

**LAPORAN PERKEMBANGAN PROYEK  
DESAIN BASIS DATA  
SISTEM INFORMASI BEBAS TANGGUNGAN  
TUGAS AKHIR  
“FINALIS JTI”**



**Disusun oleh:**

**Afifah Khoirunnisa**

***Database Designer***

**Dewita Anggraini**

***UI/UX Designer***

**Fajar Aditya Nugraha**

***UI/UX Designer***

**Farrel Augusta Dinata**

***Back-End Developer***

**Stevan Zaky Setyanto**

***Front-End Developer***

**KELOMPOK 3 ANDROMEDA - KELAS 2F  
PROGRAM STUDI D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2024**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>1</b>
<b>BAB I PENDAHULUAN.....</b>	<b>2</b>
1.1 Latar Belakang.....	2
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.3 Batasan Masalah.....	2
<b>BAB II ANALISIS SISTEM DAN FLOWCHART.....</b>	<b>3</b>
2.1 Use Case Diagram.....	3
2.2 Flowchart.....	4
a. Flowchart General Sistem Bebas Tanggungan Tugas Akhir “Finalis JTI”.....	4
b. Flowchart Mahasiswa.....	5
c. Flowchart Admin Prodi.....	6
d. Flowchart Admin TA.....	7
e. Flowchart Admin Jurusan (Super Admin).....	8
<b>BAB III ANALISIS KEBUTUHAN PENGGUNA.....</b>	<b>9</b>
3.1 Kebutuhan Fungsional.....	9
3.2 Kebutuhan Non-Fungsional.....	9
<b>BAB IV DESAIN BASIS DATA.....</b>	<b>10</b>
4.1 Desain Konseptual.....	10
4.2 Desain Logis.....	11
a. Mapping ke Model Relasional.....	11
b. Normalisasi.....	12
c. Tipe Data yang Digunakan dalam Tabel.....	12
<b>BAB V IMPLEMENTASI BASIS DATA.....</b>	<b>16</b>
5.1 Pemilihan Database Management System (DBMS).....	16
5.2 Skema Basis Data.....	16
5.3 Implementasi Basis Data Dalam Fitur Sistem.....	17
5.4 Peran Pengguna dan Perizinan.....	24
a. Peran Pengguna Sistem.....	24
b. Perizinan Akses.....	25
<b>BAB VI PENUTUP.....</b>	<b>26</b>
<b>DAFTAR PUSTAKA.....</b>	<b>27</b>

## **BAB I PENDAHULUAN**

### **1.1 Latar Belakang**

Penyelesaian tugas akhir merupakan salah satu tahapan penting yang harus dilalui oleh mahasiswa tingkat akhir sebelum menyelesaikan studinya. Selain menyelesaikan tugas akhir itu sendiri, mahasiswa juga diwajibkan untuk memenuhi serangkaian persyaratan administrasi, termasuk pengurusan surat bebas tanggungan tugas akhir. Surat ini diperlukan untuk memastikan bahwa mahasiswa tidak memiliki kewajiban yang belum diselesaikan terhadap institusi kampus, seperti peminjaman buku di perpustakaan, tanggungan laboratorium, atau administrasi lainnya.

Namun, proses pengurusan surat bebas tanggungan saat ini sering kali dilakukan secara manual dan memakan waktu. Mahasiswa harus mendatangi beberapa unit layanan kampus, menyerahkan dokumen secara fisik, dan menunggu proses verifikasi oleh petugas. Proses ini tidak hanya memakan waktu bagi mahasiswa, tetapi juga membebani staf administrasi dengan pekerjaan manual yang berulang.

Oleh karena itu, diperlukan sebuah sistem informasi berbasis digital yang dapat mempermudah proses pengajuan dan verifikasi bebas tanggungan tugas akhir. Kami beri nama sistem ini “FINALIS JTI”. Sistem ini diharapkan dapat meningkatkan efisiensi dan akurasi dalam pengelolaan dokumen, sekaligus memberikan pengalaman yang lebih baik bagi mahasiswa dan staf administrasi.

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, masalah yang dihadapi dapat dirumuskan sebagai berikut:

1. Bagaimana cara mengoptimalkan proses pengajuan dan verifikasi surat bebas tanggungan tugas akhir agar lebih efisien, mengingat saat ini proses tersebut masih dilakukan secara manual dan memakan waktu serta tenaga?
2. Apa solusi yang dapat diberikan agar mahasiswa dapat dengan mudah memantau status pengajuan surat bebas tanggungan tugas akhir mereka tanpa harus melalui proses manual yang memakan waktu?
3. Bagaimana cara mengurangi beban kerja petugas administrasi yang harus memverifikasi banyak dokumen secara manual dalam proses pengajuan surat bebas tanggungan tugas akhir?

### **1.3 Tujuan**

Adapun tujuan dari laporan ini yaitu sebagai berikut:

1. Untuk mendokumentasikan pengembangan sistem informasi yang mampu menangani pengajuan dan verifikasi bebas tanggungan tugas akhir secara digital.
2. Untuk menyediakan panduan bagi tim pengembang dalam merancang desain basis data pengguna yang efisien dan mudah digunakan.
3. Untuk memastikan bahwa sistem yang dibangun mampu memenuhi kebutuhan mahasiswa dan petugas administrasi secara efektif.

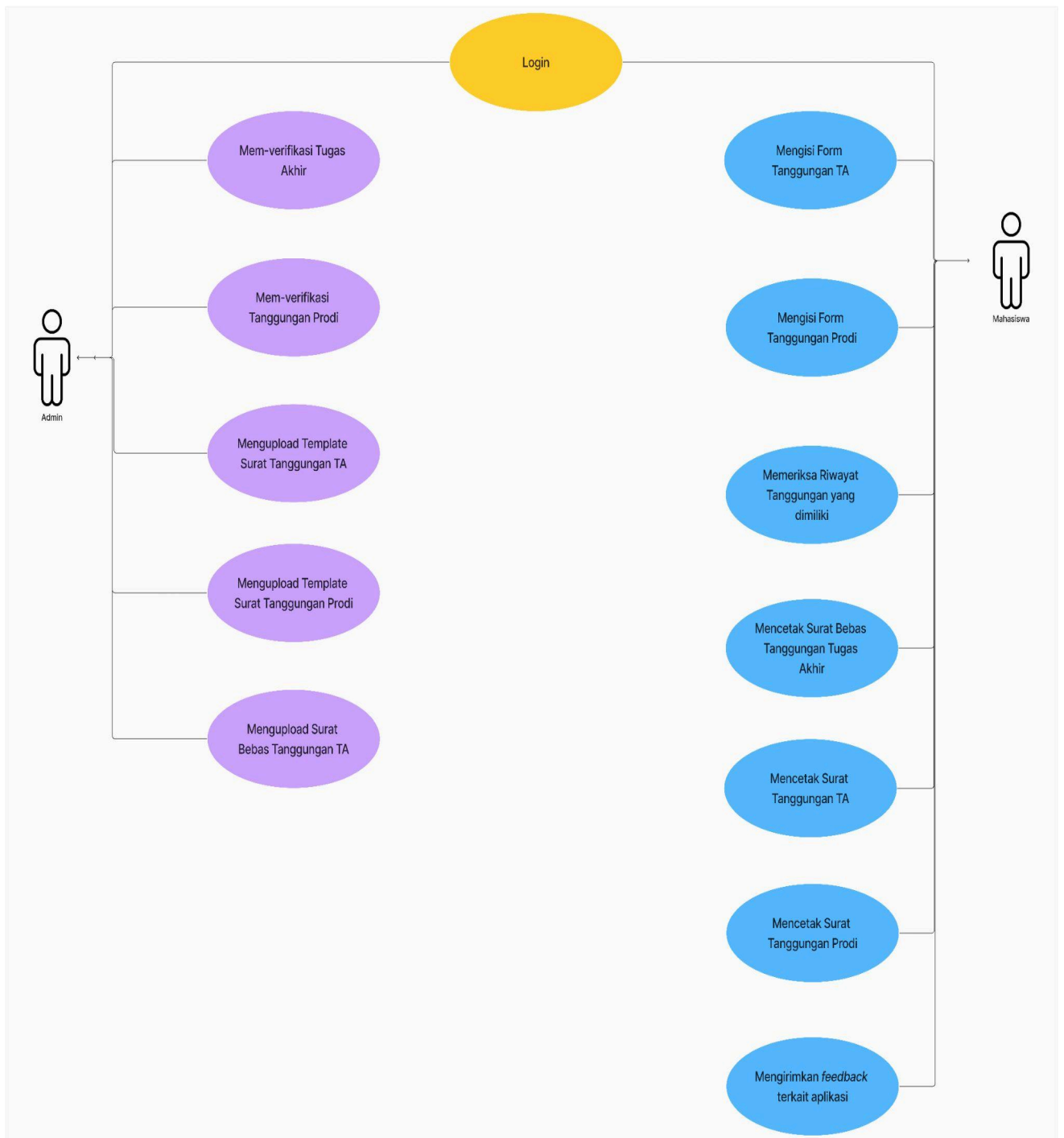
### **1.3 Batasan Masalah**

Dalam laporan ini, pembahasan difokuskan pada aspek desain antarmuka pengguna sistem informasi bebas tanggungan tugas akhir, dengan batasan sebagai berikut:

1. Sistem hanya mencakup pengelolaan dokumen bebas tanggungan tugas akhir tanpa mencakup urusan administrasi kampus lainnya.
2. Proses verifikasi dilakukan oleh admin kampus melalui sistem, sementara pengesahan akhir surat dilakukan secara otomatis setelah semua dokumen diverifikasi.

