



## INSTITUT TEKNOLOGI DEL

### Pengembangan Fitur-Fitur *Campus Information System* Berbasis *Mobile* dengan Layanan *API*

## TUGAS AKHIR

11318016

Rafika Jesika Tampubolon

11318031

Martinus Yudha Chrisanto Sitinjak

11318065

Lucy Marito Fransisca Sihite

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO  
PROGRAM STUDI DIII TEKNOLOGI INFORMASI  
SITOLUAMA  
2021



## INSTITUT TEKNOLOGI DEL

### Pengembangan Fitur-Fitur *Campus Information System* Berbasis *Mobile* dengan Layanan API

#### TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma 3 Program  
Studi Teknologi Informasi

**11318016**

**Rafika Jesika Tampubolon**

**11318031**

**Martinus Yudha Chrisanto Sitinjak**

**11318065**

**Lucy Marito Fransisca Sihite**

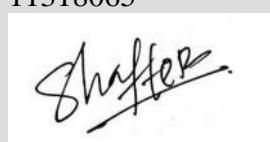
**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO  
PROGRAM STUDI DIII TEKNOLOGI INFORMASI  
SITOLUAMA  
2021**

## **HALAMAN PERNYATAAN ORISINALITAS**

Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

NAMA : Rafika Jesika Tampubolon  
NIM : 11318016  
TANDA TANGAN :   
TANGGAL : 19 Juli 2021

NAMA : Martinus Yudha Chrisanto Sitinjak  
NIM : 11318031  
TANDA TANGAN :   
TANGGAL : 19 Juli 2021

NAMA : Lucy Marito Fransiska Sihite  
NIM : 11318065  
TANDA TANGAN :   
TANGGAL : 19 Juli 2021

## **HALAMAN PENGESAHAN**

Tugas Akhir ini diajukan oleh : :

Nama : Rafika Jesika Tampubolon

NIM : 11318016

Program studi : D3 Teknologi Informasi

Nama : Martinus Yudha Chrisanto Sitinjak

NIM : 11318031

Program studi : D3 Teknologi Informasi

Nama : Lucy Marito Fransiska Sihite

NIM : 11318065

Program studi : D3 Teknologi Informasi

Judul Tugas Akhir : Pengembangan Fitur-Fitur *Campus Information*

*System Berbasis Mobile* dengan Layanan API

Telah berhasil dipertahankan dihadapannya dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Diploma 3 pada program studi D3 Teknologi Informasi Fakultas Informatika dan Teknik Elektro (FITE) Institut Teknologi Del.

## **DEWAN PENGUJI**

Pembimbing : Tegar Arifin Prasetyo, S.Si., M.Si (.tanda tangan)

Pembimbing : Regina Ayunita Tarigan, S.Si., M.Sc (.tanda tangan)

Penguji : Johannes Harungguan Sianipar, S.T., M.T (.tanda tangan)

Penguji : Tiurma Lumban Gaol, SP., M.P (.tanda tangan)

Ditetapkan : Laguboti

Tanggal : 19 Juli 2021

## KATA PENGANTAR

Puji dan syukur kehadirat Tuhan Yang Maha Esa atas penyertaannya penulis mampu menyelesaikan tugas akhir dengan judul “**Pengembangan Fitur-Fitur Campus Information System Berbasis Mobile dengan Layanan API**”. Tujuan penulisan dokumen ini adalah untuk memenuhi salah satu syarat dalam meraih gelar Ahli Madya Komputer pada program studi Diploma III Teknologi Informasi Fakultas Informatika dan Teknik Elektro Institut Teknologi Del. Dalam penulisan tugas akhir ini, mahasiswa banyak menghadapi hambatan dan kesulitan. Namun berkat usaha dan bantuan berbagai pihak baik secara langsung maupun tidak langsung maka penulisan tugas akhir ini dapat selesai. Penulis mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Allah Bapa di surga yang senantiasa mengiringi dalam setiap perjalanan hidup penulis selama pengerjaan tugas akhir ini, penulis diberikan kesehatan, kesabaran dan kekuatan untuk mampu mengerjakan tugas akhir ini dengan baik.
2. Ayah dan Ibu dari penulis yang selalu mendoakan dan menyampaikan semangat dan motivasi baik moril maupun material untuk mendukung dalam kelancaran penyelesaian tugas akhir ini.
3. Bapak Tegar Arifin Prasetyo, S.Si., M.Si dan Ibu Regina Ayunita Tarigan, S.Si., M.Sc, sebagai dosen pembimbing yang telah meluangkan waktu untuk memberikan anjuran dan kritik selama pengerjaan tugas akhir.
4. Bapak Johannes Harungguan Sianipar, S.T., M.T dan Ibu Tiurma Lumban Gaol, SP., M.P, sebagai dosen penguji dalam pelaksanaan tugas akhir yang telah menyampaikan masukan, kritik dan arahan yang bermanfaat untuk memperbaiki pengerjaan tugas akhir ini.
5. Seluruh teman-teman, terutama kepada teman-teman D3 Teknologi Informasi Institut Teknologi Del angkatan 2018 atas dukungan dan semangatnya yang mendorong tim penulis dalam menyelesaikan tugas akhir ini.
6. Seluruh pihak yang tidak dapat saya ucapkan satu per satu yang bersedia untuk berkontribusi selama ini untuk memberikan dukungan penulis mampu menyelesaikan tugas akhir.

Penulis memahami akan keterbatasan dari pengetahuan maupun pengalaman yang dimiliki, sehingga pembuatan Tugas Akhir ini masih jauh dari sempurna. Oleh sebab itu, penulis mengharapkan adanya saran dan masukan bahkan kritik yang membangun dari berbagai pihak. Hendaknya tugas akhir ini berguna dan dapat dimanfaatkan sebagai salah satu acuan ataupun pedoman bagi pembaca.

Sitoluama, 19 Juli 2021

11318016 Rafika Jesika Tampubolon

11318031 Martinus Yudha Chrisanto Sitinjak

11318065 Lucy Marito Fransisca Sihite

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Institut Teknologi Del, saya yang bertanda tangan di bawah ini:

Nama : Rafika Jesika Tampubolon  
NIM : 11318016  
Fakultas/Program Studi : FITE/ D3 Teknologi Informasi

Nama : Martinus Yudha Chrisanto Sitinjak  
NIM : 11318031  
Fakultas/Program Studi : FITE/ D3 Teknologi Informasi

Nama : Lucy Marito Fransiska Sihite  
NIM : 11318065  
Fakultas/Program Studi : FITE/ D3 Teknologi Informasi

Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan menyetujui untuk memberikan kepada Institut Teknologi Del **Hak Bebas Royalti Non Eksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul: **Pengembangan Fitur-Fitur Campus Information System Berbasis Mobile dengan Layanan API.**

Beserta perangkat yang ada (jika di perlukan). Dengan Hak Bebas Royalty *Non-eksklusif* ini Institut Teknologi Del berhak menyimpan, mengalih/media-format dalam bentuk pangkalan data (database), merawat dan mempublikasikan tugas akhir saya selama tetap mencantunkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Laguboti  
Pada tanggal: 19 Juli 2021  
Yang menyatakan:



(Rafika Jesika Tampubolon) (Martinus Yudha Chrisanto Sitinjak) (Lucy Marito Fransiska Sihite)

## ABSTRAK

Nama : D3TI-13  
Program Studi : D3 Teknologi Informasi  
Judul : Pengembangan Fitur-Fitur *Campus Information System (CIS)*  
Berbasis *Mobile* dengan Layanan API.

*Campus Information System (CIS)* merupakan sebuah sistem informasi universitas yang berguna untuk mengumpulkan data grafik dan non-grafik terkait dengan universitas. Institut Teknologi Del (IT Del) adalah sebuah perguruan tinggi yang menyediakan CIS dalam bentuk *website* guna untuk menyediakan informasi-informasi kegiatan akademik, kemahasiswaan, keuangan dan keasramaan. Dalam penyampaian informasi kepada mahasiswa menggunakan CIS *website* masih belum mendukung sebagai sistem informasi yang dapat memenuhi akan informasi yang cepat, mudah dan murah dalam pengaksesannya. Tujuan dibangunnya aplikasi CIS berbasis *mobile* dengan layanan API adalah untuk memudahkan mahasiswa dalam mendapat informasi dan mendukung kegiatan akademis di IT Del dan akan lebih praktis, cepat dan efisien dalam pengaksesannya. *Mobile* adalah sekumpulan program yang berjalan pada aplikasi tersebut dimana akan digunakan pengguna. API adalah sekumpulan aturan yang ditentukan dalam menjelaskan bagaimana komputer atau aplikasi berkomunikasi satu sama lain. Metode yang digunakan dalam pengembangan fitur-fitur CIS *mobile* adalah metode *Scrum*. *Scrum* adalah kerangka pengembangan perangkat lunak dengan pengembangan secara berulang dan incremental untuk mengelola proyek perangkat lunak dan produk atau pengembangan aplikasi. Pengeraaan Tugas Akhir ini telah menghasilkan aplikasi CIS berbasis *mobile* dengan layanan API. Aplikasi CIS *mobile* yang terdiri dari fitur-fitur mahasiswa.

Kata kunci: CIS, *Mobile*, API dan *Scrum*

## **ABSTRACT**

Name : D3TI-13  
Study Program : D3 Information Technology  
Title : Development of Campus Information System (CIS) Features  
Mobile Based with API Service.

Campus Information System (CIS) is a university information system that is useful for collecting graphic and non-graphical data related to universities. Del Institute of Technology (IT Del) is a university that provides CIS in the form of a website in order to provide information on academic, student, financial and boarding activities. In delivering information to students using the CIS website, it is still not supported as an information system that can fulfill information that is fast, easy and cheap to access. The purpose of building a mobile-based CIS application with API services is to make it easier for students to get information and support academic activities at IT Del and it will be more practical, fast and efficient in accessing it. Mobile is a set of programs that run on the application where the user will use it. An API is a set of defined rules that describe how computers or applications communicate with each other. The method used in developing CIS mobile features is the Scrum method. Scrum is a software development framework with iterative and incremental development for managing software and product projects or application development. This final project has produced a mobile-based CIS application with API services. CIS mobile application consisting of student features.

Keywords: CIS, Mobile, API and Scrum

## **DAFTAR ISI**

KATA PENGANTAR	4
ABSTRAK	7
DAFTAR ISI	1
DAFTAR TABEL	4
DAFTAR GAMBAR	5
BAB I	PENDAHULUAN
1.1 Latar Belakang	8
1.2 Tujuan	12
1.3 Lingkup	12
1.4 Pendekatan	13
1.5 Sistematika Penyajian	15
BAB II TINJAUAN PUSTAKA	16
2.1 Mobile Application	16
2.2 Android	17
2.2.1 Arsitektur Android	17
2.2.2 Komponen Android	19
2.3. <i>Application Programming Interface (API)</i>	20
2.4 Framework Laravel	23
2.5 Kotlin	25
2.6 Android Studio	27
2.7 Trello	28
2.8 Campus Information System (CIS) IT Del	29
Kesimpulan	30
BAB III	ANALISIS
DAN DESAIN	32
3.1 Inisialisasi Proyek	32
3.2 Project Plan	32
3.3 Pengumpulan Data	32
3.3.1 Persiapan Survei	32
3.3.2 Pelaksanaan Survei	33
3.3.3 Hasil Survei	33
3.4 Kebutuhan Fungsional	34
3.5 Kebutuhan Non-Fungsional	36
3.6 User Characteristic	36
3.6.1 User Group 1	37
3.7 Arsitektur Sistem	37
3.8 Arsitektur Uniform Resource Identifier (URI).	38
3.9 Arsitektur Integrasi CIS Mobile dan Website	38

3.10 Current System	39
3.11 Target Sistem	49
3.11.1 Business Process Otentikasi	49
3.11.2 Business Process Melihat Notifikasi	51
3.11.3 Business Process Mengelola Izin Bermalam (IB)	51
3.11.4 Business Process Mengelola Izin Keluar (IK)	52
3.11.5 Business Process Mengelola Perkuliahuan	53
3.11.6 Business Process Mengelola Data Diri	54
3.11.7 Business Process Mengelola Kuesioner	56
3.11.8 Business Process Melihat Pengumuman	57
3.11.9 Business Process Mengelola Kartu Rencana Studi (KRS)	58
3.11.10 Business Process Melihat Paket	59
3.11.11 Business Process Melihat Jadwal Perkuliahuan	60
3.12 Use Case Diagram	61
3.13 Use Case Scenario	62
3.14 ER-Diagram	63
3.15 Conceptual Data Model	64
3.16 Physical Data Model	65
3.17 Class Diagram	66
3.18 Desain User Interface	67
3.18.1 Desain User Interface - <i>Login</i>	67
3.18.2 Desain User Interface – Melihat Notifikasi	68
3.18.3 Desain User Interface – Mengelola Request Izin Bermalam	68
3.18.4 Desain User Interface – Mengelola Request Izin Keluar	70
3.18.5 Desain User Interface – Mengelola Perkuliahuan	72
3.18.6 Desain User Interface - Mengelola Data Diri	73
3.18.7 Desain User Interface - Mengelola Kuesioner	74
3.18.8 Desain User Interface - Melihat Pengumuman	75
3.18.9 Desain User Interface - Mengelola Kartu Rencana Studi (KRS)	76
3.18.10 Desain User Interface - Melihat Paket	77
3.18.11 Desain User Interface - Melihat Grafik IP	78
3.18.12 Desain User Interface - Melihat Jadwal	78
<b>BAB IV</b>	<b>IMPLEMENTASI</b>
4.1 Kebutuhan Implementasi	80
4.1.1 Perangkat Keras ( <i>Hardware</i> )	80
4.1.2 Perangkat Lunak ( <i>Software</i> )	81
4.2 Batasan Implementasi	81
4.3 Implementasi Aplikasi	82
<b>BAB V</b>	<b>HASIL DAN</b>
<b>PENGUJIAN</b>	<b>83</b>
5.1 Hasil Aplikasi	83
5.1.1 Login	83

5.1.2 Home	83
5.1.3 Mengelola Request Izin Bermalam (IB)	84
5.1.4 Mengelola Request Izin Keluar (IK)	86
5.1.5 Mengelola Data Diri	87
5.1.6 Mengelola Kuesioner	89
5.1.7 Melihat Pengumuman	90
5.1.8 Melihat Paket	91
5.1.9 Melihat Notifikasi	91
5.2 Pengujian	92
5.2.1 Pengujian Unit	92
5.2.1.1 Skenario Pengujian	93
5.2.1.2 Pengujian <i>Beta Testing</i> pada Aplikasi	126
5.2.2 Pengujian <i>Application Programming Interface</i> (API)	130
<b>BAB VI</b>	<b>KESIMPULAN DAN SARAN</b>
6.1 Kesimpulan	153
6.2 Saran	153
Rujukan	154
Lampiran 1 - Hasil Kuesioner	156
Lampiran 2 – Arsitektur <i>Uniform Resource Identifier</i> (URI)	157
Lampiran 3 - Use Case Scenario	169
Lampiran 4 – Implementasi Aplikasi	193
Lampiran 5 - Hasil UAT	205

## DAFTAR TABEL

Tabel 1. Komponen Android .....	19
Tabel 2. Kebutuhan Non-Fungsional .....	36
Tabel 3. User Group 1 .....	37
Tabel 4. Daftar Tabel Master dan Tabel <i>Transactional</i> .....	64
Tabel 5. Spesifikasi <i>Hardware Laptop</i> 1 .....	80
Tabel 6. Spesifikasi <i>Hardware Laptop</i> 2 .....	80
Tabel 7. Spesifikasi <i>Hardware Laptop</i> 3 .....	80
Tabel 8. <i>Hardware Smartphone</i> .....	81
Tabel 9. Spesifikasi <i>software</i> .....	81
Tabel 10. Pengujian Unit .....	92
Tabel 11. Skenario Pengujian Otentikasi .....	94
Tabel 12. Skenario Pengujian <i>Add Request Izin Bermalam</i> .....	95
Tabel 13. Skenario Pengujian <i>Edit Request Izin Bermalam</i> .....	98
Tabel 14. Skenario Pengujian <i>Cancel Request Izin Bermalam</i> .....	100
Tabel 15. Skenario Pengujian <i>Add Request Izin Keluar</i> .....	101
Tabel 16. Skenario Pengujian <i>Edit Request Izin Keluar</i> .....	103
Tabel 17. Skenario Pengujian <i>Cancel Request Izin Keluar</i> .....	106
Tabel 18. Skenario Pengujian Melihat Matakuliah.....	107
Tabel 19. Skenario Pengujian Unduh Modul.....	108
Tabel 20. Skenario Pengujian Melihat Jadwal.....	110
Tabel 21. Skenario Pengujian Melihat Data Diri.....	110
Tabel 22. Skenario Pengujian <i>Edit Data Diri</i> .....	113
Tabel 23. Skenario Pengujian Melihat Kuesioner .....	117
Tabel 24. Skenario Pengujian Mengisi Kuesioner.....	118
Tabel 25. Skenario Pengujian Melihat Pengumuman .....	120
Tabel 26. Skenario Pengujian Melihat Paket .....	121
Tabel 27. Skenario Pengujian <i>Add Request KRS</i> .....	122
Tabel 28. Skenario Pengujian <i>Edit Request KRS</i> .....	123
Tabel 29. Skenario Pengujian Melihat Notifikasi .....	125
Tabel 30. Arsitektur URI .....	157
Tabel 31. Use Case Scenario Login .....	169
Tabel 32. Use Case Scenario Logout .....	170
Tabel 33. Use Case Scenario Melihat Notifikasi .....	170
Tabel 34. Use Case Scenario <i>Add Request IB</i> .....	171
Tabel 35. Use Case Scenario <i>Edit Request IB</i> .....	172
Tabel 36. Use Case Scenario <i>Cancel Request IB</i> .....	173
Tabel 37. Use Case Scenario <i>Add Request IK</i> .....	174
Tabel 38. Use Case Scenario <i>Edit Request IK</i> .....	175
Tabel 39. Use Case Scenario <i>Cancel Request IK</i> .....	177
Tabel 40. Use Case Scenario Melihat Matakuliah.....	177
Tabel 41. Use Case Scenario Unduh Modul .....	178
Tabel 42. Use Case Scenario Melihat Jadwal Perkuliahan .....	181
Tabel 43. Use Case Scenario Melihat Data Diri .....	182
Tabel 44. Use Case Scenario <i>Edit Data Diri</i> .....	183
Tabel 45. Use Case Scenario Melihat Kuesioner.....	184
Tabel 46. Use Case Scenario Mengisi Kuesioner .....	185
Tabel 47. Use Case Scenario Melihat Pengumuman .....	186
Tabel 48. Use Case Scenario Melihat Paket .....	188
Tabel 49. Use Case Scenario Mengisi KRS.....	189
Tabel 50. Use Case Scenario <i>Edit KRS</i> .....	190
Tabel 51. Use Case Scenario Melihat Grafik IP .....	191

## DAFTAR GAMBAR

Gambar 1. Arsitektur Android .....	18
Gambar 2. Popularitas pengguna <i>framework Laravel</i> .....	24
Gambar 3. Membuat <i>Class</i> dengan Java .....	26
Gambar 4. Membuat <i>Class</i> dengan Kotlin .....	26
Gambar 5. Membuat <i>Extension Function</i> .....	27
Gambar 6. Tampilan Android Studio .....	28
Gambar 7. Arsitektur Sistem .....	37
Gambar 8. Integrasi CIS <i>mobile</i> dan CIS <i>website</i> .....	38
Gambar 9. BPMN <i>Current System</i> Melakukan <i>Login</i> .....	39
Gambar 10. BPMN <i>Current System</i> Melakukan <i>Logout</i> .....	40
Gambar 11. BPMN <i>Current System</i> Melihat Notifikasi .....	40
Gambar 12. BPMN Mengelola Izin Bermalam (IB) .....	41
Gambar 13. BPMN <i>Current System</i> Mengelola <i>Request IK</i> .....	42
Gambar 14. BPMN <i>Current System</i> Mengelola Perkuliahan .....	43
Gambar 15. BPMN <i>Current System</i> Mengelola Data Diri .....	44
Gambar 16. BPMN <i>Current system</i> Mengelola Kuesioner .....	46
Gambar 17. BPMN <i>Current System</i> Mengelola Pengumuman .....	46
Gambar 18. BPMN <i>Currents System</i> Mengelola Request KRS .....	47
Gambar 19. BPMN <i>Current System</i> Melihat Informasi Paket .....	47
Gambar 20. BPMN <i>Current System</i> Melihat Grafik IP .....	48
Gambar 21. BPMN <i>Current System</i> Melihat Jadwal Perkuliahan .....	49
Gambar 22. Target Sistem Melakukan <i>Login</i> .....	50
Gambar 23. Target Sistem Melakukan <i>Logout</i> .....	50
Gambar 24. Target Sistem Melihat Notifikasi .....	51
Gambar 25. Target Sistem Mengelola IB .....	52
Gambar 26. Target Sistem Mengelola IK .....	53
Gambar 27. Target Sistem Mengelola Perkuliahan .....	54
Gambar 28. Target Sistem Mengelola Data Diri .....	56
Gambar 29. Target Sistem Mengelola Kuesioner .....	57
Gambar 30. Target Sistem Melihat Pengumuman .....	58
Gambar 31. Target Sistem Mengelola KRS .....	59
Gambar 32. Target Sistem Melihat Paket .....	60
Gambar 33. Target Sistem Melihat Jadwal Perkuliahan .....	61
Gambar 34. <i>Use Case Diagram</i> .....	62
Gambar 35. ER-Diagram .....	63
Gambar 36. <i>Conceptual Data Model</i> .....	65
Gambar 37. <i>Physical Data Model</i> .....	66
Gambar 38. <i>Class Diagram</i> .....	67
Gambar 39. Desain <i>User Interface - Login</i> .....	68
Gambar 40. Desain <i>User Interface - Notifikasi</i> .....	68
Gambar 41. Desain <i>User Interface - Request Izin Bermalam</i> .....	69
Gambar 42. Desain <i>User Interface - Edit Request Izin Bermalam</i> .....	69
Gambar 43. Desain <i>User Interface - Data Izin Bermalam</i> .....	70
Gambar 44. Desain <i>User Interface - Request Izin Keluar</i> .....	71
Gambar 45. Desain <i>User Interface - Edit Request Izin Keluar</i> .....	71
Gambar 46. Desain <i>User Interface - Data Izin Keluar</i> .....	72
Gambar 47. Desain <i>User Interface - Melihat Mata Kuliah</i> .....	72
Gambar 48. Desain <i>User Interface - Unduh Modul</i> .....	73
Gambar 49. Desain <i>User Interface - Melihat Data Diri</i> .....	73
Gambar 50. Desain <i>User Interface - Edit Data Diri</i> .....	74
Gambar 51. Desain <i>User Interface - Melihat Kuesioner</i> .....	75
Gambar 52. Desain <i>User Interface - Mengisi Kuesioner</i> .....	75
Gambar 53. Desain <i>User Interface - Melihat Pengumuman</i> .....	76
Gambar 54. Desain <i>User Interface - Mengisi KRS</i> .....	76
Gambar 55. Desain <i>User Interface - Edit KRS</i> .....	77
Gambar 56. Desain <i>User Interface - Melihat Paket</i> .....	77
Gambar 57. Desain <i>User Interface - Melihat Grafik IP</i> .....	78

Gambar 58. Desain <i>User Interface</i> – Melihat Jadwal .....	79
Gambar 59. Implementasi <i>Login</i> .....	83
Gambar 60. Implementasi <i>Home</i> .....	84
Gambar 61. Implementasi <i>Add Request IB</i> .....	85
Gambar 62. Implementasi <i>Edit Request IB</i> .....	85
Gambar 63. Implementasi <i>Add Request IK</i> .....	86
Gambar 64. Implementasi <i>Edit Request IBK</i> .....	87
Gambar 65. Implementasi Melihat Data Diri .....	88
Gambar 66. Implementasi Edit Data Diri .....	88
Gambar 67. Implementasi Melihat Kuesioner .....	89
Gambar 68. Implementasi Mengisi Kuesioner .....	90
Gambar 69. Implementasi Melihat Pengumuman .....	90
Gambar 70. Implementasi Melihat Pengumuman .....	91
Gambar 71. Implementasi Melihat Notifikasi .....	92
Gambar 72. Diagram Pertanyaan 1 .....	126
Gambar 73. Diagram Pertanyaan 2 .....	127
Gambar 74. Diagram Pertanyaan 3 .....	127
Gambar 75. Diagram Pertanyaan 4 .....	128
Gambar 76. Diagram Pertanyaan 5 .....	128
Gambar 77. Diagram Pertanyaan 6 .....	129
Gambar 78. Diagram Pertanyaan 7 .....	129
Gambar 79. Pengujian API login .....	130
Gambar 80. Pengujian API logout .....	131
Gambar 81. Pengujian API menampilkan data <i>user</i> .....	132
Gambar 82. Pengujian API request IB .....	132
Gambar 83. Pengujian API edit request IB .....	133
Gambar 84. Pengujian API cancel request IB .....	134
Gambar 85. Pengujian API menampilkan daftar IB .....	134
Gambar 86. Pengujian API detail IB .....	135
Gambar 87. Pengujian API approve IB .....	136
Gambar 88. Pengujian API request IK .....	136
Gambar 89. Pengujian API edit request IK .....	137
Gambar 90. Pengujian API Cancel Request IK .....	138
Gambar 91. Pengujian API Menampilkan Daftar-Daftar IK .....	139
Gambar 92. Pengujian API Detail IK .....	140
Gambar 93. Pengujian API Data Pribadi .....	141
Gambar 94. Pengujian API Edit Data Pribadi .....	141
Gambar 95. Pengujian API Data Orangtua .....	142
Gambar 96. Pengujian API Edit Data Orangtua .....	143
Gambar 97. Pengujian API Data Asal Sekolah .....	143
Gambar 98. Pengujian API Edit Data Asal Sekolah .....	144
Gambar 99. Pengujian API Data Akademis .....	145
Gambar 100. Pengujian API Data Asrama .....	145
Gambar 101. Pengujian API Pengumuman .....	146
Gambar 102. Pengujian API Detail Pengumuman .....	147
Gambar 103. Pengujian API Paket .....	147
Gambar 104. Pengujian API Detail Paket .....	148
Gambar 105. Pengujian API Kuesioner .....	149
Gambar 106. Pengujian API Kuesioner Wajib .....	150
Gambar 107. Pengujian API Pertanyaan Kuesioner .....	151
Gambar 108. Pengujian API Isi Kuesioner .....	152
Gambar 109. Grafik Hasil Kuesioner 1 .....	156
Gambar 110. Grafik Hasil Kuesioner .....	156
Gambar 111. Pembuatan Database MySQL .....	194
Gambar 112. API Model .....	195
Gambar 113. API Controller .....	195
Gambar 114. API Routes .....	196
Gambar 115. Pembuatan Project pada Android Studio .....	196
Gambar 116. Pembuatan Activity .....	197

Gambar 117. Pembuatan Model Pada Android Studio .....	198
Gambar 118. Potongan Kode Model Response .....	199
Gambar 119. Pembuatan Adapter.....	200
Gambar 120. Potongan Kode Adapter.....	200
Gambar 121. Pembuatan Fragment .....	201
Gambar 122. Potongan Kode Fragment .....	202
Gambar 123. Potongan Kode Layout <i>RecyclerView</i> .....	203
Gambar 124. Potongan Kode <i>Layout</i> .....	204
Gambar 125. Hasil UAT .....	205

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Kampus adalah nama yang digunakan untuk menunjukkan suatu area yang terdiri dari beberapa gedung tempat berlangsungnya suatu kegiatan perguruan tinggi [1]. Menurut O'Brien pada tahun 2003, Sistem Informasi adalah gabungan dari beberapa sumber daya yang diorganisir dari manusia, *hardware*, *software* dan beberapa data yang mendistribusikan suatu informasi pada sebuah organisasi [2]. *Campus Information System* (CIS) merupakan sebuah sistem informasi universitas yang berguna untuk mengumpulkan data grafik dan non-grafik terkait dengan universitas [3]. Data yang berada pada CIS akan disimpan pada database dan akan ditampilkan kepada pengguna dengan berbagai format seperti *website*, *mobile application*, dan lain-lain. Adapun yang menjadi tujuan dari CIS adalah meningkatkan efisiensi dan kemajuan universitas serta menyediakan layanan yang mempunyai batasan antara administrasi unit layanan pusat dan organisasi lainnya yang ada pada universitas tersebut [3] [4]. Oleh sebab, itu sebuah lembaga pendidikan harus menyediakan layanan CIS ini guna untuk meningkatkan efisiensi lembaga pendidikan tersebut.

Institut Teknologi Del (IT Del) adalah sebuah perguruan tinggi yang menyediakan CIS dalam bentuk *website* guna untuk menyediakan informasi-informasi kegiatan akademik, kemahasiswaan, keuangan dan keasramaan. Dalam penyampaian informasi kepada mahasiswa melalui *website* yang saat ini digunakan sudah menjawab sebagian penyajian informasi yang lengkap. Namun masih belum mendukung sebagai sistem informasi yang dapat memenuhi akan informasi yang cepat, mudah dan murah dalam pengaksesannya bagi seluruh mahasiswa. Hal ini diperoleh dari hasil kuesioner dengan mahasiswa IT Del yang memiliki 160 responden. Berdasarkan hasil kuesioner terdapat beberapa mahasiswa yang mengalami kesulitan dalam penggunaan CIS berbasis *website* saat ini. Hasil kuesioner juga menunjukkan bahwa mahasiswa IT Del setuju dengan pengembangan CIS berbasis *mobile* ini dengan alasan akan mempermudah mahasiswa dalam mendapatkan informasi dari CIS. Adapun hasil kuesioner yang telah dilakukan dilampirkan pada Lampiran 1.

Menurut survei yang dilakukan oleh Comscore di Amerika Serikat, jumlah pengguna aplikasi berbasis *mobile* telah meningkat sebesar 90% selama tiga tahun terakhir (2015-2018) [5]. Hal ini terjadi dikarenakan aplikasi berbasis *mobile* sudah lebih interaktif

dibandingkan dengan *website* atau media lainnya. Hal ini juga dikarenakan kualitas komputasi dari smartphone yang sebanding dengan komputer meningkatkan pengguna akses informasi secara *mobile* bagi mahasiswa di dalam institusi akademik [6]. Saat ini IT Del belum menyediakan CIS berbasis *mobile* baik *role* mahasiswa, staf maupun dosen. Dalam hal pengembangan fitur-fitur CIS berbasis *mobile* ini akan difokuskan dalam pengembangan fitur-fitur CIS untuk *role* mahasiswa IT Del. Pengembangan fitur-fitur CIS berbasis *mobile* dengan layanan *Applications Programming Interface* (API) dan menggunakan *framework* Laravel terdapat beberapa penelitian terkait. Penelitian ini membahas tentang Sistem Informasi Akademik bagi Layanan Mahasiswa Universitas Muhammadiyah Surakarta (UMS) dengan berbasis *Mobile*.

Pada Layanan Mahasiswa UMS, penerapan terhadap penggunaan teknologi *mobile* dapat dilihat dengan adanya pembuatan aplikasi android. Aplikasi android yang akan mendukung untuk *broadcast* pengumuman, dimana pada penerapannya teknik yang digunakan yaitu dengan teknik notifikasi yang akan memberikan banyak kemudahan bagi mahasiswanya dalam mendapatkan dan menyampaikan informasi dalam bentuk pengumuman. Hal ini dapat dilihat dari penilaian calon *user* yang menunjukkan 92% menyatakan bahwa dalam penggunaannya aplikasi *mobile* ini mudah untuk digunakan, dapat meningkatkan kenyamanan dan kecepatan akses dalam penggunaan serta bermanfaat bagi mahasiswa UMS dalam menjalankan studi akademik mereka [6]. Terdapat juga penelitian lain mengenai perbandingan antara *mobile applications* dengan *mobile website*. Berdasarkan data hasil penelitian ini menunjukkan bahwa dengan menyampaikan suatu layanan melalui suatu aplikasi *mobile* akan lebih banyak yang tertarik dan berminat dalam menggunakan dibandingkan dengan menyampaikannya melalui web *mobile* sehingga hal ini membuat aplikasi *mobile* menjadi suatu alat utama yang tepat untuk digunakan di perguruan tinggi di Taiwan, dan juga dalam aktivitas yang bersifat rutinitas dan kebiasaan maka layanan aplikasi *mobile* lebih umum dan sering digunakan dibandingkan dengan *website mobile* [7].

Merujuk pada uraian penelitian yang telah dilakukan mengenai aplikasi berbasis *mobile*, perlu dilakukan optimalisasi CIS ke dalam teknologi *mobile*. CIS berbasis *mobile* ini akan dapat memudahkan mahasiswa dalam mendukung kegiatan akademis di IT Del dan akan lebih praktis, cepat dan efisien dalam pengaksesan data atau informasi dibandingkan dengan CIS berbasis *website*. Aplikasi *mobile* merupakan suatu perangkat

lunak yang akan berjalan dan terpasang pada perangkat *mobile* seperti *smartphone* atau *tablet* yang biasanya akan diinstal atau dipasang oleh *user* untuk mendukung dan membantu aktivitas dan pekerjaannya [8]. Salah satu fitur unggul yang membuat aplikasi berbasis *mobile* menjadi lebih interaktif adalah *push notification*. Fitur *push notification* memungkinkan sistem untuk dapat memberikan notifikasi kepada mahasiswa walaupun sedang tidak mengakses CIS berbasis *mobile* tersebut sehingga mahasiswa tidak akan ketinggalan informasi [8].

Pada umumnya sistem operasi *mobile* yang sering digunakan adalah *mobile android*. Android adalah suatu *platform* yang lengkap dengan pengembangan alat, vendor *industry handphone* serta dukungan dari sistem komunikasi terbuka [9]. Menurut Kusumastuti, sistem operasi android akan menjadi salah satu pilihan yang baik dan tepat khususnya bagi *vendor smartphone* yang akan mengembangkan atau membuat suatu *mobile smartphone* dikarenakan android memiliki biaya lisensi lebih mudah dan bersifat *open source* sehingga dapat mempermudah vendor dalam mengembangkan *mobile smartphone* perusahaan [10]. Dalam pengembangan fitur-fitur CIS berbasis *mobile* ini digunakan dengan layanan API menggunakan *framework Laravel*.

API adalah suatu *class* yang dirancang dan dibuat dengan tujuan untuk menghubungkan dan mengkomunikasikan suatu aplikasi *mobile* dengan *database* yang digunakan [11]. Penggunaan layanan API ini akan dapat membantu dan mempermudah tugas *programmer* dalam membongkar dan mengkomunikasikan suatu *software* yang akan dibangun atau dikembangkan serta diintegrasikan pada 2 bagian aplikasi. API memiliki beberapa elemen pendukung seperti *function*, *protocols*, dan *tools* yang mengizinkan *developer* dalam mengembangkan atau membangun aplikasi. Keuntungan dalam menggunakan layanan ini API dapat berjalan di *Apache* maupun *website* server lainnya, mendukung fungsi *Create*, *Read*, *Update* dan *Delete* (CRUD) yang bekerja lewat *HTTP protocol*, mendukung fitur *Model View Controller* (MVC) seperti *routing*, *controller*, *action results*, *model* dan lain-lain, dan mempermudah *developer* dalam mengakses data dikarenakan *response* dengan format *JSON*, *XML* ataupun format lainnya [12]. Laravel adalah *framework PHP* yang bersifat *open-source* dibuat oleh Taylor Otwell (2017) dengan tujuan untuk pengembangan aplikasi dengan memakai konsep MVC untuk mempermudah pengembangan *website service*. *Framework Laravel* mudah dipahami serta menyediakan *Routing*, *Restful Controller*, dan komponen yang paling umum digunakan. *Routing* dalam *Laravel API*

berfungsi untuk membuat jalan/route lebih mudah digunakan dan *RestFull Controller* berfungsi untuk memisahkan logika dalam melayani *GET*, *POST*, *DELETE* dan lain sebagainya [13].

Pada aplikasi CIS IT Del berbasis *website* saat ini memiliki fitur - fitur sebagai berikut:

1. Fitur Otentikasi
2. Fitur Data Diri
3. Fitur Perkuliahuan
4. Fitur Kartu Rencana Studi
5. Fitur Bursar
6. Fitur Artikel
7. Fitur Publikasi
8. Fitur *Request* Izin Bermalam
9. Fitur *Request* Izin Keluar
10. Fitur *Request* Izin Kolaboratif
11. Fitur *View* Poin Pelanggaran
12. Fitur Penggantian KTM
13. Fitur Surat Mahasiswa Aktif
14. Fitur Surat Pengantar Proyek
15. Fitur Surat Pengantar Magang
16. Fitur Surat Pengantar Kompetisi
17. Fitur Pengumuman dan Paket
18. Fitur *Request* Peminjaman Alat Laboratorium dan Pemesanan Bahan
19. Fitur *View Polling*, Kuisioner

Pada aplikasi CIS IT Del berbasis *mobile* yang akan dikembangkan untuk *role* mahasiswa tersedia fitur-fitur sebagai berikut:

1. Fitur Notifikasi: fitur ini merupakan fungsi yang akan memberikan notifikasi di *smartphone* mahasiswa apabila ada unggah modul baru, *accept* izin bermalam (IB), *accept* izin keluar (IK) dan informasi *accept request* kartu rencana studi (KRS).
2. Fitur *Request* Izin Bermalam (IB): fitur ini berfungsi untuk melakukan *request* dan *accept* IB oleh mahasiswa ke pihak keasramaan.
3. Fitur *Request* Izin Keluar (IK): fitur ini berfungsi untuk melakukan *request* dan

*accept* IK oleh mahasiswa ke pihak dosen wali, keasramaan dan BAAK.

4. Fitur Perkuliahan: fitur ini berfungsi untuk menampilkan daftar-daftar matakuliah, jadwal perkuliahan dan unduh modul.
5. Fitur Data Diri: fitur ini memuat data diri mahasiswa mulai dari data akademis, hasil studi, data pribadi, data orangtua, data asal sekolah dan nilai perilaku keasramaan.
6. Fitur Kuesioner: fitur ini memuat kuesioner-kuesioner yang diberikan kepada mahasiswa.
7. Fitur Kartu Rencana Studi (KRS): fitur ini berfungsi untuk melakukan pengisian rencana studi selama satu semester.
8. Fitur Pengumuman: fitur ini memuat fitur pengumuman untuk menampilkan pengumuman yang diberikan pihak kampus dan dosen
9. Fitur Paket: fitur ini memuat list data paket yang diterima oleh mahasiswa melalui perantara pos satpam
10. Fitur Otentikasi: fitur ini berfungsi untuk melakukan *login* dan *logout* manual.

Berdasarkan latar belakang yang telah dijelaskan sebelumnya dapat disimpulkan bahwa kami melakukan penelitian tugas akhir yang akan mengembangkan fitur-fitur CIS berbasis *mobile* untuk mahasiswa dengan layanan API yang dapat menyediakan informasi secara cepat, efisien dan mudah diakses. Hasil pengembangan akan diintegrasikan antar *class* menjadi sebuah API dengan CIS berbasis *mobile*. API akan menangkap data yang berhubungan dengan informasi akademik ke dalam CIS *mobile*. CIS berbasis *mobile* untuk mahasiswa yang dikombinasikan dengan fitur *push notification* diharapkan dapat memberi manfaat yang besar dalam suatu sistem informasi, tidak terkecuali untuk sistem informasi di dalam akademik maupun non-akademik.

## 1.2 Tujuan

Pengerjaan Tugas Akhir ini memiliki tujuan sebagai berikut:

1. Mengembangkan fitur-fitur *Campus Information System* berbasis *mobile*.
2. Mengembangkan layanan API untuk mendukung aplikasi *Campus Information System* berbasis *mobile*.

## 1.3 Lingkup

Batasan/lingkup dalam melakukan penelitian ini adalah:

1. Pengembangan aplikasi CIS berbasis *mobile* khusus untuk *role* mahasiswa.
2. Aplikasi akan dikembangkan berdasarkan *business process* yang sudah ada saat ini yaitu *business process* CIS berbasis *website*.
3. Menggunakan *Android platform* yang menggunakan bahasa pemrograman Kotlin.
4. Menggunakan layanan API dengan *framework* Laravel.
5. Di dalam aplikasi akan terdapat notifikasi yang memberikan informasi unggah modul baru oleh dosen.
6. Aplikasi juga akan memberikan notifikasi persetujuan IB, IK, informasi paket mahasiswa, KRS dan pengumuman.
7. Di dalam aplikasi mahasiswa dapat melakukan *login* dan *logout* secara manual. Dengan adanya *logout* secara manual mahasiswa tidak lagi melakukan *login* setiap kali mengakses CIS.

#### **1.4 Pendekatan**

Metode penelitian yang digunakan untuk penggerjaan Tugas Akhir ini adalah metode *Systems development life cycle* (SDLC). SDLC adalah siklus yang digunakan dalam pembuatan atau pengembangan sistem informasi yang bertujuan untuk menyelesaikan masalah secara efektif [14]. Pendekatan ini memiliki beberapa tahapan yang dapat dilakukan sehingga akhirnya menghasilkan sebuah aplikasi yang dapat menjawab permasalahan. Adapun langkah-langkah yang dilakukan untuk Tugas Akhir ini adalah:

##### **1. Observasi**

Pada tahap observasi yang akan dilakukan peneliti adalah:

- a. Melakukan wawancara dengan pihak Sumber Daya Informasi (SDI) IT Del.
- b. Membuat pertanyaan dari kuesioner yang diajukan kepada mahasiswa IT Del.
- c. Mendapatkan hasil wawancara dari pihak Sumber Daya Informasi (SDI) IT Del dan hasil kuesioner dari mahasiswa.
- d. Mengamati sistem yang berjalan saat ini untuk mengetahui bagaimana mahasiswa menggunakan CIS yang ada saat ini.

- e. Mengetahui fitur-fitur apa saja yang paling utama dibutuhkan oleh mahasiswa.

## 2. Studi Literatur

Pada tahap ini studi literatur yang akan dilakukan peneliti adalah:

- a. Mencari bahan pustaka dari berbagai sumber tertulis, baik berupa buku, artikel atau jurnal yang dapat digunakan untuk mendukung pemecahan masalah dan topik yang sudah ditetapkan oleh peneliti.
- b. Mendalami materi sesuai dengan pengembangan sistem informasi kampus, *mobile*, layanan API dan *framework* Laravel.

## 3. Analisis

Pada tahap ini analisis yang akan dilakukan peneliti adalah:

- a. Melakukan analisis terhadap sistem yang berjalan saat ini.
- b. Melakukan analisis kebutuhan pada sistem dan batasan sistem yang difokuskan kepada mahasiswa.
- c. Melakukan analisis terhadap database yang akan digunakan dalam pengembangan fitur-fitur CIS berbasis *mobile* menggunakan layanan API.
- d. Melakukan analisis terhadap hasil yang didapat dari studi literatur dan analisis terhadap penerapan layanan API dalam pengembangan fitur-fitur CIS.

## 4. Desain

Pada tahap ini desain yang akan dilakukan peneliti adalah:

- a. Merancang *user interface* sistem yang akan dibangun yaitu berupa *prototype* tampilan-tampilan dari setiap halaman CIS yang akan dibangun. Hasil dari proses ini akan menghasilkan desain *prototype* dari CIS berbasis *mobile*.

## 5. Implementasi

Pada tahap ini implementasi yang akan dilakukan peneliti adalah:

- a. Peneliti akan mulai membangun aplikasi. Aplikasi yang akan dibangun adalah aplikasi yang sudah menerapkan hasil dari analisis dan desain *prototype* yang sudah dirancang oleh peneliti. *Development team* akan mengembangkan aplikasi hingga mencapai target yang sudah ditentukan sebelumnya.

## 6. Pengujian

Pada tahap ini pengujian yang akan dilakukan peneliti adalah:

- a. Memastikan bahwa aplikasi yang sudah dibangun layak digunakan, bebas dari *error*, telah sesuai dengan hasil perancangan, dan menjawab kebutuhan pengguna.

## 1.5 Sistematika Penyajian

Secara garis besar Laporan Tugas Akhir ini terdiri dari 6 bab. Sistematika dalam penelitian dokumen ini adalah sebagai berikut:

1. Bab I merupakan Pendahuluan berisi informasi mengenai latar belakang, tujuan, lingkup, pendekatan dan sistematika penyajian penelitian ini.
2. Bab II dijelaskan mengenai tinjauan pustaka yang berkaitan dengan penelitian ini.
3. Bab III dijelaskan mengenai analisis dan desain yang membahas mengenai pengamatan lapangan, penelitian terkait serta kesimpulan mengenai antarmuka yang akan dibangun.
4. Bab IV dijelaskan mengenai implementasi antarmuka yang berisi pembahasan mengenai langkah-langkah implementasi antarmuka yang sudah dibangun untuk mahasiswa IT Del.
5. Bab V dijelaskan mengenai hasil dan pengujian yang telah dilakukan terhadap hasil implementasi antarmuka yang telah selesai dibangun.
6. Bab VI dijelaskan mengenai kesimpulan dan saran yang berkaitan dengan penelitian ini.

## **BAB II**

### **TINJAUAN PUSTAKA**

Tinjauan Pustaka menjelaskan tentang teori, metode, teknik, proses dan prosedur yang terkait dengan tema penelitian, dan kajian mengenai hasil penelitian sebelumnya yang berkaitan dengan penelitian.

#### **2.1 Mobile Application**

Pada era *modern* ini, orang banyak menggunakan *mobile application*. *Mobile application* adalah sekumpulan program yang berjalan pada aplikasi tersebut dimana akan digunakan pengguna. *Mobile application* juga banyak digunakan banyak orang seperti *browser*, *chatting*, menelepon dan masih banyak lagi [15]. *Mobile Application* adalah teknologi tercanggih yang saat ini diminati di seluruh dunia. *Mobile Application* dapat kompleks atau sederhana, tergantung dengan desain dan fitur yang ditawarkan. Berbagai fungsi dan informasi berkontribusi untuk menjadikan *mobile application* teratas. Dalam perjalanan misalnya, dalam hal komunikasi menyediakan beragam alat bantu seperti kamera dan sistem GPS dimana orang bisa menemukan jalan pulang dengan menghindari beban berat lalu lintas kota atau bahkan menemukan rute yang tidak mereka kenal [16].

Dalam Pengembangan aplikasi mobile terdapat tiga pendekatan yaitu:

##### **1. Mobile Website Application**

*Mobile website application* adalah aplikasi *website* yang diformat untuk *smartphone* atau tablet yang dapat digunakan atau berjalan pada semua *browser* dengan platform *mobile* sehingga aplikasi ini disebut multi platform. *Mobile website application* tidak dapat mengakses fitur-fitur perangkat keras yang terdapat pada *smartphone* karena *application website* hanya berjalan pada *browser smartphone*. *Website application* juga hanya dapat diakses jika terhubung dengan internet dan tergantung pada konektivitas yang menyebabkan performanya menjadi kurang stabil.

##### **2. Mobile Native Application**

*Mobile native application* adalah aplikasi yang secara khusus untuk platform *mobile* dan bahasa pemrograman tertentu. *Mobile native application* dibangun dengan menggunakan bahasa pemrograman untuk Eclipse IDE, *Objective C* untuk iOS (iphone) atau bahasa pemrograman lainnya. Tampilan antarmuka yang dihasilkan *mobile native application* adalah alami dan sesuai untuk platform yang dikembangkan. *Native* juga mendukung akses ke semua fitur perangkat keras smartphone misalnya *info device*, kamera, *compass*, file dan GPS. Pengembangan aplikasi terbatas yaitu hanya untuk satu sistem operasi saja. Hal

tersebut dikarenakan aplikasi *native* hanya dapat berjalan pada platform yang telah di spesifikasikan sejak pada tahap awal pengembangan aplikasi.

### 3. Mobile Hybrid Application

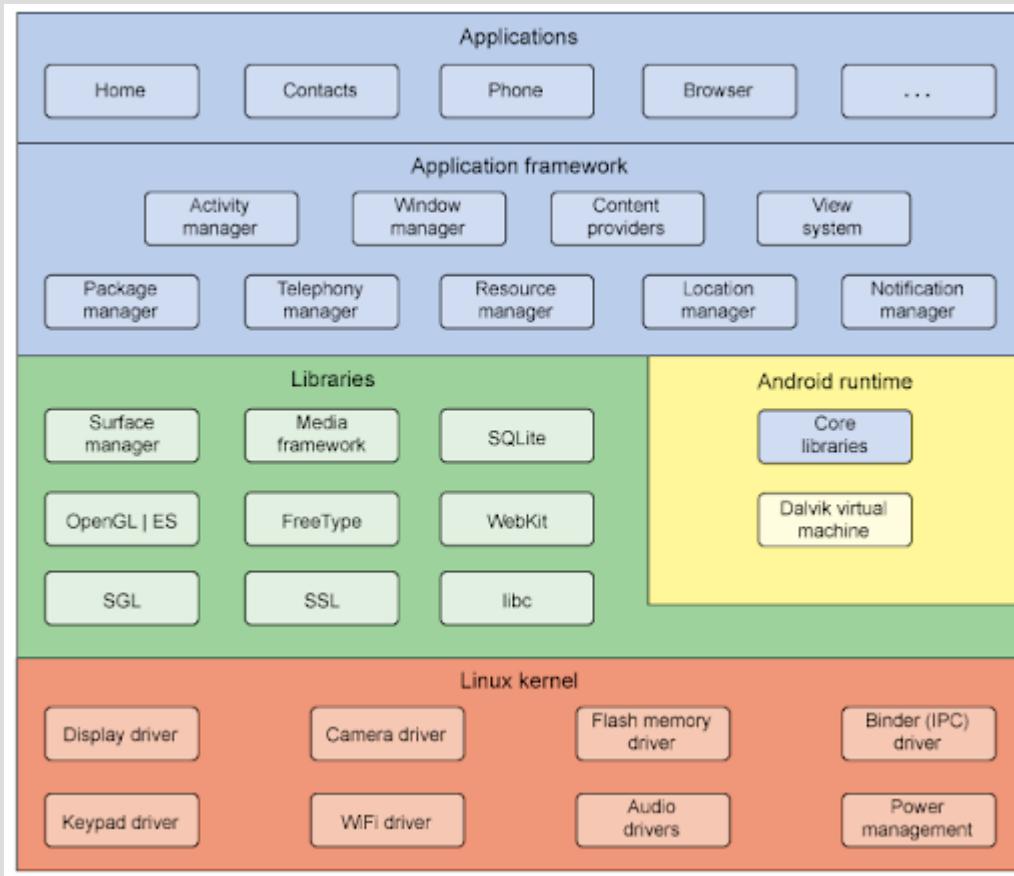
*Mobile hybrid application* adalah penggabungan dari aplikasi *website* dengan aplikasi *native*. Aplikasi hybrid mengubah aplikasi *website mobile* HTML5 ke aplikasi *native smartphone* yang akan ditargetkan. Pengubahan aplikasi *website mobile* ke dalam aplikasi *native* diperlukan semacam *framework* pengembangan aplikasi *mobile* sebagai dukungan perangkat lunak spesifik. Aplikasi hybrid memiliki tahap pengembangan yang relatif mudah dikarenakan menggunakan standar teknologi *website*. Aplikasi hybrid dapat berjalan di semua platform *mobile* yang dituju, yang berarti aplikasi hybrid dapat didistribusikan secara aplikasi *native* menuju ke platform *mobile* yang akan ditargetkan oleh pengembangan aplikasi. Aplikasi hybrid dapat mengakses fitur-fitur perangkat keras misalnya kamera, *wifi*, kalender dan *file* jika dilakukan distribusi aplikasi secara *native*.

## 2.2 Android

Android adalah platform sebuah sistem operasi yang bersifat *open source* untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi [17]. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi. Android dikembangkan oleh Android, Inc, dengan dukungan dari Google dan perusahaan yang didirikan pada tahun 2003 oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Operasi android menyediakan *memory management, processes, networks, drivers, security and abstraction*.

### 2.2.1 Arsitektur Android

Android dapat dikatakan sebagai sebuah tumpukan dari sistem operasi, *middleware* dan *key applications*. Menurut Holla [18], arsitektur sistem operasi android dapat dilihat pada Gambar 1.



