



# LAPORAN AKHIR

Kelas D - Kelompok 02

Arya Dheffan Shevchenko (24060121140160)

Arynda Anna Salsabiela (24060121120023)

Kharisma Andini Putri (24060121140119)

Yesy Margharetta Munthe (24060121120031)



**Dokumentasi Pengembangan Perangkat Lunak  
Monitoring Progress Studi Mahasiswa Informatika**

**Mata Kuliah Proyek Perangkat Lunak (PPL) - 3 sks  
(Capstones Project)**

## Kata Pengantar

---

Segala puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya yang diberikan, sehingga “Laporan Akhir Dokumentasi Pengembangan Perangkat Lunak” ini dapat terselesaikan dengan baik. Laporan ini kami susun sebagai bagian dari penugasan mata kuliah Proyek Perangkat Lunak.

Adapun maksud dan tujuan adanya laporan akhir ini adalah dapat memahami dan membuat sebuah system informasi terkait kondisi departemen Informatika yang mengalami peningkatan jumlah mahasiswa dari tahun ke tahun. Dalam penyusunan laporan ini, saya mengucapkan terima kasih kepada Bapak Dr. Aris Puji Widodo, S.Si, M.T. yang telah membantu kami dalam pemahaman materi dan dasar – dasar permbuatan sistem informasi.

Dengan ini, kami menyadari bahwa laporan praktikum ini masih jauh dari kata sempurna. Untuk itu, saya dengan sangat terbuka menerima kritik dan saran dari para pembaca. Semoga laporan praktikum ini bermanfaat untuk para pembaca dan kami khususnya dalam usaha memahami dan membuat sebuah system informasi terkait kondisi departemen Informatika yang mengalami peningkatan jumlah mahasiswa dari tahun ke tahun.

Semarang, 19 Desember 2023

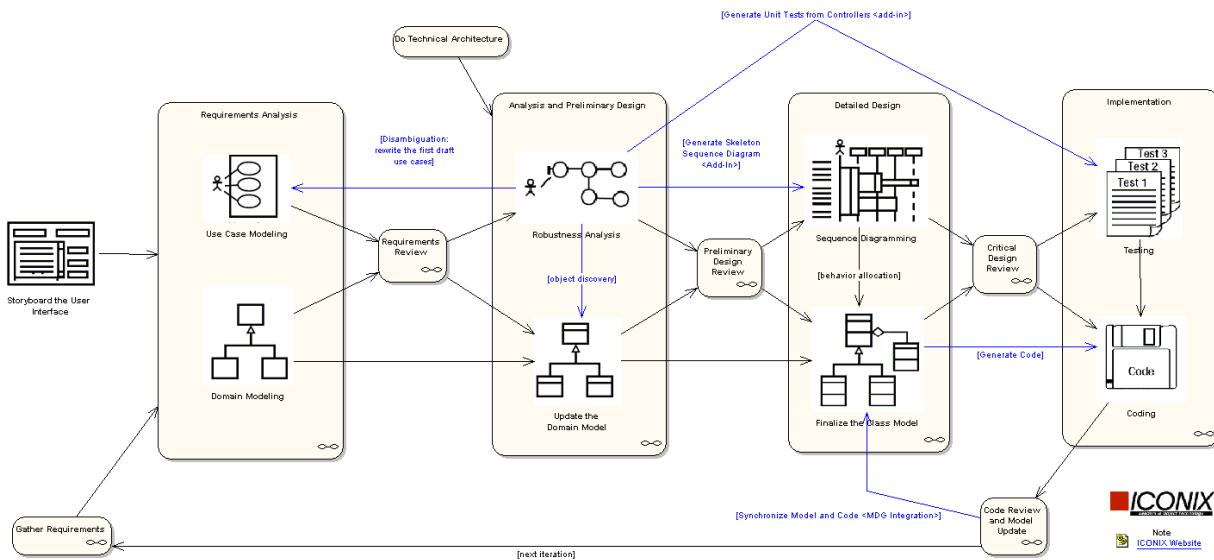
## Daftar Isi

<b>Kata Pengantar .....</b>	<b>2</b>
<b>1 ICONIX PROCESS .....</b>	<b>4</b>
<b>2 Requirement Gathering .....</b>	<b>5</b>
2.1 Deskripsi Singkat	5
2.2 Proses Bisnis	6
2.3 Software Requirement Specifications (SRS)	6
2.4 GUI Story Board	7
<b>3 Requirement Analysis .....</b>	<b>23</b>
3.1 Use Case Diagram	23
3.2 Domain Model	24
<b>4 Analysis and Preliminary Design.....</b>	<b>25</b>
4.1 Robustness Diagram	25
4.2 Update Domain Model	33
<b>5 Detailed Design .....</b>	<b>34</b>
5.1 Sequence Diagram	34
5.2 Class Diagram	39
<b>6 Implementation.....</b>	<b>40</b>
6.1 Code	40
6.2 Testing	60
<b>7 Keterlibatan Anggota Kelompok.....</b>	<b>62</b>
<b>8 Daftar Pustaka .....</b>	<b>64</b>
<b>Lampiran .....</b>	Error! Bookmark not defined.

# 1 ICONIX PROCESS

Menurut Yulianta & Mursanto (2008), ICONIX PROCESS Roadmap merupakan kumpulan diagram aktivitas yang menggambarkan semua langkah yang terlibat dalam proses ICONIX. Tujuan utama dari ICONIX proses adalah bagaimana mewujudkan use case menjadi kode di mana kode tersebut harus lengkap, telah dilakukan testing, dan dapat mengerjakan apa yang ada pada use case dan requirement yang ada. ICONIX proses akan berusaha menjawab tiap proses-proses yang dilakukan. ICONIX Process dikenalkan oleh Doug Rosenberg yang berusaha menggabungkan praktik terbaik (*best practices*) dari 3 metodologi yang sudah ada yaitu *Object Modelling Techniques* (OMT) oleh James Rumbaugh, *Object-Oriented Software Engineering* (OOSE) oleh Ivar Jacobson, dan *Structural Method* (Booch) oleh Grady Booch.

Pada ICONIX proses ini terdapat diagram yang menggambarkan keseluruhan proses sebagai berikut. Dalam ICONIX proses ini dimulai dengan storyboard yang menyesuaikan requirement kemudian dibuat use case dan domain model, selanjutnya dibuat robustness dan update domain model sebagai analisis desain. Setelah itu perincian desain dengan sequence diagram dan class diagram. Setelah dokumentasi selesai dilanjut ke proses implementasi dengan kode dan testing program (*Roadmap \_ ICONIX Process*, 2007; Yulianta & Mursanto, 2008).



Gambar 1.1. ICONIX PROCESS

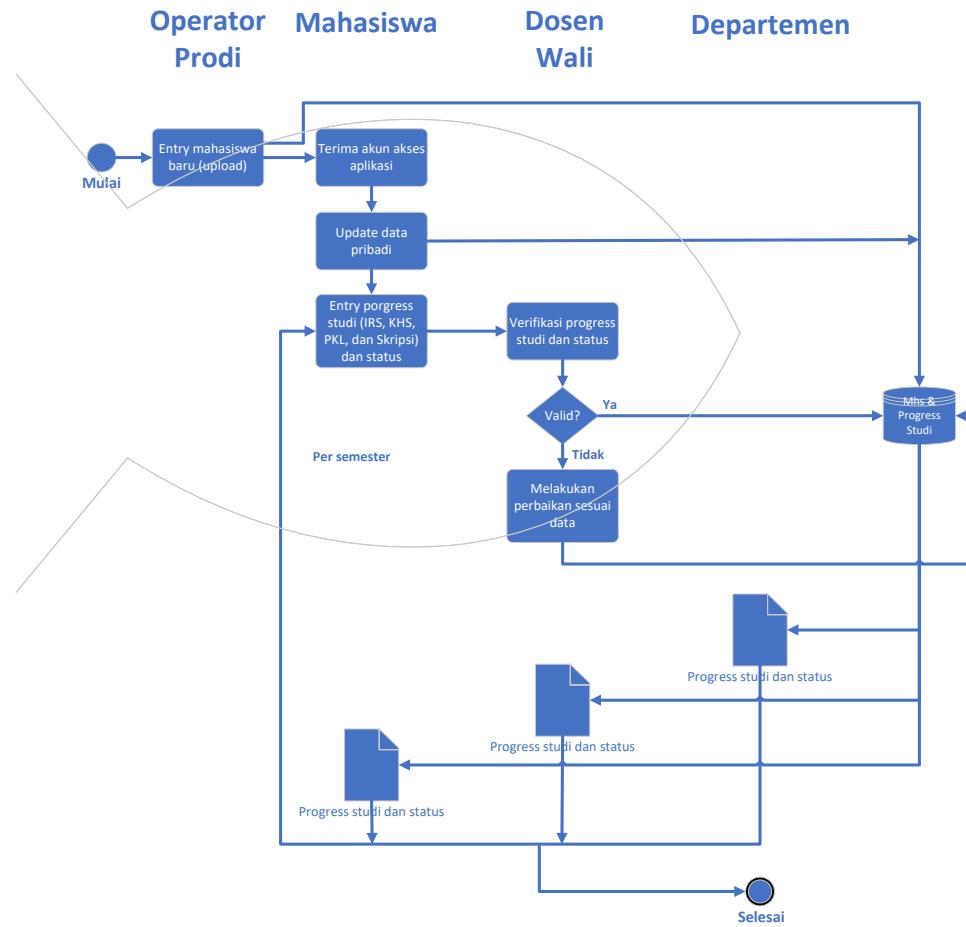
## 2 Requirement Gathering

---

### 2.1 Deskripsi Singkat

Dari tahun ke tahun jumlah mahasiswa departemen Informatika semakin bertambah, karena jumlah mahasiswa yang diterima sebagai mahasiswa baru tidak berimbang dengan jumlah kelulusan pada setiap tahunnya. Berdasarkan jumlah mahasiswa dan adanya kebutuhan untuk melakukan monitoring dan evaluasi progress perkuliahan untuk setiap mahasiswa, maka dibutuhkan sebuah perangkat lunak untuk mempermudah dan mempercepat dosen wali dan departemen dalam mendapatkan informasi untuk melakukan pengambilan keputusan. Terdapat permasalahan yaitu untuk mengetahui progress perkembangan akademik mahasiswa departemen Informatika kurang efektif dan efisien karena harus melihat satu persatu setiap dokumen IRS dan KHS mahasiswa sehingga untuk melakukan pengambilan keputusan kurang dapat di dukung berdasarkan data sehingga ada kebutuhan informasi yang diperlukan yaitu menyajikan prestasi akademik mahasiswa departemen Informatika untuk setiap semester secara rinci dan summary agar dapat digunakan untuk pengambilan keputusan. Untuk pengguna yang diperlukan yaitu mahasiswa, dosen wali, operator departemen, dan departemen dengan kebutuhan data untuk mahasiswa. Terdiri dari elemen data nim, nama, alamat, kab/kota, provinsi, angkatan, jalur masuk, email, handphone, status, dan foto mahasiswa. Elemen data status terdiri dari aktif, cuti, mangkir, DO, undur diri, lulus, dan meninggal dunia. Elemen data jalur masuk terdiri dari SNMPTN, SBMPTN, mandiri, lainnya. Elemen data kab/kota dan provinsi menggunakan data yang diberikan di Dukcapil (termasuk kode). Untuk IRS terdiri dari semester aktif, jumlah SKS diambil, dan scan IRS. Untuk KHS terdiri dari semester aktif, jumlah SKS semester, jumlah SKS kumulatif, IP semester, IP kumulatif, dan scan KHS. Untuk PKL terdapat elemen data status PKL, nilai PKL, dan scan berita acara seminar PKL. Elemen data status terdiri dari belum ambil, sedang ambil, dan lulus serta elemen nilai PKL diisikan jika statusnya lulus. Untuk skripsi terdiri dari elemen data status skripsi, nilai skripsi, dan scan berita acara siding skripsi. Elemen data status terdiri dari belum ambil, sedang ambil, dan lulus serta elemen nilai skripsi diisikan jika statusnya lulus.

## 2.2 Proses Bisnis



## 2.3 Software Requirement Specifications (SRS)

1. Entry (upload) data mahasiswa baru menggunakan template yang sudah disediakan (diantaranya terdiri: Nim, nama, angkatan, dan status=AKTIF). Pada saat upload sekaligus dilakukan generate akun akses untuk setiap mahasiswa secara default. (SRS-XXX-001)
2. Update data pribadi mahasiswa secara mandiri, untuk elemen data nim dan angkatan tidak dapat dilakukan update. Pada saat pertama kali login harus melakukan pengisian semua elemen data. Jika belum dilakukan pengisian semua elemen data mahasiswa maka belum dapat menggunakan fitur lain (SRS-XXX-002)
3. Entry pengambilan IRS per semester (SRS-XXX-003)

4. Entry prestasi akademik KHS per semester (SRS-XXX-004)
5. Entry progress PKL (SRS-XXX-005)
6. Entry progress skripsi (SRS-XXX-006) ---- tambahkan elemen data tanggal lulus/sidang, dan lama studi dalam semester
7. Verifikasi progress studi meliputi IRS, KHS, PKL, dan Skripsi untuk diberikan approval. Jika sudah sesuai dapat langsung di approve, dan jika belum sesuai harus dilakukan update sesuai data dukungnya. (SRS-XXX-007)
8. Pencarian progress studi mahasiswa untuk setiap semester berdasarkan nim dan atau nama (SRS-XXX-008)
9. Profile dan dashboard operator prodi (SRS-XXX-009)
10. Profile dan dashboard mahasiswa (SRS-XXX-010)
11. Profile dan dashboard dosen wali (SRS-XXX-011)
12. Profile dan dashboard departemen (SRS-XXX-012)
13. Rekap mahasiswa yang sudah lulus dan belum PKL untuk semua angkatan (SRS-XXX-013)
14. List daftar mahasiswa yang sudah lulus dan belum PKL untuk semua Angkatan (SRS-XXX-014)
15. Rekap mahasiswa yang sudah lulus dan belum Skripsi untuk semua Angkatan (SRS-XXX-015)
16. List daftar mahasiswa yang sudah lulus dan belum Skripsi untuk semua Angkatan (SRS-XXX-016)
17. Rekap mahasiswa berdasarkan status yang disajikan per angkatan dan dilengkapi list daftar mahasiswanya (SRS-XXX-017)

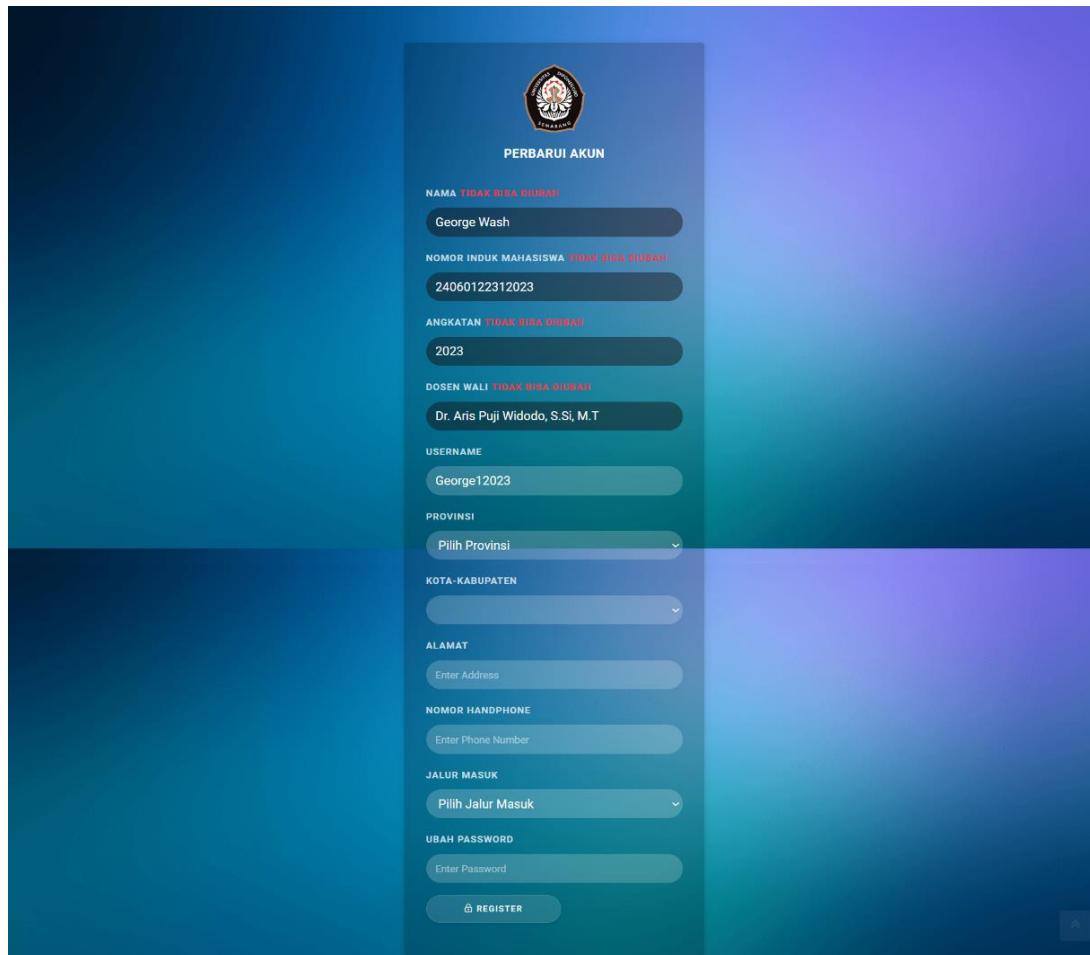
## 2.4 GUI Story Board

1. Entry (upload) data mahasiswa baru menggunakan template yang sudah disediakan (diantaranya terdiri: Nim, nama, angkatan, dan status=AKTIF). Pada saat upload sekaligus dilakukan generate akun akses untuk setiap mahasiswa secara default. (SRS-XXX-001)

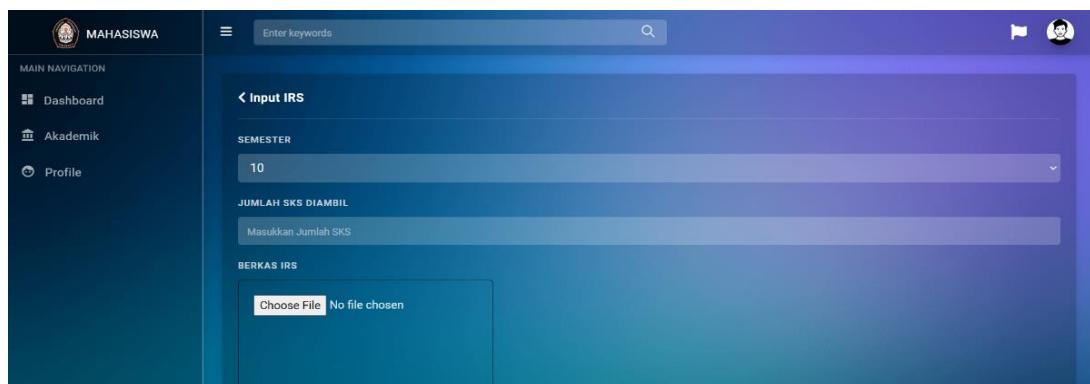
The application interface consists of two main sections. The top section is a registration form titled 'REGISTER' with fields for NAMA, NIM, ANGKATAN, STATUS, DOSEN WALI, and a 'REGISTER' button. The bottom section is a data management page titled 'Data Mahasiswa' with a navigation bar, a search bar, and a table listing student data. The table has columns: NO., NAMA LENGKAP, NIM, ANGKATAN, DOSEN WALI, and AKSI. The table contains 6 rows of student information.

