

# **PENGEMBANGAN SISTEM INFORMASI UIIPERKULIAHAN DENGAN RESTFUL API**



Disusun Oleh:

N a m a : Mustafa Widiarto Heryatno

NIM : 15523055

**PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2020**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN SISTEM INFORMASI UI PERKULIAHAN  
DENGAN RESTFUL API**

**TUGAS AKHIR**



N a m a : Mustafa Widiarto Heryatno

NIM : 15523055

الجمعة الإسلامية  
الاستاذة الباندا

Yogyakarta, 4 Maret 2020

Pembimbing,

(Ahmad Fathan Hidayatullah, S.T., M.Cs.)

## HALAMAN PENGESAHAN DOSEN PENGUJI

# PENGEMBANGAN SISTEM INFORMASI UIPERKULIAHAN DENGAN RESTFUL API

## TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 4 Maret 2020

Tim Penguji

Ahmad Fathan Hidayatullah, S.T., M.Cs.

**Anggota 1**

Dhomas Hatta Fudholi, S.T., M.Eng.,  
Ph.D.

**Anggota 2**

Ahmad M. Raf'ie Pratama, S.T., M.I.T.,  
Ph.D.

Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Mustafa Widiarto Heryatno

NIM : 15523055

Tugas akhir dengan judul:

**PENGEMBANGAN SISTEM INFORMASI UIIPERKULIAHAN  
DENGAN RESTFUL API**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 4 Maret 2020



(Mustafa Widiarto Heryatno)

## HALAMAN PERSEMBAHAN



Alhamdulillah, segala puji bagi Allah sang pemilik alam semesta, yang maha pengasih lagi maha penyayang, pemilik hari pembalasan. Hanya dengan mengingatnya, manusia dapat merasakan ketenangan hidup. Karena menyadari bahwa segala sesuatu yang baik berasal dari Allah, dan kebaikan itu bisa menjadi nikmat maupun ujian. Karena Allah yang membuat manusia bisa merasakan nikmat iman, islam, hidayah, ukhuwah, dan taqwa. Karena Dia-lah yang membimbing manusia yang mengharapkan ridho-Nya menuju jalan yang lurus. Karya ini saya persembahkan untuk orang-orang yang saya sayangi:

### **Kedua orang tua saya yang tercinta**

(Ayahanda Yatnoto dan Ibunda Suciwati)

Terimakasih untuk semua rasa sayang dan cinta yang telah kalian curahkan. Tidak ada rasa sayang terbaik selain dari kalian.

### **Ketiga saudara saya yang tersayang**

(Mukhlis Aziz Heryatno, Rachmawati Lestari, Mustaqim Adiyatno)

Terimakasih untuk setiap cerita yang menemani proses pendewasaan diri ini. Semoga kita bisa selalu membawa manfaat untuk lingkungan kita.

### **Sahabat seperjuangan**

(Muhammad Fadhilatul Maa'rif dan Riefky Fadel)

Semoga Allah selalu menuntun dan memudahkan kita untuk menuju jalan kebaikan, dan menjadikan kita sebagai golongan orang beriman.

## HALAMAN MOTO

“Manfaatkanlah lima perkara sebelum lima perkara: (1) Waktu mudamu sebelum datang waktu tuamu, (2) Waktu sehatmu sebelum datang waktu sakitmu, (3) Masa kayamu sebelum datang masa kefakiranmu, (4) Masa luangmu sebelum datang masa sibukmu, (5) Hidupmu sebelum datang matimu.” (HR. Al Hakim)

“Jika sudah terjadi ya sudah, yang perlu dipikirkan sekarang solusinya” – (Ibunda)

“Jadikan dunia itu di genggamamu, bukan dihatimu” – Pak Idham

“Jangan menunda apa yang bisa kamu kerjakan hari ini” – *Unknown*

“Bersakit-sakit dahulu bersenang-senang kemudian” – *Unknown*

## KATA PENGANTAR

Segala puji dan syukur atas limpahan rahmat, taufiq, dan hidayah dari Allah SWT, sehingga terselesaikannya penelitian dengan judul **“Pengembangan Sistem Informasi UIIPerkuliahan dengan RESTful API”**.

Laporan Penelitian ini disusun sebagai salah satu syarat untuk mendapatkan gelar sarjana pada Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Sekaligus untuk meningkatkan layanan UIIGateway, sehingga tercipta sistem yang lebih baik.

Penulisan laporan ini tidak lepas dari dukungan, saran dan bimbingan dari berbagai pihak. Oleh karena itu, ungkapan terimakasih penulis berikan kepada:

1. Allah SWT yang telah membimbing penulis, sehingga dipertemukan dengan orang-orang yang baik.
2. Bapak Ahmad Fathan Hidayatullah S.T., M.Cs. sebagai dosen pembimbing yang telah meluangkan tenaga dan waktu untuk membimbing dan memberikan saran selama penyusunan laporan ini.
3. Kedua orang tua tercinta, serta adik-adik tersayang, semoga Allah membalas dengan limpahan keberkahan, rahmat, dan hidayah.
4. Teman-teman di BSI yang telah memberikan bantuan dan dukungan.
5. Teman-teman di Centris yang telah membantu proses memperbaiki diri sehingga menjadi lebih baik.
6. Teman-teman seangkatan yang telah lulus terlebih dahulu, sehingga menjadi cambuk untuk terus maju.

Terlepas dari berbagai macam kendala. Penulis mengerti bahwa masih ada hal yang bisa ditingkatkan sehingga sistem informasi ini bisa menjadi lebih baik. Oleh karena itu, penulis menerima kritik dan saran membangun, sehingga tercipta sistem informasi yang lebih baik. Semoga penelitian ini bisa membawa manfaat untuk lingkungan Universitas Islam Indonesia.

Yogyakarta, 4 Maret 2020

(Mustafa Widiarto Heryatno)

## SARI

Selama ini, UII menggunakan aplikasi berbasis desktop bernama “W-Simak” untuk mengolah nilai dan presensi mahasiswa. Aplikasi W-Simak saat ini belum menggunakan *web service*. Hal ini menyebabkan setiap perubahan yang dilakukan mengharuskan *deploy* ulang keseluruhan aplikasi yaitu bagian *back end* dan *front end* yang berimplikasi pada waktu *deploy* aplikasi tersebut. Hal ini menjadikan *developer* merasa kurang nyaman dalam pengembangannya. Selain itu, belum adanya *web service* menyebabkan aplikasi diluar W-Simak harus membuat ulang layanan yang sebenarnya sudah ada. Apalagi ketika aplikasi W-Simak diakses oleh banyak pengguna sekaligus, hal ini akan menyebabkan turunnya performa dan kemampuan aplikasi.

Oleh karena itu, teknologi *web service* sangat diperlukan guna fleksibilitas pengembangan aplikasi dan ketangkasan aplikasi dalam mengatasi *request* dalam jumlah besar. *Web service* memiliki berbagai macam standar komunikasi, dan salah satu standar komunikasinya adalah REST. Standar komunikasi REST memiliki beberapa keunggulan yaitu mudah dikembangkan, mudah dipelajari, mudah digunakan, dapat menggunakan XML atau JSON sebagai format pertukaran data, tidak bergantung pada aplikasi pihak ketiga, dan tidak memerlukan lapisan pertukaran data. *Web service* yang dibuat nantinya akan diimplementasikan pada sistem informasi UIIPerkuliahan.

Pada penelitian ini akan membahas pengembangan RESTful API yang digunakan untuk mengolah nilai mahasiswa, presensi mahasiswa, dan pencetakan berkas perkuliahan. Di mana RESTful API terbagi menjadi tiga *service* yaitu *service presence*, *service assessment*, dan *service simak*.

Kata kunci: sistem perkuliahan, sistem presensi, sistem nilai, sistem *microservice*, RESTful API.



## GLOSARIUM

<i>Framework</i>	kerangka kerja atau lingkungan pengerjaan yang digunakan selama pengembangan sistem.
<i>Debug</i>	langkah untuk menelusuri kesalahan kode program.
<i>Back end</i>	proses yang berjalan dibelakang antarmuka
<i>Front end</i>	interaksi antara pengguna dan sistem
<i>Coding</i>	proses merealisasikan perancangan ke dalam baris kode program
<i>Debugging</i>	proses mencari kesalahan sistem
<i>Deploy</i>	proses menyimpan kode sumber sistem ke <i>repository online</i>
<i>Repository</i>	tempat penyimpanan kode program
<i>Request</i>	pesan yang dikirim ke <i>web service</i> untuk mendapatkan data
<i>Response</i>	balasan dari <i>request client</i>
<i>Microservice</i>	konsep pemecahan layanan sesuai dengan kebutuhan
<i>Web Service</i>	sekumpulan layanan yang dapat diakses oleh berbagai piranti
<i>REST</i>	standar komunikasi antar <i>platform</i> bahasa pemrograman
<i>RESTful API</i>	standar REST yang diimplementasikan pada <i>web service</i>
<i>Plain text</i>	suatu jenis berkas komputer yang berupa teks tidak terformat
<i>Endpoint</i>	url yang digunakan untuk mengakses suatu layanan

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	viii
GLOSARIUM.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR .....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Metodologi Penelitian .....	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	5
2.1 Sistem Informasi.....	5
2.2 RESTful API .....	5
2.3 <i>Web Service</i> .....	7
2.4 <i>Microservice</i> .....	8
2.5 <i>Black-box Testing</i> .....	9
2.6 Lumen PHP <i>Micro Framework</i> .....	9
2.7 <i>Database Oracle</i> .....	10
2.8 Percona XtraDB Cluster .....	11
2.9 Scrum.....	12
2.9.1 <i>Sprint Planning</i> .....	13
2.9.2 <i>Daily Scrum</i> .....	13
2.9.3 <i>Sprint Review</i> .....	13

2.9.4 <i>Sprint Retrospective</i> .....	14
2.10 Penelitian Sebelumnya .....	14
<b>BAB III ANALISIS DAN PERANCANGAN</b> .....	<b>16</b>
3.1 <i>Requirements Phase</i> .....	16
3.1.1 Analisis Masalah .....	16
3.1.2 Usulan Penyelesaian Masalah .....	16
3.1.3 Identifikasi Pengguna .....	16
3.1.4 Kebutuhan Proses .....	17
3.1.5 Kebutuhan Perangkat Lunak .....	18
3.1.6 Kebutuhan Perangkat Keras .....	18
3.2 <i>Analysis Phase</i> .....	19
3.2.1 Rancangan <i>Database</i> Sebelumnya .....	19
3.2.2 Rancangan <i>Service RESTful API</i> .....	27
3.2.3 Rancangan <i>Database</i> .....	28
3.2.4 Rancangan Pengujian .....	28
3.3 <i>Design Phase</i> .....	28
3.3.1 Desain Use Case Diagram .....	28
3.3.2 Desain <i>Database</i> .....	29
3.3.3 Desain <i>Product Backlog Scrum</i> .....	41
3.3.4 Desain <i>Service Simak</i> .....	41
3.3.5 Desain <i>Service Presence</i> .....	63
3.3.6 Desain <i>Service Assessment</i> .....	84
3.3.7 Desain Pengujian .....	95
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM</b> .....	<b>101</b>
4.1 <i>Coding Phase</i> .....	101
4.1.1 <i>Database</i> .....	101
4.1.2 Membuat <i>Controller</i> .....	101
4.1.3 Membuat <i>Endpoint</i> .....	105
4.2 <i>Test Phase</i> .....	107
4.2.1 <i>Testing Service Assessment</i> .....	108
4.2.2 <i>Testing Service Presence</i> .....	119
4.3 <i>Deployment Phase</i> .....	133
4.3.1 Antarmuka <i>Input Nilai</i> .....	137
4.3.2 Antarmuka <i>Riwayat Nilai</i> .....	138

4.3.3 Antarmuka Rekap Nilai .....	139
4.3.4 Antarmuka Presensi Harian .....	139
4.3.5 Antarmuka Laporan Presensi .....	140
4.3.6 Antarmuka <i>Import</i> Presensi .....	141
4.3.7 Antarmuka Penyerahan Nilai .....	141
4.3.8 Antarmuka Cetak Berkas .....	142
4.4 Pengujian Sistem .....	143
4.4.1 Pengujian bagian Presensi.....	143
4.4.2 Pengujian bagian Nilai .....	145
4.4.3 Pengujian bagian Penyerahan Nilai .....	147
4.4.4 Pengujian bagian Cetak Berkas.....	147
BAB V KESIMPULAN DAN SARAN .....	149
5.1 Kesimpulan.....	149
5.2 Saran .....	149
DAFTAR PUSTAKA .....	150

## DAFTAR TABEL

Tabel 2.1 Perbedaan SOAP dan REST .....	7
Tabel 2.2 Rangkuman Penelitian Sebelumnya .....	14
Tabel 3.1 Tabel TM_DETAIL_KRS .....	19
Tabel 3.2 Tabel TM_PRESENSI.....	20
Tabel 3.3 Tabel TM_DETAIL_PRESENSI .....	22
Tabel 3.4 Tabel TM_MAHASISWA.....	23
Tabel 3.5 Tabel TM_KELAS .....	24
Tabel 3.6 Tabel TM_MASTER_KRS .....	25
Tabel 3.7 Tabel TM_MATAKULIAH .....	25
Tabel 3.8 Tabel TM_DOSEN .....	26
Tabel 3.9 Tabel TR_FAKULTAS .....	26
Tabel 3.10 Tabel TR_JURUSAN .....	27
Tabel 3.11 Tabel TR_NILAI .....	27
Tabel 3.12 Tabel Nilai .....	30
Tabel 3.13 Tabel Riwayat Nilai .....	30
Tabel 3.14 Tabel Presensi .....	31
Tabel 3.15 Tabel Pertemuan .....	31
Tabel 3.16 Tabel Organisasi .....	32
Tabel 3.17 Tabel Tahun Akademik .....	33
Tabel 3.18 Tabel Dosen .....	33
Tabel 3.19 Tabel Matakuliah Ditawarkan .....	34
Tabel 3.20 Tabel Matakuliah .....	35
Tabel 3.21 Tabel Kelas .....	36
Tabel 3.22 Tabel Jadwal Reguler Union Blok.....	37
Tabel 3.23 Tabel Mahasiswa .....	38
Tabel 3.24 Tabel Ras Mahasiswa .....	39
Tabel 3.25 Tabel Ras Mahasiswa Detail.....	40
Tabel 3.26 Tabel Kurikulum.....	40
Tabel 3.27 <i>Product Backlog</i> .....	41
Tabel 3.28 Rancangan <i>Request</i> dan <i>Response</i> Kurikulum.....	42
Tabel 3.29 Rancangan Request dan Response Matakuliah .....	43
Tabel 3.30 <i>Request</i> dan <i>Response</i> Kelas.....	44
Tabel 3.31 Rancangan <i>Request</i> dan <i>Response</i> Dosen.....	44

Tabel 3.32 Rancangan <i>Request</i> dan <i>Response</i> Detail Kelas bagian Nilai.....	45
Tabel 3.33 Rancangan <i>Request</i> dan <i>Response</i> Daftar Mahasiswa bagian Nilai .....	46
Tabel 3.34 Rancangan <i>Request</i> dan <i>Response</i> Riwayat Nilai.....	47
Tabel 3.35 Rancangan <i>Request</i> dan <i>Response</i> Rekap Nilai.....	48
Tabel 3.36 Rancangan <i>Request</i> dan <i>Response</i> Penyerahan Nilai .....	49
Tabel 3.37 Rancangan <i>Request</i> dan <i>Response</i> Store Nilai.....	50
Tabel 3.38 Rancangan <i>Request</i> dan <i>Response</i> Store Pertemuan.....	50
Tabel 3.39 Rancangan <i>Request</i> dan <i>Response</i> Detail Kelas bagian Presensi .....	51
Tabel 3.40 Rancangan <i>Request</i> dan <i>Response</i> Detail Kelas Gabungan.....	51
Tabel 3.41 Rancangan <i>Request</i> dan <i>Response</i> Jumlah Pertemuan .....	52
Tabel 3.42 Rancangan <i>Request</i> dan <i>Response</i> Materi .....	52
Tabel 3.43 <i>Request</i> dan <i>Response</i> Daftar Mahasiswa bagian Presensi.....	53
Tabel 3.44 Rancangan <i>Request</i> dan <i>Response</i> Laporan Presensi .....	54
Tabel 3.45 Rancangan <i>Request</i> dan <i>Response</i> Store Presensi .....	55
Tabel 3.46 Rancangan <i>Request</i> dan <i>Response</i> Store Nilai dari <i>File</i> .....	56
Tabel 3.47 Rancangan <i>Request</i> dan <i>Response</i> Store Presensi dari <i>File</i> .....	57
Tabel 3.48 Rancangan <i>Request</i> dan <i>Response</i> Store Pertemuan dari <i>File</i> .....	58
Tabel 3.49 Rancangan <i>Request</i> dan <i>Response</i> Cetak Presensi Dosen .....	59
Tabel 3.50 Rancangan <i>Request</i> dan <i>Response</i> Cetak Persentase Kehadiran Kuliah .....	60
Tabel 3.51 Rancangan <i>Request</i> dan <i>Response</i> Cetak Beban Mengajar Dosen .....	61
Tabel 3.52 Rancangan <i>Request</i> dan <i>Response</i> Cetak Kartu Ujian.....	62
Tabel 3.53 Rancangan <i>Request</i> dan <i>Response</i> Cetak Presensi Ujian.....	63
Tabel 3.54 Rancangan <i>Request</i> dan <i>Response</i> Periode.....	64
Tabel 3.55 Rancangan <i>Request</i> dan <i>Response</i> Program Studi.....	65
Tabel 3.56 Rancangan <i>Request</i> dan <i>Response</i> Kurikulum.....	66
Tabel 3.57 Rancangan <i>Request</i> dan <i>Response</i> Matakuliah.....	67
Tabel 3.58 Rancangan <i>Request</i> dan <i>Response</i> Kelas .....	68
Tabel 3.59 Rancangan <i>Request</i> dan <i>Response</i> Detail Kelas Presensi .....	69
Tabel 3.60 Rancangan <i>Request</i> dan <i>Response</i> Detail Kelas Gabungan.....	70
Tabel 3.61 Rancangan <i>Request</i> dan <i>Response</i> Jumlah Pertemuan .....	70
Tabel 3.62 Rancangan <i>Request</i> dan <i>Response</i> Materi .....	71
Tabel 3.63 Rancangan <i>Request</i> dan <i>Response</i> Daftar Mahasiswa dengan Presensi .....	72
Tabel 3.64 Rancangan <i>Request</i> dan <i>Response</i> Laporan Presensi .....	73
Tabel 3.65 Rancangan <i>Request</i> dan <i>Response</i> Store Presensi .....	74

Tabel 3. 66 Rancangan <i>Request</i> dan <i>Response Store</i> Presensi melalui File .....	75
Tabel 3.67 Rancangan <i>Request</i> dan <i>Response Store</i> Pertemuan melalui <i>File</i> .....	76
Tabel 3.68 Rancangan <i>Request</i> dan <i>Response</i> Cetak Presensi Dosen .....	77
Tabel 3.69 Rancangan <i>Request</i> dan <i>Response</i> Cetak Kehadiran Kuliah .....	78
Tabel 3.70 Rancangan <i>Request</i> dan <i>Response</i> Cetak Kartu Ujian.....	80
Tabel 3.71 Rancangan <i>Request</i> dan <i>Response</i> Cetak Rekapitulasi Beban Mengajar Dosen ...	81
Tabel 3.72 Rancangan <i>Request</i> dan <i>Response</i> Cetak Presensi Ujian.....	82
Tabel 3.73 Rancangan <i>Request</i> dan <i>Response</i> Cetak Daftar IPK Mahasiswa .....	83
Tabel 3.74 Rancangan <i>Request</i> dan <i>Response</i> Cetak Daftar IPS Mahasiswa.....	84
Tabel 3.75 Rancangan <i>Request</i> dan <i>Response</i> Periode .....	85
Tabel 3.76 Rancangan <i>Request</i> dan <i>Response</i> Program Studi.....	86
Tabel 3.77 Rancangan <i>Request</i> dan <i>Response</i> Kurikulum.....	87
Tabel 3.78 Rancangan <i>Request</i> dan <i>Response</i> Matakuliah.....	88
Tabel 3.79 Rancangan <i>Request</i> dan <i>Response</i> Kelas .....	89
Tabel 3.80 Rancangan <i>Request</i> dan <i>Response</i> Detail Kelas Nilai .....	90
Tabel 3.81 Rancangan <i>Request</i> dan <i>Response</i> Daftar Mahasiswa dengan Nilai .....	91
Tabel 3.82 Rancangan <i>Request</i> dan <i>Response</i> Riwayat Nilai.....	92
Tabel 3.83 Rancangan <i>Request</i> dan <i>Response</i> Rekap Nilai .....	93
Tabel 3.84 Rancangan <i>Request</i> dan <i>Response</i> Penyerahan Nilai .....	94
Tabel 3.85 Rancangan <i>Request</i> dan <i>Response Store</i> Nilai.....	95
Tabel 3.86 Rancangan Skenario Pengujian Presensi .....	96
Tabel 3.87 Rancangan Skenario Pengujian Nilai .....	98
Tabel 3.88 Rancangan Skenario Pengujian Penyerahan Nilai.....	99
Tabel 3.89 Rancangan Skenarion Pengujian Cetak Berkas .....	100
Tabel 4.1 Skenario Pengujian Presensi .....	144
Tabel 4.2 Skenario Pengujian Nilai .....	146
Tabel 4.3 Skenario Pengujian Penyerahan Nilai.....	147
Tabel 4.4 Skenario Pengujian Cetak Berkas.....	148

## DAFTAR GAMBAR

Gambar 2.1 Arsitektur RESTful API.....	6
Gambar 2.2 Ilustrasi kegunaan <i>web service</i> .....	8
Gambar 2.3 Perbedaan arsitektur <i>monolithic</i> dan <i>microservice</i> .....	9
Gambar 2.4 Arsitektur Percona XtraDB Cluster .....	11
Gambar 2.5 <i>Scrum Methodology</i> Sumber: (Digite, 2020).....	13
Gambar 3.1 Use Case Diagram UIIPerkuliahan .....	29
Gambar 4.1 <i>Endpoint service presence</i> .....	105
Gambar 4.2 <i>Endpoint Service Assessment</i> .....	106
Gambar 4.3 <i>Endpoint Service Simak</i> .....	107
Gambar 4.4 <i>Response endpoint</i> periode <i>success</i> .....	108
Gambar 4.5 <i>Response endpoint</i> periode ketika data tidak ditemukan .....	109
Gambar 4.6 <i>Response endpoint</i> program studi <i>success</i> .....	109
Gambar 4.7 <i>Response endpoint</i> program studi ketika pengguna tidak memiliki otoritas .....	110
Gambar 4.8 <i>Response endpoint</i> kurikulum <i>success</i> .....	110
Gambar 4.9 <i>Response endpoint</i> kurikulum ketika data tidak ditemukan.....	110
Gambar 4.10 <i>Response endpoint</i> matakuliah <i>success</i> sebagai petugas.....	111
Gambar 4.11 <i>Response endpoint</i> matakuliah <i>success</i> sebagai dosen .....	112
Gambar 4.12 <i>Response endpoint</i> matakuliah ketika data tidak ditemukan .....	112
Gambar 4.13 <i>Response endpoint</i> kelas <i>success</i> .....	113
Gambar 4.14 <i>Response endpoint</i> kelas ketika data tidak ditemukan.....	113
Gambar 4.15 <i>Response endpoint</i> detail kelas <i>success</i> .....	114
Gambar 4.16 <i>Response endpoint</i> detail kelas ketika kelas tidak ada.....	114
Gambar 4.17 <i>Response endpoint</i> nilai mahasiswa <i>success</i> .....	115
Gambar 4.18 <i>Response endpoint</i> nilai mahasiswa ketika data tidak ditemukan.....	115
Gambar 4.19 <i>Response endpoint</i> riwayat nilai <i>success</i> .....	116
Gambar 4.20 <i>Response endpoint</i> riwayat nilai ketika belum memiliki riwayat .....	116
Gambar 4.21 <i>Response endpoint</i> rekap nilai <i>success</i> .....	117
Gambar 4.22 <i>Response endpoint</i> rekap nilai ketika belum ada data.....	117
Gambar 4.23 <i>Response endpoint</i> store nilai ketika nilai tidak sesuai ketentuan .....	118
Gambar 4.24 <i>Response endpoint</i> store nilai <i>success</i> .....	118
Gambar 4.25 <i>Response endpoint</i> store nilai ketika tidak terjadi perubahan .....	118
Gambar 4.26 <i>Response endpoint</i> penyerahan nilai <i>success</i> .....	119
Gambar 4.27 <i>Response endpoint</i> penyerahan nilai ketika data tidak ditemukan.....	119



Gambar 4.28 <i>Response endpoint</i> jumlah pertemuan <i>success</i> .....	120
Gambar 4.29 <i>Response endpoint</i> jumlah pertemuan ketika belum ada pertemuan .....	120
Gambar 4.30 <i>Response endpoint</i> materi <i>success</i> .....	120
Gambar 4.31 <i>Response endpoint</i> materi ketika belum ada pertemuan .....	120
Gambar 4.32 <i>Response endpoint</i> detail kelas <i>success</i> .....	121
Gambar 4.33 <i>Response endpoint</i> detail kelas <i>fail</i> .....	121
Gambar 4.34 <i>Response endpoint</i> presensi mahasiswa <i>success</i> .....	122
Gambar 4.35 <i>Response endpoint</i> presensi mahasiswa <i>fail</i> .....	122
Gambar 4.36 <i>Response endpoint</i> store presensi <i>success</i> .....	123
Gambar 4.37 <i>Response endpoint</i> store presensi <i>fail</i> .....	123
Gambar 4.38 <i>Response endpoint</i> laporan presensi <i>success</i> .....	123
Gambar 4.39 <i>Response endpoint</i> laporan presensi <i>fail</i> .....	124
Gambar 4.40 <i>Response endpoint</i> import pertemuan <i>success</i> .....	124
Gambar 4.41 <i>Response endpoint</i> import pertemuan <i>fail</i> .....	124
Gambar 4.42 <i>Response endpoint</i> import presensi <i>success</i> .....	125
Gambar 4.43 <i>Response endpoint</i> import presensi <i>fail</i> .....	125
Gambar 4.44 <i>Response endpoint</i> dosen <i>success</i> .....	125
Gambar 4.45 <i>Response endpoint</i> dosen <i>fail</i> .....	125
Gambar 4.46 <i>Response endpoint</i> fakultas <i>success</i> .....	126
Gambar 4.47 <i>Response endpoint</i> fakultas <i>fail</i> .....	126
Gambar 4.48 <i>Response endpoint</i> cetak presensi dosen <i>success</i> .....	127
Gambar 4.49 <i>Response endpoint</i> cetak presensi dosen <i>fail</i> .....	127
Gambar 4.50 <i>Response endpoint</i> cetak kehadiran kuliah <i>success</i> .....	128
Gambar 4.51 <i>Response endpoint</i> cetak kehadiran kuliah <i>fail</i> .....	128
Gambar 4.52 <i>Response endpoint</i> cetak kartu ujian <i>success</i> .....	129
Gambar 4.53 <i>Response endpoint</i> cetak kartu ujian <i>fail</i> .....	129
Gambar 4. 54 <i>Response endpoint</i> cetak rekapitulasi beban mengajar dosen <i>success</i> .....	130
Gambar 4.55 <i>Response endpoint</i> cetak rekapitulasi beban mengajar dosen <i>fail</i> .....	130
Gambar 4.56 <i>Response endpoint</i> cetak presensi ujian per kelas <i>success</i> .....	131
Gambar 4.57 <i>Response endpoint</i> cetak presensi ujian gabungan <i>success</i> .....	131
Gambar 4.58 <i>Response endpoint</i> presensi ujian <i>fail</i> .....	131
Gambar 4.59 <i>Response endpoint</i> cetak daftar mahasiswa dengan IPK <i>success</i> .....	132
Gambar 4.60 <i>Response endpoint</i> cetak daftar mahasiswa dengan IPK <i>fail</i> .....	132
Gambar 4.61 <i>Response endpoint</i> cetak daftar mahasiswa dengan IPS <i>success</i> .....	133

Gambar 4.62 <i>Response endpoint</i> cetak daftar mahasiswa dengan IPS <i>fail</i> .....	133
Gambar 4.63 Arsitektur otentikasi di UIIGateway .....	134
Gambar 4.64 <i>Flow</i> mengambil data.....	135
Gambar 4.65 <i>Flow</i> menyimpan data.....	136
Gambar 4.66 Antarmuka <i>Input</i> Nilai .....	138
Gambar 4.67 Antarmuka Riwayat Nilai .....	139
Gambar 4.68 Antarmuka Rekap Nilai .....	139
Gambar 4.69 Antarmuka Presensi Harian .....	140
Gambar 4.70 Antarmuka Laporan Presensi .....	140
Gambar 4.71 Antarmuka Import Presensi dan Pertemuan.....	141
Gambar 4.72 Antarmuka Penyerahan Nilai .....	142
Gambar 4.73 Antarmuka Cetak Berkas .....	143

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Pendidikan merupakan hak setiap warga negara Indonesia. Hal ini tertuang dalam Pasal 31 ayat 1 UUD 1945 yang menyatakan bahwa “Setiap warga negara berhak mendapatkan Pendidikan”. Salah satu jenjang pendidikan yaitu pendidikan tinggi. Didalam Pasal 20 ayat 4 Peraturan Menteri Riset, Teknologi, dan Pendidikan Tinggi No.44 tahun 2015 menyatakan bahwa “Prinsip objektif sebagaimana dimaksud pada ayat 1 merupakan penilaian yang didasarkan pada standar yang disepakati antara dosen dan mahasiswa serta bebas dari pengaruh subjektivitas penilai dan yang dinilai.” Peraturan ini menunjukkan pentingnya standar penilaian yang diatur oleh institusi pendidikan tinggi. Universitas Islam Indonesia yang selanjutnya disebut UII, merupakan institusi pendidikan tinggi yang mengatur standar penilaian mahasiswanya. Hal ini tertuang didalam Pasal 1 ayat 38, Pasal 31 ayat 3, Pasal 33 ayat 2, dan Pasal 39 Peraturan UII No.2 tahun 2017.

Sebagai universitas swasta yang terus mengikuti perkembangan teknologi, UII selalu berusaha untuk meningkatkan kualitas pelayanan. Salah satu bentuk peningkatan layanan adalah dengan mengembangkan teknologi yang mampu meringkas dan memenuhi kebutuhan proses bisnis di lingkungan UII. Salah satu kebutuhan UII yang harus dipenuhi berkaitan dengan penilaian dan presensi mahasiswa.

Selama ini, UII menggunakan aplikasi berbasis desktop bernama “W-Simak” untuk mengolah nilai dan presensi mahasiswa. Aplikasi ini tidak hanya mengolah nilai dan presensi mahasiswa, tetapi juga mengolah penjadwalan kelas dan masa *key-in* mahasiswa. Akibatnya, kompleksitas antar tabel menjadi sangat tinggi. Aplikasi ini juga menggunakan basis data Oracle yang merupakan basis data berbayar dan memerlukan biaya cukup besar. Selain itu, masih adanya proses yang didefinisikan secara langsung pada basis data menyebabkan sulitnya memodifikasi tabel basis data karena keterkaitan antar tabel yang cukup tinggi. Aplikasi ini juga dibuat menggunakan teknologi yang sudah usang. Sehingga *maintenance* dan modifikasi kedepannya akan menjadi sulit.

Aplikasi W-Simak saat ini belum menggunakan *web service*. Ketika tidak menggunakan *web service* akan memiliki kekurangan dan kelemahan meliputi tidak dapat melakukan pertukaran data antar aplikasi, sulit untuk mengembangkan fitur tertentu karena keterkaitan antara layanan yang cukup tinggi sehingga jika melakukan pengembangan salah satu layanan

akan mempengaruhi aplikasi keseluruhan, fitur yang ada pada aplikasi W-Simak tidak dapat digunakan oleh aplikasi lain.

Oleh karena itu, teknologi *web service* sangat diperlukan guna fleksibilitas pengembangan aplikasi dan ketangkasan aplikasi dalam mengatasi *request* dalam jumlah besar. *Web service* memiliki berbagai macam standar komunikasi, dan salah satu standar komunikasinya adalah REST. Standar komunikasi REST memiliki beberapa keunggulan yaitu mudah dikembangkan, mudah dipelajari, mudah digunakan, dapat menggunakan XML atau JSON sebagai format pertukaran data, tidak bergantung pada aplikasi pihak ketiga, dan tidak memerlukan lapisan pertukaran data. *Web service* yang dibuat nantinya akan diimplementasikan pada sistem informasi UIIPerkuliahan.

Pengembangan sistem informasi UIIPerkuliahan akan dikerjakan secara berkelompok menggunakan kerangka kerja *Scrum* dengan pembagian tugas meliputi perancangan antarmuka, pengembangan *front end* dan pengembangan *back end*. Penelitian ini akan membahas dari sisi *back end* tentang pengembangan RESTful API. Dikarenakan nilai dan presensi mahasiswa sebelumnya tersimpan pada basis data Oracle, maka RESTful API yang dikembangkan akan memiliki kemampuan untuk mengolah dua macam basis data yaitu Oracle (basis data aplikasi W-Simak) dan Mysql (basis data sistem informasi UIIPerkuliahan). RESTful API digunakan karena merupakan standar komunikasi API yang mudah digunakan, dipahami, serta dikembangkan. Metode yang akan digunakan untuk mengembangkan RESTful API sistem informasi UIIPerkuliahan adalah *Web Service Implementation Lifecycle*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah bagaimana membangun RESTful API untuk Sistem Informasi UIIPerkuliahan yang dapat mengolah nilai, presensi, dan mencetak berkas perkuliahan?

## 1.3 Batasan Masalah

Batasan masalah diperlukan untuk membatasi tingkat kompleksitas penelitian, sehingga penelitian ini lebih berfokus pada masalah utama. Berikut merupakan batasan masalah dalam penelitian ini:

- a. RESTful API ini menggunakan data FIAI, FTI dan FMIPA di Universitas Islam Indonesia sebagai sampel.

- b. RESTful API ini hanya dapat diakses oleh petugas dan dosen yang telah diberikan hak akses ke sistem.
- c. Penelitian ini hanya berfokus dalam pembuatan RESTful API untuk “Sistem Informasi UIIPerkuliahan”.
- d. RESTful API meliputi penyimpanan serta pengambilan data yang berkaitan dengan nilai mahasiswa, presensi mahasiswa dan pencetakan berkas perkuliahan.

#### 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah membangun “RESTful API Sistem Informasi UIIPerkuliahan” yang dapat mengolah nilai mahasiswa, presensi mahasiswa dan mencetak berkas-berkas perkuliahan FTI, FIAI dan FMIPA di Universitas Islam Indonesia yang dapat digunakan oleh petugas dan dosen.

#### 1.5 Metodologi Penelitian

Dalam menyelesaikan penelitian ini, diperlukan langkah penyelesaian yang tersusun secara konseptual agar penelitian yang dilakukan mejadi terarah. Penelitian ini akan menggunakan metode *Web Service Implementation Lifecycle* dengan tahapan-tahapan sebagai berikut.

a. *Requirements Phase*

Fase ini bertujuan untuk memahami proses bisnis yang terjadi dan menerjemahkan hal tersebut ke dalam fitur *web service* meliputi fungsionalitas, non-fungsionalitas dan batasan dari kemampuan *web service*.

b. *Analysis Phase*

Pada fase ini kebutuhan *web service* akan diterjemahkan menjadi model-model perancangan yang dapat dipahami secara garis besar.

c. *Design Phase*

Detail rancangan *web service* harus diselesaikan pada fase ini. Hal tersebut meliputi identifikasi bentuk *request* dan *response* serta detail dari basis data yang akan digunakan.

d. *Coding Phase*

Pada tahap ini rancangan yang sudah dibuat akan diimplementasikan kedalam bahasa pemrograman. Pada fase ini juga akan dilakukan proses *coding* dan *debugging*.

e. *Test Phase*

Pada fase ini *web service* akan mengalami pengujian meliputi *functional test*.

f. *Deployment Phase*

Pada fase ini, *web service* yang telah dibuat akan di-*deploy* ke *repository online* sehingga dapat digunakan oleh *client-side* atau disebut juga *front end*. Pada tahap ini *web service* akan diuji fungsionalitasnya dari sisi *front end*.

## 1.6 Sistematika Penulisan

Penelitian ini akan dibagi menjadi lima bagian, setiap bagian dijelaskan sebagai berikut:

a. BAB I PENDAHULUAN

Bagian pendahuluan memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan. Bagian ini akan memuat alasan yang mendasari penelitian ini.

b. BAB II LANDASAN TEORI

Bagian ini berisi teori-teori yang digunakan dalam penelitian ini, serta penelitian sebelumnya.

c. BAB III ANALISIS DAN PERANCANGAN

Bagian ini akan menjelaskan *product backlog* dalam setiap *sprint*, rancangan *service RESTful API*, rancangan *response/request*, *use case diagram*, dan rancangan *database*.

d. BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bagian ini akan memuat hasil implementasi *web service* yang dibuat beserta hasil dari pengujian.

e. BAB V KESIMPULAN DAN SARAN

Bagian ini akan memuat manfaat dari penelitian ini, serta bagian-bagian yang masih bisa ditingkatkan agar tercipta sistem yang lebih baik. Karena tidak ada sistem yang sempurna sejak awal, perlu adanya pengujian dan peningkatan berulang kali sehingga tercipta sistem yang lebih baik.