**Gambar 1. Arsitektur Android**

Berikut ini adalah keterangan dari gambar arsitektur android di atas, yaitu:

1. *Kernel Linux*

Bagian terbawah dalam lapisan android yaitu Linux 2.6 dengan 115 patch. Lapisan ini menyediakan fungsi-fungsi dasar seperti manajemen proses, manajemen memori, manajemen perangkat seperti kamera, *keypad*, tampilan, dll. Kernel juga benar-benar baik seperti jaringan di area yang cukup luas.

2. *Libraries*

Di atas lapisan kernel Linux, terdapat sekumpulan *library* yang mengandung *Website browser* WebKit yang *open source* atau sering disebut dengan *library libc*, database Structured Query Language (SQL) *Lite* untuk penyimpanan dan berbagi data aplikasi, *library* untuk memainkan dan merekam audio/video, keamanan internet, dsb.

3. *Android Runtime*

Ini adalah bagian ketiga yang setara dengan *libraries* yang berada pada *layer* kedua. Bagian ini menyediakan komponen yang disebut *Dalvik Virtual Machine* yaitu sejenis Java *Virtual Machine* khusus untuk *design* dan optimasi android.

#### 4. Application Framework

*Layer application framework* menyediakan banyak *higher-level services* untuk aplikasi Java. *Developer* aplikasi diizinkan untuk memakai layanan aplikasi tersebut.

#### 5. Applications

Dari gambar diatas dapat dilihat bahwa lapisan tertinggi dari arsitektur android adalah lapisan aplikasi. Lapisan aplikasi adalah lapisan yang paling terlihat pada pengguna saat menjalankan program. Pengguna hanya melihat dan menggunakan aplikasi tanpa mengetahui alur proses yang terjadi pada *layer* aplikasi ini. Lapisan aplikasi berjalan pada android *runtime* dengan *service* dan kelas yang telah tersedia pada *framework* aplikasi. Lapisan aplikasi android memiliki perbedaan dengan sistem operasi lainnya, yaitu semua aplikasi berjalan diatas lapisan aplikasi dengan menggunakan *library* yang sama yaitu *Application Programming Interface (API)*.

### 2.2.2 Komponen Android

Konsep dasar mengembangkan aplikasi *mobile* adalah bagaimana menampilkan informasi di layar, melakukan proses yang dibutuhkan dan berpindah dari satu halaman ke halaman lain. Berikut merupakan komponen dari android yang dapat dilihat pada Tabel 1.

**Tabel 1. Komponen Android**

Komponen	Deskripsi
<i>Activity</i>	Untuk menampilkan antarmuka aplikasi.
<i>Service</i>	Untuk menjalankan proses aplikasi di belakang layar.
<i>Broadcast Receiver</i>	Untuk komunikasi antar <i>activity</i> , aplikasi dan antar <i>device</i> .
<i>Content Provider</i>	Untuk manajemen data dan <i>database</i> .

Dalam paper *Analyzing Inter-Application Communication in Android* mengungkapkan penjelasan mengenai komponen utama yang menyusun android, yaitu [19]:

#### 1. Activity

*Activity* menyediakan antarmuka pengguna. Kegiatan dimulai dengan *Intens*, dan mereka dapat *invoking components* setelah selesai. Semua bagian dapat terlihat dari *activities* suatu aplikasi.

#### 2. Services

*Services* berjalan di latar belakang dan tidak berinteraksi dengan pengguna. Mengunduh *file* atau dekompresi arsip adalah contoh dari operasi yang mungkin terjadi dalam *service*. Komponen lain dapat *bind to a service*, yang memungkinkan *binder* meng-*invoke* metode yang dideklarasikan dalam *interface* target *service*. *Intent* digunakan untuk *Start and bind to Services*.

### 3. *Broadcast Receiver*

*Broadcast receiver* merupakan sebuah komponen yang menerima dan merespon pemberitahuan. Banyak *broadcast* yang berasal dari kode sistem, salah satu contohnya adalah pemberitahuan dan baterai lemah. Aplikasi juga dapat menginisiasi *broadcast* contohnya, untuk memberi tahu aplikasi lain bahwa suatu data telah diunduh ke dalam perangkat dan siap digunakan. Sebuah aplikasi dapat memperoleh beberapa *broadcast receiver* untuk merespon pemberitahuan yang dianggap penting. Setiap *receiver* meng-*extend class BroadcastReceiver* dan dikirim sebagai sebuah objek *Intent*. *Broadcast receiver* tidak menampilkan *user interface*. Namun, akan memulai sebuah *activity* sebagai respon atas pemberitahuan yang diterima.

### 4. *Content Provider*

Sebuah *content provider* mengatur sharing aplikasi sehingga dapat digunakan oleh aplikasi lainnya. Data tersebut dapat disimpan di *file system*, dalam sebuah database *SQL Lite*, di *website*, atau dalam bentuk lain. *Content provider* akan meng-*extend class ContentProvider* untuk mengimplementasikan serangkaian *method* yang memungkinkan aplikasi lain untuk mengambil dan menyimpan data sesuai dengan yang dikendalikannya.

## 2.3. *Application Programming Interface (API)*

API adalah sekumpulan aturan yang ditentukan yang menjelaskan bagaimana komputer atau aplikasi berkomunikasi satu sama lain [19]. API berada di antara aplikasi dan server *website*, bertindak sebagai lapisan perantara yang memproses transfer data antar sistem. Berikut cara kerja API:

1. Aplikasi klien mulai panggilan API untuk mengambil informasi atau dikenal sebagai permintaan. Permintaan ini diproses dari aplikasi ke server *website* melalui *Uniform Resource Identifier (URI)* API dan menyertakan kata kerja permintaan, header, dan permintaan.

2. Setelah menerima permintaan yang *valid*, API melakukan panggilan ke program eksternal atau server *website*.
3. Server mengirimkan *respons* ke API dengan informasi yang diminta.
4. API mentransfer data ke aplikasi permintaan awal.

Transfer data yang bergantung pada *website* yang digunakan, proses permintaan dan respon terjadi melalui API. Sedangkan *Interface* pengguna dirancang untuk digunakan oleh *user*. API dirancang untuk digunakan oleh komputer atau aplikasi. API juga menawarkan keamanan berdasarkan desain karena posisinya sebagai perantara fungsionalitas antara dua sistem. *Request* API menyertakan *authorization credentials* untuk mengurangi risiko serangan pada *server*, dan *gateway* API dapat membatasi akses untuk meminimalkan ancaman keamanan. Selain itu, selama pertukaran HTTP *headers*, *cookie* atau *query string* parameter memberikan lapisan keamanan tambahan pada data. Berikut beberapa manfaat utama API seperti: kolaborasi yang ditingkatkan, inovasi yang lebih mudah, monetisasi data dan keamanan tambahan.

Saat ini, sebagian besar *interface* pemrograman aplikasi adalah API *website* yang mengekspos data dan fungsionalitas aplikasi melalui internet. Berikut adalah empat jenis utama API *website*:

1. *Open API* adalah antarmuka pemrograman aplikasi terbuka yang dapat diakses dengan protokol HTTP. *Open API* juga dikenal sebagai API publik yang telah menentukan titik akhir API serta format permintaan dan respon.
2. *Partner API* adalah antarmuka pemrograman aplikasi yang diekspos oleh strategi mitra bisnis. Pengembang dapat mengakses API dalam mode layanan melalui portal pengembang API publik.
3. *Internal API* adalah antarmuka pemrograman aplikasi yang tetap tersembunyi dari pengguna eksternal. API pribadi tidak tersedia untuk pengguna di luar perusahaan dan meningkatkan produktivitas komunikasi tim pengembangan internal.
4. *Composite API* menggabungkan beberapa data atau layanan API. Layanan ini memungkinkan pengembang untuk mengakses beberapa titik akhir dalam satu panggilan. *Composite API* berfungsi dalam arsitektur layanan mikro di mana melakukan satu tugas yang memerlukan informasi dari beberapa sumber.

Dikarenakan penggunaan API *website* telah meningkat, protokol tertentu telah dikembangkan untuk menyediakan pengguna dengan seperangkat aturan yang menentukan

jenis data dan perintah yang diterima. Oleh karena itu API memfasilitasi pertukaran informasi standar:

1. SOAP (*Simple Object Access Protocol*) adalah protokol API yang dibangun dengan XML, memungkinkan pengguna untuk mengirim dan menerima data melalui SMTP dan HTTP. Dengan SOAP API, lebih mudah untuk berbagi informasi antara aplikasi atau komponen perangkat lunak yang berjalan di lingkungan berbeda atau ditulis dalam bahasa berbeda.
2. XML-RPC adalah protokol yang mengandalkan format XML tertentu untuk mentransfer data, sedangkan SOAP menggunakan format XML berpemilik. XML-RPC lebih tua dari SOAP, tetapi jauh lebih sederhana, dan relatif ringan karena menggunakan bandwidth minimum.
3. JSON-RPC adalah protokol yang mirip dengan XML-RPC, karena keduanya adalah panggilan prosedur jarak jauh (RPC), tetapi yang ini menggunakan JSON, bukan format XML untuk mentransfer data. Kedua protokol itu sederhana. Meskipun panggilan mungkin berisi beberapa parameter, mereka hanya mengharapkan satu hasil.
4. REST (*Representational State Transfer*) adalah sekumpulan prinsip arsitektur API *website*, yang berarti tidak ada standar resmi (tidak seperti yang memiliki protokol). Untuk menjadi REST API (RESTful API), antarmuka harus mematuhi batasan arsitektur tertentu. RESTful API dapat dibuat dengan protokol SOAP, tetapi kedua standar tersebut biasanya dipandang sebagai spesifikasi yang bersaing.

Secara tradisional, API mengacu pada antarmuka yang terhubung ke aplikasi yang mungkin telah dibuat dengan salah satu bahasa pemrograman tingkat rendah, seperti Javascript. API modern mematuhi prinsip REST dan format JSON dan biasanya dibuat untuk HTTP, menghasilkan antarmuka ramah pengembang yang mudah diakses dan dipahami secara luas oleh aplikasi yang ditulis dalam Java, Ruby, Python, dan lain sebagainya. Saat menggunakan API, ada dua pendekatan arsitektur yang umum yaitu *Service Oriented Architecture (SOA) and Microservices Architecture*.

1. SOA adalah gaya desain perangkat lunak di mana fitur-fiturnya dipecah dan tersedia sebagai layanan terpisah dalam jaringan. Biasanya, SOA diimplementasikan dengan layanan *website*, membuat blok bangunan fungsional dapat diakses melalui protokol komunikasi standar. Pengembang dapat

membangun layanan ini dari awal, tetapi biasanya membuatnya dengan menampilkan fungsi dari sistem lama sebagai antarmuka layanan.

2. *Microservices architecture* adalah gaya arsitektur alternatif yang membagi aplikasi menjadi komponen independen yang lebih kecil. Menerapkan aplikasi sebagai kumpulan layanan terpisah akan memudahkan pengujian, pemeliharaan, dan penskalaan. Metodologi ini menjadi terkenal di seluruh era komputasi awan, memungkinkan pengembang untuk bekerja pada satu komponen yang independen dari yang lain.

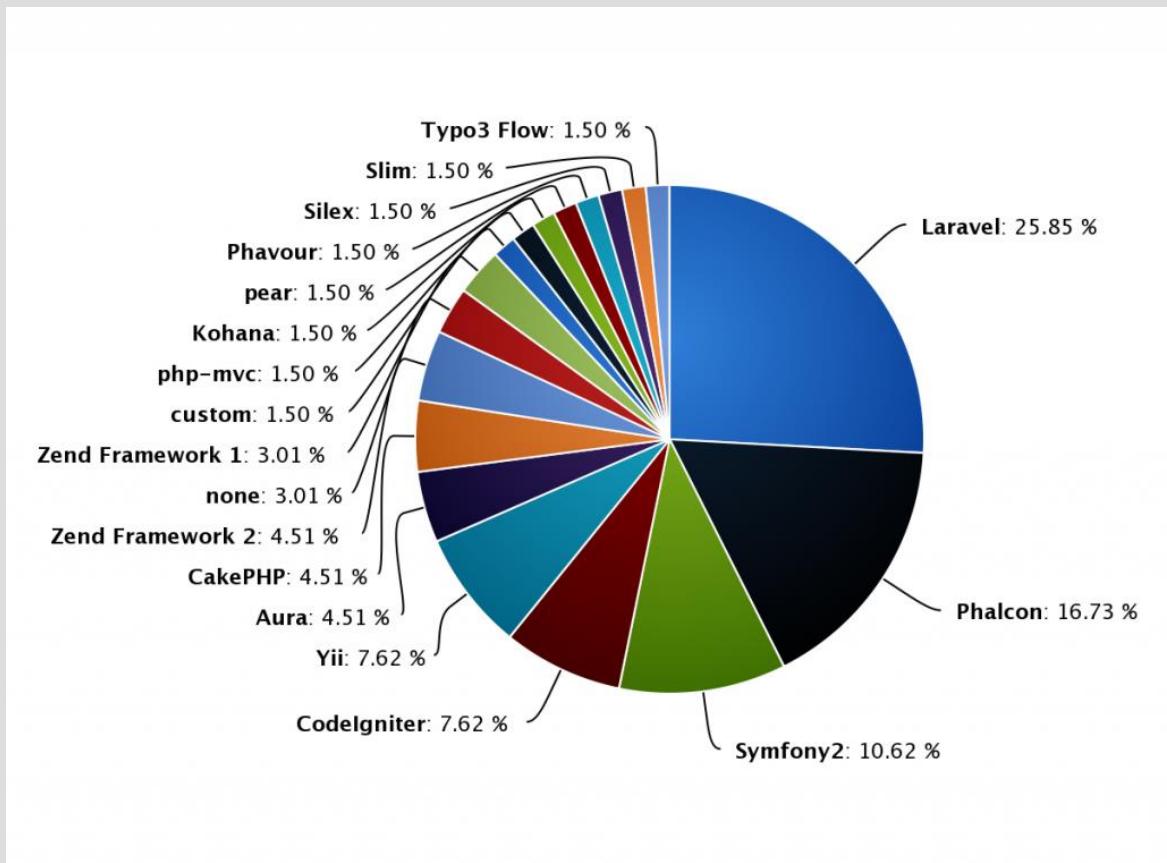
SOA adalah langkah *evolusioner* penting dalam pengembangan aplikasi, arsitektur layanan mikro dibangun untuk menyesuaikan skala, menyediakan pengembang dan perusahaan dengan kelincahan dan fleksibilitas dalam membuat, memodifikasi, menguji, dan menyebarkan aplikasi pada tingkat *granular*, dengan siklus iterasi yang lebih pendek dan sebagainya.

## 2.4 Framework Laravel

*Framework* disebut sebagai suatu pendukung dalam sebuah perancangan *software* yang dapat diorganisir dan dikembangkan. *Framework* meliputi beberapa pendukung seperti: kumpulan *source code* atau *libraries*, bahasa *scripting* atau beberapa perangkat lunak lainnya untuk membantu mengembangkan komponen-komponen yang berbeda menjadi satu perangkat lunak [20]. Keberadaan *framework* sangat membantu programmer dalam pembuatan aplikasi karena dengan adanya *framework*, programmer tidak lagi membuat segala sesuatu dari awal melainkan programmer hanya perlu memperhatikan pembuatan modul-modul pada aplikasi yang akan dibuat karena *framework* telah menyediakannya [21].

Dalam pengembangan CIS berbasis *mobile* dengan layanan API ini *framework* yang akan digunakan adalah *framework* Laravel. Laravel adalah satu *framework* PHP yang bersifat terbuka atau *open-source* dengan desain MVC dalam pembangunan sebuah aplikasi atau website [21]. Laravel dirilis pada tahun 2011 oleh programmer Amerika yang bernama Taylor Otwell. Laravel dapat disebut sebagai kumpulan-kumpulan kode program yang disusun pada sebuah folder sehingga dapat digunakan dan menjadi sebuah *framework*. Semenjak Laravel dirilis, *framework* Laravel ini telah menjadi *framework* terbaik dan yang paling sering digunakan oleh programmer dunia dengan popularitas sebesar 25,85%.

Berikut Gambar 2 yang menunjukkan kedudukan *framework* Laravel dibandingkan *framework* lainnya.



Gambar 2. Popularitas pengguna *framework* Laravel

*Framework* Laravel memiliki beberapa keunggulan yang membuat disukai para programmer dunia. Berikut keunggulan *framework* Laravel yaitu [21]:

1. Laravel mempunyai banyak fitur yang tidak dimiliki oleh *framework* lain.
2. Laravel adalah *framework* PHP yang bersifat ekspresif, dimana sintaks yang terdapat pada Laravel menggunakan bahasa yang mudah dimengerti oleh programmer baik itu pemula atau yang sudah paham karena sintaks dari *framework* Laravel mudah dipahami
3. Laravel mempunyai dokumentasi lengkap, setiap versi mempunyai dokumentasi tersendiri mulai dari cara instalasi hingga cara penggunaan fitur-fiturnya
4. Laravel banyak digunakan para programmer sehingga banyak *library* yang mendukung Laravel
5. Laravel didukung *Composer* sehingga *library* Laravel dapat diperoleh dengan mudah dari internet menggunakan *Composer*. *Composer* adalah *Dependency*

*Management PHP* yang dapat digunakan untuk mendapatkan *library* yang kita pakai dan menginstalnya dari internet.

6. Laravel mempunyai *template engine* yang diberi nama *blade*, *template* ini memudahkan kita dalam menampilkan data pada *template* HTML.

Dalam penggunaan *framework* Laravel membutuhkan beberapa *software* yang harus diinstal dalam komputer. Berikut beberapa *software* di antaranya sebagai berikut:

1. *Text Editor*

*Text editor* ini berfungsi untuk menulis pemrograman *website*. Beberapa *text editor* yang bisa digunakan oleh para programmer di antaranya adalah *Notepad++*, *Sublime Text*, *Brackets* dan sebagainya

2. *Website Server*

*Website server* berfungsi sebagai penyedia layanan *website* pada komputer lokal. *Framework* Laravel mendukung *website server* Apache, Nginx maupun Xampp yang di dalamnya menggunakan server Apache.

3. *Composer*

*Composer* berguna untuk memudahkan instalasi Laravel dan mendapatkan *library* yang dibutuhkan dari internet. Saat *install Composer*, pastikan mendeteksi letak PHP dan terletak di *folder* xampp/php.

## 2.5 Kotlin

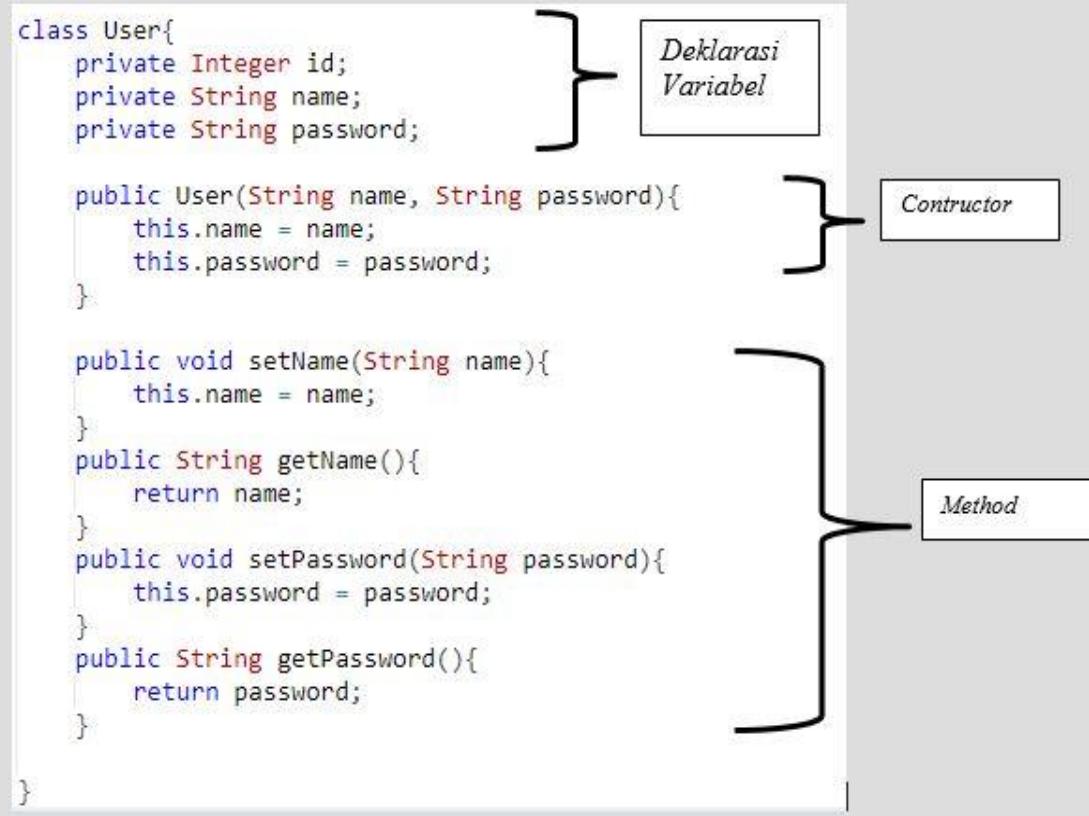
Kotlin adalah bahasa *modern*, statis, dan kompatibel dengan android yang memperbaiki banyak masalah Java seperti *null pointer exception* atau penggunaan *code* yang berlebihan [22]. Kotlin berjalan di *Java Virtual Machine* dan juga dikompilasi ke kode sumber Javascript. Kotlin dirilis ke publik pada Februari 2016. Yang utama pengembangan berasal dari tim pemrograman berbasis JetBrains di Saint Petersburg, Rusia [23].

Setiap *programmer* menginginkan kode yang sederhana. Sederhana disini ialah dimana *programmer* dapat menulis kode yang singkat namun dimengerti oleh orang lain dengan mudah. Kotlin memiliki kelebihan antara lain [23]:

1. *Data Classes*

*Class* adalah hal yang akan sering digunakan untuk menyimpan data. Pada Java, sebuah *class* memiliki 3 bagian utama yaitu *constructor*, *variable*, *method*. *Constructor* digunakan pada saat ingin menginstansiasi objek yang menggunakan kelas tersebut. *Variabel* adalah anggota dari *class* yang akan di deklarasi. *Method* adalah fungsi-fungsi yang merupakan anggota dari *class* itu sendiri. *Method* sendiri

mengandung berbagai atribut yang menentukan level akses, *return type*, nama, *logic* dan parameter.



Gambar 3. Membuat Class dengan Java

Pada Kotlin, *class* melakukan *auto-generate* pada semua properti termasuk *method* yang sering dipakai seperti *toString()*, *setter* and *getter*, dan lain nya secara langsung dapat di *generate* dengan menggunakan data *class* pada kode nya.

```
data class User(var id:Int
                ,var name:String
                ,var password:String)
```

Gambar 4. Membuat Class dengan Kotlin

## 2. Null Safety

Pada Java dalam melakukan pengembangan *mobile* sering terjadi masalah yang dihadapi oleh *programmer* adalah *Null Pointer Exception* (NPE). NPE bisa terjadi, jika aplikasi mencoba mengakses objek yang bernilai *null*. Bisa diatasi dengan *try-catch* atau *annotation*. Pada Kotlin tidak ada variabel yang nilainya berasal *null*. Akan tetapi, *programmer* dapat menambahkan “?” (tanda tanya) pada akhir dari sebuah variabel dan tipe data harus dinyatakan secara eksplisit untuk menandakan bahwa variabel tersebut bernilai *null*.

### 3. Extension Function

Dengan melakukan *Extension Function* programmer dapat menambahkan fungsi ke dalam beberapa kelas, bahkan saat kita tidak memiliki akses ke kelas tersebut. Contoh pada saat kita ingin menghapus 1 karakter pada string kita dapat menggunakan contoh seperti dibawah ini

```
fun String.removeFirstLastChar(): String =  
    this.substring(1, this.length - 1)  
fun main(args: Array<String>) {  
    val myString= "Hello Everyone"  
    val result = myString.removeFirstLastChar()  
    println("First character is: $result")  
}
```

Gambar 5. Membuat *Extension Function*

## 2.6 Android Studio

Android Studio adalah pengembang dari Eclipse IDE yaitu IntelliJ IDEA [24]. Android Studio diluncurkan pada tahun 2013 oleh *Google* di *event* konferensi *Google I/O* [25]. Android Studio adalah IDE yang digunakan untuk pengembangan aplikasi berbasis android. Dibandingkan dengan Eclipse IDE, Android Studio lebih banyak memiliki fitur-fitur baru [26].

Berikut ini fitur-fitur Android Studio:

1. *Environment*: berfungsi untuk mempermudah dalam pengembangan aplikasi berbasis android
2. *Support*: berfungsi dalam pengembangan aplikasi android TV dan *Wear*
3. *Template*: digunakan dalam menentukan desain dan komponen-komponen android
4. Memiliki tata letak Editor *interface drag and drop*
5. Khusus android mudah dalam *refactoring* dan perbaikan
6. Mendukung *build* yang berbasis Gradle
7. Memiliki integrasi *ProGuard*
8. Memiliki berbagai fitur dan emulator yang cepat
9. Dapat diintegrasikan dengan Google *Cloud Messaging* dan *App Engine*
10. Mendukung program basic C++ dan NDK

Dengan Android Studio para pengembang dengan mudah membangun aplikasi, mengedit dan melakukan *testing* dengan hanya menggunakan *tools* android studio. Android studio mempunyai sistem operasi yang bersifat terbuka sehingga akan mempermudah untuk

mengembangkan sebuah aplikasi. Android Studio *supports* banyak versi seperti Oreo, Kitkat, Jelly Bean dan lain sebagainya.

Setiap proyek di Android studio berisi satu atau beberapa modul dengan file *source code*. Jenis modul meliputi [27]:

1. Modul aplikasi android
2. Modul *Library*
3. Modul Google *App Engine*

Secara default, Android studio menampilkan proyek dalam tampilan proyek android, seperti yang ditunjukkan pada Gambar 6.

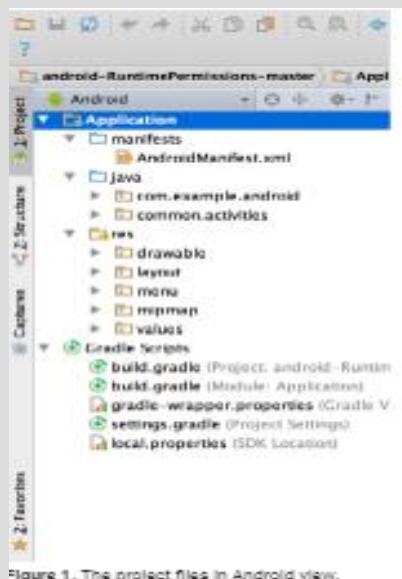


Figure 1. The project files in Android view.

Gambar 6. Tampilan Android Studio

Tampilan ini diatur berdasarkan modul untuk memberikan akses cepat ke *file source* proyek. Semua file *build* berada di tingkat atas pada *Gradle Scripts* serta setiap modul memiliki folder sebagai berikut:

1. *Manifests*: berisi *file* AndroidManifest.xml.
2. *Java*: berisi *file source code*, termasuk kode pengujian JUnit.
3. *Res*: berisi semua *source non-code*, seperti *layout XML*, *string User Interface (UI)* dan gambar bitmap.

Android studio memiliki beberapa jendela utama seperti: *toolbar*, *navigation bar*, *editor window*, *tool window bar*, *tool windows* dan *status bar*.

## 2.7 Trello

Trello adalah aplikasi berbasis *website* dan alat manajemen proyek yang menawarkan serangkaian fitur sebagai semacam *board*, tempat untuk menempatkan

pemberitahuan tugas tertentu [28]. Dalam Trello juga dapat digunakan secara kolaborasi yang tidak harus bertemu secara langsung dalam satu ruangan atau tempat. Adapun bagian-bagian dan fungsi-fungsi dari Trello adalah sebagai berikut [29]:

1. *Board* (Papan) adalah sebuah tempat atau wadah untuk menambah, mengubah atau mengatur kata-kata yang akan dikerjakan didalam *board* tersebut.
2. *List* (Daftar) adalah nama tahapan yang akan dijalani oleh sebuah *card* (tugas).
3. *Card* (Tugas) adalah unit yang merupakan tugas yang akan dikerjakan (*to do list*) yang dapat di *drag* dan *drop* dari panel *board*.

Dalam mengimplementasikan bingkai kerja *scrum* pada pengembangan fitur-fitur CIS berbasis *mobile* akan menggunakan aplikasi. Trello sebagai *project management* yang juga akan dibuatkan dokumentasi berupa laporan yang akan disertakan dalam lampiran laporan tugas akhir ini.

## **2.8 Campus Information System (CIS) IT Del**

*Campus Information System* (CIS) merupakan sebuah sistem informasi universitas yang berguna untuk mengumpulkan data grafik dan non-grafik terkait universitas. Data-data yang berada pada CIS akan disimpan dalam sebuah database dan akan ditampilkan kepada pengguna dengan berbagai format seperti *website*. Institut Teknologi Del menyediakan sistem layanan informasi yang disebut CIS. CIS berbasis *website* yang berguna untuk menyediakan informasi-informasi kegiatan akademik, kemahasiswaan, keuangan dan keasramaan. CIS IT Del saat ini masih berbasis *website* dan digunakan oleh staf, keasramaan dan mahasiswa. CIS ini memiliki banyak fitur yang berbeda untuk setiap *user*, dimana fitur staf berbeda dengan fitur keasramaan begitu juga fitur mahasiswa.

Adapun fitur-fitur yang terdapat dalam CIS IT Del untuk mahasiswa yaitu:

1. Fitur Otentikasi
2. Fitur Data Diri
3. Fitur Perkuliahan
4. Fitur Kartu Rencana Studi
5. Fitur Bursar
6. Fitur Artikel
7. Fitur Publikasi
8. Fitur *Request* Izin Bermlaan
9. Fitur *Request* Izin Keluar
10. Fitur *Request* Izin Kolaboratif

11. Fitur *View Poin Pelanggaran*
12. Fitur Penggantian KTM
13. Fitur Surat Mahasiswa Aktif
14. Fitur Surat Pengantar Proyek
15. Fitur Surat Pengantar Magang
16. Fitur Surat Pengantar Kompetisi
17. Fitur Pengumuman dan Paket
18. Fitur *Request* Peminjaman Alat Laboratorium dan Pemesanan Bahan
19. Fitur *View Polling*, Kuesioner

Melalui fitur-fitur tersebut mahasiswa dapat melihat informasi-informasi akademik maupun non akademik. Mahasiswa juga dapat melakukan *request* izin bermalam dan *request* izin keluar. Ketika mahasiswa ingin melakukan izin keluar maupun izin bermalam mahasiswa dapat melakukan *request* melalui menu keasramaan yaitu izin bermalam dan izin keluar. Mahasiswa akan mengisi *form request* lalu menggu *approve* dari keasramaan untuk izin bermalam dan dosen wali, BAAK untuk izin keluar. Mahasiswa dapat mengubah *request* izin ketika *request* belum di *approve*. Untuk informasi *approve request* dapat dilihat/diketahui mahasiswa apabila membuka menu izin bermalam dan izin keluar kembali dikarenakan belum ada notifikasi untuk pemberitahuan *approve request*.

Selain itu, mahasiswa juga dapat melihat informasi paket melalui fitur paket dan notifikasi. Dimana ketika mahasiswa mendapat paket di pos satpam, mahasiswa akan mendapat notifikasi informasi paket. Akan tetapi notifikasi ini akan didapat oleh mahasiswa apabila sedang mengakses dan *login* ke CIS. Hal ini dikarenakan CIS masih berbasis website dan belum ada notifikasi langsung ke mahasiswa. Berdasarkan uraian diatas dapat disimpulkan bahwa mahasiswa masih banyak mengalami kendala dalam mendapat informasi dari CIS IT Del berbasis *website*.

## Kesimpulan

Berdasarkan hasil dari tinjauan pustaka, maka dapat disimpulkan bahwa pengembangan Fitur-Fitur *Campus Information System* (CIS) berbasis *mobile* dengan layanan API akan diimplementasikan menggunakan beberapa *tools* dan *platform*. Adapun teknologi yang digunakan yaitu *mobile application* sebagai tempat berjalannya program yang akan digunakan mahasiswa. *Platform* yang digunakan yaitu Android sebagai *platform* sebuah sistem operasi untuk perangkat *mobile* berbasis linux. Selain itu pengembangan CIS akan dikembangkan dengan memperhatikan setiap aspek yang diperlukan seperti aspek

penggunaan trello. Trello akan digunakan oleh pengembang sebagai bingkai kerja dalam pembangunan CIS berbasis *mobile*. API akan digunakan sebagai perantara database dengan aplikasi CIS *mobile* menggunakan *framework* Laravel. Dan untuk bahasa pemrograman yang akan digunakan dalam membangun CIS berbasis *mobile* adalah bahasa pemrograman Kotlin.

## **BAB III**

### **ANALISIS DAN DESAIN**

Pada BAB III Perencanaan dan Analisis berisi uraian mengenai analisis proyek, project plan dan analisis yang dilakukan untuk mendapatkan rancangan aplikasi yang akan dikembangkan.

#### **3.1 Inisialisasi Proyek**

Mahasiswa Institut Teknologi Del (IT Del) merupakan *client* dari pengembangan sistem ini. Sistem yang akan dibangun adalah *Campus Information System* (CIS) berbasis *mobile* dengan layanan *Application Programming Interface* (API). Pengerjaan *Campus Information System* (CIS) berbasis *mobile* ini mencakup tiga belas fitur yaitu fitur otentikasi, fitur notifikasi, fitur *request Izin Bermalam* (IB), fitur *request Izin Keluar* (IK), fitur perkuliahan, fitur data diri, fitur kuesioner, fitur melihat pengumuman, fitur Kartu Rencana Studi (KRS), fitur melihat paket, fitur unduh modul, fitur melihat grafik Indeks Prestasi (IP) dan fitur melihat jadwal kuliah.

#### **3.2 Project Plan**

Berikut akan dijelaskan bagaimana *developer* membangun *Campus Information System* (CIS) berbasis *mobile* dengan layanan *Application Programming Interface* (API).

1. Melakukan bimbingan dengan pembimbing mengenai sistem yang akan dibangun.
2. Melakukan wawancara secara *online* ke pihak Sumber Daya Informasi (SDI) Institut Teknologi Del (IT Del) untuk melakukan *requirement gathering*, validasi dan *User Acceptance Testing* (UAT).
3. Untuk memanajemen waktu maka *developer* membuat target penggeraan setiap minggunya dalam Trello.

#### **3.3 Pengumpulan Data**

Pada subbab ini akan dijelaskan mengenai pengumpulan data dalam bentuk kuesioner yang dilakukan kepada mahasiswa Institut Teknologi Del (IT Del) yang nantinya akan menjadi pengguna dari sistem ini. Berikut akan dijelaskan tahapan dalam melakukan pengumpulan data.

##### **3.3.1 Persiapan Survei**

Pada tahapan ini akan dijelaskan mengenai persiapan survei yang dilakukan. Survei ini akan ditujukan kepada mahasiswa IT Del baik alumni atau pun mahasiswa aktif yang

dimana mahasiswa tersebut telah mengenal dan juga menggunakan *Campus Information System* (CIS). Teknik pengumpulan data yang dilakukan dalam bentuk kuesioner atau angket yang melalui *Google form* yang akan diisi langsung oleh mahasiswa IT Del. Sebelum menyebarkan kuesioner adapun persiapan yang dilakukan, yaitu:

1. Melakukan evaluasi terhadap CIS berbasis *website* terlebih dahulu. Hal ini bertujuan untuk mempermudah dalam penyusunan pertanyaan yang akan diberikan kepada mahasiswa IT Del nantinya.
2. Membuat daftar pertanyaan yang akan diberikan kepada mahasiswa IT Del. Pertanyaan yang akan diberikan dalam bentuk *Google form* yang tentunya akan direspon oleh mahasiswa IT Del nantinya.

### **3.3.2 Pelaksanaan Survei**

Tahapan selanjutnya yaitu melaksanakan survei ataupun kuesioner. Pelaksanaan survei ini dilakukan dengan media *Google form* sebagai perantaranya. Mahasiswa IT Del akan menjawab pertanyaan yang berada di *Google form* yang telah dibuat sebelumnya. Tujuan dari survei ini untuk mendapatkan tanggapan dari mahasiswa IT Del terkait pengembangan CIS dalam versi *mobile*. Serta bertujuan untuk membantu *developer* untuk mengetahui fitur atau fungsi - fungsi yang lebih dibutuhkan oleh mahasiswa di dalam aplikasi CIS *mobile* ini nantinya. Sehingga aplikasi ini akan dapat memenuhi kebutuhan dari mahasiswa IT Del. Adapun mahasiswa IT Del dapat mengisi *Google form* yang telah disebarluaskan melalui media komunikasi *whatsapp*.

### **3.3.3 Hasil Survei**

Setelah dilakukan beberapa uji statistika pada hasil pengujian kuesioner yang telah dilakukan, maka kami mendapatkan hasil sebagai berikut:

1. Jumlah mahasiswa IT Del yang mengisi kuesioner ada sebanyak 160 orang dengan kata lain ada sebanyak 160 responden.
2. *Developer* mengetahui fitur - fitur atau fungsi - fungsi apa saja yang lebih dibutuhkan oleh mahasiswa IT Del berdasarkan respon mahasiswa sendiri.
3. Dari seluruh responden keseluruhannya setuju dengan adanya pengembangan untuk CIS berbasis *mobile*.

Adapun hasil Kuesioner dapat dilihat pada Lampiran 1.

### **3.4 Kebutuhan Fungsional**

Kebutuhan fungsional adalah kebutuhan/layanan yang harus disediakan oleh aplikasi yang akan dibangun. Adapun kebutuhan fungsionalnya adalah:

- 1. Fungsi Notifikasi**

Fungsi ini digunakan oleh mahasiswa jika ada pemberitahuan mengenai unggah modul baru oleh dosen, informasi paket dari pihak satpam, informasi persetujuan Izin Bermalam (IB) oleh keasramaan dan informasi persetujuan Izin Keluar (IK) oleh dosen wali, keasramaan dan Biro Administrasi Akademik dan Kemahasiswaan (BAAK).

- 2. Fungsi *Add Request* Izin Bermalam (IB)**

Fungsi ini digunakan oleh mahasiswa untuk menambahkan *request* IB.

- 3. Fungsi *Edit Request* Izin Bermalam (IB)**

Fungsi ini digunakan oleh mahasiswa untuk mengedit *request* IB sebelum disetujui oleh keasramaan.

- 4. Fungsi *Cancel Request* Izin Bermalam (IB)**

Fungsi ini digunakan oleh mahasiswa untuk membatalkan *request* IB sebelum disetujui oleh keasramaan.

- 5. Fungsi *Add Request* Izin Keluar (IK)**

Fungsi ini digunakan oleh mahasiswa untuk menambahkan *request* IK..

- 6. Fungsi *Edit Request* Izin Keluar (IK)**

Fungsi ini digunakan oleh mahasiswa untuk mengedit *request* IK sebelum disetujui oleh dosen wali, keasramaan dan Biro Administrasi Akademik dan Kemahasiswaan (BAAK).

- 7. Fungsi *Cancel Request* Izin Keluar (IK)**

Fungsi ini digunakan oleh mahasiswa untuk membatalkan *request* IK sebelum disetujui oleh dosen wali, keasramaan dan Biro Administrasi Akademik dan Kemahasiswaan (BAAK).

- 8. Fungsi Melihat Matakuliah**

Fungsi ini digunakan mahasiswa untuk melihat materi perkuliahan yang telah diunggah oleh dosen pengampu.

- 9. Fungsi Unduh Modul**

Fungsi ini digunakan oleh mahasiswa untuk mengunduh materi atau modul perkuliahan yang telah diunggah oleh dosen pengampu.

## **10. Fungsi Melihat Jadwal**

Fungsi ini digunakan oleh mahasiswa untuk melihat jadwal perkuliahan serta daftar ruangan yang digunakan untuk perkuliahan.

## **11. Fungsi Melihat Data Diri**

Fungsi ini digunakan oleh mahasiswa untuk melihat data-data mahasiswa mulai dari data akademis, data pribadi, data orangtua dan data asal sekolah.

## **12. Fungsi Edit Data Diri**

Fungsi ini digunakan oleh mahasiswa untuk mengedit data diri mahasiswa mulai dari data akademis, data pribadi, data orangtua dan data asal sekolah.

## **13. Fungsi Melihat Grafik Indeks Prestasi (IP)**

Fungsi ini digunakan oleh mahasiswa untuk melihat grafik Indeks Prestasi (IP) yang di dapat setiap semesternya.

## **14. Fungsi Melihat Kuesioner**

Fungsi ini digunakan oleh mahasiswa untuk melihat daftar-daftar kuesioner yang ditujukan terhadap mahasiswa.

## **15. Fungsi Mengisi Kuesioner**

Fungsi ini digunakan oleh mahasiswa untuk mengedit atau mengisi kuesioner yang ditujukan terhadap mahasiswa.

## **16. Fungsi Melihat Pengumuman**

Fungsi ini berfungsi untuk melihat pengumuman-pengumuman yang diberikan kepada mahasiswa.

## **17. Fungsi Melihat Paket**

Fungsi ini digunakan oleh mahasiswa untuk melihat paket mahasiswa yang telah didaftarkan oleh pihak satpam.

## **18. Fungsi Add Kartu Rencana Studi (KRS)**

Fungsi ini digunakan oleh mahasiswa untuk menambahkan KRS sesuai matakuliah yang diambil dalam satu semester.

## **19. Fungsi Edit Kartu Rencana Studi (KRS)**

Fungsi ini digunakan oleh mahasiswa untuk mengedit KRS.

## **20. Fungsi *Login***

Fungsi ini digunakan oleh mahasiswa untuk dapat masuk ke aplikasi *Campus Information System (CIS)*.

## **21. Fungsi *Logout***

Fungsi ini digunakan oleh mahasiswa untuk keluar dari aplikasi *Campus Information System* (CIS).

### 3.5 Kebutuhan Non-Fungsional

Kebutuhan fungsional adalah kebutuhan/layanan yang harus disediakan oleh aplikasi yang akan dibangun [30]. Adapun kebutuhan fungsionalnya adalah:

**Tabel 2. Kebutuhan Non-Fungsional**

Non Fungsional Requirement Id	Parameter	Requirement
NFR_01	<i>Availability</i>	Sistem harus bisa kembali diakses selama 24 jam tanpa ada kegagalan.
NFR_02	<i>Reliability</i>	Dalam mengakses sistem, tidak ada kegagalan.
NFR_03	<i>Ergonomics</i>	Sistem memiliki kenyamanan bagi Mahasiswa IT Del karena desain serta informasi dalam aplikasi.
NFR_04	<i>Response time</i>	Pada saat ingin melihat informasi atau ingin membuka modul dan yang lainnya, sistem mampu menampilkan hasil kurang dari 2 menit.
NFR_05	<i>Safety</i>	Sistem nyaman untuk diakses dan digunakan.
NFR_06	<i>Security</i>	Untuk menjaga keamanan, sistem hanya dapat digunakan oleh mahasiswa IT Del yang memiliki akun.
NFR_07	Bahasa komunikasi	Semua informasi dalam bahasa Indonesia, sehingga mudah dipahami.

### 3.6 User Characteristic

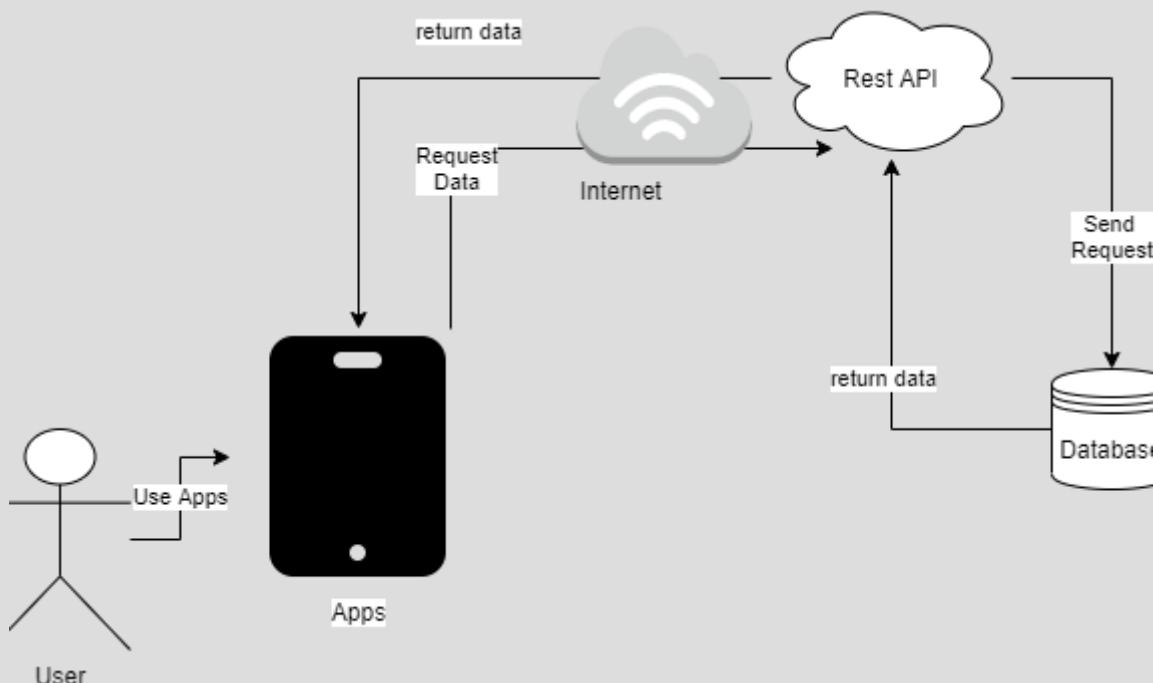
Aplikasi yang dibangun memiliki 1 target *user*, yaitu mahasiswa Institut Teknologi Del (IT Del). Berikut adalah detail karakteristik dari pengguna.

### 3.6.1 User Group 1

Tabel 3. User Group 1

<i>Description of User</i>	: <i>User</i> merupakan pengguna aplikasi yang telah ditentukan yaitu mahasiswa Institut Teknologi Del.
<i>Role</i>	: <i>User</i>
<i>Prerequisite</i>	: <i>User</i> harus <i>login</i> terlebih dahulu untuk dapat mengakses aplikasi.
<i>Task description</i>	: <i>User</i> melihat notifikasi, <i>request</i> Izin Bermalam (IB), <i>request</i> Izin Keluar (IK), perkuliahan, data diri, kuesioner, melihat pengumuman, kartu rencana studi, melihat paket, unduh modul, melihat grafik IP dan melihat jadwal perkuliahan.

### 3.7 Arsitektur Sistem



Gambar 7. Arsitektur Sistem

#### Keterangan:

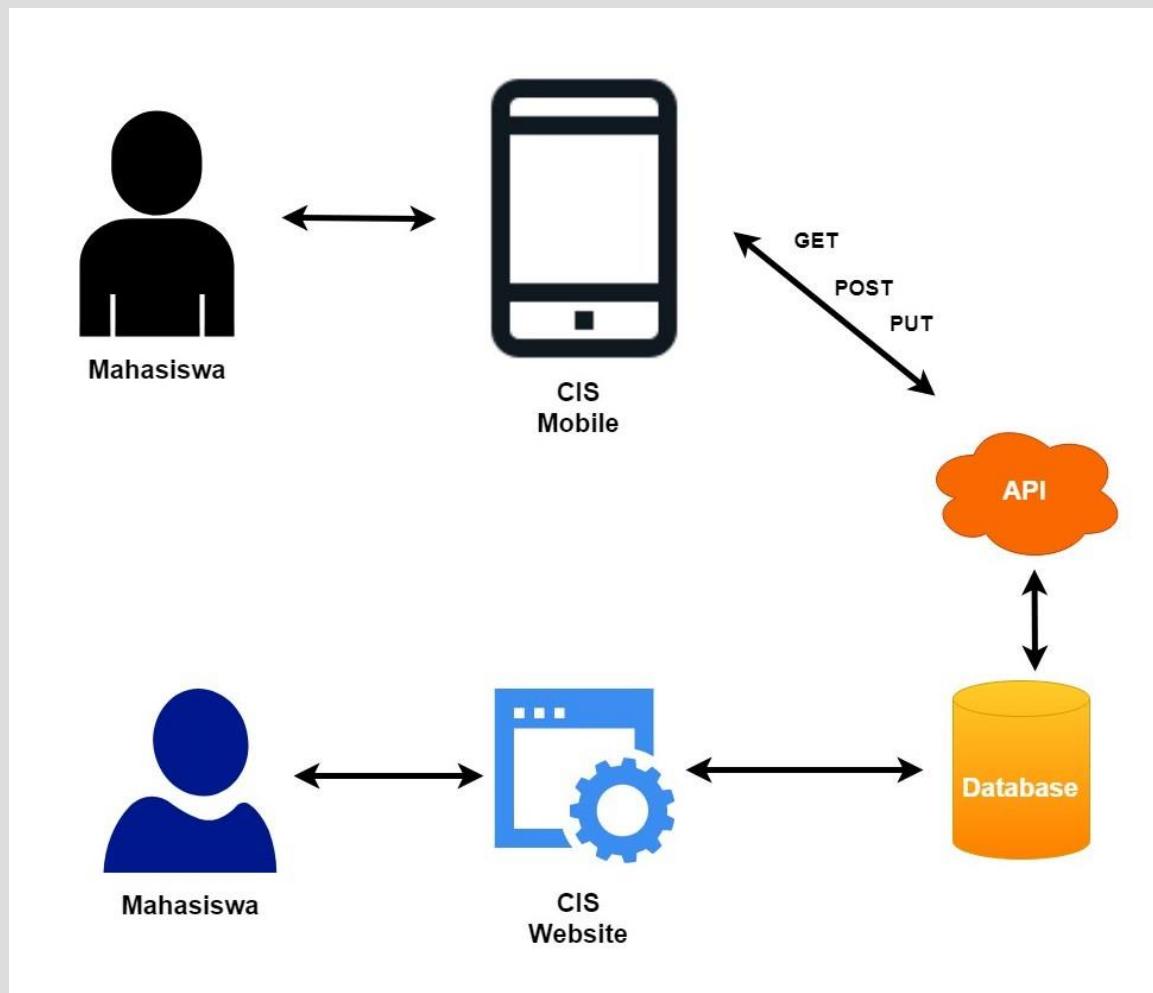
1. *User* akan mengakses aplikasi CIS melalui *smartphone* dengan membutuhkan koneksi Internet.
2. Aplikasi akan mengirimkan *request data* ke *API*, data tersebut akan diambil dari database serta akan di *load* sehingga data-data yang dibutuhkan dapat ditampilkan di sisi *user*.

### 3.8 Arsitektur Uniform Resource Identifier (URI).

Pada subbab ini akan dijelaskan arsitektur URI dalam pengembangan aplikasi pada pengerjaan Tugas Akhir. Adapun arsitektur URI dapat dilihat pada Lampiran 2.

### 3.9 Arsitektur Integrasi CIS Mobile dan Website

Berikut Gambar 8 arsitektur integrasi CIS berbasis *mobile* dengan CIS berbasis *website*.



Gambar 8. Integrasi CIS *mobile* dan CIS *website*

#### Keterangan:

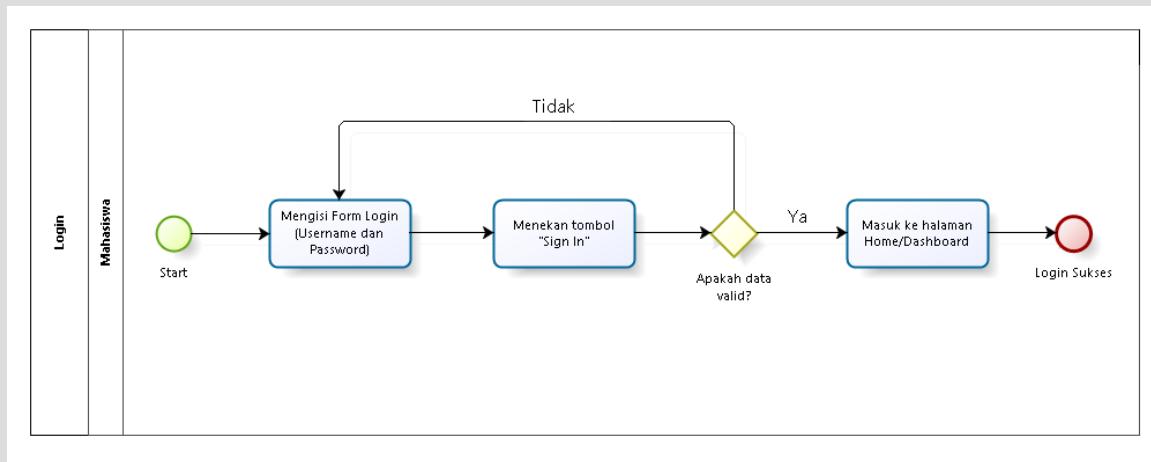
1. Mahasiswa sebagai *user* akan mengakses CIS yang ada di *website* yang dimana pada CIS *website* akan mengakses database yang ada pada CIS *website*
2. Pada CIS *mobile*, Mahasiswa akan mengakses aplikasi tersebut dan pada aplikasi tersebut akan mengambil data yang ada pada API dan API tersebut akan mengakses langsung kepada database

### 3.10 Current System

Pada sub bab ini akan diuraikan sistem yang berjalan saat ini dalam bentuk *Business Process Modeling and Notation* (BPMN). BPMN adalah sebuah standar untuk menggambarkan proses bisnis.

