

SKRIPSI

SCREENSAVER INFORMASI MAHASISWA WALI



Harry Senjaya Darmawan

NPM: 2017730067

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
«tahun»

UNDERGRADUATE THESIS

STUDENTS' INFORMATION SCREENSAVER



Harry Senjaya Darmawan

NPM: 2017730067

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

«tahun»

LEMBAR PENGESAHAN

SCREENSAVER INFORMASI MAHASISWA WALI

Harry Senjaya Darmawan

NPM: 2017730067

Bandung, «**tanggal**» «**bulan**» «**tahun**»

Menyetujui,

Pembimbing

Pascal Alfadian, Nugroho, M.Comp.

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:

SCREENSAVER INFORMASI MAHASISWA WALI

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 dijatuahkan 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» «tahun»



Harry Senjaya Darmawan
NPM: 2017730067

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

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

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

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 . . . »

Bandung, «bulan» «tahun»

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	3
2.1 Jsoup	3
2.1.1 Jsoup	3
2.1.2 Connection	3
2.1.3 Response	4
2.1.4 Elements	4
2.1.5 Element	5
2.2 JavaFX dan FXML	5
2.2.1 JavaFX	5
2.2.2 FXML	6
2.3 SIAModels	7
2.3.1 Mahasiswa	7
2.3.2 Nilai	9
2.3.3 ChronologicalComparator	10
2.3.4 MataKuliah	10
2.3.5 JenisKelamin	11
2.3.6 Status	11
2.3.7 TahunSemester	11
3 ANALISIS	13
3.1 Analisis Portal Akademik Mahasiswa	13
3.1.1 <i>Login</i>	13
3.1.2 Halaman Utama	15
3.1.3 Profil	15
3.1.4 Jadwal	16
3.1.5 Pembayaran	22
3.1.6 FRS/PRS	22
3.1.7 Nilai	23
3.1.8 Angket	26

3.1.9	Saran & Komentar	26
3.1.10	Kelulusan	27
3.1.11	Pengajuan	28
3.2	Analisis SIAKAD	28
3.2.1	Login	28
3.2.2	Mahasiswa	28
3.2.3	Pra Kuliah	34
3.2.4	Perkuliahinan	35
3.2.5	Ujian	35
3.2.6	Nilai	35
3.2.7	Evaluasi	35
3.2.8	Skripsi	35
3.2.9	Sidang	35
3.2.10	Kelulusan	35
3.2.11	Pengumuman	36
3.3	Data yang Dibutuhkan untuk <i>Screensaver</i>	36
3.3.1	Portal Akademik Mahasiswa	36
3.3.2	SIAKAD	38
3.4	Analisis Sistem <i>Screensaver</i>	39
3.4.1	Diagram <i>Use Case Screensaver</i>	39
4	PERANCANGAN	41
4.1	Perancangan Kelas	41
4.2	Perancangan Antarmuka	48
5	IMPLEMENTASI DAN PENGUJIAN	51
5.1	Lingkungan Implementasi	51
5.2	Hasil Implementasi	51
DAFTAR REFERENSI		53
A	KODE PROGRAM	55
B	HASIL EKSPERIMEN	57

DAFTAR GAMBAR

3.1 Halaman <i>Login 1</i>	14
3.2 Halaman <i>Login 2</i>	14
3.3 Halaman Utama Portal Akademik Mahasiswa	15
3.4 Halaman Profil	16
3.5 Halaman Kehadiran	16
3.6 Halaman Ketidakhadiran	17
3.7 Halaman Jadwal Kuliah Dalam Tabel Waktu	18
3.8 Halaman Jadwal Kuliah Tabel	18
3.9 Halaman UTS	19
3.10 Halaman UAS	19
3.11 Halaman MKU	20
3.12 Halaman Kalender Akademik	21
3.13 Halaman Daftar Kehadiran	21
3.14 Halaman Pembayaran	22
3.15 Tampilan FRS/PRS	23
3.16 Halaman Nilai Per Semester	23
3.17 Halaman Daftar Perkembangan Studi (1)	24
3.18 Halaman Daftar Perkembangan Studi (2)	24
3.19 Halaman Riwayat Indeks Prestasi	25
3.20 Halaman Nilai TOEFL	26
3.21 Halaman Angket	26
3.22 Halaman Saran & Komentar	27
3.23 Halaman Kelulusan	27
3.24 Halaman Pengajuan	28
3.25 Tangkapan Layar Halaman Daftar Mahasiswa Wali	29
3.26 Tangkapan Layar Pop-up Detail Mahasiswa	29
3.27 Tangkapan Layar Tab Identitas Mahasiswa	30
3.28 Tangkapan Layar Tab Riwayat Pendidikan	31
3.29 Tangkapan Layar Tab Orang Tua/Wali	31
3.30 Tangkapan Layar Tab Ringkasan Akademik	32
3.31 Tangkapan Layar Tab Nilai Semester Ini	33
3.32 Tangkapan Layar Tab Nilai Per Tahun Semester	33
3.33 Tangkapan Layar Tab Nilai Berdasarkan Kurikulum	34
3.34 Tangkapan Layar Tab Status Akademik	34
3.35 Tangkapan Layar PDF Daftar Perkembangan Studi	35
3.36 Diagram <i>Use Case Screensaver</i>	39
4.1 Diagram Kelas Keseluruhan	41
4.2 Diagram Kelas SIAKAD	42
4.3 Diagram Kelas Studentportal	43
4.4 <i>Script</i> Data Nilai Mahasiswa Pada Halaman Nilai	48
4.5 Rancangan Antarmuka <i>Screensaver</i>	49

5.1 Tampilan <i>Screensaver</i>	51
B.1 Hasil 1	57
B.2 Hasil 2	57
B.3 Hasil 3	57
B.4 Hasil 4	57

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Setiap dosen wali (dikenal juga dengan istilah penasehat akademik ¹) memiliki data mengenai mahasiswa walinya yang dapat diakses melalui SIAKAD [1]. Namun, walaupun dosen wali memiliki data mengenai mahasiswa walinya, dosen wali juga perlu melakukan pemeriksaan data mahasiswa walinya, terutama data akademiknya secara berkala. Dengan berbagai kesibukan yang dialami oleh para dosen wali dan mahasiswa, ditambah dengan situasi Indonesia saat ini yang menyebabkan perkuliahan dilakukan secara daring, akan sangat sulit bagi dosen wali untuk menemui mahasiswa walinya. Hal ini menyebabkan dosen wali kesulitan mengamati perkembangan mahasiswa walinya. Selain itu dalam mencari data akademik mahasiswa wali, SIAKAD juga memiliki kekurangan dimana dosen wali perlu melakukan *login*, kemudian mencari serta memilih mahasiswa wali yang ingin dilihat. Sehingga dapat dikatakan proses tersebut tidaklah efisien.

Maka dari itu, pada skripsi ini akan dibuat sebuah perangkat lunak yang berupa *screensaver* yang dapat menampilkan data akademik mahasiswa wali secara acak. *Screensaver* adalah program komputer yang mengosongkan layar atau mengisinya dengan gambar atau pola bergerak ketika komputer telah diam dalam waktu yang lama [2]. Dengan menggunakan perangkat lunak tersebut, dosen wali dapat tetap mengamati perkembangan mahasiswa walinya, paling tidak secara akademik.

Dikarenakan terbimbing tidak memiliki akses ke SIAKAD untuk mengakses data mahasiswa wali, namun terbimbing memiliki akses ke Portal Akademik Mahasiswa [3] maka, terbimbing mensimulasikan dengan Portal Akademik Mahasiswa, dan kemudian Pembimbing mengubah aksesnya ke SIAKAD. Struktur kelas yang akan digunakan dalam pembuatan perangkat lunak ini yaitu, struktur kelas SIAModels yang tersedia pada Github dan Maven Public Repository [4]. Simulasi Portal Akademik Mahasiswa ini berdasarkan pada skripsi yang dibuat oleh Andrianto Sugiarto [5]. Tetapi terdapat beberapa perbaikan yang perlu dilakukan, dan dapat dilihat pada sub-bab 3.1.

Perangkat lunak akan dibangun menggunakan bahasa pemrograman Java. Terdapat beberapa teknologi yang dapat dimanfaatkan dalam bahasa pemrograman Java. Teknologi yang pertama yaitu *library* jsoup. Jsoup dapat digunakan untuk melakukan *scraping*, sehingga pengambilan data mahasiswa tidak memerlukan API (*Application Programming Interface*) [6]. Teknologi lainnya yang dapat dimanfaatkan yaitu JavaFX. JavaFX dapat digunakan untuk mengonversi aplikasi tersebut menjadi *screensaver*.

1.2 Rumusan Masalah

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

- Bagaimana cara memanfaatkan jsoup untuk mengambil data mahasiswa?
- Bagaimana cara memanfaatkan JavaFX untuk mengonversi aplikasi tersebut menjadi *screensaver*?

¹Sebagaimana dituliskan pada <https://unpar.ac.id/akademik/>. Pada skripsi ini istilah yang digunakan adalah dosen wali karena istilah tersebut yang muncul pada aplikasi SIAKAD [1]

1.3 Tujuan

Tujuan yang ingin dicapai dari penulisan skripsi ini sebagai berikut:

- Memanfaatkan jsoup untuk mengambil data mahasiswa.
- Memanfaatkan JavaFX untuk mengonversi aplikasi tersebut menjadi *screensaver*.

1.4 Batasan Masalah

Dikarenakan terbimbing tidak memiliki akses ke SIAKAD untuk mengakses data mahasiswa wali, namun terbimbing memiliki akses ke Portal Akademik Mahasiswa maka, terbimbing mensimulasikan dengan Portal Akademik Mahasiswa, dan kemudian Pembimbing mengubah aksesnya ke SIAKAD.

1.5 Metodologi

Langkah-langkah yang akan dilakukan dalam melakukan penelitian ini yaitu:

1. Melakukan studi mengenai jsoup.
2. Melakukan studi mengenai cara mengonversi aplikasi menjadi *screensaver*.
3. Mempelajari struktur kelas SIAModels.
4. Menganalisis IF Portal Akademik Mahasiswa dan Portal Akademik Mahasiswa.
5. Merancang struktur kelas aplikasi.
6. Melakukan studi mengenai cara mendesain antarmuka aplikasi
7. Mendesain antarmuka aplikasi.
8. Mengimplementasikan jsoup untuk mengambil data mahasiswa.
9. Mengonversi aplikasi menjadi *screensaver* dengan menggunakan JavaFX.
10. Melakukan pengujian dan eksperimen.
11. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Dokumen dibagi ke dalam beberapa bab dengan sistematika pembahasan sebagai berikut:

- Bab 1. Pendahuluan, membahas tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, metode penelitian dan sistematika pembahasan mengenai skripsi.
- Bab 2. Landasan Teori, membahas landasan dari teori-teori yang berhubungan serta mendukung penelitian, meliputi jsoup, JavaFX, dan SIAModels.
- Bab 3. Analisis, menjelaskan tentang analisis Portal Akademik Mahasiswa, analisis SIAKAD, analisis data yang dibutuhkan untuk *screensaver*, serta analisis sistem *screensaver*.
- Bab 4. Perancangan, membahas perancangan kelas beserta deskripsi kelas dan fungsinya, serta perancangan antarmuka.
- Bab 5. Implementasi dan pengujian, membahas hasil-hasil implementasi dan pengujian secara fungsional dan eksperimental.
- Bab 6. Kesimpulan dan saran, membahas kesimpulan yang diperoleh dari penelitian ini dan saran untuk pengembangan berikutnya.

BAB 2

LANDASAN TEORI

Bab Landasan Teori ini berisi teori-teori yang menjadi dasar penelitian ini, meliputi jsoup, JavaFX, dan SIAModels.

2.1 Jsoup

Salah satu teknologi yang dapat dimanfaatkan untuk melakukan *scraping* yaitu *library* Java jsoup. Jsoup adalah *library* Java untuk mengerjakan dokumen HTML yang menyediakan API yang baik untuk mengekstraksi, memanipulasi data, dan menyelesaikan pembersihan data awal menggunakan metode terbaik dari *Document Object Model* (DOM), *Cascading Style Sheets* (CSS), dan metode lain yang mirip dengan jQuery. Jsoup mengimplementasikan spesifikasi WHATWG HTML5, dan mem-parsing HTML ke DOM yang sama seperti yang dilakukan *browser* modern. Pada skripsi ini akan digunakan jsoup versi 1.13.1. Berikut adalah layanan utama yang tersedia di jsoup [6]:

1. *Scrape* dan *parse* HTML dari URL, *file*, atau string.
2. Mencari dan ekstrak data menggunakan traversal DOM dan CSS *selector*.
3. Memanipulasi elemen HTML, atribut HTML, dan teks.
4. Membersihkan konten yang dikirim oleh pengguna yang menggunakan *safe white-lists* untuk mencegah serangan XSS.
5. Menghasilkan HTML yang rapi.

Subbab-subbab berikut menjelaskan beberapa kelas dari jsoup.

2.1.1 Jsoup

Kelas ini merupakan inti untuk mengakses fungsionalitas jsoup. Salah satu *method* yang dimiliki kelas ini adalah sebagai berikut:

- `public static Connection connect(String url)`

Berfungsi untuk membuat koneksi baru ke URL. Digunakan untuk mengambil dan mengurai halaman HTML.

Parameter: URL situs web dengan protokol HTTP atau HTTPS.

Kembalian: koneksi dengan situs web.

2.1.2 Connection

Kelas ini merupakan *interface* yang menyediakan antarmuka yang nyaman untuk mengambil konten dari web, dan menguraikannya menjadi dokumen. Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

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

Berfungsi untuk menambahkan *cookies* ke *request*.

Parameter:

– *cookies*: Map dari *cookie*.

Kembalian: koneksi yang sama tetapi sudah diubah.

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

Berfungsi untuk menambahkan parameter data *request* yang bisa dikirim melalui *method* HTTP GET atau POST.

Parameter:

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

Kembalian: koneksi yang sama tetapi sudah diubah.

- `Connection method(URLConnection.Method method)`

Berfungsi untuk mengatur *method request* yang akan digunakan, HTTP GET atau POST. *Default*-nya adalah GET.

Parameter:

- *method*: *method request* HTTP.

Kembalian: koneksi yang sama tetapi sudah diubah.

- `Connection timeout(int millis)`

Berfungsi untuk mengatur batas waktu *request*. Batas waktu *default* adalah 30 detik. Batas waktu nol akan dianggap sebagai batas waktu yang tak terhingga.

Parameter:

- *millis*: batas waktu dalam milidetik.

Kembalian: koneksi yang sama tetapi sudah diubah.

- `Connection.Response execute()`

Berfungsi untuk mengirim *request* HTTP.

Kembalian: objek Response.

2.1.3 Response

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

- `Document parse()`

Berfungsi untuk mengurai *body* jawaban menjadi dokumen.

Kembalian: dokumen yang diurai.

- `String body()`

Berfungsi untuk mendapatkan *body* jawaban dalam bentuk *string*.

Kembalian: *body* jawaban dalam bentuk *string*.

2.1.4 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()`

Berfungsi untuk mendapatkan kombinasi teks dari seluruh elemen yang sesuai.

Kembalian: seluruh teks dalam *string*.

2.1.5 Element

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

- `public Elements select(String cssQuery)`

Berfungsi untuk menemukan elemen yang cocok dengan kueri CSS Selector, dengan elemen ini sebagai konteks awal.

Parameter:

- `cssQuery`: kueri CSS berupa CSS Selector.

Kembalian: elemen-elemen HTML yang sesuai dengan kueri CSS.

- `public Element child(int index)`

Berfungsi untuk mendapatkan anak elemen berdasarkan nomor indeks.

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 gabungan dari elemen.

Kembalian: teks dalam *string*.

2.2 JavaFX dan FXML

2.2.1 JavaFX

JavaFX adalah platform aplikasi klien *open source* generasi berikutnya untuk *desktop*, *mobile*, dan *embedded systems* yang dibangun dengan Java. JavaFX memungkinkan untuk membuat aplikasi Java dengan antarmuka pengguna (UI) modern dengan akselerasi perangkat keras yang sangat portabel. [7]. Subbab-subbab berikut menjelaskan beberapa kelas dari JavaFX. [7]

Application

Titik masuk untuk aplikasi JavaFX adalah kelas `Application`. JavaFX melakukan hal berikut, secara berurutan, setiap kali aplikasi diluncurkan:

1. Membuat *instance* kelas `Application` yang ditentukan
2. Memanggil *method* `init()`
3. Memanggil *method* `start(javafx.stage.Stage)`
4. Menunggu aplikasi selesai, yang terjadi jika salah satu dari hal berikut terjadi:
 - aplikasi memanggil `Platform.exit()`
 - *window* terakhir telah ditutup dan atribut `implicitExit` di `Platform` adalah true
5. Memanggil *method* `stop()`

Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- `public void init()`

Method inisialisasi aplikasi. *Method* ini dipanggil segera setelah kelas `Application` dimuat dan dibangun. *Method* ini dapat ditimpak untuk melakukan inisialisasi sebelum aplikasi sebenarnya dimulai.

- `public abstract void start(Stage primaryStage)`

Titik masuk utama untuk semua aplikasi JavaFX. *Method* `start` dipanggil setelah *method*

`init` kembali, dan setelah sistem siap untuk aplikasi mulai berjalan.

Parameter:

- `primaryStage`: *stage* utama untuk aplikasi ini, tempat *scene* aplikasi dapat diatur.
- `public static void launch()`
Meluncurkan aplikasi. *Method* ini biasanya dipanggil dari `main()` *method*. Tidak boleh dipanggil lebih dari sekali atau *exception* akan dilemparkan. Harus merupakan *subclass* dari `Application` atau `RuntimeException` akan dilemparkan.

Stage

Kelas `Stage` adalah *container* JavaFX tingkat atas. `Stage` utama dibangun oleh platform. Objek `Stage` harus dibuat dan dimodifikasi pada JavaFX Application Thread.

Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- `public final void setScene(Scene value)`
Menentukan *scene* yang akan digunakan di *stage* ini.
- Parameter:**
 - `value`: *scene* yang akan digunakan.
- `public final void show()`
Mencoba menampilkan *window* ini dengan mengubah *visibility* menjadi true.

Scene

Kelas `Scene` adalah wadah untuk semua konten dalam grafik *scene*.

Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- `public Scene(Parent root, double width, double height)`
Merupakan *constructor* dari kelas `Scene`.
- Parameter:**
 - `root`: Node root dari grafik *scene*.
 - `width`: Lebar *scene*.
 - `height`: Tinggi *scene*.

FXMLLoader

Memuat hierarki objek dari dokumen XML.

Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- `public FXMLLoader(URL location)`
Merupakan *constructor* dari kelas `FXMLLoader`.
- Parameter:**
 - `location`: lokasi dokumen fxml.
- `public <T> T load`
Memuat hierarki objek dari dokumen FXML.

2.2.2 FXML

FXML adalah bahasa *markup* berbasis XML yang dapat dituliskan untuk membangun grafik objek Java. FXML memberikan alternatif yang nyaman untuk membuat grafik dalam kode prosedural, dan cocok untuk mendefinisikan antarmuka pengguna aplikasi JavaFX, karena struktur hierarki dari dokumen XML sangat mirip dengan struktur grafik *scene* JavaFX. [7] Subbab-subbab berikut menjelaskan beberapa bagian dari FXML.

2.3 SIAModels

SIAModels merupakan kelas-kelas dalam bahasa Java yang merepresentasikan objek-objek yang tersedia di Sistem Informasi Akademik UNPAR [4]. Pada skripsi ini akan digunakan SIAModels versi 4.0.0.

