Parameter:

- **mahasiswa:** menyimpan objek mahasiswa yang akan dikirim ke *fragment*.

Berfungsi untuk mengembalikan *context activity Android* ini.

Kembalian: tidak ada.

5. AnimateActivity

Kelas ini merepresentasikan halaman *splash screen* dari IFStudentPortal. Halaman *splash screen* ini ditampilkan saat aplikasi baru dibuka. Kelas ini merupakan ekstensi dari kelas *AppCompatActivity*. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **ImageView logo:** menyimpan gambar yang ditampilkan di halaman *splash screen*.
- **final static int delay = 2000:** durasi penampilan animasi (dalam milidetik).

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public void onCreate(Bundle savedInstanceState)**

Parameter:

- **savedInstanceState:** menyimpan informasi status instansi dari *activity Android*.

Berfungsi menginisiasi elemen *Android* dasar dan atribut-atribut kelas.

Kembalian: tidak ada.

6. HomeFragment

Kelas ini menyimpan tampilan halaman utama dari IFStudentPortal. Kelas ini merupakan ekstensi dari kelas *Fragment* dan mengimplementasi *interface IHomeActivity* dan *View.OnClickListener*. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **TextView tvNama:** menyimpan teks nama mahasiswa.
- **TextView tvNPM:** menyimpan teks nomor pokok mahasiswa.
- **CircularImageView photo:** menyimpan foto profil mahasiswa.
- **Button btnJadwal:** menyimpan tombol untuk menuju halaman jadwal.
- **Button btnPrasyarat:** menyimpan tombol untuk menuju halaman persiapan perwalian.
- **Button btnKelulusan:** menyimpan tombol untuk menuju halaman syarat kelulusan.
- **Button btnLogout:** menyimpan tombol untuk melakukan *logout*.
- **Mahasiswa mahasiswa:** menyimpan objek mahasiswa.
- **Presenter presenter:** menyimpan objek presenter.
- **String phpSessId:** menyimpan php session id yang digunakan untuk berkomunikasi dengan Portal Akademik Mahasiswa.
- **IHomeActivity ui:** menyimpan *context* dari *HomeActivity*.
- **FragmentManager fm:** menyimpan objek *fragment manager*.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public void onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)**
Parameter:
 - **inflater:** menyimpan objek yang menghubungkan kelas dengan layout.
 - **container:** menyimpan objek yang menghubungkan kelas dengan layout.
 - **savedInstanceState:** menyimpan informasi status instansi dari *activity Android*.Berfungsi menginisiasi *fragment*, elemen *Android* dasar, dan atribut-atribut kelas.
Kembalian: tidak ada.
- **public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)**
Parameter:
 - **view:** objek view tampilan halaman utama.
 - **savedInstanceState:** menyimpan informasi status instansi dari *activity Android*.Berfungsi mengisi data mahasiswa saat halaman utama dimuat.
Kembalian: tidak ada.
- **public static HomeFragment newInstance()**
Dipanggil saat objek dari kelas HomeFragment pertama kali dibuat.
Kembalian: HomeFragment.
- **public void changePage(int page)**
Parameter:
 - **page:** nomor halaman yang dituju.Berfungsi memanggil method **changePage(int page)** yang ada di kelas HomeActivity.
Kembalian: tidak ada.
- **public void displayMahasiswaInfo(Mahasiswa mahasiswa)**
Parameter:
 - **mahasiswa:** objek mahasiswa.Berfungsi menampilkan nama dan nomor pokok mahasiswa di *TextView*.
Kembalian: tidak ada.
- **public void onClick(View view)**
Parameter:
 - **view:** objek tombol yang ditekan.Berfungsi mengatur fungsi semua tombol yang ada di kelas ini.
Kembalian: tidak ada.
- **public void setMahasiswa(Mahasiswa mahasiswa)**
Parameter:
 - **mahasiswa:** objek mahasiswa.Berfungsi menampilkan dan menyimpan informasi data diri mahasiswa.
Kembalian: tidak ada.
- **Bundle saveState()**
Berfungsi menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.
Kembalian: *bundle* berisi data mahasiswa.
- **Bundle onDestoryView()**
Berfungsi memanggil method **saveState()** untuk menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.
Kembalian: tidak ada.

- **Bundle onSaveInstanceState(Bundle outState)**

Parameter:

- **outState:** *bundle* tempat menyimpan data mahasiswa.

Berfungsi menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.

Kembalian: *bundle* berisi data mahasiswa.