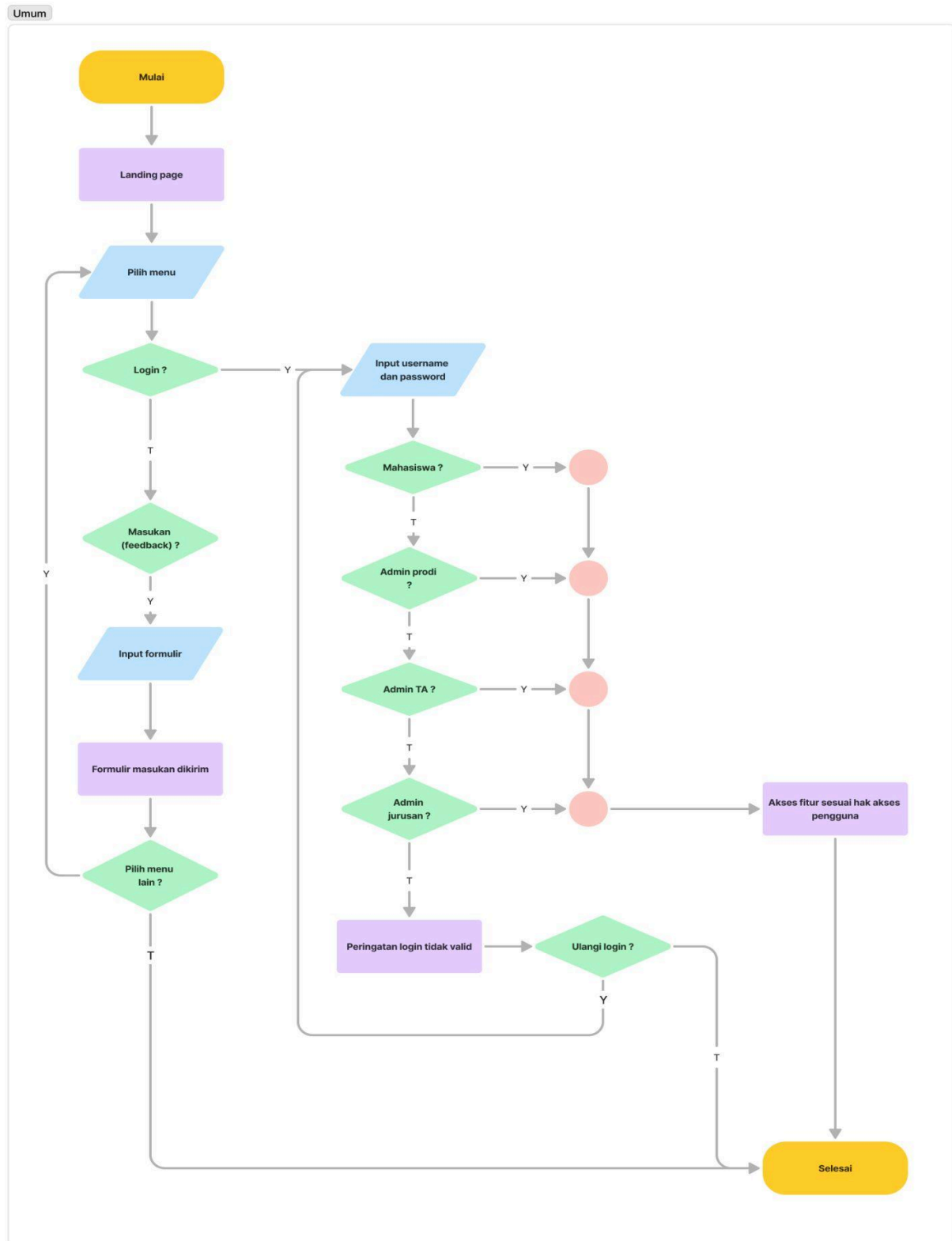
## BAB II ANALISIS SISTEM DAN FLOWCHART

### 2.1 Use Case Diagram

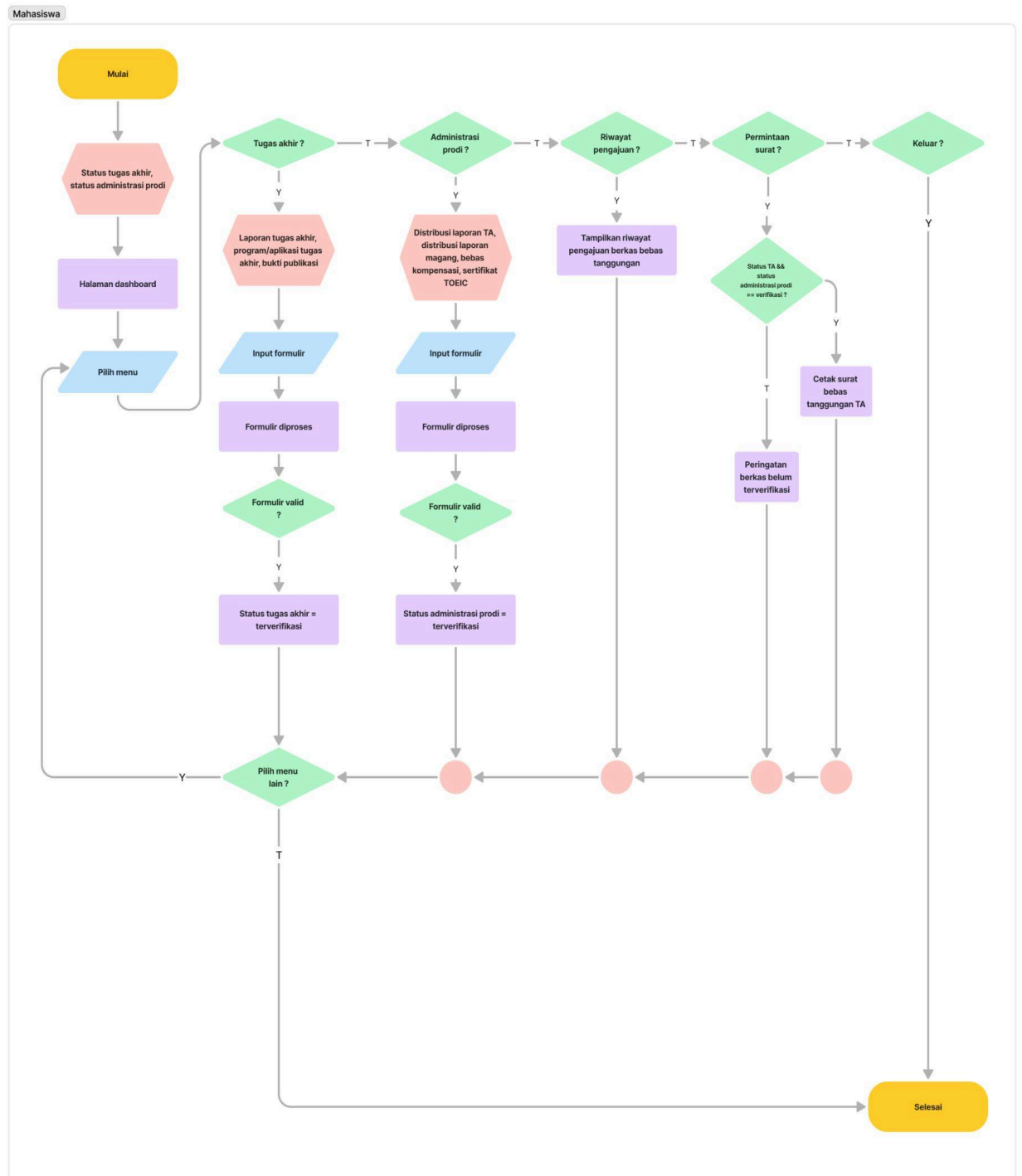


## 2.2 Flowchart

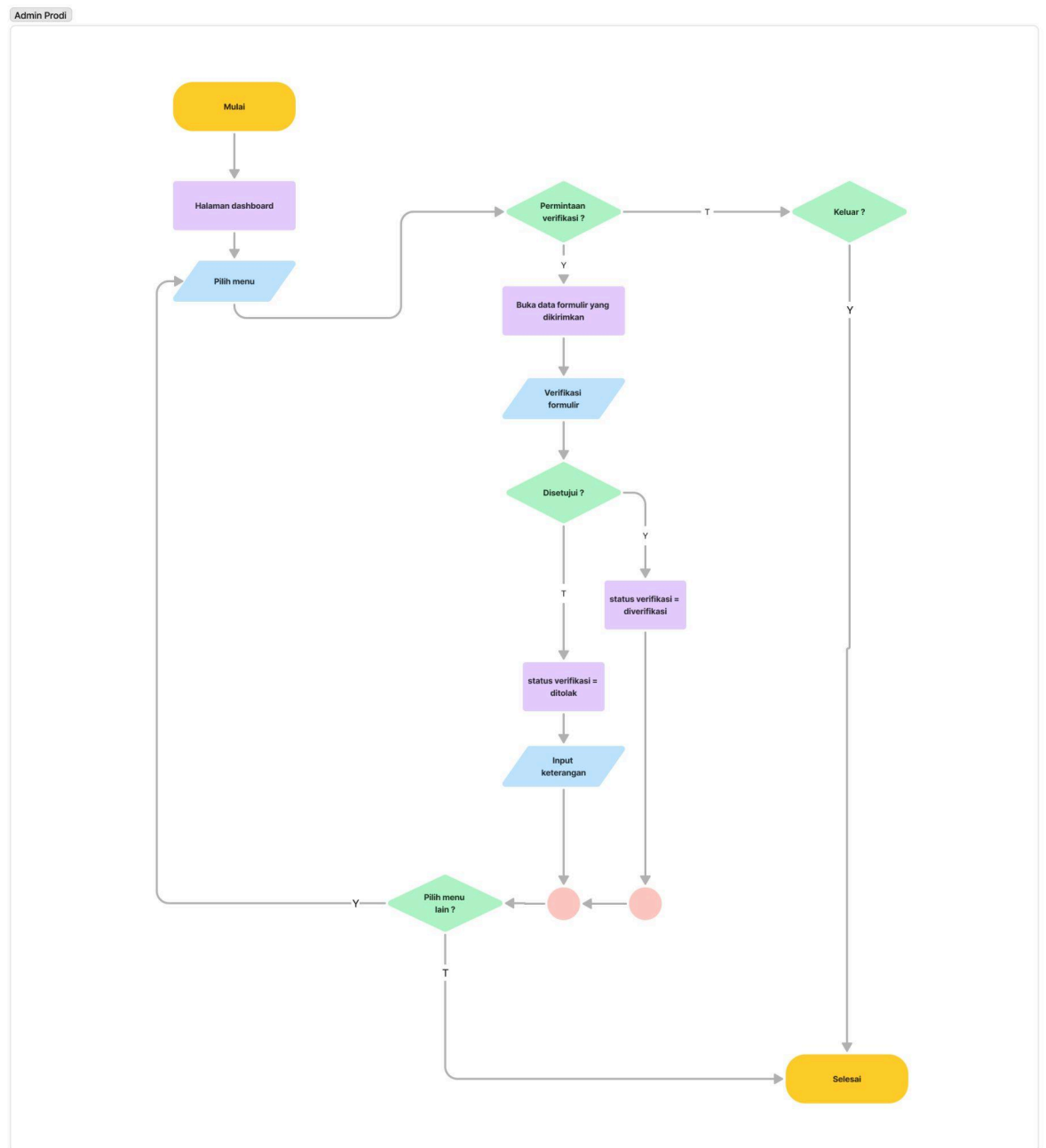
### a. Flowchart General Sistem Bebas Tanggungan Tugas Akhir “Finalis JTI”



