

## BAB III

### PELAKSANAAN KERJA

#### 3.1 Kedudukan dan Koordinasi

##### 3.1.1 Kedudukan

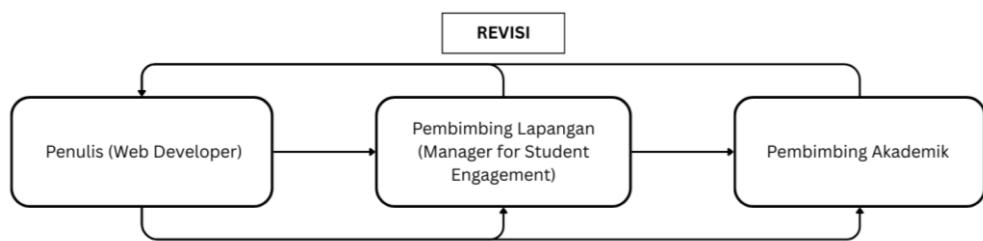
Selama pelaksanaan magang, penulis menempati posisi sebagai Web Developer di lingkungan Universitas Multimedia Nusantara (UMN), khususnya pada unit *Student Development* yang berada di bawah naungan *Student Engagement*. Divisi ini membawahi tiga sub divisi, yaitu *Student Service*, *Student Support*, dan *Student Development* yang berperan dalam mengelola berbagai layanan mahasiswa termasuk pengembangan sistem internal.

Sebagai Web Developer, penulis berperan dalam proses pengembangan dan pemeliharaan sistem berbasis web yang mendukung kegiatan operasional di bawah *Student Development*. Kegiatan magang dilaksanakan secara *Work From Office* (WFO) di kampus UMN, Tangerang, Banten dengan koordinasi langsung bersama pembimbing lapangan dari pihak *Manager for Student Engagement* dan dosen pembimbing akademik dari program studi Sistem Informasi.

##### 3.1.2 Koordinasi

Alur koordinasi selama kegiatan magang dilakukan secara berjenjang dan terstruktur. Penulis melakukan pelaporan dan diskusi teknis terkait proyek yang dikerjakan kepada pembimbing lapangan dari *Student Development*. Komunikasi dan koordinasi dilakukan secara langsung di ruang kerja maupun melalui media komunikasi seperti email dan grup internal. Selain itu, penulis bekerja sama dengan peserta magang lainnya, yaitu Olivia Tsania Putri, L., yang bertugas sebagai *Web Designer* selama proses desain antarmuka sistem. Untuk memastikan desain akhir sesuai dengan kebutuhan pengguna, kolaborasi ini berfokus pada persiapan tata

letak dan penyesuaian elemen antarmuka pengguna (UI/UX). Apabila terdapat hal yang memerlukan persetujuan atau arahan lebih lanjut, pembimbing lapangan yang menjabat sebagai *Manager for Student Engagement* akan memberikan pengarahan secara langsung kepada penulis serta berkoordinasi dengan pihak terkait apabila dibutuhkan. Sementara itu, pembimbing akademik memantau perkembangan kegiatan magang melalui laporan dan sesi konsultasi berkala.



**Gambar 3.1 Bagan Alur Koordinasi**

Pada gambar alur tersebut menggambarkan proses koordinasi dan supervisi selama kegiatan magang berlangsung. Penulis berperan sebagai Web Developer melakukan koordinasi secara langsung dengan pembimbing lapangan, yaitu *Manager for Student Engagement*, terkait pelaksanaan tugas dan pengembangan proyek. Setiap hasil pekerjaan yang telah diselesaikan akan dikonsultasikan kepada pembimbing lapangan untuk mendapatkan evaluasi dan *feedback*.

Selanjutnya, pembimbing lapangan menyampaikan laporan perkembangan magang kepada pembimbing akademik dari Universitas Multimedia Nusantara untuk memastikan kesesuaian kegiatan magang dengan tujuan akademik program studi. Apabila terdapat masukan atau revisi dari pembimbing akademik, proses tindak lanjut dapat dilakukan melalui pembimbing lapangan atau langsung pada penulis untuk menyampaikan masukannya.

### 3.2 Tugas yang Dilakukan

**Tabel 3.1** Detail Pekerjaan yang Dilakukan

No.	Minggu	Proyek	Keterangan
<b>Proyek 1</b>			
1	1-3	Perencanaan sistem <i>ticketing</i> sesuai kebutuhan <i>staff</i>	Menyusun alur kerja sistem <i>ticketing</i> dan melakukan diskusi antar sub-divisi
2	3-7	Desain tampilan sistem <i>ticketing</i> pelaporan mahasiswa	Melakukan desain tampilan <i>website ticketing</i> pelaporan melalui <i>Figma</i> sesuai <i>feedback</i> <i>staff</i> sub-divisi
3	4-10	Mengembangkan <i>website ticketing</i> pelaporan	Mengerjakan koding pembuatan <i>website ticketing</i> pelaporan melalui <i>Visual Studio Code</i>
4	4-11	Pengecekan dan perbaikan <i>website ticketing</i> pelaporan	Melakukan pengecekan dan perbaikan fungsionalitas <i>website</i> agar terhindar <i>error</i>
5	11	<i>Website ticketing deployment</i>	Melakukan <i>deployment</i> secara publik untuk pengecekan penggunaan <i>website ticketing</i> pelaporan
6	11-13	<i>Maintenance website</i>	Melakukan pemeliharaan sistem untuk penyesuaian pada terhadap tampilan antarmuka, optimasi <i>query database</i> , dll
<b>Proyek 2</b>			
7	10	Analisis kebutuhan sistem pengajuan SKKM	Melakukan analisis kebutuhan sistem untuk pengajuan SKKM bagi mahasiswa
8	10-11	Perancangan alur kerja sistem pengajuan SKKM	Menyusun alur kerja sistem pengajuan SKKM
9	11-13	Desain UI/UX tampilan <i>website</i> pengajuan SKKM	Melakukan desain tampilan pengajuan SKKM melalui <i>Figma</i>

### 3.3 Uraian Pelaksanaan Kerja

Selama pelaksanaan magang di Universitas Multimedia Nusantara, penulis terlibat dalam dua proyek utama yang berkaitan dengan pengembangan sistem berbasis web di bawah unit *Student Engagement*. Kedua proyek tersebut bertujuan mendukung proses digitalisasi layanan internal kampus, khususnya pada sub divisi *Student Service, Student Support, dan Student Development*.

#### A. Pencatatan Laporan Mahasiswa melalui Sistem Ticketing Internal

Proyek pertama berfokus pada pencatatan dan manajemen laporan yang masuk ke tiga subdivisi di bawah Student Engagement, yaitu *Student Service, Student Support, dan Student Development*. Sebelum adanya sistem, proses pelaporan dilakukan secara manual melalui email atau pesan pribadi, sehingga sulit melakukan pelacakan status laporan dan dokumentasi.

Tahapan pengembangan sistem ini mengikuti metode *Software Development Life Cycle (SDLC)* yang meliputi:

##### 1. *Planning*

Tahap perencanaan dimulai dengan mengidentifikasi masalah pada proses pelaporan manual yang digunakan oleh setiap sub divisi. Penulis bersama pembimbing lapangan melakukan diskusi untuk menentukan ruang lingkup proyek, tujuan pengembangan, serta target hasil akhir yang diharapkan.

Tujuan utama dari sistem ini adalah menciptakan platform yang mampu mencatat, mengelola, dan memantau laporan mahasiswa secara terstruktur dan terdokumentasi. Selain itu, beberapa *tools* digunakan sesuai kebutuhan untuk pengembangan web seperti perangkat lunak (PHP, MySQL, dan XAMPP), serta penjadwalan kegiatan pengembangan berdasarkan timeline magang.

## 2. *Analysis*

Pada tahap analisis, penulis melakukan observasi terhadap alur kerja pelaporan di tiga sub divisi dan melakukan diskusi singkat dengan *staff Student Engagement* untuk mengetahui kebutuhan utama sistem.

Hasil analisis menunjukkan bahwa setiap sub divisi menerima jenis laporan yang berbeda-beda, namun membutuhkan sistem yang sama dalam hal pencatatan, penentuan penanggung jawab, serta pemantauan status tiket. Kebutuhan sistem dibagi menjadi dua bagian:

1. **Kebutuhan Fungsional:** pembuatan tiket laporan, pembaruan status tiket, pencatatan tindak lanjut, dan notifikasi otomatis.
2. **Kebutuhan Non-Fungsional:** kemudahan penggunaan, keamanan data laporan, dan kecepatan akses sistem.

## 3. *Design*

Setelah kebutuhan sistem dianalisis, penulis melakukan perancangan arsitektur sistem, struktur *database*, dan tampilan antarmuka pengguna. Perancangan dilakukan menggunakan *Figma* untuk tampilan antarmuka (UI) dan *Canva* untuk diagram alur kerja sistem. Desain yang dihasilkan mencakup:

1. Halaman login untuk setiap sub divisi.
2. Formulir tiket laporan yang berisi informasi pelapor, jenis laporan, dan deskripsi masalah.
3. Halaman *dashboard* untuk melihat, mengedit dan menghapus status setiap laporan.
4. Struktur *database* MySQL yang mencakup tabel *activity\_log*, *laporan*, *mahasiswa*, *report\_notes*, dan *users*.