7. JadwalFragment

Kelas ini menyimpan tampilan halaman jadwal dari IFStudentPortal. Kelas ini merupakan ekstensi dari kelas *Fragment* dan mengimplementasi *interface IHomeActivity* dan *View.OnClickListener*. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **WeekView weekView:** menyimpan elemen jadwal.
- **Mahasiswa mahasiswa:** menyimpan objek mahasiswa.
- **Presenter presenter:** menyimpan objek presenter.
- **List<JadwalKuliah> listJadwal:** menyimpan daftar jadwal kuliah mahasiswa.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public void onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)**

Parameter:

- **inflater:** menyimpan objek yang menghubungkan kelas dengan layout.
- **container:** menyimpan objek yang menghubungkan kelas dengan layout.
- **savedInstanceState:** menyimpan informasi status instansi dari *activity Android*.

Berfungsi menginisiasi *fragment*, elemen *Android* dasar, dan atribut-atribut kelas.

Kembalian: tidak ada.

- **public static JadwalFragment newInstance()**

Dipanggil saat objek dari kelas JadwalFragment pertama kali dibuat.

Kembalian: JadwalFragment.

- **public void onClick(View view)**

Parameter:

- **view:** objek tombol yang ditekan.

Berfungsi mengatur fungsi semua tombol yang ada di kelas ini.

Kembalian: tidak ada.

- **public void setJadwal()**

Berfungsi mengolah dan menampilkan data jadwal kuliah mahasiswa.

Kembalian: tidak ada.

- **public int getColor(String kode)**

Parameter:

- **kode:** kode mata kuliah.

Berfungsi memberikan warna ke jadwal sesuai kode mata kuliah.

Kembalian: kode warna dalam bentuk *Integer*.

- **public void setPresenter(Presenter presenter)**

Parameter:

- **presenter:** objek presenter.

Berfungsi menyimpan presenter ke atribut kelas.

Kembalian: tidak ada.

8. PrasyaratFragment

Kelas ini menyimpan tampilan halaman persiapan perwalian dari IFStudentPortal. Kelas ini merupakan ekstensi dari kelas *Fragment* dan mengimplementasi *interface IHomeActivity*. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **RecyclerView recyclerView:** menyimpan elemen daftar mata kuliah.
- **Mahasiswa mahasiswa:** menyimpan objek mahasiswa.
- **Presenter presenter:** menyimpan objek presenter.
- **MataKuliahAdapter mataKuliahAdapter:** menyimpan adapter untuk *recycler view*.
- **IHomeActivity ui:** menyimpan *context* dari *HomeActivity*.
- **String phpSessId:** menyimpan *php session id* yang digunakan untuk berkomunikasi dengan Portal Akademik Mahasiswa.
- **Bundle savedInstanceState:** menyimpan data mahasiswa agar tidak hilang saat berganti halaman.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public void onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)**
Parameter:
 - **inflater:** menyimpan objek yang menghubungkan kelas dengan layout.
 - **container:** menyimpan objek yang menghubungkan kelas dengan layout.
 - **savedInstanceState:** menyimpan informasi status instansi dari *activity Android*.Berfungsi menginisiasi *fragment*, elemen *Android* dasar, dan atribut-atribut kelas.
Kembalian: tidak ada.
- **public static PrasyaratFragment newInstance()**
Dipanggil saat objek dari kelas PrasyaratFragment pertama kali dibuat.
Kembalian: PrasyaratFragment.
- **public void changePage(int page)**
Parameter:
 - **page:** nomor halaman *fragment* yang dituju.Berfungsi memanggil method **changePage(int page)** yang ada di kelas HomeActivity.
Kembalian: tidak ada.
- **Bundle saveState()**
Berfungsi menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.
Kembalian: *bundle* berisi data mahasiswa.
- **Bundle onDestoryView()**
Berfungsi memanggil method **saveState()** untuk menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.
Kembalian: tidak ada.
- **Bundle onSaveInstanceState(Bundle outState)**
Parameter:
 - **outState:** *bundle* tempat menyimpan data mahasiswa.Berfungsi menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.
Kembalian: *bundle* berisi data mahasiswa.

- **public void setMahasiswaForFragment(Mahasiswa mahasiswa)**

Parameter:

- **mahasiswa:** objek mahasiswa yang diterima dari *home activity*.

Berfungsi menyimpan dan mengirimkan informasi data mata kuliah mahasiswa ke *adapter recycler view*.

Kembalian: tidak ada.

- **public void setPresenter(Presenter presenter)**

Parameter:

- **presenter:** objek presenter.

Berfungsi menyimpan presenter ke atribut kelas.

Kembalian: tidak ada.