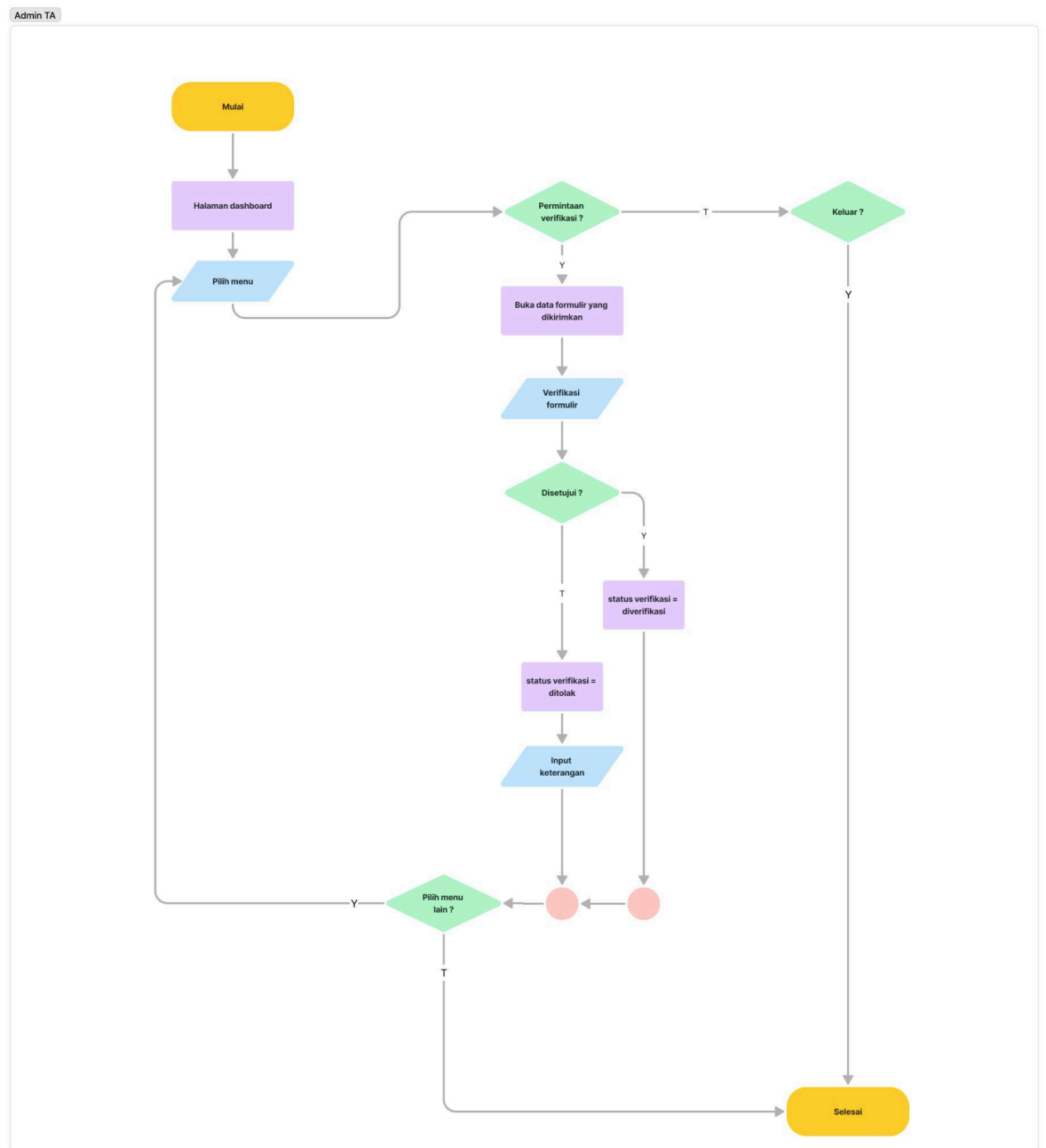
## b. Flowchart Mahasiswa



### c. Flowchart Admin Prodi

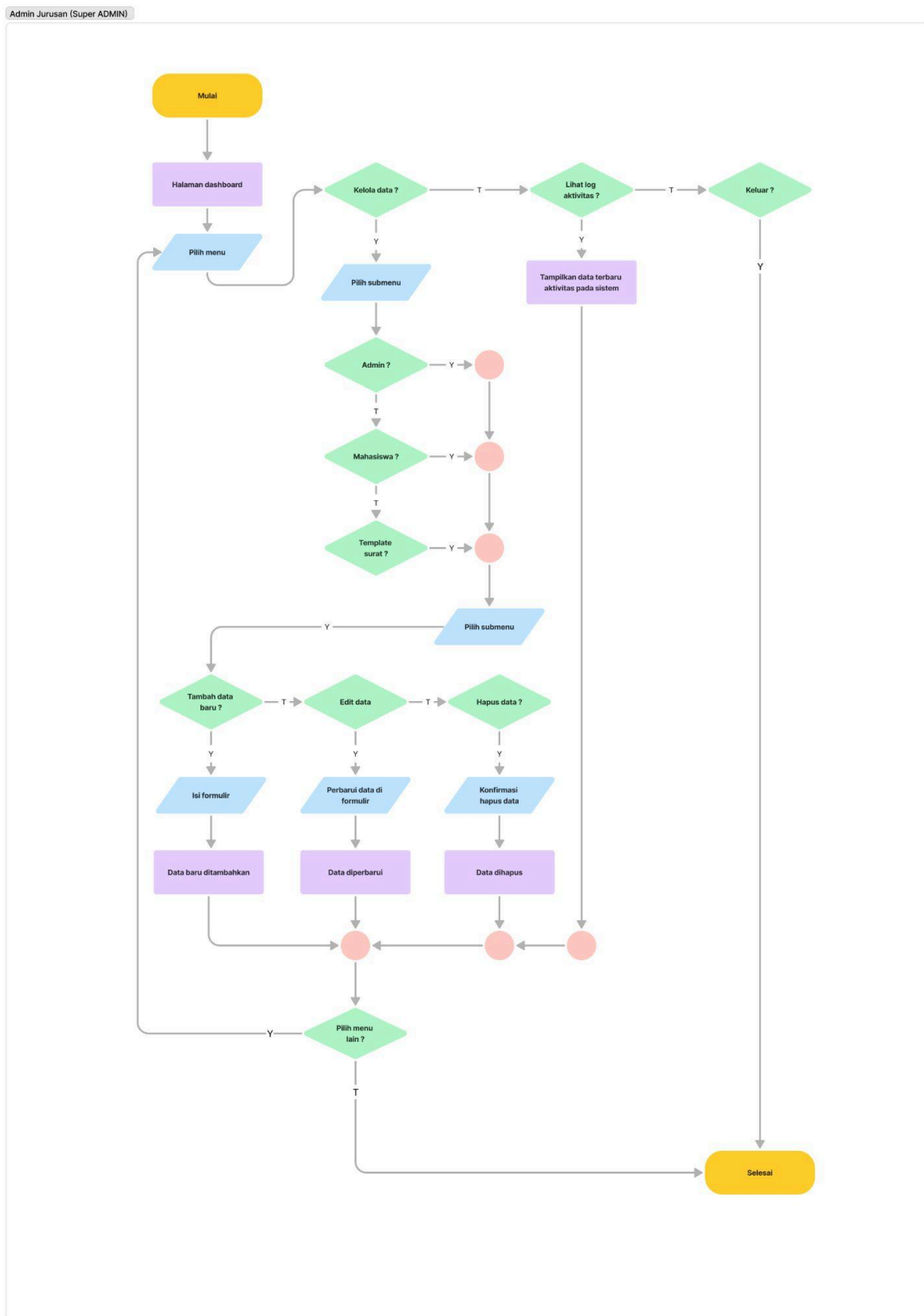


d. Flowchart Admin TA





e. Flowchart Admin Jurusan (Super Admin)



## BAB III ANALISIS KEBUTUHAN PENGGUNA

### 3.1 Kebutuhan Fungsional

Kebutuhan fungsional pada Sistem Bebas Tanggungan Tugas Akhir “Finalis JTI” yaitu sebagai berikut:

- a. Sistem memungkinkan Mahasiswa dapat melakukan input data dokumen yang diperlukan untuk menerbitkan Surat Bebas Tanggungan Tugas Akhir berupa:
  - Input Laporan Tugas Akhir atau Skripsi (berbentuk PDF)
  - Input Program Aplikasi (berbentuk ZIP/RAR)
  - Input Bukti Publikasi Jurnal (berbentuk PDF)
  - Input Laporan Magang (Berbentuk PDF)
  - Input Sertifikat TOEIC (berbentuk PDF)