2.3.1 Mahasiswa

Kelas ini merepresentasikan mahasiswa. Atribut yang dimiliki kelas ini antara lain:

- `String npm` : Nomor Pokok Mahasiswa (NPM).
- `String nama` : nama mahasiswa.
- `List<Nilai> riwayatNilai` : riwayat nilai yang dimiliki mahasiswa.
- `String photoPath` : URL dari foto mahasiswa.
- `List<JadwalKuliah> jadwalKuliahList` : daftar jadwal kuliah mahasiswa
- `SortedMap<LocalDate, Integer> nilaiTOEFL` : nilai TOEFL mahasiswa.
- `Status status` : status mahasiswa.
- `LocalDate tanggalLahir` : tanggal lahir mahasiswa.
- `JenisKelamin jenisKelamin` : jenis kelamin mahasiswa.

Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- `public Mahasiswa(String npm)`
Merupakan *constructor* dari kelas `Mahasiswa`.

Parameter:

– `npm`: nomor pokok mahasiswa.

- `public String getNama()`
Berfungsi untuk mendapatkan nama mahasiswa.

Kembalian: nama mahasiswa.

- `public void setNama(String nama)`
Berfungsi untuk mengubah nama mahasiswa.

Parameter:

– `nama`: nama mahasiswa.

- `public String getNpm()`
Berfungsi untuk mendapatkan nomor pokok mahasiswa.

Kembalian: nomor pokok mahasiswa.

- `public String getPhotoPath()`
Berfungsi untuk mendapatkan URL dari foto mahasiswa.

Kembalian: URL dari foto mahasiswa.

- `public void setPhotoPath(String photoPath)`
Berfungsi untuk mengubah URL dari foto mahasiswa.

Parameter:

– `photoPath`: URL dari foto mahasiswa.

- `public List<JadwalKuliah> getJadwalKuliahList`
Berfungsi untuk mendapatkan jadwal kuliah mahasiswa.

Kembalian: jadwal kuliah mahasiswa dalam *list*.

- `public void setJadwalKuliahList(List<JadwalKuliah> jadwalKuliahList)`
Berfungsi untuk mengubah jadwal kuliah mahasiswa.

Parameter:

– `jadwalKuliahList`: jadwal kuliah mahasiswa dalam *list*.

- `public String getEmailAddress()`
Berfungsi untuk mendapatkan *email* mahasiswa.

Kembalian: *email* mahasiswa.

- `public List<Nilai> getRiwayatNilai()`
Berfungsi untuk mendapatkan riwayat nilai mahasiswa.

Kembalian: riwayat nilai mahasiswa dalam List.

- `public SortedMap<LocalDate, Integer> getNilaiTOEFL()`
Berfungsi untuk mendapatkan nilai TOEFL mahasiswa.
- `public void setNilaiTOEFL(SortedMap<LocalDate, Integer> nilaiTOEFL)`
Berfungsi untuk mengubah nilai TOEFL mahasiswa.

Parameter:

- `nilaiTOEFL`: nilai TOEFL mahasiswa dalam SortedMap.

- `public Status getStatus()`
Berfungsi untuk mendapatkan status mahasiswa.
- `public void setStatus(Status status)`
Berfungsi untuk mengubah status mahasiswa.

Parameter:

- `status`: status mahasiswa.

- `public LocalDate getTanggalLahir()`
Berfungsi untuk mendapatkan tanggal lahir mahasiswa.
- `public void setTanggalLahir(LocalDate tanggalLahir)`
Berfungsi untuk mengubah tanggal lahir mahasiswa.

Parameter:

- `tanggalLahir`: tanggal lahir mahasiswa.

- `public JenisKelamin getJenisKelamin()`
Berfungsi untuk mendapatkan jenis kelamin mahasiswa.
- `public void setJenisKelamin(JenisKelamin jenisKelamin)`
Berfungsi untuk mengubah jenis kelamin mahasiswa.

Parameter:

- `jenisKelamin`: jenis kelamin mahasiswa.

- `public byte[] getPhotoImage()`
Berfungsi untuk mendapatkan foto mahasiswa yang dibungkus dalam kelas `java.awt.Image`. Berbeda dengan `method getPhotoPath()`, `method` ini akan menghasilkan image, apapun bentuk photo path nya (bisa berupa URL ataupun base64 string).

Kembalian: foto mahasiswa.

- `public double calculateIPKLulus()`
Berfungsi untuk menghitung IPK mahasiswa sampai saat ini, dengan aturan kuliah yang tidak lulus tidak dihitung dan jika pengambilan beberapa kali, diambil nilai terbaik. Sebelum memanggil `method` ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah.
- `public double calculateIPTempuh(boolean lulusSaja)`
Berfungsi untuk menghitung IP mahasiswa sampai saat ini, dengan aturan perhitungan kuliah yang tidak lulus ditentukan parameter, dan jika pengambilan beberapa kali, diambil nilai terbaik.

Parameter:

- `lulusSaja`: `lulusSaja` set true jika ingin membuang mata kuliah tidak lulus, false jika ingin semua (sama dengan "IP N. Terbaik" di DPS). Sebelum memanggil `method` ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah.

Kembalian: IPK lulus.

- `public double calculateIPKumulatif()`
Berfungsi untuk menghitung IP Kumulatif mahasiswa sampai saat ini, dengan aturan jika pengambilan beberapa kali, diambil semua. Sebelum memanggil `method` ini, `getRiwayatNilai()`

harus sudah mengandung nilai per mata kuliah.

Kembalian: IPK lulus.

- `public double calculateIPS()`

Berfungsi untuk menghitung IPS semester terakhir sampai saat ini, dengan aturan kuliah yang tidak lulus dihitung. Sebelum memanggil *method* ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah.

Kembalian: nilai IPS sampai saat ini.

- `public int calculateSKSLulus()`

Berfungsi untuk menghitung jumlah SKS lulus mahasiswa saat ini. Sebelum memanggil *method* ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah.

Kembalian: SKS lulus.

- `public int calculateSKSTempuh(boolean lulusSaja)`

Berfungsi untuk menghitung jumlah SKS tempuh mahasiswa saat ini. Sebelum memanggil *method* ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah.

Parameter:

- `lulusSaja`: lulusSaja set true jika ingin membuang SKS tidak lulus

Kembalian: SKS tempuh

- `public Set<TahunSemester> calculateTahunSemesterAktif()`

Berfungsi untuk mendapatkan seluruh tahun semester di mana mahasiswa ini tercatat sebagai mahasiswa aktif, dengan strategi memeriksa riwayat nilainya. Jika ada satu nilai saja pada sebuah tahun semester, maka dianggap aktif pada semester tersebut.

Kembalian: kumpulan tahun semester di mana mahasiswa ini aktif.

- `public boolean hasLulusKuliah(String kodeMataKuliah)`

Berfungsi untuk memeriksa apakah mahasiswa ini sudah lulus mata kuliah tertentu. Sebelum memanggil *method* ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah.

Parameter:

- `kodeMataKuliah`: kode mata kuliah yang ingin diperiksa kelulusannya.

Kembalian: `true` jika sudah pernah mengambil dan lulus, `false` jika belum.

- `public boolean hasTempuhKuliah(String kodeMataKuliah)`

Memeriksa apakah mahasiswa ini sudah pernah menempuh mata kuliah tertentu. Sebelum memanggil *method* ini, `getRiwayatNilai()` harus sudah mengandung nilai per mata kuliah.

Parameter:

- `kodeMataKuliah`: kode mata kuliah yang ingin diperiksa kelulusannya.

Kembalian: `true` jika sudah pernah mengambil, `false` jika belum.

- `public int getTahunAngkatan()`

Mendapatkan tahun angkatan mahasiswa ini berdasarkan NPM-nya.

Kembalian: tahun angkatan.

2.3.2 Nilai

Kelas ini merepresentasikan nilai yang ada pada riwayat nilai mahasiswa. Atribut yang dimiliki kelas ini antara lain:

- `TahunSemester tahunSemester`: tahun dan semester kuliah ini diambil
- `MataKuliah mataKuliah`: mata kuliah yang diambil.
- `Character kelas`: kelas kuliah.
- `Map<String, Double> nilaiTugas`: nilai Angka Rata-rata Tugas (ART).
- `Double nilaiUTS`: nilai Ujian Tengah Semester (UTS).
- `Double nilaiUAS`: nilai Ujian Akhir Semester (UAS).
- `String nilaiAkhir`: nilai akhir.

Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- `public Nilai(TahunSemester tahunSemester, MataKuliah mataKuliah, Character kelas, Map<String, Double> nilaiTugas, Double nilaiUTS, Double nilaiUAS, String nilaiAkhir)`

Merupakan *constructor* dari kelas *Nilai*.

Parameter:

- *tahunSemester*: tahun dan semester kuliah ini diambil.
- *mataKuliah*: mata kuliah yang diambil.
- *kelas*: kelas kuliah.
- *nilaiTugas*: nilai ART.
- *nilaiUTS*: nilai UTS.
- *nilaiUAS*: nilai UAS.
- *nilaiAkhir*: nilai akhir.

- **public MataKuliah getMataKuliah()**

Mendapatkan mata kuliah yang diambil.

Kembalian: mata kuliah.

- **public String getNilaiAkhir()**

Mengembalikan nilai akhir dalam bentuk huruf (A, B, C, D, ..., atau K).

Kembalian: nilai akhir dalam huruf, atau *null* jika tidak ada.

- **public Double getAngkaAkhir()**

Mendapatkan nilai akhir dalam bentuk angka.

Kembalian: nilai akhir dalam angka, atau *null* jika *getNilaiAkhir()* mengembalikan *null*.

- **public int getTahunAjaran()**

Mendapatkan tahun ajaran saat pengambilan mata kuliah.

Kembalian: tahun ajaran saat pengambilan mata kuliah.

- **public TahunSemester getTahunSemester()**

Mendapatkan tahun dan semester pengambilan mata kuliah.

Kembalian: tahun dan semester pengambilan mata kuliah.

- **public Semester getSemester()**

Mendapatkan semester pengambilan mata kuliah.

Kembalian: semester pengambilan mata kuliah

2.3.3 ChronologicalComparator

Pembanding antara satu nilai dengan nilai lainnya, secara kronologis waktu pengambilan. *Method* yang dimiliki kelas ini adalah sebagai berikut:

- **public int compare(Nilai o1, Nilai o2)**
Berfungsi untuk membandingkan nilai.

Parameter:

- *o1*: nilai pertama yang akan dibandingkan.
- *o2*: nilai kedua yang akan dibandingkan.

Kembalian: hasil perbandingan.

2.3.4 MataKuliah

Kelas ini merepresentasikan sebuah mata kuliah. Atribut yang dimiliki kelas ini antara lain:

- **String kode**: kode mata kuliah
- **String nama**: nama mata kuliah
- **Integer sks**: sks mata kuliah.

Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- **public MataKuliah(String kode, String nama, int sks)**
Merupakan *constructor* dari kelas *MataKuliah*.

Parameter:

- *kode*: kode mata kuliah.
- *nama*: nama mata kuliah.

- `sks`: sks mata kuliah.
- `public String getKode()`
Mendapatkan kode mata kuliah.
Kembalian: kode mata kuliah.
- `public int getSks()`
Mendapatkan sks mata kuliah.
Kembalian: sks mata kuliah.
- `public String getNama()`
Mendapatkan nama mata kuliah.
Kembalian: nama mata kuliah.

2.3.5 JenisKelamin

Kelas ini berupa enum yang merepresentasikan jenis kelamin mahasiswa. Nilai dari enum ini antara lain:

- `LAKI_LAKI("Laki-laki")`
- `PEREMPUAN("Perempuan")`

2.3.6 Status

Kelas ini berupa enum yang merepresentasikan status mahasiswa. Nilai dari enum ini antara lain:

- `SEMUA("00")`
- `AKTIF("01")`
- `GENCAT("02")`
- `CUTI_SEBELUM_FRS("03")`
- `CUTI_SETELAH_FRS("04")`
- `KELUAR("05")`
- `LULUS("06")`
- `DROP_OUT("07")`
- `SISIPAN("08")`

2.3.7 TahunSemester

Kelas ini menyimpan konstanta untuk semester beserta tahunnya di UNPAR. Atribut yang dimiliki kelas ini antara lain:

- `String kodeTahunSemester`: kode semester 3 digit, 2 digit pertama berupa tahun, digit terakhir menandakan semester dengan definisi 1 untuk ganjil, 2 untuk genap, 4 untuk pendek, dan 6 untuk transfer.

Beberapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- `public TahunSemester(String kodeTahunSemester)`
Method ini merupakan constructor dari kelas TahunSemester.
Parameter:
 - `kodeTahunSemester`: semester dalam bentuk teks (GANJIL, GENAP, PENDEK, TRANSFER, dan UNKNOWN5).
- `public TahunSemester(int tahun, Semester semester)`
Method ini merupakan constructor dari kelas TahunSemester.
Parameter:
 - `tahun`: tahun ajaran.
 - `semester`: semester dari tahun ajaran.
- `public Semester getSemester()`
Method ini berfungsi untuk mendapatkan semester.
Kembalian: semester dalam teks.

- `public int getTahun()`
Method ini berfungsi untuk mendapatkan tahun.
Kembalian: tahun ajaran.
- `private static void validateKodeSemester(String kodeTahunSemester)`
Method ini berfungsi untuk melakukan validasi terhadap kode tahun semester.
Parameter:
 - `kodeTahunSemester`: kode tahun semester.

BAB 3

ANALISIS

Pada bab ini akan dijelaskan mengenai analisis Portal Akademik Mahasiswa, analisis SIAKAD, analisis data yang dibutuhkan untuk *screensaver*, serta analisis sistem *screensaver*.

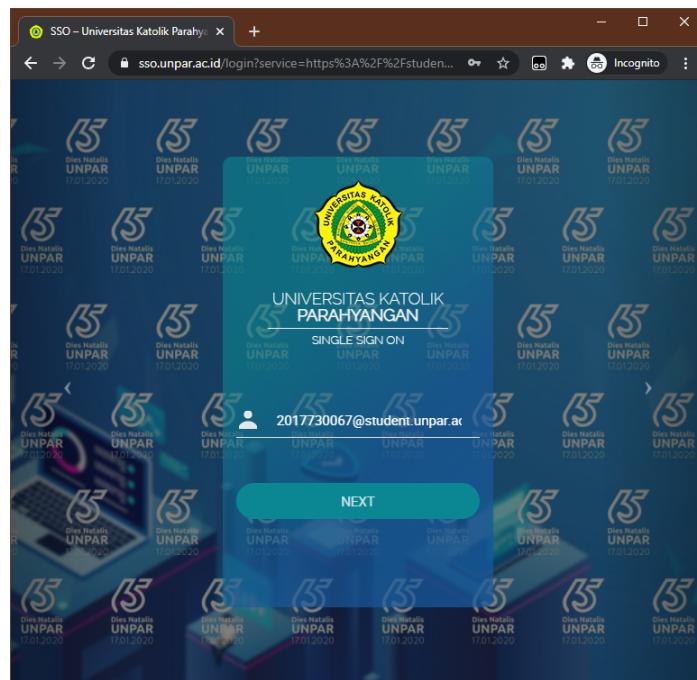
3.1 Analisis Portal Akademik Mahasiswa

Untuk mengambil data mahasiswa, diperlukan sumber data mahasiswa tersebut. Sumber data mahasiswa tersebut dapat diperoleh melalui Portal Akademik Mahasiswa. Portal Akademik Mahasiswa merupakan sebuah situs yang diperuntukkan bagi mahasiswa untuk mendapatkan informasi mengenai profil dan kegiatan akademik mahasiswa tersebut. Mahasiswa dapat mengakses Portal Akademik Mahasiswa melalui *URL* <https://studentportal.unpar.ac.id/>. Untuk mengakses Portal Akademik Mahasiswa, mahasiswa harus melakukan *login* menggunakan *email* dan *password* mahasiswa tersebut.

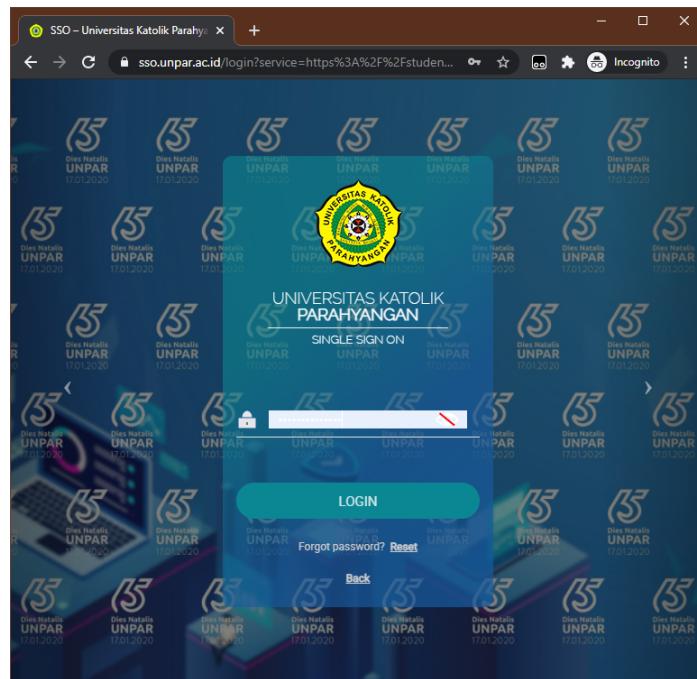
Aplikasi *screensaver* akan melakukan *http request* ke Portal Akademik Mahasiswa untuk mendapatkan data untuk setiap kebutuhan dari masing-masing fitur yang ada, dimana fitur-fitur yang terdapat dalam aplikasi Mahasiswa Wali *Screensaver* adalah informasi umum mengenai mahasiswa, serta prestasi akademik mahasiswa. Pengambilan data secara langsung dari Portal Akademik Mahasiswa dilakukan menggunakan *library jsoup*. Beberapa implementasi pemanfaatan jsoup untuk mengambil data-data tersebut sudah diimplementasikan pada skripsi Andrianto Sugiarto [5] sebelumnya. Data yang telah didapat dari Portal Akademik Mahasiswa kemudian diolah ke dalam SIAModels, dan ditampilkan sesuai dengan fitur-fitur yang ada pada aplikasi Mahasiswa Wali *Screensaver*.

3.1.1 *Login*

Halaman *Login* (Gambar 3.1 dan 3.2) merupakan halaman dimana mahasiswa memasukkan *email* dan *password* untuk mengakses menu-menu Portal Akademik Mahasiswa.



Gambar 3.1: Halaman Login 1



Gambar 3.2: Halaman Login 2

Login dilakukan dengan mengirim *request method POST*, dan kemudian mengambil session yang akan digunakan sebagai *cookies* apabila *login* berhasil. Terdapat beberapa perubahan yang terjadi pada situs Portal Akademik Mahasiswa semenjak skripsi Andrianto Sugiarto [5], yang mengakibatkan perlunya perubahan (Kode 3.1) terhadap implementasi jsoup:

1. Menghapus pemanggilan fungsi `validateTLCertificates()` dikarenakan sudah *deprecated* [6].
2. Menghapus pengambilan data dengan kueri css "`input [name=lt]`" dikarenakan kueri tersebut sudah dihapus oleh Portal Akademik Mahasiswa.
3. Menghapus *form data* dengan *key "submit"* yang memiliki *value ""*.