#### 1. Melakukan *Login*

Pada Gambar 9 akan menggambarkan *current system* bisnis proses dari *login* ke sistem.

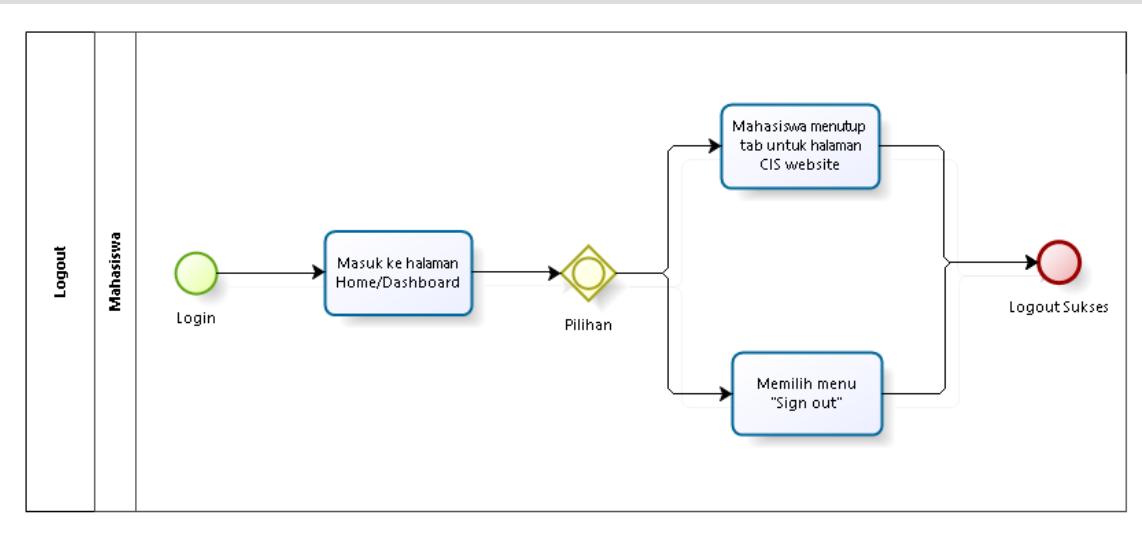


Gambar 9. BPMN *Current System* Melakukan *Login*

Pada saat ini proses *login* ke dalam *website Campus Information System* (CIS) di Institut Teknologi (IT) Del oleh mahasiswa masih dilakukan setiap kali mengakses CIS. Hal ini dikarenakan proses *logout website* dilakukan secara otomatis, sehingga ketika mahasiswa ingin mengakses CIS mahasiswa harus selalu melakukan *login* dengan memasukkan *username* dan *password* dari mahasiswa tersebut yang dimana sebelumnya *password* dan *username* tersebut telah didaftarkan oleh pihak SDI (Sumber Daya Informasi). Namun jika *username* atau *password* yang dimasukkan tidak sesuai dengan yang terdaftar maka mahasiswa tersebut tidak dapat *login* atau *login* yang dilakukan tidak *valid* dan mahasiswa tersebut harus kembali memasukkan *password* dan *username* yang benar dan yang *valid* sehingga mahasiswa tersebut sukses melakukan *login* dan dapat masuk ke halaman *home/dashboard*.

#### 2. Melakukan *Logout*

Pada Gambar 10 akan menggambarkan *current system* bisnis proses dari melakukan *logout* dari sistem.

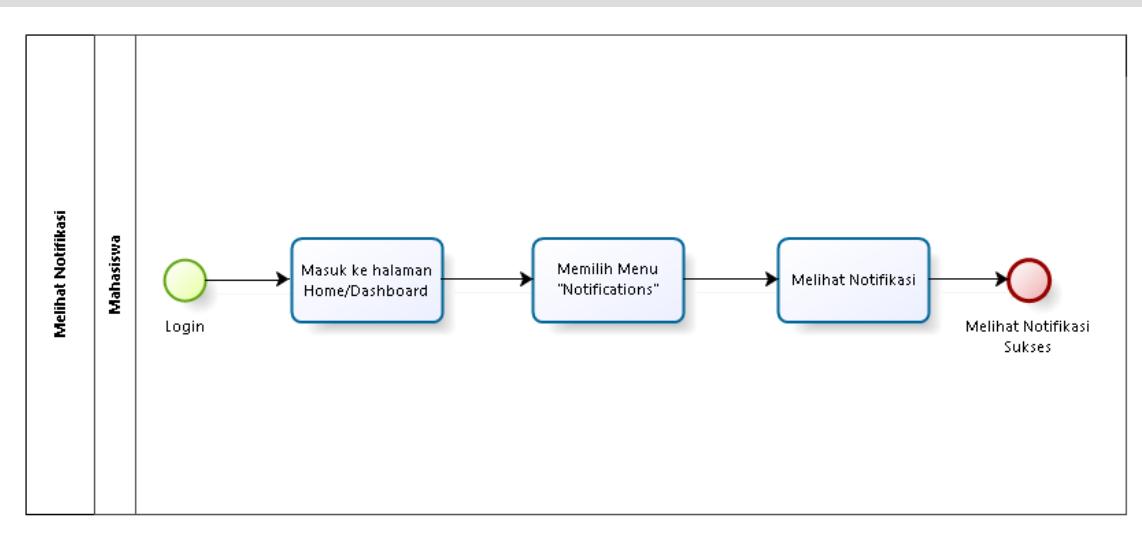


**Gambar 10. BPMN Current System Melakukan Logout**

Pada saat ini proses *logout* dari *website CIS IT Del* masih dilakukan secara otomatis. Dimana ketika mahasiswa ingin mengeluarkan akun dari sistem maka mahasiswa dapat melakukannya dengan menutup tab untuk situs *website CIS IT Del*. Proses *logout* juga dapat dilakukan dengan memilih menu *sign out*.

### 3. Melihat Notifikasi

Pada Gambar 11 akan menggambarkan *current system* bisnis proses dari melihat notifikasi.

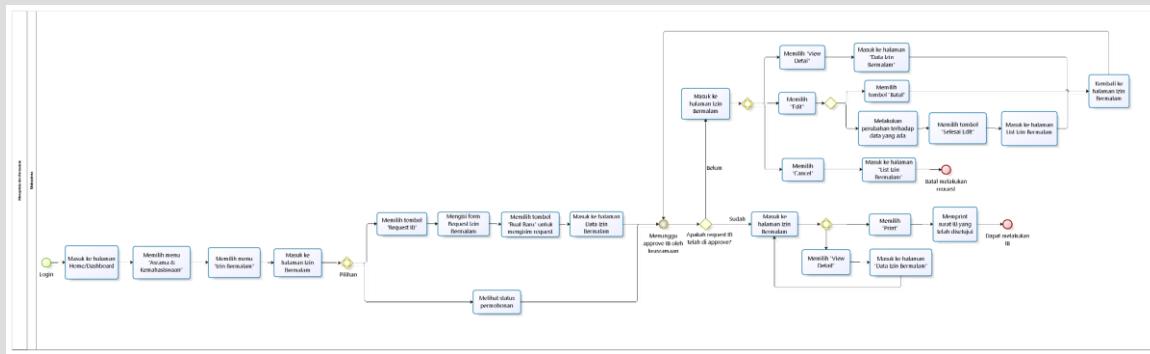


**Gambar 11. BPMN Current System Melihat Notifikasi**

Pada saat ini untuk melihat notifikasi yang ada dalam *website CIS IT Del*, mahasiswa harus melakukan *login* terlebih dahulu ke dalam *website CIS IT Del*. Selanjutnya mahasiswa memilih menu *notifications* dan hanya dapat melihat notifikasi saja.

#### 4. Mengelola Izin Bermalam (IB).

Pada Gambar 12 akan menggambarkan *current system* bisnis proses dari mengelola izin bermalam.



**Gambar 12. BPMN Mengelola Izin Bermalam (IB)**

Pada saat ini proses untuk mengelola izin bermalam oleh mahasiswa harus terlebih dahulu melakukan *login* ke dalam *website CIS IT Del*. Ketika mahasiswa telah *login* dan masuk ke halaman *home/dashboard*, mahasiswa dapat memilih menu asrama dan kemahasiswaan lalu memilih menu Izin Bermalam (IB) kemudian akan masuk ke halaman Izin Bermalam. Di halaman Izin Bermalam ini mahasiswa juga dapat melakukan *request* IB dengan memilih tombol *request* IB. Setelah itu mahasiswa harus mengisi form *request* IB dan kemudian memilih tombol “Buat Baru” untuk mengirim *request* kemudian mahasiswa akan di bawa ke halaman data izin bermalam untuk melihat data yang telah dibuat tadi. Mahasiswa akan menunggu informasi *approve* dari keasramaan. Dan untuk melihat status permohonan yang dibuat terhadap *request* tersebut dapat kembali di lihat di halaman Izin Bermalam.

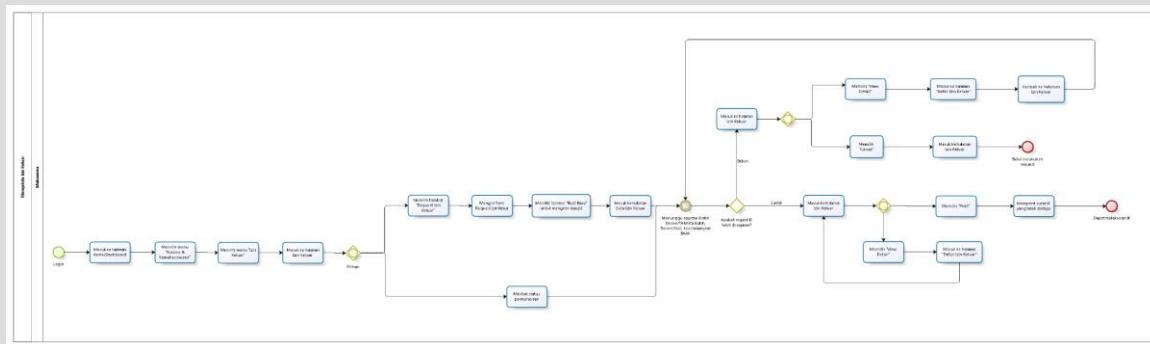
Ketika menunggu *approve* ini mahasiswa dapat melakukan beberapa hal terhadap *request* yang sebelumnya dilakukan dengan kondisi bahwa *request* tersebut masih menunggu *approve* dan belum mendapat respon dari pihak keasramaan. Pada kondisi ini mahasiswa dapat masuk ke halaman izin bermalam kemudian dapat memilih *view detail* untuk melihat *detail* dari data izin bermalam mahasiswa tersebut, atau mahasiswa dapat melakukan pengeditan terhadap data yang telah di *request* sebelumnya dengan cara memilih *edit*. Kemudian mahasiswa akan dapat melakukan perubahan terhadap data yang telah ada dan memilih tombol selesai edit untuk menyimpan perubahan data tersebut. Perubahan data yang dilakukan akan ditampilkan pada halaman setelahnya yaitu halaman list izin bermalam, atau jika mahasiswa tersebut sudah sempat memilih edit akan tetapi

tidak jadi melakukan perubahan apapun atau tidak jadi melakukan perubahan mahasiswa dapat memilih tombol batal untuk kondisi tersebut.

Mahasiswa juga dapat membatalkan *request* yang telah dilakukan dengan memilih tombol *cancel*, kemudian mahasiswa akan di bawa ke halaman list izin bermalam dan *request* yang mereka lakukan sebelumnya telah dibatalkan. Jika permohonan terhadap *request* yang dilakukan oleh mahasiswa tersebut telah di *approve* oleh pihak keasramaan maka mahasiswa tidak akan dapat melakukan edit atau pun *cancel* terhadap data *request* yang di buat. Dan setelah di *approve* oleh keasramaan mahasiswa tersebut dapat mem-*print* surat IB. Informasi persetujuan permohonan dapat dilihat oleh mahasiswa jika telah *login* ke dalam sistem dan memilih menu IB. Oleh karena hal tersebut banyak mahasiswa yang sering lupa atau tidak mengetahui informasi permohonan karena tidak adanya berupa notifikasi pemberitahuan.

## 5. Mengelola Izin Keluar

Pada Gambar 13 akan menggambarkan *current system* bisnis proses dari mengelola *request* izin keluar.



Gambar 13. BPMN *Current System* Mengelola *Request* IK

Pada saat ini proses untuk mengelola izin keluar oleh mahasiswa harus terlebih dahulu melakukan *login* ke dalam *website CIS IT Del*. Ketika mahasiswa telah *login* dan masuk ke halaman *home/dashboard*, mahasiswa dapat memilih menu asrama dan kemahasiswaan lalu memilih menu Izin Keluar (IK) kemudian akan masuk ke halaman Izin Keluar. Di halaman Izin Keluar ini mahasiswa juga dapat melakukan *request* IK dengan memilih tombol *request* Izin Keluar. Setelah itu mahasiswa harus mengisi form *request* IK dan kemudian memilih tombol “Buat Baru” untuk mengirim *request* kemudian mahasiswa akan di bawa ke halaman data izin keluar untuk melihat data yang telah dibuat tadi. Mahasiswa akan menunggu informasi

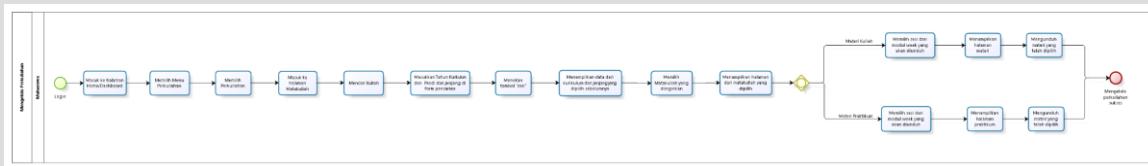
*approve* dari Dosen/TA Matakuliah, Dosen Wali, Keasramaan, dan BAAK. Dan untuk melihat status permohonan yang dibuat terhadap *request* tersebut dapat kembali di lihat di halaman Izin Bermalam.

Ketika menunggu *approve* ini mahasiswa dapat melakukan beberapa hal terhadap *request* yang sebelumnya dilakukan dengan kondisi bahwa *request* tersebut masih menunggu *approve* dan belum mendapat respon dari Dosen/TA Matakuliah, Dosen Wali, Keasramaan, dan BAAK. Pada kondisi ini mahasiswa dapat masuk ke halaman izin keluar kemudian dapat memilih *view detail* untuk melihat *detail* dari data izin keluar mahasiswa tersebut, atau mahasiswa dapat melakukan pembatalan *request* yang telah dilakukan dengan memilih tombol *cancel*, kemudian mahasiswa akan di bawa ke halaman izin keluar dan *request* yang mereka lakukan sebelumnya telah dibatalkan.

Jika permohonan terhadap *request* yang dilakukan oleh mahasiswa tersebut telah di *approve* oleh Dosen/TA Matakuliah, Dosen Wali, Keasramaan, dan BAAK maka mahasiswa tidak akan dapat melakukan *cancel* atau pembatalan terhadap data *request* yang di buat. Dan setelah di *approve* oleh Dosen/TA Matakuliah, Dosen Wali, Keasramaan, dan BAAK mahasiswa tersebut dapat mem-*print* surat IK. Informasi persetujuan permohonan dapat dilihat oleh mahasiswa jika telah *login* ke dalam sistem dan memilih menu IK. Oleh karena hal tersebut banyak mahasiswa yang sering lupa atau tidak mengetahui informasi permohonan karena tidak adanya berupa notifikasi pemberitahuan.

## 6. Mengelola Perkuliahan

Pada Gambar 14 akan menggambarkan *current system* bisnis proses dari mengelola perkuliahan.



Gambar 14. BPMN *Current System* Mengelola Perkuliahan

Pada proses melihat modul perkuliahan mahasiswa harus melakukan *login* terlebih dahulu ke dalam CIS IT Del. Setelah berhasil melakukan *login*, mahasiswa akan masuk ke halaman *home/dashboard* kemudian memilih menu perkuliahan lalu perkuliahan. Lalu mahasiswa akan masuk ke halaman Matakuliah pada halaman ini mahasiswa dapat mencari kuliah dengan cara memasukkan tahun kurikulum dan

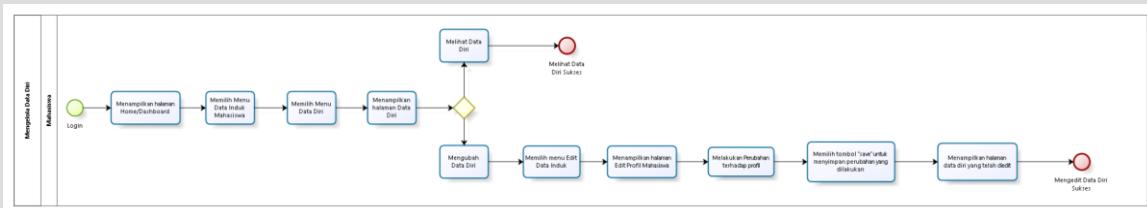
prodi dan jenjang di form pencarian lalu memilih tombol “cari”. Setelah itu CIS IT Del akan menampilkan halaman modul mata kuliah sesuai dengan tahun kurikulum dan prodi dan jenjang yang telah dipilih sebelumnya.

Kemudian mahasiswa dapat memilih matakuliah yang diinginkan sesuai dengan yang dicari sebelumnya. Jika mahasiswa telah memilih matakuliah yang ingin dilihat maka sistem akan menampilkan halaman dari matakuliah yang dipilih tersebut. Pada halaman matakuliah ini terdapat dua materi yaitu materi kuliah dan materi praktikum. Pada kedua materi ini mahasiswa dapat mengunduh modul dari setiap *week* dan sesi yang ada dimana mahasiswa tersebut harus menentukan terlebih dahulu *week* dan sesi dari modul yang akan di unduh kemudian jika sudah memilih maka sistem akan menampilkan halaman dari materi yang dipilih tersebut. Pada halaman ini mahasiswa dapat mengunduh materi yang terlampir sesuai dengan yang dipilih mereka sebelumnya.

Dalam hal ini, mahasiswa mengalami masalah dimana ketika ingin mengetahui modul baru yang diunggah oleh dosen pengampu, mahasiswa harus terlebih dahulu masuk ke halaman perkuliahan. Sehingga banyak mahasiswa yang tidak membuka halaman perkuliahan tidak mengetahui dan terlambat mengetahui modul baru yang telah diunggah oleh dosen pengampu.

## 7. Mengelola Data Diri

Pada Gambar 15 akan menggambarkan *current system* bisnis proses dari mengelola data diri.



Gambar 15. BPMN *Currents System* Mengelola Data Diri

Pada proses mengelola data diri, mahasiswa harus terlebih dahulu melakukan *login* kedalam *website CIS IT Del*. Mahasiswa akan masuk ke halaman *home/dashboard* kemudian mahasiswa memilih menu data induk mahasiswa dan memilih menu data diri lalu sistem akan menampilkan halaman data diri dan mahasiswa dapat memilih data yang ingin dilihat. Dalam halaman data diri terdapat enam menu yaitu: data diri akademis, data pribadi, data orangtua, data asal sekolah, data asrama dan

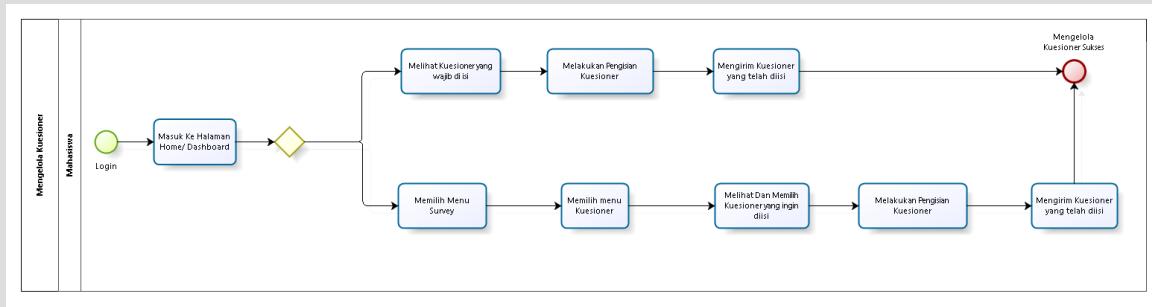
perilaku serta data hasil studi.Untuk semua data ini mahasiswa tersebut dapat melihat setiap detailnya sebagai berikut:

- a. Data akademis mahasiswa dapat melihat status akhir, angkatan, *username*, email akademik, program studi, kelas, wali mahasiswa dan jalur usm.
- b. Data pribadi mahasiswa dapat melihat foto, nama lengkap, tempat/tanggal lahir, jenis kelamin, alamat, agama, golongan darah, asal kabupaten, kode pos, email pribadi, no telepon, no HP 1 dan 2 serta Anak ke / dari.
- c. Data Orang tua mahasiswa dapat melihat nama ayah, nama ibu, alamat orang tua, pekerjaan ayah, keterangan pekerjaan ayah, rentang penghasilan ayah, penghasilan pasti ayah, No.HP ayah, pekerjaan ibu, keterangan pekerjaan ibu, rentang penghasilan ibu, penghasilan pasti ibu, No.HP ibu, total penghasilan orang tua, jumlah tanggungan, nama wali, alamat wali, pekerjaan wali, keterangan pekerjaan ibu, rentang penghasilan wali, penghasilan pasti wali, no.telepon / no.HP wali.
- d. Data Asal Sekolah mahasiswa dapat melihat nomor ijazah SMA, nama SMA, alamat SMA, kabupaten SMA, telepon SMA, dan kode pos SMA.
- e. Data Keasramaan dan perilaku mahasiswa dapat melihat nama asrama, pengurus asrama, kamar yang ditempati, teman satu kamar dan daftar nilai perilaku selama menjadi mahasiswa aktif.
- f. Hasil studi mahasiswa dapat melihat Grafik Perkembangan Nilai, Nilai IP yang didapat, *History* Dari Mata Kuliah yang Sudah diambil dan sedang di ambil beserta nilai dari setiap mata kuliah dan IP Sementara untuk hitungan setiap semester.

Selain dari itu, mahasiswa juga dapat mengubah data diri mahasiswa yaitu data pribadi, data orangtua dan data asal sekolah. Mahasiswa dapat memilih data diri yang akan diubah lalu memilih menu Edit Data Induk. Kemudian sistem akan menampilkan halaman edit profil mahasiswa, setelah mahasiswa melakukan perubahan terhadap data profil maka mahasiswa dapat memilih tombol “*save*” untuk menyimpan perubahan yang dilakukan. Sistem akan menampilkan halaman data diri yang telah diubah.

## 8. Mengelola Kuesioner

Pada Gambar 16 akan menggambarkan *current system* bisnis proses dalam mengisi kuesioner.

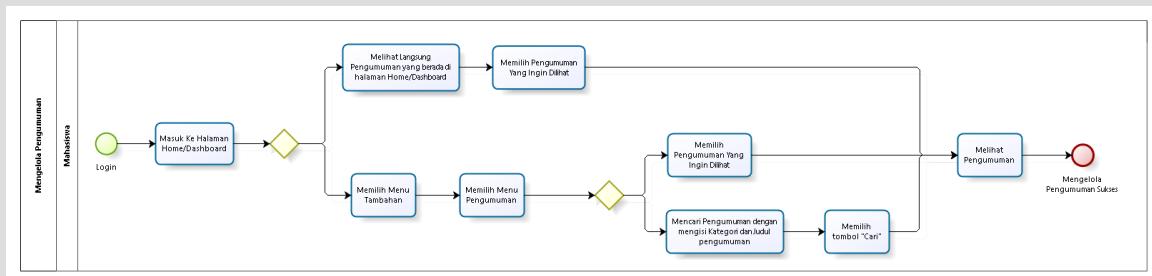


**Gambar 16. BPMN Current system Mengelola Kuesioner**

Pada proses mengelola kuesioner oleh mahasiswa dapat dilakukan dengan *login* terlebih dahulu ke dalam *website CIS IT Del* dan masuk ke halaman *home/dashboard*. Mengelola kuesioner oleh mahasiswa dapat dilakukan dengan dua cara yaitu melalui menu *home* dan menu *survey*. Kuesioner yang terdapat dalam menu *home/dashboard* adalah kuesioner wajib yang harus diisi oleh mahasiswa sebelum mengakses halaman lainnya. Sementara untuk mengisi kuesioner melalui menu *survey* adalah kuesioner yang tidak wajib. Mahasiswa dapat memilih menu *survey* lalu memilih menu kuesioner maka website akan menampilkan daftar-daftar kuesioner. Kemudian mahasiswa melihat dan memilih salah satu kuesioner yang akan diisi. Ketika mahasiswa telah memilih kuesioner, *website* akan menampilkan *form* kuesioner. Setelah mahasiswa mengisi kuesioner, mahasiswa dapat memilih tombol “Kirim” untuk menyimpan tanggapan mahasiswa.

#### 9. Mengelola Pengumuman

Pada Gambar 17 akan menggambarkan *current system* bisnis proses dari mengelola pengumuman.



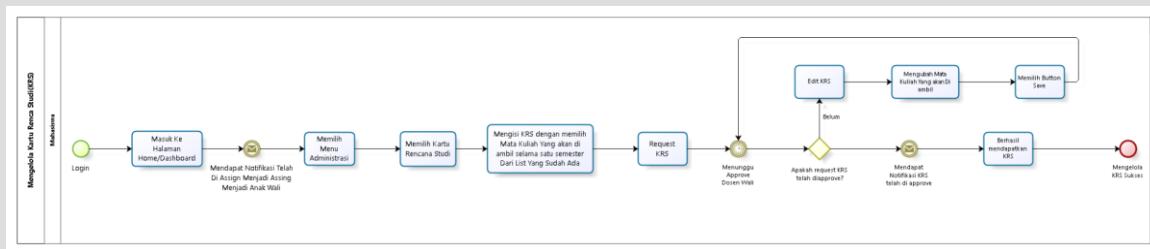
**Gambar 17. BPMN Current System Mengelola Pengumuman**

Pada proses mengelola pengumuman mahasiswa terlebih dahulu *login* ke dalam *website CIS IT Del* dan masuk ke halaman *home/dashboard*. Mengelola pengumuman dapat dilakukan oleh mahasiswa setelah masuk ke halaman *home/dashboard* atau menu tambahan. Di halaman menu *home/dashboard*, mahasiswa dapat langsung memilih pengumuman yang akan dilihat. Selain itu,

untuk mahasiswa yang akan melihat pengumuman melalui menu tambahan, mahasiswa dapat memilih menu pengumuman kemudian dapat langsung memilih dan melihat pengumuman yang diinginkan atau mahasiswa dapat mengisi judul pengumuman lalu memilih tombol “Cari”. Kemudian memilih pengumuman dan melihat pengumuman serta dapat juga melakukan pengunduhan modul yang terdapat pada pengumuman.

## 10. Mengelola Request Kartu Rencana Studi (KRS)

Pada Gambar 18 akan menggambarkan *current system* bisnis proses dari melakukan *request* kartu rencana studi.

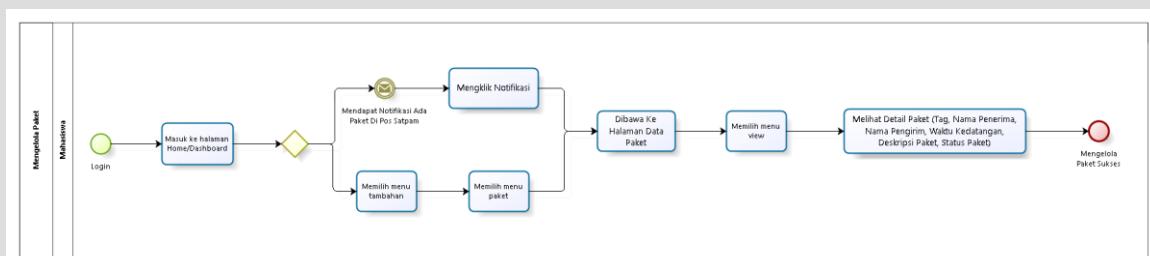


Gambar 18. BPMN *Current System* Mengelola Request KRS

Proses mengelola Kartu Rencana Studi (KRS) dapat dilakukan apabila telah *login* terlebih dahulu. Setelah masuk ke halaman sistem, mahasiswa dapat memilih menu Administrasi Akademik dan KRS. Lalu mahasiswa akan memilih nama mata kuliah untuk rencana studi satu semester kedepan. Kemudian mahasiswa akan memilih tombol “Kirim” untuk melakukan *request* KRS. Dalam halaman KRS, mahasiswa juga dapat melakukan edit KRS sebelum disetujui oleh dosen wali. Mahasiswa dapat melakukan edit KRS di menu “Perubahan KRS”. Di halaman perubahan KRS mahasiswa akan memilih kembali nama-nama rencana studi dan memilih tombol “Kirim” kembali.

## 11. Melihat Informasi Paket

Pada Gambar 19 akan menggambarkan *current system* bisnis proses melihat informasi paket.

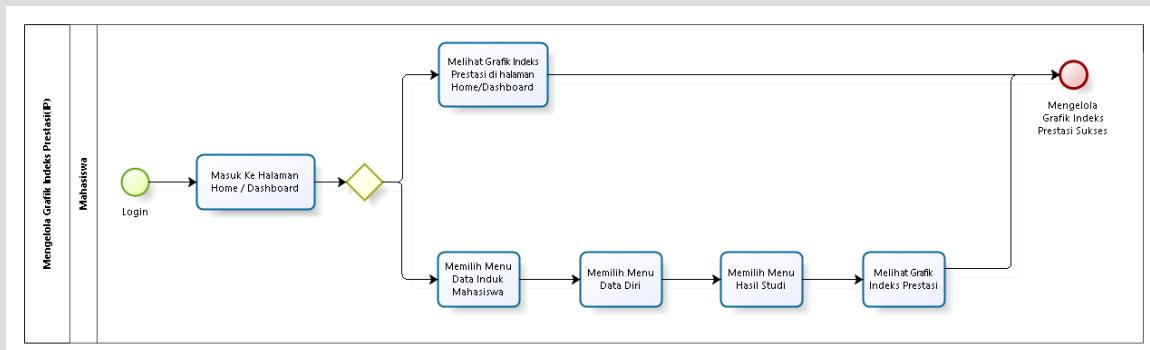


Gambar 19. BPMN *Current System* Melihat Informasi Paket

Pada proses melihat informasi paket, mahasiswa harus terlebih dahulu melakukan *login* ke dalam *website CIS IT Del*. Mahasiswa dapat memilih menu tambahan dan menu paket. Di halaman menu paket, mahasiswa dapat melihat informasi-informasi paket yang berada di pos satpam IT Del. Namun dengan sistem ini, mahasiswa sering terlambat mengetahui informasi paket dan bahkan tidak mengetahui paket atas nama mahasiswa tersebut. Hal ini dikarenakan tidak ada notifikasi pemberitahuan paket apabila belum melakukan *login* terlebih dahulu ke dalam website CIS IT Del. Oleh karena hal tersebut banyak paket yang bertumpuk di pos satpam IT Del.

## 12. Melihat Grafik Indeks Prestasi (IP)

Pada Gambar 20 akan menggambarkan *current system* bisnis proses melihat grafik Indeks Prestasi (IP).

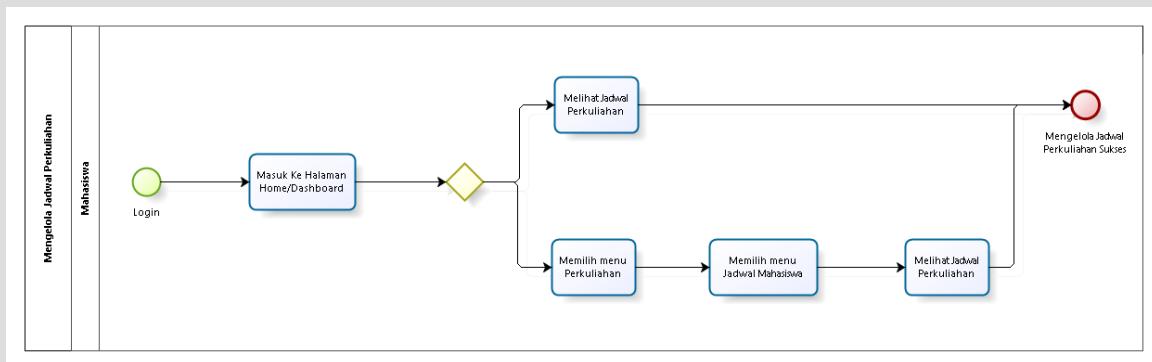


**Gambar 20. BPMN Current System Melihat Grafik IP**

Proses melihat grafik Indeks Prestasi (IP) oleh mahasiswa, harus terlebih dahulu melakukan *login* ke dalam website CIS IT Del. Di halaman *home* mahasiswa dapat melihat grafik IP mahasiswa tersebut. Melihat grafik IP juga dapat dilakukan dengan memilih menu data diri. Dalam halaman data diri mahasiswa akan memilih menu hasil studi. Maka di halaman hasil studi mahasiswa dapat melihat grafik IP.

## 13. Melihat Jadwal Perkuliahan

Pada Gambar 21 akan menggambarkan *current system* bisnis proses dari melihat jadwal perkuliahan.



**Gambar 21. BPMN Current System Melihat Jadwal Perkuliahan**

Proses melihat jadwal perkuliahan oleh mahasiswa, dapat dilakukan dengan *login* terlebih dahulu ke dalam *website CIS IT Del*. Melihat jadwal perkuliahan dapat dilakukan dengan melihatnya di halaman *home*. Melihat jadwal perkuliahan juga dapat dilakukan dengan memilih menu perkuliahan dan menu jadwal mahasiswa. Di halaman jadwal mahasiswa akan ditampilkan jadwal perkuliahan untuk satu hari.

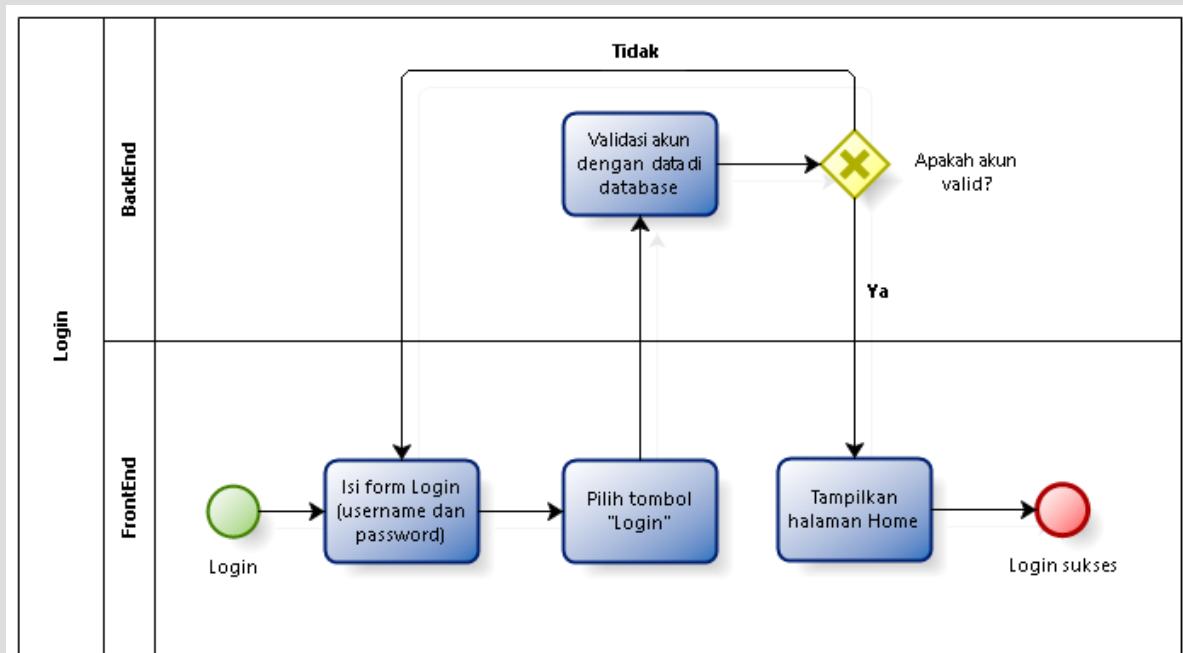
### 3.11 Target Sistem

Sub bab berikut akan menjelaskan tentang target sistem pada aplikasi yang akan dibangun dalam bentuk *Business Process Model and Notation* (BPMN). BPMN ini akan terdiri dari proses setiap bisnis baik dari segi FrontEnd (FE) maupun BackEnd (BE). Target sistem adalah sistem yang akan dibangun berdasarkan permasalahan yang ditemukan. Analisis permasalahan dilakukan dengan melakukan diskusi. Adapun target sistem pada aplikasi yang akan dibangun dapat dilihat pada sub bab berikut.

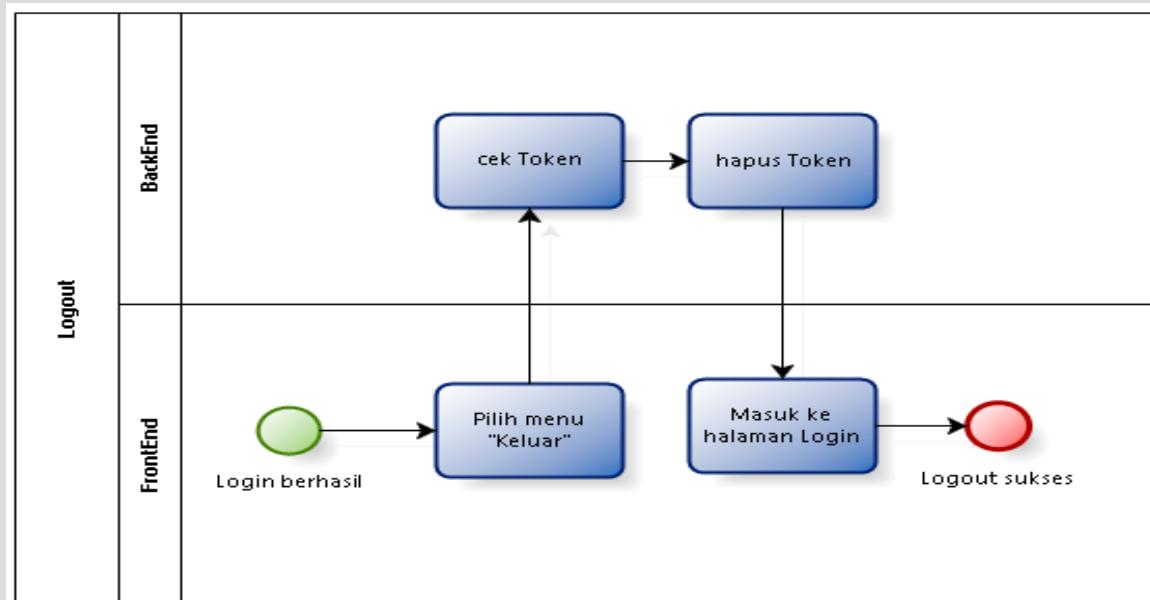
#### 3.11.1 Business Process Otentikasi

Otentikasi terdiri dari 2 jenis yaitu *login* dan *logout*. *Login* dapat dilakukan dengan membuka aplikasi terlebih dahulu, lalu aplikasi akan menampilkan *form login*. Kemudian pada FE mahasiswa mengisi *form login* dengan akun yang telah didaftarkan oleh pihak SDI (Sumber Daya Informasi) sebelumnya yaitu *username* dan *password*. Jika sudah memasukkan *username* dan *password*, selanjutnya mahasiswa memilih tombol “*Login*”. Ketika mahasiswa memilih tombol *login*, BE akan melakukan validasi akun terhadap database. Apabila akun yang dimasukkan oleh mahasiswa *valid*, maka aplikasi akan menampilkan halaman *home*. Namun apabila *username*, *password* atau *username* dan *password* yang dimasukkan mahasiswa tidak *valid* maka aplikasi akan tetap berada di halaman *login*. Sementara untuk *logout* dari aplikasi dapat dilakukan dengan memilih menu

“Keluar” maka BE akan melakukan pengecekan token dan penghapusan token, kemudian mengarahkan atau menampilkan halaman *login*. *Logout* pada target sistem adalah *logout manual*. Dimana mahasiswa tidak perlu melakukan *login* kembali atau memasukkan akun secara berulang setiap kali mengakses CIS selama mahasiswa belum memilih menu “Keluar”. *Business Process* dari autentikasi mahasiswa dapat dilihat pada Gambar 22 dan Gambar 23 berikut.



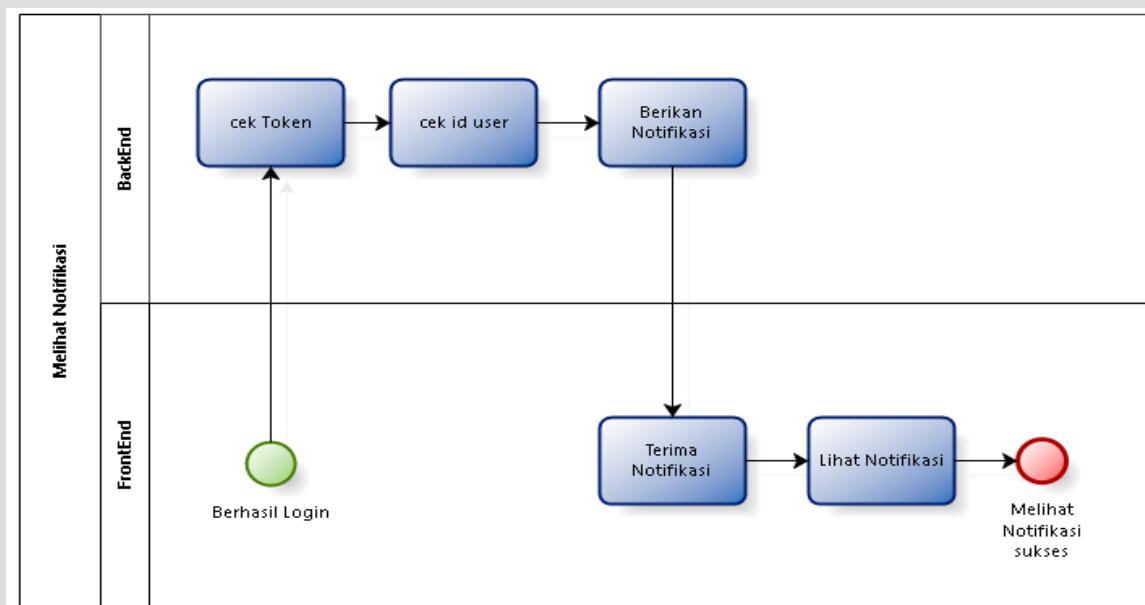
Gambar 22. Target Sistem Melakukan *Login*



Gambar 23. Target Sistem Melakukan *Logout*

### 3.11.2 Business Process Melihat Notifikasi

Pada target sistem mahasiswa dapat melihat notifikasi atau mendapat notifikasi yang masuk ke *mobile phone* mahasiswa, walau mahasiswa sedang tidak membuka aplikasi CIS *mobile*. Ketika mahasiswa telah melakukan *login*, BE akan melakukan cek token dan cek *id* mahasiswa. Setelah itu BE akan mengirimkan notifikasi ke FE. FE akan menerima Notifikasi lalu mahasiswa akan melihat notifikasi yang diterima. *Buiness Process* melihat notifikasi dapat dilihat pada Gambar 24 berikut.



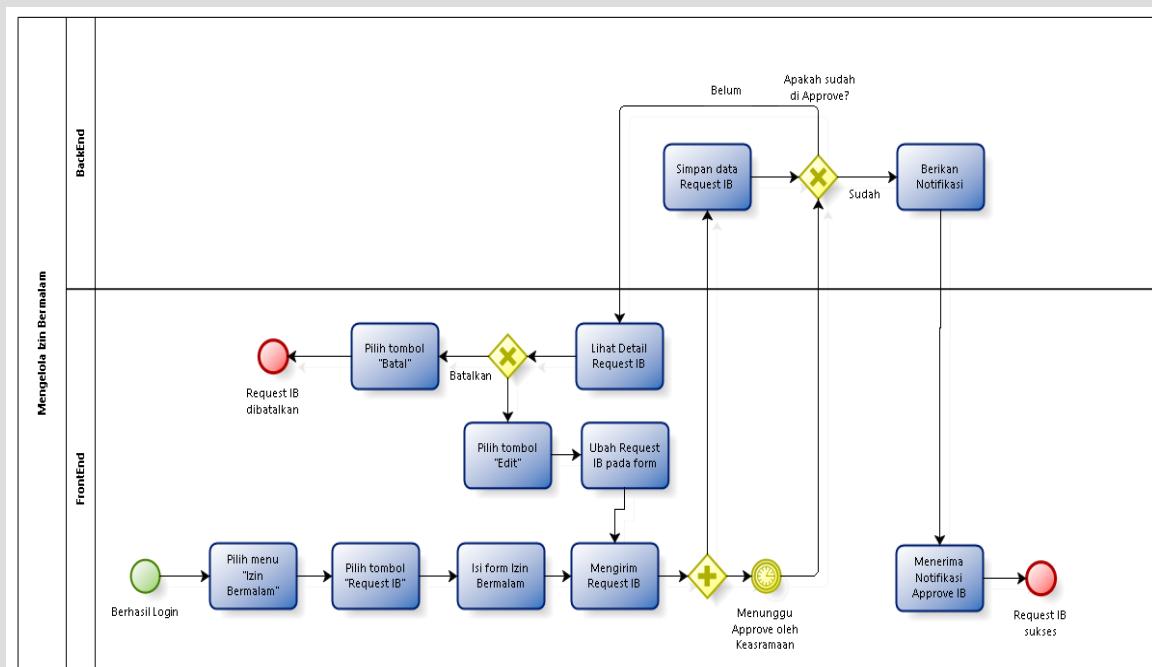
Gambar 24. Target Sistem Melihat Notifikasi

### 3.11.3 Business Process Mengelola Izin Bermalam (IB)

Pada target sistem mengelola izin bermalam mahasiswa dapat melakukan *request* IB, edit IB dan *cancel* IB dengan kondisi tertentu. *Request* IB dapat dilakukan oleh mahasiswa di FE maksimal dua hari sebelum tanggal keberangkatan IB. *Request* IB dapat dilakukan dengan memilih menu “Izin Bermalam”, lalu aplikasi akan menampilkan halaman izin bermalam. Mahasiswa memilih tombol “*Request* IB” lalu mengisi *form* berupa rencana berangkat, rencana kembali, keperluan IB dan tempat tujuan. Setelah *form* diisi mahasiswa memilih tombol “Buat Baru” untuk mengirim *request* IB. Kemudian aplikasi akan menambahkan *request* IB tersebut di halaman izin bermalam dengan status menunggu *approve* oleh keasramaan. Apabila keasramaan telah meng-*approve* *request* IB maka mahasiswa akan menerima notifikasi di mobile mahasiswa.

Selain melakukan *request* IB, mahasiswa juga dapat mengubah atau membatalkan *request* IB apabila status *request* IB masih dengan status menunggu atau belum *diapprove*.

Mahasiswa dapat memilih dan melihat detail *request IB* yang belum di *approve* lalu mengubah atau membatalkan *request IB*. Apabila mahasiswa ingin mengubah *request IB* mahasiswa dapat memilih tombol “Edit” dan mengubah data pada *form*. Setelah data diubah mahasiswa akan memilih tombol “Edit Selesai” untuk menyimpan data yang telah diubah. Aplikasi akan menampilkan kembali *request IB* yang telah diubah dan menunggu *approve*. Untuk membatalkan *request IB* mahasiswa dapat memilih tombol “Batal” dan status *request IB* akan langsung berubah menjadi “dibatalkan”. *Business Process* dari mengelola IB dapat dilihat pada Gambar 25 berikut



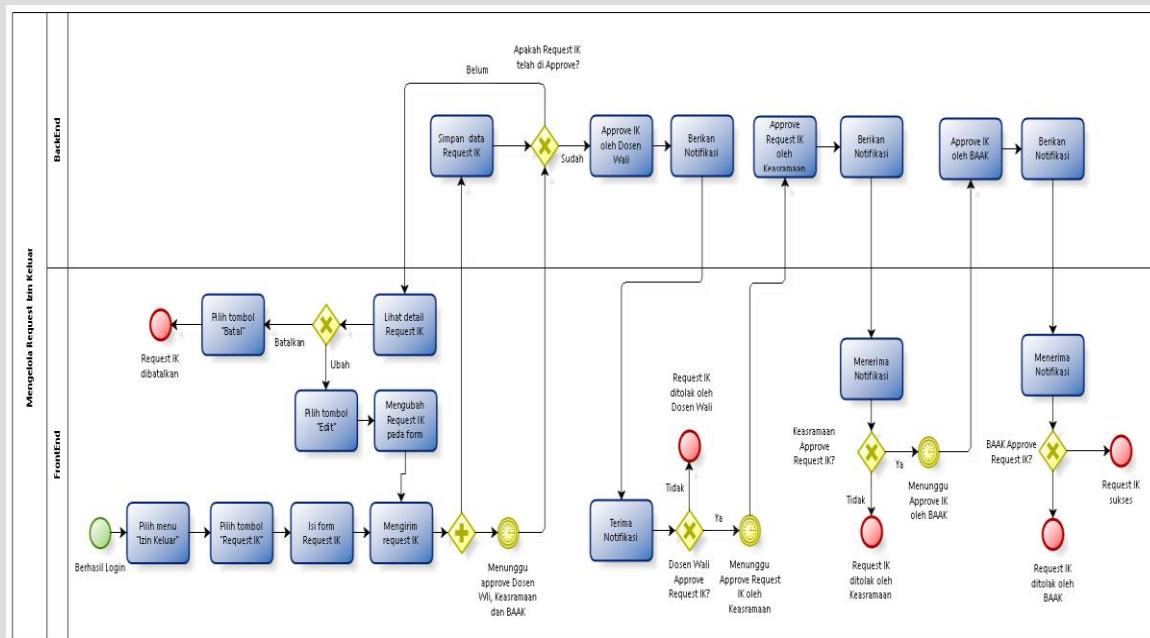
Gambar 25. Target Sistem Mengelola IB

### 3.11.4 Business Process Mengelola Izin Keluar (IK)

Pada target sistem mengelola izin keluar mahasiswa dapat melakukan *request IK*, edit IK dan *cancel IK* dengan kondisi tertentu. *Request IK* dilakukan oleh mahasiswa dengan memilih menu “Izin Keluar”, lalu aplikasi akan menampilkan halaman IK. Dalam halaman IK mahasiswa akan memilih tombol “*Request IK*”, lalu aplikasi akan menampilkan *form IK*. Mahasiswa dapat mengisi *form IK* berupa rencana berangkat, rencana kembali dan keperluan IK. Apabila *form* telah diisi maka mahasiswa dapat memilih tombol “Buat Baru” untuk mengirim *request IK*. Kemudian BE akan menyimpan data *request IK*. Setelah itu *request IK* akan ditampilkan aplikasi di halaman IK, mahasiswa dapat menunggu *approve* oleh dosen wali, keasramaan dan BAAK. *Approve request IK* memiliki beberapa kondisi, dimana *approve* harus secara berurutan dimulai dari dosen

wali, lalu keasramaan dan BAAK. Apabila dosen wali telah meng-approve *request IK*, maka mahasiswa dapat menunggu *approve* selanjutnya yaitu keasramaan dan BAAK. Namun apabila dosen wali menolak *request IK*, maka otomatis *request IK* tidak dapat dilanjutkan atau dengan kata lain status dari *request IK* akan berubah menjadi “ditolak”. Begitu juga dengan seterusnya, apabila keasramaan menolak *request IK*, maka mahasiswa tidak perlu lagi menunggu *approve* oleh BAAK. Dengan kesimpulan apabila salah satu dari user tersebut menolak *request IK* maka *request IK* tidak dapat dilanjutkan atau ditolak. *Approve request IK* akan diterima oleh mahasiswa berupa notifikasi yang muncul di *mobile* mahasiswa.

