

SKRIPSI

**INFORMATIKA STUDENT PORTAL: PENGEMBANGAN
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* DEVELOPMENT FOR UNPAR'S INFORMATICS ENGINEERING STUDENTS



HERFAN HERYANDI

NPM: 2012730012

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

LEMBAR PENGESAHAN

**INFORMATIKA STUDENT PORTAL: PENGEMBANGAN
PORTAL AKADEMIK MAHASISWA UNTUK MAHASISWA
TEKNIK INFORMATIKA UNPAR**

HERFAN HERYANDI

NPM: 2012730012

Bandung, 18 Desember 2015

Menyetujui,

Pembimbing Tunggal

Thomas Anung Basuki, Ph.D.

Ketua Tim Penguji

Anggota Tim Penguji

Gede Karya, M.T.

Luciana Abednego, M.T.

Mengetahui,

Ketua Program Studi

Mariskha Tri Aditia, PDEng

PERNYATAAN

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

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

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

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

Dinyatakan di Bandung,
Tanggal 18 Desember 2015

Meterai

Herfan Heryandi
NPM: 2012730012

ABSTRAK

Portal Akademik Mahasiswa merupakan sebuah situs web yang diperuntukkan 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 ekstraksi data situs web 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 menggunakan CSS *Selector* kemudian data tersebut akan diolah ke dalam SIA Models.

Pengujian dari aplikasi Informatika Student Portal dilakukan oleh beberapa mahasiswa dengan kriteria yang beragam. Berdasarkan hasil pengujian, 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 ekstraksi data situs web 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, jsoup, Play Framework, SIA Models, Chrome DevTools, CSS *Selector*

ABSTRACT

Portal Akademik Mahasiswa is a website that is intended for students in order to obtain information about academic activities. However, its features has still not supported the academic needs of UNPAR's Informatics Engineering students. Therefore, it is made Informatika Student Portal application.

Informatika Student Portal is a web-based application which made with Play Framework. The data which processed by Informatika Student Portal is taken directly from Portal Akademik Mahasiswa by exploiting website data extraction which is implemented by jsoup. To obtain the data, it is necessary to analyze the communication of Portal Akademik Mahasiswa. The analysis is performed using Chrome DevTools. The data that was obtained by jsoup can be selected using CSS Selector then the data will be processed into SIA Models.

The testing of Informatika Student Portal was performed by some students with various criteria. The application can run well and give precise results in all students except students who have done studies transfer. The results of Informatika Student Portal testing prove that website data extraction can be used to meet the academic needs of UNPAR's Informatics Engineering students if it is combined with SIA Models.

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

Teknik Informatika UNPAR dan diri sendiri

KATA PENGANTAR

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

Bandung, Desember 2015

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	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	8
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	38
3.3 Analisis Komunikasi Portal Akademik Mahasiswa untuk Fitur Informatika Student Portal	42
3.3.1 Kasus <i>Login</i>	42
3.3.2 Kasus Nilai	45
3.3.3 Kasus Jadwal	47
3.3.4 Kasus <i>Logout</i>	51
3.4 Analisis Arsitektur Informatika Student Portal	53
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	56
4 PERANCANGAN	59
4.1 Diagram Kelas Rinci	59
4.2 Perancangan Antarmuka	64
5 IMPLEMENTASI DAN PENGUJIAN	69
5.1 Implementasi	69
5.1.1 Lingkungan Implementasi	69
5.1.2 Hasil Implementasi	69
5.2 Pengujian	73
5.2.1 Pengujian Fungsional	73
5.2.2 Pengujian Eksperimental	75
6 KESIMPULAN DAN SARAN	77
6.1 Kesimpulan	77
6.2 Saran	77
DAFTAR REFERENSI	79
A KODE PROGRAM	81

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	28
3.2 Menu Atas Home	28
3.3 Menu Atas Kuliah	29
3.4 Menu Atas Profil	29
3.5 Menu Atas Komentar	29
3.6 Identitas Portal	30
3.7 Menu Utama	30
3.8 Tampilan Registrasi FRS/PRS	31
3.9 Tampilan Kartu Rencana Studi[4]	31
3.10 Tampilan Pindah Kelas MKU[4]	32
3.11 Tampilan Jadwal Kuliah, UTS, dan UAS	32
3.12 Tampilan Jadwal MKU	33
3.13 Tampilan Jadwal Seluruh Fakultas	34
3.14 Tampilan Riwayat Per Semester	34
3.15 Tampilan Daftar Perkembangan Studi	35
3.16 Tampilan Statistik Nilai dan IP	35
3.17 Tampilan Riwayat Indeks Prestasi	36
3.18 Tampilan TOEFL	36
3.19 Tampilan Pembayaran Uang Kuliah	37
3.20 Tampilan Informasi	37
3.21 Tampilan <i>Pop Up</i> Informasi	37
3.22 Tampilan Kalender	38
3.23 Tampilan Info Browser	38
3.24 Tampilan Jadwal pada Portal Akademik Mahasiswa	39
3.25 Tampilan Jadwal yang Diinginkan Mahasiswa	40

3.26 Tampilan <i>Mobile</i> Portal Akademik Mahasiswa	41
3.27 Tombol “input#submit.login-button” pada Halaman Depan Portal Akademik Mahasiswa	43
3.28 Kolom “ <i>Username</i> ” “input#username.required” pada Halaman CAS UNPAR	44
3.29 <i>Form Data</i> yang dikirim CAS UNPAR	45
3.30 Elemen “a” dengan teks “Riwayat Per Semester” pada Menu Nilai dan Indeks Prestasi	46
3.31 <i>Combo Box</i> “select#tahun_akd_sec” pada Halaman Riwayat Per Semester	46
3.32 <i>Form Data</i> pada pengiriman Nilai Seluruh Tahun Akademik	47
3.33 Elemen “a” dengan teks “Kuliah, UTS dan UAS” pada Menu Jadwal	48
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	52
3.42 Arsitektur Informatika Student Portal	54
3.43 Diagram <i>Use Case</i> Informatika Student Portal	54
3.44 Diagram Kelas Analisis Informatika Student Portal	57
4.2 Rancangan Halaman <i>Login</i>	64
4.3 Rancangan Halaman <i>Home</i>	65
4.4 Rancangan Halaman Prasyarat Mata Kuliah	65
4.5 Rancangan Halaman Jadwal Kuliah	66
4.6 Rancangan Halaman Data Akademik	66
4.1 Diagram Kelas Rinci	67
5.1 Halaman <i>Login</i>	70
5.2 Halaman <i>Home</i>	71
5.3 Halaman Prasyarat Mata Kuliah	71
5.4 Halaman Jadwal Kuliah	72
5.5 Rincian Jadwal Kuliah	72
5.6 Halaman Data Akademik	73

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	42
5.1	Tabel Pengujian Fungsional	74

BAB 1

PENDAHULUAN

1.1 Latar Belakang

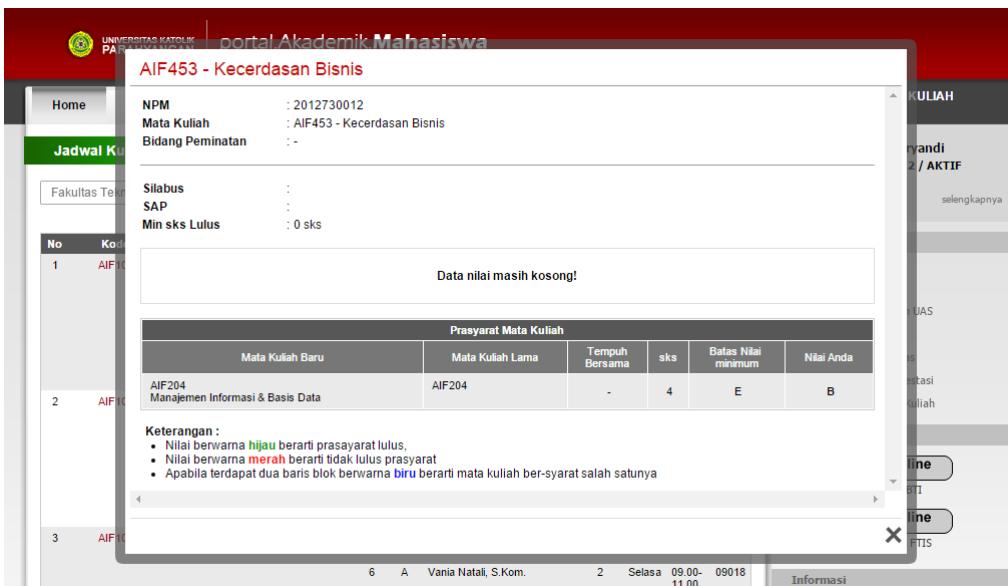
Portal Akademik Mahasiswa atau dikenal sebagai Student Portal UNPAR^[2] merupakan sistem informasi berbasis web yang digunakan oleh mahasiswa Universitas Katolik Parahyangan. Beberapa fitur yang dimiliki Portal Akademik Mahasiswa antara lain rencana studi, jadwal, nilai dan indeks prestasi, dan pembayaran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

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

jsoup^[5] merupakan *library* Java yang digunakan untuk menelusuri suatu situs web untuk mendapatkan suatu informasi. Informasi yang didapat berupa HTML yang kemudian diekstrak dan disajikan dalam bentuk *Document Object Model*. Play Framework^[3] merupakan sebuah *web framework* berbasis Java dan Scala. Play juga menggunakan *design pattern Model-View-Controller* (MVC) di mana *model* dan *controller* menggunakan bahasa Java sedangkan *view* menggunakan bahasa Scala dan HTML. SIA Models^[6] merupakan kelas-kelas dalam bahasa Java yang merepresentasikan Sistem Informasi Akademik UNPAR. Aplikasi akan dibuat dengan menggunakan Play Framework dan jsoup karena aplikasi didukung oleh 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		
4	AIF302	Penulisan Ilmiah	2. T: AIF202/292 Desain dan Analisis Algoritma		
5	AIF303	Pengantar Sistem Informasi	1. Lulus minimal 84 SKS		
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		
	AIF402	Skripsi 2	2. Lulus minimal 108 SKS 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		
2	AIF453	Kecerdasan Bisnis	1. T: AIF204/AIF 294 Manaj. Inf. & Basis Data 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		
5	AIF362	Sistem & Aplikasi Telematika	2. T: AIF315 Pemrograman Berbasis Web		
			1. T: AIF305 Jaringan Komputer		

Gambar 1.1: Prasyarat Mata Kuliah[1]



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

Untuk mendukung kebutuhan akademik mahasiswa Program Studi Teknik Informatika, fitur-fitur yang diperlukan akan dianalisis kemudian diimplementasikan ke dalam sebuah aplikasi yang akan dinamakan Informatika Student Portal. Informatika Student Portal merupakan aplikasi berbasis web yang dibuat menggunakan Play Framework. Selain itu, data-data yang akan diolah diambil langsung dari Portal Akademik Mahasiswa dengan *web scraping* menggunakan *library* jsoup. Untuk melakukan pengambilan data, jsoup harus mengetahui komunikasi dari Portal Akademik Mahasiswa. Analisis komunikasi Portal Akademik Mahasiswa akan dilakukan dengan menggunakan Chrome DevTools.

1.2 Rumusan Masalah

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

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

1.3 Tujuan

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

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

1.4 Batasan Masalah

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

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

1.5 Metode Penelitian

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

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.
2. Melakukan wawancara.
3. Menganalisis Portal Akademik Mahasiswa.
4. Merancang Informatika Student Portal.
5. Mengimplementasikan *web scraping* menggunakan *library jsoup*.
6. Melakukan eksperimen dan pengujian.

1.6 Sistematika Penulisan

Sistematika penulisan setiap bab pada skripsi ini adalah sebagai berikut:

1. Bab Pendahuluan

Bab 1 berisi latar belakang, rumusan masalah, tujuan, metode penelitian, dan sistematika penulisan yang digunakan untuk menyusun skripsi ini.

2. Bab Dasar Teori

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

3. Bab Analisis

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

4. Bab Perancangan

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

5. Bab Implementasi dan Pengujian

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

6. Bab Kesimpulan dan Saran

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

BAB 2

DASAR TEORI

Pada bab ini akan dijelaskan dasar-dasar teori mengenai *library* jsoup meliputi kelas Jsoup, Connection, Response, Document, Elements, dan Element. Selain itu akan dibahas pula mengenai CSS *selector*, Chrome DevTools meliputi panel Elements dan Network, Play Framework, dan SIA Models.

2.1 jsoup

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

Subbab-subbab berikut menjelaskan beberapa kelas dari jsoup.

2.1.1 Jsoup

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

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

Berfungsi untuk membuat koneksi baru dengan suatu situs web.

Parameter:

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

Kembalian: koneksi dengan situs web.

2.1.2 Connection

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

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

Berfungsi untuk menambahkan *cookie*.

Parameter:

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

Kembalian: koneksi yang sama tetapi sudah diubah.

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

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

Parameter:

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

Kembalian: koneksi yang sama tetapi sudah diubah.

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

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

Parameter:

- **method** metode pengiriman permintaan HTTP.

Kembalian: koneksi yang sama tetapi sudah diubah.

- **Connection timeout(int millis)**

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

Parameter:

- **millis** batas waktu dalam milidetik.

Kembalian: koneksi yang sama tetapi sudah diubah.

- **Connection validateTLCertificates(boolean value)**

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

Parameter:

- **value** status pemeriksaan sertifikat TLS.

Kembalian: koneksi yang sama tetapi sudah diubah.

- **Connection.Response execute()**

Berfungsi untuk mengirim permintaan HTTP.

Kembalian: objek Response.

2.1.3 Response

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

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

Method ini berfungsi untuk mendapatkan seluruh *cookies*.

Kembalian: seluruh *cookies*.

- **Document parse()**

Berfungsi untuk mengurai *body* jawaban menjadi dokumen.

Kembalian: koneksi yang sama tetapi sudah diubah.

- **String body()**

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

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

2.1.4 Document

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

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

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

Parameter:

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

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

2.1.5 Elements

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

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

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

Parameter:

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

Kembalian: elemen-elemen yang sudah diseleksi sesuai kueri.

- **public String val()**

Berfungsi untuk mendapatkan nilai dari elemen pertama.

Kembalian: nilai elemen.

- **public String text()**

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

Kembalian: seluruh teks dalam *string*.

2.1.6 Element

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

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

Berfungsi untuk mendapatkan anak elemen berdasarkan nomor indeks.

Parameter:

- **index** nomor index.

Kembalian: anak elemen.

- **public Element children()**

Berfungsi untuk mendapatkan seluruh anak elemen.

Kembalian: seluruh anak elemen.

- **public String className()**

Berfungsi untuk mendapatkan nama kelas elemen.

Kembalian: nama kelas elemen.

- **public String text()**

Berfungsi untuk mendapatkan teks dari elemen.

Kembalian: teks dalam *string*.

2.2 CSS Selector

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

```
h1{  
    color: gray;  
}
```

Gambar 2.1: Format Penulisan Definisi CSS

Beberapa jenis CSS *Selector* antara lain:

1. **Element selector**, memilih *tag* HTML.

Contoh: `h1`

Keterangan: *selector* mendefinisikan elemen h1.

2. **Grouping Selector**, memilih beberapa *selector* sekaligus. Setiap *selector* dipisahkan dengan `,`.

Contoh: `h1, h2, p`

Keterangan: *selector* mendefinisikan elemen h1, h2, dan p.

3. **Universal Selector**, memilih seluruh elemen. *Selector* ditampilkan sebagai “*”.

Contoh: *

Keterangan: *selector* mendefinisikan seluruh elemen.

4. **Class Selector**, memilih kelas elemen. *Selector* ditampilkan sebagai “.” kemudian diikuti nama kelas elemen.

Contoh: .top

Keterangan: *selector* mendefinisikan elemen dengan kelas “top”.

5. **ID Selector**, memilih ID elemen. *Selector* ditampilkan sebagai “#” kemudian diikuti ID elemen.

Contoh: #top

Keterangan: *selector* mendefinisikan elemen dengan ID “top”.

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

- **Simple Attribute**, memilih atribut elemen. *Selector* ditampilkan sebagai nama atribut kemudian diapit dengan kurung siku. Contoh: [name]

Keterangan: *selector* mendefinisikan elemen dengan atribut “name”.

- **Exact Value Attribute**, memilih atribut elemen dengan nilai tertentu. *Selector* ditampilkan sebagai definisi atribut kemudian diapit dengan kurung siku. Contoh: [name=Joe]

Keterangan: *selector* mendefinisikan elemen dengan atribut “name” yang memiliki nilai “Joe”.

7. **Descendant Selector**, memilih *child* elemen yang merupakan keturunan *parent* tertentu. *Selector* ditampilkan dengan mendefinisikan *parent* kemudian diikuti oleh *child* dipisahkan dengan spasi.

Contoh: p .top

Keterangan: *selector* mendefinisikan elemen dengan kelas “top” yang merupakan *child* dari elemen p.

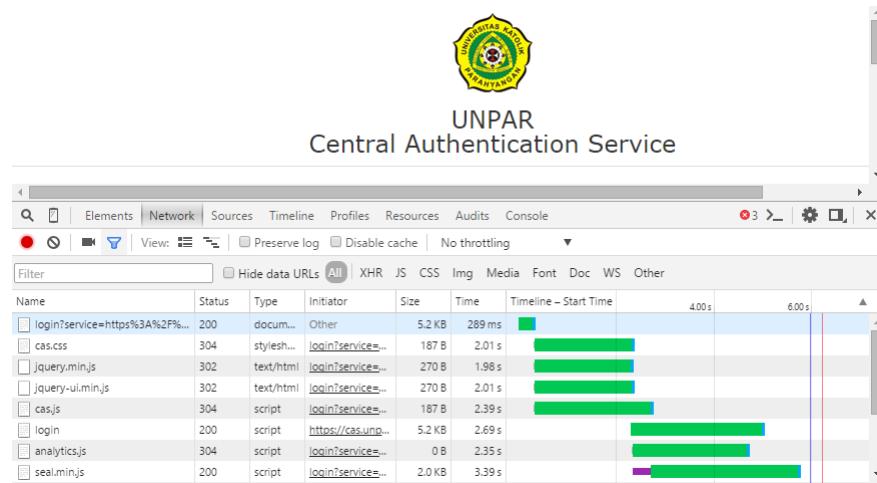
2.3 Chrome DevTools

Chrome Developer Tools (DevTools) adalah perangkat *debugging* yang dimiliki Google Chrome[9]. Saat menunjungi suatu halaman web, pengguna DevTools dapat melakukan *debugging* pada halaman tersebut. DevTools dapat diakses dengan menekan “Ctrl+Shift+I” saat sedang membuka suatu halaman web.

Panel-panel yang dimiliki DevTools (Gambar 2.2) antara lain:

1. **Elements**, memeriksa dan mengubah elemen HTML dan *style* dari suatu situs web.
2. **Console**, mendapatkan informasi pengembangan dan berinteraksi dengan dokumen.
3. **Sources**, melakukan *debugging* pada JavaScript dengan menentukan *breakpoint*.
4. **Network**, memantau aktivitas jaringan pada situs web secara *real-time*.

5. **Audits**, menganalisa halaman yang dimuat.
6. **Timeline**, menampilkan alur waktu saat memuat halaman.
7. **Profiles**, menggambarkan waktu eksekusi dan penggunaan memori saat memuat halaman.
8. **Resources**, memeriksa sumber daya halaman yang dapat berupa basis data, *cookies*, dan *cache*.



Gambar 2.2: Chrome DevTools

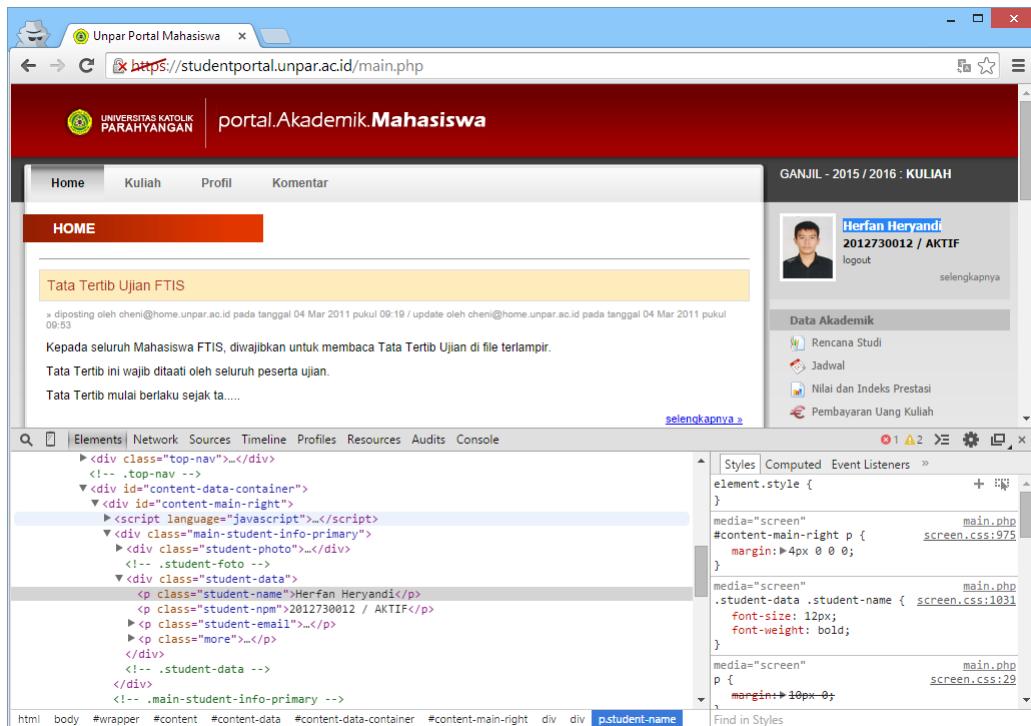
Pada subbab-subbab berikut akan dijelaskan mengenai dua panel dari DevTools.