## BAB II LANDASAN TEORI

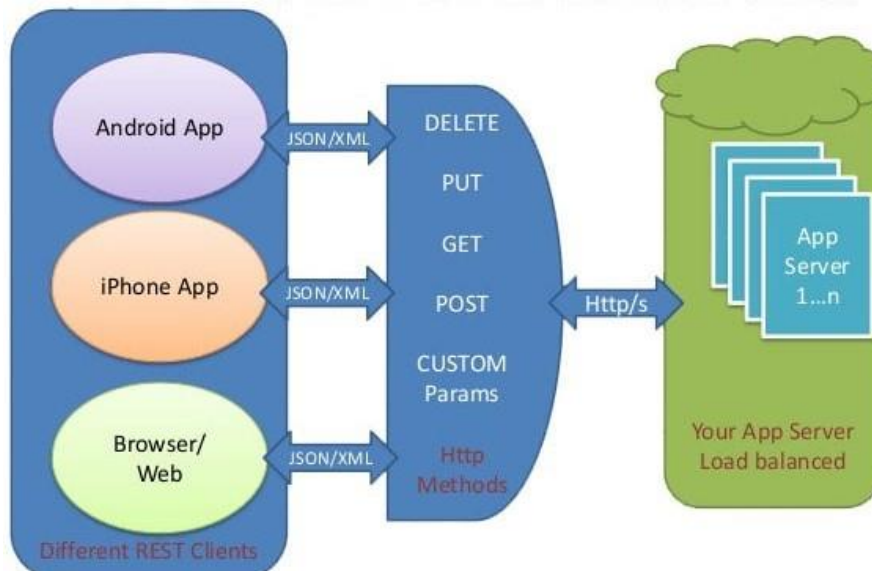
### 2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem yang dibuat untuk memberikan informasi kepada pengguna sehingga bisa membantu pengguna dalam proses pengambilan keputusan maupun mengolah suatu masalah yang sedang dihadapi. Fokus pada sistem informasi ada pada proses memasukan dan melihat data sehingga menjadi informasi yang berguna (Osd, 2020)

### 2.2 RESTful API

Sebelum memahami tentang RESTful API, terlebih dahulu perlu memahami tentang REST dan API. REST (*REpresentational State Transfer*) merupakan **standar komunikasi** yang digunakan dua atau lebih aplikasi untuk saling bertukar data menggunakan HTTP (Feridi, 2020). Sedangkan API (*Application Programming Interface*) merupakan perantara perangkat lunak yang memungkinkan dua atau lebih aplikasi dapat saling berkomunikasi dengan adanya batasan yang ditetapkan oleh pemilik API.

Sebagai perumpamaan, Indonesia terdiri dari berbagai macam suku dan bangsa yang memiliki bahasa berbeda-beda. Agar dapat saling mengerti, tokoh-tokoh terdahulu membuat bahasa persatuan yaitu bahasa Indonesia. REST menyerupai bahasa Indonesia yang menyatukan berbagai macam suku dan bangsa. Sedangkan API merupakan media perantara yang digunakan untuk dapat saling berkomunikasi seperti telepon genggam. Arsitektur dari RESTful API bisa dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur RESTful API

Pada Gambar 2.1, *Http method* serta *JSON/XML* merupakan gambaran dari standar komunikasi REST. Sedangkan *App Server* merupakan API atau media komunikasi yang akan menghubungkan antar aplikasi.

REST dan RESTful API merupakan dua hal yang berbeda. REST merupakan standar komunikasi yang digunakan, sedangkan RESTful API adalah hasil implementasi REST pada API. RESTful API digunakan karena memiliki berbagai macam kelebihan jika dibandingkan dengan standar komunikasi yang lain. Perbedaan antara SOAP dan REST sebagai standar komunikasi API dapat dilihat pada Tabel 2.1.



Tabel 2.1 Perbedaan SOAP dan REST

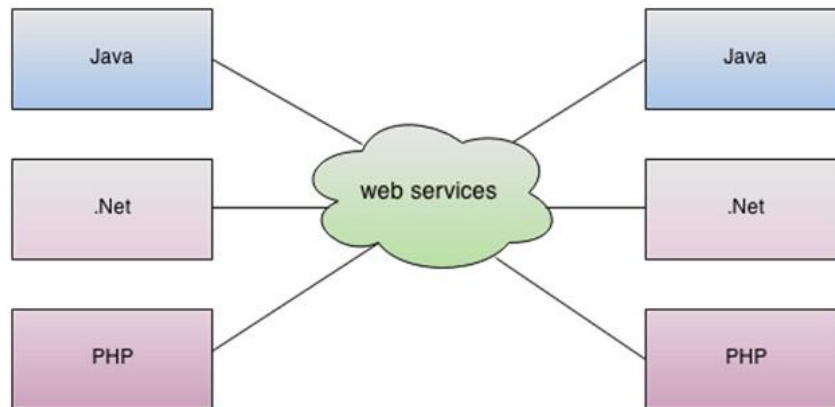
Sumber: (Nurdiyanto, 2020)

Uraian	SOAP	REST
Protokol komunikasi	HTTP, HTTPS, SMTP, FTP.	HTTP, HTTPS.
Penggunaan <i>bandwidth</i>	Dalam jumlah <i>request</i> yang banyak, relatif boros <i>bandwidth</i> . Hal ini karena banyaknya markup dalam penulisan format XML.	Relatif hemat <i>bandwidth</i> , karena markup-markup ekstra seperti pada XML tidak dipakai.
Tren penggunaan	Banyak mulai beralih ke REST, meski masih tetap ada yang mempertahankan, misalnya untuk integrasi aplikasi ke sistem legasi pada sebuah perusahaan.	Mulai populer, banyak dipakai oleh penyedia <i>web service</i> terkemuka, seperti twitter, yahoo!, flickr, bloglines, technorati, google, amazon, eBay, dsb.
Aturan penulisan	Ketat, mengikuti spesifikasi XML (SOAP v1.2).	Tidak ada spesifikasi khusus.
Format respon	XML dengan spesifikasi SOAP. Agak sulit bagi kita untuk membaca langsung dan memahaminya.	XML, JSON, atau format <i>plain text</i> lainnya. Hal ini memudahkan penerima respon membaca dan memahaminya.
<i>Attachment file</i>	Bisa (karena dapat mengembalikan respon dalam format <i>binary</i> )	Tidak bisa
Sifat <i>web service</i> pada umumnya	Tertutup, lebih ditujukan untuk vendor atau perusahaan tertentu	Terbuka, bisa diakses siapa saja
<i>Caching web</i>	Relatif sulit	Mudah, karena menggunakan URI
Penggunaan standar	Standar lama (XML, HTTP) dan baru (SOAP) digunakan bersamaan	Standar yang sudah ada, seperti XML dan HTTP
<i>Tool</i> pengembangan	Banyak, baik komersial maupun <i>opensource</i>	Beberapa, karena tidak begitu dibutuhkan
<i>Tool</i> manajemen	Perlu, bahkan kadang harganya mahal	Menggunakan <i>tool</i> yang sudah ada pada sistem jaringan
Kemudahan implementasi	Mudah jika kita sudah memiliki lingkungan berbasis SOAP	Mudah

### 2.3 Web Service

Pada tahun 1999 *web service* digagas oleh perusahaan ternama *Hewlett-Packard's e-Speak* dan mulai di populerkan oleh *Microsoft* pada tahun 2000. *Web service* dapat didefinisikan sebagai perangkat lunak yang mendukung interaksi antar dua aplikasi atau lebih melalui jaringan. *Web service* juga sering dianggap sebagai fasilitas yang disediakan oleh suatu perusahaan, organisasi atau lembaga agar aplikasi dari dalam maupun luar dapat memanfaatkan data-data yang mereka miliki. Sebagai contoh, sebuah maskapai penerbangan menyediakan

*web service* agar aplikasi lain dapat mengetahui harga tiket pesawat maskapainya. Ilustrasi dari penggunaan *web service* dapat dilihat pada Gambar 2.2.

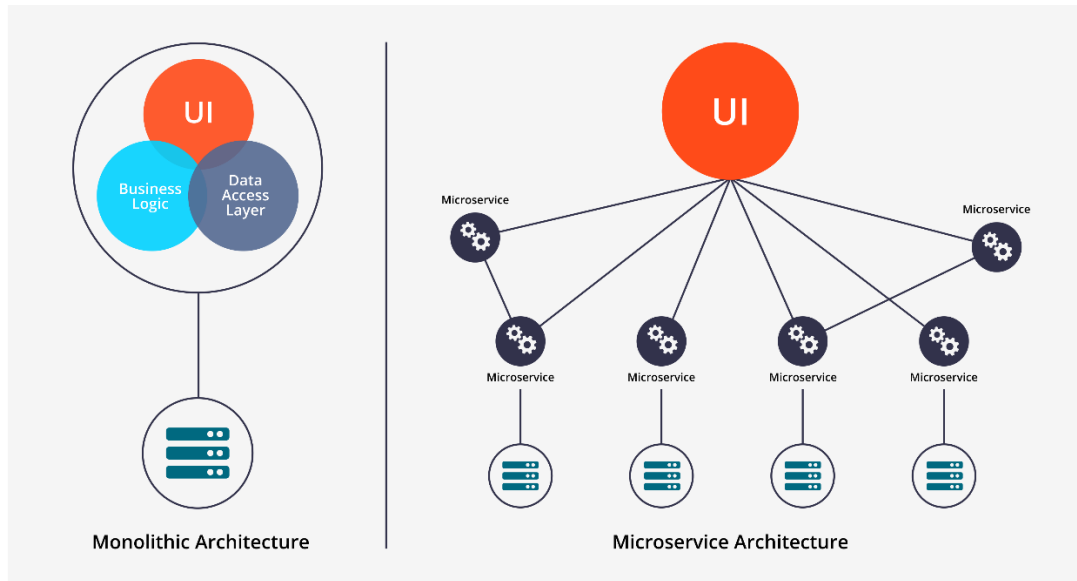


Gambar 2.2 Ilustrasi kegunaan *web service*

Pada dasarnya *web service* dan API memiliki kesamaan secara fungsi, yang membedakan ialah ada tidaknya perantara jaringan sebagai media penggunaannya. API bisa digunakan tanpa melalui perantara jaringan, sedangkan *web service* mengharuskan adanya jaringan agar dapat saling bertukar informasi. Umumnya *web service* digunakan bersamaan dengan *website*. Sebagai bagian dari *website*, *web service* memiliki peran yang cukup penting untuk memberikan layanan terbaik kepada pengguna.

#### 2.4 *Microservice*

*Microservice* merupakan arsitektur yang biasa digunakan dalam pengembangan sistem informasi. Arsitektur ini menekankan pada pemecahan layanan sistem menjadi beberapa *service* sesuai dengan lingkup kebutuhannya. Di mana ketika terjadi masalah pada salah satu *service*, tidak mempengaruhi *service* yang lain. Selain *microservice*, arsitektur sistem informasi lainnya adalah *monolithic*. Perbedaan arsitektur *microservice* dan *monolithic* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Perbedaan arsitektur *monolithic* dan *microservice*

Arsitektur *microservice* merupakan arsitektur yang lebih terukur dan fleksibel. Pada arsitektur ini, sistem informasi dirancang agar menjadi sistem yang terdistribusi dan menghasilkan layanan yang spesifik dan fokus. Permasalahan utama akan dipecah dan dibuat solusi untuk setiap pecahan tersebut yang akan disusun dalam satu *service*. Di mana setiap *service* memiliki tanggung jawabnya sendiri. Dengan pendekatan ini, suatu sistem informasi akan terdiri dari beberapa *service* yang dapat dikelola dan didistribusikan secara *independent*, hal ini akan memudahkan sistem untuk beradaptasi terhadap perubahan.

## 2.5 *Black-box Testing*

*Black-box testing* disebut juga *behavioral testing*, merupakan pengujian sistem yang dilakukan tanpa mengetahui struktur, proses, implementasi yang terjadi di belakang sistem. Sehingga hanya perlu fokus pada apa yang dilihat penguji, tujuannya untuk mencari kesalahan yang bisa terlihat oleh mata (Software Testing Fundamentals, 2020).

## 2.6 *Lumen PHP Micro Framework*

Lumen merupakan *Micro Framework* berbasis PHP yang dibuat oleh *Taylor Otwell* yang juga merupakan pembuat Laravel. Lumen dibuat untuk memenuhi kebutuhan *developer* yang ingin membuat aplikasi dengan skala lebih kecil dari Laravel. Secara garis besar, Lumen dan Laravel tidak berbeda jauh. Hal yang membedakan kedua *framework* ini terletak dari fitur yang disediakan. Laravel memiliki fitur yang lebih lengkap dibandingkan Lumen.

Jika dibandingkan dengan *micro framework* lain yang menggunakan PHP seperti Slim dan Silex, Lumen merupakan *micro framework* yang memiliki kemampuan meng-*handle request* tertinggi yaitu 1900 *request* per detik. Karena kecepatannya, Lumen sering digunakan untuk membuat RESTful API. Lumen juga menyediakan fitur-fitur yang memudahkan pekerjaan *developer*, berikut fitur-fitur yang disediakan Lumen.

- a. *Caching*, mendukung berbagai sistem *caching* seperti penggunaan *Database*, *Redis*, atau *Memcached*.
- b. *Queues*, memungkinkan Lumen untuk menunda pemrosesan tugas. Sering digunakan untuk mengirim pesan ke berbagai pengguna sekaligus melalui *email* ataupun proses yang membutuhkan banyak waktu.
- c. *Validation*, dapat memvalidasi data yang masuk. Seperti harus berupa angka, atau harus memuat parameter tertentu.
- d. *Routing*, memudahkan *developer* untuk membuat *endpoint* sesuai dengan keinginan *developer*. Serta menghubungkan endpoint tersebut dengan *method* yang ingin diakses.
- e. *Middleware*, mekanisme yang membantu *developer* untuk menetapkan syarat-syarat yang harus dipenuhi *client* sebelum dapat mengakses suatu layanan.

## 2.7 Database Oracle

*Database Oracle* merupakan RDBMS (Relational Database Management System) yang sudah ada sejak tahun 1979. Hingga saat ini, basis data ini telah mengalami pembaharuan berulang kali dan versi terbarunya ialah *Oracle Database 20c*. Meskipun begitu basis data yang saat ini digunakan oleh Badan Sistem Informasi UII ialah *Oracle Database 11g*. Perusahaan besar dan ternama cenderung menggunakan basis data ini, karena berbagai macam kelebihan yang ditawarkan. Adapun kelebihan basis data Oracle sebagai berikut:

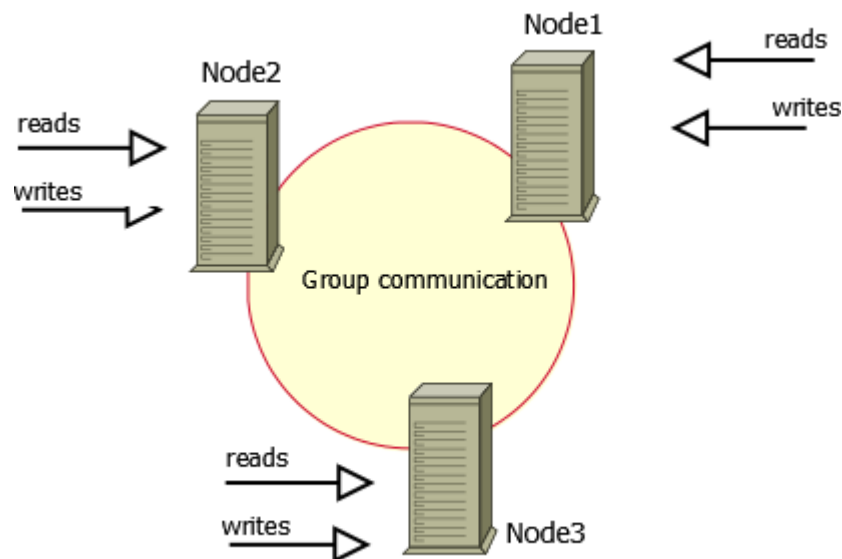
- a. Memiliki kemampuan yang baik untuk melakukan manajemen basis data.
- b. Mampu mengatasi jumlah data yang sangat besar.
- c. Dapat mengolah data dengan cepat dan akurat.
- d. Memiliki kemampuan untuk mengolah data lebih dari satu server (*cluster server*).
- e. Dapat membatasi hak akses yang dimiliki oleh pengguna dari sisi server.
- f. Dapat digunakan oleh berbagai macam *platform*.
- g. Pemrosesan data yang cepat.
- h. Memiliki kemampuan flashback.

Selain kelebihan yang sudah dipaparkan, basis data ini tetap tidak bisa lepas dari kekurangan sebagai berikut.

- Harga software DBMS yang sangat mahal dibandingkan DBMS lain.
- Spesifikasi minimum hardware yang tinggi.
- Rumit dalam mengaplikasikannya.

## 2.8 Percona XtraDB Cluster

Percona XtraDB Cluster adalah teknologi yang dapat mengelompokkan basis data MySQL. Terdiri dari dua atau lebih basis data yang saling terhubung dan disebut *node*. Teknologi ini menjamin ketersediaan penyimpanan data, mencegah *downtime* dan kehilangan data, serta memberikan skalabilitas linier untuk basis data yang terus berkembang. Arsitektur dari Percona XtraDB Cluster dapat dilihat pada Gambar 2.4.



Gambar 2.4 Arsitektur Percona XtraDB Cluster

Sumber: (Percona XtraDB Cluster, 2020)

Adapun fitur yang disediakan Percona XtraDB Cluster sebagai berikut.

- Synchronous replication*: Data ditulis ke semua *node* secara bersamaan, atau tidak ditulis sama sekali jika gagal.
- Multi-master replication*: Setiap *node* dapat menggunakan *trigger* untuk memperbarui data.
- True parallel replication*: Replikasi data dilakukan menggunakan *multi thread*.

- d. *Automatic node provisioning*: *Node* yang baru ditambahkan akan otomatis tersinkronisasi.
- e. *Data consistency*: Tidak ada lagi *node* yang tidak tersinkron.

Adapun kelebihan Percona XtraDB Cluster sebagai berikut.

- a. Saat mengeksekusi *query*, *query* tersebut akan dieksekusi secara lokal di *node* yang sedang digunakan.
- b. Tidak ada control manajemen terpusat, sehingga *node* yang sudah tidak digunakan bisa dihilangkan dan tidak mengganggu fungsi *node* yang lain.
- c. Solusi yang bagus untuk memaksimalkan kecepatan membaca data.

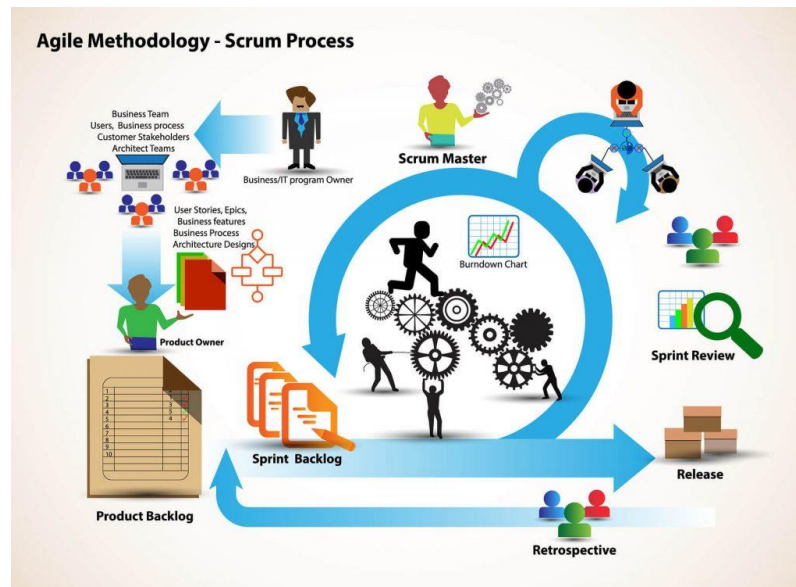
Adapun kekurangan Percona XtraDB Cluster sebagai berikut.

- a. Ketika menambahkan *node* baru, harus menyalin seluruh data dari salah satu *node* yang ada.
- b. Penulisan harus dilakukan pada semua *node*.
- c. Adanya duplikasi data, jika ada *node* maka akan ada 3 duplikasi.

## 2.9 Scrum

Scrum adalah salah satu kerangka kerja dengan menggunakan prinsip-prinsip pendekatan *agile*, yang bertumpu pada kekuatan kolaborasi anggota tim, *incremental product* dan proses iterasi untuk mewujudkan hasil akhir (Kompasiana, 2019).

Scrum bukanlah sebuah teknik, metodologi, ataupun proses. Scrum merupakan kerangka kerja tim di mana pengguna bisa menggunakan berbagai macam teknik dan mengharuskan untuk transparan terhadap hal-hal yang dikerjakan, sehingga terjadi peningkatan kerja secara terus-menerus (Schwaber & Sutherland, 2020). Alur kerja scrum dapat dilihat pada Gambar 2.5.



Gambar 2.5 Scrum Methodology

Sumber: (Digite, 2020)

Seperti yang terlihat pada gambar Gambar 2.5. Alur kerja scrum terbagi menjadi beberapa tahapan, antara lain:

### 2.9.1 *Sprint Planning*

Dalam tahap ini, *product owner* (PO) akan bertemu dengan *stake holder* untuk merumuskan manfaat yang ingin didapatkan dengan adanya produk keluaran. Setelah itu, PO akan merumuskan nilai-nilai tersebut ke dalam *product backlog*. *Product Backlog* ini didiskusikan antara PO dan tim pengembang. Diskusi antara PO dan tim pengembang ini, bertujuan untuk menentukan bobot dari setiap *backlog* dan *backlog* yang akan dikerjakan selama satu *sprint*.

### 2.9.2 *Daily Scrum*

Dalam tahap ini, tim pengembang akan menyediakan waktu paling lama 15 menit dalam sehari untuk menceritakan hal-hal yang telah dilakukan dan akan dilakukan dalam kurung waktu 24 jam. Hal ini dilakukan agar adanya transparansi dan pengawasan antar anggota tim pengembang.

### 2.9.3 *Sprint Review*

*Sprint Review* akan dilakukan setiap menyelesaikan sebuah *sprint*. Dalam tahap ini, tim pengembang harus mempresentasikan pencapaian selama satu *sprint* meliputi *product backlog* yang telah selesai dan belum kepada PO. Kemudian dilakukan penghitungan bobot yang berhasil diselesaikan selama *sprint* tersebut. Bobot ini akan menjadi acuan untuk *sprint*

selanjutnya agar *backlog* selama satu *sprint* tidak melebihi beban yang bisa ditanggung oleh tim pengembang.

#### 2.9.4 *Sprint Retrospective*

*Sprint Retrospective* dilakukan sebagai kesempatan untuk melakukan introspeksi di dalam *Scrum Team* dan membuat perencanaan yang dapat meningkatkan kualitas maupun kinerja tim di *sprint* berikutnya.

#### 2.10 Penelitian Sebelumnya

Dalam penelitian ini, akan membahas pengembangan RESTful API untuk pencetakan berkas perkuliahan, pengelolaan nilai dan presensi mahasiswa. Sebelumnya sudah ada beberapa penelitian yang menggunakan RESTful API untuk menyelesaikan berbagai permasalahan. Adapun penelitian sebelumnya dapat dirangkum seperti pada Tabel 2.2.

Tabel 2.2 Rangkuman Penelitian Sebelumnya

Pengarang	Judul	Hasil
(Nurdiyah & Handayani, 2018)	Restful <i>Web Service</i> Sistem Presensi Mahasiswa (Studi Kasus Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang)	Penelitian ini menghasilkan sistem presensi berbasis web dan android. Kedua sistem tersebut dapat saling berkomunikasi dengan memanfaatkan RESTful API. RESTful API digunakan agar sistem presensi yang berjalan di <i>platform</i> berbasis web dan <i>platform</i> berbasis android dapat saling bertukar data presensi. Metode yang digunakan pada penelitian ini ialah metode <i>prototyping</i> . Penelitian ini dilatar belakangi oleh presensi mahasiswa yang masih bersifat manual. RESTful API yang dibuat hanya terdiri dari sebuah <i>service</i> dan menggunakan basis data MySQL.



Pengarang	Judul	Hasil
(Kurniawan, Oslan, & Kristanto, 2013)	Implementasi REST - API untuk Portal Akademik UKDW Berbasis Android	Penelitian ini menghasilkan aplikasi berbasis android yang menggunakan RESTful API. RESTful API digunakan karena sebelumnya sudah ada portal sejenis yang berbasis <i>website</i> sehingga agar proses yang dilakukan melalui aplikasi android dapat terbaca juga pada <i>website</i> , perlu adanya media yang menghubungkan data dari kedua aplikasi tersebut. Penelitian ini dilatar belakangi oleh belum adanya portal akademik berbasis android di Universitas Kristen Duta Indonesia. RESTful API yang dibuat hanya terdiri dari sebuah <i>service</i> dan menggunakan basis data MySQL.
(Akbar, 2018)	Pengembangan RESTful API untuk <i>Application Specific High Level Location Service</i>	Penelitian ini menghasilkan RESTful API yang dapat digunakan sebagai layanan penyedia informasi area abstraksi level tinggi sehingga dapat menyesuaikan kebutuhan spesifik aplikasi LBS ( <i>Location Based Service</i> ). Penelitian ini dilatar belakangi oleh semakin banyaknya aplikasi yang memanfaatkan layanan lokasi dengan kebutuhan berbeda-beda. RESTful API yang dibuat hanya terdiri dari sebuah <i>service</i> dan menggunakan basis data MySQL.

Pada penelitian sebelumnya *RESTful API* dibuat dalam satu *service* yang terpusat dan hanya menggunakan satu jenis basis data. Sedangkan pada penelitian ini, RESTful API akan dibuat menjadi beberapa layanan yang terdistribusi dan menggunakan dua macam basis data.

## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1 *Requirements Phase*

##### 3.1.1 Analisis Masalah

Berdasarkan wawancara yang dilakukan kepada pihak yang mengerti keadaan aplikasi W-Simak, diketahui bahwa aplikasi “W-Simak” belum menggunakan *web service* yang menyebabkan hal-hal sebagai berikut.

- a. Sulitnya pemindahan layanan W-Simak ke media lain seperti *website* atau perangkat *mobile*.
- b. Pengembangan salah satu layanan akan mempengaruhi keseluruhan aplikasi.
- c. Menurunnya performa dan kemampuan aplikasi ketika diakses oleh banyak pengguna.

##### 3.1.2 Usulan Penyelesaian Masalah

Sejalan dengan masalah yang ditemukan, maka perlu adanya RESTful API yang dapat menyelesaikan masalah-masalah tersebut serta dapat diimplementasikan pada sistem informasi UIIPerkuliahan. RESTful API dipilih karena memiliki kelebihan jika dibandingkan dengan standar pertukaran data yang lain yaitu mudah untuk dikembangkan, mudah untuk dipelajari, tidak bergantung pada *tool*, tidak memerlukan lapisan pertukaran data, dan secara desain mudah untuk digunakan. RESTful API akan memenuhi ketentuan sebagai berikut.

- a. RESTful API dapat digunakan sebagai media pertukaran data antar aplikasi atau antar *service*.
- b. RESTful API dapat dikembangkan secara terpisah tanpa mempengaruhi fitur yang sudah ada.
- c. RESTful API dapat digunakan oleh berbagai macam aplikasi.

##### 3.1.3 Identifikasi Pengguna

RESTful API yang telah dibuat akan diimplementasikan pada sistem informasi UIIPerkuliahan yang dibuat oleh Badan Sistem Informasi UII. Di mana sistem informasi ini memiliki dua pengguna, berikut penjelasan setiap pengguna:

- a. Dosen

Dosen merupakan karyawan Universitas Islam Indonesia yang tercatat dalam *database* sebagai dosen dan memiliki tugas untuk mengampu satu atau lebih matakuliah.

- b. Petugas/Admin

Petugas/*Admin* merupakan karyawan Universitas Islam Indonesia yang diberikan amanah untuk mengolah nilai, presensi mahasiswa dan pencetakan berkas perkuliahan. Serta telah terdaftar sebagai *admin* pada sistem informasi UIIPerkuliahan.

### 3.1.4 Kebutuhan Proses

Penelitian ini hanya akan berfokus pada pengembangan RESTful API yang dapat memenuhi kebutuhan Sistem Informasi UIIPerkuliahan. Sehingga diperlukan penjabaran proses yang akan dipenuhi, sehingga pengembangan RESTful API akan menjadi lebih terarah. Terdapat beberapa proses yang ingin dipenuhi dalam penelitian ini. Adapun proses–proses tersebut adalah sebagai berikut:

a. Proses Mengolah Nilai

Proses ini bertujuan agar pengguna dapat memasukkan dan mengubah nilai mahasiswa. Khusus dosen hanya dapat mengubah nilai mahasiswa pada matakuliah yang diampu atau pernah diampu.

b. Proses Mengolah Presensi

Proses ini bertujuan agar pengguna dapat memasukkan dan mengubah status presensi mahasiswa dalam setiap pertemuan. Khusus dosen hanya dapat mengubah presensi mahasiswa pada matakuliah yang diampu atau pernah diampu

c. Proses Mencetak Berkas

Proses ini hanya dapat digunakan oleh petugas, proses ini bertujuan agar petugas dapat mencetak berkas-berkas perkuliahan antara lain presensi mahasiswa, persentase kehadiran kuliah mahasiswa, persentase kehadiran kuliah setiap pertemuan, kartu ujian, presensi ujian, daftar mahasiswa dengan indeks prestasi semester, daftar mahasiswa dengan indeks prestasi kumulatif, rekapitulasi beban mengajar dosen, dan persentase kehadiran dosen.

d. Proses Melihat Penyerahan Nilai

Petugas dapat melihat daftar kelas beserta waktu ujian, waktu penyerahan nilai dan jeda waktu antara ujian dan penyerahan nilai setiap kelas pada periode dan kurikulum tertentu.

e. Proses Melihat Rekap Nilai

Petugas dan dosen dapat melihat rekapitulasi perolehan nilai mahasiswa dalam satu kelas.

f. Proses Melihat Laporan Presensi

Petugas dan dosen dapat melihat persentase kehadiran mahasiswa dalam satu kelas selama perkuliahan berlangsung.

g. Proses *Import* Presensi

Petugas dapat memasukkan dan mengubah presensi mahasiswa menggunakan *file* dengan ekstensi \*.dbf maupun \*.xlsx.

#### h. Proses *Import* Nilai

Petugas dan dosen dapat memasukkan ataupun mengubah nilai mahasiswa menggunakan *file* dengan ekstensi \*.dbf maupun \*.xlsx.

### 3.1.5 Kebutuhan Perangkat Lunak

Berikut di bawah ini merupakan daftar perangkat lunak yang digunakan dalam penelitian:

- a. Sistem Operasi Windows 10 sebagai lingkungan kerja sistem yang digunakan.
- b. Sublime dan Visual Studio Code sebagai *text editor* yang digunakan selama pengembangan.
- c. PHP 7.2.19 sebagai bahasa pemrograman yang digunakan untuk membuat RESTful API.
- d. Lumen 5.8.12 sebagai *framework* PHP untuk mengembangkan RESTful API.
- e. Navicat, Sql Developer, dan Mysql Workbench sebagai *Database Management System* yang digunakan untuk mengolah data yang dibutuhkan.
- f. Git Bash, aplikasi yang digunakan untuk merekam semua perubahan *code* selama proses pengembangan.
- g. Git Kraken, aplikasi yang digunakan untuk men-*deploy* kode sumber ke repositori *online*.
- h. Gitlab, repositori *online* kode sumber RESTful API.
- i. Postman, aplikasi yang digunakan untuk menguji RESTful API yang dibuat.
- j. Google Chrome, aplikasi *browser* yang digunakan untuk melihat hasil implementasi RESTful API pada *web application*.

### 3.1.6 Kebutuhan Perangkat Keras

Adapun perangkat keras yang digunakan pada penelitian ini adalah sebuah laptop dengan spesifikasi sebagai berikut:

- a. Processor AMD FX-7600P
- b. RAM 8 GB
- c. Harddisk SSD 163 GB

## 3.2 Analysis Phase

### 3.2.1 Rancangan Database Sebelumnya

Pada aplikasi W-Simak, basis data yang digunakan adalah Oracle. Sedangkan Sistem Informasi yang ingin dibangun akan menggunakan basis data Mysql. Hal ini dikarenakan basis data MySQL memiliki beberapa kelebihan dibandingkan dengan basis data Oracle. Kelebihan Mysql antara lain *multi-platform*, *open source*, gratis, mudah digunakan, tipe data yang bervariasi, memiliki *tools* yang cukup lengkap, serta dapat di *install* pada komputer yang tidak memiliki spesifikasi tinggi. Meskipun begitu, RESTful API UIIPerkuliahan tetap memerlukan data terdahulu yang tersimpan dalam basis data Oracle, sehingga perlu dilakukan pemetaan terkait basis data Oracle yang memiliki keterhubungan dengan Sistem Informasi UIIPerkuliahan. Adapun basis data Oracle yang memiliki keterhubungan dengan Sistem Informasi UIIPerkuliahan sebagai berikut:

a. Tabel TM\_DETAIL\_KRS

Tabel ini menyimpan hasil *key-in* mahasiswa pada setiap semester beserta nilai yang didapatkan. Adapun struktur tabel dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel TM\_DETAIL\_KRS

Name	Type	Length	Scale	Not Null	Key
TH_AKADEMI	VARCHAR2	9	0	Yes	Primary
SMT	VARCHAR2	1	0	Yes	Primary
KD_JURUSAN	VARCHAR2	3	0	Yes	
KD_KURIKULUM	VARCHAR2	4	0	Yes	Primary
NO_MK	VARCHAR2	8	0	Yes	Primary
KELAS	VARCHAR2	1	0	No	
NO_MHS	VARCHAR2	8	0	Yes	Primary
NILAI	VARCHAR2	3	0	No	
FLAG_STATUS_A MBIL	VARCHAR2	1	0	Yes	
FLAG_POSTED	VARCHAR2	1	0	Yes	
TGL_INPUT	DATE	7	0	No	
USER_INPUT	VARCHAR2	25	0	No	

Name	Type	Length	Scale	Not Null	Key
ROLE ID	NUMBER	10	0	No	
TGL_UPDATE	DATE	7	0	No	
USER_UPDATE	VARCHAR2	25	0	No	
ID_DETAIL_KRS	NUMBER	38	0	No	

b. Tabel TM\_PRESENSI

Tabel ini menyimpan detail pertemuan setiap kelas, seperti materi pertemuan dan total pertemuan yang telah dilaksanakan. Adapun struktur tabel dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel TM\_PRESENSI

Name	Type	Length	Scale	Not Null	Key
TH_AKADEMI	VARCHAR2	9	0	Yes	Primary
SMT	VARCHAR2	1	0	Yes	Primary
KD_JURUSAN	VARCHAR2	3	0	Yes	Primary
KD_KURIKULUM	VARCHAR2	4	0	Yes	Primary
NO_MK	VARCHAR2	8	0	Yes	Primary
KELAS	VARCHAR2	1	0	Yes	Primary
KD_DOSEN	VARCHAR2	9	0	Yes	Primary
MATERI1	VARCHAR2	65	0	No	
MATERI2	VARCHAR2	65	0	No	
MATERI3	VARCHAR2	65	0	No	
MATERI4	VARCHAR2	65	0	No	
MATERI5	VARCHAR2	65	0	No	
MATERI6	VARCHAR2	65	0	No	
MATERI7	VARCHAR2	65	0	No	
MATERI8	VARCHAR2	65	0	No	
MATERI9	VARCHAR2	65	0	No	
MATERI10	VARCHAR2	65	0	No	

<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Scale</b>	<b>Not Null</b>	<b>Key</b>
MATERI11	VARCHAR2	65	0	No	
MATERI12	VARCHAR2	65	0	No	
MATERI13	VARCHAR2	65	0	No	
MATERI14	VARCHAR2	65	0	No	
MATERI15	VARCHAR2	65	0	No	
MATERI16	VARCHAR2	65	0	No	
MATERI17	VARCHAR2	65	0	No	
MATERI18	VARCHAR2	65	0	No	
MATERI19	VARCHAR2	65	0	No	
MATERI20	VARCHAR2	65	0	No	
MATERI21	VARCHAR2	65	0	No	
MATERI22	VARCHAR2	65	0	No	
MATERI23	VARCHAR2	65	0	No	
MATERI24	VARCHAR2	65	0	No	
MATERI25	VARCHAR2	65	0	No	
MATERI26	VARCHAR2	65	0	No	
MATERI27	VARCHAR2	65	0	No	
MATERI28	VARCHAR2	65	0	No	
MATERI29	VARCHAR2	65	0	No	
MATERI30	VARCHAR2	65	0	No	
TOTAL	NUMBER	2	0	Yes	

## c. Tabel TM\_DETAIL\_PRESENSI

Tabel ini menyimpan presensi kehadiran mahasiswa pada setiap pertemuan. Adapun struktur tabel dapat dilihat pada Tabel 3.3.

Tabel 3.3 Tabel TM\_DETAIL\_PRESENSI

Name	Type	Length	Scale	Not Null	Key
TH_AKADEMI	VARCHAR2	9	0	Yes	Primary
SMT	VARCHAR2	1	0	Yes	Primary
KD_JURUSAN	VARCHAR2	3	0	Yes	Primary
KD_KURIKULUM	VARCHAR2	4	0	Yes	Primary
NO_MK	VARCHAR2	8	0	Yes	Primary
KELAS	VARCHAR2	1	0	Yes	Primary
KD_DOSEN	VARCHAR2	9	0	No	
NO_MHS	VARCHAR2	15	0	Yes	Primary
HADIR1	VARCHAR2	1	0	No	
HADIR2	VARCHAR2	1	0	No	
HADIR3	VARCHAR2	1	0	No	
HADIR4	VARCHAR2	1	0	No	
HADIR5	VARCHAR2	1	0	No	
HADIR6	VARCHAR2	1	0	No	
HADIR7	VARCHAR2	1	0	No	
HADIR8	VARCHAR2	1	0	No	
HADIR9	VARCHAR2	1	0	No	
HADIR10	VARCHAR2	1	0	No	
HADIR11	VARCHAR2	1	0	No	
HADIR12	VARCHAR2	1	0	No	
HADIR13	VARCHAR2	1	0	No	
HADIR14	VARCHAR2	1	0	No	
HADIR15	VARCHAR2	1	0	No	
HADIR16	VARCHAR2	1	0	No	



Name	Type	Length	Scale	Not Null	Key
HADIR17	VARCHAR2	1	0	No	
HADIR18	VARCHAR2	1	0	No	
HADIR19	VARCHAR2	1	0	No	
HADIR20	VARCHAR2	1	0	No	
HADIR21	VARCHAR2	1	0	No	
HADIR22	VARCHAR2	1	0	No	
HADIR23	VARCHAR2	1	0	No	
HADIR24	VARCHAR2	1	0	No	
HADIR25	VARCHAR2	1	0	No	
HADIR26	VARCHAR2	1	0	No	
HADIR27	VARCHAR2	1	0	No	
HADIR28	VARCHAR2	1	0	No	
HADIR29	VARCHAR2	1	0	No	
HADIR30	VARCHAR2	1	0	No	
TOTAL	NUMBER	5	2	Yes	

d. Tabel TM\_MAHASISWA

Tabel ini menyimpan biodata dari mahasiswa seperti nim dan nama mahasiswa. Adapun struktur tabel dapat dilihat pada Tabel 3.4.

