

SKRIPSI

INFORMATIKA STUDENT PORTAL: KUSTOMISASI PORTAL
AKADEMIK MAHASISWA UNTUK MAHASISWA TEKNIK
INFORMATIKA UNPAR



HERFAN HERYANDI

NPM: 2012730012

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2015

UNDERGRADUATE THESIS

**INFORMATIKA STUDENT PORTAL: *PORTAL AKADEMIK
MAHASISWA* CUSTOMIZATION FOR UNPAR'S
INFORMATICS ENGINEERING STUDENTS**



HERFAN HERYANDI

NPM: 2012730012

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2015**

ABSTRAK

Portal Akademik Mahasiswa merupakan sebuah situs jaringan yang diperuntukan bagi mahasiswa dalam rangka mendapatkan informasi kegiatan akademik. Namun, fitur-fitur yang dimilikinya masih belum mendukung kebutuhan akademik mahasiswa Program Studi Teknik Informatika UNPAR. Oleh karena itu, dibuatlah aplikasi Informatika Student Portal.

Informatika Student Portal merupakan aplikasi berbasis web yang dibuat menggunakan Play Framework. Data yang diolah Informatika Student Portal diambil langsung dari Portal Akademik Mahasiswa dengan memanfaatkan *web scraping* yang diimplementasikan menggunakan jsoup. Untuk mendapatkan data tersebut, perlu dilakukan analisis komunikasi dari Portal Akademik Mahasiswa. Analisis tersebut dilakukan menggunakan Chrome DevTools. Data yang telah diambil menggunakan jsoup dapat diseleksi dengan memanfaatkan CSS *selector* kemudian data tersebut akan diolah menggunakan SIA Models.

Pengujian dari aplikasi Informatika Student Portal dilakukan oleh beberapa mahasiswa dengan kriteria yang beragam. Aplikasi dapat berjalan dengan baik dan memberikan hasil yang tepat pada seluruh mahasiswa kecuali mahasiswa yang pernah melakukan transfer studi. Hasil pengujian aplikasi Informatika Student Portal membuktikan bahwa *web scraping* dapat dimanfaatkan memenuhi kebutuhan akademik mahasiswa Program Studi Teknik Informatika UNPAR jika digabungkan dengan SIA Models.

Kata-kata kunci: Informatika Student Portal, Portal Akademik Mahasiswa, Teknik Informatika UNPAR, *Web Scraping*, jsoup, Play Framework, SIA Models, Chrome DevTools, CSS *Selector*

ABSTRACT

Keywords: Informatika Student Portal, Portal Akademik Mahasiswa, Informatics Engineering UNPAR, Web Scraping, jsoup, Play Framework, SIA Models, Chrome DevTools, CSS Selector

DAFTAR ISI

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metode Penelitian	3
1.6 Sistematika Penulisan	4
2 DASAR TEORI	5
2.1 jsoup	5
2.1.1 Jsoup	5
2.1.2 Connection	5
2.1.3 Response	6
2.1.4 Document	7
2.1.5 Elements	7
2.1.6 Element	7
2.2 CSS <i>Selector</i>	8
2.3 Chrome DevTools	9
2.3.1 Elements	10
2.3.2 Network	11
2.4 Play Framework	16
2.5 SIA Models	18
3 ANALISIS	27
3.1 Analisis Portal Akademik Mahasiswa	27
3.2 Analisis Kebutuhan Informatika Student Portal	37
3.3 Analisis Komunikasi Portal Akademik Mahasiswa untuk Fitur Informatika Student Portal	41
3.3.1 Kasus <i>Login</i>	41
3.3.2 Kasus Nilai	43
3.3.3 Kasus Jadwal	46
3.3.4 Kasus <i>Logout</i>	51
3.4 Analisis Arsitektur Informatika Student Portal	54
3.5 Analisis <i>Use Case</i>	54
3.5.1 Diagram <i>Use Case</i>	54
3.5.2 Skenario <i>Use Case</i>	55
3.6 Analisis Kelas	57

4 PERANCANGAN	59
4.1 Diagram Kelas Rinci	59
4.2 Perancangan Antarmuka	65
5 IMPLEMENTASI DAN PENGUJIAN	69
5.1 Implementasi	69
5.1.1 Lingkungan Implementasi	69
5.1.2 Hasil Implementasi	69
5.2 Pengujian	72
5.2.1 Pengujian Fungsional	72
5.2.2 Pengujian Eksperimental	74
6 KESIMPULAN DAN SARAN	75
6.1 Kesimpulan	75
6.2 Saran	75
DAFTAR REFERENSI	77
A KODE PROGRAM	79

DAFTAR GAMBAR

1.1 Prasyarat Mata Kuliah[1]	2
1.2 Prasyarat Mata Kuliah Portal Akademik Mahasiswa[2]	2
2.1 Format Penulisan Definisi CSS	8
2.2 Chrome DevTools	10
2.3 Panel Elements	11
2.4 Panel Network	12
2.5 Contoh Tautan Header pada Metode Permintaan GET	13
2.6 Contoh Tautan Header pada Metode Permintaan POST	13
2.7 Contoh <i>Preview</i> yang Tersedia	14
2.8 Contoh <i>Preview</i> yang Tidak Tersedia	14
2.9 Contoh Tautan Response	15
2.10 Contoh Tabel pada Tautan Cookie	16
2.11 Struktur Aplikasi Play Framework	16
2.12 Contoh Komponen Route[3]	17
2.13 Direktori app yang Dibangkitkan Play Framework[3]	17
2.14 Diagram Kelas SIA Models	19
3.1 Halaman Utama Portal Akademik Mahasiswa	27
3.2 Menu Atas Home	28
3.3 Menu Atas Kuliah	28
3.4 Menu Atas Profil	28
3.5 Menu Atas Komentar	29
3.6 Identitas Portal	29
3.7 Menu Utama	29
3.8 Tampilan Registrasi FRS/PRS	30
3.9 Tampilan Kartu Rencana Studi[4]	30
3.10 Tampilan Pindah Kelas MKU[4]	31
3.11 Tampilan Jadwal Kuliah, UTS, dan UAS	31
3.12 Tampilan Jadwal MKU	32
3.13 Tampilan Jadwal Seluruh Fakultas	32
3.14 Tampilan Riwayat Per Semester	33
3.15 Tampilan Daftar Perkembangan Studi	33
3.16 Tampilan Statistik Nilai dan IP	34
3.17 Tampilan Riwayat Indeks Prestasi	34
3.18 Tampilan TOEFL	35
3.19 Tampilan Pembayaran Uang Kuliah	35
3.20 Tampilan Informasi	36
3.21 Tampilan <i>Pop Up</i> Informasi	36
3.22 Tampilan Kalender	36
3.23 Tampilan Info Browser	36
3.24 Tampilan Jadwal pada Portal Akademik Mahasiswa	38
3.25 Tampilan Jadwal yang Diinginkan Mahasiswa	38

3.26 Tampilan <i>Mobile</i> Portal Akademik Mahasiswa	39
3.27 Tombol “input#submit.login-button” pada Halaman Depan Portal Akademik Mahasiswa	41
3.28 Kolom “ <i>Username</i> ” “input#username.required” pada Halaman CAS UNPAR	42
3.29 <i>Form Data</i> yang dikirim CAS UNPAR	43
3.30 Elemen “a” dengan teks “Riwayat Per Semester” pada Menu Nilai dan Indeks Prestasi	44
3.31 <i>Combo Box</i> “select#tahun_akd_sec” pada Halaman Riwayat Per Semester	45
3.32 <i>Form Data</i> pada pengiriman Nilai Seluruh Tahun Akademik	46
3.33 Elemen “a” dengan teks “Kuliah, UTS dan UAS” pada Menu Jadwal	47
3.34 <i>Combo Box</i> “select#tahun_akd_sec” pada Halaman Jadwal Kuliah, UTS, dan UAS	48
3.35 <i>Form Data</i> pada pengiriman Jadwal Kuliah dan Ujian	49
3.36 Elemen “a” dengan teks “Seluruh Fakultas” pada Menu Jadwal	49
3.37 <i>Combo Box</i> “select#jadwal_all_ps” pada Halaman Jadwal Seluruh Fakultas	50
3.38 <i>Form Data</i> pada pengiriman Jadwal Seluruh Fakultas	51
3.39 Elemen “a” dengan teks “logout” pada Identitas Portal	51
3.40 Pengadaluarsaan Cookie CASTGC dan CASPRIVACY	52
3.41 Pengalihan ke Halaman Depan Portal Akademik Mahasiswa	53
3.42 Arsitektur Informatika Student Portal	54
3.43 Diagram <i>Use Case</i> Informatika Student Portal	55
3.44 Diagram Kelas Analisis Informatika Student Portal	57
4.1 Diagram Kelas Rinci	60
4.2 Rancangan Halaman <i>Login</i>	65
4.3 Rancangan Halaman <i>Home</i>	66
4.4 Rancangan Halaman Prasyarat Mata Kuliah	66
4.5 Rancangan Halaman Jadwal Kuliah	67
4.6 Rancangan Halaman Data Akademik	67
5.1 Halaman <i>Login</i>	70
5.2 Halaman Utama	70
5.3 Halaman Prasyarat Mata Kuliah	71
5.4 Halaman Jadwal Kuliah	71
5.5 Rincian Jadwal Kuliah	72
5.6 Halaman Data Akademik	72

DAFTAR TABEL

2.1	Tabel Rincian Kelas pada <i>Package id.ac.unpar.siamodels.matakuliah</i>	25
3.1	Tabel Hasil Analisis Kebutuhan Informatika Student Portal	40
5.1	Tabel Pengujian Fungsional	73

¹

BAB 1

²

PENDAHULUAN

³ 1.1 Latar Belakang

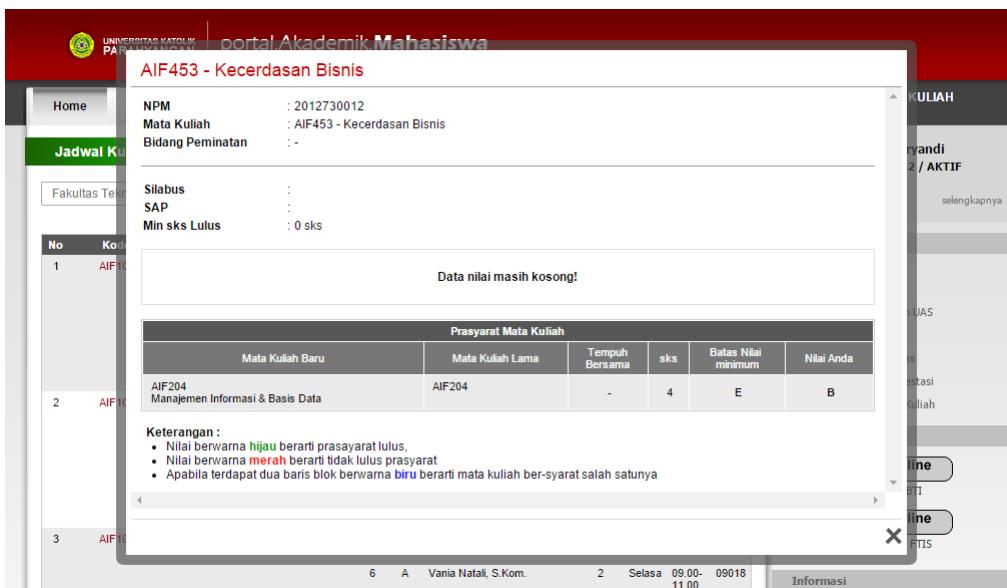
⁴ Portal Akademik Mahasiswa atau dikenal sebagai Student Portal UNPAR^[2] merupakan
⁵ sistem informasi berbasis web yang digunakan oleh mahasiswa Universitas Katolik Para-
⁶ hyangan. Beberapa fitur yang dimiliki Portal Akademik Mahasiswa antara lain rencana
⁷ studi, jadwal, nilai dan indeks prestasi, dan pembayaran uang kuliah. Namun, fitur-fitur
⁸ tersebut masih belum cukup untuk mendukung kebutuhan akademik mahasiswa Program
⁹ Studi Teknik Informatika.

¹⁰ Salah satu fitur yang diperlukan oleh mahasiswa Teknik Informatika UNPAR adalah
¹¹ prasyarat mata kuliah. Dalam Teknik Informatika UNPAR, terdapat beberapa mata kulih-
¹² ah yang membutuhkan prasyarat baik prasyarat tempuh maupun prasyarat lulus. Portal
¹³ Akademik Mahasiswa sudah menyediakan fitur prasyarat mata kuliah namun kurang men-
¹⁴ dukung karena data yang ditampilkan kurang akurat. Misalnya dalam pengambilan mata
¹⁵ kuliah “AIF453 Kecerdasan Bisnis” membutuhkan prasyarat lulus mata kuliah “AIF204 Ma-
¹⁶ najemen Informasi dan Basis Data” atau lulus mata kuliah “AIF102 Algoritma dan Struktur
¹⁷ Data” dengan IPK di atas 2.75 (Gambar 1.1). Namun dalam Portal Akademik Mahasiswa,
¹⁸ prasyarat yang dicantumkan hanya lulus mata kuliah “AIF204 Manajemen Informasi dan
¹⁹ Basis Data” (Gambar 1.2). Selain itu, pemeriksaan prasyarat mata kuliah tidak dilakukan
²⁰ secara otomatis sehingga setiap pengambilan mata kuliah tetap dianggap valid meskipun
²¹ belum memenuhi prasyarat.

²² jsoup^[5] merupakan *library* Java yang digunakan untuk menelusuri suatu situs web untuk
²³ mendapatkan suatu informasi. Informasi yang didapat berupa HTML yang kemudian dieks-
²⁴ trak dan disajikan dalam bentuk *Document Object Model*. Play Framework^[3] merupakan
²⁵ sebuah *web framework* berbasis Java dan Scala. Play juga menggunakan *design pattern*
²⁶ *Model-View-Controller* (MVC) di mana *model* dan *controller* menggunakan bahasa Java se-
²⁷ dangkan *view* menggunakan bahasa Scala dan HTML. SIA Models^[6] merupakan kelas-kelas
²⁸ dalam bahasa Java yang merepresentasikan Sistem Informasi Akademik UNPAR. Aplikasi
²⁹ akan dibuat dengan menggunakan Play Framework dan jsoup karena aplikasi didukung oleh
³⁰ SIA Models yang tersedia dalam bahasa Java.

Prasyarat Mata Kuliah					
4	AIF203	Struktur Diskret	1. T: AIF103 Matematika Diskret		
5	AIF205	Arsitektur & Organisasi Komputer	1. T: AIF106 Sistem Digital		
6	AIF301	Pengantar Sistem Cerdas	1. T: AIF104 Logika Informatika 2. T: AIF202/292 Desain dan Analisis Algoritma		
4	AIF302	Penulisan Ilmiah	1. Lulus minimal 84 SKS		
5	AIF303	Pengantar Sistem Informasi	1. T: AIF204/AIF294 Manaj. Informasi & Basis Data		
6	AIF305	Jaringan Komputer	1. T: AIF206 Sistem Operasi		
	AIF403	Komputer dan Masyarakat	1. Sudah lulus minimal 72 SKS		
7	AIF405	Proyek Sistem Informasi 2	1. T: AIF304 Proyek Sistem Informasi 1		
8	AIF401	Skripsi 1	1. L: AIF302 Penulisan Ilmiah 2. Lulus minimal 108 SKS		
9	AIF402	Skripsi 2	1. L: AIF401 Skripsi 1 ATAU 2. Ambil AIF401 Skripsi 1 & lulus min. 124 SKS		
13	APS402	Etika Profesi	1. Sudah lulus minimal 90 SKS		
Kuliah Pilihan Wajib					
1	AIF311	Pemrograman Fungsional	1. T: AIF103 Matematika Diskret		
2	AIF314	Pemrograman Basisdata	1. T: AIF204/AIF294 Manajemen Informasi & Basis Data		
3	AIF315	Pemrograman Berbasis Web	2. AIF204/AIF294 Manaj. Informasi & Basis Data		
4	AIF317	Desain Antarmuka Grafis	1. T: AIF210 Interaksi Manusia Komputer		
Kuliah Pilihan					
1	AIF438	Penambangan Data	1. T: AIF102/AIF192 Algoritma & Struktur Data 1. T: AIF204/AIF 294 Manaj. Inf. & Basis Data		
2	AIF453	Kecerdasan Bisnis	ATAU 2. T: AIF102/AIF 192 Algo. & Struk. Data & IPK >2.75		
3	AIF457	Kewirausahaan Berbasis Teknologi	1. Lulus minimal 70 SKS		
4	AIF469	Layanan Berbasis Web	1. T: AIF305 Jaringan Komputer 2. T: AIF315 Pemrograman Berbasis Web		
5	AIF362	Sistem & Aplikasi Telematika	1. T: AIF305 Jaringan Komputer		

Gambar 1.1: Prasyarat Mata Kuliah[1]



Gambar 1.2: Prasyarat Mata Kuliah Portal Akademik Mahasiswa[2]

- 1 Untuk mendukung kebutuhan akademik mahasiswa Program Studi Teknik Informatika,
- 2 fitur-fitur yang diperlukan akan dianalisis kemudian diimplementasikan ke dalam sebuah
- 3 program yang akan dinamakan Informatika Student Portal. Informatika Student Portal me-
- 4 rupakan aplikasi berbasis web menggunakan Play Framework. Selain itu, data-data yang
- 5 akan diolah diambil langsung dari Portal Akademik Mahasiswa dengan *web scraping* meng-
- 6 gunakan *library* jsoup. Untuk melakukan pengambilan data, jsoup harus mengetahui cara
- 7 kerja dari Portal Akademik Mahasiswa. Analisis komunikasi Portal Akademik Mahasiswa
- 8 akan dilakukan dengan menggunakan Chrome DevTools.

1 1.2 Rumusan Masalah

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

- 3 1. Fitur-fitur apa saja yang akan dibuat untuk Informatika Student Portal?
- 4 2. Bagaimana mengimplementasikan *web scraping* menggunakan *library jsoup*?
- 5 3. Bagaimana membangun aplikasi Informatika Student Portal?

6 1.3 Tujuan

7 Tujuan-tujuan yang hendak dicapai pada skripsi ini adalah sebagai berikut:

- 8 1. Mengetahui fitur-fitur yang akan dibuat dalam Informatika Student Portal.
- 9 2. Mengimplementasikan *web scraping* menggunakan *library jsoup*.
- 10 3. Membangun aplikasi Informatika Student Portal.

11 1.4 Batasan Masalah

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

- 13 1. Aplikasi akan diuji pada server FTIS sehingga tidak bisa diakses dari luar jaringan FTIS.
- 14 2. Prasyarat mata kuliah yang tersedia hanya mata kuliah yang didukung SIA Models.

16 1.5 Metode Penelitian

17 Metode-metode yang dilakukan pada skripsi ini adalah sebagai berikut:

- 18 1. Melakukan studi mengenai *library jsoup* untuk mengambil data dari Portal Akademik Mahasiswa, CSS Selector yang akan digunakan jsoup untuk menyeleksi data yang akan diambil, Chrome DevTools untuk menganalisis komunikasi Portal Akademik Mahasiswa, dan Play Framework untuk membangun Informatika Student Portal.
- 22 2. Melakukan wawancara.
- 23 3. Menganalisis Portal Akademik Mahasiswa.
- 24 4. Merancang Informatika Student Portal.
- 25 5. Mengimplementasikan *web scraping* menggunakan *library jsoup*.
- 26 6. Melakukan eksperimen dan pengujian.

1.6 Sistematika Penulisan

2 Sistematika penulisan setiap bab pada skripsi ini adalah sebagai berikut:

3 1. Bab Pendahuluan

4 Bab 1 berisi latar belakang, rumusan masalah, tujuan, metode penelitian, dan siste-
5 matika penulisan yang digunakan untuk menyusun skripsi ini.

6 2. Bab Dasar Teori

7 Bab 2 berisikan teori-teori yang digunakan dalam pembuatan skripsi ini. Teori yang
8 digunakan yaitu *library jsoup*, Chrome DevTools, Play Framework, SIA Models, dan
9 CSS *selector*.

10 3. Bab Analisis

11 Bab 3 berisikan analisis yang dilakukan pada skripsi ini, meliputi analisis Portal Aka-
12 demik Mahasiswa, analisis kebutuhan Informatika Student Portal, analisis komunikasi
13 Portal Akademik Mahasiswa untuk fitur Informatika Student Portal, analisis *use case*
14 meliputi diagram *use case* dan skenario, serta analisis kelas.

15 4. Bab Perancangan

16 Bab 4 berisikan perancangan aplikasi, meliputi diagram kelas rinci beserta deskripsi
17 kelas dan fungsinya dan perancangan antarmuka aplikasi.

18 5. Bab Implementasi dan Pengujian

19 Bab 5 berisikan implementasi dan pengujian aplikasi, meliputi lingkungan implemen-
20 tasi, hasil implementasi, pengujian fungsional, dan pengujian eksperimental.

21 6. Bab Kesimpulan dan Saran

22 Bab 6 berisikan kesimpulan dari hasil pembangunan aplikasi beserta saran untuk pe-
23 ngembangan berikutnya.

¹

BAB 2

²

DASAR TEORI

³ 2.1 jsoup

⁴ *Web scraping* adalah teknik mendapatkan informasi dari sebuah situs web secara otomatis^[7].
⁵ Dalam bahasa Java, *web scraping* dapat diimplementasikan menggunakan *library* jsoup^[5].
⁶ jsoup dapat mengambil data berupa HTML dari sebuah situs web. Dalam mengambil
⁷ HTML, jsoup dapat menggunakan CSS *selector* untuk memilih elemen mana saja yang
⁸ ingin diambil.

⁹ Subbab-subbab berikut menjelaskan beberapa kelas dari jsoup.

¹⁰ 2.1.1 Jsoup

¹¹ Kelas ini merupakan inti untuk mengakses fungsi jsoup. Seluruh *method* dalam kelas ini
¹² merupakan **static method** sehingga kelas ini tidak perlu dikonstruksi. Salah satu *method*
¹³ yang dimiliki kelas ini adalah sebagai berikut:

- ¹⁴ • **public static Connection connect(String url)**

¹⁵ Berfungsi untuk membuat koneksi baru dengan suatu situs web.

¹⁶ **Parameter:**

- ¹⁷ – **url** URL situs web dengan protokol HTTP atau HTTPS.

¹⁸ **Kembalian:** koneksi dengan situs web.

¹⁹ 2.1.2 Connection

²⁰ Kelas ini merupakan **interface** yang menyediakan pengambilan data dari situs web. Bebe-
²¹ rapa *method* yang dimiliki kelas ini adalah sebagai berikut:

- ²² • **Connection cookies(Map<String, String> cookies)**

²³ Berfungsi untuk menambahkan *cookie*.

²⁴ **Parameter:**

- ²⁵ – **cookies** Map dari *cookie*.

²⁶ **Kembalian:** koneksi yang sama tetapi sudah diubah.

- ²⁷ • **Connection data(String key, String value)**

²⁸ Berfungsi untuk menambahkan parameter data yang bisa dikirim melalui metode HT-
²⁹ TP GET atau POST.

³⁰ **Parameter:**

- 1 – **key** kunci data.
2 – **value** nilai data.

3 **Kembalian:** koneksi yang sama tetapi sudah diubah.

4 • **Connection method(Connection.Method method)**

5 Berfungsi untuk mengatur metode permintaan HTTP, GET atau POST. Metode pe-
6 ngiriman secara *default* adalah GET

7 **Parameter:**

- 8 – **method** metode pengiriman permintaan HTTP.

9 **Kembalian:** koneksi yang sama tetapi sudah diubah.

10 • **Connection timeout(int millis)**

11 Berfungsi untuk mengatur batas waktu *request*. Batas waktu nol akan dianggap seba-
12 gai batas waktu yang tak terhingga.

13 **Parameter:**

- 14 – **millis** batas waktu dalam milidetik.

15 **Kembalian:** koneksi yang sama tetapi sudah diubah.

16 • **Connection validateTLCertificates(boolean value)**

17 Berfungsi untuk mengatur pemeriksaan sertifikat TLS untuk permintaan HTTPS. Ni-
18 lai **true** untuk memeriksa dan nilai **false** untuk tidak memeriksa.

19 **Parameter:**

- 20 – **value** status pemeriksaan sertifikat TLS.

21 **Kembalian:** koneksi yang sama tetapi sudah diubah.

22 • **Connection.Response execute()**

23 Berfungsi untuk mengirim permintaan HTTP.

24 **Kembalian:** objek Response.

25

2.1.3 Response

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

28 • **Map<String, String> cookies()**

29 *Method* ini berfungsi untuk mendapatkan seluruh *cookies*.

30 **Kembalian:** seluruh *cookies*.

31 • **Document parse()**

32 Berfungsi untuk mengurai *body* jawaban menjadi dokumen.

33 **Kembalian:** koneksi yang sama tetapi sudah diubah.

34 • **String body()**

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

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

1 **2.1.4 Document**

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

- 4 • **public Elements select(String cssQuery)**

5 *Method* ini diturunkan dari kelas Element, berfungsi untuk menemukan elemen HTML
6 yang sesuai dengan kueri CSS.

7 **Parameter:**

- 8 – **cssQuery** kueri CSS berupa CSS Selector.

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

10 **2.1.5 Elements**

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

- 13 • **public Elements select(String query)**

14 Berfungsi untuk menemukan elemen-elemen yang sesuai dalam *list* elemen.

15 **Parameter:**

- 16 – **query** kueri CSS berupa CSS Selector.

17 **Kembalian:** elemen-elemen yang sudah diseleksi sesuai kueri.

- 18 • **public String val()**

19 Berfungsi untuk mendapatkan nilai dari elemen pertama.

20 **Kembalian:** nilai elemen.

- 21 • **public String text()**

22 *Method* Berfungsi untuk mendapatkan kombinasi teks dari seluruh elemen yang sesuai.

23 **Kembalian:** seluruh teks dalam *string*.

24 **2.1.6 Element**

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

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

28 Berfungsi untuk mendapatkan anak elemen berdasarkan nomor indeks.

29 **Parameter:**

- 30 – **index** nomor index.

31 **Kembalian:** anak elemen.

- 32 • **public Element children()**

33 Berfungsi untuk mendapatkan seluruh anak elemen.

34 **Kembalian:** seluruh anak elemen.

- **public String className()**
Berfungsi untuk mendapatkan nama kelas elemen.
Kembalian: nama kelas elemen.

- **public String text()**
Berfungsi untuk mendapatkan teks dari elemen.
Kembalian: teks dalam *string*.

7 2.2 CSS *Selector*

8 CSS(*Cascading Style Sheets*) memungkinkan adanya perubahan elemen HTML yang sudah
9 didefinisikan[8]. Seperti yang ditampilkan pada Gambar 2.1, definisi CSS memiliki dua
10 komponen yaitu selector dan properti. CSS *selector* digunakan untuk mendefinisikan elemen
11 HTML sedangkan properti mendefinisikan atribut beserta nilai.

```

h1{           selector
    color: gray; —————— properti
}

```

Gambar 2.1: Format Penulisan Definisi CSS

12 Beberapa jenis CSS *Selector* antara lain:

- 13 1. **Element Selector**, memilih *tag* html.
Contoh: **h1**
Keterangan: *selector* mendefinisikan elemen h1.

- 16 2. **Grouping Selector**, memilih beberapa *selector* sekaligus. Setiap *selector* dipisahkan
dengan “,”.
Contoh: **h1, h2, p**
Keterangan: *selector* mendefinisikan elemen h1, h2, dan p.

- 20 3. **Universal Selector**, memilih seluruh elemen. *Selector* ditampilkan sebagai “*”.
Contoh: *****
Keterangan: *selector* mendefinisikan seluruh elemen.

- 23 4. **Class Selector**, memilih kelas elemen. *Selector* ditampilkan sebagai “.” kemudian
diikuti nama kelas elemen.
Contoh: **.top**
Keterangan: *selector* mendefinisikan elemen dengan kelas “top”.

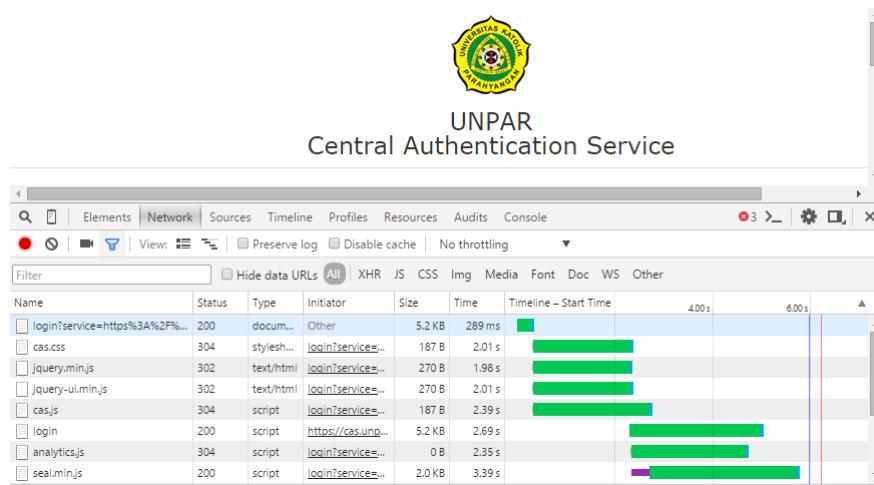
- 27 5. **ID Selector**, memilih ID elemen. *Selector* ditampilkan sebagai “#” kemudian diikuti
ID elemen.
Contoh: **#top**
Keterangan: *selector* mendefinisikan elemen dengan ID “top”.

- 31 6. **Attribute Selector**, akan dijelaskan dua *attribute selector* yaitu:

- 1 • **Simple Attribute**, memilih atribut elemen. *Selector* ditampilkan sebagai nama
2 atribut kemudian diapit dengan kurung siku. Contoh: `[name]`
3 Keterangan: *selector* mendefinisikan elemen dengan atribut “name”.
- 4 • **Exact Value Attribute**, memilih atribut elemen dengan nilai tertentu. *Selector*
5 ditampilkan sebagai definisi atribut kemudian diapit dengan kurung siku. Contoh:
6 `[name=Joe]`
7 Keterangan: *selector* mendefinisikan elemen dengan atribut “name” yang memiliki
8 nilai “Joe”.
- 9 7. **Descendant Selector**, memilih *child* elemen yang merupakan keturunan *parent* ter-
10 tentu. *Selector* ditampilkan dengan mendefinisikan parent kemudian diikuti oleh child
11 dipisahkan dengan spasi.
12 Contoh: `p .top`
13 Keterangan: *selector* mendefinisikan elemen dengan kelas “top” yang merupakan *child*
14 dari elemen p.

15 2.3 Chrome DevTools

- 16 Chrome Developer Tools (DevTools) adalah perangkat *debugging* yang dimiliki Google Chrome[9].
17 Saat menunjungi suatu halaman web, pengguna DevTools dapat melakukan *debugging* pada
18 halaman tersebut. DevTools dapat diakses dengan menekan “Ctrl+Shift+I” saat sedang
19 membuka suatu halaman web.
20 Panel-panel yang dimiliki DevTools (Gambar 2.2) antara lain:
21 1. **Elements**, memeriksa dan mengubah elemen HTML dan *style* dari suatu situs web.
22 2. **Console**, mendapatkan informasi pengembangan dan berinteraksi dengan dokumen.
23 3. **Sources**, melakukan *debugging* pada JavaScript dengan menentukan *breakpoint*.
24 4. **Network**, memantau aktivitas jaringan pada situs web secara *real-time*.
25 5. **Audits**, menganalisa halaman yang dimuat.
26 6. **Timeline**, menampilkan alur waktu saat memuat halaman.
27 7. **Profiles**, menggambarkan waktu eksekusi dan penggunaan memori saat memuat ha-
28 lamana.
29 8. **Resources**, memeriksa sumber daya halaman yang dapat berupa basis data, *cookies*,
30 dan *cache*.



Gambar 2.2: Chrome DevTools

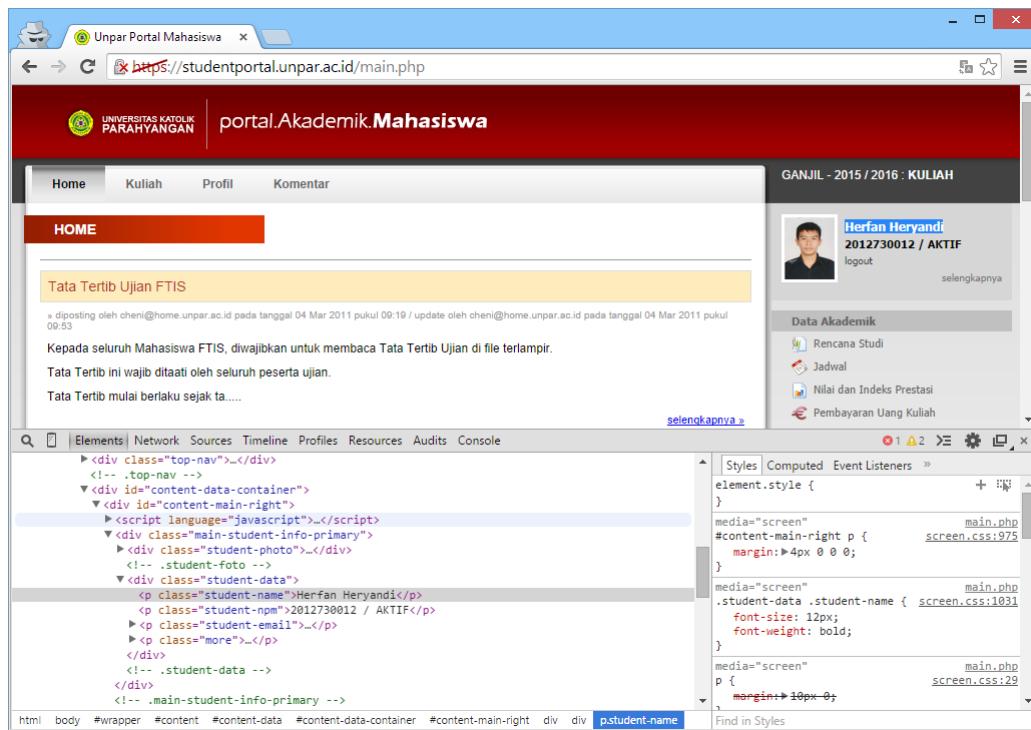
1 Pada subbab-subbab berikut akan dijelaskan mengenai dua panel dari DevTools.