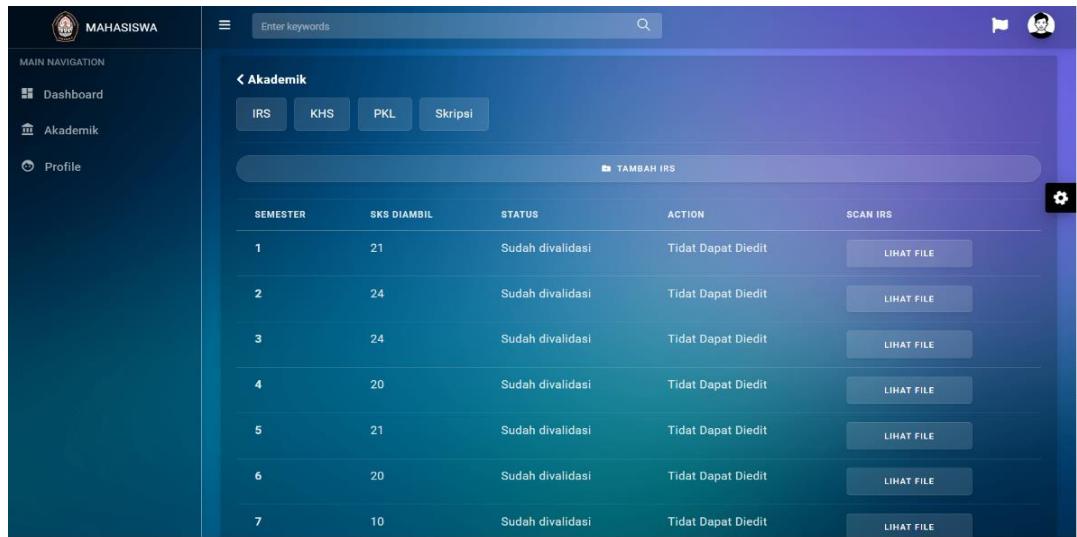
NO.	NAMA LENGKAP	NIM	ANGKATAN	DOSEN WALI	AKSI
1	Ainun Putriana	24060117120091	2017	Dr. Aris Puji Widodo, S.Si, M.T	
2	Ariana	24060117120092	2017	Dr. Aris Puji Widodo, S.Si, M.T	
3	Andri	24060117120093	2017	Dr. Aris Puji Widodo, S.Si, M.T	
4	Andrian	24060117120094	2017	Yunila Dwi Putri Ariyanti, S.Kom., M.T, M.T	
5	Ryan	24060117120095	2017	Yunila Dwi Putri Ariyanti, S.Kom., M.T, M.T	
6	Jamalukdin Siara	24060118120071	2018	Yunila Dwi Putri Ariyanti, S.Kom., M.T, M.T	

2. Update data pribadi mahasiswa secara mandiri, untuk elemen data nim dan angkatan tidak dapat dilakukan update. Pada saat pertama kali login harus melakukan pengisian semua elemen data. Jika belum dilakukan pengisian semua elemen data mahasiswa maka belum dapat menggunakan fitur lain (SRS-XXX-002)



### 3. Entry pengambilan IRS per semester (SRS-XXX-003)

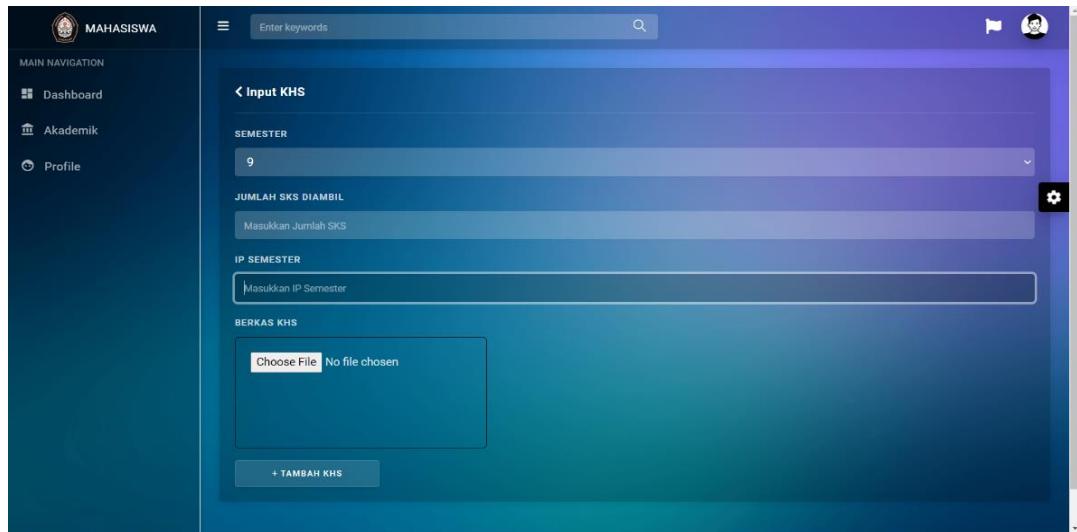




The screenshot shows the 'Akademik' section of the application. At the top, there are tabs for IRS, KHS, PKL, and Skripsi. Below this is a table with columns: SEMESTER, SKS DIAMBIL, STATUS, ACTION, and SCAN IRS. Each row contains a 'LIHAT FILE' button. The data in the table is as follows:

SEMESTER	SKS DIAMBIL	STATUS	ACTION	SCAN IRS
1	21	Sudah divalidasi	Tidak Dapat Dieedit	<button>LIHAT FILE</button>
2	24	Sudah divalidasi	Tidak Dapat Dieedit	<button>LIHAT FILE</button>
3	24	Sudah divalidasi	Tidak Dapat Dieedit	<button>LIHAT FILE</button>
4	20	Sudah divalidasi	Tidak Dapat Dieedit	<button>LIHAT FILE</button>
5	21	Sudah divalidasi	Tidak Dapat Dieedit	<button>LIHAT FILE</button>
6	20	Sudah divalidasi	Tidak Dapat Dieedit	<button>LIHAT FILE</button>
7	10	Sudah divalidasi	Tidak Dapat Dieedit	<button>LIHAT FILE</button>

#### 4. Entry prestasi akademik KHS per semester (SRS-XXX-004)



The screenshot shows the 'Input KHS' form. It includes fields for SEMESTER (set to 9), JUMLAH SKS DIAMBIL (with placeholder 'Masukkan Jumlah SKS'), IP SEMESTER (with placeholder 'Masukkan IP Semester'), and BERKAS KHS (a file upload field showing 'Choose File No file chosen'). At the bottom is a '+ TAMBAH KHS' button.

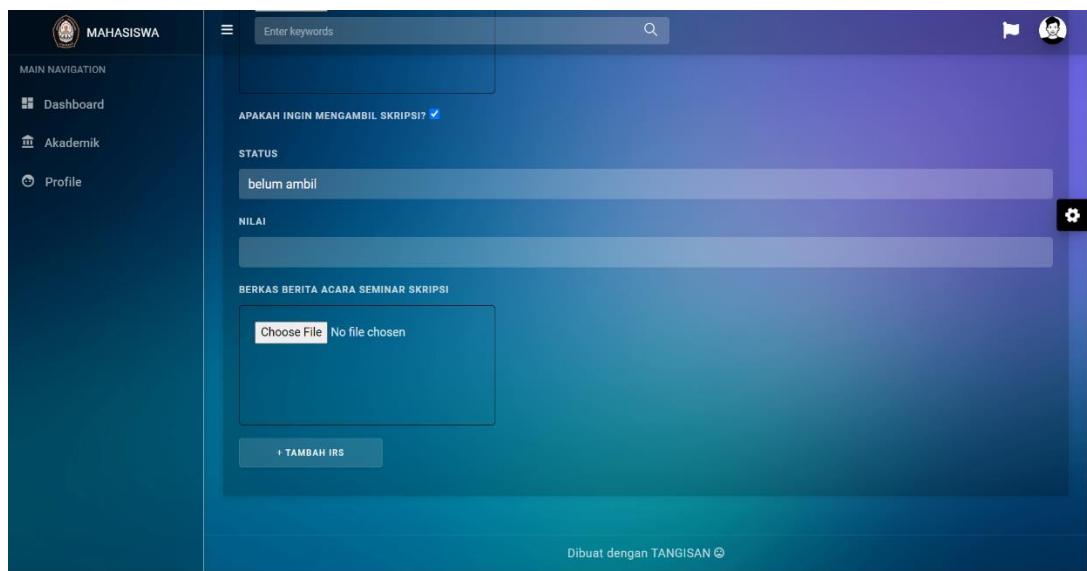
The screenshot shows the 'Akademik' section of the application. At the top, there are tabs for IRS, KHS, PKL, and Skripsi. Below them is a table with the following data:

SEMESTER	SKS SEMESTER	SKS KUMULATIF	IP SEMESTER	IP KUMULATIF	STATUS	SCAN KHS
1	21	21	4.00	4.00	Sudah divalidasi	<button>LIHAT FILE</button>
2	24	45	3.70	3.85	Sudah divalidasi	<button>LIHAT FILE</button>
3	24	69	3.25	3.55	Sudah divalidasi	<button>LIHAT FILE</button>
4	20	89	3.25	3.40	Sudah divalidasi	<button>LIHAT FILE</button>
5	21	110	3.50	3.45	Sudah divalidasi	<button>LIHAT FILE</button>
6	20	130	2.75	3.10	Sudah divalidasi	<button>LIHAT FILE</button>
7	10	140	4.00	3.55	Sudah divalidasi	<button>LIHAT FILE</button>

## 5. Entry progress PKL (SRS-XXX-005)

The screenshot shows the 'PKL' entry section. It includes fields for STATUS (set to 'belum ambil') and NILAI (empty). Below these is a file upload area for 'BERKAS BERITA ACARA SEMINAR PKL' with a placeholder message 'Choose File No file chosen'. At the bottom, there is a button '+ TAMBAH IRS'.

## 6. Entry progress skripsi(SRS-XXX-006)



7. Verifikasi progress studi meliputi IRS, KHS, PKL, dan Skripsi untuk diberikan approval. Jika sudah sesuai dapat langsung di approve, dan jika belum sesuai harus dilakukan update sesuai data dukungnya.(SRS-XXX-007)

The screenshot shows the 'DOSEN' (Professor) application interface. The left sidebar has a dark blue background with the title 'DOSEN' and navigation links: 'Dashboard', 'Progress', 'Validasi', 'Profile'. The main content area has a purple gradient background. At the top, there's a search bar with placeholder text 'Enter keywords' and a magnifying glass icon. Below it, a section titled '< Validasi Mahasiswa' with tabs 'IRS', 'KHS', 'PKL', and 'Skripsi'. A 'Filter Status' dropdown menu is shown. A table lists student progress details:

NO.	ID	NAMA	NIM	SEMESTER	STATUS	SCAN IRS
1	IRS17009101	Ainun Putriana	24060117120091	1	Sudah divalidasi	<button>LIHAT FILE</button>
2	IRS17009102	Ainun Putriana	24060117120091	2	Sudah divalidasi	<button>LIHAT FILE</button>
3	IRS17009103	Ainun Putriana	24060117120091	3	Sudah divalidasi	<button>LIHAT FILE</button>
4	IRS17009104	Ainun Putriana	24060117120091	4	Sudah divalidasi	<button>LIHAT FILE</button>
5	IRS17009105	Ainun Putriana	24060117120091	5	Sudah divalidasi	<button>LIHAT FILE</button>
6	IRS17009106	Ainun Putriana	24060117120091	6	Sudah divalidasi	<button>LIHAT FILE</button>
7	IRS17009107	Ainun Putriana	24060117120091	7	Sudah divalidasi	<button>LIHAT FILE</button>
8	IRS17009108	Ainun Putriana	24060117120091	8	Sudah divalidasi	<button>LIHAT FILE</button>

The screenshot shows a web application interface for a professor ('DOSEN'). The left sidebar contains a main navigation menu with options: Dashboard, Progress, Validasi (selected), and Profile. The main content area is titled 'Validasi Mahasiswa' and displays a table of student records. The table columns include NO., ID, NAMA, NIM, SEMESTER, IP SEMESTER, STATUS, and SCAN KI. Each row has a 'LIHAT' button. Below the table is a dropdown menu labeled '- Filter Status -'. A modal window titled 'Berkas IRS' is open, showing a preview of a document titled 'PROPOSAL PRAKTIK KERJA LAPANGAN DI PT PERTAMINA HULU ROKAN(PHR) KOTA PEKANBARU'. The document includes the university logo and the signature 'Arya Dheftan Shevchenko'. At the bottom of the modal are 'CLOSE' and 'VALIDASI' buttons.

NO.	ID	NAMA	NIM	SEMESTER	IP SEMESTER	STATUS	SCAN KI
1	KHS17009101	Ainun Putriana	24060117120091	1	3.40	Sudah divalidasi	LIHAT
2	KHS17009102	Ainun Putriana	24060117120091	2	3.70	Sudah divalidasi	LIHAT
3	KHS17009103	Ainun Putriana	24060117120091	3	3.25	Sudah divalidasi	LIHAT
4	KHS17009104	Ainun Putriana	24060117120091	4	2.75	Sudah divalidasi	LIHAT
5	KHS17009105	Ainun Putriana	24060117120091	5	3.75	Sudah divalidasi	LIHAT
6	KHS17009106	Ainun Putriana	24060117120091	6	3.30	Sudah divalidasi	LIHAT
7	KHS17009107	Ainun Putriana	24060117120091	7	3.00	Sudah divalidasi	LIHAT

127.0.0.1:8000/dashboard/dosen/197404011999031002/validation#section2

This screenshot shows the 'Validasi' section of the application. The left sidebar has the 'Validasi' option selected. The main content area lists IRS document IDs: IRS19006105, IRS19006106, IRS19006107, IRS19006108, IRS19006109, IRS21004101, IRS21004102, IRS21004103, IRS21004104, IRS21004105, and IRS23013801. A modal window titled 'Berkas IRS' is open, displaying a preview of the document with ID IRS19006106. The preview shows the title 'PROPOSAL PRAKTIK KERJA LAPANGAN DI PT PERTAMINA HULU ROKAN(PHR) KOTA PEKANBARU' and the university logo. At the bottom of the modal are 'CLOSE' and 'VALIDASI' buttons.

IRS19006105  
Enter Keywords

Kharisma Andini 24060117120091

Berkas IRS

1 / 16 | - + ⌂

PROPOSAL PRAKTIK KERJA LAPANGAN  
DI PT PERTAMINA HULU ROKAN(PHR)  
KOTA PEKANBARU

UNIVERSITAS NEGERI PADJADJARAN  
GEMBIRAH

Dibuat dengan TANGISAN ☺

CLOSE VALIDASI

8. Pencarian progress studi mahasiswa untuk setiap semester berdasarkan nim dan atau nama (SRS-XXX-008)

The screenshot shows a search interface for student progress. On the left is a sidebar with a logo and navigation links: Dashboard, Progress, Validasi, and Profile. The main area has a header with a search bar and a back button. Below it is another search bar. A table lists student names and NIMs, each with a 'VIEW' button. The footer says 'Dibuat dengan TANGISAN ☺'.

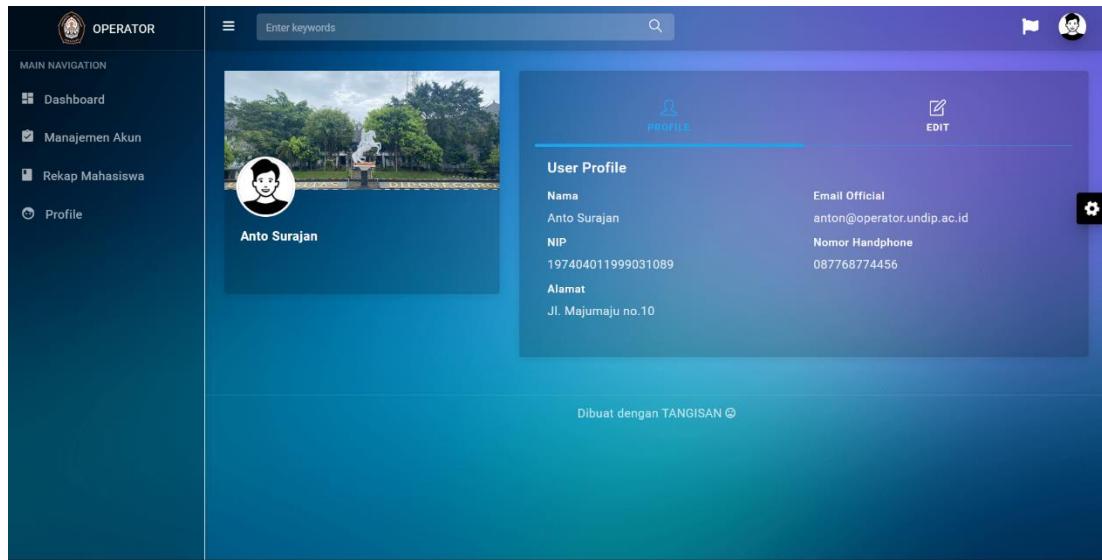
NAMA	NIM	VIEW
Ainun Putriana	24060117120091	VIEW
Kharisma Andini	24060119120061	VIEW
Arya Shevanco	24060121120041	VIEW
Iqbal Syaputra	24060123120021	VIEW
John Doe	24060123140138	VIEW

The screenshot shows a detailed student information page. The sidebar and header are identical to the previous screen. The main content displays student details: Name (Ainun Putriana), NIM (24060117120091), Angkatan (2017), and Wali (Dr. Aris Puji Widodo, S.Si, M.T). Below this is a grid of buttons labeled SEMESTER 1 through SEMESTER 13, with SEMESTER 6 highlighted in green and SEMESTER 12 highlighted in yellow. The footer says 'Dibuat dengan TANGISAN ☺'.

