

LAPORAN TUGAS AKHIR DIII

**REPOSITORY TUGAS AKHIR PADA APLIKASI  
*DIGITAL REPOSITORY (PNB REPOSITORY)***



POLITEKNIK NEGERI BALI

Oleh :

**I Putu Surya Jayantara**

NIM. 1915323025

**PROGRAM STUDI DIII MANAJEMEN  
INFORMATIKA  
JURUSAN TEKNIK ELEKTRO  
POLITEKNIK NEGERI BALI  
2022**

LAPORAN TUGAS AKHIR DIII

**REPOSITORY TUGAS AKHIR PADA APLIKASI  
*DIGITAL REPOSITORY (PNB REPOSITORY)***



POLITEKNIK NEGERI BALI

Oleh :

**I Putu Surya Jayantara**

NIM. 1915323025

**PROGRAM STUDI DIII MANAJEMEN  
INFORMATIKA  
JURUSAN TEKNIK ELEKTRO  
POLITEKNIK NEGERI BALI  
2022**

## LEMBAR PENGESAHAN TUGAS AKHIR

### REPOSITORY TUGAS AKHIR PADA APLIKASI *DIGITAL REPOSITORY* (PNB REPOSITORY)

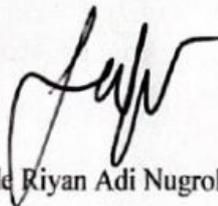
Oleh :

**I Putu Surya Jayantara**  
1915323025

Tugas Akhir ini Diajukan untuk Menyelesaikan  
Program Pendidikan Diploma III di  
Program Studi DIII Manajemen Informatika  
Jurusan Teknik Elektro - Politeknik Negeri Bali

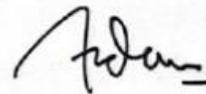
Disetujui Oleh :

Pembimbing I :



I Made Riyan Adi Nugroho, S.Si,  
M.T  
NIP. 198102202015042001

Pembimbing II :



Ir. I Wayan Raka Ardana, M.T.  
NIP. 196705021993031005

Disahkan Oleh  
Jurusan Teknik Elektro



Ir. I Wayan Raka Ardana, M.T.  
NIP. 196705021993031005

**LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI  
LAPORAN TUGAS AKHIR UNTUK  
KEPENTINGAN AKADEMIS**

Saya yang bertanda tangan di bawah ini :

Nama : I Putu Surya Jayantara  
NIM : 1915323025  
Program Studi : Manajemen Informatika  
Jurusan : Teknik Elektro  
Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Bali Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul: REPOSITORY TUGAS AKHIR PADA APLIKASI DIGITAL REPOSITORY (PNB REPOSITORY) beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Politeknik Negeri Bali berhak menyimpan, mengalihmedia atau mengalihformatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Bukit Jimbaran, 1 Agustus 2022

Yang menyatakan

  
( I r u t u Surya Jayantara)

## FORM PERNYATAAN PLAGIARISME

Saya yang bertandatangan dibawah ini:

Nama : I Putu Surya Jayantara  
NIM : 1915323025  
Program studi : Manajemen Informatika  
Jurusan : Teknik Elektro

Menyatakan dengan sesungguhnya bahwa Laporan Tugas Akhir berjudul REPOSITORY TUGAS AKHIR PADA APLIKASI DIGITAL REPOSITORY (PNB REPOSITORY) adalah betul-betul karya sendiri dan bukan menjiplak atau hasil karya orang lain. Hal-hal yang bukan karya saya, dalam Tugas Akhir tersebut diberi tanda citasi dan ditunjukkan dalam daftar pustaka. Apabila di kemudian hari terbukti pernyataan saya tidak benar, maka saya bersedia menerima sanksi akademik berupa pencabutan Tugas Akhir dan gelar yang saya peroleh dari Tugas Akhir tersebut.

Bukit Jimbaran, 1 Agustus 2022

Yang membuat pernyataan



I Putu Surya Jayantara

NIM. 1915323025

## KATA PENGANTAR

Puji Syukur penulis panjatkan kehadapan Tuhan yang Maha Esa, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan penyusunan Tugas Akhir ini yang berjudul “Repositori Tugas Akhir Pada Aplikasi Digital *Repository* (PNB Repository)” tepat pada waktunya.

Penyusunan Proyek akhir ini diajukan untuk memenuhi salah satu persyaratan kelulusan Program Pendidikan Diploma III pada Program Studi Manajemen Informati Jurusan Teknik Elektro Politeknik Negeri Bali.

Dalam penyusunan Tugas Akhir ini penulis banyak memperoleh bimbingan dan masukan dari berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu dalam kesempatan ini penulis mengucapkan terimakasih kepada:

1. Bapak I Nyoman Abdi, SE,M.eCom selaku Direktur Politeknik Negeri Bali.
2. Bapak Ir. I Wayan Raka Ardana, M.T. selaku Ketua Jurusan Teknik Elektro.
3. Bapak I Wayan Suasnawa, S.T., M.T. selaku Ketua Program Studi Manajemen Informatika Politeknik Negeri Bali.
4. Bapak I Made Riyan Adi Nugroho, S.SI, M. selaku Dosen Pembimbing Utama yang bersedia memberikan bimbingan selama proses penyusunan Tugas Akhir.
5. Bapak Ir. I Wayan Raka Ardana, M.T. selaku Dosen Pembimbing Pendamping yang bersedia memberikan bimbingan selama proses penyusunan Tugas Akhir.
6. Seluruh dosen pengajar Program Studi Manajemen Informatika Jurusan Teknik Elektro Politeknik Negeri Bali yang telah memberikan masukan serta saran untuk menyelesaikan laporan ini.
7. Ayah dan Ibu serta seluruh keluarga yang telah memberikan doa dan semangat untuk menyelesaikan Tugas Akhir ini.
8. Serta teman-teman dari kelas 6A MI yang telah memberikan dukungan moral kepada saya untuk menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Laporan Tugas Akhir ini masih jauh dari sempurna, oleh karena itu segala kritik dan saran yang bersifat membangun sangat penulis harapkan.

Akhir kata semoga Laporan Tugas Akhir ini dapat bermanfaat bagi mahasiswa Politeknik Negeri Bali khususnya, dan pembaca pada umumnya.

Bukit Jimbaran, 1 Agustus  
2022

I Putu Surya Jayantara  
1915323025

## ABSTRAK

Laporan tugas akhir ini memuat tentang tahapan dalam merancang dan membangun sistem repositori Tugas Akhir menggunakan aplikasi PNB *Repository*. Sistem ini bertujuan untuk mempermudah mahasiswa dalam mengumpulkan serta mencari informasi contoh tugas akhir dan membantu admin dalam mengelola data proposal tugas akhir . Sistem ini dibangun melalui beberapa tahap perancangan, yaitu: perancangan Flowmap Diagram, *Unified Modeling Language* dengan *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Sequence Diagram*, Perancangan *Entity Relationship Diagram* dan perancangan antarmuka. Sistem ini dibangun dengan memanfaatkan *Application Programming Interface* agar dapat digunakan atau diakses pada *website*, *desktop* atau *mobile*.

**Kata Kunci:** Sistem Repositori, Tugas Akhir , *Application Programming Interface*

## ABSTRACT

This final report contains all the stages in designing and building a repository system for the Thesis using the PNB Repository application. This system aims to make it easier for students to collect and find information on examples of Thesis and help admins in managing Thesis data. This system is built through several design stages, namely: Flowmap Diagram design, Unified Modeling Language with Use Case Diagrams, Activity Diagrams, Class Diagrams, Sequence Diagrams, Entity Relationship Diagram Design and interface design. This system is built by utilizing the Application Programming Interface so that it can be used or accessed on the website, desktop or mobile.

**Keyword:** Repository System, Thesis, Application Programming Interface

## DAFTAR ISI

|   |     |
|---|-----|
| LEMBAR PENGESAHAN TUGAS AKHIR .....   | i   |
| LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI LAPORAN TUGAS<br>AKHIR UNTUK KEPENTINGAN AKADEMIK ..... | i   |
| FORM PERNYATAAN PLAGIARISME .....   | ii  |
| KATA PENGANTAR .....  | iii |
| ABSTRAK .....   | vi  |
| DAFTAR ISI .....  | vii |
| DAFTAR TABEL .....  | x   |
| DAFTAR GAMBAR .....   | xii |
| BAB I PENDAHULUAN .....   | 1   |
| 1.1 Latar Belakang .....  | 1   |
| 1.2 Masalah dan Batasan Masalah .....   | 2   |
| 1.3 Tujuan .....  | 3   |
| 1.4 Manfaat .....   | 3   |
| 1.5 Sistematis Penulisan .....  | 3   |
| BAB II LANDASAN TEORI .....   | 5   |
| 2.1 Sistem Informasi .....  | 5   |
| 2.1.1. Pengertian Sistem .....  | 5   |
| 2.1.2. Pengertian Sistem Informasi .....  | 5   |
| 2.2 Website .....   | 5   |
| 2.3 Repositori .....  | 6   |
| 2.4 Android .....   | 6   |
| 2.5 Framework .....   | 6   |
| 2.6 Waterfall .....   | 6   |
| 2.7 Perangkat Lunak Pengembangan Sistem .....   | 8   |

|   |           |
|---|-----------|
| 2.7.1 PHP dan Laravel .....                                 | 8         |
| 2.7.2 HTML, CSS dan Javascript .....                        | 9         |
| 2.7.3 Flutter .....   | 9         |
| 2.7.4 MySQL.....  | 9         |
| 2.7.5 Microsoft Visual Studio Code.....                     | 10        |
| 2.7.6 XAMPP .....   | 10        |
| 2.8 Alat Bantu Pengembangan Sistem .....                    | 10        |
| 2.8.1. Flowmap.....   | 10        |
| 2.8.2. Class Diagram .....                                  | 11        |
| 2.8.3. Use Case Diagram.....                                | 12        |
| 2.8.4. Activity Diagram.....                                | 13        |
| 2.8.5. Sequence Diagram .....                               | 14        |
| 2.8.6. Basis Data .....                                     | 15        |
| <b>BAB III ANALISIS DAN PERANCANGAN .....</b>               | <b>17</b> |
| 3.1 Metodologi .....  | 17        |
| 3.1.1. Metode Pengembangan Sistem .....                     | 17        |
| 3.1.2. Metode Pengumpulan Data atau Analisis Kebutuhan..... | 18        |
| 3.2 Analisa Sistem.....                                     | 19        |
| 3.2.1 Analisa Sistem Berjalan .....                         | 19        |
| 3.2.2 Analisa Sistem Baru.....                              | 21        |
| 3.3 Perancangan Sistem.....                                 | 25        |
| 3.3.1 Entity Relationship Diagram (ERD) .....               | 25        |
| 3.3.2 Usecase Diagram.....                                  | 26        |
| 3.3.3 Use Case Glossary.....                                | 27        |
| 3.3.4 Actor Glossary .....                                  | 29        |
| 3.3.5 Use Case Spesification.....                           | 29        |

|   |                                       |            |
|---|---------------------------------------|------------|
| 3.3.6                                   | Activity diagram.....                 | 43         |
| 3.3.7                                   | Class Diagram .....                   | 57         |
| 3.3.8                                   | Struktur Tabel.....                   | 59         |
| 3.3.9                                   | Sequence Diagram .....                | 62         |
| 3.3.10                                  | Perancangan Antarmuka .....           | 73         |
| <b>BAB IV IMPLEMENTASI SISTEM .....</b> |                                       | <b>89</b>  |
| 4.1                                     | Perangkat Kebutuhan Implementasi..... | 89         |
| 4.1.1                                   | Spesifikasi Perangkat Keras.....      | 89         |
| 4.1.2                                   | Spesifikasi Perangkat Lunak.....      | 89         |
| 4.2                                     | Pengujian Sistem .....                | 90         |
| 4.2.1                                   | Pengujian Sistem Mobile .....         | 90         |
| 4.2.2                                   | Pengujian Sistem Website.....         | 102        |
| 4.4                                     | Tabel Pengujian.....                  | 115        |
| <b>BAB V KESIMPULAN DAN SARAN.....</b>  |                                       | <b>119</b> |
| 5.1                                     | Kesimpulan.....                       | 119        |
| 5.2                                     | Saran.....                            | 119        |
| <b>DAFTAR PUSTAKA .....</b>             |                                       | <b>120</b> |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2. 1 Simbol Flowmap .....   | 11 |
| Tabel 2. 2 Simbol Class Diagram .....   | 12 |
| Tabel 2. 3 Simbol Use Case Diagram .....  | 13 |
| Tabel 2. 4 Simbol Activity Diagram .....  | 14 |
| Tabel 2. 5 Simbol Sequance Diagram .....  | 15 |
| Tabel 3. 1 Use case glossary .....  | 28 |
| Tabel 3. 2 Actor Glossary .....   | 29 |
| Tabel 3. 3 Use case specification Login user .....  | 30 |
| Tabel 3. 4 Use case specification Registrasi mahasiswa pada sistem repositori tugas akhir .....           | 32 |
| Tabel 3. 5 Use case specification membuat repositori pada sistem repositori tugas akhir.....              | 33 |
| Tabel 3. 6 Use case specification Upload File Tugas Akhir pada sistem repositori tugas akhir .....        | 34 |
| Tabel 3. 7 Use case specification mengelola data tugas akhir pada sistem repositori tugas akhir.....      | 36 |
| Tabel 3. 8 Use case specification menambahkan tugas akhir favorit pada sistem repositori tugas akhir..... | 37 |
| Tabel 3. 9 Use case specification mengelola data Mahasiswa pada sistem repositori tugas akhir.....        | 38 |
| Tabel 3. 10 Use case specification mengelola data Repositori pada sistem repositori tugas akhir.....      | 40 |
| Tabel 3. 11 Use case specification mengelola data diri pada sistem repositori tugas akhir.....            | 41 |
| Tabel 3. 12 Use case specification logout pada sistem repositori tugas akhir .....                        | 42 |
| Tabel 3. 13 class diagram.....  | 59 |
| Tabel 3. 14 Tabel Users .....   | 59 |
| Tabel 3. 15 Tabel Users Details .....   | 60 |
| Tabel 3. 16 Tabel Studies.....  | 60 |
| Tabel 3. 17 Tabel Departements .....  | 61 |
| Tabel 3. 18 Tabel Thesis .....  | 61 |

|   |     |
|---|-----|
| Tabel 3. 19 Thesis Document.....            | 62  |
| Tabel 4. 1 Spesifikasi Perangkat Keras..... | 89  |
| Tabel 4. 2 Spesifikasi Perangkat Lunak..... | 89  |
| Tabel 4. 3 Tabel Hasil Pengujian .....      | 116 |

## **DAFTAR GAMBAR**

|  |    |
|--|----|
| Gambar 3. 1 Flowmap Sistem Berjalan .....                          | 19 |
| Gambar 3. 2 flowmap pengajuan tugas akhir.....                     | 21 |
| Gambar 3. 3 flowmap pengajuan dokumen tugas akhir.....             | 22 |
| Gambar 3. 4 flowmap pencarian tugas akhir .....                    | 24 |
| Gambar 3. 5 ERD pada sistem repositori.....                        | 25 |
| Gambar 3. 6 use case diagram sistem repositori tugas akhir .....   | 26 |
| Gambar 3. 7 Activity diagram login .....                           | 43 |
| Gambar 3. 8 Activity diagram registrasi .....                      | 44 |
| Gambar 3. 9 Activity diagram membuat repositori .....              | 46 |
| Gambar 3. 10 Activity diagram mengunggah tugas akhir .....         | 47 |
| Gambar 3. 11 Activity diagram mengelola data mahasiswa.....        | 49 |
| Gambar 3. 12 Activity diagram mengelola data repositori .....      | 50 |
| Gambar 3. 13 Activity diagram mengelola data tugas akhir.....      | 51 |
| Gambar 3. 14 Activity diagram melihat tugas akhir.....             | 53 |
| Gambar 3. 15 Activity diagram menambahkan tugas akhir favorit..... | 54 |
| Gambar 3. 16 Activity diagram mengelola data diri.....             | 55 |
| Gambar 3. 17 Activity diagram logout .....                         | 56 |
| Gambar 3. 18 class diagram .....                                   | 57 |
| Gambar 3. 19 Squence diagram login .....                           | 63 |
| Gambar 3. 20 Squence diagram registrasi.....                       | 64 |
| Gambar 3. 21 Squence diagram data diri .....                       | 65 |
| Gambar 3. 22 Squence diagram membuat repositori .....              | 66 |
| Gambar 3. 23 Squence diagram mengunggah tugas akhir.....           | 67 |
| Gambar 3. 24 Squence diagram mengelola data mahasiswa .....        | 68 |
| Gambar 3. 25 Squence diagram mengelola data tugas akhir .....      | 70 |
| Gambar 3. 26 Squence diagram melihat data tugas akhir.....         | 71 |
| Gambar 3. 27 Squence diagram menambah tugas akhir favorit .....    | 72 |
| Gambar 3. 28 Squence diagram logout favorit .....                  | 73 |
| Gambar 3. 29 Antarmuka Splashscreen Kedua.....                     | 74 |
| Gambar 3. 30 Antarmuka Splashscreen Ketiga .....                   | 74 |
| Gambar 3. 31 Antarmuka Splashscreen Keempat.....                   | 75 |

|  |    |
|--|----|
| Gambar 3. 32 Antarmuka Login .....                                   | 76 |
| Gambar 3. 33 Antarmuka Registrasi Akun Repositori .....              | 77 |
| Gambar 3. 34 Antarmuka Registrasi .....                              | 78 |
| Gambar 3. 35 Antarmuka Tampilan Dashboard .....                      | 79 |
| Gambar 3. 36 Antarmuka pencarian data tugas akhir pertama .....      | 80 |
| Gambar 3. 37 Antarmuka pencarian data tugas akhir kedua .....        | 80 |
| Gambar 3. 38 Antarmuka pencarian data tugas akhir ketiga .....       | 81 |
| Gambar 3. 39 Antarmuka membuat repositori.....                       | 81 |
| Gambar 3. 40 Antarmuka mengunggah tugas akhir.....                   | 82 |
| Gambar 3. 41 Antarmuka tampilan sub menu .....                       | 82 |
| Gambar 3. 42 Antarmuka tampilan data detail tugas akhir.....         | 83 |
| Gambar 3. 43 Antarmuka login.....                                    | 83 |
| Gambar 3. 44 Antarmuka form registrasi .....                         | 84 |
| Gambar 3. 45 Antarmuka tampilan landing page .....                   | 84 |
| Gambar 3. 46 Antarmuka tampilan memilih repositori .....             | 85 |
| Gambar 3. 47 Antarmuka membuat repositori tugas akhir .....          | 85 |
| Gambar 3. 48 Antarmuka pencarian tugas akhir.....                    | 86 |
| Gambar 3. 49 Antarmuka tampilan repositori tugas akhir.....          | 86 |
| Gambar 3. 50 Antarmuka tampilan data detail tugas akhir.....         | 87 |
| Gambar 3. 51 Antarmuka tampilan data detail dokumen tugas akhir..... | 87 |
| Gambar 3. 52 Antarmuka mengunggah tugas akhir.....                   | 88 |
| Gambar 3. 53 Antarmuka kelola data diri .....                        | 88 |
| Gambar 4. 1 Tampilan Splashscreen<br>.....                           | 9  |
| 0  |    |
| Gambar 4. 2 Halaman Login.....                                       | 90 |
| Gambar 4. 3 Halaman Memilih Akun Repositori .....                    | 91 |
| Gambar 4. 4 Halaman Registrasi .....                                 | 91 |
| Gambar 4. 5 Halaman Dashboard .....                                  | 92 |
| Gambar 4. 6 Halaman Repositori Tugas Akhir Tanpa Data .....          | 92 |
| Gambar 4. 7 Halaman Membuat Repositori Tugas Akhir .....             | 93 |
| Gambar 4. 8 Halaman Repositori Tugas Akhir Berisi Data.....          | 94 |

|  |     |
|--|-----|
| Gambar 4. 9 Halaman Melihat Detail Data Tugas Akhir.....                         | 94  |
| Gambar 4. 10 Halaman Mengedit Data Tugas Akhir .....                             | 95  |
| Gambar 4. 11 Halaman Kelola Dokumen Tugas Akhir.....                             | 96  |
| Gambar 4. 12 Halaman Tambah Dokumen Tugas Akhir .....                            | 96  |
| Gambar 4. 13Tampilan kelola dokumen tugas akhir .....                            | 97  |
| Gambar 4. 14 Halaman Edit Dokumen Tugas Akhir.....                               | 98  |
| Gambar 4. 15 Halaman Menghapus Dokumen Tugas Akhir.....                          | 98  |
| Gambar 4. 16 Halaman kelola dokumen tugas akhir .....                            | 99  |
| Gambar 4. 17 Halaman Mencari Data Tugas Akhir .....                              | 100 |
| Gambar 4. 18 Halaman pencarian tugas akhir .....                                 | 100 |
| Gambar 4. 19 Tampilan icon mencari data .....                                    | 101 |
| Gambar 4. 20 Halaman hasil pencarian tugas akhir.....                            | 102 |
| Gambar 4. 21 Halaman Login.....  | 102 |
| Gambar 4. 22 Halaman Memilih Akun Repositori .....                               | 103 |
| Gambar 4. 23 Halaman Registrasi .....  | 103 |
| Gambar 4. 24 Halaman Utama.....  | 104 |
| Gambar 4. 25 Halaman Pencarian Tugas Akhir.....                                  | 104 |
| Gambar 4. 26 Halaman Repositori.....   | 105 |
| Gambar 4. 27 Halaman Pilih Tipe Repositori.....                                  | 105 |
| Gambar 4. 28 Halaman Form Pembuatan Repositori .....                             | 106 |
| Gambar 4. 29 Halaman Repositori Data Detail Tugas Akhir .....                    | 106 |
| Gambar 4. 30 Halaman Repositori Edit Data Detail Tugas Akhir.....                | 107 |
| Gambar 4. 31 Halaman Repositori Tambah Dokumen Tugas Akhir.....                  | 107 |
| Gambar 4. 32 Halaman Repositori Data Dokumen Tugas Akhir .....                   | 108 |
| Gambar 4. 33 Halaman Repositori Edit Dan Hapus Data Dokumen Tugas Akhir<br>..... | 109 |
| Gambar 4. 34 Form edit data dokumen.....   | 109 |
| Gambar 4. 35 Halaman Data Diri .....   | 110 |
| Gambar 4. 36 Halaman Edit Data Diri.....   | 110 |
| Gambar 4. 37 Halaman Ganti Password .....  | 111 |
| Gambar 4. 38 Halaman Dashboard Admin.....  | 111 |
| Gambar 4. 39 Halaman Mengelola Data User .....                                   | 112 |

|  |     |
|--|-----|
| Gambar 4. 40 Halaman form tambah data user.....                  | 112 |
| Gambar 4. 41 Halaman form edit data user .....                   | 113 |
| Gambar 4. 42 Halaman Mengelola Data Repositori Tugas Akhir ..... | 113 |
| Gambar 4. 43 Tambah data repositori tugas akhir .....            | 114 |
| Gambar 4. 44 Tampilan form edit repositori tugas akhir.....      | 114 |
| Gambar 4. 45 Halaman Mengelola Data Dokumen Tugas Akhir.....     | 115 |
| Gambar 4. 46 Tampilan form edit data tugas akhir .....           | 115 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Tujuan pendidikan adalah menciptakan seseorang yang berkualitas dan berkarakter sehingga memiliki pandangan yang luas untuk menggapai cita-cita yang diharapkan. Selain itu, seseorang menempuh pendidikan diharapkan mampu beradaptasi secara cepat dan tepat diberbagai situasi. Pendidikan itu sendiri memotivasi diri kita untuk lebih baik dalam segala aspek kehidupan.

Pada jenjang pendidikan Perguruan Tinggi, diperlukan suatu pengujian untuk menentukan apakah peserta didik dapat memenuhi kompetensi yang telah ditetapkan oleh instansi untuk meraih kelulusan dalam jenjang pendidikan yang dipilih oleh peserta didik. Oleh karena itu, pada setiap jenjang pendidikan akan dilakukan proses pengujian pada akhir jenjang tersebut. Pengujian yang dimaksud adalah tugas akhir.

Pada pelaksanaannya, tugas akhir yang sudah selesai didokumentasikan ke dalam bentuk laporan tugas akhir. Laporan tersebut biasanya diletakkan di perpustakaan dengan tujuan untuk memberikan informasi atau referensi kepada pengunjung perpustakaan mengenai contoh tugas akhir yang sudah pernah dibuat. Namun terdapat permasalahan yang terjadi di dalamnya diantara lain adalah susah mencari referensi yang cocok dengan tugas akhir yang ingin dibuat karena tidak adanya suatu pengelompokkan antar jenis atau topik tugas akhir serta penyusunan yang kurang rapi. Kemudian tidak lengkapnya koleksi tugas akhir yang terpajang pada rak dikarenakan keterbatasan tempat dan juga aksesibilitas yang terbatas dikarenakan penerapan sistem manual dalam pendataan tugas akhir tersebut.

Berdasarkan masalah tersebut, maka dibutuhkan sebuah sistem informasi yang dapat memberikan informasi kepada mahasiswa ataupun masyarakat dan sebuah aplikasi yang dapat memanajemen dokumen. Salah satu bentuk sistem informasi yang dapat dibuat adalah Digital Repository. Repositori Digital adalah mekanisme untuk mengelola dan menyimpan konten digital yang dapat dipercaya, bisa

repository subjek, kelembagaan atau komersial. Dengan Begitu dengan adanya Digital repository ini, diharapkan menjadi sebuah media yang dapat memberikan akses kepada Mahasiswa maupun Umum untuk menjadi bahan referensi dalam penulisan maupun pengembangan kreativitas.

Pada pelaksanaannya, Sistem Informasi Digital Repositori ini akan menjadi fitur dari aplikasi yang bernama PNB REPOSITORY. Aplikasi PNB REPOSITORY merupakan Sistem Informasi penyimpanan buku atau berkas secara digital. Selain itu, aplikasi dapat diakses di website dan dapat diunduh dalam bentuk aplikasi mobile. Aplikasi ini memiliki fitur repositori berbagai macam berkas sesuai dengan jenisnya. Dalam fitur repositori Tugas Akhir menggunakan aplikasi berbasis website agar datanya dapat diakses secara mudah oleh mahasiswa yang tidak mengunduh aplikasinya dan pada fitur ini juga membantu admin kampus dalam mengelola data setiap proposal mahasiswa. Karena hal tersebutlah terbentuknya sebuah rancangan sistem informasi yang berjudul “REPOSITORY TUGAS AKHIR PADA APLIKASI DIGITAL REPOSITORY (PNB REPOSITORY)”. Aplikasi ini diharapkan dapat bermanfaat bagi kampus dalam mengelola mengelola hasil dari tugas akhir mahasiswa. Selain itu, aplikasi ini juga diharapkan dapat membantu mengurangi keramaian dan membantu mahasiswa dalam mencari data dalam memperlancar berlangsungnya pembuatan tugas akhir mahasiswa.

## **1.2 Masalah dan Batasan Masalah**

Berdasarkan latar belakang yang telah diuraikan di atas, dapat diambil rumusan masalah yang akan menjadi bahan pembahasan, yaitu :

- a. Bagaimana membuat menyelesaikan masalah penyimpanan tradisional ?.
- b. Bagaimana membuat sebuah media yang dapat mengelompokan data berdasarkan pencarian user?
- c. Bagaimana membangun sebuah sistem yang dapat diakses oleh user kapan saja dimana saja?

Sesuai dengan penelitian yang dilakukan, maka diberikan batasan masalah sebagai berikut:

- d. Sistem informasi yang dibuat akan digunakan oleh admin dan mahasiswa interal Politeknik Negeri Bali.

- e. Sistem informasi yang dibuat menghasilkan keluaran berupa data tugas akhir yang terdaftar dari sisi admin dan mahasiswa.
- f. Pencarian tugas akhir hanya berdasarkan filter judul

### **1.3 Tujuan**

Tujuan yang ingin dicapai dari penulisan dan pembuatan tugas akhir ini adalah untuk merancang Sistem informasi repositori digital, guna memberikan informasi tentang data tugas akhir serta mempermudah pengelolaan tugas akhir.

### **1.4 Manfaat**

Berdasarkan pada tujuan yang ingin dicapai, maka manfaat dari penelitian ini antara lain:

- a. Bagi penulis, penelitian ini diharapkan dapat menjadi sarana yang bermanfaat dalam pengimplementasian pengetahuan penulis tentang merancang serta membangun sebuah sistem informasi repositori.
- b. Bagi peneliti selanjutnya, hasil penelitian ini diharapkan dapat menjadi bahan referensi bagi peneliti lain dalam kajian yang sama.
- c. Bagi Politeknik Negeri Bali selaku admin atau pengelola, sistem yang dihasilkan diharapkan dapat menyelesaikan masalah pengelolaan tugas akhir, sehingga tugas akhir dapat didokumentasikan dengan lebih baik.
- d. Bagi mahasiswa Politeknik selaku pengguna, sistem yang dihasilkan diharapkan dapat mempermudah pengguna dalam mendapatkan informasi tentang data tugas akhir.

### **1.5 Sistematis Penulisan**

Tugas akhir ini terdiri dari lima bab yang saling berhubungan antara satu dengan yang lainnya dan disusun secara terperinci dan sistematis untuk memberikan gambaran dan mempermudah pembahasan tentang proyek akhir ini. Adapun sistematika penulisan proyek akhir ini adalah:

#### **BAB I PENDAHULUAN**

Memuat tentang latar belakang, rumusan masalah dan batasan masalah, tujuan, manfaat, serta sistematika penulisan proyek tugas akhir.

#### **BAB II LANDASAN TEORI**

Memuat tentang uraian buku-buku, bahan pustaka atau teori-teori yang berkaitan sebagai dasar dan landasan dalam menyelesaikan masalah yang dihadapi.

### **BAB III ANALISIS DAN PERANCANGAN**

Memuat tentang analisis sistem yang sedang berjalan di tempat penelitian. Disertakan juga Flowmap Diagram, Unified Modelling Language Diagram (UML) seperti Use Case Diagram, Class Diagram, Activity Diagram, Sequence Diagram, rancangan basis data, serta desain tampilan antarmuka sistem yang merupakan produk akhir dari penelitian ini.

### **BAB IV IMPLEMENTASI SISTEM**

Memuat tentang pengujian sistem informasi yang telah rampung dirancang dan dibangun. Disertakan pula hasil pengujian dan pengoperasian sistem informasi yang telah dilaksanakan. yang telah dilaksanakan.

### **BAB V KESIMPULAN DAN SARAN**

Memuat tentang uraian mengenai kesimpulan dan saran yang perlu disampaikan kepada pihak-pihak yang terkait dengan proyek tugas akhir ini.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem Informasi**

##### **2.1.1. Pengertian Sistem**

Seperti telah diketahui bersama bahwa telah banyak para ahli mendefinisikan mengenai sistem. Beberapa pengertian sistem yang diambil dalam penelitian kali ini adalah pengertian yang diberikan oleh Sutabri. Sutabri memberikan pengertian sistem sebagai sekelompok unsur-unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Dalam arti yang lain, sistem didefinisikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling interaksi, saling tergantung satu sama lain, dan terpadu. Pada intinya, sebuah sistem adalah sekumpulan entitas (*hardware, brainware, software*) yang saling berinteraksi, bekerjasama dan berkolaborasi untuk mencapai tujuan tertentu [1].

##### **2.1.2. Pengertian Sistem Informasi**

Pengertian sistem informasi adalah sistem yang menyediakan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerima. Secara lebih detil, sistem informasi dapat didefinisikan sebagai seperangkat entitas yang terdiri dari *hardware, software* dan *brainware* yang saling bekerjasama untuk menyediakan data yang diolah sehingga berguna dan bermanfaat bagi penerima data tersebut [1].

#### **2.2 Website**

*Website* merupakan istilah yang sudah tidak asing lagi dewasa ini. Secara umum *website* dapat diartikan sebagai sebuah halaman yang tersedia dalam sebuah *server* yang dapat diakses menggunakan jaringan internet dimana didalamnya berisi bermacam-macam informasi dari suatu konten tertentu. Sebuah halaman web yang tampil pada jejaring, umumnya dibuat melalui serangkaian *plain text* yang dikenal dengan istilah HTML (*Hyper Text Markup Language*) atau XHTML (*eXtensible HyperText Markup Language*) [1].

### **2.3 Repositori**

Repositori adalah tempat penyimpanan data besar berbasis digital. Apabila dikelola maka informasi tersebut pasti menjadi pengetahuan. Semua orang sepakat bahwa pengetahuan sangat penting bagi kebaikan. Data pada repositori sebagian tersusun secara sistematis dan sebagiannya lagi bersifat tidak beraturan. Agar data berfungsi sebagai informasi, maka data pada repositori yang tidak beraturan harus dikelola. Sebuah pengelolaan informasi secara sistematis maka akan menghasilkan pengetahuan yang sistematis. Tegastlah bahwa pengelolaan pengetahuan membutuhkan rancangan repositori dalam bentuk prototipe [2].

### **2.4 Android**

Menurut Ableson (2009), Android adalah sebuah aplikasi *platform mobile* yang *open source*. Android utamanya adalah produk Google, tetapi lebih tepatnya bagian dari *Open Handset Alliance*. *Open Handset Alliance* merupakan aliansi dari 30 organisasi yang berkomitmen untuk membawa sebuah perangkat seluler yang lebih baik dan terbuka untuk pasar. Android adalah *platform* terbuka pertama untuk perangkat *mobile*, Android adalah sebuah lingkungan perangkat lunak yang dibangun untuk perangkat berbasis *mobile*. Android termasuk kernel berbasis Linux, aplikasi *end-user*, dan *framework* aplikasi. *User application* dibangun berbasiskan bahasa pemrograman Java. Bahkan aplikasi yang dibangun juga berbasiskan Java [3].

### **2.5 Framework**

Pengertian *framework* menurut Naista (2017) adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* lebih singkat dan memudahkan dalam melakukan perbaikan [4].

### **2.6 Waterfall**

Pada pengembangan penulis menggunakan metode Air terjun (*WaterFall*) Menurut Rosa dan M. Shalahuddin Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*).

Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*).