2.3.1 Elements

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

Pemeriksaan elemen akan memperlihatkan node DOM dan CSS dari elemen yang dipilih pada *browser*. Pemeriksaan elemen dapat dilakukan dengan cara klik kanan pada elemen yang ingin diperiksa kemudian pilih “Inspect element”. Dengan melakukan pemeriksaan elemen, jendela panel Elements akan muncul. Sebagai contoh pada Gambar 2.3, saat melakukan “Inspect element” pada nama mahasiswa, panel Elements akan muncul dan menunjukkan pohon DOM dari halaman tersebut. Selain itu panel Elements juga menunjukkan CSS selector dari elemen tersebut yaitu `p.student-name`.

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

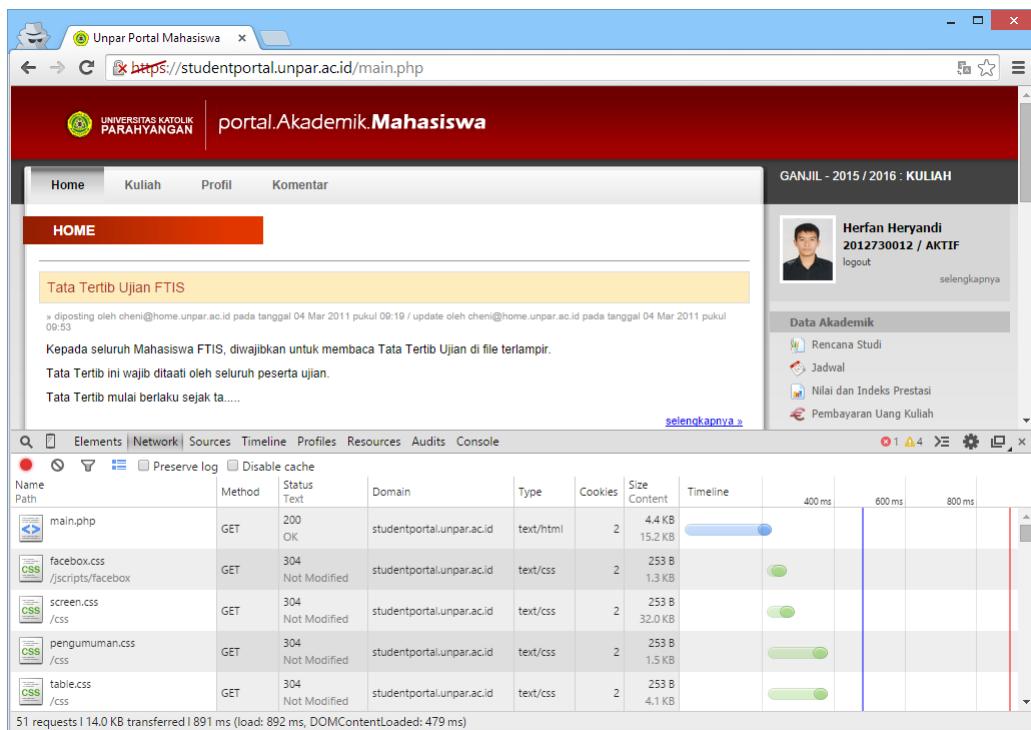


Gambar 2.3: Panel Elements

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.

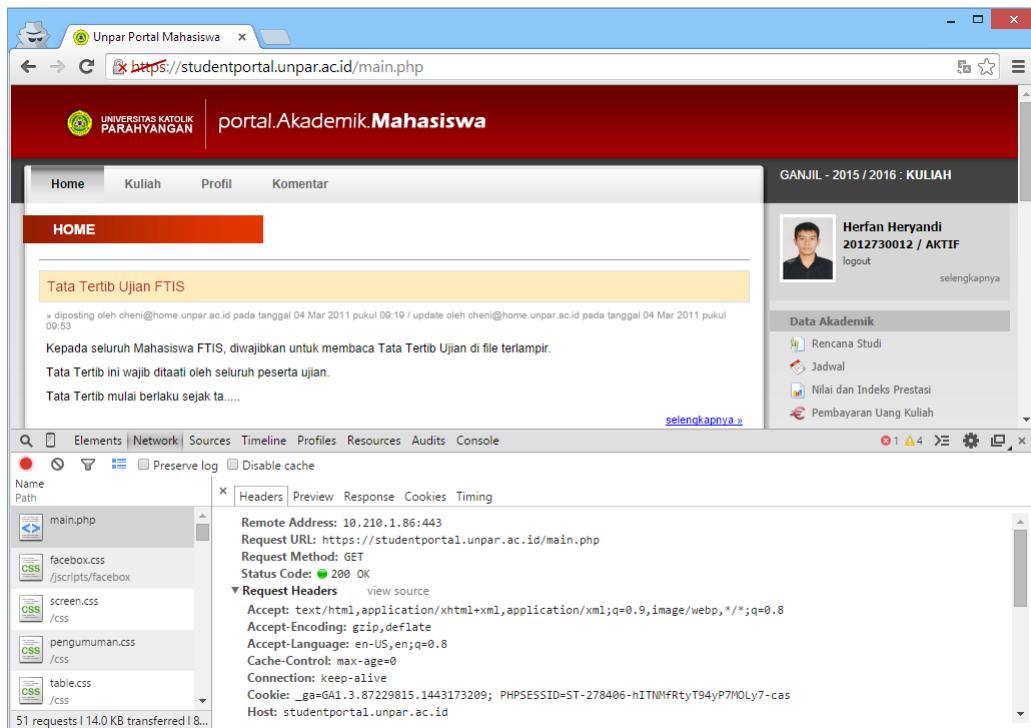


Gambar 2.4: Panel Network

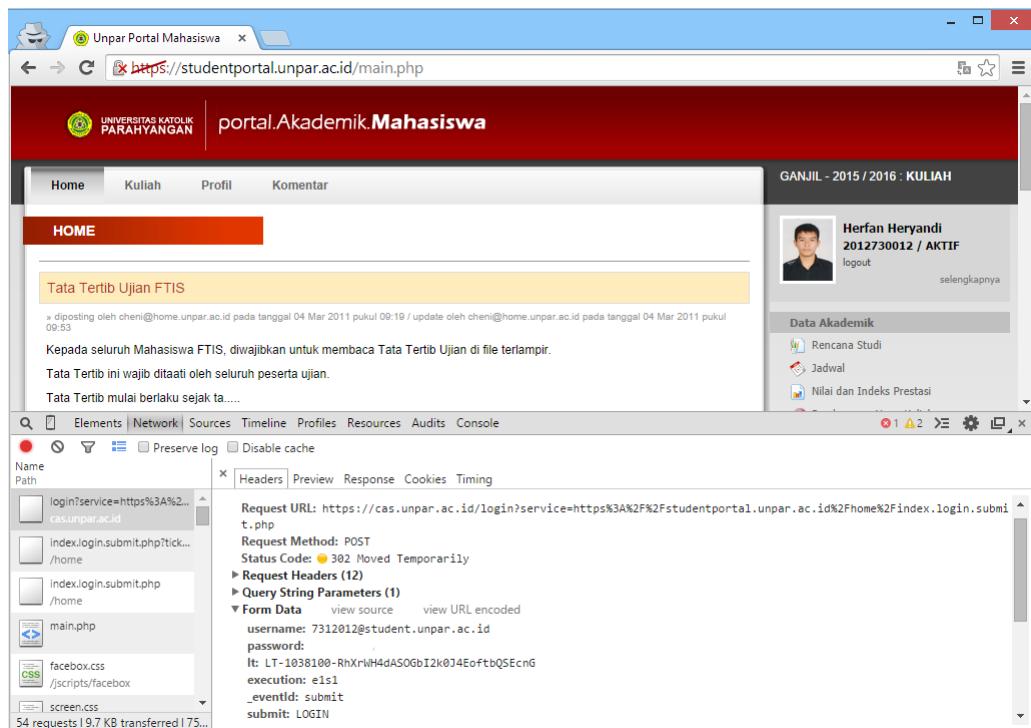
Ketika nama sumber daya dalam tabel Network diklik, maka akan muncul tautan baru yang berisi rincian tambahan sebagai berikut:

- **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, Gambar 2.5 menampilkan *header* pada metode permintaan GET sedangkan Gambar 2.6 menampilkan *header* pada metode permintaan POST.



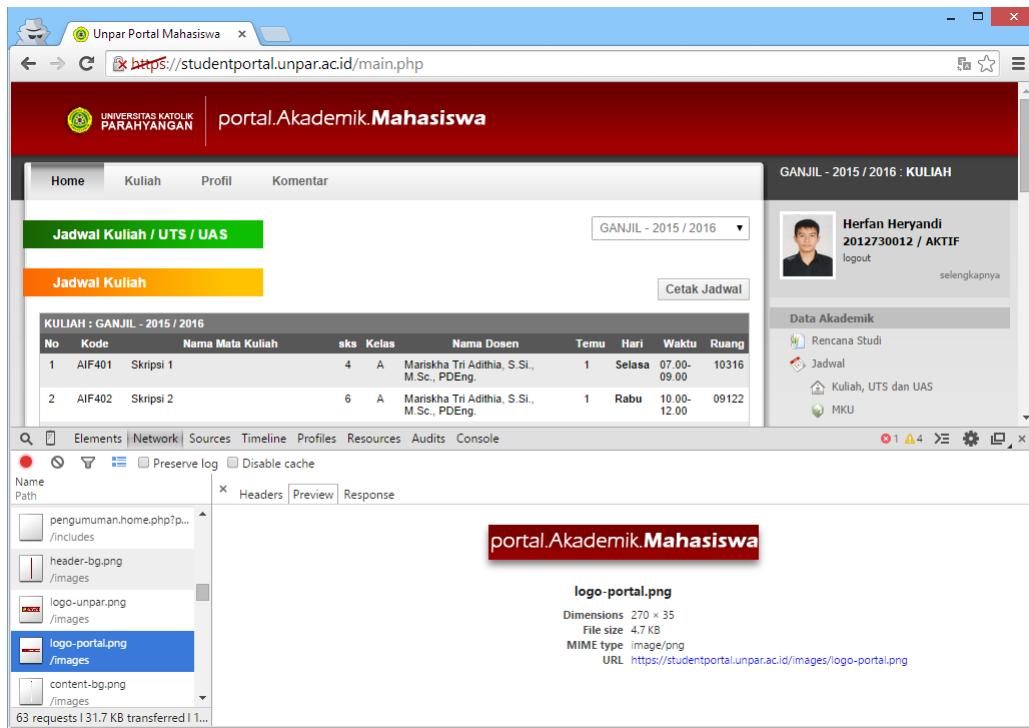
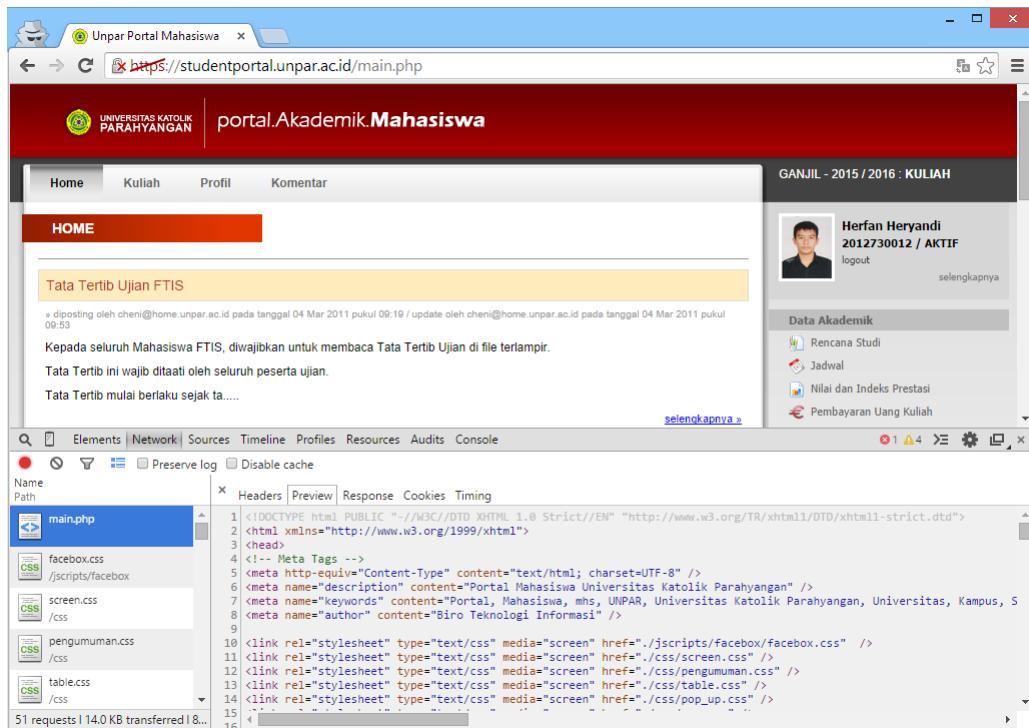
Gambar 2.5: Contoh Tautan Header pada Metode Permintaan GET



Gambar 2.6: Contoh Tautan Header pada Metode Permintaan POST

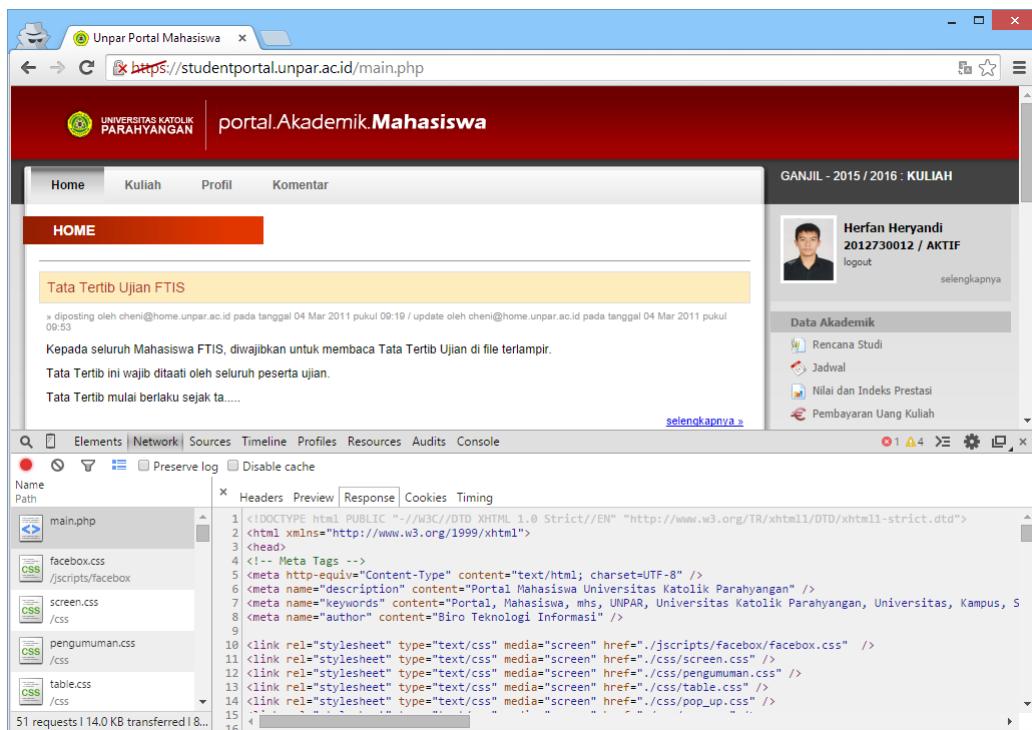
• Preview

Tautan Preview menampilkan *preview* sumber daya jika tersedia. Gambar 2.7 menampilkan *preview* yang tersedia pada sumber daya. Jika *preview* tidak tersedia maka 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`.

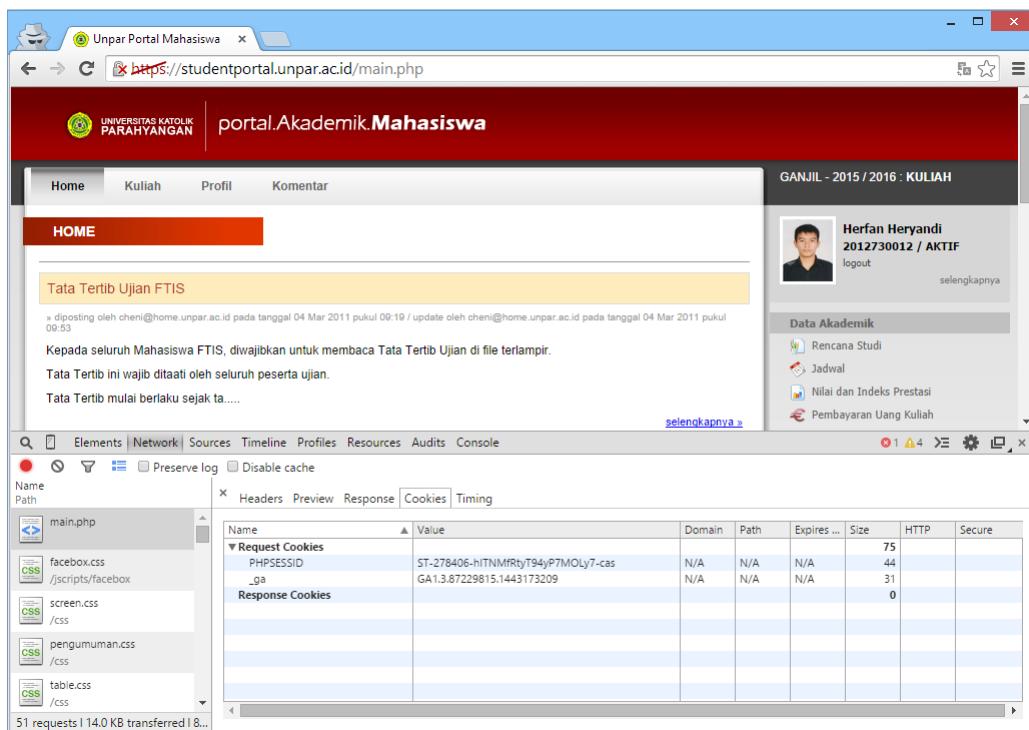


Gambar 2.9: Contoh Tautan Response

• Cookies

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

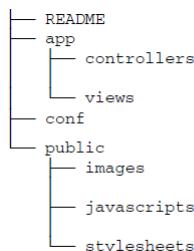
- **Name**, nama *cookie*
- **Value**, nilai *cookie*
- **Domain**, domain yang memiliki *cookie*
- **Path**, URL asal *cookie*
- **Expires/Max-Age**, batas akhir nilai *cookie*
- **Size**, ukuran *cookie* dalam byte
- **HTTP**, menunjukkan bahwa *cookie* harus ditetapkan oleh browser dalam permintaan 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

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

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*.

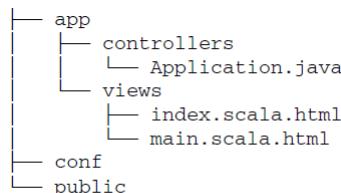


Gambar 2.12: Contoh Komponen Route[3]

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

```
1 public Result home() {
2     String nama = mahasiswa.getNama();
3     return ok(views.html.home.render(nama));
4 }
```

Method `home()` mengembalikan *view* `home` yang berada pada package `views` dengan mengirim parameter “`nama`”.

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

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

```
1 @(message: String)
2
3 <!DOCTYPE HTML>
4
5 <html lang='en'>
6   <head>
7     <title>Informatika Student Portal</title>
8   </head>
9   <body>
10     Nama: @message
11   </body>
12 </html>
```

Pada baris pertama “message” mendefinisikan nama parameter yang diterima dengan tipe *String*. Tipe yang diterima *view* harus sama dengan tipe yang dikirim *controller* begitu pula banyak parameternya. Baris ke-10 menampilkan “message” pada halaman HTML. Tanda “@” menandakan penggunaan bahasa Scala pada *view*.

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

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

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

Parameter:

- **key** kunci *session*.
- **value** nilai *session*.