Desain tersebut dibuat dengan mempertimbangkan tampilan yang sederhana agar mudah digunakan oleh *staff Student Engagement*.

#### **4. *Implementation***

Tahap implementasi dilakukan dengan menghasilkan desain menjadi program yang berfungsi sebagai web menggunakan PHP, HTML, CSS, dan JavaScript. Fitur-fitur utama yang diimplementasikan meliputi:

1. Halaman login dan autentikasi pengguna.
2. Halaman pembuatan tiket laporan mahasiswa.
3. *Dashboard* sub divisi untuk menampilkan tiket yang masuk berdasarkan kategori.
4. Fungsi pembaruan status tiket (*pending, assign, resolved, done*).
5. Riwayat tindak lanjut dan komentar untuk setiap tiket.

Selama tahap ini, penulis juga melakukan integrasi antara antarmuka pengguna dan *database MySQL* agar data dapat tersimpan dan diakses secara dinamis.

Dalam tahap implementasi, selain bahasa pemrograman utama seperti PHP, HTML, CSS, dan JavaScript, penulis juga memanfaatkan beberapa *library* eksternal untuk meningkatkan fungsionalitas sistem, khususnya pada halaman *dashboard* dalam proyek file kode (*dashboard\_report.php*). Beberapa di antaranya yaitu *Font Awesome* untuk ikon antarmuka, *Chart.js* untuk visualisasi grafik laporan, serta *ExcelJS* dan *FileSaver.js* untuk mendukung fitur ekspor data laporan ke dalam format Excel. Penggunaan *library* ini bertujuan untuk menambahkan tampilan visual, mempermudah pembuatan laporan statistik, serta mendukung efisiensi pengguna dalam mengunduh data laporan secara langsung dari *dashboard*.

## 5. *Testing & Integration*

Setelah semua fitur utama selesai dibangun, tahapan selanjutnya adalah melakukan pengujian untuk memastikan bahwa sistem benar-benar berfungsi seperti yang direncanakan. Pengujian ini dilakukan untuk melihat apakah proses *input* data laporan, penyimpanan informasi ke dalam *database*, sampai pembaruan status tiket dapat berjalan tanpa hambatan. Tujuannya adalah untuk memastikan bahwa setiap bagian sistem memberikan *respond* yang tepat sesuai dengan kebutuhan pengguna.

Tahap pengujian diawali dengan memahami kembali alur dan kebutuhan sistem, terutama terkait proses pelaporan mahasiswa dan mekanisme pembaruan status pada masing-masing sub divisi. Setelah itu, dilakukan uji coba dengan berbagai jenis *input*, seperti mengisi formulir laporan, mengubah status tiket, hingga meneruskan laporan ke divisi tertentu. Dari setiap percobaan tersebut, penulis memerhatikan apakah data berhasil tersimpan, apakah status langsung muncul di *dashboard*, dan apakah sistem memberikan konfirmasi ketika suatu tindakan berhasil dilakukan. Beberapa skenario diuji, misalnya membuat laporan baru, mengisi *form* dengan data yang berbeda-beda, memperbarui status menjadi *assign* maupun *resolved*, dan melihat daftar laporan yang ditampilkan sesuai dengan divisi pengguna.

Berdasarkan hasil uji coba tersebut, sistem dapat berjalan dengan baik dan tidak ditemukan kendala. Seluruh fungsi utama bekerja sesuai dengan alurnya, dan koneksi antara tampilan antarmuka dengan *database* MySQL juga berjalan stabil. Hal ini menunjukkan bahwa sistem sudah siap digunakan dalam mendukung proses pelaporan dan pendataan kegiatan mahasiswa di lingkungan Student Engagement.

## 6. *Deployment*

Setelah sistem selesai diuji dan berjalan sesuai kebutuhan pengguna, dilakukan tahap *deployment* agar program dapat digunakan secara publik. Proses ini dilakukan dengan menggunakan layanan *hosting* gratis *InfinityFree* yang mendukung PHP dan MySQL.

Melalui layanan ini, aplikasi web *ticketing* internal dapat diakses secara daring melalui tautan: <https://student-engagement.infinityfree.me/>.

Tahap ini melibatkan proses unggah *file* ke *server hosting*, konfigurasi *database*, serta penyesuaian pada *file* proyek yang terkoneksi dengan *database* agar sistem dapat berjalan dengan baik di lingkungan server publik.

## 7. *Maintenance*

Tahap akhir adalah pemeliharaan sistem yang dilakukan selama masa uji coba internal. Pemeliharaan meliputi perbaikan terhadap tampilan antarmuka, optimasi *query database* agar proses pemanggilan data lebih cepat, serta penyesuaian label dan kategori tiket berdasarkan *feedback* pengguna. Selain itu, penulis juga menyiapkan dokumentasi teknis yang berisi panduan penggunaan sistem dan struktur *database* untuk memudahkan proses pengembangan lanjutan oleh tim internal UMN di masa mendatang.

## B. Perancangan Website Ticketing untuk Pengajuan SKKM Mahasiswa

Proyek kedua berfokus pada perancangan sistem ticketing untuk proses pengajuan SKKM (Satuan Kredit Kegiatan Mahasiswa). Tujuannya adalah menyediakan platform terpusat yang memudahkan mahasiswa dalam mengajukan permohonan validasi kegiatan tanpa harus datang langsung ke *Student Development*. Proyek ini dikerjakan sampai pada tahap perancangan alur kerja dan antarmuka (UI/UX) tanpa implementasi penuh atau *deployment*. Tahapan yang dikerjakan antara lain:

### 1. Analisis Kebutuhan Sistem

Mengidentifikasi kendala proses pengajuan SKKM secara manual dan menentukan kebutuhan sistem seperti unggah bukti kegiatan, status verifikasi, dan notifikasi.

## 2. Perancangan Alur Kerja (*Flowchart*)

Membuat diagram alur pengajuan SKKM dari tahap pembuatan tiket oleh mahasiswa hingga proses pengajuan dari mahasiswa selesai atau diterima oleh sistem.

## 3. Desain UI/UX

Merancang tampilan halaman utama, *form* pengajuan, dan halaman status tiket menggunakan *Figma* dengan pembuatan yang *user-friendly* dan konsisten dengan tampilan sistem internal kampus.

Walaupun belum sampai tahap pengembangan kode dan implementasi, proyek ini memberikan dasar konseptual yang jelas untuk pengembangan sistem *ticketing* SKKM di masa depan.

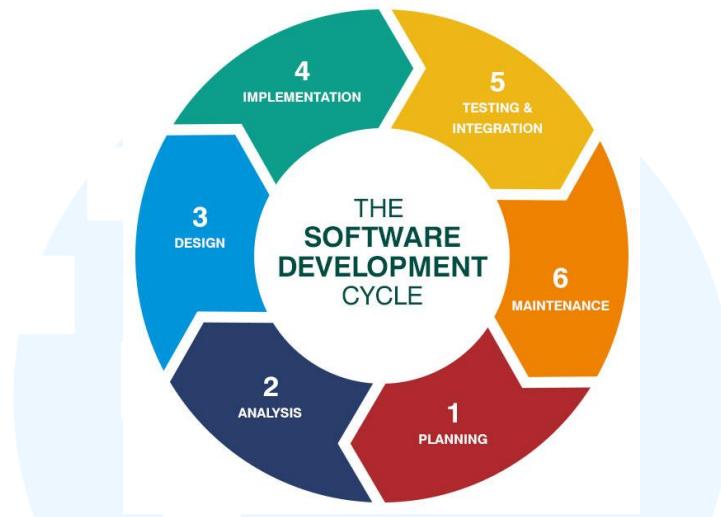
### 3.3.1 Proses Pelaksanaan

Selama pelaksanaan magang di Universitas Multimedia Nusantara (UMN), penulis berperan aktif sebagai *Web Developer* di bawah unit *Student Development*, bagian dari *Student Engagement*. Kegiatan magang ini dilakukan secara *Work From Office* (WFO) di kampus UMN, Tangerang, selama lima bulan, mulai 18 Agustus 2025 hingga 30 Desember 2025 dengan jam kerja Senin hingga Jumat pukul 08.00 -17.00 WIB.

Tujuan utama dari kegiatan magang ini adalah untuk mengimplementasikan keahlian dalam pengembangan sistem informasi berbasis web menggunakan PHP dan MySQL, serta mendukung proses digitalisasi layanan internal kampus. Selama periode magang, penulis terlibat dalam dua proyek utama, yaitu Pengembangan Sistem *Ticketing Internal* untuk Laporan Mahasiswa, dan Perancangan *Website Ticketing* untuk Pengajuan SKKM Mahasiswa.

Kedua proyek tersebut dirancang menggunakan metode *Software Development Life Cycle (SDLC)* yang meliputi tahapan *planning, analysis, design, implementation, testing & integration, deployment, dan maintenance*.

Setiap proyek melibatkan proses perancangan, pembuatan, dan dokumentasi sistem yang dilakukan secara bertahap berdasarkan kebutuhan unit kerja.