Tabel 3.4 Tabel TM\_MAHASISWA

Name	Type	Length	Scale	Not Null	Key
NO_MHS	VARCHAR2	15	0	Yes	Primary
TAHUN	VARCHAR2	4	0	No	
NAMA_MHS	VARCHAR2	60	0	Yes	
KD_FAK	VARCHAR2	2	0	No	
KD_JURUSAN	VARCHAR2	3	0	No	
STATUS	CHAR	1	0	Yes	
KD_DOSEN	VARCHAR2	9	0	Yes	

## e. Tabel TM\_KELAS

Tabel ini menyimpan data yang berkaitan dengan kelas, seperti kuota kelas, jumlah mahasiswa dan dosen yang mengampu kelas tersebut. Adapun struktur tabel dapat dilihat pada Tabel 3.5.

Tabel 3.5 Tabel TM\_KELAS

Name	Type	Length	Scale	Not Null	Key
TH_AKADEMI	VARCHAR2	9	0	Yes	Primary
SMT	VARCHAR2	1	0	Yes	Primary
KD_FAK	VARCHAR2	2	0	Yes	
KD_DOSEN	VARCHAR2	9	0	Yes	
KD_KURIKULUM	VARCHAR2	4	0	Yes	Primary
KD_JURUSAN	VARCHAR2	3	0	Yes	Primary
NO_MK	VARCHAR2	8	0	Yes	Primary
KELAS	VARCHAR2	1	0	Yes	Primary
JML_MAHASISWA	NUMBER	4	0	Yes	
KUOTA	NUMBER	4	0	Yes	
JML_PERTEMUAN	NUMBER	2	0	Yes	

## f. Tabel TM\_MASTER\_KRS

Tabel ini menyimpan rekap hasil perhitungan nilai mahasiswa setiap semester, seperti indeks prestasi kumulatif, indeks prestasi semester, sks kumulatif, sks semester, sks yang diambil dan sks jatah yang dapat diambil. Adapun struktur tabel dapat dilihat pada Tabel 3.6.

Tabel 3.6 Tabel TM\_MASTER\_KRS

Name	Type	Length	Scale	Not Null	Key
TH_AKADEMI	VARCHAR2	9	0	Yes	Primary
SMT	VARCHAR2	1	0	Yes	Primary
NO_MHS	VARCHAR2	8	0	Yes	Primary
KD_DOSEN	VARCHAR2	9	0	No	
SKS_JATAH	NUMBER	2	0	Yes	
IP_SMT	NUMBER	3	2	Yes	
SKS_AMBIL	NUMBER	3	0	Yes	
SKS_JATAH_YAD	NUMBER	2	0	Yes	
KD_JURUSAN	VARCHAR2	3	0	No	
IPK	NUMBER	6	3	Yes	
SKS_KUM	NUMBER	3	0	Yes	

## g. Tabel TM\_MATAKULIAH

Tabel ini menyimpan data-data yang berkaitan dengan matakuliah seperti nama matakuliah, total sks, kode matakuliah dan kode jurusan. Adapun struktur tabel dapat dilihat pada Tabel 3.7.

Tabel 3.7 Tabel TM\_MATAKULIAH

Name	Type	Length	Scale	Not Null	Key
KD_KURIKULUM	VARCHAR2	4	0	Yes	Primary
KD_JURUSAN	VARCHAR2	3	0	Yes	Primary
NO_MK	VARCHAR2	8	0	Yes	Primary
NAMA_MK	VARCHAR2	50	0	No	
SKS	NUMBER	2	0	Yes	

## h. Tabel TM\_DOSEN

Tabel ini menyimpan data-data yang berhubungan dengan dosen, seperti kode dosen, nama dosen, dan kode jurusan. Adapun struktur tabel dapat dilihat pada Tabel 3.8.

Tabel 3.8 Tabel TM\_DOSEN

Name	Type	Length	Scale	Not Null	Key
KD_DOSEN	VARCHAR2	9	0	Yes	Primary
NAMA	VARCHAR2	40	0	Yes	
KD_FAK	VARCHAR2	2	0	No	
KD_JURUSAN	VARCHAR2	3	0	No	
GELAR_DEPAN	VARCHAR2	30	0	No	
GELAR_BELAKANG	VARCHAR2	30	0	No	

## i. Tabel TR\_FAKULTAS

Tabel ini menyimpan data yang berkaitan dengan fakultas, seperti kode fakultas dan nama fakultas. Adapun struktur tabel dapat dilihat pada Tabel 3.9.

Tabel 3.9 Tabel TR\_FAKULTAS

Name	Type	Length	Scale	Not Null	Key
KD_FAK	VARCHAR2	2	0	Yes	Primary
FAKULTAS	VARCHAR2	50	0	Yes	
NAMA_SINGKAT	VARCHAR2	4	0	Yes	
KD_DEKAN	VARCHAR2	9	0	Yes	
MAX_SKS_DOSEN	NUMBER	2	0	Yes	
NAMA_INGGRIS	VARCHAR2	50	0	No	

## j. Tabel TR\_JURUSAN

Tabel ini menyimpan data yang berkaitan dengan jurusan, seperti kode jurusan, dan nama jurusan. Adapun struktur tabel dapat dilihat pada Tabel 3.10.

Tabel 3.10 Tabel TR\_JURUSAN

Name	Type	Length	Scale	Not Null	Key
KD_FAK	VARCHAR2	2	0	Yes	
KD_JURUSAN	VARCHAR2	3	0	Yes	Primary
NAMA_JURUSAN	VARCHAR2	50	0	Yes	
NAMA_SINGKAT	VARCHAR2	6	0	Yes	
KD_KAJUR	VARCHAR2	9	0	No	
KD_SEKJUR	VARCHAR2	9	0	No	
NAMA_INGGRIS	VARCHAR2	50	0	No	
JENJANG	VARCHAR2	3	0	No	

## k. Tabel TR\_NILAI

Tabel ini menyimpan bobot atau nilai angka dari nilai huruf yang didapatkan. Adapun struktur tabel dapat dilihat pada Tabel 3.11.

Tabel 3.11 Tabel TR\_NILAI

Name	Type	Length	Scale	Not Null	Key
KD_JURUSAN	VARCHAR2	3	0	Yes	Primary
NILAI	VARCHAR2	3	0	Yes	Primary
NILAI_ANGKA	NUMBER	3	2	Yes	

### 3.2.2 Rancangan Service RESTful API

Dalam penelitian ini, basis data yang digunakan terbagi menjadi dua yaitu Mysql dan Oracle. Sebagian sistem di Universitas Islam Indonesia menggunakan basis data Oracle sebagai sumber data dan sebagian lagi menggunakan Mysql, hal ini menyebabkan setiap penambahan ataupun perubahan nilai dan presensi harus dilakukan ke dalam dua basis data yaitu Oracle dan Mysql. Untuk mengatasi kondisi tersebut maka perlu dibuat tiga buah *service* dengan rincian sebagai berikut:

a. *Service Assessment*

*Service* ini berfungsi untuk mengolah setiap *request* dan *response* yang berkaitan dengan pengelolaan nilai mahasiswa. *Service* ini akan terhubung dengan *database* Mysql.

b. *Service Presence*

*Service* ini berfungsi untuk mengolah setiap *request* dan *response* yang berkaitan dengan pengelolaan presensi mahasiswa dan pencetakan berkas perkuliahan. *Service* akan terhubung dengan *database* Mysql.

c. *Service Simak*

*Service* ini berfungsi untuk mengolah setiap *request* dan *response* yang membutuhkan data dari basis data Oracle.

### 3.2.3 Rancangan Database

Dalam pengembangan RESTful API, terlebih dahulu perlu merancang basis data yang akan digunakan. Dalam penelitian ini, basis data akan menggunakan konsep *non-relational database* pada basis data Mysql. Sehingga, setiap tabel dalam basis data tidak memiliki relasi yang didefinisikan antar tabel. Tabel utama yang akan digunakan untuk menyimpan nilai dan presensi mahasiswa adalah tabel nilai, riwayat\_nilai, pertemuan dan presensi. Sedangkan sebagian tabel akan diambil dari tabel yang sudah tersedia.

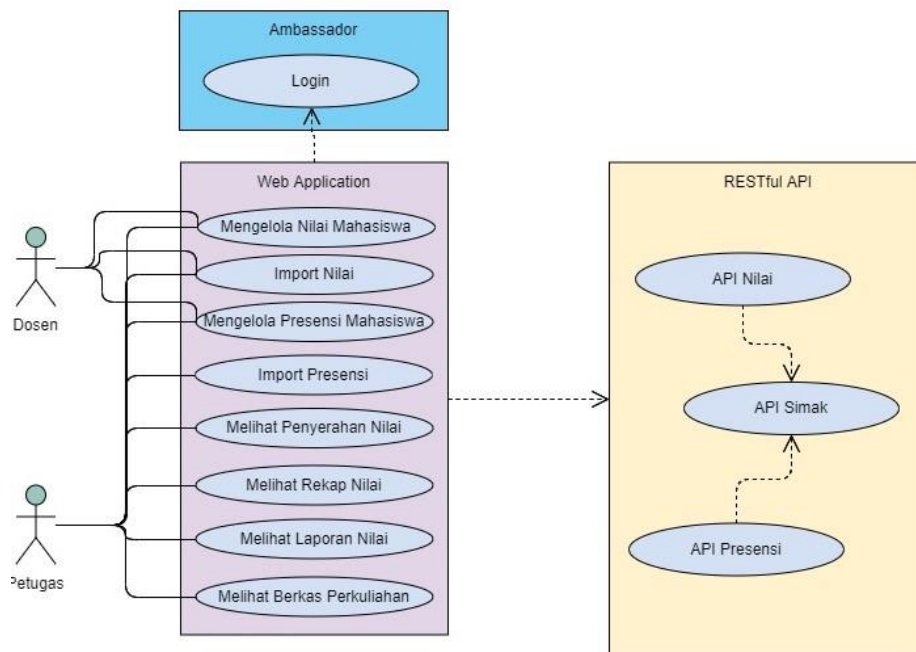
### 3.2.4 Rancangan Pengujian

Pengujian akan dilakukan dua kali, yaitu menguji fungsionalitas RESTful API menggunakan aplikasi Postman dan menguji hasil implementasi pada *web application* menggunakan metode *black-box testing*.

## 3.3 Design Phase

### 3.3.1 Desain Use Case Diagram

*Use Case Diagram* digunakan untuk memudahkan pemahaman interaksi antara *web application* dan RESTful API yang akan dibuat. Adapun interaksi antara *web application* dan RESTful API ditunjukkan pada Gambar 3.1.



Gambar 3.1 Use Case Diagram UIIPerkuliahan

Seperti yang terlihat pada Gambar 3.1, ada dua aktor yang dapat menggunakan sistem informasi UIIPerkuliahan. *Web application* akan berhubungan dengan sistem otentikasi yang telah dibuat oleh Badan Sistem Informasi UII atau disebut *ambassador*. Jika pengguna berhasil melalui proses otentikasi, maka pengguna dapat menggunakan RESTful API sesuai dengan otoritas yang telah diberikan. Penerjemahan otoritas akan dilakukan dari sisi *web application*.

### 3.3.2 Desain Database

Dalam pengembangan RESTful API, terlebih dahulu perlu merancang basis data yang akan digunakan. Dalam penelitian ini, basis data akan menggunakan konsep *non-relational database*. Sehingga, setiap tabel dalam basis data tidak memiliki relasi. Akan tetapi, hubungan antara tabel dibuat melalui fitur bawaan yang telah disediakan oleh *framework* Lumen. Adapun rancangan tabel yang digunakan sebagai berikut:

a. Tabel Nilai

Tabel ini digunakan untuk menyimpan nilai mahasiswa pada setiap kelas. Adapun struktur dari tabel ditunjukkan pada Tabel 3.12.

Tabel 3.12 Tabel Nilai

<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Decimals</b>	<b>Not Null</b>
id	Big Int	20	0	Yes
uuid	Varchar	128	0	Yes
id_kelas	Binary	16	0	Yes
nim	Varchar	20	0	Yes
nilai	Varchar	5	0	No
tgl_input	Date Time	0	0	No
user_input	Varchar	15	0	No
tgl_update	Date Time	0	0	No
user_update	Varchar	15	0	No

## b. Tabel Riwayat Nilai

Tabel ini digunakan untuk menyimpan setiap perubahan nilai yang terjadi pada tabel nilai. Adapun struktur tabel ditunjukkan pada Tabel 3.13.

Tabel 3.13 Tabel Riwayat Nilai

<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Decimals</b>	<b>Not Null</b>
id	BigInt	20	0	Yes
uuid	Varchar	128	0	Yes
nilai	Varchar	3	0	No
tgl_input	DateTime	0	0	Yes
user_input	Varchar	15	0	Yes
id_nilai	BigInt	20	0	Yes

## c. Tabel Presensi

Tabel ini digunakan untuk menyimpan setiap presensi mahasiswa di berbagai kelas. Adapun struktur tabel ditunjukkan pada Tabel 3.14.



Tabel 3.14 Tabel Presensi

Name	Type	Length	Decimals	Not Null
id	Big Int	20	0	Yes
uuid	Varchar	128	0	Yes
nim	Varchar	20	0	No
status	Varchar	20	0	No
id_pertemuan	Big Int	20	0	No
tgl_input	Date Time	0	0	No
user_input	Varchar	15	0	No
user_update	Varchar	15	0	No
tgl_update	Date Time	0	0	No

## d. Tabel Pertemuan

Tabel ini digunakan untuk menyimpan detail pertemuan setiap kelas meliputi total mahasiswa hadir dan materi. Adapun struktur tabel ditunjukkan pada Tabel 3.15.

Tabel 3.15 Tabel Pertemuan

Name	Type	Length	Decimals	Not Null
id	Big Int	20	0	Yes
uuid	Varchar	128	0	Yes
pertemuan_ke	Int	2	0	No
materi	Varchar	255	0	No
user_input	Varchar	15	0	No
tgl_input	Date Time	15	0	No
user_update	Varchar	0	0	No
tgl_update	Date Time	15	0	No
id_kelas	Binary	16	0	No
total_hadir	Big Int	4	0	No

e. Tabel Organisasi

Tabel ini merupakan basis data yang menyimpan data setiap organisasi di UII meliputi Fakultas dan Program Studi. Adapun struktur tabel ditunjukkan pada Tabel 3.16.

Tabel 3.16 Tabel Organisasi

<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Decimals</b>	<b>Not Null</b>
id	BigInt	20	0	Yes
id_tingkat	BigInt	20	0	No
id_jenjang	BigInt	20	0	No
kd_organisasi	Varchar	10	0	Yes
induk_organisasi	Varchar	10	0	Yes
nama_organisasi	Varchar	255	0	Yes
nama_tingkat	Varchar	255	0	No
nama_jenjang	Varchar	10	0	No
uuid	Varchar	128	0	Yes

f. Tabel Tahun Akademik

Tabel ini merupakan basis data yang menyimpan seluruh tahun akademik di UII. Adapun struktur tabel ditunjukkan pada Tabel 3.17.

Tabel 3.17 Tabel Tahun Akademik

<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Decimals</b>	<b>Not Null</b>
id	BigInt	20	0	Yes
kd_semester	Int	2	0	Yes
nama_semester	Varchar	255	0	Yes
nama_tahun_akademik	Varchar	10	0	Yes
flag_aktif	Int	1	0	Yes
flag_delete	Int	1	0	Yes
flag_current	Int	1	0	Yes
tgl_update	DateTime	0	0	Yes
uuid	Varchar	128	0	Yes

## g. Tabel Nilai

Tabel ini menyimpan seluruh data dosen di UII. Adapun struktur tabel ditunjukkan pada Tabel 3.18.

Tabel 3.18 Tabel Dosen

<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Decimals</b>	<b>Not Null</b>
id	BigInt	20	0	Yes
id_organisasi	BigInt	20	0	No
kd_dosen	Varchar	10	0	Yes
nama_dosen	Varchar	50	0	Yes
gelar_depan	Varchar	50	0	No
gelar_belakang	Varchar	50	0	No
flag_aktif	Int	1	0	Yes
flag_delete	Int	1	0	Yes
uuid	Varchar	128	0	Yes

## h. Tabel Matakuliah Ditawarkan

Tabel ini menyimpan seluruh data matakuliah yang ditawarkan pada setiap tahun akademik. Adapun struktur tabel ditunjukkan pada Tabel 3.19.

Tabel 3.19 Tabel Matakuliah Ditawarkan

<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Decimals</b>	<b>Not Null</b>
id	Binary	16	0	Yes
id_organisasi	BigInt	20	0	Yes
id_tahun_akademik	BigInt	20	0	No
id_matakuliah	BigInt	20	0	No
id_kurikulum	BigInt	20	0	No
kd_matakuliah	Varchar	10	0	Yes
nama_matakuliah	Varchar	50	0	Yes
sks	Int	1	0	Yes
flag_delete	Int	1	0	No
uuid	Varchar	128	0	Yes

## i. Tabel Matakuliah

Tabel ini menyimpan data seluruh matakuliah di UII. Adapun struktur tabel ditunjukkan pada Tabel 3.20.

Tabel 3.20 Tabel Matakuliah

Name	Type	Length	Decimals	Not Null
id	BigInt	20	0	Yes
id_kurikulum	BigInt	20	0	Yes
kd_prodi	Varchar	20	0	No
kd_matakuliah	varchar	20	0	Yes
nama_matakuliah	varchar	255	0	Yes
sks_tatap_muka	float	20	2	Yes
sks_praktek	float	20	2	Yes
sks_lapangan	float	20	2	Yes
total_sks_matakuliah	float	20	2	Yes
flag_aktif	int	1	0	Yes
flag_delete	int	1	0	Yes
user_input	varchar	15	0	Yes
tgl_input	datetime	0	0	Yes
user_update	varchar	15	0	No
tgl_update	datetime	0	0	No
user_delete	varchar	15	0	No
tgl_delete	datetime	0	0	No
uuid	varchar	128	0	No

## j. Tabel Kelas

Tabel ini menyimpan data yang berkaitan dengan kelas dalam setiap tahun akademik. Adapun struktur tabel ditunjukkan pada Tabel 3.21.

Tabel 3.21 Tabel Kelas

<b>Name</b>	<b>Type</b>	<b>Length</b>	<b>Decimals</b>	<b>Not Null</b>
id	binary	16	0	Yes
id_organisasi	bigint	20	0	Yes
id_tahun_akademik	bigint	20	0	No
id_matakuliah_ditawarkan	binary	16	0	No
id_kurikulum	bigint	20	0	No
nama_kelas	varchar	1	0	Yes
flag_aktif	int	1	0	Yes
user_input	varchar	15	0	Yes
tgl_input	datetime	0	0	Yes
user_update	varchar	15	0	Yes
tgl_update	datetime	0	0	Yes
flag_delete	int	1	0	Yes
user_delete	varchar	15	0	No
tgl_delete	datetime	0	0	No
uuid	varchar	128	0	Yes
tgl_ujian	varchar	45	0	No
tgl_penyerahan	varchar	45	0	No

k. Tabel Jadwal Reguler Union Blok

Tabel ini menyimpan jadwal kuliah dari setiap kelas. Adapun struktur tabel ditunjukkan pada Tabel 3.22.

Tabel 3.22 Tabel Jadwal Reguler Union Blok

Name	Type	Length	Decimals	Not Null
id	bigint	20	0	Yes
id_organisasi	bigint	20	0	No
id_matakuliah	bigint	20	0	No
id_matakuliah_ditawarkan	binary	16	0	No
id_kurikulum	bigint	20	0	No
id_dosen	bigint	20	0	No
id_kelas	binary	16	0	No
id_tahun_akademik	bigint	20	0	No
kd_dosen	varchar	20	0	No
kd_matakuliah	varchar	20	0	No
hari	int	1	0	No
pertemuan_ke	int	1	0	No
nama_ruang	varchar	50	0	No
nama_sesi	varchar	20	0	No
jam_mulai	varchar	10	0	No
jam_selesai	varchar	10	0	No
nama_matakuliah	varchar	100	0	No
sks	float	5	2	No
nama_kelas	varchar	2	0	No
nama_dosen_gelar	varchar	255	0	No
nama_dosen	varchar	100	0	No
gelar_depan	varchar	50	0	No
gelar_belakang	varchar	50	0	No
flag_aktif	int	1	0	No
flag_delete	int	1	0	No
uuid_kelas	varchar	128	0	No

l. Tabel Mahasiswa

Tabel ini menyimpan data mahasiswa di UII. Adapun struktur tabel ditunjukkan pada Tabel 3.23.

Tabel 3.23 Tabel Mahasiswa

Name	Type	Length	Decimals	Not Null
id	binary	16	0	Yes
id_kurikulum	bigint	20	0	No
kd_kurikulum	varchar	4	0	No
id_organisasi	bigint	20	0	No
kd_status	varchar	1	0	Yes
kd_organisasi	varchar	10	0	Yes
id_dosen	id_dosen	20	0	No
kd_dosen	varchar	9	0	No
tahun	varchar	4	0	Yes
nim	varchar	20	0	Yes
nama_mahasiswa	varchar	60	0	Yes
ipk	float	5	2	No
sks_kum	float	5	2	No
user_input	varchar	15	0	Yes
tgl_input	datetime	0	0	Yes
user_update	varchar	15	0	No
tgl_update	datetime	0	0	No
uuid	varchar	128	0	Yes

m. Tabel Ras Mahasiswa

Tabel ini menyimpan data mahasiswa yang melakukan *key-in* ras pada setiap tahun akademik. Adapun struktur tabel ditunjukkan pada Tabel 3.24.



Tabel 3.24 Tabel Ras Mahasiswa

Name	Type	Length	Decimals	Not Null
id	binary	16	0	Yes
id_tahun_akademik	bigint	20	0	No
id_organisasi	bigint	20	0	No
id_mahasiswa	binary	16	0	No
no_mhs	varchar	15	0	No
sks_ambil	float	5	2	No
sks_kum	float	5	2	No
ip_smt	float	3	2	No
ipk	float	6	3	No
tgl_ras	datetime	0	0	No
user_ras	varchar	15	0	No
flag_verif	int	1	0	No
user_verif	varchar	15	0	No
tgl_verif	datetime	0	0	No
user_input	varchar	15	0	Yes
tgl_input	datetime	0	0	No
user_update	varchar	15	0	Yes
tgl_update	datetime	0	0	No
uuid	varchar	128	0	Yes

n. Tabel Ras Mahasiswa Detail

Tabel ini menyimpan data hasil *key-in* mahasiswa UII pada setiap tahun akademik. Adapun struktur tabel ditunjukkan pada Tabel 3.25.

Tabel 3.25 Tabel Ras Mahasiswa Detail

Name	Type	Length	Decimals	Not Null
id	Binary	16	0	Yes
id_ras_mahasiswa	Binary	16	0	No
no_mhs	Varchar	15	0	No
id_kurikulum	BigInt	20	0	No
id_matakuliah	BigInt	20	0	No
id_kelas	Binary	16	0	No
uuid	Varchar	128	0	No

## o. Tabel Kurikulum

Tabel ini menyimpan data kurikulum setiap program studi di UII. Adapun struktur tabel ditunjukkan pada Tabel 3.26.

Tabel 3.26 Tabel Kurikulum

Name	Type	Length	Decimals	Not Null
id	bigint	20	0	Yes
id_organisasi	bigint	20	0	Yes
id_tahun_akademik	bigint	20	0	Yes
id_jenjang	bigint	20	0	Yes
kd_kurikulum	varchar	4	0	Yes
nama_kurikulum	varchar	255	0	Yes
nama_jenjang	varchar	255	0	Yes
flag_aktif	int	1	0	Yes
tgl_update	datetime	0	0	Yes
uuid	varchar	128	0	Yes

### 3.3.3 Desain Product Backlog Scrum

Dalam penelitian ini, pengerjaan akan dilakukan dalam dua kali *sprint* di mana setiap *sprint* dilakukan selama 30 hari. Adapun *product backlog* setiap *sprint* ditunjukkan dalam Tabel 3.27.

Tabel 3.27 Product Backlog

Sprint ke-	Product Backlog
1	Petugas dan dosen dapat melakukan <i>input</i> presensi
	Petugas dan dosen dapat melakukan <i>input</i> nilai
	Petugas dan dosen dapat melihat presensi mahasiswa dalam setiap pertemuan
	Petugas dan dosen dapat melihat nilai setiap mahasiswa
	Petugas dan dosen dapat melakukan <i>update</i> presensi mahasiswa
	Petugas dan dosen dapat melakukan <i>update</i> nilai mahasiswa
	Petugas dan dosen dapat melihat riwayat perubahan nilai
	Petugas dan dosen dapat melihat rekap nilai mahasiswa
	Petugas dan dosen dapat melihat laporan presensi mahasiswa
	Petugas dan dosen dapat melihat daftar penyerahan nilai dari dosen
2	Petugas dapat <i>import</i> presensi mahasiswa menggunakan file *.xlsx maupun *.dbf
	Petugas dan dosen dapat <i>import</i> nilai mahasiswa menggunakan file *.xlsx maupun *.dbf
	Petugas dapat mencetak berkas yang berkaitan dengan perkuliahan
	Melakukan <i>testing</i> menggunakan scenario yang dibuat

### 3.3.4 Desain Service Simak

*Service* ini akan menjadi pendukung *service assessment* dan *service presence* dalam mengolah data yang bersumber dari *database* Oracle. Sehingga *service* ini tidak bisa digunakan secara langsung melainkan harus melalui *service assessment* atau *presence*. Adapun RESTful API yang dibuat dalam *service* ini sebagai berikut:

a. *Request* dan *Response* Kurikulum

API ini digunakan untuk mendapatkan kumpulan kode kurikulum yang dikombinasikan dengan kode organisasi. Hal ini diperlukan karena data *response* akan digunakan untuk menyaring data-data yang lain. Adapun rancangan *Request* dan *Response* Kurikulum ditunjukkan pada Tabel 3.28.

Tabel 3.28 Rancangan *Request* dan *Response* Kurikulum

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/main/kurikulum
<b>Header</b>	-
<b>Params</b>	-th_akademi: 2015/2016 -smt: 1 -kd_jurusan: 523
<b>Keterangan</b>	Terdapat beberapa parameter yang wajib diisikan yaitu th_akademi, smt, dan kd_jurusan.
<b>Contoh Response</b>	[ { "uuid": "2010-523", "nama_kurikulum": "Kurikulum 2010" } ]

b. *Request* dan *Response* Matakuliah

API ini digunakan untuk mendapatkan kumpulan kode matakuliah yang tersedia pada periode dengan kurikulum tertentu. *Response* dari API ini berisi kumpulan kode matakuliah dan nama matakuliah. *Response* akan digunakan untuk menyaring data-data dari API lain. Adapun rancangan *Request* dan *Response* Matakuliah ditunjukkan pada Tabel 3.29.

Tabel 3.29 Rancangan Request dan Response Matakuliah

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/main/matakuliah
<b>Header</b>	-
<b>Params</b>	-th_akademi: 2013/2014 -smt: 1 -uuid_kurikulum: 2006-523 -kd_dosen -dosen
<b>Keterangan</b>	Sebagai dosen parameter yang perlu dikirim adalah th_akademi, smt, kd_dosen dan dosen. Sedangkan sebagai petugas parameter yang perlu dikirim adalah th_kademi, smt, dan uuid_kurikulum.
<b>Contoh Response</b>	[ { "kd_matakuliah": "52000341", "nama_matakuliah": "Kerja Praktek" }, { "kd_matakuliah": "10002233", "nama_matakuliah": "Tugas Akhir" } ]

c. *Request dan Response Kelas*

API ini digunakan untuk mendapatkan uuid kelas yang merupakan kombinasi dari tahun akademik, semester, kode program studi, kode kurikulum, kode matakuliah dan nama kelas. *Response* akan digunakan untuk menyaring data-data dari API lain. Adapun rancangan *Request* dan *Response* Kelas ditunjukkan pada Tabel 3.30.

Tabel 3.30 *Request dan Response Kelas*

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/main/kelas
<b>Header</b>	-
<b>Params</b>	-nama_tahun_akademik: 2013/2014 -kd_semester: 2 -kd_matakuliah: 52323639 -uuid_kurikulum: 2010-523
<b>Keterangan</b>	Terdapat beberapa parameter yang wajib diisikan yaitu nama_tahun_akademi, kd_semester, kd_matakuliah dan uuid_kurikulum.
<b>Contoh Response</b>	[ { "uuid_kelas": "2013/2014-2-523-2010-52323639-a", "nama_kelas": "A" } ]

d. *Request dan Response Dosen*

API ini digunakan untuk mendapatkan NIP dan nama dosen. *Response* dari API ini akan digunakan sebagai filter dalam pencetakan presensi ujian kelas gabungan. Adapun rancangan *Request dan Response Dosen* ditunjukkan pada Tabel 3.31.

Tabel 3.31 Rancangan *Request dan Response Dosen*

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/main/dosen
<b>Header</b>	-
<b>Params</b>	-nama_tahun_akademik: 2013/2014 -kd_semester: 2 -kd_matakuliah: 52323639 -uuid_kurikulum: 2010-523
<b>Keterangan</b>	Terdapat beberapa parameter yang wajib diisikan yaitu nama_tahun_akademi, kd_semester, kd_matakuliah dan uuid_kurikulum.
<b>Contoh Response</b>	[ { "kd_dosen": "005230199", "nama_dosen": " DOSEN JTIF FTI " } ]

e. *Request* dan *Response* Detail Kelas bagian Nilai

API ini digunakan untuk mendapatkan detail dari kelas untuk bagian nilai mahasiswa. Adapun rancangan *Request* dan *Response* Detail Kelas bagian Nilai ditunjukkan pada Tabel 3.32.

Tabel 3.32 Rancangan *Request* dan *Response* Detail Kelas bagian Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/assessment/kelasDetail
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 2015/2016-2-523-2010-52322203-a
<b>Keterangan</b>	Terdapat beberapa parameter yang wajib diisikan yaitu uuid_kelas
<b>Contoh Response</b>	<pre>{   "uuid_kelas": "2015/2016-2-523-2010-52322203-a",   "matakuliah": "Algoritma dan Pemrograman 2 - A",   "prodi": "Informatika",   "periode": "2015/2016 - Genap",   "tgl_ujian": null,   "tgl_penyerahan": "2020-01-01" }</pre>

f. *Request* dan *Response* Daftar Mahasiswa bagian Nilai

API ini digunakan untuk memberikan *response* berupa daftar mahasiswa dalam kelas beserta dengan nilai dari mahasiswa tersebut. Adapun rancangan *request* dan *response* ditunjukkan pada Tabel 3.33.

Tabel 3.33 Rancangan *Request* dan *Response* Daftar Mahasiswa bagian Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/assessment/mahasiswa
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 2015/2016-2-523-2010-52322203-a
<b>Keterangan</b>	Terdapat beberapa parameter yang wajib diisikan yaitu uuid_kelas
<b>Contoh Response</b>	<pre>[   {     "nim": "11523146",     "nama_mahasiswa": "ABDULLAH HAICHAL ALBAAR",     "nilai": "C",     "user_input": "PAKEKO",     "tgl_input": "02-AUG-16",     "user_update": null,     "tgl_update": "20-JUL-19"   },   {     "nim": "15523007",     "nama_mahasiswa": "ADE BAGAS PERMATA PUTRA",     "nilai": "F",     "user_input": "mustafa",     "tgl_input": "18-JAN-20",     "user_update": null,     "tgl_update": "20-JUL-19"   } ]</pre>

g. *Request* dan *Response* Riwayat Nilai

API ini digunakan untuk mendapatkan detail dari mahasiswa yang ingin dilihat riwayatnya. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.34.



Tabel 3.34 Rancangan *Request* dan *Response* Riwayat Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/assessment/riwayat
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 2015/2016-2-523-2010-52322203-a -no_mhs: 15523007
<b>Keterangan</b>	Terdapat beberapa parameter yang wajib terisi yaitu <code>uuid_kelas</code> dan <code>no_mhs</code>
<b>Contoh Response</b>	<pre>{   "matakuliah": "Algoritma dan Pemrograman 2 - A",   "periode": "2015/2016 - Genap",   "mahasiswa": "ADE BAGAS PERMATA PUTRA (15523007)" }</pre>

h. *Request* dan *Response* Rekap Nilai

API ini digunakan untuk mendapatkan rekapitulasi jumlah mahasiswa yang dikelompokkan berdasarkan nilai. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.35.

Tabel 3.35 Rancangan *Request* dan *Response* Rekap Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/assessment/rekap
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 2015/2016-2-523-2010-52322203-a
<b>Keterangan</b>	Terdapat beberapa parameter yang wajib terisi yaitu uuid_kelas.
<b>Contoh Response</b>	<pre> {   "prodi": "Informatika",   "matakuliah": "Algoritma dan Pemrograman 2 - A",   "periode": "2015/2016 - Genap",   "rekap": [     {       "name": "A/B",       "y": 1     },     {       "name": "B+",       "y": 2     },     {       "name": "B",       "y": 2     },     {       "name": "B-",       "y": 1     },     {       "name": "B/C",       "y": 5     },     {       "name": "C+",       "y": 1     },     {       "name": "C",       "y": 7     },     {       "name": "C-",       "y": 6     },     {       "name": "F",       "y": 2     }   ] } </pre>

i. *Request* dan *Response* Penyerahan Nilai

API ini digunakan untuk melihat daftar kelas yang telah menyerahkan nilai akhir matakuliah. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.36.

Tabel 3.36 Rancangan *Request* dan *Response* Penyerahan Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/assessment/penyerahan
<b>Header</b>	-
<b>Params</b>	-th_akademi: 2016/2017 -smt: 2 -uuid_kurikulum: 2016-523
<b>Keterangan</b>	Terdapat beberapa parameter yang harus terisi yaitu th_akademi, smt, dan uuid_kurikulum.
<b>Contoh Response</b>	<pre>[   {     "nik": "035200528",     "nama": " AANG KUNAEFI S.Ag., M.Ag.",     "matakuliah": "Ibadah dan Akhlak - A",     "tgl_ujian": null,     "tgl_penyerahan": null,     "jeda_waktu": "- hari"   },   {     "nik": "035200528",     "nama": " AANG KUNAEFI S.Ag., M.Ag.",     "matakuliah": "Ibadah dan Akhlak - B",     "tgl_ujian": null,     "tgl_penyerahan": null,     "jeda_waktu": "- hari"   },   {     "nik": "045110403",     "nama": "Drs. JAMRONI MSI.",     "matakuliah": "Ibadah dan Akhlak - D",     "tgl_ujian": null,     "tgl_penyerahan": null,     "jeda_waktu": "- hari"   }, ],</pre>

j. *Request* dan *Response* Store Nilai

API ini digunakan untuk menambahkan dan mengubah nilai mahasiswa yang ada dalam basis data Oracle. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.37.

Tabel 3.37 Rancangan *Request* dan *Response* Store Nilai

<b>Method</b>	<i>POST</i>
<b>URL</b>	localhost:1003/private/api/v1/assessment/storeAll
<b>Header</b>	- Content-Type: application/x-www-form-urlencoded
<b>Body</b>	-uuid_kelas: 2015/2016-2-523-2010-52322203-a -mahasiswa: [{"nim":"11523146","nilai":"C"}, {"nim":"15523007","nilai":"A"}] -user: mustafa
<b>Keterangan</b>	Pada bagian <i>body</i> wajib mengirim <i>uuid_kelas</i> , <i>mahasiswa</i> yang berisi <i>nim</i> dan <i>nilai mahasiswa</i> , dan <i>user</i> yang merupakan <i>NIK dosen</i> atau <i>petugas</i> yang mengirim <i>request</i> . <i>Response</i> adalah jumlah nilai yang berhasil ditambahkan dan diubah.
<b>Contoh Response</b>	1

k. *Request* dan *Response* Store Pertemuan

API ini digunakan untuk menyimpan materi pertemuan ke dalam basis data Oracle.

Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.38.

Tabel 3.38 Rancangan *Request* dan *Response* Store Pertemuan

<b>Method</b>	POST
<b>URL</b>	localhost:1003/private/api/v1/assembly/storeAll
<b>Header</b>	- Content-Type: application/x-www-form-urlencoded
<b>Body</b>	-uuid_kelas: 2016/2017-6-523-2010-52323645-a -pertemuan_ke: 1 -materi: Remediasi Genap
<b>Keterangan</b>	- <i>Request</i> wajib mengirim <i>uuid_kelas</i> , <i>pertemuan_ke</i> , dan <i>materi</i> pada bagian <i>body</i> . - <i>Response</i> adalah angka 1 yang berarti berhasil dan 0 yang berarti gagal.
<b>Contoh Response</b>	1

l. *Request* dan *Response* Detail Kelas bagian Presensi

API ini digunakan untuk mendapatkan detail dari kelas yang sedang diakses untuk bagian presensi. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.39.

Tabel 3.39 Rancangan *Request* dan *Response* Detail Kelas bagian Presensi

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/presence/kelasDetail
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 2015/2016-1-523-2010-52312107-a
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas.
<b>Contoh Response</b>	<pre>{   "fakultas": "TEKNOLOGI INDUSTRI",   "jurusan": "Informatika",   "mk_kelas": "52312107 / A",   "matakuliah": "Algoritma dan Pemrograman 1",   "dosen": " Feri Wijayanto S.T., M.T.",   "periode": "2015/2016 - Ganjil" }</pre>

m. *Request* dan *Response* Detail Kelas Gabungan

API ini digunakan untuk mendapatkan detail dari kelas yang sedang diakses untuk bagian cetak presensi ujian kelas gabungan. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.40.