2 2.3.1 Elements

3 Panel Elements memungkinkan untuk memperlihatkan informasi yang terstruktur tentang
4 halaman yang sedang dibuka. HTML akan ditampilkan dalam bentuk pohon *Document
5 Object Model* (DOM). DOM adalah sebuah struktur seperti pohon yang dibuat oleh browser
6 untuk menemukan elemen HTML¹. Tampilan pohon DOM memperlihatkan struktur DOM
7 dari halaman yang sedang dibuka. Pohon DOM adalah pohon dari node-node yang mewakili
8 setiap elemen HTML seperti <body> dan <p>.

9 Pemeriksaan elemen akan memperlihatkan node DOM dan CSS dari elemen yang dipilih
10 pada *browser*. Pemeriksaan elemen dapat dilakukan dengan cara klik kanan pada elemen
11 yang ingin diperiksa kemudian pilih “Inspect element”. Dengan melakukan pemeriksaan ele-
12 men, jendela panel Elements akan muncul. Sebagai contoh pada gambar 2.3, saat melakukan
13 “Inspect element” pada nama mahasiswa, panel Elements akan muncul dan menunjukkan po-
14 hon DOM dari halaman tersebut. Selain itu panel Elements juga menunjukkan CSS selector
15 dari elemen tersebut yaitu p.student-name.

¹<http://try.jquery.com/>, diakses 24 September 2015

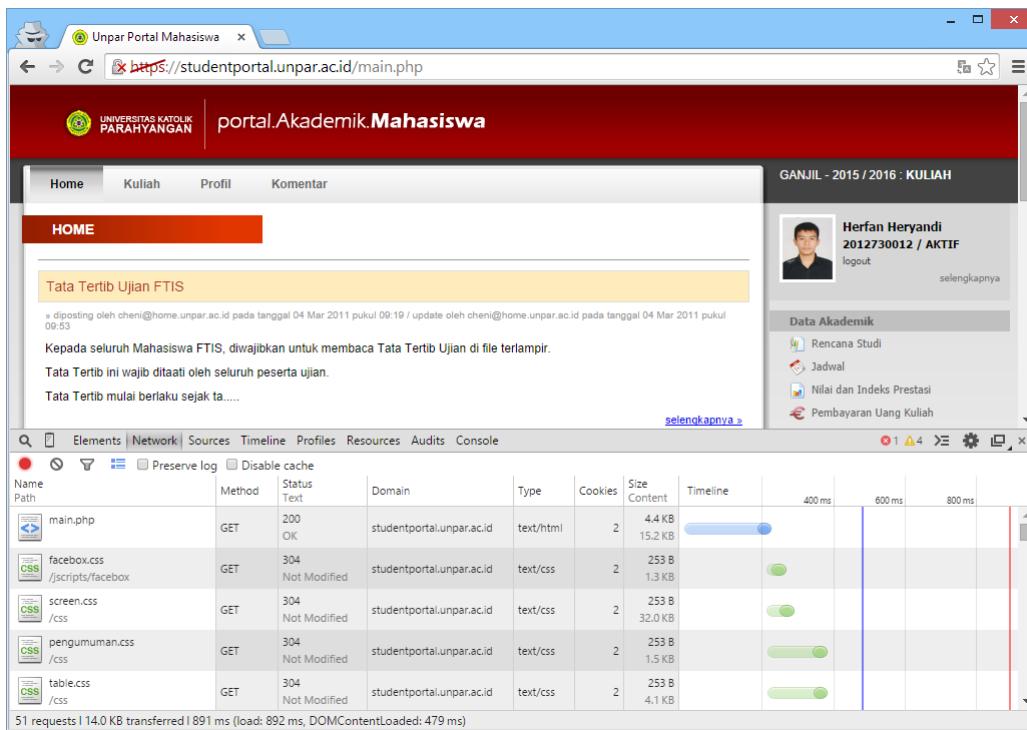


Gambar 2.3: Panel Elements

2.3.2 Network

Panel Network secara otomatis merekam semua aktivitas jaringan saat DevTools terbuka. Pertama kali dibuka, panel Network masih kosong. Halaman web harus dimuat ulang untuk mulai merekam aktivitas jaringan atau menunggu adanya aktivitas jaringan pada halaman web. Panel Network akan mencatat sumber daya dari aktivitas jaringan yang terekam. Setiap sumber daya akan ditambahkan ke dalam sebuah baris dalam tabel Network seperti pada gambar 2.4 dengan rincian kolom sebagai berikut:

- **Name dan Path**, nama dan URL dari sumber daya.
- **Method**, metode permintaan HTTP.
- **Status dan Text**, kode status HTTP dan pesan.
- **Domain**, domain dari sumber daya.
- **Type**, tipe sumber daya yang diminta.
- **Cookies**, banyaknya *cookie* yang dikirim dalam permintaan.
- **Size dan Content**, *size* merupakan ukuran dari *header* dan *body* jawaban yang dikirim server sedangkan *content* merupakan ukuran konten sumber daya.
- **Timeline**, alur waktu dari seluruh aktivitas jaringan yang diminta.

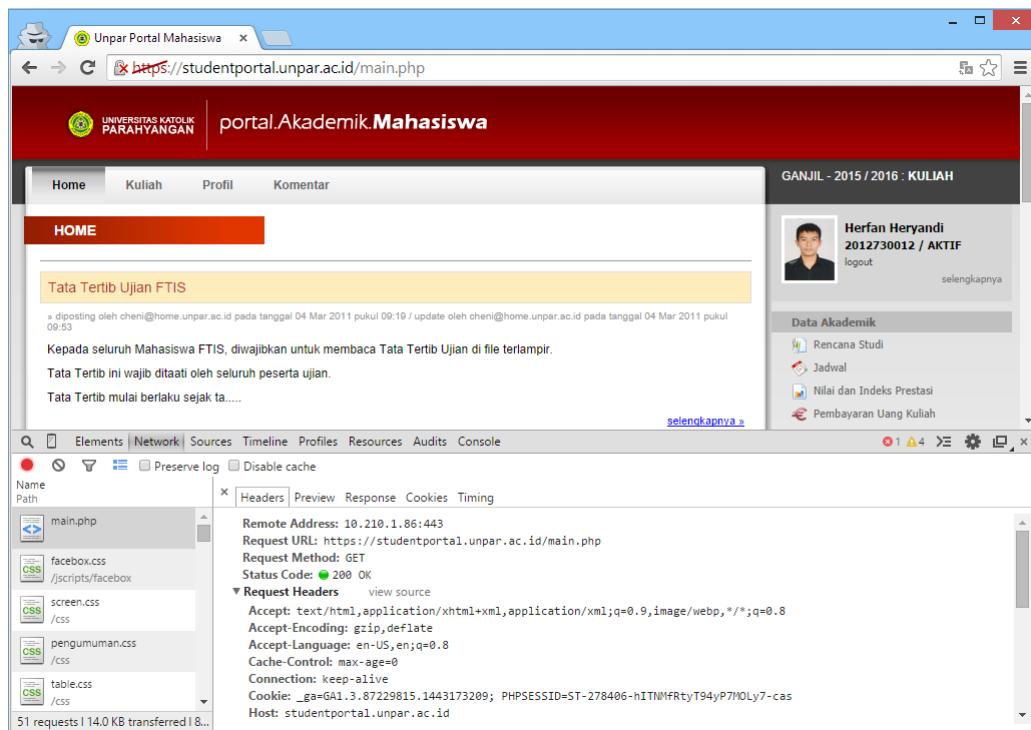


Gambar 2.4: Panel Network

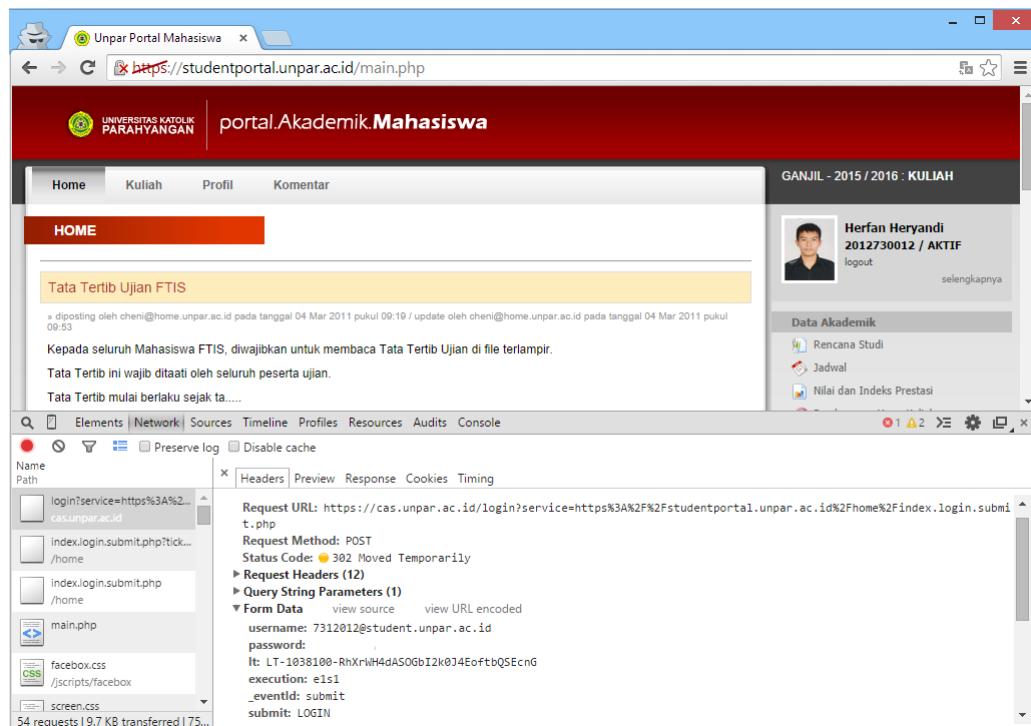
- 1 Ketika nama sumber daya dalam tabel Network diklik, maka akan muncul tautan baru
- 2 yang berisi rincian tambahan sebagai berikut:

• Header

Tautan Header menampilkan *request URL*, *request method*, *status code*, HTTP *response* dan *request header* beserta nilainya, dan *query string parameter*. HTTP header dapat ditampilkan secara terformat atau dalam bentuk sumber dengan mengklik tombol *toggle* “view parsed”/“view source”. Nilai-nilai parameter dapat ditampilkan dalam bentuk yang sudah didekodekan atau dalam bentuk URL yang dienkode dengan mengklik tombol *toggle* “view decoded”/“view URL encoded”. Sebagai contoh pada gambar 2.5 menampilkan *header* pada metode permintaan GET sedangkan gambar 2.6 menampilkan *header* pada metode permintaan POST.



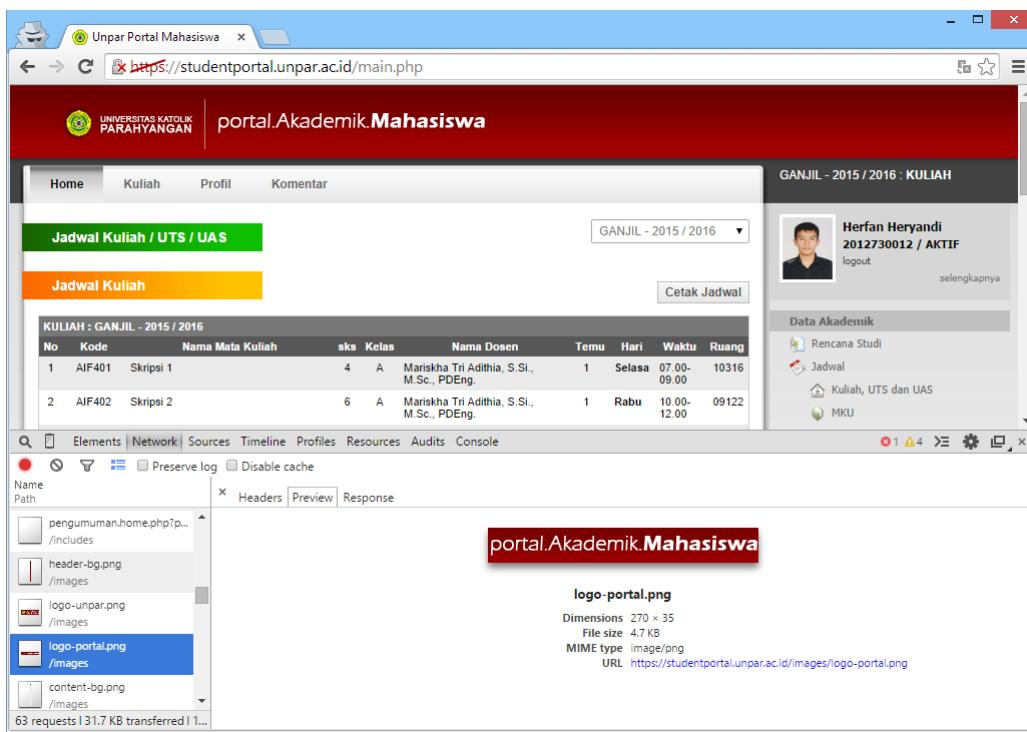
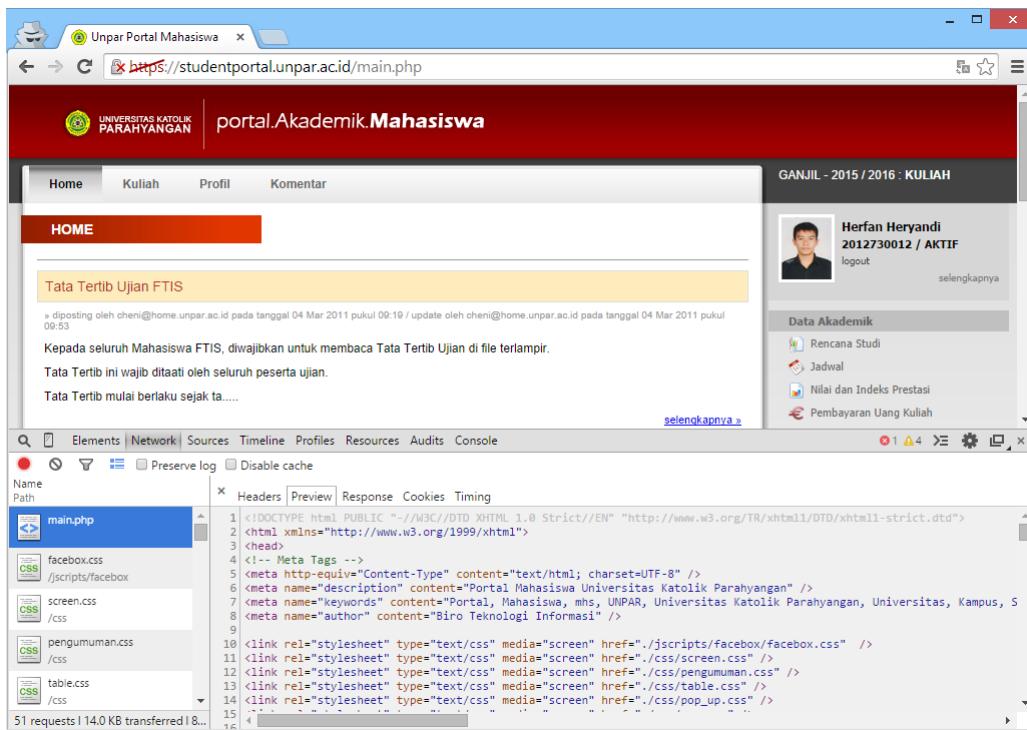
Gambar 2.5: Contoh Tautan Header pada Metode Permintaan GET



Gambar 2.6: Contoh Tautan Header pada Metode Permintaan POST

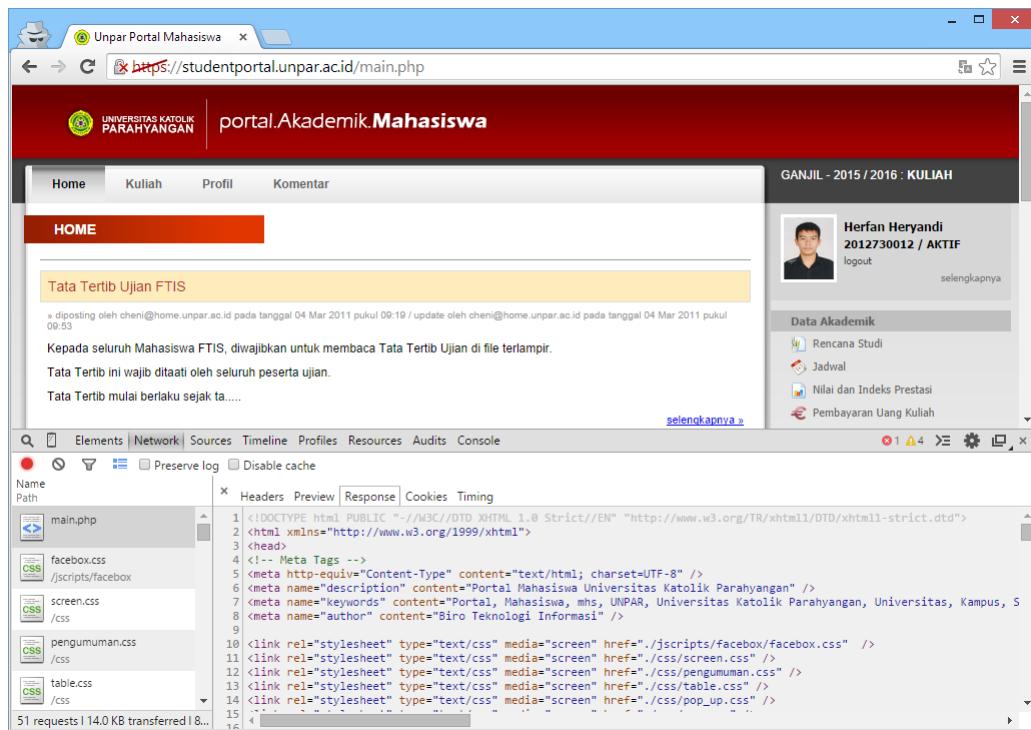
1 • Preview

2 Tautan Preview menampilkan *preview* sumber daya jika tersedia. Gambar 2.7 me-
 3 nampilkan *preview* yang tersedia pada sumber daya. Jika *preview* tidak tersedia maka
 4 akan tampilan akan sama dengan jawaban seperti yang terlihat pada gambar 2.8.

Gambar 2.7: Contoh *Preview* yang TersediaGambar 2.8: Contoh *Preview* yang Tidak Tersedia

• Response

- Tautan Response berisi konten sumber daya yang tidak terformat. Sebagai contoh pada gambar 2.9 menampilkan Tautan Response dari sumber daya `main.php`.

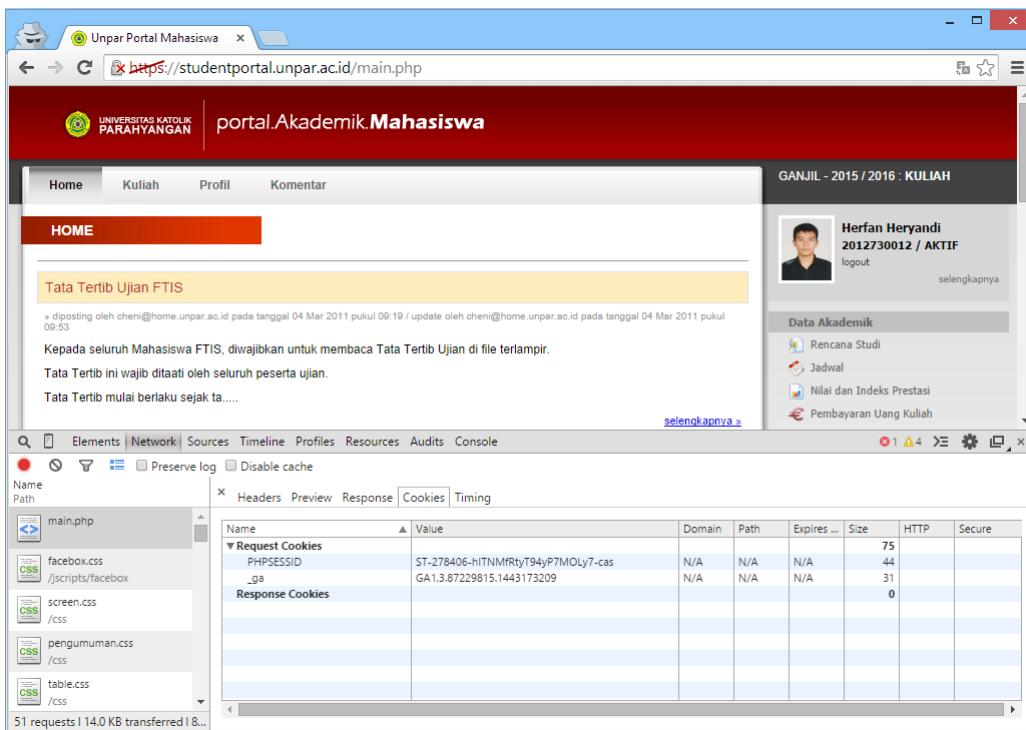


Gambar 2.9: Contoh Tautan Response

• Cookies

Tautan Cookies menampilkan sebuah tabel yang terdiri dari seluruh *cookie* yang di-transmisikan dalam *header* permintaan dan jawaban HTTP. Contoh dari tabel *cookie* dapat dilihat pada gambar 2.10 dengan rincian kolom sebagai berikut:

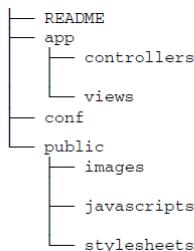
- **Name**, nama *cookie*
- **Value**, nilai *cookie*
- **Domain**, domain yang memiliki *cookie*
- **Path**, URL asal *cookie*
- **Expires/Max-Age**, batas akhir nilai *cookie*
- **Size**, ukuran *cookie* dalam byte
- **HTTP**, menunjukkan bahwa *cookie* harus ditetapkan oleh browser dalam permintaan HTTP, dan tidak dapat diakses dengan JavaScript
- **Secure**, menunjukkan bahwa *cookie* harus dikirim melalui koneksi yang aman



Gambar 2.10: Contoh Tabel pada Tautan Cookie

2.4 Play Framework

- 2 Play Framework[3] merupakan sebuah web *framework* berbasis bahasa Java dan Scala. Play Framework juga menggunakan *design pattern* Model-View-Controller (MVC) di mana *model* dan *controller* menggunakan bahasa Java sedangkan *view* menggunakan bahasa Scala dan HTML. Struktur aplikasi Play Framework dapat dilihat pada gambar 2.11.



Gambar 2.11: Struktur Aplikasi Play Framework

- 6 Dalam direktori **conf**, terdapat file **routes**. Melalui **routes**, rute aplikasi dapat ditentukan dengan memetakan URL ke kode aplikasi. Setiap **route** memiliki tiga bagian yaitu **HTTP method**, **URL path**, dan **action method**. **HTTP method** merupakan metode pengiriman **HTTP**. **URL path** merupakan URL untuk mengakses halaman. **Action method** merupakan **method** yang menangani permintaan metode pengiriman **HTTP**. Sebagai contoh pada Gambar 2.12, setiap permintaan GET pada URL /list akan ditangani oleh **method** **list()** milik kelas **Products** yang terdapat pada *package controllers*.

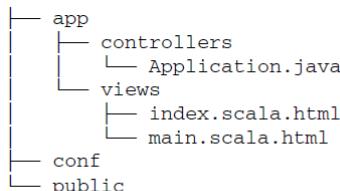


Gambar 2.12: Contoh Komponen Route[3]

1 Direktori `app` merupakan sumber dari kode program seperti file Java dan *view*. Saat
 2 pertama kali proyek Play Framework dibuat, direktori `app` berisi file-file seperti pada gam-
 3 bar 2.13. Dalam folder `controllers`, terdapat file `Application.java` yang berisi kode Java
 4 untuk menghasilkan halaman web. Kelas yang menangani permintaan HTTP dan mengem-
 5 balikan hasil HTTP disebut kelas *controller*. Kelas *controller* merupakan kelas yang memiliki
 6 *action method*. Setiap *action method* memiliki tipe kembalian *Result* yang merepresentasik-
 7 an *view*. Pada buku referensi[3], tertulis bahwa kembalian dari *Action method* harus `static`
 8 tetapi pada versi Play Framework 2.4, `static` dihilangkan. *Action method* kan berhubungan
 9 dengan *view* setelah didefinisikan di `routes`. *Controller* dapat mengirimkan parameter pada
 10 *view* melalui kembalian dari *action method*. Berikut ini adalah contoh *method* pada kelas
 11 *Controller*:

```
12 | public Result home() {
13 |     String nama = mahasiswa.getNama();
14 |     return ok(views.html.home.render(nama));
15 | }
```

16 *Method home()* mengembalikan *view home* yang berada pada *package views* dengan mengi-
 17 rim *parameter* “*nama*”.

Gambar 2.13: Direktori `app` yang Dibangkitkan Play Framework[3]

18 Dalam folder `views` terdapat dua file yaitu `index.scala.html` dan `main.scala.html`
 19 yang berfungsi untuk mendefinisikan halaman HTML. Setiap konten yang dihasilkan pada
 20 server dan dikirim ke klien dalam *body* HTTP, seperti halaman HTML, disebut *view*. *View*
 21 dapat menerima parameter dari *controller* menggunakan bahasa Scala. Berikut contoh
 22 penggunaan bahasa Scala pada *view* dalam penerimaan parameter:

```
23 | @(message: String)
24 |
25 | <!DOCTYPE HTML>
26 |
27 | <html lang="en">
28 |   <head>
29 |     <title>Informatika Student Portal</title>
30 |   </head>
31 |   <body>
32 |     Nama: @message
33 |   </body>
34 | </html>
```

35 Pada baris pertama “*message*” mendefinisikan nama parameter yang diterima dengan tipe
 36 *String*. Tipe yang diterima *view* harus sama dengan tipe yang dikirim *controller* begitu pula

1 banyak parameternya. Baris ke-10 menampilkan “message” pada halaman HTML. Tanda
2 “@” menandakan penggunaan bahasa Scala pada *view*.

3 Direktori **public** berisi sumber yang dapat diakses secara langsung sebagai asset publik.
4 Biasanya asset publik mendukung file selain aplikasi yang dibuat seperti gambar, *stylesheet*,
5 Javascript, dan halaman HTML statis. Aset publik tidak dihasilkan oleh aplikasi melainkan
6 diatur secara langsung oleh pembuat program.

7 Dalam Play Framework, objek yang disimpan pada *session* memiliki masa hidup yaitu
8 selama *browser* dibuka. *Session* tidak disimpan di server melainkan ditambahkan ke setiap
9 permintaan HTTP berikutnya menggunakan mekanisme *cookie*. Ukuran data *session* sangat
10 terbatas yaitu hingga 4 KB sehingga hanya dapat menyimpan *String*. Pada controller,
11 *session* dapat disimpan dengan *method*:

- **public static void session(String key, String value)**

Parameter:

— **key** kunci *session*.

— **value** nilai *session*.

16 Sedangkan nilai *session* dapat diperoleh menggunakan *method*:

- **public static String session(String key)**

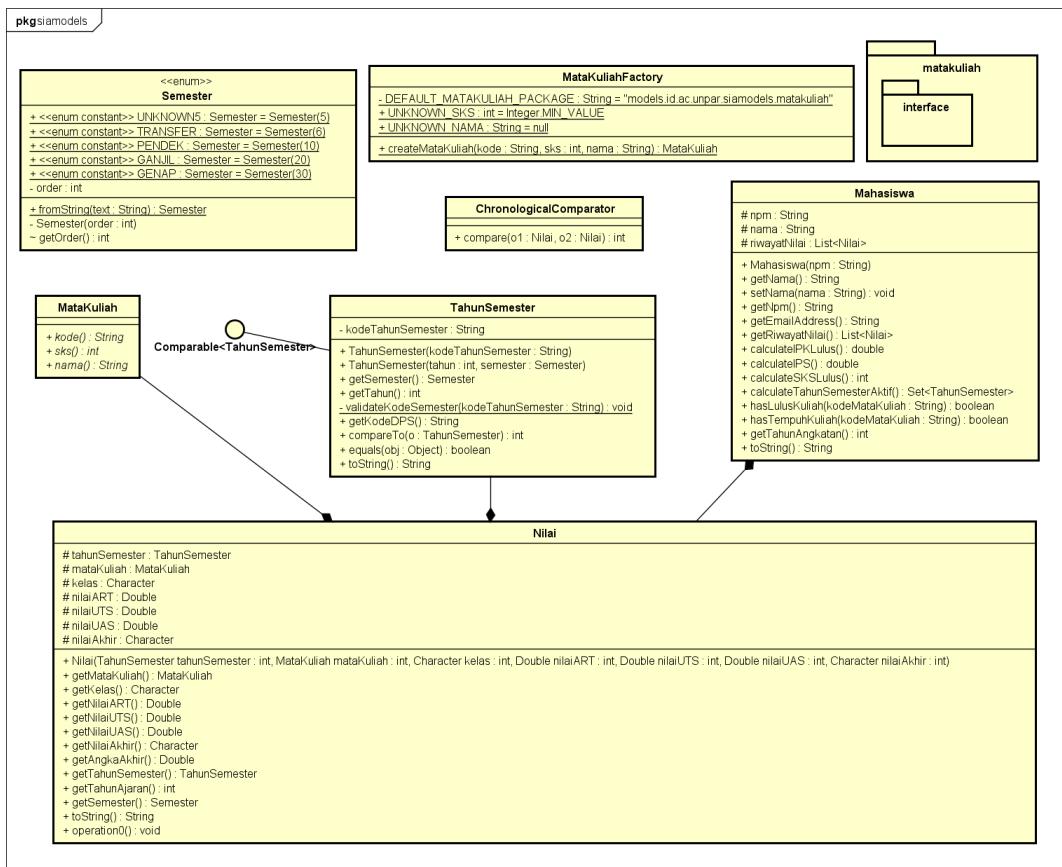
Parameter:

— **key** kunci *session*.

20 **Kembalian:** nilai *session*.

21 **2.5 SIA Models**

22 SIA Models merupakan kelas-kelas dalam bahasa Java yang merepresentasikan Sistem Infor-
23 masi Akademik UNPAR[6]. Berdasarkan diagram kelas SIA Models (Gambar 2.14), kelas-
24 kelas yang dimiliki SIA Models terbagi ke dalam tiga *package* antara lain:



Gambar 2.14: Diagram Kelas SIA Models

1. Package id.ac.unpar.siamodels

2. Package ini memiliki kelas-kelas sebagai berikut:

3. (a) Mahasiswa

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

- 6. • **String npm:** Nomor Pokok Mahasiswa (NPM).
- 7. • **String nama:** nama mahasiswa.
- 8. • **List<Nilai> riwayatNilai:** riwayat nilai yang dimiliki mahasiswa.

9. Method-method yang dimiliki kelas ini adalah sebagai berikut:

- 10. • **public Mahasiswa(String npm)**

11. Merupakan *constructor* dari kelas Mahasiswa.

12. **Parameter:**

- 13. – **npm** nomor pokok mahasiswa.

- 14. • **public String getNama()**

15. Berfungsi untuk mendapatkan nama mahasiswa.

16. **Kembalian:** nama mahasiswa.

- 17. • **public void setNama(String nama)**

18. Berfungsi untuk mengubah nama mahasiswa.

19. **Parameter:**

- 20. – **nama** nama mahasiswa.