9. KelulusanFragment

Kelas ini menyimpan tampilan halaman kelulusan dari IFStudentPortal. Kelas ini merupakan ekstensi dari kelas *Fragment* dan mengimplementasi *interface IHomeActivity*. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **RecyclerView recyclerView:** menyimpan elemen daftar syarat kelulusan.
- **TextView tvStatus:** menyimpan elemen tempat menyimpan teks status mata kuliah.
- **Mahasiswa mahasiswa:** menyimpan objek mahasiswa.
- **Presenter presenter:** menyimpan objek presenter.
- **KelulusanAdapter kelulusanAdapter:** menyimpan adapter untuk *recycler view*.
- **IHomeActivity ui:** menyimpan *context* dari *HomeActivity*.
- **String phpSessId:** menyimpan *php session id* yang digunakan untuk berkomunikasi dengan Portal Akademik Mahasiswa.
- **Bundle savedInstanceState:** menyimpan data mahasiswa agar tidak hilang saat berganti halaman.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public void onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)**

Parameter:

- **inflater:** menyimpan objek yang menghubungkan kelas dengan layout.
- **container:** menyimpan objek yang menghubungkan kelas dengan layout.
- **savedInstanceState:** menyimpan informasi status instansi dari *activity Android*.

Berfungsi menginisiasi *fragment*, elemen *Android* dasar, dan atribut-atribut kelas.

Kembalian: tidak ada.

- **public static KelulusanFragment newInstance()**

Dipanggil saat objek dari kelas KelulusanFragment pertama kali dibuat.

Kembalian: KelulusanFragment.

- **public void changePage(int page)**

Parameter:

- **page:** nomor halaman *fragment* yang dituju.

Berfungsi memanggil method **changePage(int page)** yang ada di kelas HomeActivity.

Kembalian: tidak ada.

- **Bundle saveState()**

Berfungsi menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.

Kembalian: *bundle* berisi data mahasiswa.

- **Bundle onDestroyView()**

Berfungsi memanggil method saveState() untuk menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.

Kembalian: tidak ada.

- **Bundle onSaveInstanceState(Bundle outState)**

Parameter:

- **outState:** *bundle* tempat menyimpan data mahasiswa.

Berfungsi menyimpan informasi data diri mahasiswa agar tidak hilang saat berganti halaman.

Kembalian: *bundle* berisi data mahasiswa.

- **public void setMahasiswaForFragment(Mahasiswa mahasiswa)**

Parameter:

- **mahasiswa:** objek mahasiswa yang diterima dari *home activity*.

Berfungsi menyimpan dan mengirimkan informasi data mata kuliah mahasiswa ke *adapter recycler view*.

Kembalian: tidak ada.

- **public void setPresenter(Presenter presenter)**

Parameter:

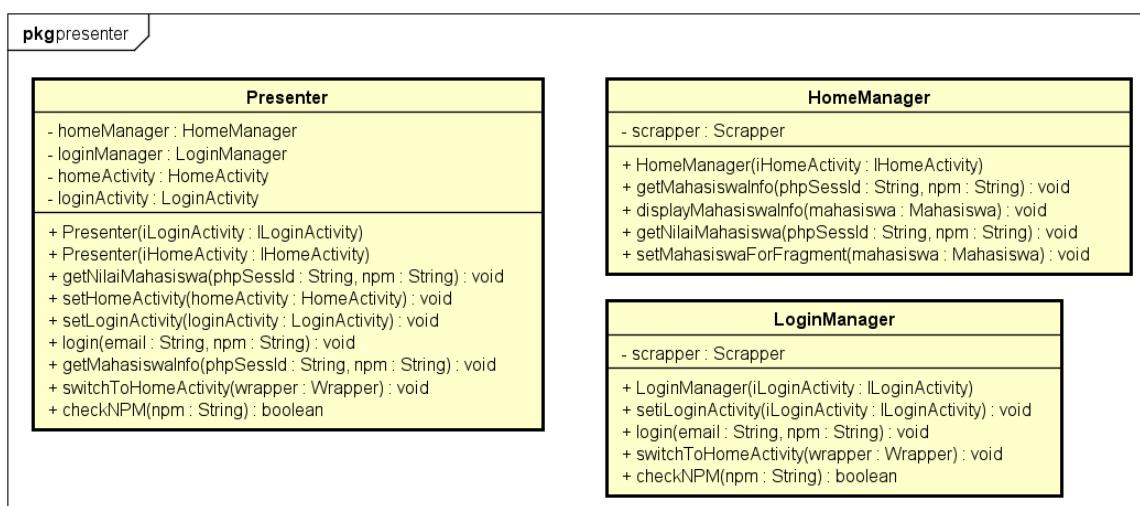
- **presenter:** objek presenter.

Berfungsi menyimpan presenter ke atribut kelas.

Kembalian: tidak ada.