Tabel 3.40 Rancangan *Request* dan *Response* Detail Kelas Gabungan

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/presence/kelasGabunganDetail
<b>Header</b>	-
<b>Params</b>	- th_akademi: 2015/2016 -smt: 1 -kd_dosen: 045230406 -kd_matakuliah: 52312105
<b>Keterangan</b>	Pada bagian parameter wajib mengirim th_akademi, smt, kd_dosen, dan kd_matakuliah
<b>Contoh Response</b>	<pre>{   "fakultas": "TEKNOLOGI INDUSTRI",   "jurusan": "Informatika",   "mk_kelas": "52312105",   "matakuliah": "Logika Matematika",   "dosen": " LIZDA ISWARI S.T., M.Sc.",   "periode": "2015/2016 - Ganjil" }</pre>

n. *Request* dan *Response* Jumlah Pertemuan

API ini digunakan untuk mendapatkan jumlah pertemuan yang sudah terlaksana pada kelas tertentu. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.41.

Tabel 3.41 Rancangan *Request* dan *Response* Jumlah Pertemuan

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/presence/jumlahPertemuan
<b>Header</b>	-
<b>Params</b>	- uuid_kelas: 2015/2016-2-523-2010-52322203-a
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas. <i>Response</i> merupakan jumlah pertemuan yang telah terlaksana.
<b>Contoh Response</b>	14

o. *Request* dan *Response* Materi

API ini digunakan untuk mendapatkan materi pada pertemuan tertentu. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.42.

Tabel 3.42 Rancangan *Request* dan *Response* Materi

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/presence/materi
<b>Header</b>	-
<b>Params</b>	- uuid_kelas: 2015/2016-1-523-2010-52323639-a -pertemuan_ke: 1
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas, dan pertemuan_ke.
<b>Contoh Response</b>	{ "uuid": "2015/2016-1-523-2010-52323639-a-1", "materi": "Kontrak Belajar", "pertemuan_ke": "1" }

p. *Request* dan *Response* Daftar Mahasiswa bagian Presensi

API ini digunakan untuk mendapatkan daftar mahasiswa beserta status setiap pertemuan. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.43.

Tabel 3.43 *Request* dan *Response* Daftar Mahasiswa bagian Presensi

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/presence/mahasiswa
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 2015/2016-1-523-2010-52323639-a -pertemuan_ke: 1
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas dan pertemuan_ke. <i>Response</i> berisi nim, nama, dan status presensi.
<b>Contoh Response</b>	<pre>[   {     "nim": "12523088",     "nama": "LUQMAN HARYANTO H",     "alpha": 0,     "hadir": 1,     "izin": 0   },   {     "nim": "12523165",     "nama": "FITRI KARUNIA",     "alpha": 0,     "hadir": 0,     "izin": 1   },   {     "nim": "12523262",     "nama": "GRA PRIMA CHENDY PRAYANI",     "alpha": 1,     "hadir": 0,     "izin": 0   } ]</pre>

q. *Request* dan *Response* Laporan Presensi

API ini digunakan untuk melihat jumlah hadir, izin, alpa, dan persentase kehadiran setiap mahasiswa dari awal pertemuan hingga akhir. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.44.

Tabel 3.44 Rancangan *Request* dan *Response* Laporan Presensi

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/presence/laporan
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 2015/2016-1-523-2010-52323639-a
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas.
<b>Contoh Response</b>	<pre>[   {     "nim": "12523088",     "nama": "LUQMAN HARYANTO H",     "hadir": "2",     "izin": "0",     "alpha": "1",     "total": 2,     "porsentase": 100   },   {     "nim": "12523165",     "nama": "FITRI KARUNIA",     "hadir": "0",     "izin": "2",     "alpha": "1",     "total": 2,     "porsentase": 100   },   {     "nim": "12523262",     "nama": "GRA PRIMA CHENDY PRAYANI",     "hadir": "0",     "izin": "0",     "alpha": "3",     "total": 2,     "porsentase": 0   } ]</pre>

r. *Request* dan *Response* Store Presensi

API ini digunakan untuk menyimpan status kehadiran mahasiswa ke dalam basis data oracle. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.45.



Tabel 3.45 Rancangan *Request* dan *Response Store* Presensi

<b>Method</b>	POST
<b>URL</b>	localhost:1003/private/api/v1/presence/storeAll
<b>Header</b>	- Content-Type: application/x-www-form-urlencoded
<b>Body</b>	-uuid_kelas: 2016/2017-6-523-2010-52323645-a -pertemuan_ke: 1 -mahasiswa: [{"nim":"14523176","status":"Hadir"}]
<b>Keterangan</b>	Pada bagian <i>body</i> wajib mengirim <i>uuid_kelas</i> , <i>pertemuan_ke</i> , <i>mahasiswa</i> . <i>Response</i> adalah jumlah dari mahasiswa status presensi mahasiswa yang berhasil diperbarui.
<b>Contoh Response</b>	1

s. *Request* dan *Response Store* Nilai dari *File*

API ini digunakan untuk menyimpan nilai mahasiswa yang menggunakan *file* ke dalam basis data Oracle. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.46.

Tabel 3.46 Rancangan *Request* dan *Response Store* Nilai dari *File*

<b>Method</b>	POST
<b>URL</b>	localhost:1003/private/api/v1/file/storeAssessment
<b>Header</b>	- Content-Type: application/x-www-form-urlencoded
<b>Body</b>	<pre>-file: {   "tahun_akademik":"2019/2020",   "semester":"1",   "kd_jurusan":"523",   "kd_kurikulum":"2016",   "no_mk":"52311001",   "nama_kelas":"a",   "mahasiswa": [     {"nim":"17523079","nilai":"A"},     {"nim":"19523002","nilai":"B"},     {"nim":"19523003","nilai":"C"}   ] } -user: mustafa -hak_akses: ["523","422","933","913","423","421","424"] -dosen: 0</pre>
<b>Keterangan</b>	<p><i>Request:</i></p> <ul style="list-style-type: none"> <li>-file, merupakan data dari <i>file</i> yang telah diolah ke dalam JSON.</li> <li>-user, merupakan NIK dari user.</li> <li>-hak_akses, merupakan kumpulan kd_organisasi yang bisa diakses oleh user.</li> <li>-dosen, menunjukkan status sebagai dosen atau petugas.</li> </ul> <p><i>Response:</i></p> <ul style="list-style-type: none"> <li>-jumlah nilai yang berhasil ditambahkan atau diperbarui.</li> </ul>
<b>Contoh Response</b>	3

t. *Request* dan *Response Store* Presensi dari *File*

API ini digunakan untuk menyimpan status presensi mahasiswa yang menggunakan *file* ke dalam basis data *Oracle*. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.47.

Tabel 3.47 Rancangan *Request* dan *Response Store* Presensi dari *File*

<b>Method</b>	POST
<b>URL</b>	localhost:1003/private/api/v1/file/storePresence
<b>Header</b>	- Content-Type: application/x-www-form-urlencoded
<b>Body</b>	<pre>-file: {   "tahun_akademik":"2019\2020",   "semester":"1",   "kd_jurusan":"523",   "kd_kurikulum":"2016",   "no_mk":"52312003",   "nama_kelas":"a",   "kd_dosen":"125200501",   "pertemuan": [     {       "nama_pertemuan":"1",       "mahasiswa": [         {"nim":"19523002", "status":1},         {"nim":"19523003", "status":1}       ]     }   ] } -hak_akses: [523, 522]</pre>
<b>Keterangan</b>	<p><i>Request:</i>  <i>-file</i>, merupakan data dari <i>file</i> yang telah diolah ke dalam json.  <i>-hak_akses</i>, merupakan kumpulan <i>kd_organisasi</i> yang bisa diakses oleh <i>user</i>.</p> <p><i>Response:</i>  -jumlah presensi mahasiswa yang berhasil ditambahkan atau diperbarui.</p>
<b>Contoh Response</b>	2

u. *Request* dan *Response Store* Pertemuan dari *File*

API ini digunakan untuk menyimpan materi setiap pertemuan yang menggunakan *file* ke dalam basis data Oracle. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.48.

Tabel 3.48 Rancangan *Request* dan *Response* Store Pertemuan dari *File*

<b>Method</b>	POST
<b>URL</b>	localhost:1003/private/api/v1/file/storeAssembly
<b>Header</b>	- Content-Type: application/x-www-form-urlencoded
<b>Body</b>	<pre>-file: {   "tahun_akademik":"2019\2020",   "semester":"1",   "kd_jurusan":"523",   "kd_kurikulum":"2016",   "no_mk":"52312003",   "nama_kelas":"a",   "kd_dosen":"125200501",   "pertemuan": [     {"nama_pertemuan":"1","materi":"Pertemuan 1"},     {"nama_pertemuan":"2","materi":"Pertemuan 2"}   ] } -hak_akses: ["523"]</pre>
<b>Keterangan</b>	<p><i>Request:</i> -file, merupakan data dari <i>file</i> yang telah diolah ke dalam JSON. -hak_akses, merupakan kumpulan kd_organisasi yang bisa diakses oleh user.</p> <p><i>Response:</i> -jumlah pertemuan yang berhasil ditambahkan atau diperbarui.</p>
<b>Contoh Response</b>	2

v. *Request* dan *Response* Cetak Presensi Dosen

API ini digunakan untuk mencetak persentase kehadiran dosen dalam setiap kelas. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.49.

Tabel 3.49 Rancangan *Request* dan *Response* Cetak Presensi Dosen

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/print/presensiDosen
<b>Header</b>	-
<b>Params</b>	-th_akademi: 2015/2016 -smt: 1 -kd_jurusan: 523
<b>Keterangan</b>	Pada bagian parameter wajib mengirim th_akademi, smt, dan kd_jurusan.
<b>Contoh Response</b>	<pre>[   {     "kd_matakuliah": "52312105",     "matakuliah": "Logika Matematika",     "kelas": "a",     "dosen": " LIZDA ISWARI S.T., M.Sc.",     "hadir": 13,     "target": 14,     "persen_hadir": "92.86"   },   {     "kd_matakuliah": "52312105",     "matakuliah": "Logika Matematika",     "kelas": "b",     "dosen": " LIZDA ISWARI S.T., M.Sc.",     "hadir": 13,     "target": 14,     "persen_hadir": "92.86"   } ]</pre>

w. *Request* and *Response* Cetak Persentase Kehadiran Kuliah

API ini digunakan untuk mencetak persentase kehadiran jumlah mahasiswa dalam setiap pertemuan untuk semua kelas dalam satu program studi dan kurikulum. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.50.

Tabel 3.50 Rancangan *Request* dan *Response* Cetak Persentase Kehadiran Kuliah

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/print/prosentaseKK
<b>Header</b>	-
<b>Params</b>	-th_akademi: 2015/2016 -smt: 2 -uuid_kurikulum: 2010-523
<b>Keterangan</b>	Pada bagian parameter wajib mengirim th_akademi, smt, dan uuid_kurikulum. <i>Response</i> merupakan <i>array</i> yang berisi obyek dengan persentase kehadiran mahasiswa setiap pertemuan.
<b>Contoh Response</b>	<pre>[   {     "matakuliah": "Data Mining",     "kelas": "A",     "jmlh_mahasiswa": 50,     "persentase": [       "96.00",       "88.00",       "84.00",       "90.00",       "82.00",       "100.00",       "100.00",       "76.00",       "84.00",       "82.00",       "90.00",       "88.00",       "90.00",       "66.00",       "-"     ],     "rerata": "0.87"   } ]</pre>

x. *Request* and *Response* Cetak Rekapitulasi Beban Mengajar Dosen

API ini digunakan untuk mencetak rekapitulasi beban mengajar dosen dalam satu periode dan fakultas. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.51.

Tabel 3.51 Rancangan *Request* dan *Response* Cetak Beban Mengajar Dosen

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/print/rekapitulasiBMD
<b>Header</b>	-
<b>Params</b>	-th_akademi: 2015/2016 -smt: 1 -kd_fakultas: 52
<b>Keterangan</b>	Pada bagian parameter wajib mengirim th_akademi, smt, dan kd_fakultas. <i>Response</i> berisi kumpulan beban mengajar dosen selama satu periode dalam satu fakultas.
<b>Contoh Response</b>	<pre>[   {     "nama_dosen": " PUNANG AMARIPUJA S.E., M.Si.",     "nip": "003100406",     "jumlah_sks": 3,     "jumlah_mk": 1,     "matakuliah": [       {         "kode_mk": "52224501",         "matakuliah": "Management Information System",         "sks": 3,         "kelas": "a",         "peserta": 24,         "jurusan": "Teknik Industri Program Internasional"       }     ]   } ]</pre>

y. *Request* and *Response* Cetak Kartu Ujian

API ini digunakan untuk mencetak kartu ujian mahasiswa dalam rentang tertentu pada periode, program studi, dan kurikulum yang sama. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.52.

Tabel 3.52 Rancangan *Request* dan *Response* Cetak Kartu Ujian

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/print/kartuUjian
<b>Header</b>	-
<b>Params</b>	-th_akademi: 2015/2016 -smt: 1 -uuid_kurikulum: 2010-523 -mulai_nim: 15523000 -sampai_nim: 15523100
<b>Keterangan</b>	Pada bagian parameter wajib mengirim th_akademi, smt, uuid_kurikulum, mulai_nim, dan sampai_nim. <i>Response</i> berisi kumpulan data mahasiswa dalam rentang mulai_nim dan sampai_nim. Data tersebut akan dimanfaatkan oleh antarmuka.
<b>Contoh Response</b>	<pre>[   {     "fakultas": "TEKNOLOGI INDUSTRI",     "jurusan": "Informatika",     "nim": "15523002",     "nama_mahasiswa": "DESTA DEVIANI PRAMANA",     "Smt_TA": "Ganjil 2015/2016",     "DPA": " Septia Rani S.T., M.Cs.",     "jumlah_sks": 6,     "matakuliah": [       {         "nama_matakuliah": "Algoritma dan Pemrograman 1",         "sks": 3,         "nama_kelas": "A",         "uuid_kelas": "2015/2016-1-523-2010-52312107-a",         "no_urut": 6,         "jadwal": [           {             "jadwal_kuliah": "Rabu 09:30-12:00 04.06 Gb",             "pengampu": "Feri Wijayanto"           }         ]       }     ],   },   {     "nama_matakuliah": "Logika Matematika",     "sks": 3,     "nama_kelas": "A",     "uuid_kelas": "2015/2016-1-523-2010-52312105-a",     "no_urut": 6,     "jadwal": [       {         "jadwal_kuliah": "Kamis 12:30-15:00 04.06 Gb",         "pengampu": "LIZDA ISWARI"       }     ]   } ] ]</pre>



z. *Request and Response* Cetak Presensi Ujian

API ini digunakan untuk mencetak presensi ujian per kelas maupun gabungan. Jika per kelas maka *response* berupa daftar mahasiswa dalam satu kelas. Sedangkan, jika gabungan maka *response* berupa daftar mahasiswa pada matakuliah dan dosen yang sama. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.53.

Tabel 3.53 Rancangan *Request* dan *Response* Cetak Presensi Ujian

<b>Method</b>	GET
<b>URL</b>	localhost:1003/private/api/v1/print/presensiUjian
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 2015/2016-2-523-2010-52322203-a -th_akademi: 2015/2016 -smt: 1 -kd_dosen: 045230406 -kd_matakuliah: 52312105
<b>Keterangan</b>	Jika API digunakan untuk cetak per kelas, maka cukup mengirim uuid_kelas pada bagian parameter. Sedangkan jika untuk cetak gabungan, maka tidak perlu mengirim uuid_kelas tapi harus mengirim th_akademi, smt, kd_dosen, dan kd_matakuliah. <i>Response</i> merupakan kumpulan nim, nama, dan persentase kehadiran mahasiswa.
<b>Contoh Response</b>	[ <pre>         {           "nim": "11523130",           "nama": "RIDHO ILHAM YULIANDA",           "hadir": "10",           "izin": "0",           "alpha": "3",           "total_pertemuan": 13,           "porsentase": 76.92307692307692         },         {           "nim": "11523167",           "nama": "PANJI DWI SASTRA",           "hadir": "11",           "izin": "0",           "alpha": "2",           "total_pertemuan": 13,           "porsentase": 84.61538461538461         }       ]</pre>

### 3.3.5 Desain *Service Presence*

*Service* ini merupakan *service* yang berfokus pada pengelolaan presensi mahasiswa dan pencetakan berkas. Adapun RESTful API yang dibuat dalam *service* ini sebagai berikut:

a. *Request* dan *Response* Periode

API ini digunakan untuk mendapatkan uuid dan nama periode yang aktif. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.54.

Tabel 3.54 Rancangan *Request* dan *Response* Periode

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/periode
<b>Header</b>	-
<b>Params</b>	-
<b>Keterangan</b>	<i>Response</i> adalah daftar semua uuid dan nama periode di UII.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 6,   "data": [     {       "uuid": "0465d84a-31bd-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Konversi"     },     {       "uuid": "f95b3adb-31bc-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Remediasi Genap"     },     {       "uuid": "ea526fb7-31bc-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Remediasi Ganjil"     },     {       "uuid": "da20e7f0-31bc-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Pendek"     },     {       "uuid": "c35fa7d7-31bc-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Genap"     },     {       "uuid": "9f03fa99-ad01-11e9-946e-506b8da96a87",       "semester": "2019/2020 - Ganjil"     }   ] }</pre>

b. *Request* dan *Response* Program Studi

API ini digunakan untuk mendapatkan uuid, nama, dan kode program studi. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.55.

Tabel 3.55 Rancangan *Request* dan *Response* Program Studi

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/prodi
<b>Header</b>	- x-organization: [{"kd_organisasi": "523"}, {"kd_organisasi": "42"}, {"kd_organisasi": "123"}]
<b>Params</b>	-
<b>Keterangan</b>	<i>Response</i> adalah daftar program studi yang bisa diakses oleh pengguna.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 7,   "data": [     {       "uuid": "ad07defc-f154-11e7-b2cf-005056806fe5",       "name": "Ahwal Al-syakhshiyah",       "kd_organisasi": "421"     },     {       "uuid": "ad07d087-f154-11e7-b2cf-005056806fe5",       "name": "Pendidikan Agama Islam",       "kd_organisasi": "422"     },     {       "uuid": "ad07db44-f154-11e7-b2cf-005056806fe5",       "name": "Ekonomi Islam",       "kd_organisasi": "423"     },     {       "uuid": "bf567c0b-c5eb-11e8-8f38-005056804a5c",       "name": "Ahwal Al-syakhshiyah Program Internasional",       "kd_organisasi": "424"     },     {       "uuid": "ad07d79a-f154-11e7-b2cf-005056806fe5",       "name": "Magister Studi Islam",       "kd_organisasi": "913"     },     {       "uuid": "ad07d446-f154-11e7-b2cf-005056806fe5",       "name": "Doktor Hukum Islam",       "kd_organisasi": "933"     },     {       "uuid": "ad081f99-f154-11e7-b2cf-005056806fe5",       "name": "Informatika",       "kd_organisasi": "523"     }   ] } </pre>

c. *Request* dan *Response* Kurikulum

API ini digunakan untuk mendapatkan uuid, dan nama kurikulum pada periode dan organisasi tertentu. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.56.

Tabel 3.56 Rancangan *Request* dan *Response* Kurikulum

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/kurikulum
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_organisasi: ad081f99-f154-11e7-b2cf-005056806fe5
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik dan uuid_organisasi
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 2,   "data": [     {       "uuid": "f5acca68-2097-11ea-889a-506b8da96a87",       "nama_kurikulum": "Kurikulum 2016"     },     {       "uuid": "f590cec1-2097-11ea-889a-506b8da96a87",       "nama_kurikulum": "Kurikulum 2010"     }   ] }</pre>

d. *Request* dan *Response* Matakuliah

API ini digunakan oleh petugas dan dosen tetapi memiliki *response* yang berbeda tergantung perannya. Sebagai petugas, *response* adalah kode dan nama matakuliah. Sedangkan sebagai dosen, *response* adalah nama matakuliah, nama kelas, dan uuid kelas. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.57.

Tabel 3.57 Rancangan *Request* dan *Response* Matakuliah

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/matakuliah
<b>Header</b>	x-member: 145230403
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kurikulum: f5acca68-2097-11ea-889a-506b8da96a87
<b>Keterangan</b>	Sebagai dosen, pada bagian parameter wajib mengirim uuid_tahun_akademik, sedangkan sebagai petugas atau <i>admin</i> wajib mengirimkan uuid_tahun_akademik dan uuid_kurikulum.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 2,   "data": [     {       "kd_matakuliah": "52312507",       "nama_matakuliah": "Interaksi Manusia dan Komputer - A",       "uuid_kelas": "11ea36f3-65d7-9026-b4d2-7eb0d4a3c7a0"     },     {       "kd_matakuliah": "52332002",       "nama_matakuliah": "Matematika Lanjut - D",       "uuid_kelas": "11ea36f3-65e4-495a-b4d2-7eb0d4a3c7a0"     }   ] }</pre>

e. *Request* dan *Response* Kelas

API ini digunakan oleh petugas, *response* adalah nama dan uuid kelas. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.58.

Tabel 3.58 Rancangan *Request* dan *Response* Kelas

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/kelas
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kurikulum: f5acca68-2097-11ea-889a-506b8da96a87 -kd_matakuliah: 52311001
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, uuid_kurikulum, dan kd_matakuliah.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 4,   "data": [     {       "uuid_kelas": "11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0",       "nama_kelas": "A"     },     {       "uuid_kelas": "11ea36f3-65e1-49dd-b4d2-7eb0d4a3c7a0",       "nama_kelas": "B"     },     {       "uuid_kelas": "11ea36f3-65e1-6dfe-b4d2-7eb0d4a3c7a0",       "nama_kelas": "C"     },     {       "uuid_kelas": "11ea36f3-65e1-9486-b4d2-7eb0d4a3c7a0",       "nama_kelas": "D"     }   ] }</pre>

f. *Request* dan *Response* Detail Kelas Presensi

API ini digunakan untuk mendapatkan detail dari kelas yang diinginkan. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.59.

Tabel 3.59 Rancangan *Request* dan *Response* Detail Kelas Presensi

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/kelasDetail
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik dan uuid_kelas
<b>Contoh Response</b>	<pre>{   "result": true,   "data": {     "fakultas": "TEKNOLOGI INDUSTRI",     "jurusan": "Informatika",     "mk_kelas": "52311001 / A",     "matakuliah": "Akidah",     "dosen": "Ahmad Baliyo Eko Prasetyo, S.Psi., M.A.",     "periode": "2019/2020 - Ganjil"   } }</pre>

g. *Request* dan *Response* Detail Kelas Gabungan

API ini digunakan untuk mendapatkan detail kelas yang digunakan ketika akan mencetak presensi ujian kelas gabungan. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.60.

Tabel 3.60 Rancangan *Request* dan *Response* Detail Kelas Gabungan

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/kelasGabunganDetail
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -kd_dosen: 045230406 -kd_matakuliah: 52362004
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, kd_dosen, dan kd_matakuliah
<b>Contoh Response</b>	<pre>{   "result": true,   "data": {     "fakultas": "TEKNOLOGI INDUSTRI",     "jurusan": "Informatika",     "mk_kelas": "52362004",     "matakuliah": "Bahasa Inggris untuk Informatika",     "dosen": "LIZDA ISWARI, S.T., M.Sc.",     "periode": "2019/2020 - Ganjil"   } }</pre>

h. *Request* dan *Response* Jumlah Pertemuan

API ini digunakan untuk mendapatkan jumlah pertemuan yang sudah terlaksana dalam setiap kelas. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.61.

Tabel 3.61 Rancangan *Request* dan *Response* Jumlah Pertemuan

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/jmlhPertemuan
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0 -uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas, dan uuid_tahun_akademik
<b>Contoh Response</b>	<pre>{   "jmlhPertemuan": 0,   "info": "Belum ada pertemuan" }</pre>

i. *Request* dan *Response* Materi



API ini digunakan untuk mendapatkan materi dari pertemuan yang dipilih. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.62.

Tabel 3.62 Rancangan *Request* dan *Response* Materi

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/materi
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0 -pertemuan_ke: 1 -uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas, pertemuan_ke, dan uuid_tahun_akademik
<b>Contoh Response</b>	{ "result": true, "data": [], "info": "Belum ada pertemuan ke 1" }

j. *Request* dan *Response* Daftar Mahasiswa dengan Presensi

API ini digunakan untuk mendapatkan daftar mahasiswa dalam kelas pilihan beserta status presensi dalam pertemuan pilihan. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.63.

Tabel 3.63 Rancangan *Request* dan *Response* Daftar Mahasiswa dengan Presensi

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/mahasiswa
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0 -pertemuan_ke: 1 -uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas, pertemuan_ke, dan uuid_tahun_akademik.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 2,   "data": [     {       "nim": "17523079",       "nama": "ALFENO MUHAYDEIN NUR",       "hadir": 0,       "izin": 0,       "alpha": 0     },     {       "nim": "19523002",       "nama": "SILVIANUR LAILA RAMADHANI",       "hadir": 0,       "izin": 0,       "alpha": 0     }   ] }</pre>

k. *Request* dan *Response* Laporan Presensi

API ini digunakan untuk mendapatkan persentase kehadiran mahasiswa pada kelas pilihan. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.64.

Tabel 3.64 Rancangan *Request* dan *Response* Laporan Presensi

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/laporan
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0 -uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas dan uuid_tahun_akademik.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 2,   "data": [     {       "nim": "17523079",       "nama": "ALFENO MUHAYDEIN NUR",       "hadir": 0,       "izin": 0,       "alpha": 0,       "total": 0,       "porsentase": 0     },     {       "nim": "19523002",       "nama": "SILVIANUR LAILA RAMADHANI",       "hadir": 0,       "izin": 0,       "alpha": 0,       "total": 0,       "porsentase": 0     }   ] }</pre>

#### 1. *Request* dan *Response* Store Presensi

API ini digunakan untuk menyimpan atau mengubah status presensi mahasiswa. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.65.

Tabel 3.65 Rancangan *Request* dan *Response Store Presensi*

<b>Method</b>	POST
<b>URL</b>	localhost:1002/public/api/v1/store
<b>Header</b>	x-member: mustafa
<b>Body</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0 -pertemuan_ke: 1 -materi: kontrak belajar -mahasiswa: [{"nim":"17523079","status":"Hadir"}, {"nim":"19523002","status":"Hadir"}]
<b>Keterangan</b>	Para bagian <i>body</i> wajib mengirim <i>uuid_tahun_akademik</i> , <i>uuid_kelas</i> , <i>pertemuan_ke</i> , <i>materi</i> dan <i>mahasiswa</i> .
<b>Contoh Response</b>	{ "result": true, "data": [], "info": "Presensi berhasil diperbarui" }

m. *Request* dan *Response Store Presensi* melalui *File*

API ini digunakan untuk menyimpan dan mengubah status presensi mahasiswa menggunakan *file*. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3. 66.

Tabel 3. 66 Rancangan *Request* dan *Response Store* Presensi melalui File

<b>Method</b>	POST
<b>URL</b>	localhost:1002/public/api/v1/file/storeFilePresensi
<b>Header</b>	x-member: mustafa x-organization: [{"kd_organisasi":"523"}, {"kd_organisasi":"42"}]
<b>Body</b>	-tahun_akademik: 2019/2020 -semester: 1 -no_mk: 52312003 -nama_kelas: a -kd_dosen: 125200501 -kd_jurusan: 523 -kd_kurikulum: 2016 -nama_pertemuan: 1 -mahasiswa: [ { "nim": "19523002", "status":1 }, { "nim": "19523003", "status":1 } ]
<b>Keterangan</b>	Pada bagian <i>body</i> wajib mengirim tahun_akademik, semester, no_mk, nama_kelas, kd_dosen, kd_jurusan, kd_kurikulum, nama_pertemuan, mahasiswa.
<b>Contoh Response</b>	{ "result": true, "data": [], "info": "2 presensi mahasiswa, pertemuan ke-1, Kelas Pemikiran Desain-A berhasil ditambahkan" }

n. *Request* dan *Response Store* Pertemuan melalui *File*

API ini digunakan untuk menyimpan dan mengubah materi pertemuan menggunakan *file*.

Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.67.

Tabel 3.67 Rancangan *Request* dan *Response Store* Pertemuan melalui *File*

<b>Method</b>	POST
<b>URL</b>	localhost:1002/public/api/v1/file/storeFilePertemuan
<b>Header</b>	x-member: mustafa x-organization: [{"kd_organisasi":"523"}, {"kd_organisasi":"42"}]
<b>Body</b>	-tahun_akademik: 2019/2020 -semester: 1 -no_mk: 52312003 -nama_kelas: a -kd_dosen: 125200501 -kd_jurusan: 523 -kd_kurikulum: 2016 -pertemuan: [{"nama_pertemuan":"1","materi":"-","jam_mulai":"15:15","jam_selesai":"17:00"}, {"nama_pertemuan":"2","materi":"-"}, {"nama_pertemuan":"3","materi":"-","jam_mulai":"15:15","jam_selesai":"17:00"}]
<b>Keterangan</b>	Pada bagian <i>body</i> wajib mengirim tahun_akademik, semester, no_mk, nama_kelas, kd_dosen, kd_jurusan, kd_kurikulum, pertemuan.
<b>Contoh Response</b>	<pre>{   "result": true,   "data": [],   "info": "2 pertemuan matakuliah Pemikiran Desain A berhasil ditambahkan" }</pre>

o. *Request* dan *Response* Cetak Presensi Dosen

API ini digunakan untuk mencetak persentase kehadiran dosen untuk setiap kelas selama satu periode. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.68.

Tabel 3.68 Rancangan *Request* dan *Response* Cetak Presensi Dosen

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/print/presensiDosen
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_organisasi: ad081f99-f154-11e7-b2cf-005056806fe5
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, dan uuid_organisasi.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 2,   "data": [     {       "kd_matakuliah": "52323639",       "matakuliah": "Administrasi Basisdata",       "kelas": "A",       "dosen": "DOSEN JTIF FTI",       "hadir": 0,       "target": 14,       "persen_hadir": "0.00"     },     {       "kd_matakuliah": "52311001",       "matakuliah": "Akidah",       "kelas": "A",       "dosen": "Ahmad Baliyo Eko Prasetyo, S.Psi., M.A.",       "hadir": 1,       "target": 14,       "persen_hadir": "7.14"     }   ] } </pre>

p. *Request* dan *Response* Cetak Kehadiran Kuliah

API ini digunakan untuk mencetak jumlah mahasiswa hadir dalam setiap pertemuan.

Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.69.

Tabel 3.69 Rancangan *Request* dan *Response* Cetak Kehadiran Kuliah

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/print/kehadiranKuliah
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9a4966fe-f6b2-11e8-9733-005056804a5c -uuid_kurikulum: ad081f99-f154-11e7-b2cf-005056806fe5
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik dan uuid_kurikulum
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 92,   "data": [     {       "matakuliah": "Akidah",       "kelas": "A",       "jmlh_mahasiswa": 49,       "persentase": [         "2.04",         "-"       ],       "renata": "0.00"     }   ] } </pre>



q. *Request* dan *Response* Cetak Kartu Ujian

API ini digunakan untuk mencetak kartu ujian mahasiswa pada rentang NIM dan periode tertentu. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.70.

Tabel 3.70 Rancangan *Request* dan *Response* Cetak Kartu Ujian

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/print/kartuUjian
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_organisasi: ad081f99-f154-11e7-b2cf-005056806fe5 -uuid_kurikulum: f5acca68-2097-11ea-889a-506b8da96a87 -mulai_nim: 17523001 -sampai_nim: 17523001
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akadmeik, uuid_organisasi, uuid_kurikulum, mulai_nim, dan sampai_nim.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 1,   "data": [     {       "fakultas": "TEKNOLOGI INDUSTRI",       "jurusan": "Informatika",       "nim": "17523001",       "nama_mahasiswa": "ACHMAD NOER AZIZ",       "Smt_TA": "Ganjil 2019/2020",       "DPA": " Almed Hamzah S.T., M.Eng.",       "jumlah_sks": 8,       "matakuliah": [         {           "nama_matakuliah": "Bahasa Inggris untuk Informatika"         },         {           "sks": 4,           "nama_kelas": "A",           "uuid_kelas": "11ea36f3-65e6-ba0e-b4d2-7eb0d4a3c7a0",           "no_urut": 6,           "jadwal": [             {               "jadwal_kuliah": "Sabtu 07:00-08:40 04.17",               "pengampu": "Lizda Iswari"             },             {               "jadwal_kuliah": "Sabtu 06:30-07:00 01.10",               "pengampu": "Lizda Iswari"             }           ]         }       ]     }   ] } </pre>

r. *Request* dan *Response* Cetak Rekapitulasi Beban Mengajar Dosen

API ini digunakan untuk mencetak rekapitulasi beban mengajar dosen pada periode dan fakultas tertentu. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.71.

Tabel 3.71 Rancangan *Request* dan *Response* Cetak Rekapitulasi Beban Mengajar Dosen

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/print/rekapitulasiBMD
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_organisasi: ad07887c-f154-11e7-b2cf-005056806fe5
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik dan uuid_organisasi
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 175,   "data": [     {       "nama_dosen": "PUNANG AMARIPUJA, S.E., M.Si.",       "nip": "003100406",       "jumlah_sks": 2,       "jumlah_mk": 1,       "matakuliah": [         {           "kode_mk": "52252006",           "matakuliah": "Management Information System",           "sks": 2,           "kelas": "A",           "peserta": 19,           "jurusan": "Teknik Industri Program Internasional"         }       ]     }   ] }</pre>

s. *Request* dan *Response* Cetak Presensi Ujian

API ini digunakan untuk mencetak presensi ujian mahasiswa. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.72.

Tabel 3.72 Rancangan *Request* dan *Response* Cetak Presensi Ujian

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/print/presensiUjian
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -kd_dosen: 045230406 -kd_matakuliah: 52362004
<b>Keterangan</b>	Pada bagian parameter wajib mengirimkan uuid_tahun_akademik, kd_dosen, dan kd_matakuliah untuk cetak gabungan. Sedangkan untuk pencetakan per kelas wajib mengirim uuid_kelas dan uuid_tahun_akademik.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 49,   "data": [     {       "total_pertemuan": 1,       "nim": "17523079",       "nama": "ALFENO MUHAYDEIN NUR",       "hadir": 0,       "izin": 0,       "alpha": 0,       "porsentase": 0     },     {       "total_pertemuan": 1,       "nim": "19523002",       "nama": "SILVIANUR LAILA RAMADHANI",       "hadir": 0,       "izin": 0,       "alpha": 0,       "porsentase": 0     }   ] }</pre>

t. *Request* dan *Response* Cetak Daftar IPK Mahasiswa

API ini digunakan untuk mencetak daftar mahasiswa dengan rentang IPK tertentu. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.73.

Tabel 3.73 Rancangan *Request* dan *Response* Cetak Daftar IPK Mahasiswa

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/print/mahasiswaIPK
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_organisasi: ad081f99-f154-11e7-b2cf-005056806fe5 -uuid_kurikulum: f5acca68-2097-11ea-889a-506b8da96a87 -mulai_ipk: 3.5 -sampai_ipk: 4
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, uuid_organisasi, uuid_kurikulum, mulai_ipk, dan sampai_ipk.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 219,   "data": [     {       "nim": "19523064",       "nama": "NURASTUTI WIJARENI",       "ipk": 4,       "sks_kum": 0,       "dosen": " Sri Mulyati S.Kom., M.Kom."     },     {       "nim": "19523201",       "nama": "MUHAMMAD RIKO BEDIATRA",       "ipk": 4,       "sks_kum": 0,       "dosen": " BENI SURANTO S.T., M.Soft.Eng."     }   ] } </pre>

u. *Request* dan *Response* Cetak Daftar IPS Mahasiswa

API ini digunakan untuk mencetak daftar mahasiswa dengan rentang IPS tertentu. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.74.

Tabel 3.74 Rancangan *Request* dan *Response* Cetak Daftar IPS Mahasiswa

<b>Method</b>	GET
<b>URL</b>	localhost:1002/public/api/v1/print/mahasiswaIPS
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_organisasi: ad081f99-f154-11e7-b2cf-005056806fe5 -uuid_kurikulum: f5acca68-2097-11ea-889a-506b8da96a87 -mulai_ipk: 3.5 -sampai_ipk: 4
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, uuid_organisasi, uuid_kurikulum, mulai_ipk dan sampai_ipk.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 355,   "data": [     {       "nim": "17523055",       "nama": "MUHAMMAD PANDU WIDODO",       "ipk_smt": 4,       "sks_ambil": 22,       "dosen": " Andhik Budi Cahyono S.T., M.T."     },     {       "nim": "18523130",       "nama": "ABYAN AHMAD NURRASYID",       "ipk_smt": 4,       "sks_ambil": 24,       "dosen": " Fayruz Rahma S.T., M.Eng."     }   ] } </pre>

### 3.3.6 Desain *Service Assessment*

*Service* ini merupakan *service* yang berfokus pada pengelolaan nilai mahasiswa dan penyerahan nilai ujian. Adapun RESTful API yang dibuat dalam *service* ini sebagai berikut:

a. *Request* dan *Response* Periode

API ini digunakan untuk mendapatkan uuid dan nama periode yang aktif. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.75.

Tabel 3.75 Rancangan *Request* dan *Response* Periode

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/periode
<b>Header</b>	-
<b>Params</b>	-
<b>Keterangan</b>	<i>Response</i> adalah daftar semua uuid dan nama periode di UII.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 6,   "data": [     {       "uuid": "0465d84a-31bd-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Konversi"     },     {       "uuid": "f95b3adb-31bc-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Remediasi Genap"     },     {       "uuid": "ea526fb7-31bc-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Remediasi Ganjil"     },     {       "uuid": "da20e7f0-31bc-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Pendek"     },     {       "uuid": "c35fa7d7-31bc-11ea-ae12-506b8da96a87",       "semester": "2019/2020 - Genap"     },     {       "uuid": "9f03fa99-ad01-11e9-946e-506b8da96a87",       "semester": "2019/2020 - Ganjil"     }   ] } </pre>

b. *Request* dan *Response* Program Studi

API ini digunakan untuk mendapatkan uuid, nama, dan kode program studi. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.76.