- 1 • **public String getNpm()**
2 Berfungsi untuk mendapatkan nomor pokok mahasiswa.
3 **Kembalian:** nomor pokok mahasiswa.
- 4 • **public String getEmailAddress()**
5 Berfungsi untuk mendapatkan *email* mahasiswa.
6 **Kembalian:** *email* mahasiswa.
- 7 • **public List<Nilai> getRiwayatNilai()**
8 Berfungsi untuk mendapatkan riwayat nilai mahasiswa.
9 **Kembalian:** riwayat nilai mahasiswa dalam List.
- 10 • **public double calculateIPKLulus()**
11 Menghitung IPK mahasiswa sampai saat ini, dengan aturan kuliah yang
12 tidak lulus tidak dihitung dan jika pengambilan beberapa kali, diambil nilai
13 terbaik. Sebelum memanggil *method* ini, *getRiwayatNilai()* harus sudah
14 mengandung nilai per mata kuliah.
15 **Kembalian:** IPK lulus.
- 16 • **public double calculateIPS()**
17 Menghitung IPS semester terakhir sampai saat ini, dengan aturan kuliah
18 yang tidak lulus dihitung. Sebelum memanggil *method* ini, *getRiwayatNilai()*
19 harus sudah mengandung nilai per mata kuliah.
20 **Kembalian:** nilai IPS sampai saat ini.
- 21 • **public int calculateSKSLulus()**
22 Menghitung jumlah SKS lulus mahasiswa saat ini. Sebelum memanggil *me-*
23 *thod* ini, *getRiwayatNilai()* harus sudah mengandung nilai per mata kuliah.
24 **Kembalian:** SKS lulus.
- 25 • **public Set<TahunSemester> calculateTahunSemesterAktif()**
26 Mendapatkan seluruh tahun semester di mana mahasiswa ini tercatat sebagai
27 mahasiswa aktif, dengan strategi memeriksa riwayat nilainya.Jika ada satu
28 nilai saja pada sebuah tahun semester, maka dianggap aktif pada semester
29 tersebut.
30 **Kembalian:** kumpulan tahun semester di mana mahasiswa ini aktif.
- 31 • **public boolean hasLulusKuliah(String kodeMataKuliah)**
32 Memeriksa apakah mahasiswa ini sudah lulus mata kuliah tertentu. Sebelum
33 memanggil *method* ini, *getRiwayatNilai()* harus sudah mengandung nilai
34 per mata kuliah.
35 **Parameter:**
 - 36 – **kodeMataKuliah** kode mata kuliah yang ingin diperiksa kelulusannya.
Kembalian: *true* jika sudah pernah mengambil dan lulus, *false* jika belum.
- 38 • **public boolean hasTempuhKuliah(String kodeMataKuliah)**
39 Memeriksa apakah mahasiswa ini sudah pernah menempuh mata kuliah ter-
40 tentu. Sebelum memanggil *method* ini, *getRiwayatNilai()* harus sudah
41 mengandung nilai per mata kuliah.
42 **Parameter:**
 - 43 – **kodeMataKuliah** kode mata kuliah yang ingin diperiksa kelulusannya.

- 1 **Kembalian:** `true` jika sudah pernah mengambil, `false` jika belum.
- 2 • **public int getTahunAngkatan()**
3 Mendapatkan tahun angkatan mahasiswa ini berdasarkan NPM-nya.
4 **Kembalian:** tahun angkatan.
- 5 (b) Nilai
6 Kelas ini merepresentasikan nilai yang ada pada riwayat nilai mahasiswa. Atribut
7 yang dimiliki kelas ini antara lain:
8 • **TahunSemester tahunSemester:** tahun dan semester kuliah ini diambil
9 • **MataKuliah mataKuliah:** mata kuliah yang diambil.
10 • **Character kelas:** kelas kuliah.
11 • **Double nilaiART:** nilai Angka Rata-rata Tugas (ART).
12 • **Double nilaiUTS:** nilai Ujian Tengah Semester (UTS).
13 • **Double nilaiUAS:** nilai Ujian Akhir Semester (UAS).
14 • **Character nilaiAkhir:** nilai akhir.
- 15 *Method-method* yang dimiliki kelas ini adalah sebagai berikut:
16 • **public Nilai(TahunSemester tahunSemester, MataKuliah mataKuliah, Character kelas, Double nilaiART, Double nilaiUTS, Double nilaiUAS, Character nilaiAkhir)**
17 Merupakan *constructor* dari kelas Nilai.
18
19 **Parameter:**
20 – **tahunSemester** tahun dan semester kuliah ini diambil.
21 – **mataKuliah** mata kuliah yang diambil.
22 – **kelas** kelas kuliah.
23 – **nilaiART** nilai ART.
24 – **nilaiUTS** nilai UTS.
25 – **nilaiUAS** nilai UAS.
26 – **nilaiAkhir** nilai akhir.
27
28 • **public MataKuliah getMataKuliah()**
29 Mendapatkan mata kuliah yang diambil.
30 **Kembalian:** mata kuliah.
31 • **public Character getKelas()**
32 Mendapatkan kelas kuliah.
33 **Kembalian:** kelas kuliah.
34 • **public Double getNilaiART()**
35 Mendapatkan nilai ART.
36 **Kembalian:** nilai ART.
37 • **public Double getNilaiUTS()**
38 Mendapatkan nilai UTS.
39 **Kembalian:** nilai UTS.
40 • **public Double getNilaiUAS()**
41 Mendapatkan nilai UAS.
42 **Kembalian:** nilai UAS.

- 1 • **public Double getNilaikhir()**
2 Mendapatkan nilai akhir dalam bentuk angka.
3 **Kembalian:** nilai akhir dalam huruf atau **null** jika tidak ada.
- 4 • **public Double getAngkaAkhir()**
5 Mengembalikan nilai akhir dalam bentuk huruf (A, B, C, D, ...).
6 **Kembalian:** nilai akhir dalam angka, atau **null** jika **getNilaiAkhir()**
7 mengembalikan **null**.
- 8 • **public int getTahunAjaran()**
9 Mendapatkan tahun ajaran saat pengambilan mata kuliah.
10 **Kembalian:** tahun ajaran saat pengambilan mata kuliah.
- 11 • **public TahunSemester getTahunSemester()**
12 Mendapatkan tahun dan semester pengambilan mata kuliah.
13 **Kembalian:** tahun dan semester pengambilan mata kuliah.
- 14 • **public Semester getSemester()**
15 Mendapatkan semester pengambilan mata kuliah.
16 **Kembalian:** semester pengambilan mata kuliah

17 (c) ChronologicalComparator

18 Pembanding antara satu nilai dengan nilai lainnya, secara kronologis waktu pe-
19 ngambilan. *Method* yang dimiliki kelas ini adalah sebagai berikut:

- 20 • **public int compare(Nilai o1, Nilai o2)**
21 Berfungsi untuk membandingkan nilai.
22 **Parameter:**
 - 23 – **o1** nilai pertama yang akan dibandingkan.
 - 24 – **o2** nilai kedua yang akan dibandingkan.
25 **Kembalian:** hasil perbandingan.

26 (d) MataKuliah

27 Kelas ini merepresentasikan sebuah mata kuliah. *Method-method* yang dimiliki
28 kelas ini adalah sebagai berikut:

- 29 • **public String kode()**
30 Mendapatkan kode mata kuliah sesuai dengan nama kelas mata kuliah ter-
31 sebut.
32 **Kembalian:** kode mata kuliah.
- 33 • **public int sks()**
34 Mendapatkan bobot sks.
35 **Kembalian:** bobot SKS.
- 36 • **public String kode()**
37 Mendapatkan nama mata kuliah.
38 **Kembalian:** nama mata kuliah.

39 (e) MataKuliahFactory

40 Kelas ini berperan dalam pembuatan objek mata kuliah baru. Atribut yang
41 dimiliki kelas ini antara lain:

- 42 • **String DEFAULT_MATAKULIAH_PACKAGE:** lokasi *package* un-
43 tuk daftar mata kuliah.

- 1 • **int UNKNOWN_SKS:** menandakan jumlah SKS tidak diketahui.
- 2 • **String UNKNOWN_NAMA:** menandakan nama mata kuliah tidak di-
- 3 ketahui.

4 *Method* yang dimiliki kelas ini adalah sebagai berikut:

- 5 • **public static MataKuliah createMataKuliah(String kode, int sks,**
- 6 **String nama)**

7 Membuat objek mata kuliah baru. Jika memungkinkan mengambil dari kelas
8 yang sudah ada.

9 **Parameter:**

- 10 – **kode** kode mata kuliah.
- 11 – **sks** bobot SKS mata kuliah, dapat diisi dengan UNKNOWN_SKS jika tidak
12 diketahui.
- 13 – **nama** nama mata kuliah, dapat diisi dengan.

14 **Kembalian:** objek mata kuliah UNKNOWN_NAMA jika tidak diketahui.

15 (f) Semester

16 Kelas ini merepresentasikan semester *Method* yang dimiliki kelas ini adalah seba-
17 gai berikut:

- 18 • **public static final Semester fromString(String text)**

19 Berfungsi untuk mengubah semester dari bentuk teks ke konstanta.

20 **Parameter:**

- 21 – **text** semester dalam bentuk teks (GANJIL, GENAP, PENDEK, TRANS-
22 FER, dan UNKNOWN5).

23 **Kembalian:** konstanta semester.

24 (g) TahunSemester

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

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

30 *Method-method* yang dimiliki kelas ini adalah sebagai berikut:

- 31 • **public TahunSemester(String kodeTahunSemester)**

32 *Method* ini merupakan constructor dari kelas TahunSemester.

33 **Parameter:**

- 34 – **kodeTahunSemester** semester dalam bentuk teks (GANJIL, GENAP,
35 PENDEK, TRANSFER, dan UNKNOWN5).

- 36 • **public TahunSemester(int tahun, Semester semester)**

37 *Method* ini merupakan constructor dari kelas TahunSemester.

38 **Parameter:**

- 39 – **tahun** tahun ajaran.
- 40 – **semester** semester dari tahun ajaran.

- 1 • **public Semester getSemester()**
2 *Method* ini berfungsi untuk mendapatkan semester.
3 **Kembalian:** semester dalam teks.
- 4 • **public int getTahun()**
5 *Method* ini berfungsi untuk mendapatkan tahun.
6 **Kembalian:** tahun ajaran.
- 7 • **private static void validateKodeSemester(String kodeTahunSemester)**
8 *Method* ini berfungsi untuk melakukan validasi terhadap kode tahun semester.
9
10
11 **Parameter:**
12 – **kodeTahunSemester** kode tahun semester.

13 2. *Package id.ac.unpar.siamodels.matakuliah.interfaces*

14 *Package* ini memiliki beberapa *interface* antara lain:

15 (a) **HasPrasyarat**

16 Mendefinisikan kelas-kelas yang memiliki prasyarat, terkustomisasi untuk seorang
17 mahasiswa. *Method* yang dimiliki *interface* ini adalah sebagai berikut:

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

19 Memeriksa prasyarat-prasyarat dari kuliah, spesifik untuk mahasiswa yang
20 dituju. Jika ada pesan-pesan khusus, akan ditambahkan pada parameter rea-
21 sonsContainer.
22

23 **Parameter:**

- 24 – **mahasiswa** prasyarat kuliah akan diperiksa spesifik pada mahasiswa ini.
- 25 – **reasonsContainer** jika pesan-pesan terkait prasyarat akan ditambahkan
26 di sini.

27 **Kembalian:** `true` jika seluruh prasyarat dipenuhi, `false` jika tidak.

28 (b) **Pilihan**

29 Mendefinisikan kelas-kelas yang merupakan mata kuliah pilihan.

30 (c) **PilihanWajib**

31 Mendefinisikan kelas-kelas yang merupakan mata kuliah pilihan wajib.

32 (d) **Wajib**

33 Mendefinisikan kelas-kelas yang merupakan mata kuliah wajib.

34 3. *Package id.ac.unpar.siamodels.matakuliah*

35 *Package* ini berisi kelas-kelas yang merepresentasikan mata kuliah yang terdapat pada
36 Program Studi Teknik Informatika UNPAR beserta aturan prasyaratnya. Rincian dari
37 kelas-kelas pada package ini dapat dilihat pada tabel 2.1.

Tabel 2.1: Tabel Rincian Kelas pada *Package id.ac.unpar.siamodels.matakuliah*

Kelas	<i>Implements</i>	Kelas	<i>Implements</i>
AIF101	-	AIF342	HasPrasyarat
AIF102	HasPrasyarat, Wajib	AIF344	HasPrasyarat
AIF103	-	AIF360	HasPrasyarat
AIF105	-	AIF362	HasPrasyarat, Pilihan
AIF200	-	AIF401	HasPrasyarat, Wajib
AIF201	HasPrasyarat, Wajib	AIF402	HasPrasyarat, Wajib
AIF202	HasPrasyarat, Wajib	AIF403	-
AIF203	HasPrasyarat, Wajib	AIF405	HasPrasyarat, Wajib
AIF204	HasPrasyarat	AIF438	HasPrasyarat, Pilihan
AIF205	HasPrasyarat, Wajib	AIF441	-
AIF206	HasPrasyarat	AIF445	HasPrasyarat
AIF208	HasPrasyarat	AIF453	HasPrasyarat, Pilihan
AIF301	HasPrasyarat, Wajib	AIF456	-
AIF302	HasPrasyarat, Wajib	AIF457	HasPrasyarat, Pilihan
AIF303	HasPrasyarat, Wajib	AIF458	HasPrasyarat
AIF304	HasPrasyarat	AIF461	HasPrasyarat
AIF305	HasPrasyarat, Wajib	AIF462	-
AIF306	HasPrasyarat	AIF469	HasPrasyarat, Pilihan
AIF311	HasPrasyarat, PilihanWajib	APS402	HasPrasyarat, Wajib
AIF312	HasPrasyarat	MKU001	-
AIF314	HasPrasyarat, PilihanWajib	MKU002	-
AIF315	HasPrasyarat, PilihanWajib	MKU003	-
AIF316	HasPrasyarat	MKU004	-
AIF317	HasPrasyarat, PilihanWajib	MKU008	-
AIF318	HasPrasyarat	MKU009	-
AIF332	HasPrasyarat	MKU010	-
AIF336	-	MKU011	-
AIF339	HasPrasyarat	MKU012	-
AIF341	-		

1

BAB 3

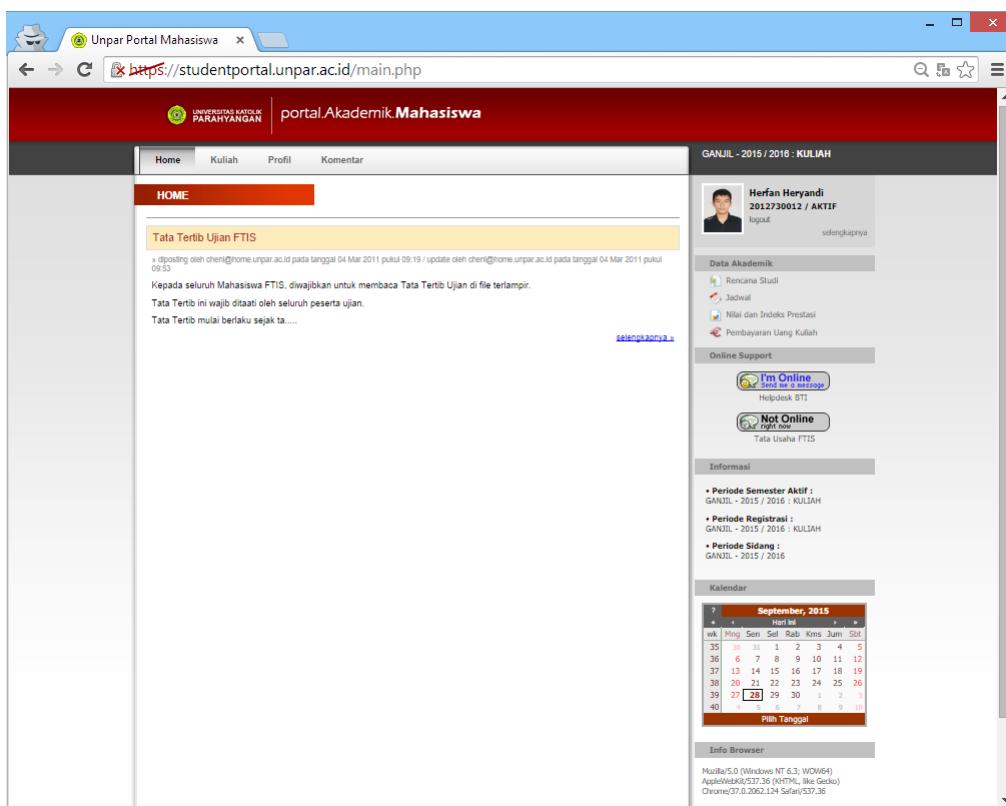
2

ANALISIS

3

3.1 Analisis Portal Akademik Mahasiswa

- 4 Portal Akademik Mahasiswa merupakan sebuah situs jaringan yang diperuntukan bagi mahasiswa dalam rangka mendapatkan informasi kegiatan akademik[4]. Mahasiswa dapat mengakses Portal Akademik Mahasiswa melalui URL <https://studentportal.unpar.ac.id/>.
- 5 Untuk mengakses Portal Akademik Mahasiswa, mahasiswa harus *login* menggunakan akun email *student*. Halaman *login* Student Portal UNPAR terintegrasi dengan CAS (*Central Authentication Service*) UNPAR¹.



Gambar 3.1: Halaman Utama Portal Akademik Mahasiswa

10 Pada halaman utama Portal Akademik Mahasiswa (gambar 3.1), terdapat beberapa
11 bagian yaitu:

12 1. Menu Atas

13 Menu ini berfungsi sebagai menu pendukung yang terdiri dari :

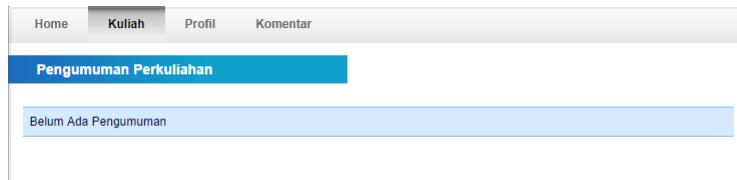
¹<https://cas.unpar.ac.id>

- 1 • **Home**, menampilkan informasi atau pengumuman yang dikeluarkan oleh fakultas
2 masing-masing (Gambar 3.2).



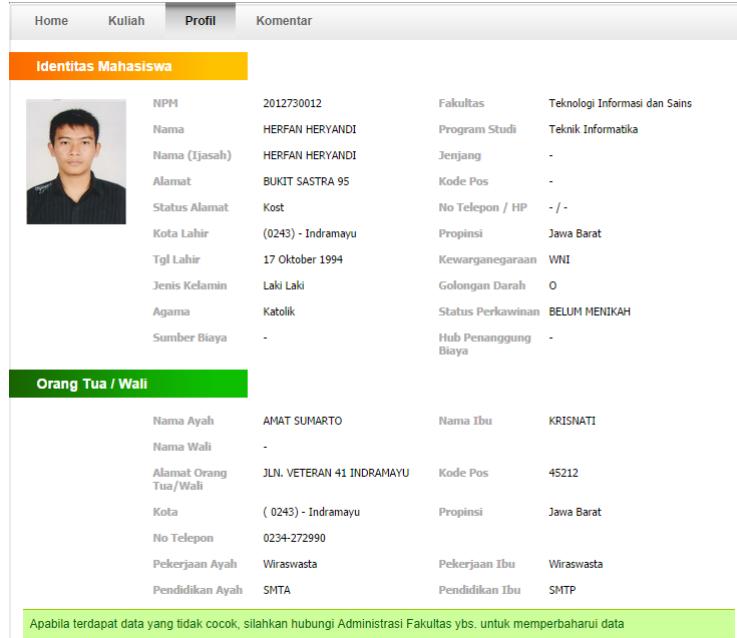
Gambar 3.2: Menu Atas Home

- 3 • **Kuliah**, menampilkan pengumuman per mata kuliah sesuai dengan mata kuliah
4 dan kelas yang diambil oleh masing-masing mahasiswa (Gambar 3.3).



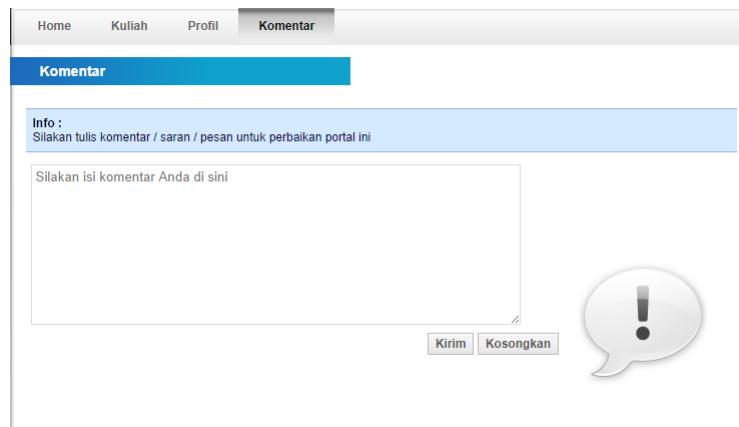
Gambar 3.3: Menu Atas Kuliah

- 5 • **Profil**, berisi tentang data diri masing-masing mahasiswa (Gambar 3.4).



Gambar 3.4: Menu Atas Profil

- 6 • **Komentar**, berisi komentar, saran, dan kritik dari mahasiswa (Gambar 3.5).



Gambar 3.5: Menu Atas Komentar

1 2. Identitas Portal

2 Bagian ini menampilkan identitas pengguna portal. Tampilan identitas ini dapat ditampilkan lengkap dengan melakukan klik pada link “selengkapnya” atau ditampilkan minimal dengan klik link “tutup”. Identitas yang ditampilkan adalah nama, Nomor Pokok Mahasiswa (NPM), status keaktifan, pas foto, email, dosen wali, program studi, dan fakultas seperti yang terlihat pada gambar 3.6.



Gambar 3.6: Identitas Portal

7 3. Menu Utama

8 Bagian ini memuat fitur utama Portal Akademik Mahasiswa mengenai data akademik (gambar 3.7) yang terdiri dari:



Gambar 3.7: Menu Utama

10 • **Rencana Studi**

11 Menu Rencana Studi terdiri dari submenu:

- 12 – Registrasi (FRS/PRS)

- 1 Digunakan sebagai formulir pengisian rencana studi awal (FRS) dan perubahan rencana studi (PRS) (Gambar 3.8).

No.	Kode MK	Nama MK	Pada Pengampuan	Pilih FRS	Pilih PRS
1	EEP101-3	Pengantar Ekonomi Mikro		<input checked="" type="checkbox"/>	-
2	EMS101-3	Pengantar Bisnis		<input checked="" type="checkbox"/>	-
3	MHU001-2	Pendidikan Pancasila		<input checked="" type="checkbox"/>	-
4	MHU002-2	Pendidikan Kinerjanegeoran		<input checked="" type="checkbox"/>	-
Mata Kuliah Pilihan					
5	EAR205-0	Responsi Akuntansi Keuangan Lanjut I	Akuntansi	<input checked="" type="checkbox"/>	-
6	EMS205-3	Perilaku Organisasi	Manajemen	<input checked="" type="checkbox"/>	-
7	EMS305-3	Statistika Non Parametrik	Manajemen	<input checked="" type="checkbox"/>	-
8	EMS306-3	Kewirausahaan	Manajemen	<input checked="" type="checkbox"/>	-
Total Sks			19	0	
Total Mata Kuliah			8	0	
Total Sks Akhir			19	sks	
Total Mata Kuliah Akhir			8	mata kuliah	

[Simpan](#) [Kembali](#)

Gambar 3.8: Tampilan Registrasi FRS/PRS

3 – Kartu Rencana Studi

4 Menampilkan informasi mata kuliah yang telah diambil melalui submenu
5 Registrasi (Gambar 3.9). Kartu Rencana Studi juga dapat dicetak melalui
6 submenu ini.

No.	Kode MK	Nama Matakuliah	sks	Kelas	SAPR
1	AIF401	Skripsi 1	4	A	
2	AIF402	Skripsi 2	6	A	
3	AIF439	Kerja Praktek 2	3	A	
4	AIF441	Administrasi Jaringan Komputer 3	3	A	P
5	AIF457	Kewirausahaan Berbasis Teknologi	3	A	
6	AIF469	Layanan Berbasis Web	3	A	
Total sks			22		

* Anda sudah pernah melakukan registrasi pada tanggal : 6 Juli 2015

[Cetak KRS](#)

Keterangan :
S : Studio
A : Asistensi
P : Praktikum
R : Responsi

Gambar 3.9: Tampilan Kartu Rencana Studi[4]

7 – Pindah Kelas MKU

8 Mahasiswa dapat memilih kelas yang masih tersedia di kolom Jadwal Baru
9 dan menekan tombol “Simpan” untuk setiap kelas yang diubah (Gambar
10 3.10).

Ubah Jadwal MKU Anda :
GANJIL - 2010 / 2011

Matakuliah	Jadwal Sekarang	Jadwal Baru
MKU001 - Pendidikan Pancasila	Kelas : C Jadwal : Kamis, 9.00 - 11.00 Ruang : 10118 Dosen :	Kelas : D (0/50) <input type="button" value="▼"/> <input type="button" value="Simpan"/> Jadwal : Kamis, 11.00 - 13.00 Ruang : 10118 Dosen : Kapasitas : 50 Peserta :
MKU002 - Pendidikan Kewarganegaraan	Kelas : A Jadwal : Senin, 8.00 - 10.00 Ruang : 2131 Dosen :	Kelas : Tidak Berubah <input type="button" value="▼"/> <input type="button" value="Simpan"/> Jadwal : - Ruang : - Dosen : - Kapasitas : - Peserta : -

Gambar 3.10: Tampilan Pindah Kelas MKU[4]

1 ● **Jadwal**

2 Menu Jadwal terdiri dari submenu:

- 3 – Kuliah, UTS, dan UAS

4 Submenu ini berisi tentang jadwal kuliah, UTS dan UAS yang dapat disusun per semester (Gambar 3.11).

Jadwal Kuliah / UTS / UAS

GANJIL - 2015 / 2016

Jadwal Kuliah

KULIAH : GANJIL - 2015 / 2016

No	Kode	Nama Mata Kuliah	skls	Kelas	Nama Dosen	Temu	Hari	Waktu	Ruang
1	AIF401	Skripsi 1	4	A	Marikha Tri Adithia, S.Si., M.Sc., PDEng.	1	Selasa	07.00-09.00	10316
2	AIF402	Skripsi 2	6	A	Marikha Tri Adithia, S.Si., M.Sc., PDEng.	1	Rabu	10.00-12.00	09122
3	AIF439	Keja Praktek 2	3	A	-		1	00.00-00.00	00000
4	AIF441	Administrasi Jaringan Komputer 3	3	A	Chandra Wijaya, S.T., M.T.	1	Kamis	07.00-10.00	09016
			3	A	Billy Susanto Panca, S.T.	2	Senin	15.00-17.00	09016
5	AIF457	Kewirausahaan Berbasis Teknologi	3	A	Dr. Veronica Sri Moertini, Ir., M.T.	1	Jumat	08.00-11.00	10317
6	AIF469	Layanan Berbasis Web	3	A	Pascal Alfadian, S.Kom., M.Com.	1	Kamis	13.00-16.00	09122
		Total sks	22	skls					

Jadwal UTS

UTS : GANJIL - 2015 / 2016

No	Kode	Nama Mata Kuliah	skls	Kelas	Tanggal Ujian	Waktu	Ruang	Kursi
1	AIF441	Administrasi Jaringan Komputer 3	3	A	13 OCT 15	08.00-10.00	09016	0007
2	AIF469	Layanan Berbasis Web	3	A	16 OCT 15	13.30-15.30	09120	0025

Jadwal UAS

Jadwal UAS untuk tahun akademik : GANJIL - 2015 / 2016 belum tersedia.

Gambar 3.11: Tampilan Jadwal Kuliah, UTS, dan UAS

6 – MKU

7 Submenu ini menampilkan seluruh jadwal Mata Kuliah Umum (MKU) yang memberikan informasi tentang kelas-kelas yang dibuka oleh Pusat Kajian

1 Humaniora (PKH) (Gambar 3.12).

Daftar Kelas & Jadwal MKU Semester Ini					
Silahkan klik Kode Mata Kuliah untuk melihat jadwal.					
MKU001 - Pendidikan Pancasila					
MKU002 - Pendidikan Kewarganegaraan					
MKU003 - Pendidikan Agama (Katolik)					
MKU004 - Pendidikan Agama (Fenomenologi)					
MKU008 - Etika					
MKU011 - Estetika					
MKU012 - Logika					
Kelas	Hari	Waktu	Ruang	Dosen	
1	Jumat	08.00 - 09.40	10119	Andreas Doweng Bolo, S.S., M.Hum.	
%	Jumat	10.00 - 11.40	10119	Bernardus Arto Tejo Sugiantoro, S.S., M.Hum.	
(Jumat	13.00 - 14.40	10119	Rudi Setiawan, S.Ag., M.M.	
)	Senin	07.00 - 08.40	10118	Yusuf Siswantara, S.S., M.Hum.	
1	Senin	15.00 - 16.40	10112	Yusuf Siswantara, S.S., M.Hum.	
2	Selasa	15.00 - 16.40	10112	Yusuf Siswantara, S.S., M.Hum.	
3	Rabu	10.00 - 11.40	10118	Yusuf Siswantara, S.S., M.Hum.	
4	Jumat	10.00 - 11.40	10116	Kristian Widya Wicaksono, S.Sos. M.Si.	
5	Senin	08.00 - 09.40	10116	Adelia, S.S., M.Si.	
6	Senin	10.00 - 11.40	10116	Adelia, S.S., M.Si.	
7	Senin	13.00 - 14.40	10116	Adelia, S.S., M.Si.	
8	Senin	15.00 - 16.40	10116	Adelia, S.S., M.Si.	
9	Selasa	08.00 - 09.40	10117	Adelia, S.S., M.Si.	
A	Kamis	10.00 - 12.00	06113	Dr. Stephanus Djunafan	
B	Selasa	10.00 - 11.40	10117	Adelia, S.S., M.Si.	
C	Selasa	13.00 - 14.40	10117	Adelia, S.S., M.Si.	
D	Selasa	09.00 - 10.40	10114	Aloysius Oscar Yasunari, S.S., M.M.	
E	Selasa	11.00 - 12.40	10114	Aloysius Oscar Yasunari, S.S., M.M.	

Gambar 3.12: Tampilan Jadwal MKU

2 – Seluruh Fakultas

3 Fitur ini memberikan informasi mengenai jadwal-jadwal yang ada di seluruh
4 fakultas (Gambar 3.13).

Jadwal Kuliah										
Fakultas Teknologi Informasi dan Sains			GANJIL - 2015 / 2016							
No	Kode	Nama Mata Kuliah	skS	Kelas	Nama Dosen	Temu	Hari	Waktu	Ruang	
1	AIF101	Pemrograman Berorientasi Objek	6	A	Husnul Hakim, S.Kom., M.T.	1	Senin	08.00-09.00	09122	
			6	A	Vania Natali, S.Kom.	2	Selasa	09.00-10.00	09018	
			6	A	Husnul Hakim, S.Kom., M.T.	3	Rabu	08.00-09.00	09018	
			6	A	Vania Natali, S.Kom.	4	Jumat	09.00-11.00	09018	
2	AIF101	Pemrograman Berorientasi Objek	6	A	Husnul Hakim, S.Kom., M.T.	1	Senin	08.00-09.00	09122	
			6	A	Vania Natali, S.Kom.	2	Selasa	09.00-11.00	09018	
			6	A	Husnul Hakim, S.Kom., M.T.	3	Rabu	08.00-09.00	09018	
			6	A	Vania Natali, S.Kom.	4	Jumat	09.00-11.00	09018	
3	AIF101	Pemrograman Berorientasi Objek	6	A	Husnul Hakim, S.Kom., M.T.	1	Senin	08.00-09.00	09122	
			6	A	Vania Natali, S.Kom.	2	Selasa	09.00-11.00	09018	
			6	A	Husnul Hakim, S.Kom., M.T.	3	Rabu	08.00-09.00	09018	
			6	A	Vania Natali, S.Kom.	4	Jumat	09.00-11.00	09018	
4	AIF101	Pemrograman Berorientasi Objek	6	A	Husnul Hakim, S.Kom., M.T.	1	Senin	08.00-09.00	09122	
			6	A	Vania Natali, S.Kom.	2	Selasa	09.00-11.00	09018	
			6	A	Husnul Hakim, S.Kom., M.T.	3	Rabu	08.00-09.00	09018	
			6	A	Vania Natali, S.Kom.	4	Jumat	09.00-11.00	09018	
5	AIF101	Pemrograman Berorientasi Objek	6	B	Vania Natali, S.Kom.	1	Senin	08.00-09.00	09121	
			6	B	Ferdian Reynaldi, S.T.	2	Selasa	09.00-11.00	09017	
			6	B	Vania Natali, S.Kom.	3	Rabu	08.00-09.00	09017	

Gambar 3.13: Tampilan Jadwal Seluruh Fakultas

1 • **Nilai dan Indeks Prestasi**

2 Menu Nilai dan Indeks Prestasi terdiri dari submenu:

- 3 – Riwayat per Semester

4 Submenu ini menampilkan informasi nilai per semester. Mahasiswa dapat
5 melihat nilai sesuai dengan semester yang dipilih atau bisa memilih pilihan
6 “Seluruh Tahun Akademik” untuk melihat seluruh nilai berdasarkan semester
7 (Gambar 3.14).