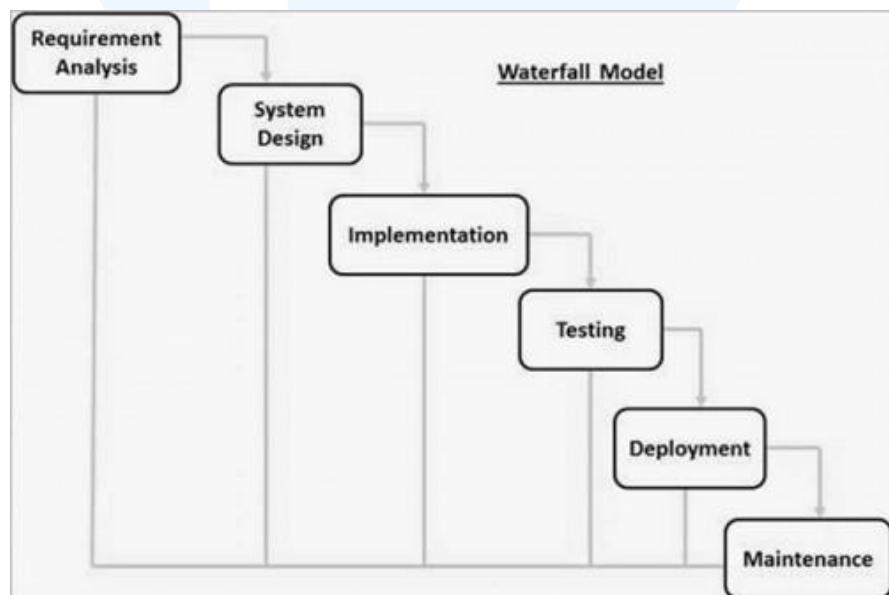
**Gambar 3.2** Diagram Software Development Life Cycle (SDLC)

Sumber: [course-net.com](http://course-net.com)

Metode *Software Development Life Cycle* (SDLC) digunakan sebagai kerangka kerja yang membantu proses pengembangan sistem agar lebih terarah dan sistematis. Dengan menggunakan SDLC, proses pembangunan sistem dapat dilakukan secara lebih terukur, mengurangi potensi kesalahan, serta memastikan hasil akhir sesuai dengan kebutuhan pengguna. Tujuan digunakan SDLC adalah menghasilkan sistem yang berkualitas, memenuhi kebutuhan unit kerja, dan tetap mudah dikelola di masa mendatang. Tahapannya dimulai dari *planning* yaitu proses merumuskan kebutuhan awal, menetapkan tujuan, serta menentukan ruang lingkup pengembangan. Setelah itu, masuk ke tahap *analysis*, di mana kebutuhan pengguna dianalisis lebih rinci dan alur kerja yang ada dipahami secara menyeluruh. Pada tahap *design*, disusun rancangan antarmuka, struktur *database*, serta arsitektur sistem berdasarkan hasil analisis sebelumnya. Selanjutnya, tahap *implementation* dilakukan untuk membangun sistem melalui pembuatan kode dan pengembangan fitur. Setelah sistem selesai dibuat, dilakukan *testing & integration* untuk memastikan setiap fungsi berjalan dengan baik dan bebas dari kendala. Ketika sistem dinyatakan siap, proses *deployment* dilakukan agar dapat digunakan oleh

pengguna di lingkungan kerja. Tahap terakhir adalah *maintenance* yang dilakukan untuk memperbaiki masalah, meningkatkan performa, dan menyesuaikan sistem apabila terdapat kebutuhan baru.

Selain SDLC, proses pengembangan sistem juga mengikuti pendekatan *Waterfall*, yaitu salah satu model SDLC yang bersifat berurutan. Pada metode ini, setiap tahapan harus diselesaikan sepenuhnya sebelum berlanjut ke tahap berikutnya, sehingga alurnya berjalan seperti air terjun dari atas ke bawah. Pendekatan *Waterfall* memberikan kejelasan alur kerja, dokumentasi yang lebih rapi, dan memudahkan proses pengendalian karena setiap tahap memiliki *output* yang jelas. Model ini cocok digunakan pada pengembangan sistem yang kebutuhannya stabil sejak awal, sehingga proses pengembangan dapat dilakukan dengan lebih terencana dan minim perubahan di tengah jalan.



Gambar 3.3 SDLC Model *Waterfall*

Sumber: [codepolitan.com](http://codepolitan.com)

Dalam model *Waterfall*, pengembangan sistem dimulai dari tahap *Requirement Analysis*, yaitu proses mengumpulkan dan memahami kebutuhan sistem secara mendalam. Semua kebutuhan yang disampaikan oleh pengguna dicatat dan disusun dalam dokumen spesifikasi sebagai acuan utama dalam pengembangan. Setelah

kebutuhan dirumuskan dengan jelas, tahap berikutnya adalah *System Design*, di mana informasi tersebut diterjemahkan menjadi rancangan sistem yang lebih teknis. Pada tahap ini ditentukan arsitektur sistem, kebutuhan perangkat keras, serta komponen teknis lain yang diperlukan agar proses pengembangan berjalan terarah.

Setelah desain disetujui, proses masuk ke tahap *Implementation*, yaitu pembuatan sistem dalam bentuk unit-unit kecil sesuai rancangan sebelumnya. Setiap unit dikembangkan dan diuji secara mandiri melalui unit testing untuk memastikan fungsinya berjalan dengan benar. Ketika seluruh unit selesai dibuat, tahap *Testing* dilakukan dengan menggabungkan seluruh unit menjadi satu sistem utuh. Pengujian menyeluruh kembali dilakukan untuk memastikan tidak ada kendala atau *error* setelah integrasi.

Jika sistem sudah dinilai stabil, tahap berikutnya adalah *Deployment*, yaitu proses menerapkan dan mengaktifkan sistem pada lingkungan pengguna agar dapat mulai digunakan dalam kegiatan operasional. Tahap terakhir adalah *Maintenance*, di mana sistem diawasi dan diperbaiki berdasarkan masukan pengguna. Pada tahap ini dilakukan pembaruan, penyempurnaan fitur, serta penanganan masalah agar sistem tetap bekerja dengan optimal dalam jangka panjang.

### **3.3.1.1 Pengembangan Sistem Ticketing Internal untuk Laporan Mahasiswa**

Sebelum sistem ini dikembangkan, proses pelaporan mahasiswa kepada unit *Student Engagement* dilakukan secara manual melalui email atau pesan pribadi. Hal ini mempersulit proses pelacakan, pendistribusian laporan ke sub divisi terkait serta dokumentasi tindak lanjut laporan. Maka, dikembangkanlah sistem ticketing internal berbasis web untuk mencatat dan memantau laporan mahasiswa secara terstruktur.

Berikut beberapa tahapan yang dilakukan selama pengembangan web dilakukan:

## ***A. Requirement Analysis***

Tahap ini dimulai dengan identifikasi permasalahan pada proses manual. Penulis berdiskusi dengan *Manager for Student Engagement* untuk menentukan ruang lingkup proyek, kebutuhan sistem, serta *timeline* pengembangan selama magang. Selama pengerjaan, penulis menggunakan beberapa *tools* utama seperti PHP, MySQL, XAMPP, dan *Visual Studio Code* sebagai perangkat utama untuk membangun sistem.

Selanjutnya, dilakukan melalui observasi langsung terhadap proses pelaporan di tiga sub divisi *Student Engagement* (*Student Service*, *Student Support*, dan *Student Development*) serta berdiskusi dengan *staff* yang terlibat dalam proses penerimaan laporan mahasiswa.

Analisis ini bertujuan untuk memahami alur kerja, permasalahan yang dihadapi dalam pencatatan laporan manual, serta menentukan kebutuhan sistem yang dibutuhkan agar proses pelaporan menjadi lebih efisien dan terdokumentasi dengan baik.

Berdasarkan hasil analisis, kebutuhan sistem dibagi menjadi dua kategori utama:

### **1. Kebutuhan Fungsional**

Sistem harus mampu menyediakan fungsi-fungsi utama sebagai berikut:

#### **a. Pembuatan tiket laporan**

Mahasiswa dapat membuat tiket berisi informasi NIM, jenis laporan, dan deskripsi permasalahan.

#### **b. Pembaruan status tiket**

Admin sub divisi dapat memperbarui status tiket menjadi *pending*, *assign*, *resolved*, atau *done* sesuai progres penyelesaian.

#### **c. Pencatatan tindak lanjut (notes)**

Admin dapat menambahkan catatan atau komentar pada setiap tiket sebagai dokumentasi proses penyelesaian laporan.

**d. Ekspor data laporan**

Admin dapat mengekspor data laporan ke dalam format Excel menggunakan library *ExcelJS* untuk kebutuhan dokumentasi dan pelaporan internal.

**e. Visualisasi laporan**

Sistem menampilkan grafik tren laporan menggunakan *Chart.js* untuk membantu analisis statistik per periode.

**f. Notifikasi pembaruan status**

Sistem menampilkan konfirmasi status atau pesan sukses saat perubahan data berhasil dilakukan.

## **2. Kebutuhan Non-Fungsional**

Selain fungsi utama, sistem juga harus memenuhi aspek pendukung sebagai berikut:

**a. Kemudahan penggunaan (*usability*)**

Antarmuka dirancang sederhana dan mudah dipahami oleh pengguna non-teknis seperti *staff* sub divisi.

**b. Keamanan data (*data security*)**

Setiap pengguna memiliki autentikasi login yang berbeda berdasarkan peran (role) seperti *superadmin*, *admin*, *student service*, *student support*, *student development*.