Tabel 3.76 Rancangan *Request* dan *Response* Program Studi

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/prodi
<b>Header</b>	- x-organization: [{"kd_organisasi": "523"}, {"kd_organisasi": "42"}, {"kd_organisasi": "123"}]
<b>Params</b>	-
<b>Keterangan</b>	<i>Response</i> adalah daftar program studi yang bisa diakses oleh pengguna.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 7,   "data": [     {       "uuid": "ad07defc-f154-11e7-b2cf-005056806fe5",       "name": "Ahwal Al-syakhshiyah",       "kd_organisasi": "421"     },     {       "uuid": "ad07d087-f154-11e7-b2cf-005056806fe5",       "name": "Pendidikan Agama Islam",       "kd_organisasi": "422"     },     {       "uuid": "ad07db44-f154-11e7-b2cf-005056806fe5",       "name": "Ekonomi Islam",       "kd_organisasi": "423"     },     {       "uuid": "bf567c0b-c5eb-11e8-8f38-005056804a5c",       "name": "Ahwal Al-syakhshiyah Program Internasional",       "kd_organisasi": "424"     },     {       "uuid": "ad07d79a-f154-11e7-b2cf-005056806fe5",       "name": "Magister Studi Islam",       "kd_organisasi": "913"     },     {       "uuid": "ad07d446-f154-11e7-b2cf-005056806fe5",       "name": "Doktor Hukum Islam",       "kd_organisasi": "933"     },     {       "uuid": "ad081f99-f154-11e7-b2cf-005056806fe5",       "name": "Informatika",       "kd_organisasi": "523"     }   ] } </pre>



c. *Request* dan *Response* Kurikulum

API ini digunakan untuk mendapatkan uuid, dan nama kurikulum pada periode dan organisasi tertentu. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.77.

Tabel 3.77 Rancangan *Request* dan *Response* Kurikulum

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/kurikulum
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_organisasi: ad081f99-f154-11e7-b2cf-005056806fe5
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, dan uuid_organisasi.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 2,   "data": [     {       "uuid": "f5acca68-2097-11ea-889a-506b8da96a87",       "nama_kurikulum": "Kurikulum 2016"     },     {       "uuid": "f590cec1-2097-11ea-889a-506b8da96a87",       "nama_kurikulum": "Kurikulum 2010"     }   ] }</pre>

d. *Request* dan *Response* Matakuliah

API ini digunakan oleh petugas dan dosen tetapi memiliki *response* yang berbeda tergantung perannya. Sebagai petugas, *response* adalah kode dan nama matakuliah. Sedangkan sebagai dosen, *response* adalah nama matakuliah, nama kelas, dan uuid kelas. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.78.

Tabel 3.78 Rancangan *Request* dan *Response* Matakuliah

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/matakuliah
<b>Header</b>	x-member: mustafa
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kurikulum: f5acca68-2097-11ea-889a-506b8da96a87
<b>Keterangan</b>	Pada bagian <i>header</i> perlu mengirimkan x-member. Jika pengguna adalah admin maka wajib mengirim uuid_tahun_akademik dan uuid_kurikulum. Sedangkan jika pengguna adalah dosen maka hanya perlu mengirimkan uuid_tahun_akademik.
<b>Contoh Response</b>	<p><i>Response</i> Petugas:</p> <pre>{   "result": true,   "count": 2,   "data": [     {       "kd_matakuliah": "52311001",       "nama_matakuliah": "Akidah"     },     {       "kd_matakuliah": "52373007",       "nama_matakuliah": "Arsitektur Sistem Operasi"     }   ] }</pre> <p><i>Response</i> Dosen:</p> <pre>{   "result": true,   "count": 2,   "data": [     {       "kd_matakuliah": "52312507",       "nama_matakuliah": "Interaksi Manusia dan Komputer - A",       "uuid_kelas": "11ea36f3-65d7-9026-b4d2-7eb0d4a3c7a0"     },     {       "kd_matakuliah": "52332002",       "nama_matakuliah": "Matematika Lanjut - D",       "uuid_kelas": "11ea36f3-65e4-495a-b4d2-7eb0d4a3c7a0"     }   ] }</pre>

e. *Request* dan *Response* Kelas

API ini hanya digunakan oleh petugas, *response* adalah nama dan uuid kelas. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.79.

Tabel 3.79 Rancangan *Request* dan *Response* Kelas

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/kelas
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kurikulum: f5acca68-2097-11ea-889a-506b8da96a87 -kd_matakuliah: 52311001
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, uuid_kurikulum, dan kd_matakuliah.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 4,   "data": [     {       "uuid_kelas": "11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0",       "nama_kelas": "A"     },     {       "uuid_kelas": "11ea36f3-65e1-49dd-b4d2-7eb0d4a3c7a0",       "nama_kelas": "B"     },     {       "uuid_kelas": "11ea36f3-65e1-6dfe-b4d2-7eb0d4a3c7a0",       "nama_kelas": "C"     },     {       "uuid_kelas": "11ea36f3-65e1-9486-b4d2-7eb0d4a3c7a0",       "nama_kelas": "D"     }   ] }</pre>

f. *Request* dan *Response* Detail Kelas

API ini digunakan untuk mendapatkan detail dari kelas yang diinginkan. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.80.

Tabel 3.80 Rancangan *Request* dan *Response* Detail Kelas Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/kelasDetail
<b>Header</b>	-
<b>Params</b>	-uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0 -uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_kelas, dan uuid_tahun_akademik.
<b>Contoh Response</b>	<pre> {   "result": true,   "data": {     "dosen": 1,     "uuid_kelas": "11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0",     "matakuliah": "Akidah - A",     "prodi": "Informatika",     "periode": "2019/2020 - Ganjil",     "tgl_ujian": "2019-11-18",     "tgl_penyerahan": "2019-12-18"   } } </pre>

g. *Request* dan *Response* Daftar Mahasiswa dengan Nilai

API ini digunakan untuk mendapatkan daftar mahasiswa dalam kelas beserta dengna nilai.

Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.81.

Tabel 3.81 Rancangan *Request* dan *Response* Daftar Mahasiswa dengan Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/mahasiswa
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, dan uuid_kelas.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 2,   "data": [     {       "nim": "17523079",       "nama_mahasiswa": "ALFENO MUHAYDEIN NUR",       "nilai": "A",       "user_input": "145230403",       "tgl_input": "2020-01-18 20:42:20",       "user_update": "mustafa",       "tgl_update": "2020-01-20 23:23:03",       "posted": "A"     },     {       "nim": "19523002",       "nama_mahasiswa": "SILVIANUR LAILA RAMADHANI",       "nilai": "A",       "user_input": "145230403",       "tgl_input": "2020-01-18 20:42:21",       "user_update": "mustafa",       "tgl_update": "2020-01-20 23:25:25",       "posted": "A"     }   ] } </pre>

h. *Request* dan *Response* Riwayat Nilai

API ini digunakan untuk melihat riwayat perubahan nilai mahasiswa. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.82.

Tabel 3.82 Rancangan *Request* dan *Response* Riwayat Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/riwayat
<b>Header</b>	
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0 -nim: 19523005
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akadmeik, uuid_kelas dan nim.
<b>Contoh Response</b>	<pre>{   "result": true,   "count": 3,   "data": {     "matakuliah": "Akidah - A",     "periode": "2019/2020 - Ganjil",     "mahasiswa": "FAHRUDIN NASIKH AZ ZUHDU (19523005)",     "riwayat": [       {         "nilai": "A",         "tgl_input": "2020-01-20 23:23:36",         "user_input": "mustafa"       },       {         "nilai": "",         "tgl_input": "2020-01-18 20:47:01",         "user_input": "mustafa"       },       {         "nilai": "D",         "tgl_input": "2020-01-18 20:46:02",         "user_input": "mustafa"       }     ]   } }</pre>

i. *Request* dan *Response* Rekap Nilai

API ini digunakan untuk melihat rekapitulasi nilai mahasiswa dikelompokkan berdasarkan nilai yang didapat. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.83.

Tabel 3.83 Rancangan *Request* dan *Response* Rekap Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/rekap
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik, dan uuid_kelas.
<b>Contoh Response</b>	<pre> {   "result": true,   "data": {     "prodi": "Informatika",     "matakuliah": "Akidah - A",     "periode": "2019/2020 - Ganjil",     "rekap": [       {         "name": "A",         "y": 4       }     ]   } } </pre>

j. *Request* dan *Response* Penyerahan Nilai

API ini digunakan untuk mendapatkan daftar kelas pada periode dan fakultas tertentu beserta tanggal ujian, tanggal penyerahan dan jeda waktu penyerahan. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.84.

Tabel 3.84 Rancangan *Request* dan *Response* Penyerahan Nilai

<b>Method</b>	GET
<b>URL</b>	localhost:1001/public/api/v1/penyerahan
<b>Header</b>	-
<b>Params</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kurikulum: f5acca68-2097-11ea-889a-506b8da96a87
<b>Keterangan</b>	Pada bagian parameter wajib mengirim uuid_tahun_akademik dan uuid_kurikulum.
<b>Contoh Response</b>	<pre> {   "result": true,   "count": 92,   "data": [     {       "nik": "145230403",       "nama": "Ahmad Fathan Hidayatullah, S.T., M.Cs.",       "matakuliah": "Matematika Lanjut - D",       "tgl_ujian": "2020-01-01",       "tgl_penyerahan": "2020-01-20",       "jeda_waktu": "19 hari"     },     {       "nik": "045201223",       "nama": "ASS. PRAK. T. INFORM. 4",       "matakuliah": "Students' Soft Skills Development - A",       "tgl_ujian": "2019-11-18",       "tgl_penyerahan": "2019-12-18",       "jeda_waktu": "30 hari"     }   ] } </pre>

k. *Request* dan *Response Store* Nilai

API ini digunakan untuk menyimpan dan mengubah nilai mahasiswa. Adapun *request* dan *response* ini ditunjukkan dalam Tabel 3.85.



Tabel 3.85 Rancangan *Request* dan *Response* Store Nilai

<b>Method</b>	POST
<b>URL</b>	localhost:1001/public/api/v1/store
<b>Header</b>	-x-member: mustafa
<b>Body</b>	-uuid_tahun_akademik: 9f03fa99-ad01-11e9-946e-506b8da96a87 -uuid_kelas: 11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0 -tgl_ujian: 2019-11-19 -tgl_penyerahan: 2019-12-18 -mahasiswa: [{"nim":"17523079","nilai":"A"}, {"nim":"19523002","nilai":"B"}]
<b>Keterangan</b>	Pada bagian body wajib mengirimkan uuid_tahun_akademik, uuid_kelas, tgl_ujian, tgl_penyerahan, dan mahasiswa.
<b>Contoh Response</b>	<pre>{   "result": true,   "data": [],   "info": "2 nilai berhasil diperbarui" }</pre>

### 3.3.7 Desain Pengujian

RESTful API yang sudah dibuat akan diimplementasikan pada *web application*, setelah itu akan dilakukan pengujian menggunakan metode *black-box testing*. Skenario pengujian akan dibagi menjadi 4 yaitu presensi, nilai, penyerahan nilai, dan cetak berkas. Adapun rancangan dari setiap skenario sebagai berikut:

#### a. Skenario Pengujian Presensi

Rancangan skenario ini digunakan untuk menguji fungsionalitas *endpoint* yang berkaitan dengan pengelolaan presensi mahasiswa. Adapun rancangan skenario pengujian dapat dilihat pada Tabel 3.86.

Tabel 3.86 Rancangan Skenario Pengujian Presensi

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
1	Filter Tab Presensi Harian	Petugas	Isi semua <i>filter</i> yang tersedia	Muncul data presensi mahasiswa sesuai dengan <i>filter</i>	
2	Simpan presensi mahasiswa	Petugas	Isi sebagian presensi mahasiswa	Presensi yang terisi akan disimpan sedangkan yang tidak terisi akan disimpan sebagai 'Alpha'	
3	Simpan presensi Mahasiswa	Petugas	Isi seluruh presensi mahasiswa	Presensi tersimpan	
4	Simpan Presensi Mahasiswa	Petugas	Mengubah presensi mahasiswa	Presensi berhasil diubah	
5	Filter Tab Laporan Presensi	Petugas	Isi semua filter yang tersedia	Muncul laporan kehadiran mahasiswa sesuai dengan filter	
6	Laporan Presensi Mahasiswa	Petugas	Mencocokkan isi laporan dengan tab presensi harian	Laporan dan presensi harian sama	
7	Import pertemuan kelas menggunakan file	Petugas	Import file pertemuan kelas menggunakan file *.dbf dan *.xlsx	Pertemuan kelas berhasil disimpan dan muncul pada Tab Presensi Harian	
8	Import pertemuan kelas menggunakan file	Petugas	Import file pertemuan yang tidak sesuai dengan hak akses organisasi	Pertemuan kelas tidak akan disimpan dan muncul pesan 'Tidak memiliki hak akses organisasi'	
9	Filter Tab presensi harian	Dosen	Isi semua filter yang tersedia	Muncul presensi mahasiswa sesuai dengan filter	
10	Simpan presensi mahasiswa	Dosen	Isi sebagian presensi mahasiswa	Presensi yang terisi akan disimpan sedangkan yang tidak terisi akan disimpan sebagai 'Alpha'	
11	Simpan presensi Mahasiswa	Dosen	Isi seluruh presensi mahasiswa	Presensi tersimpan	
12	Simpan Presensi Mahasiswa	Dosen	Mengubah presensi mahasiswa	Presensi berhasil diubah	
13	Filter Tab Laporan Presensi	Dosen	Isi semua filter yang tersedia	Muncul laporan kehadiran mahasiswa sesuai dengan filter	
14	Laporan Presensi Mahasiswa	Dosen	Mencocokkan isi laporan dengan tab presensi harian	Laporan dan presensi harian sama	

b. Skenario Pengujian Nilai

Rancangan skenario ini digunakan untuk menguji fungsionalitas *endpoint* yang berkaitan dengan pengelolaan nilai mahasiswa. Adapun rancangan skenario pengujian dapat dilihat pada Tabel Tabel 3.87.

Tabel 3.87 Rancangan Skenario Pengujian Nilai

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
1	Filter Tab Input Nilai	Petugas	Isi semua <i>filter</i> yang tersedia	Muncul data mahasiswa dan nilai sesuai <i>filter</i> .	
2	Simpan Nilai	Petugas	Isi sebagian nilai kemudian simpan	Nilai yang terisi akan disimpan kemudian tanggal input akan muncul.	
3	Simpan Nilai	Petugas	Simpan tanpa melakukan perubahan sebelumnya	Muncul pesan 'Tidak terjadi perubahan nilai'.	
4	Simpan Nilai	Petugas	Isi nilai yang tidak sesuai ketentuan contoh 'K'.	Muncul pesan nilai tidak sesuai ketentuan dan nilai tidak disimpan.	
5	Simpan Nilai	Petugas	Isi semua kolom nilai kemudian simpan.	Nilai tersimpan, tanggal input muncul, dan muncul pesan banyak data yang berhasil diubah.	
6	Riwayat Nilai	Petugas	Lihat riwayat Nilai Mahasiswa yang belum memiliki nilai.	Riwayat Nilai kosong.	
7	Riwayat Nilai	Petugas	Lihat riwayat Nilai Mahasiswa yang sudah memiliki nilai.	Menampilkan riwayat nilai mahasiswa.	
8	Filter Rekap Nilai	Petugas	Isi semua filter yang tersedia.	Muncul rekap nilai kelas sesuai filter.	
9	Filter Tab Input Nilai	Dosen	Isi semua filter yang tersedia.	Muncul data mahasiswa dan nilai sesuai filter.	
10	Simpan Nilai	Dosen	Isi nilai tidak sesuai ketentuan contoh 'X'	Muncul pesan nilai tidak sesuai ketentuan dan nilai tidak disimpan.	
11	Simpan Nilai	Dosen	Isi sebagian nilai mahasiswa kemudian simpan	Nilai berhasil disimpan dan tanggal input muncul.	
12	Simpan Nilai	Dosen	Isi semua nilai mahasiswa kemudian simpan	Nilai berhasil disimpan, muncul pesan jumlah data yang berhasil disimpan, dan tanggal penyerahan merupakan tanggal saat nilai disimpan semua.	
13	Riwayat Nilai	Dosen	Melihat riwayat nilai mahasiswa yang belum pernah dimasukkan nilai	Tidak muncul riwayat nilai akan tetapi muncul data mahasiswa.	
14	Riwayat Nilai	Dosen	Melihat riwayat nilai mahasiswa yang memiliki nilai.	Muncul detail mahasiswa dan riwayat perubahan nilai.	
15	Filter Rekap Nilai	Dosen	Isi semua filter yang tersedia	Muncul rekap nilai kelas sesuai filter.	

c. Skenario Pengujian Penyerahan Nilai

Rancangan skenario ini digunakan untuk menguji fungsionalitas *endpoint* yang berkaitan dengan daftar penyerahan nilai mahasiswa. Adapun rancangan skenario pengujian dapat dilihat pada Tabel 3.88.

Tabel 3.88 Rancangan Skenario Pengujian Penyerahan Nilai

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
1	<i>Filter</i> Penyerahan Nilai	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul data sesuai <i>filter</i> dan telah diurutkan berdasarkan jeda waktu antara tanggal ujian dan tanggal penyerahan, lalu berdasarkan NIK.	

d. Skenario Pengujian Cetak Berkas

Rancangan skenario ini digunakan untuk menguji fungsionalitas *endpoint* yang berkaitan dengan pencetakan berkas-berkas. Adapun rancangan skenario pengujian dapat dilihat pada Tabel 3.89.

Tabel 3.89 Rancangan Skenario Pengujian Cetak Berkas

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
1	Cetak Berkas Daftar Presensi Mahasiswa	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul Daftar Presensi Mahasiswa sesuai <i>filter</i> .	
2	Cetak Berkas Persentase Kehadiran Mahasiswa setiap Pertemuan	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar kelas dengan persentase kehadiran mahasiswa setiap pertemuan sesuai <i>filter</i> .	
3	Cetak Berkas Kartu Ujian	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar kartu ujian untuk setiap mahasiswa sesuai <i>filter</i> .	
4	Cetak Berkas Laporan Presensi Kehadiran Mahasiswa	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul laporan presensi mahasiswa.	
5	Cetak Berkas Daftar Mahasiswa dengan IPK.	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar mahasiswa beserta IPK sesuai dengan <i>filter</i> .	
6	Cetak Berkas Daftar Mahasiswa dengan IPS.	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar mahasiswa beserta IPS sesuai dengan <i>filter</i> .	
7	Cetak Berkas Presensi Ujian.	Petugas	Isi semua <i>filter</i> yang tersedia untuk per kelas.	Muncul daftar presensi ujian untuk kelas tertentu sesuai <i>filter</i> .	
8	Cetak Berkas Presensi Ujian.	Petugas	Isi semua <i>filter</i> yang tersedia untuk kelas gabungan.	Muncul daftar presensi ujian untuk kelas dan dosen tertentu sesuai <i>filter</i> .	
9	Cetak Berkas Rekapitulasi Beban Mengajar Dosen.	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar dosen sesuai <i>filter</i> beserta jumlah sks beban mengajar dan kelas yang diampu.	
10	Cetak Berkas Persentase Kehadiran Dosen	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul laporan persentase kehadiran dosen sesuai <i>filter</i> .	

## BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

### 4.1 *Coding Phase*

Pada tahapan ini, rancangan dan desain akan mulai direalisasikan ke dalam baris-baris kode program menggunakan *framework* lumen dan basis data Mysql. Berikut hal-hal yang dilakukan ketika berada dalam tahapan ini.:

#### 4.1.1 *Database*

Database yang akan digunakan ada dua macam, yaitu Oracle dan Mysql. Database Oracle merupakan *database* yang sudah tersedia sebelumnya dan digunakan pada aplikasi “W-Simak”. Sedangkan *database* Mysql merupakan basis data yang digunakan pada sistem informasi UIIGateway. Ketika data yang diinginkan tidak ditemukan dalam basis data Mysql, maka pencarian akan dilanjutkan pada basis data Oracle. Untuk mendapatkan data dari Oracle akan menggunakan *service* simak. Sedangkan data yang bersumber dari Mysql akan dikelola menggunakan *service assessment* dan *service presence*.

#### 4.1.2 *Membuat Controller*

*Controller* merupakan kumpulan *method* yang dapat diakses melalui *endpoint*. Setiap *endpoint* akan terhubung dengan sebuah *method*. Adapun *controller* dari setiap *service* dijelaskan sebagai berikut:

- a. *Service presence* terdiri dari tiga *controller* yaitu PresensiController, CetakController, dan FileController. Adapun fungsi dari setiap *controller* bisa dilihat pada Tabel 4.1.

Tabel 4.1 *Controller Service Presence*

No.	Nama Controller	Fungsi	Method
1	PresensiController	Memuat seluruh method yang berhubungan dengan pengelolaan presensi pada sistem.	<ul style="list-style-type: none"> <li>• getPeriode</li> <li>• getProdi</li> <li>• getFakultas</li> <li>• getKurikulum</li> <li>• getMatakuliah</li> <li>• getKelas</li> <li>• getDosen</li> <li>• getKelasDetail</li> <li>• getKelasGabunganDetail</li> <li>• getJmlhPertemuan</li> <li>• getMateri</li> <li>• getMahasiswa</li> <li>• getLaporanPresensi</li> <li>• store</li> </ul>
2	CetakController	Memuat seluruh <i>method</i> yang berhubungan dengan pencetakan berkas-berkas.	<ul style="list-style-type: none"> <li>• getPresensiDosen</li> <li>• getKehadiranKuliah</li> <li>• getKartuUjian</li> <li>• getRekapitulasiBMD</li> <li>• getPresensiUjian</li> <li>• getMahasiswaIPS</li> <li>• getMahasiswaIPK</li> </ul>
3	FileController	Memuat seluruh <i>method</i> yang berhubungan dengan penyimpanan data melalui <i>file</i> .	<ul style="list-style-type: none"> <li>• storeFilePertemuan</li> <li>• storeFilePresensi</li> </ul>

- b. Controller pada *service assessment* memiliki satu *controller* yaitu NilaiController. Adapun fungsi dari Nilai Controller dapat dilihat pada Tabel 4.2.



Tabel 4.2 *Controller Service Assessment*

No.	Nama Controller	Fungsi	Method
1	NilaiController	Memuat seluruh method yang berhubungan dengan pengelolaan nilai mahasiswa pada sistem.	<ul style="list-style-type: none"> <li>• getPeriode</li> <li>• getProdi</li> <li>• getKurikulum</li> <li>• getMatakuliah</li> <li>• getKelas</li> <li>• getKelasDetail</li> <li>• getMahasiswa</li> <li>• getRiwayat</li> <li>• getRekap</li> <li>• getPenyerahan</li> <li>• store</li> </ul>

- c. Controller pada *service* simak memiliki enam *controller* yaitu CetakController, FileController, MainController, NilaiController, PertemuanController, dan PresensiController. Adapun fungsi dari setiap *controller* dapat dilihat pada Tabel 4.3.

Tabel 4.3 *Controller Service Simak*

No.	Nama Controller	Fungsi	Method
1	CetakController	Memuat seluruh <i>method</i> yang berhubungan pencetakan berkas di mana sumber data berasal dari basis data oracle.	<ul style="list-style-type: none"> <li>• getPresensiDosen</li> <li>• getProsentaseKK</li> <li>• getRekapitulasiBMD</li> <li>• getKartuUjian</li> <li>• getPresensiUjian</li> </ul>
2	FileController	Memuat seluruh <i>method</i> yang berhubungan dengan penyimpanan data melalui <i>file</i> ke dalam basis data oracle.	<ul style="list-style-type: none"> <li>• storePertemuan</li> <li>• storePresensi</li> </ul>
3	MainController	Memuat <i>method</i> yang digunakan secara umum untuk memfilter data pada pengelolaan cetak berkas, nilai, dan presensi mahasiswa.	<ul style="list-style-type: none"> <li>• getProdi</li> <li>• getKurikulum</li> <li>• getMatakuliah</li> <li>• getKelas</li> <li>• getDosen</li> </ul>
4	NilaiController	Memuat <i>method</i> utama yang berhubungan dengan pengelolaan nilai dari basis data oracle.	<ul style="list-style-type: none"> <li>• getKelasDetail</li> <li>• getMahasiswa</li> <li>• getRiwayat</li> <li>• getRekap</li> <li>• getPenyerahan</li> <li>• storeAll</li> </ul>
5	PertemuanController	Memuat <i>method</i> yang berhubungan dengan pengelolaan pertemuan kelas pada basis data oracle.	<ul style="list-style-type: none"> <li>• storeAll</li> </ul>
6	PresensiController	Memuat <i>method</i> yang berhubungan dengan pengelolaan presensi mahasiswa pada basis data oracle.	<ul style="list-style-type: none"> <li>• getKelasDetail</li> <li>• getKelasGabunganDetail</li> <li>• getjumlahPertemuan</li> <li>• getMateri</li> <li>• getMahasiswa</li> <li>• getLaporanPresensi</li> <li>• storeAll</li> </ul>

### 4.1.3 Membuat Endpoint

#### a. Endpoint Presence

*Endpoint service presence* merupakan kumpulan dari *endpoint* yang digunakan untuk mengakses *method* yang tersedia pada *Service Presence*. Adapun *endpoint* yang dapat digunakan terlihat pada Gambar 4.1.

```

7  $router->group(['prefix' => '/public/api/v1/'], function() use ($router){
8  // $router->group(['middleware' => 'role'], function() use ($router){
9  $router->group(['middleware' => 'header'], function() use ($router){
10 $router->get('periode', 'PresensiController@getPeriode');
11 $router->get('fakultas', 'PresensiController@getFakultas');
12 $router->get('prodi', 'PresensiController@getProdi');
13 $router->get('kurikulum', 'PresensiController@getKurikulum');
14 $router->get('matakuliah', 'PresensiController@getMatakuliah');
15 $router->get('kelas', 'PresensiController@getKelas');
16 $router->get('dosen', 'PresensiController@getDosen');
17 $router->get('kelasDetail', 'PresensiController@getKelasDetail');
18 $router->get('jmlhPertemuan', 'PresensiController@getJmlhPertemuan');
19 $router->get('materi', 'PresensiController@getMateri');
20 $router->get('mahasiswa', 'PresensiController@getMahasiswa');
21 $router->get('laporan', 'PresensiController@getLaporanPresensi');
22 $router->post('store', 'PresensiController@store');
23 $router->get('kelasGabunganDetail', 'PresensiController@getKelasGabunganDetail');
24
25 $router->group(['prefix' => 'print/'], function($router){
26 $router->get('kehadiranKuliah', 'CetakController@getKehadiranKuliah');
27 $router->get('presensiDosen', 'CetakController@getPresensiDosen');
28 $router->get('rekapitulasiBMD', 'CetakController@getRekapitulasiBMD');
29 $router->get('kartuUjian', 'CetakController@getKartuUjian');
30 $router->get('mahasiswaIPK', 'CetakController@getMahasiswaIPK');
31 $router->get('mahasiswaIPS', 'CetakController@getMahasiswaIPS');
32 $router->get('mahasiswaPresensi', 'CetakController@getMahasiswa');
33 $router->get('presensiUjian', 'CetakController@getPresensiUjian');
34 $router->get('mahasiswa', 'CetakController@getMahasiswa');
35 });
36
37 $router->group(['prefix' => 'file/'], function($router){
38 $router->post('storeFilePresensi', 'FileController@storeFilePresensi');
39 $router->post('storeFilePertemuan', 'FileController@storeFilePertemuan');
40 });
41 });
42 //});
43 });
44

```

Gambar 4.1 *Endpoint service presence*

#### b. Endpoint Assessment

*Endpoint service assessment* merupakan kumpulan dari *endpoint* yang digunakan untuk mengakses *method* yang tersedia pada *Service Assessment*. Adapun *endpoint* yang dapat digunakan terlihat pada Gambar 4.2.

```
7 $router->group(['prefix' => '/public/api/v1/', 'middleware' => 'header'], function() use ($router){
8     //$router->group(['middleware' => 'role'], function() use ($router){
9         $router->get('periode', 'NilaiController@getPeriode');
10        $router->get('prodi', 'NilaiController@getProdi');
11        $router->get('kurikulum', 'NilaiController@getKurikulum');
12        $router->get('matakuliah', 'NilaiController@getMatakuliah');
13        $router->get('mahasiswa', 'NilaiController@getMahasiswa');
14        $router->get('riwayat', 'NilaiController@getRiwayat');
15        $router->get('kelas', 'NilaiController@getKelas');
16        $router->get('kelasDetail', 'NilaiController@getKelasDetail');
17        $router->post('store', 'NilaiController@store');
18        $router->get('rekap', 'NilaiController@getRekap');
19        $router->get('penyerahan', 'NilaiController@getPenyerahan');
20    });
21 };
22
```

Gambar 4.2 *Endpoint Service Assessment*

c. *Endpoint Simak*

*Endpoint service* simak merupakan kumpulan dari *endpoint* yang digunakan untuk mengakses *method* yang tersedia pada *service* simak. Adapun *endpoint* yang dapat digunakan terlihat pada Gambar 4.3.

```

21 $router->group(['prefix'=>'/private/api/v1'], function($router){
22     $router->group(['prefix'=>'/main'], function($router){
23         $router->get('/prodi', 'MainController@getProdi');
24         $router->get('/kurikulum', 'MainController@getKurikulum');
25         $router->get('/matakuliah', 'MainController@getMatakuliah');
26         $router->get('/kelas', 'MainController@getKelas');
27         $router->get('/dosen', 'MainController@getDosen');
28     });
29
30     $router->group(['prefix'=>'/assessment'], function($router){
31         $router->get('/kelasDetail', 'NilaiController@getKelasDetail');
32         $router->get('/mahasiswa', 'NilaiController@getMahasiswa');
33         $router->get('/riwayat', 'NilaiController@getRiwayat');
34         $router->get('/rekap', 'NilaiController@getRekap');
35         $router->get('/penyerahan', 'NilaiController@getPenyerahan');
36         $router->post('/storeAll', 'NilaiController@storeAll');
37         $router->put('/posted', 'NilaiController@posted');
38         $router->put('/unposted', 'NilaiController@unposted');
39         $router->post('/posting', 'NilaiController@posting');
40     });
41
42     $router->group(['prefix'=>'/file'], function($router){
43         $router->post('/storeAssessment', 'FileController@storeNilai');
44         $router->post('/storeAssembly', 'FileController@storePertemuan');
45         $router->post('/storePresence', 'FileController@storePresensi');
46     });
47
48     $router->group(['prefix'=>'/presence'], function($router){
49         $router->get('/kelasDetail', 'PresensiController@getKelasDetail');
50         $router->get('/mahasiswa', 'PresensiController@getMahasiswa');
51         $router->get('/jumlahPertemuan', 'PresensiController@getJumlahPertemuan');
52         $router->get('/materi', 'PresensiController@getMateri');
53         $router->get('/laporan', 'PresensiController@getLaporanPresensi');
54         $router->post('/storeAll', 'PresensiController@storeAll');
55         $router->get('/kelasGabunganDetail', 'PresensiController@getKelasGabunganDetail');
56     });
57
58     $router->group(['prefix'=>'/assembly'], function($router){
59         $router->post('/storeAll', 'PertemuanController@storeAll');
60     });
61
62     $router->group(['prefix'=>'/print'], function($router){
63         $router->get('/presensiDosen', 'CetakController@getPresensiDosen');
64         $router->get('/prosentaseKK', 'CetakController@getProsentaseKK');
65         $router->get('/rekapitulasiBMD', 'CetakController@getRekapitulasiBMD');
66         $router->get('/kartuUjian', 'CetakController@getKartuUjian');
67         $router->get('/mahasiswa', 'CetakController@getMahasiswa');
68         $router->get('/presensiUjian', 'CetakController@getPresensiUjian');
69     });
70 });
71

```

Gambar 4.3 Endpoint Service Simak

## 4.2 Test Phase

Pada fase ini *endpoint* yang sudah dibuat akan diuji fungsionalitasnya menggunakan aplikasi Postman. Dengan harapan *endpoint* sudah dapat memberikan *response* seperti yang telah direncanakan. *Response* yang dihasilkan akan memiliki kode status 200 ketika berhasil, 400 ketika data tidak ditemukan, dan 500 ketika terjadi kesalahan pada sisi *server* atau kode

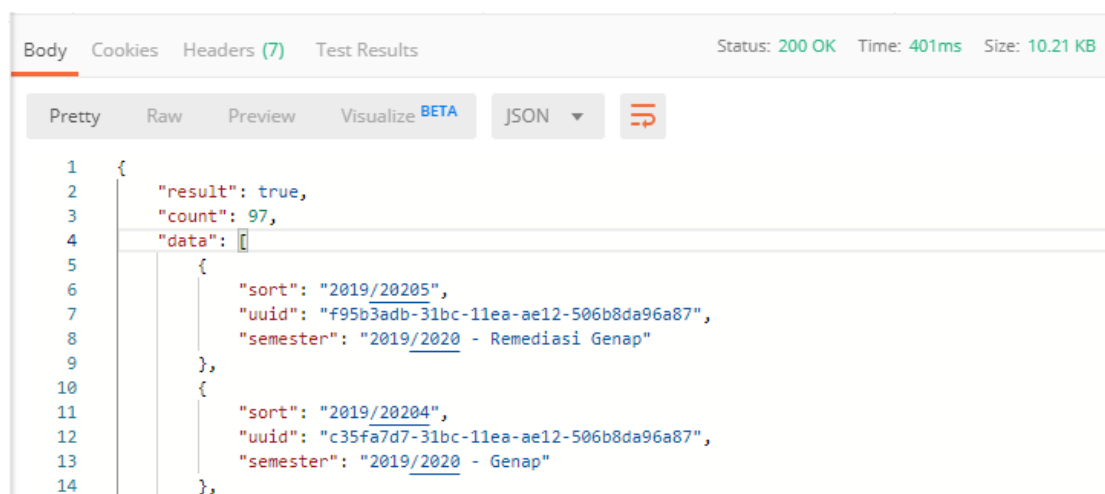
sumber pada *service*. *Service* yang diuji hanya *service assessment* dan *presence*, karena *service* simak tidak digunakan langsung akan tetapi terintegrasi dengan kedua *service* tersebut. *Service* yang sudah dibuat akan menggunakan otentikasi dari badan sistem informasi UII, sehingga tidak diperlukannya pembuatan otentikasi tersendiri. Berikut merupakan hasil pengujian *endpoint* menggunakan aplikasi Postman:

#### 4.2.1 Testing Service Assessment

*Service Assessment* bertujuan untuk mengolah setiap *request* dan *response* yang berhubungan dengan pengelolaan nilai. Setiap *endpoint* yang telah dibuat, hanya bisa digunakan setelah melalui proses otentikasi yang dibuat oleh Badan Sistem Informasi UII. Adapun hasil pengujian menggunakan aplikasi Postman sebagai berikut:

##### a. Response Periode

API ini digunakan untuk mendapatkan *uuid* tahun akademik yang akan digunakan sebagai *filter* pada antarmuka *input* nilai, rekap nilai, penyerahan nilai, presensi harian, laporan presensi, dan cetak berkas. *Web application* dapat mengakses *endpoint* ini melalui *endpoint* yang telah terdaftar pada sistem otentikasi milik Badan Sistem Informasi UII. Untuk menggunakan *endpoint* ini tidak ada parameter yang perlu dikirimkan. *Response* merupakan data periode yang tersedia pada basis data dan terbatas pada semester ganjil, remediasi ganjil, genap, dan remediasi genap. Adapun *response* ketika data periode tersedia dapat dilihat pada Gambar 4.4, dan *response* ketika tidak ada data periode yang ditemukan dapat dilihat pada Gambar 4.5.

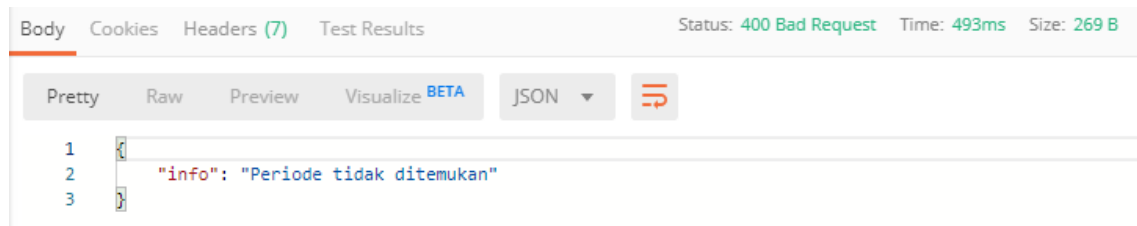


```

1  {
2    "result": true,
3    "count": 97,
4    "data": [
5      {
6        "sort": "2019/20205",
7        "uuid": "f95b3adb-31bc-11ea-ae12-506b8da96a87",
8        "semester": "2019/2020 - Remediasi Genap"
9      },
10     {
11      "sort": "2019/20204",
12      "uuid": "c35fa7d7-31bc-11ea-ae12-506b8da96a87",
13      "semester": "2019/2020 - Genap"
14     },
15   ]
16 }

```

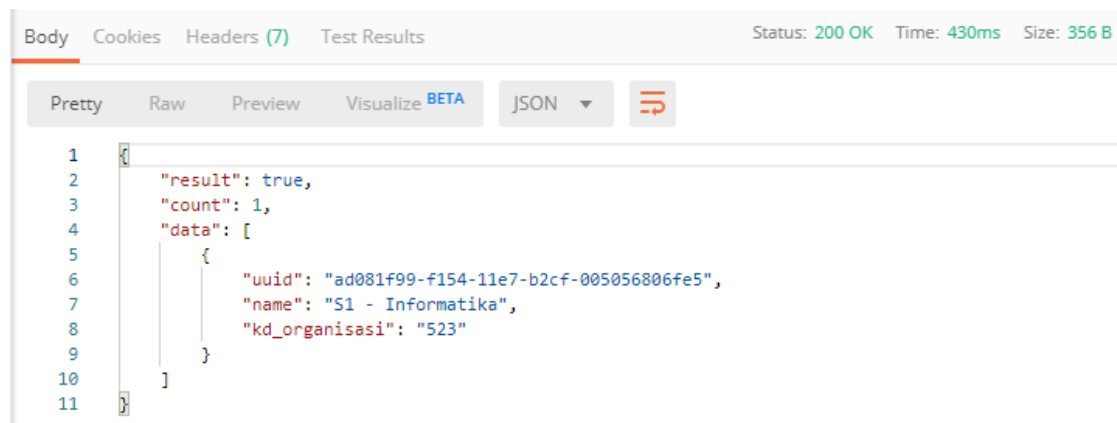
Gambar 4.4 *Response endpoint periode success*



Gambar 4.5 *Response endpoint periode* ketika data tidak ditemukan

b. *Response* Program Studi

API ini digunakan untuk mendapatkan uuid program studi yang akan digunakan untuk *filter* data. Antarmuka yang menggunakan API ini antara lain, antarmuka *input* nilai, rekap nilai, penyerahan nilai, presensi harian, laporan presensi, dan cetak berkas. Data program studi yang dimunculkan bergantung kepada otoritas yang diberikan kepada pengguna melalui layanan UIIPengaturan. Ketika akan mengakses *endpoint* yang telah dibuat, terlebih dahulu *request* dari *web application* akan dikelola oleh *ambassador* setelah itu diteruskan dengan menyertakan kode organisasi yang dapat diakses pada bagian header *request*. Kode organisasi yang diterima berupa kode fakultas ataupun kode jurusan. Kode fakultas akan terlebih dahulu diterjemahkan menjadi kode jurusan, setelah itu dilakukan pengambilan data sesuai dengan hak akses kode jurusan. Adapun detail *response* ketika memiliki kode jurusan dapat dilihat pada Gambar 4.6, dan Gambar 4.7 ketika tidak memiliki otoritas jurusan.