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

- `public static String session(String key)`

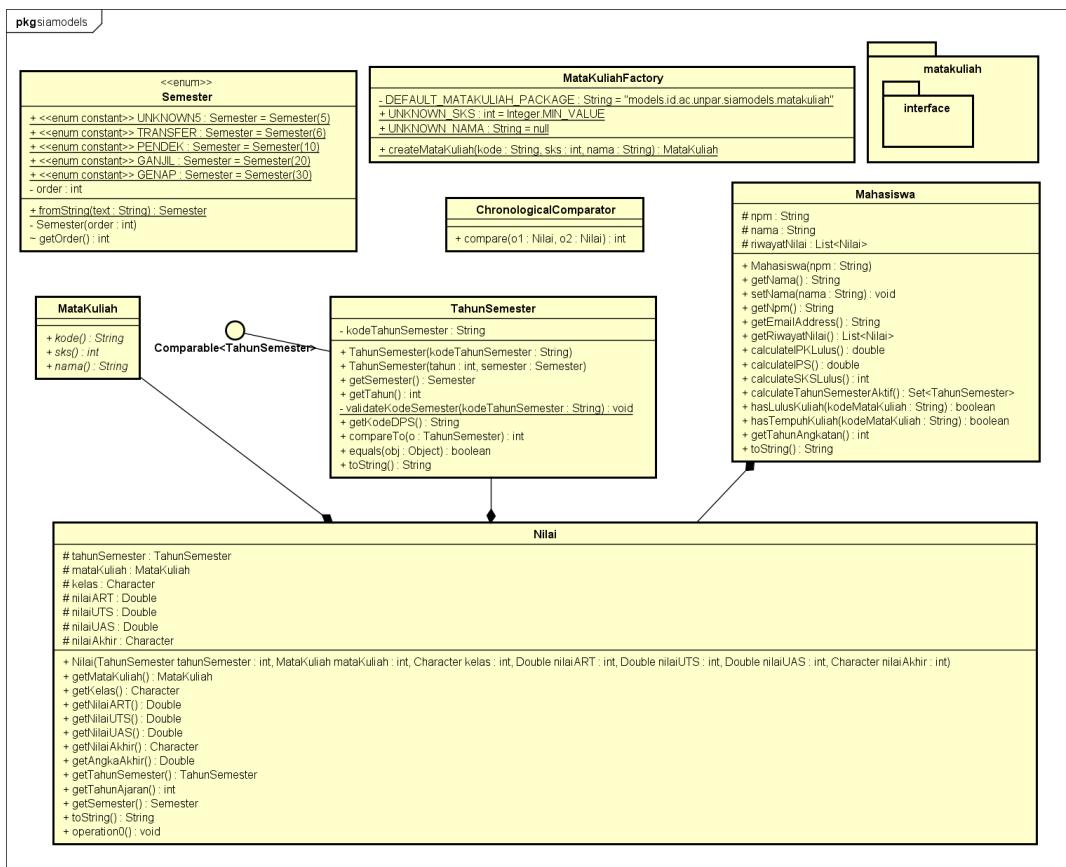
Parameter:

- **key** kunci *session*.

Kembalian: nilai *session*.

2.5 SIA Models

SIA Models merupakan kelas-kelas dalam bahasa Java yang merepresentasikan Sistem Informasi Akademik UNPAR[6]. Berdasarkan diagram kelas SIA Models (Gambar 2.14), kelas-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`

Package ini memiliki kelas-kelas sebagai berikut:

(a) Mahasiswa

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

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

Method-method yang dimiliki kelas ini adalah sebagai berikut:

- **public Mahasiswa(String npm)**

Merupakan *constructor* dari kelas Mahasiswa.

Parameter:

- **npm** nomor pokok mahasiswa.

- **public String getNama()**

Berfungsi untuk mendapatkan nama mahasiswa.

Kembalian: nama mahasiswa.

- **public void setNama(String nama)**

Berfungsi untuk mengubah nama mahasiswa.

Parameter:

– **nama** nama mahasiswa.

- **public String getNpm()**

Berfungsi untuk mendapatkan nomor pokok mahasiswa.

Kembalian: nomor pokok mahasiswa.

- **public String getEmailAddress()**

Berfungsi untuk mendapatkan *email* mahasiswa.

Kembalian: *email* mahasiswa.

- **public List<Nilai> getRiwayatNilai()**

Berfungsi untuk mendapatkan riwayat nilai mahasiswa.

Kembalian: riwayat nilai mahasiswa dalam List.

- **public double calculateIPKLulus()**

Menghitung IPK mahasiswa sampai saat ini, dengan aturan kuliah yang tidak lulus tidak dihitung dan jika pengambilan beberapa kali, diambil nilai terbaik. Sebelum memanggil *method* ini, *getRiwayatNilai()* harus sudah mengandung nilai per mata kuliah.

Kembalian: IPK lulus.

- **public double calculateIPS()**

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

Kembalian: nilai IPS sampai saat ini.

- **public int calculateSKSLulus()**

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

Kembalian: SKS lulus.

- **public Set<TahunSemester> calculateTahunSemesterAktif()**

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

Kembalian: kumpulan tahun semester di mana mahasiswa ini aktif.

- **public boolean hasLulusKuliah(String kodeMataKuliah)**

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

Parameter:

– **kodeMataKuliah** kode mata kuliah yang ingin diperiksa kelulusannya.

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

- **public boolean hasTempuhKuliah(String kodeMataKuliah)**

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

Parameter:

– **kodeMataKuliah** kode mata kuliah yang ingin diperiksa kelulusannya.

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

- **public int getTahunAngkatan()**

Mendapatkan tahun angkatan mahasiswa ini berdasarkan NPM-nya.

Kembalian: tahun angkatan.

(b) Nilai

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

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

Method-method yang dimiliki kelas ini adalah sebagai berikut:

- **public Nilai(TahunSemester tahunSemester, MataKuliah mataKuliah, Character kelas, Double nilaiART, Double nilaiUTS, Double nilaiUAS, Character nilaiAkhir)**

Merupakan *constructor* dari kelas Nilai.

Parameter:

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

- **public MataKuliah getMataKuliah()**

Mendapatkan mata kuliah yang diambil.

Kembalian: mata kuliah.

- **public Character getKelas()**

Mendapatkan kelas kuliah.

Kembalian: kelas kuliah.

- **public Double getNilaiART()**

Mendapatkan nilai ART.

Kembalian: nilai ART.

- **public Double getNilaiUTS()**

Mendapatkan nilai UTS.

Kembalian: nilai UTS.

- **public Double getNilaiUAS()**
Mendapatkan nilai UAS.
Kembalian: nilai UAS.
- **public Double getNilaikhir()**
Mendapatkan nilai akhir dalam bentuk angka.
Kembalian: nilai akhir dalam huruf atau **null** jika tidak ada.
- **public Double getAngkaAkhir()**
Mengembalikan nilai akhir dalam bentuk huruf (A, B, C, D, ...).
Kembalian: nilai akhir dalam angka, atau **null** jika **getNilaiAkhir()** mengembalikan **null**.
- **public int getTahunAjaran()**
Mendapatkan tahun ajaran saat pengambilan mata kuliah.
Kembalian: tahun ajaran saat pengambilan mata kuliah.
- **public TahunSemester getTahunSemester()**
Mendapatkan tahun dan semester pengambilan mata kuliah.
Kembalian: tahun dan semester pengambilan mata kuliah.
- **public Semester getSemester()**
Mendapatkan semester pengambilan mata kuliah.
Kembalian: semester pengambilan mata kuliah

(c) ChronologicalComparator

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

- **public int compare(Nilai o1, Nilai o2)**
Berfungsi untuk membandingkan nilai.
Parameter:
 - **o1** nilai pertama yang akan dibandingkan.
 - **o2** nilai kedua yang akan dibandingkan.**Kembalian:** hasil perbandingan.

(d) MataKuliah

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

- **public String kode()**
Mendapatkan kode mata kuliah sesuai dengan nama kelas mata kuliah tersebut.
Kembalian: kode mata kuliah.
- **public int sks()**
Mendapatkan bobot sks.
Kembalian: bobot SKS.
- **public String kode()**
Mendapatkan nama mata kuliah.
Kembalian: nama mata kuliah.

(e) MataKuliahFactory

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

- **String DEFAULT_MATAKULIAH_PACKAGE:** lokasi *package* untuk daf-
tar mata kuliah.
- **int UNKNOWN_SKS:** menandakan jumlah SKS tidak diketahui.
- **String UNKNOWN_NAMA:** menandakan nama mata kuliah tidak diketahui.

Method yang dimiliki kelas ini adalah sebagai berikut:

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

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

Parameter:

- **kode** kode mata kuliah.
- **sks** bobot SKS mata kuliah, dapat diisi dengan UNKNOWN_SKS jika tidak diketa-
hui.
- **nama** nama mata kuliah, dapat diisi dengan.

Kembalian: objek mata kuliah UNKNOWN_NAMA jika tidak diketahui.

(f) Semester

Kelas ini merepresentasikan semester *Method* yang dimiliki kelas ini adalah sebagai ber-
ikut:

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

Berfungsi untuk mengubah semester dari bentuk teks ke konstanta.

Parameter:

- **text** semester dalam bentuk teks (GANJIL, GENAP, PENDEK, TRANSFER,
dan UNKNOWN5).

Kembalian: konstanta semester.

(g) TahunSemester

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

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

Method-method yang dimiliki kelas ini adalah sebagai berikut:

- **public TahunSemester(String kodeTahunSemester)**

Method ini merupakan constructor dari kelas TahunSemester.

Parameter:

- **kodeTahunSemester** semester dalam bentuk teks (GANJIL, GENAP, PEN-
DEK, TRANSFER, dan UNKNOWN5).

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

Method ini merupakan constructor dari kelas TahunSemester.

Parameter:

- **tahun** tahun ajaran.
- **semester** semester dari tahun ajaran.

- **public Semester getSemester()**

Method ini berfungsi untuk mendapatkan semester.

Kembalian: semester dalam teks.

- **public int getTahun()**

Method ini berfungsi untuk mendapatkan tahun.

Kembalian: tahun ajaran.

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

Method ini berfungsi untuk melakukan validasi terhadap kode tahun semester.

Parameter:

- **kodeTahunSemester** kode tahun semester.

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

Package ini memiliki beberapa *interface* antara lain:

(a) HasPrasyarat

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

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

Memeriksa prasyarat-prasyarat dari kuliah, spesifik untuk mahasiswa yang dituju. Jika ada pesan-pesan khusus, akan ditambahkan pada parameter reasonsContainer.

Parameter:

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

Kembalian: true jika seluruh prasyarat dipenuhi, false jika tidak.

(b) Pilihan

Mendefinisikan kelas-kelas yang merupakan mata kuliah pilihan.

(c) PilihanWajib

Mendefinisikan kelas-kelas yang merupakan mata kuliah pilihan wajib.

(d) Wajib

Mendefinisikan kelas-kelas yang merupakan mata kuliah wajib.

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

Package ini berisi kelas-kelas yang merepresentasikan mata kuliah yang terdapat pada Program Studi Teknik Informatika UNPAR beserta aturan prasyaratnya. Rincian dari 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	-		

BAB 3

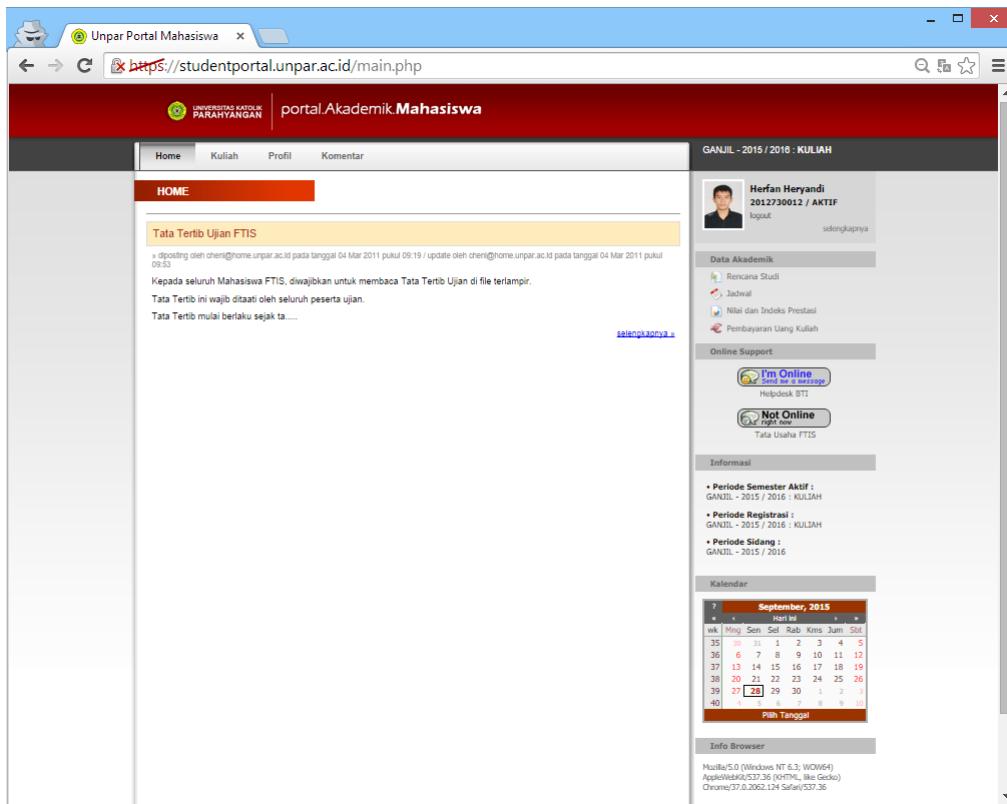
ANALISIS

Pada bab ini akan dijelaskan mengenai analisis Portal Akademik Mahasiswa, analisis kebutuhan Informatika Student Portal, analisis komunikasi Portal Akademik Mahasiswa meliputi kasus *login*, nilai, jadwal, dan *logout*, analisis arsitektur Informatika Student Portal, analisis *use case* beserta skenario dari masing-masing *use case*, dan analisis kelas.

3.1 Analisis Portal Akademik Mahasiswa

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/>. Untuk mengakses Portal Akademik Mahasiswa, mahasiswa harus *login* menggunakan akun email *student*. Halaman *login* Student Portal UNPAR terintegrasi dengan CAS (*Central Authentication Service*) UNPAR¹.

¹<https://cas.unpar.ac.id>



Gambar 3.1: Halaman Utama Portal Akademik Mahasiswa

Pada halaman utama Portal Akademik Mahasiswa (Gambar 3.1), terdapat beberapa bagian yaitu:

1. Menu Atas

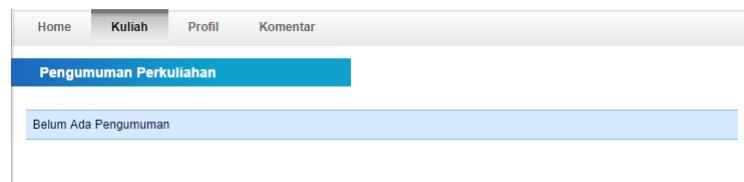
Menu ini berfungsi sebagai menu pendukung yang terdiri dari :

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



Gambar 3.2: Menu Atas Home

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



Gambar 3.3: Menu Atas Kuliah

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

NPM	2012730012	Fakultas	Teknologi Informasi dan Sains
Nama	HERFAN HERYANDI	Program Studi	Teknik Informatika
Nama (Ijazah)	HERFAN HERYANDI	Jenjang	-
Alamat	BUKIT SASTRA 95	Kode Pos	-
Status Alamat	Kost	No Telepon / HP	- / -
Kota Lahir	(0243) - Indramayu	Propinsi	Jawa Barat
Tgl Lahir	17 Oktober 1994	Kewarganegaraan	WNI
Jenis Kelamin	Laki Laki	Golongan Darah	O
Agama	Katolik	Status Perkawinan	BELUM MENIKAH
Sumber Biaya	-	Hub Penanggung Biaya	-

Nama Ayah	AMAT SUMARTO	Nama Ibu	KRISNATI
Nama Wali	-		
Alamat Orang Tua/Wali	JLN. VETERAN 41 INDRAMAYU	Kode Pos	45212
Kota	(0243) - Indramayu	Propinsi	Jawa Barat
No Telepon	0234-272990	Pekerjaan Ibu	Wiraswasta
Pekerjaan Ayah	Wiraswasta	Pendidikan Ibu	SMTA
Pendidikan Ayah	SMTA	Pendidikan Ibu	SMTP

Apabila terdapat data yang tidak cocok, silahkan hubungi Administrasi Fakultas ybs. untuk memperbarui data

Gambar 3.4: Menu Atas Profil

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

Info :
Silakan tulis komentar / saran / pesan untuk perbaikan portal ini

Silakan isi komentar Anda di sini

Kirim Kosongkan

Gambar 3.5: Menu Atas Komentar

2. Identitas Portal

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

3. Menu Utama

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



Gambar 3.7: Menu Utama

• Rencana Studi

Menu Rencana Studi terdiri dari submenu:

- Registrasi (FRS/PRS)

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

The screenshot shows the 'Registrasi Online : GANJIL - 2010 / 2011' interface. At the top, it displays student information: NIM 2008130095, Name Sharleen Kartika, Email 1308095@student.unpar.ac.id, and Advisor Samwel Wirawan. Below this, it shows academic details: IPK 3.98, IPS 4.00, and Batas 24 sks. The main area is titled 'FRS' and contains two tables: 'Berkelaruan 1' and 'Mata Kuliah Pilihan'. The first table lists four courses: EEP101-3 Pengantar Ekonomi Mikro, EMS101-3 Pengantar Bisnis, MKU001-2 Pendidikan Pancasila, and MKU002-2 Pendidikan Kewarganegaraan. The second table lists four optional courses: EAR203-0 Responsi Akuntansi Keuangan Lanjut I, EMS203-3 Perilaku Organisasi, EMS305-3 Statistika Non Parameterik, and EMS306-3 Kewirausahaan. At the bottom, there are summary rows for total credits (Total Sks 19), total courses (Total Mata Kuliah 8), and final totals (Total Sks Akhir 19 sks, Total Mata Kuliah Akhir 8 mata kuliah). Buttons for 'Simpan' and 'Kembali' are at the bottom right.

Gambar 3.8: Tampilan Registrasi FRS/PRS

- Kartu Rencana Studi

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

The screenshot shows the 'Kartu Rencana Studi : KULIAH GANJIL - 2015 / 2016' page. It lists six courses with their respective credits, classes, and SAPR values. The courses are: Skripsi 1 (4 sks, A), Skripsi 2 (6 sks, A), Kerja Praktek 2 (3 sks, A), Administrasi Jaringan Komputer 3 (3 sks, A), Kewirausahaan Berbasis Teknologi (3 sks, A), and Layanan Berbasis Web (3 sks, A). A summary row shows a total of 22 sks. Below the table, a note states: * Anda sudah pernah melakukan registrasi pada tanggal : 6 Juli 2015. On the right, there is a 'Cetak KRS' button. A legend at the bottom left defines abbreviations: S : Studio, A : Asistensi, P : Praktikum, R : Responsi.