Daftar Nilai Semester								
GANJIL - 2015 / 2016								
No.	Kode MK	Nama Matakuliah	skls	Kelas	ART	UTS	UAS	AA NA
1	AIF401	Skripsi 1 #	4	A	0	0	0	0
2	AIF402	Skripsi 2 #	6	A	0	0	0	0
3	AIF439	Kerja Praktek 2 #	3	A	0	0	0	0
4	AIF441	Administrasi Jaringan Komputer 3 #	3	A	0	0	0	0
5	AIF457	Kewirausahaan Berbasis Teknologi #	3	A	0	0	0	0
6	AIF469	Layanan Berbasis Web #	3	A	0	0	0	0
Total skls				22				

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

Gambar 3.14: Tampilan Riwayat Per Semester

- 8 – Daftar Perkembangan Studi

9 Seluruh riwayat mata kuliah dan nilai yang pernah ditempuh ditampilkan di
10 submenu ini (Gambar 3.15). Pada bagian bawah halaman, terdapat statistik
11 nilai dan indeks prestasi (Gambar 3.16).

Daftar Perkembangan Studi								
Kode MK	Nama MK	Nilai	Tahun Sem	Kode MK	Nama MK	Nilai	Tahun Sem	
SEMESTER 1								
AIF103-3	Matematika Diskrit	B	121	AIF104-4	Algoritma & Struktur Data	A	122	
AIF105-3	Pengantar Informatika	B	121	AIF104-3	Logika Informatika	A	122	
AIF191-3	Pemrograman Berorientasi Objek	B	121	AIF106-3	Sistem Dijital	A	122	
MKU001-2	Pendidikan Pancasila	B	121	AMS190-3	Matematika Informatika	B	122	
MKU008-2	Ethika	B	121	MKU009-2	Bahasa Indonesia	A	141	
MKU010-2	Bahasa Inggris	A	122	MKU011-2	Estatika	B	122	
SEMESTER 3								
AIF201-4	Analisis & Desain Berorientasi Objek	A	131	AIF202-4	Desain & Analisis Algoritma	B	132	
AIF203-4	Struktur Diskrit	A	131	AIF204-4	Manajemen Informasi & Basis Data	B	132	
AIF205-3	Arsitektur & Organisasi Komputer	A	131	AIF206-4	Sistem Operasi	B	132	
AMS200-3	Probabilitas & Statistika	A	131	AIF208-4	Rekayasa Perangkat Lunak	B	132	
MKU003-2	Pendidikan Agama (Katolik)	B	131	AIF210-2	Interaksi Manusia Komputer	B	132	
MKU012-2	Logika	A	122					
SEMESTER 5								
AIF301-3	Pengantar Sistem Cerdas	C	141	AIF302-2	Penulisan Ilmiah	B	142	
AIF303-3	Pengantar Sistem Informasi	A	141	AIF305-5	Proyek Informatika	A	142	
AIF305-4	Jaringan Komputer	A	141					
MKU002-2	Pendidikan Kewarganegaraan	B	122					
SEMESTER 7								
AIF403-2	Komputer & Masyarakat	B	141	APS402-2	Etika Profesi	A	142	
Mata Kuliah Pilihan								
AMS191-4	Kalkulus	B	121					
AIF311-2	Pemrograman Fungsional	A	141					
AIF313-2	Grafika Komputer	B	131					
AIF315-2	Pemrograman Berbasis Web	A	141					
AIF341-3	Administrasi Jaringan Komputer 1	A	141					
AIF312-2	Keamanan Informatika	D	142					
AIF318-2	Pemrograman Aplikasi Bergerak	A	142					
AIF342-3	Administrasi Jaringan Komputer 2	A	142					
AIF390-3	Pemrograman Berbasis Web Lanjut	B	142					
AIF445-3	Metode Numerik	B	132					
AIF455-3	Sistem Pendukung Keputusan	A	141					
AIF461-2	Pencarian & Temu Kembali Informasi	A	142					
AIF480-3	Pengolahan Citra	A	131					
APS302-2	Dunia Digital Dan Sains	A	142					

Gambar 3.15: Tampilan Daftar Perkembangan Studi

Kode Semester :	Ket. Mk. Bid. Peminatan :																																									
1 = Ganjil, 2 = Genap, 4 = Padat, 6 = Transfer	01 = Teknologi Informasi Bisnis 02 = Ilmu Komputer 03 = Telematika																																									
Ket. Mt Kuliah :																																										
• [M] = Mk. Kendali Mutu,																																										
• [X] = Mk. Disisihkan,																																										
• [<] = Mk. yang diambil semester (2015-1)																																										
<table border="1"> <thead> <tr> <th>Nilai Akhir</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>K</th></tr> </thead> <tbody> <tr> <td>Jml. Mata Kuliah</td><td>23</td><td>20</td><td>1</td><td>1</td><td></td><td></td></tr> <tr> <td>Jml. sks</td><td>67</td><td>56</td><td>3</td><td>2</td><td></td><td></td></tr> <tr> <td>Jml. Mk K. Mutu</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>Jml. sks K. Mutu</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	Nilai Akhir	A	B	C	D	E	K	Jml. Mata Kuliah	23	20	1	1			Jml. sks	67	56	3	2			Jml. Mk K. Mutu							Jml. sks K. Mutu							<table border="1"> <thead> <tr> <th>No</th><th>Skor</th><th>Tanggal</th></tr> </thead> <tbody> <tr> <td>1</td><td>544</td><td>30 Maret 2015</td></tr> </tbody> </table>	No	Skor	Tanggal	1	544	30 Maret 2015
Nilai Akhir	A	B	C	D	E	K																																				
Jml. Mata Kuliah	23	20	1	1																																						
Jml. sks	67	56	3	2																																						
Jml. Mk K. Mutu																																										
Jml. sks K. Mutu																																										
No	Skor	Tanggal																																								
1	544	30 Maret 2015																																								
	IP Lulus (2014-2) : 128 sks : 3.47 IP N. Terbaik (2014-2) : 128 sks : 3.47 IPS (2014-2) : 24 sks : 3.54 IPT-1 (2013-2) : 30 : 4.00 sks lulus : 80																																									
	Jumlah sks Ditempuh : 128 sks Lulus Wajib : 92 sks Lulus Pilihan : 36 sks Lulus Wajib Peminatan : 0 sks Lulus Pilihan Peminatan : 0 sks Total Lulus : 128 sks																																									
	Ditempuh Semester ini (2014-2) : 24 sks Dijinkan untuk semester yad.(2015-2) : 24 sks																																									
	Cuti studi : 0 semester Akhir Masa Studi : Semester GENAP - 2018 / 2019																																									
	Syarat Kelulusan : Lulus min. 144 sks terdiri dari : Mk. Wajib + Mk. Pilihan I.P. Lulus minimum : 2.00																																									

Gambar 3.16: Tampilan Statistik Nilai dan IP

– Riwayat Indeks Prestasi

Menampilkan daftar riwayat Indeks Prestasi Semester (IPS) dan Indeks Prestasi Kumulatif (IPK) setiap semester. Tampilan ini juga dilengkapi dengan grafik perkembangan (Gambar 3.17).



Gambar 3.17: Tampilan Riwayat Indeks Prestasi

– TOEFL

Menampilkan daftar riwayat skor *Test of English as Foreign Language* (TOEFL) yang pernah ditempuh (Gambar 3.18). Mahasiswa diwajibkan untuk menempuh TOEFL dengan skor minimal 500.

The screenshot shows the TOEFL section of the academic portal. At the top, there is a table with one row containing 'No', 'Skor' (Score), and 'Tanggal' (Date). The entry is '1' with a score of '544' and the date '30 Maret 2015'. Below this is a green header box titled 'Apa sih TOEFL ?'. The main content area contains text explaining what TOEFL is, its purpose, and its structure. It also includes a bulleted list of skills tested and a note about the score range.

Gambar 3.18: Tampilan TOEFL

1 • Pembayaran Uang Kuliah

2 Menu ini berfungsi untuk melihat data tagihan pembayaran uang kuliah serta
3 cara-cara pembayarannya (Gambar 3.19).

The screenshot shows the payment section of the academic portal. At the top, there is a search bar with 'NPM: 2012730012' and a button 'Lihat Data Tagihan'. Below this is a table with columns 'No', 'Tagihan' (Bill), and 'Deskripsi' (Description). There is one entry with 'Rp' and '0,-'. Below the table is a blue header box titled 'Keterangan' (Information). The main content area contains text about payment methods, specifically detailing steps for Bank Permata, ATM, and other banking methods. It also includes a note about transfer limits and a link for mobile access.

Gambar 3.19: Tampilan Pembayaran Uang Kuliah

4. Informasi

5 Bagian ini menampilkan informasi tentang periode-periode yang sedang aktif (Gambar
6 3.20). Sebagai contoh jika "Periode Registrasi" diklik maka akan muncul *pop up* seperti
7 pada gambar 3.21.



Gambar 3.20: Tampilan Informasi

Tahun Akademik : GANJIL - 2015 / 2016	
• Tgl Semester	: 18 Agustus 2015 s/d 23 Januari 2016
• Tgl UTS	: 5 Oktober 2015 s/d 20 Oktober 2015
• Tgl UAS	: 7 Desember 2015 s/d 19 Desember 2015
• Tgl Batas Nilai UTS	: 5 November 2015
• Tgl Batas Nilai UAS	: 6 Januari 2016
• Tgl Batas Nilai Akhir	: 8 Januari 2016
• Tgl FRS Prodi	: 6 Juli 2015 s/d 10 Juli 2015
• Tgl FRS Dosen Wali	: 6 Juli 2015 s/d 10 Juli 2015
• Tgl FRS Prodi	: 1 September 2015 s/d 2 September 2015
• Tgl FRS Dosen Wali	: 1 September 2015 s/d 2 September 2015
• Tgl Frs Mk. Dibuka	: 15 Juni 2015 s/d 3 Juli 2015
• Tgl Prs Mk. Dibuka	: 20 Agustus 2015 s/d 28 Agustus 2015
• Tgl Frs Mahasiswa	: 6 Juli 2015 08.00 s/d 10 Juli 2015 16.00
• Tgl Frs Mahasiswa	: 1 September 2015 08.00 s/d 2 September 2015 16.00
• Tgl Frs Tata Usaha	: 1 Juli 2015 s/d 25 Agustus 2015
• Tgl Frs Tata Usaha	: 1 September 2015 s/d 11 September 2015
• Tgl Tenggang PRS	: -/- -
• Tgl Aju Sidang	: 18 Agustus 2015 s/d 22 Januari 2016

Gambar 3.21: Tampilan Pop Up Informasi

1 5. Kalender

2 Bagian ini menampilkan kalender masehi (Gambar 3.22).



Gambar 3.22: Tampilan Kalender

3 6. Info Browser

4 Bagian ini menampilkan informasi tentang internet browser yang digunakan pada saat
5 membuka Portal Akademik Mahasiswa (Gambar 3.23).



Gambar 3.23: Tampilan Info Browser

3.2 Analisis Kebutuhan Informatika Student Portal

Dalam menganalisis kebutuhan Informatika Student Portal, peneliti melakukan wawancara dengan mahasiswa Program Studi Teknik Informatika UNPAR. Sampel yang diambil adalah lima orang mahasiswa dari setiap angkatan dalam empat tahun terakhir yaitu angkatan 2012 sampai 2015 dengan rincian sebagai berikut:

- Angkatan 2012: tiga orang laki-laki dan dua orang perempuan
- Angkatan 2013: dua orang laki-laki dan tiga orang perempuan
- Angkatan 2014: lima orang laki-laki
- Angkatan 2015: tiga orang laki-laki dan satu orang perempuan

Setelah melakukan wawancara, peneliti memperoleh fitur-fitur yang diinginkan mahasiswa antara lain:

1. Prasyarat mata kuliah

Mahasiswa bisa memeriksa prasyarat mata kuliah saat FRS sehingga tidak terjadi kesalahan pengambilan mata kuliah. Prasyarat mata kuliah yang ditampilkan di Portal Akademik Mahasiswa kurang akurat. Selain itu, dari 20 mahasiswa yang diwawancara, hanya ada satu mahasiswa yang mengetahui bahwa Portal Akademik Mahasiswa memiliki fitur prasyarat. Prasyarat mata kuliah untuk Program Studi Teknik Informatika juga tersedia di <http://tinyurl.com/lionov>², namun mahasiswa merasa kurang praktis karena harus memeriksa secara manual. Mahasiswa menginginkan agar fitur ini bisa dibuat untuk mempermudah FRS. Fitur ini akan menampilkan seluruh mata kuliah yang dibuka pada semester terkini beserta status pengambilannya.

2. Ringkasan data akademik

Ringkasan data akademik menampilkan data mengenai mata kuliah pilihan wajib dan sisa SKS untuk mencapai kelulusan. Mahasiswa menginginkan fitur ini dibuat untuk membantu mereka dalam mengatur perkuliahananya.

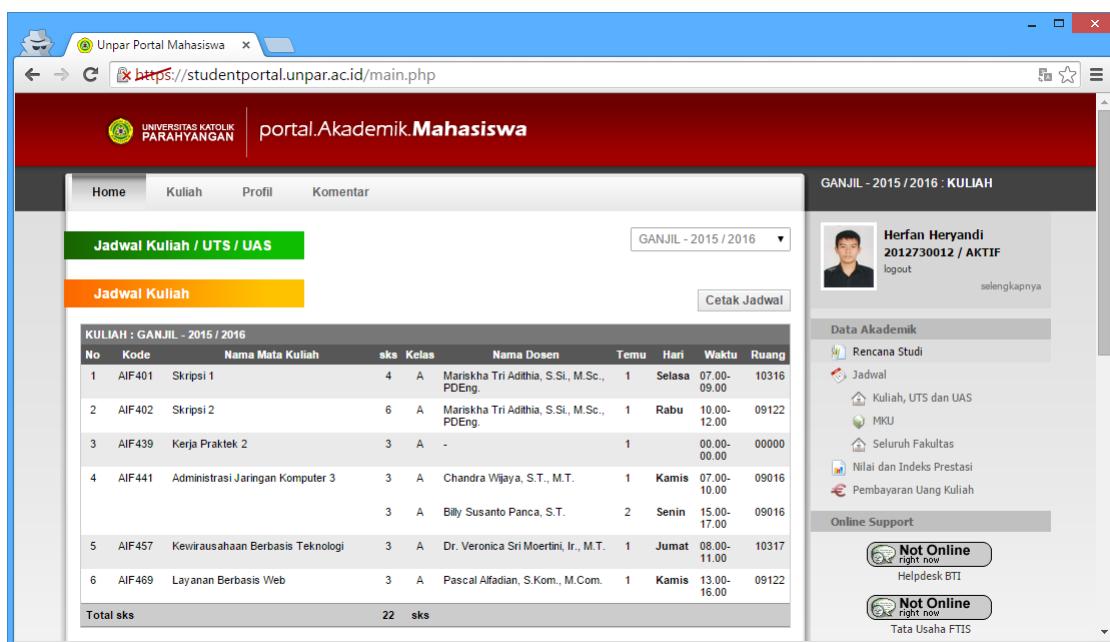
3. Perubahan IPS dan IPK berdasarkan riwayat nilai

Dalam Portal Akademik Mahasiswa, nilai pertama kali muncul dalam riwayat nilai. Riwayat IP tidak berubah secara otomatis saat seluruh nilai di riwayat nilai sudah muncul. Mahasiswa menginginkan agar IPS dan IPK dapat berubah secara otomatis saat nilai muncul.

4. Jadwal kuliah yang tersusun

Tampilan jadwal kuliah dalam Portal Akademik Mahasiswa tidak terurut berdasarkan hari seperti pada gambar 3.24 sehingga perlu direkapitulasi lagi. Mahasiswa menginginkan agar tampilan jadwal tersusun dan dalam bentuk seperti gambar 3.25.

²<http://tinyurl.com/lionov>, diakses 5 September 2015



Gambar 3.24: Tampilan Jadwal pada Portal Akademik Mahasiswa

	SENIN	SELASA	RABU	KAMIS	JUMAT	SABTU
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						

Gambar 3.25: Tampilan Jadwal yang Diinginkan Mahasiswa

5. Kalender akademik

Kalender akademik merupakan salah satu fitur pada Portal Akademik Mahasiswa namun sekarang fitur tersebut sudah tidak ada lagi. Mahasiswa menginginkan fitur kalender akademik kembali untuk mengetahui tanggal-tanggal penting pada perkuliahan.

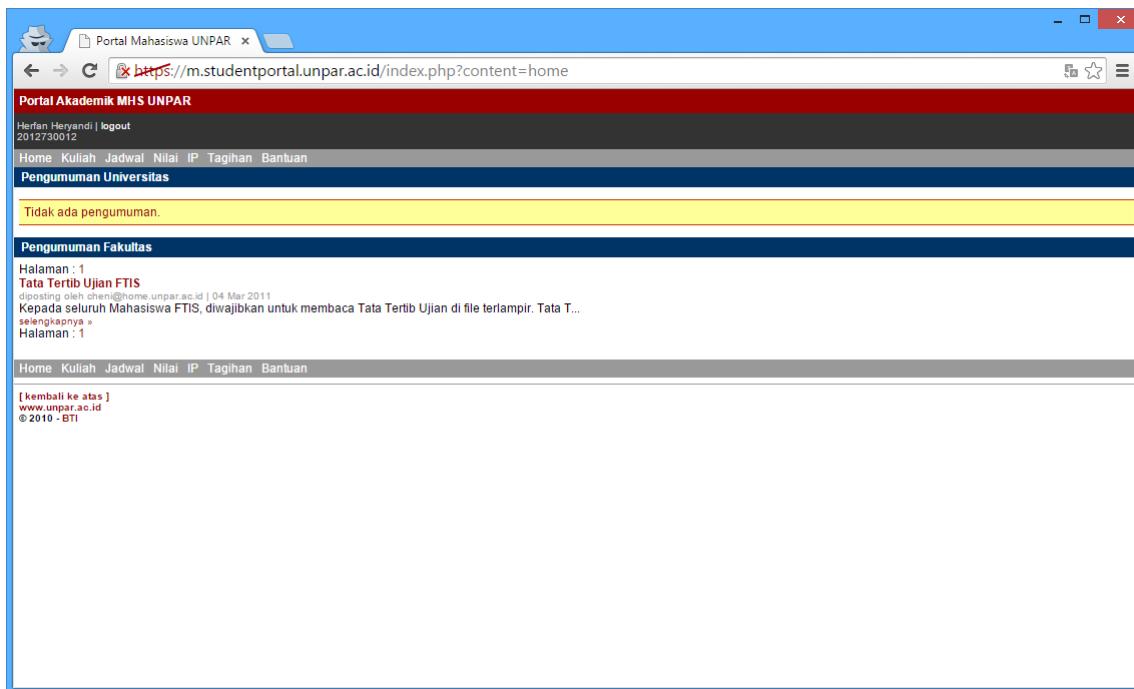
6. Rincian pembayaran

Tagihan pada Portal Akademik Mahasiswa tidak mencantumkan batas akhir pembayaran dan rincian nominal tagihan. Mahasiswa menginginkan rincian pembayaran agar tidak terlambat membayar uang kuliah dan dapat mengetahui rincian nominal tagihan.

7. Rincian mata kuliah

Setiap mata kuliah yang dibuka memiliki rincian seperti deskripsi mata kuliah dan jenis mata kuliah yaitu apakah mata kuliah tersebut wajib, pilihan, atau pilihan wajib. Mahasiswa menginginkan fitur ini agar dapat mengetahui mata kuliah apa yang akan dipelajari.

8. Tampilan situs web sama di sistem operasi manapun
 Jika tidak menggunakan sistem operasi Windows seperti Linux dan Mac, saat mengakses Portal Akademik Mahasiswa melalui <https://studentportal.unpar.ac.id/>, maka mahasiswa akan diarahkan ke <https://m.studentportal.unpar.ac.id/> yaitu Portal Akademik Mahasiswa dengan tampilan *mobile* (Gambar 3.26). Tampilan ini tidak memiliki fitur selengkap Portal Akademik Mahasiswa, hanya memiliki fitur pengumuman kuliah, jadwal kuliah, UTS, dan UAS, nilai, IP, dan tagihan. Selain itu, tampilan *mobile* pada telepon seluler akan terlihat sangat kecil sehingga tidak sulit untuk memilih menu. Mahasiswa menginginkan fitur ini agar Informatika Student Portal dapat diakses di sistem operasi manapun tanpa perubahan tampilan.



Gambar 3.26: Tampilan *Mobile* Portal Akademik Mahasiswa

9. Kontak dosen
 Kontak dosen berisi informasi email setiap dosen sehingga dapat mempermudah mahasiswa untuk menghubungi dosen. Mahasiswa juga dapat mengirim email secara langsung melalui Portal Akademik Mahasiswa.
10. Pohon kurikulum
 Mahasiswa mungkin ingin agar dapat melihat pohon kurikulum Program Studi Teknik Informatika dalam Portal Akademik Mahasiswa.
11. Pemberitahuan
 Mahasiswa menginginkan Portal Akademik Mahasiswa menampilkan pemberitahuan berupa *pop up* mengenai pengumuman terkini.
12. Unggah *Curriculum Vitae*
 Mahasiswa menginginkan agar dapat mengunggah data mengenai kegiatan dan keaktifan di universitas agar dapat digunakan oleh perusahaan untuk mencari mahasiswa dengan kriteria tertentu misalnya untuk kepentingan magang dan beasiswa.

1 Peneliti menetapkan fitur-fitur yang akan dipilih untuk diimplementasikan harus memenuhi kriteria:

- 3 • Data yang dibutuhkan dapat diambil dari Portal Akademik Mahasiswa
- 4 Aplikasi Informatika Student Portal merupakan aplikasi hasil kustomisasi Portal Aka-
- 5 demik Mahasiswa. Informasi yang ditampilkan Informatika Student Portal diperoleh
- 6 dengan cara mengolah data dari Portal Akademik Mahasiswa. Jadi data yang dibu-
- 7 tuhkan Informatika Student Portal harus dapat diambil dari Portal Akademik Maha-
- 8 siswa.
- 9 • Data hasil olahan tidak tersedia di Portal Akademik Mahasiswa
- 10 Jika data hasil olahan yang sudah tersedia di Portal Akademik Mahasiswa, maka
- 11 fitur tidak perlu dibuat lagi di Informatika Student Portal karena mahasiswa dapat
- 12 mengakses langsung Portal Akademik Mahasiswa.
- 13 • Fitur mendukung fungsi Portal Akademik Mahasiswa sebagai sumber informasi aka-
- 14 demik
- 15 Portal Akademik Mahasiswa berfungsi sebagai sumber informasi akademik karena itu
- 16 Informatika Student Portal juga harus mememiliki fitur yang dapat mendukung fungsi
- 17 tersebut.

18 Hasil analisis fitur-fitur yang diinginkan berdasarkan kriteria di atas dan batas waktu

19 pembangunan aplikasi dapat dilihat pada tabel 3.1.

Tabel 3.1. Tabel Hasil Analisis Kebutuhan Informatika Student Portal

Fitur	Dibuat/Tidak dibuat	Alasan
Prasyarat mata kuliah	Dibuat	Data dapat diambil dari Portal Akademik Mahasiswa dan aturan prasyarat mata kuliah Program Studi Teknik Informatika sudah tersedia di SIA Models
Ringkasan data akademik	Dibuat	Data dapat diambil dari Portal Akademik Mahasiswa dan didukung oleh SIA Models
Perubahan IPS dan IPK berdasarkan riwayat nilai	Dibuat	IPS dan IPK dapat dihitung melalui riwayat nilai yang dapat diperoleh dari Portal Akademik Mahasiswa
Jadwal kuliah yang tersusun	Dibuat	Jadwal yang tersusun mempermudah mahasiswa untuk
Kalender akademik	Tidak dibuat	Data tidak bisa diperoleh dari Portal Akademik Mahasiswa
Rincian pembayaran	Tidak dibuat	Data tidak bisa diperoleh dari Portal Akademik Mahasiswa
Rincian mata kuliah	Tidak dibuat	Data tidak bisa diperoleh dari Portal Akademik Mahasiswa
Tampilan situs web sama di sistem operasi manapun	Dibuat	Aplikasi yang akan dibuat merupakan situs web yang responsif
Kontak dosen	Tidak dibuat	Data tidak bisa diperoleh dari Portal Akademik Mahasiswa
Pemberitahuan	Tidak dibuat	Waktu pengerjaan yang terbatas
Unggah <i>Curriculum Vitae</i>	Tidak dibuat	Tidak mendukung Portal Akademik Mahasiswa sebagai sumber informasi akademik

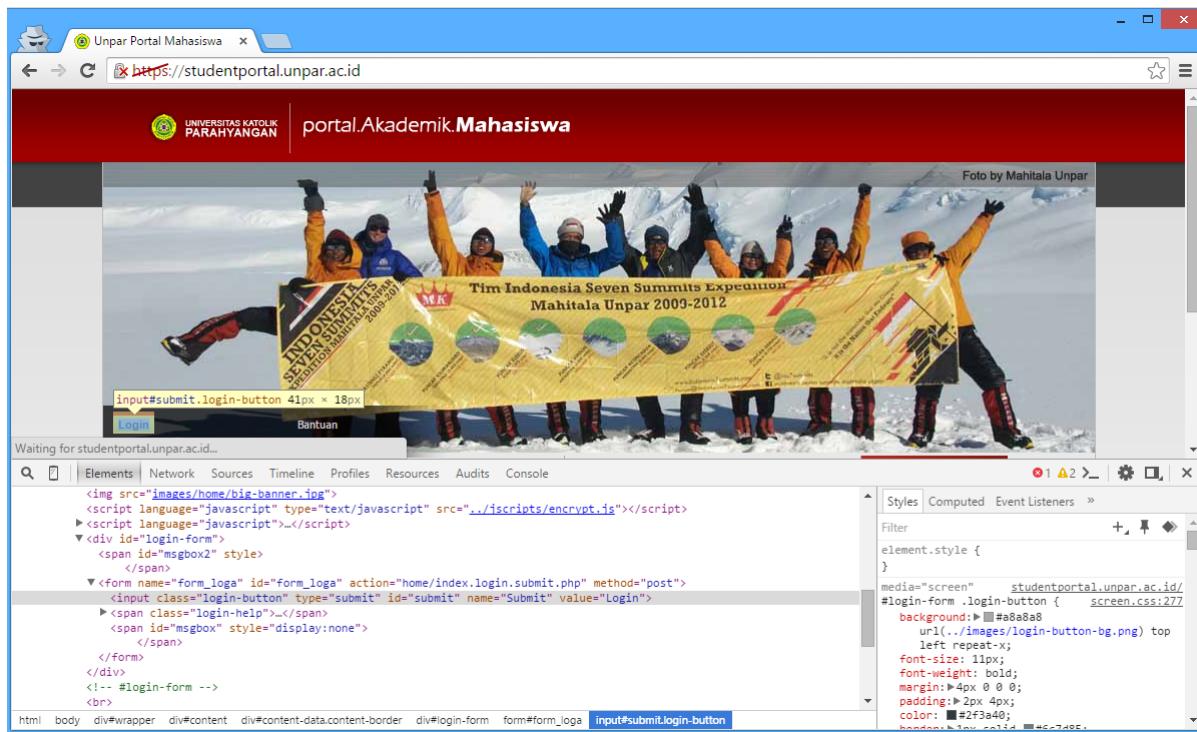
3.3 Analisis Komunikasi Portal Akademik Mahasiswa untuk Fitur Informatika Student Portal

Untuk memenuhi fitur Informatika Student Portal, peneliti menganalisis komunikasi Portal Akademik Mahasiswa ke dalam beberapa kasus yang akan dijelaskan pada subbab-subbab berikut.

3.3.1 Kasus Login

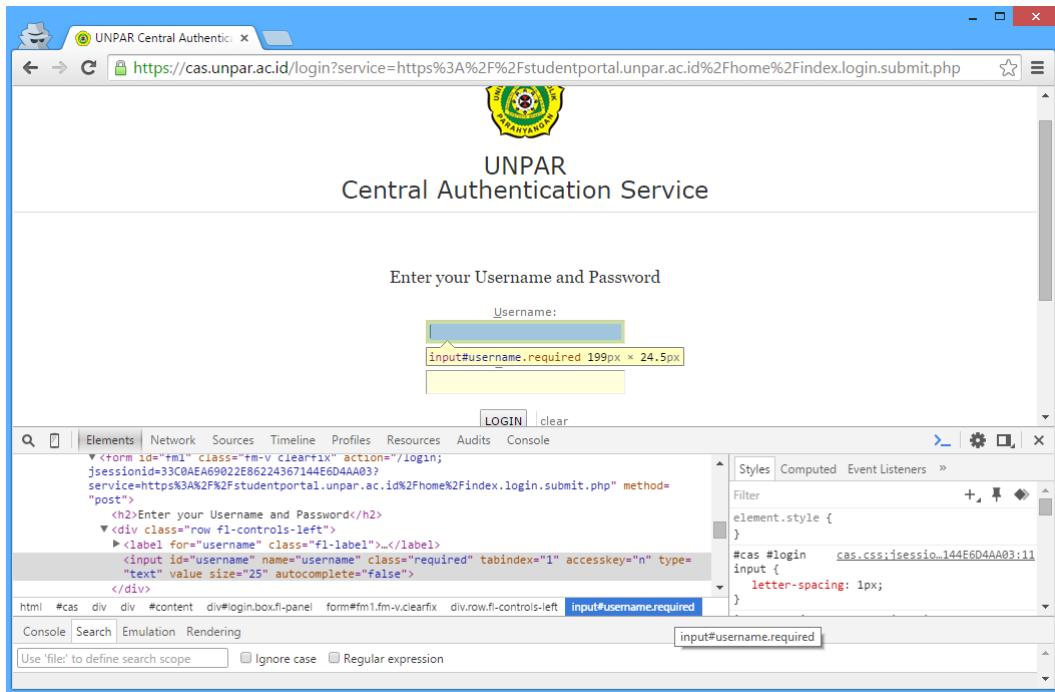
Di Portal Akademik Mahasiswa, mahasiswa dapat *login* dengan cara:

- Mengakses <https://studentportal.unpar.ac.id/> dan mengklik tombol “input#submit.login-button” (Gambar 3.27).



Gambar 3.27: Tombol “input#submit.login-button” pada Halaman Depan Portal Akademik Mahasiswa

- Saat tombol tersebut ditekan, mahasiswa akan dibawa ke halaman <index.login.submit.php> dengan *form data* berisi:
 - Submit: selalu berisi “Login”
- Secara otomatis halaman akan berpindah lagi ke <https://cas.unpar.ac.id/login?service=https%3A%2F%2Fstudentportal.unpar.ac.id%2Fhome%2Findex.login.submit.php>.
- Di sana, akan ditampilkan halaman *login* CAS UNPAR di mana mahasiswa diminta mengisi “Username” pada kolom “input#username.required” dan “Password” pada kolom “input#password.required” (Gambar 3.28).