Kode 3.1: Perubahan Implementasi Jsoup Login

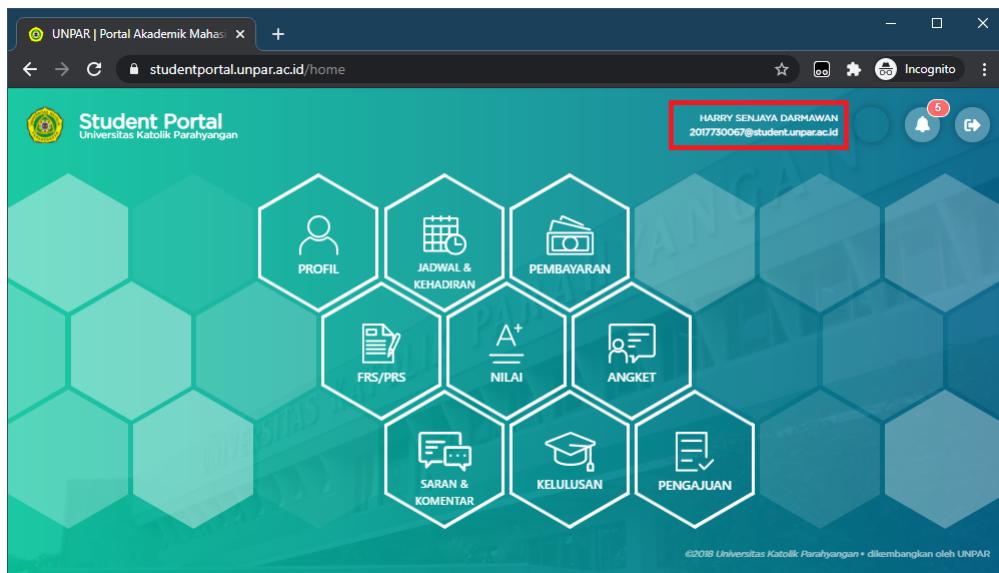
```

1 @@ -78,11 +77,9 @@ public class Scraper {
2     Connection conn = Jsoup.connect(LOGIN_URL);
3     conn.data("Submit", "Login");
4     conn.timeout(0);
5     conn.validateTLCertificates(false);
6     conn.method(Connection.Method.POST);
7     Response resp = conn.execute();
8     Document doc = resp.parse();
9     String lt = doc.select("input[name=lt]").val();
10    String execution = doc.select("input[name=execution]").val();
11    String jsessionid = resp.cookie("JSESSIONID");
12    /* SSO LOGIN */
13 @@ -90,12 +87,9 @@ public class Scraper {
14     loginConn.cookies(resp.cookies());
15     loginConn.data("username", user);
16     loginConn.data("password", pass);
17     loginConn.data("lt", lt);
18     loginConn.data("execution", execution);
19     loginConn.data("_eventId", "submit");
20     loginConn.data("submit", "");
21     loginConn.timeout(0);
22     loginConn.validateTLCertificates(false);
23     loginConn.method(Connection.Method.POST);
24     resp = loginConn.execute();
25     if (resp.body().contains(user)) {

```

3.1.2 Halaman Utama

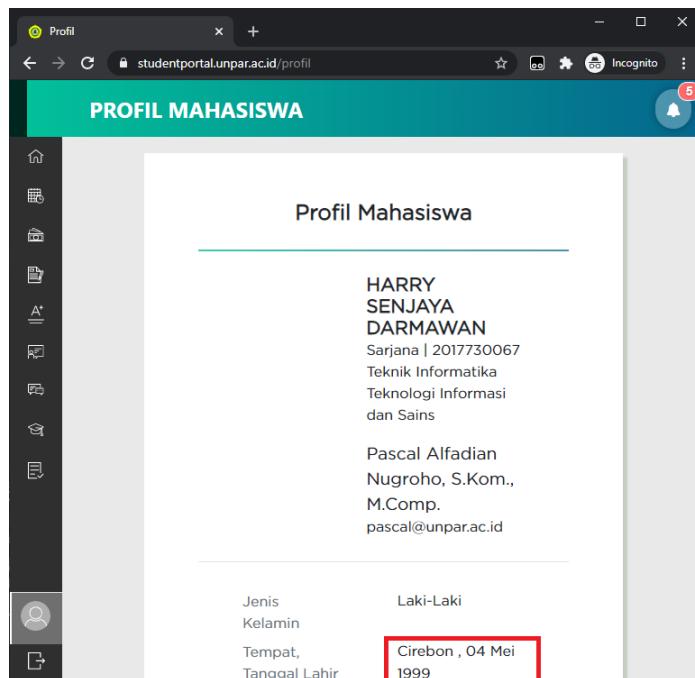
Pada Halaman Utama Portal Akademik Mahasiswa (Gambar 3.3) terdapat beberapa menu yang dapat digunakan sebagai sumber data.



Gambar 3.3: Halaman Utama Portal Akademik Mahasiswa

3.1.3 Profil

Menu Profil merupakan halaman yang menampilkan data diri mahasiswa (Gambar 3.4).



Gambar 3.4: Halaman Profil

3.1.4 Jadwal

Menu Jadwal terdiri dari beberapa submenu:

- Kehadiran

Submenu ini berfungsi untuk menandakan kehadiran mahasiswa di suatu mata kuliah pada hari dimana mahasiswa tersebut mengakses halaman tersebut (Gambar 3.5).

Gambar 3.5: Halaman Kehadiran

- Ketidakhadiran

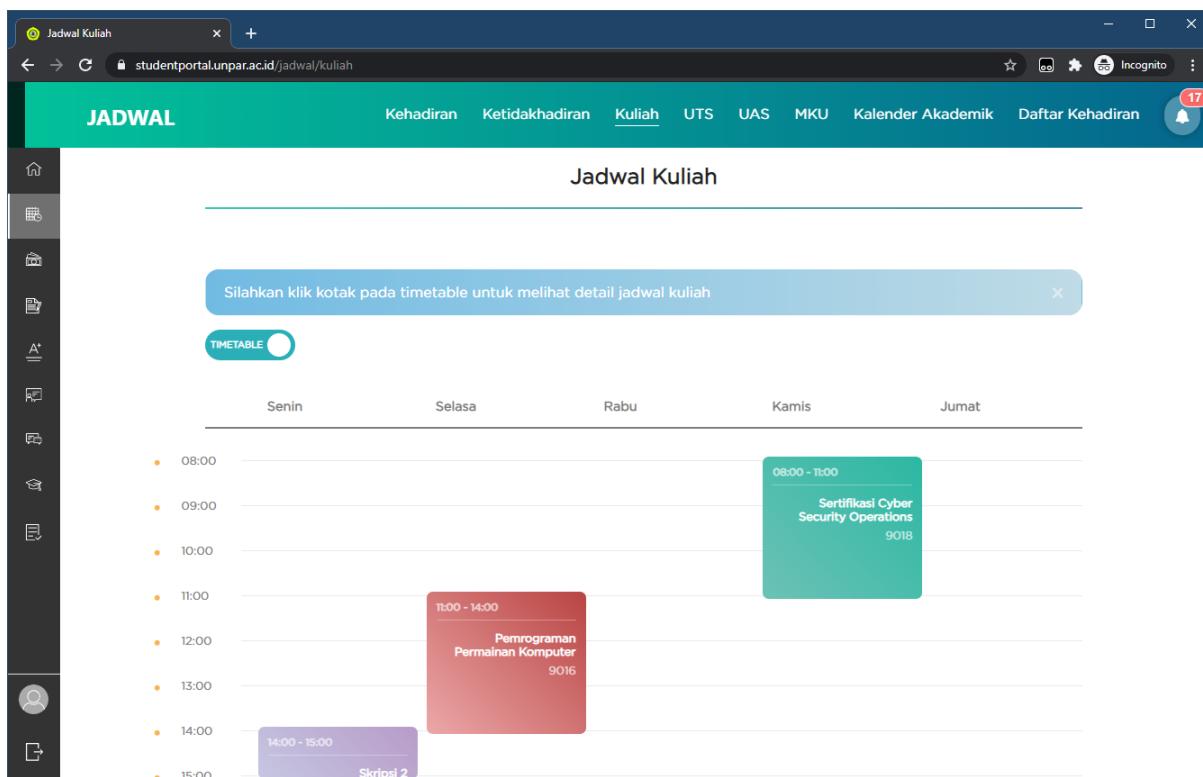
Submenu ini berfungsi untuk mengunggah surat sakit atau surat izin mahasiswa. (Gambar 3.6).

The screenshot shows a web browser window titled 'Ketidakhadiran' with the URL 'studentportal.unpar.ac.id/jadwal/dispensasi_kehadiran'. The page has a green header bar with the word 'JADWAL' in white. Below the header, there's a navigation menu with several icons. The main content area is titled 'Ketidakhadiran' and contains a form for uploading absence documents. It includes fields for 'Alasan Ketidakhadiran' (Reason for Absence) with radio buttons for 'Sakit' (selected) and 'Izin', a date input field for 'Tanggal Sakit/Izin', a file upload input for 'Unggah Surat Sakit/Izin', a browse button, and a note about PDF file size. A green 'Simpan & Kirim' (Save & Send) button is present. At the bottom, there's a table with columns: No, Tanggal Sakit/Izin (Tanggal Awal, Tanggal Akhir), Jenis Ketidakhadiran, Status, Dokumen, and Hapus. The table message says 'No data available in table'. There are also 'Show 10 entries' and 'Search' buttons.

Gambar 3.6: Halaman Ketidakhadiran

- Kuliah

Submenu ini berisi tentang jadwal kuliah yang dapat disusun per semester dan terdapat 2 tampilan, yaitu tabel waktu (Gambar 3.7) dan tabel biasa (Gambar 3.8).



Gambar 3.7: Halaman Jadwal Kuliah Dalam Tabel Waktu

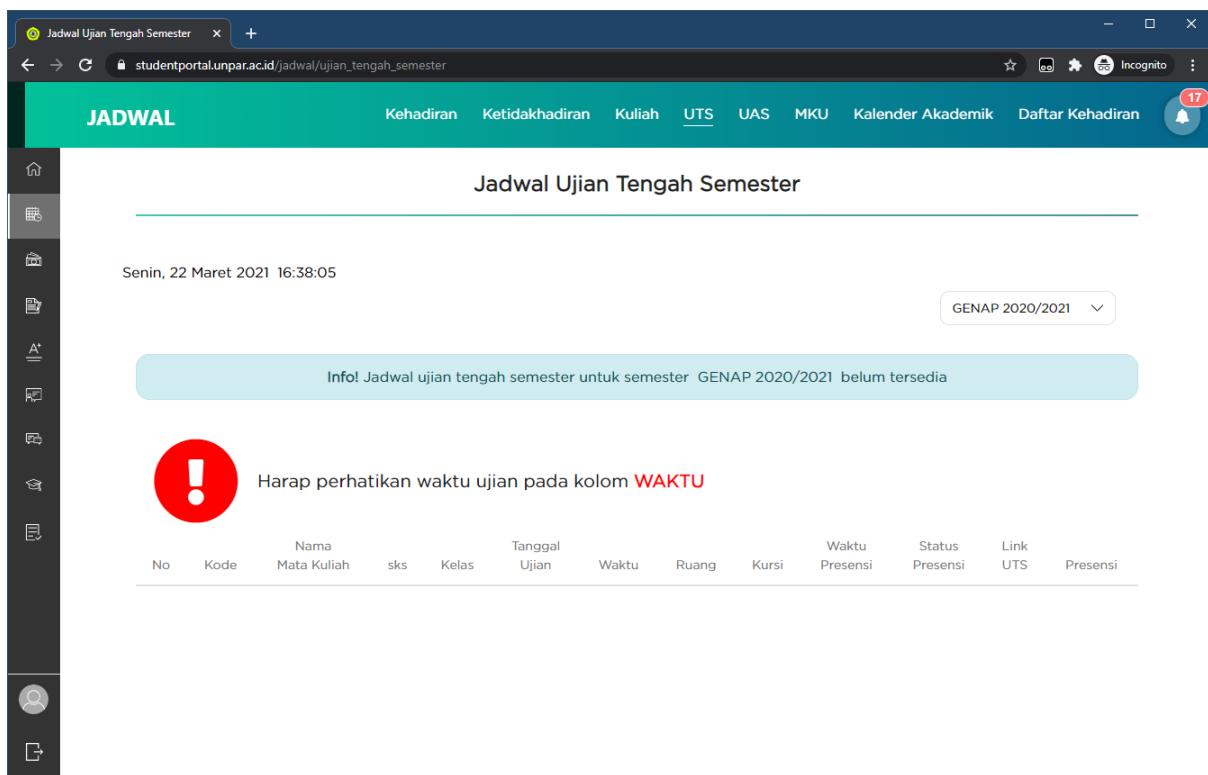
The screenshot shows the same student portal interface as in Gambar 3.7, but with the 'TABLE' view selected. The main content area displays a table of scheduled classes:

Hari	Waktu	Kode	Ruang	Nama	SKS	Kelas	Nama Dosen	Temu
Senin	14:00-15:00	AIF184002	Ruang Kuliah 9120	Skripsi 2	5	A	• Mariska Tri Adithia, S.Si., M.Sc., PDEng.	1
Selasa	11:00-14:00	AIF183120	Ruang 9016 Lab. Komputer 3	Pemrograman Permainan Komputer	3	A	• Jefvin Viriya, S.T.	1
Kamis	08:00-11:00	AIF183240	Ruang 9016 Lab Komputer 1	Sertifikasi Cyber Security Operations	3	A	• Chandra Wijaya, S.T., M.T.	1

Gambar 3.8: Halaman Jadwal Kuliah Tabel

- UTS

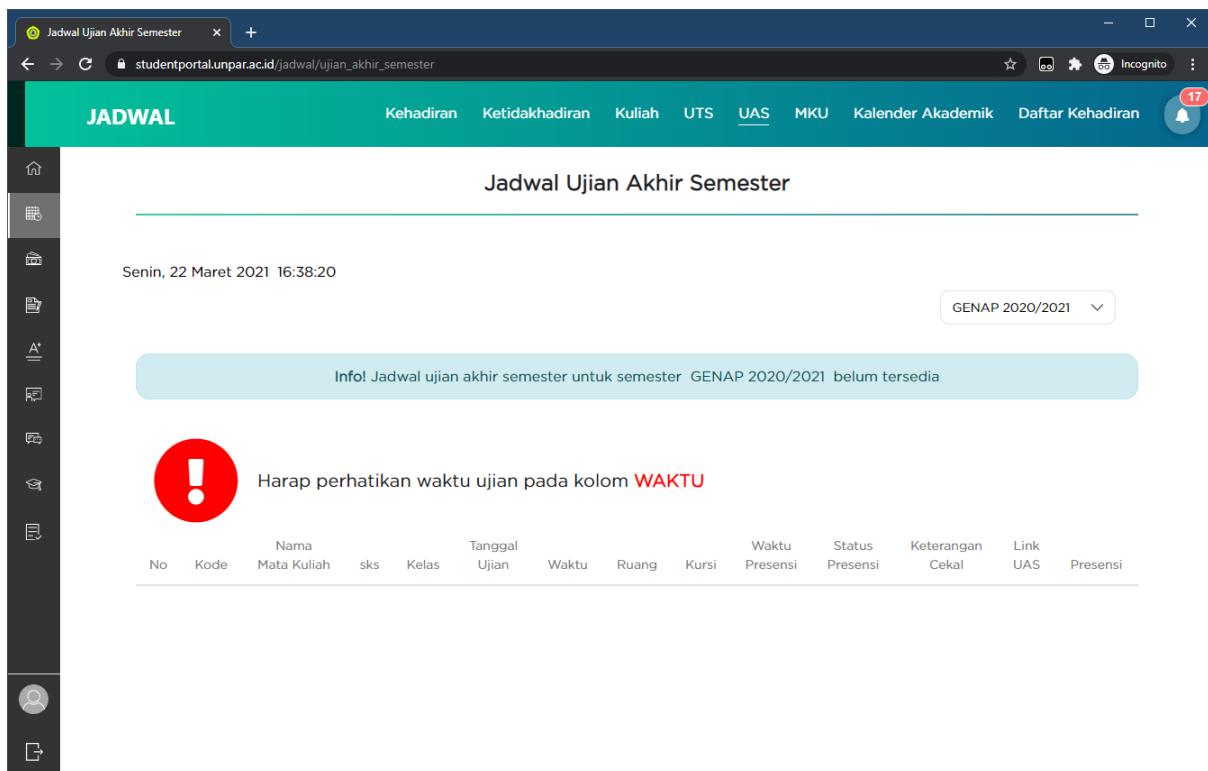
Submenu ini berisi tentang UTS yang dapat disusun per semester (Gambar 3.9).



Gambar 3.9: Halaman UTS

- UAS

Submenu ini berisi tentang UAS yang dapat disusun per semester (Gambar 3.10).



Gambar 3.10: Halaman UAS

- MKU

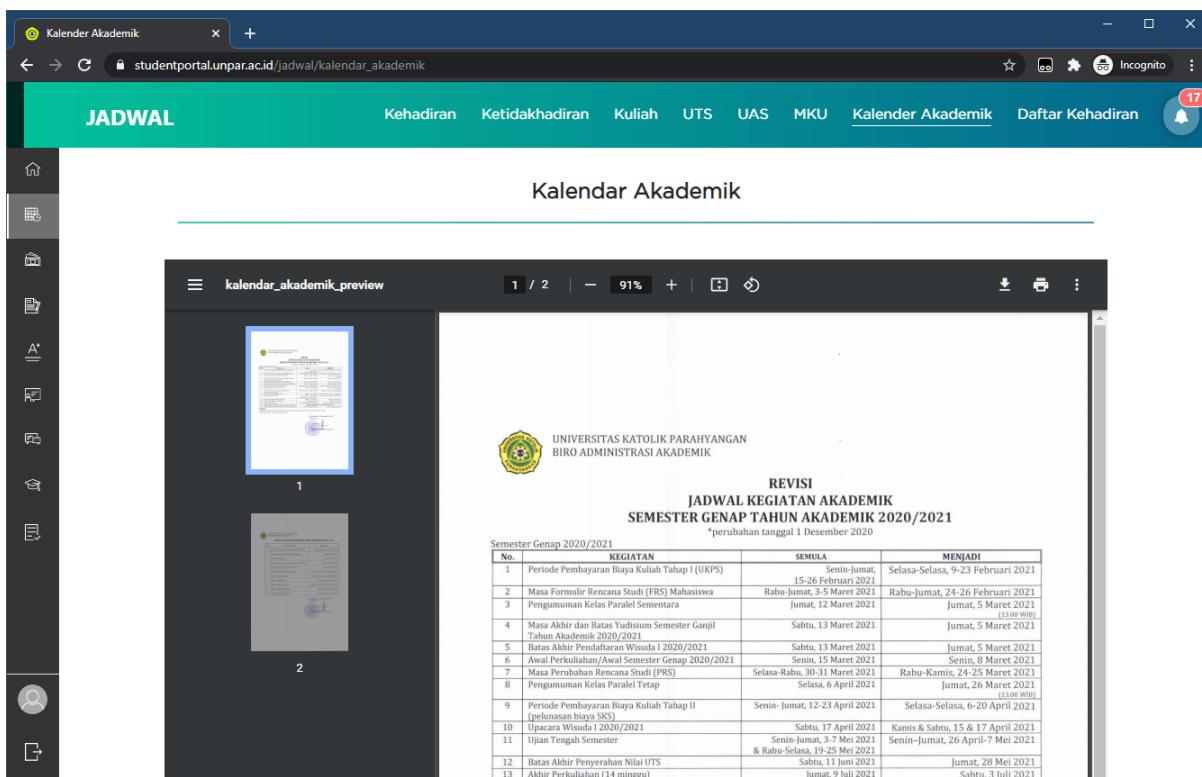
Submenu ini menampilkan seluruh jadwal Mata Kuliah Umum (MKU) yang memberikan informasi tentang kelas-kelas yang dibuka oleh Pusat Kajian Humaniora (PKH) (Gambar 3.11).