Gambar 3.9: Tampilan Kartu Rencana Studi[4]

- Pindah Kelas MKU

Mahasiswa dapat memilih kelas yang masih tersedia di kolom Jadwal Baru dan menekan tombol “Simpan” untuk setiap kelas yang diubah (Gambar 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) Jadwal : Kamis, 11.00 - 13.00 Ruang : 10118 Dosen : Kapasitas : 50 Peserta :	<input type="button" value="Simpan"/>	
MKU002 - Pendidikan Kewarganegaraan	Kelas : A Jadwal : Senin, 8.00 - 10.00 Ruang : 2131 Dosen :	Kelas : Tidak Berubah Jadwal : - Ruang : - Dosen : - Kapasitas : - Peserta : -	<input type="button" value="Simpan"/>	

Gambar 3.10: Tampilan Pindah Kelas MKU[4]

- **Jadwal**

Menu Jadwal terdiri dari submenu:

- Kuliah, UTS, dan UAS

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	Mariskha Tri Adithia, S.Si., M.Sc., PDEng.	1	Selasa	07.00-09.00	10316
2	AIF402	Skripsi 2	6	A	Mariskha Tri Adithia, S.Si., M.Sc., PDEng.	1	Rabu	10.00-12.00	09122
3	AIF439	Kerja Praktek 2	3	A	-	1		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 sks					

Jadwal UTS

UTS : GANJIL - 2015 / 2016							
No	Kode	Nama Mata Kuliah	skls	Kelas	Tanggal Ujian	Waktu	Ruang
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

- MKU

_submenu ini menampilkan seluruh jadwal Mata Kuliah Umum (MKU) yang mem-

berikan informasi tentang kelas-kelas yang dibuka oleh Pusat Kajian 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
!	Jumat	08.00 - 09.40	10119	Andreas Doweng Bolo, S.S., M.Hum.
%	Jumat	10.00 - 11.40	10119	Bernardus Ario Tejo Sugiarto, 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 Djunatan
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

– Seluruh Fakultas

Fitur ini memberikan informasi mengenai jadwal-jadwal yang ada di seluruh 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	08.00-10.00	09122	
				A	Vania Natali, S.Kom.	2	Selasa	09.00-11.00	09018	
				A	Husnul Hakim, S.Kom., M.T.	3	Rabu	08.00-10.00	09018	
				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	08.00-10.00	09122	
				A	Vania Natali, S.Kom.	2	Selasa	09.00-11.00	09018	
				A	Husnul Hakim, S.Kom., M.T.	3	Rabu	08.00-10.00	09018	
				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	08.00-10.00	09122	
				A	Vania Natali, S.Kom.	2	Selasa	09.00-11.00	09018	
				A	Husnul Hakim, S.Kom., M.T.	3	Rabu	08.00-10.00	09018	
				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	08.00-10.00	09122	
				A	Vania Natali, S.Kom.	2	Selasa	09.00-11.00	09018	
				A	Husnul Hakim, S.Kom., M.T.	3	Rabu	08.00-10.00	09018	
				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	08.00-10.00	09121	
				B	Ferdian Reynaldi, S.T.	2	Selasa	09.00-11.00	09017	
				B	Vania Natali, S.Kom.	3	Rabu	08.00-10.00	09017	

Gambar 3.13: Tampilan Jadwal Seluruh Fakultas

• Nilai dan Indeks Prestasi

Menu Nilai dan Indeks Prestasi terdiri dari submenu:

- Riwayat per Semester

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

Daftar Nilai Semester									
GANJIL - 2015 / 2016									
No.	Kode MK	Nama Matakuliah	sks	Kelas	ART	UTS	UAS	AA	NA
1	AIF401	Skripsi 1 #	4	A	0	0	0	0	0
2	AIF402	Skripsi 2 #	6	A	0	0	0	0	0
3	AIF439	kerja Praktek 2 #	3	A	0	0	0	0	0
4	AIF441	Administrasi Jaringan Komputer 3 #	3	A	0	0	0	0	0
5	AIF457	Kewirausahaan Berbasis Teknologi #	3	A	0	0	0	0	0
6	AIF469	Layanan Berbasis Web #	3	A	0	0	0	0	0
Total sks					22				
Keterangan :									
# : Nilai tidak dapat dilihat karena status pembayaran belum lunas									
# : Nilai belum tersedia									
% : Nilai sedang dalam proses									

Gambar 3.14: Tampilan Riwayat Per Semester

- Daftar Perkembangan Studi

Seluruh riwayat mata kuliah dan nilai yang pernah ditempuh ditampilkan di submenu ini (Gambar 3.15). Pada bagian bawah halaman, terdapat statistik 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	AIF102-4	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 Digital	A	122
MKU001-2	Pendidikan Pancasila	B	121	AMS190-3	Matematika Informatika	B	122
MKU008-2	Eтика	B	121	MKU009-2	Bahasa Indonesia	A	141
MKU010-2	Bahasa Inggris	A	122	MKU011-2	Estetika	B	122
SEMESTER 3							
AIF201-4	Analisis & Desain Berorientasi Objek	A	131	AIF202-4	Desain & Analisis Algoritma	B	132
AIF205-4	Struktur Diskrit	A	131	AIF204-4	Manajemen Informasi & Basis Data	B	132
AIF206-3	Arsitektur & Organisasi Komputer	A	131	AIF208-4	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	Eтика 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 Informasi	D	142				
AIF318-2	Pemrograman Aplikasi Bergerak	A	142				
AIF342-3	Administrasi Jaringan Komputer 2	A	142				
AIF360-3	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				
AIF450-3	Pengolahan Citra	A	131				
APS302-2	Dunia Digital Dan Sains	A	142				

Gambar 3.15: Tampilan Daftar Perkembangan Studi

Kode Semester :
1 = Ganjil, 2 = Genap, 4 = Padat, 6 = Transfer

Ket. Mt. Bid. Peminatan :

01 = Teknologi Informatika Bisnis

02 = Ilmu Komputer

03 = Telematika

Ket. Mt Kuliah :
• [M] = Mk. Kendali Mutu,
• [X] = Mk. Disisihkan,
• [«] = Mk. yang diambil semester (2015-1)

Nilai Akhir	A	B	C	D	E	K
Jml. Mata Kuliah	23	20	1	1		
Jml. sks	67	56	3	2		
Jml. MK. 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
Dijinjakan 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.



Gambar 3.18: Tampilan TOEFL

• Pembayaran Uang Kuliah

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

Pembayaran Uang Kuliah		
NPM :	2012730012	Lihat Data Tagihan
No	Tagihan	Deskripsi
1	Rp. 0,-	

Keterangan

Pembayaran dapat dilakukan dengan cara melalui :

- I. Tunai di Bank Permata :
 1. Bentahueller bahwa Anda membayar biaya kuliah Semester Genap 2009 / 2010 di Unpar.
- II. Kartu ATM & Mesinnya Bank Permata
 1. Pilih menu: Transaksi Lainnya -> Pembayaran -> Pembayaran Lainnya -> Virtual Account
 2. Masukkan nomor Virtual Account (16 digit) dengan format 889855 + NPM
Contoh: NPM 2012730012 -> masukkan 8898552012730012
 3. Masukkan nilai uang sesuai dengan jumlah tagihan (tidak boleh kurang atau lebih)
- III. Kartu ATM Lain Mesin Bank Permata
 1. Pilih menu: Transaksi Lainnya -> Pembayaran -> Pembayaran Lainnya -> Virtual Account
 2. Masukkan nomor Virtual Account (16 digit) dengan format 889855 + NPM
Contoh: NPM 2012730012 -> masukkan 8898552012730012
 3. Masukkan nilai uang sesuai dengan jumlah tagihan (tidak boleh kurang atau lebih)
- IV. Kartu ATM & Mesinnya Bank Lain
 1. Pilih menu: Transfer -> Transfer antar Bank
 2. Masukkan kode bank Permata (013) dan nomor Virtual Account (16 digit) dengan format 013 + 889855 + NPM
Contoh: NPM 2012730012 -> masukkan 0138898552012730012
 3. Masukkan nilai uang sesuai dengan jumlah tagihan (tidak boleh kurang atau lebih)
- V. Transfer tunai dari Bank lainnya (*pada prinsipnya sama dengan transfer biasa*)
 1. Isi di kolom Rekening penerima : 889855 + NPM
Contoh: NPM 2012730012 -> masukkan 8898552012730012
 2. Isi di kolom Nama penerima : Nama Mahasiswa
Contoh: Nama untuk npm 2012730012 : Herfan Heryandi
 3. Isi di kolom Bank Penerima : Bank Permata - Cabang UNPAR,
Alamat : Jalan Ciambuleuit 94 Bandung
 4. Masukkan nilai uang sesuai dengan jumlah tagihan (tidak boleh kurang atau lebih)

Keterlambatan Pembayaran
Jika surpass dengan batas akhir pembayaran (baik tahap I maupun II) Mahasiswa belum melunasi tagihannya, maka akan terkena sanksi denda maupun akademik

Untuk mengakses melalui mobile devices (Handphone), silahkan mengakses m.unpar.ac.id
Untuk keterangan lebih lanjut silakan hubungi Biro Keuangan Unpar dengan telepon (022) 2032655 ext. 264

Gambar 3.19: Tampilan Pembayaran Uang Kuliah

4. Informasi

Bagian ini menampilkan informasi tentang periode-periode yang sedang aktif (Gambar 3.20). Sebagai contoh jika “Periode Registrasi” diklik maka akan muncul *pop up* seperti 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 Prod1	: 6 Juli 2015 s/d 10 Juli 2015
• Tgl FRS Dosen Wali	: 6 Juli 2015 s/d 10 Juli 2015
• Tgl PRS Prod1	: 1 September 2015 s/d 2 September 2015
• Tgl PRS 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 Prs 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	: - s/d -
• Tgl Aju Sidang	: 18 Agustus 2015 s/d 22 Januari 2016

Gambar 3.21: Tampilan Pop Up Informasi

5. Kalender

Bagian ini menampilkan kalender masehi (Gambar 3.22).



Gambar 3.22: Tampilan Kalender

6. Info Browser

Bagian ini menampilkan informasi tentang internet *browser* yang digunakan pada saat 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/lionov²>, 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 perkuliahan.

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.

No	Kode	Nama Mata Kuliah	sks	Kelas	Nama Dosen	Temu	Hari	Waktu	Ruang
1	AIF401	Skripsi 1	4	A	Mariska Tri Adithia, S.Si., M.Sc., PDEng.	1	Selasa	07.00-09.00	10316
2	AIF402	Skripsi 2	6	A	Mariska Tri Adithia, S.Si., M.Sc., PDEng.	1	Rabu	10.00-12.00	09122
3	AIF439	Kerja 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	sks					

Gambar 3.24: Tampilan Jadwal pada Portal Akademik Mahasiswa

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

	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

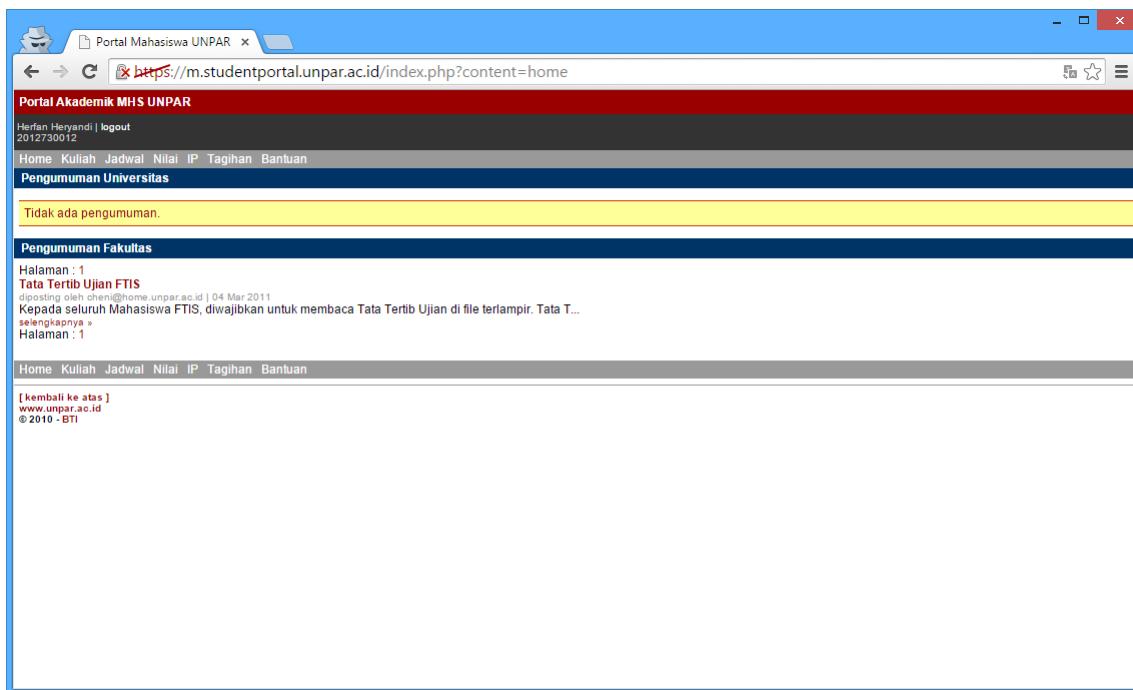
Tagihan pada Portal Akademik Mahasiswa tidak mencantumkan batas akhir pembayaran dan rincian nominal tagihan. Mahasiswa menginginkan rincian pembayaran agar tidak terlambat membayar uang kuliah dan dapat mengetahui rincian nominal tagihan.

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 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.

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

- Data yang dibutuhkan dapat diambil dari Portal Akademik Mahasiswa

Aplikasi Informatika Student Portal merupakan aplikasi hasil kustomisasi Portal Akademik Mahasiswa. Informasi yang ditampilkan Informatika Student Portal diperoleh dengan cara mengolah data dari Portal Akademik Mahasiswa. Jadi data yang dibutuhkan Informatika Student Portal harus dapat diambil dari Portal Akademik Mahasiswa.

- Data hasil olahan tidak tersedia di Portal Akademik Mahasiswa

Jika data hasil olahan yang sudah tersedia di Portal Akademik Mahasiswa, maka fitur tidak

perlu dibuat lagi di Informatika Student Portal karena mahasiswa dapat mengakses langsung Portal Akademik Mahasiswa.

- Fitur mendukung fungsi Portal Akademik Mahasiswa sebagai sumber informasi akademik Portal Akademik Mahasiswa berfungsi sebagai sumber informasi akademik karena itu Informatika Student Portal juga harus mememiliki fitur yang dapat mendukung fungsi tersebut.

Hasil analisis fitur-fitur yang diinginkan berdasarkan kriteria di atas dan batas waktu 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 penggeraan yang terbatas
Unggah <i>Curriculum Vitae</i>	Tidak dibuat	Tidak mendukung Portal Akademik Mahasiswa sebagai sumber informasi akademik

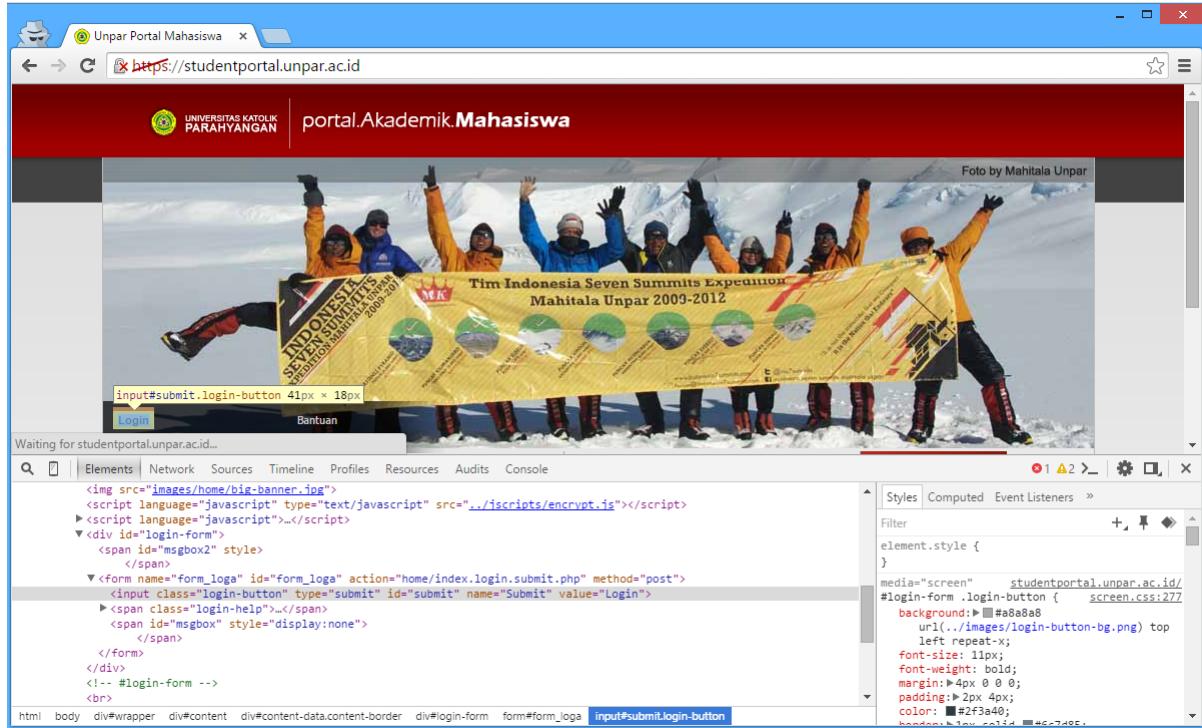
3.3 Analisis Komunikasi Portal Akademik Mahasiswa untuk Fitur Informatika Student Portal

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*

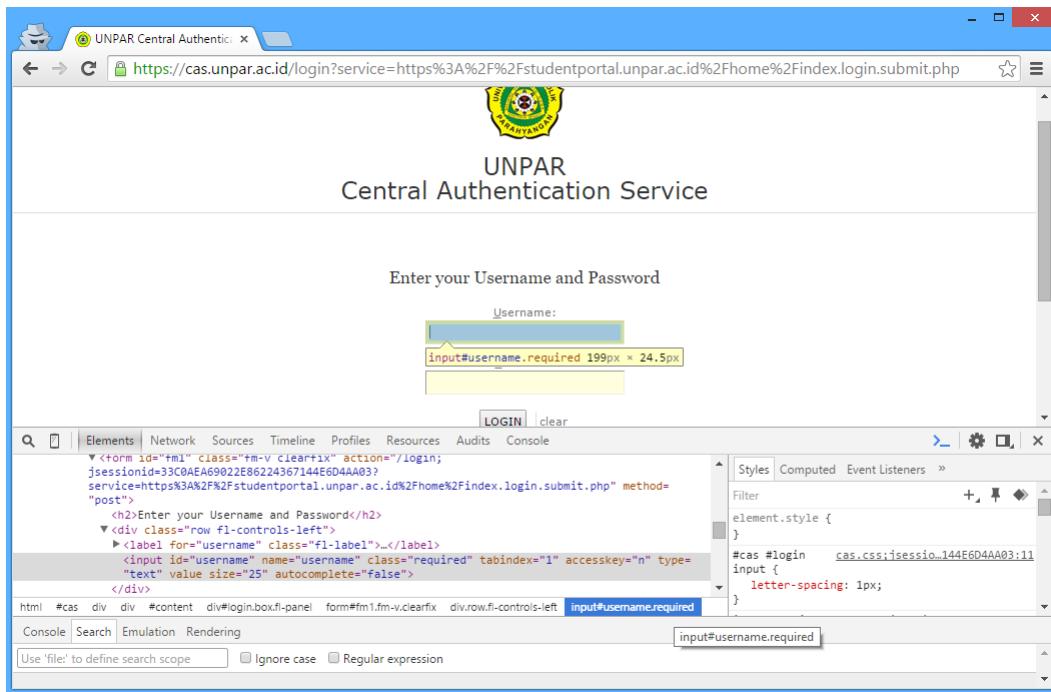
Di Portal Akademik Mahasiswa, mahasiswa dapat *login* dengan cara:

- 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).



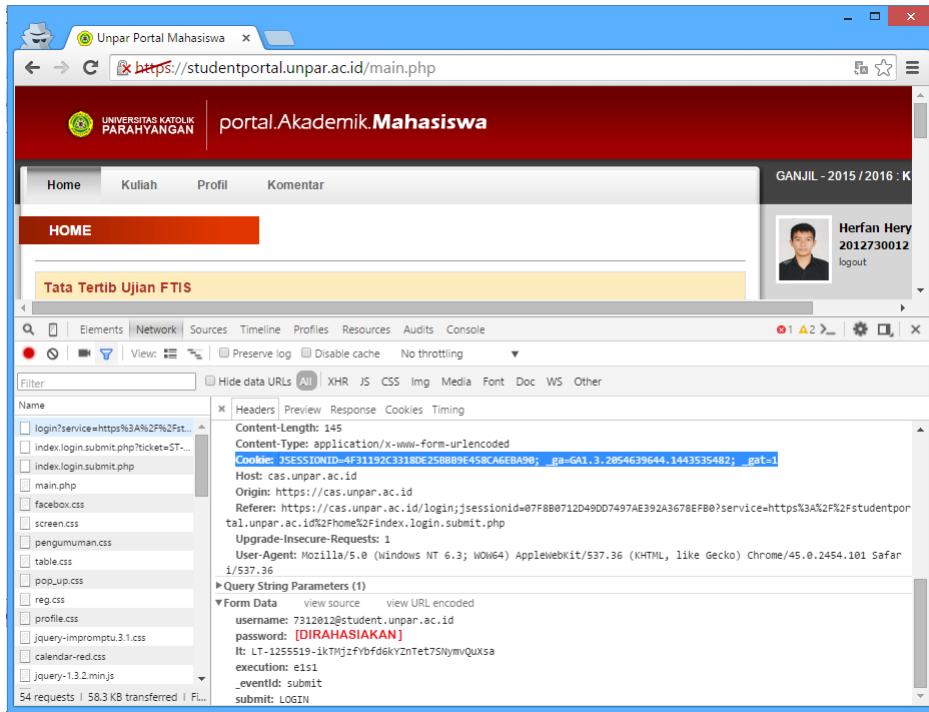
Gambar 3.28: Kolom “Username” “input#username.required” pada Halaman CAS UNPAR

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

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

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

- username: diambil dari nilai elemen “input#username.required”
- password: diambil dari nilai elemen “input#password.required”
- lt: diambil dari nilai elemen “input” dengan nama “lt”
- execution: diambil dari nilai elemen “input” dengan nama “execution”
- _eventId: selalu berisi “submit”
- submit: selalu berisi “LOGIN”

Gambar 3.29: *Form Data* yang dikirim CAS UNPAR

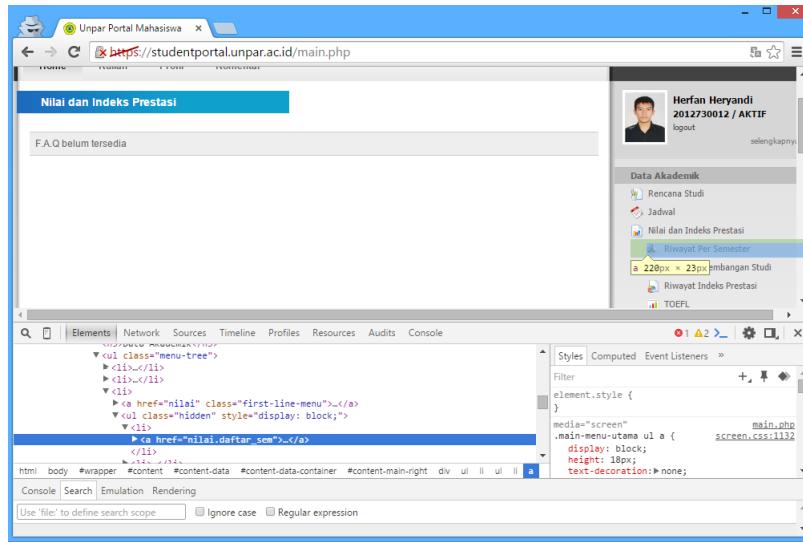
6. Jika berhasil, akan dilakukan pengalihan beberapa kali dan diakhiri di <https://studentportal.unpar.ac.id/main.php> dengan *cookie* sebagai berikut:
 - PHPSESSID: diambil dari *cookie* yang di-set pada beberapa pengalihan sebelumnya
 - _ga dan _gat: tidak diperlukan, digunakan oleh Google Analytics³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