Selain itu mahasiswa juga dapat Edit dan *cancel request IK*, yang dimana status dari *request IK* masih menunggu dari ketiga *user* tersebut. Mahasiswa dapat memilih dan melihat detail *request IK* yang masih dengan status menunggu, lalu memilih tombol “Edit”. Kemudian mahasiswa akan mengubah data pada *form* dan memilih tombol “Edit Selesai” untuk mengirim *request* kembali dan menunggu *approve*. Untuk membatalkan *request IB*, mahasiswa dapat memilih tombol “batal” maka *request IB* akan otomatis dibatalkan. *Business Process* dari mengelola *IK* dapat dilihat pada Gambar 26 berikut

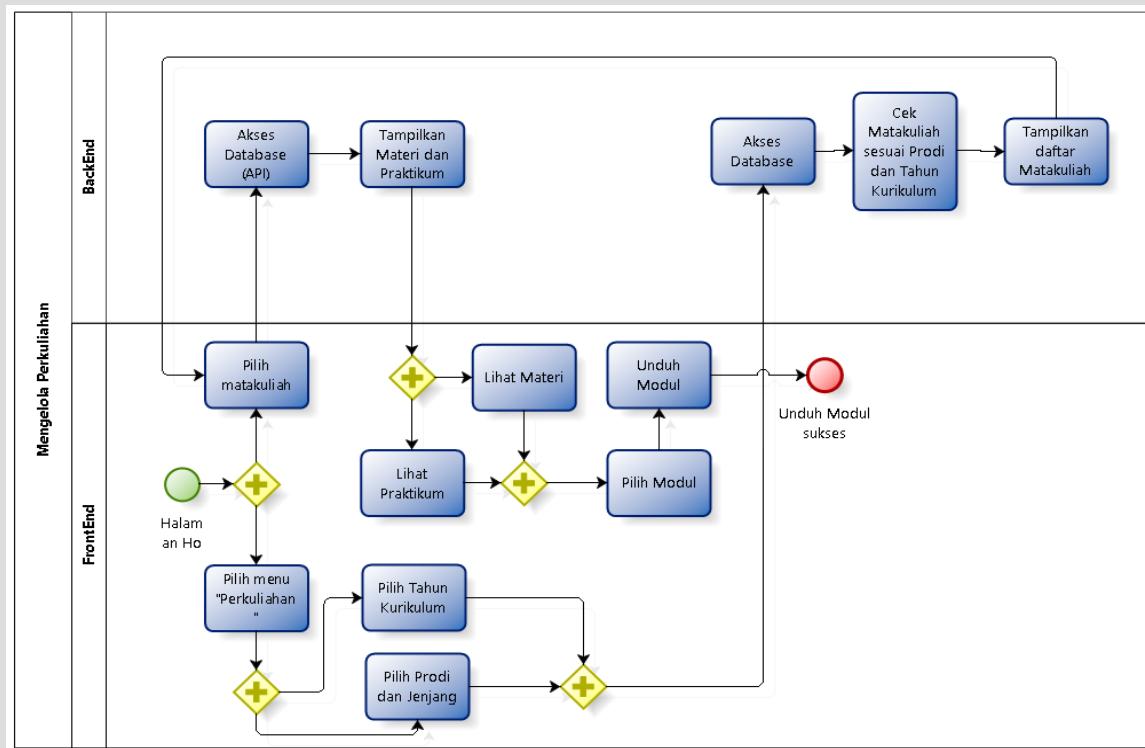


Gambar 26. Target Sistem Mengelola IK

### 3.11.5 Business Process Mengelola Perkuliahan

Pada target sistem mengelola perkuliahan, mahasiswa dapat melihat mata kuliah dan mengunduh modul perkuliahan. Mahasiswa dapat memilih menu perkuliahan, lalu aplikasi akan menampilkan *form* pencarian berupa tahun kurikulum dan prodi dan jenjang. Apabila

mahasiswa telah memilih tahun kurikulum, prodi dan jenjang, BE akan melakukan cek data matakuliah. Lalu FE akan menampilkan daftar-daftar matakuliah. Mahasiswa dapat memilih matakuliah materi atau praktikum. Lalu FE akan menampilkan modul, selanjutnya mahasiswa memilih modul yang akan diunduh. Untuk mengunduh modul perkuliahan, mahasiswa dapat meng-klik modul maka otomatis modul akan terunduh. *Business Process* dari mengelola perkuliahan dapat dilihat pada Gambar 27 berikut.



Gambar 27. Target Sistem Mengelola Perkuliahan

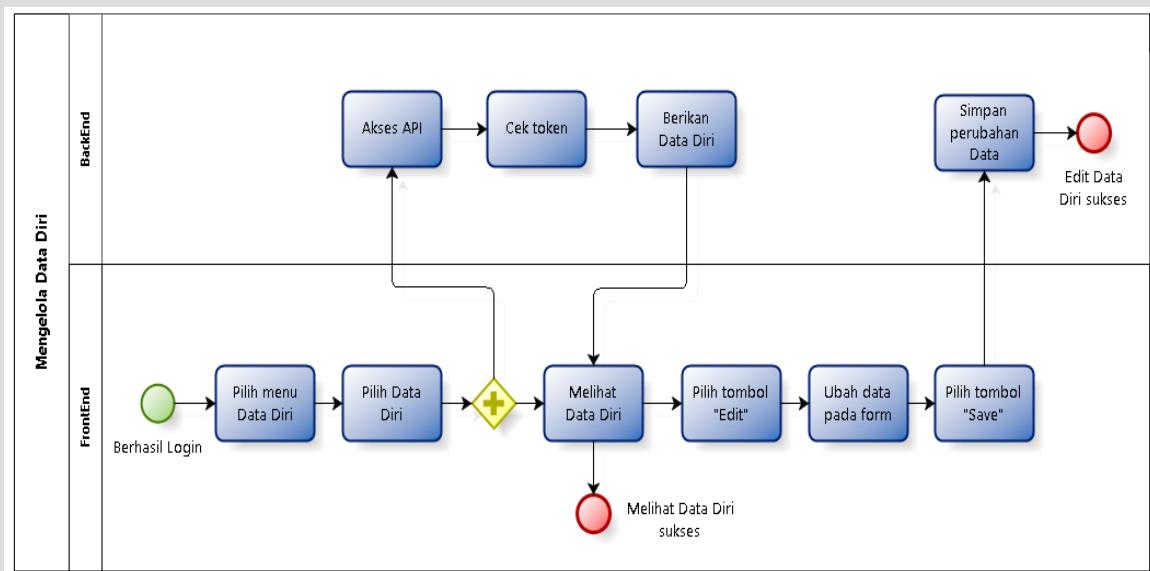
### 3.11.6 Business Process Mengelola Data Diri

Pada target sistem mahasiswa dapat mengelola data diri seperti melihat data diri dan mengubah data diri. Pada saat mahasiswa melihat data diri, mahasiswa dapat memilih menu “Data Diri” lalu BE akan melakukan cek token yang sebelumnya telah disimpan oleh FE. Kemudian mahasiswa dapat memilih data diri yang akan dilihat oleh mahasiswa. Data diri terdiri dari data akademis, data pribadi, data orang tua, data asal sekolah, keasramaan, nilai perilaku dan hasil studi. Untuk semua data ini mahasiswa tersebut dapat melihat setiap detailnya sebagai berikut:

- Data akademis mahasiswa dapat melihat status akhir angkatan, *username*, email akademik, program studi, kelas, wali mahasiswa dan jalur usm.

- h. Data pribadi mahasiswa dapat melihat foto, nama lengkap, tempat/tanggal lahir, jenis kelamin, alamat, agama, golongan darah, asal kabupaten, kode pos, email pribadi, no telepon, no HP 1 dan 2 serta Anak ke / dari.
- i. Data Orang tua mahasiswa dapat melihat nama ayah, ibu, alamat orang tua, pekerjaan ayah, ibu, keterangan pekerjaan ayah, ibu, rentang penghasilan ayah, ibu, penghasilan pasti ayah, ibu, no.HP ayah, ibu, total penghasilan orang tua, jumlah tanggungan, nama wali, alamat wali, pekerjaan wali, rentang penghasilan wali, penghasilan pasti wali, no.telepon / no.HP wali.
- j. Data Asal Sekolah mahasiswa dapat melihat nomor ijazah SMA, nama SMA, alamat SMA, kabupaten SMA, telepon SMA, dan kode pos SMA.
- k. Data Keasramaan dan perilaku mahasiswa dapat melihat nama asrama, pengurus asrama, kamar yang ditempati, teman satu kamar dan daftar nilai perilaku selama menjadi mahasiswa aktif.
- l. Hasil studi mahasiswa dapat melihat Grafik Perkembangan Nilai, Nilai IP yang didapat, *History* Dari Mata Kuliah yang Sudah diambil dan sedang di ambil beserta nilai dari setiap mata kuliah dan IP Sementara untuk hitungan setiap semester.

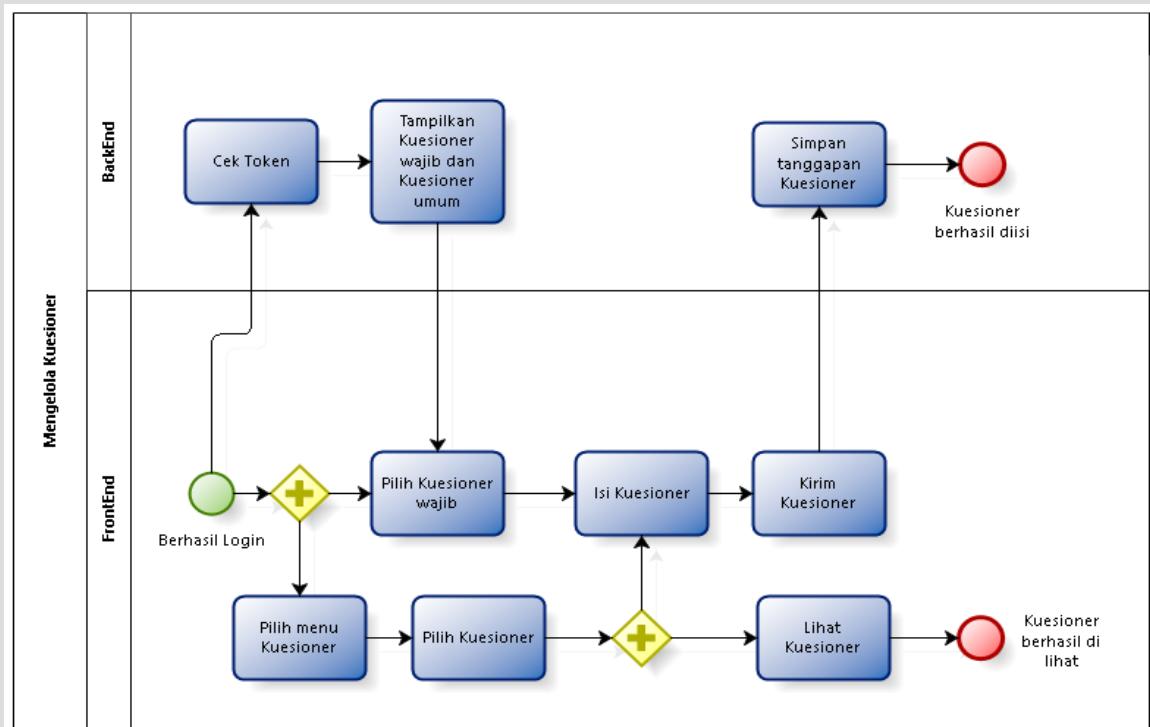
Selain dari itu, mahasiswa juga dapat mengubah data diri mahasiswa seperti data pribadi, data orangtua dan data asal sekolah. Mahasiswa dapat memilih data diri yang akan diubah lalu memilih tombol “Edit”. Kemudian FE akan menampilkan form edit data, setelah mahasiswa mengubah data maka mahasiswa memilih tombol “Save”, lalu BE akan menyimpan perubahan data. FE akan menampilkan data yang telah diubah di dalam detail data diri. *Business Process* dari mengelola data diri dapat dilihat pada Gambar 28 berikut



Gambar 28. Target Sistem Mengelola Data Diri

### 3.11.7 Business Process Mengelola Kuesioner

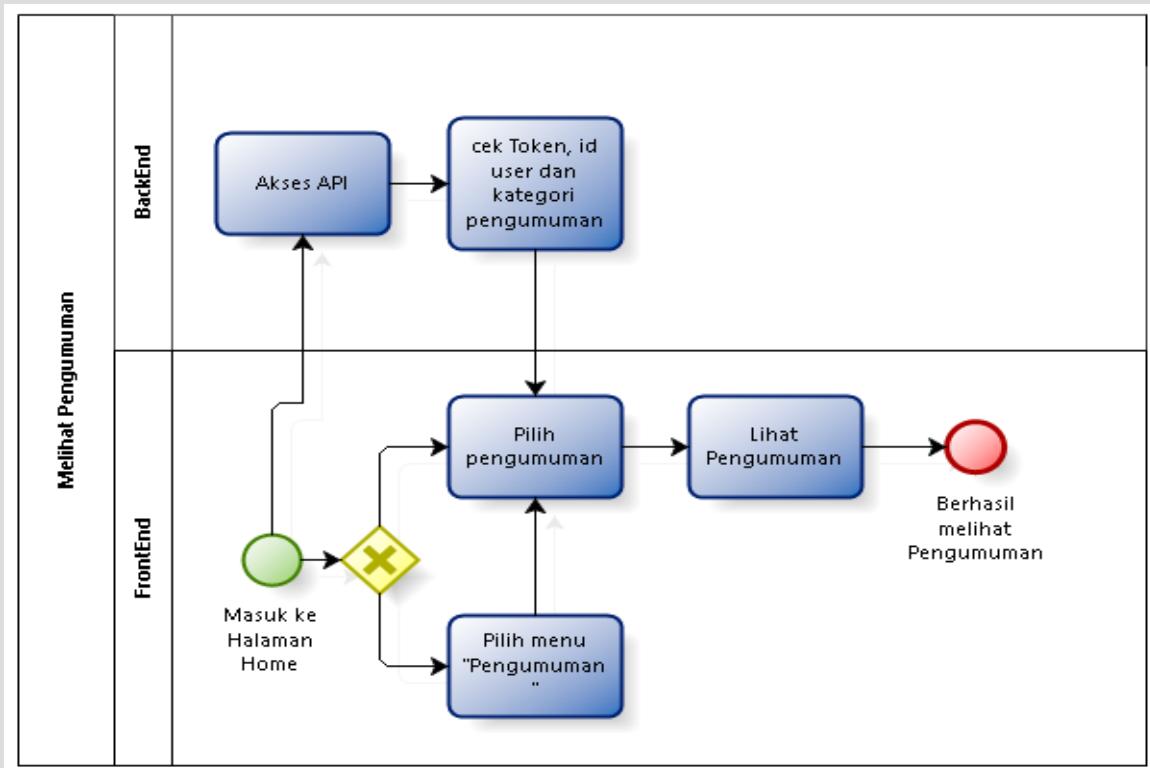
Pada target sistem mengelola kuesioner terdiri dari melihat dan mengisi kuesioner oleh mahasiswa. Mahasiswa dapat melihat dan mengisi kuesioner pada halaman *home*, yang dimana kuesioner tersebut termasuk dalam kategori wajib untuk diisi oleh mahasiswa. Selain itu, mahasiswa juga dapat melihat dan mengisi kuesioner dari menu “Kuesioner”. Ketika mahasiswa memilih menu “Kuesioner”, FE akan menampilkan daftar-daftar kuesioner. Lalu mahasiswa dapat melihat daftar-daftar atau kuesioner. Mahasiswa yang akan mengisi kuesioner dapat memilih kuesioner yang akan diisi kemudian mengisi kuesioner atau menjawab kuesioner. Setelah mahasiswa mengisi kuesioner mahasiswa dapat memilih tombol “Kirim” untuk menyimpan jawaban kuesioner. Jawaban kuesioner ini akan disimpan oleh BE pada database. Apabila kuesioner telah diisi, maka mahasiswa tidak dapat mengisinya lagi. *Business Process* dari mengelola kuesioner dapat dilihat pada Gambar 29 berikut.



Gambar 29. Target Sistem Mengelola Kuesioner

### 3.11.8 Business Process Melihat Pengumuman

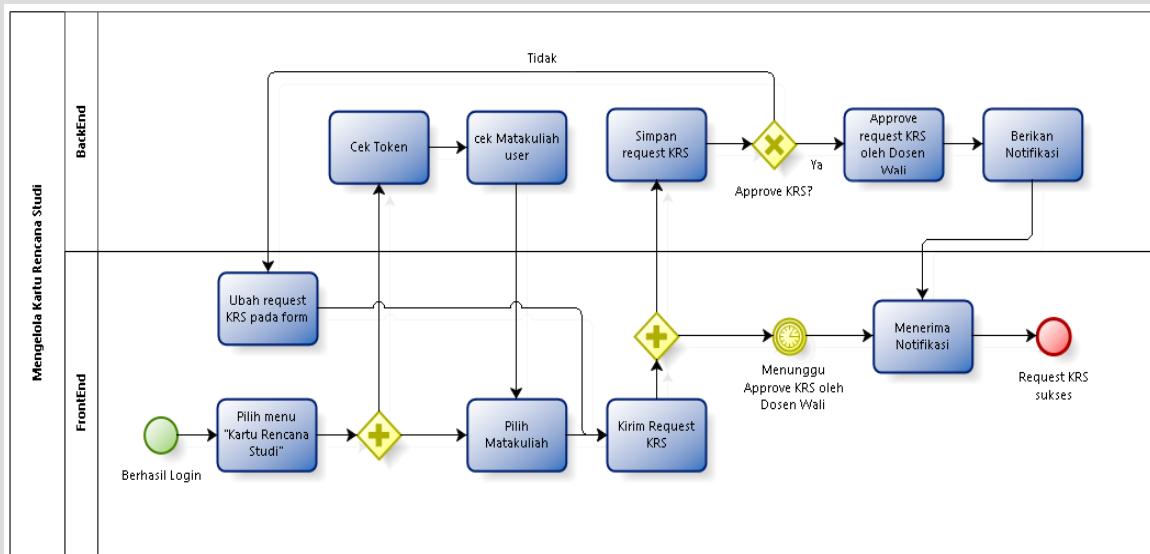
Pada target sistem melihat pengumuman oleh mahasiswa dapat dilakukan dengan dua cara yaitu melihat pengumuman dari halaman “Home” dan “Pengumuman”. Mahasiswa dapat memilih dan melihat daftar-daftar pengumuman yang terdapat di halaman *home*. Selain itu mahasiswa juga dapat melihat pengumuman di halaman pengumuman. Dimana mahasiswa memilih menu “Pengumuman” lalu FE akan menampilkan daftar-daftar pengumuman sesuai dengan pengecekan token yang dilakukan oleh BE. Kemudian mahasiswa dapat memilih pengumuman yang akan dilihat oleh mahasiswa. *Business Process* dari melihat pengumuman oleh mahasiswa dapat dilihat pada Gambar 30 berikut.



Gambar 30. Target Sistem Melihat Pengumuman

### 3.11.9 Business Process Mengelola Kartu Rencana Studi (KRS)

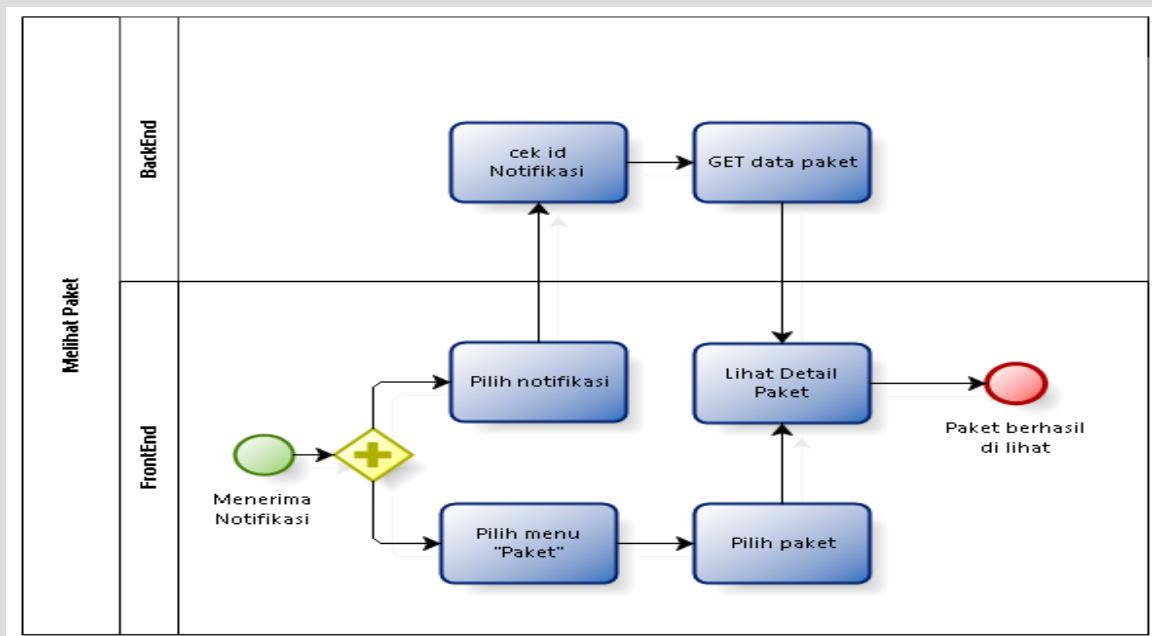
Target sistem mengelola KRS terdiri dari dua yaitu melakukan *request* KRS dan edit KRS. *Request* KRS dapat dilakukan dengan memilih menu “Kartu Rencana Studi”, lalu memilih matakuliah yang akan diambil dalam satu semester. Setelah mahasiswa selesai memilih matakuliah, mahasiswa dapat memilih tombol “kirim”. Data KRS yang dikirim oleh mahasiswa akan disimpan dalam BE. Kemudian mahasiswa akan menunggu *approve* KRS oleh dosen wali. Selain itu, apabila status *request* KRS masih menunggu mahasiswa dapat mengubah *request* KRS. Edit *request* KRS dapat dilakukan dengan memilih menu “Edit” kemudian memilih matakuliah kembali dan tombol “Kirim” dan disimpan oleh BE. Apabila *request* KRS mahasiswa telah di *approve* mahasiswa akan menerima notifikasi *approve request* KRS. *Business Process* dari mengelola KRS dapat dilihat pada Gambar 31 berikut.



Gambar 31. Target Sistem Mengelola KRS

### 3.11.10 Business Process Melihat Paket

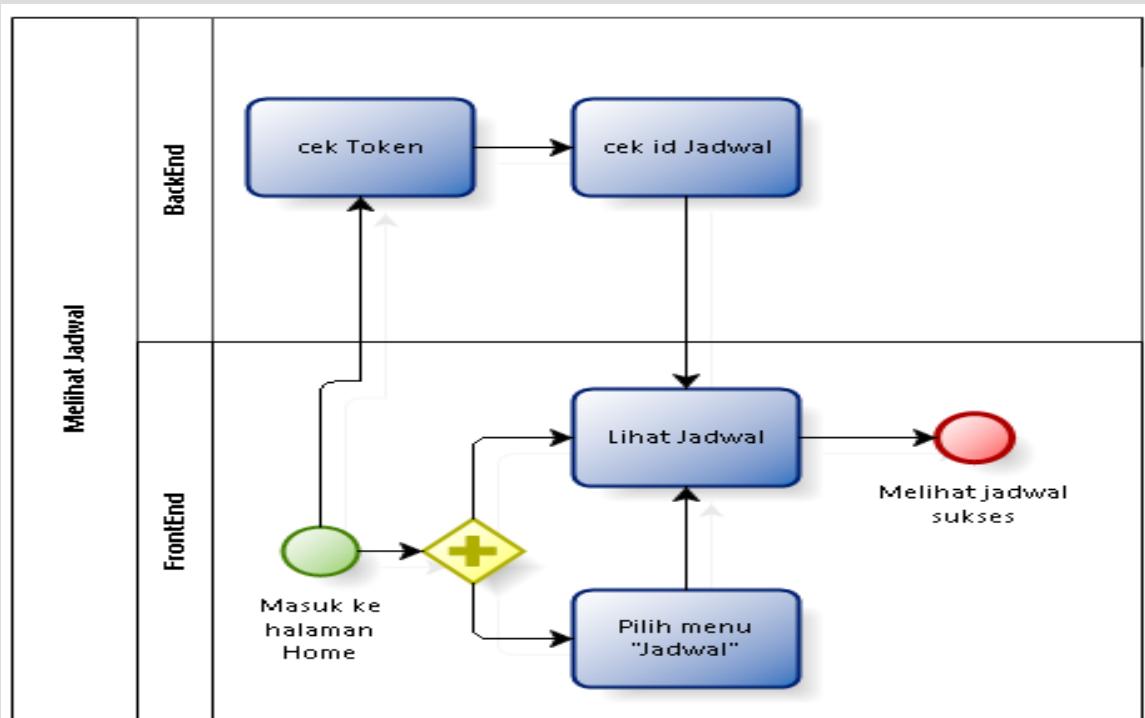
Target sistem melihat paket oleh mahasiswa dapat dilakukan dengan dua cara yaitu melalui notifikasi dan melalui menu “Paket”. Melalui menu “Paket”. Mahasiswa dapat memilih menu “Paket”, lalu aplikasi akan menampilkan daftar-daftar paket. Kemudian mahasiswa memilih paket yang akan dilihat dan aplikasi akan menampilkan detail paket. Selain itu melalui notifikasi, BE akan melakukan cek *id* notifikasi dan GET data paket untuk ditampilkan di FE. Dalam detail paket, nama mahasiswa pemilik paket akan ditampilkan oleh sistem apabila *id* mahasiswa sama dengan *id* paket. Dan untuk paket yang *id* nya berbeda dengan *id* pemilik akun maka nama paket dalam detail paket tidak akan ditampilkan. *Business Process* dari melihat paket oleh mahasiswa dapat dilihat pada Gambar 32 berikut.



Gambar 32. Target Sistem Melihat Paket

### 3.11.11 Business Process Melihat Jadwal Perkuliahan

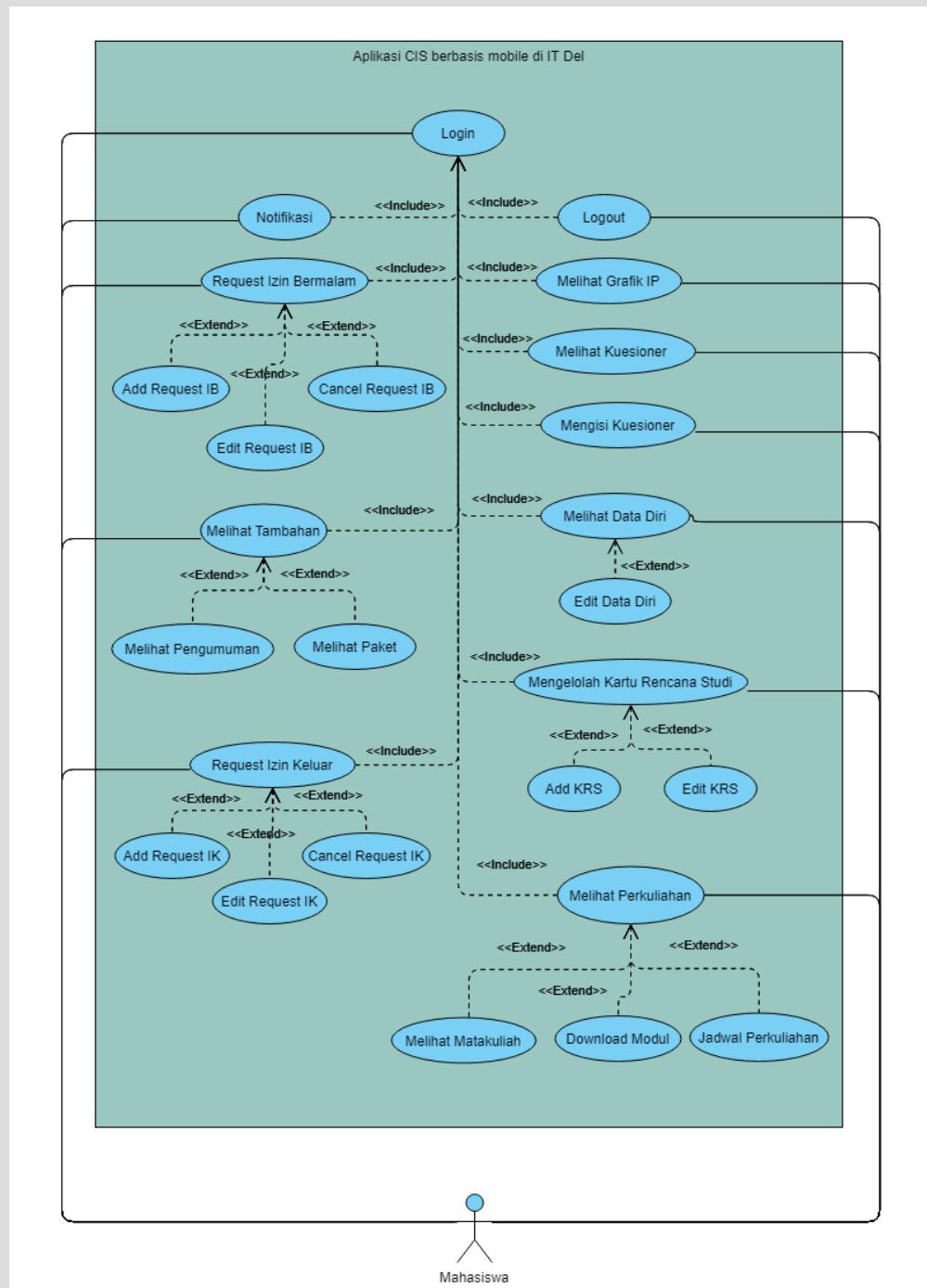
Pada target sistem melihat jadwal perkuliahan dapat dilakukan mahasiswa melalui dua cara yaitu melihat jadwal perkuliahan yang terdapat di halaman *home* dan di halaman jadwal. Mahasiswa dapat melihat jadwal perkuliahan yang terdapat di halaman *home*, dimana BE akan melakukan pengecekan token untuk menyesuaikan jadwal mahasiswa. Jadwal yang ditampilkan di *home* adalah jadwal perkuliahan yang sesuai dengan hari itu juga. Selain itu mahasiswa dapat juga melihat jadwal perkuliahan pada halaman jadwal dengan memilih menu “Jadwal”, lalu aplikasi akan menampilkan jadwal-jadwal perkuliahan mahasiswa secara keseluruhan mulai dari hari Senin hingga hari Jumat. *Business Process* dari melihat jadwal perkuliahan oleh mahasiswa dapat dilihat pada Gambar 33 berikut.



Gambar 33. Target Sistem Melihat Jadwal Perkuliahan

### 3.12 Use Case Diagram

*Use case diagram* merupakan gambaran dari kemampuan atau hal apa saja yang dapat dilakukan oleh aktor terhadap sistem. Pada Gambar 34 akan dijelaskan tentang *use case diagram* yang akan dibangun.



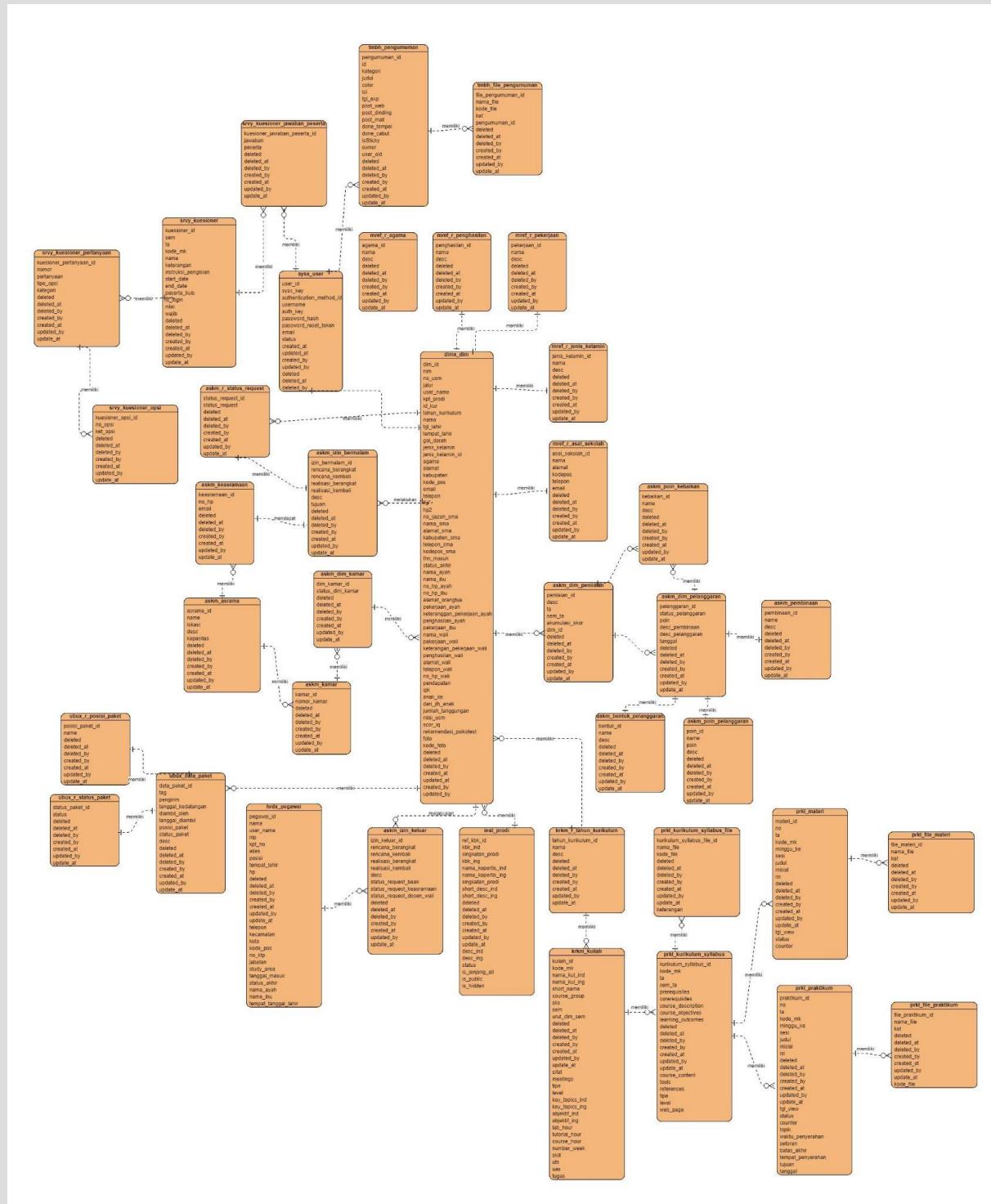
**Gambar 34. Use Case Diagram**

### 3.13 Use Case Scenario

Pada subbab berikut akan dijelaskan tentang *use case scenario* sistem dan aplikasi yang akan dibangun. Adapun gambaran dari *use case scenario* aplikasi dapat dilihat pada Lampiran 3.

### 3.14 ER-Diagram

ER-Diagram yang digunakan dalam pembangunan aplikasi CIS berbasis *mobile* dengan layanan API dapat dilihat pada Gambar 35 berikut.



### Gambar 35. ER-Diagram

Database yang digunakan terdiri dari 2 tabel yaitu tabel master dan tabel *transactional*. Tabel Master adalah tabel yang berdiri sendiri, datanya cenderung tidak berubah-ubah.

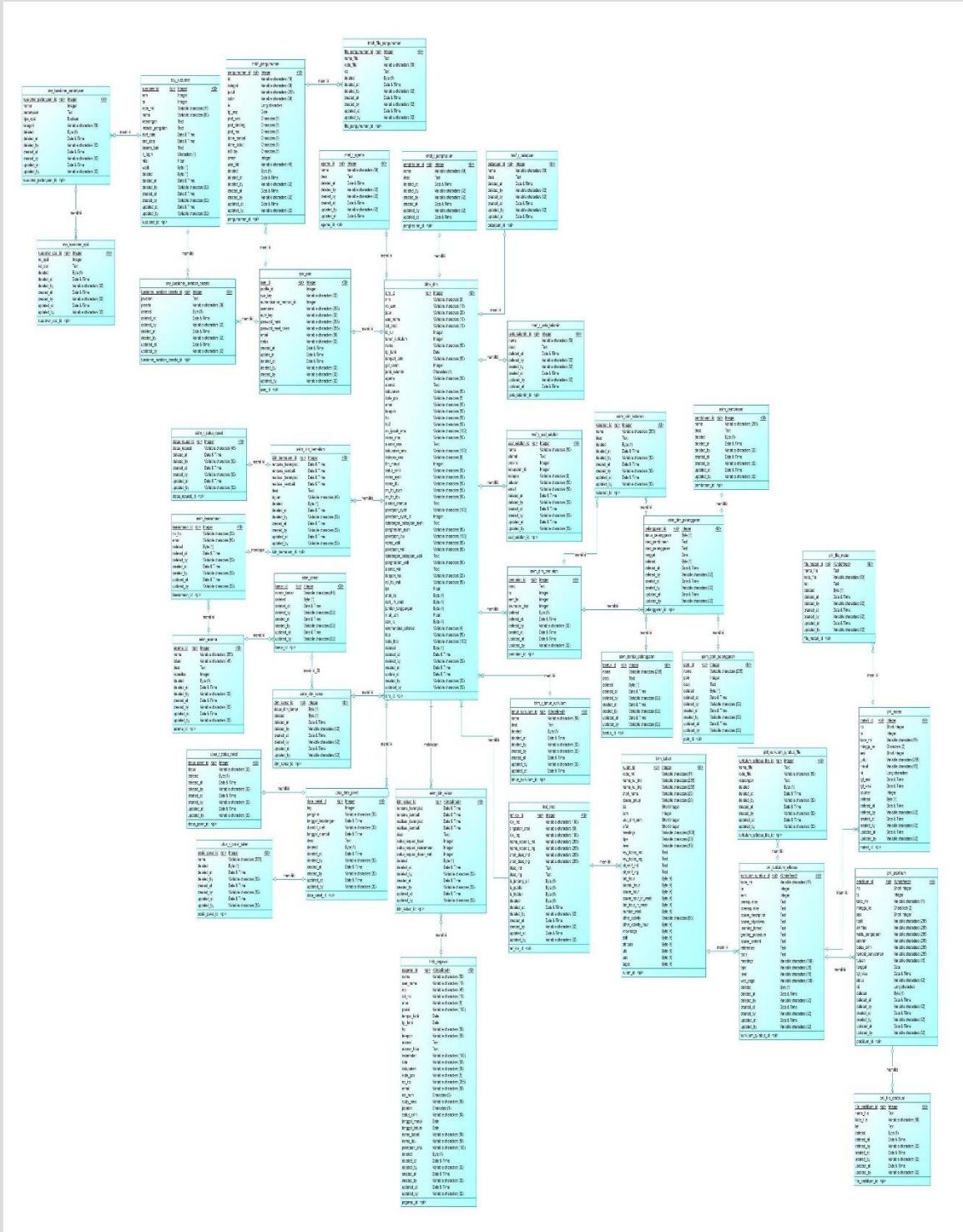
Sedangkan tabel *Transactional* adalah tabel yang bersifat relatif atau yang isi datanya dapat berubah-ubah, tabel yang tidak dapat berdiri sendiri. Berikut daftar tabel master dan tabel *transactional* yang digunakan dalam database Pengembangan Fitur-Fitur CIS berbasis *Mobile* dengan layanan API.

**Tabel 4. Daftar Tabel Master dan Tabel *Transactional***

No	Tabel Master	Tabel Transactional
1	mref_r_agama	dimx_dim
2	mref_r_penghasilan	askm_dim_penilaian
3	mref_r_pekerjaan	askm_dim_pelanggaran
4	mref_r_jenis_kelamin	krkm_tahun_kurikulum
5	mref_r_asal_sekolah	krkm_kuliah
6	askm_poin_kebaikan	prkl_kurikulum_syllabus
7	askm_pembinaan	prkl_materi
8	askm_poin_pelanggaran	prkl_praktikum
9	askm_bentuk_pelanggaran	askm_izin_keluar
10	prkl_kurikulum_syllabus_file	ubux_data_paket
11	prkl_file_materi	askm_kamar
12	prkl_file_praktikum	askm_dim_kamar
13	hrdx_pegawai	askm_izin_bermalam
14	inst_prodi	srvy_kuesioner_pertanyaan
15	ubux_r_status_paket	srvy_kuesioner
16	ubux_r_posisi_paket	sysx_user
17	askm_keasramaan	srvy_kuesioner_jawaban_peserta
18	askm_r_status_request	tmbh_pengumuman
19	srvy_kuesioner_opsi	
20	tmbh_file_pengumuman	

### 3.15 Conceptual Data Model

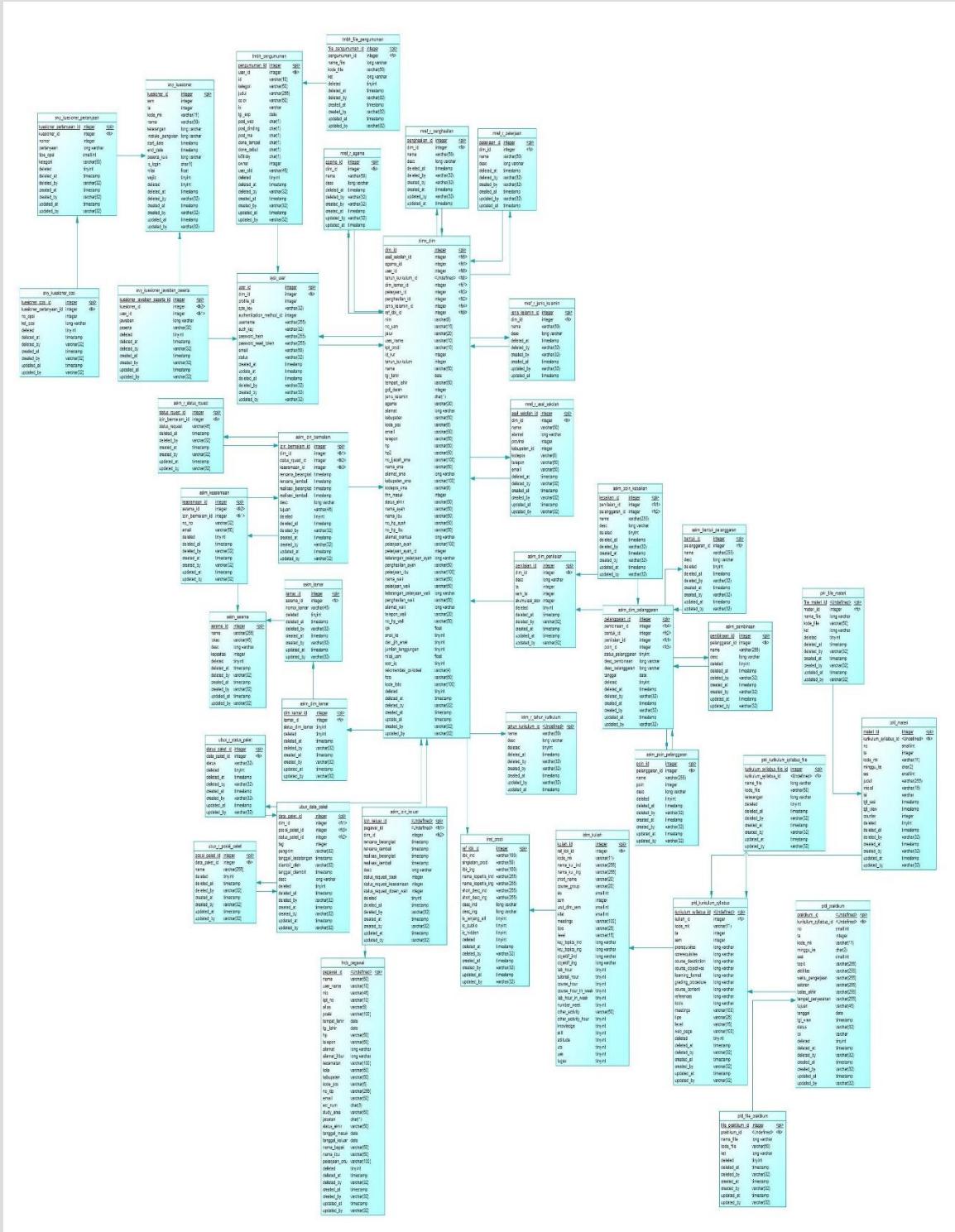
Pada Gambar 36 merupakan *Conceptual Data Model* pada aplikasi yang akan dibangun.



**Gambar 36. Conceptual Data Model**

### 3.16 Physical Data Model

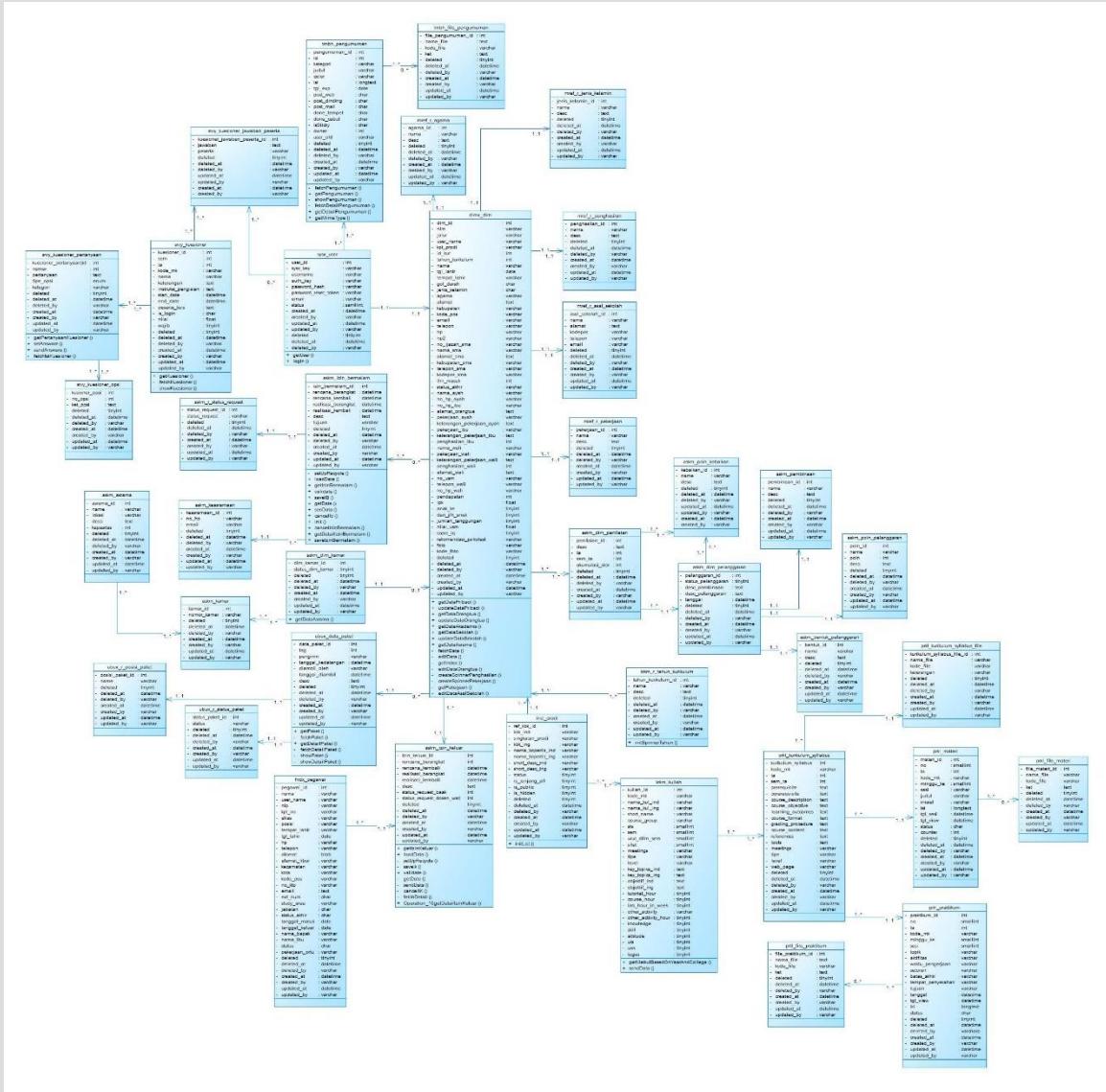
Pada Gambar 37 merupakan *Physical Data Model* pada aplikasi yang akan dibangun.



### Gambar 37. Physical Data Model

### 3.17 Class Diagram

Pada Gambar 38 merupakan *Class Diagram* pada aplikasi yang akan dibangun.



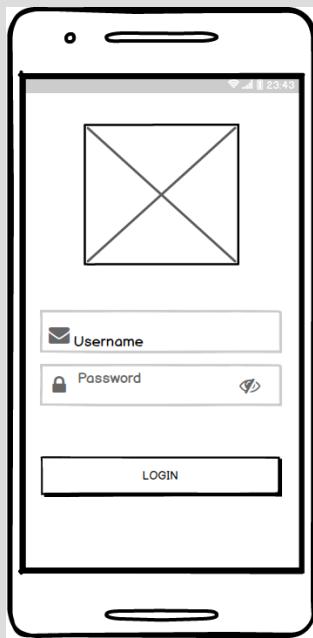
### Gambar 38. Class Diagram

### 3.18 Desain User Interface

Pada sub bab ini akan dijelaskan tentang desain yang akan diimplementasikan oleh mahasiswa dalam sistem.

### **3.18.1 Desain User Interface - *Login***

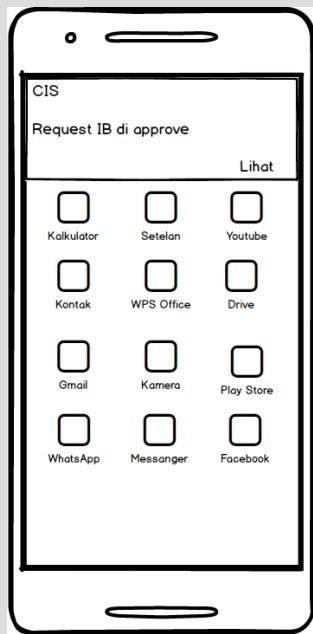
Tampilan desain aplikasi untuk melakukan *login* dapat dilihat pada Gambar 39 berikut.



Gambar 39. Desain User Interface - Login

### 3.18.2 Desain User Interface – Melihat Notifikasi

Tampilan desain aplikasi untuk melihat notifikasi dapat dilihat pada Gambar 40 berikut.



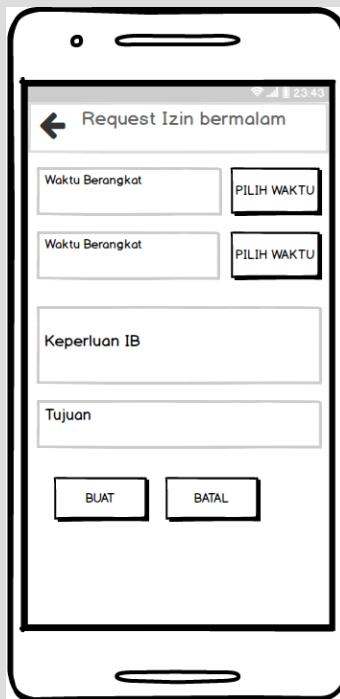
Gambar 40. Desain User Interface - Notifikasi

### 3.18.3 Desain User Interface – Mengelola Request Izin Bermalam

Tampilan desain aplikasi mengelola *request* izin bermalam terdiri dari *request* izin bermalam, edit izin bermalam dan data izin bermalam. Berikut desain aplikasi mengelolaa *request* izin bermalam.