Gambar 4.6 *Response endpoint program studi success*

```

Body Cookies Headers (7) Test Results Status: 400 Bad Request Time: 270ms Size: 333 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "message": "Isian yang diberikan tidak valid",
3   "info": "Tidak memiliki otoritas organisasi manapun"
4 }

```

Gambar 4.7 *Response endpoint* program studi ketika pengguna tidak memiliki otoritas

c. *Response Kurikulum*

API ini digunakan untuk mendapatkan uuid kurikulum yang akan digunakan saat *filter* data. Antarmuka yang menggunakan API ini, antara lain antarmuka *input* nilai, rekap nilai, penyerahan nilai, presensi harian, laporan presensi, dan cetak berkas. *Endpoint* ini memerlukan dua parameter yang dikirimkan, yaitu uuid tahun akademik dan uuid jurusan. Setelah itu, *response* merupakan uuid kurikulum pada tahun dan jurusan yang diinginkan. Adapun detail *response* ketika data kurikulum ditemukan dapat dilihat pada Gambar 4.8, dan Gambar 4.9 ketika data kurikulum tidak ditemukan.

```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 475ms Size: 424 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": true,
3   "count": 2,
4   "data": [
5     {
6       "uuid": "f5acca68-2097-11ea-889a-506b8da96a87",
7       "nama_kurikulum": "Kurikulum 2016"
8     },
9     {
10      "uuid": "f590cec1-2097-11ea-889a-506b8da96a87",
11      "nama_kurikulum": "Kurikulum 2010"
12    }
13  ]
14 }

```

Gambar 4.8 *Response endpoint* kurikulum *success*

```

Body Cookies Headers (7) Test Results Status: 400 Bad Request Time: 899ms Size: 271 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "info": "Kurikulum tidak ditemukan"
3 }

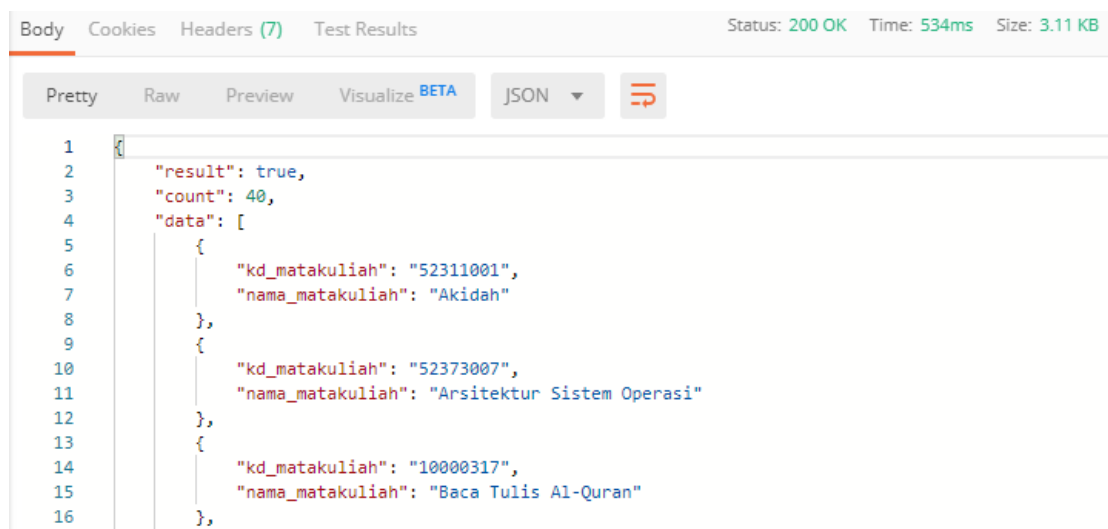
```

Gambar 4.9 *Response endpoint* kurikulum ketika data tidak ditemukan



d. *Response* Matakuliah

API ini memiliki dua macam *response*, yaitu *response* untuk dosen dan petugas. Sebagai dosen, API ini digunakan untuk mendapatkan kode matakuliah, nama matakuliah, dan uuid kelas yang diampu oleh dosen tersebut. Sebagai petugas, API ini digunakan untuk mendapatkan kode matakuliah, dan nama matakuliah. Untuk membedakan pengguna sebagai dosen atau petugas, terlebih dahulu dilakukan pengecekan melalui NIK atau NIP pengguna. NIK dan NIP ini didapatkan pada bagian *header request* setelah melalui *ambassador authentication*. Adapun API ini digunakan untuk menyaring data yang akan ditampilkan. Antarmuka yang menggunakan API ini antara lain antarmuka *input* nilai, presensi harian, laporan presensi, cetak berkas, dan rekap nilai. Adapun detail *response* dapat dilihat pada Gambar 4.10, Gambar 4.11, dan Gambar 4.12.



```
Body Cookies Headers (7) Test Results Status: 200 OK Time: 534ms Size: 3.11 KB
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": true,
3   "count": 40,
4   "data": [
5     {
6       "kd_matakuliah": "52311001",
7       "nama_matakuliah": "Akidah"
8     },
9     {
10      "kd_matakuliah": "52373007",
11      "nama_matakuliah": "Arsitektur Sistem Operasi"
12    },
13    {
14      "kd_matakuliah": "10000317",
15      "nama_matakuliah": "Baca Tulis Al-Quran"
16    },
17  ],
18 }
```

Gambar 4.10 *Response endpoint* matakuliah *success* sebagai petugas

The screenshot shows a REST client interface with the following details:

- Body: Cookies Headers (7) Test Results
- Status: 200 OK Time: 477ms Size: 519 B
- View options: Pretty Raw Preview Visualize BETA JSON
- Response body (JSON):
 

```

1 {
2   "result": true,
3   "count": 2,
4   "data": [
5     {
6       "kd_matakuliah": "52312507",
7       "nama_matakuliah": "Interaksi Manusia dan Komputer - A",
8       "uuid_kelas": "11ea36f3-65d7-9026-b4d2-7eb0d4a3c7a0"
9     },
10    {
11      "kd_matakuliah": "52332002",
12      "nama_matakuliah": "Matematika Lanjut - D",
13      "uuid_kelas": "11ea36f3-65e4-495a-b4d2-7eb0d4a3c7a0"
14    }
15  ]
16 }
```

Gambar 4.11 *Response endpoint* matakuliah *success* sebagai dosen

The screenshot shows a REST client interface with the following details:

- Body: Cookies Headers (7) Test Results
- Status: 400 Bad Request Time: 921ms Size: 272 B
- View options: Pretty Raw Preview Visualize BETA JSON
- Response body (JSON):
 

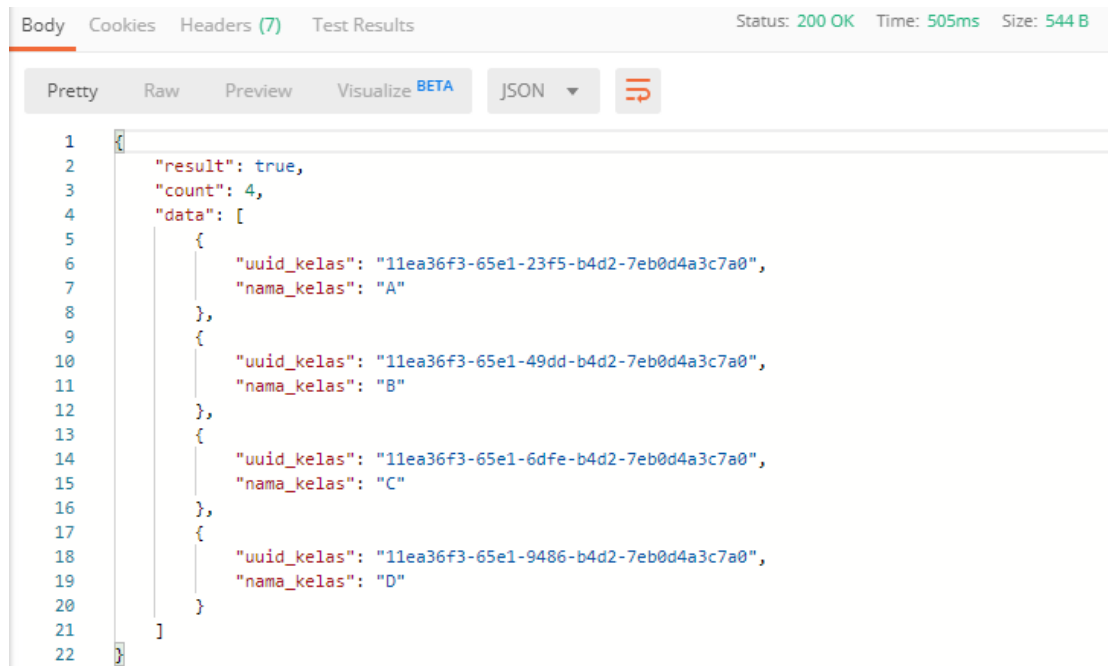
```

1 {
2   "info": "Matakuliah tidak ditemukan"
3 }
```

Gambar 4.12 *Response endpoint* matakuliah ketika data tidak ditemukan

e. *Response* Kelas

API ini digunakan hanya oleh petugas. *Response* yang didapatkan adalah nama kelas dan uuid kelas. Adapun API ini digunakan pada antarmuka *input* nilai, presensi harian, laporan presensi, cetak berkas, dan rekap nilai. Ada tiga parameter yang harus dikirimkan, yaitu uuid kurikulum, uuid kelas, dan kode matakuliah. Adapun detail *response* ketika kelas ditemukan dapat dilihat pada Gambar 4.13, dan ketika kelas tidak ditemukan dapat dilihat pada Gambar 4.14.

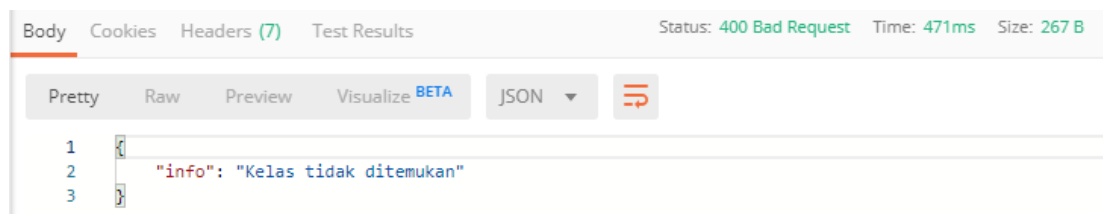


```

1  {
2    "result": true,
3    "count": 4,
4    "data": [
5      {
6        "uuid_kelas": "11ea36f3-65e1-23f5-b4d2-7eb0d4a3c7a0",
7        "nama_kelas": "A"
8      },
9      {
10     "uuid_kelas": "11ea36f3-65e1-49dd-b4d2-7eb0d4a3c7a0",
11     "nama_kelas": "B"
12   },
13   {
14     "uuid_kelas": "11ea36f3-65e1-6dfe-b4d2-7eb0d4a3c7a0",
15     "nama_kelas": "C"
16   },
17   {
18     "uuid_kelas": "11ea36f3-65e1-9486-b4d2-7eb0d4a3c7a0",
19     "nama_kelas": "D"
20   }
21 ]
22 }

```

Gambar 4.13 *Response endpoint kelas success*



```

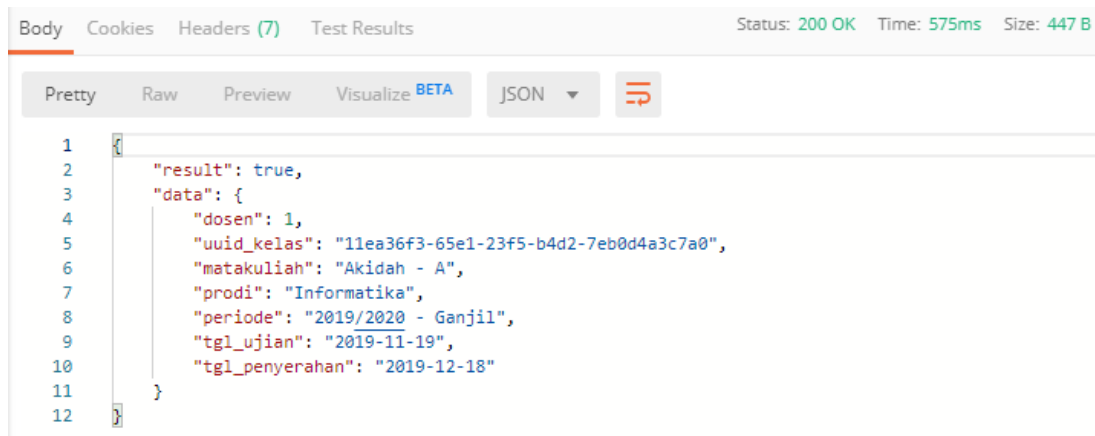
1  {
2    "info": "Kelas tidak ditemukan"
3  }

```

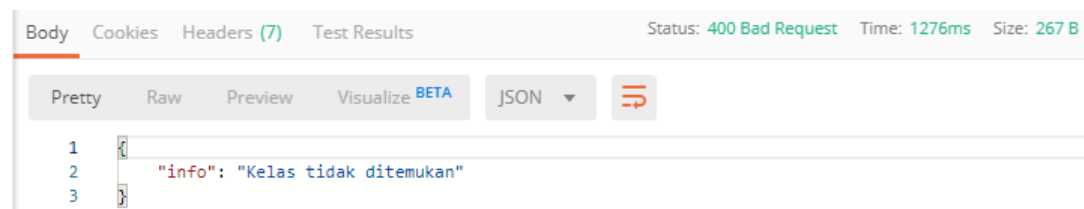
Gambar 4.14 *Response endpoint kelas ketika data tidak ditemukan*

f. *Response Detail Kelas*

API ini digunakan untuk mendapatkan detail dari kelas. Pada saat menggunakannya, perlu mengirimkan uuid kelas, detail kelas akan dicari menggunakan *query* berdasarkan uuid kelas tersebut. Adapun *response* dari API ini yaitu nama matakuliah, nama jurusan, tanggal ujian, tanggal penyerahan, dan periode. API ini akan digunakan pada antarmuka *input* nilai, presensi harian, laporan presensi, dan rekap nilai. Adapun detail *response* dapat dilihat pada Gambar 4.15, dan Gambar 4.16.



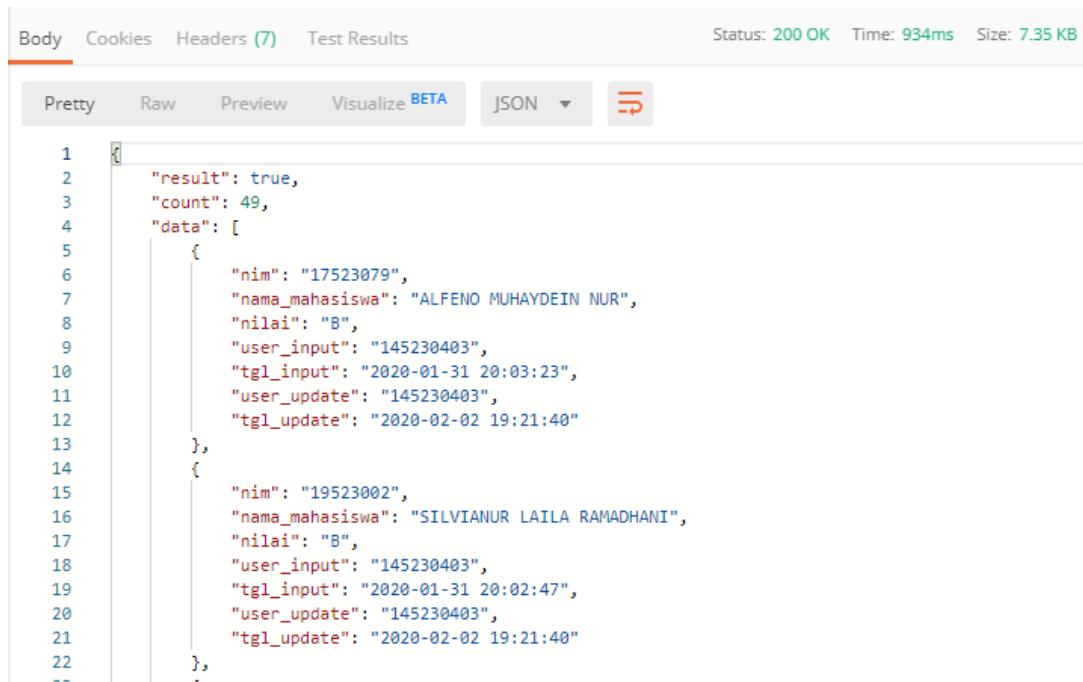
Gambar 4.15 *Response endpoint* detail kelas *success*



Gambar 4.16 *Response endpoint* detail kelas ketika kelas tidak ada

g. *Response* Nilai Mahasiswa

API ini akan digunakan untuk mendapatkan nilai mahasiswa pada kelas tertentu. Antarmuka yang menggunakan API ini yaitu antarmuka *input* nilai. Sebelumnya nilai mahasiswa akan dicek pada basis data mysql, jika tidak ditemukan maka akan dilakukan pengecekan pada basis data oracle melalui *service* simak yang telah dibuat. Adapun detail *response* ketika data ditemukan dapat dilihat pada Gambar 4.17, dan ketika data tidak ditemukan dapat dilihat pada Gambar 4.18.

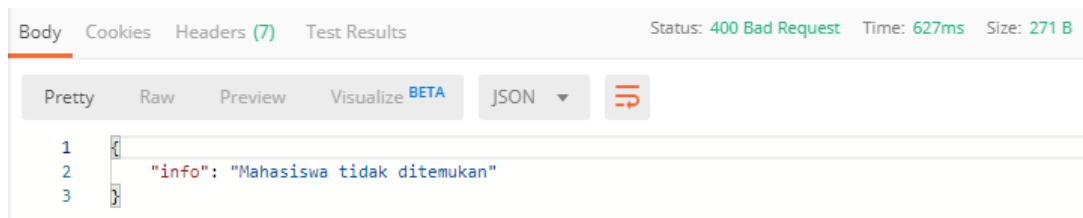


```

1  {
2    "result": true,
3    "count": 49,
4    "data": [
5      {
6        "nim": "17523079",
7        "nama_mahasiswa": "ALFENO MUHAYDEIN NUR",
8        "nilai": "B",
9        "user_input": "145230403",
10       "tgl_input": "2020-01-31 20:03:23",
11       "user_update": "145230403",
12       "tgl_update": "2020-02-02 19:21:40"
13     },
14     {
15       "nim": "19523002",
16       "nama_mahasiswa": "SILVIANUR LAILA RAMADHANI",
17       "nilai": "B",
18       "user_input": "145230403",
19       "tgl_input": "2020-01-31 20:02:47",
20       "user_update": "145230403",
21       "tgl_update": "2020-02-02 19:21:40"
22     },
23   ]
24 }

```

Gambar 4.17 *Response endpoint nilai mahasiswa success*



```

1  {
2    "info": "Mahasiswa tidak ditemukan"
3  }

```

Gambar 4.18 *Response endpoint nilai mahasiswa ketika data tidak ditemukan*

#### h. *Response Riwayat Nilai*

API ini digunakan untuk mendapatkan catatan perubahan nilai mahasiswa. Untuk menggunakan API ini, pengguna perlu mengirimkan uuid kelas dan nim mahasiswa. Setelah itu akan dilakukan pencarian pada database. *Response* yang dikembalikan yaitu nama matakuliah, periode, nama mahasiswa, dan riwayat perubahan nilai mahasiswa. Antarmuka yang menggunakan API ini adalah antarmuka riwayat nilai. Adapun detail *response* berhasil dapat dilihat pada Gambar 4.19, dan *response* tidak ditemukan pada Gambar 4.20.

```

1  {
2    "result": true,
3    "count": 3,
4    "data": {
5      "matakuliah": "Akidah - A",
6      "periode": "2019/2020 - Ganjil",
7      "mahasiswa": "ALFENO MUHAYDEIN NUR (17523079)",
8      "riwayat": [
9        {
10       "nilai": "C",
11       "tgl_input": "2020-02-02 19:21:40",
12       "user_input": "145230403"
13     },
14     {
15       "nilai": "A",
16       "tgl_input": "2020-02-01 15:50:06",
17       "user_input": "mustafa"
18     },
19     {
20       "nilai": "C",
21       "tgl_input": "2020-01-31 20:03:55",
22       "user_input": "145230403"
23     }
24   ]
25 }
26

```

Gambar 4.19 *Response endpoint riwayat nilai success*

```

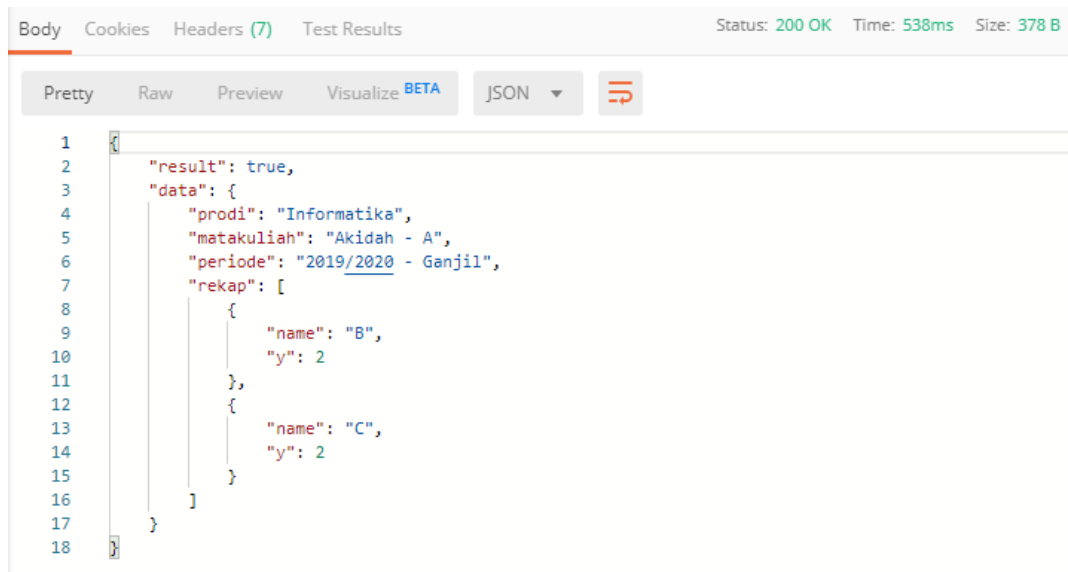
1  {
2    "result": true,
3    "count": 0,
4    "data": {
5      "matakuliah": "Akidah - A",
6      "periode": "2019/2020 - Ganjil",
7      "mahasiswa": "ALFENO MUHAYDEIN NUR (17523079)",
8      "riwayat": []
9    }
10 }

```

Gambar 4.20 *Response endpoint riwayat nilai ketika belum memiliki riwayat*

i. *Response Rekap Nilai*

API ini digunakan untuk mendapatkan rekap nilai mahasiswa dalam suatu kelas. Setiap nilai mahasiswa yang sudah tersimpan di dalam basis data akan dikelompokkan berdasarkan nilainya kemudian dihitung jumlah mahasiswa yang mendapat nilai tersebut. Untuk menggunakan API ini, pengguna perlu mengirimkan uuid kelas. Antarmuka yang menggunakan API ini ialah antarmuka rekap nilai. Adapun detail *response* ketika sudah ada nilai mahasiswa dapat dilihat pada Gambar 4.21, dan ketika belum ada nilai mahasiswa pada kelas tersebut dapat dilihat pada Gambar 4.22.

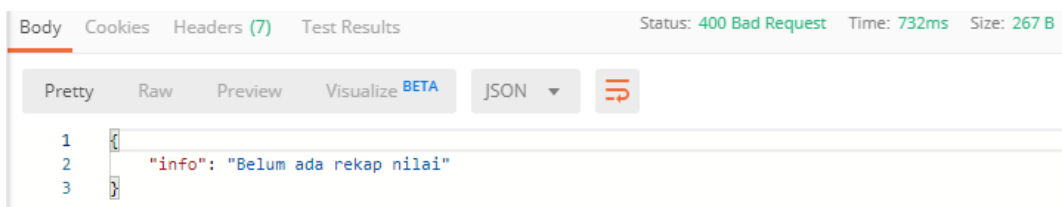


```

1  {
2    "result": true,
3    "data": {
4      "prodi": "Informatika",
5      "matakuliah": "Akidah - A",
6      "periode": "2019/2020 - Ganjil",
7      "rekap": [
8        {
9          "name": "B",
10         "y": 2
11        },
12        {
13          "name": "C",
14          "y": 2
15        }
16      ]
17    }
18  }

```

Gambar 4.21 *Response endpoint* rekap nilai *success*



```

1  {
2    "info": "Belum ada rekap nilai"
3  }

```

Gambar 4.22 *Response endpoint* rekap nilai ketika belum ada data

j. *Response Store Nilai*

API ini digunakan untuk menyimpan nilai mahasiswa. Dalam API ini akan menggunakan API dari *service* Simak yang berguna untuk menyimpan nilai ke dalam basis data oracle. Sebelum nilai disimpan, akan terlebih dahulu dilakukan pengecekan untuk mengetahui kredibilitas nilai. Nilai yang kredibel adalah A, A-, A/B, B+, B, B-, B/C, C+, C, C-, D+, D, E, F. Adapun detail *response* ketika nilai tidak sesuai ketentuan dapat dilihat pada Gambar 4.23, ketika nilai mengalami perubahan pada Gambar 4.24, dan ketika tidak terjadi perubahan pada Gambar 4.25.

```

Body Cookies Headers (7) Test Results Status: 400 Bad Request Time: 1215ms Size: 349 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "message": "Isian yang diberikan tidak valid",
3   "info": "Nilai mahasiswa dengan NIM 17421190 tidak sesuai ketentuan"
4 }

```

Gambar 4.23 *Response endpoint store* nilai ketika nilai tidak sesuai ketentuan

```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 3.44s Size: 288 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": true,
3   "data": [],
4   "info": "1 nilai berhasil diperbarui"
5 }

```

Gambar 4.24 *Response endpoint store* nilai success

```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 3.14s Size: 295 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": false,
3   "data": [],
4   "info": "Nilai tidak mengalami pembaharuan"
5 }

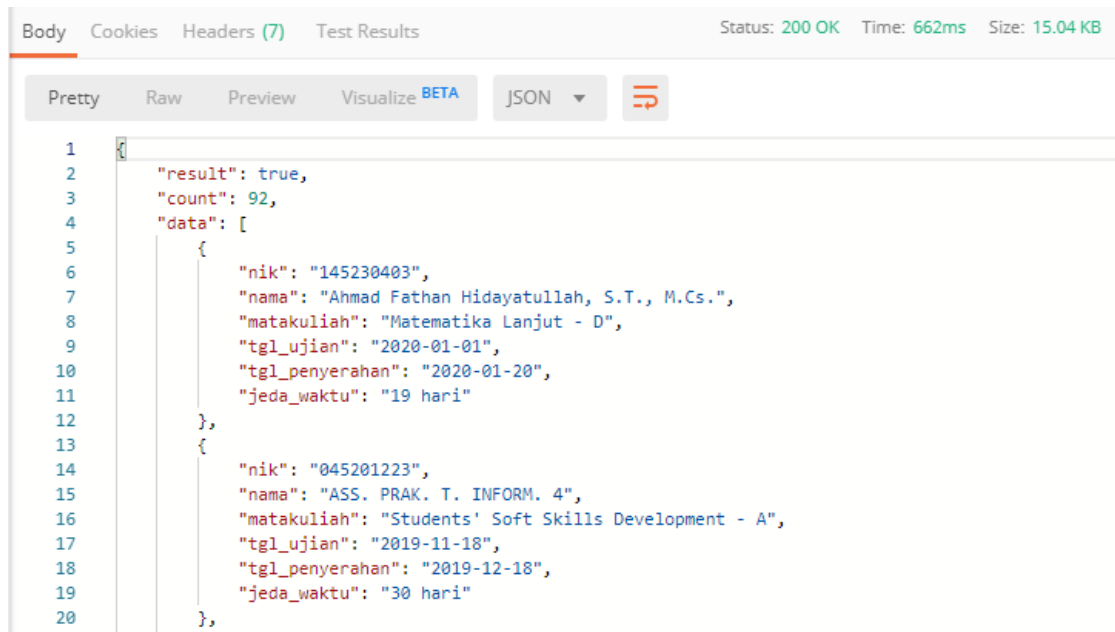
```

Gambar 4.25 *Response endpoint store* nilai ketika tidak terjadi perubahan

k. *Response Penyerahan Nilai*

API ini digunakan untuk menampilkan waktu penyerahan nilai setiap kelas pada periode, program studi dan kurikulum tertentu. Pengguna perlu mengirimkan uuid tahun akademik dan uuid kurikulum. Parameter tersebut akan digunakan untuk mencari data yang diinginkan. Antarmuka yang menggunakan API ini adalah antarmuka penyerahan nilai. Adapun detail *response* ditemukan dapat dilihat pada Gambar 4.26 dan tidak ditemukan pada Gambar 4.27.



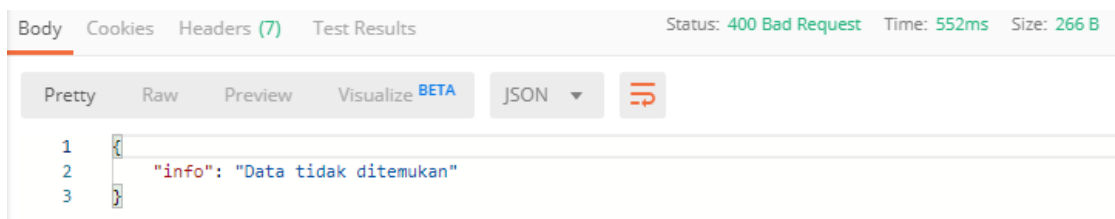


```

1  {
2    "result": true,
3    "count": 92,
4    "data": [
5      {
6        "nik": "145230403",
7        "nama": "Ahmad Fathan Hidayatullah, S.T., M.Cs.",
8        "matakuliah": "Matematika Lanjut - D",
9        "tgl_ujian": "2020-01-01",
10       "tgl_penyerahan": "2020-01-20",
11       "jeda_waktu": "19 hari"
12     },
13     {
14       "nik": "045201223",
15       "nama": "ASS. PRAK. T. INFORM. 4",
16       "matakuliah": "Students' Soft Skills Development - A",
17       "tgl_ujian": "2019-11-18",
18       "tgl_penyerahan": "2019-12-18",
19       "jeda_waktu": "30 hari"
20     }
21   ]
22 }

```

Gambar 4.26 *Response endpoint* penyerahan nilai *success*



```

1  {
2    "info": "Data tidak ditemukan"
3  }

```

Gambar 4.27 *Response endpoint* penyerahan nilai ketika data tidak ditemukan

#### 4.2.2 *Testing Service Presence*

Hasil pengujian untuk *endpoint* periode, program studi, kurikulum, matakuliah dan kelas pada *service presence* memiliki hasil yang sama dengan *service assessment*. Hal ini dikarenakan tujuan *endpoint* yang sama, hanya saja dibuat pada *service* yang berbeda agar setiap *service* tidak saling mempengaruhi. Berikut hasil pengujian *endpoint* dari *service presence*:

##### a. *Response* Jumlah Pertemuan

*API* ini digunakan untuk mendapatkan jumlah pertemuan yang telah dilaksanakan. Parameter yang perlu dikirimkan adalah *uuid\_kelas*. Jika belum ada pertemuan, maka *response* yang diberikan terlihat pada Gambar 4.29. sedangkan *response* ketika sudah ada pertemuan yang terlaksana dapat terlihat pada Gambar 4.28.

The screenshot shows a REST client interface with the following details:

- Body: Pretty, Raw, Preview, Visualize BETA, JSON
- Status: 200 OK, Time: 3.31s, Size: 260 B
- Response Body (JSON):
 

```

1 {
2   "result": true,
3   "jumlahPertemuan": 14
4 }
```

Gambar 4.28 *Response endpoint* jumlah pertemuan *success*

The screenshot shows a REST client interface with the following details:

- Body: Pretty, Raw, Preview, Visualize BETA, JSON
- Status: 200 OK, Time: 812ms, Size: 274 B
- Response Body (JSON):
 

```

1 {
2   "jumlahPertemuan": 0,
3   "info": "Belum ada pertemuan"
4 }
```

Gambar 4.29 *Response endpoint* jumlah pertemuan ketika belum ada pertemuan

b. *Response Materi*

API ini digunakan untuk mendapatkan materi dari pertemuan tertentu. Pengguna perlu mengirimkan uuid kelas dan pertemuan yang ingin diambil. Adapun detail *response* ketika pertemuan telah terlaksana dapat dilihat pada Gambar 4.30, dan ketika belum terlaksana pada Gambar 4.31.

The screenshot shows a REST client interface with the following details:

- Body: Pretty, Raw, Preview, Visualize BETA, JSON
- Status: 200 OK, Time: 470ms, Size: 336 B
- Response Body (JSON):
 

```

1 {
2   "result": true,
3   "data": {
4     "uuid_pertemuan": "6e4d84b9-43ca-11ea-b751-7eb0d4a3c7a0",
5     "pertemuan_ke": 1,
6     "materi": "-"
7   }
8 }
```

Gambar 4.30 *Response endpoint* materi *success*

The screenshot shows a REST client interface with the following details:

- Body: Pretty, Raw, Preview, Visualize BETA, JSON
- Status: 200 OK, Time: 487ms, Size: 286 B
- Response Body (JSON):
 

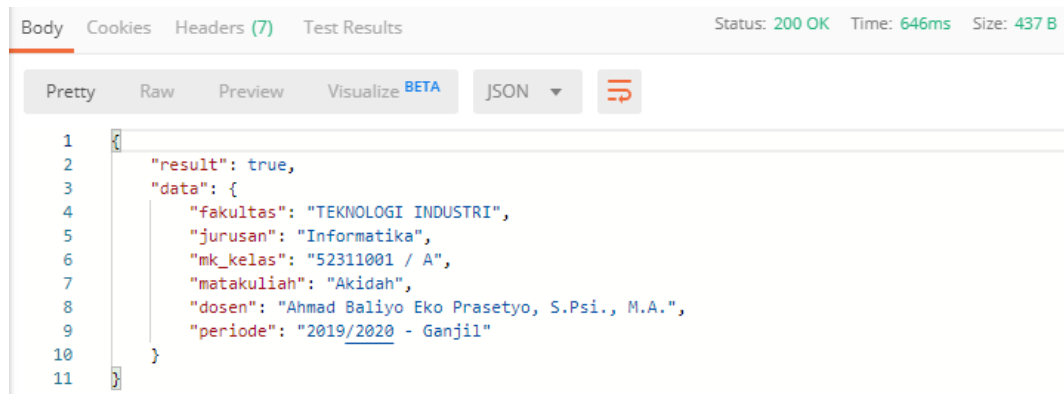
```

1 {
2   "result": true,
3   "data": [],
4   "info": "Belum ada pertemuan ke 15"
5 }
```

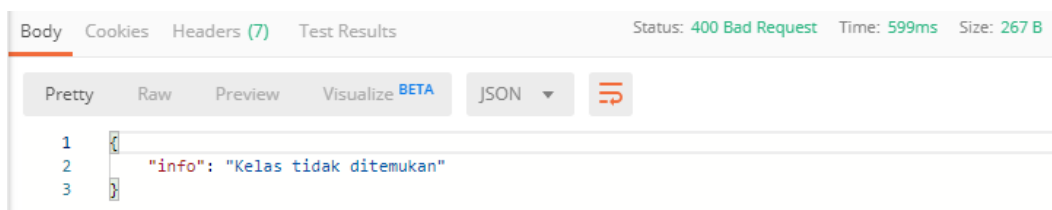
Gambar 4.31 *Response endpoint* materi ketika belum ada pertemuan

c. *Response* Detail Kelas

API ini digunakan untuk mendapatkan detail dari kelas yang ingin diakses. Pengguna akan mendapatkan *response* detail kelas sesuai dengan uuid kelas yang disertakan ketika menggunakan *endpoint* ini. Adapun detail *response* ketika kelas ditemukan dapat dilihat pada Gambar 4.32, dan Gambar 4.33 ketika kelas tidak ditemukan.