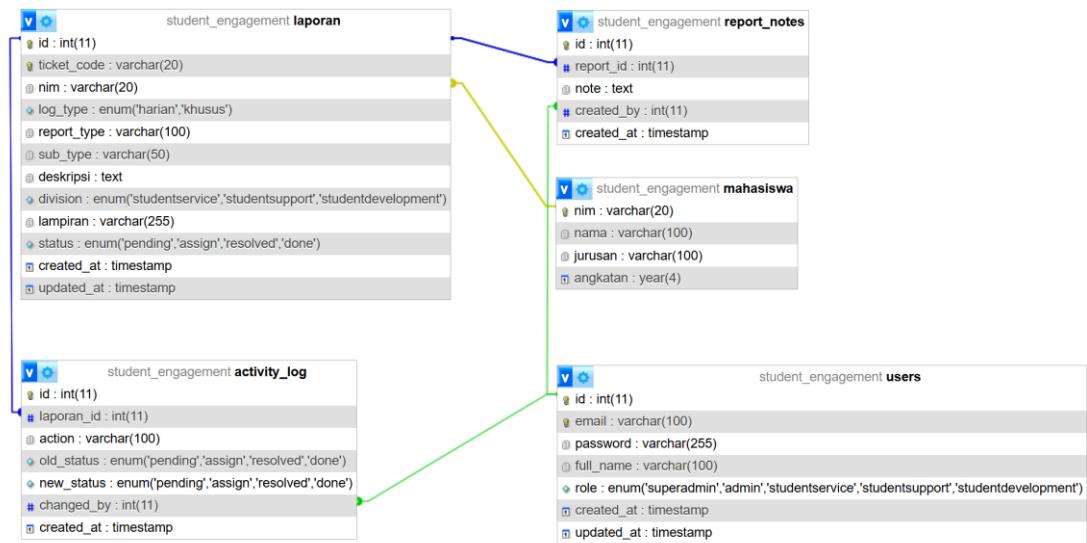
**c. Efisiensi akses (*performance*)**

Sistem dioptimalkan agar proses pemanggilan dan penyimpanan data di *database MySQL* berjalan cepat dan stabil.

#### d. Keterhubungan sistem (*integration*)

Data laporan, aktivitas, dan catatan tindak lanjut terintegrasi dalam satu *database* sehingga mudah untuk dilacak.

### B. System Design



Gambar 3.4 Struktur *Database Student Engagement*

Berdasarkan gambar 3.4 diatas, menunjukkan struktur *database* yang digunakan dalam sistem *ticketing* internal pada unit *Student Engagement* Universitas Multimedia Nusantara. *Database* ini terdiri dari lima tabel utama, yaitu: tabel *laporan*, *users*, *mahasiswa*, *report\_notes*, dan *activity\_log*. Masing-masing tabel memiliki fungsi dan hubungan yang saling terintegrasi untuk mendukung proses pelaporan, tindak lanjut, serta pengelolaan aktivitas sistem.

Perancangan database dilakukan menggunakan *Entity Relationship Diagram* (ERD) dengan tabel berikut:

## 1. Tabel laporan

Tabel ini merupakan inti dari sistem ticketing yang menyimpan semua data laporan mahasiswa.

Kolom-kolom yang digunakan antara lain:

1. **id**: kunci utama laporan.
2. **ticket\_code**: kode unik laporan/tiket.
3. **nim**: nomor induk mahasiswa pelapor (relasi ke tabel mahasiswa).
4. **log\_type**: tipe laporan (harian/khusus).
5. **report\_type**: kategori laporan.
6. **sub\_type**: subkategori laporan.
7. **division**: subdivisi tujuan laporan (*studentservice*, *studentsupport*, *studentdevelopment*).
8. **lampiran**: menyimpan nama file bukti laporan.
9. **status**: status tiket, terdiri dari *pending*, *assign*, *resolved*, dan *done*.
10. **created\_at** dan **updated\_at**: mencatat waktu pembuatan dan pembaruan laporan.

Tabel ini berperan sebagai pusat relasi yang terhubung ke tabel lain seperti *report\_notes* dan *activity\_log*.

## 2. Tabel users

Tabel users menyimpan data pengguna baik admin maupun sub divisi terkait.

Berikut kolom-kolom yang digunakan meliputi:

1. **id**: identitas pengguna (primary key).
2. **email** dan **password**: untuk autentikasi login.
3. **full\_name**: nama lengkap pengguna.

4. **role**: menentukan hak akses, terdiri dari *superadmin*, *admin*, *studentservice*, *studentsupport*, dan *studentdevelopment*.
5. **created\_at** dan **updated\_at**: mencatat waktu pembuatan dan pembaruan akun.

Dalam relasi antar tabel, setiap pengguna (users.id) dapat menjadi pembuat catatan tindak lanjut di tabel *report\_notes* dan melakukan perubahan status laporan di tabel *activity\_log*.

### 3. Tabel mahasiswa

Tabel ini menyimpan data identitas mahasiswa yang telah membuat laporan. Berikut kolom-kolom yang digunakan meliputi:

1. **nim**: kunci utama (primary key).
2. **nama**: nama mahasiswa.
3. **jurusan**: program studi mahasiswa.
4. **angkatan**: tahun angkatan mahasiswa.

Pada relasi antar tabel, tabel ini memiliki relasi *One-to-one* dengan tabel laporan melalui kolom nim.

### 4. Tabel report\_notes

Tabel ini digunakan untuk mencatat tindak lanjut atau komentar tambahan terhadap laporan tertentu.

Berikut kolom-kolom yang digunakan meliputi:

1. **id**: kunci utama (primary key).
2. **report\_id**: merujuk ke laporan yang ditindaklanjuti (laporan.id).
3. **note**: isi catatan atau komentar tindak lanjut.
4. **created\_by**: pengguna (admin) yang membuat catatan (relasi ke users.id).
5. **created\_at**: waktu pembuatan catatan.

Pada relasi antar tabel, dilakukan *Many-to-one* dengan laporan karena setiap laporan dapat memiliki banyak catatan tindak lanjut.

## 5. Tabel activity\_log

Tabel ini berfungsi sebagai log aktivitas sistem untuk mencatat semua perubahan status tiket oleh pengguna.

Berikut kolom-kolom yang digunakan meliputi:

1. **id**: kunci utama (primary key).
2. **laporan\_id**: identitas laporan yang diubah (relasi ke laporan.id).
3. **action**: jenis aksi yang dilakukan (contohnya “update status” atau “delete”).
4. **old\_status**: status laporan sebelum diubah.
5. **new\_status**: status laporan setelah diubah.
6. **changed\_by**: pengguna yang melakukan perubahan (relasi ke users.id).
  - a. **created\_at**: waktu aksi dilakukan.

Pada relasi antar tabel, dilakukan *Many-to-one* dengan tabel laporan karena setiap laporan dapat memiliki banyak aktivitas perubahan.

## 6. Entity Relationship Diagram (ERD) Deskriptif

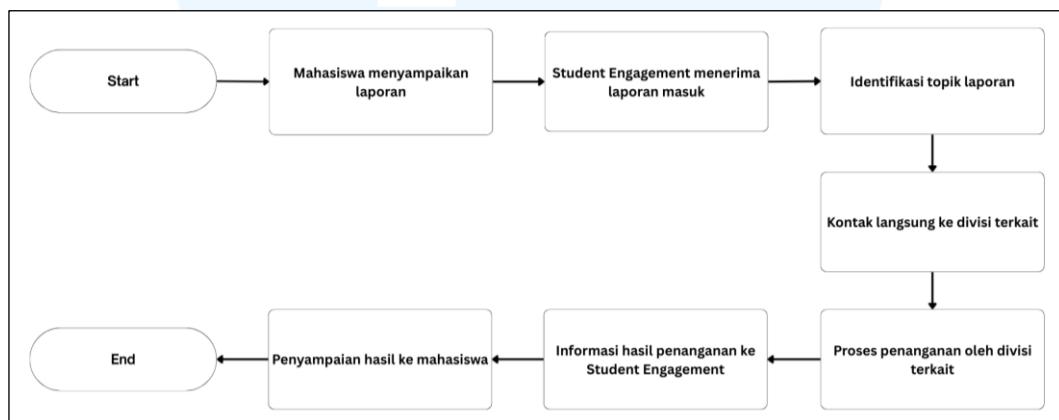
Berdasarkan struktur tabel sebelumnya, relasi antar tabel dalam *database* dapat dijelaskan sebagai berikut:

1. **Entity users**: memiliki relasi one-to-many dengan *report\_notes* dan *activity\_log*, di mana setiap pengguna dapat memberikan catatan tindak lanjut serta melakukan perubahan status pada banyak laporan.
2. **Entity mahasiswa**: memiliki relasi one-to-many dengan laporan, karena satu mahasiswa dapat membuat lebih dari satu laporan dengan NIM sebagai pengenal utama.

3. **Entity laporan:** sebagai entitas utama yang berelasi one-to-many dengan *report\_notes* dan *activity\_log*, serta many-to-one dengan mahasiswa sebagai pelapor.
4. **Entity report\_notes:** mencatat setiap catatan atau tindak lanjut terhadap laporan tertentu, serta mencatat siapa pengguna yang menambahkannya.
5. **Entity activity\_log:** mencatat setiap aktivitas sistem seperti perubahan status tiket, penghapusan laporan, dan tindakan lainnya, beserta pengguna yang melakukan aksi tersebut.