1. Desain User Interface – Request Izin Bermalam

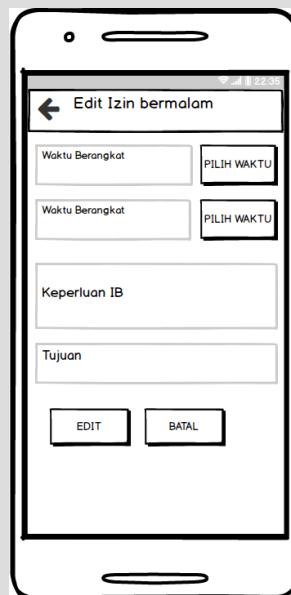
Tampilan desain aplikasi *request izin bermalam* dapat dilihat pada Gambar 41 berikut.



**Gambar 41. Desain User Interface – Request Izin Bermalam**

## 2. Desain User Interface – Edit Request Izin Bermalam

Tampilan desain aplikasi untuk melakukan edit *request izin bermalam* dapat dilihat pada Gambar 42 berikut.



**Gambar 42. Desain User Interface – Edit Request Izin Bermalam**

## 3. Desain User Interface – Data Izin Bermalam

Tampilan desain aplikasi untuk melihat data izin bermalam dapat dilihat pada Gambar 43 berikut.



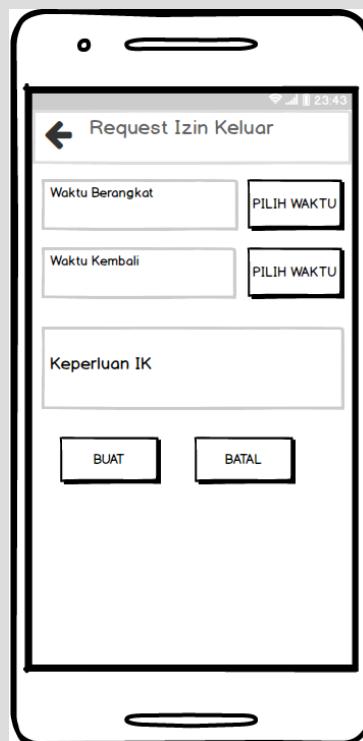
Gambar 43. Desain *User Interface* – Data Izin Bermalam

#### 3.18.4 Desain User Interface – Mengelola Request Izin Keluar

Tampilan desain aplikasi mengelola *request izin keluar* terdiri dari *request izin keluar*, edit *request izin keluar* dan data izin keluar. Berikut desain aplikasi mengelola *request izin keluar*.

##### 1. Desain *User Interface* – Request Izin Keluar

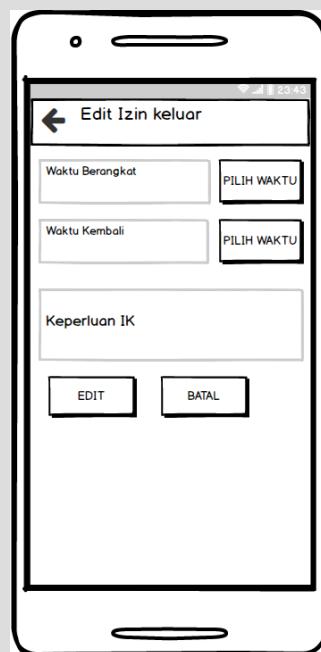
Tampilan desain aplikasi *request izin keluar* dapat dilihat pada Gambar 44 berikut.



**Gambar 44. Desain User Interface – Request Izin Keluar**

2. Desain User Interface – Edit Request Izin Keluar

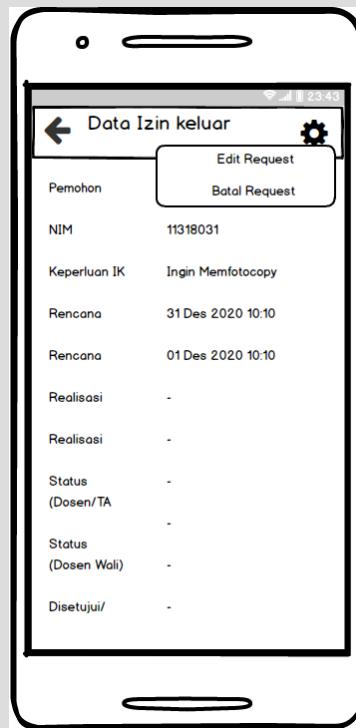
Tampilan desain aplikasi untuk melakukan edit *request* izin keluar dapat dilihat pada Gambar 45 berikut.



**Gambar 45. Desain User Interface - Edit Request Izin Keluar**

3. Desain User Interface – Data Izin Keluar

Tampilan desain aplikasi untuk melihat data izin keluar dapat dilihat pada Gambar 46 berikut.



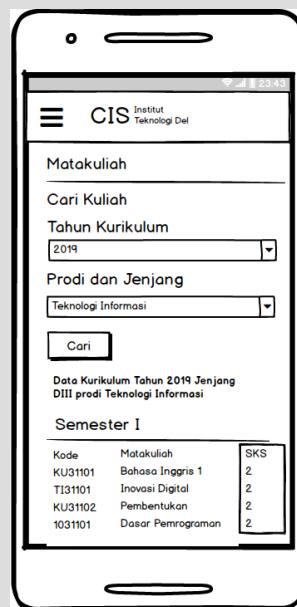
Gambar 46. Desain *User Interface* – Data Izin Keluar

### 3.18.5 Desain *User Interface* – Mengelola Perkuliahan

Tampilan desain aplikasi mengelola perkuliahan terdiri dari melihat mata kuliah dan unduh modul perkuliahan. Berikut desain aplikasi mengelola perkuliahan.

#### 1. Desain *User Interface* – Melihat Matakuliah

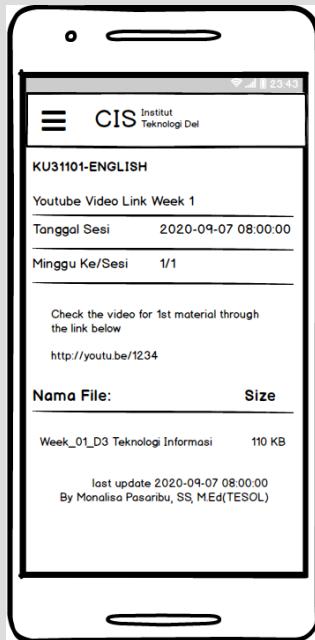
Tampilan desain aplikasi melihat matakuliah dapat dilihat pada Gambar 47 berikut.



Gambar 47. Desain *User Interface* – Melihat Mata Kuliah

## 2. Desain *User Interface* – Unduh Modul

Tampilan desain aplikasi untuk melakukan unduh modul dapat dilihat pada Gambar 48 berikut.



Gambar 48. Desain *User Interface* – Unduh Modul

### 3.18.6 Desain User Interface - Mengelola Data Diri

Tampilan desain aplikasi mengelola data diri terdiri dari melihat data diri dan edit data diri.

Berikut desain aplikasi mengelola data diri.

#### 1. Desain *User Interface* – Melihat Data Diri

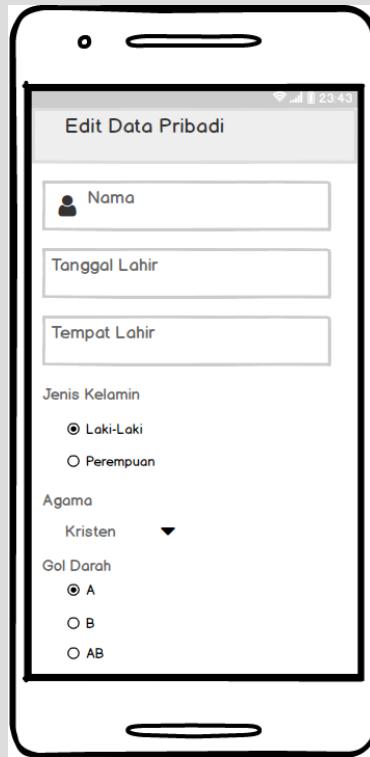
Tampilan desain aplikasi melihat data diri dapat dilihat pada Gambar 49 berikut.



Gambar 49. Desain *User Interface* – Melihat Data Diri

## 2. Desain *User Interface* – Edit Data Diri

Tampilan desain aplikasi untuk melakukan edit data diri dapat dilihat pada Gambar 50 berikut.



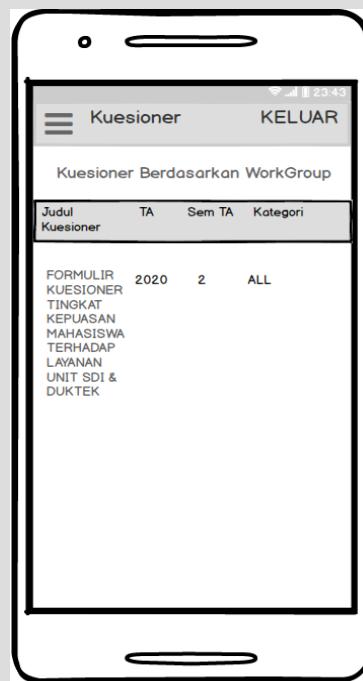
Gambar 50. Desain *User Interface* – Edit Data Diri

### 3.18.7 Desain *User Interface* - Mengelola Kuesioner

Tampilan desain aplikasi untuk mengelola kuesioner terdiri dari melihat kuesioner dan mengisi kuesioner. Berikut desain aplikasi mengelola kuesioner.

#### 1. Desain *User Interface* – Melihat Kuesioner

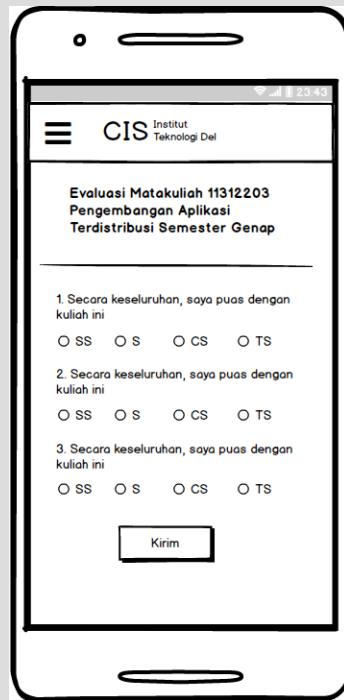
Tampilan desain aplikasi melihat kuesioner dapat dilihat pada Gambar 51 berikut.



Gambar 51. Desain *User Interface* – Melihat Kuesioner

## 2. Desain *User Interface* – Mengisi Kuesioner

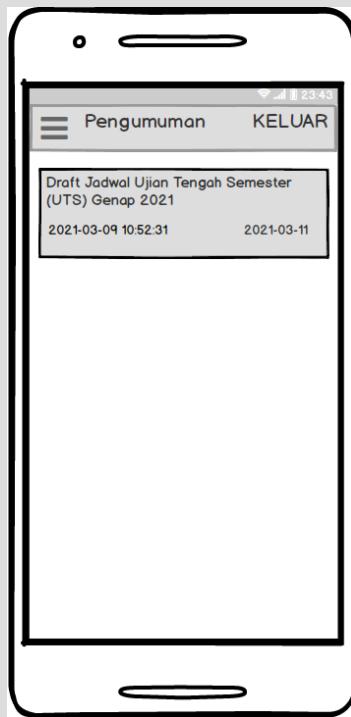
Tampilan desain aplikasi untuk mengisi kuesioner dapat dilihat pada Gambar 52 berikut.



Gambar 52. Desain *User Interface* – Mengisi Kuesioner

## 3.18.8 Desain *User Interface* - Melihat Pengumuman

Tampilan desain aplikasi untuk melihat pengumuman dapat dilihat pada Gambar 53 berikut.



Gambar 53. Desain *User Interface* – Melihat Pengumuman

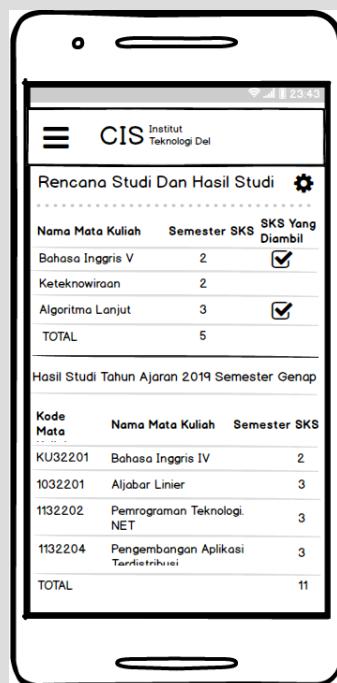
### 3.18.9 Desain *User Interface* - Mengelola Kartu Rencana Studi (KRS)

Tampilan desain aplikasi untuk mengelola KRS terdiri dari mengisi KRS dan edit KRS.

Berikut desain aplikasi mengelola KRS.

#### 1. Desain *User Interface* – Mengisi KRS

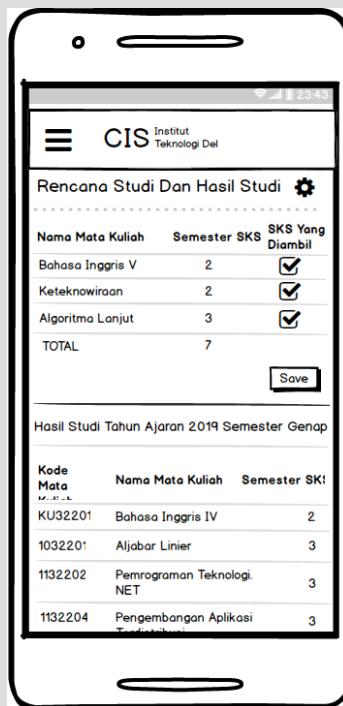
Tampilan desain aplikasi mengisi KRS dapat dilihat pada Gambar 54 berikut.



Gambar 54. Desain *User Interface* – Mengisi KRS

## 2. Desain *User Interface* – Edit KRS

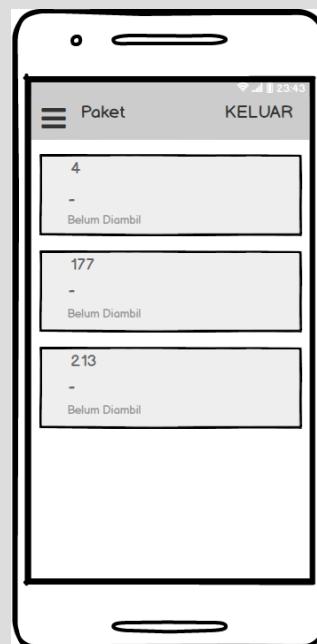
Tampilan desain aplikasi untuk melakukan edit KRS dapat dilihat pada Gambar 55 berikut.



Gambar 55. Desain *User Interface* – Edit KRS

### 3.18.10 Desain User Interface - Melihat Paket

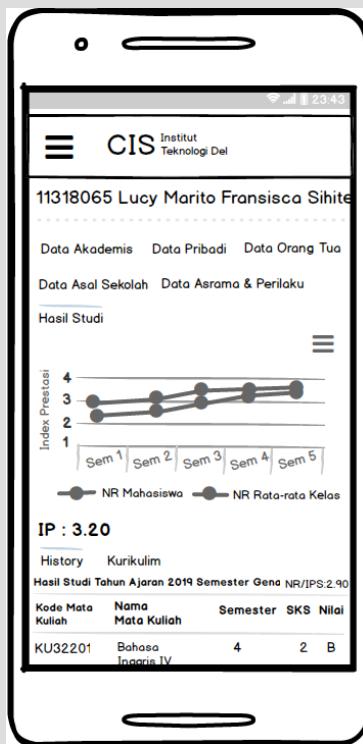
Tampilan desain aplikasi untuk melihat paket dapat dilihat pada Gambar 56 berikut.



Gambar 56. Desain *User Interface* – Melihat Paket

### 3.18.11 Desain User Interface - Melihat Grafik IP

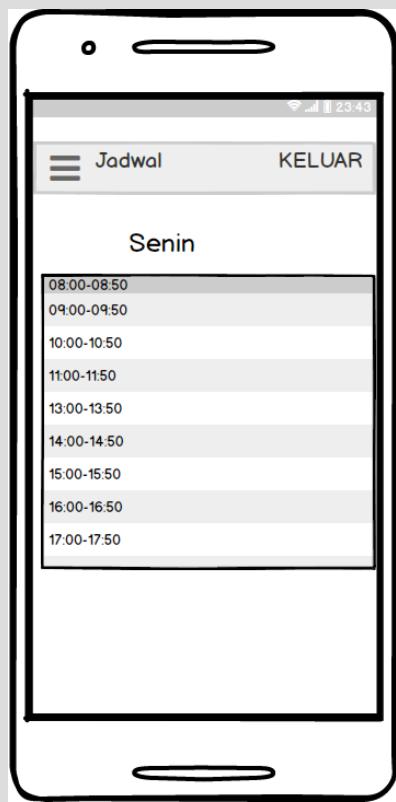
Tampilan desain aplikasi untuk melihat grafik IP dapat dilihat pada Gambar 57 berikut.



Gambar 57. Desain User Interface – Melihat Grafik IP

### 3.18.12 Desain User Interface - Melihat Jadwal

Tampilan desain aplikasi untuk melihat jadwal dapat dilihat pada Gambar 58 berikut.



**Gambar 58. Desain User Interface – Melihat Jadwal**

## **BAB IV**

### **IMPLEMENTASI**

Pada bab ini akan dijelaskan deskripsi umum sistem yang meliputi kebutuhan implementasi, batasan implementasi, implementasi metode pengembangan dan implementasi aplikasi

#### **4.1 Kebutuhan Implementasi**

Pada subbab ini akan dijelaskan mengenai spesifikasi *hardware* dan *software* yang dibutuhkan dalam proses implementasi. Berikut akan dijelaskan mengenai suatu kebutuhan implementasi yang digunakan oleh penulis dalam melakukan implementasi.

##### **4.1.1 Perangkat Keras (*Hardware*)**

Implementasi yang dilakukan dalam penggerjaan Tugas Akhir menggunakan perangkat keras (*hardware*). Spesifikasi *hardware* yang digunakan untuk melakukan implementasi dapat dilihat di bawah ini.

**Tabel 5. Spesifikasi *Hardware Laptop 1***

<b>Laptop 1</b>	Lenovo
<b>Processor</b>	Intel(R) Core(TM) i5-7200U 2.71 GHz
<b>CPU</b>	2.71 GHz
<b>RAM</b>	8.00 GB
<b>Hardisk</b>	HDD 1 TB

**Tabel 6. Spesifikasi *Hardware Laptop 2***

<b>Laptop 2</b>	Lenovo
<b>Processor</b>	Intel(R) Core(TM) i5-7200U 2.71 GHz
<b>CPU</b>	2.71 GHz
<b>RAM</b>	8.00 GB
<b>Hardisk</b>	HDD 1 TB

**Tabel 7. Spesifikasi *Hardware Laptop 3***

<b>Laptop 3</b>	Lenovo
-----------------	--------

<b>Processor</b>	Intel(R) Core(TM) i5-7200U 2.71 GHz
<b>CPU</b>	2.71 GHz
<b>RAM</b>	8.00 GB
<b>Hardisk</b>	HDD 1 TB

**Tabel 8. Hardware Smartphone**

<b>Smartphone</b>	Oppo A83
<b>ROM</b>	32 GB
<b>RAM</b>	3 GB
<b>Sistem Operasi</b>	Android 7.1.1 Nougat

#### **4.1.2 Perangkat Lunak (Software)**

Rincian spesifikasi *software* yang digunakan pada pelaksanaan implementasi diuraikan pada Tabel 9 berikut:

**Tabel 9. Spesifikasi software**

<b>No</b>	<b>Software</b>	<b>Spesifikasi</b>
1	<i>Operating System</i>	Windows 10
2	<i>Development Tools</i>	<i>Android Studio, PhpStorm, Postman</i>
3	<i>Programming Language</i>	Php dan Kotlin
4	<i>Database Tools</i>	<i>SQLyog</i>
5	<i>Design Tools</i>	<i>Bizagi, Balsamiq Mockups, Enterprise Architect, Power Designer</i>
6	<i>Web Server</i>	<i>Apache</i>
7	Microsoft Office	Ms. Excel 2010, Ms. Word 2010, PowerPoint 2010

#### **4.2 Batasan Implementasi**

Adapun batasan dari pengembangan aplikasi CIS ini adalah sebagai berikut.

1. Aplikasi CIS ini terdiri dari satu jenis *user interface*, yaitu versi *mobile*.
2. Aplikasi versi *mobile* ini digunakan oleh mahasiswa IT Del untuk melihat informasi-informasi akademik maupun non-akademik yang disediakan oleh dosen,

staf maupun keasramaan yaitu: melihat pengumuman, melihat paket, melakukan izin bermalam, melakukan izin keluar, melihat modul perkuliahan, melihat data diri, melakukan pengisian Kartu Rencana Studi dan melihat Kuesioner.

### **4.3 Implementasi Aplikasi**

Pada sub bab ini akan dijelaskan mengenai aplikasi yang diimplementasikan dalam pengerjaan Tugas Akhir. Pengembangan fitur-fitur CIS yang dibangun merupakan aplikasi berbasis *mobile* dengan layanan API. API dibangun berupa *website* menggunakan PHP dengan *framework* Laravel dan berada di antara aplikasi dan *database* yang bertindak sebagai lapisan perantara yang memproses *transfer* data antar aplikasi. Pada *mobile* dibangun menggunakan Android Studio dengan bahasa pemrograman Kotlin. Dalam melakukan implementasi aplikasi *developer* telah melakukan *hosting* terhadap *website* API menggunakan *protocol* HTTPS (*Hypertext Transfer Protocol Secure*). Hal ini dilakukan supaya data yang dikirimkan melalui *browser* tidak dapat dilihat dan disalahgunakan oleh pihak lain sehingga data pengguna aplikasi akan terlindungi integrasi dan kerahasiaan datanya. Adapun implementasi dari aplikasi dapat dilihat pada Lampiran 4.

## **BAB V**

### **HASIL DAN PENGUJIAN**

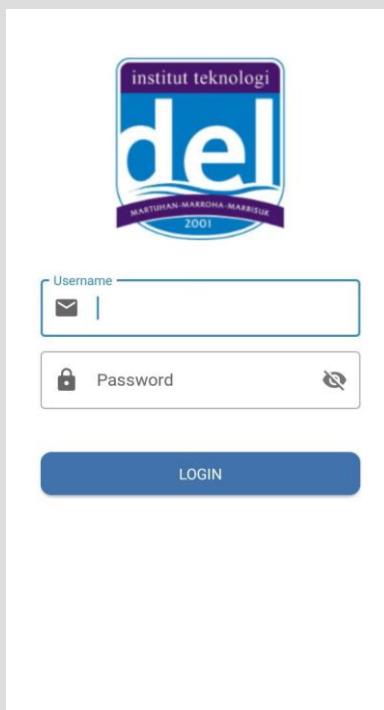
Bab ini menjelaskan mengenai hasil dan pengujian dari Pengembangan Fitur-Fitur *Campus Information System* Berbasis *Mobile* dengan Layanan API.

#### **5.1 Hasil Aplikasi**

Pada sub bab ini akan dijelaskan mengenai hasil Pengembangan Fitur-Fitur *Campus Information System* Berbasis *Mobile* dengan Layanan API. Pada aplikasi juga telah dilakukan pengujian terhadap masing-masing fungsi. Adapun hasil *User Interface* yang dihasilkan adalah sebagai berikut.

##### **5.1.1 Login**

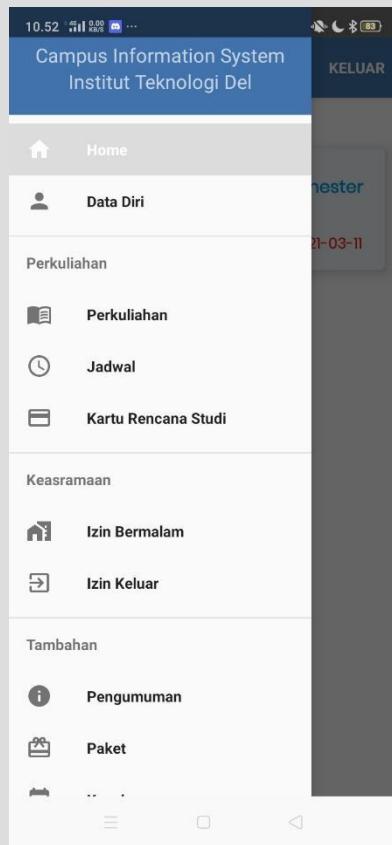
Halaman dari *login* oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 59 berikut.



**Gambar 59. Implementasi Login**

##### **5.1.2 Home**

Halaman dari *Home* oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 60 berikut.



Gambar 60. Implementasi *Home*

### 5.1.3 Mengelola Request Izin Bermalam (IB)

Halaman dari mengelola *request* izin bermalam terdiri dari *request* izin bermalam dan edit *request* izin bermalam. Berikut halaman mengelola *request* izin bermalam.

#### 1. *Request* Izin Bermalam

Halaman dari *Request* IB oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 61 berikut.



Gambar 61. Implementasi Add Request IB

## 2. Edit Request Izin Bermalam

Halaman dari Edit Request IB oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 62 berikut.



Gambar 62. Implementasi Edit Request IB

#### **5.1.4 Mengelola Request Izin Keluar (IK)**

Halaman mengelola *request izin keluar* terdiri dari *request izin keluar* dan edit *request izin keluar*. Berikut halaman mengelola *request izin keluar*.

##### **1. Request Izin Keluar**

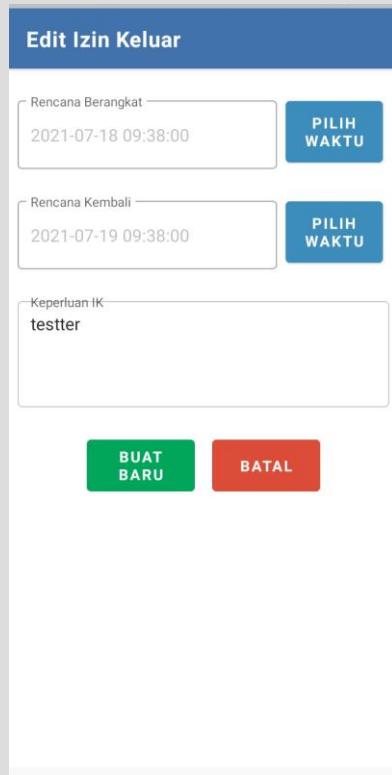
Halaman dari *Request IK* oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 63 berikut.



**Gambar 63. Implementasi Add Request IK**

##### **2. Edit Request Izin Keluar**

Halaman dari Edit *Request IK* oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 64 berikut.



Gambar 64. Implementasi Edit Request IBK

### 5.1.5 Mengelola Data Diri

Halaman mengelola data diri terdiri dari melihat data diri dan edit data diri. Berikut halaman mengelola data diri.

#### 1. Melihat Data Diri

Halaman Melihat Data Diri oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 65 berikut.



**Gambar 65. Implementasi Melihat Data Diri**

## 2. Edit Data Diri

Halaman Edit Data Diri oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 66 berikut.

The screenshot shows a mobile application interface titled "Edit Data Pribadi" (Edit Personal Data). It contains the following fields:

- Nama:** Martinus Yudha Chrisanto Sitinjak
- Tanggal Lahir:** 2000-03-18
- Tempat Lahir:** Medan
- Jenis Kelamin:** Laki-laki (radio button selected)
- Agama:** Kristen
- Gol Darah:** A (radio button selected)

**Gambar 66. Implementasi Edit Data Diri**

### **5.1.6 Mengelola Kuesioner**

Halaman mengelola kuesioner terdiri dari melihat kuesioner dan mengisi kuesioner. Berikut halaman mengelola kuesioner.

#### **1. Melihat Kuesioner**

Halaman dari Melihat Kuesioner oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 67 berikut.



**Gambar 67. Implementasi Melihat Kuesioner**

#### **2. Mengisi Kuesioner**

Halaman dari Mengisi Kuesioner oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 68 berikut.

**Isi Kuesioner**

---

[PPKHA] Data Diri Lulusan IT Del Tahun 2021

---

**Data Diri**

1. Tahun Masuk (Angkatan)
  - 2015
  - 2016
  - 2017
  - 2018
2. Program Studi
  - D3 Teknologi Informasi
  - D3 Teknologi Komputer
  - D4 Teknologi Rekayasa Perangkat Lunak
  - S1 Informatika
  - S1 Sistem Informasi
  - S1 Teknik Elektro
  - S1 Manajemen Rekayasa
  - S1 Teknik Bioproses
3. NIM (contoh: 11S15047)

**Gambar 68. Implementasi Mengisi Kuesioner**

### 5.1.7 Melihat Pengumuman

Halaman dari Melihat Pengumuman oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 69 berikut.

The image shows two screenshots of a mobile application interface. The left screenshot displays a list of announcements with the title "Pengumuman". One announcement is selected, showing its details. The right screenshot shows the "Detail Pengumuman" screen for the selected announcement. The announcement content is as follows:

**Draft Jadwal Ujian Tengah Semester (UTS) Genap 2021**

2021-03-09 10:52:31      2021-03-11

Dear Mahasiswa,

Berikut ini terlampir kami sampaikan Draft Jadwal Ujian Tengah Semester (UTS) Genap 2021.

Apabila terdapat Jadwal UTS yang bentrok, Diharapkan mahasiswa dapat menginformasikan ke e-mail [baak@del.ac.id](mailto:baak@del.ac.id) dengan Subjek : **Prodi - Jadwal UTS Bentrok - Angkatan**. Konfirmasi Jadwal UTS yang bentrok diterima paling lama hari **Rabu, 10 Maret 2021, Pukul 12.00 WIB**.

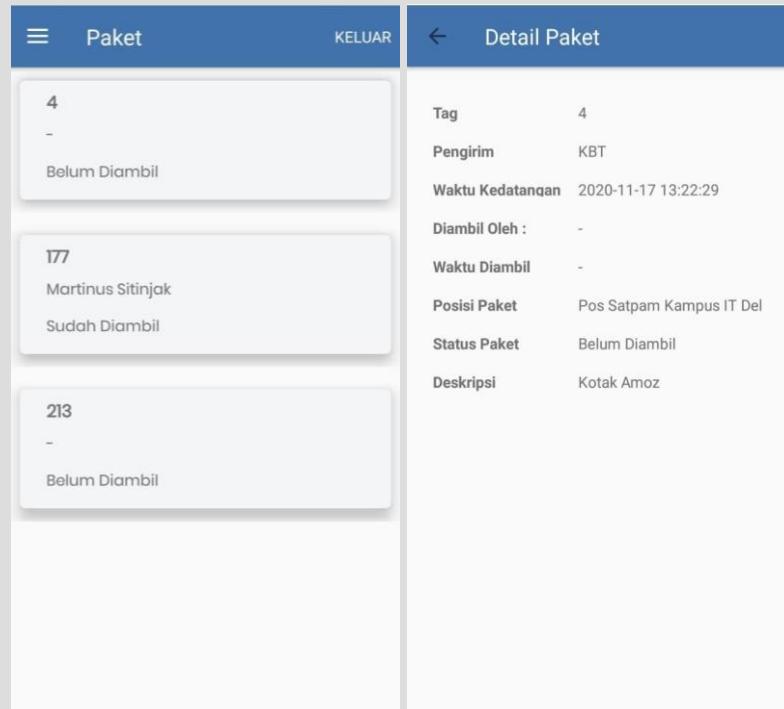
Demikian informasi ini kami sampaikan, terima kasih.  
Salam.

**Nama File**  
Draft\_Jadwal\_UTS\_Semester\_Genap\_2020\_2021.xlsx  
Haloo File  
ttd Nama Pembuat

**Gambar 69. Implementasi Melihat Pengumuman**

### 5.1.8 Melihat Paket

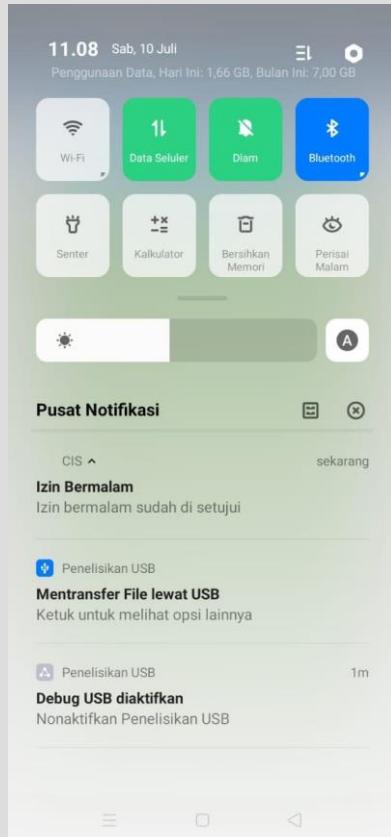
Halaman dari Melihat Paket oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 70 berikut.



Gambar 70. Implementasi Melihat Pengumuman

### 5.1.9 Melihat Notifikasi

Halaman Melihat Notifikasi oleh mahasiswa pada *mobile* dapat dilihat pada Gambar 71 berikut.



Gambar 71. Implementasi Melihat Notifikasi

## 5.2 Pengujian

Pada sub bab ini akan dijelaskan mengenai skenario pengujian yang akan dilakukan terhadap aplikasi yang dibangun.

### 5.2.1 Pengujian Unit

Pengujian unit merupakan teknik pengujian untuk menguji fungsionalitas yang terdapat dalam Pengembangan Fitur-Fitur *Campus Information System* Berbasis *Mobile* dengan Layanan API. Pengujian unit dikelompokkan menjadi beberapa kasus uji berdasarkan fungsional yang dimiliki aplikasi. Jenis pengujian yang dilakukan adalah *Beta Testing*, yaitu metode pengujian secara langsung kepada pengguna aplikasi. Pengujian ini dilakukan untuk memvalidasi kegunaan dan fungsi perangkat lunak yang sudah dikembangkan. Berikut merupakan butir uji yang dilakukan pada aplikasi.

Tabel 10. Pengujian Unit

No Butir Uji	Butir Uji
BU-01	Otentikasi

BU-02	<i>Add Request Izin Bermalam (IB)</i>
BU-03	<i>Edit Request Izin Bermalam (IB)</i>
BU-04	<i>Cancel Request Izin Bermalam (IB)</i>
BU-05	<i>Add Request Izin Keluar (IK)</i>
BU-06	<i>Edit Request Izin Keluar (IK)</i>
BU-07	<i>Cancel Request Izin Keluar (IK)</i>
BU-08	Melihat Matakuliah
BU-09	Unduh Modul
BU-10	Melihat Jadwal
BU-11	Melihat Data Diri
BU-12	Edit Data Diri
BU-13	Melihat Kuesioner
BU-14	Mengisi Kuesioner
BU-15	Melihat Pengumuman
BU-16	Melihat Paket
BU-17	<i>Add Kartu Rencana Studi (KRS)</i>
BU-18	<i>Edit Kartu Rencana Studi (KRS)</i>
BU-19	Melihat Notifikasi

### 5.2.1.1 Skenario Pengujian

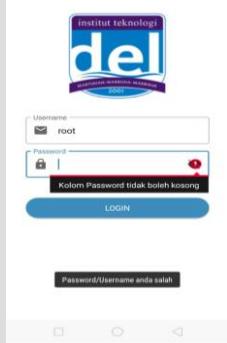
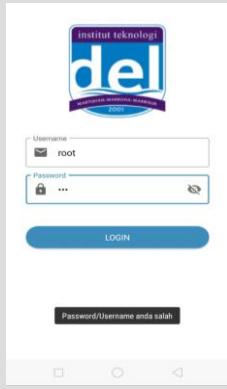
Berikut akan dijelaskan skenario pengujian pada fungsi yang terdapat pada sistem berdasarkan *use case scenario*.

#### 1. Skenario Pengujian Otentikasi

Pengujian otentikasi dilakukan untuk memastikan bahwa mahasiswa dapat masuk ke dalam sistem. Hasil pengujian dapat dilihat pada Tabel 11 berikut.

**Tabel 11. Skenario Pengujian Otentikasi**

<b>Identifikasi</b>	BU-01		
<b>Nama Butir Uji</b>	Test Butir Uji <i>Login</i> .		
<b>Tujuan</b>	Menguji apakah mahasiswa dapat masuk ke dalam aplikasi dengan memasukkan <i>username</i> dan <i>password</i> .		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk dapat mengakses aplikasi.		
<b>Kondisi Awal</b>	Mahasiswa mengakses aplikasi dan berada pada halaman <i>login</i> .		
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Mahasiswa memasukkan <i>username</i> dan <i>password</i> untuk masuk ke aplikasi dan mengklik tombol <i>login</i>.</li> <li>2. Sistem memeriksa kebenaran <i>input</i> dengan yang sudah terdapat pada <i>database</i>.</li> <li>3. Mahasiswa dapat mengakses <i>home</i></li> </ol>		
<b>Kriteria Evaluasi Hasil</b>	Mahasiswa berhasil masuk ke dalam aplikasi.		
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mahasiswa memasukkan <i>username</i> dan <i>password</i>	Mahasiswa berhasil <i>login</i> ke dalam aplikasi	Sesuai dengan yang diharapkan 	Diterima
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			

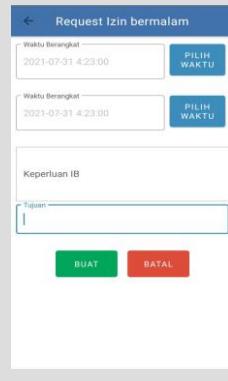
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa tidak mengisi salah satu kolom <i>username</i> atau <i>password</i>	Aplikasi menampilkan pesan <i>error</i>	Sesuai dengan yang diharapkan 	Diterima
Mahasiswa memasukkan <i>username</i> atau <i>password</i> yang salah	Aplikasi menampilkan pesan <i>error</i>	Sesuai dengan yang diharapkan 	Diterima

## 2. Skenario Pengujian *Add Request Izin Bermalam (IB)*

Pengujian *add request* IB dilakukan untuk memastikan bahwa mahasiswa dapat melakukan *add request* IB. Hasil pengujian dapat dilihat pada Tabel 12 berikut.

**Tabel 12. Skenario Pengujian *Add Request Izin Bermalam***

<b>Identifikasi</b>	BU-02
<b>Nama Butir Uji</b>	Test Butir Uji <i>Add Request Izin Bermalam</i>
<b>Tujuan</b>	Menguji fungsi <i>Add Request Izin Bermalam</i>
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk menambah <i>request</i> IB

<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan berada pada halaman <i>home</i> .					
<b>Skenario Uji</b>						
	<ol style="list-style-type: none"> <li>1. Memilih menu Izin Bermalam</li> <li>2. Mengklik tombol “Request IB”</li> <li>3. Mengisi <i>form request IB</i></li> <li>4. Mengklik tombol “Buat Baru”</li> </ol>					
<b>Kriteria Evaluasi Hasil</b>						
Mahasiswa berhasil melakukan <i>request IB</i>						
<b>Kasus dan Hasil Pengujian (Data Normal)</b>						
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan			
Memilih menu Izin Bermalam	Aplikasi menampilkan list-list IB	<p>Sesuai dengan yang diharapkan</p> 	Diterima			
Mengklik tombol “Request IB”	Aplikasi menampilkan halaman <i>request IB</i>	<p>Sesuai dengan yang diharapkan</p> 	Diterima			

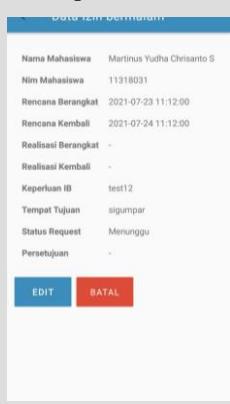
Mengisi <i>form request IB</i> (rencana berangkat, rencana kembali, keperluan IB, tempat tujuan)	Aplikasi berhasil menambah data sesuai inputan mahasiswa	Sesuai dengan yang diharapkan	Diterima
Mengklik tombol “Buat Baru”	<i>Request IB</i> berhasil ditambahkan dan ditampilkan di halaman <i>IB</i> berupa list-list <i>IB</i>	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa mengklik tombol “Buat Baru” namun tidak mengisi sebagian atau secara keseluruhan <i>form</i>	Aplikasi akan menampilkan pesan <i>error</i> dan <i>request IB</i> tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima

### 3. Skenario Pengujian Edit *Request Izin Bermalam (IB)*

Pengujian *edit request IB* dilakukan untuk memastikan bahwa mahasiswa dapat melakukan *edit request IB*. Hasil pengujian dapat dilihat pada Tabel 13 berikut.

**Tabel 13. Skenario Pengujian Edit Request Izin Bermalam**

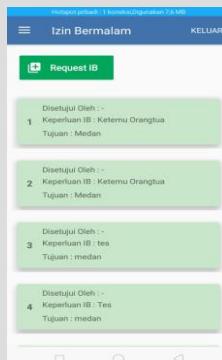
<b>Identifikasi</b>	BU-03		
<b>Nama Butir Uji</b>	Test Butir Uji Edit Request Izin Bermalam		
<b>Tujuan</b>	Menguji fungsi Edit Request Izin Bermalam		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melakukan edit request IB.		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan telah melakukan <i>add request</i> IB		
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Memilih menu Izin Bermalam</li> <li>2. Memilih list IB yang akan diubah</li> <li>3. Mengklik tombol “Edit”</li> <li>4. Mengubah data <i>request</i> IB pada <i>form</i></li> <li>5. Mengklik tombol “Edit Selesai”</li> </ol>		
<b>Kriteria Evaluasi Hasil</b>	Mahasiswa berhasil melakukan edit <i>request</i> IB		
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Memilih menu Izin Bermalam	Aplikasi menampilkan list-list IB	Sesuai dengan yang diharapkan 	Diterima

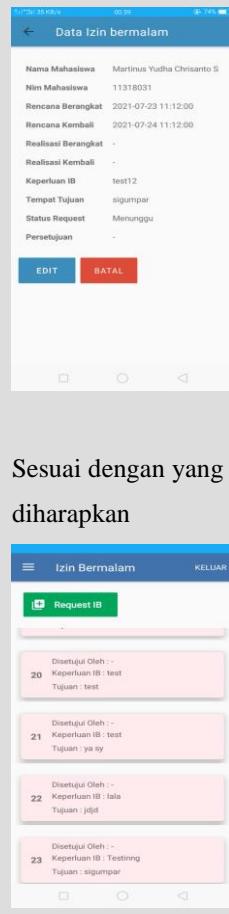
Memilih list IB yang akan diubah	Aplikasi menampilkan detail <i>request IB</i>	Sesuai dengan yang diharapkan 	Diterima
Mengklik tombol “Edit”	Aplikasi menampilkan <i>form edit request IB</i>	Sesuai dengan yang diharapkan	Diterima
Mengubah data <i>request IB</i> (rencana berangkat, rencana kembali, keperluan IB, tempat tujuan)	Data <i>request IB</i> akan berubah sesuai masukan mahasiswa	Sesuai dengan yang diharapkan	Diterima
Mengklik tombol “Edit Selesai”	Data <i>request IB</i> berhasil diedit dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa mengubah data <i>request IB</i> namun tidak mengklik tombol “Edit Selesai”	Data <i>request IB</i> tidak berubah	Sesuai dengan yang diharapkan	Diterima

#### 4. Skenario Pengujian *Cancel Request Izin Bermalam (IB)*

Pengujian *cancel request IB* dilakukan untuk memastikan bahwa mahasiswa dapat melakukan *cancel request IB*. Hasil pengujian dapat dilihat pada Tabel 14 berikut.

**Tabel 14. Skenario Pengujian *Cancel Request Izin Bermalam***

<b>Identifikasi</b>	BU-04		
<b>Nama Butir Uji</b>	Test Butir Uji <i>Cancel Request Izin Bermalam</i>		
<b>Tujuan</b>	Menguji fungsi <i>Cancel Request Izin Bermalam</i>		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melakukan <i>cancel request</i> IB.		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan telah melakukan <i>add request</i> IB		
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Memilih menu Izin Bermalam</li> <li>2. Memilih list IB yang akan dibatalkan</li> <li>3. Mengklik tombol “Batal”</li> </ol>		
<b>Kriteria Evaluasi Hasil</b>	Mahasiswa berhasil melakukan <i>cancel request</i> IB		
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Memilih menu Izin Bermalam	Aplikasi menampilkan list-list IB	Sesuai dengan yang diharapkan 	Diterima
Memilih list IB yang akan dibatalkan	Menampilkan detail data <i>request IB</i>	Sesuai dengan yang diharapkan 	Diterima

Mengklik tombol “Batal”	Data <i>request IB</i> berhasil di <i>cancel</i> dan ditampilkan menjadi list IB warna merah muda		Diterima
-------------------------	---------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------	----------

##### 5. Skenario Pengujian *Add Request Izin Keluar* (IK)

Pengujian *add request* IK dilakukan untuk memastikan bahwa mahasiswa dapat melakukan *add request* IK. Hasil pengujian dapat dilihat pada Tabel 15 berikut.

**Tabel 15. Skenario Pengujian *Add Request Izin Keluar***

<b>Identifikasi</b>	BU-05
<b>Nama Butir Uji</b>	Test Butir Uji <i>Add Request Izin Keluar</i>
<b>Tujuan</b>	Menguji fungsi <i>Add Request Izin Keluar</i>
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk menambah <i>request</i> IK
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi
<b>Skenario Uji</b>	

	<ol style="list-style-type: none"> <li>1. Memilih menu Izin Keluar</li> <li>2. Mengklik tombol “Request Izin Keluar”</li> <li>3. Mengisi <i>form request IK</i></li> <li>4. Mengklik tombol “Buat Baru”</li> </ol>		
<b>Kriteria Evaluasi Hasil</b>			
Mahasiswa berhasil melakukan <i>add request IK</i>			
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memilih menu Izin Keluar	Aplikasi menampilkan daftar-daftar IK	Sesuai dengan yang diharapkan  	Diterima
Mengklik tombol “Request IK”	Aplikasi menampilkan <i>form request IK</i>	Sesuai dengan yang diharapkan  	Diterima

Mengisi <i>form request IK</i> (rencana berangkat, rencana kembali, keperluan IK)	Aplikasi berhasil menambah data sesuai inputan mahasiswa	Sesuai dengan yang diharapkan 	Diterima
Mengklik tombol “Buat Baru”	<i>Request IK</i> berhasil ditambahkan dan ditampilkan di halaman IK	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa mengklik tombol “Buat Baru” namun tidak mengisi sebagian atau secara keseluruhan <i>form</i>	Aplikasi akan menampilkan pesan <i>error</i> dan <i>request IK</i> tidak akan bertambah	Sesuai dengan yang diharapkan	Diterima

## 6. Skenario Pengujian Edit *Request Izin Keluar* (IK)

Pengujian edit *request IK* dilakukan untuk memastikan bahwa mahasiswa dapat melakukan edit *request IK*. Hasil pengujian dapat dilihat pada Tabel 16 berikut.

**Tabel 16. Skenario Pengujian Edit *Request Izin Keluar***

<b>Identifikasi</b>	BU-06
<b>Nama Butir Uji</b>	Test Butir Uji Edit <i>Request Izin Keluar</i>
<b>Tujuan</b>	Menguji fungsi Edit <i>Request Izin Keluar</i>

<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melakukan edit <i>request IK</i>		
<b>Kondisi Awal</b>	Mahasiswa telah login ke dalam aplikasi dan telah melakukan add <i>request IK</i>		
<b>Skenario Uji</b>			
<ol style="list-style-type: none"> <li>1. Memilih menu Izin Keluar</li> <li>2. Memilih daftar IK yang akan diubah</li> <li>3. Mengklik tombol “Edit”</li> <li>4. Mengubah data <i>request IK</i> pada <i>form</i></li> <li>5. Mengklik tombol “Edit Selesai”</li> </ol>			
<b>Kriteria Evaluasi Hasil</b>			
Mahasiswa berhasil melakukan edit <i>request IK</i>			
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memilih menu Izin Keluar	Aplikasi menampilkan list-list IK	Sesuai dengan yang diharapkan	Diterima
			
Memilih list IK yang	Aplikasi menampilkan	Sesuai dengan yang diharapkan	Diterima

akan diubah	detail <i>request IK</i>		
Mengklik tombol “Edit”	Aplikasi menampilkan form edit <i>request IK</i>		Diterima
Mengubah data <i>request IK</i> (rencana berangkat, rencana kembali, keperluan IK)	Data <i>request IK</i> akan berubah sesuai masukan mahasiswa	Sesuai dengan yang diharapkan	Diterima
Mengklik tombol “Edit Selesai”	Data <i>request IK</i> berhasil diedit dan ditampilkan	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan

Mahasiswa mengubah data <i>request</i> IK namun tidak mengklik tombol “Edit Selesai”	Data <i>request</i> IK tidak berubah	Sesuai dengan yang diharapkan	Diterima
--------------------------------------------------------------------------------------	--------------------------------------	-------------------------------	----------

#### 7. Skenario Pengujian *Cancel Request* Izin Keluar (IK)

Pengujian *cancel request* IK dilakukan untuk memastikan bahwa mahasiswa dapat melakukan *cancel request* IK. Hasil pengujian dapat dilihat pada Tabel 17 berikut.

**Tabel 17. Skenario Pengujian *Cancel Request* Izin Keluar**

<b>Identifikasi</b>	BU-07		
<b>Nama Butir Uji</b>	Test Butir Uji <i>Cancel Request</i> Izin Keluar		
<b>Tujuan</b>	Menguji fungsi <i>Cancel Request</i> Izin Keluar		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melakukan <i>cancel request</i> IK		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan telah melakukan <i>add request</i> IK		
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Memilih menu Izin Bermalam</li> <li>2. Memilih daftar IK yang akan dibatalkan</li> <li>3. Mengklik tombol “Batal”</li> </ol>		
<b>Kriteria Evaluasi Hasil</b>	Mahasiswa berhasil melakukan <i>cancel request</i> IK		
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Memilih menu Izin Keluar	Aplikasi menampilkan list-list IK	Sesuai dengan yang diharapkan	Diterima

Memilih list IK yang akan dibatalkan	Aplikasi menampilkan detail data request IK	Sesuai dengan yang diharapkan	Diterima
--------------------------------------	---------------------------------------------	-------------------------------	----------

#### 8. Skenario Pengujian Melihat Matakuliah

Pengujian melihat matakuliah dilakukan untuk memastikan bahwa mahasiswa dapat melihat daftar-daftar matakuliah. Hasil pengujian dapat dilihat pada Tabel 18 berikut.

**Tabel 18. Skenario Pengujian Melihat Matakuliah**

<b>Identifikasi</b>	BU-08		
<b>Nama Butir Uji</b>	Test Butir Uji Melihat Matakuliah		
<b>Tujuan</b>	Menguji fungsi Melihat Matakuliah		
<b>Deskripsi</b>	Fungsi ini digunakan mahasiswa untuk melihat materi perkuliahan yang telah di <i>upload</i> oleh dosen pengampu.		
<b>Kondisi Awal</b>	Mahasiswa telah login dan berada di halaman home		
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Memilih menu “Perkuliahannya”</li> <li>2. Memilih “Prodi dan Jenjang dan Tahun Kurikulum”</li> <li>3. Memilih tombol “Cari”</li> </ol>		
<b>Kriteria Evaluasi Hasil</b>	Mahasiswa berhasil melihat matakuliah		
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Memilih menu “Perkuliahannya”	Aplikasi menampilkan halaman perkuliahan Aplikasi menampilkan pilihan	Sesuai dengan yang diharapkan	Diterima

Memilih prodi dan jenjang dan tahun kurikulum	Aplikasi menampilkan daftar-daftar matakuliah	Sesuai dengan yang diharapkan	Diterima
Memilih tombol “Cari”		Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa tidak memilih prodi dan jenjang	Aplikasi menampilkan pesan error	Sesuai dengan yang diharapkan	Diterima

#### 9. Skenario Pengujian Unduh Modul

Pengujian unduh modul dilakukan untuk memastikan bahwa mahasiswa dapat mengunduh modul perkuliahan. Hasil pengujian dapat dilihat pada Tabel 19 berikut.