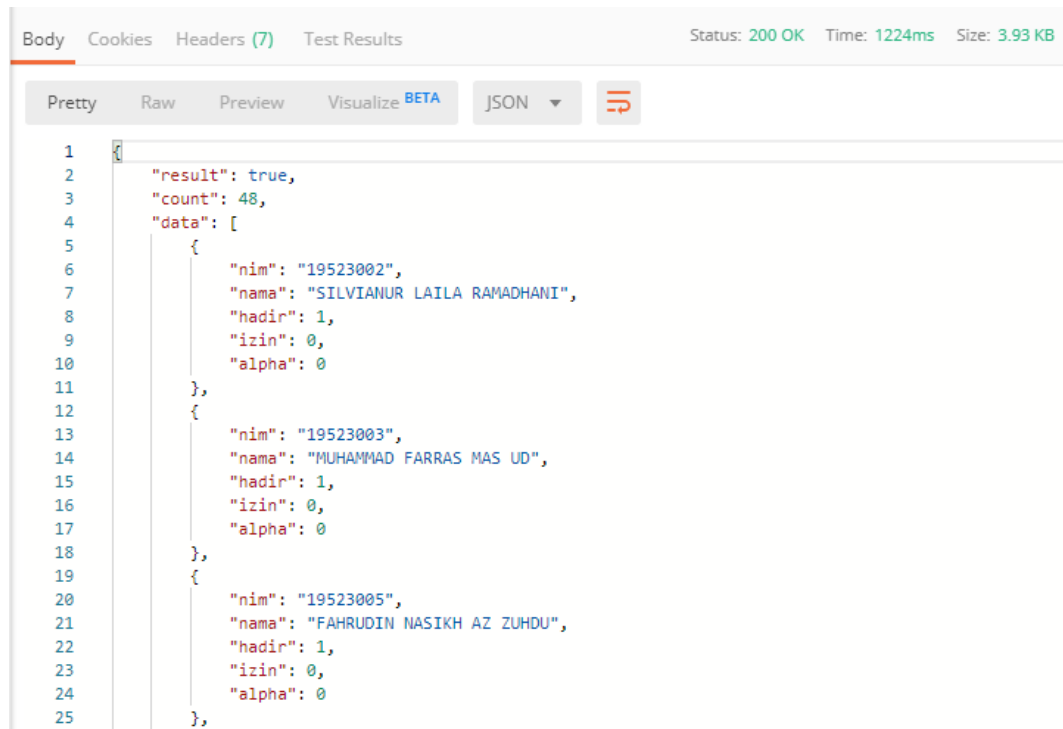
Gambar 4.32 *Response endpoint* detail kelas *success*



Gambar 4.33 *Response endpoint* detail kelas *fail*

d. *Response* Presensi Mahasiswa

API ini digunakan untuk mendapatkan daftar mahasiswa beserta dengan presensi mahasiswa tersebut pada suatu pertemuan. *Response* akan menyesuaikan dengan uuid kelas yang disertakan ketika mengakses *endpoint* ini. Adapun detail *response* ketika kelas tersebut memiliki mahasiswa dapat dilihat pada Gambar 4.34, dan Gambar 4.35 ketika kelas tersebut tidak memiliki mahasiswa.

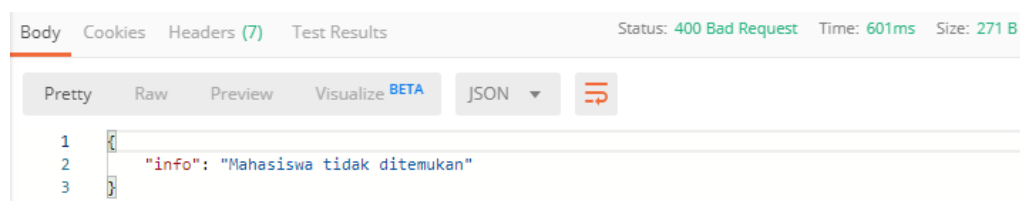


```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 1224ms Size: 3.93 KB
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": true,
3   "count": 48,
4   "data": [
5     {
6       "nim": "19523002",
7       "nama": "SILVIANUR LAILA RAMADHANI",
8       "hadir": 1,
9       "izin": 0,
10      "alpha": 0
11    },
12    {
13      "nim": "19523003",
14      "nama": "MUHAMMAD FARRAS MAS UD",
15      "hadir": 1,
16      "izin": 0,
17      "alpha": 0
18    },
19    {
20      "nim": "19523005",
21      "nama": "FAHRUDIN NASIKH AZ ZUHDU",
22      "hadir": 1,
23      "izin": 0,
24      "alpha": 0
25    }
  ],
}

```

Gambar 4.34 *Response endpoint* presensi mahasiswa success



```

Body Cookies Headers (7) Test Results Status: 400 Bad Request Time: 601ms Size: 271 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "info": "Mahasiswa tidak ditemukan"
3 }

```

Gambar 4.35 *Response endpoint* presensi mahasiswa fail

e. *Response Store* Presensi

API ini digunakan untuk menyimpan presensi mahasiswa dan materi pada pertemuan tertentu. Pertemuan dan presensi mahasiswa juga akan disimpan ke dalam basis data oracle menggunakan *service* simak. Adapun detail *response* ketika berhasil menyimpan presensi mahasiswa dapat dilihat pada Gambar 4.36, dan Gambar 4.37 ketika gagal menyimpan presensi mahasiswa.

The screenshot shows a REST client interface with the following details:

- Body: Cookies Headers (7) Test Results
- Status: 200 OK Time: 7.86s Size: 289 B
- View: Pretty Raw Preview Visualize BETA JSON
- Response Body (JSON):
 

```

1 {
2   "result": true,
3   "data": [],
4   "info": "Presensi berhasil diperbarui"
5 }

```

Gambar 4.36 *Response endpoint store presensi success*

The screenshot shows a REST client interface with the following details:

- Body: Cookies Headers (7) Test Results
- Status: 400 Bad Request Time: 1410ms Size: 271 B
- View: Pretty Raw Preview Visualize BETA JSON
- Response Body (JSON):
 

```

1 {
2   "info": "Presensi gagal diperbarui"
3 }

```

Gambar 4.37 *Response endpoint store presensi fail*f. *Response Laporan Presensi*

API ini digunakan untuk mendapatkan rekap kehadiran mahasiswa selama kegiatan belajar mengajar berlangsung dari pertemuan pertama hingga terakhir. Parameter yang perlu dikirimkan adalah `uuid_kelas`. Adapun detail *response* dapat dilihat pada Gambar 4.38 ketika data sudah tersedia, dan Gambar 4.39 ketika belum ada data.

The screenshot shows a REST client interface with the following details:

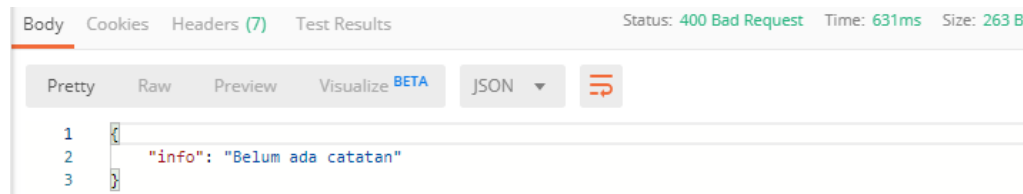
- Body: Cookies Headers (7) Test Results
- Status: 200 OK Time: 784ms Size: 6.23 KB
- View: Pretty Raw Preview Visualize BETA JSON
- Response Body (JSON):
 

```

1 {
2   "result": true,
3   "count": 48,
4   "data": [
5     {
6       "nim": "19523002",
7       "nama": "SILVIANUR LAILA RAMADHANI",
8       "hadir": "2",
9       "izin": "0",
10      "alpha": "0",
11      "total": 14,
12      "porsentase": 14.285714285714285
13    },
14    {
15      "nim": "19523003",
16      "nama": "MUHAMMAD FARRAS MAS UD",
17      "hadir": "2",
18      "izin": "0",
19      "alpha": "0",
20      "total": 14,
21      "porsentase": 14.285714285714285
22    },
23  ]
24 }

```

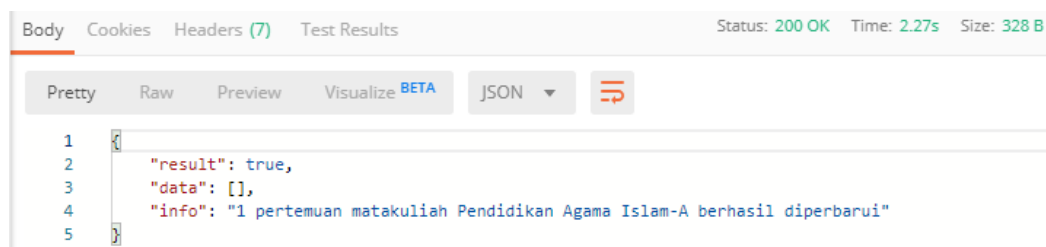
Gambar 4.38 *Response endpoint laporan presensi success*



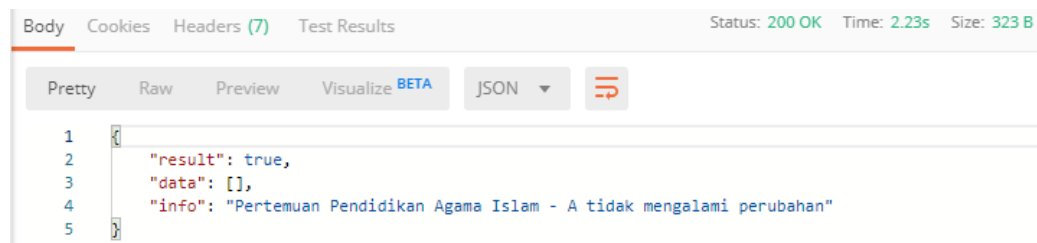
Gambar 4.39 *Response endpoint* laporan presensi *fail*

g. *Response Import* Pertemuan

API ini digunakan untuk menyimpan data pertemuan yang berasal dari *file*. *File* ini akan terlebih dahulu dikelola oleh bagian antarmuka dan dikirimkan ke API dalam bentuk data JSON. Adapun detail *response* dapat dilihat pada Gambar 4.40, dan Gambar 4.41.



Gambar 4.40 *Response endpoint* import pertemuan *success*



Gambar 4.41 *Response endpoint* import pertemuan *fail*

h. *Response Import* Presensi

API ini digunakan untuk menyimpan data presensi yang berasal dari *file*. *File* akan terlebih dahulu dikelola oleh bagian antarmuka kemudian dikirim ke *endpoint* sebagai data JSON. Adapun detail *response* dapat dilihat pada Gambar 4.42, dan 4.43.

```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 3.29s Size: 302 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": true,
3   "data": [],
4   "info": "96 presensi mahasiswa berhasil diperbarui"
5 }

```

Gambar 4.42 *Response endpoint import presensi success*

```

Body Cookies Headers (7) Test Results Status: 400 Bad Request Time: 437ms Size: 333 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "message": "Isian yang diberikan tidak valid",
3   "info": "Tidak memiliki hak akses program studi 523"
4 }

```

Gambar 4.43 *Response endpoint import presensi fail*i. *Response Dosen*

API ini digunakan untuk mendapatkan daftar dosen pada matakuliah tertentu. Pengguna perlu mengirimkan uuid tahun akademik, uuid kurikulum, dan kode matakuliah. Adapun detail *response* dapat dilihat pada Gambar 4.44, dan Gambar 4.45.

```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 694ms Size: 340 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": true,
3   "count": 1,
4   "data": [
5     {
6       "kd_dosen": "125200501",
7       "nama_dosen": "Ahmad Baliyo Eko Prasetyo, S.Psi., M.A."
8     }
9   ]
10 }

```

Gambar 4.44 *Response endpoint dosen success*

```

Body Cookies Headers (7) Test Results Status: 400 Bad Request Time: 447ms Size: 267 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "info": "Dosen tidak ditemukan"
3 }

```

Gambar 4.45 *Response endpoint dosen fail*j. *Response Fakultas*

API ini digunakan untuk mendapatkan daftar fakultas yang bisa diakses oleh pengguna. *Response* akan menyesuaikan dengan hak akses fakultas pengguna yang tersedia pada bagian *header request*. Adapun detail *response* dapat dilihat pada Gambar 4.46, dan Gambar 4.47.

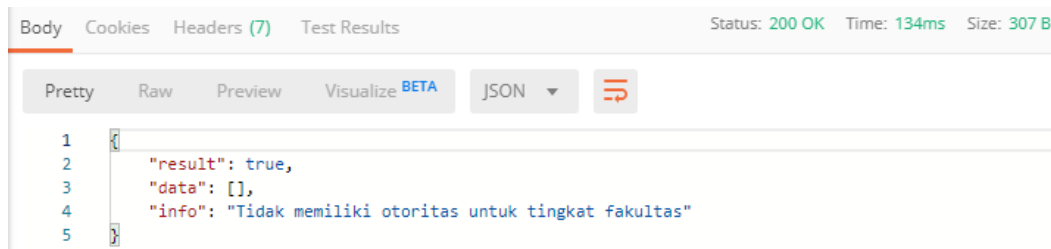


```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 424ms Size: 336 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": true,
3   "count": 1,
4   "data": [
5     {
6       "uuid": "ad07887c-f154-11e7-b2cf-005056806fe5",
7       "name": "Teknologi Industri"
8     }
9   ]
10 }

```

Gambar 4.46 *Response endpoint fakultas success*



```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 134ms Size: 307 B
Pretty Raw Preview Visualize BETA JSON
1 {
2   "result": true,
3   "data": [],
4   "info": "Tidak memiliki otoritas untuk tingkat fakultas"
5 }

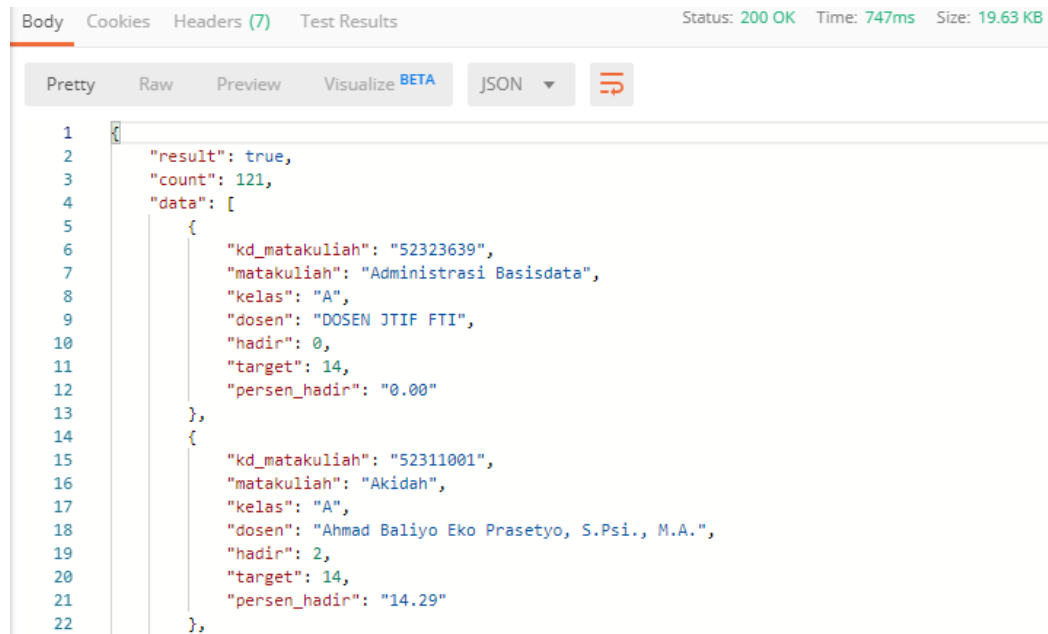
```

Gambar 4.47 *Response endpoint fakultas fail*

k. *Response* Cetak Presensi Dosen

API ini digunakan untuk mendapatkan rekapitulasi kehadiran dosen selama satu periode kegiatan belajar mengajar. Adapun detail *response* dapat dilihat pada Gambar 4.48, dan Gambar 4.49.



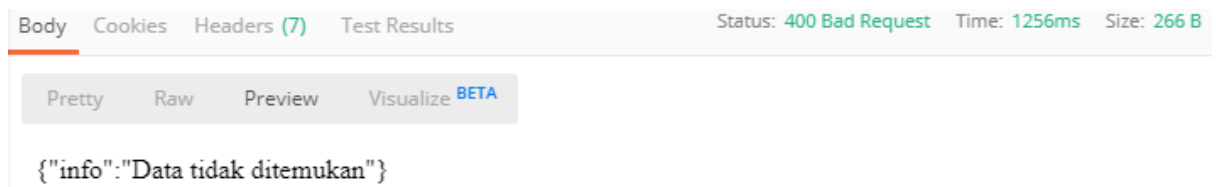


```

1  {
2    "result": true,
3    "count": 121,
4    "data": [
5      {
6        "kd_matakuliah": "52323639",
7        "matakuliah": "Administrasi Basisdata",
8        "kelas": "A",
9        "dosen": "DOSEN JTIF FTI",
10       "hadir": 0,
11       "target": 14,
12       "persen_hadir": "0.00"
13     },
14     {
15       "kd_matakuliah": "52311001",
16       "matakuliah": "Akidah",
17       "kelas": "A",
18       "dosen": "Ahmad Baliyo Eko Prasetyo, S.Psi., M.A.",
19       "hadir": 2,
20       "target": 14,
21       "persen_hadir": "14.29"
22     }
23   ]
24 }

```

Gambar 4.48 *Response endpoint* cetak presensi dosen *success*



```

{"info":"Data tidak ditemukan"}

```

Gambar 4.49 *Response endpoint* cetak presensi dosen *fail*

#### 1. *Response* Cetak Kehadiran Kuliah

API ini digunakan untuk mendapatkan persentase mahasiswa yang hadir dalam setiap pertemuan. Adapun detail *response* dapat dilihat pada Gambar 4.50, Gambar 4.51.

The screenshot shows a REST client interface with the following details:

- Body: Cookies Headers (7) Test Results
- Status: 200 OK Time: 995ms Size: 20.3 KB
- View: Pretty Raw Preview Visualize BETA JSON

```

1  {
2    "result": true,
3    "count": 91,
4    "data": [
5      {
6        "matakuliah": "Akidah",
7        "kelas": "A",
8        "jmlh_mahasiswa": 49,
9        "persentase": [
10       "2.04",
11       "2.04",
12       "-",
13       "-",
14       "-",
15       "-",
16       "-",
17       "-",
18       "-",
19       "-",
20       "-",
21       "-",
22       "-"

```

Gambar 4.50 *Response endpoint* cetak kehadiran kuliah *success*

The screenshot shows a REST client interface with the following details:

- Body: Cookies Headers (7) Test Results
- Status: 400 Bad Request Time: 691ms Size: 276 B
- View: Pretty Raw Preview Visualize BETA JSON

```

1  {
2    "info": "Tidak ada kelas yang ditemukan"
3  }

```

Gambar 4.51 *Response endpoint* cetak kehadiran kuliah *fail*

m. *Response Cetak Kartu Ujian*

API ini digunakan untuk mendapatkan data yang diperlukan ketika mencetak kartu ujian, meliputi biodata mahasiswa, matakuliah yang memiliki ujian, dan foto mahasiswa. Adapun detail *response* dapat dilihat pada Gambar 4.52, dan Gambar 4.53.

```

1  {
2    "result": true,
3    "count": 2,
4    "data": [
5      {
6        "fakultas": "TEKNOLOGI INDUSTRI",
7        "jurusan": "Informatika",
8        "nim": "16523006",
9        "nama_mahasiswa": "YORDAN PATRA SAVIRA",
10       "Smt_TA": "Ganjil 2019/2020",
11       "DPA": " Zainudin Zukhri S.T., M.I.T.",
12       "jumlah_sks": 19,
13       "matakuliah": [
14         {
15           "nama_matakuliah": "Metodologi Perintisan Bisnis",
16           "sks": 3,
17           "nama_kelas": "A",
18           "no_urut": 1,
19           "jadwal": [
20             {
21               "jadwal_kuliah": "Rabu 12:30-14:10 Lab TF 3-2A",
22               "pengampu": "Beni Suranto"
23             }
24           ]
25         }
26       ],
27       "fotoMhs": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD//

```

Gambar 4.52 *Response endpoint* cetak kartu ujian *success*

```

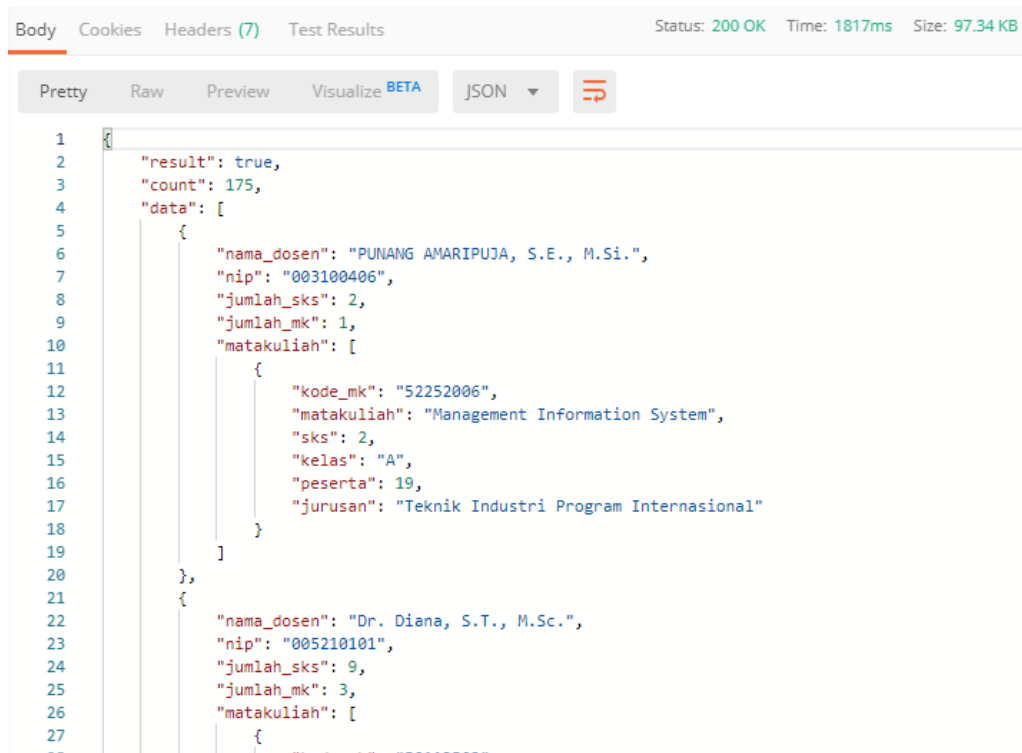
{"info":"Mahasiswa tidak ditemukan"}

```

Gambar 4.53 *Response endpoint* cetak kartu ujian *fail*

n. *Response Cetak Rekapitulasi Beban Mengajar Dosen*

API ini digunakan untuk mendapatkan data yang diperlukan ketika mencetak daftar beban mengajar dosen selama satu periode. Adapun detail *response* dapat dilihat pada Gambar 4.54, dan Gambar 5.55.

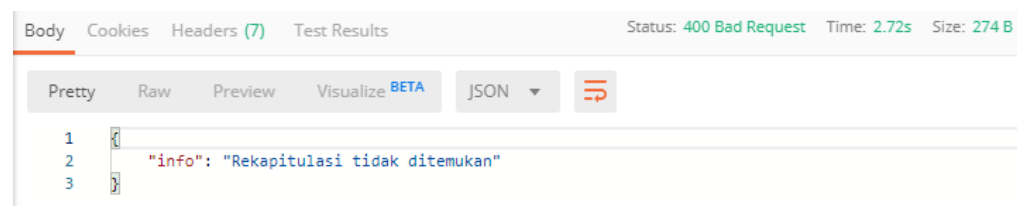


```

1  {
2    "result": true,
3    "count": 175,
4    "data": [
5      {
6        "nama_dosen": "PUNANG AMARIPUJA, S.E., M.Si.",
7        "nip": "003100406",
8        "jumlah_sks": 2,
9        "jumlah_mk": 1,
10       "matakuliah": [
11         {
12           "kode_mk": "52252006",
13           "matakuliah": "Management Information System",
14           "sks": 2,
15           "kelas": "A",
16           "peserta": 19,
17           "jurusan": "Teknik Industri Program Internasional"
18         }
19       ]
20     },
21     {
22       "nama_dosen": "Dr. Diana, S.T., M.Sc.",
23       "nip": "005210101",
24       "jumlah_sks": 9,
25       "jumlah_mk": 3,
26       "matakuliah": [
27         {
28           "kode_mk": "52252006",
29           "matakuliah": "Management Information System",
30           "sks": 2,
31           "kelas": "A",
32           "peserta": 19,
33           "jurusan": "Teknik Industri Program Internasional"
34         }
35       ]
36     }
37   ]
38 }

```

Gambar 4.54 *Response endpoint* cetak rekapitulasi beban mengajar dosen *success*



```

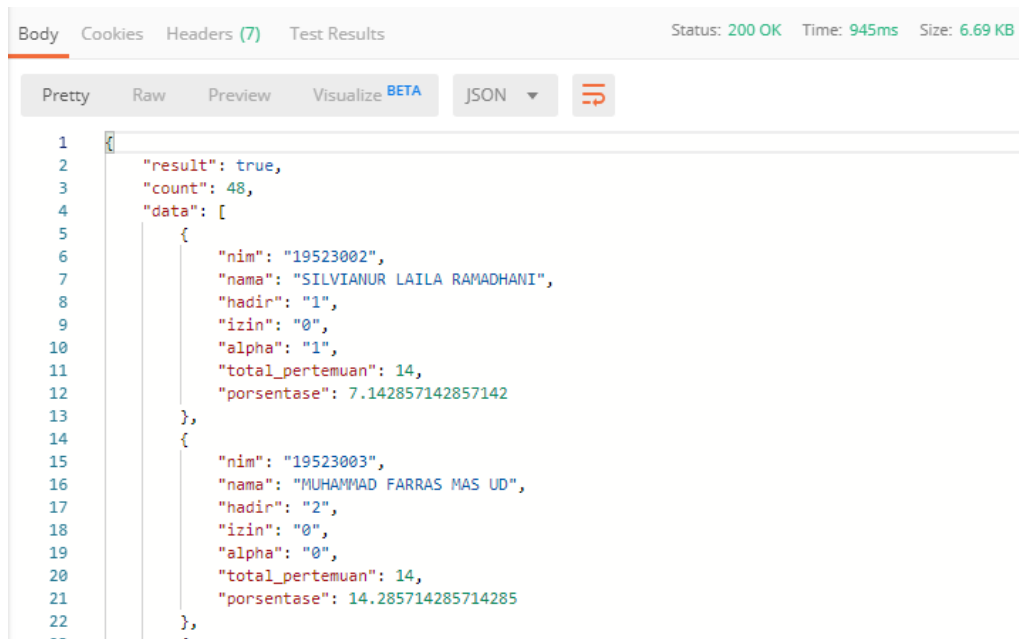
1  {
2    "info": "Rekapitulasi tidak ditemukan"
3  }

```

Gambar 4.55 *Response endpoint* cetak rekapitulasi beban mengajar dosen *fail*

o. *Response Cetak Presensi Ujian*

API ini dibuat untuk mendapatkan data mahasiswa beserta persentase kehadiran mahasiswa selama satu periode. *Response* yang diberikan ada dua macam, yaitu untuk presensi mahasiswa per kelas dan gabungan. Adapun detail *response* dapat dilihat pada Gambar 4.56, Gambar 4.57, dan Gambar 4.58.

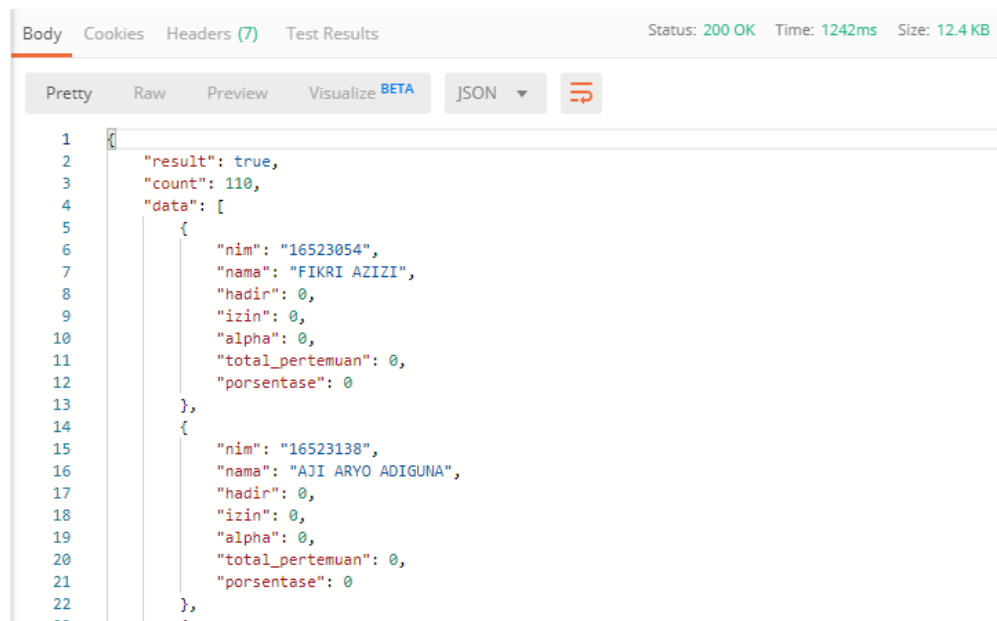


```

1  {
2    "result": true,
3    "count": 48,
4    "data": [
5      {
6        "nim": "19523002",
7        "nama": "SILVIANUR LAILA RAMADHANI",
8        "hadir": "1",
9        "izin": "0",
10       "alpha": "1",
11       "total_pertemuan": 14,
12       "porsentase": 7.142857142857142
13     },
14     {
15       "nim": "19523003",
16       "nama": "MUHAMMAD FARRAS MAS UD",
17       "hadir": "2",
18       "izin": "0",
19       "alpha": "0",
20       "total_pertemuan": 14,
21       "porsentase": 14.285714285714285
22     }
23   ]
24 }

```

Gambar 4.56 *Response endpoint* cetak presensi ujian per kelas *success*

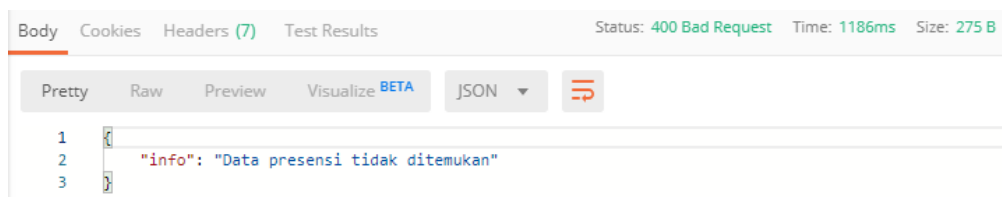


```

1  {
2    "result": true,
3    "count": 110,
4    "data": [
5      {
6        "nim": "16523054",
7        "nama": "FIKRI AZIZI",
8        "hadir": 0,
9        "izin": 0,
10       "alpha": 0,
11       "total_pertemuan": 0,
12       "porsentase": 0
13     },
14     {
15       "nim": "16523138",
16       "nama": "AJI ARYO ADIGUNA",
17       "hadir": 0,
18       "izin": 0,
19       "alpha": 0,
20       "total_pertemuan": 0,
21       "porsentase": 0
22     }
23   ]
24 }

```

Gambar 4.57 *Response endpoint* cetak presensi ujian gabungan *success*



```

1  {
2    "info": "Data presensi tidak ditemukan"
3  }

```

Gambar 4.58 *Response endpoint* presensi ujian *fail*

p. *Response Cetak Daftar Mahasiswa dengan IPK*

API ini digunakan untuk mendapatkan daftar mahasiswa dengan rentang indeks prestasi kumulatif tertentu pada periode yang dipilih. Adapun detail *response* dapat dilihat pada Gambar 4.59, dan Gambar 4.60.

The screenshot shows a REST client interface with the following details:

- Status: 200 OK
- Time: 877ms
- Size: 7.47 KB
- Response body (JSON):
 

```

1  {
2    "result": true,
3    "count": 67,
4    "data": [
5      {
6        "nim": "18523136",
7        "nama": "RISCA NAQUITASIA",
8        "ipk": 3.93,
9        "sks_kum": 0,
10       "dosen": "Fayruz Rahma S.T., M.Eng."
11     },
12     {
13       "nim": "16523055",
14       "nama": "ANNISSA KURNIASARI",
15       "ipk": 3.92,
16       "sks_kum": 0,
17       "dosen": "HENDRIK S.T., M.Eng."
18     }
19   ]
20 }
      
```

Gambar 4.59 *Response endpoint* cetak daftar mahasiswa dengan IPK *success*

The screenshot shows a REST client interface with the following details:

- Status: 400 Bad Request
- Time: 570ms
- Size: 266 B
- Response body (JSON):
 

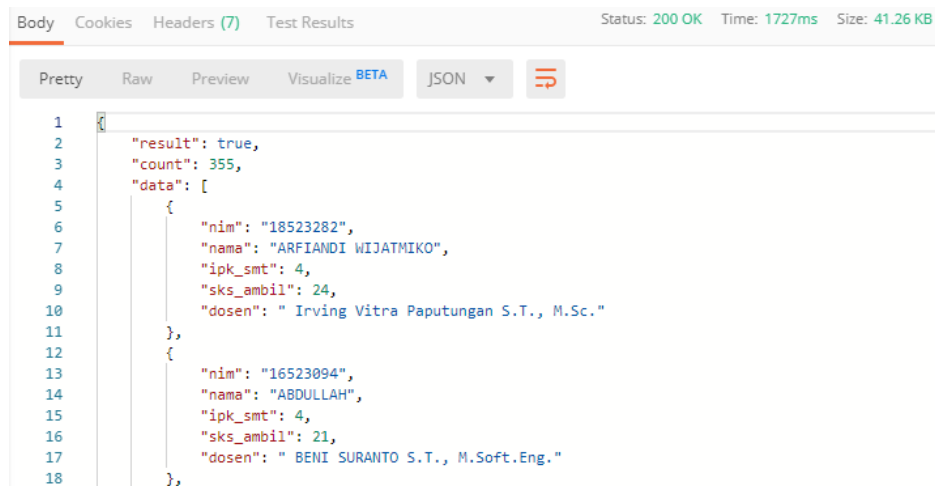
```

1  {
2    "info": "Data tidak ditemukan"
3  }
      
```

Gambar 4.60 *Response endpoint* cetak daftar mahasiswa dengan IPK *fail*

q. *Response Cetak Daftar Mahasiswa dengan IPS*

API ini digunakan untuk mendapatkan daftar mahasiswa pada rentang indeks prestasi semester tertentu pada periode pilihan. Adapun detail *response* dapat dilihat pada Gambar 4.61, dan Gambar 4.62.

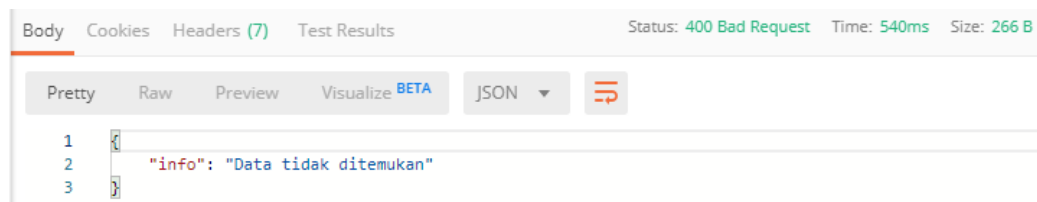


```

1  {
2    "result": true,
3    "count": 355,
4    "data": [
5      {
6        "nim": "18523282",
7        "nama": "ARFIANDI WIJATMIKO",
8        "ipk_smt": 4,
9        "sks_ambil": 24,
10       "dosen": "Irving Vitra Papatungan S.T., M.Sc."
11     },
12     {
13       "nim": "16523094",
14       "nama": "ABDULLAH",
15       "ipk_smt": 4,
16       "sks_ambil": 21,
17       "dosen": "BENI SURANTO S.T., M.Soft.Eng."
18     }
19   ]
20 }

```

Gambar 4.61 *Response endpoint* cetak daftar mahasiswa dengan IPS *success*



```

1  {
2    "info": "Data tidak ditemukan"
3  }

```

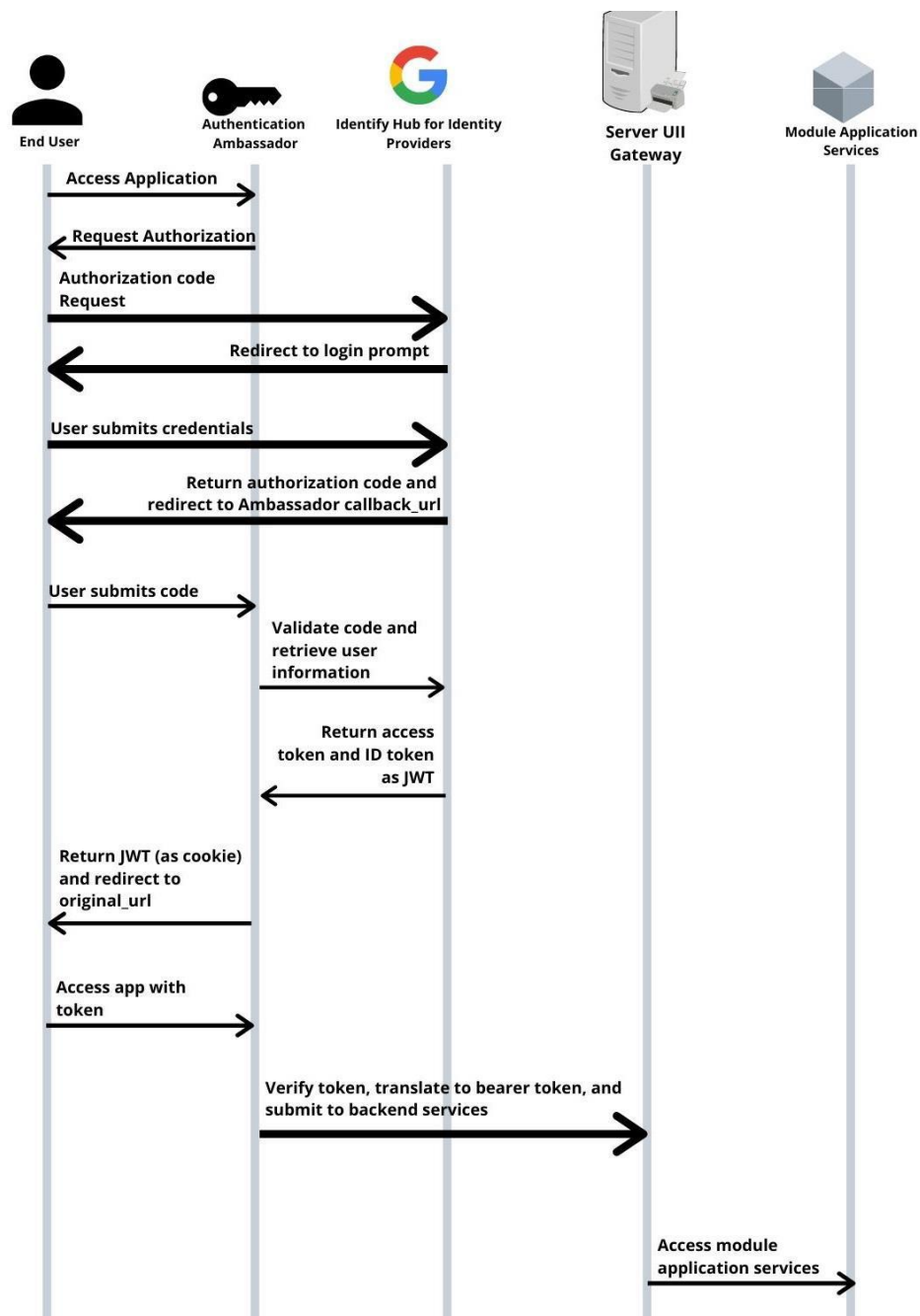
Gambar 4.62 *Response endpoint* cetak daftar mahasiswa dengan IPS *fail*

Berdasarkan pengujian yang telah dilakukan, RESTful API yang dibuat berhasil memenuhi kebutuhan pencetakan berkas, pengelolaan nilai, dan pengelolaan presensi dengan cukup baik. Hal ini dapat dibuktikan dengan hasil pengujian menggunakan aplikasi Postman di mana setiap *endpoint* dapat mengembalikan *response* yang memenuhi kebutuhan sistem informasi UIIPerkuliahan dengan kode status kembalian 200 ketika berhasil, 400 ketika data tidak ditemukan, dan 500 ketika terjadi kesalahan pada *endpoint* (bisa berupa kesalahan kode sumber ataupun parameter yang dikirimkan). Selain itu, dilakukan pengujian menggunakan *black-box testing* pada hasil implementasi RESTful API ke antarmuka Sistem Informasi UIIPerkuliahan, di mana RESTful API dapat memberikan *response* sesuai dengan harapan pada skenario yang telah dibuat.