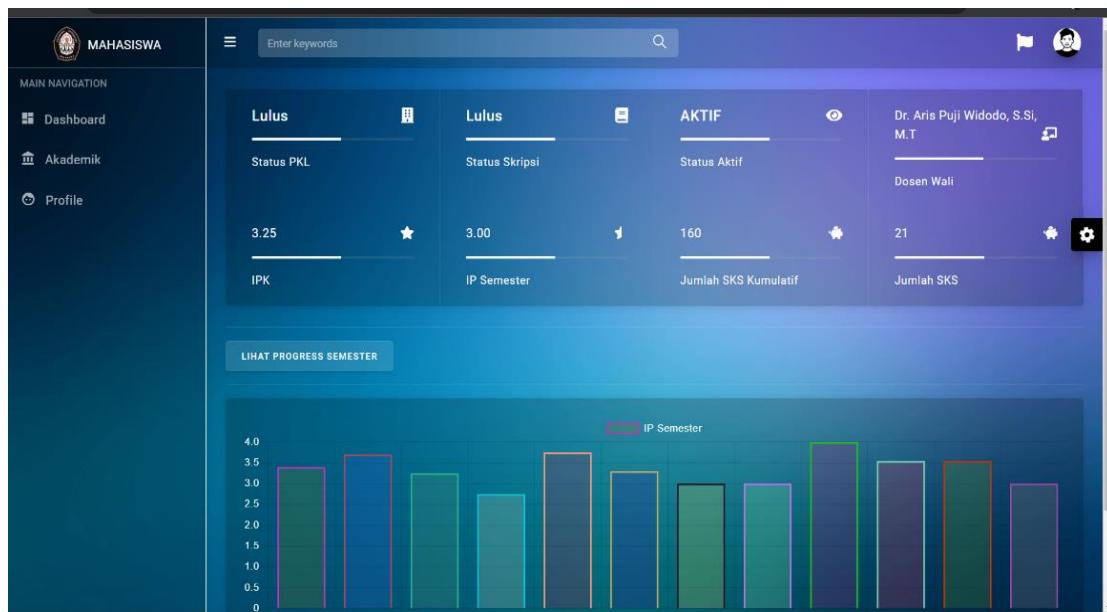
The screenshot shows the DOSEN (Faculty) dashboard. On the left sidebar, under 'MAIN NAVIGATION', the 'Progress' option is selected. The main content area displays a modal window titled 'Detail Mahasiswa' for a student named Ainun Putriana with NIM 24060117120091. The modal shows the student's grade distribution across semesters: IRS (21 SKS), KHS (SEMESTER 1-10), and PKL (SEMESTER 4-13). A large green bar at the bottom indicates the total number of credits: IRS17009106. The background shows other navigation links like 'Dashboard', 'Validasi', and 'Profile'.

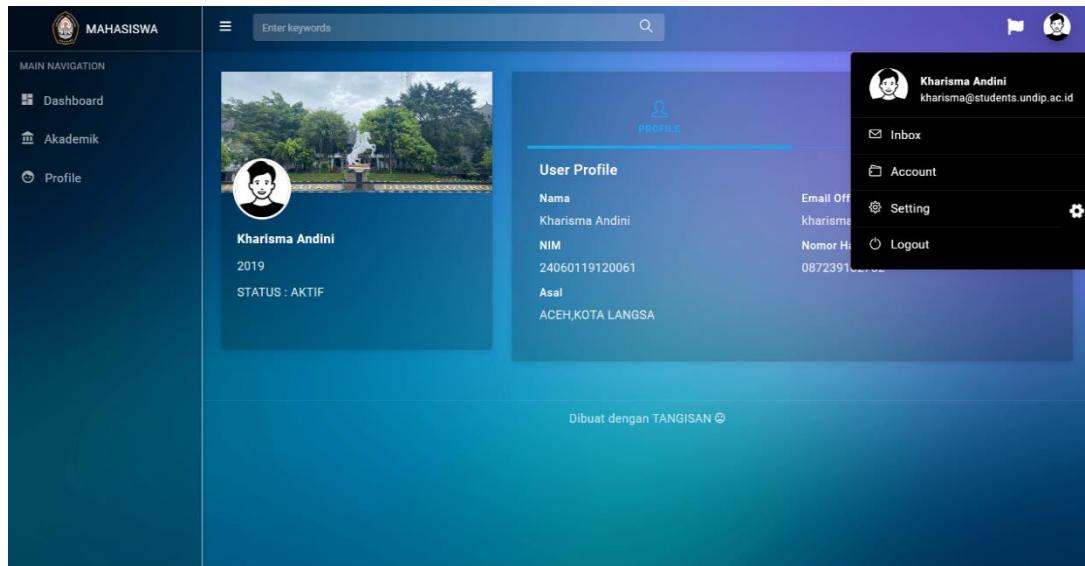
## 9. Profile dan dashboard operator prodi (SRS-XXX-009)

The screenshot shows the OPERATOR (Operator) dashboard. On the left sidebar, under 'MAIN NAVIGATION', the 'Manajemen Akun' option is selected. The main content area displays several statistics: Total Mahasiswa (29), Total Dosen (2), Total Mahasiswa Aktif (23), and Total Mahasiswa Tidak Aktif (6). Below these, a bar chart titled 'IP Semester Rata-rata' shows average grades for each semester from Semester 1 to Semester 14. The chart uses a color gradient from purple to green. A legend indicates the color for 'IP Semester Rata-rata'. The background shows other navigation links like 'Dashboard', 'Rekap Mahasiswa', and 'Profile'.

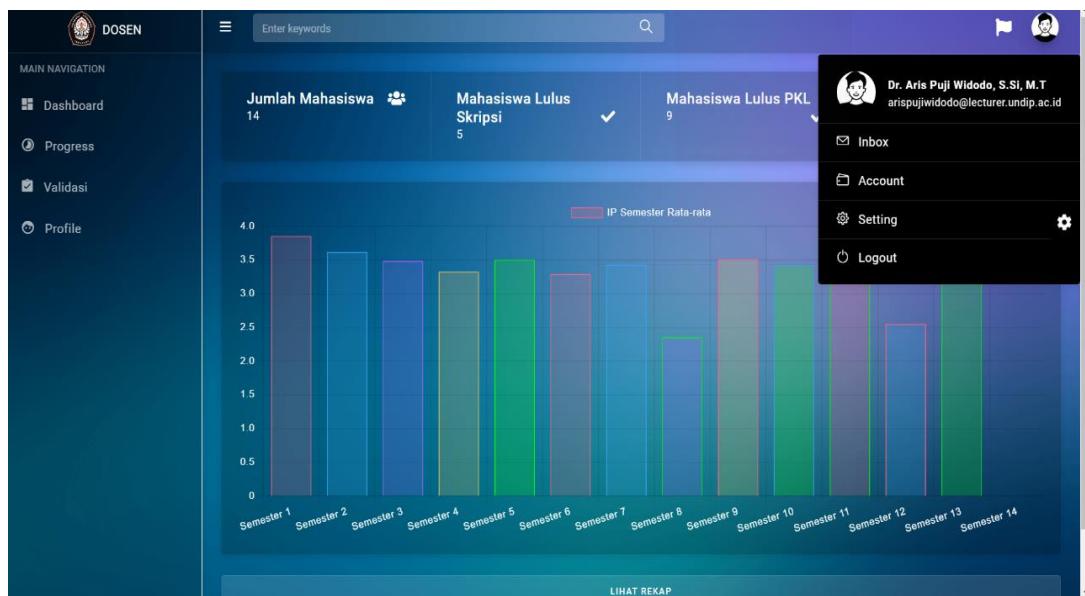


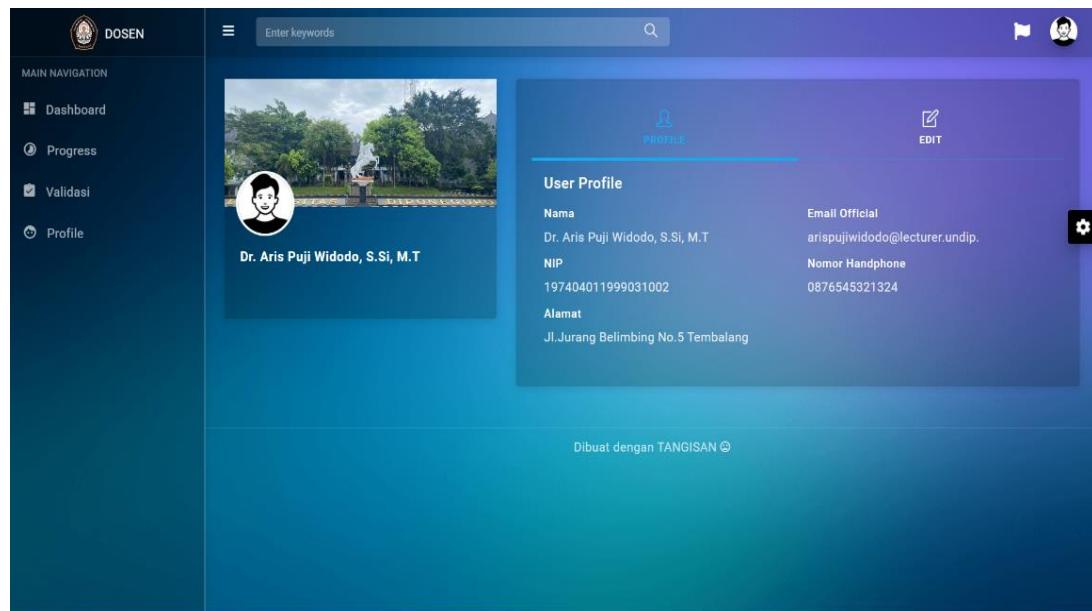
## 10. Profile dan dashboard mahasiswa (SRS-XXX-010)



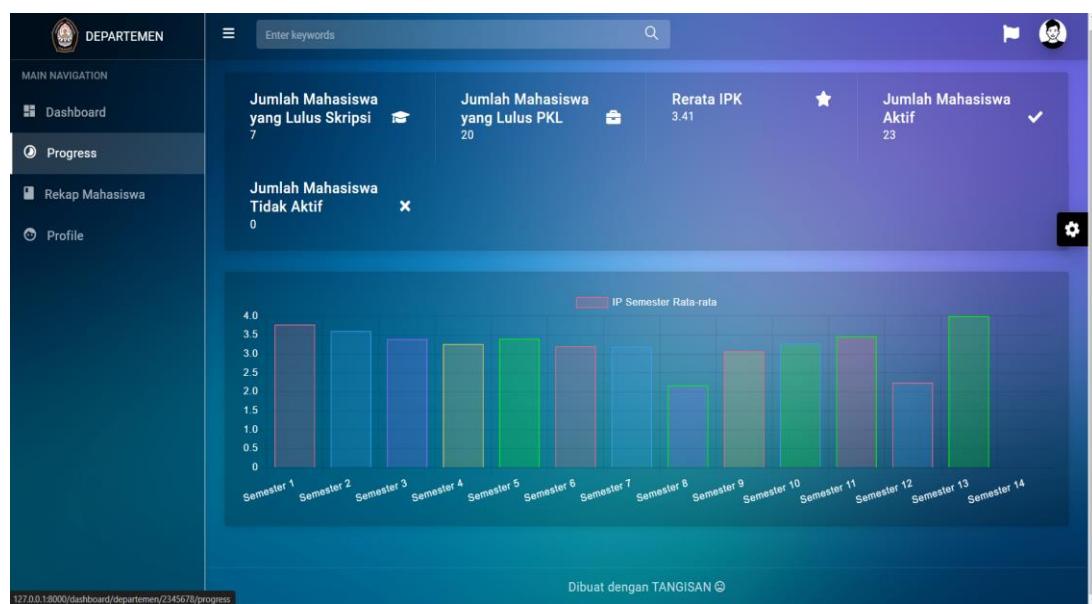


### 11. Profile dan dashboard dosen wali (SRS-XXX-011)





## 12. Profile dan dashboard departemen (SRS-XXX-012)



The screenshot shows a dark-themed dashboard for 'DEPARTEMEN'. On the left, a sidebar lists 'MAIN NAVIGATION' items: Dashboard, Progress, Rekap Mahasiswa, and Profile. The main area features a search bar with 'Enter keywords' and a magnifying glass icon. Below it is a profile section for 'IF' (Institut Teknologi Sepuluh Nopember), which includes a photo of a building, a user icon, and the text: 'ID Departemen : 2345678', 'Thn Berdiri : 2003', and 'Akreditasi: A'. To the right is a 'Sejarah Singkat' (Brief History) box containing two paragraphs about the department's history and changes in study programs. At the bottom right of the dashboard is a 'PRINT' button.

**13. Rekap mahasiswa yang sudah lulus dan belum PKL untuk semua angkatan (SRS-XXX-013)**

This screenshot shows a report titled 'REKAP PKL MAHASISWA' from the 'Rekap Mahasiswa' section. It displays a table with data for each year from 2017 to 2022, categorized by 'ANGKATAN' (Year). The columns represent 'SUDAH' (Completed) and 'BELUM' (Not Yet Completed) counts for PKL, Skripsi, and Status. The table shows the following data:

ANGKATAN	2017		2018		2019		2020		2021		2022	
	SUDAH	BELUM										
	1	0	1	0	1	0	0	1	0	1	0	1

At the bottom right of the report area is a 'PRINT' button.

14. List daftar mahasiswa yang sudah lulus dan belum PKL untuk semua Angkatan (SRS-XXX-014)

The screenshot shows a software interface titled "DEPARTEMEN". The left sidebar contains a "MAIN NAVIGATION" with options: Dashboard, Progress, Rekap Mahasiswa (selected), and Profile. The main content area has a search bar at the top. Below it, a table titled "List Mahasiswa lulus pkl Angkatan 2017" displays student information:

NO	NIM	NAMA	ANGKATAN	NILAI
1	24060117120091	Ainun Putriana	2017	A
2	24060117120092	Ariana	2017	A
3	24060117120093	Andri	2017	B
4	24060117120094	Andrian	2017	A
5	24060117120095	Ryan	2017	A

At the bottom right of the content area is a small gear icon.

15. Rekap mahasiswa yang sudah lulus dan belum Skripsi untuk semua Angkatan (SRS-XXX-015)

The screenshot shows a software interface titled "DEPARTEMEN". The left sidebar contains a "MAIN NAVIGATION" with options: Dashboard, Progress, Rekap Mahasiswa (selected), and Profile. The main content area has a search bar at the top. Below it, a section titled "Rekap Mahasiswa" shows tabs for "PKL", "Skripsi" (selected), and "Status". A table titled "REKAP SKRIPSI MAHASISWA" displays data for various years:

ANGKATAN	REKAP SKRIPSI MAHASISWA											
	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028
MH	BELUM	SUDAH	BELUM	SUDAH	BELUM	SUDAH	BELUM	SUDAH	BELUM	SUDAH	BELUM	
	0	0	1	0	1	0	1	0	1	0	1	

Below the table is a horizontal scroll bar. At the bottom right of the content area is a "PRINT" button.

16. List daftar mahasiswa yang sudah lulus dan belum Skripsi untuk semua Angkatan  
(SRS-XXX-016)

The screenshot shows a software interface titled "DEPARTEMEN". The main navigation menu on the left includes "Dashboard", "Progress", "Rekap Mahasiswa", and "Profile". The current page is titled "List Mahasiswa lulus skripsi Angkatan 2017". It displays a table with columns: NO, NIM, NAMA, ANGKATAN, and NILAI. Two rows of data are shown:

NO	NIM	NAMA	ANGKATAN	NILAI
1	24060117120091	Ainun Putriana	2017	B
2	24060117120092	Ariana	2017	A

At the bottom right of the table area is a small gear icon.

17. Rekap mahasiswa berdasarkan status yang disajikan per angkatan dan dilengkapi list daftar mahasiswanya (SRS-XXX-017)

The screenshot shows a software interface titled "DEPARTEMEN". The main navigation menu on the left includes "Dashboard", "Rekap Mahasiswa" (which is checked), and "Profile". The current page is titled "Rekap Mahasiswa". It features three tabs: "PKL", "Skripsi", and "Status". Below the tabs is a section titled "REKAP STATUS MAHASISWA" with a table showing student counts for each year from 2017 to 2021. The table has columns for "IF", "TIDAK AKTIF", "AKTIF", and "TIDAK AKTIF" for each year. A horizontal scroll bar is present below the table. At the bottom right of the table area is a "PRINT" button.

IF	2017		2018		2019		2020		2021	
	TIDAK AKTIF	AKTIF								
	0	1	0	1	0	1	0	1	0	1

At the bottom right of the table area is a small gear icon.