## 7. Alur Sistem (*Flowchart* Deskriptif)

### A. Sebelum menggunakan sistem (Manual)

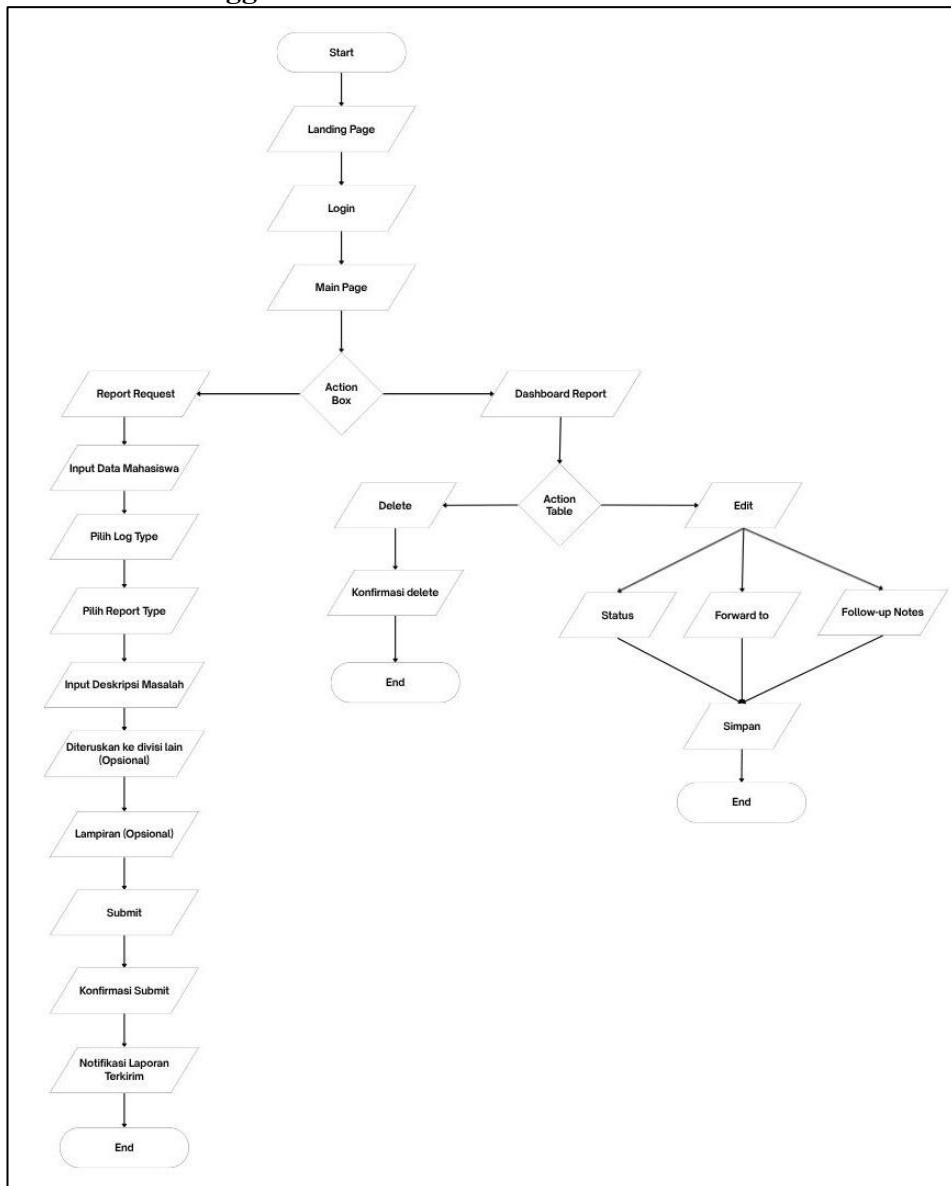


**Gambar 3.5** Alur Sebelum Menggunakan Sistem Ticketing Student Engagement

Pada gambar alur sistem tersebut, proses pelaporan mahasiswa dilakukan secara manual dan tidak terpusat. Ketika terdapat laporan dari mahasiswa, *staff Student Engagement* terlebih dahulu mengidentifikasi topik atau permasalahan yang dilaporkan. Setelah mengetahui divisi yang bertanggung jawab, *staff* kemudian menghubungi salah satu perwakilan divisi terkait secara langsung melalui telepon, pesan singkat, atau email. Proses tindak lanjut sepenuhnya bergantung pada komunikasi personal antar *staff* tanpa adanya sistem pencatatan terpusat. Setelah permasalahan ditangani oleh divisi terkait, informasi penyelesaian disampaikan kembali kepada *Student Engagement* yang kemudian meneruskan informasi tersebut kepada mahasiswa. Alur ini menyebabkan sulitnya pelacakan status

laporan, kurangnya dokumentasi historis, serta potensi terjadinya keterlambatan atau laporan yang terlewat.

## B. Sesudah menggunakan sistem



Gambar 3.6 Flowchart Tampilan Website Ticketing Student Engagement

Berdasarkan gambar tersebut, *flowchart* tersebut menggambarkan alur proses utama dalam sistem *ticketing* laporan mahasiswa pada unit *Student Engagement* Universitas Multimedia Nusantara. Sistem ini dirancang untuk mengelola laporan mahasiswa secara terstruktur dari proses *input* hingga tindak lanjut oleh admin sub divisi terkait.

Berikut penjelasan tahapan proses secara deskriptif:

1. Mahasiswa atau *staff* membuka halaman utama dan melakukan login menggunakan akun terdaftar sesuai sub divisi masing-masing (*Student Service, Student Support, atau Student Development*).
2. Setelah login, pengguna diarahkan ke Main Page yang menampilkan dua menu utama:
  - a. Report Request digunakan untuk membuat laporan baru.
  - b. Dashboard Report digunakan untuk melihat laporan yang sudah ada.
3. Pada menu Report Request, pengguna mengisi *form* pelaporan yang terdiri dari:
  - a. Input data mahasiswa (NIM atau data mahasiswa otomatis terambil dari tabel *mahasiswa*).
  - b. Pemilihan *log type* (harian atau khusus).
  - c. Pemilihan *report type* sesuai kategori laporan.
  - d. Deskripsi masalah yang ingin dilaporkan.
  - e. Menentukan sub divisi lain untuk diteruskan dan melampirkan file pendukung (opsional).
4. Setelah *form* diisi, sistem akan melakukan validasi data dan menampilkan konfirmasi submit. Jika berhasil, sistem akan melakukan hal-hal berikut:
  - a. Menyimpan data ke tabel *laporan*.
  - b. Mencatat aktivitas ke tabel *activity\_log*.
  - c. Mengirim notifikasi bahwa laporan berhasil diterima.

5. Pengguna atau admin kemudian dapat mengakses *Dashboard Report* untuk memantau laporan. *Dashboard* ini menampilkan daftar tiket laporan dengan berbagai status (*pending*, *assign*, *resolved*, *done*), serta menyediakan fitur berupa *Edit* laporan untuk memperbarui status atau menambahkan *follow-up notes* dan *Delete* laporan jika laporan tidak valid (menampilkan konfirmasi penghapusan).
6. Saat admin memperbarui laporan, sistem secara otomatis:
  - a. Memperbarui status di tabel *laporan*.
  - b. Mencatat aktivitas perubahan di tabel *activity\_log*.
  - c. Menyimpan catatan tindak lanjut di tabel *report\_notes*.
7. Data laporan dapat diekspor menjadi file excel (.xlsx) menggunakan library ExcelJS dan FileSaver.js untuk kebutuhan rekap dan pelaporan internal.

Penerapan sistem *ticketing* memberikan peningkatan signifikan dibandingkan proses manual sebelumnya. Sistem ini mampu mengurangi ketergantungan pada komunikasi personal, memperjelas alur distribusi laporan, serta menyediakan riwayat penanganan yang terdokumentasi dengan baik. Dengan adanya sistem terpusat, proses pelaporan menjadi lebih terkontrol, mudah dipantau, dan mendukung koordinasi yang lebih efektif antar sub divisi di lingkungan *Student Engagement*.