3.3.2 Kasus Nilai

Di halaman utama Portal Akademik Mahasiswa, mahasiswa dapat melihat riwayat nilai dengan cara:

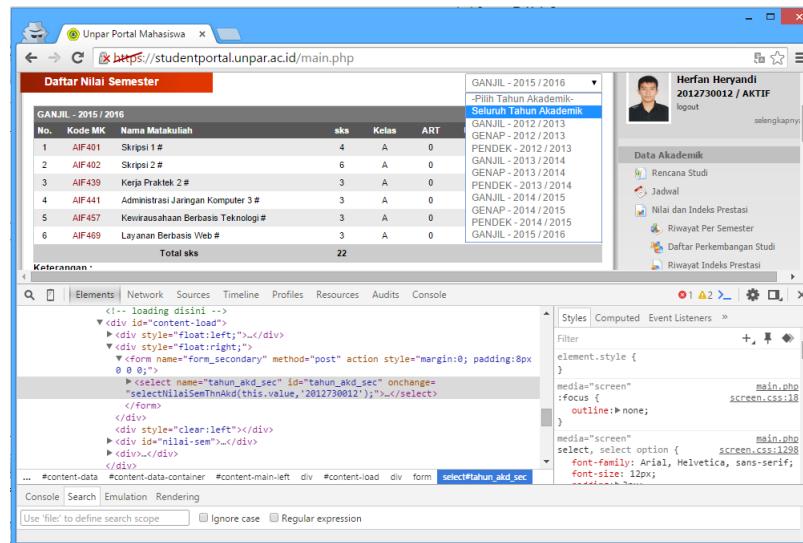
1. Mengklik “a.first-line-menu” dengan teks “Nilai dan Indeks Prestasi”
2. Setelah diklik, akan muncul list “ul.hidden”, kemudian mahasiswa harus mengklik elemen “a” dengan teks “Riwayat Per Semester” (Gambar 3.30)

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

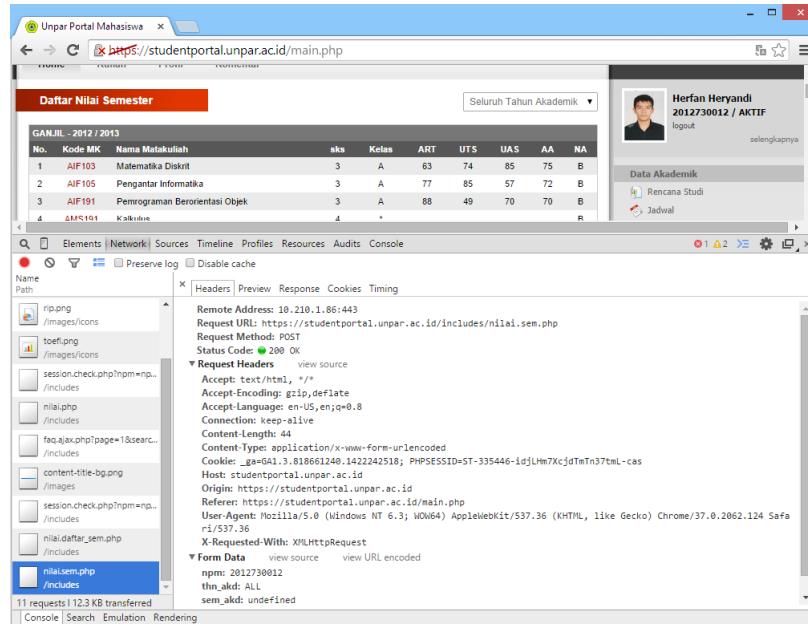
3. Setelah mengklik “Riwayat Per Semester”, mahasiswa akan diarahkan ke <https://studentportal.unpar.ac.id/includes/nilai.daftar.sem.php> dengan *cookie* sebagai berikut:
 - PHPSESSID: diambil dari *cookie* yang di-set saat pengalihan ke halaman utama
4. Halaman “Riwayat Per Semester” menampilkan nilai 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).



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

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*:

- npm: diperoleh dari NPM mahasiswa
- thn_akd: berisi tahun akademik semester atau berisi “ALL” jika memilih “Seluruh Tahun Akademik”
- sem_akd: berisi semester akademik dalam angka (1: ganjil, 2: genap, 4: pendek) atau tidak didefinisikan jika memilih “Seluruh Tahun Akademik”

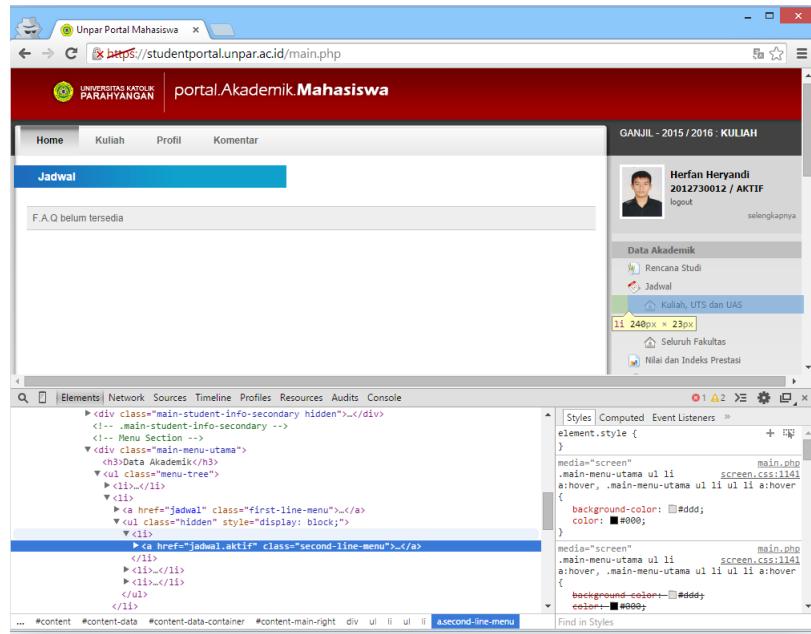


Gambar 3.32: *Form Data* pada pengiriman Nilai Seluruh Tahun Akademik

3.3.3 Kasus Jadwal

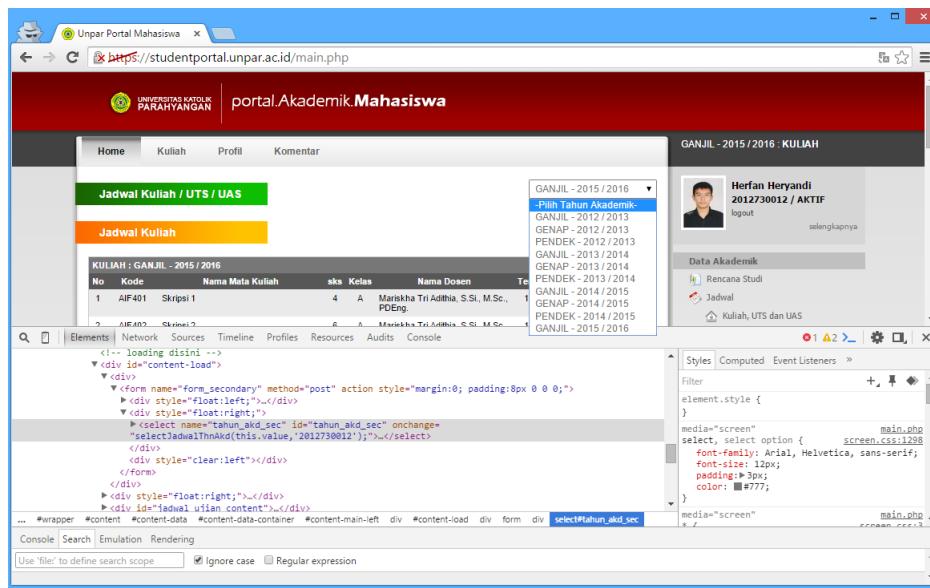
Di halaman utama Portal Akademik Mahasiswa, mahasiswa dapat melihat jadwal dengan cara:

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

3. Setelah mengklik “Kuliah, UTS dan UAS”, mahasiswa akan diarahkan ke <https://studentportal.unpar.ac.id/includes/jadwal.aktif.php> dengan *cookie* sebagai berikut:
 - PHPSESSID: diambil dari *cookie* yang di-set saat pengalihan ke halaman utama
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).

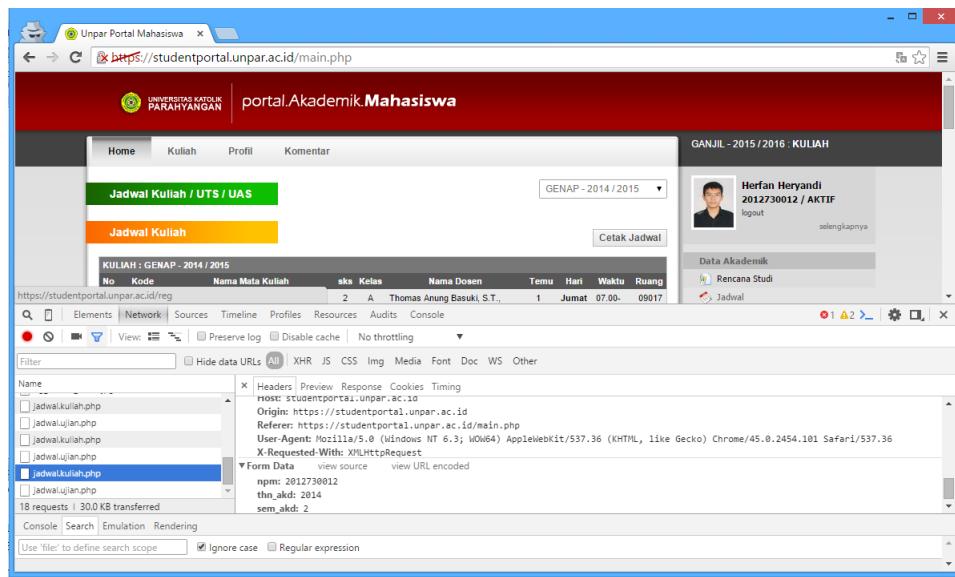


Gambar 3.34: Combo Box “select#tahun_akd_sec” pada Halaman Jadwal Kuliah, UTS, dan UAS

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:

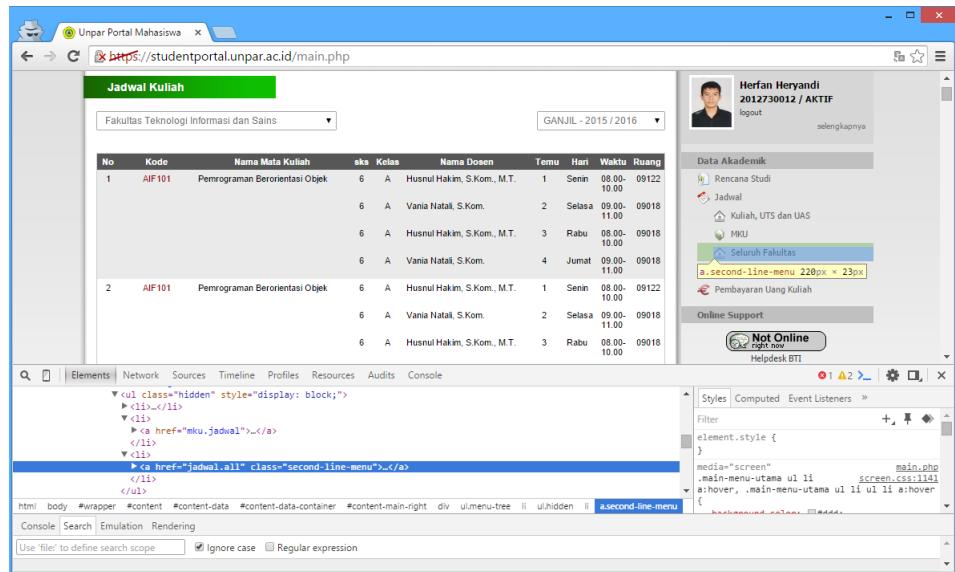
- npm: diperoleh dari NPM mahasiswa
- thn_akd: berisi tahun akademik semester
- sem_akd: berisi semester akademik dalam angka (1: ganjil, 2: genap, 4: pendek)



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

Mahasiswa juga dapat melihat jadwal seluruh fakultas dengan cara:

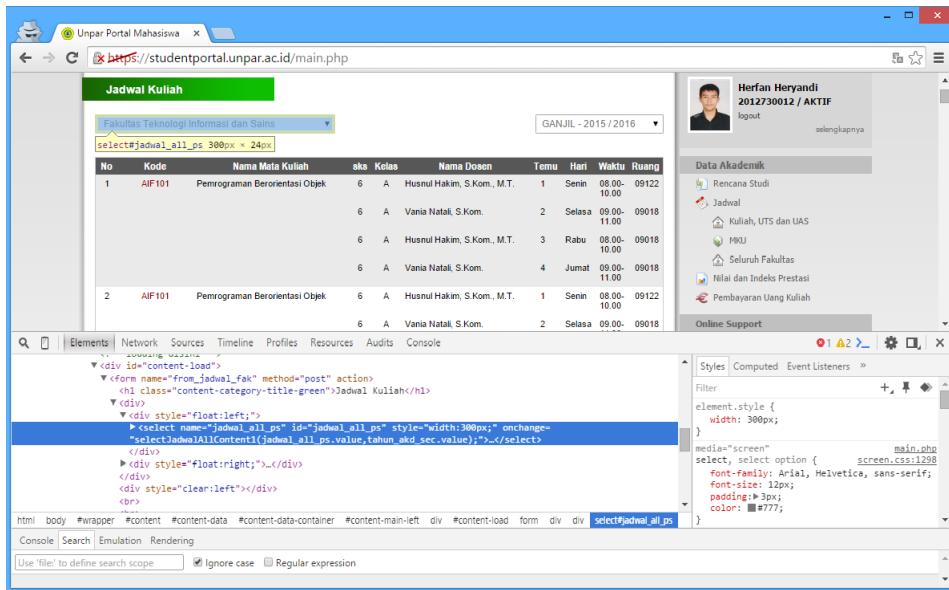
1. Mengklik “a.second-line-menu” dengan teks “Seluruh Fakultas” pada list “ul.hidden” (Gambar 3.36)



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

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

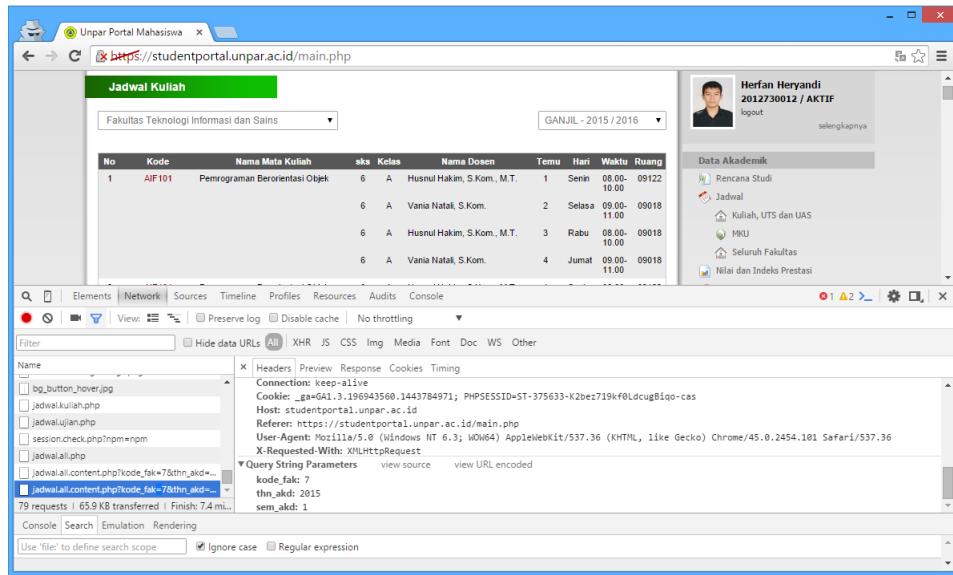
- PHPSESSID: diambil dari *cookie* yang di-set saat pengalihan ke halaman utama
3. Halaman “Seluruh Fakultas” menampilkan seluruh jadwal semester terkini pada fakultas tempat mahasiswa menempuh studi. Jika ingin melihat jadwal semester sebelumnya, mahasiswa dapat mengklik *combo box* “select#tahun_akd_sec”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

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 menggunakan *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)

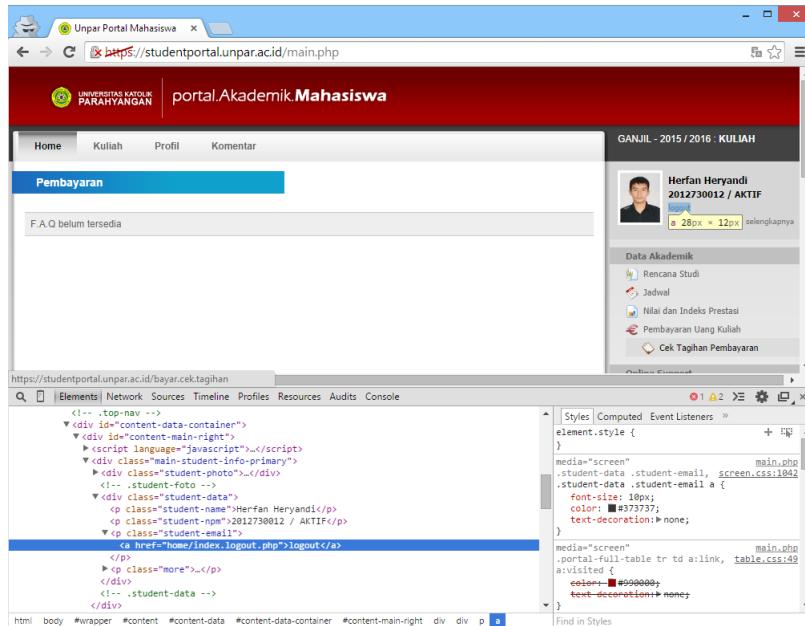


Gambar 3.38: *Form Data* pada pengiriman Jadwal Seluruh Fakultas

3.3.4 Kasus *Logout*

Mahasiswa dapat melakukan *logout* dengan cara:

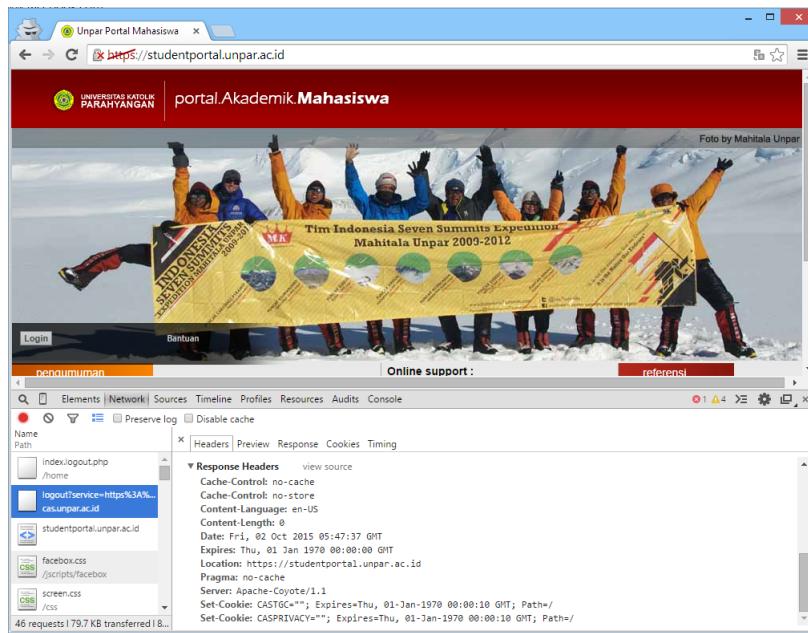
1. Mengklik “a” dengan teks “logout” pada bagian identitas portal (Gambar 3.39)



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

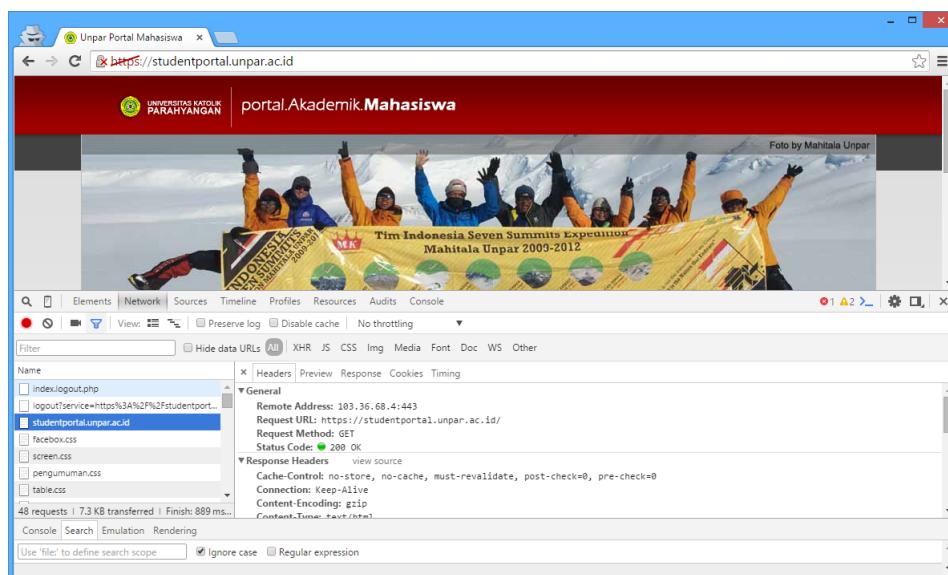
2. Setelah diklik, mahasiswa akan diarahkan ke <https://studentportal.unpar.ac.id/home/index.logout.php> dengan *cookie* sebagai berikut:
 - PHPSESSID: diambil dari *cookie* yang di-set saat pengalihan ke halaman utama
3. Kemudian dilakukan pengalihan ke <https://cas.unpar.ac.id/logout?service=https%3A%2F%2Fstudentportal.unpar.ac.id%2Fhome%2Findex.logout.php>

`2F%2Fstudentportal.unpar.ac.id` lalu mengadaluarsakan cookie CASTGC dan CASPRI-VACY yang sebelumnya di-set saat pengalihan ke halaman utama (Gambar 3.40)



Gambar 3.40: Pengadaluarsaan Cookie CASTGC dan CASPRIVACY

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

Hasil analisis komunikasi Portal Akademik Mahasiswa akan digunakan untuk membangun koneksi yang akan diimplementasikan menggunakan jsoup. Berdasarkan hasil analisis tersebut, terdapat tiga hal yang perlu diperhatikan antara lain:

- Perpindahan halaman

Saat pengguna melakukan suatu aksi seperti klik dan memilih *option*, Portal Akademik Mahasiswa akan mengarahkan pengguna ke suatu halaman. Alamat dari halaman tersebut perlu diketahui untuk koneksi jsoup. Metode pengirimannya juga perlu diketahui. Dari seluruh halaman yang dianalisis, metode pengirimannya adalah GET kecuali halaman *login* CAS UNPAR yang menggunakan metode POST.