The screenshot shows a web-based application interface. On the left, there is a dark sidebar titled "DEPARTEMEN" with a logo at the top. Below the logo, the sidebar contains a "MAIN NAVIGATION" section with the following items: Dashboard, Progress, Rekap Mahasiswa, and Profile. The main content area has a blue header bar with a search bar containing "Enter keywords" and a magnifying glass icon. To the right of the search bar are two icons: a flag and a user profile. The main content area displays a table titled "List Mahasiswa aktif Angkatan 2017". The table has columns: NO AKTIF, NIM, NAMA, ANGKATAN, ASAL, and ALAMAT. The data in the table is as follows:

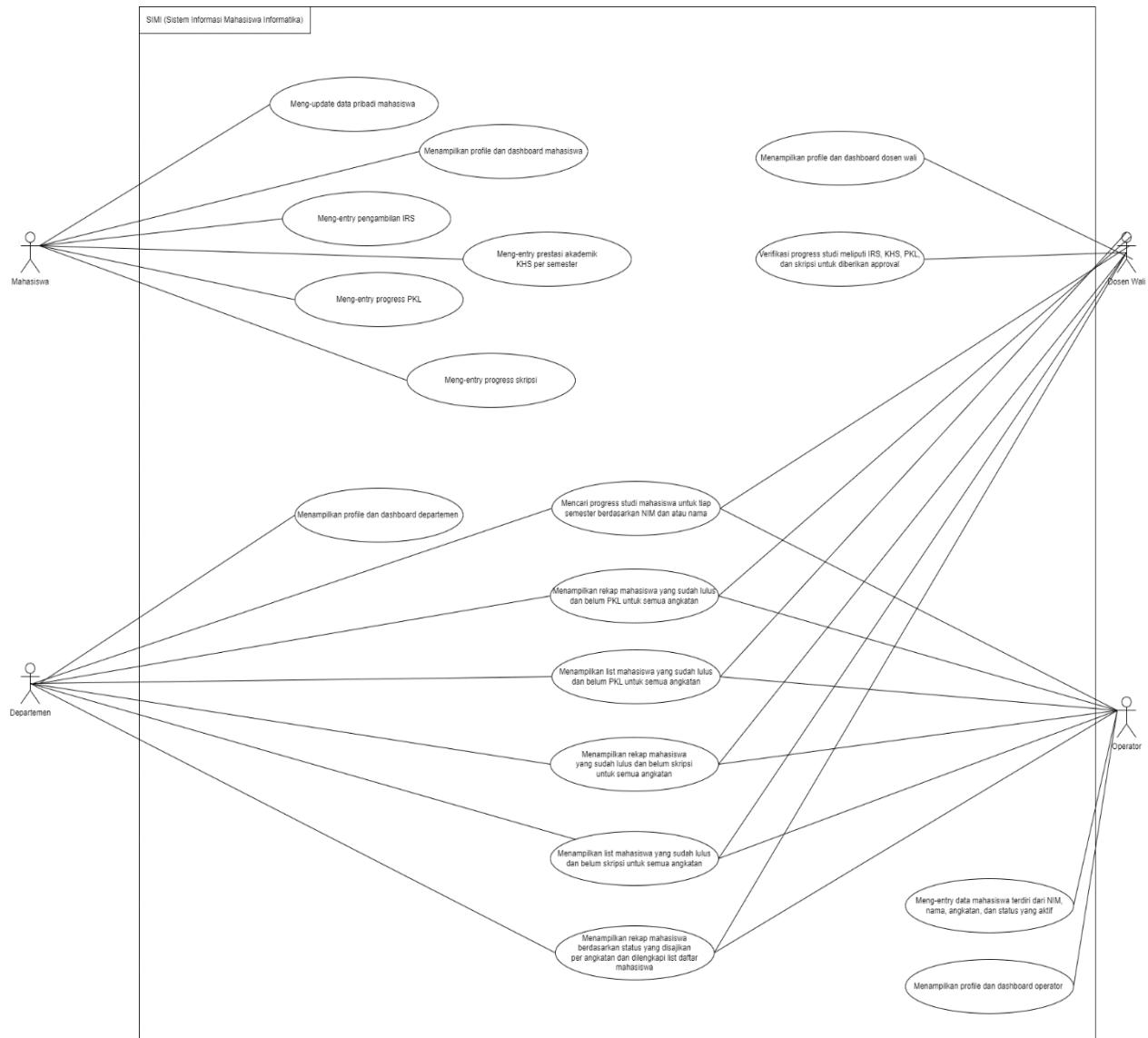
NO AKTIF	NIM	NAMA	ANGKATAN	ASAL	ALAMAT
1	24060117120091	Ainun Putriana	2017	ACEH, KABUPATEN ACEH SINGKIL	Jl.Menter
2	24060117120092	Ariana	2017	SUMATERA BARAT, KABUPATEN LIMA PULUH KOTA	Tanjung P
3	24060117120095	Ryan	2017	SUMATERA UTARA, KABUPATEN TAPANULI UTARA	Doksa

At the bottom of the main content area, there is a small text that reads "Dibuat dengan TANGISAN ☺".

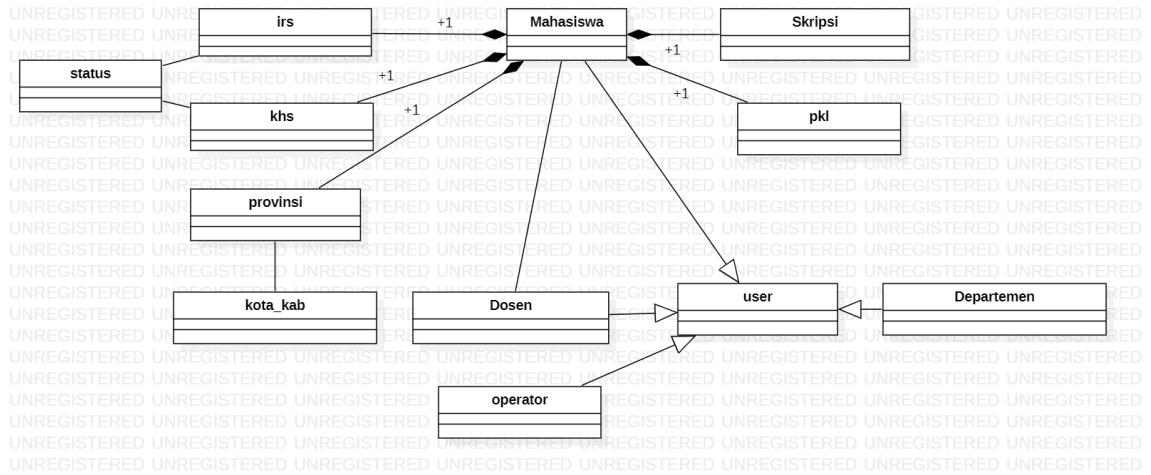
## 3 Requirement Analysis

---

### 3.1 Use Case Diagram



## 3.2 Domain Model

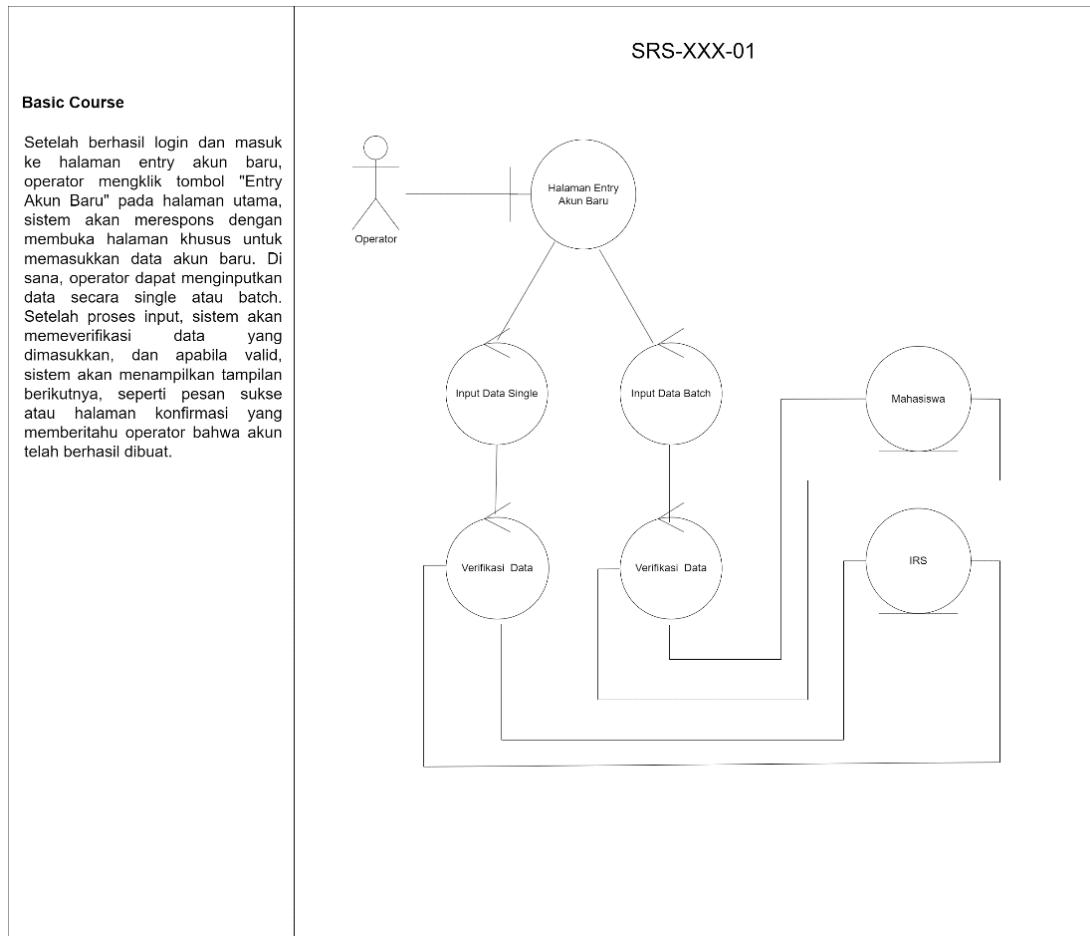


## 4 Analysis and Preliminary Design

---

### 4.1 Robustness Diagram

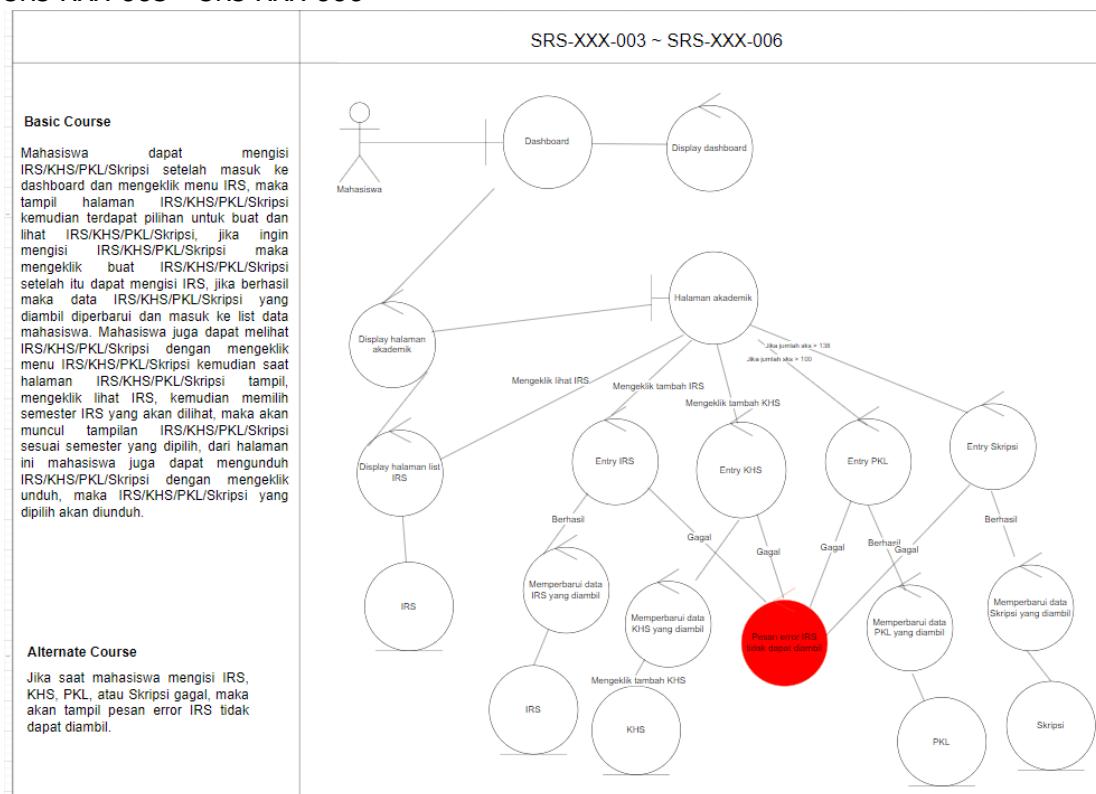
#### 1. SRS-XXX-001



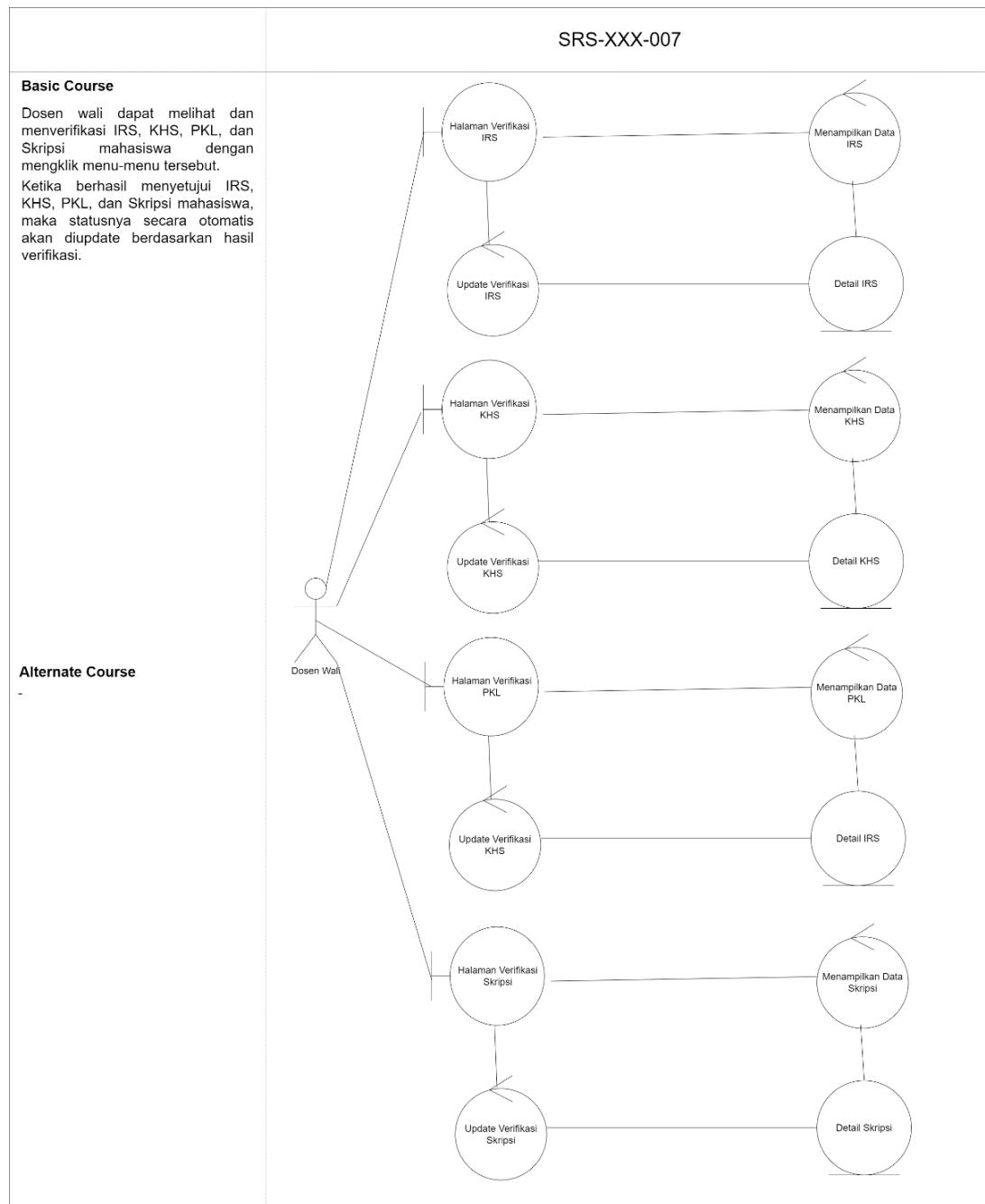
## 2. SRS-XXX-002

	SRS-XXX-02
<p><b>Basic Course</b></p> <p>Setelah berhasil login dan masuk ke halaman update data baru, mahasiswa melakukan tindakan seperti mengklik tombol "Update Profile". Sistem merespons dengan menampilkan halaman formulir perbaruan profil, memungkinkan mahasiswa untuk memperbarui data mereka. Setelah mahasiswa mengisi formulir dan mengklik "Submit", sistem memberikan respons edgan menampilkan pesan sukses kembali ke halaman [rofil yang diperbarui. Mahasiswa dapat melihat perubahan yang baru mereka lakukan dan jika diinginkan, melanjutkan interaksi lebih lanjut.</p>	<pre> stateDiagram-v2     [*] --&gt; HalamanUpdateDataBaru     HalamanUpdateDataBaru --&gt; MenampilkanDataInitial     HalamanUpdateDataBaru --&gt; MemperbaruiData     MenampilkanDataInitial --&gt; User     MemperbaruiData --&gt; User     User --&gt; Mahasiswa     User --&gt; PKL     User --&gt; Skripsi   </pre>

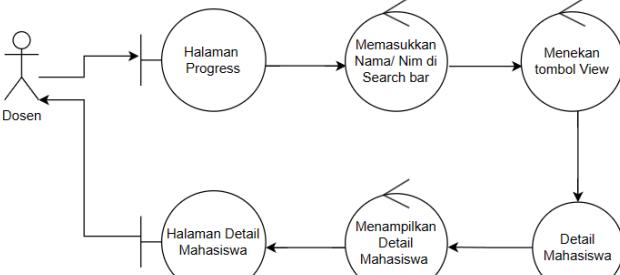
### 3. SRS-XXX-003 – SRS-XXX-006



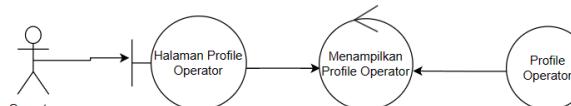
#### 4. SRS-XXX-007



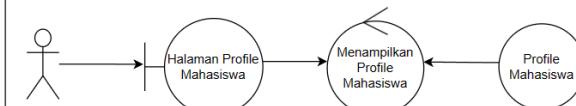
## 5. SRS-XXX-008

SRS - XXX - 08	
<b>Basic Course</b> Setelah berhasil login dan masuk ke Halaman progress, Dosen dapat memasukkan Nama/Nim di search bar dan kemudian dapat menekan tombol view yang akan menampilkan detail akademik mahasiswa.	 <pre> graph LR     Dosen((Dosen)) --&gt; HP[Halaman Progress]     HP --&gt; MN[Memasukkan Nama/ Nim di Search bar]     MN --&gt; MT[Menekan tombol View]     MT --&gt; DM[Detail Mahasiswa]     DM --&gt; MD[Menampilkan Detail Mahasiswa]     MD --&gt; HP   </pre>