## 8. Perancangan UI/UX Design

Tampilan halaman UI/UX dirancang menggunakan *Figma* yang meliputi:

### 1. Halaman Awal

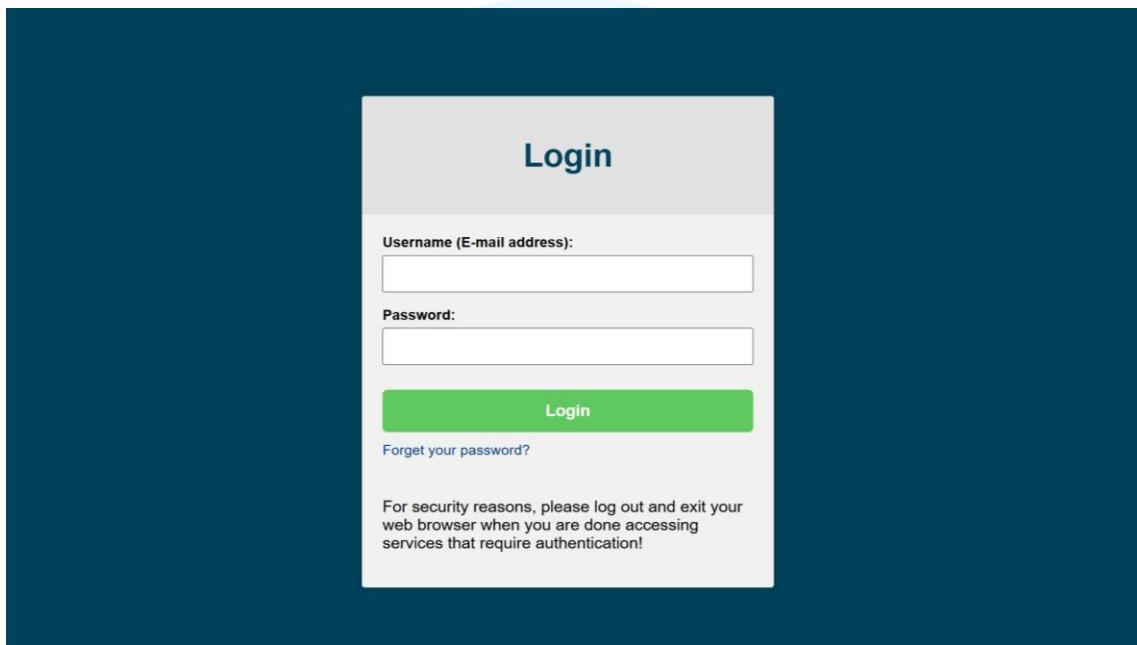


Gambar 3.7 Halaman awal website *Student Engagement*

Pada gambar 3.7 diatas, halaman awal merupakan tampilan pertama yang diakses pengguna sebelum masuk ke dalam sistem. Pada halaman ini, ditampilkan logo resmi *Student Engagement* UMN di bagian kiri atas yang berfungsi sebagai identitas visual sistem. Di tengah halaman terdapat judul “Student Engagement Report” disertai subjudul “Universitas Multimedia Nusantara” sebagai penanda bahwa sistem ini merupakan platform internal kampus. Tampilan ini dirancang dengan gaya minimalis untuk menjaga fokus pengguna pada fungsi utama halaman, yaitu tombol “Login” yang terletak di pojok kanan atas. Selain itu, *sidebar* tampil dalam bentuk ikon sederhana yang belum aktif digunakan sebelum pengguna melakukan login. Keseluruhan desain

dibuat ringkas, dan mudah dipahami untuk menciptakan pengalaman yang nyaman bagi pengguna baru.

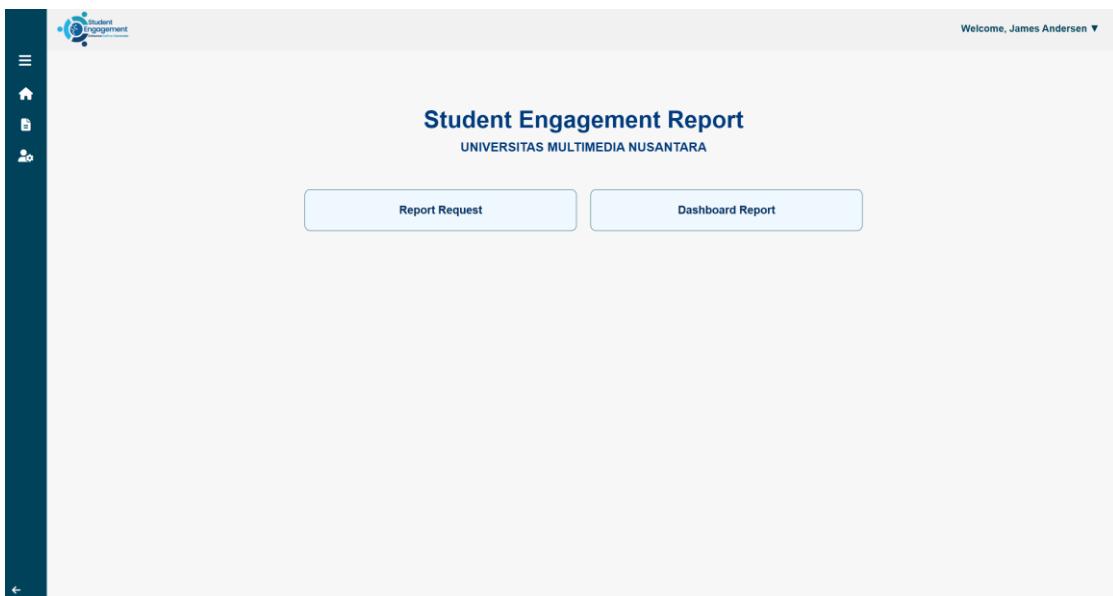
## 2. Halaman Login



**Gambar 3.8** Halaman login website *Student Engagement*

Pada gambar 3.8 menampilkan halaman login yang merupakan gerbang autentikasi yang menghubungkan pengguna dengan sistem. Pada halaman ini, ditampilkan formulir login yang terdiri dari dua komponen penting, yaitu *input email* dan *password*. Tampilan dibuat sederhana dengan latar gelap dan kartu login berwarna terang untuk memberikan kontras yang tinggi sehingga pengguna dapat langsung fokus pada formulir. Tombol "Login" ditampilkan dengan warna hijau yang menonjol sebagai tombol utama. Selain itu, tersedia tautan "*Forget your password?*" yang memberikan pengalaman pengguna yang lebih baik melalui opsi pemulihan akun. Pada bagian bawah terdapat informasi keamanan yang mengingatkan pengguna untuk keluar dari sistem setelah selesai digunakan. Halaman ini dirancang agar mudah dipahami oleh seluruh pengguna termasuk *staff* yang mungkin tidak terbiasa dengan platform digital, sehingga proses masuk ke sistem dapat dilakukan tanpa hambatan.

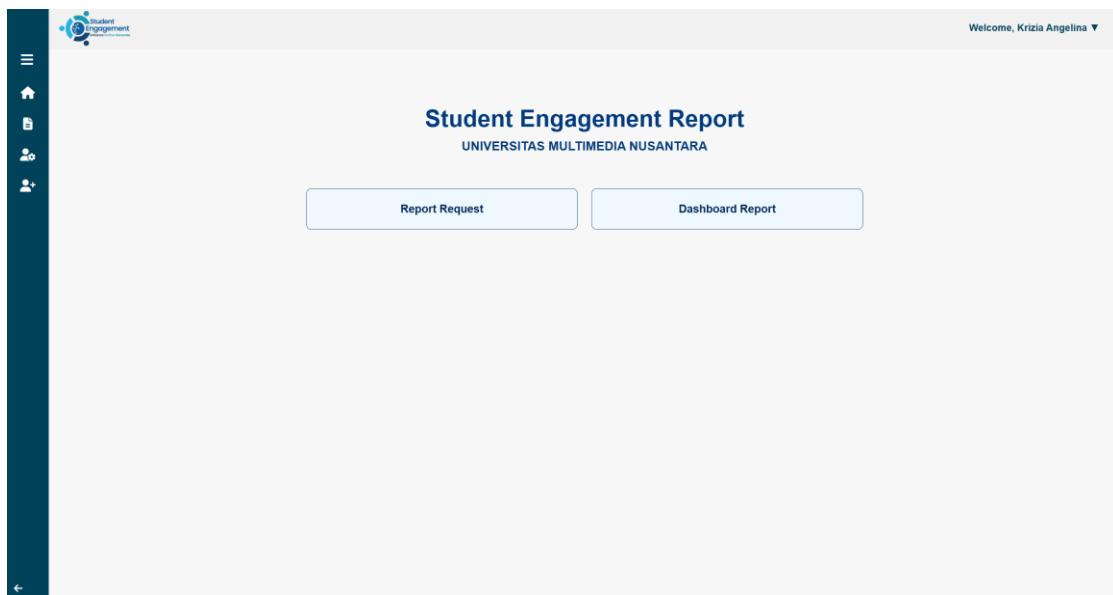
### 3. Halaman Utama



**Gambar 3.9** Halaman utama website *Student Engagement*

Pada gambar 3.9 diatas, jika berhasil login, maka pengguna akan diarahkan ke halaman menu utama yang berfungsi sebagai pusat navigasi awal. Pada halaman ini, pengguna disambut dengan teks “Welcome, (Nama User)” yang ditampilkan di bagian kanan atas untuk memberikan kesan personal. Di bagian tengah halaman terdapat dua tombol besar yaitu “Report Request” dan “Dashboard Report” yang menjadi akses utama terhadap fungsi sistem. Desain halaman ini sengaja dibuat sederhana dengan fokus pada dua pilihan tersebut untuk memastikan pengguna dapat langsung melakukan tugas yang diinginkan tanpa kebingungan. *Sidebar* di sisi kiri juga mulai aktif dan menampilkan ikon navigasi seperti beranda, pengajuan laporan, dan dashboard laporan (sesuai role pengguna). Dengan tampilan yang ringkas dan berpusat pada fungsi utama, halaman ini membantu mengarahkan pengguna secara cepat dan efektif menuju fitur yang diperlukan.