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. Pembuatan Kode Program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis

spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru **Invalid source specified.**

## **2.7 Perangkat Lunak Pengembangan Sistem**

### **2.7.1 PHP dan Laravel**

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang digunakan untuk membuat website atau situs dinamis dan menangani rangkaian bahasa pemrograman antara client side scripting dan server side scripting. PHP merupakan singkatan dari Personal Home Page Hypertext Processor. PHP merupakan bahasa script yang ditempatkan dalam server dan diproses di server hasilnya dikirimkan ke klien, tempat pemakainya menggunakan browser. Seperti bahasa pemrograman yang lain, PHP memiliki kelebihan dan juga kelemahan. Adapun kelebihan dari PHP antara lain :

- a. PHP merupakan suatu bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. PHP dapat berjalan pada web server yang dirilis oleh Microsoft, juga pada Apache yang bersifat open source.
- c. Karena sifatnya yang open source, maka perubahan dan perkembangan interpreted pada PHP lebih cepat dan mudah, karena banyak milis-milis dan developer yang siap membantu pengembangannya.
- d. PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami

Laravel adalah framework berbasis PHP yang sifatnya open source, dan menggunakan konsep model – view – controller. Laravel berada di bawah lisensi MIT License dengan menggunakan Github sebagai tempat berbagi code. Pada Desember 2013, laravel sebagai framework terpopuler. Dalam penggunaannya laravel memiliki beberapa kekurangan salah satunya yaitu ukuran file yang cukup besar. Di dalam laravel terdapat file yang sifatnya default seperti vendor. File tersebut tidak boleh dihapus sembarangan sehingga ukuran website yang dibuat berukuran cukup besar. Selain itu, dibutuhkan koneksi internet untuk instalasi dan mengunduh library laravel, dan PHP [4].

### **2.7.2 HTML, CSS dan Javascript**

HTML (Hypertext Markup Language) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi seperti gambar, teks, video, dan suara pada penjelajah web internet, yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.

CSS (cascading style sheets) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web, sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Sama halnya seperti styles dalam aplikasi pengolahan kata, Microsoft Word, yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan Bahasa HTML dan XHTML. Invalid source specified.

Javascript adalah bahasa pemrograman tingkat tinggi untuk membuat website dinamis, dan JQuery adalah library atau pustaka dari javascript yang dirancang untuk memudahkan penerapan clientside scripting dan menyajikan sebuah paradigma baru pada penanganan event pada Javascript [5].

### **2.7.3 Flutter**

Flutter adalah kerangka kerja lintas platform yang bertujuan untuk mengembangkan aplikasi mobile berkinerja tinggi. Flutter dirilis secara publik pada tahun 2016 oleh Google. Aplikasi Flutter tidak hanya dapat berjalan di Android dan iOS, tetapi juga Fuschia, sistem operasi generasi berikutnya dari Google, memilih Flutter sebagai framework level aplikasinya [6].

### **2.7.4 MySQL**

MySQL adalah salah satu jenis database server yang menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Dengan menggunakan script PHP dan PERL Software database ini dapat berfungsi atau berjalan pada semua platform sistem operasi yang biasa digunakan (Windows, Linux, OS/2, berbagai varian Unix) [6].

### **2.7.5 Microsoft Visual Studio Code**

Visual Studio Code yaitu suatu pengelola kode yang berasal dari Microsoft bagi Windows, Linux dan MacOS dengan adanya visual studio code diharapkan dapat mempermudah pengguna dalam mengubah source code untuk pengembangan perangkat lunak. Visual Studio Code sudah didukung untuk melakukan debugging, git control dan kode refactoring. Editor ini juga berbentuk open source dan bisa diunggah secara gratis di bawah lisensi proprietary [7].

### **2.7.6 XAMPP**

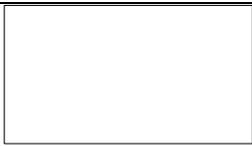
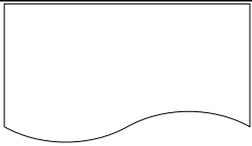
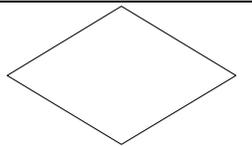
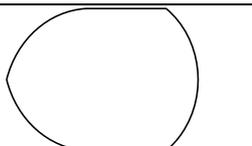
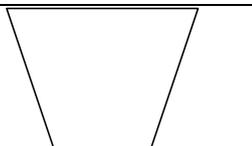
XAMPP adalah sebuah web server yang digunakan untuk melakukan simulasi membuat server web lokal sebelum diupload ke dalam server yang sesungguhnya. XAMPP dapat meliputi sebuah aplikasi server yaitu Apache, database yaitu MySQL dan bahasa skrip yaitu PHP. XAMPP bersifat lintas platform, artinya dapat digunakan dalam berbagai sistem operasi yaitu Windows, Linux dan Mac. Diharapkan dengan adanya XAMPP membuat pengembangan dan pengujian aplikasi semakin mudah karena ukuran dari XAMPP ini kecil dan mudah dibawa kemana saja [7].

## **2.8 Alat Bantu Pengembangan Sistem**

### **2.8.1. Flowmap**

Sebelum mengetahui pengertian flowmap akan terlebih dahulu memberikan definisi flowchart. Flowchart adalah bagan (chart) yang menunjukkan alir (flow) didalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan dokumentasi. flowmap adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Flowmap merupakan campuran peta dan flowchart, yang menunjukkan pergerakan benda dari suatu lokasi ke lokasi lain [8].

**Tabel 2. 1 Simbol *Flowmap***

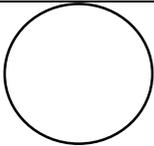
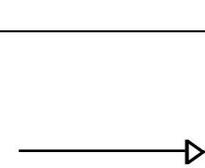
| Simbol  | Nama              | Keterangan  |
|---|-------------------|---|
|    | Proses            | Menunjukkan proses yang dilakukan pada <i>flowmap</i> .           |
|    | Dokumen           | Menunjukkan proses <i>input</i> dan <i>output</i> dokumen.        |
|    | <i>Database</i>   | Menunjukkan proses penyimpanan data ke dalam <i>database</i> .    |
|   | Kondisi           | Menunjukkan pemilihan kondisi yang ditandai dengan ya atau tidak. |
|  | Terminator        | Menandai awal dan akhir dari proses <i>flowmap</i> .              |
|  | Garis alir        | Menunjukkan arah dari alur <i>flowmap</i> .                       |
|  | Display / Monitor | Memunjukkan Tampilan yang akan dilihat oleh user                  |
|  | Manual Operation  | Menunjukkan Proses manual yang terjadi didalam Sistem             |

### 2.8.2. Class Diagram

Class diagram atau diagram kelas merupakan suatu diagram yang digunakan untuk menampilkan kelas-kelas berupa pake-paket untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya. Namun, pada Class diagram desain modelnya

dibagi menjadi 2 bagian. Class diagram yang pertama merupakan penjabaran dari domain model yang merupakan abstraksi dari basis data. Class diagram yang kedua merupakan bagian dari modul program MVC pattern (Model View Controller), di mana terdapat class boundary sebagai class interface, class control sebagai tempat ditemukannya algoritma, dan class entity sebagai tabel dalam basis data dan query program [9].

**Tabel 2. 2 Simbol *Class Diagram***

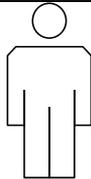
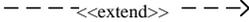
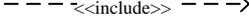
| Simbol  | Nama                        | Keterangan  |
|---|-----------------------------|---|
|    | Kelas                       | Kelas pada struktur sistem.   |
|   | <i>Interface</i>            | Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek.        |
|  | <i>Association</i>          | Relasi antar kelas dengan makna umum.   |
|  | <i>Directed association</i> | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |
|  | Generalisasi                | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).        |
|  | Dependensi                  | Relasi antar kelas dengan makna kebergantungan antar kelas.                     |
|  | Agregasi                    | Relasi antar kelas dengan makna semua-bagian ( <i>wholepart</i> ).              |

### 2.8.3. Use Case Diagram

Use Case Diagram adalah satu jenis dari diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. Use

Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. Use Case merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada Use Case [9].

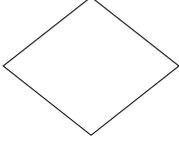
**Tabel 2. 3 Simbol Use Case Diagram**

| Simbol  | Nama            | Keterangan  |
|---|-----------------|---|
|    | Aktor           | Orang atau <i>user</i> yang terlibat dengan proses sistem.  |
|   | <i>Use Case</i> | Abstraksi dan interaksi antara sistem dengan aktor.   |
|  | Asosiasi        | Penghubung antara actor dengan <i>use case</i> .  |
|  | <i>Extend</i>   | Suatu <i>use case</i> merupakan fungsionalitas dari <i>use case</i> lain apabila suatu kondisi terpenuhi. |
|  | <i>Include</i>  | Suatu <i>use case</i> sepenuhnya merupakan fungsionalitas dari suatu <i>use case</i> .                    |

#### 2.8.4. Activity Diagram

Activity diagram atau dalam bahasa Indonesia berarti diagram aktivitas, merupakan sebuah diagram yang dapat memodelkan berbagai proses yang terjadi pada sistem. Seperti layaknya runtutan proses berjalannya suatu sistem dan digambarkan secara vertikal.

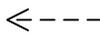
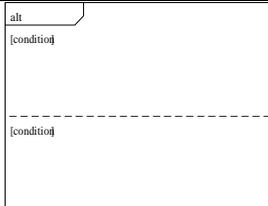
**Tabel 2. 4 Simbol *Activity Diagram***

| <b>Simbol</b>  | <b>Nama</b>  | <b>Keterangan</b>   |
|--|--------------|---|
|   | Status Awal  | Status awal dari activity diagram untuk mengawali sebuah penggambaran fitur.    |
|   | Status Akhir | Status akhir setelah suatu fitur pada sistem setelah berjalan.                  |
|   | Aktivitas    | Aktivitas pada suatu sistem atau fitur yang biasanya diawali dengan kata kerja. |
|  | Percabangan  | Menunjukkan pemilihan kondisi yang ditandai dengan ya atau tidak.               |

### **2.8.5. Sequence Diagram**

*Sequence diagram* merupakan diagram yang menjelaskan interaksi objek berdasarkan urutan waktu. *Sequence* dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu, seperti yang tertera pada *Use Case diagram* [9].

**Tabel 2. 5 Simbol *Sequence Diagram***

| <b>Simbol</b>   | <b>Nama</b>           | <b>Keterangan</b>  |
|---|-----------------------|--|
|    | <i>Message</i>        | <i>Object</i> mengirim pesan ke suatu <i>object</i> lain.  |
|    | <i>Return Message</i> | <i>Object</i> membalas Kembali pesan yang sebelumnya dikirim oleh <i>object</i> lain.                |
|    | <i>Condition</i>      | Menandakan kondisi dalam suatu <i>sequence diagram</i> .   |
|   | <i>Lifeline</i>       | Menandakan kehidupan <i>object</i> selama <i>sequence</i> berjalan.                                  |
|  | <i>Object</i>         | Instansi dari <i>model, view, controller</i> yang berpartisipasi di sebuah <i>sequence diagram</i> . |
|  | Aktivasi              | Menandakan bahwa sebuah <i>object</i> sedang   |

### 2.8.6. Basis Data

Menurut Adi Nugroho dalam jurnal Fery Wongso. basis data sebagai kumpulan terorganisasi dari data yang berhubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi serta dipanggil oleh pengguna. Terminologi hubungan berarti data mendeskripsikan domain (ranah) tertentu sehingga pengguna mudah untuk mendapatkan jawaban atas pertanyaan yang diajukan ke basis data tersebut. Sedangkan pengertian sistem basis data adalah sebagai koleksi dari data yang

terorganisasi sedemikian rupa sehingga data mudah disimpan dan dimanipulasi (diperbarui, dicari, diolah dengan perhitungan tertentu, serta dihapus) [10].

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Metodologi**

##### **3.1.1. Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan untuk membangun sistem repositori tugas akhir ilmiah dosen menggunakan metode pengembangan waterfall. Pada metode waterfall ini menekankan tahapan yang berurutan dan sistematis. Dalam metode ini setiap kemajuan dalam pengembangan sistem ini menyerupai air terjun, diaman pengembangannya terus mengalir ke bawah melewati fase-fase yang terjadi seperti perencanaan, pemodelan, pembangunan, dan pengujian. Berikut ini tahapan-tahapan yang akan dilewati dalam pengembangan sistem repositori tugas akhir ilmiah dosen menggunakan aplikasi Warebook.

##### **A. Analisis Kebutuhan**

Pada tahapan ini dilakukan analisi terhadap kebutuhan sistem yang akan dibangun. Pengumpulan data pada tahap ini dilakukan dengan pencarian data-data tugas akhir milik mahasiswa sebelumnya serta melalui studi literatur. Data yang didapatkan akan menjaddi acuan dalam pengembangan sistem sehingga dapat dilakukan ke tahapan selanjutnya

##### **B. Desain Sistem**

Setelah tahapan sebelumnya sudah terlewati, maka dilakukan tahapan mendesain sistem. Pada tahapan ini dilakukan pengembangan berupa analisis sistem, analisis data, perancangan desain sistem, dan perancangan desain antarmuka. Apabila tahapan ini sudah terselesaikan dapat dilakukan ke tahapan selanjutnya.

##### **C. Pengkodean dan Testing**

Setelah tahapan tahapan sebelumnya telah dilakukan, pada tahapan ini merupakan tahapan untuk melakukan penerjemahan dari desain sistem ke kode kode program yang telah dirancang sebelumnya. Kemudian dilanjutkan ke tahapan pengujian untuk mengetahui dan memperbaiki kesalahan-kesalahan sistem yang telah dibangun.

### **3.1.2. Metode Pengumpulan Data atau Analisis Kebutuhan**

Dalam pengerjaan tugas akhir ini, penulis akan menggunakan beberapa metode penelitian sebagai sarana untuk memahami objek permasalahan yaitu sebagai berikut:

1. Studi Pustaka

Studi pustaka dilakukan dengan mencari literatur menggunakan buku, internet dan sumber lainnya yang berhubungan dengan materi penulisan tugas akhir.

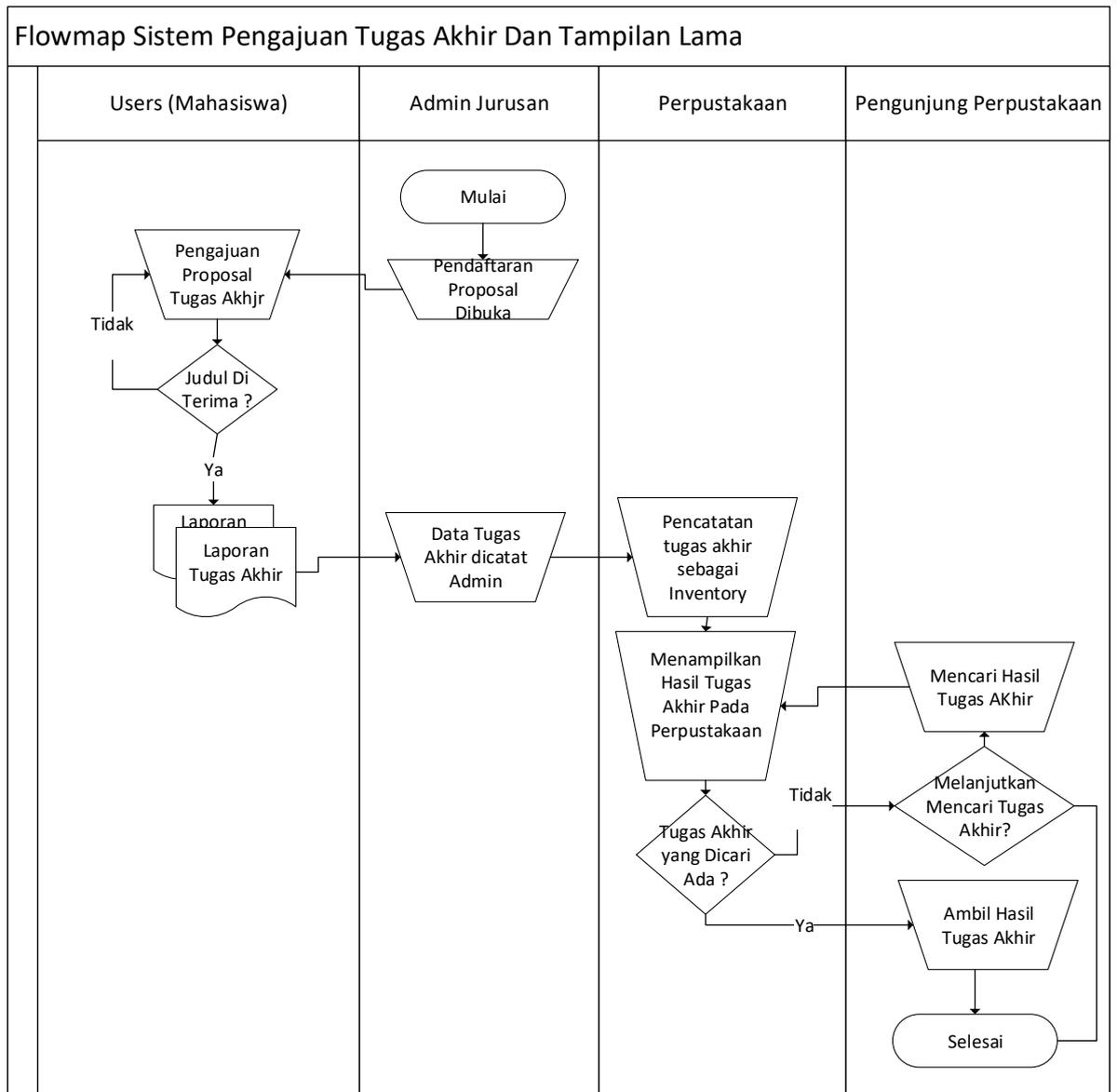
2. Observasi

Tujuan dari dilakukannya proses observasi atau pengamatan ini adalah untuk mengetahui prosedur pengumpulan data tugas akhir yang telah dibuat oleh dosen Politeknik Negeri Bali, dengan cara mengamati secara langsung proses pengumpulan tugas akhir.

### 3.2 Analisa Sistem

#### 3.2.1 Analisa Sistem Berjalan

Berdasarkan hasil observasi (pengamatan) yang telah dilakukan, adapun proses kegiatan tugas akhir yang terjadi saat ini pada Politeknik Negeri Bali dapat dilihat pada Gambar 3.1 dibawah ini.



**Gambar 3. 1 Flowmap Sistem Berjalan**

Sumber: Hasil Perancangan

Keterangan Flowmap Sistem Berjalan pada Gambar 3.1:

- Admin Jurusan Membuka pendaftaran tugas akhir berdasarkan jurusan masing masing.

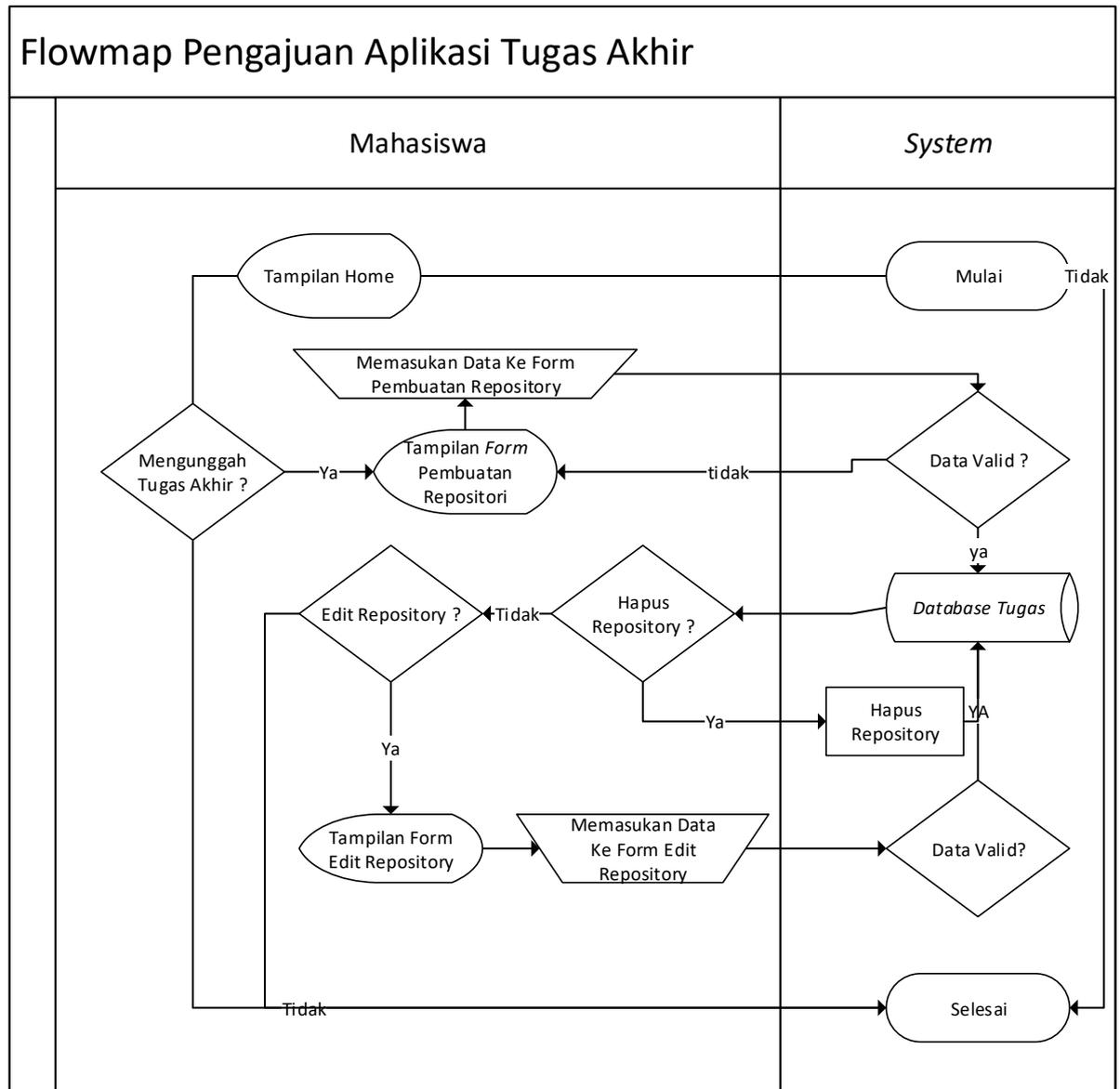
- b. Mahasiswa melakukan Pengajuan tugas akhir berupa judul, dosen pembimbing serta proposal kepada admin.
- c. Jika judul diterima , Mahasiswa dapat melanjutkan dengan proses Pengumpulan tugas akhir.
- d. Ketika menerima tugas akhir, petugas kampus akan melakukan registrasi dengan cara mencatat manual tugas akhir
- e. Jika Telah dicatat, tugas akhir akan diletakan pada rak berdasarkan jurusan masing – masing.
- f. Untuk Mahasiswa Baru , Dapat mencari tugas akhir pada rak Tugas akhir yang terdapat pada perpustakaan.
- g. Pada sistem lama, terdapat kekurangan pada saat pencarian tugas akhir yang manual yaitu sulitnya mencari judul yang tepat sesuai keinginan pengunjung perpustakaan
- h. Pada Saat diberlakukan sistem *Work From Home* , pengumpulan data secara luring menjadi tidak efisien

Pada sisi pegawai perpustakaan, Sangat rentan kehilangan data fisik diakibatkan tidak teraturnya pengunjung perpustakaan.

### 3.2.2 Analisa Sistem Baru

Setelah melakukan pengamatan (observasi) di Politeknik Negeri Bali, berikut ini merupakan beberapa gambaran flowmap terkait sistem yang nantinya akan dibuat.

#### a. Flowmap Pengajuan Tugas Akhir



**Gambar 3. 2 flowmap pengajuan tugas akhir**

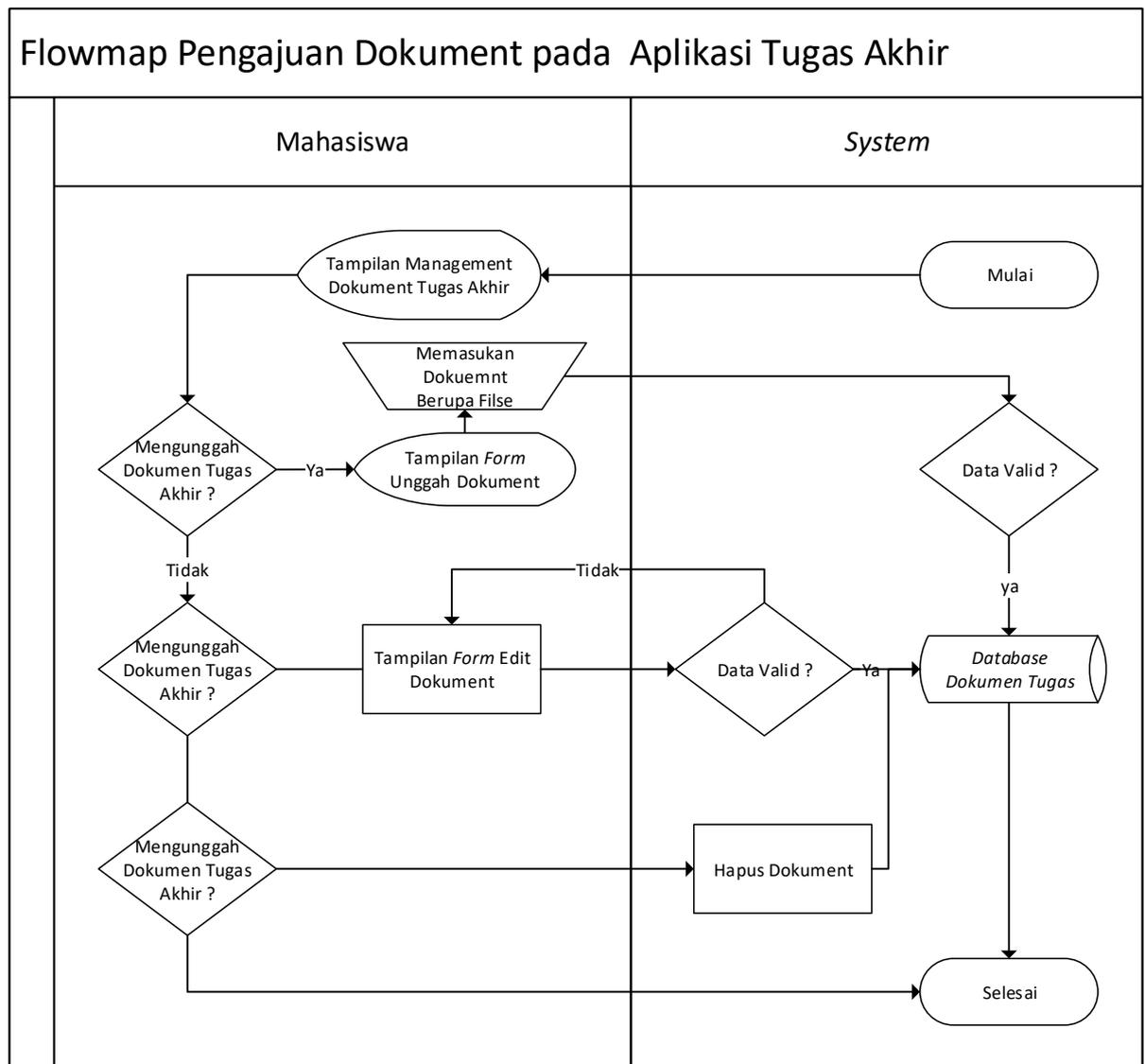
Sumber: Hasil Perancangan

Keterangan *flowmap* pengajuan tugas akhir pada Gambar 3.2.

1. Pertama, mahasiswa masuk ke dalam sistem tampilan *home*. Mahasiswa yang ingin mengunggah tugas akhir akan diarahkan pada tampilan *form* pembuatan

- repositori. Jika datanya valid, maka akan disimpan pada *database* dan apabila tidak valid, maka akan dikembalikan pada tampilan *form* pembuatan repositori.
2. Mahasiswa yang sudah membuat repositori tugas akhir dapat melakukan aksi hapus repositori atau edit repositori. Apabila mahasiswa memilih hapus repositori, maka sistem akan menghapus datanya yang ada pada *database*.
  3. Apabila mahasiswa memilih edit repositori, maka akan muncul tampilan *form* edit repositori. Jika datanya valid, maka akan disimpan pada *database* dan apabila tidak valid, maka akan dikembalikan pada tampilan *form* edit repositori.

**b. Flowmap Pengajuan Dokumen Tugas Akhir**



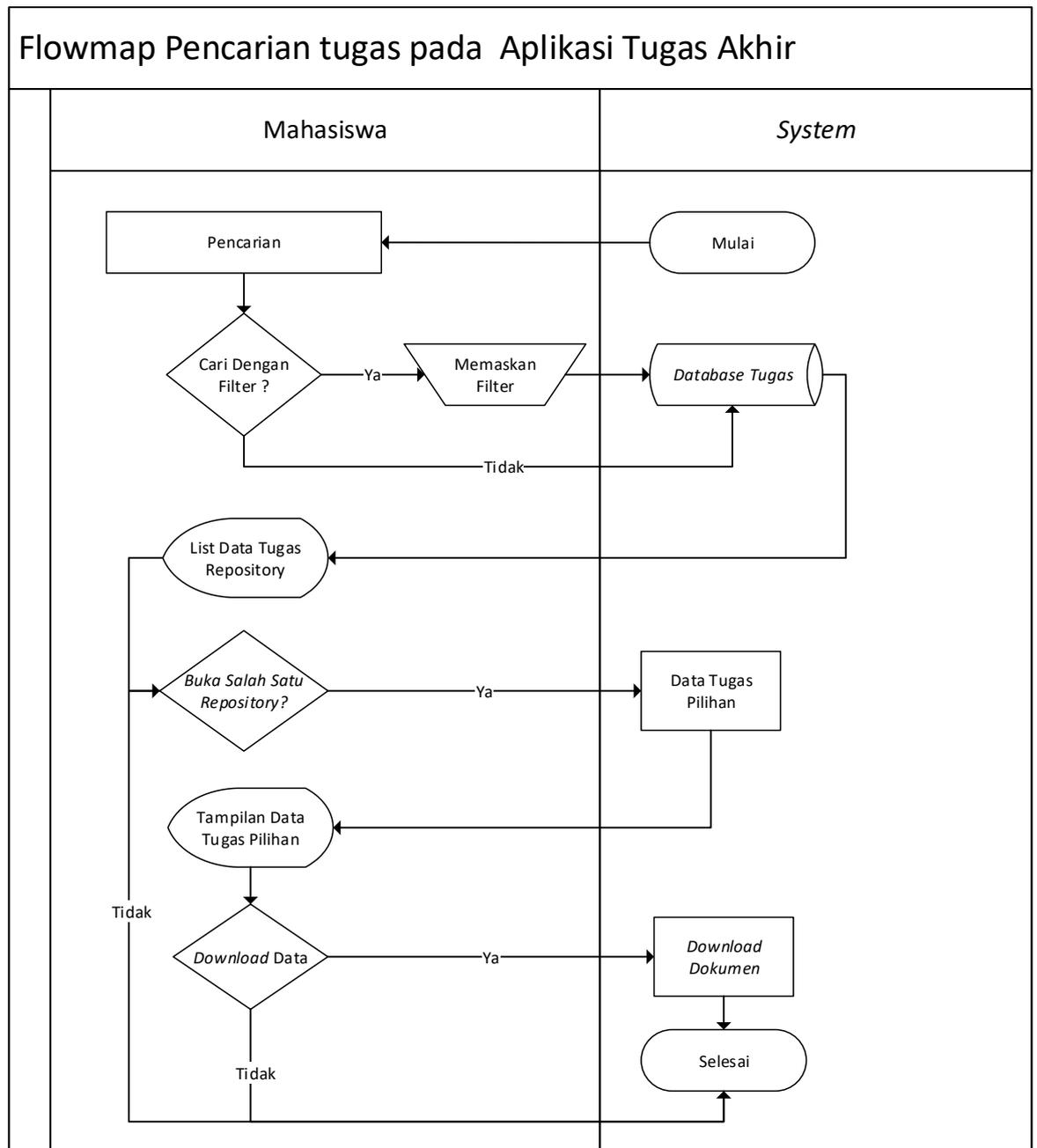
**Gambar 3. 3 flowmap pengajuan dokumen tugas akhir**

Sumber: Hasil Perancangan

Keterangan *flowmap* pengajuan dokumen tugas akhir pada Gambar 3.3.

1. Pertama, mahasiswa masuk ke dalam sistem tampilan *management* dokumen tugas akhir. Mahasiswa yang ingin mengunggah dokumen tugas akhir akan diarahkan pada tampilan *form* unggah dokumen. Jika datanya valid, maka akan disimpan pada *database* dan apabila tidak valid, maka akan dikembalikan pada tampilan *form* unggah dokumen.
2. Mahasiswa yang sudah mengunggah dokumen tugas akhir dapat melakukan aksi hapus dokumen atau edit dokumen. Apabila mahasiswa memilih edit dokumen, maka akan muncul tampilan *form* edit dokumen. Jika datanya valid, maka akan disimpan pada *database* dan apabila tidak valid, maka akan dikembalikan pada tampilan *form* edit dokumen.
3. Apabila mahasiswa memilih hapus dokumen, maka sistem akan menghapus datanya yang ada pada *database*.

c. Flowmap Pencarian Tugas Akhir



**Gambar 3. 4 flowmap pencarian tugas akhir**

Sumber: Hasil Perancangan

Keterangan *flowmap* pencarian tugas akhir pada Gambar 3.4.