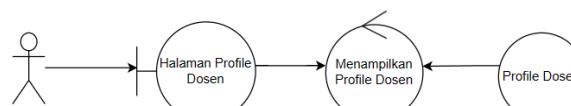
## 6. SRS-XXX-009

SRS - XXX - 09	
<b>Basic Course</b> Setelah berhasil login, operator dapat masuk ke halaman profile.	 <pre> graph LR     Operator((Operator)) --&gt; HPO[Halaman Profile Operator]     HPO --&gt; MPO[Menampilkan Profile Operator]     MPO --&gt; PO[Profile Operator]     PO --&gt; HPO   </pre>

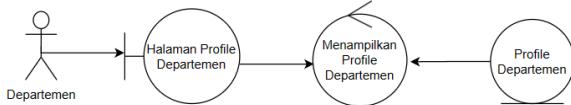
## 7. SRS-XXX-010

SRS - XXX - 10	
<b>Basic Course</b> Setelah berhasil login, Mahasiswa dapat masuk ke halaman profile.	 <pre> graph LR     Mahasiswa((Mahasiswa)) --&gt; HPM[Halaman Profile Mahasiswa]     HPM --&gt; MPM[Menampilkan Profile Mahasiswa]     MPM --&gt; PM[Profile Mahasiswa]     PM --&gt; HPM   </pre>

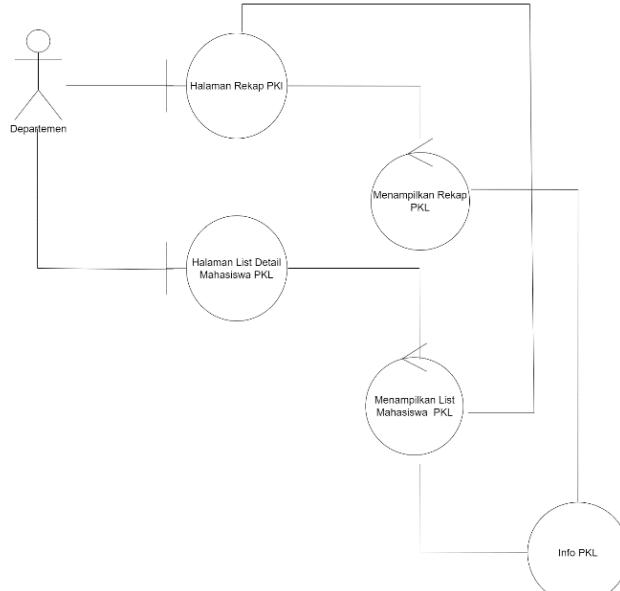
## 8. SRS-XXX-011

SRS - XXX - 11	
<b>Basic Course</b> Setelah berhasil login, Dosen dapat masuk ke halaman profile.	 <pre> graph LR     Dosen((Dosen)) --&gt; HPD[Halaman Profile Dosen]     HPD --&gt; MPD[Menampilkan Profile Dosen]     MPD --&gt; PD[Profile Dosen]     PD --&gt; HPD   </pre>

## 9. SRS-XXX-012

	SRS - XXX - 12
<b>Basic Course</b> Setelah berhasil login, Departemen dapat masuk ke halaman profile.	 <pre> graph LR     Departemen((Departemen)) --&gt; HalamanProfile(Halaman Profile Departemen)     HalamanProfile --&gt; MenampilkanProfile(Menampilkan Profile Departemen)     MenampilkanProfile --&gt; ProfileDepartemen(Profile Departemen)     </pre>

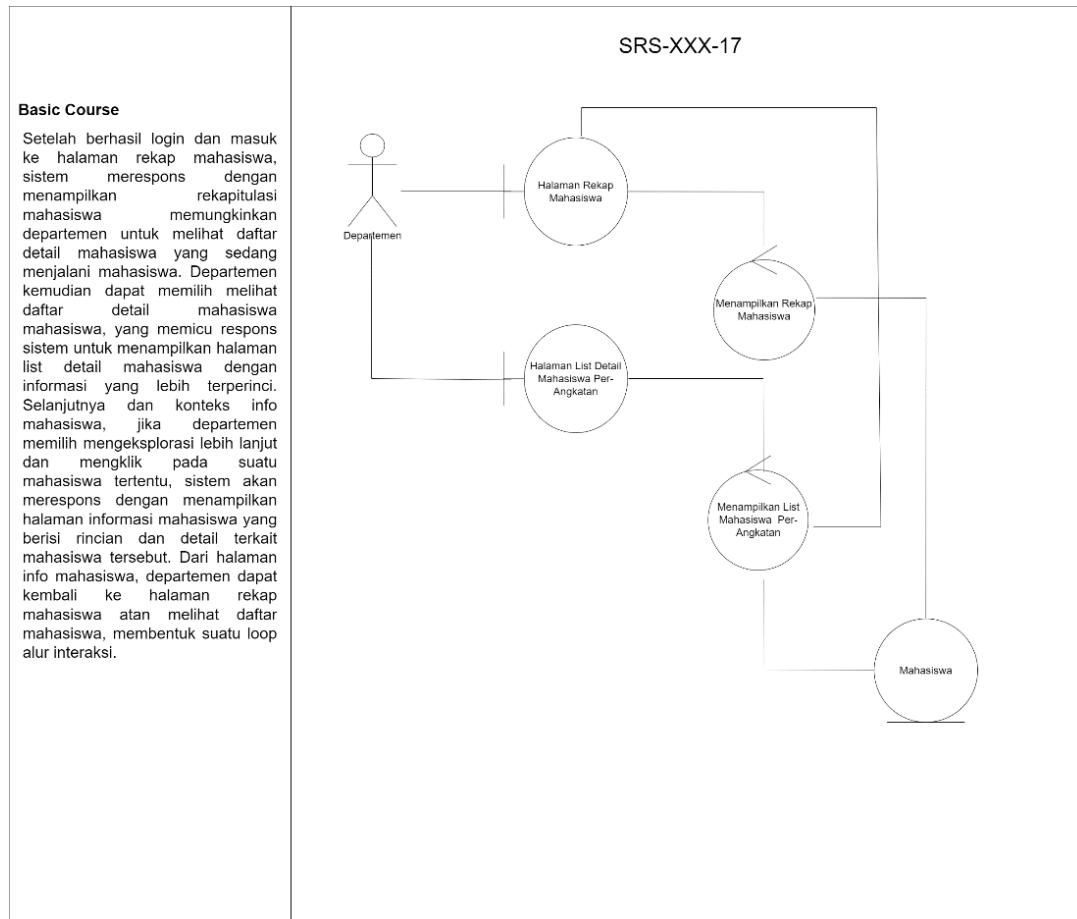
## 10. SRS-XXX-013 dan SRS-XXX-014

	SRS-XXX-13 ~ SRS-XXX-14
<b>Basic Course</b> Setelah berhasil login dan masuk ke halaman rekap pkl, sistem merespons dengan menampilkan rekapitulasi PKL, memungkinkan departemen untuk melihat daftar detail mahasiswa yang sedang menjalani PKL. Departemen kemudian dapat memilih melihat daftar detail mahasiswa PKL, yang memicu respons sistem untuk menampilkan halaman list detail mahasiswa dengan informasi yang lebih terperinci. Selanjutnya dan konteks info PKL, jika departemen memilih mengeksplorasi lebih lanjut dan mengklik pada suatu PKL tertentu, sistem akan merespons dengan menampilkan halaman informasi PKL yang berisi rincian dan detail terkait PKL tersebut. Dari halaman info PKL, departemen dapat kembali ke halaman rekap PKL atau melihat daftar mahasiswa PKL, membentuk suatu loop alur interaksi.	 <pre> graph TD     Departemen((Departemen)) --&gt; HalamanRekap(Halaman Rekap PKL)     HalamanRekap --&gt; MenampilkanRekap(Menampilkan Rekap PKL)     HalamanRekap --&gt; HalamanList(Halaman List Detail Mahasiswa PKL)     MenampilkanRekap --&gt; InfoPKL(Info PKL)     HalamanList --&gt; MenampilkanList(Menampilkan List Mahasiswa PKL)     MenampilkanList --&gt; HalamanRekap     InfoPKL --&gt; HalamanRekap     </pre>

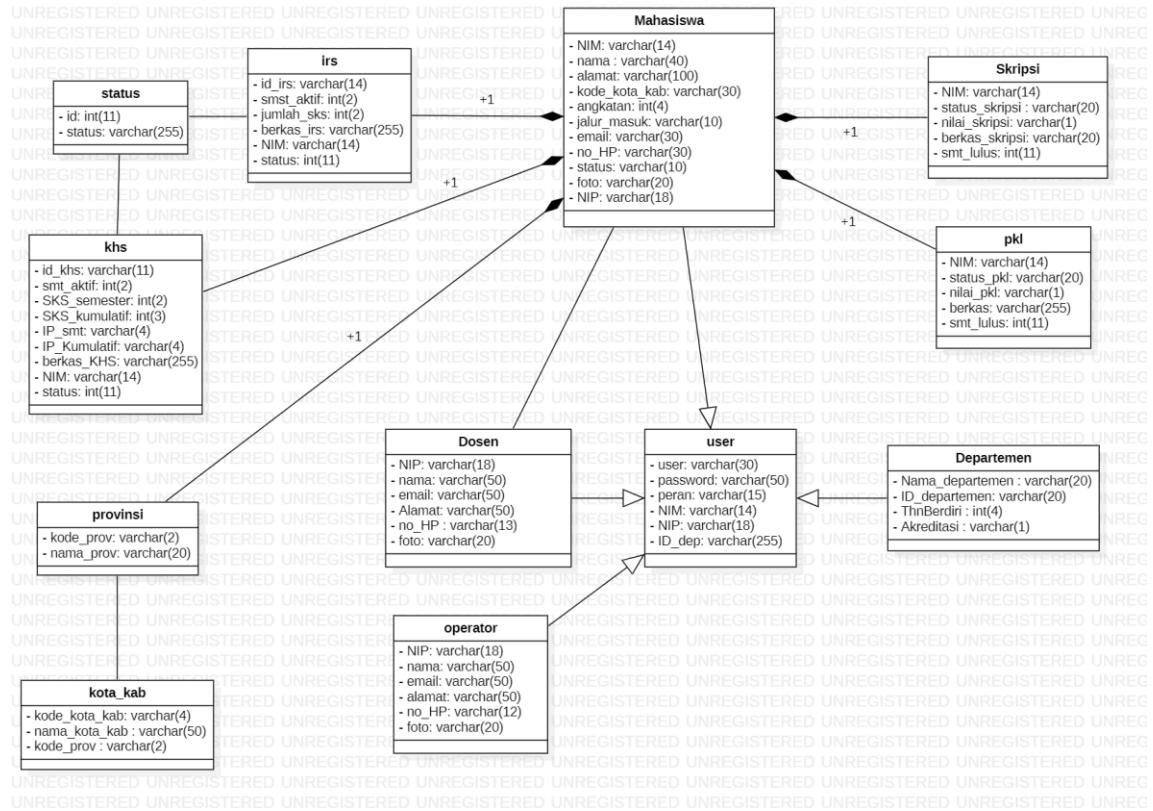
## 11. SRS-XXX-015 dan SRS-XXX-016

	SRS-XXX-15 ~ SRS-XXX-16
<b>Basic Course</b> <p>Setelah berhasil login dan masuk ke halaman rekap skripsi, sistem merespons dengan menampilkan rekapitulasi skripsi, memungkinkan departemen untuk melihat daftar detail mahasiswa yang sedang menjalani skripsi. Departemen kemudian dapat memilih melihat daftar detail mahasiswa skripsi, yang memicu respons sistem untuk menampilkan halaman list detail mahasiswa dengan informasi yang lebih terperinci. Selanjutnya dan konteks info skripsi, jika departemen memilih mengeksplorasi lebih lanjut dan mengklik pada suatu skripsi tertentu, sistem akan merespons dengan menampilkan halaman informasi skripsi yang berisi rincian dan detail terkait skripsi tersebut. Dari halaman info skripsi, departemen dapat kembali ke halaman rekap skripsi atau melihat daftar mahasiswa skripsi, membentuk suatu loop alur interaksi.</p>	<pre> stateDiagram-v2     [*] --&gt; HalamanRekapSkripsi     HalamanRekapSkripsi --&gt; MenampilkanRekapSkripsi     HalamanRekapSkripsi --&gt; HalamanListDetailMahasiswaSkripsi     MenampilkanRekapSkripsi --&gt; HalamanRekapSkripsi     HalamanListDetailMahasiswaSkripsi --&gt; MenampilkanListMahasiswaSkripsi     MenampilkanListMahasiswaSkripsi --&gt; InfoSkripsi   </pre>