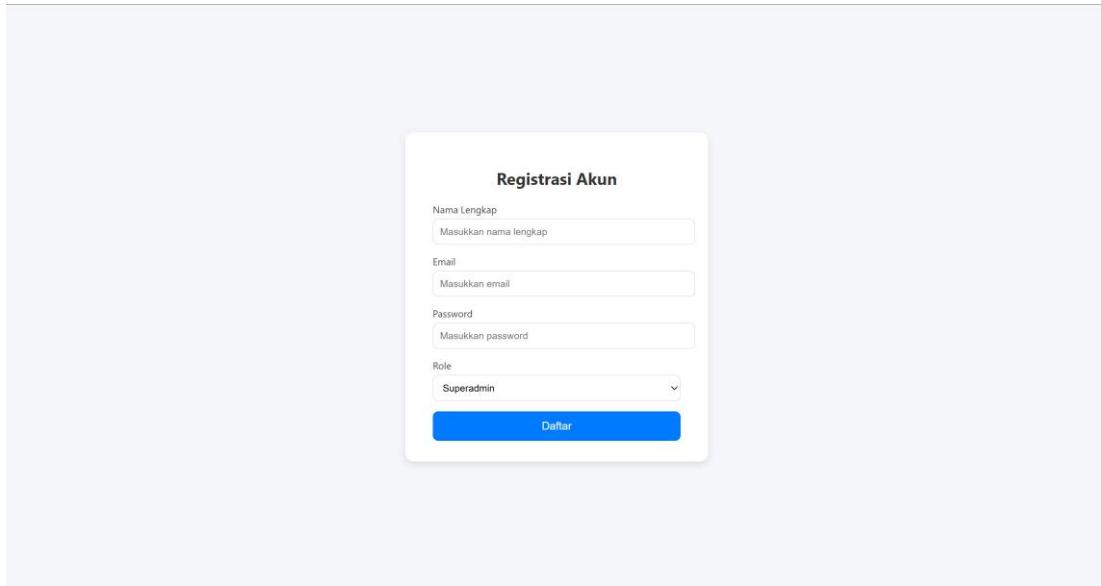
#### 4. Halaman Utama (superadmin)



**Gambar 3.10** Halaman utama website *Student Engagement* (superadmin)

Pada gambar 3.10 diatas, halaman tersebut merupakan tampilan utama yang muncul setelah pengguna berhasil melakukan proses login ke dalam sistem *Student Engagement Report*. Pada bagian tengah halaman, pengguna disambut dengan dua tombol utama, yaitu “Report Request” untuk membuat tiket laporan baru, serta “Dashboard Report” yang digunakan untuk melihat ringkasan data laporan dalam bentuk grafik dan statistik. Pada halaman untuk pengguna “superadmin”, terdapat ikon tambahan di bagian paling bawah sidebar, yaitu ikon berbentuk user-plus (register akun). Ikon ini merupakan fitur yang hanya dapat diakses oleh role Superadmin yang memiliki kewenangan khusus untuk melakukan pendaftaran akun baru ke dalam sistem. Fitur ini sengaja dipisahkan dari menu utama untuk menjaga keamanan dan memastikan bahwa proses pembuatan akun hanya dilakukan oleh pihak yang memiliki otoritas penuh. Secara keseluruhan, halaman ini berfungsi sebagai pusat navigasi awal sekaligus memberikan akses cepat ke seluruh modul sistem sesuai hak akses pengguna masing-masing.

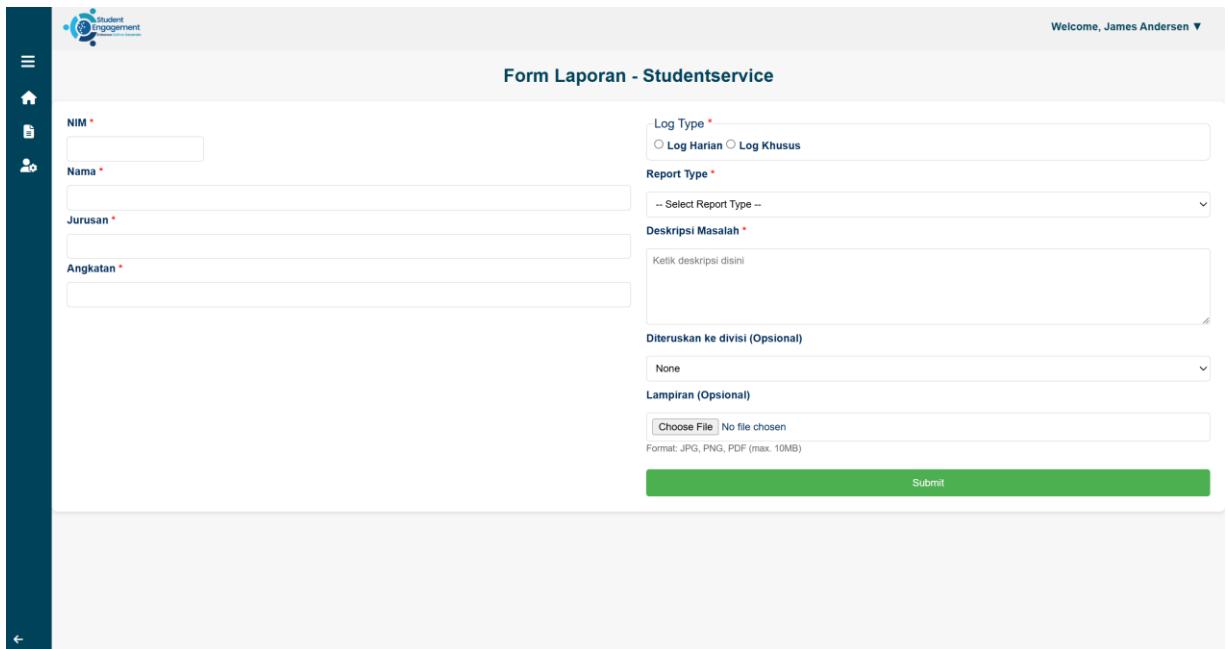
## 5. Halaman Registrasi Akun (superadmin)



**Gambar 3.11** Halaman registrasi akun (superadmin)

Pada gambar 3.11 diatas, halaman yang ditampilkan merupakan fitur khusus yang hanya dapat diakses oleh pengguna dengan peran Superadmin. Halaman ini berfungsi sebagai pusat pendaftaran akun baru ke dalam sistem *Student Engagement Report*. Pada bagian tengah halaman, terdapat sebuah *card* berisi *form* registrasi yang mencakup beberapa kolom penting seperti Nama Lengkap, Email, dan Password sebagai data identitas dasar pengguna. Selain itu, terdapat *dropdown role* yang memungkinkan Superadmin menentukan peran akun yang didaftarkan, seperti *Student Service*, *Student Support*, *Student Development* dan *Admin* sesuai kebutuhan operasional sistem. Setelah seluruh data diisi, pengguna dapat menekan tombol “Daftar” untuk menyimpan akun ke dalam *database*. Kehadiran halaman ini mendukung pengelolaan akses sistem secara terpusat, menjaga keamanan, dan memastikan bahwa hanya pihak berwenang yang dapat menambahkan pengguna baru. Hal ini menjadi bagian penting dalam manajemen *user access* dan operasional sistem internal, terutama karena struktur *Student Engagement* melibatkan beberapa sub divisi dengan tanggung jawab yang berbeda.

## 6. Formulir tiket laporan mahasiswa



Form Laporan - Studentservice

NIM \*

Nama \*

Jurusan \*

Angkatan \*

Log Type \*  Log Harian  Log Khusus

Report Type \*

Deskripsi Masalah \*

Klik deskripsi disini

Diteruskan ke divisi (Opsiional)

None

Lampiran (Opsiional)

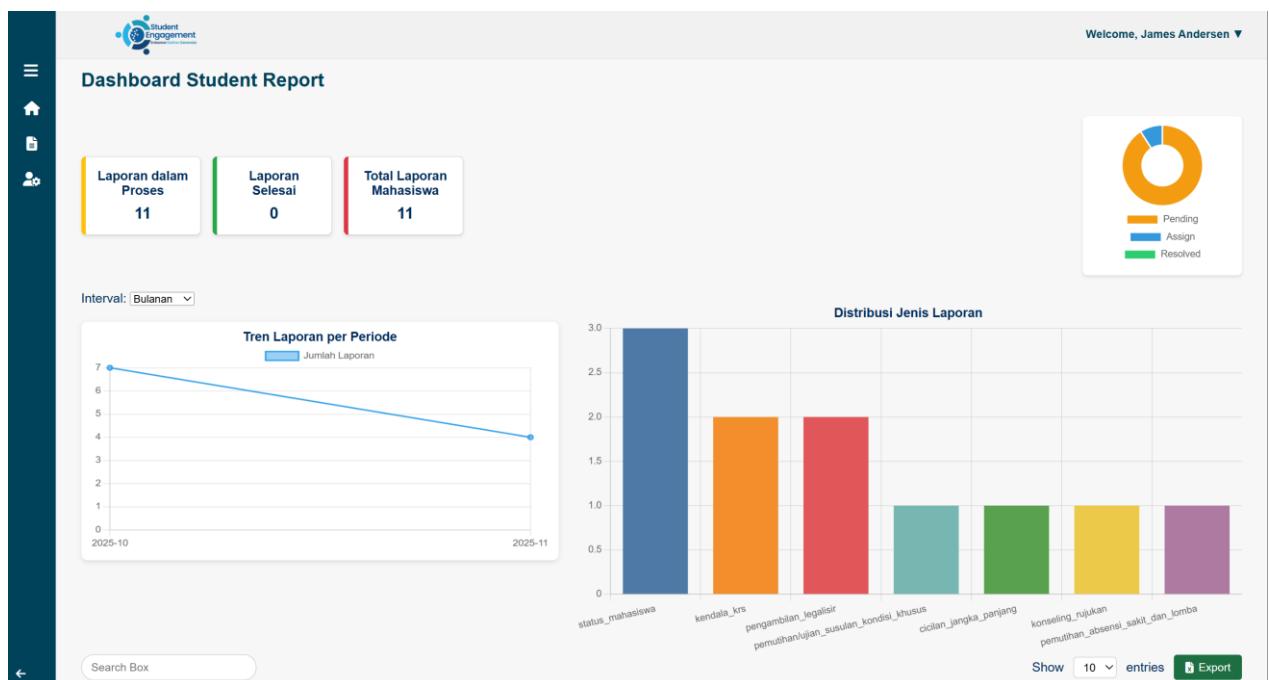
Choose File No file chosen Format: JPG, PNG, PDF (max. 10MB)