  - Input Distribusi Laporan Skripsi (berbentuk PDF)
  - Input Surat Bebas Kompen (PDF)
- b. Sistem memungkinkan Admin dapat melakukan verifikasi untuk menerima atau menolak dokumen yang dimasukkan oleh mahasiswa .
- c. Sistem memungkinkan Admin dan Mahasiswa untuk mengganti identitas diri.
- d. Sistem dapat menampilkan status pengajuan dokumen yang diajukan oleh Mahasiswa.
- e. Sistem memungkinkan Mahasiswa untuk mengunduh Surat Bebas Tanggungan Tugas Akhir apabila pengguna telah memenuhi semua dokumen persyaratan yang diperlukan.
- f. Sistem memungkinkan pengguna untuk melakukan *feedback* akan permasalahan yang terjadi di sistem.

### 3.2 Kebutuhan Non-Fungsional

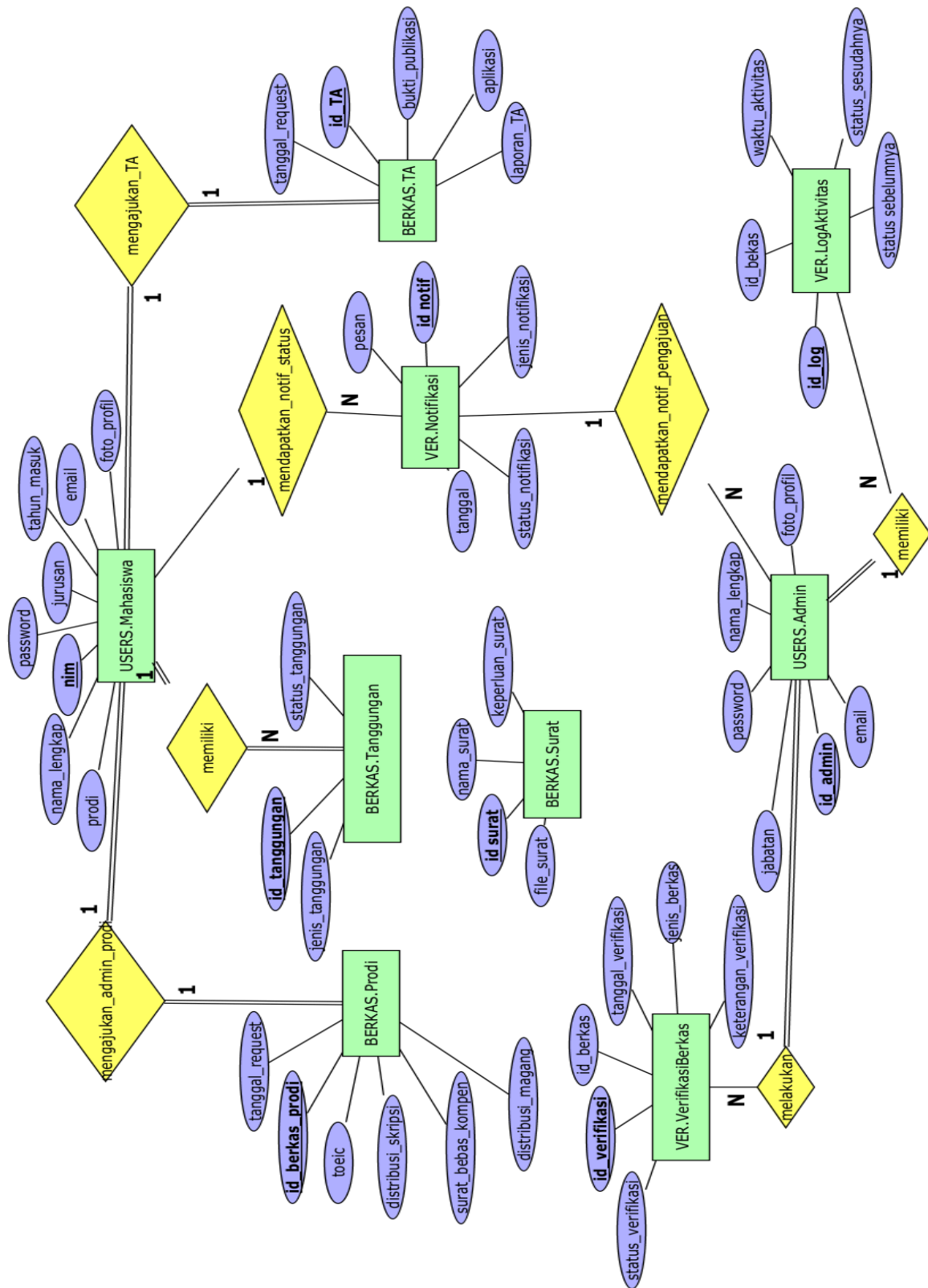
Kebutuhan Non-Fungsional pada Sistem Bebas Tanggungan Tugas Akhir “Finalis JTI” yaitu sebagai berikut:

- a. Keamanan  
Sistem harus aman dari akses yang tidak sah. Artinya, hanya pengguna yaitu Mahasiswa dan Admin yang dapat melakukan *login* pada sistem ini.
- b. Kinerja  
Saat sistem ini diakses secara bersamaan, sistem harus tetap berjalan tanpa adanya penurunan kinerja.
- c. Skalabilitas  
Sistem harus dapat ditingkatkan atau diturunkan sesuai kebutuhan. *Update* dari sistem harus dapat dipertimbangkan apabila terjadi kesalahan yang terjadi dalam sistem.
- d. Tersedianya  
Sistem harus tersedia saat dibutuhkan. Pengembang sistem harus memastikan bahwa sistem ini dapat dibuka dan digunakan setiap saat.
- e. Pemeliharaan  
Sistem harus mudah dirawat dan diperbarui.
- f. Portabilitas  
Sistem harus dapat berjalan pada platform yang berbeda dengan sedikit perubahan.
- g. Keandalan  
Sistem harus dapat diandalkan dan memenuhi persyaratan pengguna.
- h. Kegunaan  
Sistem harus mudah digunakan dan dipahami.

## BAB IV DESAIN BASIS DATA

### 4.1 Desain Konseptual

Desain konseptual mencakup rancangan diagram hubungan antar entitas (*Entity Relationship Diagram / ERD*)



## 4.2 Desain Logis

Melakukan proses mapping ke model relasional, melakukan proses normalisasi, menentukan batasan (constraint) basis data seperti primary key, foreign key, index, dll.

### a. Mapping ke Model Relasional

- Tabel USERS.Mahasiswa

nim<pk>	nama_lengkap	prodi	jurusan	email	password	tahun_masuk	foto_profil
---------	--------------	-------	---------	-------	----------	-------------	-------------

- Tabel USERS.Admin

id_admin <PK>	nama_lengkap	email	password	jabatan	telepon	tanggal_bergabung	foto_profil
---------------	--------------	-------	----------	---------	---------	-------------------	-------------

- Tabel BERKAS.Prodi

id_berkas_prodi<pk>	toeic	distribusi_skripsi	surat_bebas_kompen	distribusi_magang	tanggal_request
---------------------	-------	--------------------	--------------------	-------------------	-----------------

- Tabel BERKAS.TA

id_berkas_TA	laporan_ta	bukti_publicasi	aplikasi	tanggal_request
--------------	------------	-----------------	----------	-----------------

- Tabel BERKAS.Tanggungan

id_tanggungan<pk>	jenis_tanggungan	status_tanggungan
-------------------	------------------	-------------------

- Tabel VER.Notifikasi

id_notifikasi<pk>	pesan	jenis_notifikasi	status_notifikasi	tanggal
-------------------	-------	------------------	-------------------	---------

- Tabel VER.VerifikasiBerkas

id_verifikasi <PK>	id_berkas	status_verifikasi	jenis_berkas	keterangan_verifikasi	status_sebelum
--------------------	-----------	-------------------	--------------	-----------------------	----------------

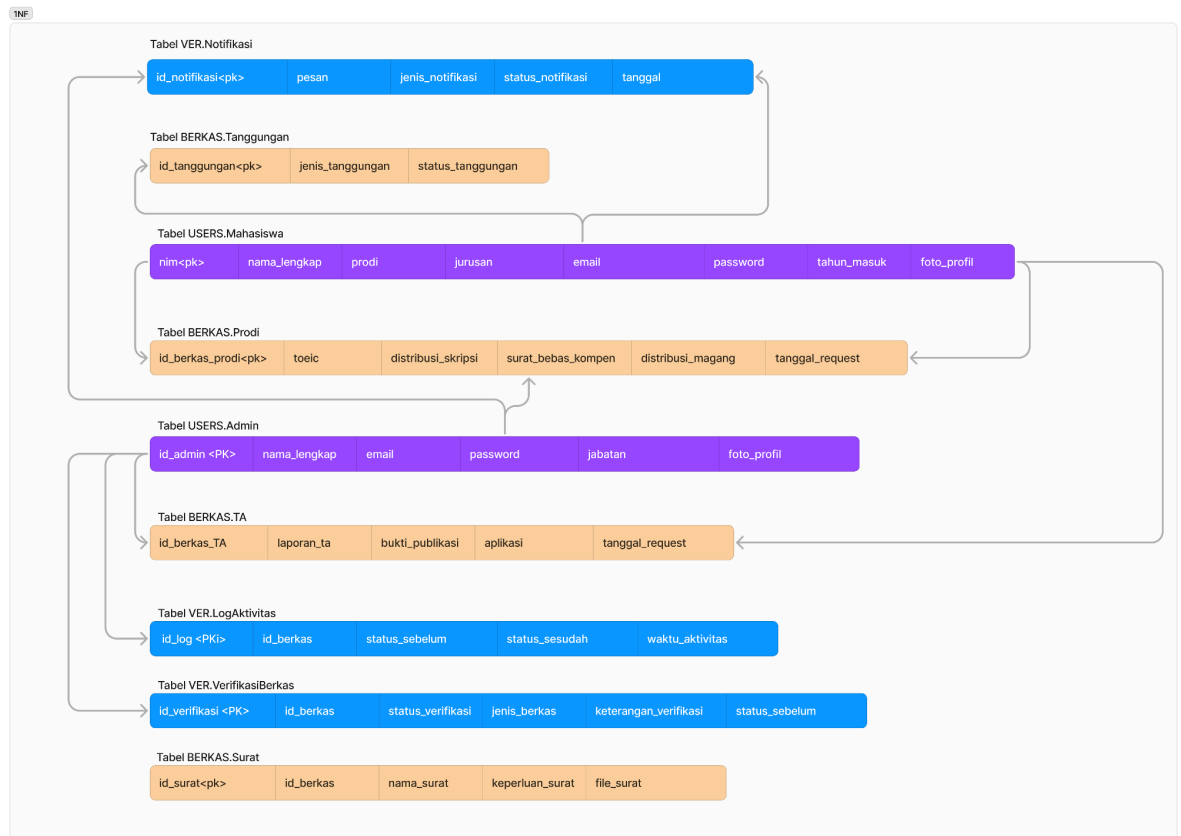
- Tabel VER.LogAktivitas

id_log <PKi>	id_berkas	status_sebelum	status_sesudah	waktu_aktivitas
--------------	-----------	----------------	----------------	-----------------

- Tabel BERKAS.Surat

id_admin <PK>	nama_lengkap	email	password	jabatan	telepon	tanggal_bergabung	foto_profil
---------------	--------------	-------	----------	---------	---------	-------------------	-------------

## b. Normalisasi



## c. Tipe Data yang Digunakan dalam Tabel

### • Tabel USERS.Admin

Nama Kolom	Tipe Data	Constraint
id_admin	VARCHAR(12)	Primary key
nama_lengkap	VARCHAR(50)	Not Null
email	VARCHAR(50)	Not Null
password	VARCHAR(255)	Not Null
jabatan	VARCHAR(20)	Not Null
foto_profil	VARCHAR(MAX)	Not Null