4.1.3 Presenter

Kelas-kelas dalam *package* ini merepresentasikan tampilan antarmuka yang ada di aplikasi IFStudentPortal. Diagram kelas rinci untuk *package* *Presenter* dapat di lihat di gambar 4.4. Penjelasan mengenai kelas dan *method* yang ada di dalam *package* *Presenter* adalah sebagai berikut :



Gambar 4.4: Diagram Kelas Package Presenter

1. Presenter

Kelas ini merepresentasikan presenter yang berfungsi menjadi pengantara transaksi data antara model dan view. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **HomeManager homeManager:** menyimpan objek yang mengatur transaksi data ke halaman utama.
- **HomeActivity homeActivity:** menyimpan objek yang merepresentasikan halaman utama.
- **LoginManager loginManager:** menyimpan objek yang mengatur transaksi data ke halaman *login*.
- **LoginActivity loginActivity:** menyimpan objek yang merepresentasikan halaman *login*.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public Presenter(ILoginActivity iLoginActivity)**

Parameter:

- **iLoginActivity:** digunakan untuk inisiasi objek loginManager.

Berfungsi sebagai constructor kelas saat digunakan di halaman *login*.

Kembalian: tidak ada.

- **public Presenter(IHomeActivity iHomeActivity)**

Parameter:

- **iHomeActivity:** digunakan untuk inisiasi objek homeManager.

Berfungsi sebagai constructor kelas saat digunakan di halaman utama.

Kembalian: tidak ada.

- **public void login(String email, String password)**

Parameter:

- **email:** *email* mahasiswa yang dipakai *login*.

- **password:** *password* mahasiswa yang dipakai *login*.

Berfungsi memanggil *method* *login* milik objek loginManager.

Kembalian: tidak ada.

- **public void switchToHomeActivity(Wrapper wrapper)**

Parameter:

- **wrapper:** objek yang berfungsi membungkus objek dan data agar mudah dikirim objek *Intent* ke *activity Android* utama.

Berfungsi untuk memanggil *method* *switchToHomeActivity* milik objek loginManager.

Kembalian: tidak ada.

- **public boolean checkNPM(String npm)**

Parameter:

- **npm:** nomor pokok mahasiswa yang akan diperiksa validitasnya.

Berfungsi untuk memanggil *method* *checkNPM* milik loginManager.

Kembalian: boolean yang menunjukan validitas npm.

2. LoginManager

Kelas ini merepresentasikan pengatur *login* yang berfungsi menjadi pengantara transaksi data antara model dan view. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **ILoginActivity iLoginActivity:** menyimpan objek iLoginActivity.

- **Scrapper scrapper:** menyimpan objek scrapper.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public LoginManager(ILoginActivity iLoginActivity)**

Parameter:

- **iLoginActivity:** digunakan untuk inisiasi objek scrapper.

Berfungsi sebagai constructor kelas saat digunakan di halaman *login*.

Kembalian: tidak ada.

- **public void login(String email, String password)**

Parameter:

- **email:** *email* mahasiswa yang dipakai *login*.

- **password:** *password* mahasiswa yang dipakai *login*.

Berfungsi memanggil *method* *login* milik objek scrapper.

Kembalian: tidak ada.

- **public void switchToHomeActivity(Wrapper wrapper)**

Parameter:

- **wrapper:** objek yang berfungsi membungkus objek dan data agar mudah dikirim objek *Intent* ke *activity Android* utama.

Berfungsi untuk memanggil method *switchToHomeActivity* milik *iLoginActivity*. **Kembalian:** tidak ada.

- **public boolean checkNPM(String npm)**

Parameter:

- **npm:** nomor pokok mahasiswa yang akan diperiksa validitasnya.

Berfungsi untuk memeriksa validitas nomor pokok mahasiswa.

Kembalian: boolean yang menunjukkan validitas npm.

3. HomeManager

Kelas ini merepresentasikan pengatur halaman utama yang berfungsi menjadi pengantara transaksi data antara model dan view. Atribut yang dimiliki kelas ini adalah sebagai berikut :

- **IHomeActivity iHomeActivity:** menyimpan objek *iHomeActivity*.

- **Scrapper scrapper:** menyimpan objek scrapper.

Method yang dimiliki kelas ini adalah sebagai berikut :

- **public HomeManager(IHomeActivity iHomeActivity)**

Parameter:

- **iHomeActivity:** digunakan untuk inisiasi objek scrapper.

Berfungsi sebagai constructor kelas saat digunakan di halaman utama.

Kembalian: tidak ada.

- **public void getMahasiswaInfo(String phpSessId, String npm)**

Parameter:

- **phpSessId:** php session id yang dipakai untuk koneksi ke Portal Akademik Mahasiswa.

- **npm:** nomor pokok mahasiswa yang dipakai *login*.

Berfungsi memanggil *method* *getMahasiswaInfo* milik objek scrapper.

Kembalian: tidak ada.

- **public void displayMahasiswaInfo(Mahasiswa mahasiswa)**

Parameter:

- **mahasiswa:** objek yang berfungsi menyimpan mahasiswa.