## 12. SRS-XXX-017



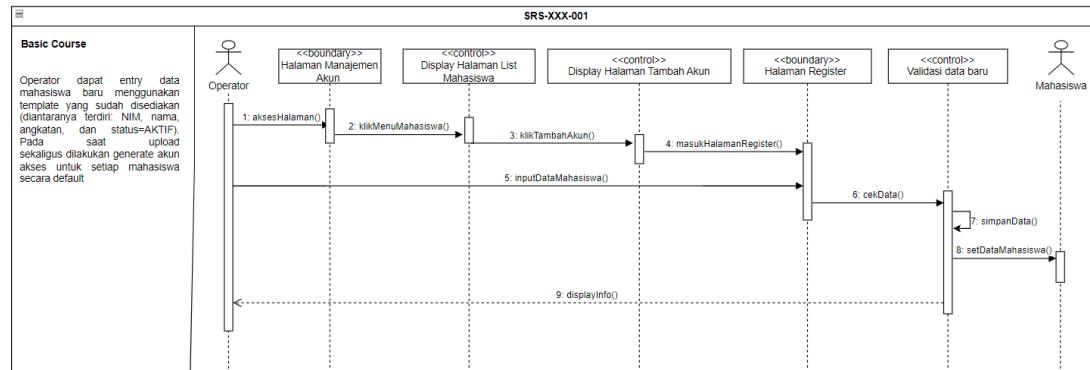
## 4.2 Update Domain Model



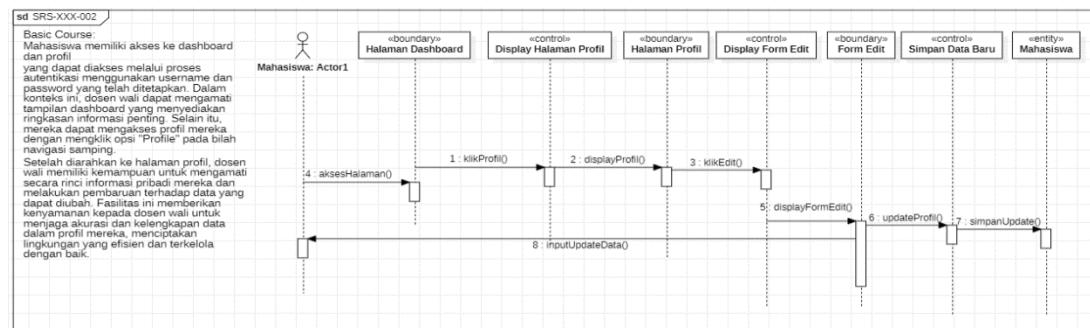
## 5 Detailed Design

### 5.1 Sequence Diagram

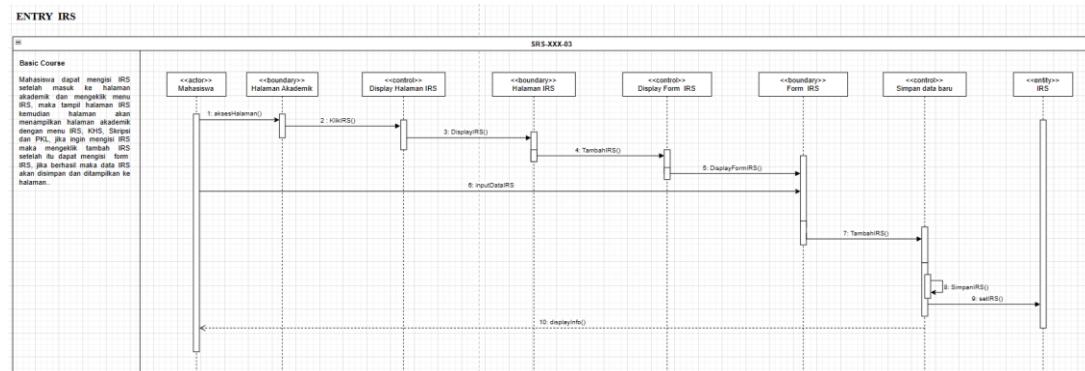
#### 1. SRS-XXX-001



#### 2. SRS-XXX-002

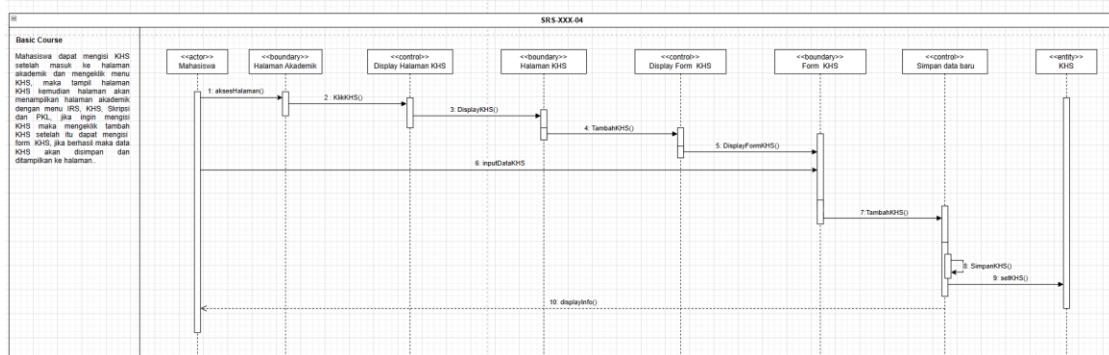


#### 3. SRS-XXX-003

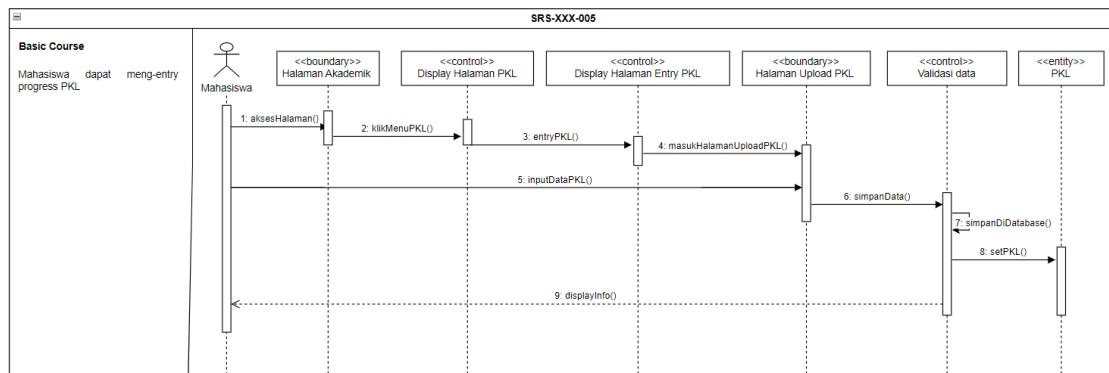


#### 4. SRS-XXX-004

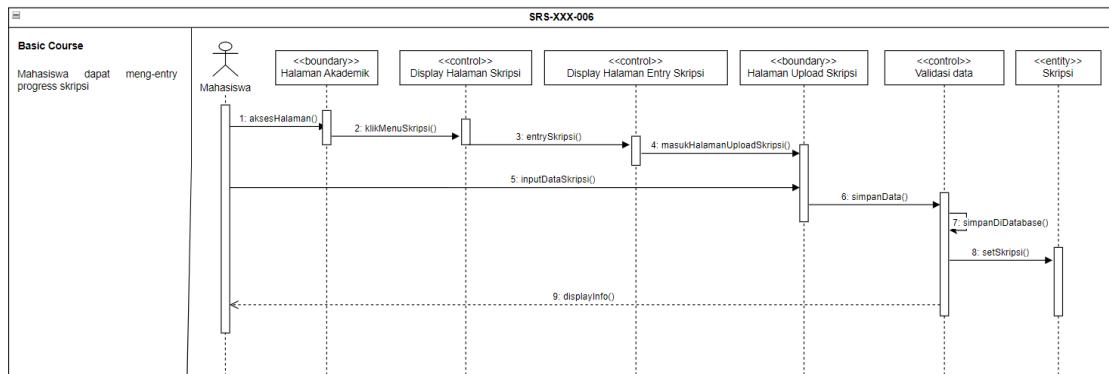
ENTRY KHS



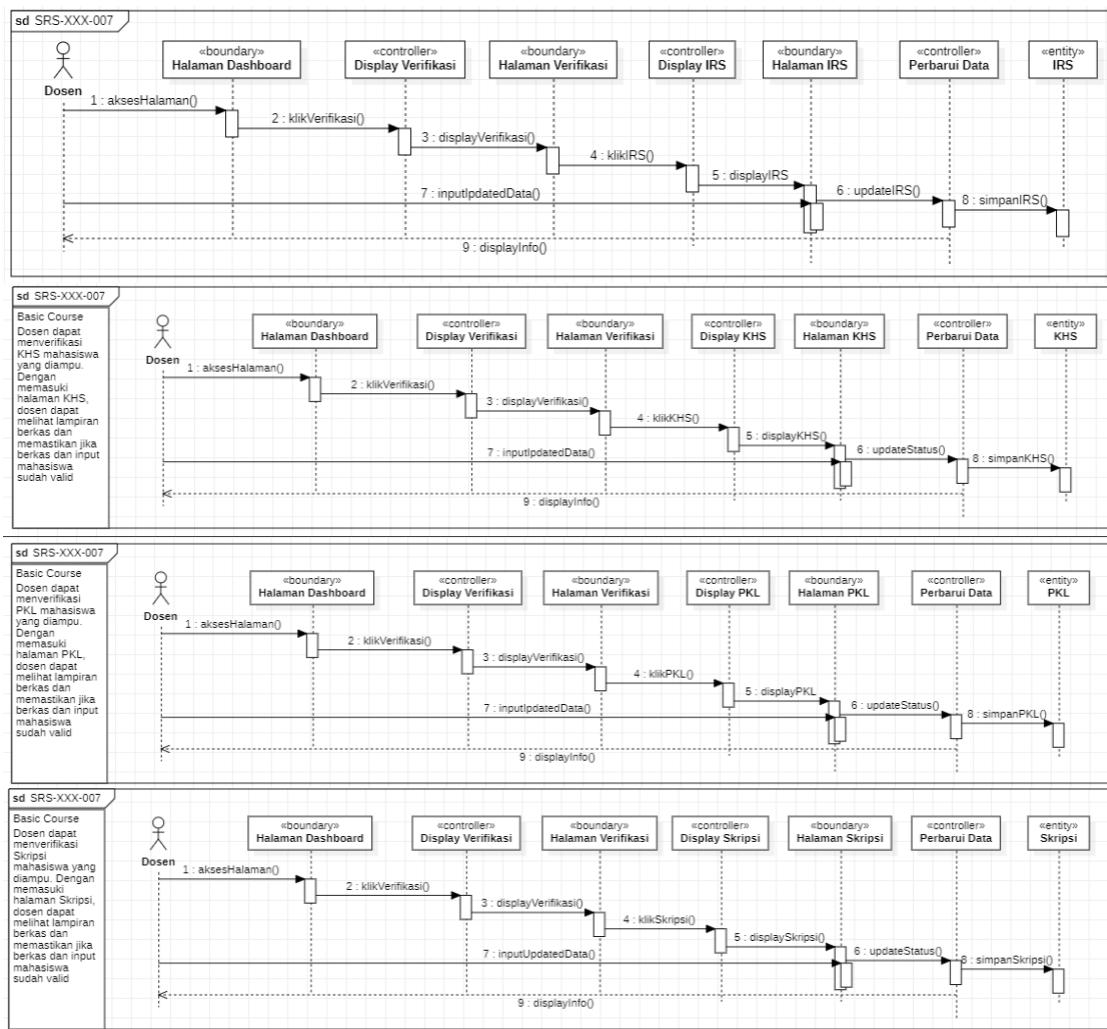
#### 5. SRS-XXX-005



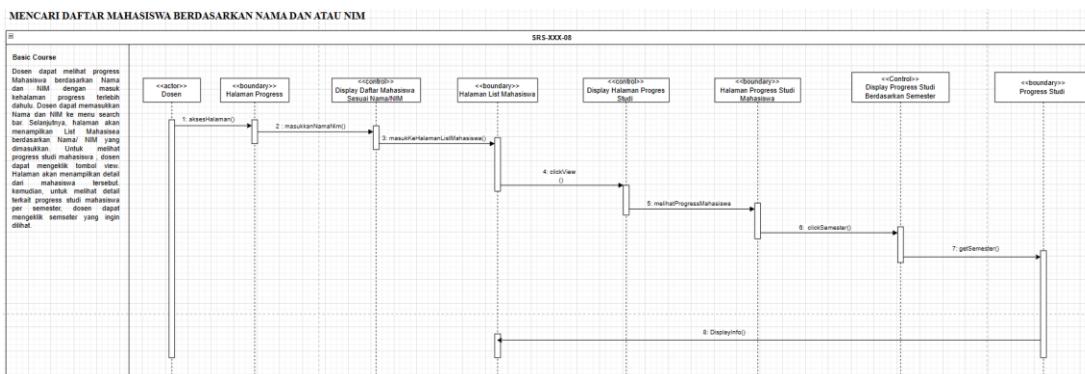
#### 6. SRS-XXX-006



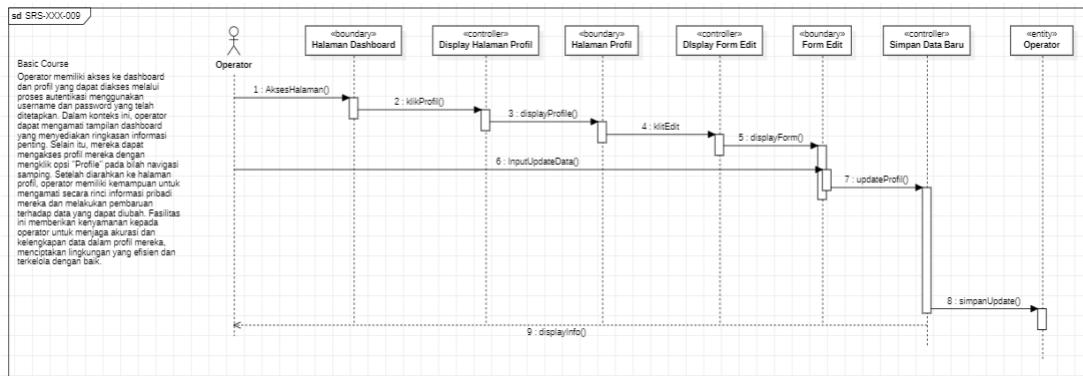
## 7. SRS-XXX-007



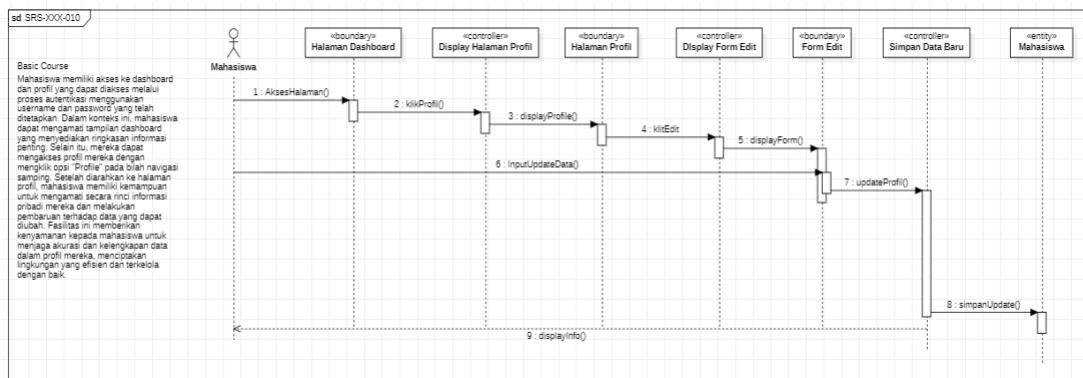
## 8. SRS-XXX-008



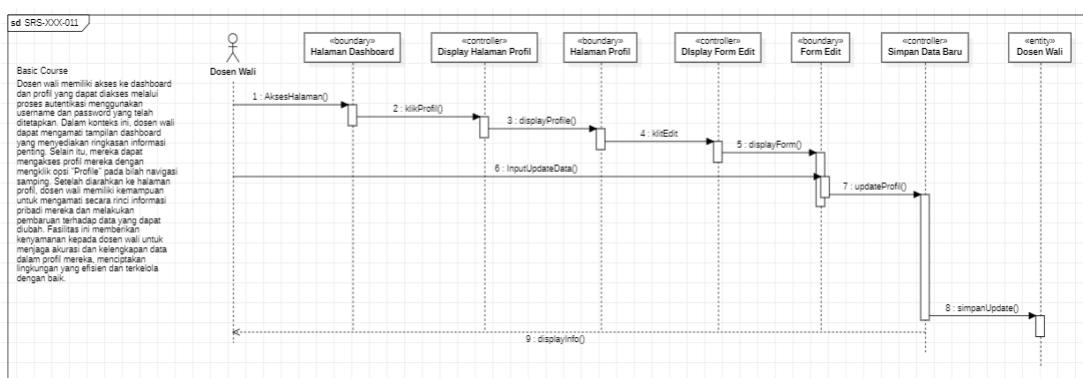
## 9. SRS-XXX-009



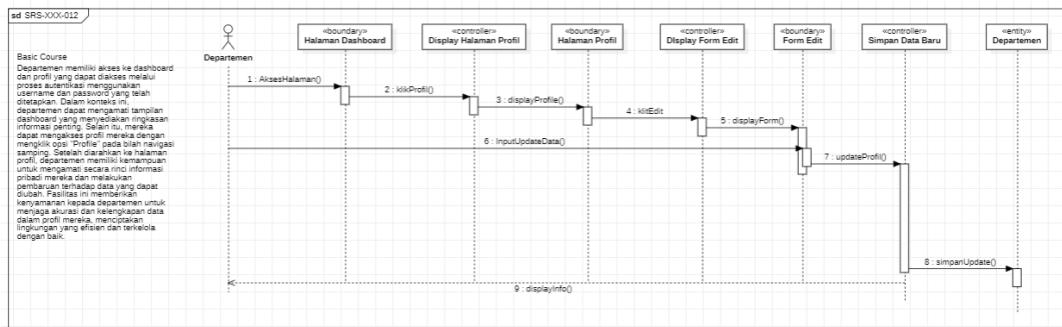
## 10. SRS-XXX-010



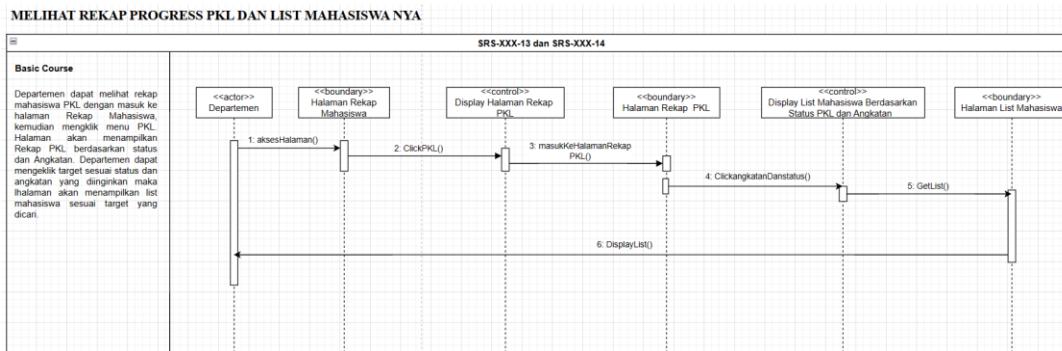
## 11. SRS-XXX-011



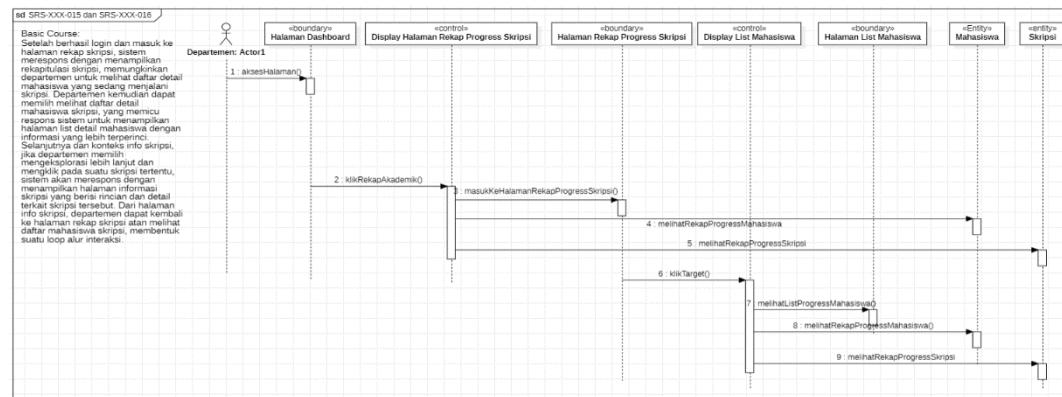
## 12. SRS-XXX-012



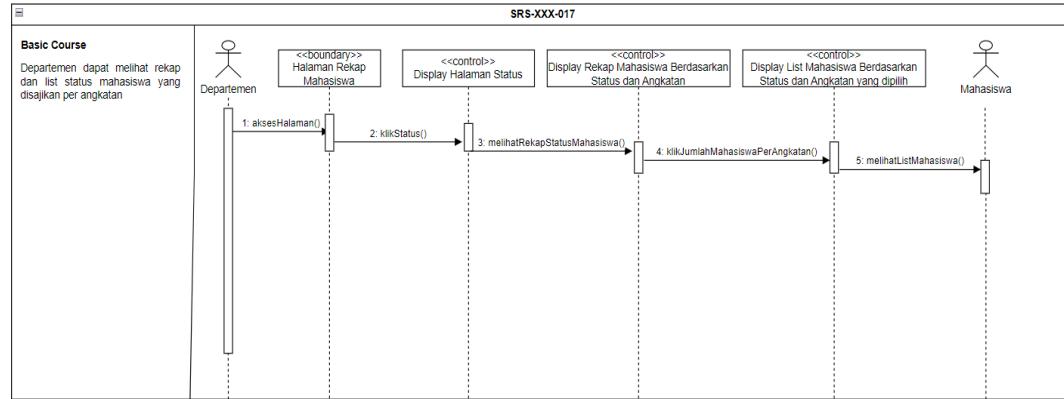
## 13. SRS-XXX-013 dan SRS-XXX-014



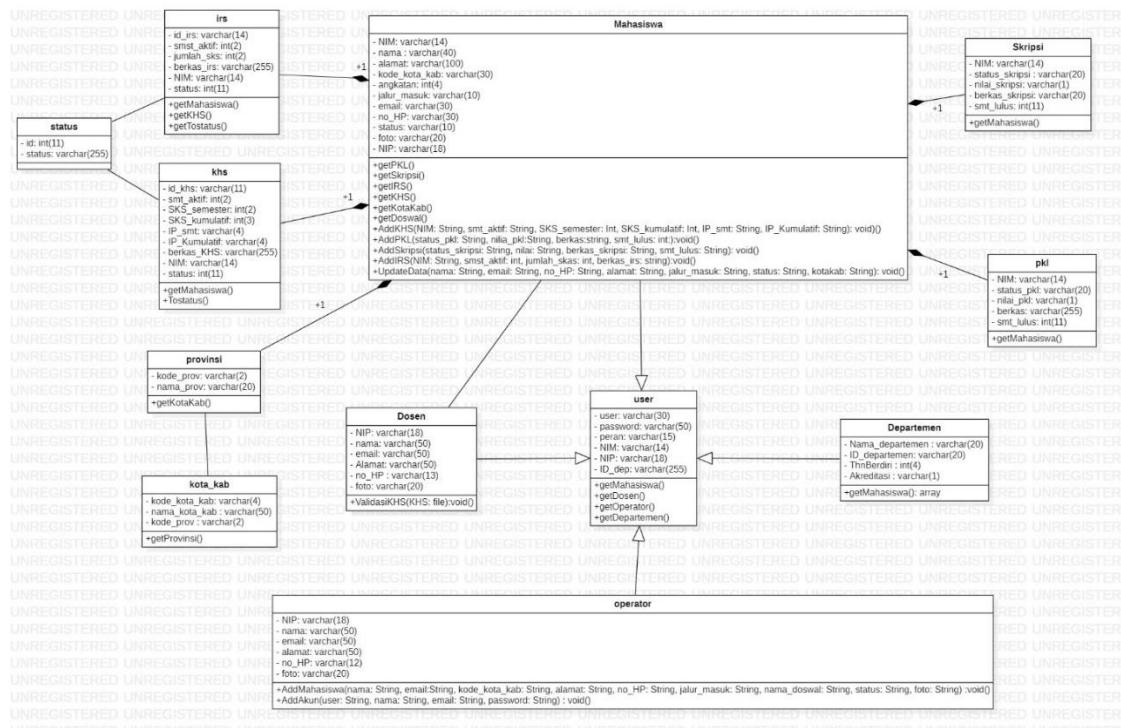
## 14. SRS-XXX-015 dan SRS-XXX-016



## 15. SRS-XXX-017



## 5. 2 Class Diagram



## 6 Implementation

---

### 6.1 Code

#### 1. Model dan Controller

##### a. LoginController.php

```
function login(Request $request)
{
    $email = $request->input('email');
    $password = $request->input('password');

    $users = Users::all();

    foreach ($users as $user) {
        if ($user->user == $email && $user->password == $password
&& $user->peran == "mahasiswa") {
            if ($user->mahasiswa->email != NULL) {
                // Authentication successful
                Session::put("email", $email);
                Session::keep("email"); // Memastikan data sesi
                "email" tetap ada
                return redirect("/dashboard/mahasiswa/{$user-
>NIM}");
            } else {
                return redirect("/dashboard/mahasiswa/{$user-
>NIM}/updateAcc");
            }
        } else if ($user->user == $email && $user->password ==
$password && $user->peran == "dosen") {
            // Authentication successful
            Session::put("email", $email);
            Session::keep("email"); // Memastikan data sesi
            "email" tetap ada
            return redirect("/dashboard/dosen/{$user->NIP}");
        } else if ($user->user == $email && $user->password ==
$password && $user->peran == "operator") {
    }}
```

```
// Authentication successful
Session::put("email", $email);
Session::keep("email"); // Memastikan data sesi
'email' tetap ada
    return redirect("/dashboard/operator/{$user->NIP}");
} else if ($user->user == $email && $user->password ==
$password && $user->peran == "departemen") {
    // Authentication successful
    Session::put("email", $email);
    Session::keep("email"); // Memastikan data sesi
'email' tetap ada
    return redirect("/dashboard/departemen/{$user-
>ID_dep}");
}
}

// Authentication failed
return "Username atau password salah";
}
```

### b. Departemen.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class Departemen extends Model
{
    public $timestamps = false;
    protected $table = 'departemen';
    protected $primaryKey = 'ID_departemen';
    protected $fillable = ['Nama_departemen', 'ID_departemen',
'ThnBerdiri', 'Akreditasi'];
    public $incrementing = false;
    use HasFactory;
}
```

### c. DepartemenController.php

```

function masukDepartemen()
{
    $user = Users::where('peran', 'departemen')->first();
    return view("dashboard.departemen")->with("user", $user);
}

function profileDep($ID_dep)
{
    $user = Users::where('ID_dep', $ID_dep)->first();
    $provinsi = Provinsi::all();
    $kota = KotaKab::all();

    if ($user) {
        return view("profile.departemen")->with("user", $user)-
>with("provinsi", $provinsi)->with("kota", $kota);
    } else {
        return "User not found";
    }
}

function progressDep($ID_dep)
{
    $user = Users::where('ID_dep', $ID_dep)->first();
    $mahasiswa = Mahasiswa::all();
    return view("progress.departemen")->with("user", $user)-
>with("mahasiswa", $mahasiswa);
}

function recapDepartemen()
{
    $user = Users::where("peran", 'departemen')->first();
    $mahasiswa = Mahasiswa::all();
    return view("recap")->with("user", $user)->with("mahasiswa",
$mahasiswa);
}

function progressMHSDep($ID_dep, $nim)
{
    $user = Users::where('ID_dep', $ID_dep)->first();
    $mahasiswa = Mahasiswa::where('NIM', $nim)->first();
    $pkl = PKL::where('NIM', $nim)->first();
    $skripsi = Skripsi::where('NIM', $nim)->first();
    $irs = IRS::all();
    $khs = KHS::all();
    return view("details.student")->with("user", $user)-
>with("mahasiswa", $mahasiswa)->with("pkl", $pkl)->with("skripsi",
$skripsi)->with("irs", $irs)->with("khs", $khs);
}

function listRecapDepartemen($ID_dep, $angkatan, $jenis, $status)
{
    if (Users::where("peran", 'departemen')) {

```

```

    }
    $user = Users::where("ID_dep", $ID_dep)->first();
    $mahasiswa = Mahasiswa::all();
    return view('list.recap')->with('user', $user)-
>with('mahasiswa', $mahasiswa)->with('angkatan', $angkatan)-
>with('status', $status)->with('jenis', $jenis);
}

function listRecapSkripsiDep($ID_dep, $angkatan, $jenis, $status){
    $user = Users::where("ID_dep", $ID_dep)->first();
    $mahasiswa = Mahasiswa::all();
    return view('list.recapSkripsi')->with('user', $user)-
>with('mahasiswa', $mahasiswa)->with('angkatan', $angkatan)-
>with('status', $status)->with('jenis', $jenis);
}

function listStatus($ID_dep, $angkatan, $status)
{
    $user = Users::where("ID_dep", $ID_dep)->first();
    $mahasiswa = Mahasiswa::all();
    return view('list.status')->with('user', $user)-
>with('mahasiswa', $mahasiswa)->with('angkatan', $angkatan)-
>with('status', $status);
}

function progressDep($ID_dep)
{
    $user = Users::where('ID_dep', $ID_dep)->first();
    $mahasiswa = Mahasiswa::all();
    return view("progress.departemen")->with("user", $user)-
>with("mahasiswa", $mahasiswa);
}

```

#### d. Operator.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Operator extends Model
{
    public $timestamps = false;
    protected $table = 'operator';
    protected $primaryKey = 'NIP';
    protected $fillable = ['NIP', 'nama', 'alamat', 'no_HP', 'foto'];
    public $incrementing = false;
    use HasFactory;
}

```

### e. OperatorController.php

```

function masukKaryawan($nip)
{
    $user = Users::where('NIP', $nip)->first();
    $mahasiswa = Mahasiswa::all();

    if ($user->peran == 'dosen') {
        return view("dashboard.dosen")->with("user", $user)-
>with("mahasiswa", $mahasiswa);
    } else if ($user->peran == 'operator') {
        return view("dashboard.operator")->with("user", $user);
    } else {
        return "User not found"; // You can change this message as
needed
    }
}
function profileOperator($nip)
{
    $user = Users::where('NIP', $nip)->first();
    $provinsi = Provinsi::all();
    $kota = KotaKab::all();

    if ($user) {
        return view("profile.operator")->with("user", $user)-
>with("provinsi", $provinsi)->with("kota", $kota);
    } else {
        return "User not found";
    }
}
function recapOperator($nip)
{
    $user = Users::where('NIP', $nip)->first();
    $mahasiswa = Mahasiswa::all();
    return view("recap")->with("user", $user)->with("mahasiswa",
$mahasiswa);
}
function addMahasiswa(Request $request)
{
    $doswal = Dosen::all();
    $randomDosen = $doswal->random();
    $mahasiswa = new Mahasiswa();
    $user = new Users();
    $randomNumber = str_pad(mt_rand(1, 99999999999999), 14, '0',
STR_PAD_LEFT);
    $mahasiswa->NIM = $randomNumber;
    $user->NIM = $randomNumber;
    $user->password = $request->input('password');
    $user->peran = "mahasiswa";
    $mahasiswa->nama = $request->input('nama');
    $mahasiswa->email = $request->input('email');
    $user->email = $request->input('email');
    $mahasiswa->kode_kota_kab = $request->input('kotakab');
    $mahasiswa->alamat = $request->input('alamat');
    $mahasiswa->no_HP = $request->input('noHP');
    $mahasiswa->jalur_masuk = $request->input('jalur_masuk');
}

```

```

$mahasiswa->nama_doswal = $randomDosen->nama_doswal;
$mahasiswa->persetujuan = "Belum Disetujui";
$mahasiswa->status = "AKTIF";
$mahasiswa->foto = "";
$mahasiswa->save();
$user->save();
return redirect('/')->with('success', 'Data Mahasiswa Berhasil
Diregister');
}
function getMahasiswa($nip)
{
    $user = Users::where('NIP', $nip)->first();
    $mahasiswa = Mahasiswa::all();
    $dosen = Dosen::all();

    return view("operation.operation")->with("user", $user)-
>with("mahasiswa", $mahasiswa)->with("dosen", $dosen);
}
function detailMahasiswa($nip, $nim)