The screenshot shows a web browser window titled "Jadwal Mata Kuliah Umum". The URL in the address bar is "studentportal.unpar.ac.id/jadwal/mata_kuliah_umum". The main content area is titled "Jadwal Mata Kuliah Umum" and displays a message: "Silahkan klik Kode Mata Kuliah untuk melihat jadwal." Below this message is a list of course codes with their descriptions: MKU180110 - Pendidikan Kewarganegaraan, MKU180120 - Logika, MKU180130 - Bahasa Indonesia, MKU180240 - Etika Dasar, MKU180250 - Pendidikan Pancasila, MKU180360 - Estetika, MKU180370 - Pendidikan Agama Katolik, and MKU180380 - Fenomenologi Agama. To the left of the main content is a vertical sidebar with various icons: a home icon, a calendar icon, a document icon, a magnifying glass icon, a person icon, and a refresh/circular arrow icon.

Gambar 3.11: Halaman MKU

- Kalender Akademik

Submenu ini menampilkan informasi mengenai kalender akademik UNPAR (Gambar 3.12).



Gambar 3.12: Halaman Kalender Akademik

- Daftar Kehadiran

Submenu ini menampilkan informasi mengenai daftar kehadiran mahasiswa pada setiap mata kuliah yang dapat disusun per semester (Gambar 3.13).

No	Mata Kuliah	Dosen Pengajar	Sesi	Proses Pembelajaran	Hadir	Sakit	Izin	Alpa	Total Pertemuan Terverifikasi	Persentase Pembelajaran	Persentase Keseluruhan
1	AIF183120-03 Pemrograman Permainan Komputer	• Jefvin Viriya, S.T.	• Sesi 1: Selasa , 11:00 -14:00, Ruang 9016 Lab. Komputer 3	Kuliah	2	0	0	0	2	100%	100%
2	AIF183240-03 Sertifikasi Cyber Security Operations	• Chandra Wijaya, S.T., M.T.	• Sesi 1: Kamis , 08:00 -11:00, Ruang 9018 Lab Komputer 1	Kuliah	1	0	0	0	1	100%	100%
3	AIF184002-05 Skripsi 2	• Mariska Tri Adithia, S.Si., M.Sc., PDEng.	• Sesi 1: Senin , 14:00 -15:00, Ruang Kuliah 9120	Kuliah	1	0	0	0	1	100%	100%

Gambar 3.13: Halaman Daftar Kehadiran

3.1.5 Pembayaran

Menu ini berfungsi untuk melihat data tagihan pembayaran uang kuliah, riwayat pembayaran, dan keterangan cara-cara pembayaran uang kuliah yang dapat disusun per semester (Gambar 3.14).

The screenshot shows a web browser window titled 'Pembayaran' from the URL 'studentportal.unpar.ac.id/pembayaran'. The main title is 'PEMBAYARAN' and the sub-section is 'Tagihan Pembayaran GENAP 2020/2021'. A dropdown menu shows 'GENAP 2020/2021'. On the left is a vertical sidebar with icons for home, documents, grades, and search. The main content table lists tuition fees:

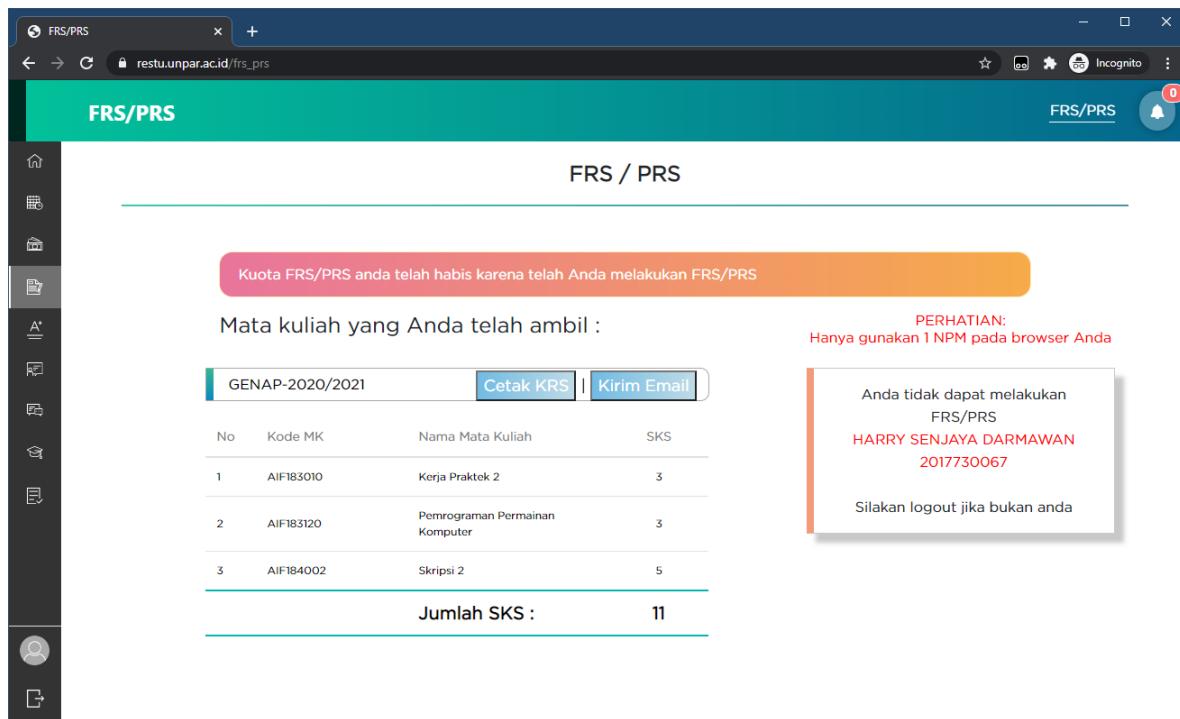
Jenis Tagihan	Jumlah Tagihan
HUTANG SEBELUMNYA	Rp. 0,-
Tahap 01	Rp. 6.000.000,-
Denda Tahap 01	Rp. 0,-
Tahap 02	Rp. 0,-
Denda Tahap 02	Rp. 0,-
PENAMBAHAN	Rp. 0,-
PENGEMBALIAN	Rp. 0,-
TOTAL	Rp. 6.000.000,-

At the bottom is a link 'Riwayat Pembayaran'.

Gambar 3.14: Halaman Pembayaran

3.1.6 FRS/PRS

Menu ini berfungsi sebagai formulir pengisian rencana studi awal (FRS), perubahan rencana studi (PRS) dan menampilkan informasi mata kuliah yang telah diambil saat FRS atau PRS (Gambar 3.15).



Gambar 3.15: Tampilan FRS/PRS

3.1.7 Nilai

Menu Nilai terdiri dari beberapa submenu:

- Nilai per Semester

Submenu ini menampilkan informasi nilai per semester. Mahasiswa dapat melihat nilai sesuai dengan semester yang dipilih (Gambar 3.16).

Nilai per Semester

GANJIL 2020/2021

No	Kode Mata Kuliah	Nama Mata Kuliah	SKS	Kelas	Nilai	AA	NA	Statistik Hasil Nilai
1	AIF183119	Keamanan Informasi	2	A	Tampilkan Detail Nilai	-	-	Tampilkan Grafik Nilai
2	AIF183341	Pola Komputasi Big Data	3	A	Tampilkan Detail Nilai	-	-	Tampilkan Grafik Nilai
3	AIF183348	Sistem Kecerdasan Bisnis	3	A	Tampilkan Detail Nilai	-	-	Tampilkan Grafik Nilai
4	AIF184001	Skripsi 1	3	A	Tampilkan Detail Nilai	-	-	Tampilkan Grafik Nilai
5	AIF184005	Komputer dan Masyarakat	2	A	Tampilkan Detail Nilai	-	-	Tampilkan Grafik Nilai
6	AIF184235	Layanan Berbasis Web	3	A	Tampilkan Detail Nilai	-	-	Tampilkan Grafik Nilai
7	AIF184303	Proyek Sistem Informasi 2	3	A	Tampilkan Detail Nilai	-	-	Tampilkan Grafik Nilai

Keterangan:

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

Gambar 3.16: Halaman Nilai Per Semester

- Daftar Perkembangan Studi

Submenu ini menampilkan seluruh riwayat mata kuliah dan nilai yang pernah ditempuh

mahasiswa (Gambar 3.17). Submenu ini juga menampilkan statistik sks, nilai, dan indeks prestasi mahasiswa (Gambar 3.18).

The screenshot shows a web-based application titled 'Daftar Perkembangan Studi'. The main content area is titled 'Daftar Perkembangan Studi' and displays a table of courses taken by the student. The table has columns for Kode MK (Course Code), Nama MK (Course Name), Nilai (Grade), and Tahun Sem (Year Semester). The table is divided into two sections: Semester 1 and Semester 2.

Kode MK	Nama MK	Nilai	Tahun Sem
AIF131105	Pengantar Informatika	A	2017/1
AIF181101	Pemodelan untuk Komputasi		
AIF181103	Matematika Dasar	A	2017/2
AIF181105	Pengantar Informatika		
AIF181107	Matematika Diskret	A	2017/1
MKU180110	Pendidikan Kewarganegaraan	A-	2018/1
MKU180120	Logika	A	2018/1
MKU180130	Bahasa Indonesia		
Semester 2			
AIF131101	Pemrograman Berorientasi Objek	A	2017/1
AIF32205	Arsitektur Komputer	B	2017/2
AIF181100	Dasar-dasar Pemrograman		
AIF181104	Logika Informatika	A	2017/2

Gambar 3.17: Halaman Daftar Perkembangan Studi (1)

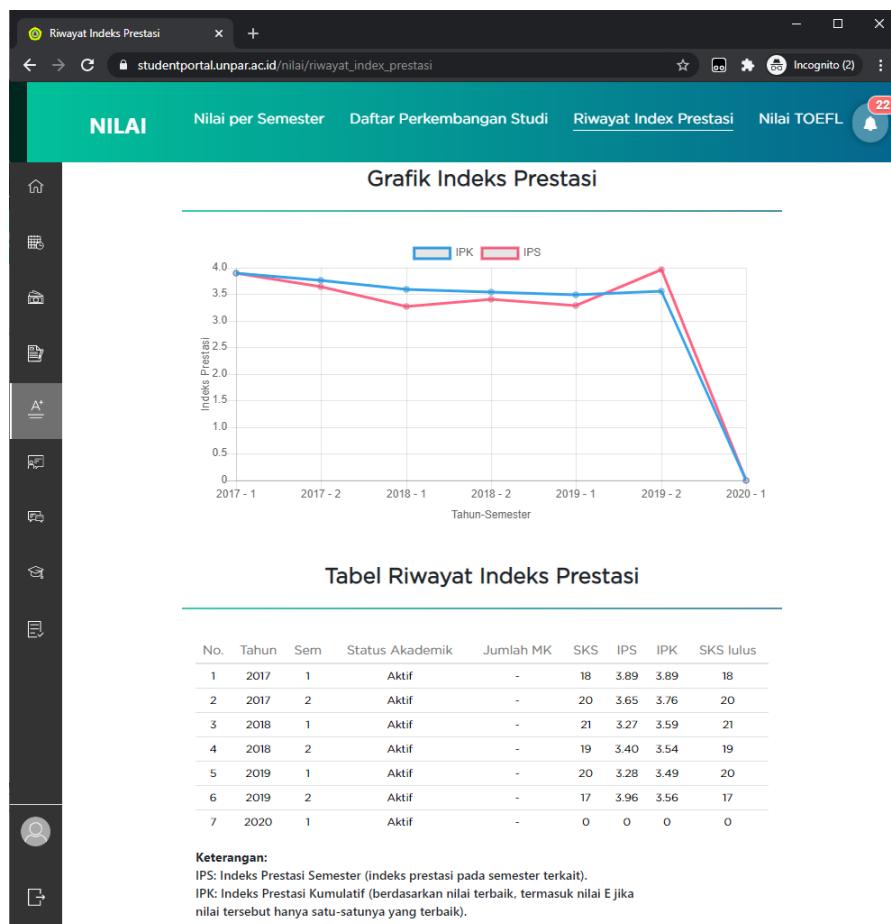
This screenshot shows the same application interface as in Gambar 3.17, but it is focused on a single course entry for 'MKU130011 Estetika' from the 'Nilai per Semester' tab. It includes detailed information such as the semester code, grade point average (IPK), and final grade (IPS). Below this, there are sections for 'Nilai Akhir' (Final Grade) and 'Nilai TOEFL' (TOEFL Score), along with various academic statistics and requirements.

MKU130011	Estetika	B	2017/2																														
Kode Semester: 1 = Ganjil, 2 = Genap, 4 = Padat, 6 = Transfer	Ket. Mk. Bid. Peminatan: 01 = Teknologi Informasi Bisnis 02 = Ilmu Komputer 03 = Telematika	Ket. Mt Kuliah: • [M] = Mk. Kendali Mutu • [X] = Mk. Disisisikan • ["] = Mk. yang diambil semester (2017-2)																															
Nilai Akhir																																	
<table border="1"> <tr><td>Nilai</td><td>A</td><td>A-</td><td>B+</td><td>B</td><td>B-</td><td>C+</td><td>C</td><td>D</td><td>E</td></tr> <tr><td>Akhir</td><td>20</td><td>4</td><td>8</td><td>5</td><td>1</td><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>Mata Kuliah</td><td>59</td><td>10</td><td>21</td><td>12</td><td>4</td><td>6</td><td>3</td><td>0</td><td>0</td></tr> </table>				Nilai	A	A-	B+	B	B-	C+	C	D	E	Akhir	20	4	8	5	1	2	1	0	0	Mata Kuliah	59	10	21	12	4	6	3	0	0
Nilai	A	A-	B+	B	B-	C+	C	D	E																								
Akhir	20	4	8	5	1	2	1	0	0																								
Mata Kuliah	59	10	21	12	4	6	3	0	0																								
Jumlah sks																																	
<table border="1"> <tr><td>Ditempuh</td><td>: 115 sks</td></tr> <tr><td>Lulus Wajib</td><td>: 83 sks</td></tr> <tr><td>Lulus Pilihan</td><td>: 32 sks</td></tr> <tr><td>Lulus Wajib Peminatan</td><td>: 0 sks</td></tr> <tr><td>Lulus Pilihan Peminatan</td><td>: 0 sks</td></tr> </table>				Ditempuh	: 115 sks	Lulus Wajib	: 83 sks	Lulus Pilihan	: 32 sks	Lulus Wajib Peminatan	: 0 sks	Lulus Pilihan Peminatan	: 0 sks																				
Ditempuh	: 115 sks																																
Lulus Wajib	: 83 sks																																
Lulus Pilihan	: 32 sks																																
Lulus Wajib Peminatan	: 0 sks																																
Lulus Pilihan Peminatan	: 0 sks																																
Nilai TOEFL																																	
No	Tanggal	Skor																															
1	05-10-2020	540																															
<table border="1"> <tr><td>Total Lulus</td><td>: 115 sks</td></tr> <tr><td>Ditempuh Semester ini</td><td>: 17 sks</td></tr> <tr><td colspan="2">Diizinkan untuk semester yad.: 24 sks</td></tr> <tr><td>Cuti Studi</td><td>: 0 semester</td></tr> <tr><td>Akhir masa Studi</td><td>: SEMESTER GENAP 2023/2024</td></tr> </table>				Total Lulus	: 115 sks	Ditempuh Semester ini	: 17 sks	Diizinkan untuk semester yad.: 24 sks		Cuti Studi	: 0 semester	Akhir masa Studi	: SEMESTER GENAP 2023/2024																				
Total Lulus	: 115 sks																																
Ditempuh Semester ini	: 17 sks																																
Diizinkan untuk semester yad.: 24 sks																																	
Cuti Studi	: 0 semester																																
Akhir masa Studi	: SEMESTER GENAP 2023/2024																																
<p>Syarat Kelulusan Lulus min. 144 sks terdiri dari : Mk. Wajib + Mk. Pilihan I.P. Lulus minimum : 2.00</p>																																	

Gambar 3.18: Halaman Daftar Perkembangan Studi (2)

- Riwayat Indeks Prestasi

Submenu ini menampilkan seluruh riwayat Indeks Prestasi Semester (IPS) dan Indeks Prestasi Kumulatif (IPK) setiap semester mahasiswa (Gambar 3.19).



Gambar 3.19: Halaman Riwayat Indeks Prestasi

- Nilai TOEFL

Submenu ini menampilkan seluruh riwayat skor dan detail skor *Test of English as Foreign Language* (TOEFL) yang pernah ditempuh mahasiswa (Gambar 3.20).

The screenshot shows a web browser window titled 'Nilai TOEFL' from the URL 'studentportal.unpar.ac.id/nilai/toefl'. The page has a teal header bar with the word 'NILAI' and several navigation links: 'Nilai per Semester', 'Daftar Perkembangan Studi', 'Riwayat Index Prestasi', and 'Nilai TOEFL'. On the left, there's a vertical sidebar with various icons. The main content area is titled 'TOEFL' and displays a table of results:

No	Tanggal	Listening	Structure	Reading	Total
1	05-10-2020	167	181	192	540

A red box highlights the 'Total' column value '540'. Below the table, a section titled 'Apa itu TOEFL?' contains text about the test and its structure.

Gambar 3.20: Halaman Nilai TOEFL

3.1.8 Angket

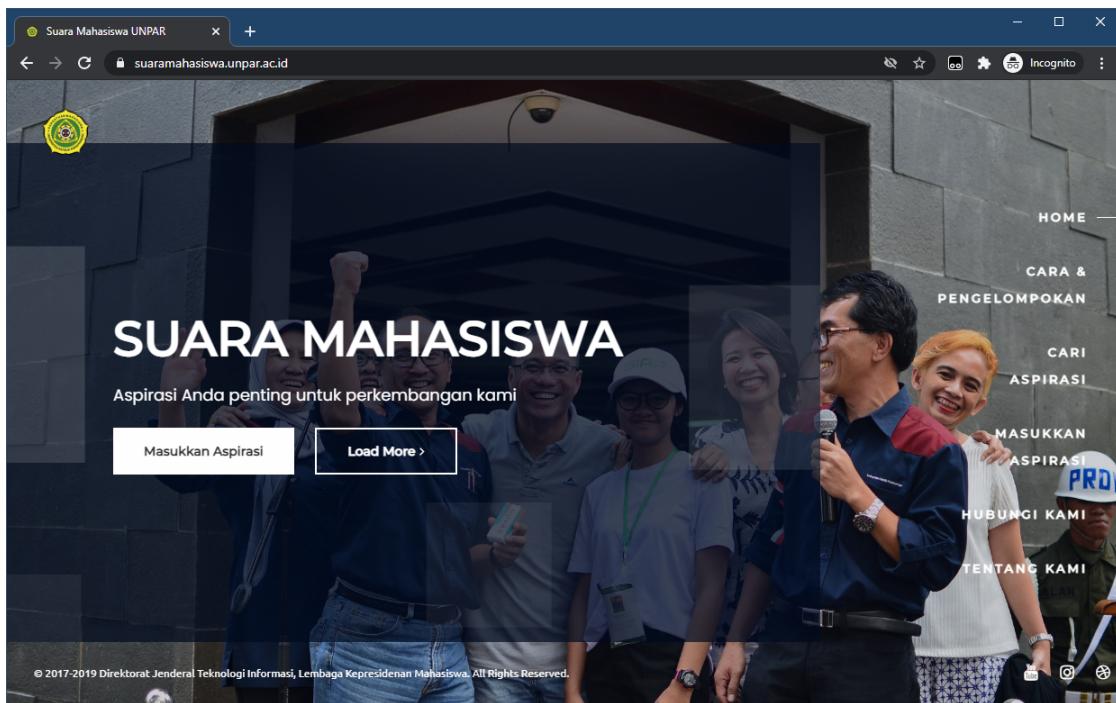
Menu Angket merupakan halaman dimana mahasiswa diminta untuk mengisi angket dari para dosen yang mengajarnya di semester yang sedang ditempuh mahasiswa tersebut (Gambar 3.21).