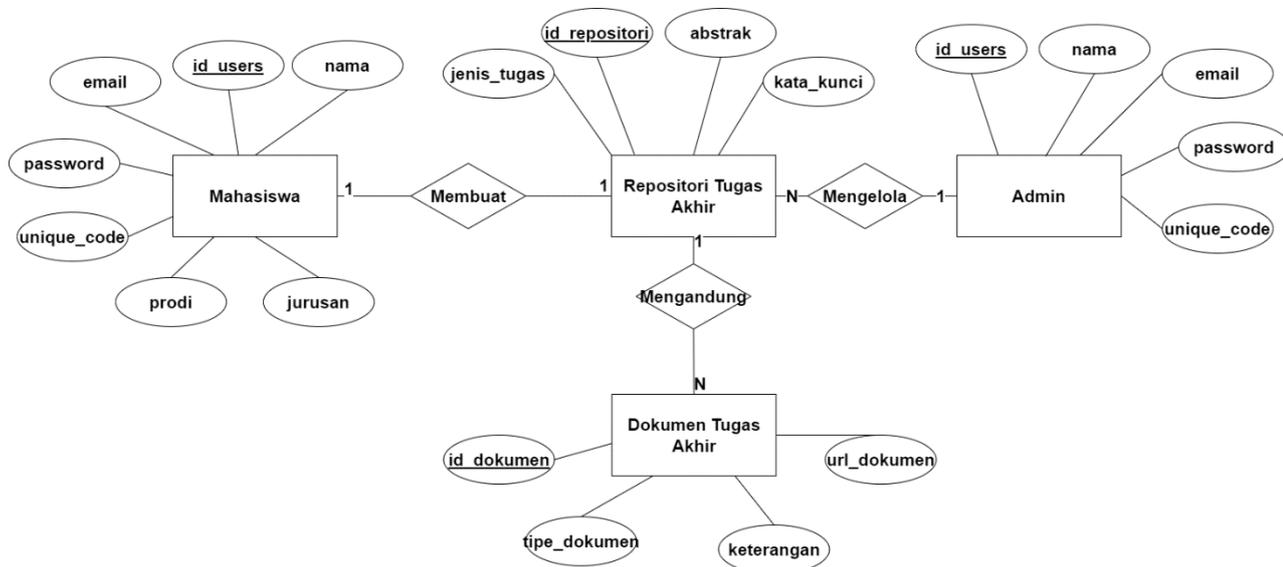
1. Pertama, mahasiswa mengakses tampilan pencarian dan memasukkan *filter* pencarian, maka sistem akan menampilkan data sesuai dengan *filter*. Apabila pencarian tidak menggunakan *filter*, maka sistem akan menampilkan semua data tugas akhir.

- Mahasiswa dapat melihat data detail tugas akhir dengan memilih salah satu *list* data tugas akhir dan sistem akan menampilkan data sesuai dengan pilihan mahasiswa. Jika mahasiswa ingin mengunduh dokumennya maka sistem akan memproses *download* dokumen.

### 3.3 Perancangan Sistem

#### 3.3.1 Entity Relationship Diagram (ERD)

Salah satu pemodelan yang digunakan dalam merancang basis data dalam pengembangan sistem repositori tugas akhir ini menggunakan Entity Relationship Diagram (ERD). ERD digunakan untuk menggambarkan relasi antar setiap entitas pada sistem yang akan dibuat nantinya.



**Gambar 3. 5 ERD pada sistem repositori**

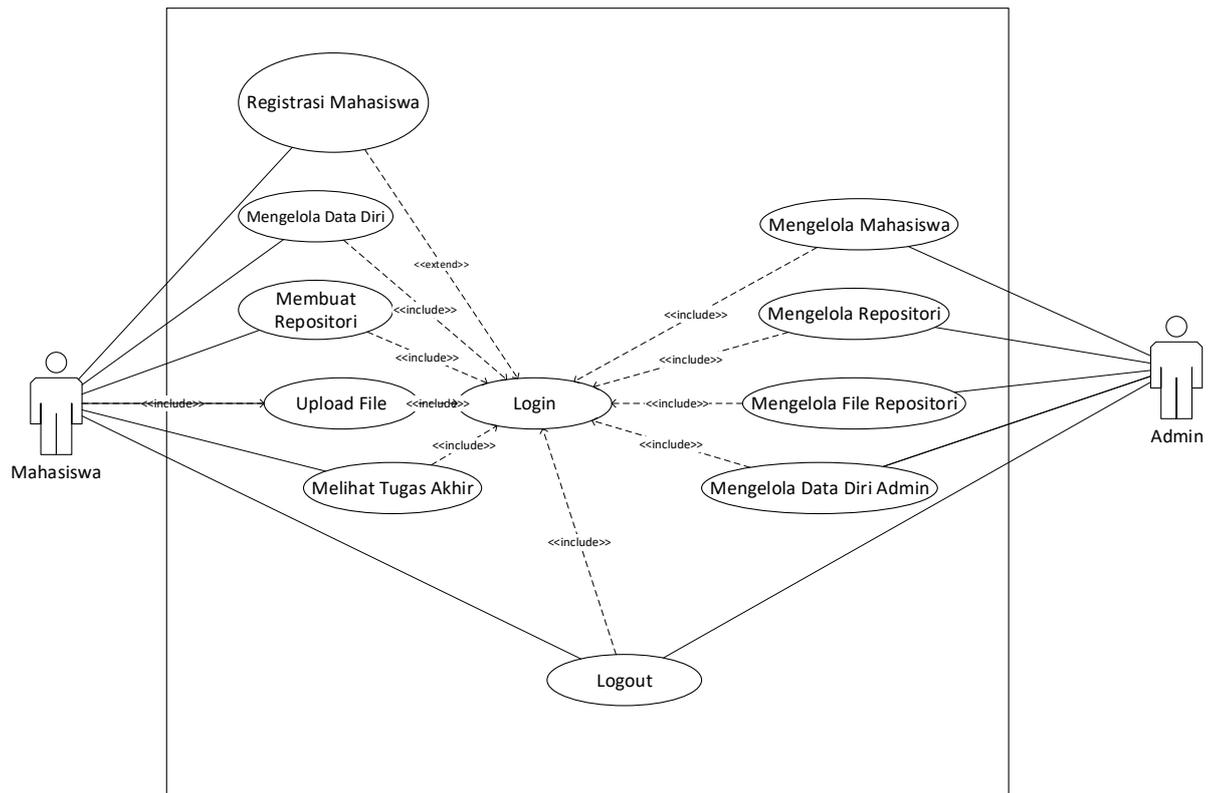
Sumber: Hasil Perancangan

ERD pada Gambar 3.5 terdiri dari 4 entitas yaitu mahasiswa, admin, dokument tugas akhir dan repositori tugas akhir. Antara entitas mahasiswa dan repositori tugas akhir terjadi relasi membuat data dengan kardinalitas *one to one* karena seorang mahasiswa hanya dapat memiliki satu repositori dan dalam repositori tugas akhir hanya dimiliki oleh satu mahasiswa. Antara entitas repositori tugas akhir dan dokumen tugas akhir terjadi relasi mangandung data dengan kardinalitas *one to many* karena repositori tugas akhir dapat mengandung lebih dari satu dokument tugas akhir dan dalam dokumen tugas akhir hanya memiliki satu repositori tugas akhir. Antara entitas admin dan repositori tugas akhir terjadi relasi mengelola data

dengan kardinalitas *one to many* karena admin dapat mengelola lebih dari satu repositori tugas akhir dan dalam repositori tugas akhir hanya memiliki satu admin.

### 3.3.2 Usecase Diagram

Perancangan use case diagram dapat menunjukkan hubungan interaksi antara aktor dengan sistem. Use case diagram pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.6.



**Gambar 3. 6 use case diagram sistem repositori tugas akhir**

Sumber: Hasil Perancangan

Keterangan *use case diagram* sistem repositori tugas akhir:

- Sistem memiliki 2 aktor, yaitu mahasiswa dan admin.
- Aktor mahasiswa dapat menjalankan *use case* registrasi mahasiswa, *login* mahasiswa, mengelola data diri, membuat repositori, *upload file*, melihat tugas akhir, menambahkan tugas akhir favorit, dan *logout* mahasiswa.
- Aktor admin dapat menjalankan *use case* *login* admin, mengelola mahasiswa, mengelola repositori, mengelola *file* repositori, mengelola admin, dan *logout* admin.

### 3.3.3 Use Case Glossary

*Use case glossary* merupakan deskripsi singkat dari setiap *use case* dan actor yang terkatik dalam *use case* yang sudah digambarkan pada *use case diagram*. *Use case glossary* sistem repositori tugas akhir dapat dilihat dalam tabel berikut ini.

| No | Use Case                          | Deskripsi   | Aktor               |
|----|-----------------------------------|---|---------------------|
| 1  | <i>Login</i>                      | Use case ini berjalan ketika pengguna mencoba untuk masuk ke dalam sistem. Aktor akan diminta untuk mengisi email dan password yang sebelumnya sudah didaftarkan pada sistem.   | Admin dan Mahasiswa |
| 2  | Registrasi                        | <i>Use case</i> ini berjalan ketika pengguna baru ingin membuat sebuah akun agar dapat mengakses sistem repositori tugas akhir. Aktor diminta untuk mengisi nama, email, password, jurusan, program studi, dan nomor induk mahasiswa. | Mahasiswa           |
| 3  | Membuat Repositori                | <i>Use case</i> ini berjalan ketika mahasiswa ingin membuat sebuah repositori dalam sistem. Data yang dilengkapi berupa nama repositori, deskripsi repositori, dan jenis dari repositori yang ingin dibuat.                           | Mahasiswa           |
| 4  | <i>Upload File</i><br>Tugas Akhir | <i>Use Case</i> ini berjalan ketika dosen ingin mengupload sebuah tugas akhir pada repositori yang dibuat sebelumnya. Mahasiswa harus melengkapi data yang diperlukan   | Mahasiswa           |

|   |                            |  |                     |
|---|----------------------------|--|---------------------|
|   |                            | dalam mengunggah file tugas akhir.   |                     |
| 5 | Mengelola Data Tugas Akhir | <i>Use case</i> ini berjalan ketika admin mengelola data repositori yang telah diupload oleh mahasiswa. Admin dapat mengubah dan menghapus data tugas akhir jika diperlukan. | Admin               |
| 6 | Mengelola Data Mahasiswa   | <i>Use case</i> ini berjalan ketika admin ingin mengelola mahasiswa. Admin dapat mengubah dan menghapus data mahasiswa jika diperlukan.                                      | Admin               |
| 7 | Mengelola Data Repositori  | <i>Use case</i> ini berjalan ketika admin mengelola data repositori yang dibuat oleh mahasiswa. Admin dapat mengubah dan menghapus data repositori jika diperlukan           | Admin               |
| 8 | Mengelola Data Diri        | <i>Use case</i> ini berjalan ketika pengguna ingin melengkapi data diri atau memperbarui data pada akun repositori.  | Admin dan Mahasiswa |
| 9 | <i>Logout</i>              | <i>Use case</i> ini berjalan ketika pengguna ingin keluar dari sistem repositori.  | Admin dan Mahasiswa |

***Tabel 3. 1 Use case glossary***

Sumber: Hasil Perancangan

### 3.3.4 Actor Glossary

*Actor Glossary* merupakan penjelasan singkat terhadap aktor dan *use case* yang terkait pada *use case diagram*. *Actor Glossary* dapat dilihat pada table berikut ini.

| No | Aktor     | Deskripsi  | Use Case   |
|----|-----------|--|--|
| 1  | Admin     | Aktor yang dapat mengelola sistem secara penuh   | <ul style="list-style-type: none"> <li>- Login</li> <li>- Mengelola User</li> <li>- Mengelola Repositori</li> <li>- Mengelola Tugas Akhir</li> <li>- Mengelola Data Diri</li> <li>- Melihat Tugas Akhir</li> <li>- Logout</li> </ul> |
| 2  | Mahasiswa | Aktor yang login ke dalam sistem dapat melihat Tugas Akhir dan menambahkan Tugas Akhir favorit | <ul style="list-style-type: none"> <li>- Login</li> <li>- Registrasi</li> <li>- Melihat Tugas Akhir</li> <li>- Mengelola Data Diri</li> <li>- Menambahkan Tugas Akhir</li> <li>- Logout</li> </ul>                                   |

**Tabel 3. 2 Actor Glossary**

Sumber: Hasil Perancangan

### 3.3.5 Use Case Spesification

#### a. Use Case Spesification Login

*Use case specification Login user* pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.3.

|                    |   |        |
|--------------------|---|--------|
| <i>Actor</i>       | Admin dan Mahasiswa   |        |
| <i>Description</i> | <i>Use case</i> login digunakan saat pengguna mencoba untuk masuk ke sistem repositori menggunakan email dan password yang terdaftar. Tujuannya untuk masuk kedalam sistem. |        |
| <i>Basic Flow</i>  | Aktor   | Sistem |

|                            |  |   |
|----------------------------|--|---|
|                            | 1. Pengguna mengakses sistem repositori  |   |
|                            |  | 2. Menampilkan halaman login  |
|                            | 3. Aktor memasukan email dan password  |   |
|                            |  | 4. Sistem melakukan validasi email dan password yang dimasukkan.<br>5. Jika email dan password cocok dengan akun yang terdaftar, maka akan dilanjutkan ke dashboard sesuai dengan role. |
| <i>Alternative Flow</i>    | Jika username dan password yang dimasukan salah. Maka, sistem akan menampilkan <i>error</i> dan tidak sistem tidak lanjut ke dashboard sistem. |   |
| <i>Special Requirement</i> | Aktor belum melakukan login ke sistem  |   |
| <i>Pre-condition</i>       | Aktor belum melakukan login ke sistem  |   |
| <i>Post-condition</i>      | Aktor sudah melakukan login ke sistem  |   |
| <i>Extension Point</i>     | N/A  |   |

**Tabel 3. 3 Use case specification Login user**

Sumber: Hasil Perancangan

**b. Use Case Specification Registrasi**

*Use case specification* Registrasi mahasiswa pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.4.

|                    |  |  |
|--------------------|--|--|
| <i>Actor</i>       | Mahasiswa  |  |
| <i>Description</i> | <i>Use case</i> registrasi digunakan saat pengguna baru ingin membuat akun untuk masuk ke sistem repositori dengan mengisi <i>form</i> registrasi berupa nama, email, password, nomor identitas, Jurusan, program studi, dan tipe akun yang akan dibuat. |  |
| <i>Basic Flow</i>  | Aktor  | Sistem   |
|                    | 1. Pengguna yang belum memiliki akun mengakses halaman registrasi.   |  |
|                    |  | 2. Sistem menampilkan pilihan jenis akun yang akan dibuat, yaitu mahasiswa.  |
|                    | 3. Pengguna memilih jenis akun yang akan dibuat.   |  |
|                    |  | 4. Sistem menampilkan form registrasi.   |
|                    | 5. Pengguna mengisi data yang diperlukan untuk membuat akun baru.  |  |
|                    |  | 6. Sistem melakukan validasi kecocokan data yang diperlukan.<br>7. Jika data sesuai maka data pengguna akan didaftarkan pada sistem. |

|                            |   |
|----------------------------|---|
| <i>Alternative Flow</i>    | Jika data yang dimasukkan pengguna tidak sesuai. Maka, sistem akan meminta memasukan ulang dan pengguna akun tidak didaftarkan ke sistem. |
| <i>Special Requirement</i> | Pengguna baru belum memiliki akun yang terdaftar pada sistem.   |
| <i>Pre-condition</i>       | Pengguna baru belum memiliki akun yang terdaftar pada sistem.   |
| <i>Post-condition</i>      | Pengguna baru berhasil mendaftarkan akun pada sistem.   |
| <i>Extension Point</i>     | N/A   |

**Tabel 3. 4 Use case specification Registrasi mahasiswa pada sistem repositori tugas akhir**

Sumber: Hasil Perancangan

**c. Use Case Spesification Membuat Repositori**

*Use case specification* Membuat Repositori pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.5.

|                    |   |  |
|--------------------|---|--|
| <i>Actor</i>       | Mahasiswa   |  |
| <i>Description</i> | <i>Use case</i> membuat repositori digunakan saat mahasiswa ingin membuat sebuah repositori sebelum menyimpan data tugas akhir. |  |
| <i>Basic Flow</i>  | Aktor   | Sistem   |
|                    | 1. Mahasiswa mengakses halaman repositori.  |  |
|                    |   | 2. Sistem menampilkan halaman repositori             |
|                    | 3. Mahasiswa memilih tambah repositori.   |  |
|                    |   | 4. Sistem menampilkan <i>form</i> tambah repositori. |
|                    | 5. Mahasiswa mengisi data repositori.   |  |

|                            |   |   |
|----------------------------|---|---|
|                            |   | <p>6. Sistem melakukan validasi terhadap data yang dimasukkan oleh mahasiswa.</p> <p>7. Jika data yang dimasukkan sesuai maka sistem akan menyimpan data ke sistem.</p> |
| <i>Alternative Flow</i>    | Jika data yang dimasukkan tidak sesuai. Maka, sistem meminta memasukkan data ulang, dan data yang salah tidak disimpan pada sistem. |   |
| <i>Special Requirement</i> | Mahasiswa yang ingin membuat repositori baru.   |   |
| <i>Pre-condition</i>       | Mahasiswa yang ingin membuat repositori baru.   |   |
| <i>Post-condition</i>      | Mahasiswa berhasil membuat repositori baru.   |   |
| <i>Extension Point</i>     | N/A   |   |

**Tabel 3. 5 Use case specification membuat repositori pada sistem repositori tugas akhir**

Sumber: Hasil Perancangan

**d. Use Case Spesification Upload File Tugas Akhir**

Use case specification Upload File Tugas Akhir pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.6.

|                    |   |  |
|--------------------|---|--|
| <i>Actor</i>       | Mahasiswa   |  |
| <i>Description</i> | Use case mengupload tugas akhir dan digunakan saat Mahasiswa ingin menambahkan tugas akhir pada repositori yang sudah dibuat. |  |
| <i>Basic Flow</i>  | Aktor   | Sistem                                   |
|                    | 1. Mahasiswa mengakses halaman repositori.  |  |
|                    |   | 2. Sistem menampilkan halaman repositori |

|                            |   |  |
|----------------------------|---|--|
|                            | 3. Mahasiswa memilih tambah tugas akhir.  |  |
|                            |   | 4. Sistem menampilkan form tambah repositori.  |
|                            | 5. Dosen mengisi data tugas akhir.  |  |
|                            |   | 6. Sistem melakukan validasi terhadap data yang dimasukkan oleh Mahasiswa.<br>7. Jika data yang dimasukkan sesuai maka sistem akan menyimpan data ke sistem. |
| <i>Alternative Flow</i>    | Jika data yang dimasukkan tidak sesuai. Maka, sistem meminta memasukkan data ulang, dan data yang salah tidak disimpan pada sistem. |  |
| <i>Special Requirement</i> | Mahasiswa yang ingin menambahkan tugas akhir baru.  |  |
| <i>Pre-condition</i>       | Mahasiswa yang ingin menambahkan tugas akhir baru.  |  |
| <i>Post-condition</i>      | Mahasiswa berhasil menambahkan tugas akhir baru.  |  |
| <i>Extension Point</i>     | N/A   |  |

**Tabel 3. 6 Use case specification Upload File Tugas Akhir pada sistem repositori tugas akhir**

Sumber: Hasil Perancangan

e. **Use Case Spesification Mengelola Data Tugas Akhir**

*Use case specification* mengelola data tugas akhir pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.7.

|                    |  |   |
|--------------------|--|---|
| <i>Actor</i>       | Admin  |   |
| <i>Description</i> | <i>Use case</i> mengelola data tugas akhir, digunakan saat admin ingin mengelola data tugas akhir yang sudah diupload. |   |
| <i>Basic Flow</i>  | Aktor  | Sistem  |
|                    | 1. Admin memilih repositori yang ingin dikelola data tugas akhirnya.   |   |
|                    |  | 2. Sistem menampilkan repositori yang dipilih.        |
|                    | 3. Admin memilih tugas akhir yang ingin dikelola.  |   |
|                    |  | 4. Sistem menampilkan tugas akhir yang dipilih admin. |
|                    | 5. Admin memilih menu perbarui data tugas akhir.   |   |
|                    |  | 6. Sistem menampilkan form perbarui data tugas akhir. |
|                    | 7. Admin memasukkan data yang ingin diperbarui.  |   |
|                    |  | 8. Sistem melakukan validasi terhadap data            |

|                            |  |   |
|----------------------------|--|---|
|                            |  | yang dimasukkan oleh admin.<br>9. Jika data yang dimasukkan sesuai maka sistem akan menyimpan data ke sistem. |
| <i>Alternative Flow</i>    | a. Jika admin memilih menu hapus tugas akhir, sistem akan menampilkan konfirmasi hapus, kemudian sistem akan menghapus data tugas akhir.<br>b. Jika data yang dimasukkan tidak sesuai, sistem meminta memasukkan data ulang, dan data yang salah tidak disimpan pada sistem. |   |
| <i>Special Requirement</i> | Admin ingin mengelola data tugas akhir.  |   |
| <i>Pre-condition</i>       | Admin ingin mengelola data tugas akhir.  |   |
| <i>Post-condition</i>      | Admin berhasil mengelola data tugas akhir.   |   |
| <i>Extension Point</i>     | N/A  |   |

**Tabel 3. 7 Use case specification mengelola data tugas akhir pada sistem repositori tugas akhir**

Sumber: Hasil Perancangan

**f. Use Case Spesification Menambahkan Tugas Akhir Favorit**

*Use case specification* menambahkan tugas akhir favorit pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.8.

|                    |  |        |
|--------------------|--|--------|
| <i>Actor</i>       | Mahasiswa  |        |
| <i>Description</i> | <i>Use case</i> menambahkan tugas akhir favorit, digunakan saat mahasiswa ingin menyimpan tugas akhir agar mudah ditemukan jika suatu saat nanti memerlukan tugas akhir tersebut lagi. |        |
| <i>Basic Flow</i>  | Aktor  | Sistem |

|                            |  |  |
|----------------------------|--|--|
|                            | 1. Dosen dan mahasiswa menambahkan tugas akhir ke favorit. |  |
|                            |  | 2. Sistem menyimpan tugas akhir ke favorit akun. |
| <i>Alternative Flow</i>    | -  |  |
| <i>Special Requirement</i> | Mahasiswa yang ingin menambahkan tugas akhir ke favorit.   |  |
| <i>Pre-condition</i>       | Mahasiswa yang ingin menambahkan tugas akhir ke favorit.   |  |
| <i>Post-condition</i>      | Mahasiswa berhasil menambahkan tugas akhir ke favorit.     |  |
| <i>Extension Point</i>     | N/A  |  |

**Tabel 3. 8 Use case specification menambahkan tugas akhir favorit pada sistem repositori tugas akhir**

Sumber: Hasil Perancangan

**g. Use Case Spesification Mengelola Data Mahasiswa**

*Use case specification* mengelola data Mahasiswa pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.9.

|                    |   |   |
|--------------------|---|---|
| <i>Actor</i>       | Admin   |   |
| <i>Description</i> | <i>Use case</i> mengelola data mahasiswa digunakan saat admin ingin mengelola data pengguna yang sudah terregistrasi. |   |
| <i>Basic Flow</i>  | Aktor   | Sistem  |
|                    | 1. Admin mengakses halaman Kelola data mahasiswa.   |   |
|                    |   | 2. Sistem menampilkan daftar pengguna yang sudah terdaftar. |

|                            |   |  |
|----------------------------|---|--|
|                            | 3. Admin memilih pengguna yang ingin dikelola.  |  |
|                            |   | 4. Sistem menampilkan data pengguna.   |
|                            | 5. Admin memilih menu perbarui data pengguna.   |  |
|                            |   | 6. Sistem melakukan validasi terhadap data yang dimasukkan oleh admin.<br>7. Jika data yang dimasukkan sesuai maka sistem akan menyimpan data ke sistem. |
| <i>Alternative Flow</i>    | <p>a. Admin memilih hapus data pengguna, sistem menampilkan konfirmasi, kemudian sistem akan menghapus data pengguna.</p> <p>b. Jika data yang dimasukkan tidak sesuai, sistem meminta memasukkan data ulang, dan data yang salah tidak disimpan pada sistem.</p> |  |
| <i>Special Requirement</i> | Admin ingin mengelola data mahasiswa.   |  |
| <i>Pre-condition</i>       | Admin ingin mengelola data mahasiswa.   |  |
| <i>Post-condition</i>      | Admin berhasil mengelola data mahasiswa.  |  |
| <i>Extension Point</i>     | N/A   |  |

**Tabel 3. 9 Use case specification mengelola data Mahasiswa pada sistem repositori tugas akhir**

Sumber: Hasil Perancangan

#### **h. Use Case Spesification Mengelola Data Repositori**

*Use case specification* mengelola data Repositori pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.10.

|                    |  |   |
|--------------------|--|---|
| <i>Actor</i>       | Admin  |   |
| <i>Description</i> | <i>Use case</i> mengelola data repositori, digunakan saat admin ingin mengelola data repositori. |   |
| <i>Basic Flow</i>  | Aktor  | Sistem  |
|                    | 1. Admin mengakses halaman Kelola data repositori.   |   |
|                    |  | 2. Sistem menampilkan daftar repositori yang terdapat pada sistem.  |
|                    | 3. Admin memilih repositori yang ingin diubah atau dihapus.                                      |   |
|                    |  | 4. Sistem menampilkan repositori yang dipilih oleh admin.   |
|                    | 5. Admin memilih menu perbarui data repositori.  |   |
|                    |  | 6. Sistem menampilkan form perbarui data repositori   |
|                    | 7. Admin mengisi data yang ingin diperbarui.   |   |
|                    |  | 8. Sistem melakukan validasi terhadap data yang dimasukkan oleh admin.<br>9. Jika data yang dimasukkan sesuai |

|                            |  |  |
|----------------------------|--|--|
|                            |  | maka sistem akan menyimpan data ke sistem. |
| <i>Alternative Flow</i>    | a. Jika admin memilih menu hapus, maka sistem akan menampilkan konfirmasi hapus dan menghapus data repositori.<br>b. Jika data yang dimasukkan tidak sesuai, sistem meminta memasukkan data ulang, dan data yang salah tidak disimpan pada sistem. |  |
| <i>Special Requirement</i> | Admin ingin mengelola data repositori.   |  |
| <i>Pre-condition</i>       | Admin ingin mengelola data repositori.   |  |
| <i>Post-condition</i>      | Admin berhasil mengelola data repositori.  |  |
| <i>Extension Point</i>     | N/A  |  |

**Tabel 3. 10 Use case specification mengelola data Repositori pada sistem repositori tugas akhir**

Sumber: Hasil Perancangan

**i. Use Case Spesification Mengelola Data Diri**

*Use case specification* mengelola data diri pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.11.

|                    |   |   |
|--------------------|---|---|
| <i>Actor</i>       | Admin dan Mahasiswa   |   |
| <i>Description</i> | <i>Use case</i> mengelola data diri digunakan saat pengguna ingin memperbarui data diri pada sistem repositori. |   |
| <i>Basic Flow</i>  | Aktor   | Sistem                                    |
|                    | 1. Pengguna mengakses halaman kelola data diri.   |   |
|                    |   | 2. Sistem menampilkan data diri pengguna. |
|                    | 3. Pengguna memilih menu perbarui data diri.  |   |

|                            |  |   |
|----------------------------|--|---|
|                            |  | 4. Sistem menampilkan form untuk memperbarui data.                  |
|                            | 5. Pengguna mengubah data lama menjadi data yang baru. |   |
|                            |  | 6. Sistem memvalidasi data, kemudian menyimpan data baru ke sistem. |
| <i>Alternative Flow</i>    | Sistem meminta memasukan ulang data yang tidak sesuai. |   |
| <i>Special Requirement</i> | Pengguna yang ingin memperbarui data diri pada sistem  |   |
| <i>Pre-condition</i>       | Pengguna akan memperbarui data diri pada sistem.       |   |
| <i>Post-condition</i>      | Pengguna berhasil menyimpan data baru pada sistem.     |   |
| <i>Extension Point</i>     | N/A  |   |

**Tabel 3. 11 Use case specification mengelola data diri pada sistem repositori tugas akhir**

Sumber: Hasil Perancangan

**j. Use Case Spesification Logout**

*Use case specification logout* pada sistem repositori tugas akhir ditunjukkan oleh Tabel 3.12.

|                    |  |  |
|--------------------|--|--|
| <i>Actor</i>       | Admin dan Mahasiswa  |  |
| <i>Description</i> | <i>Use case logout</i> digunakan saat pengguna ingin keluar dari sistem. |  |
| <i>Basic Flow</i>  | Aktor  | Sistem                                       |
|                    | 1. Pengguna memilih menu logout.   |  |
|                    |  | 2. Sistem mengeluarkan pengguna dari sistem. |

|                            |   |
|----------------------------|---|
| <i>Alternative Flow</i>    | -                                       |
| <i>Special Requirement</i> | Pengguna yang ingin keluar dari sistem. |
| <i>Pre-condition</i>       | Pengguna yang ingin keluar dari sistem. |
| <i>Post-condition</i>      | Pengguna berhasil keluar dari sistem.   |
| <i>Extension Point</i>     | N/A                                     |

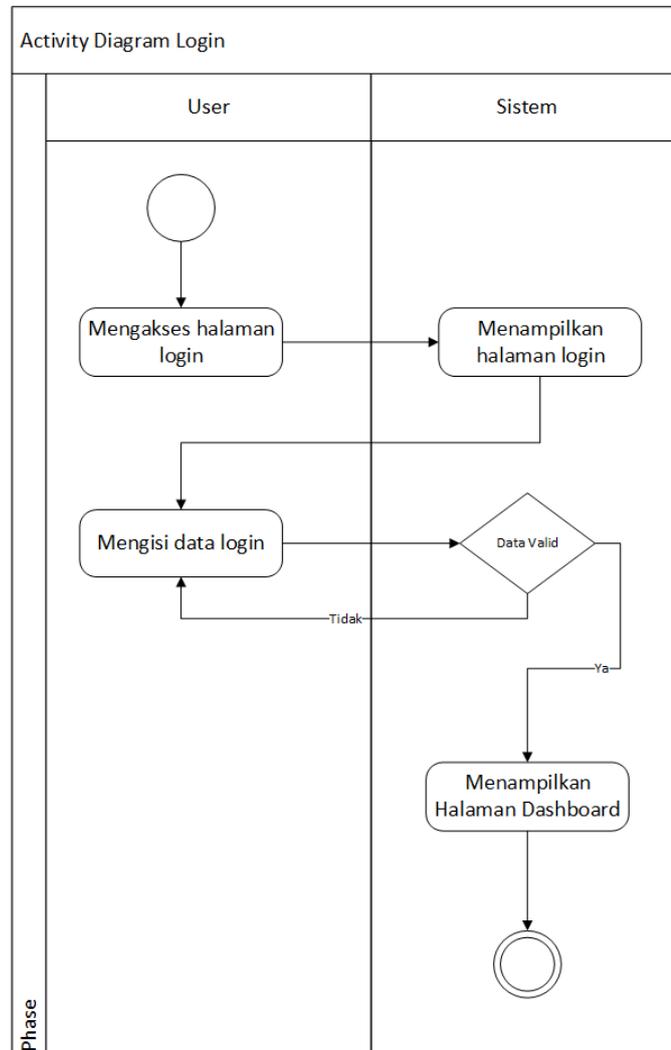
***Tabel 3. 12 Use case specification logout pada sistem repositori tugas akhir***

Sumber: Hasil Perancangan

### 3.3.6 Activity diagram

#### a. Activity Diagram Login

Activity diagram login pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.7.



**Gambar 3. 7 Activity diagram login**

Sumber: Hasil Perancangan

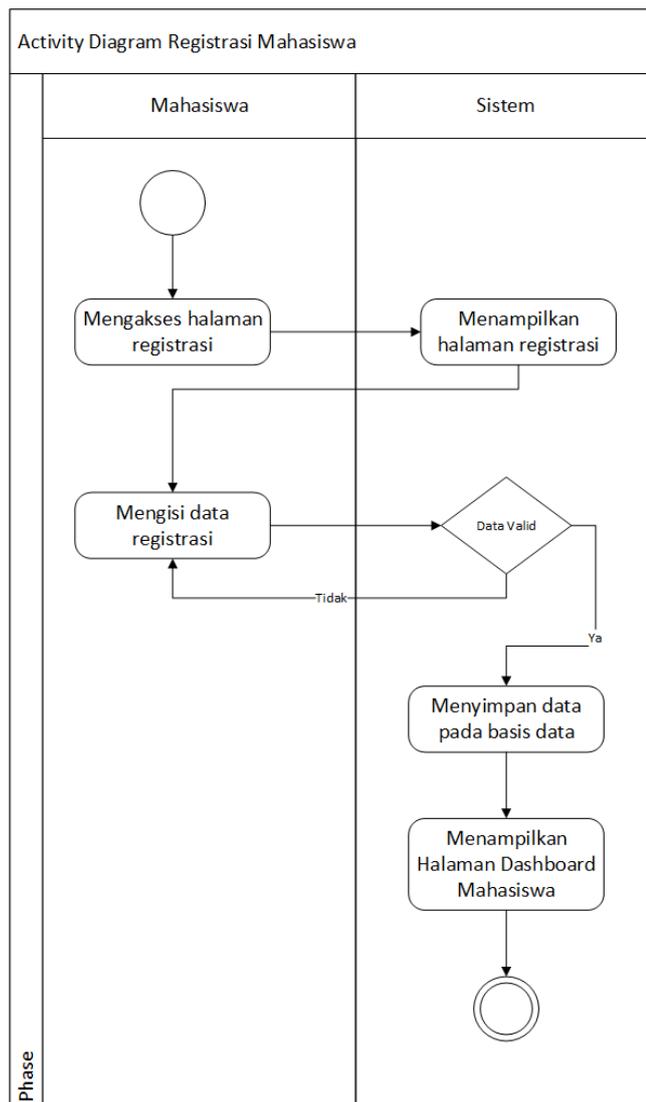
Keterangan :

1. Pengguna mengakses sistem, selanjutnya sistem akan menampilkan halaman *login*.
2. Pengguna mengisi data *login* berupa *email* dan *password*, kemudian menekan tombol *login*.
3. Sistem memvalidasi data yang dimasukkan oleh pengguna.

4. Apabila data yang dimasukkan tidak sesuai, pengguna diminta memasukkan ulang data login.
5. Apabila data yang dimasukkan sesuai maka sistem akan menampilkan halaman dashboard pengguna.

**b. Activity Diagram Registrasi**

Activity diagram registrasi pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.8.