### • Tabel USERS.Mahasiswa

Nama Kolom	Tipe Data	Constraint
nim	VARCHAR(12)	Primary key
nama_lengkap	VARCHAR(50)	Not Null

email	VARCHAR(50)	Not Null
password	VARCHAR(255)	Not Null
prodi	VARCHAR(30)	Not Null
jurusan	VARCHAR(50)	Not Null
tahun_masuk	VARCHAR(10)	Not Null
foto_profil	VARCHAR(MAX)	Not Null

- Tabel BERKAS.Prodi

Nama Kolom	Tipe Data	Constraint
id_berkas_prodi	VARCHAR(10)	Primary key
nim	VARCHAR(12)	Foreign key
tanggal_request	DATE	Not Null
toeic	VARCHAR(MAX)	Not Null
distribusi_skripsi	VARCHAR(MAX)	Not Null
distribusi_magang	VARCHAR(MAX)	Not Null
surat_bebas_kompen	VARCHAR(MAX)	Not Null

- Tabel BERKAS.TA

Nama Kolom	Tipe Data	Constraint
id_ta	VARCHAR(10)	Primary key
nim	VARCHAR(12)	Foreign key
tanggal_request	DATE	Not Null
laporan_ta	VARCHAR(MAX)	Not Null
aplikasi	VARCHAR(MAX)	Not Null
bukti_publicasi	VARCHAR(MAX)	Not Null

- Tabel BERKAS.Tanggungan

Nama Kolom	Tipe Data	Constraint
id_tanggungan	INT	Primary key, Auto_increment
nim	VARCHAR(12)	Foreign Key
jenis_tanggungan	VARCHAR(20)	Not Null
status_tanggungan	VARCHAR(20)	Not Null

- Tabel VER.DetailBerkas

Nama Kolom	Tipe Data	Constraint
id_notifikasi	VARCHAR(20)	Primary key
id_berkas	VARCHAR(10)	Foreign Key
nim	VARCHAR(12)	Foreign Key
tanggal_persetujuan	DATE	Not Null
status	VARCHAR(50)	Not Null
jenis_berkas	VARCHAR(20)	Not Null
keterangan	VARCHAR(MAX)	Not Null

- Tabel VER.Notifikasi

Nama Kolom	Tipe Data	Constraint
id_notifikasi	VARCHAR(20)	Primary key
id_admin	VARCHAR(12)	Foreign Key
nim	VARCHAR(12)	Foreign Key
jenis_notifikasi	VARCHAR(20)	Not Null
status_notifikasi	VARCHAR(20)	Not Null

- VER.LogAktivitas

Nama Kolom	Tipe Data	Constraint
id_log	INT	Primary key
id_admin	VARCHAR(12)	Foreign Key
id_berkas	VARCHAR(20)	Not Null
status_sebelumnya	VARCHAR(50)	Not Null
status_sesudahnya	VARCHAR(50)	Not Null
waktu_aktivitas	DATE	Not Null

- VER.VerifikasiBerkas

Nama Kolom	Tipe Data	Constraint
id_verifikasi	INT	Primary key
id_admin	VARCHAR(12)	Foreign Key
id_berkas	VARCHAR(20)	Not Null
jenis_berkas	VARCHAR(MAX)	Not Null
status_verifikasi	VARCHAR(50)	Not Null
tanggal_verifikasi	DATE	Null
keterangan_verifikasi	VARCHAR(MAX)	Not Null



## BAB V IMPLEMENTASI BASIS DATA

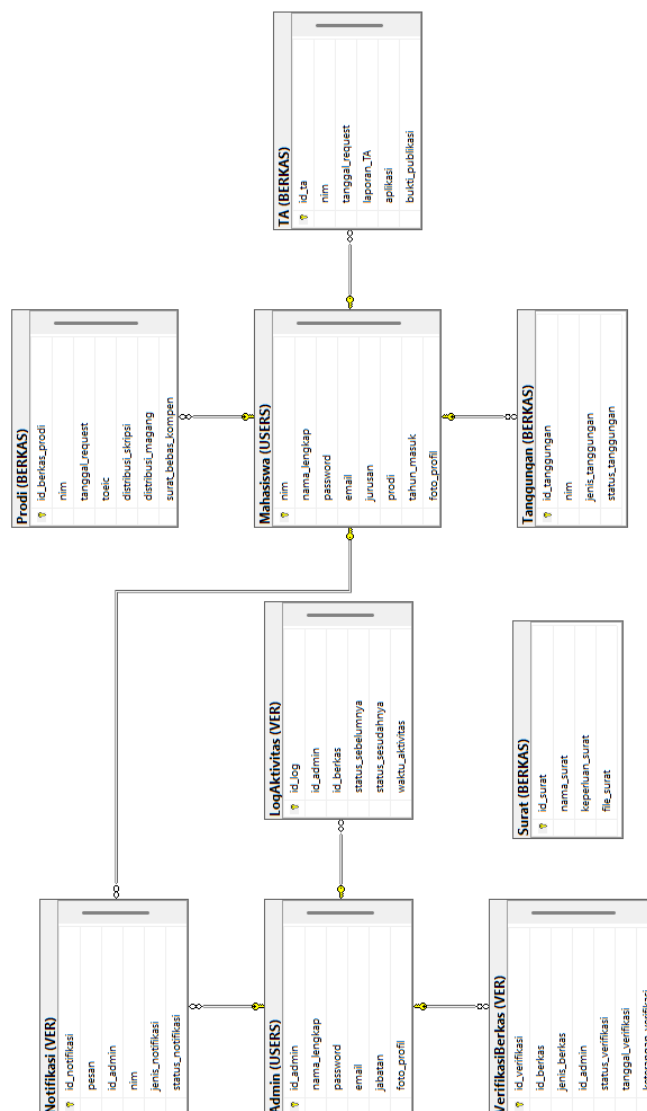
### 5.1 Pemilihan Database Management System (DBMS)

*Database Management System* (DBMS) yang digunakan pada Sistem Bebas Tanggungan Tugas Akhir “Finalis JTI” ini yaitu SQL Server. DBMS tersebut digunakan karena mempertimbangkan kelebihan yang diberikan yaitu sebagai berikut:

- Bekerja sangat baik pada sistem operasi Windows untuk segala versi. Hal tersebut dikarenakan SQL Server dikembangkan sendiri oleh *Microsoft* sehingga dipastikan bisa berjalan dengan baik di semua sistem operasi Windows.
- Memiliki fitur *recovery* dan *restore* data. Hal ini berguna ketika ada sebagian data dari database anda mengalami kerusakan maka dapat dilakukan *recovery* lalu *restore* data.
- Memiliki sistem keamanan dan handal. *Database Administrator* dapat melakukan *login* menggunakan dua cara, yaitu:
  - *Windows Authentication* memungkinkan untuk dapat *login* menggunakan *username* dan *password* pada akun Windows.
  - *SQL Authentication* memungkinkan untuk dapat *login* menggunakan *username* dan *password* yang tersimpan pada SQL Server.

### 5.2 Skema Basis Data

Keterikatan Antar-Tabel dalam Sistem



## 5.3 Implementasi Basis Data Dalam Fitur Sistem

### a. *Data Definition Language (DDL)*

- Store Procedure untuk Insert BERKAS.TA

```
CREATE PROCEDURE sp_InsertBerkasTA
    @nim VARCHAR(12),
    @laporan_TA VARCHAR(MAX),
    @aplikasi VARCHAR(MAX),
    @bukti_publicasi VARCHAR(MAX),
    @id_admin VARCHAR(12)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @last_id VARCHAR(10);
    DECLARE @new_id VARCHAR(10);
    DECLARE @next_number INT;

    -- Ambil ID terakhir
    SELECT @last_id = MAX(id_ta) FROM BERKAS.TA;

    -- Angka dari ID terakhir dan tambahkan 1
    SET @next_number = ISNULL(CAST(SUBSTRING(@last_id, 3,
        LEN(@last_id) - 2) AS INT), 0) + 1;

    -- Buat ID baru dengan format "TA" + angka yang di-increment
    SET @new_id = 'TA' + RIGHT('0000' + CAST(@next_number AS
        VARCHAR), 4);

    -- Masukkan data baru ke tabel
    INSERT INTO BERKAS.TA (id_ta, nim, laporan_TA, aplikasi,
        bukti_publicasi)
        VALUES (@new_id, @nim, @laporan_TA, @aplikasi,
            @bukti_publicasi);

    --Masukkan data tanggungan
    INSERT INTO BERKAS.Tanggungan (nim, jenis_tanggungan,
        status_tanggungan)
        VALUES (@nim, 'Tanggungan TA', 'Belum Selesai')

    -- Menyisipkan data ke tabel VerifikasiBerkas
    INSERT INTO VER.VerifikasiBerkas (id_berkas, jenis_berkas,
        id_admin, status_verifikasi, tanggal_verifikasi,
        keterangan_verifikasi)
        VALUES (@new_id, 'Prodi', @id_admin, 'Diajukan', NULL ,
            'Berkas TA telah diajukan untuk verifikasi.');
```