### 4.3 Deployment Phase

Pada fase ini, *service* yang telah dibuat akan di-*deploy* ke *repository* gitlab milik Badan Sistem Informasi UII, sehingga bisa digunakan oleh *web application*. Setelah *service* berhasil di-*deploy*, agar dapat menggunakan *service* tersebut pengguna harus melalui otentikasi pihak ketiga yang disediakan oleh Badan Sistem Informasi UII yaitu *Ambassador authentication*.

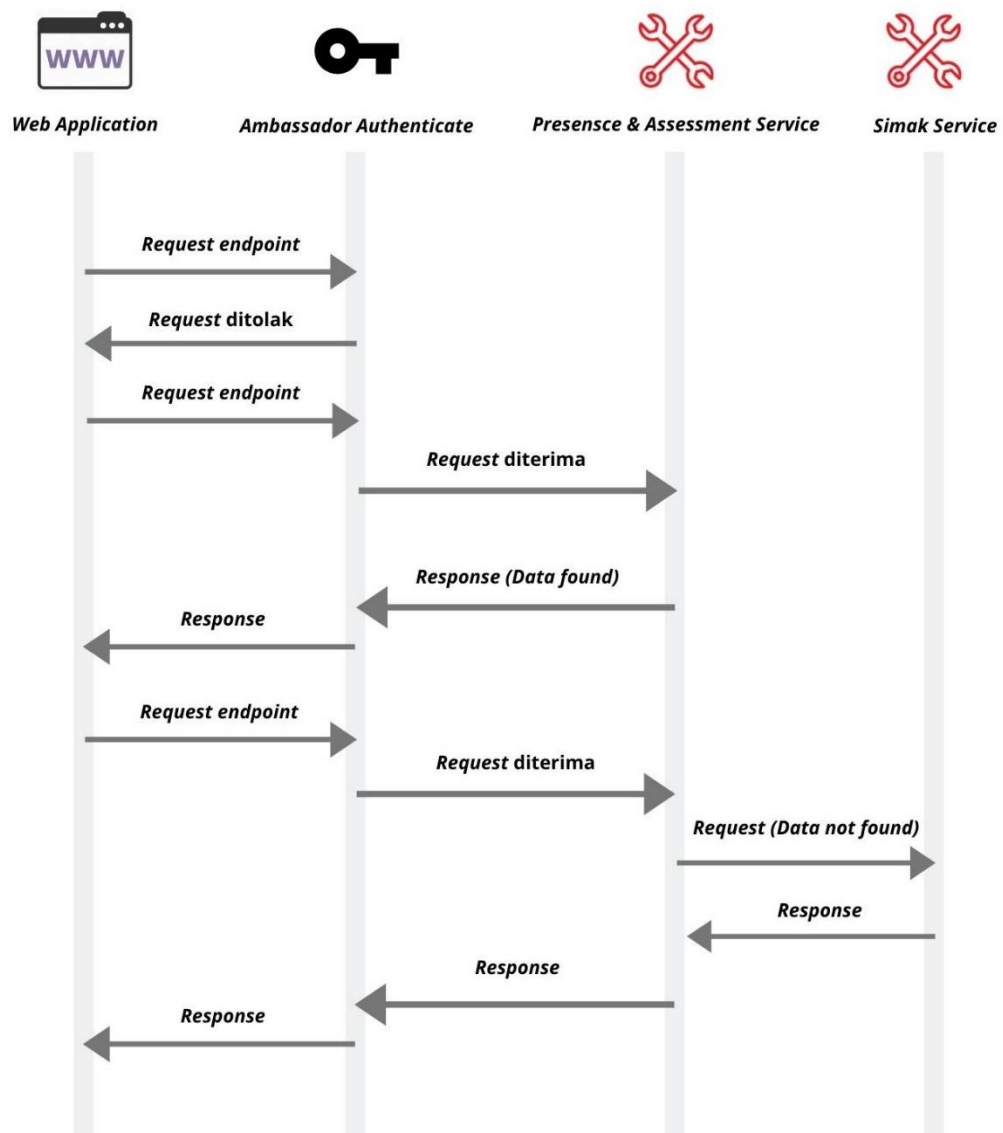
Arsitektur dari proses otentikasi sehingga pengguna dapat menggunakan *service* dapat dilihat pada Gambar 4.63.



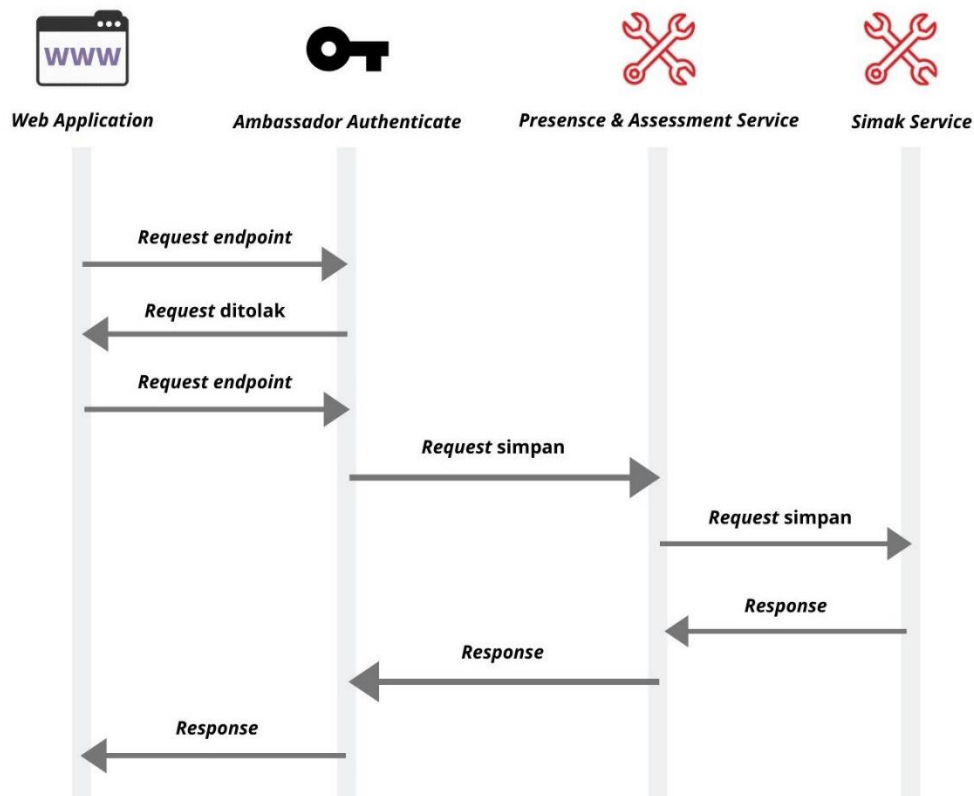
Gambar 4.63 Arsitektur otentikasi di UI Gateway

Selain itu, proses yang terjadi ketika mengambil dan menyimpan data dapat dilihat pada Gambar 4.64, dan Gambar 4.65.





Gambar 4.64 *Flow* mengambil data



Gambar 4.65 Flow menyimpan data

Seperti yang terlihat pada Gambar 4.64, dan Gambar 4.65. *Request* dari *web application* tidak langsung diteruskan ke *service* utama (*service presence* dan *service assessment*). Melainkan harus melewati otentikasi ambassador terlebih dahulu. Jika *request* tersebut memiliki *access token*, maka *request* akan diteruskan ke *service* yang dituju. Sebaliknya, jika *request* tidak memiliki *access token*, maka *request* akan ditolak. *Request* yang diteruskan ke *service* utama akan terlebih dahulu ditambahkan variabel *x-member* pada bagian *request header*, *x-member* ini akan digunakan untuk mengetahui identitas pengguna yang mengirimkan *request*. Untuk proses pengambilan data, ketika data yang diinginkan tidak didapatkan melalui *service* utama, *request* tersebut akan dilanjutkan ke *service* simak agar dilakukan pengambilan data dari basis data oracle. Setelah itu, *response* dari *service* simak akan dikembalikan ke *service* utama kemudian ambassador dan *web application*. Sedangkan untuk proses menyimpan atau mengubah data, *request* yang telah melalui proses otentikasi akan dilanjutkan ke *service* utama kemudian dilanjutkan ke *service* simak. Jika data berhasil di simpan ke basis data oracle, maka data tersebut akan disimpan juga pada basis data mysql melalui *service* utama. Setelah itu, *response* dari *service* utama akan dikembalikan ke ambassador kemudian ke *web application*.

Setelah *service* berhasil di-*deploy*, *service* tersebut akan digunakan oleh *web application*. Antarmuka dari *web application* akan disediakan oleh pihak Badan Sistem Informasi UII dan bukan merupakan bagian yang dikerjakan dalam penelitian ini. Penyertaan gambaran antarmuka hanya sebagai gambaran dari sistem informasi yang akan menggunakan *service* yang telah dibuat. Berikut tampilan dari *web application* yang akan menggunakan *service* tersebut:

#### **4.3.1 Antarmuka *Input* Nilai**

Pada antarmuka ini, petugas dan dosen dapat melihat daftar mahasiswa sebuah kelas berdasarkan *filter* yang tersedia. Selain itu proses *update* dan *store* nilai dilakukan pada halaman ini. Jika ingin menghapus nilai cukup menghapus nilai yang ditampilkan setelah itu menekan tombol simpan. Nilai yang akan dimasukkan akan terlebih dahulu melalui proses validasi, agar nilai yang dimasukkan sesuai ketentuan. Jika petugas yang melakukan *input* nilai maka tanggal penyerahan dimasukkan oleh petugas sesuai dengan tanggal dosen menyerahkan nilai tersebut. Sedangkan jika dosen yang memasukkan nilai, maka tanggal penyerahan adalah tanggal ketika dosen telah memasukkan semua nilai mahasiswa dalam kelas itu. Sehingga, jika ada nilai yang belum dimasukkan, maka tanggal penyerahan tidak akan tersimpan. Antarmuka *input* nilai dapat dilihat pada Gambar 4.66.

UNIVERSITAS SULAIMAN INDONESIA

- Beranda - Nilai

Presensi  
Nilai  
Penyerahan nilai  
Cetak Berkas

Input nilai

Rekap Nilai

Nama matakuliah : Algoritma & Pemrograman 1 - A  
Prodi : Informatika  
Periode : 2018/2019 Ganjil

Pilih mata kuliah

Tanggal ujian : 06/02/2018  
Tanggal penyerahan : 06/02/2018

Template Nilai Import Nilai

Tampilkan 10 data

No.	NIM	Nama	Tanggal input	Nilai	Aksi
1	15628272	Wah Setyo Hutomo	10/02/2018	A/B	
2	15523268	Muhammad Iqbal	10/02/2018	C	
3	15523262	Iham Yogi Nugraha	10/02/2018	A	
4	15523126	Putra Yovana	-	B	
5	15628096	Cholil Firo Fahriza	-	A-	
6	14523189	Wah Sacongko	-	B+	
7	14523027	Muhammad Ihsani	-	B/C	
8	15523266	Fahumzy Akbar Hafidalem	-		
9	15523181	Muhammad Izza Pradana	-		
10	12023292	Reha Setiawan	-		

Menampilkan 1 sampai 10 dari 25 data

Bersihkan data Simpan

Gambar 4.66 Antarmuka *Input Nilai*

### 4.3.2 Antarmuka Riwayat Nilai

Pada antarmuka ini, petugas dan dosen dapat melihat rekam perubahan nilai mahasiswa beserta NIK dari pengguna yang melakukan perubahan. Adapun tampilan antarmuka dapat terlihat pada Gambar 4.67.

No.	Waktu input	Nilai	Operator(NIK)
1	2020-01-20 23:23:03	A	mustafa
2	2020-01-20 13:27:59		145230403
3	2020-01-20 12:02:52	A	mustafa
4	2020-01-20 12:01:48	B	mustafa
5	2020-01-20 10:25:20	A	125200501
6	2020-01-20 10:25:13	B	125200501
7	2020-01-18 20:42:20	A	145230403

Gambar 4.67 Antarmuka Riwayat Nilai

### 4.3.3 Antarmuka Rekap Nilai

Pengguna bisa melihat rekap perolehan nilai dalam sebuah kelas melalui tab rekap nilai pada *menu* nilai. Adapun tampilan antarmuka dapat dilihat pada Gambar 4.68.



Gambar 4.68 Antarmuka Rekap Nilai

### 4.3.4 Antarmuka Presensi Harian

Pengguna dapat memasukkan dan mengubah presensi kehadiran mahasiswa melalui tab presensi harian pada *menu* presensi. Selain itu, pengguna hanya dapat memasukkan data

presensi dan pertemuan secara berurutan dari pertemuan pertama. Adapun tampilan dari tab presensi harian dapat dilihat pada Gambar 4.69.

No.	NIM	Nama mahasiswa	Hadir	Izin	Alpha
1	16523100	Hani Hanum	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	16523101	Muhammad Rizki	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	16523102	Aisyah Rahmah	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	16523100	Hani Hanum	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	16523101	Muhammad Rizki	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Gambar 4.69 Antarmuka Presensi Harian

#### 4.3.5 Antarmuka Laporan Presensi

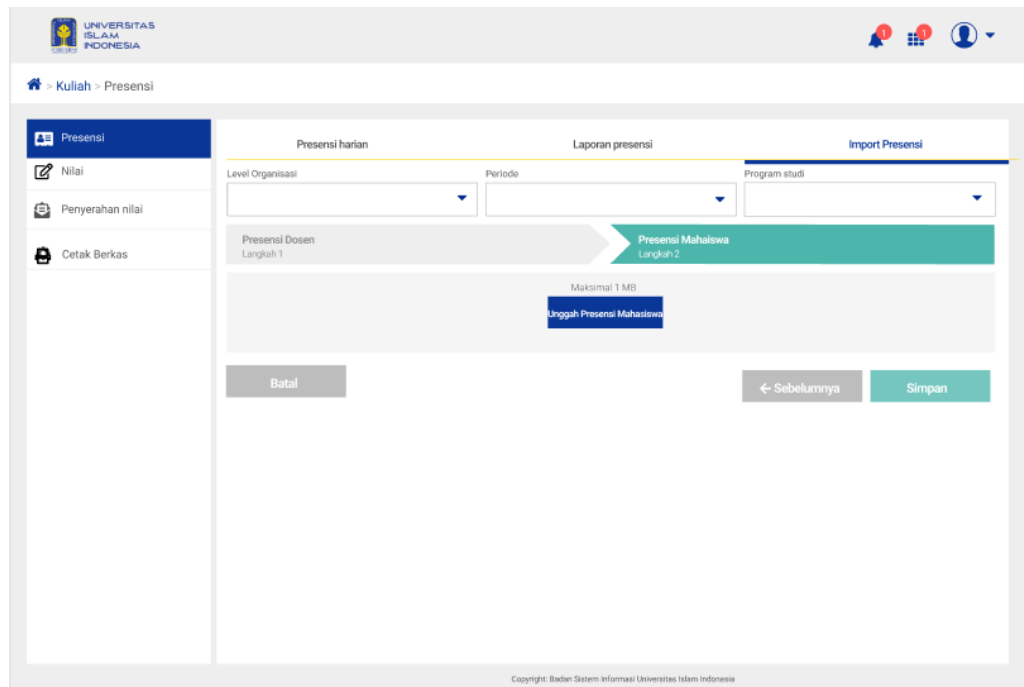
Pengguna dapat melihat rekapitulasi presensi kehadiran mahasiswa dari awal pertemuan melalui tab laporan presensi pada menu presensi. Seperti yang terlihat pada Gambar 4.70, pengguna dapat melihat jumlah kehadiran mahasiswa yang terbagi menjadi jumlah hadir, jumlah izin, jumlah ketidakhadiran, dan total pertemuan yang telah terlaksana.

No.	NIM	Nama mahasiswa	Hadir	Izin	Alpha	Total pertemuan	% Kehadiran
1	16523100	Hani Hanum	14	0	14	14	100
2	16523101	Muhammad Rizki	9	2	2	14	64
3	16523102	Aisyah Rahmah	10	1	1	14	71

Gambar 4.70 Antarmuka Laporan Presensi

#### 4.3.6 Antarmuka *Import Presensi*

Pengguna dapat memasukkan detail pertemuan dan presensi mahasiswa melalui tab *import presensi* pada menu presensi. *File* yang diunggah harus dalam format \*.dbf atau \*.xlsx dan berisi data-data yang telah ditentukan. Adapun tampilan dari tab presensi dapat dilihat pada Gambar 4.71.



Gambar 4.71 Antarmuka Import Presensi dan Pertemuan

#### 4.3.7 Antarmuka Penyerahan Nilai

Pada antarmuka ini, pengguna dapat melihat jeda waktu antara tanggal ujian dan penyerahan nilai dari setiap kelas. Adapun antarmuka bisa dilihat pada Gambar 4.72.

UNIVERSITAS ISLAMIA INDONESIA

UIN Ilmiah > Penyerahan nilai

Penyerahan nilai

Periode: - pilih - Prodi: - pilih - Kurikulum: - pilih -

Tampilkan: 10 data Cari

No.	NIK	Nama	Matakuliah	Tgl. ujian	Tgl. penyerahan	Jeda waktu
1	985240102	Izzati Muhiimah S.T., M.Sc., Ph.D.	Pencitraan Medis - A	06/02/2018	08/02/2018	2 hari
2	985240102	Izzati Muhiimah S.T., M.Sc., Ph.D.	Pencitraan Medis - B	06/02/2018	08/02/2018	2 hari
3	985240102	Izzati Muhiimah S.T., M.Sc., Ph.D.	Pencitraan Medis - C	06/02/2018	08/02/2018	2 hari
4	985240102	Izzati Muhiimah S.T., M.Sc., Ph.D.	Pencitraan Medis - D	06/02/2018	08/02/2018	2 hari
5	945230102	Dr. Sri Kusumadewi S.Si., M.T.	Sistem Pakar - A	06/02/2018	07/02/2018	1 hari
6	945230102	Dr. Sri Kusumadewi S.Si., M.T.	Sistem Cerdas dan Pendukung Keputusan - D	07/02/2018	09/02/2018	2 hari
7	945230102	Dr. Sri Kusumadewi S.Si., M.T.	Sistem Pendukung Keputusan Klinis - A	08/02/2018	09/02/2018	1 hari
8	965240102	Zainudin Zukhri S.T., M.I.T.	Bahasa Indonesia untuk Komunikasi Ilmiah - C	07/02/2018	11/02/2018	4 hari
9	965240102	Zainudin Zukhri S.T., M.I.T.	Bahasa Indonesia untuk Komunikasi Ilmiah - D	07/02/2018	11/02/2018	4 hari
10	965240102	Zainudin Zukhri S.T., M.I.T.	Fundamen Pengembangan Aplikasi - B	10/02/2018	11/02/2018	1 hari

Menampilkan 1 sampai 10 dari 25 data

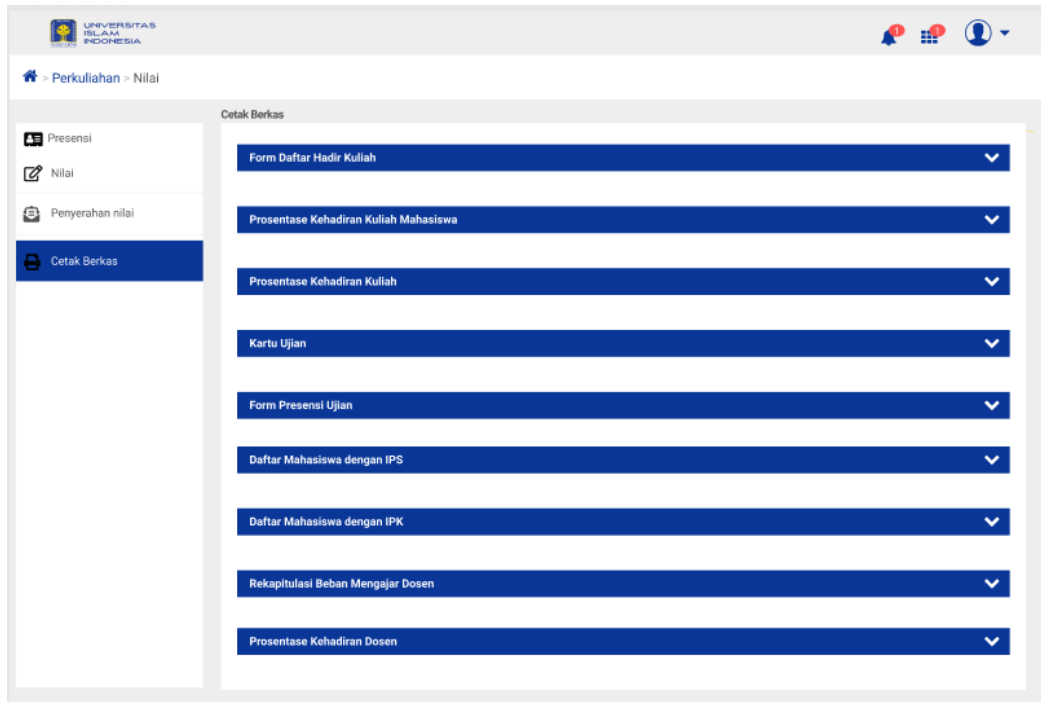
Copyright Badan Sistem Informasi Universitas Islam Indonesia

Gambar 4.72 Antarmuka Penyerahan Nilai

#### 4.3.8 Antarmuka Cetak Berkas

Pengguna dapat melakukan pencetakan berkas melalui menu cetak berkas. Adapun tampilan dari menu cetak berkas dapat dilihat pada Gambar 4.73.





Gambar 4.73 Antarmuka Cetak Berkas

Seperti terlihat pada Gambar 4.73, petugas dapat mencetak beberapa berkas menggunakan *filter* yang disediakan. Berkas yang dapat dicetak antara lain *form* daftar hadir kuliah, persentase kehadiran kuliah mahasiswa, persentase kehadiran kuliah, kartu ujian, *form* presensi ujian, mahasiswa dengan IPK, mahasiswa dengan IPS, rekapitulasi beban mengajar dosen, persentase kehadiran dosen.

#### 4.4 Pengujian Sistem

Pada bagian ini akan dilakukan pengujian RESTful API setelah diimplementasikan pada *web application*. Pengujian dilakukan menggunakan skenario yang telah dibuat pada tahap perancangan, sehingga dapat terlihat kemampuan RESTful API dalam menghadapi berbagai macam input dari pengguna. Berikut hasil pengujian menggunakan skenario pada sistem:

##### 4.4.1 Pengujian bagian Presensi

Skenario yang telah dibuat sebelumnya, akan digunakan untuk menguji hasil implementasi RESTful API pada antarmuka yang berkaitan dengan presensi. Skenario ini bertujuan untuk menguji fungsionalitas bagian presensi dari RESTful API setelah terintegrasi pada *web application*. Adapun hasil pengujian untuk bagian presensi dapat dilihat pada Tabel 4.4.

Tabel 4.4 Skenario Pengujian Presensi

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
1	<i>Filter</i> Tab Presensi Harian	Petugas	Isi semua <i>filter</i> yang tersedia	Muncul data presensi mahasiswa sesuai dengan <i>filter</i>	Valid
2	Simpan presensi mahasiswa	Petugas	Isi sebagian presensi mahasiswa	Presensi yang terisi akan disimpan sedangkan yang tidak terisi akan disimpan sebagai 'Alpha'	Valid
3	Simpan presensi Mahasiswa	Petugas	Isi seluruh presensi mahasiswa	Presensi tersimpan	Valid
4	Simpan Presensi Mahasiswa	Petugas	Mengubah presensi mahasiswa	Presensi berhasil diubah	Valid
5	<i>Filter</i> Tab Laporan Presensi	Petugas	Isi semua <i>filter</i> yang tersedia	Muncul laporan kehadiran mahasiswa sesuai dengan <i>filter</i>	Valid
6	Laporan Presensi Mahasiswa	Petugas	Mencocokkan isi laporan dengan tab presensi harian	Laporan dan presensi harian sama	Valid
7	<i>Import</i> presensi dosen menggunakan <i>file</i>	Petugas	<i>Import file</i> presensi dosen menggunakan <i>file</i> *.dbf dan *.xlsx	Presensi dosen berhasil disimpan dan bisa dilihat pada Tab Presensi Harian	Valid

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
8	<i>Import</i> presensi mahasiswa menggunakan <i>file</i>	Petugas	<i>Import file</i> presensi mahasiswa menggunakan <i>file</i> *dbf dan *xlsx	Presensi mahasiswa berhasil disimpan dan bisa dilihat pada Tab Presensi Harian	Valid
9	<i>Filter</i> Tab presensi harian	Dosen	Isi semua <i>filter</i> yang tersedia	Muncul presensi mahasiswa sesuai dengan <i>filter</i>	Valid
10	Simpan presensi mahasiswa	Dosen	Isi sebagian presensi mahasiswa	Presensi yang terisi akan disimpan sedangkan yang tidak terisi akan disimpan sebagai 'Alpha'	Valid
11	Simpan presensi Mahasiswa	Dosen	Isi seluruh presensi mahasiswa	Presensi tersimpan	Valid
12	Simpan Presensi Mahasiswa	Dosen	Mengubah presensi mahasiswa	Presensi berhasil diubah	Valid
13	<i>Filter</i> Tab Laporan Presensi	Dosen	Isi semua <i>filter</i> yang tersedia	Muncul laporan kehadiran mahasiswa sesuai dengan <i>filter</i>	Valid
14	Laporan Presensi Mahasiswa	Dosen	Mencocokkan isi laporan dengan tab presensi harian	Laporan dan presensi harian sama	Valid

#### 4.4.2 Pengujian bagian Nilai

Pengujian hasil implementasi yang berkaitan dengan nilai akan menggunakan skenario yang telah dibuat pada tahap perancangan. Adapun hasil dari pengujian dapat dilihat pada Tabel 4.5.

Tabel 4.5 Skenario Pengujian Nilai

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
1	<i>Filter Tab Input Nilai</i>	Petugas	Isi semua <i>filter</i> yang tersedia	Muncul data mahasiswa dan nilai sesuai <i>filter</i> .	Valid
2	Simpan Nilai	Petugas	Isi sebagian nilai kemudian simpan	Nilai yang terisi akan disimpan kemudian tanggal input akan muncul.	Valid
3	Simpan Nilai	Petugas	Simpan tanpa melakukan perubahan sebelumnya	Muncul pesan 'Tidak terjadi perubahan nilai'.	Valid
4	Simpan Nilai	Petugas	Isi nilai yang tidak sesuai ketentuan contoh 'K'.	Muncul pesan nilai tidak sesuai ketentuan dan nilai tidak disimpan.	Valid
5	Simpan Nilai	Petugas	Isi semua kolom nilai kemudian simpan.	Nilai tersimpan, tanggal input muncul, dan muncul pesan banyak data yang berhasil diubah.	Valid
6	Riwayat Nilai	Petugas	Lihat riwayat Nilai Mahasiswa yang belum memiliki nilai.	Riwayat Nilai kosong.	Valid
7	Riwayat Nilai	Petugas	Lihat riwayat Nilai Mahasiswa yang sudah memiliki nilai.	Menampilkan riwayat nilai mahasiswa.	Valid
8	<i>Filter Rekap Nilai</i>	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul rekap nilai kelas sesuai <i>filter</i> .	Valid
9	<i>Filter Tab Input Nilai</i>	Dosen	Isi semua filter yang tersedia.	Muncul data mahasiswa dan nilai sesuai filter.	Valid
10	<i>Simpan Nilai</i>	Dosen	Isi nilai tidak sesuai ketentuan contoh 'X'	Muncul pesan nilai tidak sesuai ketentuan dan nilai tidak disimpan.	Valid
11	<i>Simpan Nilai</i>	Dosen	Isi sebagian nilai mahasiswa kemudian simpan	Nilai berhasil disimpan dan tanggal input muncul.	Valid

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
12	<i>Simpan Nilai</i>	Dosen	Isi semua nilai mahasiswa kemudian simpan	Nilai berhasil disimpan, muncul pesan jumlah data yang berhasil disimpan, dan tanggal penyerahan merupakan tanggal saat nilai disimpan semua.	Valid
13	<i>Riwayat Nilai</i>	Dosen	Melihat riwayat nilai mahasiswa yang belum pernah dimasukkan nilai	Tidak muncul riwayat nilai akan tetapi muncul data mahasiswa.	Valid
14	<i>Riwayat Nilai</i>	Dosen	Melihat riwayat nilai mahasiswa yang memiliki nilai.	Muncul detail mahasiswa dan riwayat perubahan nilai.	Valid
15	<i>Filter Rekap Nilai</i>	Dosen	Isi semua filter yang tersedia	Muncul rekap nilai kelas sesuai filter.	Valid

#### 4.4.3 Pengujian bagian Penyerahan Nilai

Pengujian bagian penyerahan nilai akan menggunakan skenario yang telah dibuat pada tahap perancangan. Adapun hasil pengujian dengan skenario dapat dilihat pada Tabel 4.6.

Tabel 4.6 Skenario Pengujian Penyerahan Nilai

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
1	<i>Filter Penyerahan Nilai</i>	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul data sesuai <i>filter</i> dan telah diurutkan berdasarkan tanggal penyerahan, lalu berdasarkan NIK.	Valid

#### 4.4.4 Pengujian bagian Cetak Berkas

Pengujian bagian cetak berkas dilakukan setelah RESTful API berhasil diimplementasikan pada tampilan sistem Informasi UIIPerkuliahan. Adapun hasil pengujian dengan skenario dapat dilihat pada Tabel 4.7.

Tabel 4.7 Skenario Pengujian Cetak Berkas

No.	Fitur	Aktor	Skenario	Kondisi Benar	Valid / Not Valid
1	Cetak Berkas Daftar Presensi Mahasiswa	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul Daftar Presensi Mahasiswa sesuai <i>filter</i> .	Valid
2	Cetak Berkas Persentase Kehadiran Mahasiswa setiap Pertemuan	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar kelas dengan persentase kehadiran mahasiswa setiap pertemuan sesuai <i>filter</i> .	Valid
3	Cetak Berkas Kartu Ujian	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar kartu ujian untuk setiap mahasiswa sesuai <i>filter</i> .	Valid
4	Cetak Berkas Laporan Presensi Kehadiran Mahasiswa	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul laporan presensi mahasiswa.	Valid
5	Cetak Berkas Daftar Mahasiswa dengan IPK.	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar mahasiswa beserta IPK sesuai dengan <i>filter</i> .	Valid
6	Cetak Berkas Daftar Mahasiswa dengan IPS.	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar mahasiswa beserta IPS sesuai dengan <i>filter</i> .	Valid
7	Cetak Berkas Presensi Ujian.	Petugas	Isi semua <i>filter</i> yang tersedia untuk per kelas.	Muncul daftar presensi ujian untuk kelas tertentu sesuai <i>filter</i> .	Valid
8	Cetak Berkas Presensi Ujian.	Petugas	Isi semua <i>filter</i> yang tersedia untuk kelas gabungan.	Muncul daftar presensi ujian untuk kelas dan dosen tertentu sesuai <i>filter</i> .	Valid
9	Cetak Berkas Rekapitulasi Beban Mengajar Dosen.	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul daftar dosen sesuai <i>filter</i> beserta jumlah sks beban mengajar dan kelas yang diampu.	Valid
10	Cetak Berkas Persentase Kehadiran Dosen	Petugas	Isi semua <i>filter</i> yang tersedia.	Muncul laporan persentase kehadiran dosen sesuai <i>filter</i> .	Valid

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut.

- a. Penelitian ini berhasil mengembangkan RESTful API yang dapat memenuhi kebutuhan Sistem Informasi UIIPerkuliahan meliputi pencetakan berkas, pengelolaan nilai, dan pengelolaan presensi mahasiswa.
- b. RESTful API yang dibuat terbagi menjadi tiga *service* yaitu *service presence*, *service assessment*, dan *service simak* serta menggunakan dua macam basis data yaitu Mysql dan Oracle. Di mana *service presence* digunakan untuk mengolah presensi mahasiswa serta pencetakan berkas, *service assessment* digunakan untuk mengolah nilai mahasiswa, dan *service simak* untuk mengolah setiap kebutuhan sistem informasi UIIPerkuliahan meliputi pencetakan berkas, pengelolaan nilai, dan pengelolaan presensi mahasiswa yang berhubungan dengan basis data oracle.
- c. Berdasarkan hasil pengujian menggunakan aplikasi Postman dan metode *black-box testing*, RESTful API telah memenuhi rancangan pengujian dengan berbagai kasus uji.

#### 5.2 Saran

Pada penelitian ini, RESTful API yang dibuat masih sebatas memasukkan dan menampilkan nilai mahasiswa, presensi mahasiswa, dan pencetakan berkas. Masih ada beberapa layanan yang bisa ditambahkan lagi seperti mengkonversi nilai mahasiswa ketika terjadi perubahan kurikulum, mencetak transkrip nilai mahasiswa, dan menghitung Indeks Prestasi Kumulatif atau Indeks Prestasi Semester setiap semester yang didapatkan mahasiswa tersebut. Harapannya, hal-hal tersebut dapat dikembangkan lebih lanjut, sehingga tercipta sistem informasi yang memenuhi kebutuhan civitas UII.

## DAFTAR PUSTAKA

- Akbar, M. (2018). *Pengembangan RESTful API untuk Application Specific High Lvel Location Service*. Yogyakarta: Universitas Islam Indonesia.
- Bhaswara, F. A., Sarno, R., & Dwi Sunaryono, D. (2017). Perbandingan Kemampuan Database NoSQL dan SQL dalam Kasus ERP Retail. *JURNAL TEKNIK ITS Vol. 6, No. 2*, A510-A514.
- Digite. (2020, January 2). *What is Scrum?* Retrieved from digite.com: <https://www.digite.com/agile/scrum-methodology/>
- Feridi. (2020, January 3). *Mengenal RESTful Web Services*. Retrieved from codepolitan: <https://www.codepolitan.com/mengenal-restful-web-services>
- Kompasiana. (2019, December 30). *Apa Itu Scrum? Apa Saja Manfaatnya?* Retrieved from kompasiana.com: <https://www.kompasiana.com/advessbusiness/5807261545afbd38183aab3a/apa-itu-scrum-apa-saja-manfaatnya>
- Kurniawan, Y. K., Oslan, Y., & Kristanto, H. (2013). Implementasi REST - API untuk Portal Akaddemik UKDW Berbasis Android. *Jurnal EKSIS Vol 06 No 02*, 29-40.
- Lee, E. W. (2020, January 21). *Web Service Implementation Methodology*. Retrieved from oasis-open.org: [https://www.oasis-open.org/committees/download.php/13420/fwsi-im-1.0-guidlines-doc-wd-publicReviewDraft.htm#\\_Toc105485362](https://www.oasis-open.org/committees/download.php/13420/fwsi-im-1.0-guidlines-doc-wd-publicReviewDraft.htm#_Toc105485362)
- Nurdiyah, D., & Handayani, S. (2018, January 25). Restful Web Service Sistem Presensi Mahasiswa (Studi Kasus Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang). *Tranformatika*, pp. 60-66.
- Nurdiyanto, W. (2020, January 22). *Perbandingan SOAP dan REST sebagai Web Service*. Retrieved from PUSDIKLAT BPS: <http://pusdiklat.bps.go.id/index.php?r=artikel/view&id=206>
- Osd, D. (2020, February 7). *Pengertian SI (Sistem Informasi)*. Retrieved from Kompasiana: <https://www.kompasiana.com/dimasosd/55291077f17e6126268b48b6/pengertian-si-sistem-informasi>
- Peraturan Menteri Riset, Teknologi, dan Pendidikan Tinggi Nomor 44 Tahun 2015 Tentang Standar Nasional Pendidikan*. (2020, January 12). Retrieved from Kementerian Riset Teknologi dan Pendidikan Tinggi Republik Indonesia: <http://kopertis3.or.id/v2/wp-content/uploads/PERMENRISTEKDIKTI-NOMOR-44-TAHUN-2015-TENTANG-SNPT-SALINAN.pdf>



- Percona XtraDB Cluster. (2020, Maret 23). *About Percona XtraDB Cluster*. Retrieved from Percona: <https://www.percona.com/doc/percona-xtradb-cluster/LATEST/intro.html>
- Schwaber, K., & Sutherland, J. (2020, January 13). *Definiton of Scrum*. Retrieved from Scrum Guides: <https://www.scrumguides.org/scrum-guide.html#definition>
- Software Testing Fundamentals. (2020, 2 11). *Black Box Testing*. Retrieved from Software Testing Fundamentals: <http://softwaretestingfundamentals.com/black-box-testing/>
- Undang Undang Dasar 1945*. (2020, January 12). Retrieved from Dewan Perwakilan Rakyat Indonesia: <http://www.dpr.go.id/jdih/uu1945>