**Gambar 3. 8 Activity diagram registrasi**

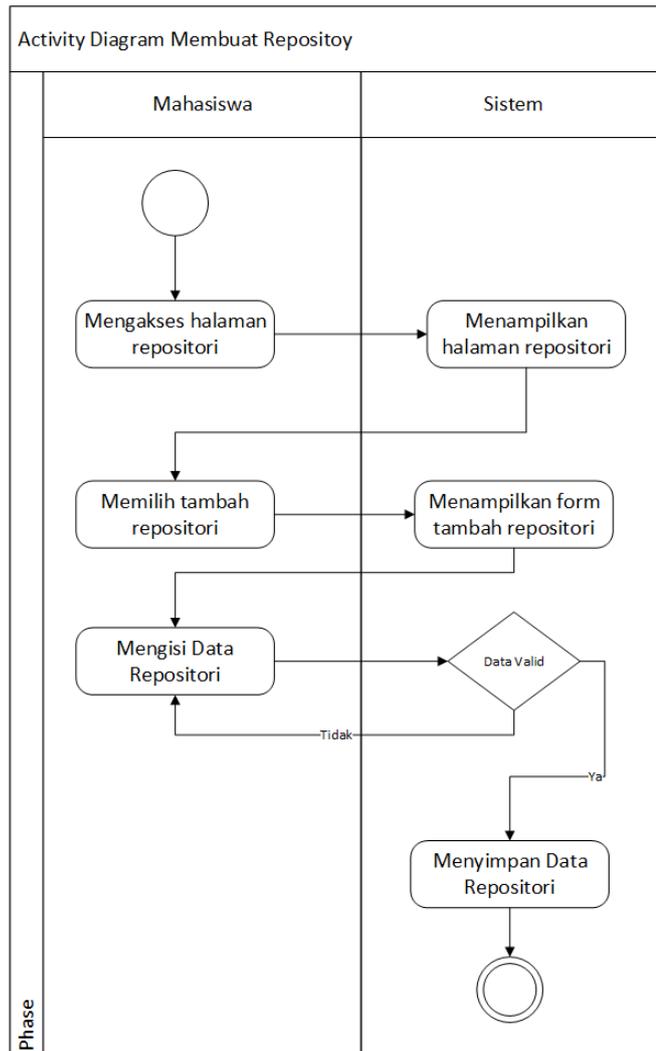
Sumber: Hasil Perancangan

Keterangan :

1. Pengguna yang belum memiliki akun yang terdaftar di sistem, memilih menu daftar akun. Kemudian sistem akan menampilkan form daftar.
2. Pengguna mengisi data pendaftaran yang diperlukan oleh sistem.
3. Sistem akan melakukan validasi untuk mencocokkan data data yang diperlukan dengan data yang dimasukkan oleh pengguna.
4. Apabila data yang dimasukkan belum sesuai maka sistem akan meminta memasukkan data ulang.
5. Apabila data yang dimasukkan sudah sesuai maka sistem akan mendaftarkan pengguna ke sistem.
6. Setelah registrasi berhasil maka sistem akan mengarahkan pengguna ke halaman login.

**c. *Activity Diagram* Membuat Repositori**

*Activity diagram* membuat repositori pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.9.



**Gambar 3. 9 Activity diagram membuat repositori**

Sumber: Hasil Perancangan

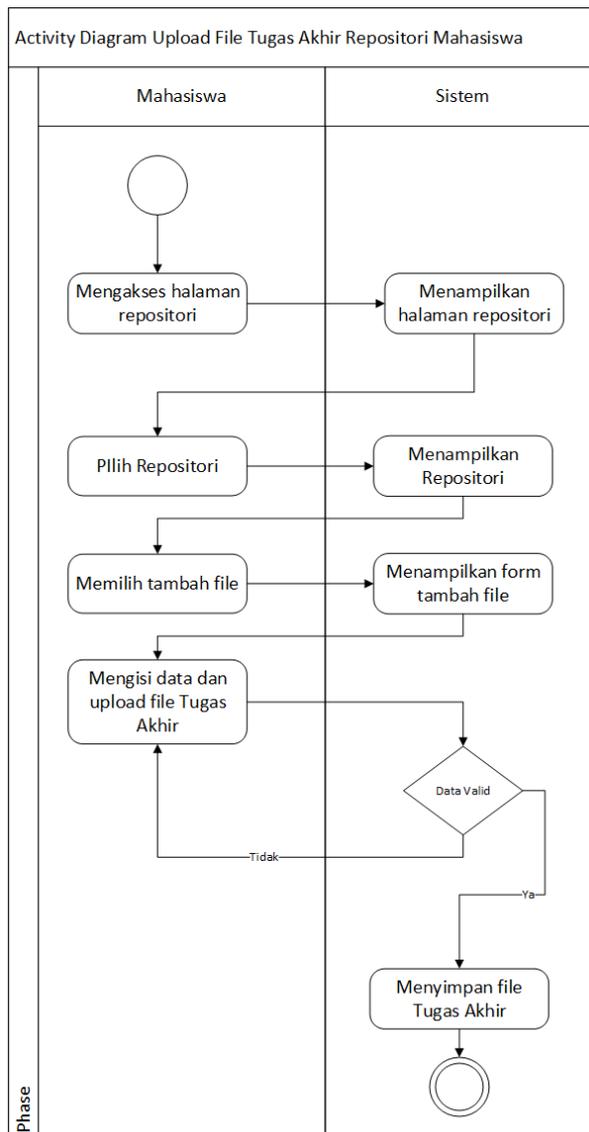
Keterangan :

1. Mahasiswa yang ingin membuat repositori baru mengakses halaman repositori terlebih dahulu. Kemudian sistem menampilkan halaman repositori.
2. Mahasiswa memilih menu tambah repositori, kemudian sistem menampilkan form tambah repositori.
3. Mahasiswa memasukkan data yang diperlukan sistem untuk membuat repositori baru.
4. Sistem melakukan validasi terhadap data yang dimasukkan oleh mahasiswa sebelumnya.

5. Apabila data yang dimasukkan tidak sesuai maka sistem akan meminta mahasiswa untuk memasukkan data ulang.
6. Apabila data yang dimasukkan oleh mahasiswa sudah sesuai maka sistem akan melanjutkan untuk menyimpan data repositori ke dalam sistem.
7. Kemudian sistem menampilkan halaman repositori.

**d. Activity Diagram Mengunggah Tugas Akhir**

Activity diagram mengunggah tugas akhir pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.10.



**Gambar 3. 10 Activity diagram mengunggah tugas akhir**

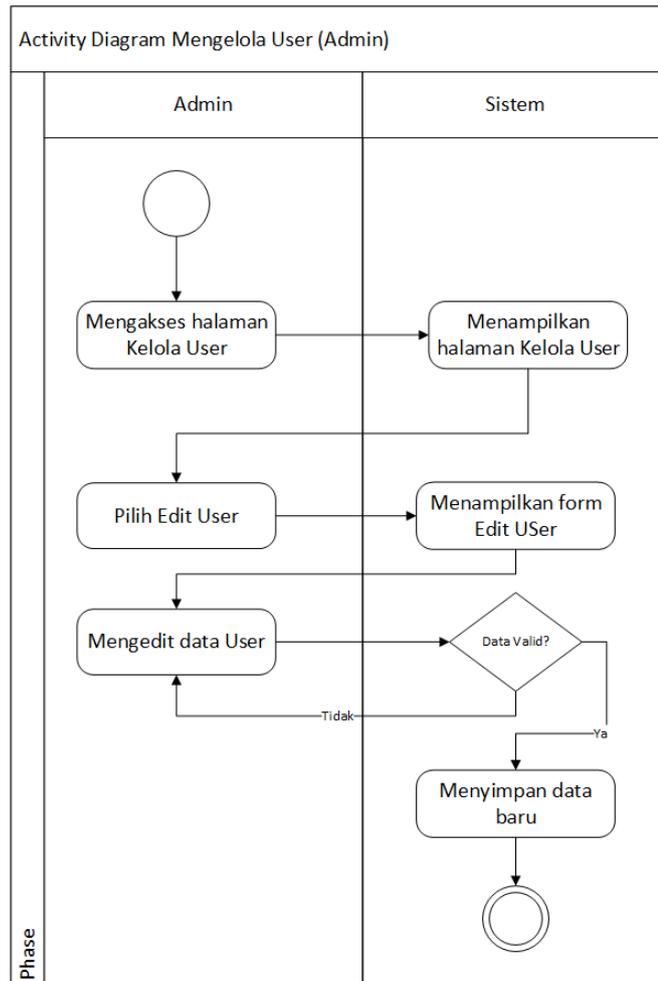
Sumber: Hasil Perancangan

Keterangan :

1. Mahasiswa yang ingin mengunggah tugas akhir harus mengakses repositori terlebih dahulu, kemudian sistem menampilkan daftar repositori
2. Setelah repositori dipilih, sistem menampilkan halaman detail repositori.
3. Mahasiswa memilih menu upload tugas akhir pada halaman repositori. Maka sistem akan menampilkan form upload tugas akhir.
4. Mahasiswa memasukkan data data yang diperlukan untuk mengupload tugas akhir.
5. Sistem memvalidasi data yang sudah dimasukkan oleh mahasiswa.
6. Apabila data yang telah dimasukkan tidak sesuai, sistem akan meminta untuk memasukkan data ulang
7. Apabila data yang dimasukkan sudah sesuai, maka sistem akan menyimpan data data tersebut.

**e. *Activity Diagram* Mengelola Data Mahasiswa**

*Activity diagram* mengelola data mahasiswa pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.11.



**Gambar 3. 11 Activity diagram mengelola data mahasiswa**

Sumber: Hasil Perancangan

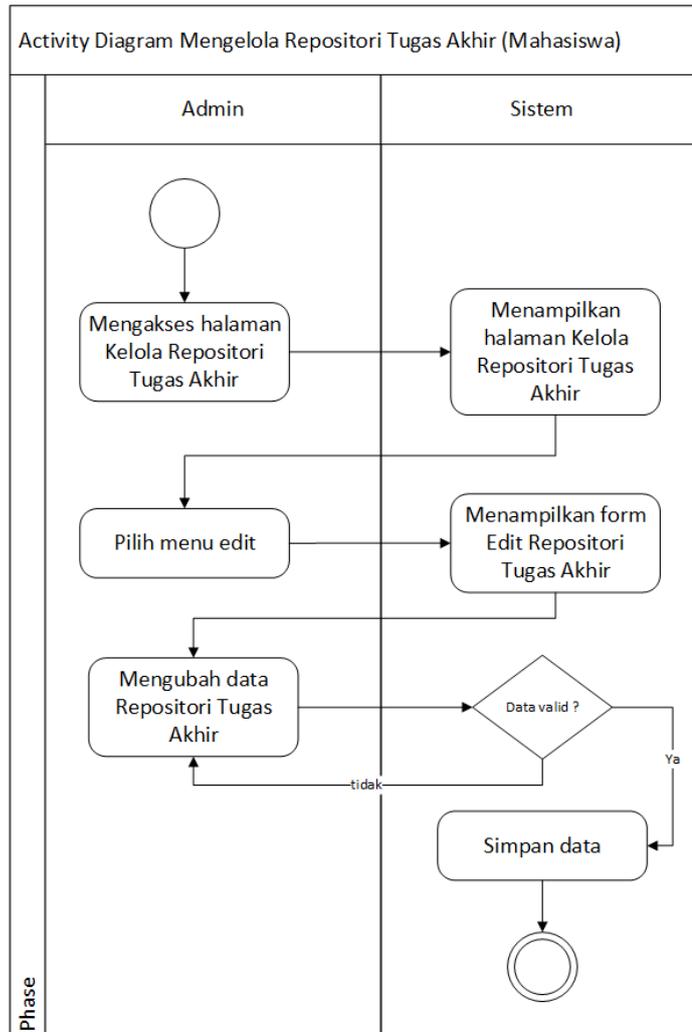
Keterangan:

1. Admin yang ingin mengelola data akun pengguna dapat mengakses halaman kelola user. Kemudian sistem menampilkan daftar akun pengguna.
2. Admin memilih akun pengguna mana yang akan dikelola oleh admin, kemudian sistem akan menampilkan detail akun pengguna.
3. Admin memilih menu edit user, kemudian sistem menampilkan form perbarui data pengguna.
4. Admin memasukkan data pengguna yang baru.
5. Sistem akan memvalidasi data yang sudah dimasukkan oleh admin.
6. Apabila data yang tidak dimasukkan oleh admin sesuai maka sistem akan meminta memasukkan data ulang.

7. Apabila data yang dimasukkan oleh admin sudah sesuai maka sistem akan menyimpan data pengguna yang sudah diperbarui.

**f. Activity Diagram Mengelola Data Repositori**

Activity diagram mengelola data repositori pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.12.



**Gambar 3. 12 Activity diagram mengelola data repositori**

Sumber: Hasil Perancangan

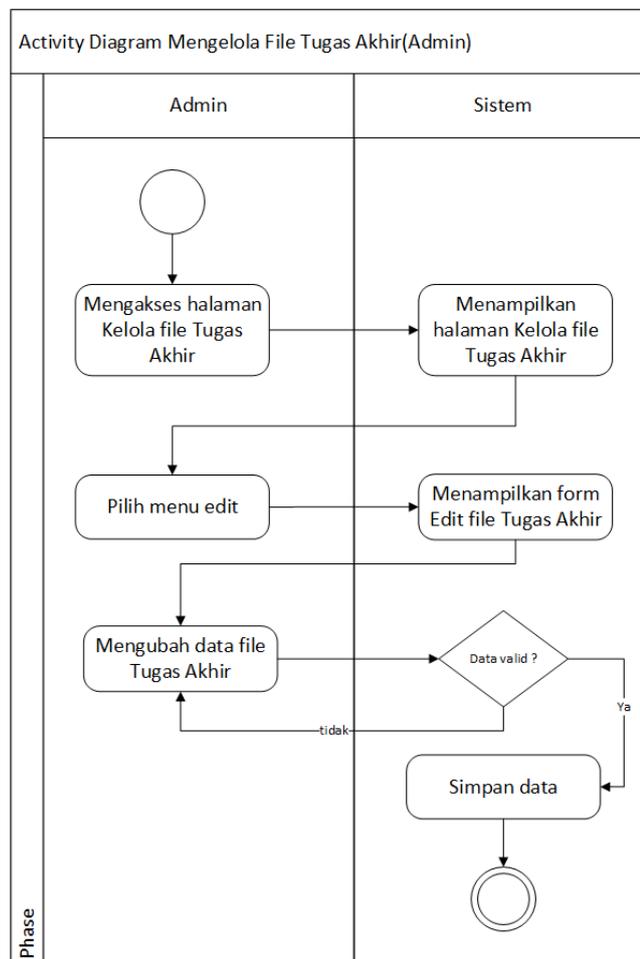
Keterangan:

1. Admin yang ingin mengelola data repositori mengakses halaman repositori kemudian sistem akan menampilkan daftar repositori.
2. Admin memilih repositori yang ingin dikelola oleh admin, kemudian sistem akan menampilkan halaman detail repositori yang dipilih oleh admin.

3. Admin memilih menu edit repositori, kemudian sistem akan menampilkan form perbarui data repositori.
4. Admin memasukkan data baru ke form yang ditampilkan oleh sistem.
5. Sistem memvalidasi data yang telah dimasukkan oleh admin.
6. Apabila admin tidak memasukkan data yang sesuai, maka sistem akan meminta untuk memasukkan data ulang.
7. Apabila data yang sudah dimasukkan sesuai maka data baru akan disimpan kedalam sistem.

**g. Activity Diagram Mengelola Data Tugas Akhir**

Activity diagram mengelola data tugas akhir pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.13.



**Gambar 3. 13 Activity diagram mengelola data tugas akhir**

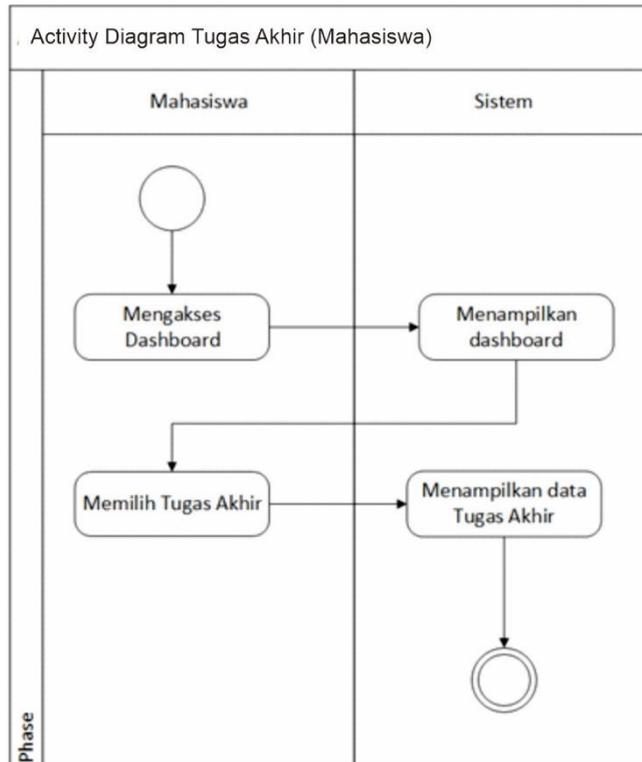
Sumber: Hasil Perancangan

Keterangan:

1. Admin yang ingin mengelol tugas akhir dalam repositori mengakses halaman repositori, kemudian sistem akan menampilkan daftar tugas akhir dalam repositori.
2. Admin memilih tugas akhir yang ingin diperbarui datanya. Kemudian sistem akan menampilkan halaman detail tugas akhir.
3. Admin memilih menu edit tugas akhir, sistem akan menampilkan form perbarui data tugas akhir.
4. Admin mengisi data tugas akhir yang ingin diperbarui.
5. Sistem akan memvalidasi data tugas akhir yang sudah dimasukkan oleh admin sebelumnya
6. Apabila data yang dimasukkan oleh admin tidak sesuai maka sistem meminta untuk memasukkan ulang data.
7. Apabila data yang dimasukkan sudah sesuai maka sistem akan langsung menyimpan data yang dimasukkan kedalam sistem.

**h. Activity Diagram Melihat Tugas Akhir**

*Activity diagram* melihat tugas akhir pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.14.



**Gambar 3. 14 Activity diagram melihat tugas akhir**

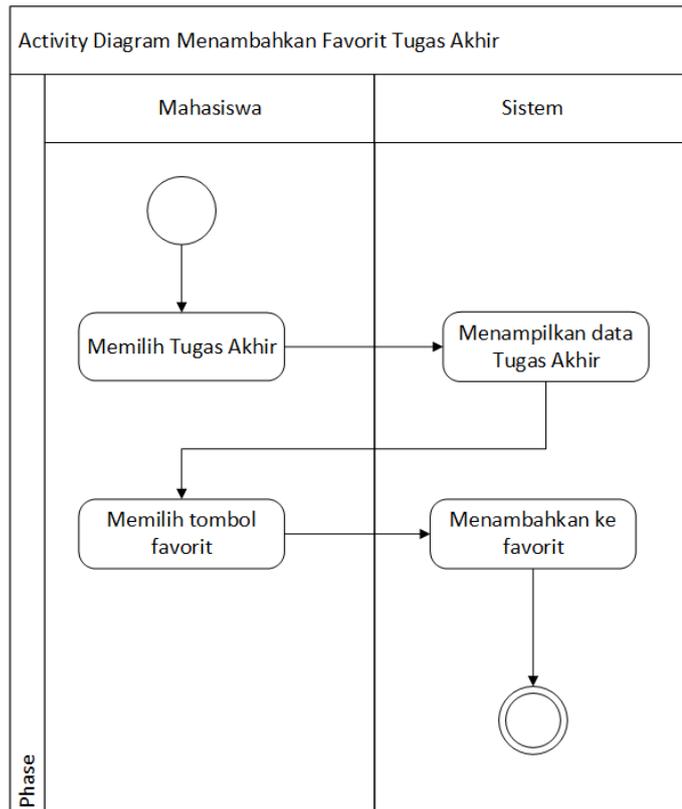
Sumber: Hasil Perancangan

Keterangan :

1. Mahasiswa yang ingin melihat tugas akhir dapat mencari tugas akhir di dashboard, kemudian sistem akan menampilkan tugas akhir.
2. Mahasiswa dapat memilih tugas akhir yang diinginkan.
3. Sistem akan menampilkan tugas akhir yang dipilih.

**i. Activity Diagram Menambahkan Tugas Akhir Favorit**

Activity diagram menambahkan tugas akhir favorit pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.15.



**Gambar 3. 15 Activity diagram menambahkan tugas akhir favorit**

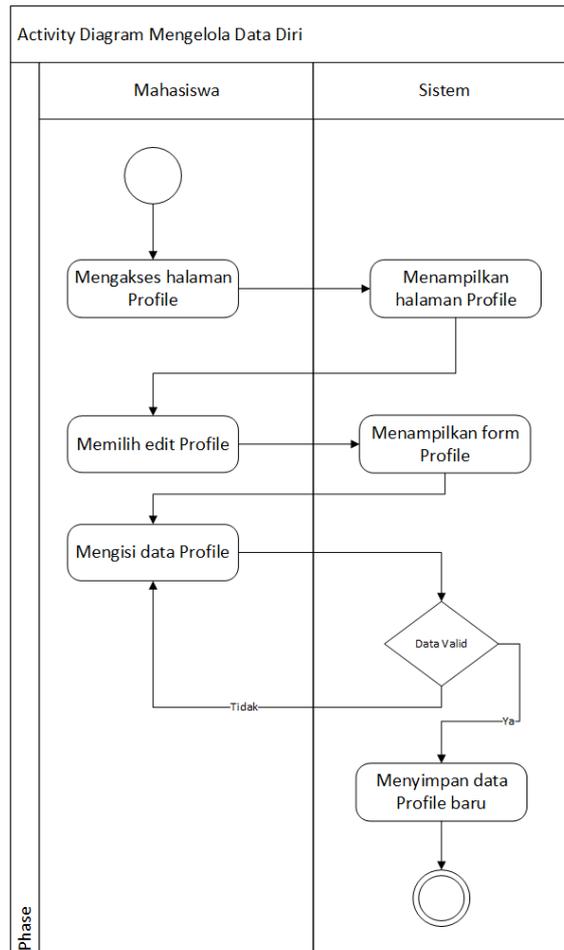
Sumber: Hasil Perancangan

Keterangan:

1. Mahasiswa yang ingin menambahkan tugas akhir kefavorit akunya, memilih tugas akhir yang ingin ditambahkan terlebih dahulu.
2. Sistem menampilkan detail tugas akhir.
3. Mahasiswa memilih menu tambah favorit.
4. Sistem menambahkan tugas akhir ke favorit akun pengguna.

**j. Activity Diagram Mengelola Data Diri**

*Activity diagram* mengelola data diri pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.16.



**Gambar 3. 16 Activity diagram mengelola data diri**

Sumber: Hasil Perancangan

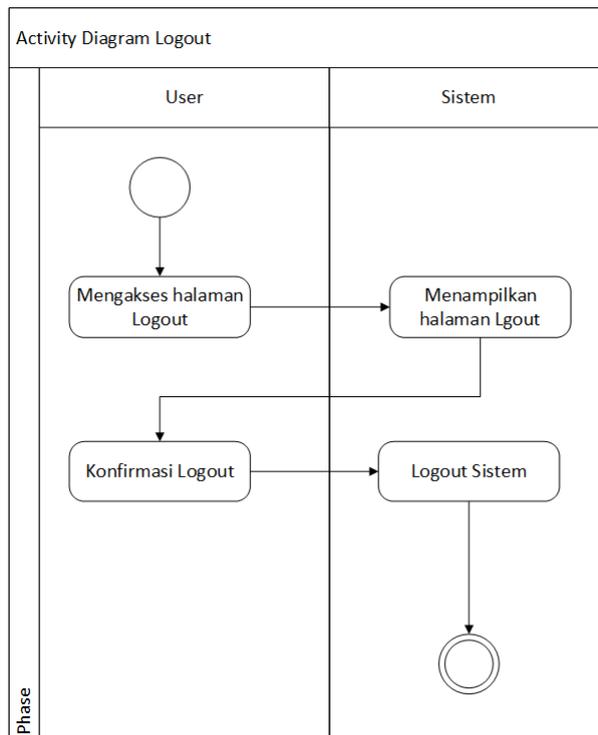
Keterangan :

1. Mahasiswa yang ingin memperbarui data diri mengakses halaman profile. Kemudian sistem akan menampilkan halaman *profile*.
2. Mahasiswa memilih menu edit *profile*, kemudian sistem menampilkan form edit profile.
3. Mahasiswa mengisi data profile yang baru, kemudian memilih tombol simpan.
4. Sistem akan memvalidasi data yang sudah dimasukkan oleh mahasiswa.
5. Apabila data yang dimasukkan tidak sesuai maka sistem akan meminta mahasiswa untuk memasukkan ulang data yang ingin diperbarui.

6. Apabila data yang dimasukkan oleh mahasiswa sudah sesuai dengan data yang diperlukan maka sistem akan menyimpan data baru yang sudah dimasukkan oleh mahasiswa.
7. Sistem akan menampilkan lagi halaman *profile* dengan data yang sudah diperbarui.

**k. Activity Diagram Logout**

Activity diagram *logout* pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.17.



**Gambar 3. 17 Activity diagram logout**

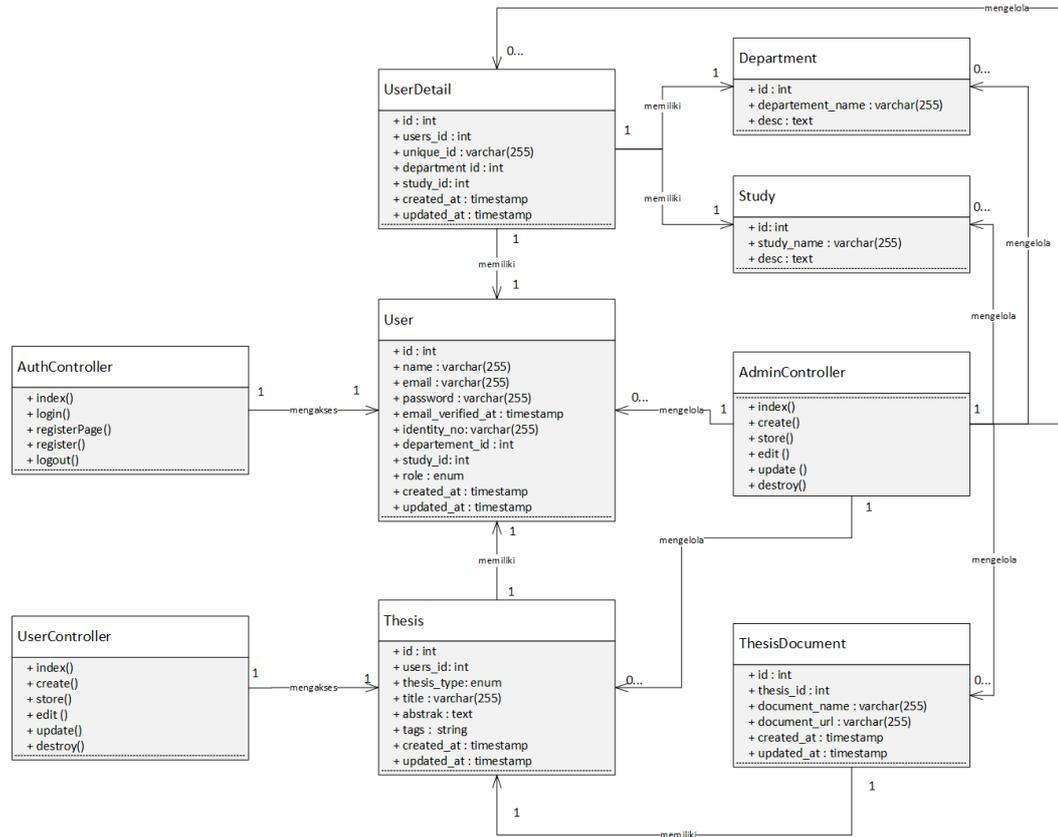
Sumber: Hasil Perancangan

Keterangan:

1. Pengguna yang ingin keluar sistem dapat memilih menu logout
2. Sistem akan menampilkan konfirmasi *logout*
3. Pengguna mengkonfirmasi *logout*
4. Sistem menjalankan fungsi *logout* pengguna.

### 3.3.7 Class Diagram

Class diagram dapat digunakan untuk menggambarkan hubungan antara class class pada sistem yang dibangun. Hasil Perancangan class diagram pada sistem repository tugas akhir dapat dilihat pada gambar berikut ini.



**Gambar 3. 18 class diagram**

Sumber: Hasil Perancangan

Keterangan class diagram pada sistem repository tugas akhir ditunjukkan oleh Tabel 3.13.

| No | Nama Class  | Keterangan  |
|----|-------------|---|
| 1  | User        | Merupakan Class Model yang digunakan untuk memproses segala data yang berhubungan dengan table User pada Database |
| 2  | User Detail | Merupakan Class Model yang digunakan untuk memproses segala data yang   |

|   |                 |   |
|---|-----------------|---|
|   |                 | berhubungan dengan table User Detail pada Database  |
| 3 | Thesis          | Merupakan Class Model yang digunakan untuk memproses segala data yang berhubungan dengan table Thesis atau tugas akhir pada Database  |
| 4 | Thesis Document | Merupakan Class Model yang digunakan untuk memproses segala data yang berhubungan dengan table Student Thesis Document atau dokumen tugas akhir pada Database   |
| 4 | Departement     | Merupakan Class Model yang digunakan untuk memproses segala data yang berhubungan dengan table Departement atau Jurusan pada Database   |
| 5 | Study           | Merupakan Class Model yang digunakan untuk memproses segala data yang berhubungan dengan table Study atau Program Studi pada Database   |
| 6 | AuthController  | Merupakan Class Controller yang terdiri dari beberapa fungsi yang digunakan untuk menangani autentikasi baik pengguna biasa maupun admin.   |
| 7 | UserController  | Merupakan Class Controller yang terdiri dari fungsi-fungsi yang menangani perubahan data yang dilakukan oleh user dosen untuk menambahkan, mengedit, dan menghapus repositori. Selain itu dapat digunakan untuk menampilkan data yang diperlukan oleh mahasiswa |
| 8 | AdminController | Merupakan Class Controller yang menangani perubahan data dilakukan oleh   |

|  |  |   |
|--|--|---|
|  |  | admin untuk menedit, dan menghapus data User, Repositori, dan data Tugas Akhir. |
|--|--|---|

**Tabel 3. 13 class diagram**

Sumber: Hasil Perancangan

### 3.3.8 Struktur Tabel

#### a. Tabel Users

Tabel *Users* digunakan untuk menyimpan data mahasiswa dan admin yang terdiri dari id, name, email, password, created\_at dan update\_at. Kolom id merupakan *primary key* dari data *user*. Kolom name, email dan password digunakan sebagai tempat menyimpan data autentikasi yang nantinya digunakan untuk *login* ke dalam sistem. Kolom created\_at dan update\_at digunakan untuk menyimpan tanggal serta waktu dalam penambahan maupun update data dari data *user*. Struktur tabel *User* ditunjukkan oleh Tabel *Users*

| Kolom      | Tipe Data | Ukuran | Index       |
|------------|-----------|--------|-------------|
| id         | int       | 20     | Primary key |
| name       | varchar   | 255    |             |
| email      | varchar   | 255    |             |
| password   | varchar   | 255    |             |
| created_at | timestamp |        |             |
| updated_at | timestamp |        |             |
| created_at | timestamp |        |             |

**Tabel 3. 14 Tabel Users**

Sumber: Hasil Perancangan

#### b. Tabel User Details

Tabel *User Details* digunakan untuk menyimpan data mahasiswa yang terdiri dari id, users\_id, unique\_id, department\_id, study\_id, created\_at dan update\_at. Kolom id merupakan *primary key* dari data *user details*. Kolom users\_id merupakan *foreign key* dari tabel *users*, kolom department\_id merupakan *foreign key* dari tabel *department*, dan kolom study\_id merupakan *foreign key* dari tabel *studies*. Kolom unique\_id digunakan untuk menyimpan data NIM dari mahasiswa ke dalam tabel *Users Details*. Kolom created\_at dan update\_at digunakan untuk menyimpan tanggal serta waktu dalam

penambahan maupun update data dari data *user detail*. Struktur tabel *User Details*

| Kolom          | Tipe Data | Ukuran | Index       |
|----------------|-----------|--------|-------------|
| id             | int       | 20     | Primary key |
| unique_id      | varchar   | 255    |             |
| departement_id | int       | 20     | Foreign key |
| study_id       | int       | 20     | Foreign key |
| created_at     | timestamp |        |             |
| updated_at     | timestamp |        |             |

**Tabel 3. 15 Tabel Users Details**

Sumber: Hasil Perancangan

**c. Tabel Studies**

Tabel *Studies* digunakan untuk menyimpan data mahasiswa yang terdiri dari id, study\_name, desc, created\_at dan update\_at. Kolom id merupakan *primary key* dari data *studies*. Kolom study\_name dan desc digunakan sebagai tempat menyimpan data program studi. Kolom created\_at dan update\_at digunakan untuk menyimpan tanggal serta waktu dalam penambahan maupun update data dari data *study*. Struktur tabel *Studies*

| Kolom      | Tipe Data | Ukuran |
|------------|-----------|--------|
| id         | int       | 20     |
| study_name | varchar   | 255    |
| desc       | text      |        |
| created_at | timestamp |        |
| updated_at | timestamp |        |

**Tabel 3. 16 Tabel Studies**

Sumber: Hasil Perancangan

**d. Tabel Departements**

Tabel *Departements* digunakan untuk menyimpan data mahasiswa yang terdiri dari id, departement\_name, desc, created\_at dan update\_at. Kolom id merupakan *primary key* dari data *studies*. Kolom departement\_name dan desc digunakan sebagai tempat menyimpan data jurusan. Kolom created\_at dan

update\_at digunakan untuk menyimpan tanggal serta waktu dalam penambahan maupun update data. Struktur tabel *Departements*

| Kolom            | Tipe Data | Ukuran |
|------------------|-----------|--------|
| id               | int       | 20     |
| departement_name | varchar   | 255    |
| desc             | text      |        |
| created_at       | timestamp |        |
| updated_at       | timestamp |        |

**Tabel 3. 17 Tabel Departements**

Sumber: Hasil Perancangan

**e. Tabel Thesis**

Tabel *Thesis* digunakan untuk menyimpan data mahasiswa yang terdiri dari id, users\_id, thesis\_type, title, abstrak, created\_year, tags, created\_at dan update\_at. Kolom id merupakan *primary key* dari data *Thesis*. Kolom users\_id merupakan *foreign key* dari tabel *users*. Kolom thesis\_type, title, abstrak, created\_year, dan tags digunakan sebagai tempat menyimpan data detail tugas akhir. Kolom created\_at dan update\_at digunakan untuk menyimpan tanggal serta waktu dalam penambahan maupun update data. Struktur tabel *Thesis*

| Kolom        | Tipe Data | Ukuran                    | Index       |
|--------------|-----------|---------------------------|-------------|
| id           | int       | 20                        | Primary key |
| users_id     | int       | 20                        | Foreign key |
| thesis_type  | enum      | ['Tugas Akhir','Skripsi'] |             |
| title        | varchar   | 255                       |             |
| abstrak      | text      |                           |             |
| created_year | integer   | 50                        |             |
| tags         | text      |                           |             |
| created_at   | timestamp |                           |             |
| updated_at   | timestamp |                           |             |

**Tabel 3. 18 Tabel Thesis**

Sumber: Hasil Perancangan

**f. Tabel Thesis Document**

Tabel *Thesis Document* digunakan untuk menyimpan data mahasiswa yang terdiri dari id, thesis\_id, document\_name, document\_url, created\_at, dan update\_at. Kolom id merupakan *primary key* dari data *Thesis*. Kolom thesis\_id merupakan *foreign key* dari tabel *thesis*. Kolom document\_name dan document\_url digunakan sebagai tempat menyimpan data detail dokumen tugas akhir. Kolom created\_at dan update\_at digunakan untuk menyimpan tanggal serta waktu dalam penambahan maupun update data. Struktur tabel *Thesis Document*

| Kolom         | Tipe Data | Ukuran | Index       |
|---------------|-----------|--------|-------------|
| id            | int       | 20     | Primary key |
| thesis_id     | int       | 20     | Foreign key |
| document_name | varchar   | 255    |             |
| document_url  | varchar   | 255    |             |
| created_at    | timestamp |        |             |
| updated_at    | timestamp |        |             |

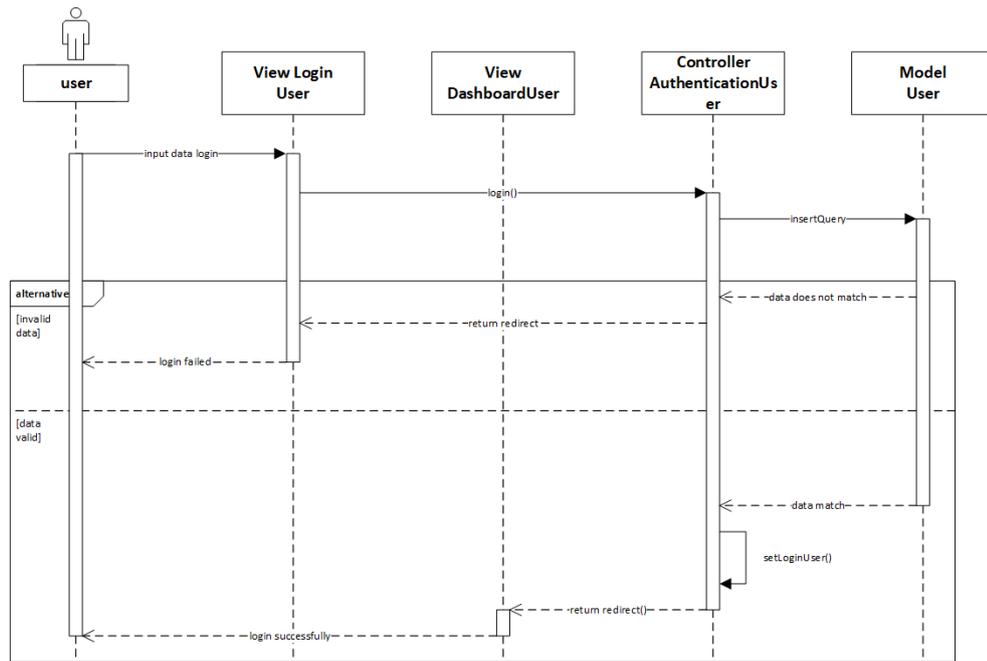
**Tabel 3. 19 Thesis Document**

Sumber: Hasil Perancangan

### 3.3.9 Sequence Diagram

#### a. *Squence Diagram Login*

*Squence diagram login* pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.19.