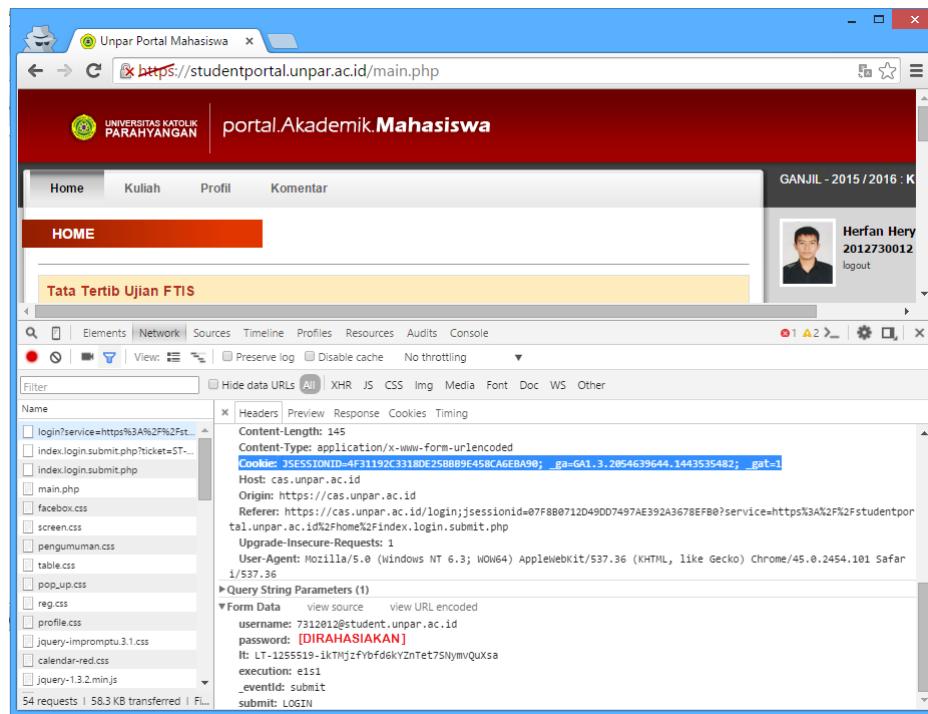
Gambar 3.28: Kolom “Username” “input#username.required” pada Halaman CAS UNPAR

1 5. Setelah itu mahasiswa harus menekan tombol “input.btn-submit”. Data tersebut akan
2 dikirimkan ke `/login;jsessionid=...?service=https://studentportal.unpar.ac.`
3 `id/home/index.login.submit.php` dengan *cookie*:

- 4 • JSESSIONID: diambil dari *cookie* yang di-set pada halaman `https://cas.unpar.`
5 `ac.id/login?service=https%3A%2F%2Fstudentportal.unpar.ac.id%2Fhome%2Findex.`
6 `login.submit.php`

7 Data yang dikirim juga mengandung *form data* sebagai berikut (Gambar 3.29):

- 8 • username: diambil dari nilai elemen “input#username.required”
- 9 • password: diambil dari nilai elemen “input#password.required”
- 10 • lt: diambil dari nilai elemen “input” dengan nama “lt”
- 11 • execution: diambil dari nilai elemen “input” dengan nama “execution”
- 12 • _eventId: selalu berisi “submit”
- 13 • submit: selalu berisi “LOGIN”



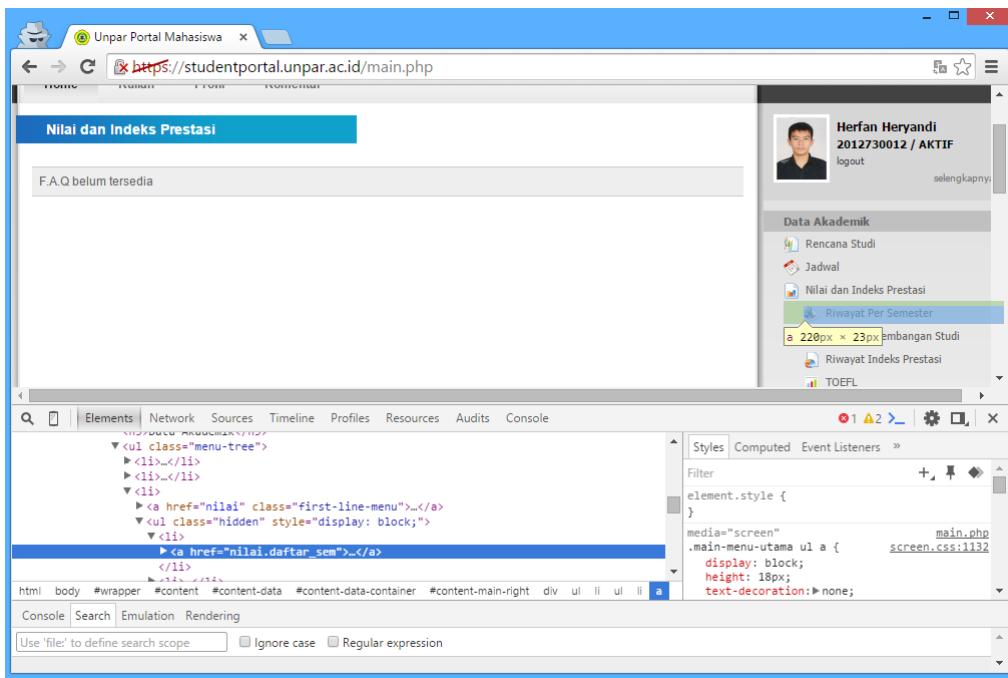
Gambar 3.29: *Form Data* yang dikirim CAS UNPAR

- 1 6. Jika berhasil, akan dilakukan pengalihan beberapa kali dan diakhiri di <https://studentportal.unpar.ac.id/main.php> dengan *cookie* sebagai berikut:
- 2
 - 3 ● PHPSESSID: diambil dari *cookie* yang di-set pada beberapa pengalihan sebelumnya
 - 4 ● _ga dan _gat: tidak diperlukan, digunakan oleh Google Analytics³
- 5 7. Pada halaman utama, mahasiswa dapat melihat foto profil pada “div.student-photo img”.

8 3.3.2 Kasus Nilai

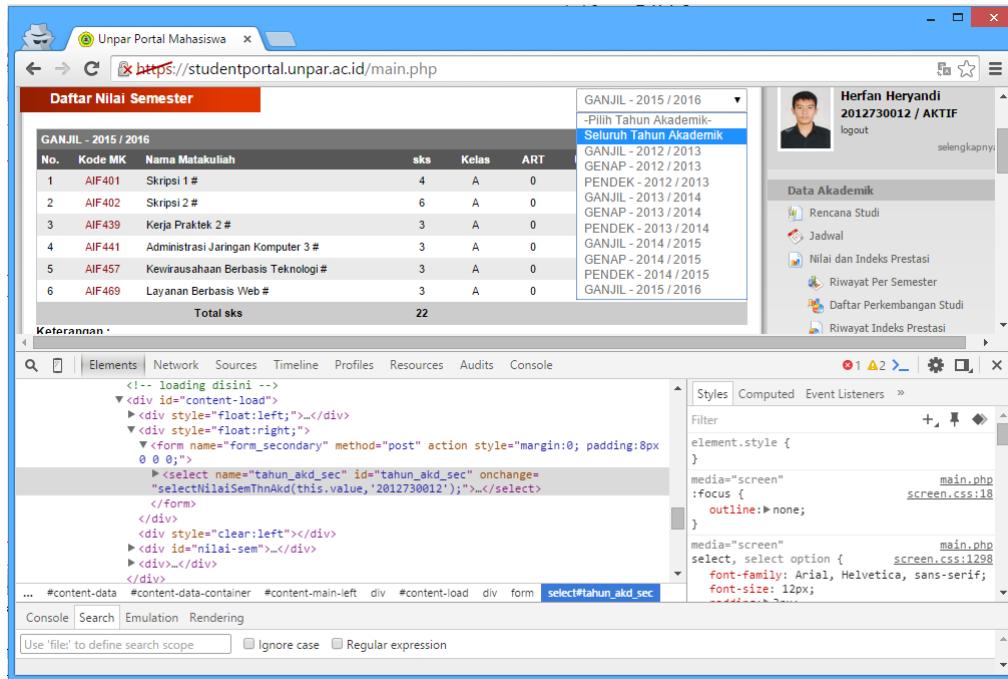
- 9 Di halaman utama Portal Akademik Mahasiswa, mahasiswa dapat melihat riwayat nilai dengan cara:
 - 10 1. Mengklik “a.first-line-menu” dengan teks “Nilai dan Indeks Prestasi”
 - 11 2. Setelah diklik, akan muncul list “ul.hidden”, kemudian mahasiswa harus mengklik elemen “a” dengan teks “Riwayat Per Semester” (Gambar 3.30)

³<https://developers.google.com/analytics/devguides/collection/analyticsjs/cookie-usage>



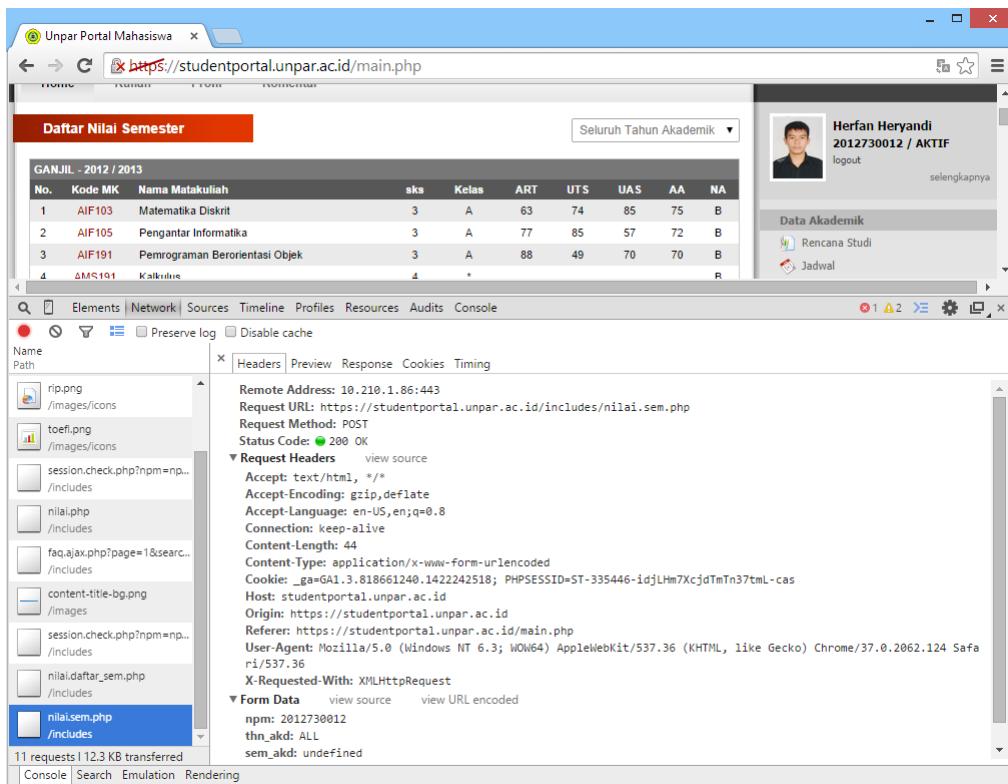
Gambar 3.30: Elemen “a” dengan teks “Riwayat Per Semester” pada Menu Nilai dan Indeks Prestasi

- 1 3. Setelah mengklik “Riwayat Per Semester”, mahasiswa akan diarahkan ke https://studentportal.unpar.ac.id/includes/nilai.daftar_sem.php dengan *cookie* sebagai berikut:
- 2
 - 4 • PHPSESSID: diambil dari *cookie* yang di-set saat pengalihan ke halaman utama
- 5 4. Halaman “Riwayat Per Semester” menampilkan nilai semester terkini. Jika ingin melihat nilai semester sebelumnya, mahasiswa dapat mengklik *combo box* “select#tahun_akd_sec” kemudian memilih “option” yang diinginkan atau “Seluruh Tahun Akademik” untuk melihat nilai seluruh semester (Gambar 3.31).
- 6
- 7
- 8



Gambar 3.31: *Combo Box “select#tahun_akd_sec” pada Halaman Riwayat Per Semester*

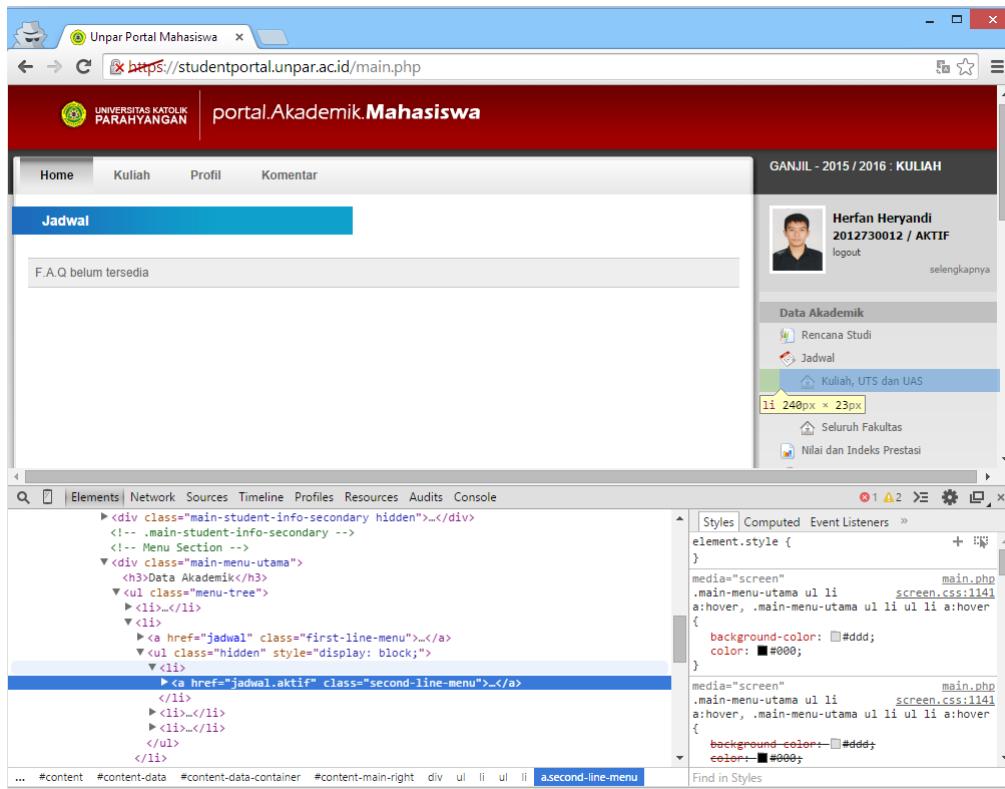
- 1 5. Setelah memilih “option”, mahasiswa akan dibawa ke <https://studentportal.unpar.ac.id/includes/nilai.sem.php> (Gambar 3.32) dengan cookie PHPSESSID dan mengandung *form data*:
- 2 6.
 - 4 • npm: diperoleh dari NPM mahasiswa
 - 5 • thn_akd: berisi tahun akademik semester atau berisi “ALL” jika memilih “Seluruh Tahun Akademik”
 - 6 • sem_akd: berisi semester akademik dalam angka (1: ganjil, 2: genap, 4: pendek) atau tidak didefinisikan jika memilih “Seluruh Tahun Akademik”



Gambar 3.32: *Form Data* pada pengiriman Nilai Seluruh Tahun Akademik

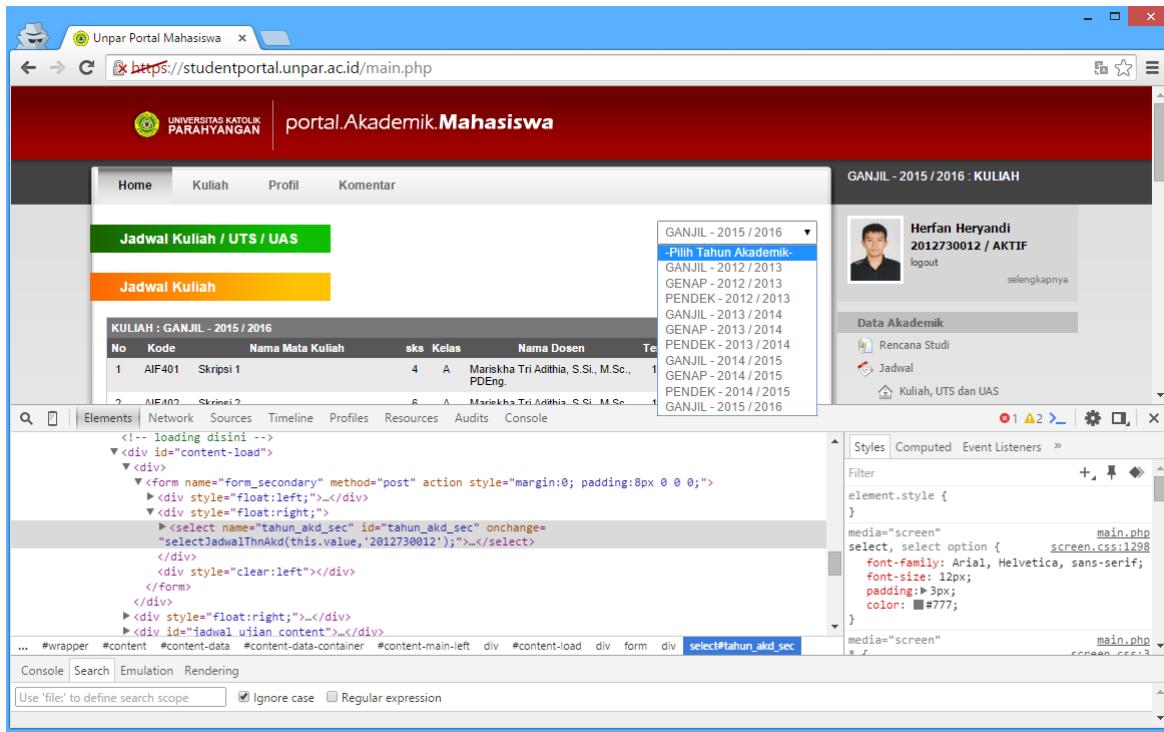
3.3.3 Kasus Jadwal

- 2 Di halaman utama Portal Akademik Mahasiswa, mahasiswa dapat melihat jadwal dengan cara:
1. Mengklik “a.first-line-menu” dengan teks “Jadwal”
2. Setelah diklik, akan muncul list “ul.hidden”, kemudian mahasiswa harus mengklik elemen “a” dengan teks “Kuliah, UTS dan UAS”(Gambar 3.33)



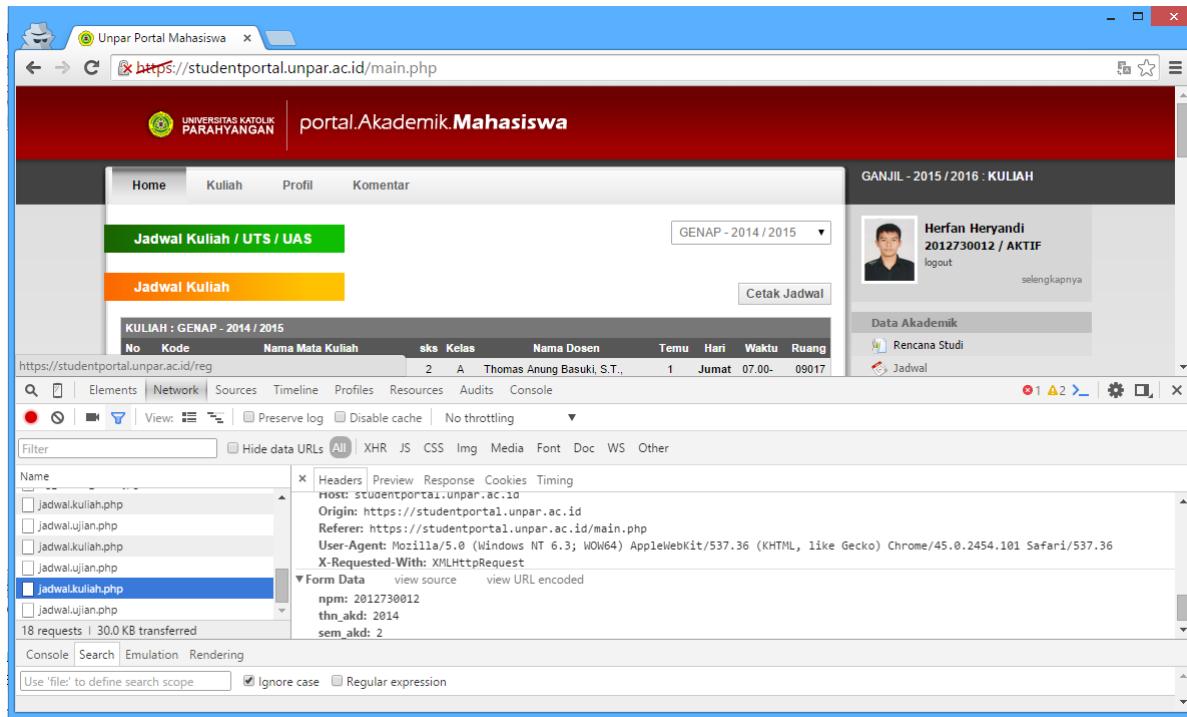
Gambar 3.33: Elemen “a” dengan teks “Kuliah, UTS dan UAS” pada Menu Jadwal

- 1 3. Setelah mengklik “Kuliah, UTS dan UAS”, mahasiswa akan diarahkan ke <https://studentportal.unpar.ac.id/includes/jadwal.aktif.php> dengan *cookie* sebagai berikut:
- 2
 - 4 • PHPSESSID: diambil dari *cookie* yang di-set saat pengalihan ke halaman utama
- 5 4. Halaman “Kuliah, UTS dan UAS” menampilkan jadwal kuliah, UTS, dan UAS semester terkini. Jika ingin melihat jadwal semester sebelumnya, mahasiswa dapat mengklik *combo box* “select#tahun_akd_sec” kemudian memilih “option” yang diinginkan (Gambar 3.34).
- 6
- 7
- 8



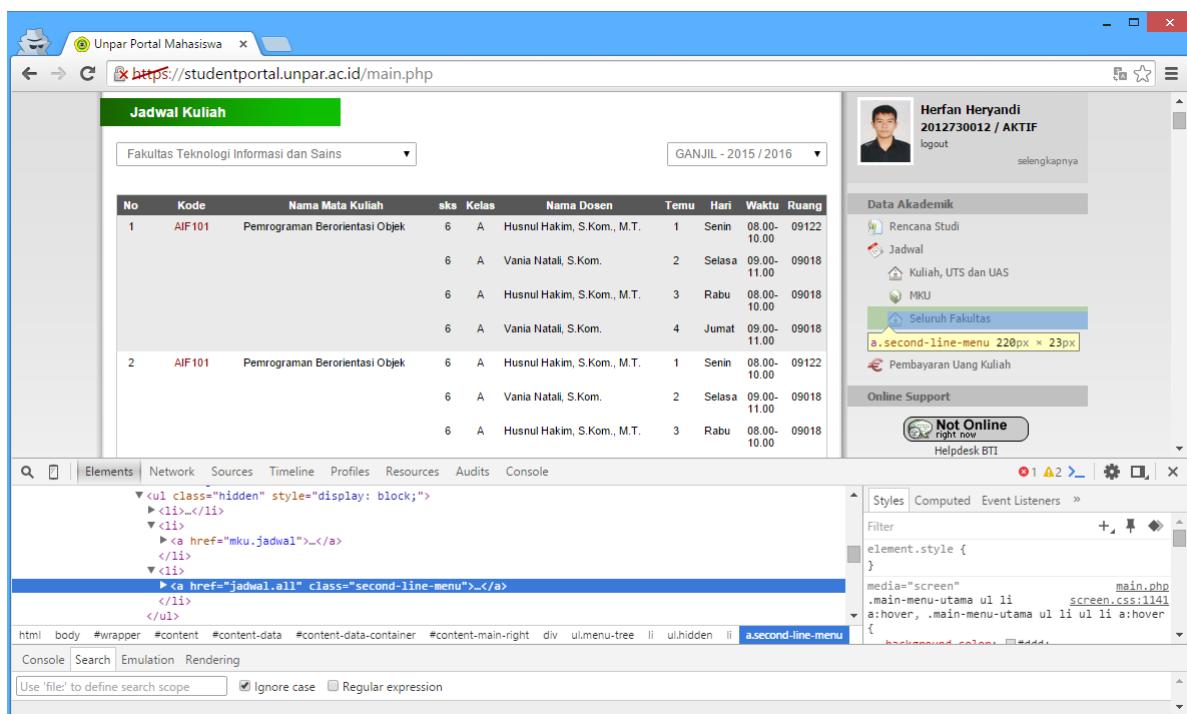
Gambar 3.34: Combo Box “select#tahun_akd_sec” pada Halaman Jadwal Kuliah, UTS, dan UAS

- 1 5. Setelah memilih “option”, mahasiswa akan ditampilkan jawaban dari <https://studentportal.unpar.ac.id/includes/jadwal.kuliah.php> dan <https://studentportal.unpar.ac.id/includes/jadwal.ujian.php> (Gambar 3.35) dengan cookie PHPSESSID dan mengandung form data:
 - 5 • npm: diperoleh dari NPM mahasiswa
 - 6 • thn_akd: berisi tahun akademik semester
 - 7 • sem_akd: berisi semester akademik dalam angka (1: ganjil, 2: genap, 4: pendek)



Gambar 3.35: *Form Data* pada pengiriman Jadwal Kuliah dan Ujian

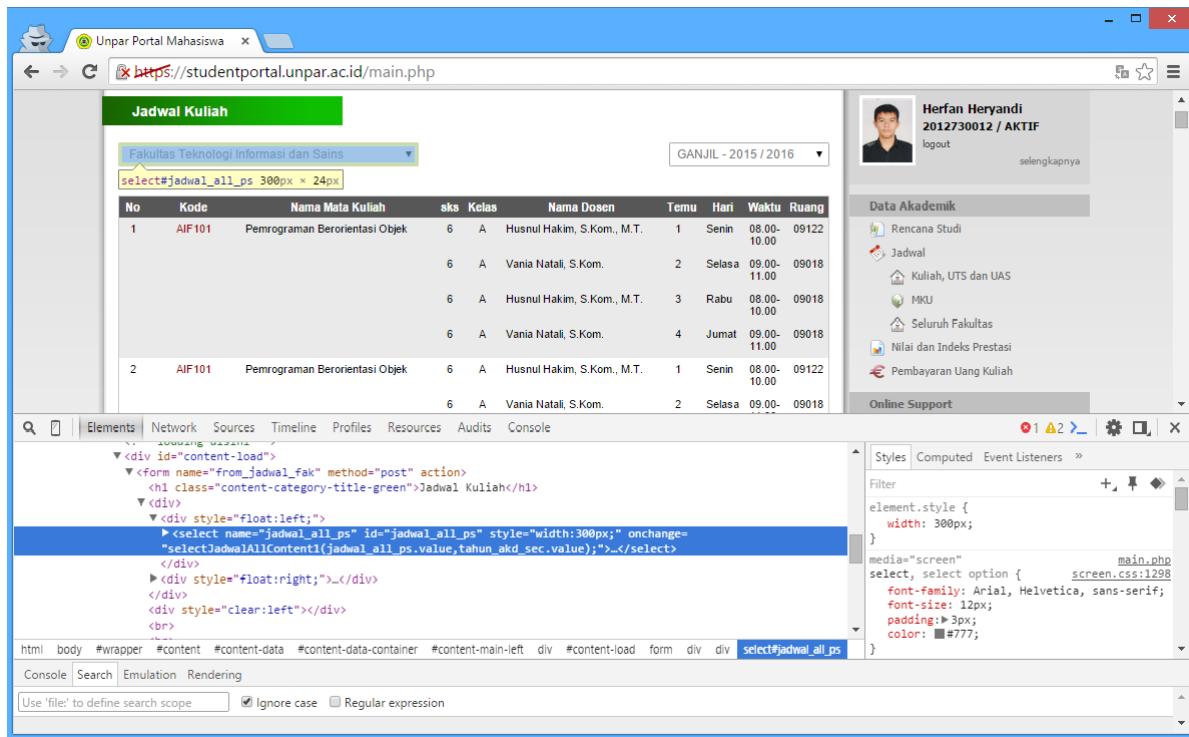
- 1 Mahasiswa juga dapat melihat jadwal seluruh fakultas dengan cara:
- 2 Mengklik “a.second-line-menu” dengan teks “Seluruh Fakultas” pada *list* “ul.hidden” (Gambar 3.36)



Gambar 3.36: Elemen “a” dengan teks “Seluruh Fakultas” pada Menu Jadwal

- 4 Setelah diklik, mahasiswa akan diarahkan ke <https://studentportal.unpar.ac.id/includes/jadwal.all.php> dengan *cookie* sebagai berikut:

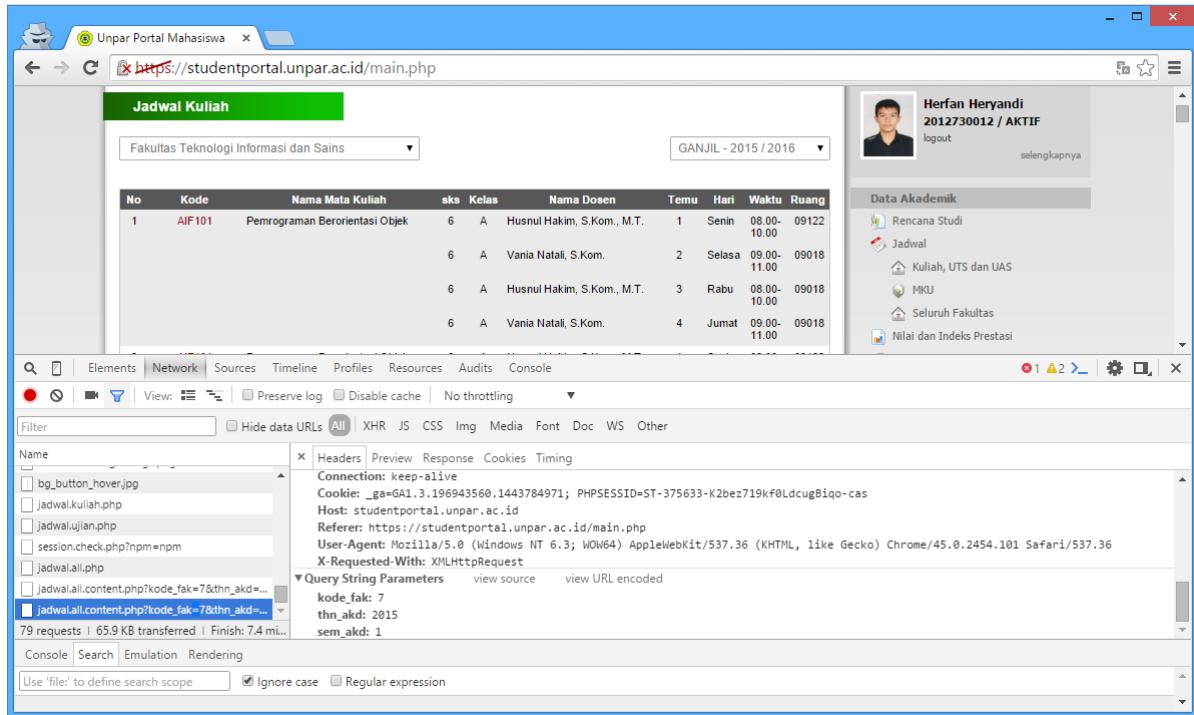
- PHPSESSID: diambil dari *cookie* yang di-set saat pengalihan ke halaman utama
3. Halaman “Seluruh Fakultas” menampilkan seluruh jadwal semester terkini pada fakultas tempat mahasiswa menempuh studi. Jika ingin melihat jadwal semester sebelumnya, mahasiswa dapat mengklik *combo box* “select#tahun_akd_sec” kemudian memilih “option” yang diinginkan. Jika ingin melihat jadwal fakultas lain, mahasiswa dapat mengklik *combo box* “select#jadwal_all_ps” kemudian memilih “option” fakultas yang diinginkan (Gambar 3.37).