**Tabel 19. Skenario Pengujian Unduh Modul**

<b>Identifikasi</b>	BU-09
<b>Nama Butir Uji</b>	Test Butir Uji Unduh Modul
<b>Tujuan</b>	Menguji fungsi Unduh Modul
<b>Deskripsi</b>	Fungsi ini digunakan oleh mahasiswa untuk mengunduh materi atau modul perkuliahan yang telah diunggah oleh dosen pengampu.
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> dan berada di halaman perkuliahan
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih “Prodi dan Jenjang dan Tahun Kukulum”</li> <li>2. Mahasiswa memilih tombol “Cari”</li> <li>3. Mahasiswa memilih matakuliah</li> <li>4. Mahasiswa memilih materi atau praktikum</li> <li>5. Mahasiswa memilih sesi</li> </ol>

6. Mahasiswa meng-klik modul			
<b>Kriteria Evaluasi Hasil</b>			
Mahasiswa berhasil mengunduh modul			
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa memilih prodi dan jenjang dan mata kuliah	Aplikasi menampilkan pilihan	Sesuai dengan yang diharapkan	Diterima
Mahasiswa memilih tombol “Cari”	Aplikasi menampilkan daftar-daftar matakuliah	Sesuai dengan yang diharapkan	Diterima
Mahasiswa memilih matakuliah	Aplikasi menampilkan materi dan praktikum	Sesuai dengan yang diharapkan	Diterima
Mahasiswa memilih materi atau praktikum	Aplikasi menampilkan modul per sesi	Sesuai dengan yang diharapkan	Diterima
Mahasiswa memilih sesi	Aplikasi menampilkan modul	Sesuai dengan yang diharapkan	Diterima
Mahasiswa meng-klik modul	Aplikasi mengunduh modul	Sesuai dengan yang diharapkan	Diterima

## 10. Skenario Pengujian Melihat Jadwal

Pengujian melihat jadwal dilakukan untuk memastikan bahwa mahasiswa dapat melihat jadwal. Hasil pengujian dapat dilihat pada Tabel 20 berikut

**Tabel 20. Skenario Pengujian Melihat Jadwal**

<b>Identifikasi</b>	BU-10		
<b>Nama Butir Uji</b>	Test Butir Uji Melihat Jadwal		
<b>Tujuan</b>	Menguji Fungsi Melihat Jadwal		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melihat jadwal dari perkuliahan yang mereka ikuti selama 1 hari tersebut		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi		
<b>Skenario Uji</b>			
1.	Memilih menu Jadwal		
<b>Kriteria Evaluasi Hasil</b>			
Mahasiswa berhasil melihat jadwal			
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mahasiswa memilih menu jadwal	Aplikasi menampilkan list-list jadwal	Sesuai dengan yang diharapkan	Diterima

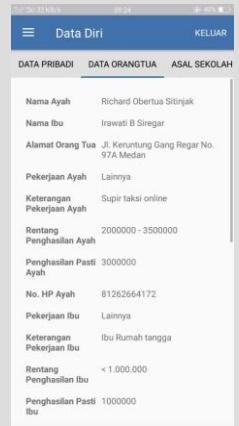
### 11. Skenario Pengujian Melihat Data Diri

Pengujian melihat data diri dilakukan untuk memastikan mahasiswa dapat melihat data diri mereka. Hasil pengujian dapat dilakukan pada Tabel 21 berikut.

**Tabel 21. Skenario Pengujian Melihat Data Diri**

<b>Identifikasi</b>	BU-11
<b>Nama Butir Uji</b>	Test Butir Uji Melihat Data Diri
<b>Tujuan</b>	Menguji fungsi Melihat Data Diri
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melihat data diri.

<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi		
<b>Skenario Uji</b>			
	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih menu Data Diri</li> <li>2. Mahasiswa melihat data akademis</li> <li>3. Mahasiswa melihat data pribadi</li> <li>4. Mahasiswa melihat data orangtua</li> <li>5. Mahasiswa melihat data asal sekolah</li> <li>6. Mahasiswa melihat data keasramaan</li> </ol>		
<b>Kriteria Evaluasi Hasil</b>			
Mahasiswa berhasil melihat data diri			
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa melihat data akademis	Aplikasi menampilkan data akademis	<p>Sesuai dengan yang diharapkan</p>	Diterima
Mahasiswa melihat data pribadi	Aplikasi menampilkan data pribadi	<p>Sesuai dengan yang diharapkan</p>	Diterima

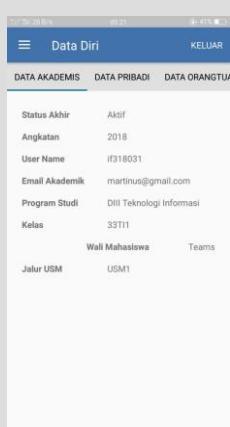
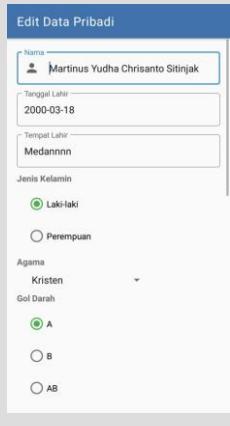
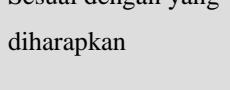
Mahasiswa melihat data orangtua	Aplikasi menampilkan data orangtua		Diterima
Mahasiswa melihat data sekolah	Aplikasi menampilkan data sekolah		Diterima
Mahasiswa melihat data keasramaan	Aplikasi menampilkan data keasramaan		Diterima

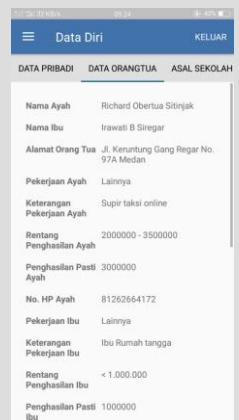
## 12. Skenario Pengujian Edit Data Diri

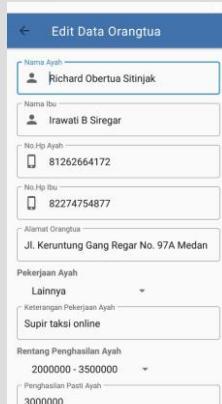
Pengujian edit data diri dilakukan untuk memastikan bahwa mahasiswa dapat melakukan edit data diri. Hasil pengujian dapat dilihat pada Tabel 22 berikut

**Tabel 22. Skenario Pengujian Edit Data Diri**

<b>Identifikasi</b>	BU-12		
<b>Nama Butir Uji</b>	Test Butir Uji Edit Data Diri		
<b>Tujuan</b>	Menguji fungsi Edit Data Diri		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melakukan pengeditan data diri terhadap data yang terdaftar yang mereka miliki.		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan ada data diri mahasiswa yang salah atau perlu diubah		
<b>Skenario Uji</b>	<ol style="list-style-type: none"><li>1. Mahasiswa memilih menu Data Diri</li><li>2. Mahasiswa memilih data pribadi</li><li>3. Mahasiswa memilih tombol “Edit”</li><li>4. Mahasiswa mengubah data pribadi</li><li>5. Mahasiswa memilih tombol “Save”</li><li>6. Mahasiswa memilih data orangtua</li><li>7. Mahasiswa memilih tombol “Edit”</li><li>8. Mahasiswa mengubah data orangtua</li><li>9. Mahasiswa memilih tombol “Save”</li><li>10. Mahasiswa memilih data sekolah</li><li>11. Mahasiswa memilih tombol “Edit”</li><li>12. Mahasiswa mengubah data sekolah</li><li>13. Mahasiswa memilih tombol “Save”</li></ol>		
<b>Kriteria Evaluasi Hasil</b>			
	Mahasiswa berhasil melakukan edit data diri		
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>

Mahasiswa memilih menu Data Diri	Aplikasi menampilkan halaman data diri	Sesuai dengan yang diharapkan 	Diterima
Mahasiswa memilih data pribadi	Aplikasi menampilkan data pribadi	Sesuai dengan yang diharapkan 	Diterima
Mahasiswa memilih tombol “Edit”	Aplikasi menampilkan form edit	Sesuai dengan yang diharapkan 	Diterima
Mahasiswa mengubah data pribadi	Aplikasi mengubah data pribadi pada form	Sesuai dengan yang diharapkan 	Diterima

		Sesuai dengan yang diharapkan	
Mahasiswa memilih tombol “Save”	Aplikasi menyimpan data pribadi	Sesuai dengan yang diharapkan	Diterima
Mahasiswa memilih data orangtua	Aplikasi menampilkan data orangtua		Diterima
Mahasiswa memilih tombol “Edit”	Aplikasi menampilkan form edit		Diterima
Mahasiswa mengubah data orangtua	Aplikasi mengubah data orangtua pada form		Diterima
Mahasiswa memilih tombol “Save”	Aplikasi menyimpan data orangtua	Sesuai dengan yang diharapkan	Diterima

		 <p>Sesuai dengan yang diharapkan</p>  <p>Diterima</p>
Mahasiswa memilih data sekolah	Aplikasi menampilkan data sekolah	 <p>Sesuai dengan yang diharapkan</p> <p>Diterima</p>
Mahasiswa memilih tombol “Edit”	Aplikasi menampilkan form edit	 <p>Sesuai dengan yang diharapkan</p> <p>Diterima</p>
Mahasiswa mengubah data sekolah	Aplikasi mengubah data sekolah pada form	 <p>Diterima</p>

Mahasiswa memilih tombol “Save”	Aplikasi menyimpan data sekolah	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa mengubah data namun tidak memilih tombol “Save”	Aplikasi tidak menyimpan data	Sesuai dengan yang diharapkan	Diterima

### 13. Skenario Pengujian Melihat Kuesioner

Pengujian melihat kuesioner dilakukan untuk memastikan bahwa mahasiswa dapat melihat kuesioner. Hasil pengujian dapat dilihat pada Tabel 23 berikut.

**Tabel 23. Skenario Pengujian Melihat Kuesioner**

<b>Identifikasi</b>	BU-14
<b>Nama Butir Uji</b>	Test Butir Uji Melihat Kuesioner
<b>Tujuan</b>	Menguji fungsi Melihat Kuesioner
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melihat kuesioner yang diberikan oleh dosen, keasramaan ataupun staff IT Del yang lainnya
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan berada di halaman home
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih menu Kuesioner</li> </ol>
<b>Kriteria Evaluasi Hasil</b>	

Mahasiswa berhasil melihat data kuesioner			
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa memilih menu Kuesioner	Aplikasi menampilkan list-list kuesioner	Sesuai dengan yang diharapkan 	Diterima

#### 14. Skenario Pengujian Mengisi Kuesioner

Pengujian mengisi kuesioner dilakukan untuk memastikan bahwa mahasiswa dapat melakukan pengisian kuesioner. Hasil pengujian dapat dilihat pada Tabel 24 berikut.

**Tabel 24. Skenario Pengujian Mengisi Kuesioner**

<b>Identifikasi</b>	BU-14
<b>Nama Butir Uji</b>	Test Butir Uji Mengisi Kuesioner
<b>Tujuan</b>	Menguji fungsi Mengisi Kuesioner
<b>Deskripsi</b>	Fungsi ini digunakan oleh mahasiswa untuk mengedit atau mengisi kuesioner yang ditujukan terhadap mahasiswa.
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan berada di halaman kuesioner
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih kuesioner</li> <li>2. Mahasiswa mengisi kuesioner</li> </ol>

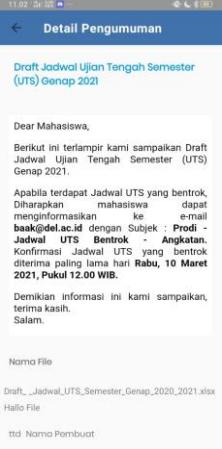
3. Mahasiswa memilih tombol “Kirim”			
<b>Kriteria Evaluasi Hasil</b>			
Mahasiswa berhasil mengisi kuesioner			
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa memilih kuesioner	Aplikasi menampilkan kuesioner	Sesuai dengan yang diharapkan  	Diterima
Mahasiswa mengisi kuesioner	Aplikasi menyimpan tanggapan	Sesuai dengan yang diharapkan	Diterima
Mahasiswa memilih tombol “Kirim”	Aplikasi menyimpan tanggapan dan menampilkan daftar-daftar kuesioner	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Pengujian (Data Tidak Normal)</b>			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa tidak memilih tombol “Kirim” pada saat sudah mengisi kuesioner	Aplikasi tidak akan menyimpan tanggapan	Sesuai dengan yang diharapkan	Diterima

## 15. Skenario Pengujian Melihat Pengumuman

Pengujian melihat pengumuman dilakukan untuk memastikan bahwa mahasiswa dapat melihat pengumuman. Hasil pengujian dapat dilihat pada Tabel 25 berikut.

**Tabel 25. Skenario Pengujian Melihat Pengumuman**

<b>Identifikasi</b>	BU-16		
<b>Nama Butir Uji</b>	Test Butir Uji Melihat Pengumuman		
<b>Tujuan</b>	Menguji fungsi Melihat Pengumuman		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melihat pengumuman yang ada.		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi		
<b>Skenario Uji</b>	<ol style="list-style-type: none"><li>1. Mahasiswa memilih menu Pengumuman</li><li>2. Memilih pengumuman yang akan dilihat</li></ol>		
<b>Kriteria Evaluasi Hasil</b>	Mahasiswa berhasil melihat pengumuman		
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mahasiswa memilih menu Pengumuman	Aplikasi menampilkan pengumuman	Sesuai dengan yang diharapkan 	Diterima

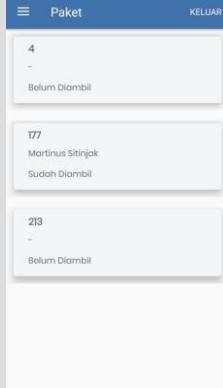
Memilih pengumuman yang akan dilihat	Aplikasi menampilkan detail pengumuman	Sesuai dengan yang diharapkan  	Diterima
--------------------------------------	----------------------------------------	-------------------------------------------------------------------------------------------------------------------------	----------

## 16. Skenario Pengujian Melihat Paket

Pengujian melihat paket dilakukan untuk memastikan bahwa mahasiswa dapat melihat paket. Hasil pengujian dapat dilihat pada Tabel 26 berikut

**Tabel 26. Skenario Pengujian Melihat Paket**

<b>Identifikasi</b>	BU-17
<b>Nama Butir Uji</b>	Test Butir Uji Melihat Paket
<b>Tujuan</b>	Menguji fungsi Melihat Paket
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melihat paket yang dapat di post satpam
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Mahasiswa memilih menu Paket</li> <li>2. Mahasiswa memilih paket yang akan dilihat</li> </ol>
<b>Kriteria Evaluasi Hasil</b>	Mahasiswa berhasil melihat paket
<b>Kasus dan Hasil Pengujian (Data Normal)</b>	

Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa memilih menu Paket	Aplikasi menampilkan halaman paket	Sesuai dengan yang diharapkan 	Diterima
Mahasiswa memilih paket yang akan dilihat	Aplikasi menampilkan detail paket	Sesuai dengan yang diharapkan 	Diterima

#### 17. Skenario Pengujian Add Kartu Rencana Studi (KRS)

Pengujian add KRS dilakukan untuk memastikan bahwa mahasiswa dapat melakukan add KRS. Hasil pengujian dapat dilihat pada Tabel 27 berikut.

**Tabel 27. Skenario Pengujian Add Request KRS**

<b>Identifikasi</b>	BU-17
<b>Nama Butir Uji</b>	Test Butir Uji Add Kartu Rencana Studi
<b>Tujuan</b>	Menguji fungsi Add Kartu Rencana Studi
<b>Deskripsi</b>	Fungsi ini digunakan oleh mahasiswa untuk menambahkan KRS sesuai

	matakuliah yang diambil dalam satu semester.		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan berada di <i>home</i>		
<b>Skenario Uji</b>			
<ol style="list-style-type: none"> <li>1. Mahasiswa memilih menu “Kartu Rencana Studi”</li> <li>2. Mahasiswa memilih matakuliah pada form</li> <li>3. Mahasiswa memilih tombol “Kirim”</li> </ol>			
<b>Kriteria Evaluasi Hasil</b>			
Mahasiswa berhasil <i>Add</i> Kartu Rencana Studi			
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mahasiswa memilih menu “Kartu Rencana Studi”	Aplikasi menampilkan halaman KRS	Tidak sesuai dengan yang diharapkan	Belum di <i>consume</i>
Mahasiswa memilih matakuliah pada form	Aplikasi menampilkan form KRS	Tidak sesuai dengan yang diharapkan	Belum di <i>consume</i>
Mahasiswa memilih tombol “kirim”	Aplikasi mengirim <i>request</i> KRS	Tidak sesuai dengan yang diharapkan	Belum di <i>consume</i>

#### 18. Skenario Pengujian Edit Kartu Rencana Studi (KRS)

Pengujian edit KRS dilakukan untuk memastikan bahwa mahasiswa dapat melakukan edit KRS. Hasil pengujian dapat dilihat pada Tabel 28 berikut.

**Tabel 28. Skenario Pengujian Edit Request KRS**

<b>Identifikasi</b>	BU-18
<b>Nama Butir Uji</b>	Test Butir Uji Edit Kartu Rencana Studi (KRS)

<b>Tujuan</b>	Menguji fungsi Edit Kartu Rencana Studi (KRS)		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melakukan pengedit terhadap data Kartu Rencana Studi (KRS) yang sebelumnya ditambahkan.		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi dan telah melakukan <i>add KRS</i> dan data yang ditambahkan belum mendapatkan <i>approval</i> dari dosen wali mahasiswa tersebut.		
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Mahasiswa telah melakukan <i>add KRS</i></li> <li>2. Mahasiswa memilih menu “Kartu Rencana Studi”</li> <li>3. Mahasiswa memilih menu edit KRS</li> <li>4. Mahasiswa melakukan pengubahan terhadap KRS</li> <li>5. Mahasiswa memilih tombol “save”</li> </ol>		
<b>Kriteria Evaluasi Hasil</b>			
	Mahasiswa berhasil Edit Kartu Rencana Studi		
	<b>Kasus dan Hasil Pengujian (Data Normal)</b>		
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mahasiswa telah melakukan add KRS	Aplikasi menampilkan halaman KRS yang telah di add sebelumnya	Tidak sesuai dengan yang diharapkan	Belum di <i>consume</i>
Mahasiswa memilih menu “Kartu Rencana Studi”	Aplikasi menampilkan halaman KRS yang menunggu persetujuan dosen wali	Tidak sesuai dengan yang diharapkan	Belum di <i>consume</i>
Mahasiswa memilih menu edit KRS	Aplikasi menampilkan menu edit KRS	Tidak sesuai dengan yang diharapkan	Belum di <i>consume</i>
Mahasiswa melakukan pengubahan terhadap	Aplikasi menampilkan form KRS yang akan diubah	Tidak sesuai dengan yang diharapkan	Belum di <i>consume</i>

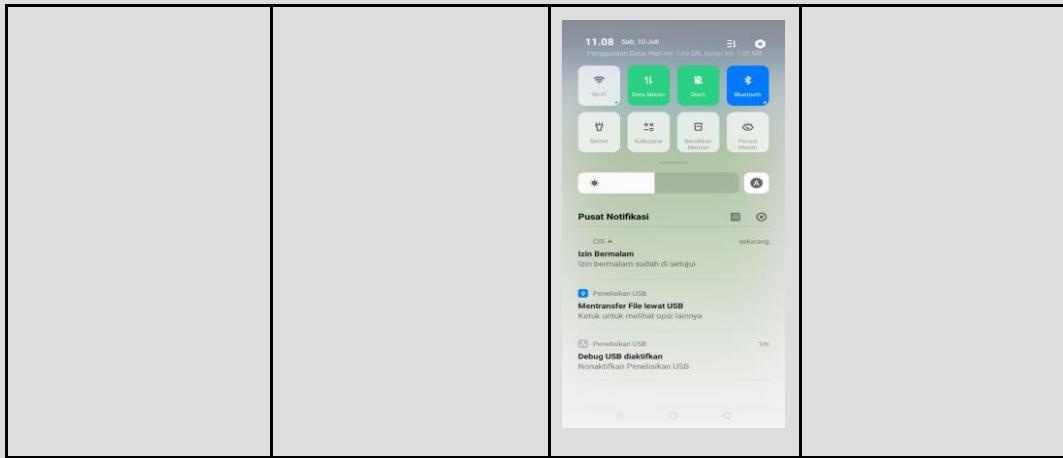
KRS  Mahasiswa memilih tombol “save”	Aplikasi menyimpan perubahan data	diharapkan  Tidak sesuai dengan yang diharapkan	Belum di <i>consume</i>
--------------------------------------------	-----------------------------------	-------------------------------------------------------	-------------------------

#### 19. Skenario Pengujian Melihat Notifikasi

Pengujian melihat notifikasi dilakukan untuk memastikan bahwa mahasiswa dapat melihat notifikasi. Hasil pengujian dapat dilihat pada Tabel 29 berikut.

**Tabel 29. Skenario Pengujian Melihat Notifikasi**

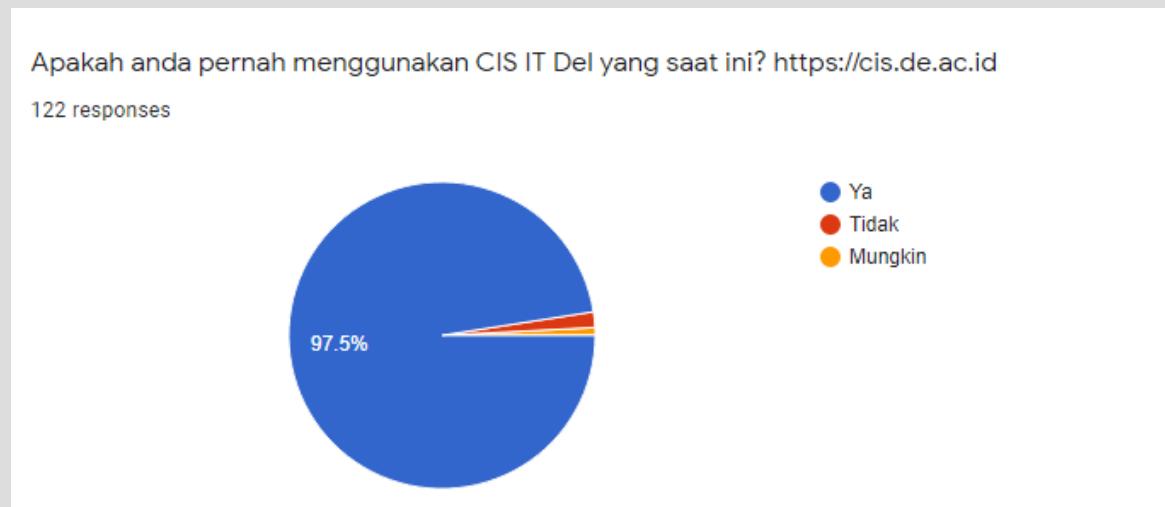
<b>Identifikasi</b>	BU-20		
<b>Nama Butir Uji</b>	Test Butir Uji Melihat Notifikasi		
<b>Tujuan</b>	Menguji fungsi Melihat Notifikasi		
<b>Deskripsi</b>	Fungsi ini akan digunakan oleh mahasiswa untuk melihat notifikasi yang masuk		
<b>Kondisi Awal</b>	Mahasiswa telah <i>login</i> ke dalam aplikasi		
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Mahasiswa melakukan <i>request</i> IB</li> <li>2. Mahasiswa mendapat notifikasi <i>approve</i> IB</li> </ol>		
<b>Kriteria Evaluasi Hasil</b>	Mahasiswa berhasil melihat notifikasi		
<b>Kasus dan Hasil Pengujian (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mahasiswa melakukan <i>request</i> IB	Aplikasi berhasil mengirim <i>request</i> IB	Sesuai dengan yang diharapkan	Diterima
Mahasiswa mendapat <i>approve</i> IB	Aplikasi menerima notifikasi	Sesuai dengan yang diharapkan	Diterima



### 5.2.1.2 Pengujian *Beta Testing* pada Aplikasi

Pada pengujian ini mahasiswa menggunakan dan menguji aplikasi CIS berbasis *mobile* secara langsung. Selain itu pada pengujian ini akan diberikan sebuah kuesioner yang akan menentukan pemahaman dan *feedback* mahasiswa terhadap aplikasi. Adapun kuesioner yang diberikan kepada mahasiswa adalah sebagai berikut.

1. Apakah anda pernah menggunakan CIS IT Del yang saat ini? <https://cis.del.ac.id>



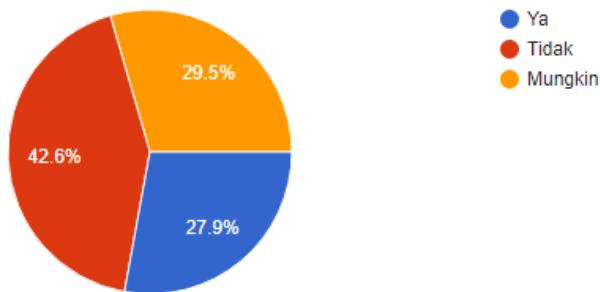
Gambar 72. Diagram Pertanyaan 1

Berdasarkan hasil kuesioner dapat dilihat bahwa dari 122 responden mahasiswa IT Del terdapat sebesar 97.5% mahasiswa sudah pernah menggunakan CIS berbasis *website* yang saat ini.

2. Saat ini, IT Del hanya menyediakan CIS berbasis *website*, menurut anda apakah anda mengalami kesulitan dalam menggunakan CIS yang saat ini?

Saat ini, IT Del hanya menyediakan CIS berbasis website, menurut anda apakah anda mengalami kesulitan dalam menggunakan CIS yang saat ini?

122 responses



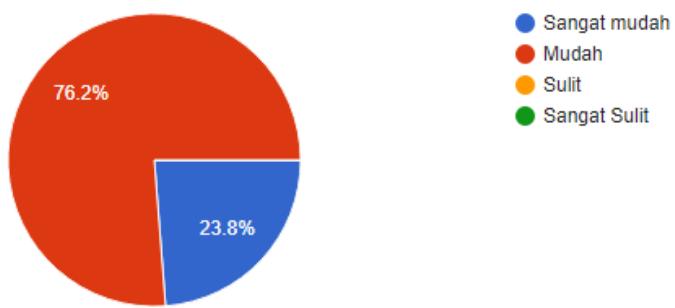
Gambar 73. Diagram Pertanyaan 2

Berdasarkan hasil kuesioner dari 122 responden mahasiswa IT Del dapat dilihat bahwa 42.6% mahasiswa tidak mengalami kesulitan dalam menggunakan CIS berbasis *website*. Namun sebesar 27.9% dan 29.5% mahasiswa mengalami kesulitan. Maka dapat disimpulkan bahwa mahasiswa mengalami kesulitan dalam menggunakan CIS *website* saat ini.

3. Berdasarkan video aplikasi CIS berbasis *mobile* yang telah dibagikan, Menurut anda apakah aplikasi mudah digunakan?

Berdasarkan video aplikasi CIS berbasis mobile yang telah dibagikan, Menurut anda apakah aplikasi mudah digunakan?

122 responses



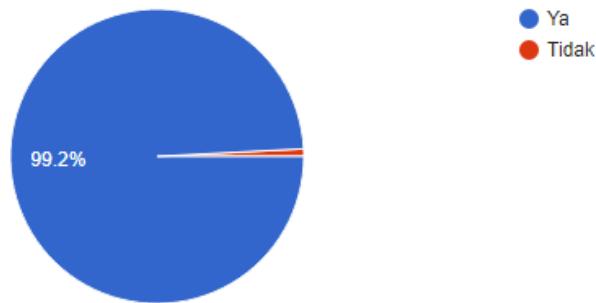
Gambar 74. Diagram Pertanyaan 3

Berdasarkan hasil kuesioner dari 122 responden mahasiswa IT Del diatas dapat dilihat bahwa 76.2% mahasiswa mudah menggunakan aplikasi CIS *mobile* yang dikembangkan. Dan 23.8% mahasiswa memilih sangat mudah. Maka dapat disimpulkan bahwa aplikasi CIS *mobile* mudah digunakan.

4. Apakah dengan adanya fitur notifikasi ke *mobile phone* rekan-rekan akan membantu rekan-rekan dalam mendapatkan informasi?

Apakah dengan adanya fitur notifikasi ke mobile phone rekan-rekan akan membantu rekan-rekan sekalian dalam mendapatkan informasi?

122 responses



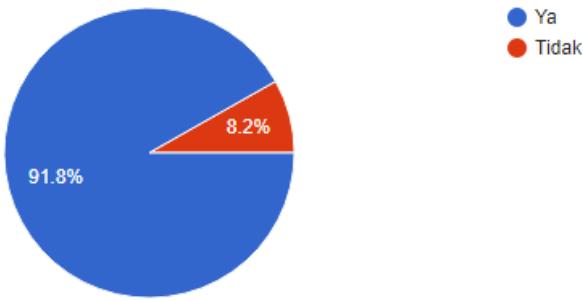
Gambar 75. Diagram Pertanyaan 4

Berdasarkan hasil kuesioner dari 122 responden mahasiswa IT Del dapat dilihat bahwa 99.2% mahasiswa memilih ya dan 0.8% mahasiswa memilih tidak. Maka dapat disimpulkan bahwa pembuatan fitur notifikasi pada aplikasi CIS berbasis *mobile* sangat membantu mahasiswa dalam mendapatkan informasi.

5. Bagaimana dengan fitur logout manual, apakah fitur ini juga membantu rekan-rekan dalam mengakses CIS?

Bagaimana dengan fitur logout manual, apakah fitur ini juga membantu rekan-rekan dalam mengakses CIS?

122 responses

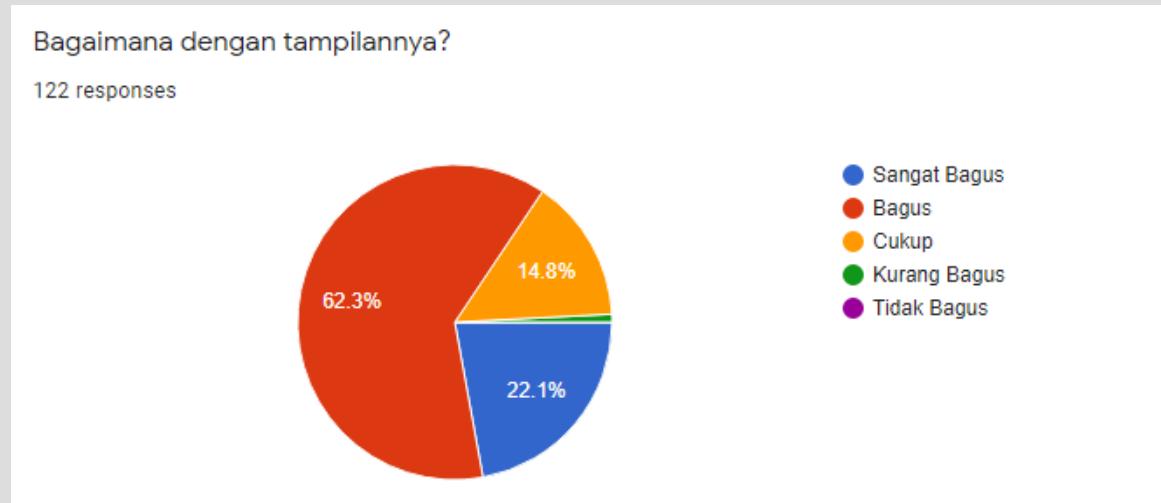


Gambar 76. Diagram Pertanyaan 5

Berdasarkan hasil kuesioner dari 122 responden mahasiswa IT Del dapat dilihat bahwa 91.8% mahasiswa memilih ya dan 8.2% mahasiswa memilih tidak. Maka dapat disimpulkan bahwa penambahan fitur logout manual pada aplikasi CIS

berbasis *mobile* sangat membantu mahasiswa dalam mengakses CIS berbasis *mobile*.

6. Bagaimana dengan tampilannya?



Gambar 77. Diagram Pertanyaan 6

Berdasarkan hasil kuesioner dari 122 responden mahasiswa IT Del dapat dilihat bahwa 62.3% memilih bagus, 22.1% memilih sangat bagus dan 14.8% memilih cukup. Maka dapat disimpulkan bahwa mahasiswa menyukai tampilannya.

7. Apakah saran atau *feedback* anda untuk aplikasi ini kedepannya?

Apakah saran atau feedback anda terhadap aplikasi ini kedepannya?

122 responses

Semoga aplikasi ini dapat segera digunakan user yang membutuhkan mudah mudahan app nya bisa diakses untuk pengguna iOS juga yaa, Tetap Semangat!

untuk layoutnya diperbaiki

Sebaiknya surat IB yg sudah dikonfirmasi tidak bisa diedit lagi.

Mungkin bisa di tambahkan untuk setiap pemberitahuan yang di post di cis, maka akan ter notifikasi ke pengguna, agar dapat melihat informasi/pemberitahuan terupdate...

saran aplikasi tidak dpt dibuka di dua pc agar tetap mejaga keamanan dan kenyamanan pengguna.

color yg abu2 yg paling atas samakan dgn warna tollbar

Gambar 78. Diagram Pertanyaan 7

### 5.2.2 Pengujian Application Programming Interface (API)

Pengujian API merupakan pengujian yang dilakukan untuk menguji fungsionalitas yang terdapat dalam API. Pengujian API ini dilakukan menggunakan aplikasi yang bernama Postman. Postman adalah sebuah aplikasi yang berfungsi sebagai *Rest Client* untuk uji coba API. Terdapat beberapa metode proses testing dalam postman yaitu:

1. *GET*, berfungsi untuk mendapatkan data yang terdapat pada sistem.
2. *POST*, berfungsi untuk memberikan data baru kedalam sistem.
3. *PATCH*, berfungsi untuk mengubah sebagian data pada sistem.
4. *PUT*, berfungsi untuk merubah keseluruhan data pada sistem.
5. *DELETE*, berfungsi untuk menghapus data yang ada pada sistem.

Berikut merupakan *testing unit* API pada Postman.

1. Pengujian API *login*

Pengujian API *login* pada postman menggunakan metode POST dengan url pengujian API *login* yaitu <https://sotardok.web.id/backend/api/login>. Adapun hasil pengujian API *login* pada postman dapat dilihat pada Gambar 79 berikut.

The screenshot shows the Postman interface for a 'Login' request. The 'Body' tab is selected, showing two form-data fields: 'username' with value 'root' and 'password' with value 'r00t'. The response status is 200 OK, and the JSON response body is displayed as:

```
1 [ { "status": 200, "data": { "user": { "dimId": "1", "name": "Martinus Yudha C Sitinjak", "userId": "1" }, "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9. eyJpc3MiOiJodHRwczpcL1wvc290YXJkb2sud2ViLm1kXC9iYWNrZW5kXC9hcG1cL2xvZ2luTiwiWF0IjoxNjI2NDA0MTcwLCJuYmYiOjE2MjT01DeT15UjJRmMiLCJzdWIiOjEsInBydiI6IjRjMDA0MjdmYWNiMjI0MjQ5ZjNmOTY5MDUzMjJhMGZlY2N1NGIi.fQ. UYB4s1HEjMSSp10vKKhEe94dasgfH5IWpC_CirbzAa0" } }
```

Gambar 79. Pengujian API login

Pada gambar diatas terdapat *request* dan *response* sebuah data yang terdapat pada sistem. Dimana *request* pada *body* berupa *username* dan *password*. Selain itu,

untuk *response* nya menampilkan seluruh data mahasiswa dalam bentuk JSON.

Dalam hal ini, data mengikuti dengan data yang terdapat dalam database.

## 2. Pengujian API *logout*

Pengujian API *logout* pada postman menggunakan metode GET dengan url pengujian API *logout* yaitu <https://sotardok.web.id/backend/api/logout>. Adapun hasil pengujian API *logout* pada postman dapat dilihat pada Gambar 80 berikut.

The screenshot shows the Postman interface for a 'Logout' API test. The request method is 'GET' and the URL is 'https://sotardok.web.id/backend/api/logout'. The 'Headers' tab is selected, showing an 'Authorization' header with the value 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiO...'. The 'Body' tab is selected, showing a JSON response with the message 'Successfully logged out'. The status bar at the bottom indicates a 200 OK status and a response time of 2.16 s.

Gambar 80. Pengujian API *logout*

## 3. Pengujian API menampilkan data user

Pengujian API menampilkan data user pada postman menggunakan metode GET dengan url pengujian API menampilkan data user yaitu <https://sotardok.web.id/backend/api/user>. Adapun hasil pengujian API menampilkan data user pada postman dapat dilihat pada Gambar 81 berikut.

The screenshot shows a Postman collection named 'getUser'. A GET request is made to <https://sotardok.web.id/backend/api/user>. In the Headers tab, there is a checked 'Authorization' field with the value 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...'. The Body tab shows a JSON response with the following structure:

```

1  {
2      "status": 200,
3      "data": {
4          "dim_id": 1,
5          "nim": "11318031",
6          "no_usm": "",
7          "jalur": "USM1",
8          "user_name": "if318031",
9          "kbk_id": null,
10         "ref_kbk_id": "1",
11         "kpt_prodi": null,
12         "id_kur": "0",
13         "tahun_kurikulum_id": null,
14         "-----"
15     }
  
```

Gambar 81. Pengujian API menampilkan data *user*

#### 4. Pengujian API *request Izin Bermalam (IB)*

Pengujian API *request IB* pada postman menggunakan metode POST dengan url pengujian API *request IB* yaitu <https://sotardok.web.id/backend/api/izinBermalam/<user id>>. Adapun hasil pengujian API *request IB* pada postman dapat dilihat pada Gambar 82 berikut.

The screenshot shows a Postman collection named 'MaketzinBermalam'. A POST request is made to <https://sotardok.web.id/backend/api/izinBermalam/1>. In the Body tab, the 'form-data' option is selected, and four parameters are added: 'rencana\_berangkat', 'rencana\_kembali', 'desc', and 'tujuan'. The Body response shows the following JSON:

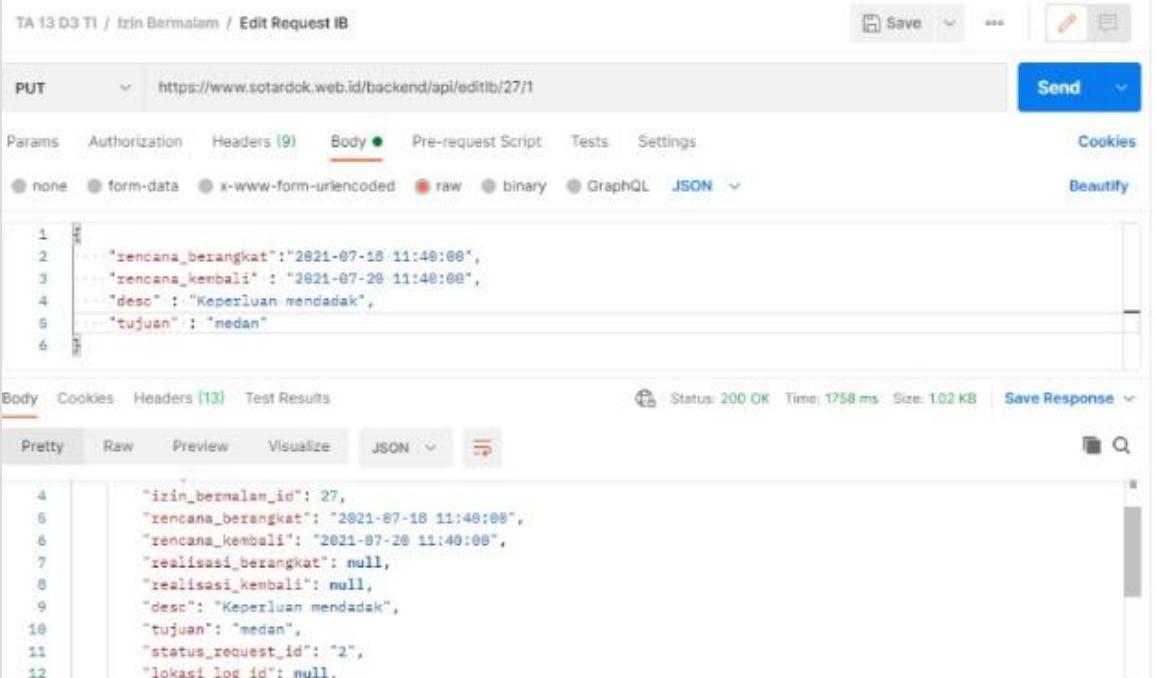
```

1  {
2      "status": 200,
3      "data": null,
4      "message": "Data berhasil disimpan"
  
```

Gambar 82. Pengujian API *request IB*

## 5. Pengujian API edit request Izin Bermalam (IB)

Pengujian API edit *request* IB pada postman menggunakan metode PUT dengan url pengujian API edit *request* IB yaitu <https://sotardok.web.id/backend/api/editIb/{izinId}/{userId}>. Adapun hasil pengujian API edit *request* IB pada postman dapat dilihat pada Gambar 83 berikut.



```
1
2
3
4
5
6
7
8
9
10
11
12
```

```
1
2
3
4
5
6
7
8
9
10
11
12
```

Body: `{"izin_bermalam_id": 27, "rencana_berangkat": "2021-07-18 11:40:00", "rencana_kembali": "2021-07-20 11:40:00", "realisasi_berangkat": null, "realisasi_kembali": null, "desc": "Keperluan mendadak", "tujuan": "medan", "status_request_id": "2", "lokasi_log_id": null, "lokasi_kembali_id": null}`

Gambar 83. Pengujian API edit request IB

## 6. Pengujian API cancel request Izin Bermalam (IB)

Pengujian API cancel *request* IB pada postman menggunakan metode PUT dengan url pengujian API cancel *request* IB yaitu <https://sotardok.web.id/backend/api/batalib>. Adapun hasil pengujian API cancel *request* IB pada postman dapat dilihat pada Gambar 84 berikut.

The screenshot shows a POST request to `https://www.sotardok.web.id/backend/api/batallb`. The 'Body' tab is selected, showing a key-value pair `idizin: 1`. The response status is 200 OK, and the message is "Izin sudah di batalkan".

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> idizin	1	

```

1 [
2   "status": 200,
3   "data": null,
4   "message": "Izin sudah di batalkan"
5 ]

```

Gambar 84. Pengujian API cancel request IB

#### 7. Pengujian API menampilkan daftar-daftar Izin Bermalam (IB)

Pengujian API menampilkan daftar-daftar IB pada postman menggunakan metode GET dengan url pengujian API menampilkan daftar-daftar IB yaitu `https://sotardok.web.id/backend/api/izinBermalam/<user id>`. Adapun hasil pengujian API menampilkan daftar-daftar IB pada postman dapat dilihat pada Gambar 85 berikut.

The screenshot shows a GET request to `https://sotardok.web.id/backend/api/izinBermalam/1`. The 'Headers' tab is selected, showing an Authorization header with a Bearer token. The response status is 200 OK, and it returns a JSON array of two objects representing IB details.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...	

```

1 [
2   {
3     "status": 200,
4     "data": [
5       {
6         "id_izin": "1",
7         "request_id": "4",
8         "disetujui_oleh": null,
9         "keperluan_ib": "Ketemu Orangtua",
10        "tujuan": "Medan"
11      },
12      {
13        "id_izin": "2",
14        "request_id": "2",
15        "disetujui_oleh": null,
16      }
17    ]
18  }
19 ]

```

Gambar 85. Pengujian API menampilkan daftar IB

#### 8. Pengujian API detail Izin Bermalam (IB)

Pengujian API detail IB pada postman menggunakan metode GET dengan url pengujian API detail IB yaitu [https://sotardok.web.id/backend/api/getIzin/<user id>/<izin\\_bermalam\\_id>](https://sotardok.web.id/backend/api/getIzin/<user id>/<izin_bermalam_id>). Adapun hasil pengujian API detail IB pada postman dapat dilihat pada Gambar 86 berikut.

The screenshot shows the Postman interface for a GET request to <https://sotardok.web.id/backend/api/getIzin/1/1>. The 'Headers' tab is selected, showing an Authorization header with a Bearer token. The 'Body' tab displays a JSON response with status 200 OK and a response time of 1323 ms. The response body is a JSON object with fields like status, data (containing student information), and request\_id.

KEY	VALUE	DESCRIPTION
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...	

```
1 {  
2   "status": 200,  
3   "data": {  
4     "nama_mahasiswa": "Martinus Yudha C Sitinjak",  
5     "nim_mahasiswa": "11318031",  
6     "rencana_berangkat": "2021-03-18 20:00:00",  
7     "rencana_kembali": "2021-03-20 20:00:00",  
8     "realisasi_berangkat": null,  
9     "realisasi_kembali": null,  
10    "keperluan_ib": "Ketemu Orangtua",  
11    "tujuan": "Medan",  
12    "request_id": "4",  
13  }  
14}
```

Gambar 86. Pengujian API detail IB

#### 9. Pengujian API *approve* Izin Bermalam (IB)

Pengujian API *approve* IB pada postman menggunakan metode POST dengan url pengujian API *approve* IB yaitu

<https://sotardok.web.id/backend/api/approveIB/izinId>. Adapun hasil pengujian API *approve* IB pada postman dapat dilihat pada Gambar 87 berikut.

Approve IB

POST https://www.sotardok.web.id/backend/api/approveIB/3

Body (8)

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> status	2	
<input checked="" type="checkbox"/> user_id	1	

Pretty Raw Preview Visualize JSON

```

1 [
2   "status": 200,
3   "data": null,
4   "message": "Data sudah di update"
5 ]

```

Status: 200 OK Time: 983 ms

Gambar 87. Pengujian API approve IB

#### 10. Pengujian API *request Izin Keluar (IK)*

Pengujian API *request IK* pada postman menggunakan metode POST dengan url pengujian API *request IK* yaitu

<https://sotardok.web.id/backend/api/izinKeluar/{dimId}>. Adapun hasil pengujian API *request IK* pada postman dapat dilihat pada Gambar 88 berikut.

Make Izin Keluar

POST https://www.sotardok.web.id/backend/api/izinKeluar/1

Body (9)

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> rencana_berangkat	2021-03-18 20:00:00	
<input checked="" type="checkbox"/> rencana_kembali	2021-03-20 20:00:00	
<input checked="" type="checkbox"/> keperluank	Mendadak	

Pretty Raw Preview Visualize JSON

```

1 [
2   "status": 200,
3   "data": null,
4   "message": "Izin Keluar berhasil direquest"
5 ]

```

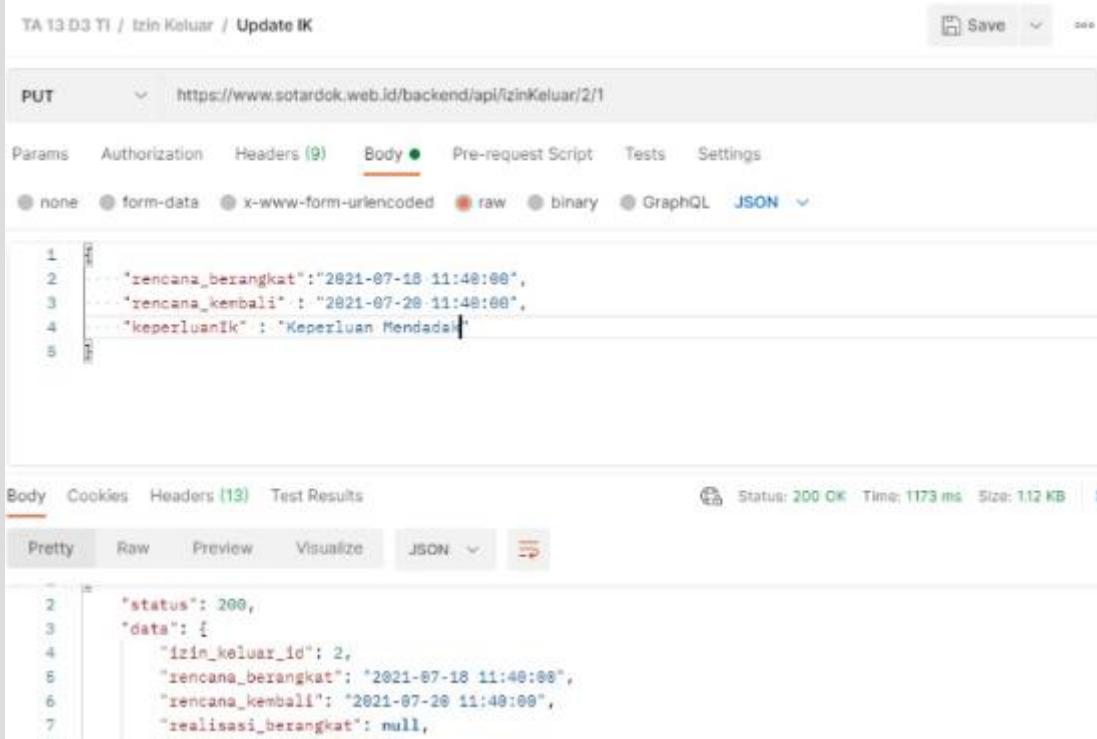
Status: 200 OK Time: 681 ms

Gambar 88. Pengujian API request IK

#### 11. Pengujian API edit *request Izin Keluar (IK)*

Pengujian API edit *request* IK pada postman menggunakan metode PUT dengan url pengujian API edit *request* IK yaitu

<https://sotardok.web.id/backend/api/izinKeluar/{izinId}/{id}>. Adapun hasil pengujian API edit *request* IK pada postman dapat dilihat pada Gambar 89 berikut.



```
PUT https://www.sotardok.web.id/backend/api/izinKeluar/2/1

Params Authorization Headers (9) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON

1
2   "rencana_berangkat": "2021-07-18 11:40:00",
3   "rencana_kembali": "2021-07-20 11:40:00",
4   "keperluanik": "Keperluan Mendadak"
5

Body Cookies Headers (13) Test Results
Pretty Raw Preview Visualize JSON
status: 200 OK Time: 1173 ms Size: 1.12 KB
{
  "status": 200,
  "data": {
    "izin_keluar_id": 2,
    "rencana_berangkat": "2021-07-18 11:40:00",
    "rencana_kembali": "2021-07-20 11:40:00",
    "realisasi_berangkat": null,
    "realisasi_kembali": null
  }
}
```

Gambar 89. Pengujian API edit *request* IK

## 12. Pengujian API cancel *request* Izin Keluar (IK)

Pengujian API cancel *request* IK pada postman menggunakan metode PUT dengan url pengujian API cancel *request* IK yaitu

<https://sotardok.web.id/backend/api/izinKeluar>. Adapun hasil pengujian API cancel *request* IK pada postman dapat dilihat pada Gambar 90 berikut.

The screenshot shows a POST request in Postman to the URL `https://www.sotardok.web.id/backend/api/izinKeluar`. The 'Body' tab is selected, showing a single key-value pair: `idizin: 1`. The response status is 200 OK, and the message is "Izin sudah di batalkan".

KEY	VALUE	DESCRIPTION
<code>idizin</code>	1	

Pretty Raw Preview Visualize JSON

```
1 [{}]
2   "status": 200,
3   "data": null,
4   "message": "Izin sudah di batalkan"
5 ]
```

Gambar 90. Pengujian API Cancel Request IK

### 13. Pengujian API menampilkan daftar-daftar Izin Keluar (IK)

Pengujian API menampilkan daftar-daftar IK pada postman menggunakan metode GET dengan url pengujian API menampilkan daftar-daftar IK yaitu `https://sotardok.web.id/backend/api/izinKeluar/{dimId}`. Adapun hasil pengujian API menampilkan daftar-daftar IK pada postman dapat dilihat pada Gambar 91 berikut.

The screenshot shows the Postman interface for a 'GET' request to <https://www.sotardok.web.id/backend/api/izinKeluar/1>. The 'Headers' tab is selected, showing an 'Authorization' header with the value 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiO...'. The 'Body' tab displays a JSON response with the following structure:

```
1 "status": 200,
2 "data": [
3     {
4         "izinKeluarId": "1",
5         "keperluanIk": "Ada Keperluan mendadak",
6         "rencanaBerangkat": "2021-07-12 15:19:00",
7         "rencanaKembali": "2021-07-12 20:19:00",
8         "realisasiBerangkat": null,
9         "realisasiKembali": null,
10        "statusDosen": "4",
11        "statusKeasramaan": "4",
12        "statusBaak": "4",
13    }
]
```

Gambar 91. Pengujian API Menampilkan Daftar-Daftar IK

#### 14. Pengujian API detail Izin Keluar (IK)

Pengujian API detail IK pada postman menggunakan metode GET dengan url pengujian API detail IK yaitu

<https://sotardok.web.id/backend/api/izinKeluar/{id}/{izinId}>. Adapun hasil pengujian API detail IK pada postman dapat dilihat pada Gambar 92 berikut.