- *Cookie*

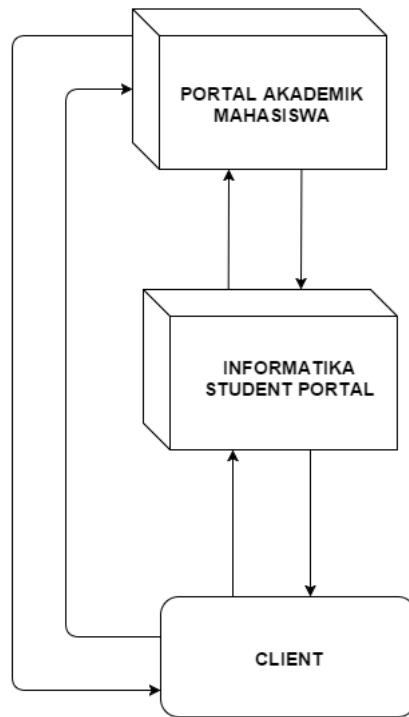
Hal yang harus diperhatikan dalam penggunaan *cookie* yaitu kapan suatu *cookie* di-set dan kapan *cookie* tersebut dikirimkan. Jika suatu halaman memerlukan *cookie* saat membangun koneksi, maka perlu diketahui kapan *cookie* tersebut di-set sehingga *cookie* tersebut bisa diperoleh menggunakan jsoup.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

- Parameter

Saat berpindah halaman, terkadang halaman tersebut membutuhkan parameter data. Dalam membangun koneksi, jsoup memerlukan juga memerlukan parameter data, karena itu perlu diketahui parameter data apa saja yang harus dilewatkan saat menuju suatu halaman.

3.4 Analisis Arsitektur Informatika Student Portal

Arsitektur Informatika Student Portal dapat dilihat pada Gambar 3.42. *Client* dapat mengakses Portal Akademik Mahasiswa dan Informatika Student Portal untuk mendapatkan informasi akademik. Namun saat mengakses Informatika Student Portal, *client* akan diberikan data akademik dengan fitur-fitur yang berbeda. Saat *client* mengakses Informatika Student Portal, Informatika Student Portal akan mengambil data dari Portal Akademik Mahasiswa. Data yang telah diambil akan diolah kemudian ditampilkan kepada *client*.

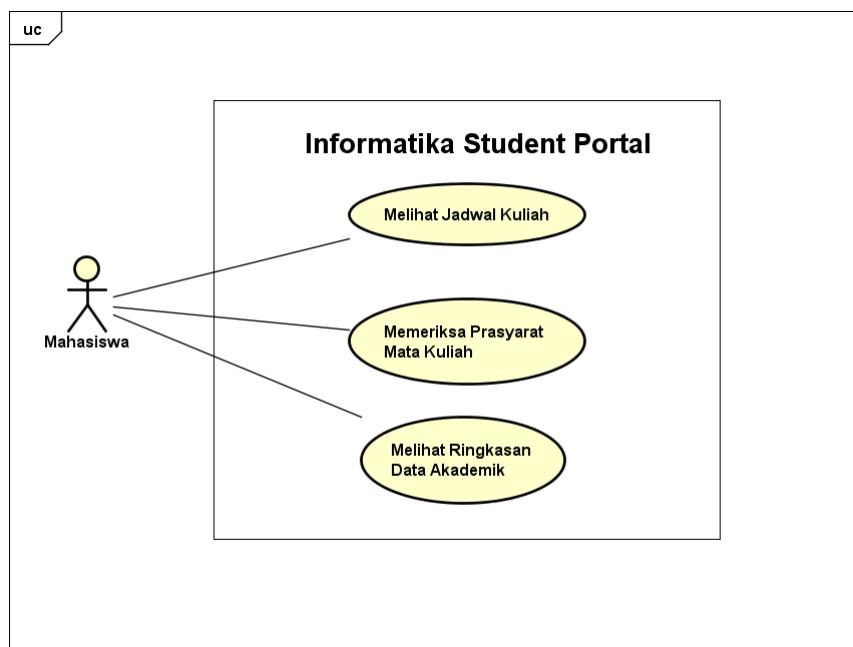


Gambar 3.42: Arsitektur Informatika Student Portal

3.5 Analisis *Use Case*

3.5.1 Diagram *Use Case*

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



Gambar 3.43: Diagram *Use Case* Informatika Student Portal

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

- **Memeriksa Prasyarat Mata Kuliah**, mahasiswa dapat memeriksa mata kuliah yang dibuka pada semester terkini apakah memenuhi prasyarat atau tidak.
- **Melihat Jadwal Kuliah**, mahasiswa dapat melihat jadwal kuliah yang sudah tersusun dan terurut berdasarkan hari.
- **Melihat Ringkasan Data Akademik**, mahasiswa dapat 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.

3.5.2 Skenario Use Case

1. Memeriksa Prasyarat Mata Kuliah

- Nama: Memeriksa prasyarat mata kuliah
- Aktor: Mahasiswa
- Deskripsi: Memeriksa prasyarat mata kuliah yang dibuka pada semester terkini
- Kondisi awal: Mahasiswa telah *login*
- Kondisi akhir: Halaman prasyarat mata kuliah ditampilkan dan berisi mata kuliah yang dibuka pada semester terkini beserta status prasyaratnya
- Skenario utama:

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

- Eksepsi: Mahasiswa sedang menempuh semester 1

2. Melihat Jadwal Kuliah

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

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

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

3. Melihat Ringkasan Data Akademik

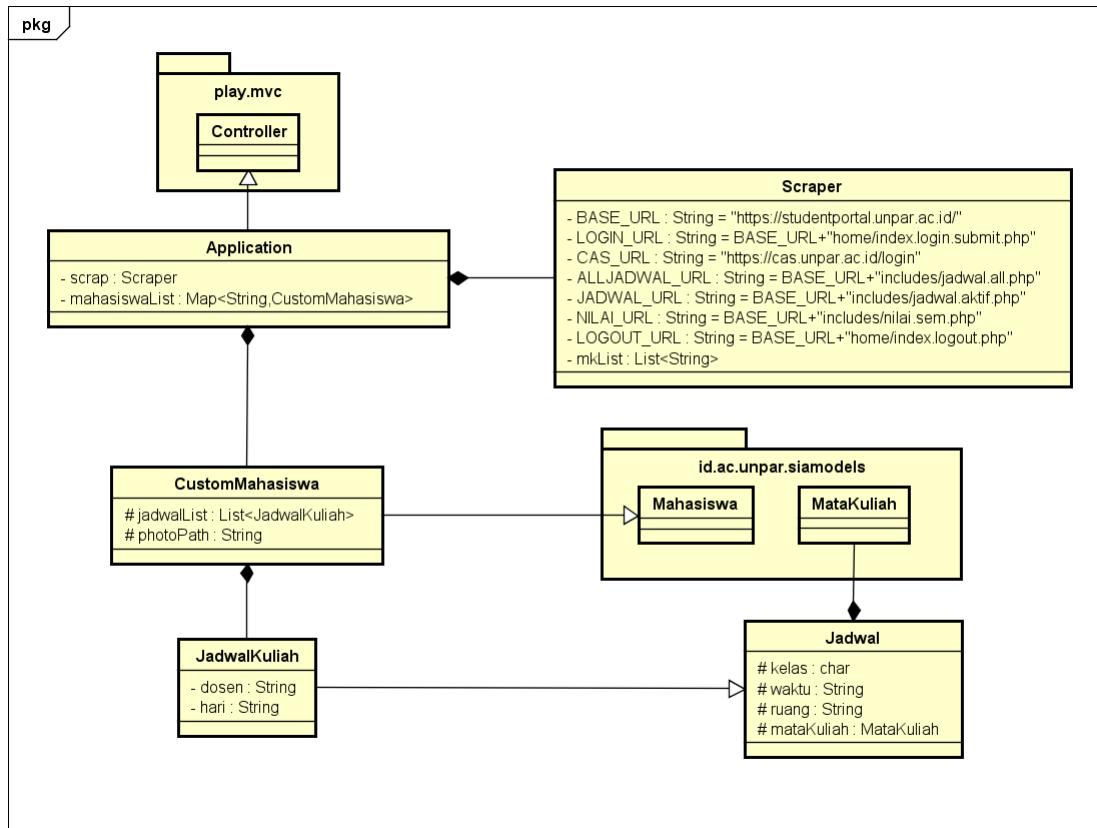
- Nama: Melihat ringkasan data akademik
- Aktor: Mahasiswa
- Deskripsi: melihat data mengenai mata kuliah apa saja yang sudah lulus beserta jenis mata kuliahnya(wajib, pilihan, atau pilihan wajib), sisa SKS untuk mencapai kelulusan, dan mata kuliah wajib yang belum ditempuh. Mahasiswa juga dapat melihat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

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

- 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. Scrapper

Kelas ini merupakan kelas yang mengimplementasikan *web scraping* menggunakan *library json-up*. 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.

5. Jadwal

Kelas ini merepresentasikan jadwal di UNPAR.

6. Jadwal Kuliah

Kelas ini merupakan turunan dari kelas Jadwal yang merepresentasikan jadwal kuliah di UN-PAR.

BAB 4

PERANCANGAN

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

4.1 Diagram Kelas Rinci

Diagram kelas rinci diperoleh dari hasil pengembangan diagram kelas analisis pada subbab 3.6. Diagram kelas rinci dapat dilihat pada Gambar 4.1. 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:

- **Scaper scrap:** objek Scaper 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:

- **public Result index()**

Berfungsi untuk mengarahkan pengguna ke halaman Informatika Student Portal.

Kembalian: halaman *login* jika pengguna belum *login* atau halaman utama jika pengguna sudah *login*.

- **public Result login()**

Berfungsi untuk mengarahkan pengguna ke halaman *login*.

Kembalian: halaman *login* jika pengguna belum *login* atau halaman utama jika pengguna sudah *login*.

- **public Result submitLogin()**

Berfungsi untuk mengirimkan data dari halaman *login* sekaligus melakukan validasi akun.

Kembalian: halaman utama jika *login* berhasil atau halaman *login* jika *login* gagal.

- **public Result home()**

Berfungsi untuk mengarahkan pengguna ke halaman utama.

Kembalian: halaman utama.

- **public Result prasyarat()**

Berfungsi untuk mengarahkan pengguna ke halaman prasyarat mata kuliah.

Kembalian: halaman prasyarat mata kuliah.

- **public Result jadwalKuliah()**

Berfungsi untuk mengarahkan pengguna ke halaman jadwal kuliah.

Kembalian: halaman jadwal kuliah.

- **public Result ringkasan()**

Berfungsi untuk mengarahkan pengguna ke halaman ringkasan data akademik.

Kembalian: halaman ringkasan data akademik.

- **public Result logout()**

Berfungsi untuk mengeluarkan pengguna yang sedang *login*.

Kembalian: halaman *login*.

2. Scaper

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

- **String BASE_URL:** URL Portal Akademik Mahasiswa.
- **String LOGIN_URL:** URL *login* Portal Akademik Mahasiswa.
- **String CAS_URL:** URL *login* CAS UNPAR.
- **String ALLJADWAL_URL:** URL jadwal seluruh fakultas pada Portal Akademik Mahasiswa.
- **String JADWAL_URL:** URL jadwal mahasiswa pada Portal Akademik Mahasiswa.
- **String NILAI_URL:** URL riwayat nilai mahasiswa pada Portal Akademik Mahasiswa.
- **String LOGOUT_URL:** URL *logout* Portal Akademik Mahasiswa.
- **TahunSemester currTahunSemester:** tahun ajaran dan semester terkini.
- **List<MataKuliah> mkList:** daftar mata kuliah yang dibuka pada semester terkini.

Method-method yang dimiliki kelas ini adalah sebagai berikut:

- **public void init()**

Berfungsi untuk menginisialisasi koneksi ke Portal Akademik Mahasiswa.

Kembalian: tidak ada.

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

Berfungsi untuk melakukan *login*.

Parameter:

- **npm** NPM mahasiswa.
- **pass** password mahasiswa.

Kembalian: objek Mahasiswa.

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

Berfungsi untuk mendapatkan daftar mata kuliah yang dibuka pada semester terkini.

Parameter:

- **login_cookies** cookie mahasiswa yang telah *login*.

Kembalian: tidak ada.

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

Berfungsi untuk mendapatkan jadwal kuliah mahasiswa pada semester terkini.

Parameter:

- **login_cookies** cookie mahasiswa yang telah *login*.

Kembalian: jadwal kuliah mahasiswa pada semester terkini.

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

Berfungsi untuk mendapatkan riwayat nilai mahasiswa.

Parameter:

- **login_cookies** cookie mahasiswa yang telah *login*.

- **logged_mhs** objek Mahasiswa dari mahasiswa yang telah *login*.

Kembalian: tidak ada.

- **public void logout()**

Berfungsi untuk melakukan *logout*.

Kembalian: tidak ada.

3. CustomMahasiswa

Kelas ini merupakan turunan dari kelas Mahasiswa. Kelas ini merepresentasikan mahasiswa yang memiliki jadwal kuliah dan URL foto profil. Atribut yang dimiliki kelas ini antara lain:

- **String photoPath:** URL foto profil mahasiswa pada Portal Akademik Mahasiswa.
- **List<JadwalKuliah> jadwalList:** daftar jadwal kuliah mahasiswa.

Method-method yang dimiliki kelas ini adalah sebagai berikut:

- **public CustomMahasiswa(String npm)**

Merupakan *constructor* dari kelas CustomMahasiswa.

Parameter:

- **npm** Nomor Pokok Mahasiswa.

4. Jadwal

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

- **MataKuliah mataKuliah:** mata kuliah.
- **char kelas:** kelas kuliah.
- **String waktu:** waktu pelaksanaan kuliah.
- **String ruang:** ruang kuliah.

5. JadwalKuliah

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

- **MataKuliah mataKuliah:** mata kuliah.
- **char kelas:** kelas kuliah.
- **String waktu:** waktu pelaksanaan kuliah.
- **String ruang:** ruang kuliah.

6. PrasyaratDisplay

Kelas berfungsi sebagai media pengiriman data ke halaman prasyarat mata kuliah. Atribut yang dimiliki kelas ini antara lain:

- **MataKuliah mataKuliah:** mata kuliah.
- **String[] status:** status pengambilan mata kuliah.

7. JadwalDisplay

Kelas berfungsi sebagai media pengiriman data ke halaman jadwal mata kuliah. Atribut yang dimiliki kelas ini antara lain:

- **JadwalKuliah[][] kuliahCalendar:** jadwal kuliah mahasiswa dalam *array*.
- **String[] hariList:** nama-nama hari dalam String.
- **List<JadwalKuliah> jadwalList:** daftar jadwal kuliah mahasiswa.

Method-method yang dimiliki kelas ini adalah sebagai berikut:

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

Merupakan *constructor* dari kelas JadwalDisplay.

Parameter:

– **jadwalList** jadwal kuliah mahasiswa.

- **public String getHariByIndex(int index)**

Berfungsi untuk mendapatkan hari berdasarkan angka index. Angka index dimulai dari 0 sedangkan hari dimulai dari Senin.

Parameter:

– **index** angka index hari.

Kembalian: hari dalam String.

- **public JadwalKuliah getJadwalKuliah(int indexHari, int indexWaktu)**

Berfungsi untuk mendapatkan jadwal kuliah dari atribut kuliahCalendar.

Parameter:

– **indexHari** angka index hari.

– **indexWaktu** angka index waktu.

Kembalian: jadwal kuliah.

- **public boolean isKuliahEmpty()**

Berfungsi untuk memeriksa apakah nilai dari jadwal kuliah kosong.

Kembalian: true jika kosong, false jika tidak kosong.

- **private void fillKuliahCalendar()**

Berfungsi untuk mengisi atribut kuliahCalendar berdasarkan atribut jadwalList.

Kembalian: tidak ada.

8. RingkasanDisplay

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

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

Method-method yang dimiliki kelas ini adalah sebagai berikut:

- **public RingkasanDisplay(String IPS, String IPK, int sksLulusTotal)**

Merupakan *constructor* dari kelas Ringkasan Display.

Parameter:

- **IPS** IPS mahasiswa.
- **IPK** IPK mahasiswa.
- **sksLulusTotal** total sks lulus mahasiswa.

- **public String getNamaPilWajib(String kode)**

Berfungsi untuk mendapatkan nama mata kuliah pilihan wajib.

Parameter:

- **kode** kode mata kuliah pilihan wajib.

Kembalian: nama mata kuliah pilihan wajib.

- **public void setDataSemTerakhir(String semTerakhir, int sksLulusSemTerakhir)**

Berfungsi untuk mengisi data semester terakhir.

Parameter:

- **semTerakhir** semester terakhir yang telah ditempuh mahasiswa.
- **sksLulusSemTerakhir** sks lulus mahasiswa pada semester terakhir.

Kembalian: tidak ada.

- **public int getMinSisaSKS()**

Berfungsi untuk mendapatkan sisa sks minimum untuk kelulusan.

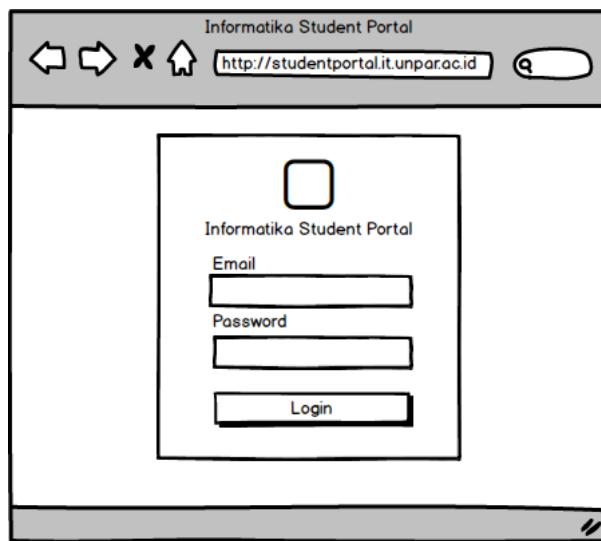
Kembalian: sisa sks untuk minimum kelulusan.