END;

- Store Procedure untuk generate ID BERKAS.Prodi

```

CREATE PROCEDURE sp_InsertBerkasProdi
    @nim VARCHAR(12),
    @toeic VARCHAR(MAX),
    @distribusi_skripsi VARCHAR(MAX),
    @distribusi_magang VARCHAR(MAX),
    @surat_bebas_kompen VARCHAR(MAX),
    @id_admin VARCHAR(12)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @last_id VARCHAR(10);
    DECLARE @new_id VARCHAR(10);
    DECLARE @next_number INT;

    -- Ambil ID terakhir
    SELECT @last_id = MAX(id_berkas_prodi) FROM BERKAS.Prodi;

    SET @next_number = ISNULL(CAST(SUBSTRING(@last_id, 3,
    LEN(@last_id) - 2) AS INT), 0) + 1;

    SET @new_id = 'PR' + RIGHT('0000' + CAST(@next_number AS
    VARCHAR), 4);

    INSERT INTO BERKAS.Prodi (id_berkas_prodi, nim, toeic,
    distribusi_skripsi, distribusi_magang, surat_bebas_kompen)
    VALUES (@new_id, @nim, @toeic, @distribusi_skripsi,
    @distribusi_magang, @surat_bebas_kompen);

    INSERT INTO BERKAS.Tanggungans (nim, jenis_tanggungans,
    status_tanggungans)
    VALUES (@nim, 'Tanggungans Prodi', 'Belum Selesai')

    INSERT INTO VER.VerifikasiBerkas (id_berkas, jenis_berkas,
    id_admin, status_verifikasi, tanggal_verifikasi,
    keterangan_verifikasi)
    VALUES (@new_id, 'Prodi', @id_admin, 'Diajukan', NULL ,
    'Berkas Prodi telah diajukan untuk verifikasi.');
```

END;

- **Store Procedure untuk update Status Verifikasi Prodi**

```

CREATE PROCEDURE sp_UpdateVerifikasiBerkasProdi
    @id_berkas VARCHAR(10),
    @status_verifikasi VARCHAR(50),
    @keterangan_verifikasi VARCHAR(MAX),
    @id_admin VARCHAR(12)
AS
BEGIN
    -- Update status verifikasi
```

```

UPDATE VER.VerifikasiBerkas
SET
    status_verifikasi =
        CASE
            WHEN @status_verifikasi = 'Disetujui' THEN
'Disetujui'
            WHEN @status_verifikasi = 'Ditolak' THEN
'Ditolak'
            ELSE 'Diajukan'
        END,
    tanggal_verifikasi = GETDATE(),
    keterangan_verifikasi = @keterangan_verifikasi
WHERE id_berkas = @id_berkas AND jenis_berkas = 'Prodi';

-- Menyimpan aktivitas ke dalam log
INSERT INTO VER.LogAktivitas (id_admin, id_berkas,
status_sesudahnya, waktu_aktivitas)
VALUES
    (@id_admin, @id_berkas, @status_verifikasi, GETDATE());
END;

```

- **Store Procedure untuk update Status Verifikasi TA**

```

CREATE PROCEDURE sp_UpdateVerifikasiBerkasTA
    @id_berkas VARCHAR(10),
    @status_verifikasi VARCHAR(50),
    @keterangan_verifikasi VARCHAR(MAX),
    @id_admin VARCHAR(12)
AS
BEGIN
    -- Update status verifikasi
    UPDATE VER.VerifikasiBerkas
    SET
        status_verifikasi =
            CASE
                WHEN @status_verifikasi = 'Disetujui' THEN
'Disetujui'
                WHEN @status_verifikasi = 'Ditolak' THEN
'Ditolak'
                ELSE 'Diajukan'
            END,
        tanggal_verifikasi = GETDATE(),
        keterangan_verifikasi = @keterangan_verifikasi
    WHERE id_berkas = @id_berkas AND jenis_berkas = 'TA';

    -- Menyimpan aktivitas ke dalam log
    INSERT INTO VER.LogAktivitas (id_admin, id_berkas,
status_sesudahnya, waktu_aktivitas)
    VALUES
        (@id_admin, @id_berkas, @status_verifikasi, GETDATE());

```

END;

b. *Data Manipulation Language (DML)*

- **Fitur General (Login)**

```
-- Login untuk mahasiswa
SELECT nim, nama_lengkap, password, 'mahasiswa' AS role
FROM USERS.Mahasiswa
WHERE nim = '?' AND password = '?'
```

UNION

```
-- Login untuk admin
SELECT id_admin, nama_lengkap, password, jabatan
FROM USERS.Admin
WHERE id_admin = '?' AND password = '?';
```

- **Fitur Mahasiswa**

- **Fitur Riwayat Tanggungan**

```
SELECT nim, jenis_tanggungan, status_tanggungan
FROM
BERKAS.Tanggungan
WHERE nim = '?';
```

- **Fitur Riwayat Pengajuan Surat**

```
SELECT ROW_NUMBER() OVER (ORDER BY tanggal_request ASC) AS No,
       tanggal_request AS 'Tanggal Request',
       jenis_berkas AS 'Jenis Formulir',
       status_verifikasi AS 'Status',
       keterangan_verifikasi AS 'Keterangan'
FROM (
  SELECT P.tanggal_request,
         V.status_verifikasi,
         V.keterangan_verifikasi,
         'Formulir Administrasi Prodi' AS jenis_berkas
  FROM VER.VerifikasiBerkas V
  INNER JOIN BERKAS.Prodi P ON P.id_berkas_prodi = V.id_berkas
  WHERE P.nim = '?'
  UNION ALL
  SELECT T.tanggal_request,
         V.status_verifikasi,
         V.keterangan_verifikasi,
         'Formulir Tugas Akhir' AS jenis_berkas
  FROM VER.VerifikasiBerkas V
  INNER JOIN BERKAS.TA T ON T.id_ta = V.id_berkas
  WHERE T.nim = '?'
) AS Combined
```

```
ORDER BY tanggal_request ASC;
```

- **Fitur Admin Prodi**

- **Mem-verifikasi Pengajuan Tanggungan Prodi**

```
EXEC sp_UpdateVerifikasiBerkasProdi
    @id_berkas = '?',
    @status_verifikasi = '?',
    @keterangan_verifikasi = '?',
    @id_admin = '?';
```

- **Melihat seluruh jumlah pengajuan surat tanggungan prodi**

```
SELECT COUNT(*) AS total_pengajuan_Prodi
    FROM VER.VerifikasiBerkas
    WHERE jenis_berkas = 'Prodi';
```

- **Melihat jumlah pengajuan surat tanggungan prodi yang telah disetujui**

```
SELECT COUNT(*) AS total_pengajuan_Prodi
    FROM VER.VerifikasiBerkas
    WHERE jenis_berkas = 'Prodi' AND status_verifikasi =
'Disetujui';
```

- **Melihat jumlah pengajuan surat tanggungan prodi yang ditolak**

```
SELECT COUNT(*) AS total_pengajuan_Prodi
    FROM VER.VerifikasiBerkas
    WHERE jenis_berkas = 'Prodi' AND status_verifikasi =
'Ditolak';
```

- **Menampilkan Rincian Pengajuan Berkas Prodi**

```
SELECT
    M.nim AS NIM,
    M.nama_lengkap AS 'Nama Lengkap',
    V.jenis_berkas AS 'Jenis',
    P.surat_bebas_kompen AS 'Surat Bebas Kompen',
    P.distribusi_skripsi AS 'Distribusi Skripsi',
    P.distribusi_magang AS 'Distribusi Magang',
    P.toeic AS 'TOEIC'
FROM USERS.Mahasiswa M
INNER JOIN BERKAS.Prodi P ON M.nim = P.nim
INNER JOIN VER.VerifikasiBerkas V ON V.id_berkas =
P.id_berkas_prodi
WHERE M.nim = '?';
```