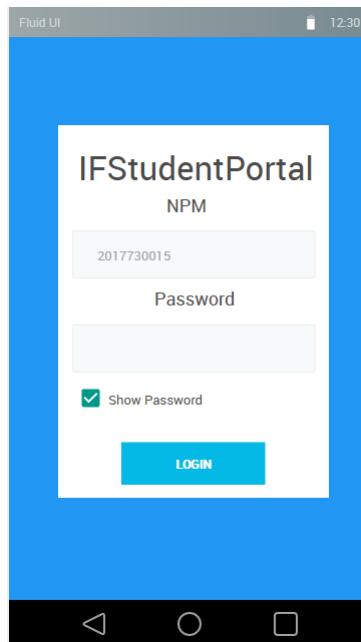
Berfungsi untuk memanggil *method displayMahasiswaInfo* milik iHomeActivity. **Kembalian:** tidak ada.

4.2 Perancangan Antarmuka

Pengguna akan dapat berinteraksi dengan aplikasi melalui antarmuka. Terdapat 5 antarmuka yang akan dirancang yaitu halaman *login*, halaman utama, halaman persiapan perwalian, halaman jadwal, dan halaman syarat kelulusan.

1. Antarmuka Halaman *Login*

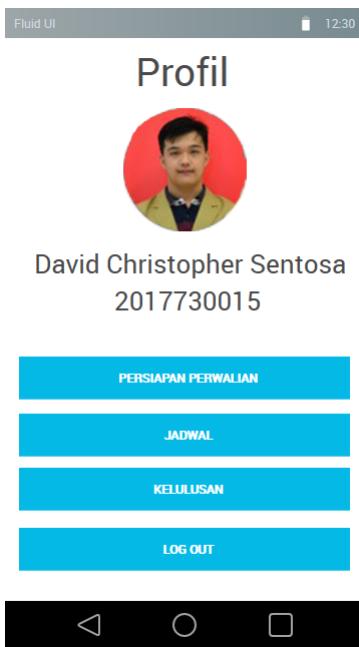
Halaman ini digunakan untuk melakukan *login*. Komponen halaman ini terdiri dari logo aplikasi, kolom *email*, kolom *password*, tombol untuk menampilkan atau menyembunyikan *password*, dan tombol *login* seperti yang ditunjukkan di Gambar 4.5. Untuk melakukan *login*, mahasiswa perlu memasukkan *email* mahasiswa dan *password* yang sesuai kemudian menekan tombol *login*. Jika berhasil pengguna akan diarahkan ke halaman utama.



Gambar 4.5: Rancangan Halaman *Login*

2. Antarmuka Halaman Utama

Halaman utama dituju setelah mahasiswa berhasil melakukan **login**. Halaman ini menampilkan identitas mahasiswa seperti foto profil, nama, dan Nomor Pokok Mahasiswa(NPM) mahasiswa. Terdapat tombol untuk mengakses halaman lain (halaman jadwal, halaman persiapan perwalian, dan halaman kelulusan) di bagian bawah halaman utama. Rancangan antarmuka halaman utama dapat dilihat pada Gambar 4.6.



Gambar 4.6: Rancangan Halaman Utama

3. Antarmuka Halaman Prasyarat Mata Kuliah

Halaman ini menampilkan tabel prasyarat mata kuliah yang dibuka pada semester terkini. Tabel tersebut memiliki tiga kolom yaitu kode mata kuliah, nama mata kuliah, dan status pengambilan mata kuliah. Rancangan antarmuka halaman prasyarat mata kuliah dapat dilihat pada Gambar 4.7.

Kode	Nama	Status
AIF181100	Dasar Pemrograman	Sudah Lulus
AIF182111	Pemrograman Kompetitif 1	Memenuhi Syarat
AIF183002	Penulisan Ilmiah	Tidak Memiliki prasyarat
AIF182112	Pemrograman Kompetitif 2	Tidak memenuhi prasyarat lulus AIF182111 (minimum B)

Gambar 4.7: Rancangan Halaman Prasyarat Mata Kuliah

4. Antarmuka Halaman Jadwal Kuliah

Halaman ini menampilkan jadwal kuliah semester terkini yang tersusun dan terurut berdasarkan hari. Rancangan antarmuka halaman jadwal kuliah dapat dilihat pada Gambar 4.8.



Gambar 4.8: Rancangan Halaman Jadwal Kuliah

5. Antarmuka Halaman Syarat Kelulusan

Halaman ini menampilkan ringkasan informasi akademik pengguna yaitu IPS semester terakhir, IPK, SKS lulus, sisa SKS menuju kelulusan, dan status pengambilan mata kuliah pilihan wajib. Rancangan antarmuka halaman data akademik dapat dilihat pada Gambar 4.9.