The screenshot shows a web browser window titled 'Angket Dosen Universitas Katolik' from the URL 'angketedosen.unpar.ac.id/listAngket'. The page has a dark blue header bar with the title 'Angket Dosen' and a user ID '2017730067'. On the left, there's a sidebar with a profile picture, the user ID, and a 'List Angket' button. The main content area shows a message: 'Senin, 22 Maret 2021 | 16:40:59 WIB' and 'Daftar Angket Matakuliah'. Below it, a red message says 'Tidak ada periode angket yang aktif'.

Gambar 3.21: Halaman Angket

3.1.9 Saran & Komentar

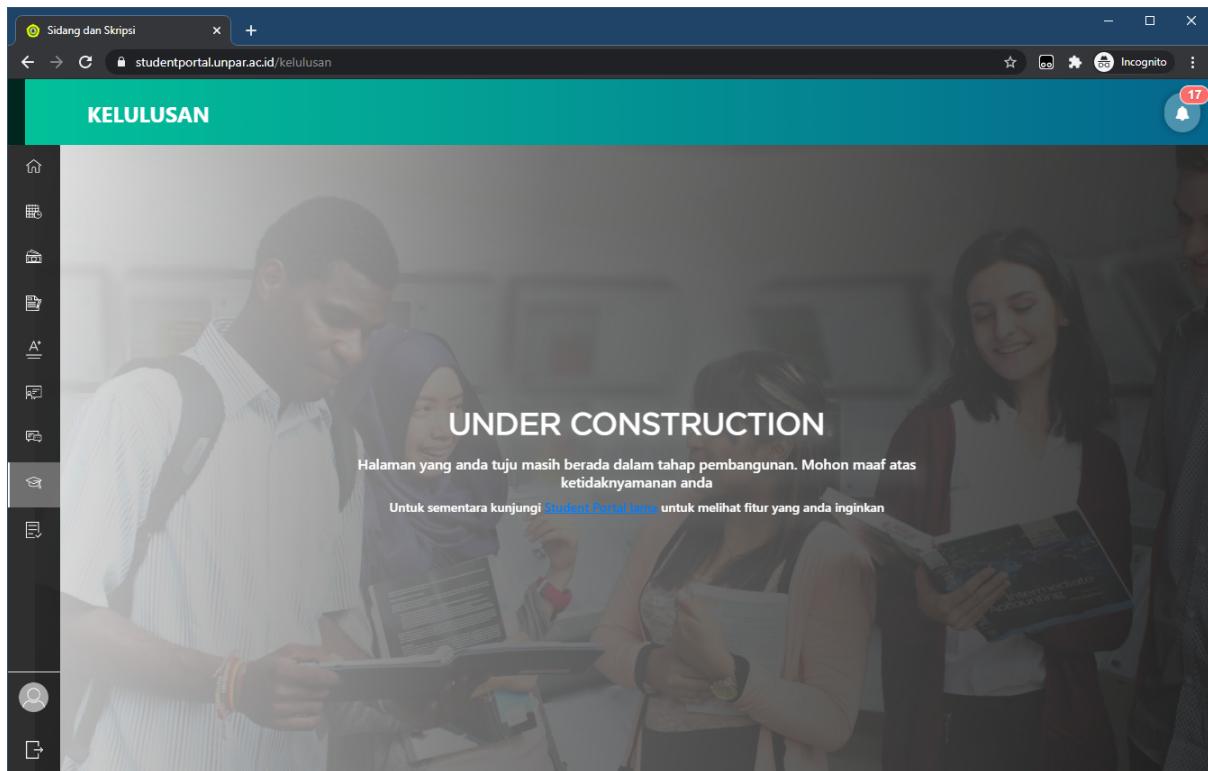
Menu Saran & Komentar akan membuka halaman <https://suaramahasiswa.unpar.ac.id/> (Gambar 3.22).



Gambar 3.22: Halaman Saran & Komentar

3.1.10 Kelulusan

Menu Kelulusan sedang dalam tahap pembangunan (Gambar 3.23).



Gambar 3.23: Halaman Kelulusan

3.1.11 Pengajuan

Menu Pengajuan merupakan halaman dimana mahasiswa dapat mengajukan topik skripsi atau tugas akhir (Gambar 3.24).

No	Semester	Mata Kuliah	Tanggal Pengajuan	Usulan Topik	Dosen Pembimbing		Status Pengajuan	Cetak Kartu Pembimbing	Edit
					Usulan	Ditunjuk			
1	2020/2021 Genap	AIF184002 - Skripsi 2						<button>Cetak</button>	<button>Edit</button>

Gambar 3.24: Halaman Pengajuan

3.2 Analisis SIAKAD

Subbab ini ditulis oleh dosen pembimbing.

SIAKAD adalah sistem informasi yang disediakan oleh Biro Teknologi Informasi kepada staf UNPAR, untuk menangani hal-hal yang berkaitan dengan akademik. SIAKAD dapat diakses pada alamat <https://siakad.unpar.ac.id> dari lingkungan jaringan UNPAR atau melalui VPN (*Virtual Private Network*). Modul-modul yang ditampilkan pada SIAKAD dapat berbeda, bergantung pada peran pengguna yang login. Pada subbab ini, akan dijabarkan modul-modul yang ditampilkan kepada dosen.

3.2.1 Login

Tata cara login untuk mengakses SIAKAD tidak jauh berbeda dengan pada Portal Akademik Mahasiswa. Dosen atau staf diarahkan ke situs web SSO (Gambar 3.1 dan 3.2). Perbedaannya hanyalah bahwa SIAKAD membatasi hanya akun dosen atau staf yang diperbolehkan masuk, berdasarkan alamat e-mailnya.

3.2.2 Mahasiswa

Untuk dosen, hanya ada satu submenu dari menu Mahasiswa, yaitu “Cari Mahasiswa”. Halaman ini memungkinkan dosen untuk mendapatkan daftar mahasiswa waliinya. Untuk setiap mahasiswa, ada tiga tautan yang dapat diklik, yaitu: NPM (untuk membuka pop-up Detail Mahasiswa), Data Diri, dan Data Akademik. Gambar 3.25 menunjukkan tampilan daftar mahasiswa wali dosen.

No	NPM	Nama Mahasiswa	Dosen Wali	Program Studi	Bidang Peminatan	Status Akademik	Aksi
1	2016730014	RICHARD WIJAYA		Teknik Informatika		Gencat	Data Diri Data Akademik
2	2016730068	GABRIEL PANJI LAZUARDI		Teknik Informatika		Aktif	Data Diri Data Akademik
3	2016730081	JONATHAN LAKSAMANA PURNOMO		Teknik Informatika		Aktif	Data Diri Data Akademik
4	2017730005	DIONISIUS SALVAVICTORI WANGGUR		Teknik Informatika		Aktif	Data Diri Data Akademik
5	2017730015	DAVID CHRISTOPHER SENTOSA		Teknik Informatika		Aktif	Data Diri Data Akademik
6	2017730016	STEPHEN HADI		Teknik Informatika		Aktif	Data Diri Data Akademik
7	2017730018	NICHOLAS ADITYA HALIM		Teknik Informatika		Aktif	Data Diri Data Akademik
8	2017730028	JOSHUA DELAVO SETIADI		Teknik Informatika		Aktif	Data Diri

Gambar 3.25: Tangkapan Layar Halaman Daftar Mahasiswa Wali

Detail Mahasiswa Detail ringkas mahasiswa ditampilkan dalam bentuk pop-up (Gambar 3.26) sehingga dosen wali dapat melihat secara sekilas data mahasiswa satu per satu tanpa perlu berpindah ke halaman lain.

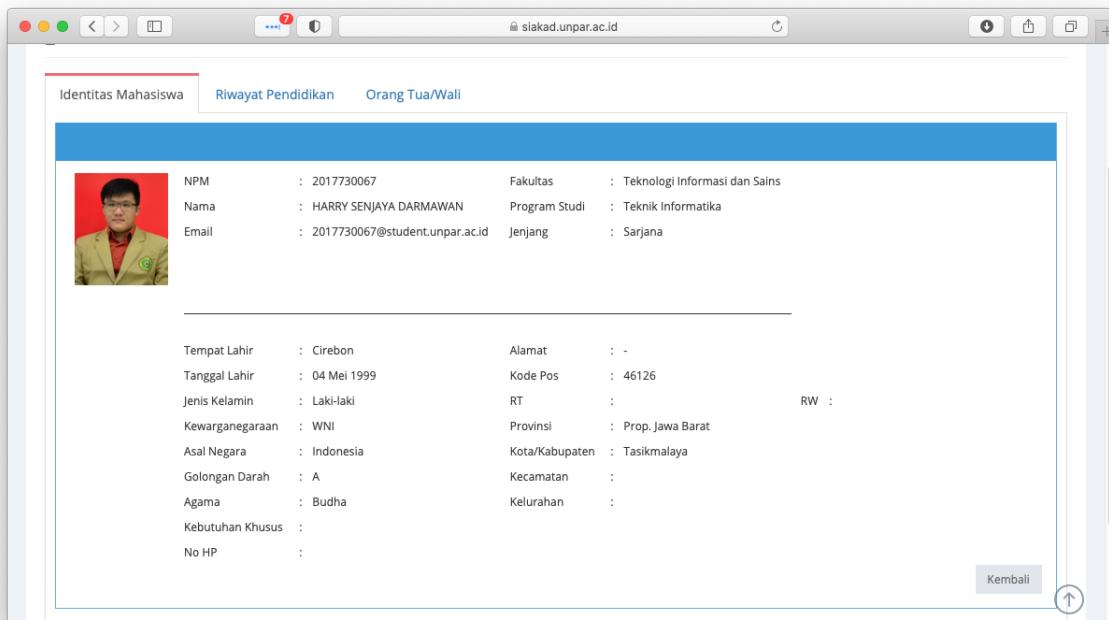
Detail Mahasiswa					
Mahasiswa		Info Status Akademik & Pembayaran			
		NPM :2017730067 Nama :HARRY SENJAYA DARMAWAN Email :2017730067@student.unpar.ac.id Telp/Hp :			
		Fakultas :Teknologi Informatika dan Sains Program Studi :Teknik Informatika Jenjang :Sarjana Dosen Wali :-			
		Email Dosen :			
Info Status Akademik & Pembayaran					
No	Tahun Akademik	Semester Akademik	Status Akademik	Status Keuangan	
				Tahap 1	Tahap 2
1	2020	Genap	Aktif	Sudah Bayar	Belum Bayar
2	2020	Ganjil	Aktif	Sudah Bayar	Sudah Bayar
3	2019	Pendek	Gencat		Sudah Bayar
4	2019	Genap	Aktif	Sudah Bayar	Sudah Bayar
5	2019	Ganjil	Aktif	Sudah Bayar	Sudah Bayar
6	2018	Genap	Aktif	Sudah Bayar	Sudah Bayar
7	2018	Ganjil	Aktif	Sudah Bayar	Sudah Bayar
8	2017	Pendek	Gencat		Sudah Bayar
9	2017	Genap	Aktif		
10	2017	Ganjil	Aktif		

Gambar 3.26: Tangkapan Layar Pop-up Detail Mahasiswa

Data Diri

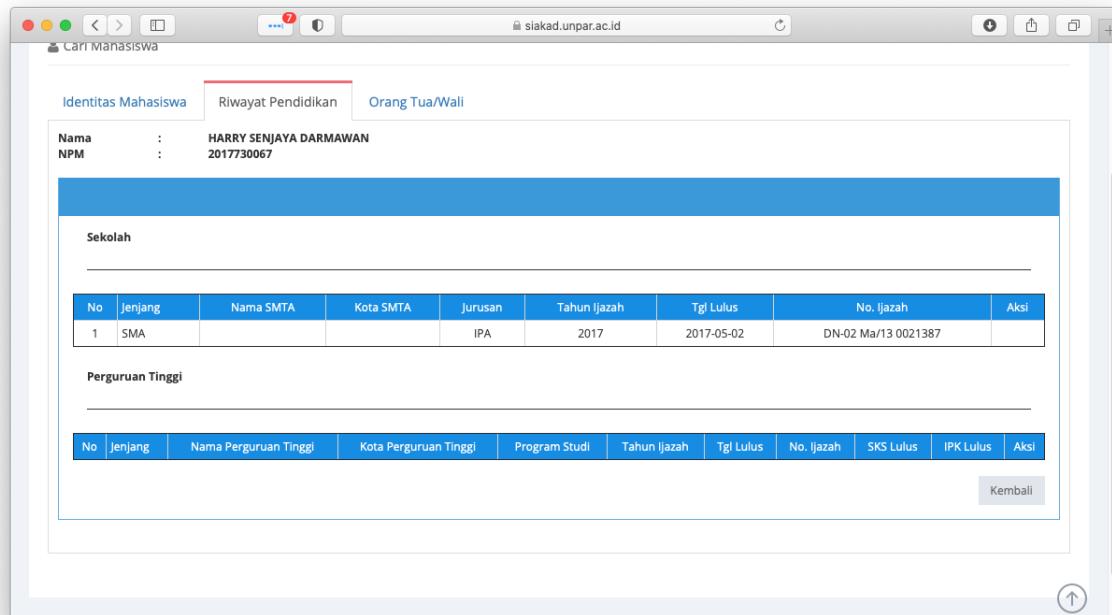
Data diri mahasiswa ditampilkan pada halaman baru yang terdiri dari beberapa tab. Halaman ini berisi detail data diri / pribadi mahasiswa yang tidak terkait data akademik di UNPAR. Halaman ini terdiri dari tiga tab, yaitu Identitas Mahasiswa, Riwayat Pendidikan, dan Orang Tua/Wali.

Identitas Mahasiswa Tab ini berisi data identitas mahasiswa, seperti foto, tempat/tanggal lahir, jenis kelamin, kewarganegaraan, serta data pribadi lainnya (Gambar 3.27).



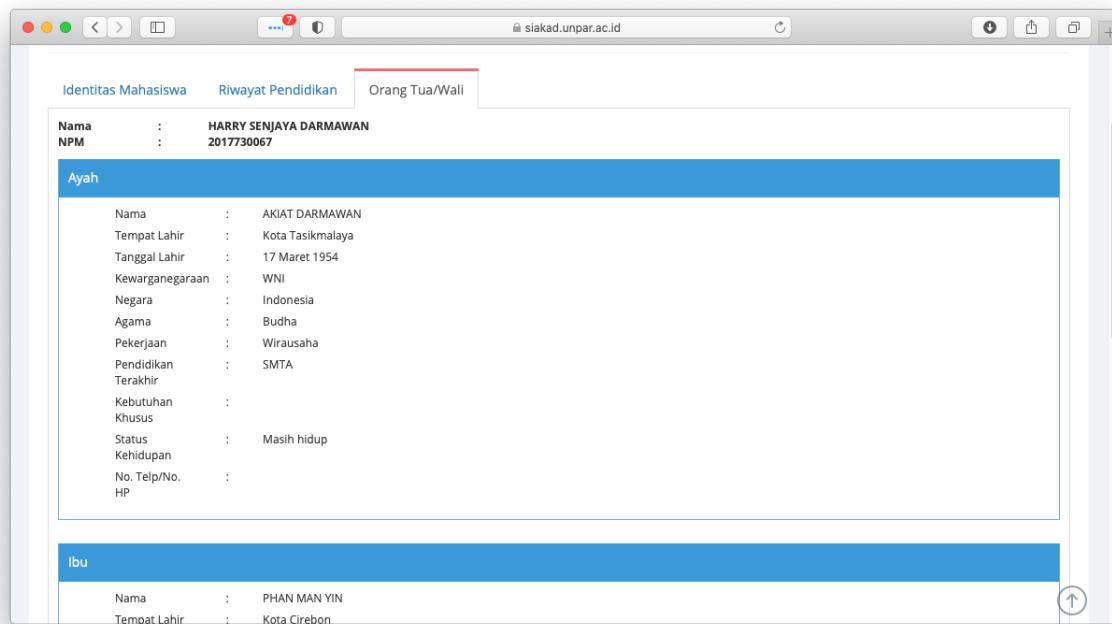
Gambar 3.27: Tangkapan Layar Tab Identitas Mahasiswa

Riwayat Pendidikan Tab ini berisi riwayat pendidikan mahasiswa sebelum berkuliahan di UNPAR, baik SMA maupun di perguruan tinggi lain (Gambar 3.28).



Gambar 3.28: Tangkapan Layar Tab Riwayat Pendidikan

Orang Tua/Wali Tab ini berisi data identitas orang tua atau wali mahasiswa, seperti nama, tempat/tanggal lahir, dan sebagainya (Gambar 3.29).

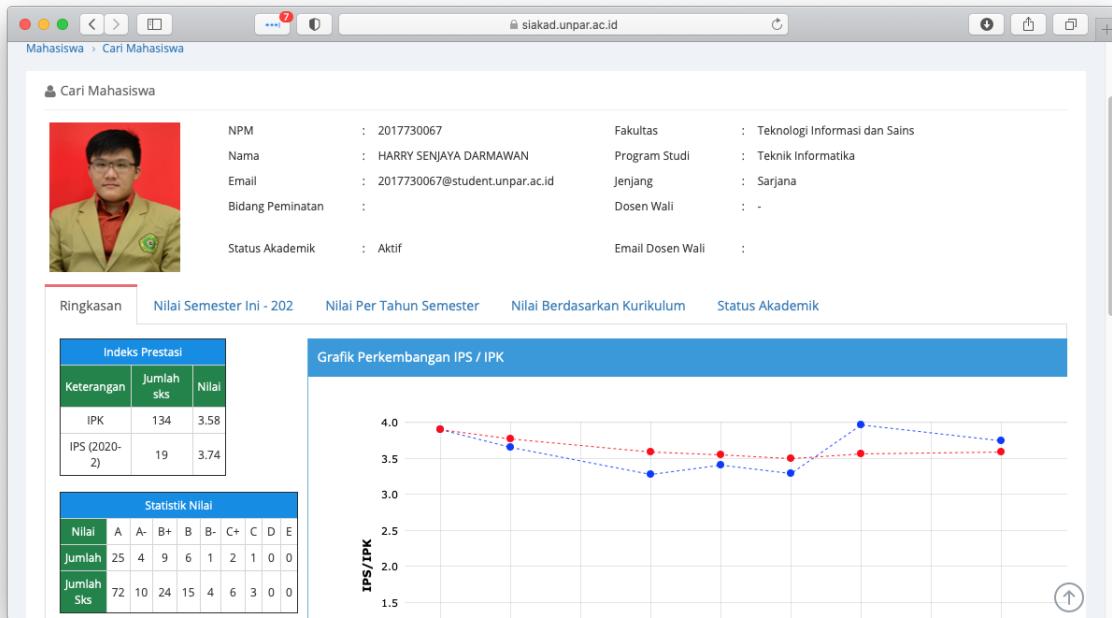


Gambar 3.29: Tangkapan Layar Tab Orang Tua/Wali

Data Akademik

Data akademik mahasiswa ditampilkan pada halaman baru yang terdiri dari beberapa tab. Halaman ini berisi detail data akademik di UNPAR. Halaman ini terdiri dari lima tab, yaitu Ringkasan, Nilai Semester Ini, Nilai Per Tahun Semester, Nilai Berdasarkan Kurikulum, dan Status Akademik.