- **Menampilkan Top 3 Pengajuan Berkas Prodi**

```
SELECT TOP 3
    M.nim AS NIM,
    M.nama_lengkap AS 'Mahasiswa',
    V.keterangan_verifikasi AS 'Aktivitas',
```

```

        P.tanggal_request AS 'Tanggal Request'
FROM USERS.Mahasiswa M
INNER JOIN BERKAS.Prodi P ON M.nim = P.nim
INNER JOIN VER.VerifikasiBerkas V ON V.id_berkas =
P.id_berkas_prodi
ORDER BY P.tanggal_request DESC;

```

- **Fitur Admin TA**

- **Mem-verifikasi Pengajuan Tanggungan TA**

```

EXEC sp_UpdateVerifikasiBerkasProdi
    @id_berkas = '?',
    @status_verifikasi = '?',
    @keterangan_verifikasi = '?',
    @id_admin = '?';

```

- **Melihat seluruh pengajuan tanggungan TA**

```

SELECT COUNT(*) AS total_pengajuan_Prodi
    FROM VER.VerifikasiBerkas
    WHERE jenis_berkas = 'TA';

```

- **Melihat jumlah pengajuan tanggungan TA yang telah disetujui**

```

SELECT COUNT(*) AS total_pengajuan_TA
    FROM VER.VerifikasiBerkas
    WHERE jenis_berkas = 'TA' AND status_verifikasi =
'Disetujui';

```

- **Melihat jumlah pengajuan tanggungan TA yang ditolak**

```

SELECT COUNT(*) AS total_pengajuan_TA
    FROM VER.VerifikasiBerkas
    WHERE jenis_berkas = 'TA' AND status_verifikasi =
'Ditolak';

```

- **Menampilkan Rincian Pengajuan Berkas TA**

```

SELECT
    M.nim AS NIM,
    M.nama_lengkap AS 'Mahasiswa',
    V.keterangan_verifikasi AS 'Aktivitas',
    T.tanggal_request AS 'Tanggal Request'
FROM USERS.Mahasiswa M
INNER JOIN BERKAS.TA T ON M.nim = T.nim
INNER JOIN VER.VerifikasiBerkas V ON V.id_berkas = T.id_ta;

```

- **Menampilkan TOP 3 Pengajuan Berkas TA**

```

SELECT TOP 3
    M.nim AS NIM,
    M.nama_lengkap AS 'Mahasiswa',
    V.keterangan_verifikasi AS 'Aktivitas',

```

```

        T.tanggal_request AS 'Tanggal Request'
FROM USERS.Mahasiswa M
INNER JOIN BERKAS.TA T ON M.nim = T.nim
INNER JOIN VER.VerifikasiBerkas V ON V.id_berkas = T.id_ta
ORDER BY T.tanggal_request DESC;

```

- **Fitur Admin Jurusan (SuperAdmin)**

- **Menambahkan Admin**

```

INSERT INTO USERS.Admin (id_admin, nama_lengkap, password,
email, jabatan, foto_profil)
VALUES
(?, ?, ?, ?, ?, ?);

```

- **Menghapus Admin**

```

DELETE USERS.Admin
WHERE id_admin = '';

```

- **Meng-update Data Admin**

```

UPDATE USERS.Admin
SET
    id_admin = '?',
    nama_lengkap = '?',
    password = '?',
    email = '?',
    jabatan = '?',
    foto_profil = '?'
WHERE id_admin = '?';

```

- **Menampilkan Semua Data Admin**

```

SELECT
    nama_lengkap,
    email,
    tanggal_bergabung
FROM USERS.Admin;

```

- **Menambahkan Data Mahasiswa**

```

INSERT INTO USERS.Mahasiswa
(nim, nama_lengkap, password, email, jurusan, prodi,
tahun_masuk, foto_profil)
VALUES
(?, ?, ?, ?, ?, ?, ?, ?)

```

- **Men-update Data Mahasiswa**

```

Update USERS.Mahasiswa
SET
    nim = `?`,
    nama_lengkap = `?`,
    password = `?`,

```



```
email = '?',
jurusan = '?',
prodi = '?',
tahun_masuk = '?',
foto_profil = '?';
```

- **Menghapus Data Mahasiswa**

```
DELETE FROM USERS.Mahasiswa
WHERE nim = '?';
```

- **Melihat jumlah seluruh pengajuan**

```
SELECT COUNT(*) AS total_pengajuan
FROM VER.VerifikasiBerkas;
```

- **Melihat jumlah pengajuan yang telah disetujui**

```
SELECT COUNT(*) AS total_pengajuan
FROM VER.VerifikasiBerkas
WHERE status_verifikasi = 'Disetujui';
```

- **Melihat jumlah pengajuan yang ditolak**

```
SELECT COUNT(*) AS total_pengajuan
FROM VER.VerifikasiBerkas
```

## 5.4 Peran Pengguna dan Perizinan

### a. Peran Pengguna Sistem

#### 1. Mahasiswa

Mahasiswa menggunakan sistem untuk:

- **Fitur Riwayat Tanggungan:** Melihat daftar tanggungan seperti TOEIC, distribusi skripsi, dan surat bebas kompen.
- **Fitur Status Permintaan Surat:** Memantau status pengajuan surat, seperti apakah sudah disetujui atau sedang diproses.
- **Fitur Riwayat Pengajuan Surat:** Mengakses riwayat surat yang pernah diajukan dan statusnya.
- **Fitur Verifikasi Dokumen:** Mengunggah dokumen terkait persyaratan pengajuan surat atau tugas akhir.

#### 2. Admin TA

Admin TA bertanggung jawab untuk:

- Memverifikasi dokumen yang diajukan mahasiswa terkait tugas akhir (skripsi).
- Mengatur status persetujuan atau penolakan tanggungan tugas akhir.
- Melakukan pelaporan data pengajuan surat tugas akhir.

#### 3. Admin Prodi

Admin Prodi memiliki peran untuk:

- Memastikan dokumen mahasiswa terkait program studi, seperti distribusi magang, telah sesuai.
- Memproses verifikasi atau penolakan dokumen di tingkat program studi.
- Mengelola laporan pengajuan surat dari mahasiswa dalam lingkup program studi.

#### 4. **Admin Jurusan**

Admin Jurusan bertanggung jawab atas:

- Memastikan koordinasi antara admin prodi dan admin TA berjalan dengan baik.
- Memantau dan mengelola data mahasiswa di lingkup jurusan.
- Melakukan audit sistem terkait proses pengajuan dan verifikasi dokumen di tingkat jurusan.

#### b. Perizinan Akses

##### 1. **Mahasiswa:**

- Akses terbatas hanya ke data pribadi, daftar tanggungan, status pengajuan surat, dan riwayat pengajuan.

##### 2. **Admin TA:**

- Akses penuh ke data pengajuan tugas akhir mahasiswa.
- Izin memverifikasi, menolak, atau menyetujui pengajuan terkait tugas akhir.

##### 3. **Admin Prodi:**

- Akses ke data pengajuan dokumen mahasiswa dalam lingkup program studi.
- Izin memverifikasi, menolak, atau menyetujui pengajuan program studi.

##### 4. **Admin Jurusan:**

- Akses penuh ke seluruh data pengajuan dokumen di tingkat jurusan.
- Izin untuk melakukan audit dan memastikan proses berjalan sesuai aturan.

## **BAB VI PENUTUP**

Demikian adalah laporan perkembangan proyek desain basis data sistem informasi bebas tanggungan tugas akhir “Finalis JTI”. Laporan ini disusun sebagai bagian dari implementasi *Project-Based Learning* (PBL) akhir semester 3. Kami berharap hasil proyek ini dapat memberikan kontribusi positif dalam pengembangan keterampilan dan pemahaman terkait desain basis data. Kritik dan saran dari dosen sangat kami harapkan agar proyek ini dapat diselesaikan dengan lebih baik sesuai target yang telah direncanakan

## DAFTAR PUSTAKA

- MateriDosen. 22 Maret 2017. *Perbedaan Kebutuhan Fungsional dan Non Fungsional, Lengkap Contoh dan Penjelasan*. Diakses pada tanggal 24 November 2024 dari <https://www.materidosen.com/2017/03/perbedaan-kebutuhan-fungsional-dan-non.html>
- Potgieter, Jan. 10 November 2022. *Setup Security for a SQL Server Database via SSMS and T-SQL*. Diakses pada tanggal 24 November 2024 dari <https://www.mssqltips.com/sqlservertip/7447/sql-server-database-security-logins-users/>