Gambar 3.37: Combo Box “select#jadwal_all_ps” pada Halaman Jadwal Seluruh Fakultas

- 8 4. Setelah memilih “option”, mahasiswa akan dibawa ke <https://studentportal.unpar.ac.id/includes/jadwal.all.content.php>(Gambar 3.38) dengan *cookie* PHPSESSID dan mengandung *query string parameter*:

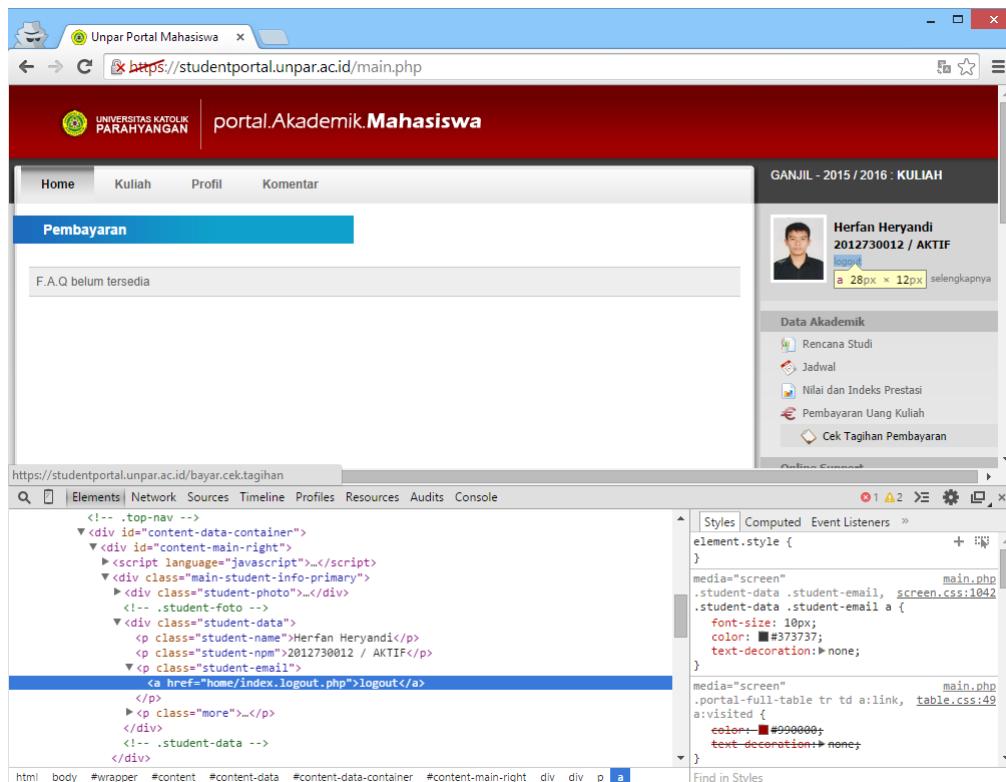
- kode_fak: berisi kode fakultas
- thn_akd: berisi tahun akademik semester
- sem_akd: berisi semester akademik dalam angka (1: ganjil, 2: genap, 4: pendek)



Gambar 3.38: *Form Data* pada pengiriman Jadwal Seluruh Fakultas

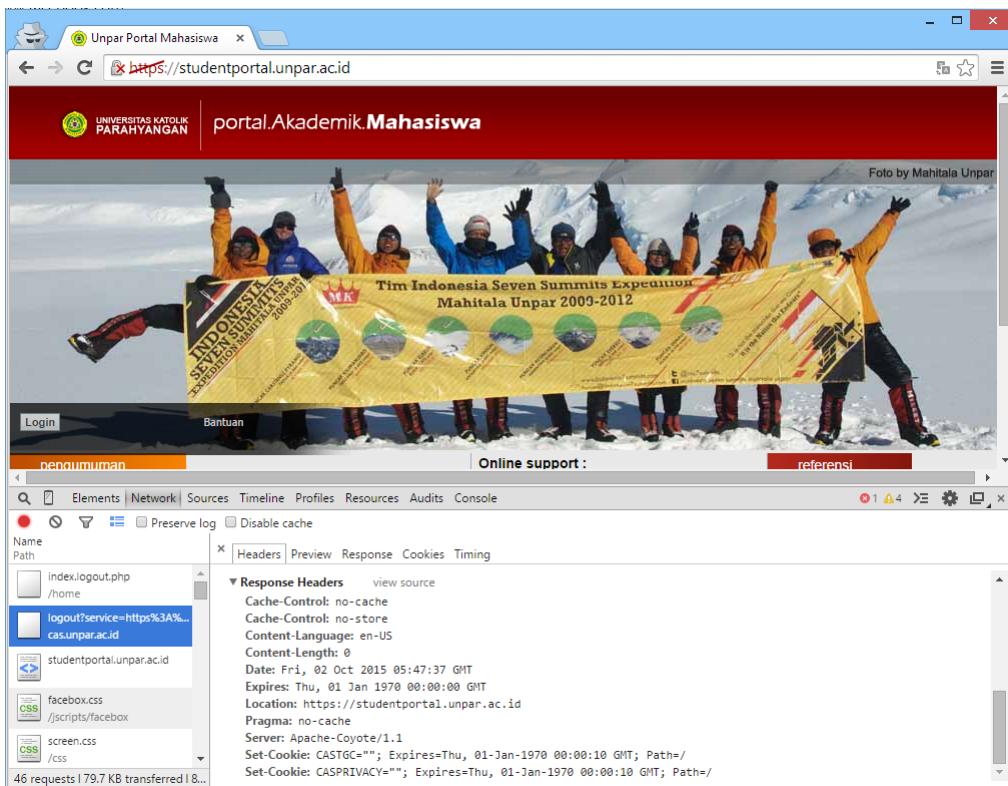
3.3.4 Kasus *Logout*

- 1 Mahasiswa dapat melakukan *logout* dengan cara:
- 2 Mengklik “a” dengan teks “logout” pada bagian identitas portal (Gambar 3.39)



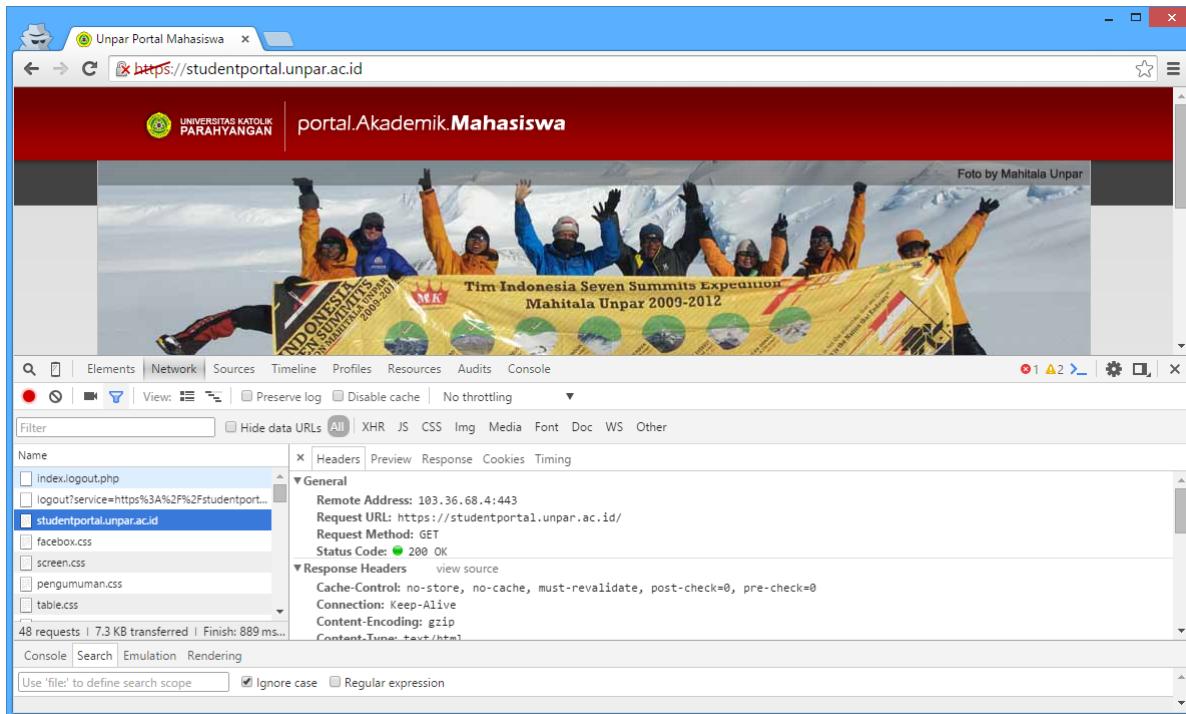
Gambar 3.39: Elemen “a” dengan teks “logout” pada Identitas Portal

- 1 2. Setelah diklik, mahasiswa akan diarahkan ke <https://studentportal.unpar.ac.id/home/index.logout.php> dengan *cookie* sebagai berikut:
- 3 • PHPSESSID: diambil dari *cookie* yang di-set saat pengalihan ke halaman utama
- 4 3. Kemudian dilakukan pengalihan ke <https://cas.unpar.ac.id/logout?service=https%3A%2F%2Fstudentportal.unpar.ac.id> lalu mengadaluarsakan *cookie* CASTGC dan CASPRIVACY yang sebelumnya di-set saat pengalihan ke halaman utama (Gambar 3.40)



Gambar 3.40: Pengadaluarsaan Cookie CASTGC dan CASPRIVACY

- 8 4. Pengalihan diakhiri di <https://studentportal.unpar.ac.id/> yaitu halaman depan Portal Akademik Mahasiswa (Gambar 3.41)



Gambar 3.41: Pengalihan ke Halaman Depan Portal Akademik Mahasiswa

1 Hasil analisis komunikasi Portal Akademik Mahasiswa akan digunakan untuk memba-
2 ngun koneksi yang akan diimplementasikan menggunakan jsoup. Berdasarkan hasil analisis
3 tersebut, terdapat tiga hal yang perlu diperhatikan antara lain:

4 • Perpindahan halaman

5 Saat pengguna melakukan suatu aksi seperti klik dan memilih *option*, Portal Akade-
6 mik Mahasiswa akan mengarahkan pengguna ke suatu halaman. Alamat dari halaman
7 tersebut perlu diketahui untuk koneksi jsoup. Metode pengirimannya juga perlu di-
8 ketahui. Dari seluruh halaman yang dianalisis, metode pengirimannya adalah GET
9 kecuali halaman *login* CAS UNPAR yang menggunakan metode POST.

10 • *Cookie*

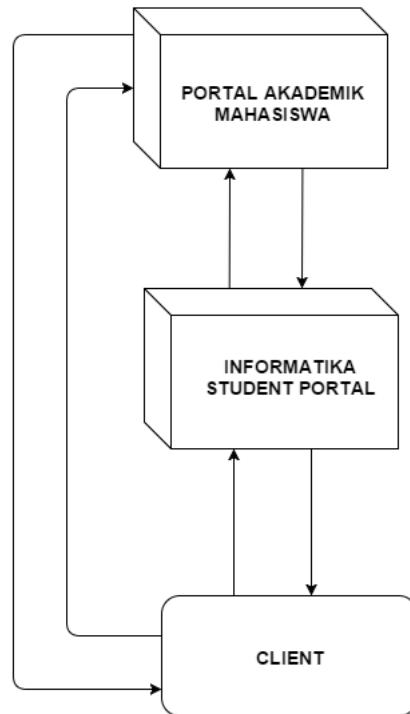
11 Hal yang harus diperhatikan dalam penggunaan *cookie* yaitu kapan suatu *cookie* di-
12 set dan kapan *cookie* tersebut dikirimkan. Jika suatu halaman memerlukan *cookie*
13 saat membangun koneksi, maka perlu diketahui kapan *cookie* tersebut di-set sehingga
14 *cookie* tersebut bisa diperoleh menggunakan jsoup. Pengiriman *cookie* tersebut akan
15 diimplementasikan dengan jsoup saat membangun koneksi ke suatu halaman yang
16 memerlukan *cookie* tersebut.

17 • Parameter

18 Saat berpindah halaman, terkadang halaman tersebut membutuhkan parameter data.
19 Dalam membangun koneksi, jsoup memerlukan juga memerlukan parameter data, ka-
20 rena itu perlu diketahui parameter data apa saja yang harus dilewatkan saat menuju
21 suatu halaman.

3.4 Analisis Arsitektur Informatika Student Portal

- 2 Arsitektur Informatika Student Portal dapat dilihat pada Gambar 3.42. *Client* dapat meng-
3 akses Portal Akademik Mahasiswa dan Informatika Student Portal untuk mendapatkan in-
4 formasi akademik. Namun saat mengakses Informatika Student Portal, *client* akan diberikan
5 data akademik dengan fitur-fitur yang berbeda. Saat *client* mengakses Informatika Student
6 Portal, Informatika Student Portal akan mengambil data dari Portal Akademik Mahasiswa.
7 Data yang telah diambil akan diolah kemudian ditampilkan kepada *client*.

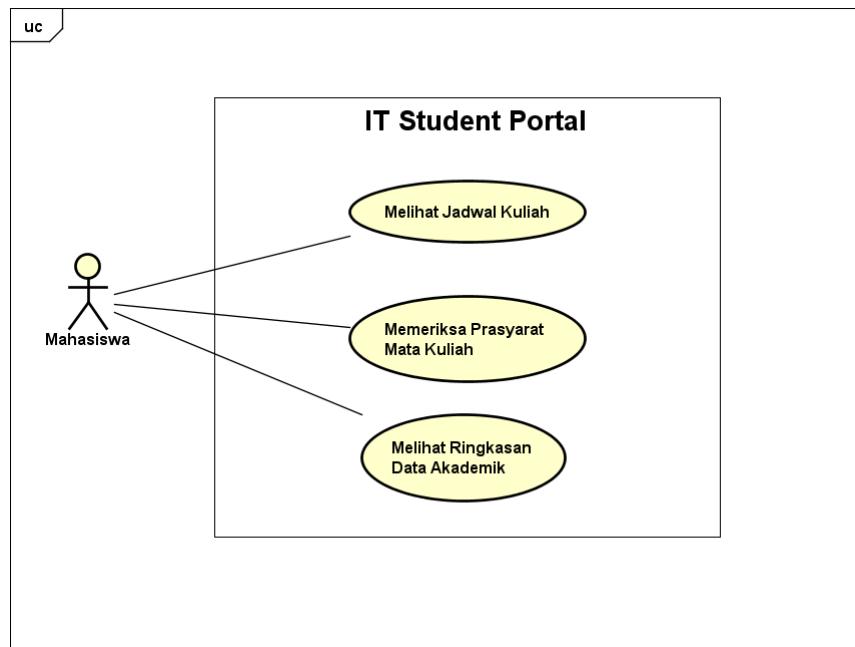


Gambar 3.42: Arsitektur Informatika Student Portal

3.5 Analisis Use Case

3.5.1 Diagram Use Case

- 10 Diagram *use case* pada perangkat lunak yang akan dibangun hanya mengandung satu aktor,
11 yaitu mahasiswa. Diagram *use case* dapat dilihat pada gambar 3.43.



Gambar 3.43: Diagram Use Case Informatika Student Portal

1 Berdasarkan hasil analisis pada subbab 3.2, dari lima fitur yang akan dibuat, dibentuk
2 tiga *use case* antara lain:

- 3 • **Memeriksa Prasyarat Mata Kuliah**, mahasiswa dapat memeriksa mata kuliah
4 yang dibuka pada semester terkini apakah memenuhi prasyarat atau tidak.
- 5 • **Melihat Jadwal Kuliah**, mahasiswa dapat melihat jadwal kuliah yang sudah tersusun
6 dan terurut berdasarkan hari.
- 7 • **Melihat Ringkasan Data Akademik**, mahasiswa dapat melihat data mengenai mata
8 kuliah apa saja yang sudah lulus beserta jenis mata kuliahnya(wajib, pilihan, atau
9 pilihan wajib), sisa SKS untuk mencapai kelulusan, dan mata kuliah wajib yang belum
10 ditempuh. Mahasiswa juga dapat melihat IPS dan IPK yang berubah berdasarkan
11 riwayat nilai.

12 3.5.2 Skenario Use Case

13 1. Memeriksa Prasyarat Mata Kuliah

- 14 • Nama: Memeriksa prasyarat mata kuliah
- 15 • Aktor: Mahasiswa
- 16 • Deskripsi: Memeriksa prasyarat mata kuliah yang dibuka pada semester terkini
- 17 • Kondisi awal: Mahasiswa telah *login*
- 18 • Kondisi akhir: Halaman prasyarat mata kuliah ditampilkan dan berisi mata kuliah
19 yang dibuka pada semester terkini beserta status prasyaratnya
- 20 • Skenario utama:

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

- 2 • Eksepsi: Mahasiswa sedang menempuh semester 1

3 2. Melihat Jadwal Kuliah

- 4 • Nama: Melihat jadwal kuliah
- 5 • Aktor: Mahasiswa
- 6 • Deskripsi: Melihat jadwal kuliah yang sudah tersusun dan terurut berdasarkan hari
- 7 • Kondisi awal: Mahasiswa telah *login*
- 8 • Kondisi akhir: Halaman jadwal ditampilkan dan berisi jadwal kuliah yang sudah tersusun dan terurut berdasarkan hari te
- 9 • Skenario utama:

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

- 14 • Eksepsi: Mahasiswa sedang cuti studi atau jadwal kuliah belum keluar

15 3. Melihat Ringkasan Data Akademik

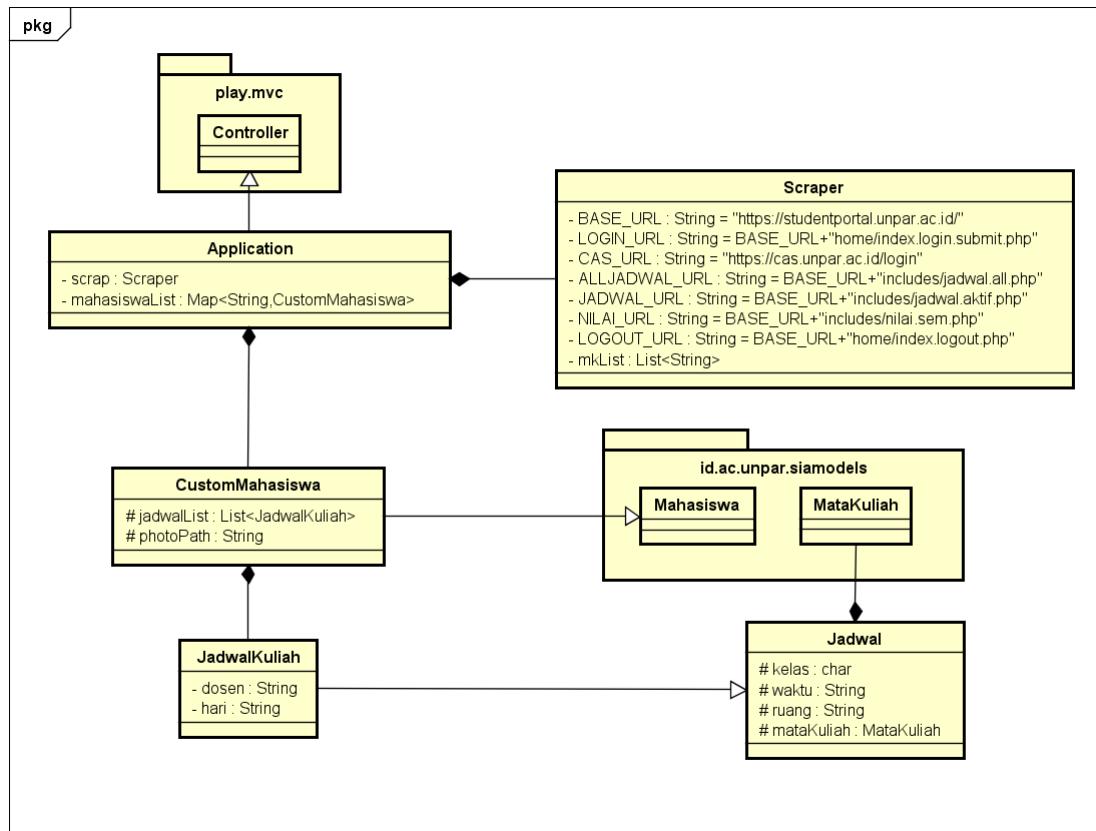
- 16 • Nama: Melihat ringkasan data akademik
- 17 • Aktor: Mahasiswa
- 18 • Deskripsi: melihat data mengenai mata kuliah apa saja yang sudah lulus beserta jenis mata kuliahnya(wajib, pilihan, atau pilihan wajib), sisa SKS untuk mencapai kelulusan, dan mata kuliah wajib yang belum ditempuh. Mahasiswa juga dapat melihat IPS dan IPK yang berubah berdasarkan riwayat nilai
- 19 • Kondisi awal: Mahasiswa telah *login*
- 20 • Kondisi akhir: Halaman jadwal ditampilkan dan berisi jadwal ringkasan data akademik
- 21 • Skenario utama:

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

- 28 • Eksepsi: Mahasiswa sedang menempuh semester 1

3.6 Analisis Kelas

- Diagram kelas analisis untuk Informatika Student Portal ditunjukkan pada gambar 3.44.



Gambar 3.44: Diagram Kelas Analisis Informatika Student Portal

- Kelas-kelas dari *package SIA Models* telah dibahas pada subbab 2.5. Sedangkan penjelasan dari kelas-kelas lainnya sebagai berikut:

1. Controller

Kelas ini merupakan kelas yang sudah disediakan oleh Play Framework. Turunan dari kelas ini akan berperan sebagai *controller*.

2. Application

Kelas ini merupakan *controller* pada Informatika Student Portal yang berfungsi untuk menghubungkan *model* dengan *view*.

3. Scraper

Kelas ini merupakan kelas yang mengimplementasikan *web scraping* menggunakan *library jsoup*. Kelas ini digunakan untuk memperoleh data dari Portal Akademik Mahasiswa kemudian menghubungkannya ke dalam SIA Models. Data-data yang sudah dihubungkan dengan SIA Models akan ditampilkan ke *view* melalui kelas *Application*.

4. CustomMahasiswa

Kelas ini turunan dari kelas Mahasiswa pada SIA Models. Kelas ini merepresentasikan mahasiswa yang memiliki jadwal kuliah dan URL foto profil pada Portal Akademik Mahasiswa.

1 5. Jadwal

2 Kelas ini merepresentasikan jadwal di UNPAR.

3 6. Jadwal Kuliah

4 Kelas ini merupakan turunan dari kelas Jadwal yang merepresentasikan jadwal kuliah
5 di UNPAR.

¹

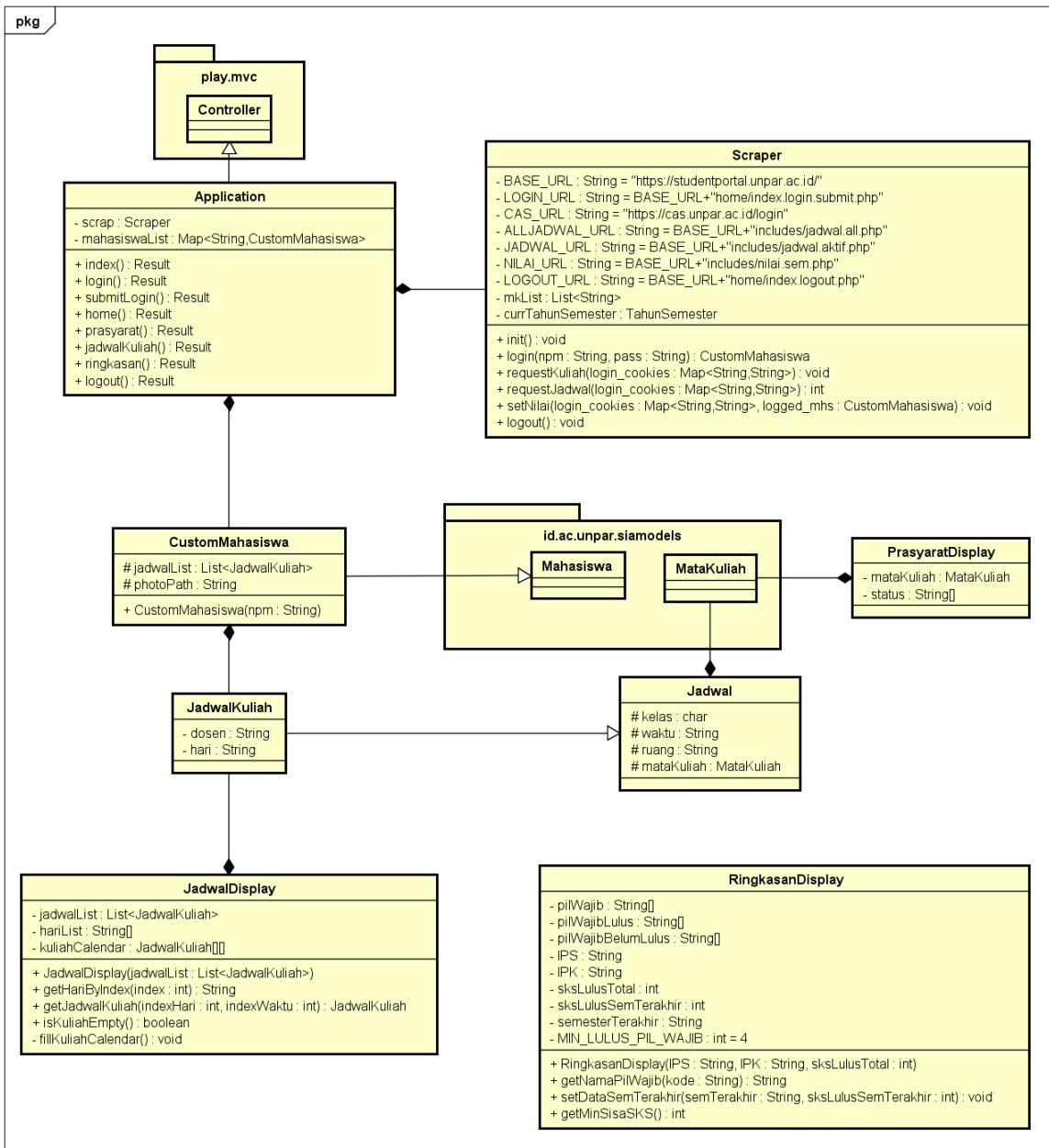
BAB 4

²

PERANCANGAN

³ 4.1 Diagram Kelas Rinci

⁴ Diagram kelas rinci diperoleh dari hasil pengembangan diagram kelas analisis pada subbab
⁵ 3.6. Diagram kelas rinci dapat dilihat pada gambar 4.1.



Gambar 4.1: Diagram Kelas Rinci

- 1 Deskripsi kelas beserta fungsi dari diagram kelas rinci tersebut adalah sebagai berikut:
1. Application
- Kelas ini merupakan turunan dari kelas Controller yang dimiliki oleh Play Framework sehingga menjadikan kelas ini sebagai controller dari aplikasi Informatika Student Portal. Atribut yang dimiliki kelas ini antara lain:
- **Scrapper scrap:** objek Scrapper yang berperan dalam pengambilan data dari Porta Akademik Mahasiswa.
 - **Map<String,CustomMahasiswa> mahasiswaList:** kumpulan objek Mahasiswa merepresentasikan mahasiswa yang sedang menggunakan aplikasi.
- Method-method yang dimiliki kelas ini merupakan *action method* dengan rincian sebagai berikut:

- 1 ● **public Result index()**
2 Berfungsi untuk mengarahkan pengguna ke halaman Informatika Student Portal.
3 **Kembalian:** halaman *login* jika pengguna belum *login* atau halaman utama jika pengguna sudah *login*.
- 4
5 ● **public Result login()**
6 Berfungsi untuk mengarahkan pengguna ke halaman *login*.
7 **Kembalian:** halaman *login* jika pengguna belum *login* atau halaman utama jika pengguna sudah *login*.
- 8
9 ● **public Result submitLogin()**
10 Berfungsi untuk mengirimkan data dari halaman *login* sekaligus melakukan validasi akun.
11
12 **Kembalian:** halaman utama jika *login* berhasil atau halaman *login* jika *login* gagal.
- 13
14 ● **public Result home()**
15 Berfungsi untuk mengarahkan pengguna ke halaman utama.
16 **Kembalian:** halaman utama.
- 17
18 ● **public Result prasyarat()**
19 Berfungsi untuk mengarahkan pengguna ke halaman prasyarat mata kuliah.
20 **Kembalian:** halaman prasyarat mata kuliah.
- 21
22 ● **public Result jadwalKuliah()**
23 Berfungsi untuk mengarahkan pengguna ke halaman jadwal kuliah.
24 **Kembalian:** halaman jadwal kuliah.
- 25
26 ● **public Result ringkasan()**
27 Berfungsi untuk mengarahkan pengguna ke halaman ringkasan data akademik.
28 **Kembalian:** halaman ringkasan data akademik.
- 29
30 ● **public Result logout()**
31 Berfungsi untuk mengeluarkan pengguna yang sedang *login*.
32 **Kembalian:** halaman *login*.

29 2. Scaper

30 Kelas ini mengimplementasikan *library* jsoup untuk melakukan pengambilan data dari
31 Portal Akademik Mahasiswa. Atribut yang dimiliki kelas ini antara lain:

- 32 ● **String BASE_URL:** URL Portal Akademik Mahasiswa.
- 33 ● **String LOGIN_URL:** URL *login* Portal Akademik Mahasiswa.
- 34 ● **String CAS_URL:** URL *login* CAS UNPAR.
- 35 ● **String ALLJADWAL_URL:** URL jadwal seluruh fakultas pada Portal Akademik Mahasiswa.
- 36
37 ● **String JADWAL_URL:** URL jadwal mahasiswa pada Portal Akademik Ma-
38 hasiswa.
- 39
40 ● **String NILAI_URL:** URL riwayat nilai mahasiswa pada Portal Akademik Ma-
41 hasiswa.
- 42 ● **String LOGOUT_URL:** URL *logout* Portal Akademik Mahasiswa.

- 1 • **TahunSemester currTahunSemester:** tahun ajaran dan semester terkini.
- 2 • **List<MataKuliah> mkList:** daftar mata kuliah yang dibuka pada semester
- 3 terkini.

4 *Method-method* yang dimiliki kelas ini adalah sebagai berikut:

- 5 • **public void init()**

6 Berfungsi untuk menginisialisasi koneksi ke Portal Akademik Mahasiswa.

7 **Kembalian:** tidak ada.

- 8 • **public CustomMahasiswa login(String npm, String pass)**

9 Berfungsi untuk melakukan *login*.

10 **Parameter:**

11 – **npm** NPM mahasiswa.

12 – **pass** *password* mahasiswa.

13 **Kembalian:** objek Mahasiswa.

- 14 • **public void requestKuliah(Map<String, String> login_cookies)**

15 Berfungsi untuk mendapatkan daftar mata kuliah yang dibuka pada semester

16 terkini.

17 **Parameter:**

18 – **login_cookies** *cookie* mahasiswa yang telah *login*.

19 **Kembalian:** tidak ada.

- 20 • **public List<JadwalKuliah> requestJadwal(Map<String, String> login_cookies)**

21 Berfungsi untuk mendapatkan jadwal kuliah mahasiswa pada semester terkini.

22 **Parameter:**

23 – **login_cookies** *cookie* mahasiswa yang telah *login*.

24 **Kembalian:** jadwal kuliah mahasiswa pada semester terkini.

- 25 • **public void setNilai(Map<String, String> login_cookies, Mahasiswa**
- 26 **logged_mhs)**

27 Berfungsi untuk mendapatkan riwayat nilai mahasiswa.

28 **Parameter:**

29 – **login_cookies** *cookie* mahasiswa yang telah *login*.

30 – **logged_mhs** objek Mahasiswa dari mahasiswa yang telah *login*.

31 **Kembalian:** tidak ada.

- 32 • **public void logout()**

33 Berfungsi untuk melakukan *logout*.

34 **Kembalian:** tidak ada.

35 3. CustomMahasiswa

36 Kelas ini merupakan turunan dari kelas Mahasiswa. Kelas ini merepresentasikan ma-

37 hasiswa yang memiliki jadwal kuliah dan URL foto profil. Atribut yang dimiliki kelas

38 ini antara lain:

- 39 • **String photoPath:** URL foto profil mahasiswa pada Portal Akademik Maha-
- 40 siswa.

- 1 • **List<JadwalKuliah> jadwalList:** daftar jadwal kuliah mahasiswa.

2 *Method-method* yang dimiliki kelas ini adalah sebagai berikut:

- 3 • **public CustomMahasiswa(String npm)**

4 Merupakan *constructor* dari kelas CustomMahasiswa.

5 **Parameter:**

- 6 – **npm** Nomor Pokok Mahasiswa.

7 4. Jadwal

8 Kelas ini merepresentasikan jadwal di UNPAR. Atribut yang dimiliki kelas ini antara
9 lain:

- 10 • **MataKuliah mataKuliah:** mata kuliah.

- 11 • **char kelas:** kelas kuliah.

- 12 • **String waktu:** waktu pelaksanaan kuliah.

- 13 • **String ruang:** ruang kuliah.

14 5. JadwalKuliah

15 Kelas ini merepresentasikan jadwal di UNPAR. Atribut yang dimiliki kelas ini antara
16 lain:

- 17 • **MataKuliah mataKuliah:** mata kuliah.

- 18 • **char kelas:** kelas kuliah.

- 19 • **String waktu:** waktu pelaksanaan kuliah.

- 20 • **String ruang:** ruang kuliah.

21 6. PrasyaratDisplay

22 Kelas berfungsi sebagai media pengiriman data ke halaman prasyarat mata kuliah.

23 Atribut yang dimiliki kelas ini antara lain:

- 24 • **MataKuliah mataKuliah:** mata kuliah.

- 25 • **String[] status:** status pengambilan mata kuliah.

26 7. JadwalDisplay

27 Kelas berfungsi sebagai media pengiriman data ke halaman jadwal mata kuliah. Atri-
28 but yang dimiliki kelas ini antara lain:

- 29 • **JadwalKuliah[][] kuliahCalendar:** jadwal kuliah mahasiswa dalam *array*.

- 30 • **String[] hariList:** nama-nama hari dalam String.

- 31 • **List<JadwalKuliah> jadwalList:** daftar jadwal kuliah mahasiswa.

32 *Method-method* yang dimiliki kelas ini adalah sebagai berikut:

- 33 • **public JadwalDisplay(List<JadwalKuliah> jadwalList)**

34 Merupakan *constructor* dari kelas JadwalDisplay.

35 **Parameter:**

- 1 – **jadwalList** jadwal kuliah mahasiswa.
- 2 • **public String getHariByIndex(int index)**
3 Berfungsi untuk mendapatkan hari berdasarkan angka index. Angka index dimulai dari 0 sedangkan hari dimulai dari Senin.
- 4 **Parameter:**
- 5 – **index** angka index hari.
- 6 **Kembalian:** hari dalam String.
- 7 • **public JadwalKuliah getJadwalKuliah(int indexHari, int indexWaktu)**
8 Berfungsi untuk mendapatkan jadwal kuliah dari atribut kuliahCalendar.
- 9 **Parameter:**
- 10 – **indexHari** angka index hari.
- 11 – **indexWaktu** angka index waktu.
- 12 **Kembalian:** jadwal kuliah.
- 13 • **public boolean isKuliahEmpty()**
14 Berfungsi untuk memeriksa apakah nilai dari jadwal kuliah kosong.
15 **Kembalian:** `true` jika kosong, `false` jika tidak kosong.
- 16 • **private void fillKuliahCalendar()**
17 Berfungsi untuk mengisi atribut kuliahCalendar berdasarkan atribut jadwalList.
18 **Kembalian:** tidak ada.

20 8. RingkasanDisplay

21 Kelas berfungsi sebagai media pengiriman data ke halaman ringkasan data akademik.
22 Atribut yang dimiliki kelas ini antara lain:

- 23 • **String[] pilWajib:** kode-kode mata kuliah pilihan wajib.
- 24 • **String[] pilWajibLulus:** kode-kode mata kuliah pilihan wajib yang sudah lulus.
- 25 • **String[] pilWajibBelumLulus:** kode-kode mata kuliah pilihan wajib yang belum lulus.
- 26 • **String IPS:** IPS mahasiswa.
- 27 • **String IPK:** IPK mahasiswa.
- 28 • **String semesterTerakhir:** semester terakhir yang telah ditempuh mahasiswa.
- 29 • **int sksLulusTotal:** total sks lulus mahasiswa.
- 30 • **int sksLulusSemTerakhir:** sks lulus mahasiswa pada semester terakhir.
- 31 • **int MIN_LULUS_PIL_WAJIB:** banyaknya mata kuliah pilihan wajib minimum untuk kelulusan dengan nilai 4.