Ringkasan Tab ini berisi ringkasan data akademik mahasiswa, seperti statistik nilai, IPS, IPK, serta grafik perkembangannya (Gambar 3.30).



Gambar 3.30: Tangkapan Layar Tab Ringkasan Akademik

Nilai Semester Ini Tab ini berisi data nilai dari kuliah-kuliah yang sedang diambil pada semester ini (Gambar 3.31).

The screenshot shows a student profile with the following details:

- NPM:** 2017730067
- Nama:** HARRY SENJAYA DARMAWAN
- Email:** 2017730067@student.unpar.ac.id
- Bidang Peminatan:** -
- Status Akademik:** Aktif
- Fakultas:** Teknologi Informasi dan Sains
- Program Studi:** Teknik Informatika
- Jenjang:** Sarjana
- Dosen Wali:** -
- Email Dosen Wali:** -

Below the profile, there are tabs: Ringkasan, Nilai Semester Ini - 202, Nilai Per Tahun Semester, Nilai Berdasarkan Kurikulum, and Status Akademik. The "Nilai Semester Ini - 202" tab is selected, displaying the following table:

No.	Kode MK	Nama Matakuliah	sks	Kelas	Kelompok Tugas																				Kelompok Ujian	AA	NA
					TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	TP11	TP12	TP13	TP14	TP15	TP16	TP17	TP18	TP19	TP20			
1	AIF183010	Kerja Praktek 2	3																		0	0					
2	AIF183120	Pemrograman Permainan Komputer	3	A																	0	0					
3	AIF184002	Skripsi 2	5	A																	0	0					
Total sks					11																						

Keterangan: Jika nilai tidak muncul mungkin dikarenakan Anda belum melunasi seluruh pembayaran.

Kembalil

Gambar 3.31: Tangkapan Layar Tab Nilai Semester Ini

Nilai Per Tahun Semester Tab ini berisi data nilai mahasiswa, dibagi per tahun semester (Gambar 3.32).

The screenshot shows the "Nilai Per Tahun Semester" tab, displaying data for two academic years:

- Tahun Akademik 2020/2021** (Semester Genap):

No.	Kode MK	Nama Matakuliah	sks	Kelas	Kelompok Tugas																				Kelompok Ujian	AA	NA
					TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	TP11	TP12	TP13	TP14	TP15	TP16	TP17	TP18	TP19	TP20	UTS	UAS	
1	AIF183010	Kerja Praktek 2	3																				0	0			
2	AIF183120	Pemrograman Permainan Komputer	3	A																			0	0			
3	AIF184002	Skripsi 2	5	A																			0	0			
Total sks					11																						
- Tahun Akademik 2020/2021** (Semester Ganjil):

No.	Kode MK	Nama Matakuliah	sks	Kelas	Kelompok Tugas																				Kelompok Ujian	AA	NA
					TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	TP11	TP12	TP13	TP14	TP15	TP16	TP17	TP18	TP19	TP20	UTS	UAS	
1	AIF183119	Keamanan Informasi	2	A			65																	95	81	81	A
2	AIF183341	Pola Komputasi Big Data Sistem	3	A	44	90	70	70	70	66														73	74	71	B

Gambar 3.32: Tangkapan Layar Tab Nilai Per Tahun Semester

Nilai Berdasarkan Kurikulum Tab ini berisi data nilai mahasiswa, dibagi berdasarkan acuan linimasa kurikulum (Gambar 3.33).

The screenshot shows a web-based student information system (SIAKAD) interface. The top navigation bar includes tabs for 'Ringkasan', 'Nilai Semester ini - 202', 'Nilai Per Tahun Semester', 'Nilai Berdasarkan Kurikulum' (which is the active tab), and 'Status Akademik'. Below the tabs is a table titled 'Komposisi sks:' showing the distribution of credits by year (2017-2018 to 2024-2025) across various academic categories like Wajib, Peminatan, and Pilhan. Another table below shows 'Nilai Akhir' (Final Grades) for different subjects (Matakuliah) over the same years, with grades ranging from A to E. The tables include columns for 'Kode' (Subject Code), 'Matakuliah' (Subject Name), 'Sks' (Credits), and 'Nilai Akhir' (Final Grade).

Gambar 3.33: Tangkapan Layar Tab Nilai Berdasarkan Kurikulum

Status Akademik Tab ini berisi riwayat status akademik mahasiswa, termasuk perubahan dosen wali (Gambar 3.34).

The screenshot shows the 'Status Akademik' tab. It displays a table with columns for 'No.', 'Tahun' (Year), 'Semester' (Semester), 'Status' (Status), and 'Dosen Wali' (Adviser). The table lists 10 entries from 2017 to 2020, detailing the status of each student and their respective advisers. The adviser column often contains multiple names separated by commas. A 'Kembali' button is visible at the bottom right of the table area.

Gambar 3.34: Tangkapan Layar Tab Status Akademik

3.2.3 Pra Kuliah

Bagian ini tidak dijelaskan karena tidak berkaitan dengan informasi mahasiswa.

3.2.4 Perkuliahan

Bagian ini tidak dijelaskan karena tidak berkaitan dengan informasi mahasiswa.

3.2.5 Ujian

Bagian ini tidak dijelaskan karena tidak berkaitan dengan informasi mahasiswa.

3.2.6 Nilai

Bagian ini tidak dijelaskan karena tidak berkaitan dengan informasi mahasiswa.

3.2.7 Evaluasi

Pada menu Evaluasi, hanya ada satu submenu untuk dosen, yaitu Laporan Evaluasi, yang memiliki satu subsubmenu juga, yaitu Daftar Perkembangan Studi. Di halaman ini, dosen dapat mencari data mahasiswa walinya, berupa Daftar Perkembangan Studi dalam bentuk PDF, seperti ditunjukkan pada Gambar 3.35.

The screenshot shows a web browser displaying a PDF document. The document header includes the university logo and name ('UNIVERSITAS KATOLIK PARAHYANGAN'), address ('Jalan Ciwideyut 94 Bandung 40141, telp. : +62.22.2030918-20 ext. 100401, 100422 fax: (022)203 1110, Telp.(022) 203 2655;(022) 204 204'), and contact information ('Dosen Wali : [redacted], Fakultas : Teknologi Informasi dan Sains, Program Studi : Teknik Informatika, Email : [redacted], Bidang Peminatan : [redacted]'). Below this, there are two tables:

- Table 1:** A grid for course selection from 2017/2018 to 2025/2026. It shows columns for Status tempuh (Wajib, Pilihan), Rincian (Jumlah sks), and a grid where each cell contains a number representing the number of students taking that specific course and year.
- Table 2:** Two large tables for academic records. The top table is for Semester 1 and the bottom one for Semester 2. Both tables have columns for Matakuliah (subject), sks (credit hours), and Nilai Akhir (Final Grade). The final grade column uses a scale from 1 to 6, with letter grades A, B, C, D, E, and F corresponding to certain ranges. The tables list various courses such as Pengantar Informatika, Pemrograman Berorientasi Objek, Logika, and others.

Gambar 3.35: Tangkapan Layar PDF Daftar Perkembangan Studi

3.2.8 Skripsi

Bagian ini tidak dijelaskan karena tidak berkaitan dengan informasi mahasiswa.

3.2.9 Sidang

Bagian ini tidak dijelaskan karena tidak berkaitan dengan informasi mahasiswa.

3.2.10 Kelulusan

Bagian ini tidak dijelaskan karena tidak berkaitan dengan informasi mahasiswa.

3.2.11 Pengumuman

Bagian ini tidak dijelaskan karena tidak berkaitan dengan informasi mahasiswa.

3.3 Data yang Dibutuhkan untuk *Screensaver*

3.3.1 Portal Akademik Mahasiswa

Halaman Utama

Pada Halaman Utama Portal Akademik Mahasiswa terdapat nama lengkap dan foto dari mahasiswa tersebut yang dapat diambil dan digunakan (Gambar 3.3 dengan kotak merah). Nama mahasiswa dapat diambil dengan mencari elemen "div" yang memiliki kelas "namaUser d-none d-lg-block mr-3", sehingga kueri css yang dihasilkan adalah "div[class=namaUser d-none d-lg-block mr-3]". Foto mahasiswa dapat diambil dengan mencari elemen "img" yang memiliki kelas "img-fluid fotoProfil", sehingga kueri css yang dihasilkan adalah "img[class=img-fluid fotoProfil]". Terdapat beberapa perubahan (Kode 3.2) yang perlu dilakukan terhadap skripsi Andrianto Sugiarto [5]:

1. Menghapus pemanggilan fungsi validateTLCertificates() dikarenakan sudah *deprecated* [6].
2. Sebelumnya semester yang sedang dijalani mahasiswa diambil dari menu FRS/PRS, namun karena terdapat perubahan dimana menu FRS/PRS tidak menuju ke Portal Akademik Mahasiswa melainkan menuju ke https://restu.unpar.ac.id/frs_prs, sehingga perlu melakukan *login* kembali apabila mengakses URL tersebut, maka semester yang sedang dijalani mahasiswa diambil dari menu Nilai. Dengan begitu, kueri css untuk pengambilan semester yang sedang dijalani juga perlu diubah, yaitu dengan mencari elemen "select" yang memiliki kelas "custom-select mr-3", sehingga kueri css yang dihasilkan adalah "select[class=custom-select mr-3]". Kemudian mengolah data yang didapat sehingga dapat dijadikan objek bertipe TahunSemester sesuai dengan SIAModels.

Kode 3.2: Perubahan Implementasi Jsoup Halaman Utama

```

1 @@ -96,24 +96,22 @@ public class Scraper {
2     Connection connection = Jsoup.connect(HOME_URL);
3     connection.cookie("ci_session", phpsessid);
4     connection.timeout(0);
5     - connection.validateTLCertificates(false);
6     connection.method(Connection.Method.GET);
7     Response resp = connection.execute();
8     Document doc = resp.parse();
9     String nama = doc.select("div[class=namaUser d-none d-lg-block mr-3]").text();
10    mhs.setNama(nama.substring(0, nama.indexOf(mhs.getEmailAddress())));
11    Element photo = doc.select("img[class=img-fluid fotoProfil]").first();
12    String photoPath = photo.attr("src");
13    mhs.setPhotoPath(photoPath);
14    - connection = Jsoup.connect(FRSPRS_URL);
15    + connection = Jsoup.connect(NILAI_URL);
16    connection.cookie("ci_session", phpsessid);
17    connection.timeout(0);
18    - connection.validateTLCertificates(false);
19    connection.method(Connection.Method.GET);
20    resp = connection.execute();
21    doc = resp.parse();
22    - String curr_sem = doc.select(".custom-selectContent span").text();
23    - String[] sem_set = parseSemester(curr_sem);
24    + Elements curr_sem = doc.select("select[class=custom-select mr-3]");
25    + String[] sem_set = parseSemester(curr_sem.first().child(curr_sem.first().childrenSize() - 1).text());
26    TahunSemester currTahunSemester = new TahunSemester(Integer.parseInt(sem_set[0]),
27                Semester.fromString(sem_set[1]));
28    return currTahunSemester;
29 @@ -214,7 +212,7 @@ public class Scraper {
30 }
31
32     public String[] parseSemester(String sem_raw) {
33     -     String[] sem_set = sem_raw.split("/")[0].split("-");
34     +     String[] sem_set = sem_raw.split("/")[0].split(" ");
35     return new String[]{sem_set[1].trim(), sem_set[0].trim()};
36 }
```

Halaman Profil

Pada Halaman Profil (Gambar 3.4 dengan kotak merah), tanggal lahir mahasiswa akan diambil untuk ditampilkan pada *screensaver*. Implementasi jsoup tersebut belum diimplementasikan pada skripsi Andrianto Sugiarto [5], sehingga perlu dilakukan penambahan fitur untuk mengambil data tersebut. Tanggal lahir mahasiswa dapat diambil dengan mencari elemen "div" yang memiliki kelas "offset-md-1 col-md-10 col-12 headerWrapper my-0 border-bottom", sehingga kueri css yang dihasilkan adalah "div[class=offset-md-1 col-md-10 col-12 headerWrapper my-0 border-bottom]".

Halaman Daftar Perkembangan Studi

Pada Halaman Daftar Perkembangan Studi (Gambar 3.17 dan 3.18 dengan kotak merah), data yang dapat dimanfaatkan dari halaman ini adalah IPK, IPS, jumlah sks yang lulus, dan jumlah sks yang ditempuh. Namun, pada SIAModels sudah terdapat *method* yang melakukan kalkulasi untuk mendapatkan data-data tersebut, sehingga tidak perlu dilakukan pengambilan data menggunakan jsoup. Untuk dapat memanfaatkan *method* tersebut diperlukan seluruh riwayat mata kuliah dan nilai yang pernah ditempuh mahasiswa, sehingga perlu dilakukan pengambilan data menggunakan jsoup. Implementasi pengambilan data tersebut sudah diimplementasikan sebelumnya pada skripsi Andrianto Sugiarto [5]. Proses dari pengambilan data tersebut yaitu:

- Mengambil data nilai berdasarkan tahun dan semester dengan mencari elemen "select" yang memiliki id "dropdownSemester", dan memiliki kelas "custom-select mr-3" sehingga kueri css yang dihasilkan adalah "select#dropdownSemester.custom-select.mr-3".
- Dikarenakan perlunya melakukan koneksi berkali-kali sebanyak semester yang telah ditempuh mahasiswa, sehingga dibutuhkan waktu yang tidak sebentar. Karena pada halaman nilai tidak dapat menampilkan seluruh semester seperti Portal Akademik Mahasiswa yang lama, sehingga untuk mengatasi masalah ini dibuat menjadi paralel. Untuk itu dibuat kelas yang mengimplementasikan kelas *interface Runnable*, yaitu kelas *MultipleRequest*. Kelas inilah yang melakukan koneksi ke setiap semester yang telah ditempuh mahasiswa, dan mengambil data-data tersebut.

Namun, terdapat beberapa perubahan (Kode 3.3) yang perlu dilakukan:

1. Menghapus pemanggilan fungsi validateTLSCertificates() dikarenakan sudah *deprecated* [6].
2. Indeks yang digunakan untuk melakukan kueri css berubah, sehingga untuk mengantisipasi tersebut, indeks yang digunakan menjadi *size* dari kueri css dikurangi 1.

Kode 3.3: Perubahan Implementasi Jsoup Halaman Daftar Perkembangan Studi

```

1 @@ -50,7 +50,7 @@ public class MultipleRequest implements Runnable {
2     Connection.Response resp = connection.execute();
3     Document doc = resp.parse();
4
5     Element script = doc.select("script").get(10);
6     + Element script = doc.select("script").get(doc.select("script").size()-1);
7     String scriptDataMataKuliah = script.html().substring(script.html().indexOf("var data_mata_kuliah = []"), script.
8         html().indexOf("var data_angket = []"));
9     engine.eval(scriptDataMataKuliah);
10    ScriptObjectMirror dataMataKuliah = (ScriptObjectMirror) engine.get("data_mata_kuliah");
11
12 @@ -131,7 +131,6 @@ public class Scraper {
13     Connection connection = Jsoup.connect(NILAI_URL);
14     connection.cookie("ci_session", phpsessid);
15     connection.timeout(0);
16     - connection.validateTLSCertificates(false);
17     connection.method(Connection.Method.POST);
18     Response resp = connection.execute();
19     Document doc = resp.parse();

```

Halaman TOEFL

Pada Halaman TOEFL (Gambar 3.20 dengan kotak merah), riwayat skor TOEFL dapat diambil dengan mencari elemen "table" yang memiliki elemen "tbody" didalamnya, serta memiliki elemen "tr" didalam elemen "tbody". Terdapat beberapa perubahan yang terjadi pada halaman TOEFL

semenjak skripsi Andrianto Sugiarto [5], yang mengakibatkan perlunya perubahan (Kode 3.4) terhadap implementasi jsoup:

1. Menghapus pemanggilan fungsi validateTLCertificates() dikarenakan sudah *deprecated* [6].
2. Perubahan format tanggal TOEFL pada Portal Akademik Mahasiswa menjadi dd-mm-yyyy.

Kode 3.4: Perubahan Implementasi Jsoup TOEFL

```

1  @@ -174,7 +174,6 @@ public class Scraper {
2      Connection connection = Jsoup.connect(TOEFL_URL);
3      connection.cookie("ci_session", phpsessid);
4      connection.timeout(0);
5      -     connection.validateTLCertificates(false);
6      connection.method(Connection.Method.POST);
7      Response resp = connection.execute();
8      Document doc = resp.parse();
9  @@ -183,45 +182,7 @@ public class Scraper {
10         for (int i = 0; i < nilaiTOEFL.size(); i++) {
11             Element nilai = nilaiTOEFL.get(i).select("td").get(5);
12             Element tgl_toefl = nilaiTOEFL.get(i).select("td").get(1);
13             String[] tanggal = tgl_toefl.text().split(" ");
14             switch (tanggal[1].toLowerCase()) {
15                 case "januari":
16                     tanggal[1] = "1";
17                     break;
18                 case "februari":
19                     tanggal[1] = "2";
20                     break;
21                 case "maret":
22                     tanggal[1] = "3";
23                     break;
24                 case "april":
25                     tanggal[1] = "4";
26                     break;
27                 case "mei":
28                     tanggal[1] = "5";
29                     break;
30                 case "juni":
31                     tanggal[1] = "6";
32                     break;
33                 case "juli":
34                     tanggal[1] = "7";
35                     break;
36                 case "agustus":
37                     tanggal[1] = "8";
38                     break;
39                 case "september":
40                     tanggal[1] = "9";
41                     break;
42                 case "oktober":
43                     tanggal[1] = "10";
44                     break;
45                 case "november":
46                     tanggal[1] = "11";
47                     break;
48                 case "desember":
49                     tanggal[1] = "12";
50                     break;
51             }
52             +     String[] tanggal = tgl_toefl.text().split("-");
53
54             LocalDate localDate = LocalDate.of(Integer.parseInt(tanggal[2]), Integer.parseInt(tanggal[1]),
55                                         Integer.parseInt(tanggal[0]));

```

3.3.2 SIAKAD

Subbab ini ditulis oleh dosen pembimbing.