4.2 Perancangan Antarmuka

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

1. Antarmuka Halaman *Login*

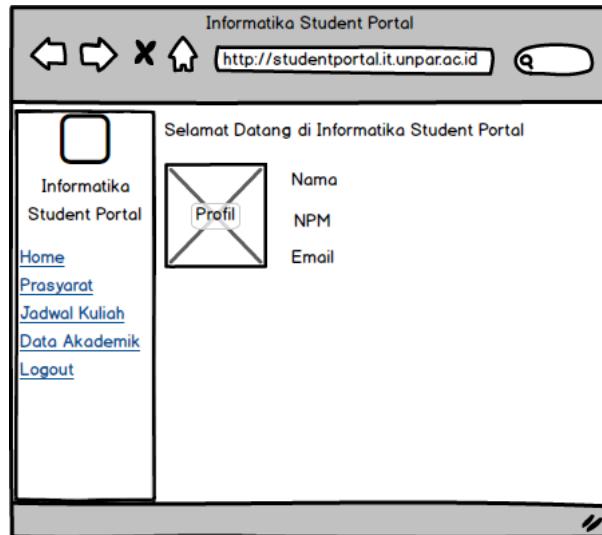
Halaman ini digunakan untuk melakukan *login*. Komponen halaman ini terdiri dari logo aplikasi, kolom *email*, kolom *password*, dan tombol *login* seperti yang ditunjukkan pada Gambar 4.2. Untuk melakukan *login*, pengguna perlu memasukkan *email* dan *password* yang sesuai kemudian menekan tombol *login*.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



Gambar 4.2: Rancangan Halaman *Login*

2. Antarmuka Halaman *Home*

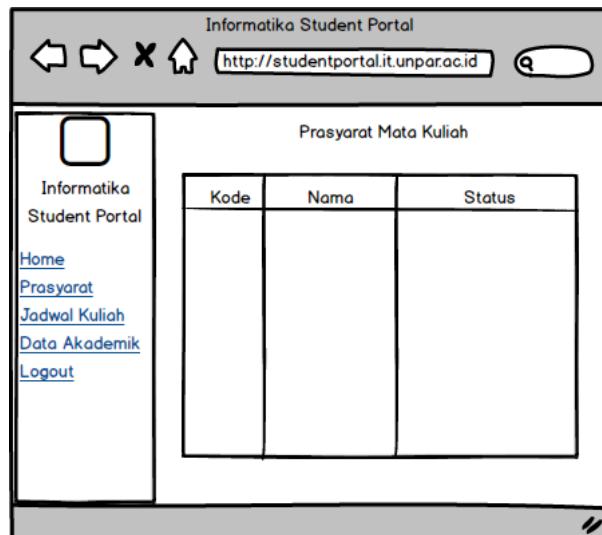
Halaman **home** merupakan halaman yang pertama dituju setelah melakukan **login**. Halaman ini, menampilkan identitas pengguna seperti foto profil, nama, Nomor Induk Mahasiswa(NPM), dan *email* mahasiswa. Rancangan antarmuka halaman *home* dapat dilihat pada Gambar 4.3.



Gambar 4.3: Rancangan Halaman *Home*

3. Antarmuka Halaman Prasyarat Mata Kuliah

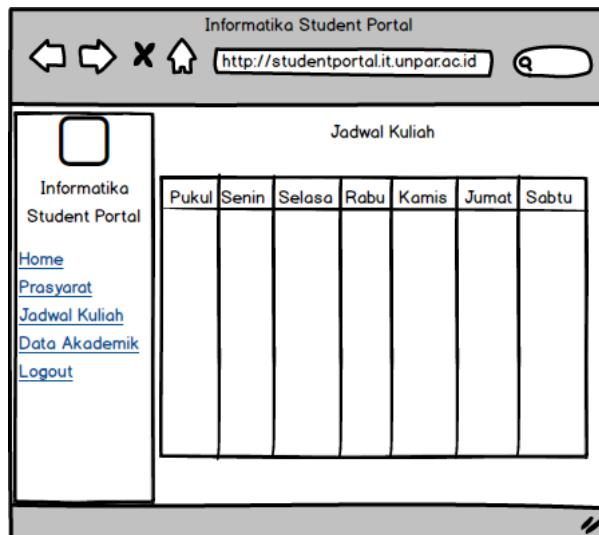
Halaman ini menampilkan tabel prasyarat mata kuliah yang dibuka pada semester terkini. Tabel tersebut memiliki tiga kolom yaitu kode mata kuliah, nama mata kuliah, dan status pengambilan mata kuliah. Rancangan antarmuka halaman prasyarat mata kuliah dapat dilihat pada Gambar 4.4.



Gambar 4.4: Rancangan Halaman Prasyarat Mata Kuliah

4. Antarmuka Halaman Jadwal Kuliah

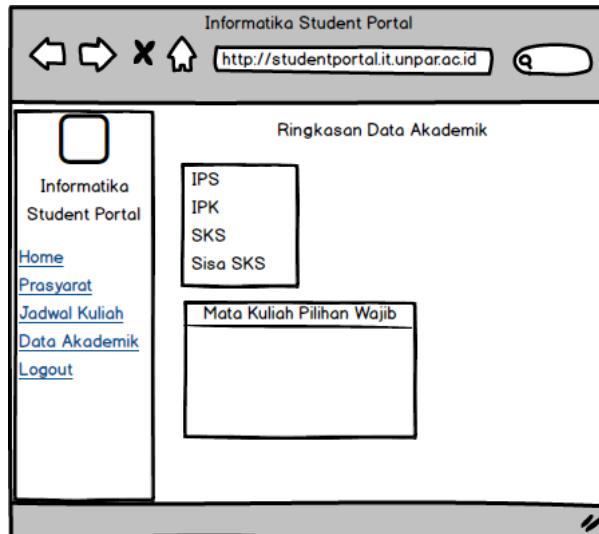
Halaman ini menampilkan jadwal kuliah semester terkini yang tersusun dan terurut berdasarkan hari. Rancangan antarmuka halaman jadwal kuliah dapat dilihat pada Gambar 4.5.



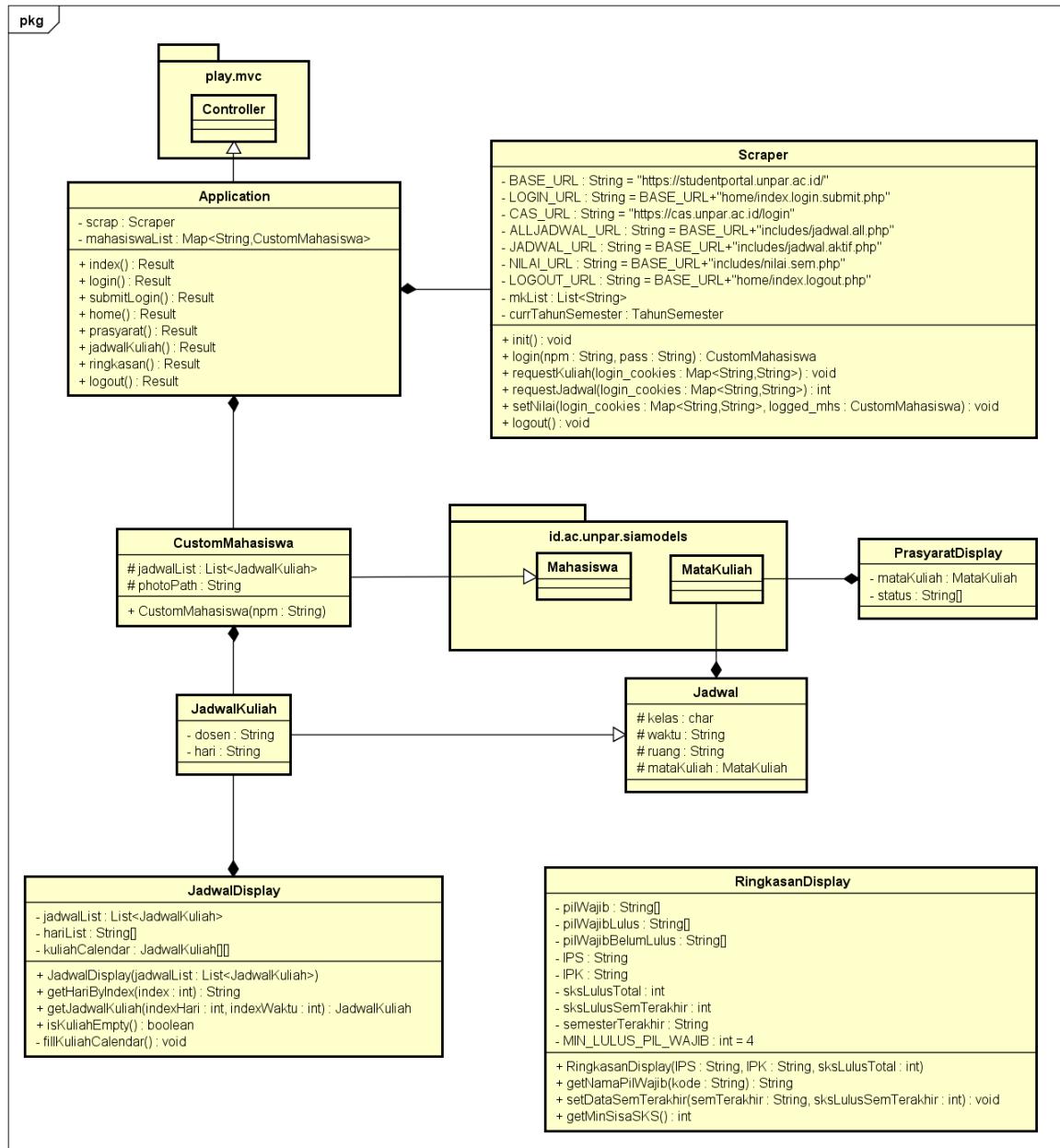
Gambar 4.5: Rancangan Halaman Jadwal Kuliah

5. Antarmuka Halaman Data Akademik

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



Gambar 4.6: Rancangan Halaman Data Akademik



Gambar 4.1: Diagram Kelas Rinci

BAB 5

IMPLEMENTASI DAN PENGUJIAN

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

5.1 Implementasi

5.1.1 Lingkungan Implementasi

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

1. Processor: 3.20Ghz
2. RAM: 4.00 GB DDR3
3. Sistem Operasi: Windows 8.1 Pro 64-bit
4. Versi Java: 1.8.0_40

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

1. Processor: 2.66Ghz
2. RAM: 4.00 GB DDR2
3. Sistem Operasi: Ubuntu server amd-64
4. Versi Java: 1.8.0_66

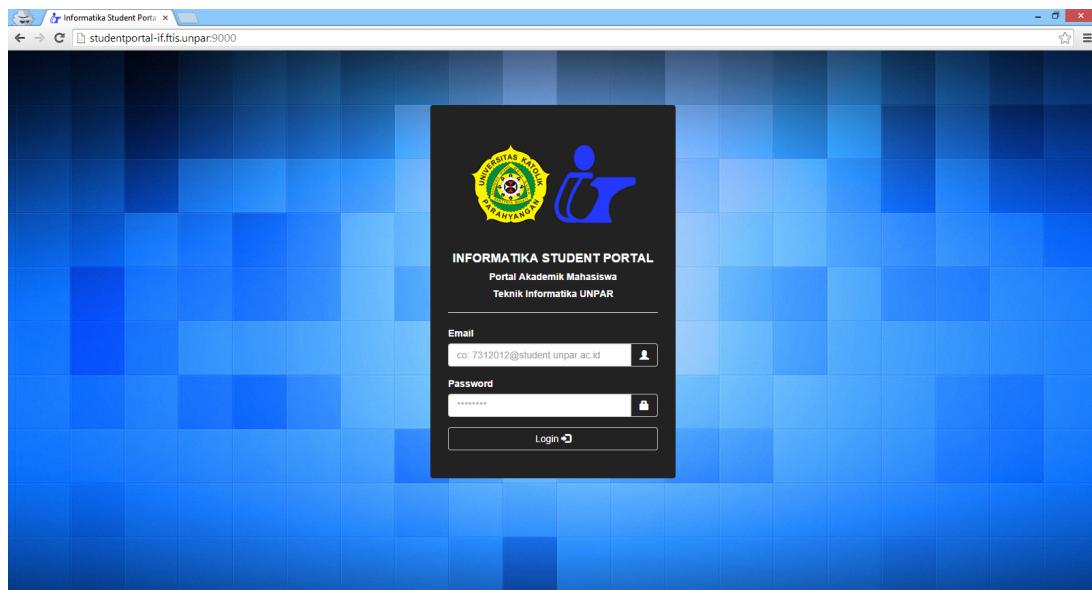
5.1.2 Hasil Implementasi

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:

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

1. Halaman *Login*

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

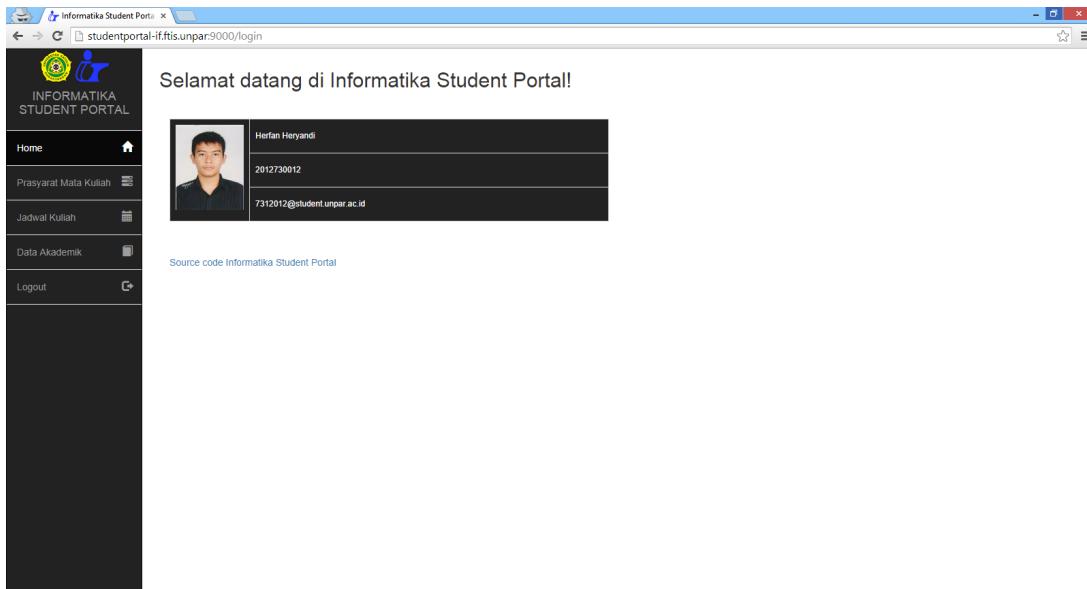


Gambar 5.1: Halaman *Login*

2. Halaman *Home*

Halaman utama merupakan halaman yang pertama kali dituju setelah melakukan *login*. Halaman utama menampilkan identitas pengguna dan *link* menuju kode sumber aplikasi Informatika Student Portal. Pada halaman ini juga terdapat *sidebar menu* yang diperoleh dari *template* StartBootstrap². *Template* tersebut juga digunakan pada halaman prasyarat mata kuliah, jadwal kuliah, dan data akademik. Tangkapan layar dari halaman utama dapat dilihat pada Gambar 5.2.

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

Gambar 5.2: Halaman *Home*

3. Halaman Prasyarat Mata Kuliah

Halaman ini menampilkan tabel prasyarat mata kuliah. Jika prasyarat mata kuliah tersedia, pengguna dapat mengklik kode mata kuliah, kemudian akan diarahkan ke kode sumber aturan prasyarat mata kuliah tersebut. Tangkapan layar dari halaman 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 Grafiti	memenuhi syarat ✓
AIF336	Algoritma Kriptografi	tidak memiliki prasyarat ✓
AIF341	Administrasi Jaringan Komputer 1	sudah lulus ?

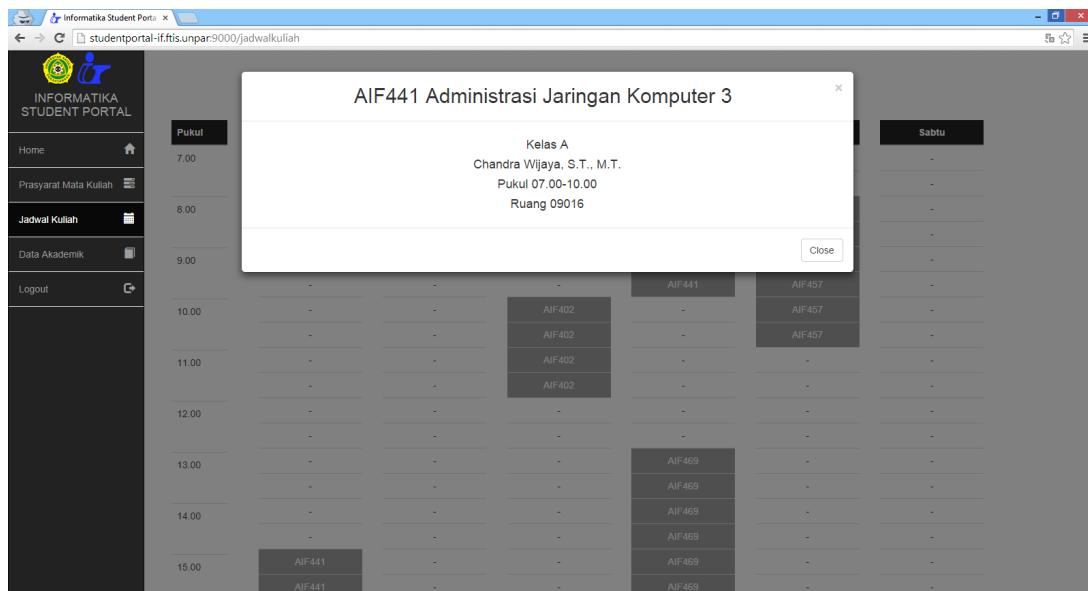
Gambar 5.3: Halaman Prasyarat Mata Kuliah

4. Halaman Jadwal Kuliah

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

JADWAL KULIAH						
SEMESTER GANJIL 2015/2016						
Pukul	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
7.00	-	-	-	-	-	-
8.00	-	AIF401 AIF401 AIF401 AIF401	-	-	-	-
9.00	-	-	-	-	-	-
10.00	-	-	-	AIF402 AIF402 AIF402 AIF402	-	-
11.00	-	-	-	-	-	-
12.00	-	-	-	-	-	-
13.00	-	-	-	-	AIF469 AIF469 AIF469 AIF469 AIF469	-
14.00	-	-	-	-	-	-
15.00	AIF441 AIF441	-	-	-	-	-

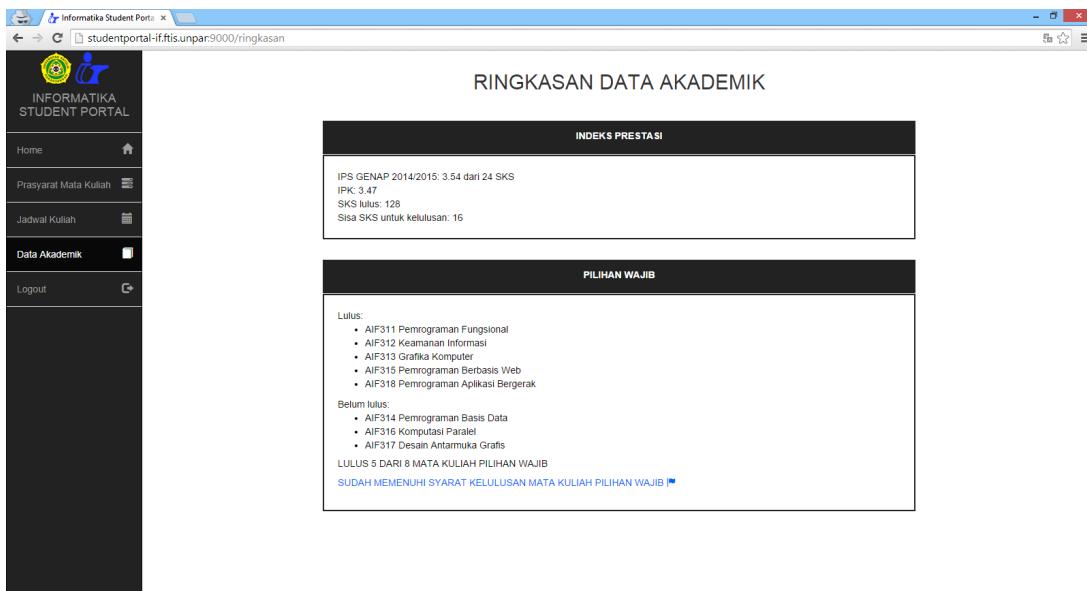
Gambar 5.4: Halaman Jadwal Kuliah



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 pada 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” Jika <i>email</i> yang dimasukkan bukan <i>email</i> mahasiswa teknik informatika, akan ditampilkan pesan “Maaf, Anda bukan mahasiswa teknik informatika”	sesuai, namun foto profil tidak muncul jika belum menambahkan eksepsi terhadap sertifikat SSL UNPAR sesuai sesuai
3.	Pengguna memilih menu “Prasyarat Mata Kuliah”	Jika pengguna belum memiliki riwayat nilai(masih menempuh semester 1), akan ditampilkan pesan “PRASYARAT BELUM TERSEDIA” 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” 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” 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 tiga 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

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

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

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

6.2 Saran

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

1. Penelitian ini memanfaatkan SIA Model terutama *package* “matakuliah” yang berisi aturan mengenai prasyarat mata kuliah. Aturan-aturan tersebut dapat berubah suatu saat. Oleh karena itu, sebaiknya SIA Models dipisahkan dari *project* Play sehingga jika terjadi perubahan aturan pada SIA Models, tidak perlu dilakukan perubahan pada *project* yang dapat mengakibatkan kesalahan kode program.
2. Dalam pengembangan berikutnya, perlu diperhatikan data dari mahasiswa yang pernah melakukan transfer studi. Jika memungkinkan, sebaiknya dianalisis langsung dari akun Portal Akademik Mahasiswa milik mahasiswa tersebut agar dapat mengambil data yang sesuai. Saat ini, aplikasi mengambil data langsung dari halaman riwayat nilai Portal Akademik Mahasiswa. Terdapat halaman lain yang memuat nilai mahasiswa pada Portal Akademik Mahasiswa yaitu halaman Daftar Perkembangan Studi (DPS). Sebaiknya pengambilan nilai dilakukan pada halaman DPS.