Gambar 4.9: Rancangan Halaman Syarat Kelulusan

BAB 5

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Bab ini terdiri dari dua bagian, yaitu Implementasi Perangkat Lunak dan Pengujian Perangkat Lunak. Bagian implementasi berisi penjelasan lingkungan pengembangan perangkat lunak dan hasil implementasi. Sedangkan bagian pengujian berisi hasil pengujian fungsional dan eksperimental terhadap perangkat lunak yang telah dibangun.

5.1.1 Lingkungan Implementasi

Implementasi perangkat lunak ini dilakukan di komputer penulis dengan spesifikasi berikut :

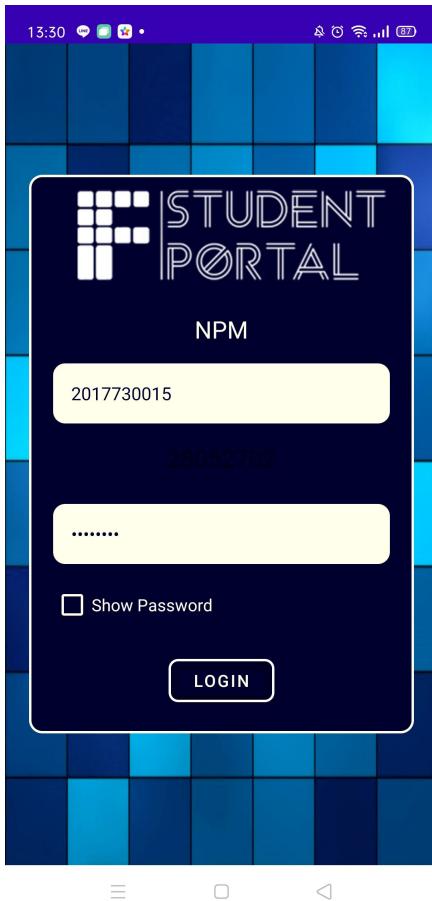
1. Processor: Intel Core i5 9400F
2. RAM: 8.00 GB DDR4
3. Sistem Operasi: Windows 10
4. Versi Android Software Development Kit : API 29 (Android 10.0(Q))
5. Versi Java : Java 8

5.1.2 Hasil Implementasi

Hasil implementasi berupa aplikasi *Android* IFStudentPortal. Aplikasi dapat diunduh melalui *Google Play Store*. Aplikasi IFStudentPortal terdiri dari 5 halaman yaitu:

1. Halaman *Login*

Halaman *login* digunakan pengguna untuk masuk ke dalam aplikasi. Pada halaman ini, pengguna dapat melakukan *login* dengan mengisi *email* pada kolom *email* dan *password* pada kolom *password* kemudian mengklik tombol *login*. Tangkapan layar dari halaman *login* dapat dilihat pada Gambar 5.1.



Gambar 5.1: Halaman *Login*

2. Halaman *Home* / Halaman Utama

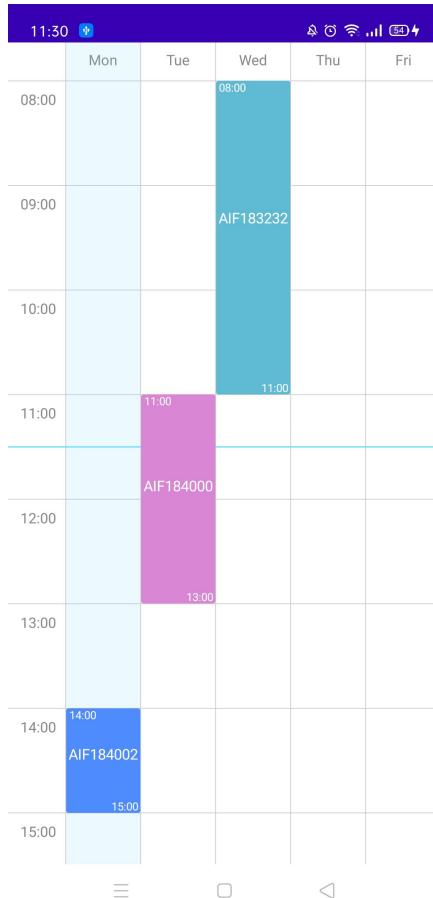
Halaman utama ini ditampilkan saat pengguna berhasil melakukan *login*. Pada halaman ini, pengguna dapat melihat informasi profil pengguna seperti nama, npm, dan foto profil. Pada halaman ini terdapat tombol-tombol untuk mengakses fitur yang ada seperti jadwal, syarat kelulusan, dan persiapan perwalian. Tangkapan layar dari halaman utama dapat dilihat pada Gambar 5.2.