34 Method-method yang dimiliki kelas ini adalah sebagai berikut:

- 35 • **public RingkasanDisplay(String IPS, String IPK, int sksLulusTotal)**
36 Merupakan *constructor* dari kelas Ringkasan Display.
- 37 **Parameter:**
- 38 – **IPS** IPS mahasiswa.

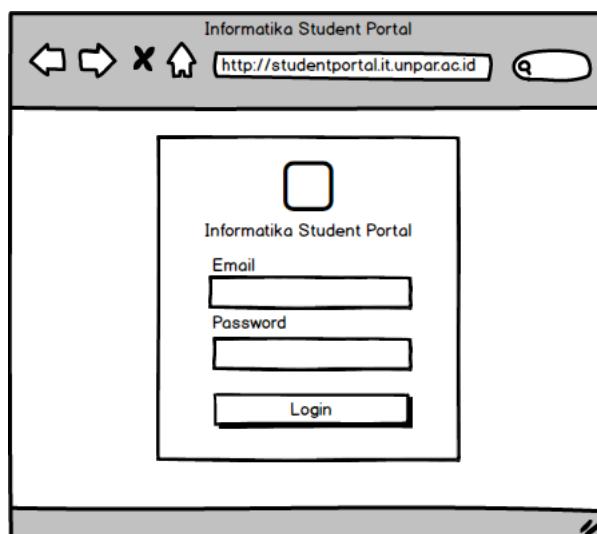
- 1 – **IPK** IPK mahasiswa.
- 2 – **sksLulusTotal** total sks lulus mahasiswa.
- 3 • **public String getNamaPilWajib(String kode)**
Berfungsi untuk mendapatkan nama mata kuliah pilihan wajib.
- 4 **Parameter:**
 - 5 – **kode** kode mata kuliah pilihan wajib.
- 6 **Kembalian:** nama mata kuliah pilihan wajib.
- 7 • **public void setDataSemTerakhir(String semTerakhir, int sksLulusSemTerakhir)**
Berfungsi untuk mengisi data semester terakhir.
- 8 **Parameter:**
 - 9 – **semTerakhir** semester terakhir yang telah ditempuh mahasiswa.
 - 10 – **sksLulusSemTerakhir** sks lulus mahasiswa pada semester terakhir.
- 11 **Kembalian:** tidak ada.
- 12 • **public int getMinSisaSKS()**
Berfungsi untuk mendapatkan sisa sks minimum untuk kelulusan.
- 13 **Kembalian:** sisa sks untuk minimum kelulusan.

18 4.2 Perancangan Antarmuka

19 Untuk memenuhi kebutuhan interaksi antara pengguna dengan sistem, maka dirancanglah
20 sebuah antarmuka dari Informatika Student Portal. Rancangan antarmuka dibagi menjadi
21 lima halaman web antara lain:

22 1. Antarmuka Halaman *Login*

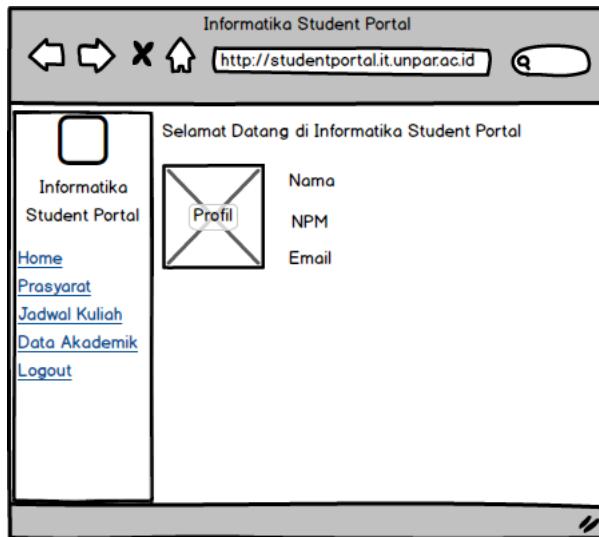
23 Halaman ini digunakan untuk melakukan *login*. Komponen halaman ini terdiri dari
24 logo aplikasi, kolom *email*, kolom *password*, dan tombol *login* seperti yang ditunjukkan
25 pada gambar 4.2. Untuk melakukan *login*, pengguna perlu memasukkan *email* dan
26 *password* yang sesuai kemudian menekan tombol *login*. Jika berhasil pengguna akan
27 diarahkan ke halaman *home*.



Gambar 4.2: Rancangan Halaman *Login*

1 2. Antarmuka Halaman *Home*

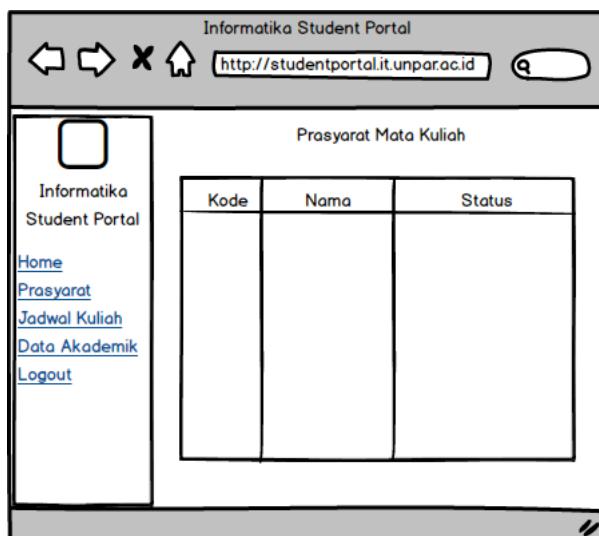
2 Halaman **home** merupakan halaman yang pertama dituju setelah melakukan **login**.
 3 Halaman ini, menampilkan identitas pengguna seperti foto profil, nama, Nomor Induk
 4 Mahasiswa(NPM), dan *email* mahasiswa. Rancangan antarmuka halaman *home* dapat
 5 dilihat pada gambar 4.3.



Gambar 4.3: Rancangan Halaman *Home*

6 3. Antarmuka Halaman Prasyarat Mata Kuliah

7 Halaman ini menampilkan tabel prasyarat mata kuliah yang dibuka pada semester
 8 terkini. Tabel tersebut memiliki tiga kolom yaitu kode mata kuliah, nama mata kuliah,
 9 dan status pengambilan mata kuliah. Rancangan antarmuka halaman prasyarat mata
 10 kuliah dapat dilihat pada gambar 4.4.

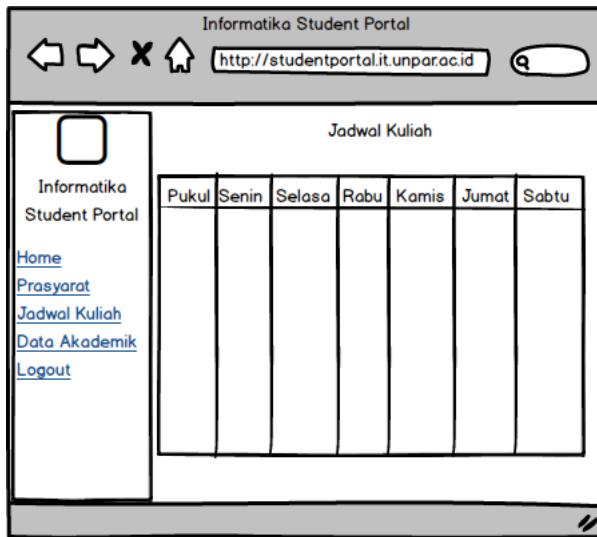


Gambar 4.4: Rancangan Halaman Prasyarat Mata Kuliah

11 4. Antarmuka Halaman Jadwal Kuliah

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

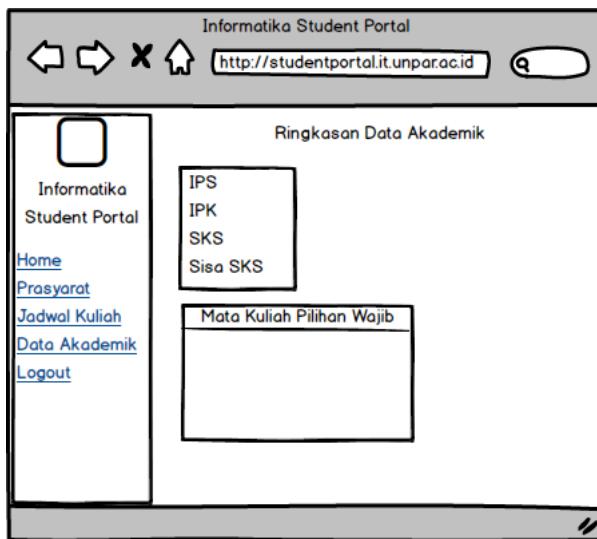
1 gambar 4.5.



Gambar 4.5: Rancangan Halaman Jadwal Kuliah

2 5. Antarmuka Halaman Data Akademik

3 Halaman ini menampilkan ringkasan informasi akademik pengguna yaitu IPS semester
4 terakhir, IPK, SKS lulus, sisa SKS menuju kelulusan, dan status pengambilan mata
5 kuliah pilihan wajib. Rancangan antarmuka halaman data akademik dapat dilihat
6 pada gambar 4.6.



Gambar 4.6: Rancangan Halaman Data Akademik

BAB 5

IMPLEMENTASI DAN PENGUJIAN

3 5.1 Implementasi

5.1.1 Lingkungan Implementasi

5 Implementasi perangkat lunak ini dilakukan di dua buah komputer. Implementasi pertama
6 dilakukan pada komputer peneliti untuk keperluan pengujian fungsional. Komputer tersebut
7 memiliki spesifikasi sebagai berikut:

- 8 1. Processor: 3.20Ghz
 - 9 2. RAM: 4.00 GB DDR3
 - 10 3. Sistem Operasi: Windows 8.1 Pro 64-bit
 - 11 4. Versi Java: 1.8.0 40

12 Implementasi kedua dilakukan pada komputer server yang terhubung dengan jaringan FTIS
13 untuk keperluan pengujian eksperimental. Komputer tersebut memiliki spesifikasi sebagai
14 berikut:

- 15 1. Processor: 2.66Ghz
 - 16 2. RAM: 4.00 GB DDR2
 - 17 3. Sistem Operasi: Ubuntu server amd-64
 - 18 4. Versi Java: 1.8.0 66

19 5.1.2 Hasil Implementasi

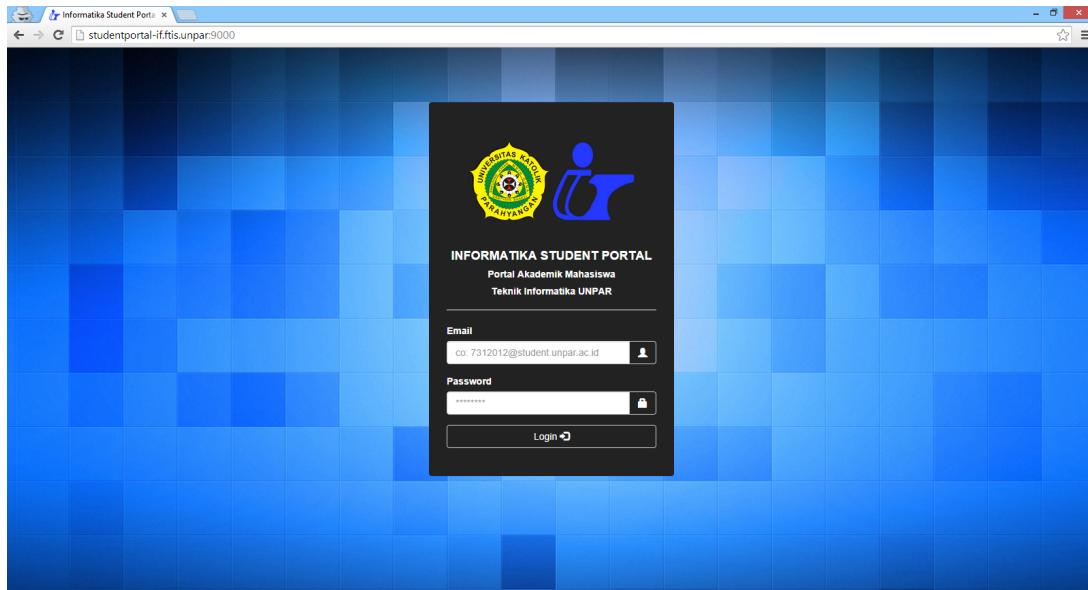
Hasil implementasi berupa aplikasi berbasis web responsif yang menggunakan *framework* Bootstrap¹. Aplikasi dapat diakses pada jaringan FTIS dengan URL <http://studentportal-if.ftis.unpar:9000>. Aplikasi Informatika Student Portal terdiri dari lima halaman antara lain:

24 1. Halaman *Login*

Halaman *login* digunakan pengguna untuk masuk ke dalam aplikasi. Pada halaman ini, pengguna dapat melakukan *login* dengan mengisi *email* pada kolom *email* dan

¹ <http://getbootstrap.com/>, diakses 20 November 2015

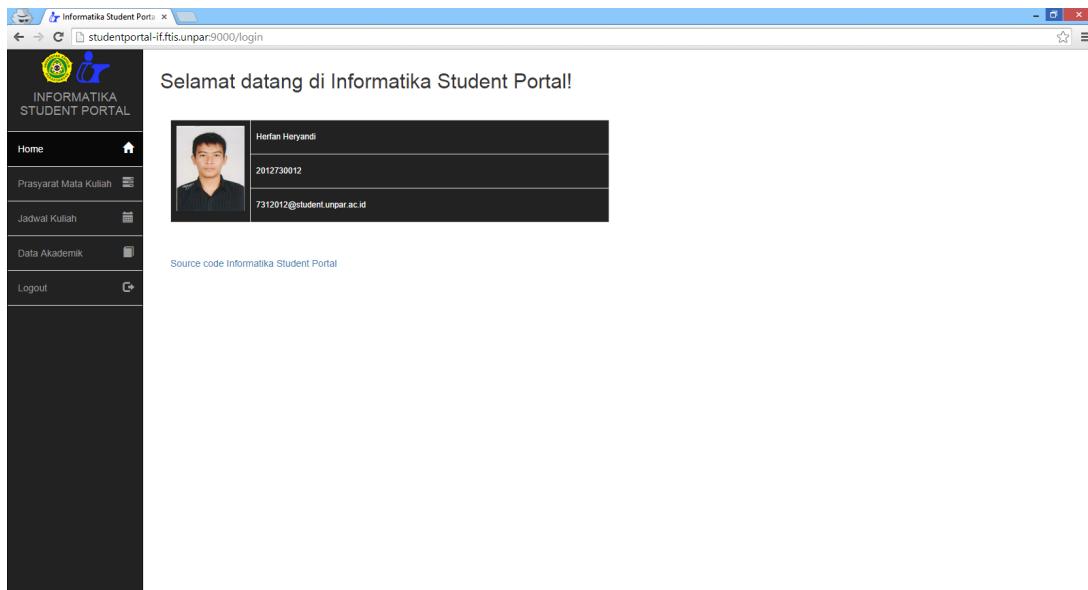
1 *password* pada kolom *password* kemudian mengklik tombol login. Tangkapan layar
 2 dari halaman *login* dapat dilihat pada gambar 5.1.



Gambar 5.1: Halaman *Login*

3.2. Halaman Utama

4 Halaman utama merupakan halaman yang pertama kali dituju setelah melakukan *log-
 5 in*. Halaman utama menampilkan identitas pengguna dan *link* menuju kode sumber
 6 aplikasi Informatika Student Portal. Pada halaman ini juga terdapat *sidebar menu*
 7 yang diperoleh dari *template* StartBootstrap². *Template* tersebut juga digunakan pa-
 8 da halaman prasyarat mata kuliah, jadwal kuliah, dan data akademik. Tangkapan
 9 layar dari halaman utama dapat dilihat pada gambar 5.2.

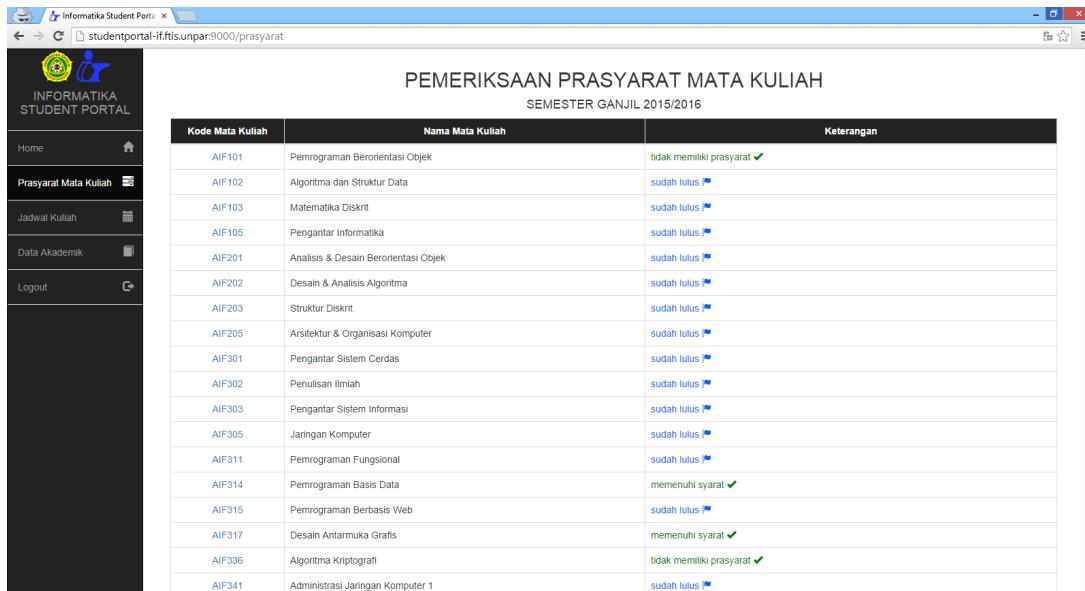


Gambar 5.2: Halaman Utama

3.3. Halaman Prasyarat Mata Kuliah

²<http://startbootstrap.com/>, diakses 20 November 2015

- 1 Halaman ini menampilkan tabel prasyarat mata kuliah. Jika prasyarat mata kuliah
 2 tersedia, pengguna dapat mengklik kode mata kuliah, kemudian akan diarahkan ke
 3 kode sumber aturan prasyarat mata kuliah tersebut. Tangkapan layar dari halaman
 4 prasyarat mata kuliah dapat dilihat pada gambar 5.3.

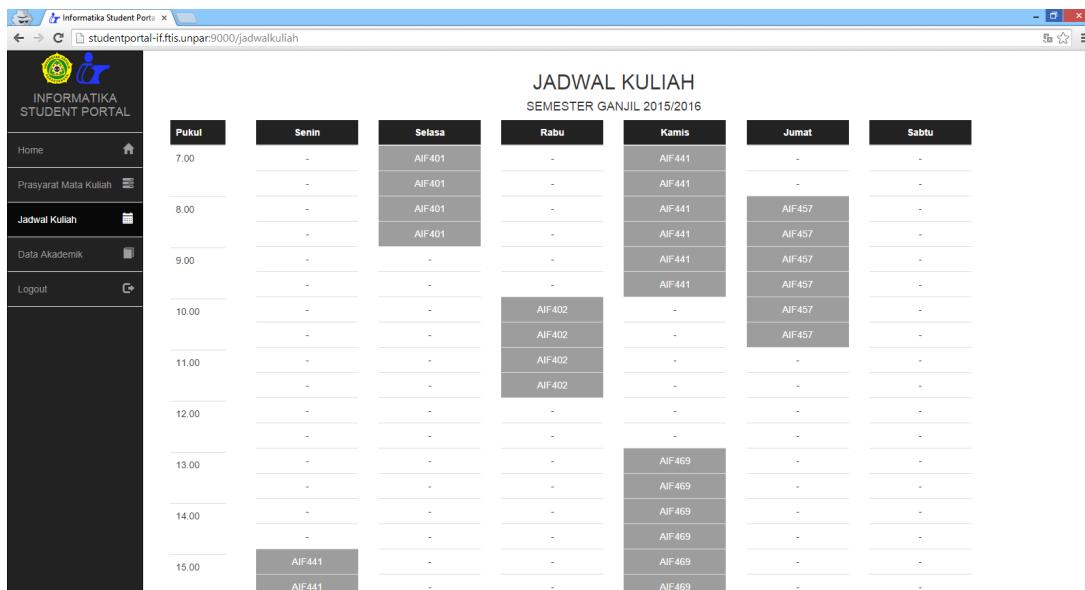


Kode Mata Kuliah	Nama Mata Kuliah	Keterangan
AIF101	Pemrograman Berorientasi Objek	tidak memiliki prasyarat ✓
AIF102	Algoritma dan Struktur Data	sudah lulus ✅
AIF103	Matematika Diskrit	sudah lulus ✅
AIF105	Pengantar Informatika	sudah lulus ✅
AIF201	Analisis & Desain Berorientasi Objek	sudah lulus ✅
AIF202	Desain & Analisis Algoritma	sudah lulus ✅
AIF203	Struktur Diskrit	sudah lulus ✅
AIF205	Arsitektur & Organisasi Komputer	sudah lulus ✅
AIF301	Pengantar Sistem Cerdas	sudah lulus ✅
AIF302	Penulisan Ilmiah	sudah lulus ✅
AIF303	Pengantar Sistem Informasi	sudah lulus ✅
AIF305	Jaringan Komputer	sudah lulus ✅
AIF311	Pemrograman Fungsional	sudah lulus ✅
AIF314	Pemrograman Basis Data	memenuhi syarat ✓
AIF315	Pemrograman Berbasis Web	sudah lulus ✅
AIF317	Desain Antarmuka Grafs	memenuhi syarat ✓
AIF336	Algoritma Kriptografi	tidak memiliki prasyarat ✓
AIF341	Administrasi Jaringan Komputer 1	sudah lulus ✅

Gambar 5.3: Halaman Prasyarat Mata Kuliah

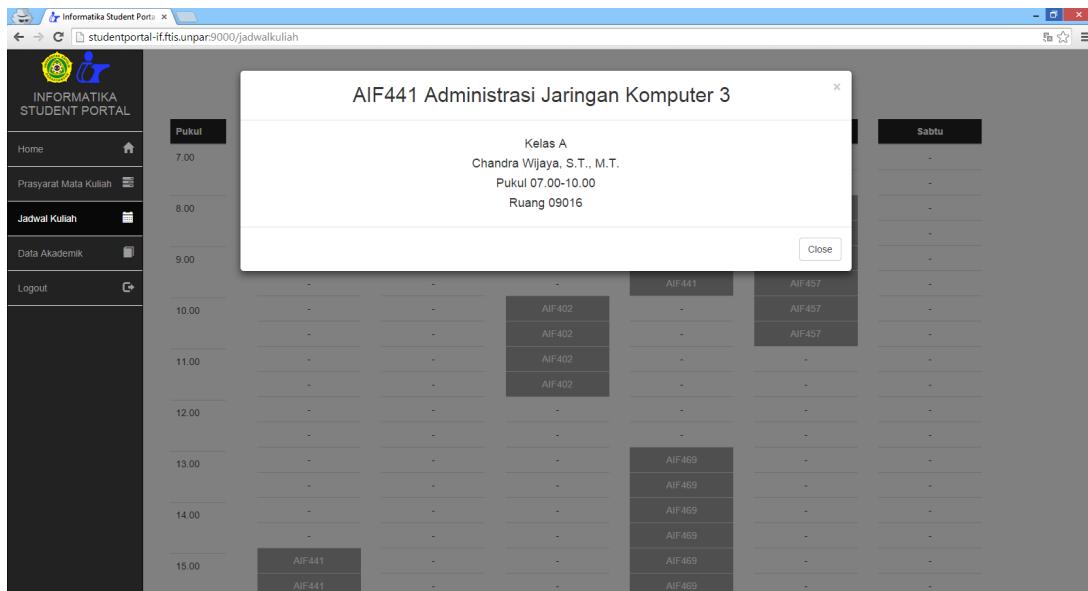
5 4. Halaman Jadwal Kuliah

- 6 Halaman ini menampilkan jadwal kuliah yang tersusun dan terurut berdasarkan hari.
 7 Tangkapan layar dari halaman jadwal kuliah dapat dilihat pada gambar 5.4. Jika kode
 8 mata kuliah diklik, akan muncul *popup* seperti pada gambar 5.5 yang berisi rincian
 9 dari jadwal kuliah tersebut.



Pukul	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
7.00	-	-	-	-	-	-
8.00	-	-	-	-	-	-
9.00	-	-	-	-	-	-
10.00	-	-	-	AIF402	-	-
11.00	-	-	-	AIF402	-	-
12.00	-	-	-	AIF402	-	-
13.00	-	-	-	-	-	-
14.00	-	-	-	-	-	-
15.00	-	AIF441	-	-	-	-
		AIF441	-	-	-	-

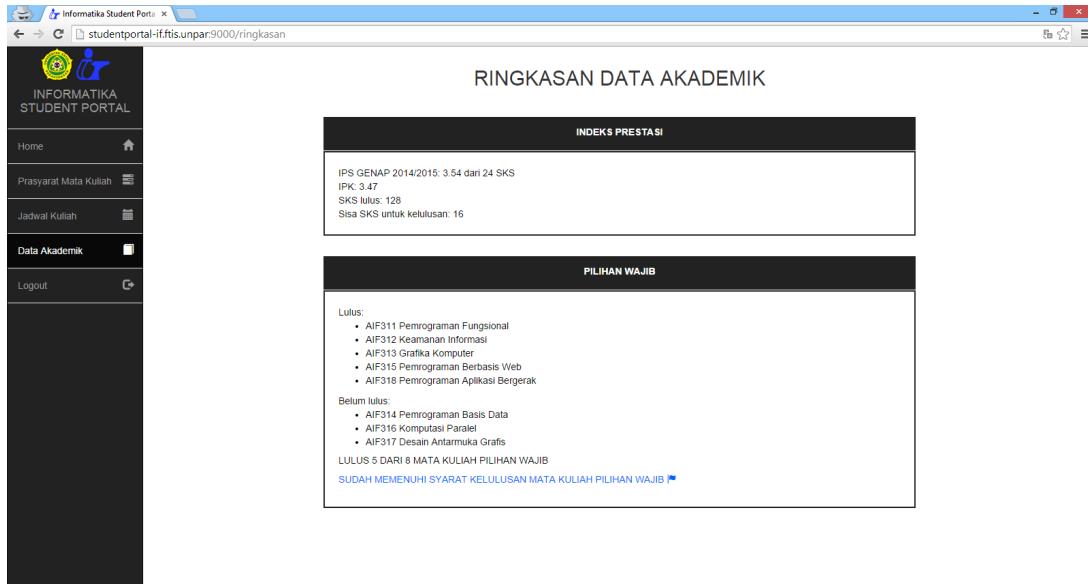
Gambar 5.4: Halaman Jadwal Kuliah



Gambar 5.5: Rincian Jadwal Kuliah

5. Halaman Data Akademik

Halaman ini menampilkan data akademik berupa IPS dan IPK yang langsung berubah ketika nilai sudah muncul, sisa SKS, dan status kelulusan mata kuliah pilihan wajib. Tangkapan layar dari halaman data akademik dapat dilihat pada gambar 5.6.



Gambar 5.6: Halaman Data Akademik

5.2 Pengujian

5.2.1 Pengujian Fungsional

Pengujian fungsional dilakukan untuk mengetahui kesesuaian reaksi perangkat lunak dengan reaksi yang diharapkan berdasarkan aksi pengguna terhadap perangkat lunak. Pengujian ini dilakukan pada berbagai sistem yaitu Windows, Linux, MacOS, dan Android. Terdapat tujuh tes kasus yang diujikan, detail serta hasilnya dapat dilihat di tabel 5.1.

Tabel 5.1: Tabel Pengujian Fungsional

No.	Aksi Pengguna	Reaksi yang diharapkan	Reaksi Perangkat Lunak
1.	Pengguna menjalankan aplikasi	Halaman <i>login</i> akan ditampilkan	sesuai
2.	Pengguna memasukkan <i>email</i> dan <i>password</i>	Jika <i>email</i> dan <i>password</i> sesuai, pengguna akan diarahkan ke halaman utama. Jika <i>email</i> yang dimasukkan bukan <i>email student</i> UNPAR, akan ditampilkan pesan “Email tidak valid”	sesuai, namun foto profil tidak muncul jika belum menambahkan eksepsi terhadap sertifikat SSL UNPAR sesuai
		Jika <i>email</i> yang dimasukkan bukan <i>email mahasiswa teknik informatika</i> , akan ditampilkan pesan “Maaf, Anda bukan mahasiswa teknik informatika”	sesuai
		Jika <i>email</i> dan <i>password</i> tidak sesuai atau mahasiswa bukan mahasiswa aktif, akan ditampilkan pesan “Password yang Anda masukkan salah atau Anda bukan mahasiswa aktif”	sesuai
3.	Pengguna memilih menu “Prasyarat Mata Kuliah”	Jika pengguna belum memiliki riwayat nilai(masih menempuh semester 1), akan ditampilkan pesan “PRASYARAT BELUM TERSEDIA”	sesuai
		Jika pengguna sudah memiliki riwayat nilai akan ditampilkan tabel prasyarat mata kuliah beserta status pengambilannya	sesuai
4.	Pengguna memilih menu “Jadwal Kuliah”	Jika pengguna belum melakukan FRS, cuti studi, atau jadwal kuliah pengguna belum tersedia, akan ditampilkan pesan “JADWAL KULIAH BELUM TERSEDIA”	sesuai
		Jika jadwal kuliah pengguna sudah tersedia, akan ditampilkan jadwal kuliah dalam bentuk kalender yang sudah diurutkan berdasarkan hari	sesuai
5.	Pengguna memilih menu “Data Akademik”	Jika pengguna belum memiliki riwayat nilai(masih menempuh semester 1), akan ditampilkan pesan “DATA AKADEMIK BELUM TERSEDIA”	sesuai
		Jika pengguna sudah memiliki riwayat nilai, akan ditampilkan ringkasan data akademik mahasiswa berupa IPS semester terakhir, IPK, SKS lulus, sisa SKS kelulusan, dan ringkasan data mengenai mata kuliah pilihan wajib	sesuai
6.	Pengguna memilih tombol <i>logout</i>	Pengguna akan diarahkan kembali ke halaman <i>login</i>	sesuai
7.	Dua pengguna menggunakan aplikasi secara bersamaan	Pengguna dapat menggunakan aplikasi dengan akun yang sesuai	sesuai

5.2.2 Pengujian Eksperimental

Pengujian eksperimental dilakukan terhadap mahasiswa angkatan 2012 sampai 2015. Dari setiap angkatan, diambil 3 orang untuk melakukan pengujian. Setiap responden diminta untuk melakukan *login* kemudian melihat data dari setiap halaman pada Portal Akademik Mahasiswa dan memastikan apakah data tersebut sudah sesuai dengan data sebenarnya. Responden menolak jika hasil pengujian diperlihatkan dengan alasan privasi, karena itu hasil pengujian dirangkum sebagai berikut:

- Angkatan 2012

Saat melakukan pemeriksaan prasyarat, terdapat mahasiswa dengan status belum lulus mata kuliah “AIF101 Pemrograman Berorientasi Objek”. Hal ini disebabkan oleh perubahan kurikulum pada tahun 2013. Kode mata kuliah Pemrograman Berorientasi Objek yang sudah ditempuh angkatan 2012 adalah AIF 191. “AIF191 Pemrograman Berorientasi Objek” merupakan hasil konversi mata kuliah “AKS141 Dasar-dasar Pemrograman”³. Hal tersebut sudah ditangani oleh aturan prasyarat SIA Models sehingga tidak mengakibatkan kesalahan pemeriksaan prasyarat mata kuliah. Selain itu, terdapat juga mahasiswa yang pernah cuti studi. Mahasiswa tersebut memiliki mata kuliah “XCT001 Cuti Studi” pada riwayat nilai yang tidak memiliki nilai akhir, namun aplikasi yang dibuat sudah dapat menanganinya dengan cara mengabaikan mata kuliah tanpa nilai akhir. Hasil pengujian eksperimental mahasiswa angkatan 2012 sesuai dengan hasil yang diharapkan.

- Angkatan 2013

Dalam pengujian terhadap mahasiswa angkatan 2013, terdapat mahasiswa yang pernah melakukan transfer studi. Total SKS lulus dari mahasiswa tersebut tidak sesuai dikarenakan aplikasi hanya menghitung SKS lulus saat dia melakukan perkuliahan di Program Studi Teknik Informatika. Akibatnya, pemeriksaan prasyarat mata kuliah dengan aturan SKS lulus memberikan hasil yang tidak sesuai. Selain itu, IPS, IPK, dan sisa SKS juga memberikan hasil yang tidak sesuai. Hasil pengujian eksperimental mahasiswa angkatan 2013 sesuai dengan hasil yang diharapkan kecuali terhadap mahasiswa yang transfer studi.

- Angkatan 2014

Mahasiswa angkatan 2014 belum pernah menempuh mata kuliah pilihan wajib. Pada halaman data akademik, ditampilkan bahwa mata kuliah pilihan wajib yang sudah lulus masih belum ada. Hasil pengujian eksperimental mahasiswa angkatan 2014 sesuai dengan hasil yang diharapkan.