**Gambar 3. 19 Squence diagram login**

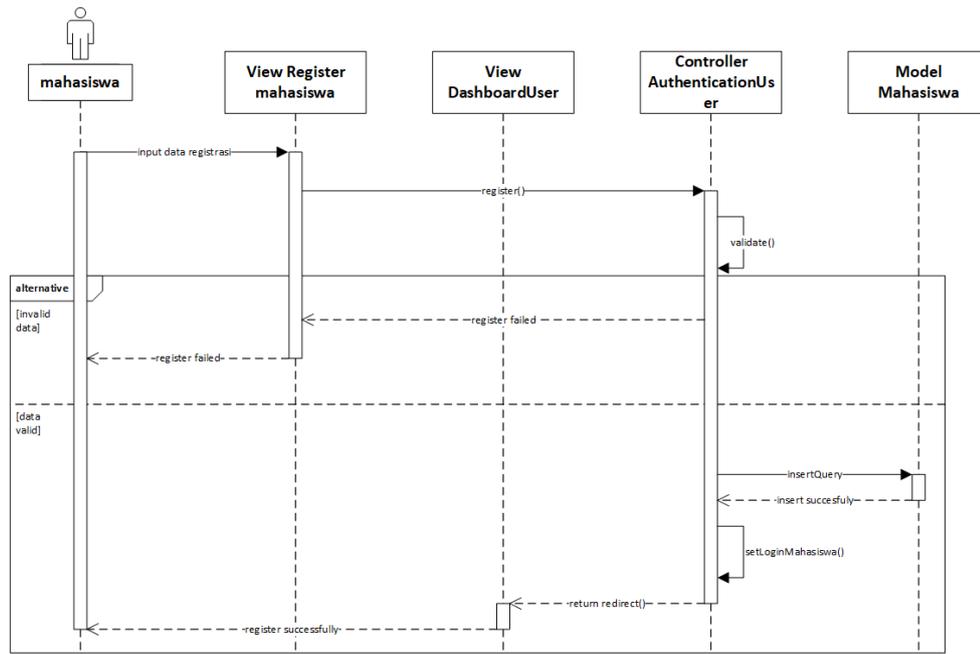
Sumber: Hasil Perancangan

Keterangan:

1. Aktor yang merupakan *user* memasukkan data *login* yang terdiri dari email dan *password* pada halaman *View Login User*.
2. Sistem akan menjalankan fungsi *login()* pada *Controller AuthenticationUser* yang berfungsi untuk mencocokkan data *user* pada *database* melalui *Model User*.
3. Apabila tidak ada data yang cocok, maka sistem akan menampilkan pesan *login failed* pada halaman *View Login User*.
4. Apabila data cocok, maka sistem akan menjalankan fungsi *setLoginUser()* untuk menginformasikan pesan *login successfully* kepada *user* dan mengarahkan *user* yang telah berhasil login untuk masuk ke halaman *View DashboardUser*.

**b. Squence Diagram Registrasi**

*Squence diagram* registrasi pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.20.



**Gambar 3. 20** *Sequence diagram registrasi*

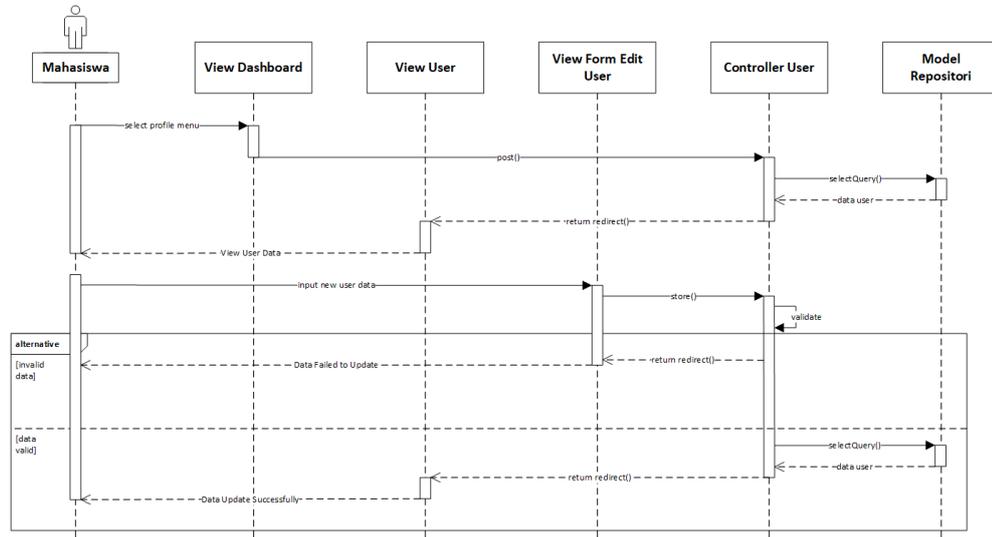
Sumber: Hasil Perancangan

Keterangan :

1. Aktor yang merupakan mahasiswa memasukkan data registrasi yang terdiri dari nama, nomor identitas, email, password, jurusan, dan program studi pada halaman *View Register Mahasiswa*.
2. Sistem kemudian menjalankan fungsi `register()` pada `Controller AuthenticationUser` yang berfungsi untuk mengecek data mahasiswa yang melakukan registrasi melalui fungsi `validate()`.
3. Apabila data tidak valid, sistem akan mengarahkan mahasiswa ke halaman *View Registrasi Mahasiswa* dan menampilkan pesan *register failed*.
4. Apabila data valid, sistem akan menyimpan data registrasi ke dalam *database* melalui `Model Mahasiswa` lalu menjalankan fungsi `setLoginMahasiswa()` untuk menyatakan aktor telah *login* sebagai mahasiswa dan mengarahkan aktor tersebut ke halaman *View Dashboard User*.

c. **Sequence Diagram Mengelola Data Diri**

Sequence diagram data diri pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.21.



**Gambar 3. 21** Sequence diagram data diri

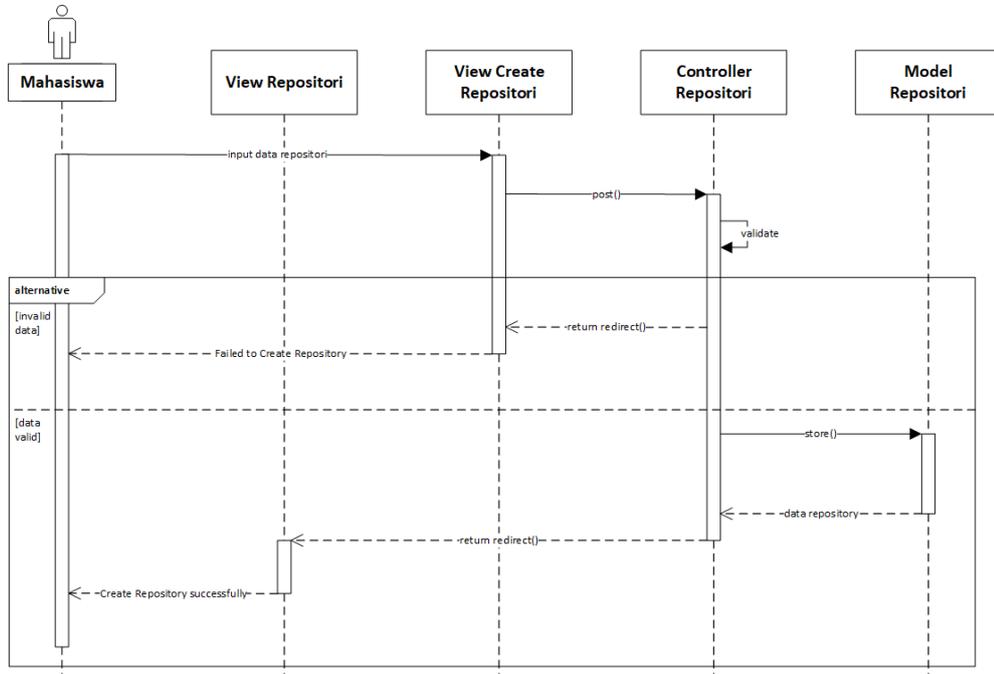
Sumber: Hasil Perancangan

Keterangan :

1. Aktor yang merupakan memilih menu *profile* pada tampilan *View Dashboard User* dan sistem akan menjalankan fungsi *get()* pada *Controller User*.
2. *Controller User* menjalankan fungsi *selectQuery()* untuk mendapatkan data pada *Model User* dan menampilkan data user yang telah dipilih pada tampilan *View Use*.
3. Apabila user ingin melakukan edit data maka user akan diarahkan pada tampilan *View Form Edit User* dan sistem akan menjalankan fungsi *store()* pada *Controller User* serta menjalankan fungsi *validate()*.
4. Apabila data tidak valid, sistem akan mengarahkan mahasiswa ke halaman *View Form Edit User* dan menampilkan pesan *Update data failed*.
5. Apabila data valid, sistem akan menyimpan data user ke dalam *database* melalui *Model Repositori* lalu menampilkan pesan *Successful Update Data* pada tampilan *View User*.

**d. Sequence Diagram Membuat Repositori**

Sequence diagram membuat repositori pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.22.



**Gambar 3. 22 Sequence diagram membuat repositori**

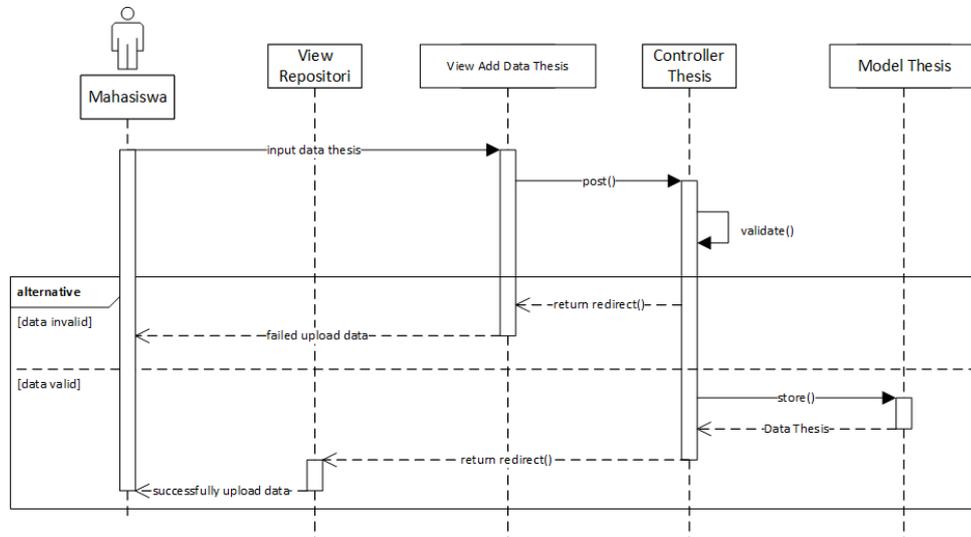
Sumber: Hasil Perancangan

Keterangan :

1. Aktor yang merupakan mahasiswa memasukkan data repositori yang terdiri dari jenis user yang *login* kesistem dan pilihan jenis repositori yang ingin di buat misalnya untuk keperluan Tugas Akhir, tugas akhir pada halaman *View Create Repositori*.
2. Sistem kemudian menjalankan fungsi *post()* pada *Controller Repositori* yang berfungsi untuk mengecek data mahasiswa yang melakukan pembuatan repositori melalui fungsi *validate()*.
3. Apabila data tidak valid, sistem akan mengarahkan mahasiswa ke halaman *View Create Repositori* dan menampilkan pesan *failed to create repository*.
4. Apabila data valid, sistem akan menyimpan data repositori ke dalam *database* melalui *Model Repositori* dan mengarahkan user pada tampilan *View Repositori*.

e. **Sequence Diagram Mengunggah Tugas Akhir**

Sequence diagram mengunggah tugas akhir pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.23.



**Gambar 3. 23** Sequence diagram mengunggah tugas akhir

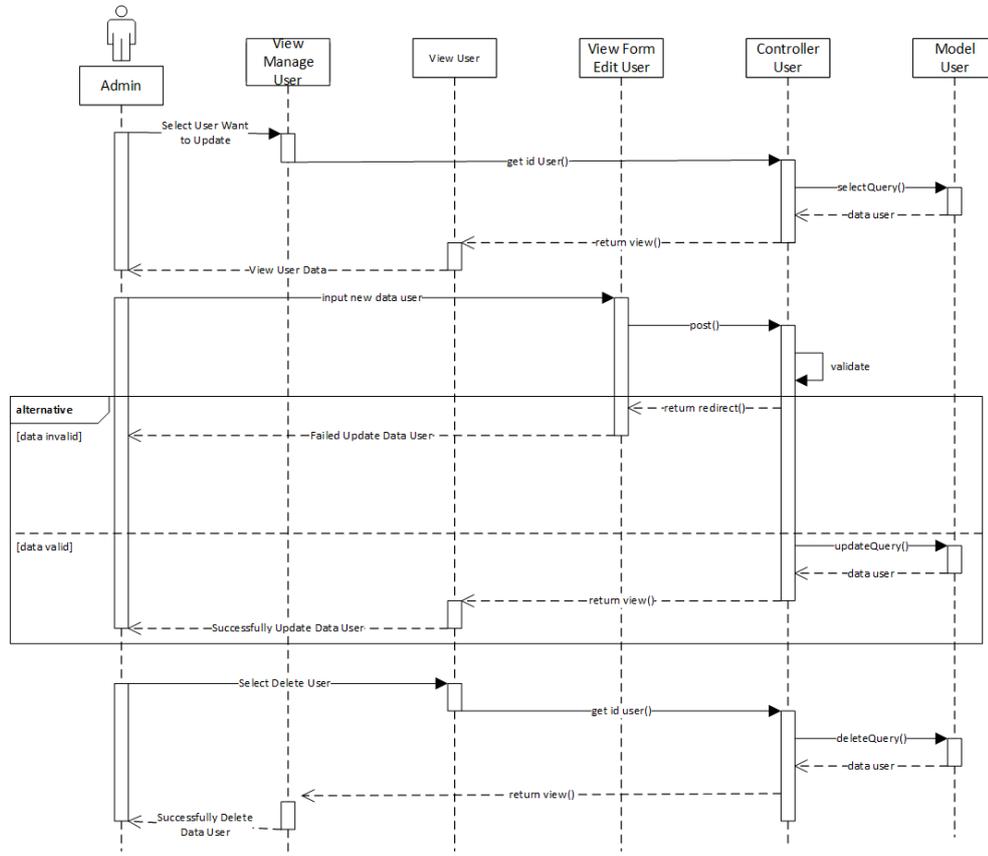
Sumber: Hasil Perancangan

Keterangan :

1. Aktor yang merupakan mahasiswa memasukkan data tugas akhir yang terdiri dari jenis tugas akhir, judul, abstrak, tahun pembuatan, tags, dan file tugas akhir pada halaman *View Add Data Thesis*.
2. Sistem kemudian menjalankan fungsi `post()` pada *Controller Thesis* yang berfungsi untuk mengecek data tugas akhir yang melakukan penambahan data melalui fungsi `validate()`.
3. Apabila data tidak valid, sistem akan mengarahkan mahasiswa ke halaman *View Add Data Thesis* dan menampilkan pesan *failed upload data*.
4. Apabila data valid, sistem akan menjalankan fungsi `store()` untuk menambah data tugas akhir ke dalam *database* melalui *Model Thesis* tugas akhir lalu menampilkan datanya pada tampilan *View Repositori* dan menampilkan pesan *successfully upload data*.

**f. Sequence Diagram Mengelola Data Mahasiswa**

Sequence diagram mengelola data mahasiswa pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.24.



**Gambar 3. 24** Sequence diagram mengelola data mahasiswa

Sumber: Hasil Perancangan

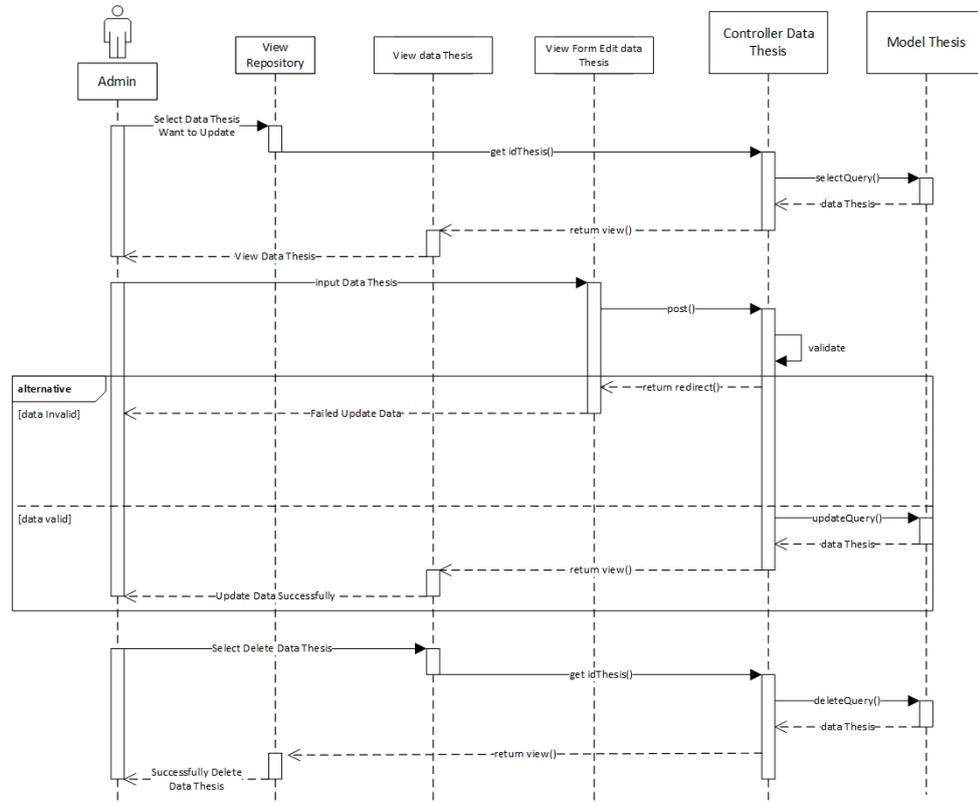
Keterangan :

1. Aktor yang merupakan admin memilih data user yang ingin diedit datanya pada halaman *View Manage User*.
2. Sistem kemudian menjalankan fungsi `getIdUser()` untuk mendapatkan salah satu data *user* pada *Controller User*, lalu menjalankan fungsi `selectQuery()` untuk mendapatkan data pada *Model User* dan menampilkan data user yang telah dipilih pada tampilan *View User*.
3. Kemudian admin melakukan kegiatan edit data pada tampilan *View Form Edit User*.

4. Sistem kemudian menjalankan fungsi `post()` pada Controller User yang berfungsi untuk mengecek data user yang melakukan telah diedit melalui fungsi `validate()`.
5. Apabila data tidak valid, sistem akan mengarahkan mahasiswa ke halaman View Form Edit User dan menampilkan pesan *failed update data*.
6. Apabila data valid, sistem akan menjalankan fungsi `updateQuery()` untuk mengedit data user ke dalam *database* melalui Model User, lalu menampilkan datanya pada tampilan View User dan menampilkan pesan *update data successfully*.
7. Jika admin memilih menghapus data user pada tampilan View User.
8. Sistem kemudian menjalankan fungsi `getIdUser()` untuk mendapatkan salah satu data user pada Controller User.
9. Sistem kemudian menjalankan fungsi `deleteQuery()` pada Controller User yang berfungsi untuk menghapus data user yang telah dipilih, lalu menampilkan datanya pada tampilan View User dan menampilkan pesan *successfully delete data user*.

**g. Sequence Diagram Mengelola Data Tugas Akhir**

Sequence diagram mengelola data tugas akhir pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.25.



**Gambar 3. 25** Sequence diagram mengelola data tugas akhir

Sumber: Hasil Perancangan

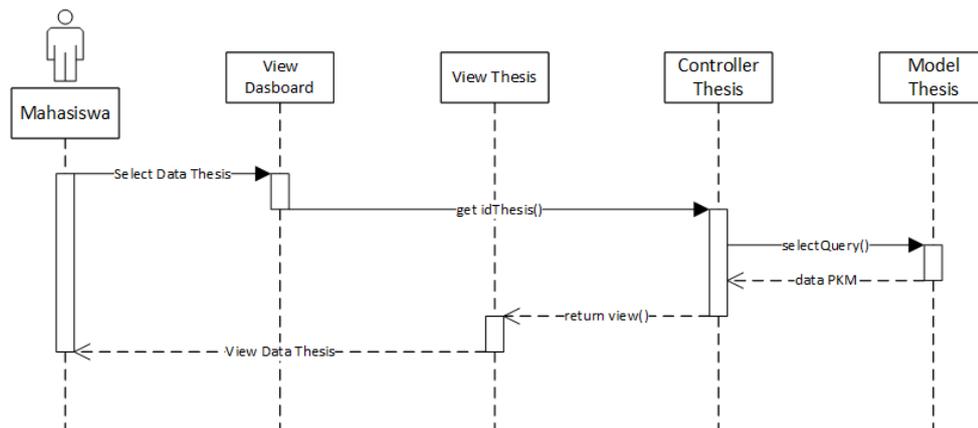
Keterangan :

1. Aktor yang merupakan admin memilih data tugas akhir yang ingin diedit datanya pada halaman *View Repository*.
2. Sistem kemudian menjalankan fungsi *getIdThesis()* untuk mendapatkan salah satu data tugas akhir pada *Controller Data Thesis*, lalu menjalankan fungsi *selectQuery()* untuk mendapatkan data pada *Model Thesis* dan menampilkan data tugas akhir yang telah dipilih pada tampilan *View data Thesis*.
3. Kemudian admin melakukan kegiatan edit data pada tampilan *View Form Edit Data Thesis*.

4. Sistem kemudian menjalankan fungsi *post()* pada *Controller Thesis* yang berfungsi untuk mengecek data tugas akhir yang melakukan telah diedit melalui fungsi *validate()*.
5. Apabila data tidak *valid*, sistem akan mengarahkan mahasiswa ke halaman *View Form Edit Data Thesis* dan menampilkan pesan *failed update data*.
6. Apabila data *valid*, sistem akan menjalankan fungsi *updateQuery()* untuk mengedit data tugas akhir ke dalam *database* melalui Model Data tugas akhir, lalu menampilkan datanya pada tampilan *View Data* tugas akhir dan menampilkan pesan *update data successfully*.
7. Jika admin memilih menghapus data tugas akhir pada tampilan *View* data tugas akhir.
8. Sistem kemudian menjalankan fungsi *getIdThesis()* untuk mendapatkan salah satu data tugas akhir pada *Controller* Data tugas akhir.
9. Sistem kemudian menjalankan fungsi *deleteQuery()* pada *Controller* Data tugas akhir yang berfungsi untuk menghapus data tugas akhir yang telah dipilih, lalu menampilkan datanya pada tampilan *View Data* tugas akhir dan menampilkan pesan *successfully delete* data tugas akhir.

**h. Sequence Diagram Melihat Data Tugas Akhir**

*Sequence diagram* melihat data tugas akhir pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.26.



**Gambar 3. 26 Sequence diagram melihat data tugas akhir**

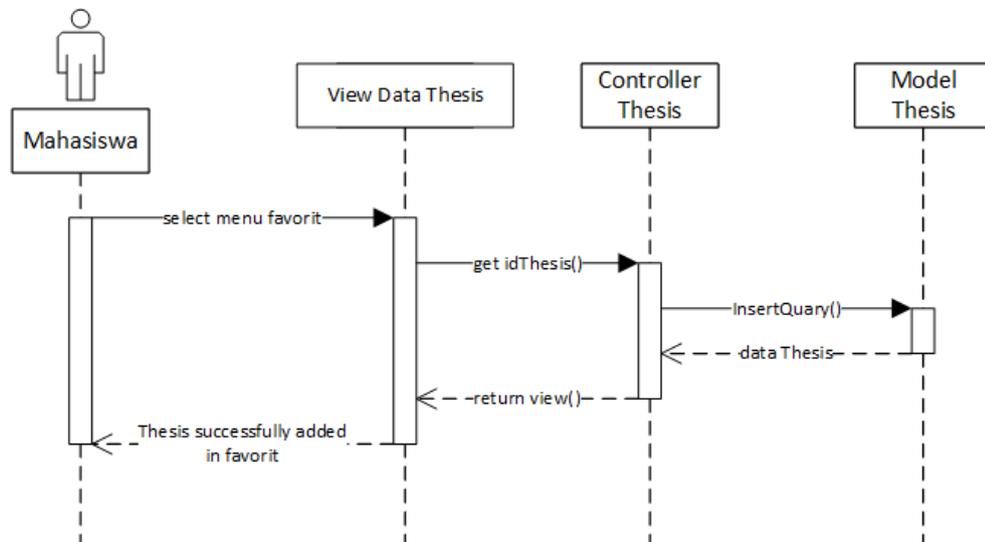
Sumber: Hasil Perancangan

Keterangan:

1. Aktor yang merupakan mahasiswa memilih data tugas akhir yang ingin dilihat datanya pada halaman *View Dashboard*.
2. Sistem kemudian menjalankan fungsi *getIdThesis()* untuk mendapatkan salah satu data tugas akhir pada *Controller* tugas akhir, lalu menjalankan fungsi *selectQuery()* untuk mendapatkan data pada Model *Thesis* dan menampilkan data tugas akhir yang telah dipilih pada tampilan *View Thesis*.

**i. Squence Diagram Menambah Tugas Akhir Favorit**

*Squence diagram* menambah tugas akhir favorit pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.27.



**Gambar 3. 27 Squence diagram menambah tugas akhir favorit**

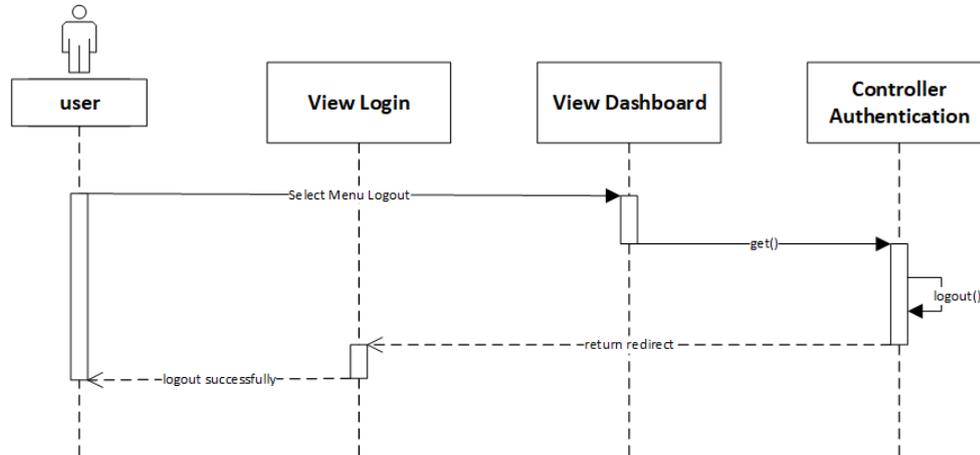
Sumber: Hasil Perancangan

Keterangan:

1. Aktor yang merupakan mahasiswa memilih data tugas akhir yang ingin dilihat datanya pada halaman *View Data Thesis*.
2. Sistem kemudian menjalankan fungsi *getIdThesis()* untuk mendapatkan salah satu data tugas akhir pada *Controller Thesis*, lalu menjalankan fungsi *insertQuery()* untuk menambahkan data favorit pada Model *Thesis* dan menampilkan data tugas akhir yang telah ditambah sebagai favorit pada tampilan *View Data Thesis*.

**j. Sequence Diagram Logout**

Sequence diagram logout favorit pada sistem repositori tugas akhir ditunjukkan oleh Gambar 3.28.



**Gambar 3. 28 Squence diagram logout favorit**

Sumber: Hasil Perancangan

Keterangan:

1. Aktor yang merupakan *user* memilih menu logout pada halaman *View Dashboard*.
2. Sistem kemudian menjalankan fungsi *get()* untuk mendapatkan data *user* pada *Controller Authentication*, lalu menjalankan fungsi *logout()* untuk keluar dari repositori dan mengarahkan user pada tampilan *View Login* serta menampilkan pesan *logout successfully*.

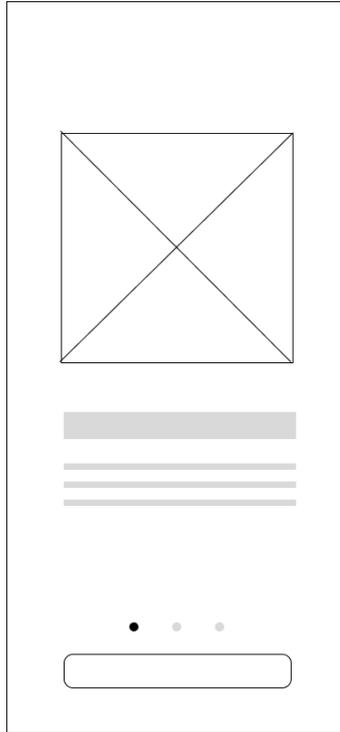
**3.3.10 Perancangan Antarmuka**

**3.3.10.1 Perancangan Antarmuka Mobile**

**a. Antarmuka *Splashscreen***

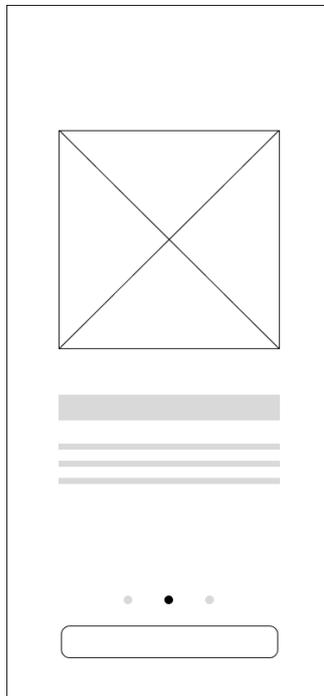
Antarmuka *splashscreen* akan digunakan saat pertama kali membuka aplikasi untuk pertama kalinya. Rancangan antarmuka *splashscreen* memiliki 4 bagian pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.29, gambar 3.30, dan gambar 3.31.