Gambar 5.2: Halaman *Home*

3. Halaman Jadwal

Halaman jadwal ini ditampilkan saat pengguna menekan tombol jadwal di halaman utama. Pada halaman ini, pengguna dapat melihat informasi jadwal kuliah pengguna dalam bentuk grafik yang terurut berdasarkan hari. Jadwal dapat ditekan seperti tombol untuk melihat informasi detail mata kuliah tersebut. Tangkapan layar dari halaman jadwal dapat dilihat pada Gambar 5.3.



Gambar 5.3: Halaman Jadwal

4. Halaman Persiapan Perwalian

Halaman persiapan perwalian ini ditampilkan saat pengguna menekan persiapan perwalian di halaman utama. Pada halaman ini, pengguna dapat melihat informasi mata kuliah pengguna dalam bentuk tabel dengan keterangan mata kuliah tersebut. Untuk setiap mata kuliah akan terdapat salah 1 keterangan antara sudah lulus, tidak memiliki prasyarat, belum memenuhi prasyarat, atau sudah memenuhi prasyarat. Tangkapan layar dari halaman persiapan perwalian dapat dilihat pada Gambar 5.4.

DATA AKADEMIK		
IPS : 4 IPK : 3.47 SKS Lulus : 130 Nilai TOEFL : [621]		
Kode MK		
AIF181100	Dasar Pemrograman	memenuhi syarat
AIF181101	Pemodelan untuk Komputasi	sudah lulus
AIF181103	Matematika Dasar	sudah lulus
AIF181104	Logika Informatika	sudah lulus
AIF181105	Pengantar Informatika	tidak memiliki prasyarat
AIF181106	Matriks dan Ruang Vektor	tidak memiliki prasyarat
AIF181107	Matematika Diskret	sudah lulus
AIF182007	Teknik Presentasi	sudah lulus
AIF182100	Analisis dan Desain Perangkat Lunak	sudah lulus
AIF182101	Algoritma dan Struktur Data	Tidak memenuhi prasyarat lulus AIF181100
AIF182103	Struktur Diskret	sudah lulus
AIF182105	Pemrograman Berorientasi Objek	Tidak memenuhi prasyarat lulus AIF181100
AIF182106	Desain dan Analisis Algoritma	sudah lulus
AIF182109	Statistika untuk Komputasi	sudah lulus

Gambar 5.4: Halaman Persiapan Perwalian

5. Halaman Syarat Kelulusan

Halaman syarat kelulusan ini ditampilkan saat pengguna menekan tombol syarat kelulusan di halaman utama. Pada halaman ini, pengguna dapat melihat informasi syarat-syarat kelulusan dan statusnya dalam bentuk tabel. Tangkapan layar dari halaman syarat kelulusan dapat dilihat pada Gambar 5.5.



Gambar 5.5: Halaman Syarat Kelulusan

5.2 Pengujian

Pengujian perangkat lunak ini dilakukan di perangkat telepon genggam OPPO F7 penulis dengan spesifikasi berikut :

1. Processor: 8 core (4x2.0 GHz Cortex-A73 4x2.0 GHz Cortex-A53)
2. RAM: 4.00 GB
3. Sistem Operasi: ColorOS V7.1
4. Versi *Android*: 10
5. Dimensi : 156 x 75,3 x 7,8 mm
6. Ukuran Layar : 6,23 inci

DAFTAR REFERENSI

- [1] Tim Pengembang Portal Akademik Mahasiswa (PAM) 2018 (2018) Portal Akademik Mahasiswa. https://studentportal.unpar.ac.id/assets/BUKU_PANDUAN_PENGGUNAAN_FRS_GABUNGAN.pdf. [Online; diakses 1-Oktober-2020].
- [2] Heryandi, H. (2015) Informatika student portal: Pengembangan portal akademik mahasiswa untuk mahasiswa teknik informatika unpar. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [3] Hedley, J. (2009-2015) jsoup: Java HTML Parser. <http://jsoup.org>. [Online; diakses 1-Oktober-2020].
- [4] Nugroho, P. A. (2015) SIA Models. <https://github.com/pascalalfadian/SIAModels>. [Online; diakses 1-Oktober-2020].
- [5] Banes, C. (2014) Material Design. <https://github.com/material-components/material-components-android>. [Online; diakses 1-Oktober-2020].
- [6] Android Developers (2014) Android Design. <https://developer.android.com/design>. [Online; diakses 1-Oktober-2020].
- [7] Android Developers (2014) Android Documentation. <https://developer.android.com>. [Online; diakses 1-Oktober-2020].
- [8] Sugiarto, A. (2018) Penyesuaian siamodels dan if student portal ke kurikulum 2018. Skripsi. Universitas Katolik Parahyangan, Indonesia.