- Angkatan 2015

Mahasiswa angkatan 2015 belum memiliki riwayat nilai sehingga belum bisa melihat ringkasan data akademik dan prasyarat mata kuliah. Hasil pengujian eksperimental mahasiswa angkatan 2015 sesuai dengan hasil yang diharapkan.

³<http://tinyurl.com/lionov><Kurikulum 2013><ATURAN KONVERSI KURIKULUM 2003/2008 ke KURIKLUM 2013>, diakses 25 November 2015

¹ **BAB 6**

² **KESIMPULAN DAN SARAN**

³ **6.1 Kesimpulan**

⁴ Dari hasil pembangunan aplikasi Informatika Student Portal, didapatkanlah kesimpulan-
⁵ kesimpulan sebagai berikut:

- ⁶ 1. Fitur-fitur yang dibuat untuk Informatika Student Portal antara lain prasyarat mata
⁷ kuliah, jadwal kuliah yang tersusun dan terurut berdasarkan hari, data akademik
⁸ berupa IPS dan IPK yang langsung berubah ketika nilai sudah muncul, sisa SKS, dan
⁹ status kelulusan mata kuliah pilihan wajib, dan aplikasi yang dapat diakses dari sistem
¹⁰ operasi manapun.
- ¹¹ 2. Telah berhasil mengimplementasikan *web scraping* menggunakan *library* jsoup. De-
¹² ngan *library* jsoup, data dari Portal Akademik Mahasiswa dapat diperoleh secara
¹³ langsung.
- ¹⁴ 3. Aplikasi Informatika Student Portal telah berhasil dibangun dengan menggunakan
¹⁵ Play Framework. Selain itu, aplikasi Informatika Student Portal juga memperoleh
¹⁶ data dari Portal Akademik Mahasiswa secara langsung menggunakan *library* jsoup
¹⁷ kemudian mengolah data tersebut dengan bantuan SIA Models.

¹⁸ **6.2 Saran**

¹⁹ Dari hasil penelitian termasuk kesimpulan yang didapat, berikut adalah beberapa saran
²⁰ untuk pengembangan:

- ²¹ 1. Penelitian ini memanfaatkan SIA Model terutama *package* “id.ac.unpar.siamodels.matakuliah”
²² yang berisi aturan mengenai prasyarat mata kuliah. Aturan-aturan tersebut dapat
²³ berubah suatu saat. Oleh karena itu, sebaiknya SIA Models dipisahkan dari *project*
²⁴ Play sehingga jika terjadi perubahan aturan pada SIA Models, tidak perlu dilakukan
²⁵ perubahan pada *project* yang dapat mengakibatkan kesalahan kode program.
- ²⁶ 2. Dalam pengembangan berikutnya, perlu diperhatikan data dari mahasiswa yang per-
²⁷ nah melakukan transfer studi. Jika memungkinkan, sebaiknya dianalisis langsung dari
²⁸ akun Portal Akademik Mahasiswa milik mahasiswa tersebut agar dapat mengambil
²⁹ data yang sesuai. Saat ini, aplikasi mengambil data langsung dari halaman riwayat ni-
³⁰ lai Portal Akademik Mahasiswa. Terdapat halaman lain yang memuat nilai mahasiswa

- 1 pada Portal Akademik Mahasiswa yaitu halaman Daftar Perkembangan Studi (DPS).
2 Sebaiknya pengambilan nilai dilakukan pada halaman DPS.
- 3 3. Saat ini, aplikasi berjalan pada port 9000 dan tidak dijalankan secara otomatis saat
4 server mulai berjalan. Sebaiknya aplikasi dapat langsung berjalan saat server mulai
5 berjalan dan aplikasi juga dapat berjalan pada port 80.
- 6 4. Aplikasi belum dapat menangani *path* yang tidak terdapat pada *routes* sehingga aplikasi
7 akan memberikan pesan error jika memasuki *path* tersebut. Dalam pengembangan
8 berikutnya, path yang tidak ditemukan sebaiknya diarahkan ke halaman “404 Not Fo-
9 und”.

1

DAFTAR REFERENSI

- 2 [1] Program Studi Teknik Informatika, "Prasyarat Mata Kuliah Semester Ganjil 2015/2016."
3 <https://tinyurl.com/lionov>, Juli 2015. [Online; diakses 1-Juli-2015].
- 4 [2] Biro Teknologi Informasi UNPAR, "Portal Akademik Mahasiswa." <https://studentportal.unpar.ac.id>, 2012. [Online; diakses 1-Juli-2015].
- 6 [3] N. Leroux and S. de Kaper, *Play for Java*. Manning Publications Co., 2014.
- 7 [4] Biro Teknologi Informasi UNPAR, *Buku Panduan Layanan TIK 2012*. Biro Teknologi
8 Informasi UNPAR, 2012.
- 9 [5] J. Hedley, "jsoup: Java HTML Parser." <http://jsoup.org>, 2009-2015. [Online; diakses
10 1-Juli-2015].
- 11 [6] P. Alfadian, "SIA Models." <https://github.com/pascalalfadian/SIAModels>, 2015.
12 [Online; diakses 1-Juli-2015].
- 13 [7] E. Vargiu and M. Urru, "Exploiting web scraping in a collaborative filtering-based approach
14 to web advertising," in *Artificial Intelligence Research*, (Barcelona,Spain), pp. 44–50,
15 2013.
- 16 [8] E. A. Meyer, *Selectors, Specificity, and the Cascade*. O'Reilly Media, 2012.
- 17 [9] Google, "Chrome DevTools." <https://developer.chrome.com/devtools>, 2013. [Online;
18 diakses 1-Juli-2015].

1

LAMPIRAN A

2

KODE PROGRAM

Listing A.1: Application.java

```

3 package controllers;
4
5 import java.io.IOException;
6 import java.lang.instrument.Instrumentation;
7 import java.util.ArrayList;
8 import java.util.HashMap;
9 import java.util.List;
10 import java.util.Map;
11
12 import org.jsoup.Connection;
13 import org.jsoup.nodes.Document;
14
15 import models.display.JadwalDisplay;
16 import models.display.PrasyaratDisplay;
17 import models.display.RingkasDisplay;
18 import models.id.ac.unpar.siamodels.Mahasiswa;
19 import models.id.ac.unpar.siamodels.MataKuliah;
20 import models.id.ac.unpar.siamodels.Nilai;
21 import models.id.ac.unpar.siamodels.MataKuliahFactory;
22 import models.id.ac.unpar.siamodels.Semester;
23 import models.id.ac.unpar.siamodels.matakuliah.interfaces.HasPrasyarat;
24 import models.support.CustomMahasiswa;
25 import models.support.JadwalKuliah;
26 import models.support.Scrap;
27 import play.*;
28 import play.data.DynamicForm;
29 import play.data.Form;
30 import play.mvc.*;
31 import views.html.*;
32
33 public class Application extends Controller {
34     Scrap scrap = new Scrap();
35     Map<String,CustomMahasiswa> mahasiswaList = new HashMap<String,CustomMahasiswa>();
36
37     public Result index() {
38         if(session("npm") == null) {
39             return ok(views.html.login.render(""));
40         }
41         else{
42             return home();
43         }
44     }
45
46     public Result login() {
47         if(session("npm") == null) {
48             return index();
49         }
50         else{
51             return home();
52         }
53     }
54
55     public Result submitLogin() throws IOException{
56         String errorHtml =
57             "<div class='alert alert-danger' role='alert'>" +
58             "<span class='glyphicon glyphicon-exclamation-sign' aria-hidden='true'></span>" +
59             "<span class='sr-only'>Error:</span>";
60         DynamicForm dynamicForm = Form.form().bindFromRequest();
61         String email = dynamicForm.get("email");
62         String pass = dynamicForm.get("pass");
63         if(!email.matches("[0-9]{7}@[student.unpar.ac.id]")){
64             return ok(views.html.login.render(errorHtml + "Email tidak valid" + "</div>"));
65         }
66         if(!(email.charAt(0) == '7' & email.charAt(1) == '3')) {
67             return ok(views.html.login.render(errorHtml + "Maaf, Anda bukan mahasiswa teknik informatika" + "</div>"));
68         }
69         String npm = "20" + email.substring(2,4) + email.substring(0,2) + "0" + email.substring(4,7);
70         CustomMahasiswa login_mhs = this.scrap.login(npm, pass);
71         if(login_mhs != null){
72             session("npm", npm);
73             mahasiswaList.put(session("npm"), login_mhs);
74             return home();
75         }
76     }
77
78 }
```

```

1     return ok(views.html.login.render(errorHtml+"Password yang Anda masukkan salah atau
2           Anda bukan mahasiswa aktif"+ "</div>"));
3   }
4 }
5
6 public Result home() {
7   if(session("npm") == null){
8     return index();
9   }
10  else{
11    return ok(views.html.home.render(mahasiswaList.get(session("npm"))));
12  }
13 }
14
15 public Result prasyarat() throws IOException{
16   if(session("npm") == null){
17     return index();
18   }
19   else if(mahasiswaList.get(session("npm")).getRiwayatNilai().size() == 0){
20     List<PrasyaratDisplay> table = null;
21     String semester = scrap.getSemester();
22     return ok(views.html.prasyarat.render(table,semester));
23   }
24   else{
25     List<PrasyaratDisplay> table = checkPrasyarat();
26     String semester = scrap.getSemester();
27     return ok(views.html.prasyarat.render(table,semester));
28   }
29 }
30
31 public Result jadwalKuliah() throws IOException{
32   if(session("npm") == null){
33     return index();
34   }
35   else{
36     JadwalDisplay table = new JadwalDisplay(mahasiswaList.get(session("npm")).getJadwalList());
37     String semester = scrap.getSemester();
38     return ok(views.html.jadwalKuliah.render(table,semester));
39   }
40 }
41
42 public Result ringkasan() throws IOException{
43   if(session("npm") == null){
44     return index();
45   }
46   else if(mahasiswaList.get(session("npm")).getRiwayatNilai().size() == 0){
47     RingkasanDisplay display = null;
48     return ok(views.html.ringkasan.render(display));
49   }
50   else{
51     Mahasiswa currMahasiswa = mahasiswaList.get(session("npm"));
52     RingkasanDisplay display = new RingkasanDisplay(
53       String.format("%.2f", currMahasiswa.calculateIPS()),
54       String.format("%.2f", currMahasiswa.calculateIPKLulus()),
55       currMahasiswa.calculateSKSLulus());
56     );
57     List<Nilai> riwayatNilai = currMahasiswa.getRiwayatNilai();
58     int lastIndex = riwayatNilai.size() - 1;
59     Semester semester = riwayatNilai.get(lastIndex).getSemester();
60     int tahunAjaran = riwayatNilai.get(lastIndex).getTahunAjaran();
61     int totalSKS = 0;
62     for(int i = lastIndex; i >= 0; i--){
63       Nilai nilai = riwayatNilai.get(i);
64       if(nilai.getSemester() == semester && nilai.getTahunAjaran() == tahunAjaran) {
65         if(nilai.getAngkaAkhir() != null) {
66           totalSKS += nilai.getMataKuliah().sks();
67         }
68       } else {
69         break;
70       }
71     }
72   }
73
74   String semTerakhir = semester + " " + tahunAjaran + "/" + (tahunAjaran + 1);
75   display.setDataSemTerakhir(semTerakhir, totalSKS);
76   String pilWajibLulus = new String();
77   String pilWajibBelumLulus = new String();
78   for(int i=0; i < display.getPilWajib().length; i++){
79     if(mahasiswaList.get(session("npm")).hasLulusKuliah(display.getPilWajib()[i])){
80       pilWajibLulus += display.getPilWajib()[i] + ",";
81     }
82     else{
83       pilWajibBelumLulus += display.getPilWajib()[i] + ",";
84     }
85   }
86   if(!pilWajibLulus.isEmpty()){
87     display.setPilWajibLulus(pilWajibLulus.split(","));
88   }
89   else{
90     display.setPilWajibLulus(new String[]{} );
91   }
92   if(!pilWajibBelumLulus.isEmpty()){
93     display.setPilWajibBelumLulus(pilWajibBelumLulus.split(","));
94   }
95   else{
96     display.setPilWajibBelumLulus(new String[]{} );
97   }
98   return ok(views.html.ringkasan.render(display));
99 }
100
101 public Result logout() throws IOException {
102   session().clear();
103 }
```

```

1     mahasiswaList.remove(session("npm"));
2     return index();
3 }
4
5     private List<PrasyaratDisplay> checkPrasyarat() throws IOException{
6         List<PrasyaratDisplay> table = new ArrayList<PrasyaratDisplay>();
7         List<MataKuliah> mkList = scrap.getMkList();
8         String MATAKULIAH_REPOSITORY_PACKAGE = "models.id.ac.unpar.siamodels.matakuliah";
9         List<Object> mkKnown = new ArrayList<Object>();
10        List<MataKuliah> mkUnknown = new ArrayList<MataKuliah>();
11        for(MataKuliah mk : mkList){
12            try {
13                Class<?> mkClass = Class.forName(MATAKULIAH_REPOSITORY_PACKAGE + "." + mk.kode());
14                Object matakuliah = mkClass.newInstance();
15                mkKnown.add(matakuliah);
16            } catch (ClassNotFoundException e) {
17                mkUnknown.add(mk);
18            } catch (InstantiationException e) {
19                e.printStackTrace();
20            } catch (IllegalAccessException e) {
21                e.printStackTrace();
22            }
23        }
24
25        for (Object mk: mkKnown) {
26            if (mk instanceof HasPrasyarat) {
27                List<String> reasons = new ArrayList<String>();
28                ((HasPrasyarat)mk).checkPrasyarat(mahasiswaList.get(session("npm")), reasons);
29                if (!reasons.isEmpty()) {
30                    String status = new String();
31                    for (String reason: reasons) {
32                        status+=reason + ";";
33                    }
34                    table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.getClass()
35                                         ().getSimpleName(), MataKuliahFactory.UNKNOWN_SKS, MataKuliahFactory.
36                                         UNKNOWN_NAMA),status.split(";")));
37                }
38            else{
39                if(mahasiswaList.get(session("npm")).hasLulusKuliah(mk.getClass().
40                                         getSimpleName())){
41                    table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.
42                                         getClass().getSimpleName(), MataKuliahFactory.UNKNOWN_SKS,
43                                         MataKuliahFactory.UNKNOWN_NAMA),new String[]{"sudah_lulus"}));
44                }
45            else{
46                table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.
47                                         getClass().getSimpleName(), MataKuliahFactory.UNKNOWN_SKS,
48                                         MataKuliahFactory.UNKNOWN_NAMA),new String[]{"memenuhi_prasyarat"}));
49            }
50        }
51    }
52    else{
53        if(mahasiswaList.get(session("npm")).hasLulusKuliah(mk.getClass().getSimpleName()))
54        {
55            table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.getClass()
56                                         ().getSimpleName(), MataKuliahFactory.UNKNOWN_SKS, MataKuliahFactory.
57                                         UNKNOWN_NAMA),new String[]{"sudah_lulus"}));
58        }
59        else{
56            table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.
57                                         getClass().getSimpleName(), MataKuliahFactory.UNKNOWN_SKS,
58                                         MataKuliahFactory.UNKNOWN_NAMA),new String[]{"tidak_memiliki_prasyarat"}));
59        }
60    }
61
62    for (MataKuliah mk: mkUnknown) {
63        table.add(new PrasyaratDisplay(mk,new String[]{"data_prasyarat_tidak_tersedia"}));
64    }
65
66    return table;
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }

```

Listing A.2: JadwalDisplay.java

```

75 package models.display;
76
77 import java.util.List;
78
79 import models.support.JadwalKuliah;
80
81 public class JadwalDisplay {
82     private List<JadwalKuliah> jadwalList;
83     private JadwalKuliah[][] kuliahCalendar;
84     private String[] hariList;
85
86     public JadwalDisplay(List<JadwalKuliah> jadwalList) {
87         this.jadwalList = jadwalList;
88         kuliahCalendar = new JadwalKuliah[6][22];
89         hariList = new String[]{"Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"};
90         fillKuliahCalendar();
91     }
92
93     public String getHariByIndex(int index) {
94         return this.hariList[index];
95     }
96
97     public JadwalKuliah getJadwalKuliah(int indexHari, int indexWaktu) {

```

```

1     return kuliahCalendar[indexHari][indexWaktu];
2 }
3
4 public boolean isKuliahEmpty(){
5     return jadwalList.isEmpty();
6 }
7
8 private void fillKuliahCalendar(){
9     if(!jadwalList.isEmpty()){
10         for (int i = 0; i < kuliahCalendar.length; i++) {
11             for (int j = 0; j < kuliahCalendar[i].length; j++) {
12                 kuliahCalendar[i][j] = new JadwalKuliah();
13             }
14         }
15         for (int i = 0; i < jadwalList.size(); i++) {
16             JadwalKuliah jdw = jadwalList.get(i);
17             int day = dayTranslate(jdw.getHari());
18             if(day!=-1){
19                 String[] timePair = jdw.getWaktu().split("-");
20                 String start = timePair[0];
21                 String end = timePair[1];
22                 int range = (Integer.parseInt(end.substring(0, 2))- Integer.parseInt(start.substring(0, 2)))*2;
23                 int beginIndex = 0;
24                 int half = Character.getNumericValue(start.charAt(3));
25                 if(half<3){
26                     beginIndex = (Integer.parseInt(start.substring(0, 2))-7)*2;
27                 }
28                 else if(half>=3){
29                     beginIndex =((Integer.parseInt(start.substring(0, 2))-7)*2)+1;
30                 }
31                 int endHalf = Character.getNumericValue(end.charAt(3));
32                 if(endHalf>3){
33                     range++;
34                 }
35                 for (int j = beginIndex; j < beginIndex+range; j++) {
36                     kuliahCalendar[day][j] = jdw;
37                 }
38             }
39         }
40     }
41 }
42
43
44 private int dayTranslate(String hari){
45     int day = -1;
46     switch(hari){
47         case "Senin":
48             day = 0;
49             break;
50         case "Selasa":
51             day = 1;
52             break;
53         case "Rabu":
54             day = 2;
55             break;
56         case "Kamis":
57             day = 3;
58             break;
59         case "Jumat":
60             day = 4;
61             break;
62         case "Sabtu":
63             day = 5;
64             break;
65         default:
66             day = -1;
67             break;
68     }
69     return day;
70 }
71 }

```

Listing A.3: PrasyaratDisplay.java

```

72 package models.display;
73
74 import models.id.ac.unpar.siamodels.MataKuliah;
75
76 public class PrasyaratDisplay {
77     private MataKuliah mataKuliah;
78     private String[] status;
79
80     public PrasyaratDisplay(MataKuliah mataKuliah, String[] status){
81         this.mataKuliah = mataKuliah;
82         this.status = status;
83     }
84
85     public MataKuliah getMataKuliah(){
86         return mataKuliah;
87     }
88
89     public String[] getStatus(){
90         return status;
91     }
92 }

```

Listing A.4: RingkasanDisplay.java

```

1  package models.display;
2
3  import models.id.ac.unpar.siamodels.MataKuliahFactory;
4
5
6  public class RingkasanDisplay {
7      private String[] pilWajib;
8      private String[] pilWajibLulus;
9      private String[] pilWajibBelumLulus;
10     private String IPS;
11     private String IPK;
12     private int sksLulusTotal;
13     private int sksLulusSemTerakhir;
14     private String semesterTerakhir;
15     private final int MIN_LULUS_PIL_WAJIB = 4;
16
17     public RingkasanDisplay(String IPS, String IPK, int sksLulusTotal){
18         this.IPS = IPS;
19         this.IPK = IPK;
20         this.sksLulusTotal = sksLulusTotal;
21         /*create mata kuliah pilihan wajib*/
22         pilWajib = new String[]{"AIF311","AIF312","AIF313","AIF314","AIF315","AIF316","AIF317",
23             "AIF318"};
24     }
25
26     public int getMinLulusPilWajib(){
27         return this.MIN_LULUS_PIL_WAJIB;
28     }
29
30     public String getNamaPilWajib(String kode){
31         return MataKuliahFactory.createMataKuliah(kode, MataKuliahFactory.UNKNOW_SKS,
32             MataKuliahFactory.UNKNOW_NAMA).nama()+"";
33     }
34
35     public String[] getPilWajibLulus() {
36         return pilWajibLulus;
37     }
38
39     public void setPilWajibLulus(String[] pilWajibLulus) {
40         this.pilWajibLulus = pilWajibLulus;
41     }
42
43     public String[] getPilWajibBelumLulus() {
44         return pilWajibBelumLulus;
45     }
46
47
48     public void setPilWajibBelumLulus(String[] pilWajibBelumLulus) {
49         this.pilWajibBelumLulus = pilWajibBelumLulus;
50     }
51
52     public String[] getPilWajib() {
53         return this.pilWajib;
54     }
55
56     public String getIPS(){
57         return this.IPS;
58     }
59
60     public String getIPK(){
61         return this.IPK;
62     }
63
64     public void setDataSemTerakhir(String semTerakhir, int sksLulusSemTerakhir) {
65         this.semesterTerakhir = semTerakhir;
66         this.sksLulusSemTerakhir = sksLulusSemTerakhir;
67     }
68
69     public String getSemesterTerakhir(){
70         return semesterTerakhir;
71     }
72     public int getSKSLulusTotal(){
73         return this.sksLulusTotal;
74     }
75
76     public int getSKSLulusSemTerakhir(){
77         return this.sksLulusSemTerakhir;
78     }
79
80     public int getMinSisaSKS(){
81         if(sksLulusTotal >=144){
82             return 0;
83         }
84         else{
85             return 144-sksLulusTotal;
86         }
87     }
88 }

```

Listing A.5: CustomMahasiswa.java

```

89 package models.support;
90
91 import java.util.ArrayList;
92 import java.util.List;
93
94 import models.id.ac.unpar.siamodels.Mahasiswa;
95
96 public class CustomMahasiswa extends Mahasiswa{

```

```

1   protected List<JadwalKuliah> jadwallist;
2   protected String photoPath;
3
4   public CustomMahasiswa(String npm) throws NumberFormatException {
5       super(npm);
6       this.jadwallist = new ArrayList<JadwalKuliah>();
7   }
8
9   public String getPhotoPath(){
10    return photoPath;
11}
12
13   public void setPhotoPath(String photoPath) {
14       this.photoPath = photoPath;
15   }
16
17   public List<JadwalKuliah> getJadwallist(){
18       return jadwallist;
19   }
20
21   public void setJadwallist(List<JadwalKuliah> jadwallist){
22       this.jadwallist = jadwallist;
23   }
24
25 }
```

Listing A.6: Jadwal.java

```

26 package models.support;
27
28 import models.id.ac.unpar.siamodels.MataKuliah;
29
30 public class Jadwal {
31     protected MataKuliah mataKuliah;
32     protected char kelas;
33     protected String waktu;
34     protected String ruang;
35
36     public Jadwal(MataKuliah mataKuliah, char kelas, String waktu, String ruang) {
37         this.mataKuliah = mataKuliah;
38         this.kelas = kelas;
39         this.waktu = waktu;
40         this.ruang = ruang;
41     }
42
43     public Jadwal() {
44
45     }
46
47     public MataKuliah getMataKuliah() {
48         return mataKuliah;
49     }
50
51     public char getKelas() {
52         return kelas;
53     }
54
55     public String getWaktu() {
56         return waktu;
57     }
58
59     public String getRuang() {
60         return ruang;
61     }
62
63 }
```

Listing A.7: JadwalKuliah.java

```

64 package models.support;
65
66 import models.id.ac.unpar.siamodels.MataKuliah;
67
68 public class JadwalKuliah extends Jadwal{
69     private String dosen;
70     private String hari;
71
72     public JadwalKuliah(MataKuliah mataKuliah, char kelas, String dosen, String hari, String waktu,
73             , String ruang) {
74         super(mataKuliah, kelas, waktu, ruang);
75         this.dosen = dosen;
76         this.hari = hari;
77     }
78
79     public JadwalKuliah() {
80         super();
81     }
82
83     public String getDosen() {
84         return dosen;
85     }
86
87     public String getHari() {
88         return hari;
89     }
90
91 }
```

Listing A.8: Scraper.java

```

1 package models.support;
2
3 import java.io.IOException;
4 import java.util.ArrayList;
5 import java.util.List;
6 import java.util.Map;
7
8 import models.id.ac.unpar.siamodels.Mahasiswa;
9 import models.id.ac.unpar.siamodels.Mahasiswa.Nilai;
10 import models.id.ac.unpar.siamodels.MataKuliah;
11 import models.id.ac.unpar.siamodels.MataKuliahFactory;
12 import models.id.ac.unpar.siamodels.Semester;
13 import models.id.ac.unpar.siamodels.TahunSemester;
14
15 import org.jsoup.Connection;
16 import org.jsoup.Connection.Response;
17 import org.jsoup.Jsoup;
18 import org.jsoup.nodes.Document;
19 import org.jsoup.nodes.Element;
20 import org.jsoup.select.Elements;
21
22 public class Scraper {
23     private final String BASE_URL = "https://studentportal.unpar.ac.id/";
24     private final String LOGIN_URL = BASE_URL + "home/index.login.submit.php";
25     private final String CAS_URL = "https://cas.unpar.ac.id/login";
26     private final String ALLJADWAL_URL = BASE_URL + "includes/jadwal.all.php";
27     private final String JADWAL_URL = BASE_URL + "includes/jadwal.aktif.php";
28     private final String NILAI_URL = BASE_URL + "includes/nilai.sem.php";
29     private final String LOGOUT_URL = BASE_URL + "home/index.logout.php";
30     private TahunSemester currTahunSemester;
31     private List<MataKuliah> mkList;
32
33     public List<MataKuliah> getMkList() {
34         return this.mkList;
35     }
36
37     public String getSemester() {
38         return currTahunSemester.getSemester() + "+" + currTahunSemester.getTahun() + "/" + (
39             currTahunSemester.getTahun() + 1);
40     }
41
42     public void init() throws IOException{
43         Connection baseConn = Jsoup.connect(BASE_URL);
44         baseConn.timeout(0);
45         baseConn.validateTLCertificates(false);
46         baseConn.method(Connection.Method.GET);
47         baseConn.execute();
48     }
49
50     public CustomMahasiswa login(String npm, String pass) throws IOException{
51         init();
52         CustomMahasiswa logged_mhs = new CustomMahasiswa(npm);
53         String user = logged_mhs.getEmailAddress();
54         Connection conn = Jsoup.connect(LOGIN_URL);
55         conn.data("Submit", "Login");
56         conn.timeout(0);
57         conn.validateTLCertificates(false);
58         conn.method(Connection.Method.POST);
59         Response resp = conn.execute();
60         Document doc = resp.parse();
61         String lt = doc.select("input[name=lt]").val();
62         String execution = doc.select("input[name=execution]").val();
63         /*CAS LOGIN*/
64         Connection loginConn = Jsoup.connect(CAS_URL);
65         loginConn.cookies(resp.cookies());
66         loginConn.data("username", user);
67         loginConn.data("password", pass);
68         loginConn.data("lt", lt);
69         loginConn.data("execution", execution);
70         loginConn.data("_eventId", "submit");
71         loginConn.timeout(0);
72         loginConn.validateTLCertificates(false);
73         loginConn.method(Connection.Method.GET);
74         resp = loginConn.execute();
75         if(resp.body().contains("Data_Akademik")){
76             Map<String, String> login_cookies = resp.cookies();
77             doc = resp.parse();
78             String nama = doc.select("p[class=student-name]").text();
79             logged_mhs.setNama(nama);
80             String curr_sem = doc.select(".main-info-semester-a").text();
81             String[] sem_set = this.parseSemester(curr_sem);
82             Element photo = doc.select(".student-photo-img").first();
83             String photoPath = photo.absUrl("src");
84             logged_mhs.setPhotoPath(photoPath);
85             currTahunSemester = new TahunSemester(Integer.parseInt(sem_set[0]), Semester.fromString(
86                 sem_set[1]));
87             this.requestKuliah(login_cookies);
88             List<JadwalKuliah> jadwalList = this.requestJadwal(login_cookies);
89             logged_mhs.setJadwalList(jadwalList);
90             this.setNilai(login_cookies, logged_mhs);
91             logout();
92             return logged_mhs;
93         }
94         else{
95             return null;
96         }
97     }
98
99     public void requestKuliah(Map<String, String> login_cookies) throws IOException{
100         Connection kuliahConn = Jsoup.connect(ALLJADWAL_URL);
101         kuliahConn.cookies(login_cookies);
102         kuliahConn.timeout(0);

```

```

1  kuliahConn.validateTLCertificates(false);
2  kuliahConn.method(Connection.Method.GET);
3  Response resp = kuliahConn.execute();
4  Document doc = resp.parse();
5  Elements jadwal = doc.select("tr");
6  String prev = "";
7  mList = new ArrayList<MataKuliah>();
8  for (int i = 1; i < jadwal.size() - 1; i++) {
9    Elements row = jadwal.get(i).children();
10   if (!row.get(1).text().equals("")) {
11     String kode = row.get(1).text();
12     String nama = row.get(2).text();
13     String sks = row.get(3).text();
14     if (!kode.equals(prev)) {
15       MataKuliah curr = MataKuliahFactory.createMataKuliah(kode, Integer.parseInt(
16         sks), nama);
17       mList.add(curr);
18     }
19     prev = kode;
20   }
21 }
22 }
23
24 public List<JadwalKuliah> requestJadwal(Map<String, String> login_cookies) throws IOException{
25   Connection jadwalConn = Jsoup.connect(JADWAL_URL);
26   jadwalConn.cookies(login_cookies);
27   jadwalConn.timeout(0);
28   jadwalConn.validateTLCertificates(false);
29   jadwalConn.method(Connection.Method.GET);
30   Response resp = jadwalConn.execute();
31   Document doc = resp.parse();
32   Elements jadwalTable = doc.select(".portal-full-table");
33   List<JadwalKuliah> jadwalList = new ArrayList<JadwalKuliah>();
34
35 /*Kuliah*/
36 if(jadwalTable.size()>0){
37   Elements tableKuliah = jadwalTable.get(0).select("tbody tr");
38   String kode = new String();
39   String nama = new String();
40   for(Element elem : tableKuliah){
41     if(elem.className().contains("row")){
42       if (!(elem.child(1).text().isEmpty() && elem.child(2).text().isEmpty())){
43         kode = elem.child(1).text();
44         nama = elem.child(2).text();
45       }
46       MataKuliah currMk = MataKuliahFactory.createMataKuliah(kode, Integer.parseInt(
47         elem.child(3).text()), nama);
48       jadwalList.add(new JadwalKuliah(currMk, elem.child(4).text().charAt(0), elem.
49         child(5).text(), elem.child(7).text(), elem.child(8).text(), elem.child(9).
50         text()));
51     }
52   }
53 }
54 return jadwalList;
55 }
56
57 public void setNilai(Map<String, String> login_cookies, Mahasiswa logged_mhs) throws
58   IOException{
59   Connection nilaiConn = Jsoup.connect(NILAI_URL);
60   nilaiConn.cookies(login_cookies);
61   nilaiConn.data("npm",logged_mhs.getNpm());
62   nilaiConn.data("thn_akd","ALL");
63   nilaiConn.timeout(0);
64   nilaiConn.validateTLCertificates(false);
65   nilaiConn.method(Connection.Method.POST);
66   Response resp = nilaiConn.execute();
67   Document doc = resp.parse();
68   Elements mk = doc.select("table");
69
70   for(Element tb:mk){
71     Elements tr = tb.select("tr");
72     String[] sem_set = this.parseSemester(tr.get(0).text());
73     String thn = sem_set[0];
74     String sem = sem_set[1];
75
76     for(Element td:tr){
77       if(td.className().contains("row")){
78         String kode = td.child(1).text();
79         int sks = Integer.parseInt(td.child(3).text());
80         String nama_mk = td.child(2).text();
81         MataKuliah curr_mk = MataKuliahFactory.createMataKuliah(kode, sks, nama_mk);
82         char kelas = td.child(4).text().charAt(0);
83         double ART = 0;
84         double UTS = 0;
85         double UAS = 0;
86         if(kelas=='*'){
87           ART = Double.parseDouble(td.child(5).text());
88           UTS = Double.parseDouble(td.child(6).text());
89           UAS = Double.parseDouble(td.child(7).text());
90         }
91         if(!td.child(9).text().equals ""){
92           char NA = td.child(9).text().charAt(0);
93           TahunSemester tahunSemesterNilai = new TahunSemester(Integer.parseInt(thn),
94             Semester.fromString(sem));
95           logged_mhs.getRiwayatNilai().add(new Nilai(tahunSemesterNilai, curr_mk,
96             kelas, ART, UTS, UAS, NA));
97         }
98       }
99     }
100   }
101 }
102 }
103 }
```

```
1  public void logout() throws IOException{
2      Connection logoutConn = Jsoup.connect(LOGOUT_URL);
3      logoutConn.timeout(0);
4      logoutConn.validateTLCSCertificates(false);
5      logoutConn.method(Connection.Method.GET);
6      logoutConn.execute();
7  }
8
9  private String[] parseSemester(String sem_raw){
10     String[] sem_set = sem_raw.split("/")[0].split("-");
11     return new String[]{sem_set[1].trim(),sem_set[0].trim()};
12 }
13 }
```