{
    $user = Users::where('NIP', $nip)->first();
    $mahasiswa = Mahasiswa::where('NIM', $nim)->first();
    $khs = KHS::where('NIM', $nim);
    $irs = IRS::where('NIM', $nim);
    $pkl = PKL::where('NIM', $nim);
    $skripsi = Skripsi::where('NIM', $nim);
    $provinsi = Provinsi::all();
    $kota = KotaKab::all();

    return view("operation.details")->with("user", $user)-
>with("mahasiswa", $mahasiswa)->with("khs", $khs)->with("irs", $irs)-
>with("pkl", $pkl)->with("skripsi", $skripsi)->with("provinsi",
$provinsi)->with("kota", $kota);
}
function addAccount()
{
    $lectures = Dosen::all();
    $prov = Provinsi::all();
    $kotaKabData = DB::table('kota_kab')
        ->join('provinsi', 'kota_kab.kode_prov', '=', 'provinsi.kode_prov')
        ->select('kota_kab.kode_kota_kab',
'kota_kab.nama_kota_kab', 'provinsi.kode_prov')
        ->get()
        ->groupBy('kode_prov')
        ->map(function ($kotaKab) {
            return $kotaKab->pluck('nama_kota_kab',
'kode_kota_kab');
        });
    return view("operation.addAccount")->with('prov', $prov)-
>with('kotaKabData', $kotaKabData)->with('lectures', $lectures);
}
function confirmAddAccount(Request $request)
{
    $user = new Users();
    $mahasiswa = new Mahasiswa();

```

```

$mahasiswa->nama = $request->nama;
$mahasiswa->NIM = $request->nim;
$mahasiswa->angkatan = $request->angkatan;
$mahasiswa->status = $request->status;
$mahasiswa->NIP = $request->doswal;
$mahasiswa->save();

$user->NIM = $mahasiswa->NIM;
$user->peran = 'mahasiswa';
$length = 10;
$characters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
$password = '';

for ($i = 0; $i < $length; $i++) {
    $password .= $characters[rand(0, strlen($characters) - 1)];
}

$user->password = $password;
$nama = $request->nama;
$nim = $request->nim;

// Ambil first name dari nama
$namaParts = explode(' ', $nama);
$firstName = $namaParts[0];

// Ambil 5 angka terakhir dari NIM
nimLast5 = substr($nim, -5);

// Gabungkan first name dan 5 angka terakhir dari NIM
$username = $firstName . $nimLast5;
$user->user = $username;
$user->save();

return redirect()->back();
}

```

#### f. Dosen.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Dosen extends Model
{
    public $timestamps = false;
    protected $table = 'dosen';
    protected $primaryKey = 'NIP';
    protected $fillable = ['NIP', 'nama', 'email', 'alamat', 'no_HP',
    'foto'];
}

```

```
    public $incrementing = false;
    use HasFactory;
}
```

#### g. DoswalController.php

```
function masukKaryawan($nip)
{
    $user = Users::where('NIP', $nip)->first();
    $mahasiswa = Mahasiswa::all();

    if ($user->peran == 'dosen') {
        return view("dashboard.dosen")->with("user", $user)-
>with("mahasiswa", $mahasiswa);
    } else if ($user->peran == 'operator') {
        return view("dashboard.operator")->with("user", $user);
    } else {
        return "User not found"; // You can change this message as
needed
    }
}
function profileDos($nip)
{
    $user = Users::where('NIP', $nip)->first();
    $provinsi = Provinsi::all();
    $kota = KotaKab::all();

    if ($user) {
        return view("profile.dosen")->with("user", $user)-
>with("provinsi", $provinsi)->with("kota", $kota);
    } else {
        return "User not found";
    }
}
function addMahasiswa(Request $request)
{
    $doswal = Dosen::all();
    $randomDosen = $doswal->random();
    $mahasiswa = new Mahasiswa();
    $user = new Users();
    $randomNumber = str_pad(mt_rand(1, 99999999999999), 14, '0',
STR_PAD_LEFT);
    $mahasiswa->NIM = $randomNumber;
    $user->NIM = $randomNumber;
    $user->password = $request->input('password');
    $user->peran = "mahasiswa";
    $mahasiswa->nama = $request->input('nama');
    $mahasiswa->email = $request->input('email');
    $user->email = $request->input('email');
    $mahasiswa->kode_kota_kab = $request->input('kotakab');
    $mahasiswa->alamat = $request->input('alamat');
    $mahasiswa->no_HP = $request->input('noHP');
    $mahasiswa->jalur_masuk = $request->input('jalur_masuk');
    $mahasiswa->nama_doswal = $randomDosen->nama_doswal;
    $mahasiswa->persetujuan = "Belum Disetujui";
```

```

        $mahasiswa->status = "AKTIF";
        $mahasiswa->foto = "";
        $mahasiswa->save();
        $user->save();
        return redirect('/')->with('success', 'Data Mahasiswa Berhasil
Diregister');
    }

function validasi($nip)
{
    $user = Users::where('NIP', $nip)->first();
    $mahasiswa = Mahasiswa::all();
    $irs = IRS::all();
    $khs = KHS::all();
    $pkl = PKL::all();
    $skripsi = Skripsi::all();

    if ($user->peran == 'dosen') {
        return view("validation")->with("user", $user)-
>with("mahasiswa", $mahasiswa)->with("irs", $irs)->with("khs", $khs)-
>with("pkl", $pkl)->with("skripsi", $skripsi);
    } else if ($user->peran == 'operator') {
        return view("dashboard.operator")->with("user", $user);
    } else {
        return "User not found"; // You can change this message as
needed
    }
}

function getMahasiswa($nip)
{
    $user = Users::where('NIP', $nip)->first();
    $mahasiswa = Mahasiswa::all();
    $dosen = Dosen::all();

    return view("operation.operation")->with("user", $user)-
>with("mahasiswa", $mahasiswa)->with("dosen", $dosen);
}

function addAccount()
{
    $lectures = Dosen::all();
    $prov = Provinsi::all();
    $kotaKabData = DB::table('kota_kab')
        ->join('provinsi', 'kota_kab.kode_prov', '=', 'provinsi.kode_prov')
        ->select('kota_kab.kode_kota_kab',
    'kota_kab.nama_kota_kab', 'provinsi.kode_prov')
        ->get()
        ->groupBy('kode_prov')
        ->map(function ($kotaKab) {
            return $kotaKab->pluck('nama_kota_kab',
    'kode_kota_kab');
        });
    return view("operation.addAccount")->with('prov', $prov)-
>with('kotaKabData', $kotaKabData)->with('lectures', $lectures);
}

```

```

    }

function validasiIRS($nip, $id_irs)
{
    $irs = IRS::find($id_irs);
    $irs->status = 1;
    $irs->save();
    return redirect("/dashboard/dosen/{$nip}");
}
function tolakPKL($nip, $nim)
{
    $pkl = PKL::where('NIM', $nim)->first();
    $pkl->status_pkl = 'SEGERA DIREVISI';
    $pkl->save();
    return redirect("/dashboard/dosen/{$nip}");
}

function tolakSkripsi($nip, $nim) {
    $skripsi = Skripsi::where('NIM', $nim)->first();
    $skripsi->status_skripsi = 'SEGERA DIREVISI';
    $skripsi->save();
    return redirect("/dashboard/dosen/{$nip}");
}
function validasiKHS($nip, $id_khs)
{
    $khs = KHS::find($id_khs);
    $khs->status = 1;
    $khs->save();
    return redirect("/dashboard/dosen/{$nip}");
}
function validasiPKL($nip, $nim)
{
    $pkl = PKL::find($nim);
    $smt_terakhir = KHS::where('NIM', $nim)
        ->orderBy('smst_aktif', 'desc')
        ->first()
        ->smst_aktif;
    $pkl->status_pkl = 'lulus';
    $pkl->smt_lulus = $smt_terakhir;
    $pkl->save();
    return redirect("/dashboard/dosen/{$nip}");
}
function validasiSkripsi($nip, $nim)
{
    $skripsi = Skripsi::find($nim);
    $smtr_terakhir = IRS::where('NIM', $nim)
        ->orderBy('smst_aktif', 'desc')
        ->first()
        ->smst_aktif;
    $skripsi->status_skripsi = 'lulus';
    $skripsi->smt_lulus = $smtr_terakhir;
    $skripsi->tanggal_sidang = Carbon::now();
    $skripsi->save();
    return redirect("/dashboard/dosen/{$nip}");
}

function recapDosen($nip)
{
}

```

```
$user = Users::where('NIP', $nip)->first();
$mahasiswa = Mahasiswa::all();
return view("recap.recapDos")->with("user", $user)-
>with("mahasiswa", $mahasiswa);
}
```

#### h. Mahasiswa.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\PKL;
use App\Models\Skripsi;
use App\Models\IRS;
use App\Models\KHS;
use App\Models\KotaKab;
use App\Models\Dosen;

class Mahasiswa extends Model
{
    public $timestamps = false;
    protected $table = 'mahasiswa';
    protected $primaryKey = 'NIM';
    protected $fillable = ['NIM', 'nama', 'email', 'alamat', 'no_HP',
    'angkatan', 'status', 'jalur_masuk', 'foto', 'kode_kota_kab', 'NIP'];
    public $incrementing = false;

    public function pkl(){
        return $this->belongsTo(PKL::class, 'NIM', 'NIM');
    }

    public function skripsi(){
        return $this->belongsTo(Skripsi::class, 'NIM', 'NIM');
    }

    public function irs(){
        return $this->belongsTo(IRS::class, 'NIM', 'NIM');
    }

    public function khs(){
        return $this->belongsTo(KHS::class, 'NIM', 'NIM');
    }

    public function kotakab(){
        return $this->belongsTo(KotaKab::class, 'kode_kota_kab',
        'kode_kota_kab');
    }
}
```

```

    public function doswal(){
        return $this->belongsTo(Dosen::class, 'NIP', 'NIP');
    }
    use HasFactory;
}