The screenshot shows the Postman interface with the following details:

- Request Method:** GET
- URL:** <https://www.sotardok.web.id/backend/api/izinKeluar/1/1>
- Headers:** Authorization (set to Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiO...)
- Body:** JSON (Pretty) - The response body is displayed in a multi-line JSON format.
- Status:** 200 OK
- Time:** 1088 ms

```
1
2   "status": 200,
3   "data": {
4     "pemohon": "Martinus Yudha C Sitinjak",
5     "nimMahasiswa": "11318031",
6     "keperluanIk": "Ada Keperluan mendadak",
7     "rencanaBerangkat": "2021-07-12 15:19:00",
8     "rencanaKembali": "2021-07-12 20:19:00",
9     "realisasiBerangkat": null,
10    "realisasiKembali": null,
11    "statusDosen": "4",
12    "statusKeasramaan": "4",
13    "statusBaak": "4",
14    "namaDosen": null,
```

Gambar 92. Pengujian API Detail IK

### 15. Pengujian API data pribadi

Pengujian API data pribadi pada postman menggunakan metode GET dengan url pengujian API data pribadi yaitu <https://sotardok.web.id/backend/api/dim>.

Adapun hasil pengujian API data pribadi pada postman dapat dilihat pada Gambar 93 berikut.

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** https://sotardok.web.id/backend/api/dim
- Headers:** Authorization (Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...
- Body:** JSON response (Pretty):
 

```

1   "status": 200,
2   "data": {
3     "nim_mahasiswa": "i1318031",
4     "nama_mahasiswa": "Martinus Yudha C Sitinjak",
5     "tanggal_lahir": "2000-03-18",
6     "tempat_lahir": "Medan",
7     "alamat": "Jl. Kerumung Gang Regar No. 97A Medan",
8     "kabupaten": null,
9     "kode_pos": "20222",
10    "email": "martinus@gmail.com",
11    "hp": "81263812093",
12    "hp2": "-",
13    "anak_ke": "1",
14
      
```
- Status:** 200 OK
- Time:** 563 ms

Gambar 93. Pengujian API Data Pribadi

## 16. Pengujian API edit data pribadi

Pengujian API edit data pribadi pada postman menggunakan metode PUT dengan url pengujian API edit data pribadi yaitu <https://sotardok.web.id/backend/api/updateDim>. Adapun hasil pengujian API edit data pribadi pada postman dapat dilihat pada Gambar 94 berikut.

The screenshot shows a Postman interface with the following details:

- Method:** PUT
- URL:** https://sotardok.web.id/backend/api/updateDim
- Headers:** Authorization (Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...
- Body:** JSON response (Pretty):
 

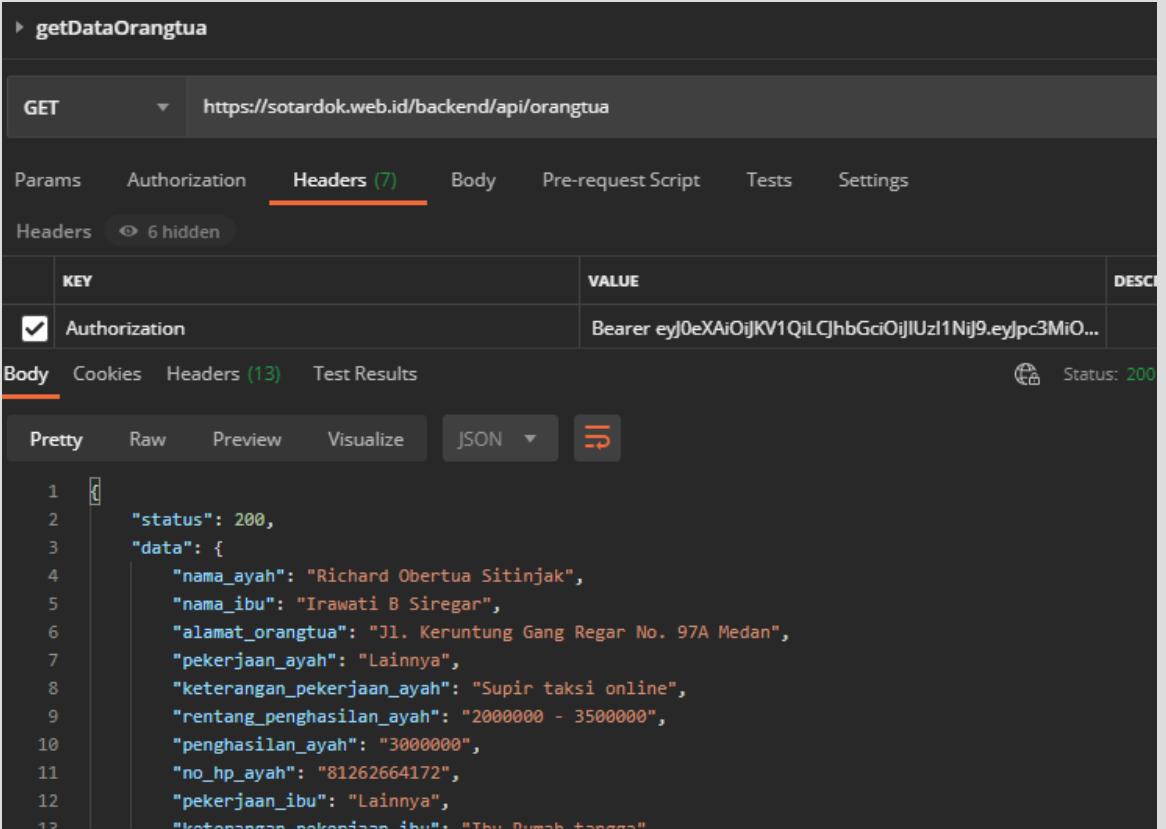
```

1   "status": 200,
2   "data": {
3     "nim_mahasiswa": "i1318031",
4     "nama_mahasiswa": "Martinus Yudha C Sitinjak",
5     "tanggal_lahir": "2000-03-18",
6     "tempat_lahir": "Medan",
7     "alamat": "Jl. Kerumung Gang Regar No. 97A Medan",
8     "kabupaten": null,
9     "kode_pos": "20222",
10    "email": "martinus@gmail.com",
11    "hp": "81263812093",
12    "hp2": "-",
13    "anak_ke": "1",
14
      
```
- Status:** 200 OK
- Time:** 1000 ms

Gambar 94. Pengujian API Edit Data Pribadi

## 17. Pengujian API data orangtua

Pengujian API data orangtua pada postman menggunakan metode PUT dengan url pengujian API data orangtua yaitu <https://sotardok.web.id/backend/api/orangtua>. Adapun hasil pengujian API data orangtua pada postman dapat dilihat pada Gambar 95 berikut.



The screenshot shows a Postman interface with the following details:

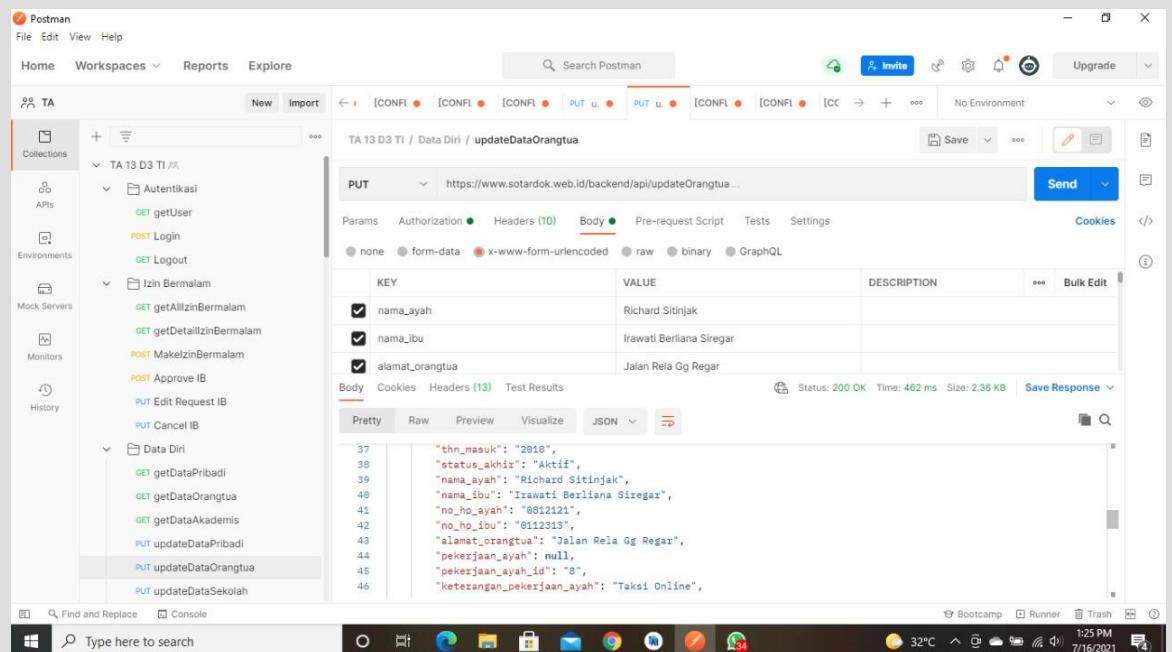
- Request Method:** GET
- URL:** <https://sotardok.web.id/backend/api/orangtua>
- Headers:** Authorization (set to Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiO...)
- Body (JSON Response):**

```
1  {
2     "status": 200,
3     "data": {
4         "nama_ayah": "Richard Obertua Sitinjak",
5         "nama_ibu": "Irawati B Siregar",
6         "alamat_orangtua": "Jl. Keruntung Gang Regar No. 97A Medan",
7         "pekerjaan_ayah": "Lainnya",
8         "keterangan_pekerjaan_ayah": "Supir taksi online",
9         "rentang_penghasilan_ayah": "2000000 - 3500000",
10        "penghasilan_ayah": "3000000",
11        "no_hp_ayah": "81262664172",
12        "pekerjaan_ibu": "Lainnya",
13        "keterangan_pekerjaan_ibu": "Tidak Diumumkan"
```
- Status:** 200

Gambar 95. Pengujian API Data Orangtua

## 18. Pengujian API edit data orangtua

Pengujian API edit data orangtua pada postman menggunakan metode PUT dengan url pengujian API edit data orangtua yaitu <https://sotardok.web.id/backend/api/updateOrangtua>. Adapun hasil pengujian API edit data orangtua pada postman dapat dilihat pada Gambar 96 berikut.



Gambar 96. Pengujian API Edit Data Orangtua

## 19. Pengujian API data asal sekolah

Pengujian API data asal sekolah pada postman menggunakan metode GET dengan url pengujian API data asal sekolah yaitu

<https://sotardok.web.id/backend/api/sekolah>. Adapun hasil pengujian API data asal sekolah pada postman dapat dilihat pada Gambar 97 berikut.

KEY	VALUE	DESCRIPTION
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiO...	

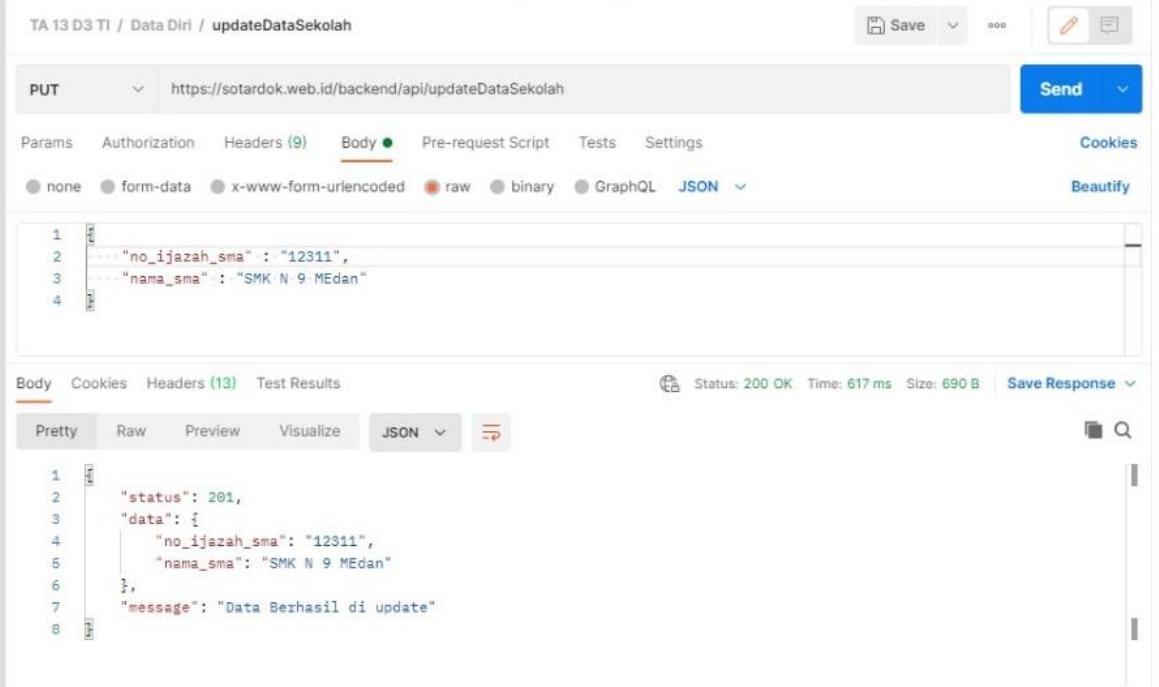
```

1 {
2   "status": 200,
3   "data": [
4     {
5       "noIjazah": "1231",
6       "namaSekolah": "SMK N 9 Medan",
7       "alamatSekolah": "Jl. Kartini Sopo Surung",
8       "kabupatenSma": "Toba Samosir",
9       "kodePos": "22382",
10      "telepon": "081234874980"
11    },
12  ],
13  "message": "Data Sekolah Didapat"
}
  
```

Gambar 97. Pengujian API Data Asal Sekolah

## 20. Pengujian API edit data asal sekolah

Pengujian API edit data asal sekolah pada postman menggunakan metode PUT dengan url pengujian API edit data asal sekolah yaitu <https://sotardok.web.id/backend/api/updateDataSekolah>. Adapun hasil pengujian API edit data asal sekolah pada postman dapat dilihat pada Gambar 98 berikut.



The screenshot shows the Postman interface for a PUT request to the URL <https://sotardok.web.id/backend/api/updateDataSekolah>. The request body contains the following JSON:

```
1
2   ...
3   ...
4   ...
```

The response status is 200 OK, with a message indicating the data was successfully updated.

```
1
2   "status": 201,
3   "data": {
4     ...
5     ...
6   },
7   "message": "Data berhasil di update"
```

**Gambar 98. Pengujian API Edit Data Asal Sekolah**

## 21. Pengujian API data akademis

Pengujian API data akademis pada postman menggunakan metode GET dengan url pengujian API data akademis yaitu <https://sotardok.web.id/backend/api/akademis>. Adapun hasil pengujian API data akademis pada postman dapat dilihat pada Gambar 99 berikut.

```

1  {
2      "status": 200,
3      "data": {
4          "status_akhir": "Aktif",
5          "thn_masuk": "2018",
6          "user_name": "if318031",
7          "email_akademik": "martinus@gmail.com",
8          "prodi": "DIII Teknologi Informasi",
9          "kelas": "33TI1",
10         "wali_mahasiswa": "Teamsar Muliadi Panggabean, S.Kom, PGCert",
11         "jalur": "USM1"
12     },
13 }

```

**Gambar 99. Pengujian API Data Akademis**

## 22. Pengujian API data asrama

Pengujian API data asrama pada postman menggunakan metode GET dengan url pengujian API data asrama yaitu <https://sotardok.web.id/backend/api/dataAsrama>. Adapun hasil pengujian API data asrama pada postman dapat dilihat pada Gambar 100 berikut.

```

1  {
2      "status": 200,
3      "data": [
4          {
5              "teman_sekamar": [
6                  {
7                      "nama": "Palti Mangaruhut Gudmen Siregar",
8                      "tahun_masuk": "2018",
9                      "prodi": "DIII Teknologi Informasi"
10                 },
11                 {
12                     "nama": "Samuel Ambarita",
13                     "tahun_masuk": null,
14                     "prodi": "DIII Teknologi Informasi"
15                 }
16             ]
17         }
18     ]
19 }

```

**Gambar 100. Pengujian API Data Asrama**

### 23. Pengujian API pengumuman

Pengujian API pengumuman pada postman menggunakan metode GET dengan url pengujian API pengumuman yaitu <https://sotardok.web.id/backend/api/pengumuman>. Adapun hasil pengujian API pengumuman pada postman dapat dilihat pada Gambar 101 berikut.

The screenshot shows the Postman interface with the following details:

- Method: GET
- URL: https://sotardok.web.id/backend/api/pengumuman
- Headers tab is selected, showing an Authorization header with the value: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...
- Body tab is selected, showing a JSON response:

```
1  {
2      "status": 200,
3      "data": [
4          {
5              "id": "1",
6              "judul": "Draft Jadwal Ujian Tengah Semester (UTS) Genap 2021",
7              "kategori": "All",
8              "kadaluarsa": "2021-03-11",
9              "pembuatan": "2021-03-09 10:52:31"
10         }
11     ],
12     "message": "Data Sukses Didapat"
13 }
```
- Status: 200 OK Time: 1346 ms

Gambar 101. Pengujian API Pengumuman

### 24. Pengujian API detail pengumuman

Pengujian API detail pengumuman pada postman menggunakan metode GET dengan url pengujian API detail pengumuman yaitu [https://sotardok.web.id/backend/api/<pengumuman\\_id>](https://sotardok.web.id/backend/api/<pengumuman_id>). Adapun hasil pengujian API detail pengumuman pada postman dapat dilihat pada Gambar 102 berikut.

```

1
2 "status": 200,
3 "data": {
4     "judul": "Draft Jadwal Ujian Tengah Semester (UTS) Genap 2021",
5     "kategori": "All",
6     "tglKadaluarsa": "2021-03-11",
7     "tglBuat": "2021-03-09 10:52:31",
8     "pembuat": "oka.simatupang",
9     "isi": "<p style=\"text-align: justify;\">Dear Mahasiswa,</p><p style=\"text-align: justify;\">Berikut ini terlampir Jadwal Ujian Tengah Semester (UTS) Genap 2021. </p><p style=\"text-align: justify;\">Apabila terdapat Jadwal Ujian yang diharapkan mahasiswa dapat menginformasikan ke e-mail <strong>baak@del.ac.id</strong> dengan Subjek : <strong>Bentrok - Angkatan. </strong>Konfirmasi Jadwal UTS yang bentrok diterima paling lama hari <strong>Rabu, 10 Maret 2021</strong></p><p style=\"text-align: justify;\">Demikian informasi ini kami sampaikan, terima kasih dan Selamat Berjuang. </p>",
10    "files": [
        ...
    ]
}

```

Gambar 102. Pengujian API Detail Pengumuman

## 25. Pengujian API paket

Pengujian API paket pada postman menggunakan metode GET dengan url pengujian API paket yaitu <https://sotardok.web.id/backend/api/paket>. Adapun hasil pengujian API paket pada postman dapat dilihat pada Gambar 103 berikut.

```

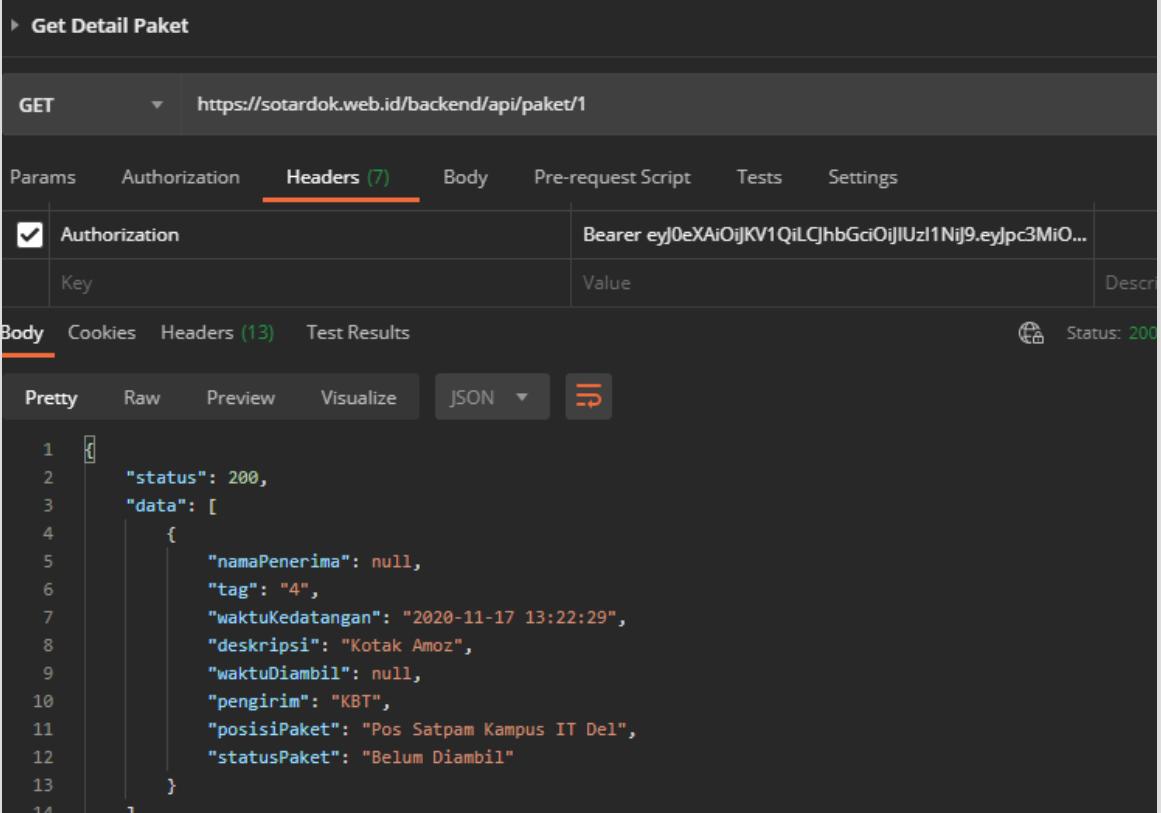
1
2 "status": 200,
3 "data": {
4     "myPaket": [
5         {
6             "namaPenerima": null,
7             "tag": "4",
8             "status": "Belum Dambil",
9             "waktuKedatangan": "2020-11-17 13:22:29",
10            "deskripsi": "Kotak Amoz",
11            "pengirim": "KBT",
12            "id": "1"
13        }
]
}

```

Gambar 103. Pengujian API Paket

## 26. Pengujian API detail paket

Pengujian API detail paket pada postman menggunakan metode GET dengan url pengujian API detail paket yaitu <https://sotardok.web.id/backend/api/paket/><paket id>. Adapun hasil pengujian API detail paket pada postman dapat dilihat pada Gambar 104 berikut.



The screenshot shows a Postman interface with the following details:

- Request Method:** GET
- URL:** https://sotardok.web.id/backend/api/paket/1
- Headers:** Authorization (Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiO...)
- Body (Pretty):**

```
1  {
2      "status": 200,
3      "data": [
4          {
5              "namaPenerima": null,
6              "tag": "4",
7              "waktuKedatangan": "2020-11-17 13:22:29",
8              "deskripsi": "Kotak Amoz",
9              "waktuDiambil": null,
10             "pengirim": "KBT",
11             "posisiPaket": "Pos Satpam Kampus IT Del",
12             "statusPaket": "Belum Diambil"
13         }
14     ],
15 }
```
- Status:** 200

Gambar 104. Pengujian API Detail Paket

## 27. Pengujian API kuesioner

Pengujian API kuesioner pada postman menggunakan metode GET dengan url pengujian API kuesioner yaitu https://sotardok.web.id/backend/api/kuesioner. Adapun hasil pengujian API kuesioner pada postman dapat dilihat pada Gambar 105 berikut.

```

1   {
2     "status": 200,
3     "data": [
4       {
5         "kuesioner_id": 3755,
6         "sem": "2",
7         "ta": "2020",
8         "kode_mk": "All",
9         "nama": "FORMULIR KUESIONER TINGKAT KEPUASAN MAHASISWA TERHADAP LAYANAN UNIT SDI & DUKTEK",
10        "keterangan": "",
11        "instruksi_pengisian": "<p><strong>I. Pilihlah kriteria yang menurut Anda paling sesuai dari pernyataan mengenai Duktek di Institut Teknologi Del dengan memberikan titik (.) pada isian yang disediakan (satu nomor satuan</p><p><strong><br></strong></p>",
12        "start_date": "0000-00-00 00:00:00",
13      }
14    ]
15  }

```

Status: 200 OK Time: 672 ms

Gambar 105. Pengujian API Kuesioner

## 28. Pengujian API kuesioner wajib

Pengujian API kuesioner wajib pada postman menggunakan metode GET dengan url pengujian API kuesioner wajib yaitu <https://sotardok.web.id/backend/api/kuesionerWajib>. Adapun hasil pengujian API kuesioner wajib pada postman dapat dilihat pada Gambar 106 berikut.

```

1  {
2      "status": 200,
3      "data": [
4          {
5              "kuesioner_id": 3758,
6              "sem": "3",
7              "ta": "2020",
8              "kode_mk": "All",
9              "nama": "Kuesioner Belajar Mandiri untuk Mahasiswa",
10             "keterangan": null,
11             "instruksi_pengisian": null,
12             "start_date": "2021-07-14 17:45:55",
13             "end_date": "2021-07-15 17:45:55",
14             "peserta_kuis": null,

```

**Gambar 106. Pengujian API Kuesioner Wajib**

## 29. Pengujian API pertanyaan kuesioner

Pengujian API pertanyaan kuesioner pada postman menggunakan metode GET dengan url pengujian API pertanyaan kuesioner yaitu <https://sotardok.web.id/backend/api/pertanyaan/idkuesioner>. Adapun hasil pengujian API pertanyaan kuesioner pada postman dapat dilihat pada Gambar 107 berikut.

```
1  {
2      "status": 200,
3      "data": [
4          {
5              "kuesioner_pertanyaan_id": 96337,
6              "kuesioner_id": "3755",
7              "nomor": "0",
8              "pertanyaan": "<p>Kemampuan dari staf teknis sesuai dengan bidang keahliannya dalam melayani kebutuhan mahasiswa</p>",
9              "tipe_opsi": "R",
10             "kategori": "A.Keandalan (reliability): kemampuan",
11             "deleted": "0",
12             "deleted_at": "0000-00-00 00:00:00",
13             "deleted_by": "N",
14             "created_at": "-0001-11-30 00:00:00",
15             "updated_at": "-0001-11-30 00:00:00",
16             "created_by": "0",
17             "updated_by": "vinni.lubis",
18             "id_kuesioner": "0"
19         }
20     ]
21 }
```

Gambar 107. Pengujian API Pertanyaan Kuesioner

### 30. Pengujian API isi kuesioner

Pengujian API isi kuesioner pada postman menggunakan metode POST dengan

url pengujian API isi kuesioner yaitu

<https://sotardok.web.id/backend/api/isiKuesioner>. Adapun hasil pengujian API isi kuesioner pada postman dapat dilihat pada Gambar 108 berikut.

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** <https://sotardok.web.id/backend/api/isiKuesioner>
- Body (JSON):**

```
1 []
2   "kuesioner" : [
3     {
4       "jawaban" : 2,
5       "peserta" : "if318031",
6       "user_id" : 1,
7       "kuesioner_pertanyaan_id":96337,
8       "kuesioner_id" : 3755
```
- Response Status:** 200 OK
- Response Time:** 699 ms
- Response Body (Pretty JSON):**

```
1 {}
2   "status": 200,
3   "data": [
4     {
5       "jawaban": 2,
6       "peserta": "if318031",
```

Gambar 108. Pengujian API Isi Kuesioner

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan dijelaskan mengenai kesimpulan dan saran selama pengerjaan Tugas Akhir.

#### **6.1 Kesimpulan**

Adapun kesimpulan yang diperoleh selama pengerjaan Tugas Akhir ini, yaitu sebagai berikut:

1. Sistem yang dibangung telah menghasilkan Pengembangan Fitur-Fitur *Campus Information System Berbasis Mobile* dengan Layanan API.
2. Penggunaan API dalam pengaksesan data sangat membantu *developer* dalam membangun aplikasi ini. Sistem akan mengakses *service* sesuai dengan apa yang dibutuhkan oleh *request* dari setiap fungsi tanpa memuat keseluruhan data yang ada pada sistem.
3. Penggunaan Kotlin dalam pembangunan aplikasi mempermudah tim pengembang dalam menulis *code* pada proses pembangunan aplikasi.
4. Aplikasi yang dibangun mendapat respon positif dari *client* karena dilihat dari hasil UAT. Adapun hasil *User Acceptance Testing* (UAT) dapat dilihat pada Lampiran 5.

#### **6.2 Saran**

Saran yang dapat diperhatikan untuk pengembangan Tugas Akhir selanjutnya adalah sebagai berikut:

1. Tim pengembang melengkapi modul diluar modul otentikasi, melihat notifikasi, *request* Izin Bermalam (IB), *request* Izin Keluar (IK), perkuliahan, data diri, kuesioner, melihat pengumuman, Kartu Rencana Studi (KRS), melihat paket, unduh modul, melihat grafik Indeks Prestasi (IP) dan melihat jadwal.
2. Mengimplementasikan keamanan sistem pada aplikasi.

## Rujukan

- [1] L. Alam, “Internalisasi Nilai-Nilai Pendidikan Islam Dalam Perguruan Tinggi Umum Melalui Lembaga Dakwah Kampus,” *Istawa J. Pendidik. Islam*, vol. 1, no. 2, p. 101, 2016, doi: 10.24269/ijpi.v1i2.171.
- [2] A. Heriadi, M. Suyanto, and S. Sudarmawan, “Perencanaan Strategis Sistem Informasi STMIK Cahaya Surya Kediri,” *Creat. Inf. Technol. J.*, vol. 1, no. 1, p. 15, 2015, doi: 10.24076/citec.2013v1i1.6.
- [3] G. I. S. On and T. H. E. Internet, “named as ESRI ’ s XML is designed as a protocol data exchange between ArcIMS program,” 2002.
- [4] C. Bischof, M. Gebhardt, and P. Steves, “The Integrated CAMPUS Information System: Bridging the Gap between Administrative and E-Learning Processes,” *EUNIS 2005 - Leadersh. Strateg. a Cyberworld, Manchester, June 21 - 24, 2005*, 2005.
- [5] M. Muharir and N. Alamsyah, “Pemanfaatan Mobile Push Notification Dalam Penyampaian Informasi Perkuliahan Mahasiswa Pada Fakultas Teknologi Informasi Berbasis Android,” *Technologia*, vol. 11, no. 2, pp. 111–118, 2020.
- [6] H. Amrullah and B. Handaga, “Sistem Informasi Akademik Untuk Layanan Mahasiswa Ums Berbasis Mobile,” *J. Emit. Vol.17 No. 01 ISSN 1411-8890*, vol. 17, no. 01, pp. 1–5, 2018.
- [7] H. C. Cheng, T. P. Kung, C. M. Li, and Y. J. Sun, “The current state of mobile apps development of higher education in Taiwan,” *Int. Conf. Adv. Commun. Technol. ICACT*, vol. 5, no. 2, pp. 780–786, 2017, doi: 10.23919/ICACT.2017.7890227.
- [8] F. Kartika, S. Dewi, T. D. Indriasari, and Y. Prayogo, “77144-ID-rancang-bangun-aplikasi-pengingat-kegiatan(1),” *Ranc. Bangun Apl. Pengingat Kegiat. Akad. Berbas. Mob.*, vol. 7, no. 4, pp. 303–312, 2016.
- [9] and R. A. A. Hariyanto, E. Sugiharti, “Rancang Bangun Aplikasi Mobile Sistem Universitas Negeri Semarang Berbasis,” 2017.
- [10] F. Kusumatuti, “Aplikasi Perpustakaan Online Berbasis Android Dan Php Menggunakan Protokol Json,” pp. 1–11, 2009.
- [11] S. Sunardi, I. Riadi, and P. A. Raharja, “Analisis Application Programming Interface Pada Mobile E-Voting Menggunakan Metode Test-Driven Development,” *Techno (Jurnal Fak. Tek. Univ. Muhammadiyah Purwokerto)*, vol. 20, no. 2, p. 87, 2019, doi: 10.30595/techno.v20i2.4266.
- [12] A. Sandi, “Mengenal Apa itu Web API,” 16 November, 2017. [Online]. Available: <https://www.codepolitan.com/mengenal-apa-itu-web-api-5a0c2855799c8>.
- [13] F. Yerisha, H. Ridwan, B. Priyambadha, and H. Nurwarsito, “Pengembangan Aplikasi Katalog Wedding Organizer berbasis Android menggunakan Restful Web Service di Ruang Lingkup Kota Malang,” vol. 3, no. 9, pp. 8572–8581, 2019.
- [14] F. Sari, *Metode dalam Pengambilan Keputusan - Febrina Sari - Google Books*. Deepublish, 2018.
- [15] R. Islam and T. Mazumder, “Mobile application and its global impact,” *Int. J. Eng.* ..., no. 06, pp. 72–78, 2010.
- [16] A. Ismail, S. A. S. A. Kadir, A. Aziz, M. Mokshin, and A. M. Lokman, “ITourism travel buddy mobile application,” *Int. Conf. Next Gener. Mob. Appl. Serv. Technol.*, pp. 82–87, 2016, doi: 10.1109/NGMAST.2016.22.
- [17] R. Bangun and A. Pembelajaran, “Rancang Bangun Aplikasi Pembelajaran Hadis Untuk Perangkat Mobile Berbasis Android,” *J. Inform.*, vol. 8, no. 2, pp. 907–920,

- 2014, doi: 10.26555/jifo.v8i2.a2057.
- [18] S. Mukherjee, P. J. Prakash, and D. Kumar, “Android Application Development & Its Security,” *Int. J. Comput. Sci. Mob. Comput.*, vol. 4, no. 3, pp. 714–719, 2015.
- [19] E. Chin, A. P. Felt, K. Greenwood, and D. Wagner, “Analyzing inter-application communication in Android,” *MobiSys’11 - Compil. Proc. 9th Int. Conf. Mob. Syst. Appl. Serv. Co-located Work.*, pp. 239–252, 2011, doi: 10.1145/1999995.2000018.
- [20] N. NA, “aplikasi sistem informasi rumah sakit berbasis web pada sub-sistem farmasi menggunakan framework prado,” *Int. J. Ther. Rehabil. Res.*, vol. 6, no. 1, p. 148, 2017, doi: 10.5455/ijtrr.000000235.
- [21] R. Abdulloh, “Membuat Aplikasi Point of Sale dengan Laravel dan Ajax,” pp. 1–252, 2017.
- [22] M. Moskala and I. Wojda, “Android Development With Kotlin,” □□□□□□, 2017. [Online]. Available: [https://books.google.co.id/books?hl=id&lr=&id=PJZGDwAAQBAJ&oi=fnd&pg=PP1&dq=Kotlin+for+Android+Developers+&ots=3Kci9APUsC&sig=XsrYZ2sCqiMC7dCMUEL12pAyFYw&redir\\_esc=y#v=onepage&q=Expressiveness&f=false](https://books.google.co.id/books?hl=id&lr=&id=PJZGDwAAQBAJ&oi=fnd&pg=PP1&dq=Kotlin+for+Android+Developers+&ots=3Kci9APUsC&sig=XsrYZ2sCqiMC7dCMUEL12pAyFYw&redir_esc=y#v=onepage&q=Expressiveness&f=false).
- [23] S. Bose, “a Comparative Study: Java Vs Kotlin Programming in Android Application Development,” *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 3, pp. 41–45, 2018, doi: 10.26483/ijarcs.v9i3.5978.
- [24] A. K. S. Putra, “Program studi teknik informatika fakultas teknologi informasi dan elektro universitas teknologi yogyakarta 2017,” 2017.
- [25] B. B. Premana, “Aplikasi Sistem Digital Trainer Support Atlet Ikatan Pencak Silat Berbasis Android,” *Univ. 17 Agustus 1945 Surabaya*, pp. 5–14, 2005.
- [26] <Https://idcloudhost.com>, “Mengenal Apa itu Android Studio : Fungsi, Manfaat, dan Cara Installasinya | IDCLOUDHOST,” *Https://Idcloudhost.Com/Mengenal-Apa-Itu-Android-Studio-Fungsi-Manfaat-Dan-Cara-Installasinya/*, 2019. [Online]. Available: <https://idcloudhost.com/mengenal-apa-itu-android-studio-fungsi-manfaat-dan-cara-installasinya/>.
- [27] A. Developers, “Meet Android Studio,” *Online*, 2020. [Online]. Available: <https://developer.android.com/studio/intro>.
- [28] A. Kaur, “App Review: Trello,” *J. Hosp. Librariansh.*, vol. 18, no. 1, pp. 95–101, 2018, doi: 10.1080/15323269.2018.1400840.
- [29] D. Parsons, R. Thorn, M. Inkila, and K. MacCallum, “Using Trello to Support Agile and Lean Learning with Scrum and Kanban in Teacher Professional Development,” *Proc. 2018 IEEE Int. Conf. Teaching, Assessment, Learn. Eng. TALE 2018*, no. December, pp. 720–724, 2019, doi: 10.1109/TALE.2018.8615399.
- [30] D. Mairiza, D. Zowghi, and N. Nurmuliani, “An investigation into the notion of non-functional requirements,” *Proc. ACM Symp. Appl. Comput.*, no. January 2010, pp. 311–317, 2010, doi: 10.1145/1774088.1774153.

## **Lampiran 1 - Hasil Kuesioner**

Berikut hasil *review* pandangan mahasiswa dalam Penggunaan CIS berbasis *Website* dan hasil rekapitulasi kuesioner mahasiswa IT Del yang terdiri dari 160 responden:

[https://docs.google.com/spreadsheets/d/1UZBZvLN4Kt2i8br8Vuk4\\_boOcEOydq34cDXfFt7WtHs/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1UZBZvLN4Kt2i8br8Vuk4_boOcEOydq34cDXfFt7WtHs/edit?usp=sharing)



**Gambar 109. Grafik Hasil Kuesioner 1**



### **Gambar 110. Grafik Hasil Kuesioner**

## Lampiran 2 – Arsitektur Uniform Resource Identifier (URI)

Berikut akan dijelaskan tentang arsitektur URI sistem dari aplikasi yang akan dibangun.

**Tabel 30. Arsitektur URI**

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
1	/api/user	GET	200	“Data Sukses Didapat”		{         "status": 200,         "data": {             "dim_id": ,             "nim": "",             "no_usm": "",             "jalur": "",             "user_name": "",             "kbk_id": ,             "ref_kbk_id": "",             "kpt_prodi": ,             "id_kur": ""         },         "tahun_kurikulum_id": ,         "nama": "",         "tgl_lahir": "",         "tempat_lahir": "",         "gol_darah": ,         "golongan_darah_id": "",         "jenis_kelamin": ,         "jenis_kelamin_id": ,         "agama": ,         "agama_id": "",         "alamat": ,         "kabupaten": ,         "kode_pos": ,         "email": ,         "telepon": ,         "hp": ,         "hp2": ,         "no_ijazah_sma": ,         "nama_sma": ,         "asal_sekolah_id": ,         "alamat_sma": ,         "kabupaten_sma": ,         "telepon_sma": ,         "kodepos_sma": ,         "thn_masuk": ,         "status_akhir": ,         "nama_ayah": ,         "nama_ibu": ,         "no_hp_ayah": ,         "no_hp_ibu": ,         "alamat_orangtua": "     },

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
						<pre>"pekerjaan_ayah": , "pekerjaan_ayah_id": "", ", "keterangan_pekerjaan_ayah": "", "penghasilan_ayah": "", ", "penghasilan_ayah_id": "", "pekerjaan_ibu": , "pekerjaan_ibu_id": "", ", "keterangan_pekerjaan_ibu": "", "penghasilan_ibu": "", ", "penghasilan_ibu_id": "", "nama_wali": null, "pekerjaan_wali": , "pekerjaan_wali_id": , ", "keterangan_pekerjaan_wali": , "penghasilan_wali": , ", "penghasilan_wali_id": , "alamat_wali": , "telepon_wali": , "no_hp_wali": , "pendapatan": , "ipk": "", "anak_ke": "", "dari_jlh_anak": "", ", "jumlah_tanggungan": "", "nilai_usm": , "score_iq": , ", "rekomendasi_psikotest": null, "foto": , "kode_foto": "", "user_id": "", "deleted": "", "deleted_at": , "deleted_by": , "created_at": , "updated_at": "", "created_by": ,</pre>

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
						<pre>"updated_by": }, "message": "Data Sukses Didapat" }</pre>
2	/api/login	POST	200	“Data Sukses Didapat”	<pre>{   "username": "",   "password": "" }</pre>	<pre>{   "status": 200,   "data": {     "user": {       "dimId": "",       "nama": ""     },     "token": ""   },   "message": "Data Sukses Didapat" }</pre>
3	/api/logout	GET		“Successf ully logged out”		<pre>{   "message": "Successfully logged out" }</pre>
4	/api/izinBer malam/<yo ur user id>	GET	200	“Data Sukses Didapat”		<pre>{   "status": 200,   "data": [     {       "id_izin": "",       "request_id": "",       "disetujui_oleh": ,       "keperluan_ib": "",       "tujuan": ""     }   ],   "message": "Data Sukses Didapat" }</pre>
5	/api/getIzin /<user_id>/ <izin_berm alam_id>	GET	200	“Data Sukses Didapat”		<pre>{   "status": 200,   "data": {     "nama_mahasiswa": "",     "nim_mahasiswa": "",     "rencana_berangkat": "",     "rencana_kembali": "",     "realisasi_berangkat": ,     "realisasi_kembali": 1,     "keperluan_ib": ""   } }</pre>

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
						<pre>"tujuan": "", "request_id": "", "disetujui_oleh": }, "message": "Data Sukses Didapat" }</pre>
6	/api/izinBer malam/<us er id>	POST	200	“Data berhasil disimpan”	{     “rencana_berangka t”: “” “rencana_kembali” : “” “desc” : “” “tujuan” : “” }	{     “status”: 200, “data”: , “message”: “Data berhasil disimpan” }
7	/api/dim	GET	200	“Data Sukses Didapat”		{     “status”: 200, “data”: { “nim_mahasiswa”: “”, “nama_mahasiswa”: “”, “tanggal_lahir”: “”, “tempat_lahir”: “”, “alamat”: “”, “kabupaten”: , “kode_pos”: “”, “email”: , “hp”: “”, “hp2”: “”, “anak_ke”: “”, “kode_foto”: “”, “dari_jlh_anak”: “”, “jenis_kelamin”: “”, “agama”: “”, “gol_darah”: “” }, “message”: “Data Sukses Didapat” }
8	/api/update Dim	PUT	200	“Data berhasil disimpan”	{     “kabupaten”: “”, “kode_pos”: “”, “nama”: “”, “tgl_lahir”: “”, “jenis_kelamin”: “”, “agama”: “”, “gol_dar”: “”, “alamat”: “”, “email”: “”, }	“status”: 200, “data”: { “kabupaten”: “”, “kode_pos”: “”, “nama”: “”, “tgl_lahir”: “”, “jenis_kelamin”: “”, “agama”: “”, “gol_dar”: “”, “alamat”: “”, “email”: “”, }

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
					{"telepon": "", "hp": "", "hp2": "", "anak_ke": "", "jumlah_anak": "", "nim": ""}	{"telepon": "", "hp": "", "hp2": "", "anak_ke": "", "jumlah_anak": "", "nim": ""} "message": "Data berhasil disimpan"
9	/api/orangtua	GET	200	Data Sukses Didapat		{     "status": 200,     "data": {         "nama_ayah": "",         "nama_ibu": "",         "alamat_orangtua": "",         "pekerjaan_ayah": "",         "keterangan_pekerjaan_ayah": "",         "rentang_penghasilan_ayah": "",         "penghasilan_ayah": "",         "no_hp_ayah": "",         "pekerjaan_ibu": "Lainnya",         "keterangan_pekerjaan_ibu": "",         "rentang_penghasilan_ibu": "",         "penghasilan_ibu": "",         "no_hp_ibu": "",         "jumlah_tanggungan": "",         "nama_wali": ,         "alamat_wali": ,         "pekerjaan_wali": ,         "keterangan_pekerjaan_wali": ,         "telepon_wali": ,         "no_hp_wali": ,         "message": "Data Sukses Didapat"     } }

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
10	/api/update Orangtua	PUT	200	“Data berhasil diupdate”	{           “nama_ayah”: “”, “nama_ibu”: “”, “alamat_orangtua”: “”, “pekerjaan_ayah”: “”, “keterangan_pekerjaan_ayah”: “”, “keterangan_pekerjaan_ibu”: “”, “pekerjaan_ibu”: “”, “rentang_penghasilan_ayah”: “”, “penghasilan_ayah”: “”, “no_hp_ayah”: “”, “no_hp_ibu”: “”, “rentang_penghasilan_ibu”: “”, “penghasilan_ibu”: “”, “jumlah_tanggungan”: “”, “nama_wali”: “”, “alamat_wali”: “”, “keterangan_pekerjaan_wali”: “”, “rentang_penghasilan_wali”: “”, “penghasilan_wali”: “”, “telepon_wali”: “”, “no_hp_wali”: “”         }	{           “status”: 200, “data”: { “nama_ayah”: “”, “nama_ibu”: “”, “alamat_orangtua”: “”, “pekerjaan_ayah”: “”, “keterangan_pekerjaan_ayah”: “”, “keterangan_pekerjaan_ibu”: “”, “pekerjaan_ibu”: “”, “rentang_penghasilan_ayah”: “”, “penghasilan_ayah”: “”, “no_hp_ayah”: “”, “no_hp_ibu”: “”, “rentang_penghasilan_ibu”: “”, “penghasilan_ibu”: “”, “jumlah_tanggungan”: “”, “nama_wali”: “”, “alamat_wali”: “”, “keterangan_pekerjaan_wali”: “”, “rentang_penghasilan_wali”: “”, “penghasilan_wali”: “”, “telepon_wali”: “”, “no_hp_wali”: “”           }, “message”: “Data Sukses Didapat”         }
11	/api/akademis	GET	200	Data Sukses Didapat		{           “status”: 200, “data”: { “status_akhir”: “”, “thn_masuk”: “”, “user_name”: “”, “email_akademik”: “”, “prodi”: “”, “kelas”: “”, “wali_mahasiswa”: “”, “jalur”: “”           }, “message”: “Data Sukses Didapat”         }

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
12	/api/sekolah	GET	200	Data Sukses Didapat		{         "status": 200,         "data": [           {             "noIjazah": "",             "namaSekolah": "",             "alamatSekolah": "",             "kabupatenSma": "",             "kodePos": "",             "telepon": ""           }         ],         "message": "Data Sukses Didapat"       }
13	/api/update DataSekolah	PUT	200	Data Berhasil Diupdate	{         "no_ijazah": "",         "nama_sma": ""       }	{         "status": 200,         "data": {           "no_ijazah": "",           "nama_sma": ""         },         "message": "Data Berhasil Diupdate"       }
14	/api/pengumuman	GET	200	Data Sukses Didapat		{         "status": 200,         "data": [           {             "id": "",             "judul": "",             "kategori": "",             "kadaluarsa": "",             "pembuatan": ""           }         ],         "message": "Data Sukses Didapat"       }
15	/api/pengumuman<p engumuman id>	GET	200	Data Sukses Didapat		{         "status": 200,         "data": [           {             "judul": "",             "kategori": "",             "tglKadaluarsa": "",             "tglBuat": "",             "pembuat": ""           }         ]       }

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
						<pre>         "isi": "",          "namaFile": "",          "kodefile": ""        }      ],      "message": "Data  Sukses Didapat"  } </pre>
16	/api/paket	GET	200	Data Sukses Didapat		<pre> {   "status": 200,    "data": {     "myPaket": [        {          "namaPenerima": ,          "tag": "",          "status": "",          "waktuKedatangan": "",          "deskripsi": "",          "pengirim": "",          "id": ""        }      ],      "otherPaket": [        {          "namaPenerima": ,          "tag": "",          "waktuKedatangan": "",          "deskripsi": "",          "status": "",          "pengirim": "",          "id": ""        }      ]    },    "message": "Data  Sukses Didapat"  } </pre>
17	/api/paket/ <paket id>	GET	200	Data Sukses Didapat		<pre> {   "status": 200,    "data": [      {        "namaPenerima": ,        "tag": "",        "waktuKedatangan": "",        "deskripsi": "",        "waktuDiambil": ,     }   ] } </pre>

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
						<pre>     "pengirim": "",      "posisiPaket": "",      "statusPaket": ""      }  ],  "message": "Data  Sukses Didapat"  } </pre>
18	/api/dataAsrama	GET	200	Data Sukses Didapat		<pre> {   "status": 200,   "data": [     {       "teman_sekamar": [         {           "nama": "",            "tahun_masuk": "",            "prodi": ""         }       ],       "pembinaAsrama": [         {           "nama": ""         }       ],       "asrama": [         {           "nama_asrama": "",            "lokasi_asrama": "",            "kamar": "",            "kamar_id": "",            "asrama_id": ""         }       ],       "message": "Data  Sukses Didapat"     }   ] } </pre>
19	/api/kuesioner	GET	200	Data Sukses Didapat		<pre> {   "status": 200,   "data": [     {       "kuesioner_id": 3755,        "sem": "",        "ta": "",        "kode_mk": "",        "nama": "",        "keterangan": "",        "instruksi_pengisian": "",        "start_date": "",        "end_date": ""     }   ] } </pre>

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
						<pre> "peserta_kuis": "",  "is_login": "",  "nilai": "",  "wajib": "",  "deleted": "",  "deleted_at": "",  "deleted_by": "",  "created_at": "",  "updated_at": "",  "created_by": "",  "updated_by": "",  "id_kuesioner": ""  }  ],  "message": "Data  Sukses Didapat"  } </pre>
20	/api/kuesionerWajib	GET	200	Data Sukses Didapat		<pre>{   "status": 200,    "data": [],    "message": "Data  Sukses Didapat"  } </pre>
21	/api/pertanyaan/pertanyaanId	GET	200	Data Sukses Didapat		<pre> {   "status": 200,    "data": [      {        "kuesioner_pertanyaan_id": "",        "kuesioner_id": "",        "nomor": "",        "pertanyaan": "<p>&lt;/p&gt;","        "tipe_opsi": "",        "kategori": "",        "deleted": "",        "deleted_at": "",        "deleted_by": "",        "created_at": "",        "updated_at": "",        "created_by": "",        "updated_by": "",        "id_kuesioner": "",        "id_pertanyaan": "",        "opsi": [          {            "no_opsi": "",            "ket_opsi": ""          },          ...       ]     }   ] } </p></pre>

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
						<pre>{   "no_opsi": "",   "ket_opsi": "" }, {   "no_opsi": "",   "ket_opsi": "" }, {   "no_opsi": "",   "ket_opsi": "" } ], "message": "Data Sukses Didapat" }</pre>
22	/api/izinKeluar/{dimId}	POST	200	"Izin Keluar berhasil direquest"	<pre>{   "rencana_berangkat": "",   "rencana_kembali": "",   "keperluanIK": "" }</pre>	<pre>{   "status": 200,   "data": null,   "message": "Izin Keluar berhasil direquest" }</pre>
23	/api/izinKeluar/{id}/{izinId}	GET	200	"Data Sukses Didapat"		<pre>{   "status": 200,   "data": [     {       "izinKeluarId": "",       "keperluanIk": "",        "rencanaBerangkat": "",       "rencanaKembali": "",        "realisasiBerangkat": ,       "realisasiKembali": ,       "statusDosen": "",        "statusKeasramaan": ,       "statusBaak": "",        "namaDosen": ,       "namaKeasramaan": ,       "namaBaak": ""     }   ],   "message": "Data Sukses Didapat" }</pre>

No	URI	HTTP Methods	HTTP Response		JSON Request	JSON Response
			Code	Header Message		
						}
24	/api/izinKeluар/{izinId}/{id}	PUT	200	“Data Berhasil Diupdate”	{           “rencana_berangkat”: “”, “rencana_kembali”: “”, “keperluanIK”: “”         }	{           “status”: 200, “data”: { “rencana_berangkat” : “”, “rencana_kembali”: “”, “keperluanIK”: “” }, “message” : “Data Berhasil Diupdate”         }

### Lampiran 3 - Use Case Scenario

Berikut akan dijelaskan tentang *use case scenario* sistem dari aplikasi yang akan dibangun.

#### 1. Use Case Scenario Login

Pada Tabel 31 berikut merupakan *use case scenario login*.