Mendapatkan daftar mahasiswa

Untuk mendapatkan daftar mahasiswa, bisa memanfaatkan halaman “Cari Mahasiswa”. Caranya dengan mensimulasikan penekanan tombol “Cari” dengan parameter *default*, yang akan menampilkan daftar mahasiswa dari dosen wali yang bersangkutan. Hasilnya dalam bentuk tabel dan di-*parsing*, dengan ketentuan sebagai berikut:

1. NPM tersedia pada kolom kedua.
2. Nama Mahasiswa tersedia pada kolom ketiga.
3. Kolom-kolom lainnya diabaikan.

Mendapatkan detail mahasiswa

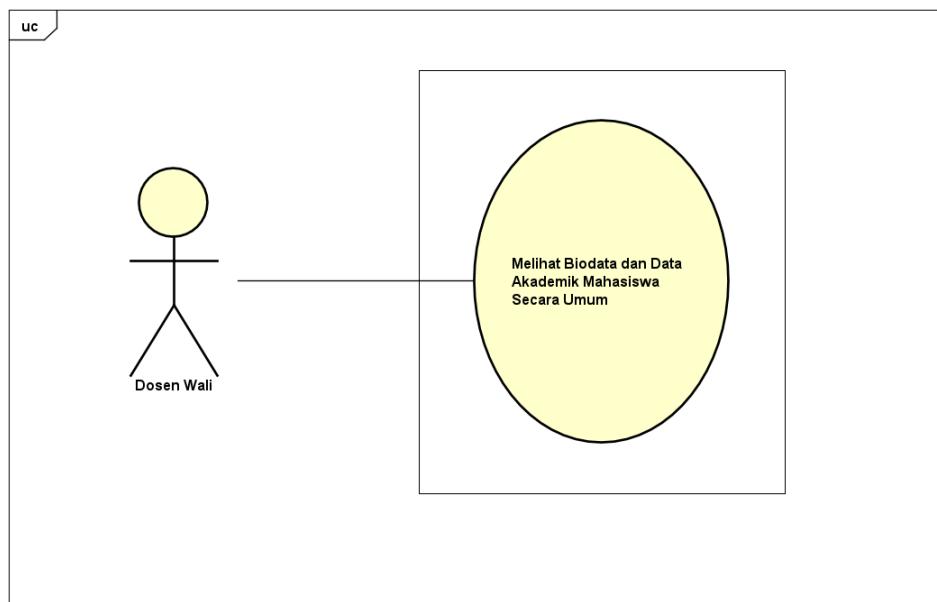
Riwayat Nilai Pada SIAKAD, riwayat nilai mahasiswa tersedia pada beberapa halaman:

1. **Nilai Per Tahun Semester** berisikan tabel nilai yang dibagi per tahun semester. Data dari halaman inilah yang dipakai untuk mendapatkan nilai mahasiswa, mencakup: kode dan nama mata kuliah, SKS, kelas, TP1 s.d. TP20, UTS, UAS, AA, dan NA.
2. **Nilai Berdasarkan Kurikulum** berisikan tabel nilai yang dibagi berdasarkan linimasa kurikulum. Data ini kurang cocok dipakai karena tidak mengandung nilai tugas dan ujian, hanya nilai akhir saja.
3. **Evaluasi** berisikan tabel nilai lengkap dalam bentuk Daftar Perkembangan Studi dengan format PDF. Data ini sulit diekstraksi karena tidak dalam bentuk HTML.
Kesimpulannya, riwayat nilai diambil dari halaman “Nilai Per Tahun Semester”.

Data Diri Pada SIAKAD, data diri dapat diambil dari halaman “Data Diri”, tab identitas mahasiswa. Foto tersedia dalam bentuk Data URL¹ yang di-*encode* dalam format *base64*. Data-data diri lainnya dapat diambil dengan cukup sederhana dari berbagai elemen yang ada di tab tersebut.

3.4 Analisis Sistem *Screensaver*

3.4.1 Diagram *Use Case Screensaver*



Gambar 3.36: Diagram *Use Case Screensaver*

Pada diagram *use case screensaver* (Gambar 3.36) terdapat 1 fitur utama dari *screensaver* yaitu, melihat biodata dan data akademik mahasiswa secara umum. Dimana biodata yang dimaksud adalah nama, angkatan, tanggal lahir. Serta data akademik yang dimaksud adalah status, email, nilai TOEFL, IPS, IPK, SKS lulus, SKS tempuh mahasiswa.

¹https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Data_URIs

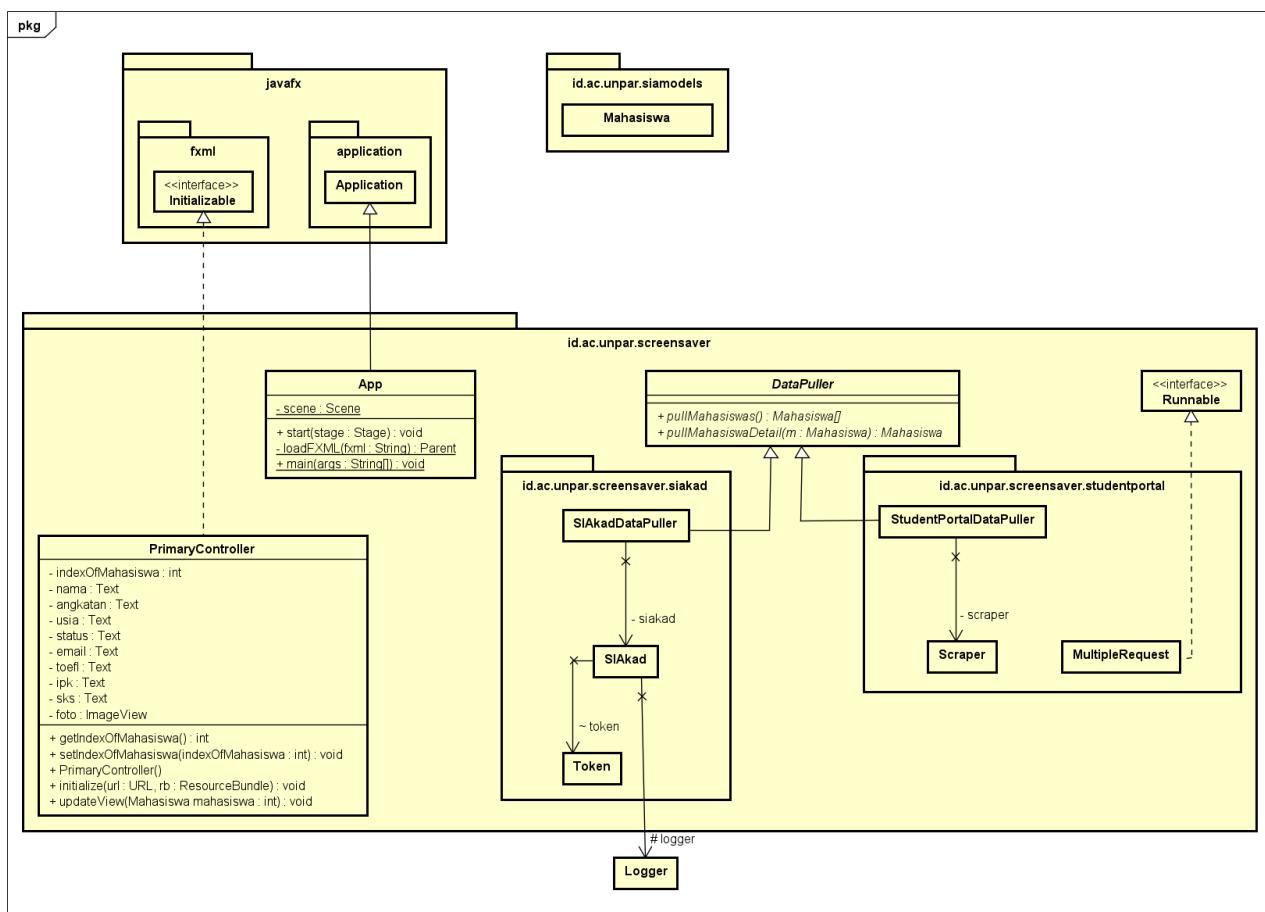
BAB 4

PERANCANGAN

Pada bab ini akan dijelaskan mengenai perancangan kelas beserta deskripsi kelas dan fungsinya, serta perancangan antarmukanya.

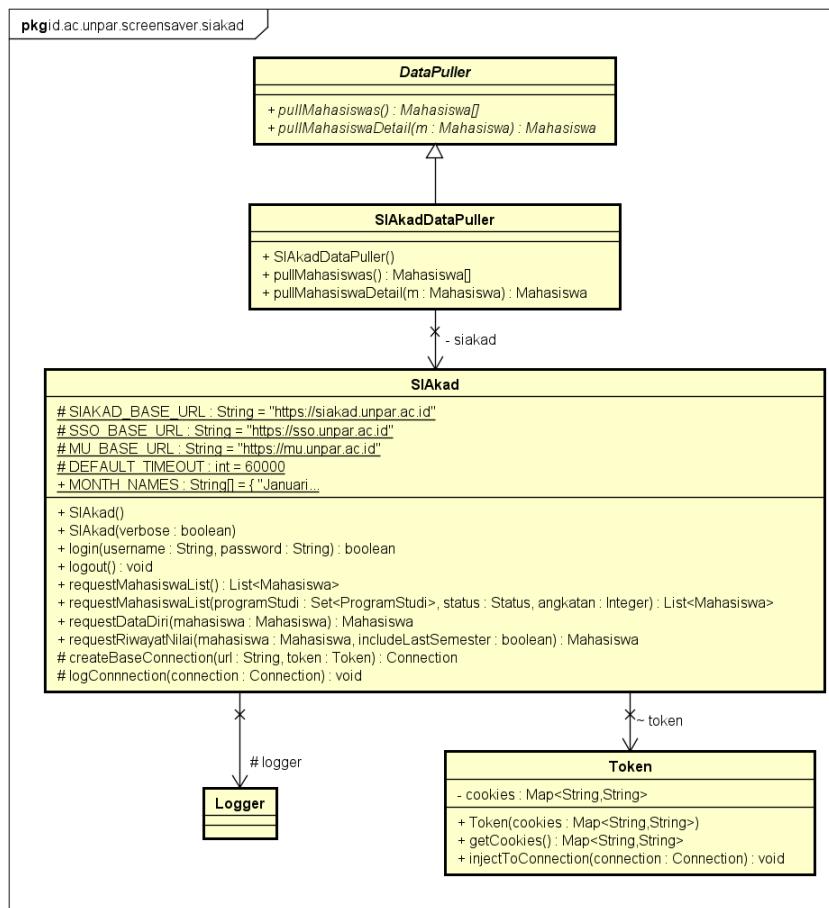
4.1 Perancangan Kelas

Diagram kelas secara keseluruhan dapat dilihat pada gambar 4.1, dimana terdapat *package screensaver* yang didalamnya terdapat *package siakad* (Gambar 4.2), dan *package studentportal* (Gambar 4.3). Penjelasan mengenai kelas, *method*, dan atribut jsoup terdapat pada bagian 2.1. Penjelasan mengenai kelas, *method*, dan atribut JavaFX serta FXML terdapat pada bagian 2.2. Penjelasan mengenai kelas, *method*, dan atribut SIAModels terdapat pada bagian 2.3.



Gambar 4.1: Diagram Kelas Keseluruhan

Penjelasan mengenai kelas, *method*, dan atribut pada *package screensaver* adalah sebagai berikut:



Gambar 4.2: Diagram Kelas SIAKAD

1. App

Kelas ini merupakan turunan dari kelas [Application 2.2.1](#). Atribut yang dimiliki kelas ini adalah sebagai berikut:

- `private static Scene scene`: menyimpan objek `Scene`.

Method yang dimiliki kelas ini adalah sebagai berikut:

- `public void start(Stage stage)`

Berfungsi untuk menampilkan `Stage` dengan `Scene` yang digunakan.

Parameter:

- `stage`: `stage` utama untuk aplikasi ini.

- `private static Parent loadFXML(String fxml)`

Berfungsi untuk memuat hierarki objek dari dokumen FXML.

Parameter:

- `fxml`: dokumen fxml.

Kembalian: hierarki objek dari dokumen FXML.

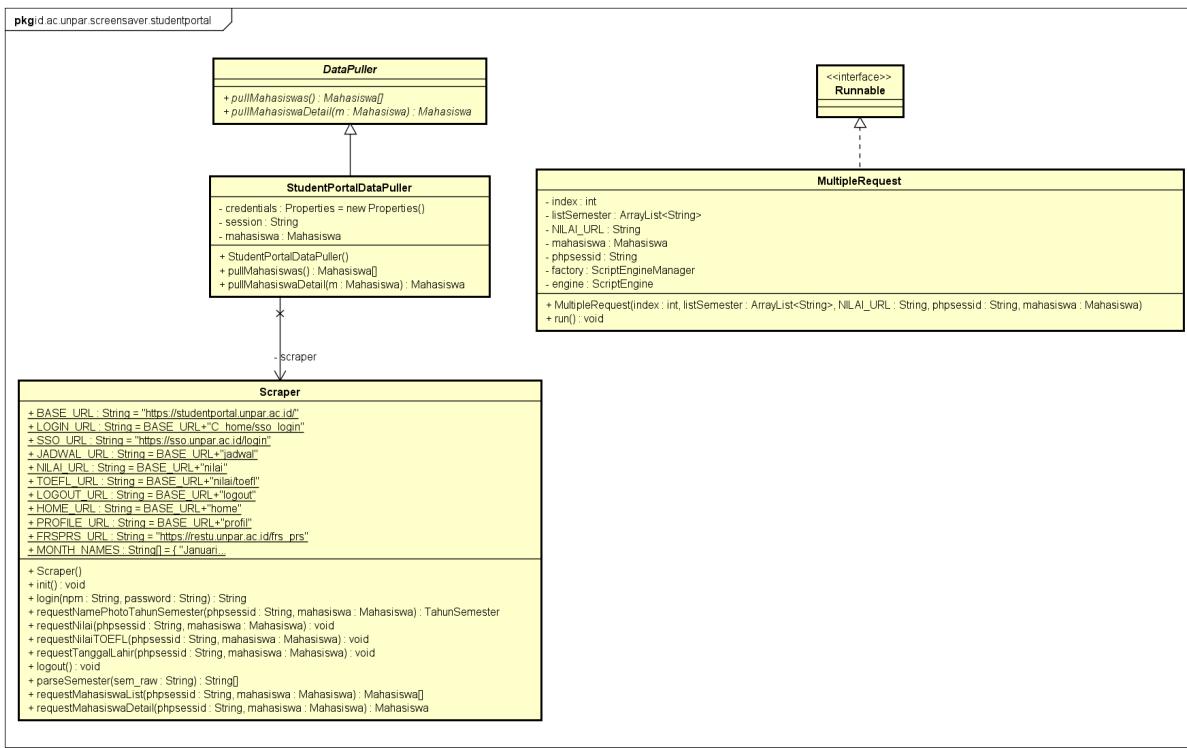
- `public static void main(String[] args)`

Berfungsi untuk memanggil `method launch()` pada kelas `Application`.

2. PrimaryController

Kelas ini mengimplementasikan kelas `interface Initializable 2.2.2`. Atribut yang dimiliki kelas ini adalah sebagai berikut:

- `private int indexOfMahasiswa`: menyimpan indeks mahasiswa yang akan ditampilkan pada `screensaver`.
- `private Text nama`: menyimpan objek `Text` yang digunakan untuk keterangan nama



Gambar 4.3: Diagram Kelas Studentportal

mahasiswa.

- `private Text angkatan`: menyimpan objek `Text` yang digunakan untuk keterangan angkatan mahasiswa.
- `private Text usia`: menyimpan objek `Text` yang digunakan untuk keterangan usia mahasiswa.
- `private Text status`: menyimpan objek `Text` yang digunakan untuk keterangan status mahasiswa.
- `private Text email`: menyimpan objek `Text` yang digunakan untuk keterangan *email* mahasiswa.
- `private Text toefl`: menyimpan objek `Text` yang digunakan untuk keterangan nilai TOEFL mahasiswa.
- `private Text ipk`: menyimpan objek `Text` yang digunakan untuk keterangan IPK mahasiswa.
- `private Text sks`: menyimpan objek `Text` yang digunakan untuk keterangan SKS mahasiswa.
- `private ImageView foto`: menyimpan objek `ImageView` yang digunakan untuk foto mahasiswa.

Method yang dimiliki kelas ini adalah sebagai berikut:

- `public int getIndexOfMahasiswa()`
Berfungsi untuk mengambil indeks mahasiswa.
Kembalian: indeks mahasiswa.
- `public void setIndexOfMahasiswa(int indexOfMahasiswa)`
Berfungsi untuk menyimpan indeks mahasiswa.
Parameter:
 - `indexOfMahasiswa`: indeks mahasiswa.
- `public PrimaryController()`
Berfungsi sebagai *constructor* kelas `PrimaryController`.

- `public void initialize(URL url, ResourceBundle rb)`
Berfungsi untuk mengambil *list* mahasiswa beserta dengan keterangan detil mahasiswa.
- `public void updateView(Mahasiswa mahasiswa)`
Berfungsi untuk memperbarui tampilan *screensaver* dengan data mahasiswa yang menjadi parameter.

Parameter:

- `mahasiswa`: objek mahasiswa.

Penjelasan mengenai kelas, *method*, dan atribut pada *package studentportal* adalah sebagai berikut:

1. DataPuller

Kelas ini merupakan *abstract class*. *Method* yang dimiliki kelas ini adalah sebagai berikut:

- `public abstract Mahasiswa[] pullMahasiswas()`
Merupakan *abstract method* yang akan diimplementasikan oleh turunannya.
- `public abstract Mahasiswa pullMahasiswaDetail()`
Merupakan *abstract method* yang akan diimplementasikan oleh turunannya.

2. StudentPortalDataPuller

Kelas ini merupakan turunan dari kelas DataPuller 1. Kelas ini memiliki fungsi untuk mengambil *npm* dan *password* mahasiswa yang disimpan dalam sebuah *file*, dan kemudian memanggil *method* pada kelas Scraper. Atribut yang dimiliki kelas ini adalah sebagai berikut:

- `private Scraper scraper: menyimpan objek Scraper.`
- `private final Properties credentials: mengakses file yang berisi npm dan password mahasiswa.`
- `private Mahasiswa mahasiswa: menyimpan objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.`
- `private String session: menyimpan session yang dapat digunakan untuk mengakses menu di Portal Akademik Mahasiswa.`

Method yang dimiliki kelas ini adalah sebagai berikut:

- `public StudentPortalDataPuller()`
Berfungsi sebagai *constructor* kelas StudentPortalDataPuller.
- `public Mahasiswa[] pullMahasiswas()`
Berfungsi untuk memanggil *method requestMahasiswaList* pada kelas Scraper (3).
Kembalian: *array* yang berisi seluruh mahasiswa yang memiliki dosen wali yang sama dengan yang melakukan *login* (dalam implementasi menggunakan Portal Akademik Mahasiswa, mahasiswa yang diambil adalah hanya mahasiswa yang melakukan *login*).
- `public Mahasiswa pullMahasiswaDetail()`
Berfungsi untuk memanggil *method requestMahasiswaDetail* pada kelas Scraper (3).
Kembalian: objek mahasiswa yang telah ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.

3. Scraper

Kelas ini memiliki fungsi untuk melakukan pengambilan data dari Portal Akademik Mahasiswa untuk kemudian diolah dan ditampilkan. Atribut yang dimiliki kelas ini adalah sebagai berikut:

- `public static final String BASE_URL: menyimpan url utama Portal Akademik Mahasiswa.`
- `public static final String LOGIN_URL: menyimpan url halaman login Portal Akademik Mahasiswa.`
- `public static final String SSO_URL: menyimpan url halaman login Single Sign On(SSO) UNPAR.`
- `public static final String JADWAL_URL: menyimpan url halaman jadwal Portal Akademik Mahasiswa.`
- `public static final String NILAI_URL: menyimpan url halaman nilai Portal Akademik Mahasiswa.`

- `public static final String TOEFL_URL`: menyimpan *url* halaman nilai TOEFL Portal Akademik Mahasiswa.
- `public static final String LOGOUT_URL`: menyimpan *url* untuk melakukan *logout*.
- `public static final String HOME_URL`: menyimpan *url* halaman utama Portal Akademik Mahasiswa setelah melakukan *login*.
- `public static final String PROFILE_URL`: menyimpan *url* halaman profil mahasiswa Portal Akademik Mahasiswa.
- `public static final String FRSPRS_URL`: menyimpan *url* halaman FRS/PRS Portal Akademik Mahasiswa.
- `public static final String[] MONTH_NAMES`: menyimpan nama-nama bulan dalam bahasa Indonesia.