3. Saat ini, aplikasi berjalan pada port 9000 dan tidak dijalankan secara otomatis saat server mulai berjalan. Sebaiknya aplikasi dapat langsung berjalan saat server mulai berjalan dan aplikasi juga dapat berjalan pada port 80.
4. Aplikasi belum dapat menangani *path* yang tidak terdapat pada *routes* sehingga aplikasi akan memberikan pesan error jika memasuki *path* tersebut. Dalam pengembangan berikutnya, *path* yang tidak ditemukan sebaiknya diarahkan ke halaman “404 Not Found”.

DAFTAR REFERENSI

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

LAMPIRAN A

KODE PROGRAM

Listing A.1: Application.java

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

```

72         return home();
73     }
74     else{
75         return ok(views.html.login.render(errorHtml+"Password yang Anda masukkan salah atau Anda bukan
76         mahasiswa aktif"+ "</div>"));
77     }
78 }
79 public Result home(){
80     if(session("npm") == null){
81         return index();
82     }
83     else{
84         return ok(views.html.home.render(mahasiswaList.get(session("npm"))));
85     }
86 }
87
88 public Result prasyarat() throws IOException{
89     if(session("npm") == null){
90         return index();
91     }
92     else if(mahasiswaList.get(session("npm")).getRiwayatNilai().size() == 0){
93         List<PrasyaratDisplay> table = null;
94         String semester = scrap.getSemester();
95         return ok(views.html.prasyarat.render(table,semester));
96     }
97     else{
98         List<PrasyaratDisplay> table = checkPrasyarat();
99         String semester = scrap.getSemester();
100        return ok(views.html.prasyarat.render(table,semester));
101    }
102 }
103
104 public Result jadwalKuliah() throws IOException{
105     if(session("npm") == null){
106         return index();
107     }
108     else{
109         JadwalDisplay table = new JadwalDisplay(mahasiswaList.get(session("npm")).getJadwalList());
110         String semester = scrap.getSemester();
111         return ok(views.html.jadwalKuliah.render(table,semester));
112     }
113 }
114
115 public Result ringkasan() throws IOException{
116     if(session("npm") == null){
117         return index();
118     }
119     else if(mahasiswaList.get(session("npm")).getRiwayatNilai().size() == 0){
120         RingkasanDisplay display = null;
121         return ok(views.html.ringkasan.render(display));
122     }
123     else{
124         Mahasiswa currMahasiswa = mahasiswaList.get(session("npm"));
125         RingkasanDisplay display = new RingkasanDisplay(
126             String.format("%.2f", currMahasiswa.calculateIPS()),
127             String.format("%.2f", currMahasiswa.calculateIPKLulus()),
128             currMahasiswa.calculateSKSLulus());
129     };
130     List<Nilai> riwayatNilai = currMahasiswa.getRiwayatNilai();
131     int lastIndex = riwayatNilai.size() - 1;
132     Semester semester = riwayatNilai.get(lastIndex).getSemester();
133     int tahunAjaran = riwayatNilai.get(lastIndex).getTahunAjaran();
134     int totalSKS = 0;
135     for(int i = lastIndex; i >= 0; i--){
136         Nilai nilai = riwayatNilai.get(i);
137         if(nilai.getSemester() == semester && nilai.getTahunAjaran() == tahunAjaran) {
138             if(nilai.getAngkaAkhir() != null) {
139                 totalSKS += nilai.getMataKuliah().sks();
140             }
141         } else {
142             break;
143         }
144     }
145
146     String semTerakhir = semester + " " + tahunAjaran + " / " + (tahunAjaran + 1);
147     display.setDataSemTerakhir(semTerakhir, totalSKS);
148     String pilWajibLulus = new String();
149     String pilWajibBelumLulus = new String();
150     for(int i=0; i < display.getPilWajib().length; i++){
151         if(mahasiswaList.get(session("npm")).hasLulusKuliah(display.getPilWajib()[i])){
152             pilWajibLulus += display.getPilWajib()[i] + ";";
153         }
154         else{
155             pilWajibBelumLulus += display.getPilWajib()[i] + ";";
156         }
157     }
158     if(!pilWajibLulus.isEmpty()){
159         display.setPilWajibLulus(pilWajibLulus.split(";"));
160     }
161     else{
162         display.setPilWajibLulus(new String[]{} );
163     }
164     if(!pilWajibBelumLulus.isEmpty()){
165         display.setPilWajibBelumLulus(pilWajibBelumLulus.split(";"));
166     }
167     else{
168         display.setPilWajibBelumLulus(new String[]{} );
169     }

```

```

170         return ok(views.html.ringkasan.render(display));
171     }
172 }
173
174 public Result logout() throws IOException {
175     session().clear();
176     mahasiswaList.remove(session("npm"));
177     return index();
178 }
179
180 private List<PrasyaratDisplay> checkPrasyarat() throws IOException{
181     List<PrasyaratDisplay> table = new ArrayList<PrasyaratDisplay>();
182     List<MataKuliah> mkList = scrap.getMkList();
183     String MATAKULIAH_REPOSITORY_PACKAGE = "models.id.ac.unpar.siamodels.matakuliah";
184     List<Object> mkKnown = new ArrayList<Object>();
185     List<MataKuliah> mkUnknown = new ArrayList<MataKuliah>();
186     for(MataKuliah mk : mkList){
187         try {
188             Class<?> mkClass = Class.forName(MATAKULIAH_REPOSITORY_PACKAGE + " ." + mk.kode());
189             Object matakuliah = mkClass.newInstance();
190             mkKnown.add(matakuliah);
191         } catch (ClassNotFoundException e) {
192             mkUnknown.add(mk);
193         } catch (InstantiationException e) {
194             e.printStackTrace();
195         } catch (IllegalAccessException e) {
196             e.printStackTrace();
197         }
198     }
199
200     for (Object mk: mkKnown) {
201         if (mk instanceof HasPrasyarat) {
202             List<String> reasons = new ArrayList<String>();
203             ((HasPrasyarat)mk).checkPrasyarat(mahasiswaList.get(session("npm")), reasons);
204             if (!reasons.isEmpty()) {
205                 String status = new String();
206                 for (String reason: reasons) {
207                     status+=reason + ";";
208                 }
209                 table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.getClass(),
210                                         getSimpleName(), MataKuliahFactory.UNKNOWN_SKS, MataKuliahFactory.UNKNOWN_NAMA),
211                                         status.split(";")));
212             } else{
213                 if(mahasiswaList.get(session("npm")).hasLulusKuliah(mk.getClass().getSimpleName())){
214                     table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.getClass(),
215                                         getSimpleName(), MataKuliahFactory.UNKNOWN_SKS, MataKuliahFactory.UNKNOWN_NAMA),
216                                         new String[]{"sudah_lulus"}));
217                 }
218             }
219         } else{
220             if(mahasiswaList.get(session("npm")).hasLulusKuliah(mk.getClass().getSimpleName())){
221                 table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.getClass(),
222                                         getSimpleName(), MataKuliahFactory.UNKNOWN_SKS, MataKuliahFactory.UNKNOWN_NAMA),new
223                                         String[]{"sudah_lulus"}));
224             } else{
225                 table.add(new PrasyaratDisplay(MataKuliahFactory.createMataKuliah(mk.getClass(),
226                                         getSimpleName(), MataKuliahFactory.UNKNOWN_SKS, MataKuliahFactory.UNKNOWN_NAMA),new
227                                         String[]{"tidak_mempeliki_prasyarat"}));
228             }
229         }
230         for (MataKuliah mk: mkUnknown) {
231             table.add(new PrasyaratDisplay(mk,new String[]{"data_prasyarat_tidak_tersedia"}));
232         }
233     }
234     return table;
235 }
236 }
237 }
```

Listing A.2: JadwalDisplay.java

```

1 package models.display;
2
3 import java.util.List;
4
5 import models.support.JadwalKuliah;
6
7 public class JadwalDisplay {
8     private List<JadwalKuliah> jadwallist;
9     private JadwalKuliah[][] kuliahCalendar;
10    private String[] hariList;
11
12    public JadwalDisplay(List<JadwalKuliah> jadwallist){
13        this.jadwallist = jadwallist;
14        kuliahCalendar = new JadwalKuliah[6][22];
15        hariList = new String[]{"Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"};
16    }
17}
```

```

16     fillKuliahCalendar();
17 }
18
19 public String getHariByIndex(int index){
20     return this.hariList[index];
21 }
22
23 public JadwalKuliah getJadwalKuliah(int indexHari, int indexWaktu){
24     return kuliahCalendar[indexHari][indexWaktu];
25 }
26
27 public boolean isKuliahEmpty(){
28     return jadwalList.isEmpty();
29 }
30
31 private void fillKuliahCalendar(){
32     if(!jadwalList.isEmpty()){
33         for (int i = 0; i < kuliahCalendar.length; i++) {
34             for (int j = 0; j < kuliahCalendar[i].length; j++) {
35                 kuliahCalendar[i][j] = new JadwalKuliah();
36             }
37         }
38         for (int i = 0; i < jadwalList.size(); i++) {
39             JadwalKuliah jdw = jadwalList.get(i);
40             int day = dayTranslate(jdw.getHari());
41             if(day!=-1){
42                 String[] timePair = jdw.getWaktu().split("-");
43                 String start = timePair[0];
44                 String end = timePair[1];
45                 int range = (Integer.parseInt(end.substring(0, 2))- Integer.parseInt(start.substring(0, 2)))*2;
46                 int beginIndex = 0;
47                 int half = Character.getNumericValue(start.charAt(3));
48                 if(half<3){
49                     beginIndex = (Integer.parseInt(start.substring(0, 2))-7)*2;
50                 }
51                 else if(half>=3){
52                     beginIndex =((Integer.parseInt(start.substring(0, 2))-7)*2)+1;
53                 }
54                 int endHalf = Character.getNumericValue(end.charAt(3));
55                 if(endHalf>3){
56                     range++;
57                 }
58                 for (int j = beginIndex; j < beginIndex+range; j++) {
59                     kuliahCalendar[day][j] = jdw;
60                 }
61             }
62         }
63     }
64 }
65
66 private int dayTranslate(String hari){
67     int day = -1;
68     switch(hari){
69         case "Senin":
70             day = 0;
71             break;
72         case "Selasa":
73             day = 1;
74             break;
75         case "Rabu":
76             day = 2;
77             break;
78         case "Kamis":
79             day = 3;
80             break;
81         case "Jumat":
82             day = 4;
83             break;
84         case "Sabtu":
85             day = 5;
86             break;
87         default:
88             day = -1;
89             break;
90     }
91     return day;
92 }
93 }
```

Listing A.3: PrasyaratDisplay.java

```

1 package models.display;
2
3 import models.id.ac.unpar.siamodels.MataKuliah;
4
5 public class PrasyaratDisplay {
6     private MataKuliah mataKuliah;
7     private String[] status;
8
9     public PrasyaratDisplay(MataKuliah mataKuliah, String[] status){
10        this.mataKuliah = mataKuliah;
11        this.status = status;
12    }
13
14    public MataKuliah getMataKuliah() {
```

```

15     return mataKuliah;
16 }
17
18 public String[] getStatus(){
19     return status;
20 }
21 }
```

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", "AIF318"};
23    }
24
25    public int getMinLulusPilWajib(){
26        return this.MIN_LULUS_PIL_WAJIB;
27    }
28
29    public String getNamaPilWajib(String kode){
30        return MataKuliahFactory.createMataKuliah(kode, MataKuliahFactory.UNKNOWN_SKS, MataKuliahFactory.UNKNOWN_NAMA).nama()+"";
31    }
32
33    public String[] getPilWajibLulus() {
34        return pilWajibLulus;
35    }
36
37    public void setPilWajibLulus(String[] pilWajibLulus) {
38        this.pilWajibLulus = pilWajibLulus;
39    }
40
41    public String[] getPilWajibBelumLulus() {
42        return pilWajibBelumLulus;
43    }
44
45
46    public void setPilWajibBelumLulus(String[] pilWajibBelumLulus) {
47        this.pilWajibBelumLulus = pilWajibBelumLulus;
48    }
49
50    public String[] getPilWajib(){
51        return this.pilWajib;
52    }
53
54    public String getIPS(){
55        return this.IPS;
56    }
57
58    public String getIPK(){
59        return this.IPK;
60    }
61
62    public void setDataSemTerakhir(String semTerakhir, int sksLulusSemTerakhir) {
63        this.semesterTerakhir = semTerakhir;
64        this.sksLulusSemTerakhir = sksLulusSemTerakhir;
65    }
66
67    public String getSemesterTerakhir(){
68        return semesterTerakhir;
69    }
70    public int getSKSLulusTotal(){
71        return this.sksLulusTotal;
72    }
73
74    public int getSKSLulusSemTerakhir(){
75        return this.sksLulusSemTerakhir;
76    }
77
78    public int getMinSisaSKS(){
79        if(sksLulusTotal >=144){
80            return 0;
81        }
82        else{
83            return 144-sksLulusTotal;
84        }
85    }
}
```

86| }

Listing A.5: CustomMahasiswa.java

```

1 package models.support;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import models.id.ac.unpar.siamodels.Mahasiswa;
7
8 public class CustomMahasiswa extends Mahasiswa{
9     protected List<JadwalKuliah> jadwalList;
10    protected String photoPath;
11
12    public CustomMahasiswa(String npm) throws NumberFormatException {
13        super(npm);
14        this.jadwalList = new ArrayList<JadwalKuliah>();
15    }
16
17    public String getPhotoPath(){
18        return photoPath;
19    }
20
21    public void setPhotoPath(String photoPath) {
22        this.photoPath = photoPath;
23    }
24
25    public List<JadwalKuliah> getJadwalList(){
26        return jadwalList;
27    }
28
29    public void setJadwalList(List<JadwalKuliah> jadwalList){
30        this.jadwalList = jadwalList;
31    }
32}
33

```

Listing A.6: Jadwal.java

```

1 package models.support;
2
3 import models.id.ac.unpar.siamodels.MataKuliah;
4
5 public class Jadwal {
6     protected MataKuliah mataKuliah;
7     protected char kelas;
8     protected String waktu;
9     protected String ruang;
10
11    public Jadwal(MataKuliah mataKuliah, char kelas, String waktu, String ruang) {
12        this.mataKuliah = mataKuliah;
13        this.kelas = kelas;
14        this.waktu = waktu;
15        this.ruang = ruang;
16    }
17
18    public Jadwal() {
19    }
20
21    public MataKuliah getMataKuliah() {
22        return mataKuliah;
23    }
24
25    public char getKelas() {
26        return kelas;
27    }
28
29    public String getWaktu() {
30        return waktu;
31    }
32
33    public String getRuang() {
34        return ruang;
35    }
36}
37
38

```

Listing A.7: JadwalKuliah.java

```

1 package models.support;
2
3 import models.id.ac.unpar.siamodels.MataKuliah;
4
5 public class JadwalKuliah extends Jadwal{
6     private String dosen;
7     private String hari;
8

```

```

9  public JadwalKuliah(MataKuliah mataKuliah, char kelas, String dosen, String hari, String waktu, String
10    ruang) {
11      super(mataKuliah, kelas, waktu, ruang);
12      this.dosen = dosen;
13      this.hari = hari;
14  }
15  public JadwalKuliah() {
16      super();
17  }
18
19  public String getDosen() {
20      return dosen;
21  }
22
23  public String getHari() {
24      return hari;
25  }
26}
27}

```

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() + "/" + (currTahunSemester.
39             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);

```

```

73     resp = loginConn.execute();
74     if(resp.body().contains("Data_Akademik")){
75         Map<String ,String> login_cookies = resp.cookies();
76         doc = resp.parse();
77         String nama = doc.select("p[class=student-name]").text();
78         logged_mhs.setNama(nama);
79         String curr_sem = doc.select(".main-info-semester-a").text();
80         String [] sem_set = this.parseSemester(curr_sem);
81         Element photo = doc.select(".student-photo-img").first();
82         String photoPath = photo.absUrl("src");
83         logged_mhs.setPhotoPath(photoPath);
84         logged_mhs.setTahunSemester(new TahunSemester(Integer.parseInt(sem_set[0]), Semester.fromString(sem_set[1])));
85         this.requestKuliah(login_cookies);
86         List<JadwalKuliah> jadwalList = this.requestJadwal(login_cookies);
87         logged_mhs.setJadwalList(jadwalList);
88         this.setNilai(login_cookies, logged_mhs);
89         logout();
90         return logged_mhs;
91     }
92     else{
93         return null;
94     }
95 }
96
97 public void requestKuliah(Map<String ,String> login_cookies) throws IOException{
98     Connection kuliahConn = Jsoup.connect(ALLJADWAL_URL);
99     kuliahConn.cookies(login_cookies);
100    kuliahConn.timeout(0);
101    kuliahConn.validateTLCertificates(false);
102    kuliahConn.method(Connection.Method.GET);
103    Response resp = kuliahConn.execute();
104    Document doc = resp.parse();
105    Elements jadwal = doc.select("tr");
106    String prev = "";
107    mkList = new ArrayList<MataKuliah>();
108    for (int i = 1; i < jadwal.size()-1; i++) {
109        Elements row = jadwal.get(i).children();
110        if(!row.get(1).text().equals("")){
111            String kode = row.get(1).text();
112            String nama = row.get(2).text();
113            String sks = row.get(3).text();
114            if(!kode.equals(prev)){
115                MataKuliah curr = MataKuliahFactory.createMataKuliah(kode, Integer.parseInt(sks), nama);
116                mkList.add(curr);
117            }
118            prev = kode;
119        }
120    }
121 }
122
123 public List<JadwalKuliah> requestJadwal(Map<String ,String> login_cookies) throws IOException{
124     Connection jadwalConn = Jsoup.connect(JADWAL_URL);
125     jadwalConn.cookies(login_cookies);
126     jadwalConn.timeout(0);
127     jadwalConn.validateTLCertificates(false);
128     jadwalConn.method(Connection.Method.GET);
129     Response resp = jadwalConn.execute();
130     Document doc = resp.parse();
131     Elements jadwalTable = doc.select(".portal-full-table");
132     List<JadwalKuliah> jadwalList = new ArrayList<JadwalKuliah>();
133
134     /*Kuliah*/
135     if(jadwalTable.size()>0){
136         Elements tableKuliah = jadwalTable.get(0).select("tbody-tr");
137         String kode = new String();
138         String nama = new String();
139         for(Element elem : tableKuliah){
140             if(elem.className().contains("row")){
141                 if(!(elem.child(1).text().isEmpty() && elem.child(2).text().isEmpty())){
142                     kode = elem.child(1).text();
143                     nama = elem.child(2).text();
144                 }
145                 MataKuliah currMk = MataKuliahFactory.createMataKuliah(kode, Integer.parseInt(elem.child(3).text()), nama);
146                 jadwalList.add(new JadwalKuliah(currMk, elem.child(4).text().charAt(0), elem.child(5).text(), elem.child(7).text(), elem.child(8).text(), elem.child(9).text()));
147             }
148         }
149     }
150     return jadwalList;
151 }
152
153 public void setNilai(Map<String ,String> login_cookies, Mahasiswa logged_mhs) throws IOException{
154     Connection nilaiConn = Jsoup.connect(NILAI_URL);
155     nilaiConn.cookies(login_cookies);
156     nilaiConn.data("npm",logged_mhs.getNpm());
157     nilaiConn.data("thn_akd","ALL");
158     nilaiConn.timeout(0);
159     nilaiConn.validateTLCertificates(false);
160     nilaiConn.method(Connection.Method.POST);
161     Response resp = nilaiConn.execute();
162     Document doc = resp.parse();
163     Elements mk = doc.select("table");
164
165     for(Element tb:mk){
166         Elements tr = tb.select("tr");
167         String [] sem_set = this.parseSemester(tr.get(0).text());

```

```

168     String thn = sem_set[0];
169     String sem = sem_set[1];
170
171     for(Element td:tr){
172         if(td.className().contains("row")){
173             String kode = td.child(1).text();
174             int sks = Integer.parseInt(td.child(3).text());
175             String nama_mk = td.child(2).text();
176             MataKuliah curr_mk = MataKuliahFactory.createMataKuliah(kode, sks, nama_mk);
177             char kelas = td.child(4).text().charAt(0);
178             double ART = 0;
179             double UTS = 0;
180             double UAS = 0;
181             if(kelas!='*'){
182                 ART = Double.parseDouble(td.child(5).text());
183                 UTS = Double.parseDouble(td.child(6).text());
184                 UAS = Double.parseDouble(td.child(7).text());
185             }
186             if(!td.child(9).text().equals ""){
187                 char NA = td.child(9).text().charAt(0);
188                 TahunSemester tahunSemesterNilai = new TahunSemester(Integer.parseInt(thn), Semester.
189                             fromString(sem));
190                 logged_mhs.getRiwayatNilai().add(new Nilai(tahunSemesterNilai, curr_mk, kelas, ART,
191                                         UTS, UAS, NA));
192             }
193         }
194     }
195 }
196
197 public void logout() throws IOException{
198     Connection logoutConn = Jsoup.connect(LOGOUT_URL);
199     logoutConn.timeout(0);
200     logoutConn.validateTLCertificates(false);
201     logoutConn.method(Connection.Method.GET);
202     logoutConn.execute();
203 }
204
205 private String[] parseSemester(String sem_raw){
206     String[] sem_set = sem_raw.split("/")[0].split("-");
207     return new String[]{sem_set[1].trim(), sem_set[0].trim()};
208 }
209 }
```