**Tabel 31. Use Case Scenario Login**

Use Case ID Number	UC-01	
Use Case Name	<i>Login</i>	
Brief Description	<i>Use case</i> ini digunakan agar mahasiswa dapat masuk ke dalam sistem.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa sudah terhubung pada jaringan <i>smartphone</i> . PRE-2 Mahasiswa sudah membuka halaman <i>login</i> .	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa mengisi <i>form login</i> dan mengklik tombol <i>login</i> .	
Post Condition	2. Sistem menampilkan halaman utama sistem.	
	Mahasiswa dapat masuk ke dalam sistem.	
Alternate Flow of Events	User Action	System Response
	1a. <i>Username</i> dan <i>password</i> tidak diisi atau kosong.  1b. Mahasiswa mengisi <i>Username</i> dan <i>password</i> yang salah pada <i>form</i> .	2a. Sistem menampilkan <i>alert username</i> dan <i>password</i> kosong.  2b. Sistem menampilkan halaman <i>login</i> dan pemberitahuan <i>Username</i> dan <i>Password</i> salah.

#### 2. Use Case Scenario Logout

Pada Tabel 32 berikut merupakan *use case scenario logout*.

**Tabel 32. Use Case Scenario Logout**

Use Case ID Number	UC-02	
Use Case Name	<i>Logout</i>	
Brief Description	<i>Use case</i> ini digunakan agar mahasiswa dapat keluar dari dalam sistem.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i> .	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu <i>logout</i> .	
		2. Sistem kembali ke halaman <i>login</i> .
Post Condition	Mahasiswa telah keluar dari sistem.	

### 3. Use Case Scenario Melihat Notifikasi

Pada Tabel 33 berikut merupakan *use case scenario* Melihat Notifikasi.

**Tabel 33. Use Case Scenario Melihat Notifikasi**

Use Case ID Number	UC-03	
Use Case Name	Melihat Notifikasi	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk melihat notifikasi dari sistem.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah terhubung ke jaringan <i>smartphone</i> . PRE-2 Mahasiswa telah <i>login</i> .	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa mengklik notifikasi.	

	2. Mahasiswa melihat notifikasi	
		3. Sistem menampilkan notifikasi
Post Condition	Mahasiswa dapat melihat pemberitahuan baru yang masuk melalui notifikasi yang didapat.	

#### 4. Use Case Scenario Add Request Izin Bermalam (IB)

Pada Tabel 34 berikut merupakan *use case scenario Add Request Izin Bermalam (IB)*

**Tabel 34. Use Case Scenario Add Request IB**

Use Case ID Number	UC-04	
Use Case Name	<i>Add Request Izin Bermalam (IB)</i>	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk menambahkan <i>request IB</i> atau melakukan <i>request IB</i> .	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu Asrama dan Kemahasiswaan.	
	2. Mahasiswa memilih menu Izin Bermalam	
		3. Sistem menampilkan halaman Izin Bermalam.
	4. Mahasiswa memilih tombol “request IB”.	
		5. Sistem menampilkan <i>form request</i> Izin Bermalam.
	6. Mahasiswa mengisi <i>form request</i> IB dan tombol “Buat Baru”.	
		7. Sistem menyimpan dan mengirimkan <i>request</i> .

	8. Mahasiswa menunggu <i>approve</i> dari keasramaan.	
	9. Mahasiswa mendapatkan notifikasi IB telah disetujui.	
	10. Mahasiswa mengklik notifikasi yang masuk.	
		11. Sistem menampilkan halaman Izin Bermalam.
	12. Mahasiswa mengklik tombol “tools” dan memilih <i>print</i> .	
		13. Sistem menampilkan lebar IB yang telah disetujui dan akan di <i>print</i> .
Post Condition	Mahasiswa dapat melakukan Izin Bermalam.	

## 5. Use Case Scenario Edit Request Izin Bermalam (IB)

Pada Tabel 35 berikut merupakan *use case scenario* Edit Request Izin Bermalam (IB).

**Tabel 35. Use Case Scenario Edit Request IB**

Use Case ID Number	UC-05	
Use Case Name	Edit Request Izin Bermalam (IB)	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk mengedit form IB yang telah di <i>request</i> sebelumnya	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i> . PRE-2 Mahasiswa telah mengirim <i>request</i> Izin Bermalam (IB). PRE-3 Mahasiswa sedang menunggu <i>approve</i> dari keasramaan.	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu Izin Bermalam.	
		2. Sistem menampilkan halaman Izin Bermalam.
	3. Mahasiswa mengklik tombol “tools” dan memilih Edit dari <i>request</i> IB yang telah dibuat	

	sebelumnya dan memiliki status permohonan sedang menunggu <i>approve</i> dari keasramaan.	
	4. Sistem menampilkan <i>form</i> edit data izin bermalam.	
	5. Mahasiswa mengubah <i>form</i> IB yang telah diisi sebelumnya dan menyesuaikan dengan kebutuhan mahasiswa tersebut.	
	6. Mahasiswa mengklik tombol “selesai edit”.	
	7. Sistem menyimpan hasil edit dan mengirim <i>request</i> .	
	8. Mahasiswa menunggu <i>approve</i> dari keasramaan.	
Post Condition	Mahasiswa berhasil mengedit <i>form</i> IB yang telah di <i>request</i> sebelumnya dan menunggu approve dari keasramaan.	
Alternate Flow of Events	User Action	System Response
	1a. Mahasiswa batal melakukan edit terhadap <i>form</i> IB nya.	2a. Sistem kembali menampilkan halaman Izin Bermalam yang sebelumnya.

## 6. Use Case Scenario Cancel Request Izin Bermalam (IB)

Pada Tabel 36 berikut merupakan *use case scenario* *Cancel Request Izin Bermalam (IB)*.

**Tabel 36. Use Case Scenario Cancel Request IB**

Use Case ID Number	UC-06
Use Case Name	<i>Cancel Request Izin Bermalam (IB)</i>
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk meng <i>cancel</i> atau membatalkan <i>request</i> Izin Bermalam (IB) yang telah dilakukan sebelumnya.
Actor	Mahasiswa
Precondition	PRE-1 Mahasiswa telah <i>login</i> PRE-2 Mahasiswa telah mengirim <i>request</i> Izin Bermalam (IB) PRE-3 Mahasiswa sedang menunggu approve dari keasramaan

Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu Izin Bermalam.	
	2. Sistem menampilkan halaman Izin Bermalam.	
	3. Mahasiswa mengklik tombol “tools” dan memilih Cancel dari request IB yang telah dibuat sebelumnya dan memiliki status permohonan sedang menunggu approve dari keasramaan.	
	4. Sistem menampilkan halaman Izin Bermalam dengan status <i>request</i> yang sebelumnya telah dibatalkan.	
Post Condition	Mahasiswa berhasil meng cancel IB dan tidak dapat melakukan IB.	

## 7. Use Case Scenario Add Request Izin Keluar (IK)

Pada Tabel 37 berikut merupakan *use case scenario Add Request Izin Keluar (IK)*

**Tabel 37. Use Case Scenario Add Request IK**

Use Case ID Number	UC-07	
Use Case Name	<i>Add Request Izin Keluar (IK)</i>	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk menambahkan <i>request</i> IK atau melakukan <i>request</i> IK	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu Asrama dan Kemahasiswaan.	
	2. Mahasiswa memilih menu Izin Keluar.	

		3. Sistem menampilkan halaman Izin Keluar.
	4. Mahasiswa memilih tombol “request Izin Keluar”.	
		5. Sistem menampilkan <i>form request</i> Izin Keluar.
	6. Mahasiswa mengisi <i>form request</i> Izin Keluar dan tombol “Buat Baru”	
		7. Sistem menyimpan dan mengirimkan <i>request</i> .
	8. Mahasiswa menunggu approve dari Dosen atau TA Matakuliah, Dosen Wali, Keasramaan, dan BAAK	
	9. Mahasiswa mendapatkan notifikasi IK telah disetujui	
	10. Mahasiswa mengklik notifikasi yang masuk	
		11. Sistem menampilkan halaman Izin Keluar
	12. Mahasiswa mengklik tombol “tools” dan memilih print	
		13. Sistem menampilkan lebar IK yang telah disetujui dan akan di print
Post Condition	Mahasiswa dapat melakukan Izin Keluar	

## 8. Use Case Scenario Edit Request Izin Keluar (IK)

Pada Tabel 38 berikut merupakan *use case scenario* Edit Request Izin Keluar (IK)

**Tabel 38. Use Case Scenario Edit Request IK**

Use Case ID Number	UC-08
Use Case Name	Edit Request Izin Keluar (IK)
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk mengedit <i>form</i> IK yang telah di <i>request</i> sebelumnya
Actor	Mahasiswa

Precondition	PRE-1 Mahasiswa telah <i>login</i> PRE-2 Mahasiswa telah mengirim <i>request</i> Izin Keluar (IK) PRE-3 Mahasiswa sedang menunggu approve dari Dosen Matakuliah atau TA Matakuliah	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu Izin Keluar	
		2. Sistem menampilkan halaman Izin Keluar
	3. Mahasiswa mengklik tombol “tools” dan memilih Edit dari <i>request</i> IK yang telah dibuat sebelumnya dan memiliki status permohonan sedang menunggu approve dari Dosen Matakuliah atau TA Matakuliah	
		4. Sistem menampilkan <i>form Update</i> Izin
	5. Mahasiswa mengubah <i>form</i> IK yang telah diisi sebelumnya dan menyesuaikan dengan kebutuhan mahasiswa tersebut	
	6. Mahasiswa mengklik tombol “selesai edit”	
		7. Sistem menyimpan hasil edit dan mengirim <i>request</i>
	8. Mahasiswa menunggu approve dari Dosen Matakuliah atau TA Matakuliah	
Post Condition	Mahasiswa berhasil mengedit <i>form</i> IK yang telah di <i>request</i> sebelumnya dan menunggu approve dari Dosen Matakuliah atau TA Matakuliah	
Alternate Flow of Events	User Action	System Response
	1a. Mahasiswa batal melakukan edit terhadap <i>form</i> IK nya	2a. Sistem kembali menampilkan halaman Izin Keluar yang sebelumnya

## 9. Use Case Scenario Cancel Request Izin Keluar (IK)

Pada Tabel 39 berikut merupakan *use case* scenario *Cancel Request Izin Keluar (IK)*

**Tabel 39. Use Case Scenario Cancel Request IK**

Use Case ID Number	UC-09	
Use Case Name	<i>Cancel Request Izin Keluar (IK)</i>	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk meng <i>cancel</i> atau membatalkan <i>request</i> Izin Keluar (IK) yang telah dilakukan sebelumnya.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i> PRE-2 Mahasiswa telah mengirim <i>request</i> Izin Keluar (IK) PRE-3 Mahasiswa sedang menunggu approve dari Dosen Matakuliah atau TA Matakuliah	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu Izin Keluar	
		2. Sistem menampilkan halaman Izin Keluar
	3. Mahasiswa mengklik tombol “tools” dan mengklik Cancel dari <i>request</i> IK yang telah dibuat sebelumnya dan memiliki status <i>request</i> sedang menunggu approve dari Dosen Matakuliah/ TA Matakuliah	
		4. Sistem menampilkan halaman Izin Keluar dengan status <i>request</i> yang sebelumnya telah dibatalkan
Post Condition	Mahasiswa berhasil meng cancel IK dan tidak dapat melakukan IK	

## 10. Use Case Scenario Melihat Mata Kuliah

Pada Tabel 40 berikut merupakan *use case scenario* Melihat Matakuliah

**Tabel 40. Use Case Scenario Melihat Matakuliah**

Use Case ID Number	UC-10
Use Case Name	Melihat Matakuliah

Brief Description	Use case ini digunakan oleh mahasiswa untuk melihat daftar-daftar mata kuliah.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i> ke sistem	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu perkuliahan dan menu lihat matakuliah.	
		2. Sistem menampilkan halaman pencarian mata kuliah berdasarkan tahun kurikulum dan prodi.
	3. Mahasiswa memilih tahun kurikulum dan prodi. Lalu memilih tombol “cari”.	
		4. Sistem menampilkan daftar-daftar mata kuliah sesuai dengan tahun kurikulum dan prodi yang telah di input oleh mahasiswa.
Post Condition	Mahasiswa dapat melihat daftar-daftar mata kuliah.	

## 11. Use Case Scenario Unduh Modul

Pada Tabel 41 berikut merupakan *use case scenario* Unduh Modul.

**Tabel 41. Use Case Scenario Unduh Modul**

Use Case ID Number	UC-11
Use Case Name	Unduh Modul
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk mengunduh modul perkuliahan
Actor	Mahasiswa
Precondition	PRE-1 Mahasiswa telah <i>login</i>

Primary Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau dashboard	
	2. Sistem menampilkan halaman utama atau halaman dashboard	
	3. Mahasiswa memilih matakuliah yang ingin di unduh modulnya	
	4. Sistem menampilkan halaman matakuliah yang telah dipilih mahasiswa	
	5. Mahasiswa memilih antara materi kuliah atau materi praktikum dari halaman yang telah ditampilkan	
	6. Sistem menampilkan antara halaman materi kuliah atau materi praktikum yang dipilih mahasiswa	
	7. Mahasiswa memilih week yang akan di unduh modulnya	
	8. Mahasiswa memilih materi yang akan di unduh dari week yang sudah dipilih sebelumnya	
	9. Sistem menampilkan halaman materi yang dipilih mahasiswa	
	10. Mahasiswa mengklik materi yang sudah dipilih	
	11. Sistem menampilkan halaman detail materi yang diklik mahasiswa	
	12. Mahasiswa mengklik file yang ada pada halaman detail materi	
Post Condition	Mahasiswa dapat mengunduh modul materi perkuliahan yang diinginkannya.	
Alternate Flow of Events	User Action	System Response

	1. Mahasiswa masuk ke halaman utama atau dashboard	
	2. Sistem menampilkan halaman utama atau dashboard	
	3. Mahasiswa memilih menu perkuliahan	
	4. Sistem menampilkan halaman matakuliah	
	5. Mahasiswa mencari matakuliah dengan memasukkan tahun kurikulum dan prodi serta jenjang	
	6. Mahasiswa mengklik tombol “cari” setelah mengisi yang	
	7. Sistem menampilkan halaman yang dicari mahasiswa	
	8. Mahasiswa memilih semester dan matakuliah yang akan di unduh modulnya	
	9. Mahasiswa memilih antara materi kuliah atau materi praktikum dari halaman yang telah ditampilkan	
	10. Sistem menampilkan antara halaman materi kuliah atau materi praktikum yang dipilih mahasiswa	
	11. Mahasiswa memilih week yang akan di unduh modulnya	
	12. Mahasiswa memilih materi yang akan di unduh dari week yang sudah dipilih sebelumnya	
	13. Sistem menampilkan halaman materi yang dipilih mahasiswa	
	14. Mahasiswa mengklik materi yang sudah dipilih	
	15. Sistem menampilkan halaman detail materi yang diklik mahasiswa	

	16. Mahasiswa mengklik file yang ada pada halaman detail materi	
--	-----------------------------------------------------------------	--

## 12. Use Case Scenario Melihat Jadwal Perkuliahuan

Pada Tabel 42 berikut merupakan *use case scenario* Melihat Jadwal Perkuliahuan.

**Tabel 42. Use Case Scenario Melihat Jadwal Perkuliahuan**

Use Case ID Number	UC-12	
Use Case Name	Melihat Jadwal Perkuliahuan	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk dapat melihat jadwal perkuliahan (waktu perkuliahan, mata kuliah, dan ruangan) yang diikutinya selama satu hari tersebut dan satu minggu kedepan.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau dashboard	
		2. Sistem menampilkan halaman utama atau halaman dashboard
	3. Mahasiswa melihat jadwal langsung di halaman utama di bagian bawah daftar course yang dimiliki mahasiswa (Jadwal yang dilihat adalah jadwal hari itu saja)	
Post Condition	Mahasiswa dapat melihat jadwal perkuliahananya.	
Alternate Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau dashboard.	
		2. Sistem menampilkan halaman utama atau halaman dashboard.

	3. Mahasiswa memilih menu perkuliahan.	
	4. Mahasiswa memilih menu jadwal mahasiswa.	
		5. Sistem menampilkan halaman Jadwal.
	6. Mahasiswa dapat melihat jadwal selama satu minggu penuh.	

### 13. Use Case Scenario Melihat Data Diri

Pada Tabel 43 berikut merupakan *use case scenario* Melihat Data Diri.

**Tabel 43. Use Case Scenario Melihat Data Diri**

Use Case ID Number	UC-13	
Use Case Name	Melihat Data Diri.	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk melihat data dirinya.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	
		2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .
	3. Mahasiswa memilih menu data induk mahasiswa.	
	4. Mahasiswa memilih menu data diri.	
		5. Sistem menampilkan Halaman Data Diri Mahasiswa.
Post Condition	Mahasiswa dapat melihat data akademis, pribadi, data orang tua, data asal sekolah, data asrama dan perilaku, dan hasil studi.	

### 14. Use Case Scenario Edit Data Diri

Pada Tabel 44 berikut merupakan *use case scenario* Edit Data Diri.

**Tabel 44. Use Case Scenario Edit Data Diri**

Use Case ID Number	UC-14	
Use Case Name	Edit Data Diri	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk dapat mengedit data diri yaitu data pribadi, data orang tua dan data asal sekolah	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	
		2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .
	3. Mahasiswa memilih menu data induk mahasiswa.	
	4. Mahasiswa memilih menu data diri.	
		5. Sistem menampilkan halaman data diri mahasiswa.
	6. Mahasiswa melihat antara data pribadi, data orang tua, dan data asal sekolah.	
		7. Sistem menampilkan halaman yang dipilih mahasiswa.
	8. Mahasiswa memilih menu edit data induk.	
		9. Sistem menampilkan halaman edit data pribadi, edit data orang tua atau edit data asal sekolah.
	10. Mahasiswa melakukan edit terhadap data yang ingin diubah.	
	11. Mahasiswa mengklik tombol “save” setelah selesai	

	mengedit atau melakukan perubahan data.	
		12. Sistem menyimpan perubahan data yang dilakukan mahasiswa.
		13. Sistem menampilkan halaman data diri mahasiswa.
Post Condition	Mahasiswa berhasil mengedit atau melakukan perubahan terhadap data terhadap data pribadi, data orang tua atau data asal sekolah.	

## 15. Use Case Scenario Melihat Kuesioner

Pada Tabel 45 berikut merupakan *use case scenario* Melihat Kuesioner.

**Tabel 45. Use Case Scenario Melihat Kuesioner**

Use Case ID Number	UC-15	
Use Case Name	Melihat Kuesioner	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk dapat melihat kuesioner.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
		1. Sistem menampilkan kuesioner yang wajib untuk mahasiswa.
	2. Mahasiswa melihat kuesioner wajib yang harus diisi.	
Post Condition	Mahasiswa melihat kuesioner yang wajib untuk diisi dan lanjut dengan pengisian kuesioner	
Alternate Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	
	2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .	

	3. Mahasiswa memilih menu <i>survey</i> .	
	4. Mahasiswa memilih menu kuesioner.	
		5. Sistem menampilkan halaman kuesioner
	6. Mahasiswa memilih kuesioner yang ingin dilihat dan akan diisi.	

## 16. Use Case Scenario Mengisi Kuesioner

Pada Tabel 46 berikut merupakan *use case scenario* Mengisi Kuesioner

**Tabel 46. Use Case Scenario Mengisi Kuesioner**

Use Case ID Number	UC-16	
Use Case Name	Mengisi Kuesioner	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk mengisi kuesioner yang masuk dan kuesioner yang dipilih untuk diisi.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
		1. Sistem menampilkan kuesioner yang wajib untuk mahasiswa.
	2. Mahasiswa melihat kuesioner wajib yang harus diisi.	
	3. Mahasiswa mengklik kuesioner yang ada.	
		4. Sistem menampilkan halaman kuesioner yang akan diisi.
	5. Mahasiswa mengisi kuesioner yang telah ditampilkan dan mengklik tombol “submit”.	
		6. Sistem mengirim hasil tanggapan dari mahasiswa.
Post Condition	Mahasiswa berhasil mengisi kuesioner dan dapat masuk ke halaman utama atau <i>dashboard</i> .	

Alternate Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	
	2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .	
	3. Mahasiswa memilih menu <i>survey</i> .	
	4. Mahasiswa memilih menu kuesioner.	
	5. Sistem menampilkan halaman kuesioner.	
	6. Mahasiswa memilih kuesioner yang ingin dilihat dan akan diisi.	
	7. Mahasiswa mengklik kuesioner yang dipilihnya.	
	8. Sistem menampilkan halaman kuesioner yang akan diisi.	
	9. Mahasiswa mengisi kuesioner yang telah ditampilkan dan mengklik tombol “submit”.	
	10. Sistem mengirim hasil tanggapan dari mahasiswa.	

## 17. Use Case Scenario Melihat Pengumuman

Pada Tabel 47 berikut merupakan *use case scenario* Melihat Pengumuman

**Tabel 47. Use Case Scenario Melihat Pengumuman**

Use Case ID Number	UC-17
Use Case Name	Melihat Pengumuman
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk melihat pengumuman yang masuk.
Actor	Mahasiswa
Precondition	PRE-1 Mahasiswa telah <i>login</i>

Primary Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	
	2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .	
	3. Mahasiswa melihat langsung pengumuman yang berada di halaman utama atau halaman <i>dashboard</i> .	
	4. Mahasiswa memilih dan mengklik pengumuman yang ingin dilihat.	
	5. Sistem menampilkan isi pengumuman yang diklik oleh mahasiswa.	
Post Condition	Mahasiswa dapat melihat pengumuman yang diinginkan.	
Alternate Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	
	2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .	
	3. Mahasiswa memilih menu tambahan.	
	4. Mahasiswa memilih menu pengumuman.	
	5. Sistem menampilkan halaman daftar pengumuman.	
	6. Memilih pengumuman yang ingin dilihat.	
	7. Menampilkan pengumuman yang dipilih mahasiswa.	
Alternate Flow of Events	User Action	System Response

	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	
	2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .	
	3. Mahasiswa memilih menu tambahan.	
	4. Mahasiswa memilih menu pengumuman.	
	5. Sistem menampilkan halaman daftar pengumuman.	
	6. Mahasiswa mencari pengumuman yang ingin dilihat dengan mengisi kategori dan judul pengumuman.	
	7. Mahasiswa mengklik tombol “Cari”.	
	8. Sistem mencari pengumuman yang dimaksud.	
	9. Sistem menampilkan hasil pencarinya.	
	10. Mahasiswa melihat pengumuman yang ditampilkan di halaman pengumuman.	
	11. Mahasiswa mengklik pengumuman tersebut.	
	12. Sistem menampilkan isi pengumuman yang diklik oleh mahasiswa.	

## 18. Use Case Scenario Melihat Paket

Pada Tabel 48 berikut merupakan *use case scenario* Melihat Paket

**Tabel 48. Use Case Scenario Melihat Paket**

Use Case ID Number	UC-18
Use Case Name	Melihat Paket

Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk dapat melihat paket yang diterima di pos satpam.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
		1. Mengirimkan notifikasi ada paket kepada mahasiswa.
	2. Menerima notifikasi ada paket dan membuka notifikasi tersebut.	
	3. Mahasiswa berada di halaman data paket dan melihat detail paket yang diterimanya.	
Post Condition	Mahasiswa dapat menjemput paket ke pos satpam sesuai dengan detail yang diterimanya.	

## 19. Use Case Scenario Mengisi Kartu Rencana Studi (KRS)

Pada Tabel 49 berikut merupakan *use case scenario* Mengisi Kartu Rencana Studi (KRS)

**Tabel 49. Use Case Scenario Mengisi KRS**

Use Case ID Number	UC-19	
Use Case Name	Mengisi Kartu Rencana Studi (KRS)	
Brief Description	<i>Use case</i> ini digunakan	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	

		2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .
	3. Mahasiswa memilih menu Administrasi Akademik.	
	4. Mahasiswa memilih menu Kartu Rencana Studi.	
		5. Sistem akan menampilkan halaman Rencana Studi dan Hasil Studi.
	6. Mahasiswa mengisi KRS dengan memilih matakuliah yang akan diambil selama satu semester dari list yang sudah ada.	
	7. Mahasiswa mengirim <i>request</i> KRS .	
		8. Sistem menerima <i>request</i> dan mengirim info kepada dosen wali mahasiswa yang bersangkutan.
Post Condition	Mahasiswa berhasil mengisi KRS dan menunggu approve dari dosen walinya.	

## 20. Use Case Scenario Edit Kartu Rencana Studi (KRS)

Pada Tabel 50 berikut merupakan *use case scenario* Edit Kartu Rencana Studi (KRS)

**Tabel 50. Use Case Scenario Edit KRS**

Use Case ID Number	UC-20	
Use Case Name	Edit Kartu Rencana Studi (KRS)	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk dapat mengedit KRS yang sebelumnya telah di <i>request</i> .	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i> PRE-2 Mahasiswa telah mengirim <i>request</i> Kartu Rencana Studi (KRS) PRE-3 Mahasiswa sedang menunggu approve dari Dosen Wali	
Primary Flow of Events	User Action	System Response

	1. Mahasiswa memilih menu Perubahan KRS.	
	2. Sistem menampilkan halaman Edit KRS.	
	3. Mahasiswa melakukan edit terhadap KRS yang akan diambilnya selama satu semester kedepan.	
	4. Mahasiswa mengklik tombol “save” untuk menyimpan perubahan yang dilakukannya.	
	5. Sistem menyimpan hasil edit dan mengirim <i>request</i> .	
	6. Mahasiswa menunggu <i>approve</i> dari Dosen Wali.	
Post Condition	Mahasiswa berhasil mengedit KRS dan menunggu approve dari dosen walinya	

## 21. Use Case Scenario Melihat Grafik Indeks Prestasi (IP)

Pada Tabel 51 berikut merupakan *use case scenario* Melihat Grafik Indeks Prestasi (IP)

**Tabel 51. Use Case Scenario Melihat Grafik IP**

Use Case ID Number	UC-21	
Use Case Name	Melihat Grafik Indeks Prestasi (IP)	
Brief Description	<i>Use case</i> ini digunakan oleh mahasiswa untuk dapat melihat grafik indeks prestasi (IP) yang dimilikinya.	
Actor	Mahasiswa	
Precondition	PRE-1 Mahasiswa telah <i>login</i>	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau halaman <i>dashboard</i> .	
		2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .

	3. Mahasiswa melihat grafik indeks prestasi (IP) nya langsung di halaman utama di bagian bawah daftar course dan jadwal.	
Post Condition	Mahasiswa dapat melihat grafik indeks prestasi (IP)	
Alternate Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman utama atau <i>dashboard</i> .	
	2. Sistem menampilkan halaman utama atau halaman <i>dashboard</i> .	
	3. Mahasiswa memilih menu data induk mahasiswa.	
	4. Mahasiswa memilih menu data diri.	
	5. Sistem menampilkan halaman data diri mahasiswa yang bersangkutan	
	6. Mahasiswa memilih menu hasil studi.	
	7. Sistem menampilkan halaman hasil studi.	
	8. Mahasiswa dapat melihat grafik indeks prestasi nya.	

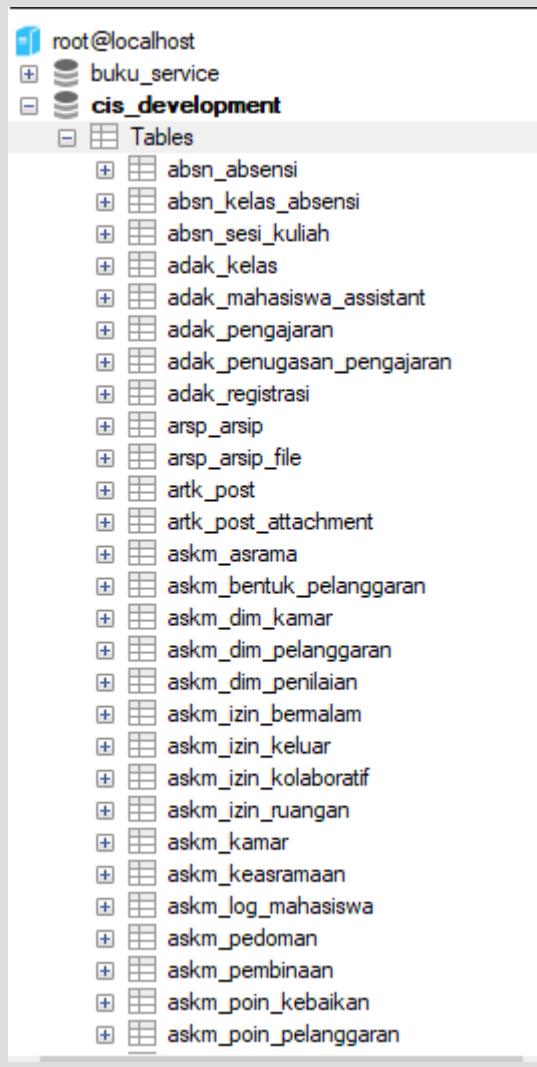
## Lampiran 4 – Implementasi Aplikasi

Pada Lampiran 5 ini menjelaskan implementasi aplikasi yang telah dilakukan.

### 1. Pembuatan Database MySQL

Pada tahap ini tim pengembang menggunakan database sistem saat ini yang sudah digunakan dalam pembangunan *current system* yaitu CIS berbasis *website*. Database yang berisi nama database, tabel berserta dengan atribut dari setiap tabel. Berikut merupakan struktur database dalam pengembangan proyek Tugas Akhir.

Nama database yang digunakan adalah *cis\_development*. Tabel yang digunakan yaitu tabel *sysx\_user*, *dimx\_dim*, *mref\_r\_agama*, *mref\_r\_penghasilan*, *mref\_r\_pekerjaan*, *mref\_r\_jenis\_kelamin*, *mref\_r\_asal\_sekolah*, *askm\_r\_status\_request*, *askm\_izin\_bermalam*, *askm\_dim\_pelanggaran*, *askm\_pembinaan*, *askm\_poin\_pelanggaran*, *askm\_poin\_kebaikan*, *askm\_bentuk\_pelanggaran*, *askm\_dim\_penilaian*, *askm\_dim\_kamar*, *askm\_kamar*, *askm\_keasramaan*, *askm\_asrama*, *tmbh\_pengumuman*, *tmbh\_file\_pengumuman*, *srvy\_kuesioner\_opsi*, *srvy\_kuesioner\_jawaban\_peserta*, *srvy\_kuesioner\_pertanyaan*, *srvy\_kuesioner*, *uvux\_r\_status\_paket*, *ubux\_r\_paket* dan *ubux\_data\_paket*.



Gambar 111. Pembuatan Database MySQL

## 2. Pembuatan API

Berikut merupakan potongan kode dalam pembuatan API yang terdiri dari model, controller dan route.

```

File Edit Selection View Go Run Terminal Help DimController.php - TA workspace (Workspace) - Visual Studio Code
EXPLORER DimController.php 7, M DimDim.php .env testNotif app.php testNotif + config StatusLiked.php testNotif + app...
AskmDimPelanggaran.php AskmDimPenilaian.php AskmPoinKebaikan.php
DimDim.php FilePengumuman.php IzinBermalam.php Keasramaan.php KrmTahunKurikulum.php Kuesioner.php KuesionerLawabahPeserta.php
MrefAgama.php MrefAsalsekolah.php MrefGolonganDarah.php MrefJenisKelamin.php MrefPekerjaan.php MrefPenghasilan.php OpsiPertanyaan.php PertanyaanKuesioner.php SysUser.php TmbhPengumuman.php UbusDataPaket.php User.php > bootstrap > config > database > OUTLINE > TIMELINE > MAVEN
master* 44 0 Type here to search 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class DimDim extends Model
{
    protected $table = 'dimx_dim';
    protected $primaryKey = 'dim_id';

    public function sysxuser()
    {
        return $this->belongsTo('App\SysxUser');
    }
    public function izinbermalam()
    {
        return $this->hasMany('App\IzinBermalam');
    }
    public function ubuxdatapaket()
    {
        return $this->hasMany('App\UbxDataPaket');
    }
}

```

Gambar 112. API Model

*Controller* akan dibuat pada direktori **./app/Http/Controllers** dengan meng-*import* **Illuminate\Http\Request** pada *Controller* yang dibuat secara otomatis. Di dalam *class Controller* yang telah dibuat, dapat dibuat suatu *endpoint* berupa fungsi-fungsi yang dipakai untuk memenuhi kebutuhan *request* yang datang dari luar.

```

File Edit Selection View Go Run Terminal Help DimController.php - TA workspace (Workspace) - Visual Studio Code
EXPLORER DimController.php 7, M .env testNotif app.php testNotif + config StatusLiked.php testNotif + app... welcome.blade.php test
idea vscode
> app
> Console
> Events
> NotificationsEvent.php
> Exceptions
> Http
> Controllers
Auth Service Controller.php DimController.php IzinBermalamController.php JadwalController.php KeasramaanController.php KuesionerController.php ResponseController.php TambahanController.php UserController.php
master* 44 0 Type here to search 02 03 04 05 06 07 08 09 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091 092 093 094
HTTP > Controllers > DimController.php > PHP Intelephense > DimController > updateDataOrangtua
$puser->save();
return $this->successResponse(201,$user,"Data Berhasil di update");

public function getDataAkademis()
{
    $userLogin = \JWTAuth::parseToken()->authenticate();
    $id = $userLogin->user_id;
    $user = SysxUser::leftJoin("dimx_dim", function ($join) {
        $join->on('sysx_user.user_id', '=', 'dimx_dim.user_id');
    })->leftJoin("inst_prodi", function ($join) {
        $join->on("dimx_dim.ref_kbk_id", '=', 'inst_prodi.ref_kbk_id');
    })->leftJoin("adak_kelas", function ($join) {
        $join->on('inst_prodi.ref_kbk_id', '=', 'adak_kelas.prodi_id');
    })->leftJoin("hrdx_dosen", function ($join) {
        $join->on('adak_kelas.dosen_wali_id', '=', 'hrdx_dosen.dosen_id');
    })->leftJoin("hrdx_pegawai", function ($join) {
        $join->on('hrdx_dosen.pegawai_id', '=', 'hrdx_pegawai.pegawai_id');
    }))->where('dimx_dim.user_id', '=', $id)->first([
        'dimx_dim.status_akhir',
        'dimx_dim.thn_masuk',
        'dimx_dim.user_name',
        'dimx_dim.email as email_akademik',
        'inst_prodi.nama_kopertis_id as prodi',
        'adak_kelas.nama as kelas',
        'hrdx_pegawai.nama as wali_mahasiswa',
        'dimx_dim.jalur'
    ]);
    if ($user) {
        return $this->successResponse(200, $user);
    }
}

```

Gambar 113. API Controller

Untuk mendaftarkan seluruh *endpoint* yang ada dapat dilakukan pada direktori **./routes/api/php**.

The screenshot shows the Visual Studio Code interface with several tabs open. The active tab is 'api.php' which contains PHP code for API routes. The code includes routes for user authentication, academic data, student data, dormitory data, and questionnaires. The sidebar on the left shows the project structure with files like 'OpsiPertanyaan.php', 'PertanyaanKuesioner.php', 'SysUser.php', 'TmbhPengumuman.php', 'UbusDataPaket.php', 'User.php', 'bootstrap', 'config', 'database', 'public', 'resources', and 'routes'. The bottom status bar shows the file is 9+, M, and the current line is 41.

Gambar 114. API Routes

### 3. Pembuatan *Project* pada Android Studio

Setelah *project* telah terbuka dengan baik maka yang dilakukan adalah dengan menambahkan perintah permission Internet pada *file Manifest*. Hal ini berfungsi agar aplikasi yang dibangun dapat mengakses internet.

The screenshot shows the 'AndroidManifest.xml' file in Android Studio. It contains XML code defining permissions. Specifically, it declares two permissions: 'android.permission.INTERNET' and 'android.permission.ACCESS\_NETWORK\_STATE'.

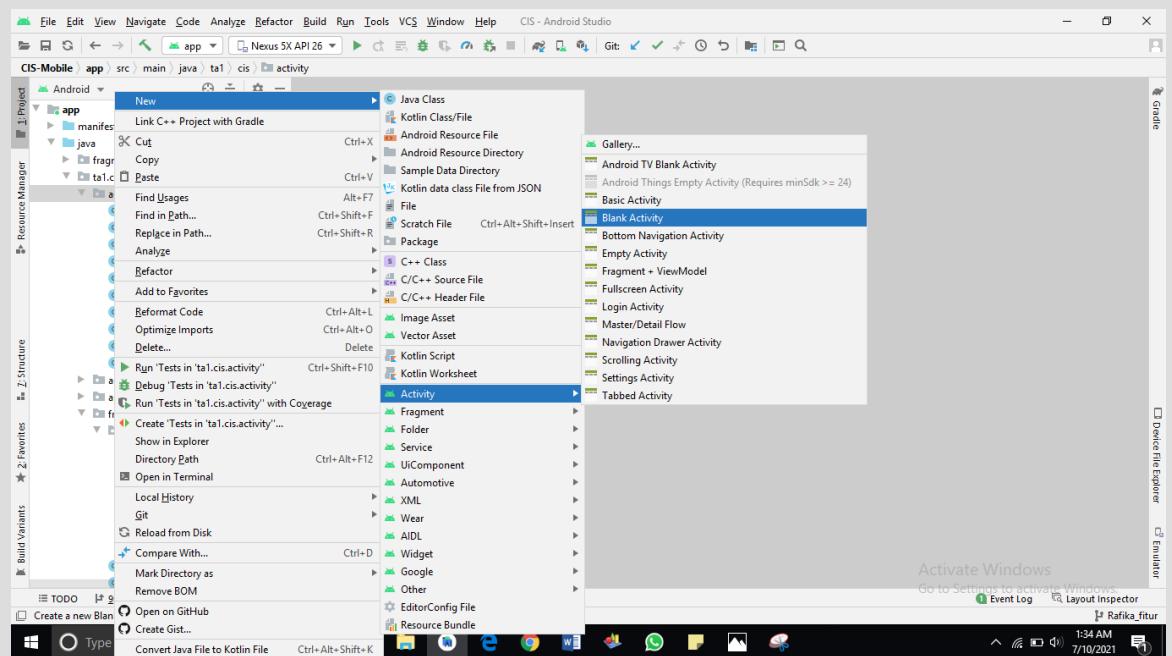
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ta1.cis">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Gambar 115. Pembuatan Project pada Android Studio

### 4. Pembuatan *Activity* dan *List Activity*

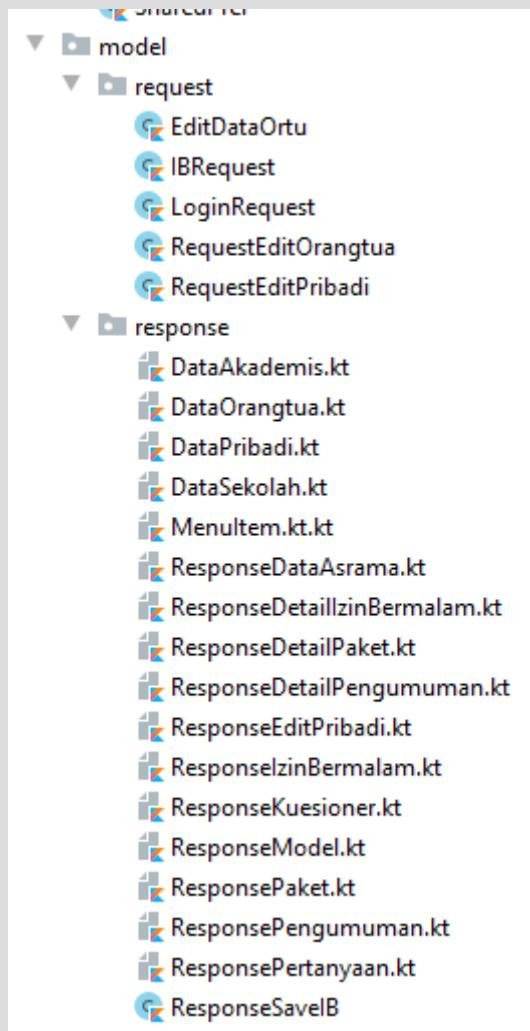
Berikut merupakan pembuatan *activity* dan *list activity* dalam sebuah android studio.



Gambar 116. Pembuatan Activity

## 5. Pembuatan Model

Pada *package* model terdiri dari model *request* dan *response*. Dalam model response terdapat beberapa model yaitu: *EditDataOrtu*, *IBRequest*, *LoginRequest*, *RequestEditOrangtua* dan *RequestEditPribadi*. Sama halnya dengan model request terdapat beberapa model yaitu: *DataAkademis*, *DataOrangtua*, *DataPribadi*, *DataSekolah*, *ManuItem*, *ResponseDetailIzinBermalam*, *ResponseDetailPaket*, *ResponseDetailPengumuman*, *ResponseEditPribadi*, *ResponseIzinBermalam*, *ResponseKuesioner*, *ResponseModel*, *ResponsePaket*, *ResponsePengumuman*, *ResponsePertanyaan* dan *ResponseSaveIB*. Berikut Gambar pembuatan model dalam android studio.



**Gambar 117. Pembuatan Model Pada Android Studio**

Berikut merupakan potongan kode pada model DataAkademis. Pada setiap *class* berisi representasi struktur dari database, yaitu berisi atribut-atribut yang sesuai dengan database.

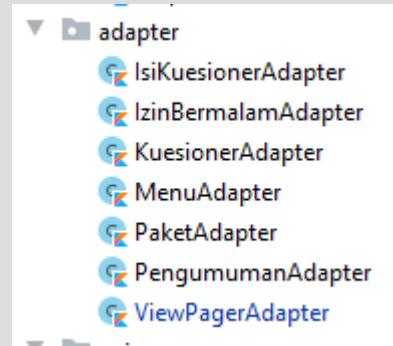
The screenshot shows a code editor window with two files open. The top file is `ResponseDataAsrama.kt`, which contains the following code:

```
1 package ta1.cis.model.response
2
3 import ...
4
5 @Parcelize
6 data class ResponseDataAsrama(
7
8     @field:SerializedName( value: "data")
9     val data: DataAsrama? = null,
10
11     @field:SerializedName( value: "message")
12     val message: String? = null,
13
14     @field:SerializedName( value: "status")
15     val status: Int? = null
16 ) : Parcelable
17
18 @Parcelize
19 data class Asrama(
20
21     @field:SerializedName( value: "nama_asrama")
22     val namaAsrama: String? = null,
23
24     @field:SerializedName( value: "lokasi_asrama")
25     val lokasiAsrama: String? = null,
26
27     @field:SerializedName( value: "kamar_id")
28     val kamarId: String? = null,
29
30 ) : Parcelable
```

Gambar 118. Potongan Kode Model Response

## 6. Pembuatan Adapter

Pembuatan `package` Adapter terdiri dari `IsiKuesionerAdapter`, `IzinBermalamAdapter`, `KuesionerAdapter`, `MenuAdapter`, `PaketAdapter`, `PengumumanAdapter` dan `ViewPagerAdapter`.



Gambar 119. Pembuatan Adapter

Berikut merupakan potongan kode pada adapter. Pada class PaketAdapter adalah *class* yang berisi atribut yang terdapat di dalam model yang akan menghubungkan terhadap sebuah *view*.

```

1 package ta1.cis.adapter
2
3 import ...
19
20 class PaketAdapter(val pakets: List<Paket>) : RecyclerView.Adapter<PaketAdapter.PaketViewHolder>() {
21
22     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PaketViewHolder {
23         return PaketViewHolder (
24             LayoutInflater.from(parent.context)
25             .inflate(R.layout.paket_layout, parent, attachToRoot: false)
26         )
27     }
28     override fun getItemCount() = pakets.size
29
30     override fun onBindViewHolder(holder: PaketViewHolder, position: Int) {
31         val paket = pakets[position]
32         holder.bind(paket)
33         holder.view.setOnClickListener { it: View!
34             Toast.makeText(it.context, paket.toString(), Toast.LENGTH_SHORT).show()
35             val intent = Intent(it.context, DetailPaketActivity::class.java).apply { this: Intent
36                 putExtra(DetailPaketActivity.DATA_PAKET, paket)
37             }
38             it.context.startActivity(intent)
39         }
40     }
41
42     inner class PaketViewHolder(val view: View) : RecyclerView.ViewHolder(view){
43         fun bind(paket: Paket){

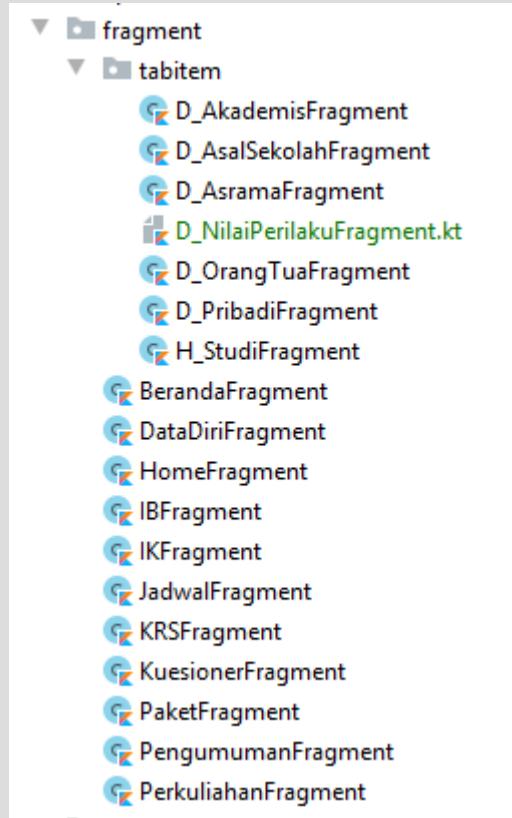
```

Gambar 120. Potongan Kode Adapter

## 7. Pembuatan Fragment dan *Layout*

Fragment merupakan bagian dari user *interface* yang ditempatkan pada sebuah *activity*. Pada pembangunan aplikasi terdapat 18 fragment yang akan di tampilkan di dalam satu *activity* yaitu D\_AkademisFragment, D\_AsalSekolahFragment, D\_AsramaFragment, D\_NilaiPerilakuFragment, D\_OrangTuaFragment,

D\_PribadiFragment, H\_StudiFragment, BerandaFragment, DataDiriFragment, HomeFragment, IBFragment, IKFragment, JadwalFragment, KRSFragment, KuesionerFragment, KuesionerFragment, PaketFragment, PengumumanFragment dan PerkuliahFragment.



**Gambar 121. Pembuatan Fragment**

Berikut merupakan potongan kode yang terdapat pada PaketFragment. Pada fragment berisi kode dalam pemanggilan Adapter, model dan fungsi yang akan ditampilkan. Pada kode juga akan memanggil layout yaitu paket2\_fragment.

```
1 package ta1.cis.fragment
2
3 import ...
4
5 class PaketFragment : Fragment() {
6
7     private lateinit var sharedPref: SharedPref
8
9     companion object {
10         fun newInstance() = PaketFragment()
11     }
12
13     private lateinit var viewModel: PaketViewModel
14
15     override fun onCreateView(
16         inflater: LayoutInflater, container: ViewGroup?,
17         savedInstanceState: Bundle?
18     ): View? {
19         sharedPref = SharedPref(activity as Context)
20         return inflater.inflate(R.layout.paket2_fragment, container, attachToRoot: false)
21     }
22
23     override fun onActivityCreated(savedInstanceState: Bundle?) {
24         super.onActivityCreated(savedInstanceState)
25         viewModel = ViewModelProvider(owner: this).get(PaketViewModel::class.java)
26
27         refreshlayout.setOnRefreshListener {
28             fetchPaket()
29         }
30     }
31 }
```

### **Gambar 122. Potongan Kode Fragment**

Berikut merupakan potongan kode pada *layout* yang dipanggil pada *fragment*. *RecyclerView* merupakan versi *ListView* yang fleksibel untuk menampilkan rangkaian data besar yang bias digulir sangat efisien dengan mempertahankan tampilan.

The screenshot shows the Android Studio XML layout editor with the file 'paket2\_fragment.xml' open. The code displays a fragment layout structure:

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:app="http://schemas.android.com/apk/res-auto"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     tools:context="fragment.PaketFragment">
7
8     <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
9         android:id="@+id/refreshlayout"
10        android:layout_width="match_parent"
11        android:layout_height="match_parent">
12
13         <androidx.recyclerview.widget.RecyclerView
14             tools:listitem="@layout/paket_layout"
15             android:id="@+id/recyclerViewPaket"
16             android:layout_width="match_parent"
17             android:layout_height="match_parent"/>
18
19     </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>
20 </RelativeLayout>
```

**Gambar 123. Potongan Kode Layout RecyclerView**

Berikut merupakan potongan kode layout\_paket yang merupakan *layout* dalam memndesain user interface yang akan ditampilkan saat aplikasi dijalankan.

The screenshot shows a code editor window for an Android XML layout file named 'paket\_layout.xml'. The code is written in XML and defines a layout structure. The XML code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_marginBottom="8dp">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="108dp"
        app:cardElevation="10dp"
        app:cardCornerRadius="5dp"
        android:layout_margin="12dp"
        android:id="@+id/card_paket">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:padding="8dp"
            android:background="@color/bgrecyclerview">

            <TextView
                android:id="@+id/TagPaket"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="16dp"

```

Gambar 124. Potongan Kode *Layout*

## Lampiran 5 - Hasil UAT

Pada Lampiran 5 ini menampilkan hasil UAT yang telah dilakukan.

### Project Information

Client Name : Jane Mitaria Sinambela	Project Name : Pengembangan Fitur-Fitur <i>Campus Information System (CIS)</i> Berbasis Mobile dengan Layanan API
Project Phase : User Acceptance Testing	Date : 10 Juli 2021

### Description

On July 10, 2021 at the Del Institute of Technology, has been conducted User Acceptance Testing for “Pengembangan Fitur-Fitur *Campus Information System (CIS)* Berbasis Mobile dengan Layanan API”

This activity has been carried out by D3TI13, as the party of Del Institut of Technology and witnesses by Jane Mitaria Sinambela as the student of Del Institute of Technology.

### Note (If Any)

Kembangkan dengan modul lebih komplex

### User Rate

80

**Gambar 125. Hasil UAT**

## ACKNOWLEDGE

Laguboti , 10 Juli 2021

Tim

Martinus Yudha Chrisanto Sitinjak

Jane Mitaria Sinambela

Rafika Tampubolon

Lucy Sihite

### 1. Short Description

Aplikasi yang dikembangkan adalah Sistem informasi kampus untuk Institut Teknologi Del. Sistem Informasi ini mencakup 6 modul yaitu modul autentikasi, modul data diri, modul zin bermalam, modul pengumuman, modul paket, modul notifikasi

## 2. Testing Scenario

No	Description	Expected Result	Criteria
TC-01	Login	User Masuk Ke Sitem	✓
TC-02	Melihat Tampilan Menu	Tampil Menu	✓
TC-03	Melihat Data Akademis	Tampil Data Akademis	✓
TC-04	Melihat Data Pribadi	Tampil Data Pribadi	✓
TC-05	Mengubah Data Pribadi	Data Pribadi dapat diubah	✓
TC-06	Melihat Data Orangtua	Tampil Data Orangtua	✓
TC-07	Mengubah Data Orangtua	Data Orangtua dapat diubah	✓
TC-08	Melihat Data Sekolah	Tampil Data Sekolah	✓
TC-09	Mengubah Data Sekolah	Data Sekolah dapat diubah	✓
TC-10	Melihat Data Keasramaan	Tampil Data Asrama	✓
TC-11	Melihat Izin Bermalam	Tampil Izin Bermalam	✓
TC-12	Menambahkan Izin Bermalam	Izin Bermalam bertambah	✓
TC-13	Melihat Notifikasi Ketika Izin Bermalam di update	Notifikasi dapat dilihat	✓
TC-14	Melihat Detail Izin Bermalam	Detail Izin Bermalam dapat dilihat	✓

TC-15	Melihat Pengumuman	Pengumuman Dapat Dilihat	✓
TC-16	Melihat Detail Pengumuman	Detail Pengumuman dapat dilihat	✓
TC-17	Mendownload File Pengumuman	File Pengumuman dapat didownload	✓
TC-18	Melihat data Paket	Data Paket dapat dilihat	✓
TC-19	Melihat Detail Paket	Detail Paket dapat dilihat	✓
TC-20	Melihat Izin Keluar	Izin Keluar dapat dilihat	✓
TC-21	Mengedit Izin Keluar	Izin Keluar dapat diedit	✓
TC-22	Membatalkan Izin Keluar	Izin Keluar dapat dibatalkan	✓
TC-23	Melihat Detail Izin Keluar	Detail Izin Keluar dapat dilihat	✓