Sumber: Hasil Perancangan



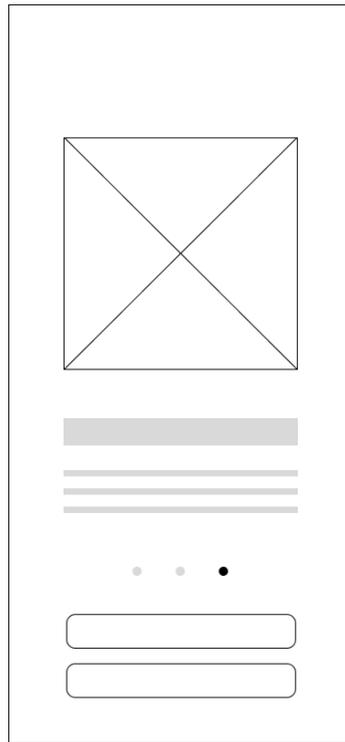
**Gambar 3. 29** Antarmuka *Splashscreen* Kedua

Sumber: Hasil Perancangan



**Gambar 3. 30** Antarmuka *Splashscreen* Ketiga

Sumber: Hasil Perancangan

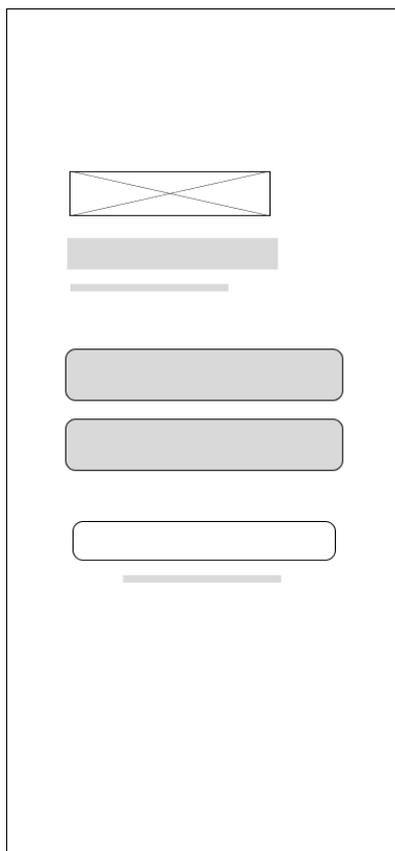


**Gambar 3. 31 Antarmuka *Splashscreen* Keempat**

Sumber: Hasil Perancangan

**b. Antarmuka *Login***

Antarmuka *login* pelanggan digunakan untuk menerima *input* data *login* dari pengguna yang ingin masuk ke dalam sistem. Rancangan antarmuka *login* pelanggan pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.32.

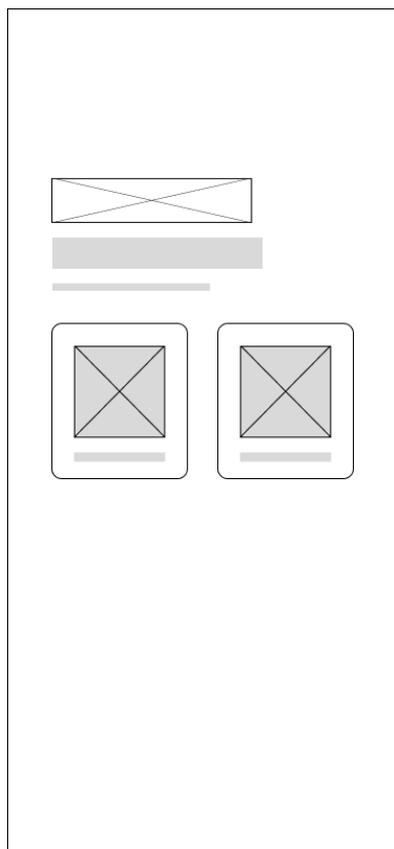


***Gambar 3. 32 Antarmuka Login***

Sumber: Hasil Perancangan

**c. Antarmuka Registrasi Akun Repositori**

Antarmuka registrasi akun repositori digunakan untuk menerima input data pilihan untuk memilih jenis akun repositori dari pengguna. Rancangan antarmuka registrasi akun repositori pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.33.

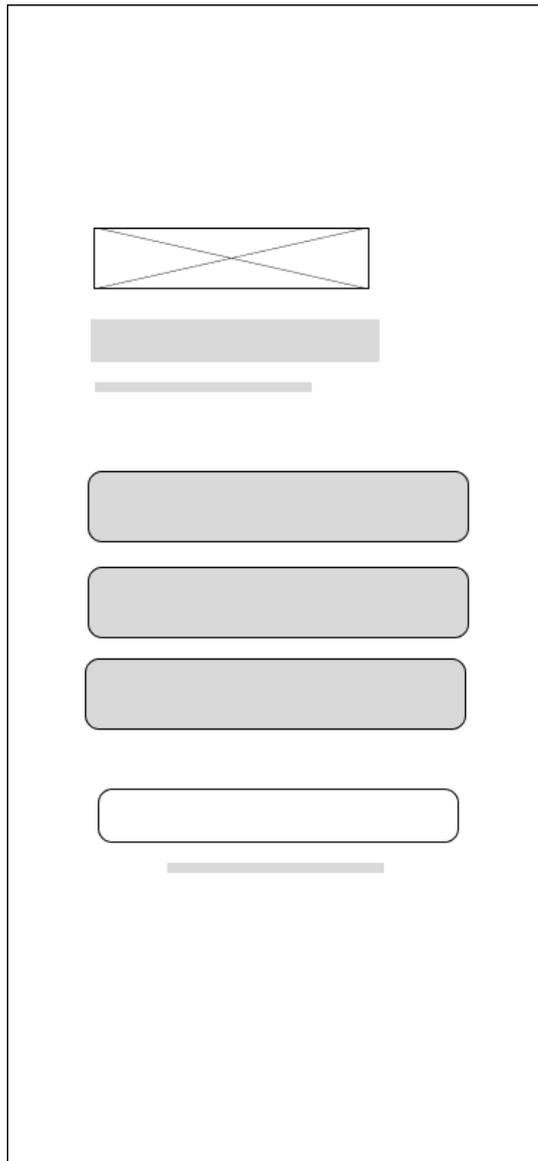


**Gambar 3. 33 Antarmuka Registrasi Akun Repositori**

Sumber: Hasil Perancangan

**d. Antarmuka *Form* Registrasi**

Antarmuka *form* registrasi digunakan untuk menerima input data registrasi dari pengguna. Rancangan antarmuka *form* registrasi pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.34.



**Gambar 3. 34 Antarmuka Registrasi**

Sumber: Hasil Perancangan

**e. Antarmuka Tampilan *Dashboard***

Antarmuka tampilan *dashboard* digunakan untuk menampilkan tampilan awal setelah berhasil melakukan *login*. Rancangan antarmuka tampilan *dashboard* pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.35.

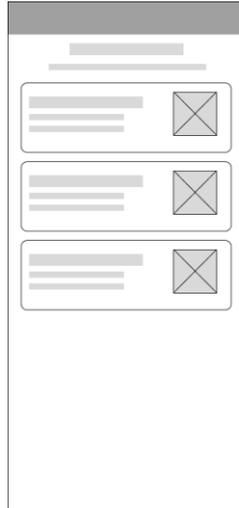


**Gambar 3. 35 Antarmuka Tampilan Dashboard**

Sumber: Hasil Perancangan

**f. Antarmuka Pencarian Data Tugas Akhir**

Antarmuka pencarian data tugas akhir merupakan halaman yang muncul ketika *user* ingin mencari repositori tugas akhir pada sistem melalui aplikasi *mobile*. Rancangan antarmuka pencarian data tugas akhir pada sistem repositori tugas akhir Menggunakan ditunjukkan pada Gambar 3.36, Gambar 3.37, dan Gambar 3.38.



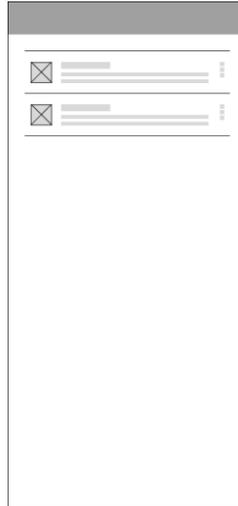
***Gambar 3. 36 Antarmuka pencarian data tugas akhir pertama***

Sumber: Hasil Perancangan



***Gambar 3. 37 Antarmuka pencarian data tugas akhir kedua***

Sumber: Hasil Perancangan

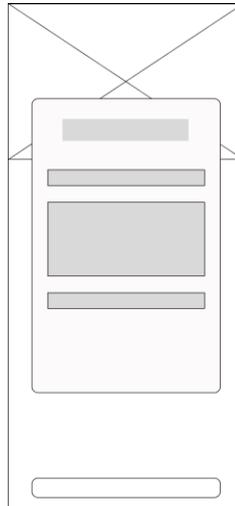


**Gambar 3. 38 Antarmuka pencarian data tugas akhir ketiga**

Sumber: Hasil Perancangan

**g. Antarmuka Membuat Repositori**

Antarmuka membuat repositori digunakan untuk menerima input data repositori dari pengguna. Rancangan antarmuka membuat repositori pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.39.

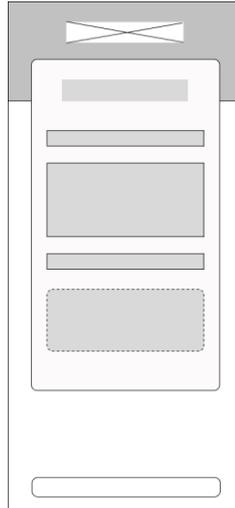


**Gambar 3. 39 Antarmuka membuat repositori**

Sumber: Hasil Perancangan

**h. Antarmuka Mengunggah Tugas Akhir**

Antarmuka mengunggah tugas akhir digunakan untuk menerima input data tugas akhir dari pengguna. Rancangan antarmuka mengunggah tugas akhir pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.40.



**Gambar 3. 40 Antarmuka mengunggah tugas akhir**

Sumber: Hasil Perancangan

**i. Antarmuka Tampilan Sub Menu**

Antarmuka tampilan sub menu digunakan untuk menampilkan data repositori dari pengguna. Rancangan antarmuka registrasi mahasiswa pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.41.

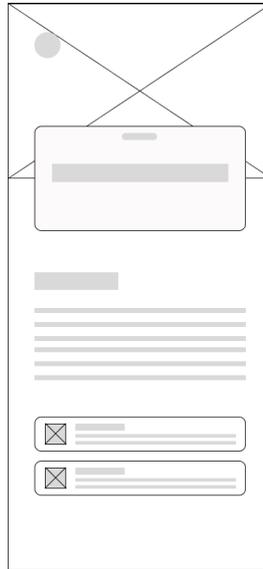


**Gambar 3. 41 Antarmuka tampilan sub menu**

Sumber: Hasil Perancangan

**j. Antarmuka Tampilan Data Detail Tugas Akhir**

Antarmuka tampilan data detail tugas akhir digunakan untuk menampilkan data detail tugas akhir dari pengguna. Rancangan antarmuka tampilan data detail tugas akhir pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.42.



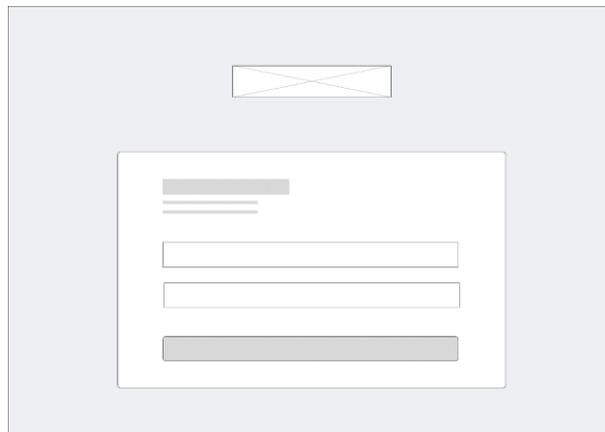
**Gambar 3. 42 Antarmuka tampilan data detail tugas akhir**

Sumber: Hasil Perancangan

**3.3.10.2 Perancangan Antarmuka Web**

**a. Antarmuka *Login***

Antarmuka *login* pelanggan digunakan untuk menerima *input* data *login* dari pengguna yang ingin masuk ke dalam sistem. Rancangan antarmuka *login* pelanggan pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.43.

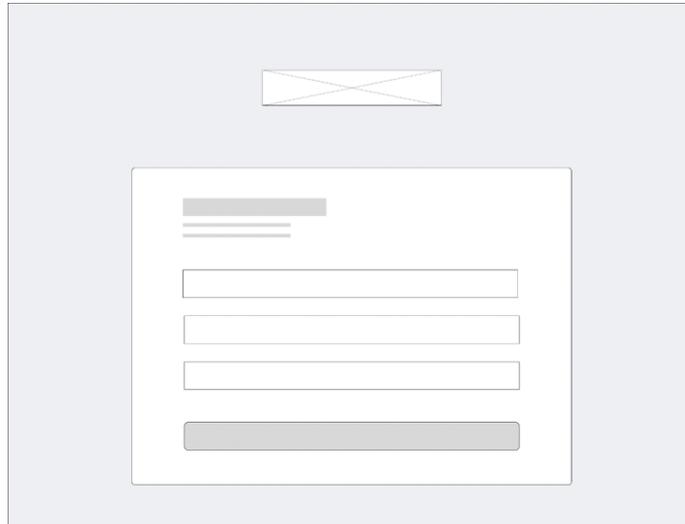


**Gambar 3. 43 Antarmuka *login***

Sumber: Hasil Perancangan

**b. Antarmuka *Form* Registrasi**

Antarmuka *form* registrasi digunakan untuk menerima input data registrasi dari pengguna. Rancangan antarmuka *form* registrasi pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.44.

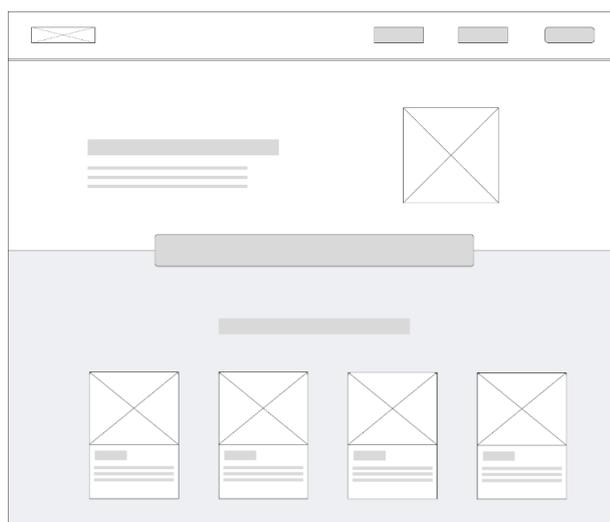


**Gambar 3. 44 Antarmuka form registrasi**

Sumber: Hasil Perancangan

**c. Antarmuka Tampilan *Landing Page***

Antarmuka tampilan *landing page* digunakan untuk menampilkan tampilan awal setelah berhasil melakukan *login*. Rancangan antarmuka tampilan *landing page* pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.45.

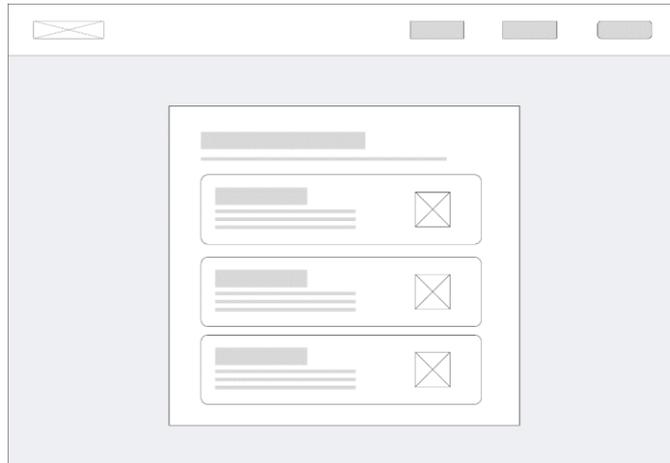


**Gambar 3. 45 Antarmuka tampilan landing page**

Sumber: Hasil Perancangan

**d. Antarmuka Tampilan Memilih Repositori**

Antarmuka tampilan memilih repositori digunakan untuk menerima pilihan repositori yang ingin dibuat oleh pengguna. Rancangan antarmuka tampilan memilih repositori pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.46.



**Gambar 3. 46 Antarmuka tampilan memilih repositori**

Sumber: Hasil Perancangan

**e. Antarmuka Membuat Repositori Tugas Akhir**

Antarmuka membuat repositori digunakan untuk menerima input data repositori dari pengguna. Rancangan antarmuka membuat repositori tugas akhir pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.47.

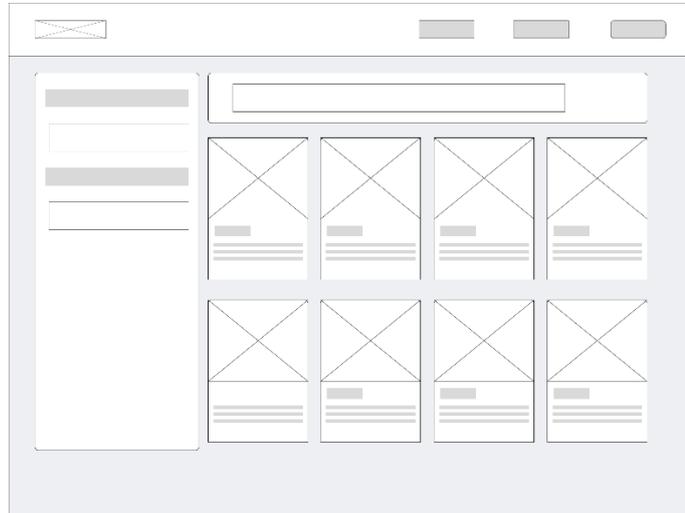


**Gambar 3. 47 Antarmuka membuat repositori tugas akhir**

Sumber: Hasil Perancangan

**f. Antarmuka Pencarian Tugas Akhir**

Antarmuka pencarian tugas akhir merupakan halaman yang muncul ketika user ingin mencari repositori tugas akhir. Rancangan antarmuka pencarian tugas akhir pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.48.

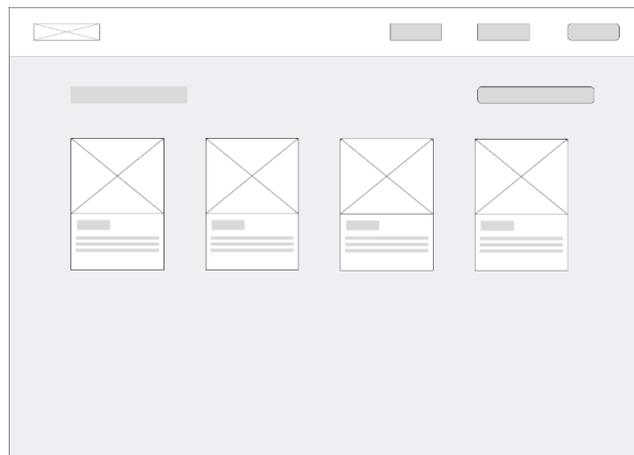


**Gambar 3. 48 Antarmuka pencarian tugas akhir**

Sumber: Hasil Perancangan

**g. Antarmuka Tampilan Repositori Tugas Akhir**

Antarmuka tampilan repositori tugas akhir digunakan untuk menampilkan repositori yang telah berhasil dibuat oleh pengguna. Rancangan antarmuka repositori tugas akhir pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.49.

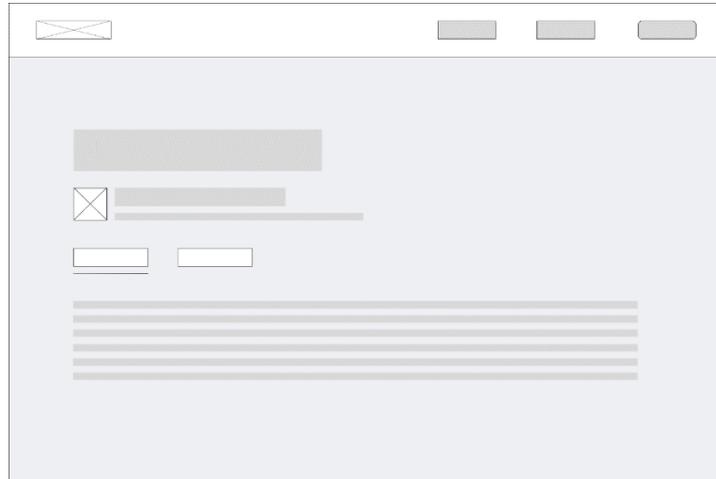


**Gambar 3. 49 Antarmuka tampilan repositori tugas akhir**

Sumber: Hasil Perancangan

**h. Antarmuka Tampilan Data Detail Tugas Akhir**

Antarmuka tampilan data detail tugas akhir digunakan untuk menampilkan data detail tugas akhir dari pengguna. Rancangan antarmuka tampilan data detail tugas akhir pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.50.

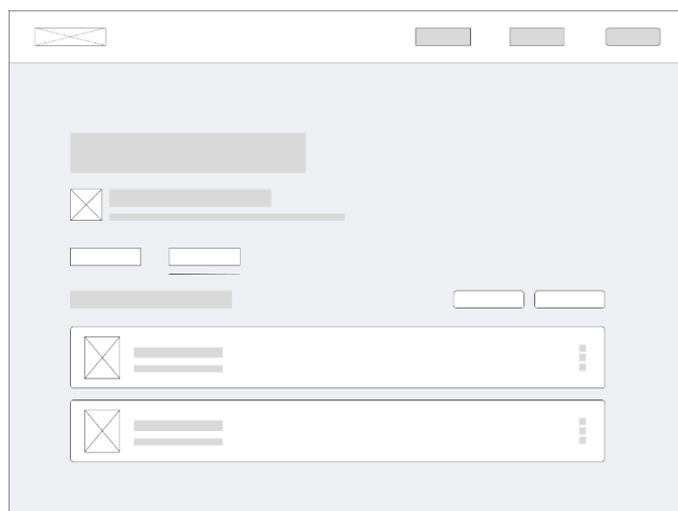


**Gambar 3. 50 Antarmuka tampilan data detail tugas akhir**

Sumber: Hasil Perancangan

**i. Antarmuka Tampilan Data Detail Dokumen Tugas Akhir**

Antarmuka tampilan data detail dokumen tugas akhir digunakan untuk menampilkan data detail dokumen tugas akhir dari pengguna. Rancangan antarmuka tampilan data detail dokument tugas akhir pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.51.

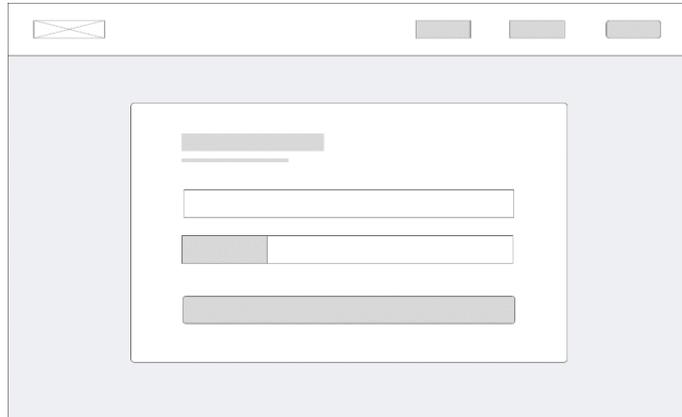


**Gambar 3. 51 Antarmuka tampilan data detail dokumen tugas akhir**

Sumber: Hasil Perancangan

**j. Antarmuka Mengunggah Tugas Akhir**

Antarmuka mengunggah tugas akhir digunakan untuk menerima input data tugas akhir dari pengguna. Rancangan antarmuka mengunggah tugas akhir pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.52.

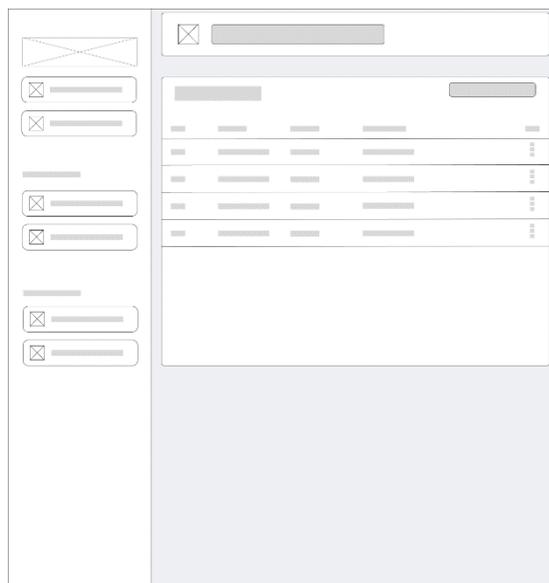


**Gambar 3. 52 Antarmuka mengunggah tugas akhir**

Sumber: Hasil Perancangan

**k. Antarmuka Kelola Data Diri**

Antarmuka kelola data diri digunakan untuk menampilkan detail data diri dari pengguna. Rancangan antarmuka Kelola data diri pada sistem repositori tugas akhir ditunjukkan pada Gambar 3.53.



**Gambar 3. 53 Antarmuka kelola data diri**

Sumber: Hasil Perancangan

## BAB IV

### IMPLEMENTASI SISTEM

#### 4.1 Perangkat Kebutuhan Implementasi

##### 4.1.1 Spesifikasi Perangkat Keras

Adapun perangkat keras yang digunakan untuk mengimplementasikan sistem repositori tugas akhir dapat dilihat pada Tabel 4.1 dibawah ini :

| No | Perangkat Keras | Keterangan                 |
|----|-----------------|----------------------------|
| 1  | Prosesor        | AMD Ryzen 5 4600H          |
| 2  | RAM             | 16 <i>GigaByte</i>         |
| 3  | VGA             | NVIDIA GeForce GTX 1650 TI |
| 4  | Penyimpanan     | 512 <i>GigaByte</i>        |
| 5  | Display         | 15.6 <i>inch</i>           |

***Tabel 4. 1 Spesifikasi Perangkat Keras***

Sumber: Hasil Perancangan

##### 4.1.2 Spesifikasi Perangkat Lunak

Adapun perangkat lunak yang digunakan untuk mengimplementasikan sistem repositori tugas akhir dapat dilihat pada Tabel 4.2 dibawah ini :

| No | Perangkat Lunak            | Keterangan                      |
|----|----------------------------|---------------------------------|
| 1  | Sistem Operasi             | Microsoft Windows 11            |
| 2  | <i>Text Editor</i>         | <i>Visual Studio Code</i>       |
| 3  | <i>Aplikasi Web Server</i> | XAMPP Versi 3.3.0               |
| 4  | Bahasa Pemrograman         | Dart, PHP dan <i>Javascript</i> |
| 5  | Basis Data                 | MySQL                           |
| 6  | <i>Web Browser</i>         | Google Chrome                   |

***Tabel 4. 2 Spesifikasi Perangkat Lunak***

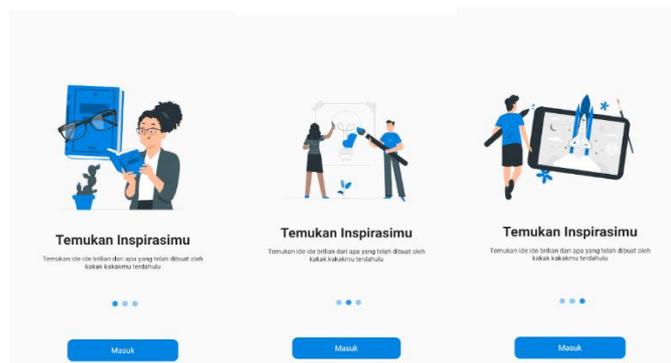
Sumber: Hasil Perancangan

## 4.2 Pengujian Sistem

### 4.2.1 Pengujian Sistem *Mobile*

#### a. Halaman *Splashscreen*

Pengujian dilakukan dengan membuka aplikasi pertama kali, maka tampilan yang muncul pertama kali adalah tampilan *splashscreen* yang memberikan informasi singkat pada aplikasi PNB *repository*. Halaman *splashscreen* pada sistem repositori Tugas Akhir ditunjukkan pada gambar 4.1.



**Gambar 4. 1 Tampilan *Splashscreen***

Sumber: Hasil Pengujian

#### b. Halaman *Login*

Pengujian dilakukan dengan mengakses halaman *login* setelah menekan tombol masuk, maka tampilan yang muncul pertama kali adalah tampilan *form login* yang berisikan kolom email dan *password*. Halaman login pada sistem repositori Tugas Akhir ditunjukkan pada gambar 4.2.



**Gambar 4. 2 Halaman *Login***

Sumber: Hasil Pengujian

### c. Halaman Memilih Akun Repositori

Pengujian dilakukan dengan mengakses halaman memilih akun setelah menekan tombol daftar sekarang. Pengguna diminta untuk memilih akun mahasiswa atau dosen untuk menentukan repositori yang akan ditampilkan pada aplikasi. Halaman memilih akun repositori pada sistem repositori tugas akhir ditunjukkan pada gambar 4.3.



**Gambar 4. 3 Halaman Memilih Akun Repositori**

Sumber: Hasil Perancangan

### d. Halaman Registrasi

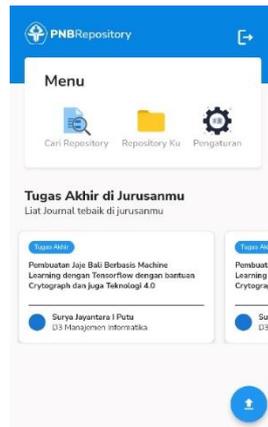
Pengujian dilakukan dengan mengakses halaman registrasi setelah memilih akun repositori, maka tampilan yang akan muncul adalah *form* registrasi yang berisikan kolom nama, email, nomor identitas, password, jurusan dan program studi. Halaman registrasi pada sistem repositori tugas akhir ditunjukkan pada gambar 4.4.

**Gambar 4. 4 Halaman Registrasi**

Sumber: Hasil Perancangan

### e. Halaman Dashboard

Pengujian dilakukan dengan mengakses halaman *dashboard* setelah berhasil melakukan *login* atau registrasi, maka tampilan yang akan dimunculkan adalah tampilan data PKM dan tugas akhir apabila *login* dengan akun mahasiswa. Halaman *dashboard* pada sistem repositori tugas akhir ditunjukkan pada gambar 4.5.

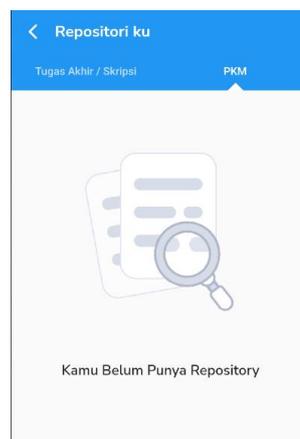


**Gambar 4. 5 Halaman Dashboard**

Sumber: Hasil Perancangan

### f. Halaman Repositori Tugas Akhir Tanpa Data

Pengujian dilakukan dengan mengakses halaman repositori tugas akhir tanpa data setelah menekan tombol repositoriku, maka akan menampilkan *icon* serta tulisan kamu belum punya *repository*. Halaman repositori tugas akhir tanpa data pada sistem repositori tugas akhir ditunjukkan pada gambar 4.6.

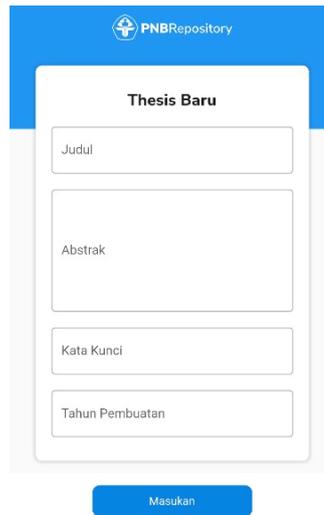


**Gambar 4. 6 Halaman Repositori Tugas Akhir Tanpa Data**

Sumber: Hasil Perancangan

#### **g. Halaman Membuat Repositori Tugas Akhir**

Pengujian dilakukan dengan mengakses halaman membuat repositori tugas akhir setelah menekan tombol tambah, maka akan menampilkan *form* pendaftaran pembuatan repositori tugas akhir yang berisi kolom judul, abstrak, kata kunci, dan tahun pembuatan. Halaman membuat repositori tugas akhir sistem repositori tugas akhir ditunjukkan pada gambar 4.7.

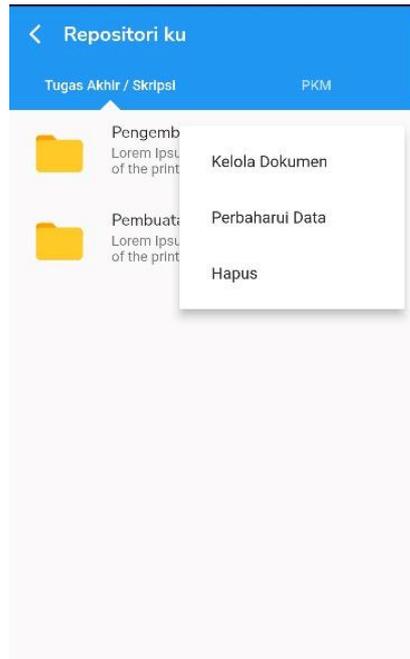


***Gambar 4. 7 Halaman Membuat Repositori Tugas Akhir***

Sumber: Hasil Perancangan

#### **h. Halaman Repositori Tugas Akhir Berisi Data**

Pengujian dilakukan dengan mengakses halaman repositori tugas akhir berisi data setelah berhasil melakukan penambahan data tugas akhir, maka akan menampilkan list data dari data tugas akhir yang telah ditambah. Halaman repositori tugas akhir berisi data pada sistem repositori tugas akhir ditunjukkan pada gambar 4.8.

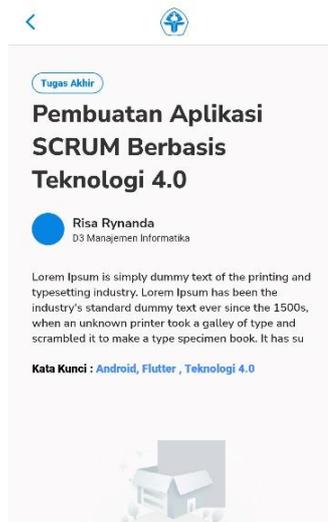


**Gambar 4. 8 Halaman Repositori Tugas Akhir Berisi Data**

Sumber: Hasil Perancangan

**i. Halaman Melihat Detail Data Tugas Akhir**

Pengujian dilakukan dengan mengakses halaman melihat detail data tugas akhir setelah memilih data pada *dashboard* atau dari repositori, maka akan menampilkan data detail tugas akhir. Halaman melihat detail data tugas akhir pada sistem repositori tugas akhir ditunjukkan pada gambar 4.9.