Method yang dimiliki kelas ini adalah sebagai berikut:

- `public Scrapper()`

Berfungsi sebagai *constructor* kelas `Scrapper`.

- `public void init()`

Berfungsi untuk melakukan koneksi ke halaman utama Portal Akademik Mahasiswa.

- `public String login(String npm, String password)`

Berfungsi untuk melakukan *login* ke Portal Akademik Mahasiswa.

Parameter:

– `npm`: *npm* mahasiswa yang dipakai *login*.

– `password`: *password* mahasiswa yang dipakai *login*.

Kembalian: *session* yang dapat digunakan untuk mengakses menu di Portal Akademik Mahasiswa.

- `public TahunSemester requestNamePhotoTahunSemester(String phpsessid, Mahasiswa mahasiswa)`

Berfungsi untuk melakukan pengambilan nama, foto, serta semester yang sedang dijalani mahasiswa dari Portal Akademik Mahasiswa.

Parameter:

– `phpsessid`: *session* yang didapatkan dari proses *login* ke Portal Akademik Mahasiswa.

– `mahasiswa`: objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.

Kembalian: semester yang sedang dijalani mahasiswa.

- `public void requestNilai(String phpsessid, Mahasiswa mahasiswa)`

Berfungsi untuk melakukan pengambilan seluruh nilai mata kuliah mahasiswa dari Portal Akademik Mahasiswa.

Parameter:

– `phpsessid`: *session* yang didapatkan dari proses *login* ke Portal Akademik Mahasiswa.

– `mahasiswa`: objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.

- `public void requestNilaiTOEFL(String phpsessid, Mahasiswa mahasiswa)`

Berfungsi untuk melakukan pengambilan seluruh riwayat nilai TOEFL mahasiswa dari Portal Akademik Mahasiswa.

Parameter:

– `phpsessid`: *session* yang didapatkan dari proses *login* ke Portal Akademik Mahasiswa.

– `mahasiswa`: objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.

- `public void requestTanggalLahir(String phpsessid, Mahasiswa mahasiswa)`

Berfungsi untuk melakukan pengambilan data tanggal lahir mahasiswa dari Portal

Akademik Mahasiswa.

Parameter:

- `phpsessid`: *session* yang didapatkan dari proses *login* ke Portal Akademik Mahasiswa.
- `mahasiswa`: objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.

- `public void logout()`

Berfungsi untuk melakukan *logout* dari Portal Akademik Mahasiswa.

- `public String[] parseSemester(String sem_raw)`

Berfungsi untuk melakukan *parsing* data semester mahasiswa dari Portal Akademik Mahasiswa agar dapat diolah lebih lanjut.

Parameter:

- `sem_raw`: data semester yang didapat dari Portal Akademik Mahasiswa.

Kembalian: *array* yang berisi semester yang sudah dilakukan *parsing*.

- `public Mahasiswa[] requestMahasiswaList(String phpsessid, Mahasiswa mahasiswa)`

Berfungsi untuk mengambil seluruh mahasiswa dengan data yang diambil berupa nama, foto, semester yang sedang dijalani, dimana seluruh mahasiswa tersebut memiliki dosen wali yang sama dengan yang melakukan *login* (dalam implementasi menggunakan Portal Akademik Mahasiswa, mahasiswa yang diambil adalah hanya mahasiswa yang melakukan *login*).

Parameter:

- `phpsessid`: *session* yang didapatkan dari proses *login* ke Portal Akademik Mahasiswa.
- `mahasiswa`: objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.

Kembalian: *array* yang berisi seluruh mahasiswa yang memiliki dosen wali yang sama dengan yang melakukan *login* (dalam implementasi menggunakan Portal Akademik Mahasiswa, mahasiswa yang diambil adalah hanya mahasiswa yang melakukan *login*).

- `public Mahasiswa requestMahasiswaDetail(String phpsessid, Mahasiswa mahasiswa)`

Berfungsi untuk mengambil detail lebih lanjut mengenai data mahasiswa berupa nilai TOEFL, seluruh nilai mata kuliah, tanggal lahir mahasiswa.

Parameter:

- `phpsessid`: *session* yang didapatkan dari proses *login* ke Portal Akademik Mahasiswa.
- `mahasiswa`: objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.

Kembalian: objek mahasiswa yang telah ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.

4. MultipleRequest

Kelas ini mengimplementasikan kelas `interface Runnable` milik Java. Kelas ini memiliki fungsi untuk melakukan koneksi ke setiap semester yang telah ditempuh mahasiswa pada halaman nilai di Portal Akademik Mahasiswa (penjelasan lebih lanjut di 3.3.1). Atribut yang dimiliki kelas ini adalah sebagai berikut:

- `private int index`: menyimpan indeks semester yang akan digunakan untuk mengakses `listSemester`.
- `private ArrayList<String> listSemester`: menyimpan daftar semester yang telah ditempuh mahasiswa.
- `private String NILAI_URL`: menyimpan *url* halaman nilai Portal Akademik Mahasiswa.
- `private String phpsessid`: menyimpan *session* yang dapat digunakan untuk mengakses menu di Portal Akademik Mahasiswa.

- `private Mahasiswa mahasiswa`: menyimpan objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.
- `private ScriptEngineManager factory`: untuk menjalankan *javascript*.
- `private ScriptEngine engine`: untuk menjalankan *javascript*.

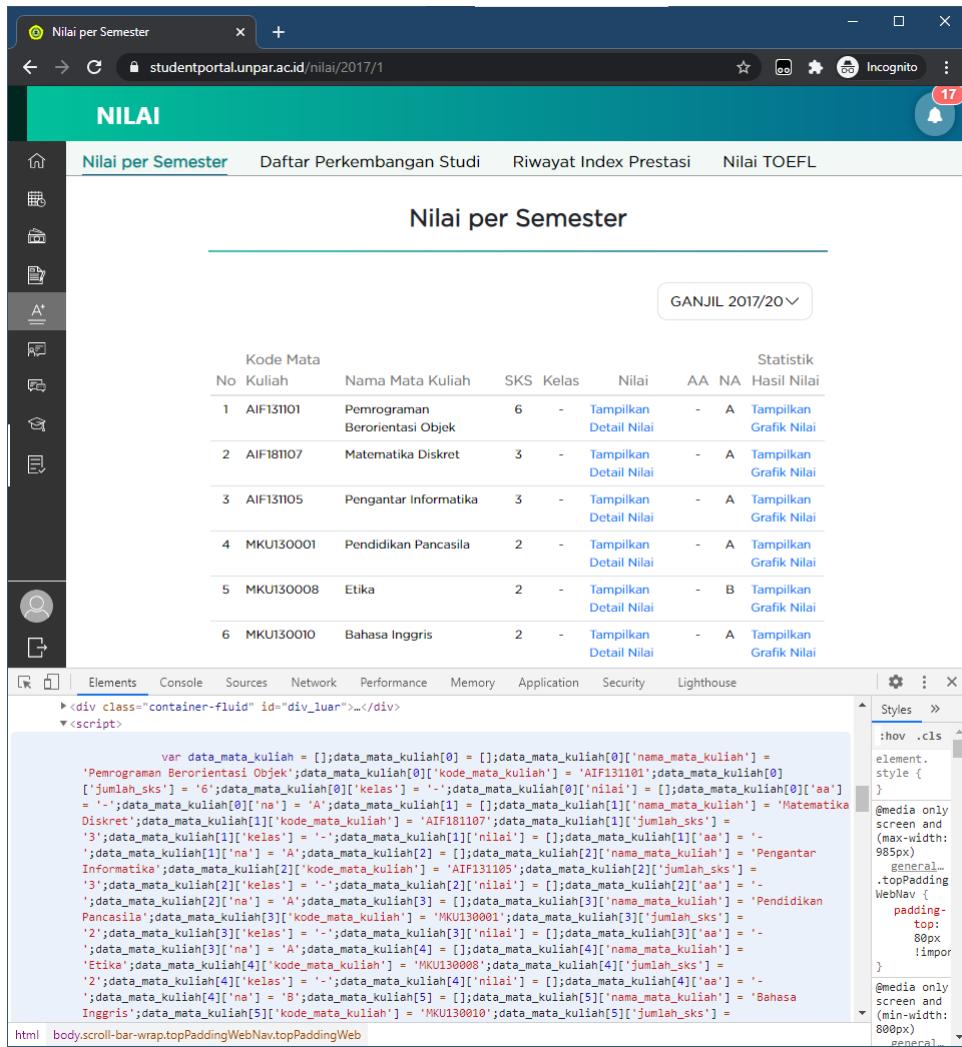
Method yang dimiliki kelas ini adalah sebagai berikut:

- `public MultipleRequest(int index, ArrayList<String> listSemester, String NILAI_URL, String phpsessid, Mahasiswa mahasiswa)`

Berfungsi sebagai *constructor* kelas `MultipleRequest`.

Parameter:

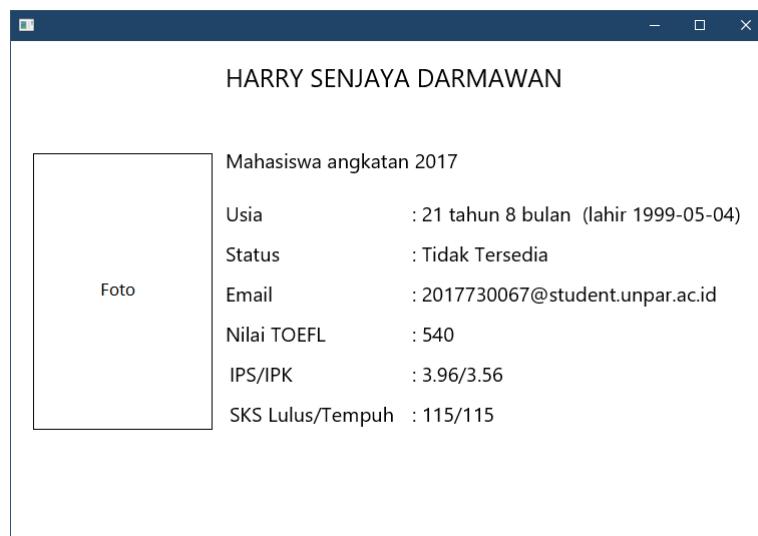
- `index`: indeks semester.
 - `listSemester`: *list* semester yang ditempuh mahasiswa.
 - `NILAI_URL`: *url* halaman nilai pada Portal Akademik Mahasiswa.
 - `phpsessid`: *session* yang didapatkan dari proses login ke Portal Akademik Mahasiswa.
 - `mahasiswa`: objek mahasiswa yang akan ditambahkan datanya dari pengambilan data di Portal Akademik Mahasiswa.
- `public void run()`
Method ini merupakan *method* turunan dari kelas `interface Runnable`. Untuk mendapatkan data nilai dilakukan dengan cara:
 - (a) Mendapatkan tahun dan semester yang ditempuh mahasiswa dari atribut `ArrayList<String> listSemester` diambil dari value atribut `int index`. Kemudian String dibagi menjadi tahun dan semester yang dibutuhkan.
 - (b) Setelah mendapatkan tahun dan semester. Kemudian melakukan koneksi ke *url* nilai berdasarkan tahun dan semester (sebagai contoh dalam gambar 4.4, *url* yang digunakan yaitu <https://studentportal.unpar.ac.id/nilai/2017/1>).
 - (c) Setelah berhasil, kemudian melakukan kueri css berdasarkan script yang mengandung nilai mahasiswa (Gambar 4.4).
 - (d) Selanjutnya adalah mendapatkan script yang mengandung script “`var data_mata_kuliah = [];`” sampai indeks dari “`var data_angket = [];`”
 - (e) Setelah mendapatkan script yang dibutuhkan, selanjutnya menjalankan script menggunakan *method* milik kelas `ScriptEngine` yaitu `Object eval(String script)`.
 - (f) Setelah berhasil, data yang didapatkan bertipe `ScriptObjectMirror` yang membungkus hasil eksekusi. Data nilai didapatkan dengan menggunakan *method* `Object get(Object key)`.
 - (g) Setelah berhasil, kemudian memasukkan data nilai ke daftar riwayat nilai mahasiswa pada atribut kelas `Mahasiswa` yaitu `List<Nilai> riwayatNilai` menggunakan *method* `List<Nilai> getRiwayatNilai()`. Proses ini dilakukan berulang kali sebanyak jumlah mata kuliah per semesternya.



Gambar 4.4: Script Data Nilai Mahasiswa Pada Halaman Nilai

4.2 Perancangan Antarmuka

Gambar 4.5 menunjukkan rancangan antarmuka dari aplikasi screensaver.



Gambar 4.5: Rancangan Antarmuka *Screensaver*

BAB 5

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dijelaskan mengenai lingkungan pengembangan perangkat lunak dan hasil implementasi.

5.1 Lingkungan Implementasi

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

1. *Processor*: Intel Core i5-10300H
2. *Random Access Memory(RAM)*: 16 GB DDR4
3. Sistem Operasi: Windows 10
4. Versi Java: 15.0.1
5. Versi JavaFX: 8.0.202
6. Versi Netbeans: 12.1
7. Versi Scenebuilder: 11.0.0

5.2 Hasil Implementasi

Hasil implementasi berupa aplikasi *screensaver*, dimana aplikasi tersebut akan dijalankan secara otomatis setelah komputer tidak digunakan selama beberapa saat. Gambar 5.1 merupakan tampilan dari aplikasi *screensaver*.

HARRY SENJAYA DARMAWAN



Mahasiswa angkatan 2017

Usia: 22 tahun 0 bulan (lahir 1999-05-04)

Status: Tidak Tersedia

Email: 2017730067@student.unpar.ac.id

Nilai TOEFL: 540

IPS/IPK: 3.74/3.58

SKS Lulus/Tempuh: 134/134

Gambar 5.1: Tampilan *Screensaver*

DAFTAR REFERENSI

- [1] Alfadian, P. (2016) SI Akademik. <http://bti.unpar.ac.id/akademik/>. 19 Oktober 2020.
- [2] Wikipedia (2021) Screensaver. <https://en.wikipedia.org/wiki/Screensaver>. 07 Maret 2021.
- [3] 2018, T. P. P. A. M. P. (2018) BUKU PANDUAN PENGGUNAAN PORTAL AKADEMIK MAHASISWA (PAM) 2018. https://studentportal.unpar.ac.id/assets/BUKU_PANDUAN_PENGGUNAAN_FRS_GABUNGAN.pdf. 28 Februari 2021.
- [4] Alfadian, P. (2020) SIAModels. <https://github.com/pascalalfadian/SIAModels>. 19 Oktober 2020.
- [5] Sugiarto, A. (2018) PENYESUAIAN SIAMODELS DAN IFSTUDENTPORTAL KE KURIKULUM 2018. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [6] Version 1.13.1 (2009-2020) *jsoup: Java HTML Parser*. Jonathan Hedley. Seattle, Washington.
- [7] Version 15 (2008-2020) *JavaFX 15*. Gluon HQ. Industriegweg 3, Leuven, BE 3001, BE.

LAMPIRAN A

KODE PROGRAM

Kode 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[i] ) {
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 + &dcaa) / (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, nepsweb.co.uk
21 // 8 October 2012
22 // http://nepsweb.co.uk/docs/progfonts.pdf
23

```

Kode A.2: MyCode.java

```

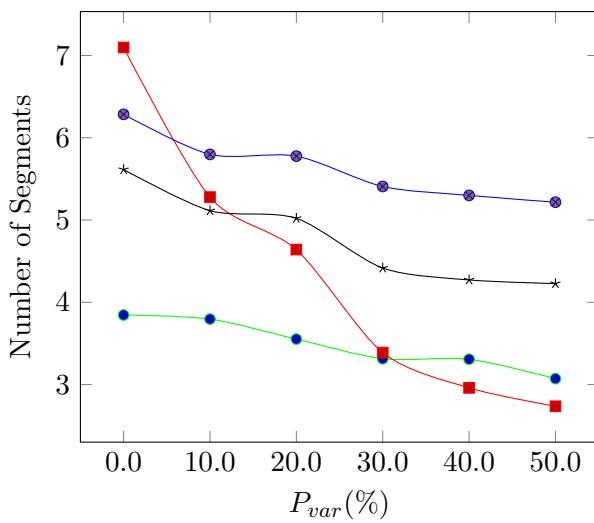
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id;                                //id of the set
8     protected MyEdge FurthestEdge;                   //the 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

```

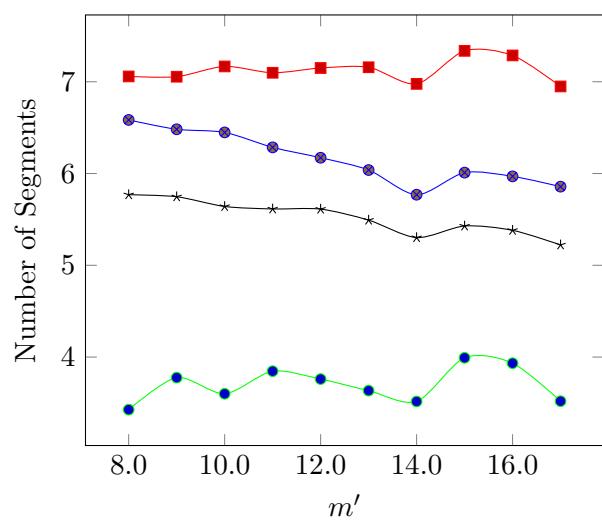

LAMPIRAN B

HASIL EKSPERIMENT

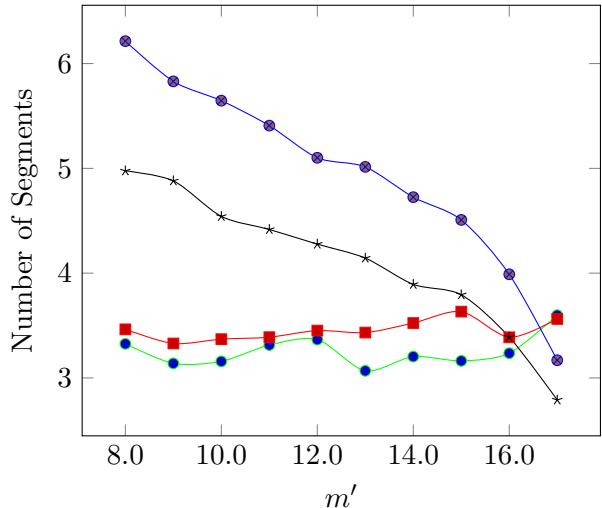
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



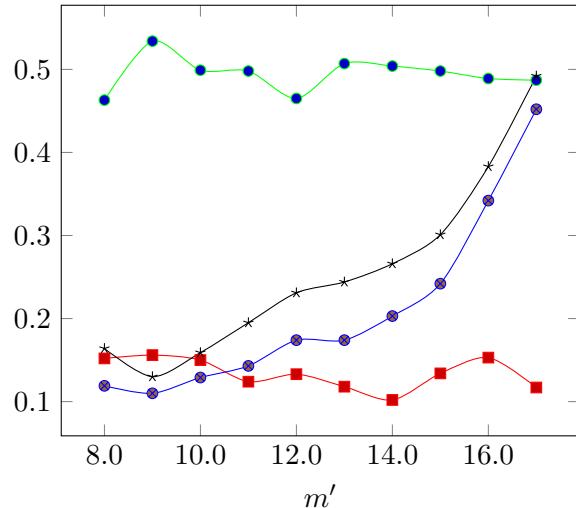
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4