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[1] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = ( -aaa + &daa ) / ( bbb++ - ccc % 2 );
14             strcpy(a,"hello_@?");
15         }
16     count = ~mask | 0x00FF00AA;
17 }
18
19 // Fonts for Displaying Program Code in LATEX
20 // Adrian P. Robson, nepswb.co.uk
21 // 8 October 2012
22 // http://nepswb.co.uk/docs/progfonts.pdf
23

```

Listing A.2: MyCode.java

```

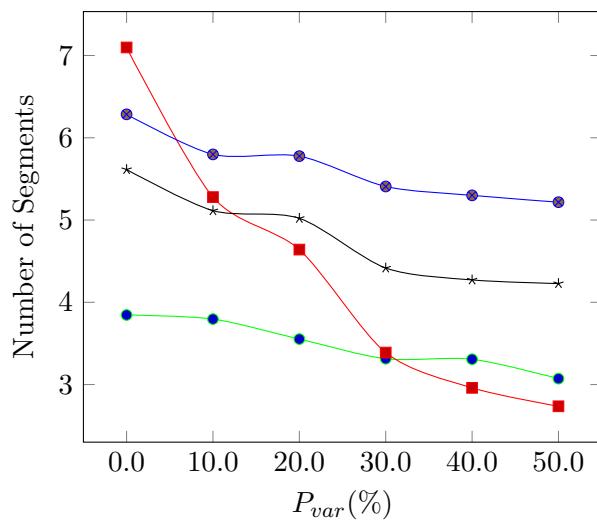
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id;                                //id of the set
8     protected MyEdge FurthestEdge;                   //the furthest edge
9     protected HashSet<MyVertex> set;                //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID;           //store the ID of all vertices
12    protected ArrayList<Double> closeDist;          //store the distance of all vertices
13    protected int totaltrj;                          //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i < totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35}
36

```

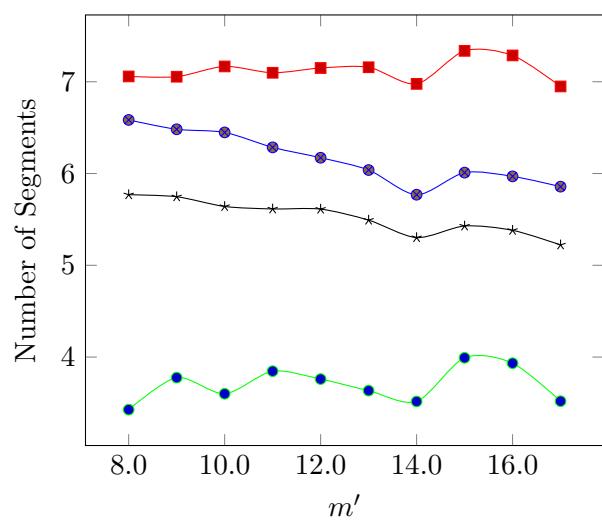

LAMPIRAN B

HASIL EKSPERIMENT

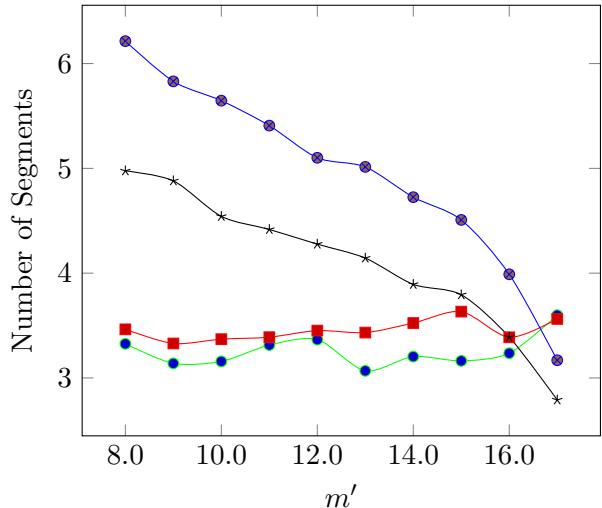
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



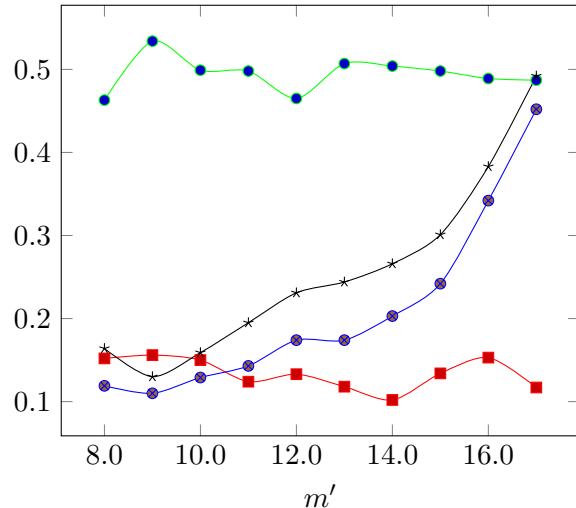
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4