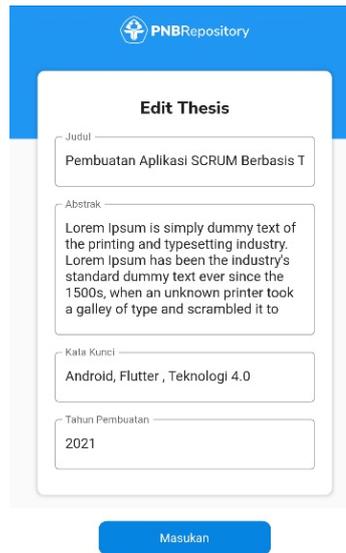


**Gambar 4. 9 Halaman Melihat Detail Data Tugas Akhir**

Sumber: Hasil Perancangan

#### j. Halaman Mengedit Data Tugas Akhir

Pengujian dilakukan dengan mengakses halaman mengedit data setelah menekan tombol edit data, maka akan menampilkan *form* yang telah berisi data pada kolom judul, abstrak, kata kunci, dan tahun pembuatan. Halaman mengedit data tugas akhir pada sistem repositori tugas akhir ditunjukkan pada gambar 4.10.



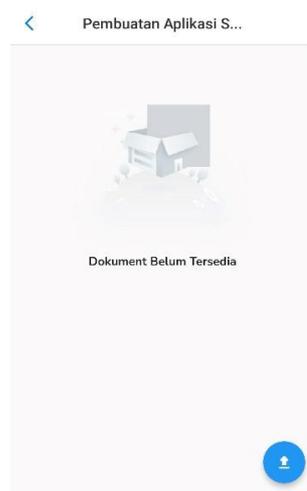
The image shows a mobile application interface for editing a thesis. At the top, there is a blue header with the PNBRepository logo and name. Below the header is a white card titled 'Edit Thesis'. The card contains four input fields: 'Judul' (Title) with the value 'Pembuatan Aplikasi SCRUM Berbasis T', 'Abstrak' (Abstract) with a paragraph of Lorem Ipsum text, 'Kata Kunci' (Keywords) with the value 'Android, Flutter, Teknologi 4.0', and 'Tahun Pembuatan' (Year of Creation) with the value '2021'. At the bottom of the card is a blue button labeled 'Masukan' (Submit).

**Gambar 4. 10 Halaman Mengedit Data Tugas Akhir**

Sumber: Hasil Perancangan

#### k. Halaman Kelola Dokumen Tugas Akhir

Pengujian dilakukan dengan mengakses halaman kelola dokumen tugas akhir dengan memilih menu kelola dokumen pada *dropdown* data repositori tugas akhir. Halaman dokumen tugas akhir sistem repositori tugas akhir ditunjukkan pada Gambar 4.11.



**Gambar 4. 11 Halaman Kelola Dokumen Tugas Akhir**

Sumber: Hasil Perancangan

### **I. Halaman Tambah Dokumen Tugas Akhir**

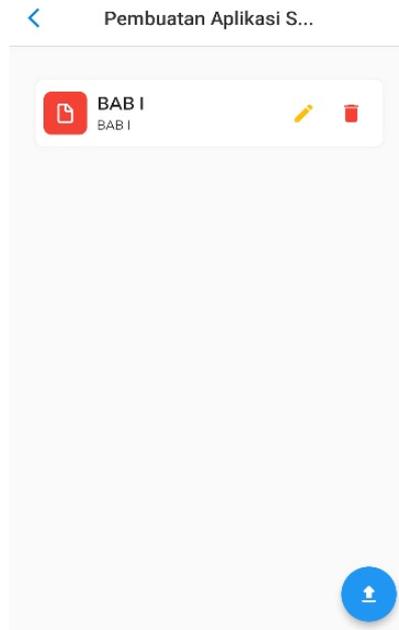
Pengujian dilakukan dengan mengakses halaman tambah dokumen tugas akhir setelah menekan tombol tambah dokumen, maka akan menampilkan *form* pendaftaran tambah dokumen tugas akhir yang berisi kolom judul, abstrak, kata kunci, dan tahun pembuatan. Halaman tambah dokumen tugas akhir sistem repositori tugas akhir ditunjukkan pada gambar 4.10.



**Gambar 4. 12 Halaman Tambah Dokumen Tugas Akhir**

Sumber: Hasil Perancangan

Apabila berhasil menambahkan data, maka pengguna diarahkan kembali pada tampilan kelola dokumen tugas akhir. Halaman data kelola dokumen sistem repositori tugas akhir ditunjukkan pada Gambar 4.13.

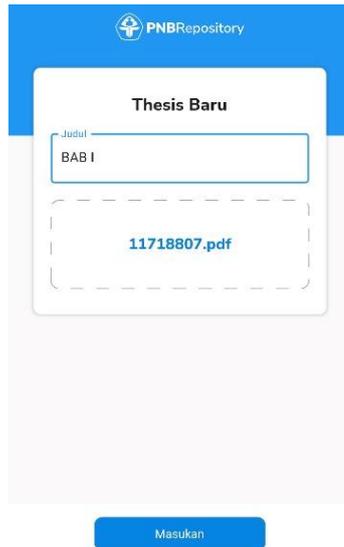


**Gambar 4. 13**Tampilan kelola dokumen tugas akhir

Sumber: Hasil Perancangan

#### **m. Halaman Edit Dokumen Tugas Akhir**

Pengujian dilakukan dengan mengakses halaman mengedit dokumen tugas akhir setelah menekan tombol *icon* edit data, maka akan menampilkan *form* yang telah berisi data pada kolom judul, abstrak, kata kunci, dan tahun pembuatan. Halaman mengedit dokumen tugas akhir pada sistem repositori tugas akhir ditunjukkan pada gambar 4.13.

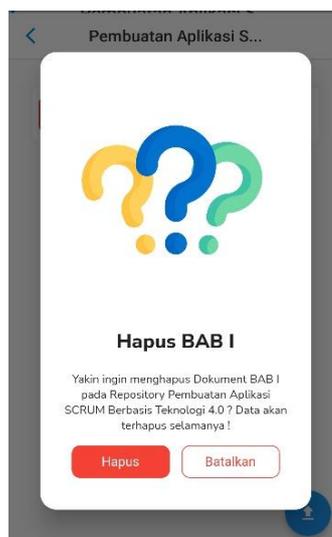


**Gambar 4. 14 Halaman Edit Dokumen Tugas Akhir**

Sumber: Hasil Perancangan

#### **n. Halaman Menghapus Dokumen Tugas Akhir**

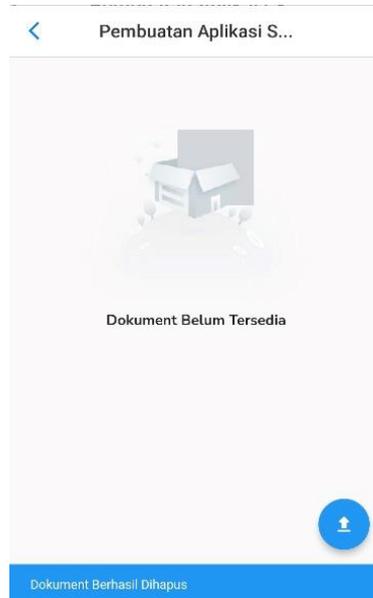
Pengujian dilakukan dengan mengakses halaman menghapus dokumen tugas akhir setelah menekan tombol hapus data, maka akan menampilkan tampilan pertanyaan ingin menghapus data atau tidak dan jika memilih hapus maka dokumen tugas akhir tersebut akan terhapus. Halaman menghapus dokumen tugas ditunjukkan pada gambar 4.15.



**Gambar 4. 15 Halaman Menghapus Dokumen Tugas Akhir**

Sumber: Hasil Perancangan

Apabila berhasil menghapus data dokumen tugas akhir, maka pengguna akan diarahkan kembali pada halaman kelola dokumen tugas akhir dengan pesan dokumen berhasil dihapus. Halaman berhasil hapus data dokumen tugas akhir ditunjukkan pada Gambar 4.16.



***Gambar 4. 16 Halaman kelola dokumen tugas akhir***

Sumber: Hasil Perancangan

**o. Halaman Mencari Data Tugas Akhir**

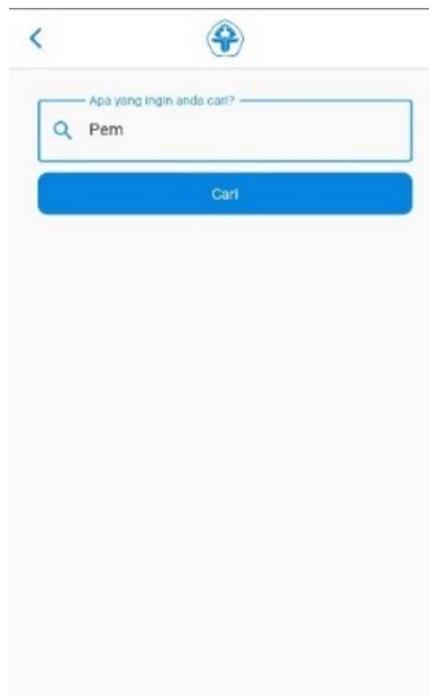
Pengujian dilakukan dengan mengakses halaman mencari data tugas akhir setelah menekan tombol search dan pengguna diminta untuk memilih data yang ingin dicari. Halaman mencari data tugas akhir ditunjukkan pada Gambar 4.17.



**Gambar 4. 17 Halaman Mencari Data Tugas Akhir**

Sumber: Hasil Perancangan

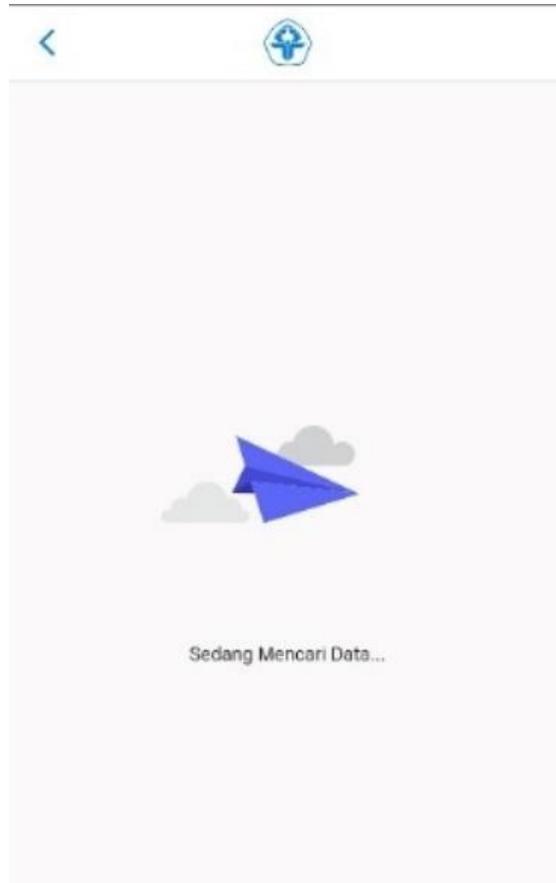
Setelah memilih data yang ingin dicari, selanjutnya masukan judul tugas akhir yang ingin dicari dan tekan tombol cari. Halaman pencarian tugas akhir ditunjukkan pada Gambar 4.18.



**Gambar 4. 18 Halaman pencarian tugas akhir**

Sumber: Hasil Perancangan

Apabila tombol cari sudah ditekan, maka sistem akan menampilkan *icon* sedang mencari data. Tampilan *icon* mencari data ditunjukkan pada Gambar 4.19.



***Gambar 4. 19 Tampilan icon mencari data***

Sumber: Hasil Perancangan

Jika sistem telah menemukan data yang dicari, maka sistem akan menampilkan halaman tugas akhir sesuai dengan hasil dari pencarian. Halaman hasil pencarian tugas akhir ditunjukkan pada Gambar 4.20.



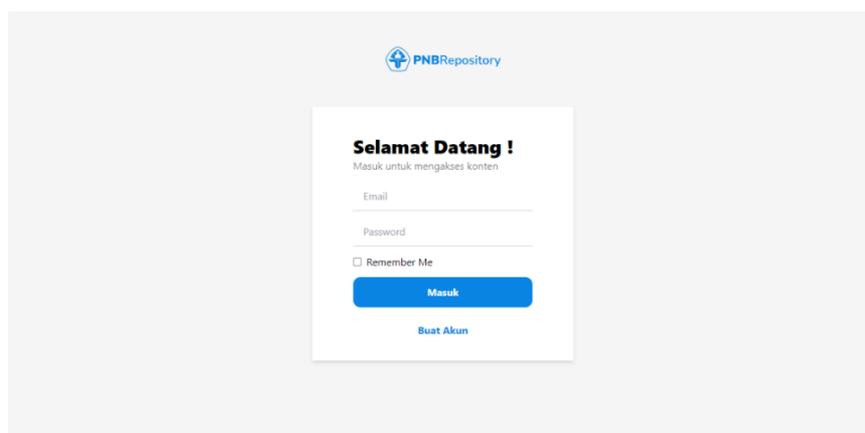
**Gambar 4. 20 Halaman hasil pencarian tugas akhir**

Sumber: Hasil Perancangan

#### **4.2.2 Pengujian Sistem Website**

##### **a. Halaman Login**

Pengujian dilakukan dengan membuka situs website pertama kali, maka tampilan yang muncul pertama kali adalah tampilan login. Halaman login pada sistem repositori tugas akhir ditunjukkan pada gambar 4.21.

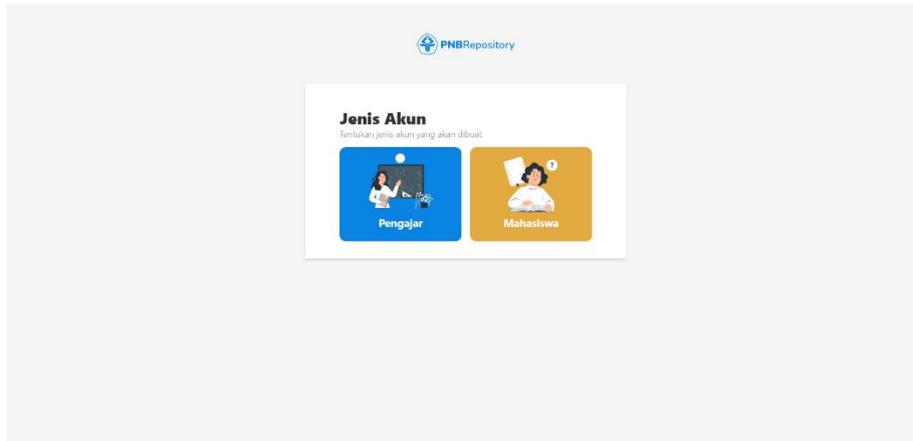


**Gambar 4. 21 Halaman Login**

Sumber: Hasil Perancangan

**b. Halaman Memilih Akun Repositori**

Pengujian dilakukan dengan mengakses halaman memilih akun setelah menekan tombol buat akun. Pengguna diminta untuk memilih akun mahasiswa atau dosen untuk menentukan repositori yang akan ditampilkan pada *website*. Halaman memilih akun repositori sistem repositori tugas akhir ditunjukkan pada gambar 4.22.

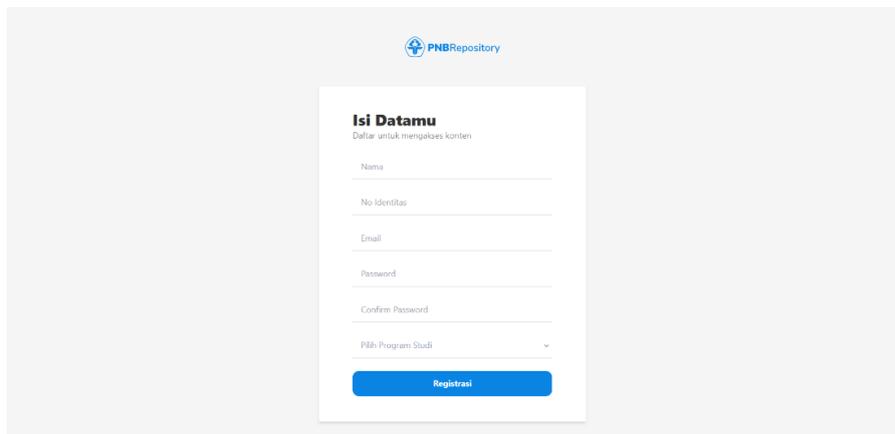


**Gambar 4. 22 Halaman Memilih Akun Repositori**

Sumber: Hasil Perancangan

**c. Halaman Registrasi**

Pengujian dilakukan dengan mengakses halaman registrasi dengan memilih tombol buat akun pada halaman *login*. Sistem akan menampilkan form registrasi dengan kolom nama, email, nomor identitas, password, jurusan dan program studi yang ditunjukkan pada gambar 4.23.

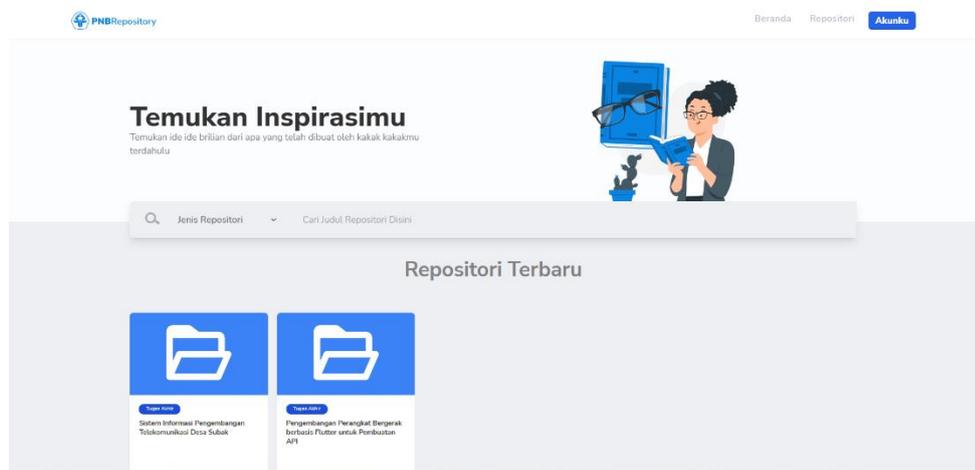


**Gambar 4. 23 Halaman Registrasi**

Sumber: Hasil Perancangan

#### d. Halaman Utama

Pengujian dilakukan dengan mengakses halaman utama yang akan ditampilkan apabila user berhasil melakukan registrasi atau *login*. Halaman utama sistem repositori tugas akhir ditunjukkan pada gambar 4.24.

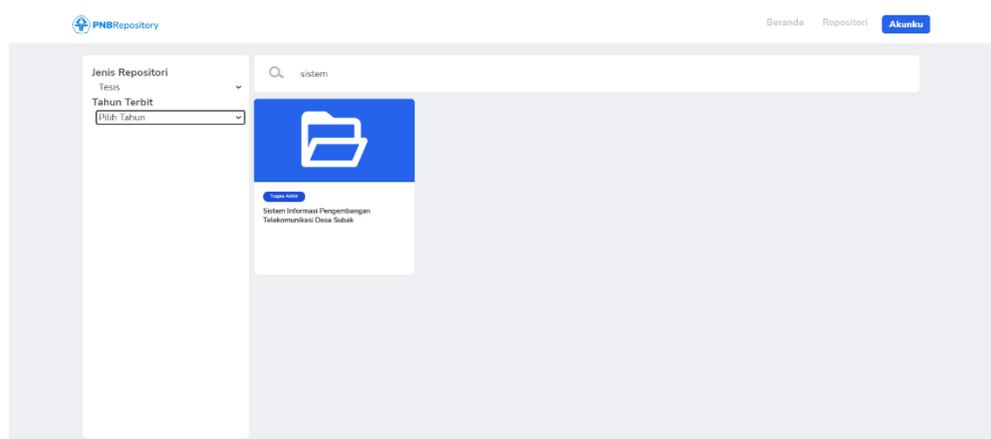


**Gambar 4. 24 Halaman Utama**

Sumber: Hasil Perancangan

#### e. Halaman Pencarian Tugas Akhir

Pengujian dilakukan dengan mengakses halaman pencarian tugas akhir melalui *textbox search*, dengan memilih jenis repositori dan tekan tombol *enter* pada *keyboard*. Setelah itu, ketikkan repositori yang ingin dicari. Tampilan pencarian tugas akhir ditunjukkan pada gambar 4.25.

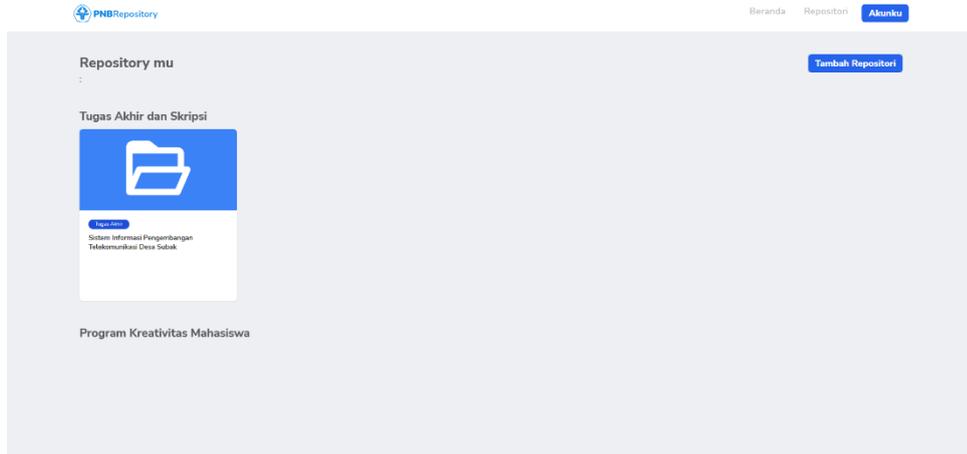


**Gambar 4. 25 Halaman Pencarian Tugas Akhir**

Sumber: Hasil Perancangan

#### f. Halaman Repositori

Pengujian dilakukan dengan mengakses halaman repositori setelah memilih menu repositori, maka sistem akan menampilkan tampilan repositori sesuai dengan akun yang dipilih saat melakukan registrasi. Halaman repositori pada sistem repositori tugas akhir ditunjukkan pada gambar 4.26.



**Gambar 4. 26 Halaman Repositori**

Sumber: Hasil Perancangan

#### g. Halaman Pilih Tipe Repositori

Pengujian dilakukan dengan mengakses halaman pilih tipe repositori setelah menekan tombol tambah repositori, kemudian akan muncul halaman pemilihan jenis repositori. Pengguna diminta untuk memilih tipe repositori yang akan dibuat. Tampilan pilih tipe repositori ditunjukkan pada gambar 4.27.

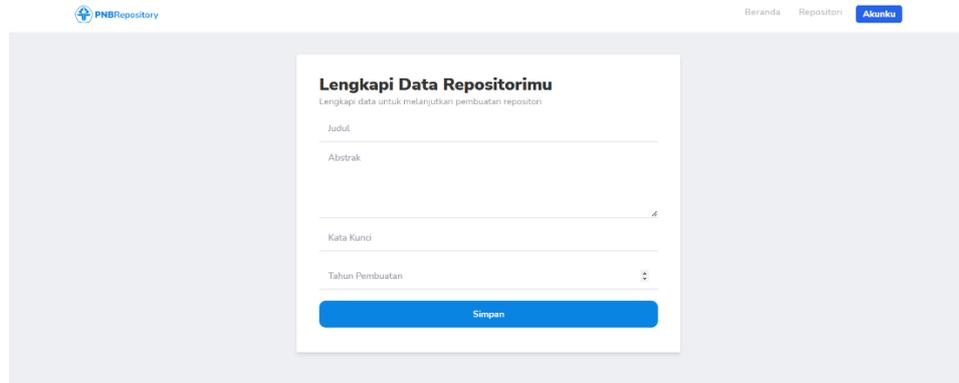


**Gambar 4. 27 Halaman Pilih Tipe Repositori**

Sumber: Hasil Perancangan

#### **h. Halaman *Form* Pembuatan Repositori**

Pengujian dilakukan dengan mengakses halaman *form* pembuatan repositori tugas akhir yang akan ditampilkan apabila user telah memilih tipe tugas akhir. Halaman *form* pembuatan repositori sistem repositori tugas akhir ditunjukkan pada gambar 4.28.



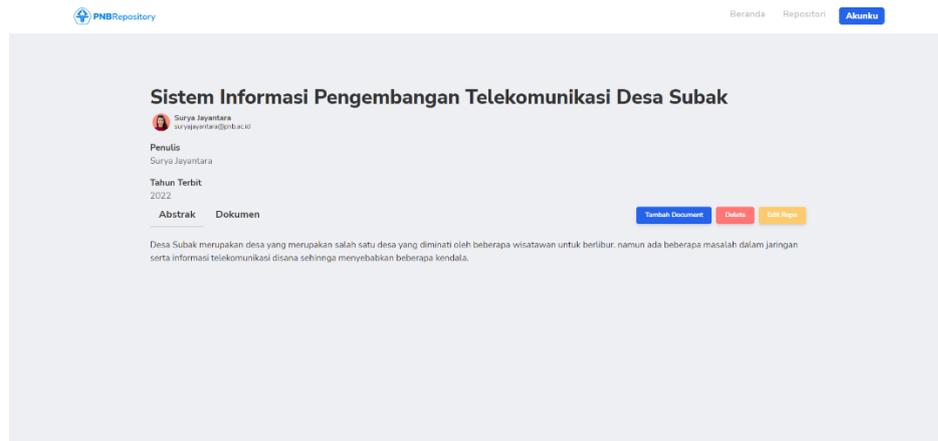
The screenshot shows a web form titled "Lengkapi Data Repositorimu" (Complete Your Repository Data). The form is part of the PNBRepository system, as indicated by the logo in the top left and navigation links (Beranda, Repositori, Akunku) in the top right. The form contains the following fields: "Judul" (Title), "Abstrak" (Abstract), "Kata Kunci" (Keywords), and "Tahun Pembuatan" (Year of Publication). A blue "Simpan" (Save) button is located at the bottom of the form.

**Gambar 4. 28 Halaman *Form* Pembuatan Repositori**

Sumber: Hasil Perancangan

#### **i. Halaman Repositori Data Detail Tugas Akhir**

Pengujian dilakukan dengan mengakses halaman repositori data detail tugas akhir yang akan ditampilkan apabila memilih salah satu data pada repositorinya. Halaman repositori data detail tugas akhir sistem repositori tugas akhir ditunjukkan pada gambar 4.29.



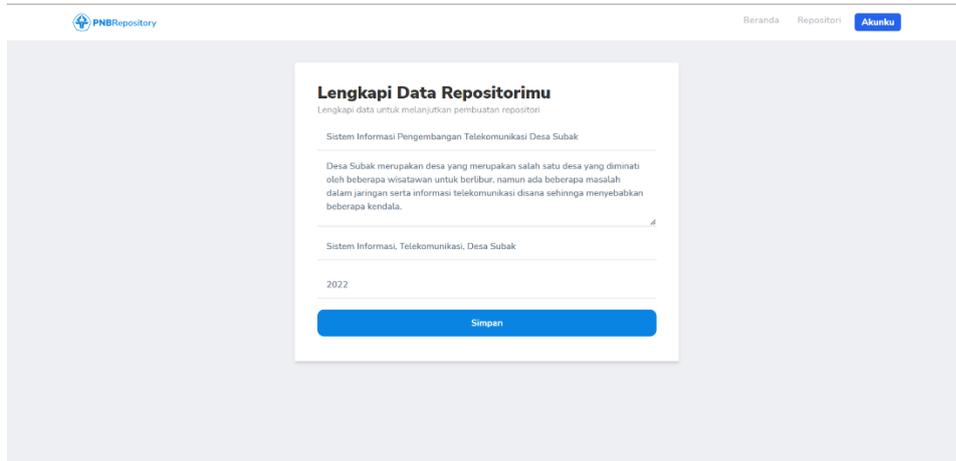
The screenshot shows a repository entry titled "Sistem Informasi Pengembangan Telekomunikasi Desa Subak". The entry is by the author "Surya Jayantara" (suryajayantara@pnb.ac.id) and was published in 2022. It includes an "Abstrak" (Abstract) and a "Dokumen" (Document) section. The abstract text reads: "Desa Subak merupakan desa yang merupakan salah satu desa yang diminati oleh beberapa wisatawan untuk berlibur, namun ada beberapa masalah dalam jaringan serta informasi telekomunikasi disana sehingga menyebabkan beberapa kendala." The page also features navigation buttons: "Tambah Dokumen" (Add Document), "Delete", and "Edit Page".

**Gambar 4. 29 Halaman Repositori Data Detail Tugas Akhir**

Sumber: Hasil Perancangan

**j. Halaman Repositori Edit Data Detail Tugas Akhir**

Pengujian dilakukan dengan mengakses halaman repositori edit data detail tugas akhir yang akan ditampilkan apabila memilih tombol edit repo. Halaman repositori edit data detail tugas akhir sistem repositori tugas akhir ditunjukkan pada gambar 4.30.



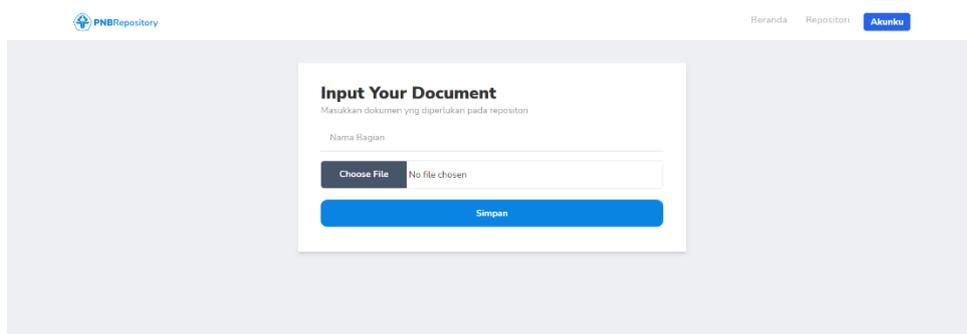
The screenshot shows a web interface for editing repository data. At the top left is the 'PNBRepository' logo, and at the top right are navigation links for 'Beranda', 'Repositori', and 'Akunku'. The main content area features a white form titled 'Lengkapi Data Repositorimu' with the subtitle 'Lengkapi data untuk melanjutkan pembuatan repositori'. The form contains the following fields: 'Sistem Informasi Pengembangan Telekomunikasi Desa Subak', a text area with a placeholder paragraph about Desa Subak, 'Sistem Informasi, Telekomunikasi, Desa Subak', and the year '2022'. A blue 'Simpan' button is located at the bottom of the form.

**Gambar 4. 30 Halaman Repositori Edit Data Detail Tugas Akhir**

Sumber: Hasil Perancangan

**k. Halaman Repositori Tambah Dokumen Tugas Akhir**

Pengujian dilakukan dengan mengakses halaman repositori tambah dokumen tugas akhir yang akan ditampilkan apabila memilih tombol tambah dokumen. Halaman repositori edit data detail tugas akhir sistem repositori tugas akhir ditunjukkan pada gambar 4.31.



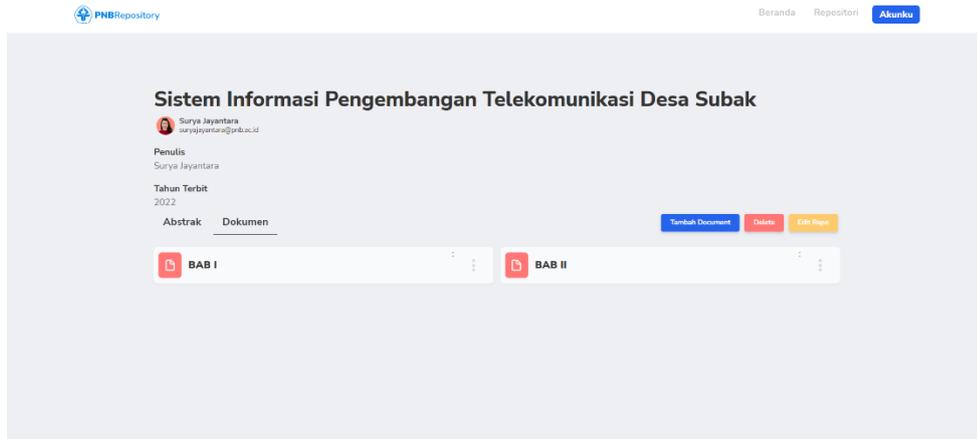
The screenshot shows a web interface for adding a document to the repository. At the top left is the 'PNBRepository' logo, and at the top right are navigation links for 'Beranda', 'Repositori', and 'Akunku'. The main content area features a white form titled 'Input Your Document' with the subtitle 'Masukkan dokumen yang diperlukan pada repositori'. The form contains the following fields: 'Nama Bagian', a file selection button labeled 'Choose File' with the text 'No file chosen' next to it, and a blue 'Simpan' button at the bottom.

**Gambar 4. 31 Halaman Repositori Tambah Dokumen Tugas Akhir**

Sumber: Hasil Perancangan

### l. Halaman Repositori Data Dokumen Tugas Akhir

Pengujian dilakukan dengan mengakses halaman repositori data dokumen tugas akhir yang akan ditampilkan apabila menekan pilihan dokumen. Halaman repositori edit data detail tugas akhir sistem repositori tugas akhir ditunjukkan pada gambar 4.32.