```

### i. MahasiswaController.php

```

function masukMahasiswa($nim)
{
    $user = Users::where('NIM', $nim)->first();
    $khs = KHS::all();

    if ($user) {
        return view("dashboard.mahasiswa")->with("user", $user)->with("khs", $khs);
    } else {
        return view('updateAcc')->with("user", $user); // You can
change this message as needed
    }
}
function edit(Request $request, $nim)
{
    $user = Users::where('NIM', $nim)->first();
    $user->mahasiswa->nama = $request->input("nama");
    $user->mahasiswa->email = $request->input("email");
    $phone_number = $request->input("hp");
    $user->mahasiswa->no_HP = $phone_number;
    $user->mahasiswa->alamat = $request->input("alamat");
    $user->mahasiswa->jalur_masuk = $request-
>input("jalur_masuk");
    $user->mahasiswa->status = $request->input("status");
    $user->mahasiswa->kode_kota_kab = $request->input("kotakab");
    $user->mahasiswa->save(); // Use "->save()" to save the changes
to the database

    return redirect()->back()->with("user", $user)->with('success', 'Profil berhasil diedit');
}
function editStudent(Request $request, $nim)
{
    $user = Users::where('NIM', $nim)->first();
    $phone_number = $request->input("hp");
    $user->mahasiswa->no_HP = $phone_number;
    $user->mahasiswa->save(); // Use "->save"

    return redirect()->back()->with("user", $user)->with('success', 'Profil berhasil diedit');
}
function profile($nim)
{
    $user = Users::where('NIM', $nim)->first();
    $provinsi = Provinsi::all();
    $kota = KotaKab::all();
}

```

```

        if ($user) {
            return view("profile.mahasiswa")->with("user", $user)-
>with("provinsi", $provinsi)->with("kota", $kota);
        } else {
            return "User not found"; // You can change this message as
needed
        }
    }
function addIRS($nim)
{
    $user = Users::where('NIM', $nim)->first();
    $khs = KHS::all();

    if ($user) {
        return view("entry.irs")->with("user", $user)->with("khs",
$khs);
    } else {
        return "User not found"; // You can change this message as
needed
    }
}

function updatePage($nim)
{
    $user = Users::where('NIM', $nim)->first();
    $prov = Provinsi::all();
    $kotaKabData = DB::table('kota_kab')
        ->join('provinsi', 'kota_kab.kode_prov', '=', 'provinsi.kode_prov')
        ->select('kota_kab.kode_kota_kab',
'kota_kab.nama_kota_kab', 'provinsi.kode_prov')
        ->get()
        ->groupBy('kode_prov')
        ->map(function ($kotaKab) {
            return $kotaKab->pluck('nama_kota_kab',
'kode_kota_kab');
        });
    return view('updateAcc')->with("prov", $prov)-
>with("kotaKabData", $kotaKabData)->with("user", $user);
}
function updateAcc(Request $request, $nim)
{
    $mahasiswa = Mahasiswa::where('NIM', $nim)->first();
    $user = Users::where('NIM', $nim)->first();
    $pkl = new PKL();
    $skripsi = new Skripsi();
    $username = $request->input('username');
    $mahasiswa->email = $username . '@students.undip.ac.id';
    $mahasiswa->alamat = $request->input('alamat');
    $mahasiswa->no_HP = $request->input('noHP');
    $mahasiswa->jalur_masuk = $request->input('jalur_masuk');
    $mahasiswa->kode_kota_kab = $request->input('kotakab');
    $user->password = $request->input('password');
    $user->user = $request->input('username');
    $pkl->NIM = $nim;
    $pkl->status_pkl = 'belum ambil';
}

```

```

$skripsi->NIM = $nim;
$skripsi->status_skripsi = 'belum ambil';
$mahasiswa->save();
$user->save();
$pkl->save();
$skripsi->save();
return redirect("/dashboard/mahasiswa/{$user->NIM}");
}
public function confirmAddIRS(Request $request, $nim)
{
    $mahasiswa = Mahasiswa::where('NIM', $nim)->first();
    $irs = new IRS();
    $irs->NIM = $nim;
    $irs->smst_aktif = $request->input('smst_aktif');
    $irs->jumlah_sks = $request->input('jumlah_sks');
    $irs->status = 0;

    $filename = '';

    if ($request->hasFile('file')) {
        $file = $request->file('file');
        $filename = $file->getClientOriginalName(); // Mengisi
$filename dengan nama file yang diunggah

        // Simpan file sementara di direktori penyimpanan
$file->storeAs('berkas_irs', $filename);

        // Simpan path file dalam database
$irs->berkas_irs = $filename;
    }

    $this->copyFileIRS($filename);

    if ($request->hasFile('file-pkl')) {
        $pkl = PKL::where('NIM', $nim)->first();
        $file = $request->file('file-pkl');
        $filename = $file->getClientOriginalName(); // Mengisi
$filename dengan nama file yang diunggah

        // Simpan file sementara di direktori penyimpanan
$file->storeAs('berkas_pkl', $filename);

        // Simpan path file dalam database
$pkl->status_pkl = 'sedang ambil';
$pkl->berkas = $filename;
$pkl->nilai_pkl = $request->input('nilai_pkl');
$pkl->save();

        $this->copyFilePKL($filename);
    }

    if ($request->hasFile('file-skripsi')) {
        $skripsi = Skripsi::where('NIM', $nim)->first();
        $file = $request->file('file-skripsi');
        $filename = $file->getClientOriginalName(); // Mengisi
$filename dengan nama file yang diunggah
    }
}

```

```

        // Simpan file sementara di direktori penyimpanan
        $file->storeAs('berkas_skripsi', $filename);

        // Simpan path file dalam database
        $skripsi->status_skripsi = 'sedang ambil';
        $skripsi->berkas_skripsi = $filename;
        $skripsi->n nilai_skripsi = $request->input('nilai_skripsi');
        $skripsi->save();

        $this->copyFileSkripsi($filename);
    }

    $angkatan = substr($mahasiswa->angkatan, -2);
    $semester = $request->input('smst_aktif');
    $formattedSemester = str_pad($semester, 2, '0', STR_PAD_LEFT);
    $last5nim = substr($nim, -4);
    $irs->id_irs = 'IRS' . $angkatan . $last5nim . $formattedSemester;
    $irs->save();
    return redirect("/dashboard/mahasiswa/{$mahasiswa->NIM}");
}

function addKHS($nim)
{
    $user = Users::where('NIM', $nim)->first();
    $khs = KHS::all();

    if ($user) {
        return view("entry.khs")->with("user", $user)->with("khs",
    $khs);
    } else {
        return "User not found"; // You can change this message as
needed
    }
}

public function confirmAddKHS(Request $request, $nim)
{
    $mahasiswa = Mahasiswa::where('NIM', $nim)->first();
    $smt_terakhir = $request->input('smst_aktif') - 1;
    if ($smt_terakhir == 0) {
        $smt_terakhir = 1;
    }
    if ($smt_terakhir == 1) {
        $skks_terakhir = 0;
        $ipk_terakhir = 0;
    } else {
        $skks_terakhir = KHS::where('NIM', $nim)-
>where('smst_aktif', $smt_terakhir)->first()->SKS_kumulatif;
        $ipk_terakhir = KHS::where('NIM', $nim)-
>where('smst_aktif', $smt_terakhir)->first()->IP_Kumulatif;
    }
    $khs = new KHS();
    $khs->NIM = $nim;
    $khs->smt_aktif = $request->input('smst_aktif');
    $khs->SKS_semester = $request->input('jumlah_sks');
    $khs->SKS_kumulatif = $skks_terakhir + $khs->SKS_semester;
}

```

```

        $khs->IP_smt = $request->input('ips');
        $khs->IP_Kumulatif = number_format((($ipk_terakhir + $khs-
>IP_smt) / 2, 2, '.', '')));
        $khs->status = 0;

        $filename = '';

        if ($request->hasFile('file')) {
            $file = $request->file('file');
            $filename = $file->getClientOriginalName(); // Mengisi
$filename dengan nama file yang diunggah

            // Simpan file sementara di direktori penyimpanan
            $file->storeAs('berkas_khs', $filename);

            // Simpan path file dalam database
            $khs->berkas_khs = $filename;
        }

        $angkatan = substr($mahasiswa->angkatan, -2);
        $semester = $request->input('smst_aktif');
        $formattedSemester = str_pad($semester, 2, '0', STR_PAD_LEFT);
        $last5nim = substr($nim, -4);
        $khs->id_khs = 'KHS' . $angkatan . $last5nim . .
$formattedSemester;
        $khs->save();

        $this->copyFileKHS($filename);

        return redirect("/dashboard/mahasiswa/{$mahasiswa->NIM}");
    }
    function editIRS($nim, $id_irs)
    {
        $user = Users::where('NIM', $nim)->first();
        $irs = IRS::where('id_irs', $id_irs)->first();
        return view("edit.irs")->with("user", $user)->with("irs",
$irs);
    }

    function confirmEditIRS($nim, $id_irs)
    {
        $irs = IRS::where('id_irs', $id_irs)->first();
        $irs->jumlah_sks = request('jumlah_sks');

        // Check if a file is provided
        if (request()->hasFile('file')) {
            $irs->berkas_irs = request('file');
        }

        $irs->save();

        return redirect("/dashboard/mahasiswa/{$nim}");
    }

    function editKHS($nim, $id_khs)
    {

```

```
$user = Users::where('NIM', $nim)->first();
$khs = KHS::where('id_khs', $id_khs)->first();
return view("edit.khs")->with("user", $user)->with("khs",
$khs);
}
```

#### j. IRS.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Mahasiswa;
use App\Models\KHS;
use App\Models>Status;

class IRS extends Model
{
    public $timestamps = false;
    protected $table = 'irs';
    protected $primaryKey = 'id_irs';
    protected $fillable = ['id_irs', 'smst_aktif', 'jumlah_sks',
    'berkas_irs', 'NIM'];
    public $incrementing = false;

    public function mahasiswa(){
        return $this->belongsTo(Mahasiswa::class, 'NIM', 'NIM');
    }

    public function khs(){
        return $this->belongsTo(KHS::class, 'NIM', 'NIM');
    }

    public function toStatus(){
        return $this->belongsTo(Status::class, 'status', 'id');
    }
    use HasFactory;
}
```

#### k. KHS.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Mahasiswa;
use App\Models>Status;
```

```
class KHS extends Model
{
    public $timestamps = false;
    protected $table = 'khs';
    protected $primaryKey = 'id_khs';
    protected $fillable = ['id_khs', 'smt_aktif', 'SKS_semester',
'SKS_kumulatif', 'IP_smt', 'IP_Kumulatif', 'berkas_KHS', 'NIM',
'status'];
    public $incrementing = false;

    public function mahasiswa()
    {
        return $this->belongsTo(Mahasiswa::class, 'NIM', 'NIM');
    }

    public function toStatus(){
        return $this->belongsTo(Status::class, 'status', 'id');
    }
    use HasFactory;
}
```

## I. PKL.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Mahasiswa;

class PKL extends Model
{
    public $timestamps = false;
    protected $table = 'pk1';
    protected $primaryKey = 'NIM';
    protected $fillable = ['NIM', 'status_pk1', 'nilai_pk1',
'berkas'];
    public $incrementing = false;

    use HasFactory;

    public function mahasiswa(){
        return $this->belongsTo(Mahasiswa::class,'NIM','NIM');
    }
}
```

## m. Skripsi.php

```
<?php

namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Skripsi extends Model
{
    public $timestamps = false;
    protected $table = 'skripsi';
    protected $primaryKey = 'NIM';
    protected $fillable = ['NIM', 'nilai_skripsi', 'berkas_skripsi'];
    public $incrementing = false;
    use HasFactory;

    function mahasiswa(){
        return $this->belongsTo(Mahasiswa::class, 'NIM', 'NIM');
    }
}
```

#### n. Users.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Mahasiswa;
use App\Models\Dosen;
use App\Models\Operator;

class Users extends Model
{
    public $timestamps = false;
    protected $table = 'user';
    protected $primaryKey = 'user';
    protected $fillable = ['password', 'peran', 'user', 'NIM', 'NIP',
'ID_dep'];
    public $incrementing = false;

    public function mahasiswa(){
        return $this->belongsTo(Mahasiswa::class,'NIM', 'NIM');
    }

    public function dosen(){
        return $this->belongsTo(Dosen::class,'NIP', 'NIP');
    }

    public function operator(){
        return $this->belongsTo(Operator::class,'NIP', 'NIP');
    }

    public function departemen(){
        return $this->belongsTo(Departemen::class,'ID_dep',
'ID_departemen');
    }
}
```

```
    use HasFactory;
}
```

#### o. Provinsi.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Provinsi extends Model
{
    public $timestamps = false;
    protected $table = 'provinsi';
    protected $primaryKey = 'kode_prov';
    protected $fillable = ['kode_prov', 'nama_prov'];
    public $incrementing = false;

    public function kotaKab(){
        return $this->belongsTo(KotaKab::class, 'kode_prov',
'kode_prov');
    }
    use HasFactory;
}
```

#### p. KotaKab.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Provinsi;

class KotaKab extends Model
{
    public $timestamps = false;
    protected $table = 'kota_kab';
    protected $primaryKey = 'kode_kota_kab';
    protected $fillable = ['kode_kota_kab', 'nama_kota_kab',
'kode_prov'];
    public $incrementing = false;

    public function provinsi(){
        return $this->belongsTo(Provinsi::class, 'kode_prov',
'kode_prov');
    }
    use HasFactory;
```

{}

## 6.2 Testing

Metoda testing yang digunakan pada pengembangan perangkat lunak ini adalah Black Box Testing. Black Box Testing adalah merupakan metoda pengujian yang difokuskan pada kebutuhan fungsional perangkat lunak tanpa dibutuhkan pengetahuan struktur internal program. Black Box Testing dilakukan dengan cara menjalankan (eksekusi) semua SRS kemudian mengamati hasilnya apakah sudah sesuai atau belum dengan spesifikasi yang telah didefinisikan.

Tabel 6.1. Hasil Pengujian

No	SRS	Deskripsi SRS	Hasil Uji	Keterangan
1	SRS-XXX-001	Entry (upload) data mahasiswa baru menggunakan template yang sudah disediakan (diantaranya terdiri: NIM, nama, angkatan, dan status = AKTIF). Pada saat upload sekaligus dilakukan generate akun akses untuk setiap mahasiswa secara default	[OK]	-
2	SRS-XXX-002	Update data pribadi mahasiswa secara mandiri, untuk elemen data NIM dan angkatan tidak dapat dilakukan update. Pada saat pertama kali login harus melakukan pengisian semua	[OK]	-

No	SRS	Deskripsi SRS	Hasil Uji	Keterangan
		elemen data. Jika belum dilakukan pengisian semua elemen data mahasiswa maka belum dapat menggunakan fitur lain		
3	SRS-XXX-003	Entry pengambilan IRS per semester	[OK]	-
4	SRS-XXX-004	Entry prestasi akademik KHS per semester	[OK]	-
5	SRS-XXX-005	Entry progress PKL	[OK]	-
6	SRS-XXX-006	Entry progress skripsi	[OK]	-
7	SRS-XXX-007	Verifikasi progress studi meliputi IRS, KHS, PKL, dan Skripsi untuk diberikan approval	[OK]	-
8	SRS-XXX-008	Pencarian progress studi mahasiswa untuk setiap semester berdasarkan nim dan atau nama	[OK]	-
9	SRS-XXX-009	Profile dan dashboard operator prodi	[OK]	-
10	SRS-XXX-010	Profile dan dashboard operator mahasiswa	[OK]	-
11	SRS-XXX-011	Profile dan dashboard operator dosen wali	[OK]	-
12	SRS-XXX-012	Profile dan dashboard operator departemen	[OK]	-

No	SRS	Deskripsi SRS	Hasil Uji	Keterangan
13	SRS-XXX-013	Rekap mahasiswa yang sudah lulus dan belum PKL untuk semua angkatan	[OK]	-
14	SRS-XXX-014	List daftar mahasiswa yang sudah lulus dan belum PKL untuk semua Angkatan	[OK]	-
15	SRS-XXX-015	Rekap mahasiswa yang sudah lulus dan belum Skripsi untuk semua Angkatan	[OK]	-
16	SRS-XXX-016	List daftar mahasiswa yang sudah lulus dan belum Skripsi untuk semua Angkatan	[OK]	-
17	SRS-XXX-017	Rekap mahasiswa berdasarkan status yang disajikan per angkatan dan dilengkapi list daftar mahasiswanya	[OK]	-

## 7 Keterlibatan Anggota Kelompok

No	Nama Nim	Keterlibatan	Tanda Tangan
1	Arya Dheffan Shevchenko - 24060121140160	Membuat back-end berupa program fungsional supaya pengguna dapat berinteraksi dengan website, seperti menambah, mengedit, dan menghapus data, serta dokumentasi untuk sequence diagram	

No	Nama Nim	Keterlibatan	Tanda Tangan
2	Arynda Anna Salsabiela - 24060121120023	Membuat front-end seperti susunan halaman, rute halaman, serta tampilan halaman supaya menarik untuk dilihat, serta dokumentasi untuk robustness diagram	
3	Kharisma Andini Putri - 24060121140119	Menjalankan program (testing) untuk mengevaluasi website jika ada yang kurang tepat dan dokumentasi untuk use-case diagram	
4	Yesy Margharetta Munthe - 24060121120031	Membuat database sesuai requirement yang dibutuhkan dan dokumentasi untuk domain model, update domain model, dan class diagram	

## 8 Daftar Pustaka

---

*Roadmap\_ ICONIX Process.* (2007).

Yulianta, & Mursanto, P. (2008). PENGEMBANGAN APLIKASI WEB DENGAN ICONIX PROCESS DAN UML STUDI KASUS: SISTEM MANAJEMEN ISI. *Jurnal Sistem Informasi MTI-UI*, 4, 115–121.