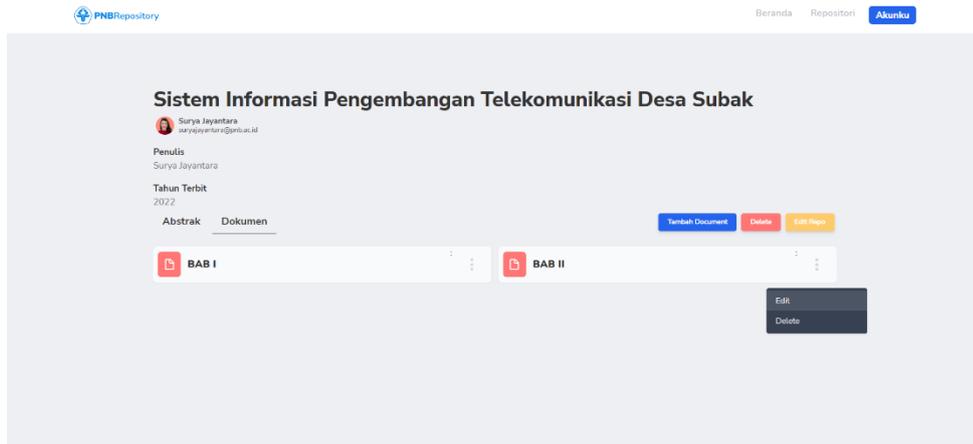


**Gambar 4. 32 Halaman Repositori Data Dokumen Tugas Akhir**

Sumber: Hasil Perancangan

### m. Halaman Repositori Edit Dan Hapus Data Dokumen Tugas Akhir

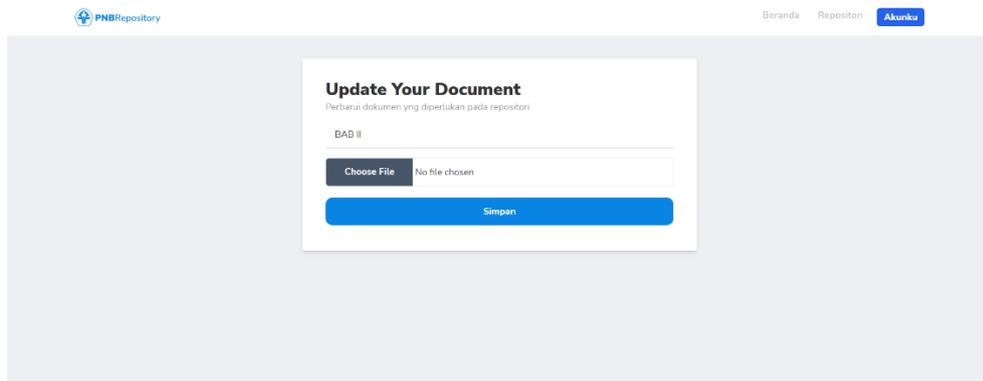
Pengujian dilakukan dengan mengakses halaman repositori edit dan hapus dokumen tugas akhir yang akan ditampilkan apabila memilih tombol edit atau *delete*. Halaman repositori tombol edit dan hapus data dokumen tugas akhir sistem repositori tugas akhir ditunjukkan pada Gambar 4.33.



**Gambar 4. 33 Halaman Repositori Edit Dan Hapus Data Dokumen Tugas Akhir**

Sumber: Hasil Perancangan

Setelah itu pilih tombol hapus untuk menghapus data dan jika ingin mengedit data, maka akan muncul form edit data dokumen seperti pada Gambar 4.34.

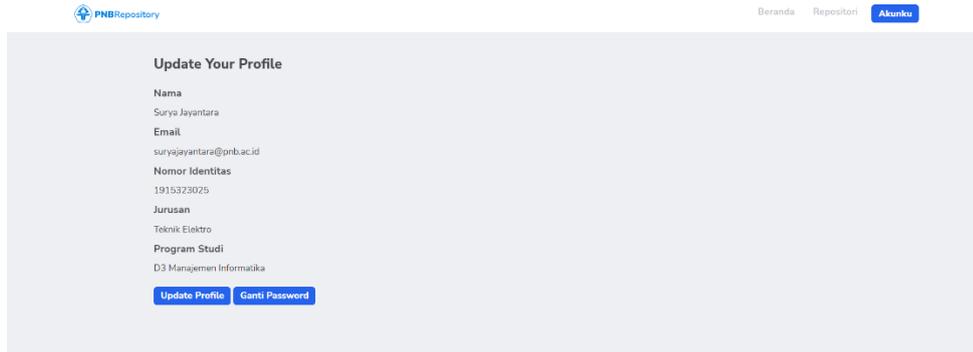


**Gambar 4. 34 Form edit data dokumen**

Sumber: Hasil Perancangan

#### n. Halaman Data Diri

Pengujian dilakukan dengan mengakses halaman data diri yang akan ditampilkan apabila memilih tombol akunku pada saat berada di halaman utama. Halaman halaman data diri sistem repositori tugas akhir ditunjukkan pada gambar 4.35.

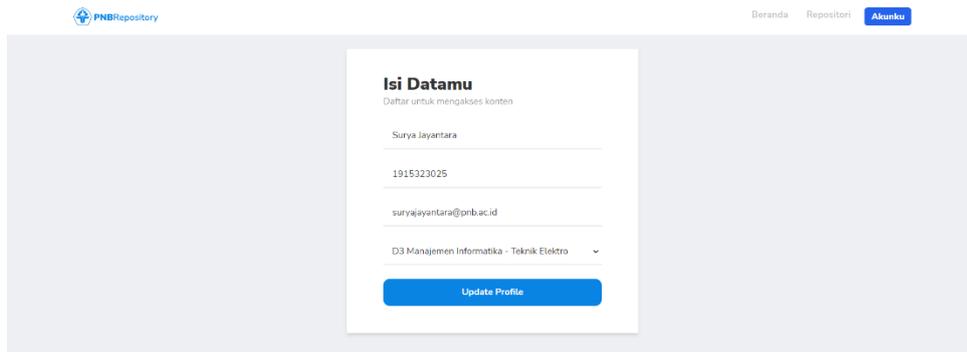


**Gambar 4. 35 Halaman Data Diri**

Sumber: Hasil Perancangan

**o. Halaman Edit Data Diri**

Pengujian dilakukan dengan mengakses halaman repositori edit data diri yang akan ditampilkan apabila memilih tombol *update profile*. Halaman repositori edit data diri sistem repositori tugas akhir ditunjukkan pada gambar 4.36.

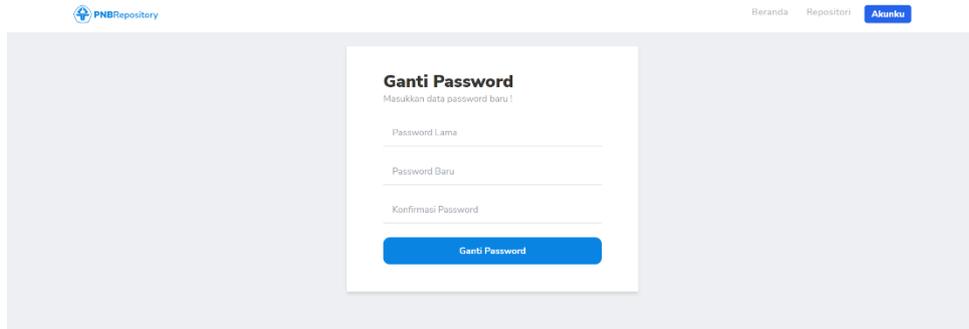


**Gambar 4. 36 Halaman Edit Data Diri**

Sumber: Hasil Perancangan

**p. Halaman Ganti Password**

Pengujian dilakukan dengan mengakses halaman repositori ganti *password* yang akan ditampilkan apabila memilih tombol ganti *password*. Halaman ganti *password* sistem repositori tugas akhir ditunjukkan pada gambar 4.37.

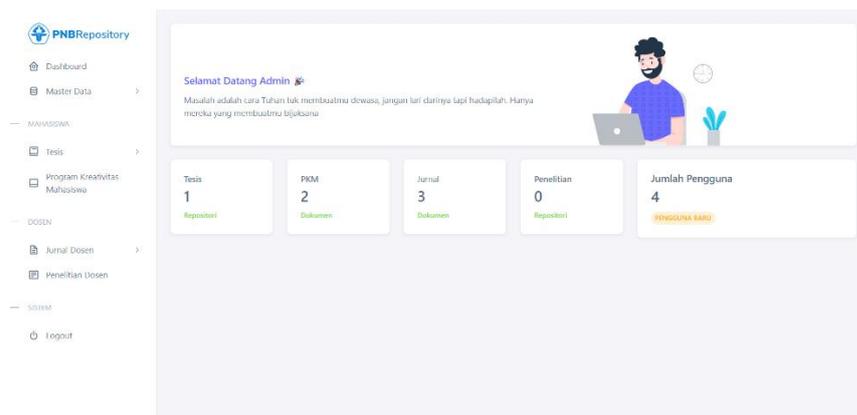


**Gambar 4. 37 Halaman Ganti Password**

Sumber: Hasil Perancangan

**q. Halaman Dashboard Admin**

Pengujian dilakukan dengan mengakses halaman *dashboard* admin yang akan ditampilkan apabila *login* menggunakan akun admin. Halaman *dashboard* admin sistem repositori tugas akhir ditunjukkan pada Gambar 4.38.

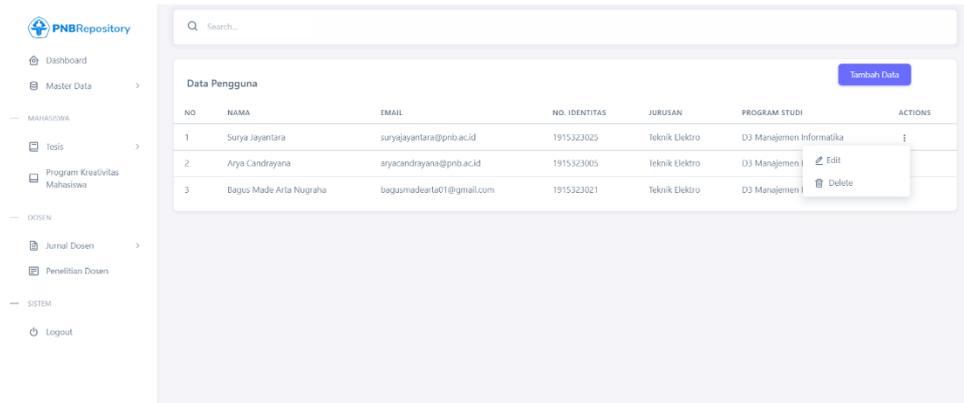


**Gambar 4. 38 Halaman Dashboard Admin**

Sumber: Hasil Perancangan

**r. Halaman Mengelola Data User**

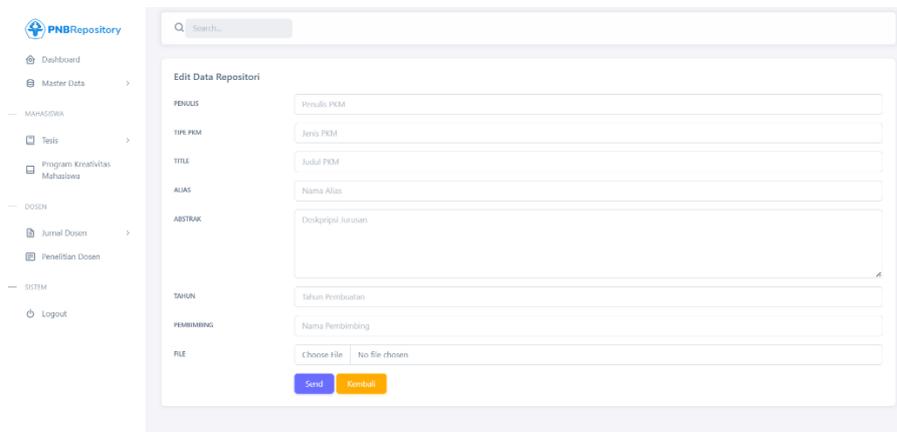
Pengujian dilakukan dengan mengakses halaman mengelola data *user* yang dapat dilakukan pada menu master data, kemudian pilih pengguna. Admin akan diarahkan ke halaman kelola data *user*, maka sistem akan menampilkan data user berupa nama, email, nomor identitas, jurusan, program studi, dan dilengkapi *action* berupa *dropdown* untuk mengedit dan menghapus data. Halaman mengelola data user ditunjukkan pada Gambar 4.39.



**Gambar 4. 39 Halaman Mengelola Data User**

Sumber: Hasil Perancangan

Untuk menambahkan data user baru dapat dilakukan dengan menekan tombol tambah data. Kemudian akan tampil *form* tambah data *user* dengan mengisi data *user* berupa jenis akun yang dibuat, nama, email, nomor identitas, program studi dan *password*. Halaman *form* tambah data *user* ditunjukkan pada Gambar 4.40.



**Gambar 4. 40 Halaman form tambah data user**

Sumber: Hasil Perancangan

Apabila data berhasil disimpan maka akan kembali ke halaman kelola data user. Untuk menghapus data bisa di pilih pada menu *action* kemudian *delete*. Kemudian untuk mengubah data *user*, dapat dilakukan dengan memilih data edit. Halaman *form* edit data *user* ditunjukkan pada Gambar 4.41.

**Gambar 4. 41 Halaman form edit data user**

Sumber: Hasil Perancangan

**s. Halaman Mengelola Data Repositori Tugas Akhir**

Pengujian dilakukan dengan mengakses halaman mengelola data Repositori tugas akhir dapat dilakukan dengan memilih menu tesis. Pada menu kelola repositori tugas akhir ditampilkan data judul, jenis tugas akhir, pembimbing, tahun, penulis, pengunggah dan *actions*. Halaman mengelola data repositori tugas akhir ditampilkan pada Gambar 4.42.

| NO | JUDUL TESIS  | JENIS TESIS | PENULIS            | PENGUNGGAH      | ACTIONS |
|----|--|-------------|--------------------|-----------------|---------|
| 1  | Pengembangan Perangkat Bergerak berbasis Flutter untuk Pembuatan API | Tugas Akhir | Surya Jayantara    | Admin           | ⋮       |
| 2  | Sistem Informasi Pengembangan Telekomunikasi Desa Subak              | Tugas Akhir | Surya Jayantara    | Surya Jayantara | ⋮       |
| 3  | Sistem Informasi Penjualan Toko Aquascapae                           | Tugas Akhir | Made Sanjaya Putra | Admin           | ⋮       |

**Gambar 4. 42 Halaman Mengelola Data Repositori Tugas Akhir**

Sumber: Hasil Perancangan

Untuk menambahkan repositori tugas akhir dapat dilakukan dengan menekan tombol tambah data kemudian admin akan diarahkan ke *form* tambah data repositori tugas akhir. Pada halaman tambah repositori tugas akhir admin diminta mengisi data penulis, jenis tesis, judul, kata kunci, abstrak, dan tahun pembuatan. Halaman tambah data repositori tugas akhir ditunjukkan pada Gambar 4.42.

**Gambar 4. 43 Tambah data repositori tugas akhir**

Sumber: Hasil Perancangan

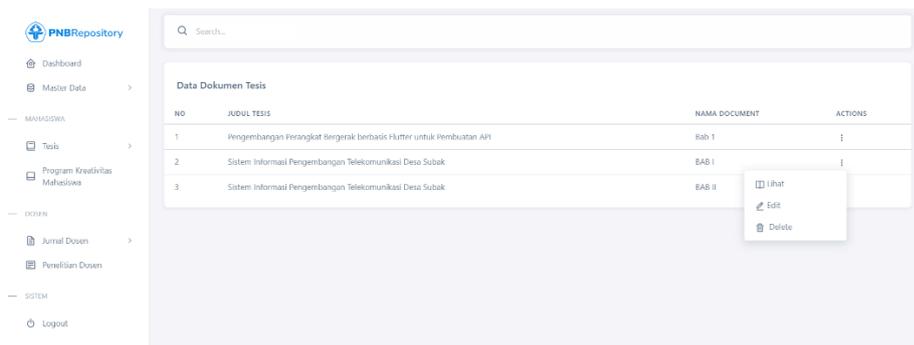
Untuk mengubah data repositori dapat dilakukan pada bagian *action* yang akan memunculkan *dropdown* lihat, edit, dan *delete*. Pilih edit untuk mengubah data repositori. Selanjutnya akan diarahkan ke *form* ubah data repositori tugas akhir. Tampilan *form* edit repositori tugas akhir ditunjukkan pada Gambar 4.44.

**Gambar 4. 44 Tampilan form edit repositori tugas akhir**

Sumber: Hasil Perancangan

**t. Halaman Mengelola Data Dokumen Tugas Akhir**

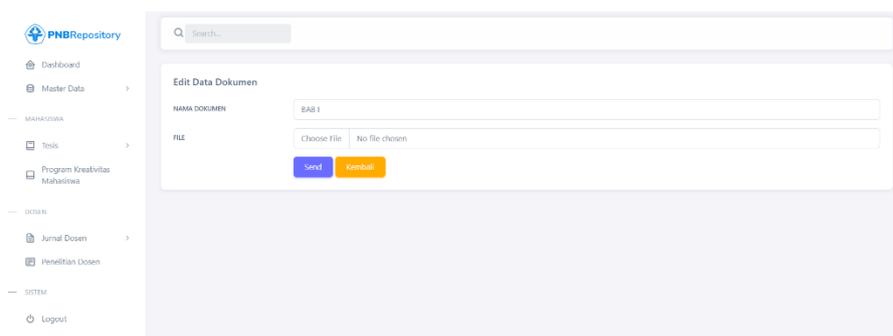
Pengujian dilakukan dengan mengakses halaman mengelola data dokumen tugas akhir dapat dilakukan pada menu tesis, kemudian pilih dokumen. Pada halaman kelola data tugas akhir ditampilkan judul tesis, nama *document*, dan actions. mengelola data dokumen tugas akhir ditunjukkan pada gambar 4.45.



**Gambar 4. 45 Halaman Mengelola Data Dokumen Tugas Akhir**

Sumber: Hasil Perancangan

Untuk mengubah data tugas akhir dapat dilakukan pada menu *action*, yang akan menampilkan *dropdown edit* dan *delete*. *Delete* untuk menghapus sedangkan *edit* untuk memunculkan halaman form edit data tugas akhir. Tampilan form edit data tugas akhir ditunjukkan pada gambar 4.46.



**Gambar 4. 46 Tampilan form edit data tugas akhir**

Sumber: Hasil Perancangan

#### 4.4 Tabel Pengujian

Sistem yang telah dibangun dilakukan beberapa pengujian untuk mengetahui bahwa fitur-fitur yang dibuat sudah berfungsi dengan baik sesuai dengan kebutuhan yang diperlukan. Pengujian yang dilakukan dapat dilihat pada tabel 4.3.

**Tabel 4. 3 Tabel Hasil Pengujian**

| <b>No.</b> | <b>Fitur</b>                   | <b>Hasil yang Diharapkan</b>  | <b>Keterangan</b>  | <b>Hasil Pengujian</b> |
|------------|--------------------------------|---|--|------------------------|
| 1          | <i>Login</i>                   | Dapat memvalidasi user yang ingin masuk ke sistem. Dan membedakan setiap Role user                                | Proses login berhasil dan dapat membedakan beberapa <i>role</i> user ke sistem | Berhasil               |
| 2          | Registrasi                     | Dapat mendaftarkan user baru yang ingin mendaftarkan diri ke sistem sesuai dengan <i>role</i> dosen dan mahasiswa | Registrasi berhasil dan user dapat terdaftar sesuai dengan role yang dipilih   | Berhasil               |
| 3          | Membuat Repositori             | Dapat menyimpan data repositori tugs akhir  | Sistem berhasil menyimpan data repositori akhir                                | Berhasil               |
| 4          | <i>Upload File</i> Tugas Akhir | Dapat menyimpan data dokumen setiap repositori tugas akhir  | Sistem dapat menyimpan data dokumen yang diupload oleh user                    | Berhasil               |

**Tabel 4.3 Tabel Hasil Pengujian (Lanjutan)**

| <b>No.</b> | <b>Fitur</b>              | <b>Hasil yang Diharapkan</b>  | <b>Keterangan</b>   | <b>Hasil Pengujian</b> |
|------------|---------------------------|---|---|------------------------|
| 5          | Mengelola Data Mahasiswa  | Dapat menampilkan data user, menambahkan data user, memperbaiki data user dan menghapus data user oleh admin  | Sistem dapat menampilkan data user, menambah data user, memperbaiki data user, dan menghapus data user  | Berhasil               |
| 6          | Mengelola Data Repositori | Dapat menampilkan data repositori tugas akhir, menambahkan data repositori, memperbaiki data repositori dan menghapus repositori tugas akhir oleh admin | Sistem dapat menampilkan data repositori tugas akhir, menambahkan data repositori, memperbaiki data repositori dan menghapus repositori tugas akhir | Berhasil               |
| 7          | Mengelola Data Diri       | Dapat menampilkan data diri user dan memperbaiki data diri dan password   | Sistem berhasil dapat menampilkan data diri user dan memperbaiki data user dan password   | Berhasil               |

**Tabel 4.3 Tabel Hasil Pengujian (Lanjutan)**

| <b>No.</b> | <b>Fitur</b>               | <b>Hasil yang Diharapkan</b>  | <b>Keterangan</b>   | <b>Hasil Pengujian</b> |
|------------|----------------------------|---|---|------------------------|
| 8          | Mengelola Data Tugas Akhir | Dapat menampilkan data dokumen tugas akhir, menambahkan data dokumen, memperbarui data dokumen dan menghapus dokumen tugas akhir oleh admin | Sistem dapat menampilkan data dokumen tugas akhir, menambahkan data dokumen, memperbarui data dokumen dan menghapus dokumen tugas akhir | Berhasil               |
| 9          | <i>Logout</i>              | Dapat mengeluarkan user dari sistem sehingga tidak dapat mengakses sistem lagi  | Sistem berhasil mengeluarkan user dari sistem.  | Berhasil               |

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil analisis dan perancangan serta implementasi dari sistem repositori tugas akhir, dapat diambil kesimpulan sebagai berikut:

- a. Analisis dan perancangan sistem repositori tugas akhir menggunakan *flowmap*, *use case*, ERD, *activity diagram*, *class diagram*, *sequence diagram* dan perancangan antarmuka.
- b. Sistem repositori tugas akhir dalam implementasinya menggunakan rancangan API dan basis data MySQL.
- c. Hasil pengujian sistem repositori tugas akhir menunjukkan seluruh fungsionalitas yang dapat berjalan dengan perancangan yang telah dibuat.

#### **5.2 Saran**

Berdasarkan kesimpulan yang telah diuraikan sebelumnya, maka ada beberapa saran agar sistem repositori tugas akhir ini, agar dapat berkembang lebih baik lagi. Saran tersebut adalah sebagai berikut:

- a. Pengembangan fitur pada bagian verifikasi email agar dapat dilakukan pada *platform website* dan *android*.
- b. Pengembangan fitur yang diperlukan untuk membuat sistem repositori lebih mudah digunakan dari pihak dosen, mahasiswa dan admin.

## DAFTAR PUSTAKA

- [1] A. Herliana and P. M. Rasyid, "SISTEM INFORMASI MONITORING PENGEMBANGAN SOFTWARE PADA TAHAP DEVELOPMENT BERBASIS WEB," *Jurnal Informatika*, vol. III, no. 1, pp. 41-50, 2016.
- [2] W. Darmalaksana and E. M. Kulsum, "Rancang Bangun Prototipe Repositori Sederhana Menggunakan Metode Design Thinking Hadis," *Gunung Djati Conference Series*, vol. 8, no. 2, pp. 258-269, 2022.
- [3] I. F. Hanif and G. M. Sinambela, "PEMBUATAN APLIKASI E-TATIB BERBASIS ANDROID MENGGUNAKAN BAHASA PEMROGRAMAN DART," *Jurnal Teknologi dan Terapan Bisnis (JTTB)*, vol. 4, no. 1, pp. 1-7, 2021.
- [4] D. Ambriani and A. I. Nurhidayat, "RANCANG BANGUN REPOSITORY PUBLIKASI ILMIAH DOSEN BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL," *Jurnal Manajemen Informatika*, vol. 10, no. 1, pp. 58-66, 2020.
- [5] Alvian, "Hubungan dan Perbedaan Javascript, HTML, CSS , JQuery dan PHP di dalam Web Development," BINUS UNIVERSITY School of information Systems, 25 Februari 2019. [Online]. Available: <https://sis.binus.ac.id/2019/02/25/hubungan-dan-perbedaan-javascript-html-css-jquery-dan-php-di-dalam-web-development/>. [Accessed 1 Mei 2022].
- [6] C. Kartiko, . A. C. Wardhana and D. P. Rakhmadani, "Pengembangan Mobile Learning Management System Dengan User Centered Design (UCD) Menggunakan Flutter Framework," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 2, pp. 960-968, 2022.
- [7] A. G. Setiawan and A. U. Zailani, "Perancangan Sistem Informasi Akademik Berbasis Web Mobile pada Pondok Pesantren Dar El Amir

dengan Metode Waterfal," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 5, no. 1, pp. 6-13, 2022.

- [8] S. Butsianto, "APLIKASI MEDIA PEMBELAJARAN UNTUK ANAK USIA DINI BERBASIS ANDROID," *Jurnal Teknologi Pelita Bangsa*, vol. 6, no. 1, pp. 112-121, 2017.
- [9] R. Juliarto, "Apa itu UML? Beserta Pengertian dan Contohnya," Dicoding Intern, 12 Mei 2021. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-uml/>. [Accessed 1 Mei 2022].
- [10] M. S. Novendri, A. Saputra and C. E. Firman, "APLIKASI INVENTARIS BARANG PADA MTS NURUL ISLAM DUMAI MENGGUNAKAN PHP DAN MYSQL," *JURNAL MANAJENEN DAN TEKNOLOGI INFORMASI*, vol. 10, no. 2, pp. 46-57, 2019.
- [11] W. Darmalaksana and E. M. Kulsum, "Rancang Bangun Prototipe Repositori Sederhana Menggunakan Metode Design Thinking Hadis," *Gunung Djati Conference Series*, vol. 8, no. 2, pp. 258-269, 2022.
- [12] I. F. Hanif and G. M. Sinambela, "PEMBUATAN APLIKASI E-TATIB BERBASIS ANDROID MENGGUNAKAN," *Jurnal Teknologi dan Terapan Bisnis (JTTB)*, vol. 4, no. 1, pp. 1-7, 2021.
- [13] D. Ambriani and A. I. Nurhidayat, "RANCANG BANGUN REPOSITORY PUBLIKASI ILMIAH DOSEN BERBASIS WEB MENGGUNAKAN," *Jurnal Manajemen Informatika*, vol. 10, no. 1, pp. 58-66, 2020.

## **LAMPIRAN**



**LEMBAR PERBAIKAN**  
**UJIAN KOMPREHENSIF**  
**TUGAS AKHIR TH. AKADEMIK 2021/2022**

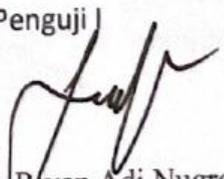
Nama Mahasiswa : I Putu Surya Jayantara  
NIM : 1915323025  
Program Studi : DIII Manajemen Informatika  
Judul Tugas Akhir : REPOSITORY TUGAS AKHIR PADA APLIKASI DIGITAL REPOSITORY  
(PNB REPOSITORY)

| NO | CATATAN PERBAIKAN           | BAB | HALAMAN |
|----|-----------------------------|-----|---------|
|    | <p>30/09<br/>AIC / 2022</p> |     |         |

Catatan :

- Lembaran ini ditanda tangani  
Setelah selesai melakukan perbaikan

Jimbaran, 8 September 2022 .  
Panita Ujian Komprehensif  
Program Studi DIII Manajemen Informatika  
Jurusan Teknik Elektro  
Dosen Penguji

  
(I Made Ryan Adi Nugroho, S.SI, M.T.)  
NIP. 198102202015042001



**LEMBAR PERBAIKAN**  
**UJIAN KOMPREHENSIF**  
TUGAS AKHIR TH. AKADEMIK 2021/2022

Nama Mahasiswa : I Putu Surya Jayantara  
NIM : 1915323025  
Program Studi : DIII Manajemen Informatika  
Judul Tugas Akhir : REPOSITORY TUGAS AKHIR PADA APLIKASI DIGITAL REPOSITORY  
(PNB REPOSITORY)

| NO | CATATAN PERBAIKAN   | BAB | HALAMAN |
|----|---|-----|---------|
| 1  | Daftar Gambar dan Daftar Tabel, diperbaiki disesuaikan dengan format jangan di bold |     |         |
| 2  | Simbul yang digunakan dimasukkan ke BAB 2   |     |         |
| 3  | Cara mengutip (Bab 2)   |     |         |
| 4  | Gambar ERD diperbesar ,tingkat kardinalitas diperjelas                              |     |         |

*Ace 26/9/2022*

*Catatan :*

- Lembaran ini ditanda tangani  
Setelah selesai melakukan perbaikan

Jimbaran, <sup>26</sup>September 2022  
Panita Ujian Komprehensif  
Program Studi DIII Manajemen Informatika  
Jurusan Teknik Elektro  
Dosen Pengujian

(I Nyoman Eddy Indrayana, S.Kom.,M.T.)  
NIP. 197602202006041001



**LEMBAR PERBAIKAN**  
**UJIAN KOMPREHENSIF**  
**TUGAS AKHIR TH. AKADEMIK 2021/2022**

Nama Mahasiswa : I Putu Surya Jayantara  
NIM : 1915323025  
Program Studi : DIII Manajemen Informatika  
Judul Tugas Akhir : REPOSITORY TUGAS AKHIR PADA APLIKASI DIGITAL REPOSITORY  
(PNB REPOSITORY)

| NO | CATATAN PERBAIKAN  | BAB    | HALAMAN |
|----|--|--------|---------|
| 1  | Rumusan masalah diperbaiki gunakan yang di ppt, dan isi tanda tanya                            | 1      |         |
| 2  | Batasan masalah ditambah untuk pencarian tugas akhir hanya dengan judul                        | 1<br>3 |         |
| 3  | Tulisan system pada flowmap sistem baru diganti jadi sistem                                    |        |         |
| 4  | Untuk ERD diisi yang mana saja primary key untuk setiap entitas diberi garis bawah             | 3      |         |
| 5  | Usecase diagram register extends ke login, logout mahasiswa dan admin jadi satu menjadi logout | 3      |         |
| 6  | Usecase glossary disesuaikan jumlahnya dengan usecase diagram                                  | 3      |         |

ACC  
28-09-2022

**Catatan :**

- Lembaran ini ditanda tangani  
Setelah selesai melakukan perbaikan

Jimbaran, 8 September 2022  
Panita Ujian Komprehensif  
Program Studi DIII Manajemen Informatika  
Jurusan Teknik Elektro  
Dosen Penguji III

(I Putu Bagus Arya Pradnyana, S.Kom, M.Kom.)  
NIP. 202111015



## **SURAT KETERANGAN PERBAIKAN/REVISI**

**TUGAS AKHIR TH. AKADEMIK 2021/2022**

Yang bertanda-tangan dibawah ini, Dosen Pembimbing Tugas Akhir Program Studi DIII Manajemen Informatika Jurusan Teknik Elektro Politeknik Negeri Bali, menerangkan bahwa :

**Nama Mahasiswa** : I Putu Surya Jayantara  
**NIM** : 1915323025  
**Program Studi** : DIII Manajemen Informatika  
**Judul Tugas Akhir** : REPOSITORY TUGAS AKHIR PADA APLIKASI DIGITAL REPOSITORY (PNB REPOSITORY)

Telah diadakan perbaikan/revisi oleh mahasiswa yang bersangkutan dan dinyatakan dapat diterima untuk melengkapi laporan Tugas Akhir.

Jimbaran, September 2022  
Panita Ujian Komprehensif  
Program Studi DIII Manajemen Informatika  
Jurusan Teknik Elektro  
Pembimbing Utama,

Pembimbing Pendamping,

(Ir. I Wayan Raka Ardana, M.T.)  
NIP. 196705021993031005

(I Made Riyan Ad Nugroho, S.SI, M.T.)  
NIP. 198102202015042001

Disetujui  
Ketua Program Studi

(I Wayan Suasnawa, S.T., M.T.)  
NIP. 197511102001121002



**FORM BIMBINGAN**  
**TUGAS AKHIR TH. AKADEMIK 2021/2022**

Nama Mahasiswa : I Putu Surya Jayantara  
NIM : 1915323025  
Program Studi : D3 Manajemen Informatika  
Judul Tugas Akhir : REPOSITORY TUGAS AKHIR PADA APLIKASI DIGITAL REPOSITORY  
(PNB REPOSITORY)  
Pembimbing (Utama/Pendamping) : I Made Riyan Adi Nugroho, S.Si, M.T

| MG KE | TANGGAL | URAIAN PERKEMBANGAN DAN PERMASALAHAN | PARAF PEMBIMBING |
|-------|---------|--------------------------------------|------------------|
| I     | 23/6/21 | Revisi Bab I                         |                  |
| II    | 7/7/22  | Revisi Bab II                        |                  |
| III   | 14/7/22 | Revisi Bab III (Lampiran)            |                  |
| IV    | 8/8/22  | Diskusi Program<br>Revisi Diagram    |                  |
| V     | 9/8/22  | Revisi Bab 4                         |                  |

|      |                    |                |      |
|------|--------------------|----------------|------|
| VI   | 10/8 <sup>22</sup> | Revisi Program | Juli |
|      |                    |                |      |
| VII  | 11/8 <sup>22</sup> | Revisi Program | Juli |
|      |                    |                |      |
| VIII |                    |                | Juli |
|      |                    |                |      |
| IX   | 20/8               | Revisi Program | Juli |
|      |                    |                |      |
| X    | 7/9                | Revisi Program | Juli |
|      |                    |                |      |
| XI   |                    |                |      |
|      |                    |                |      |
| XII  |                    |                |      |
|      |                    |                |      |
| XIII |                    |                |      |
|      |                    |                |      |



**FORM BIMBINGAN**  
**TUGAS AKHIR TH. AKADEMIK 2021/2022**

**Nama Mahasiswa** : I Putu Surya Jayantara  
**NIM** : 1915323025  
**Program Studi** : Manajemen Informatika  
**Judul Tugas Akhir** : *Repository* Tugas Akhir Pada Aplikasi Digital *Repository* (PNB REPOSITORY)

**Pembimbing** : Ir. I Wayan Raka Ardana, M.T.  
(Utama/Pendamping)

| MG KE | TANGGAL | URAIAN PERKEMBANGAN DAN PERMASALAHAN | PARAF PEMBIMBING |
|-------|---------|--------------------------------------|------------------|
| I     | 24/6/22 | Revisi Bab I                         |                  |
| II    | 7/7/22  | Revisi Bab II                        |                  |
| III   | 9/7/22  | Revisi Bab II Lanjutan               |                  |
| IV    | 8/8/22  | Revisi Bab III                       |                  |
| V     | 9/8/22  | Revisi Bab III                       |                  |

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|

FORM : TA-14 (lanjutan)

|      |         |                            |     |
|------|---------|----------------------------|-----|
| VI   | 11/8/22 | Revisi Flowmap (Bab III)   | Adm |
|      |         |                            |     |
| VII  | 12/8/22 | Revisi Bab III (Rancangan) | Adm |
|      |         |                            |     |
| VIII | 20/8/22 | Revisi Bab IV (Design UI)  | Adm |
|      |         |                            |     |
| IX   | 21/8/22 | Revisi Bab IV (Design UI)  | Adm |
|      |         |                            |     |
| X    | 22/8/22 | Revisi Bab IV (Program)    | Adm |
|      |         |                            |     |
| XI   | 7/9/22  | Revisi Bab V               | Adm |
|      |         |                            |     |
| XII  |         |                            |     |
|      |         |                            |     |
| XIII |         |                            |     |
|      |         |                            |     |