Submit

**Gambar 3.12** Halaman formulir pengajuan laporan website Student Engagement

Pada gambar 3.12, halaman formulir laporan adalah halaman yang paling sering digunakan dalam sistem, terutama oleh *staff* Student Engagement yang melakukan *input* laporan mahasiswa setiap hari. Halaman ini menampilkan dua bagian utama yang tersusun secara horizontal untuk mempermudah proses *input*. Pada bagian kiri halaman ditampilkan data mahasiswa seperti NIM, nama, jurusan, dan angkatan, yang dapat terisi otomatis melalui fitur pencarian NIM, sehingga mengurangi kesalahan input dan mempercepat proses. Sementara itu, bagian kanan menampilkan informasi laporan yang harus diisi *staff* seperti log type, report type, deskripsi masalah, tujuan divisi lain (opsional), serta unggahan lampiran jika diperlukan. Seluruh *input* yang wajib diisi ditandai dengan ikon bintang merah agar pengguna lebih mudah membedakan antara data yang wajib diisi atau opsional. Tombol *submit* ditempatkan secara jelas pada bagian bawah sebagai langkah akhir pengajuan laporan. Desain halaman ini dibuat *user-friendly* dengan penataan komponen yang rapi dan terstruktur sehingga proses pengisian dapat dilakukan secara efisien dan tanpa hambatan.

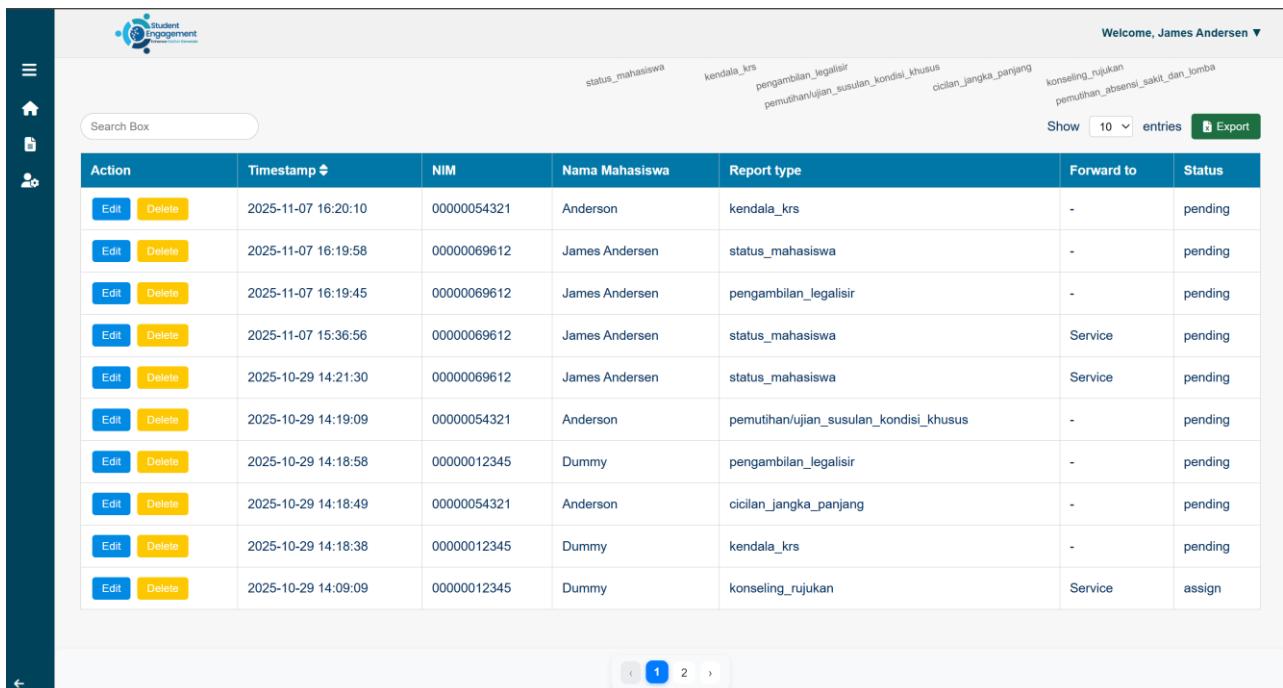
## 7. Dashboard laporan divisi



Gambar 3.13 Halaman dashboard laporan website Student Engagement

Pada gambar 3.13 diatas, dashboard student report tersebut menampilkan rangkuman visual dari seluruh laporan yang masuk ke sistem. Pada halaman ini, pengguna dapat melihat informasi secara menyeluruh melalui tampilan statistik yang disajikan dalam bentuk kartu ringkas seperti jumlah laporan yang sedang diproses, laporan yang sudah selesai, serta total laporan mahasiswa. Halaman ini dilengkapi dengan *chart* interaktif yang dihasilkan menggunakan Chart.js, termasuk grafik garis untuk menampilkan tren laporan dari waktu ke waktu dan grafik batang untuk melihat distribusi laporan berdasarkan jenisnya. Selain itu, terdapat diagram *pie* yang menampilkan proporsi status tiket seperti pending, assign, dan resolved. Pengguna juga dapat memilih interval data (harian, mingguan, bulanan) untuk menyesuaikan tampilan grafik dengan kebutuhan *monitoring*. Desain dashboard dibuat responsif, informatif, dan

mudah dipahami agar pengguna dapat melakukan pemantauan kondisi laporan secara real-time dengan cepat dan akurat.



Action	Timestamp	NIM	Nama Mahasiswa	Report type	Forward to	Status
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-07 16:20:10	00000054321	Anderson	kendala_krs	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-07 16:19:58	00000069612	James Andersen	status_mahasiswa	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-07 16:19:45	00000069612	James Andersen	pengambilan_legalisir	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-07 15:36:56	00000069612	James Andersen	status_mahasiswa	Service	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 14:21:30	00000069612	James Andersen	status_mahasiswa	Service	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 14:19:09	00000054321	Anderson	permintaan/ujian_susulan_kondisi_khusus	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 14:18:58	00000012345	Dummy	pengambilan_legalisir	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 14:18:49	00000054321	Anderson	cicilan_jangka_panjang	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 14:18:38	00000012345	Dummy	kendala_krs	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 14:09:09	00000012345	Dummy	konseling_reffukan	Service	assign

**Gambar 3.14** Halaman *dashboard* tabel laporan website *Student Engagement*

Pada gambar 3.14 diatas, halaman tabel utama laporan merupakan bagian yang digunakan untuk menampilkan seluruh data laporan mahasiswa dalam bentuk tabel yang rapi dan mudah dipahami. Pada halaman ini, setiap laporan ditampilkan dalam beberapa kolom seperti *Action*, *Timestamp*, NIM, Nama Mahasiswa, *Report Type*, *Forward To*, dan Status. Di bagian *Action*, terdapat tombol *Edit* dan *Delete* yang memungkinkan *staff* untuk langsung mengubah ataupun menghapus laporan tanpa harus berpindah halaman.

Informasi waktu pada kolom *Timestamp* digunakan untuk menunjukkan kapan laporan dibuat, sehingga *staff* dapat dengan cepat melacak laporan berdasarkan urutan pengirimannya. Untuk memudahkan pencarian, halaman ini dilengkapi *Search Box* di bagian atas yang memungkinkan pengguna mencari laporan berdasarkan nama, jenis laporan, atau NIM. Selain itu, pengguna juga bisa mengatur jumlah data yang ingin ditampilkan melalui menu “*Show entries*” dan berpindah halaman menggunakan navigasi *pagination* di bagian bawah tabel.

Di sisi kanan atas tabel terdapat tombol *Export* yang berfungsi untuk mengunduh seluruh data laporan dalam format Excel. Fitur ini memanfaatkan integrasi *library* seperti ExcelJS dan FileSaver.js, sehingga *staff* dapat dengan mudah menyimpan arsip laporan atau menyusun dokumen administrasi lainnya berdasarkan data yang ada.

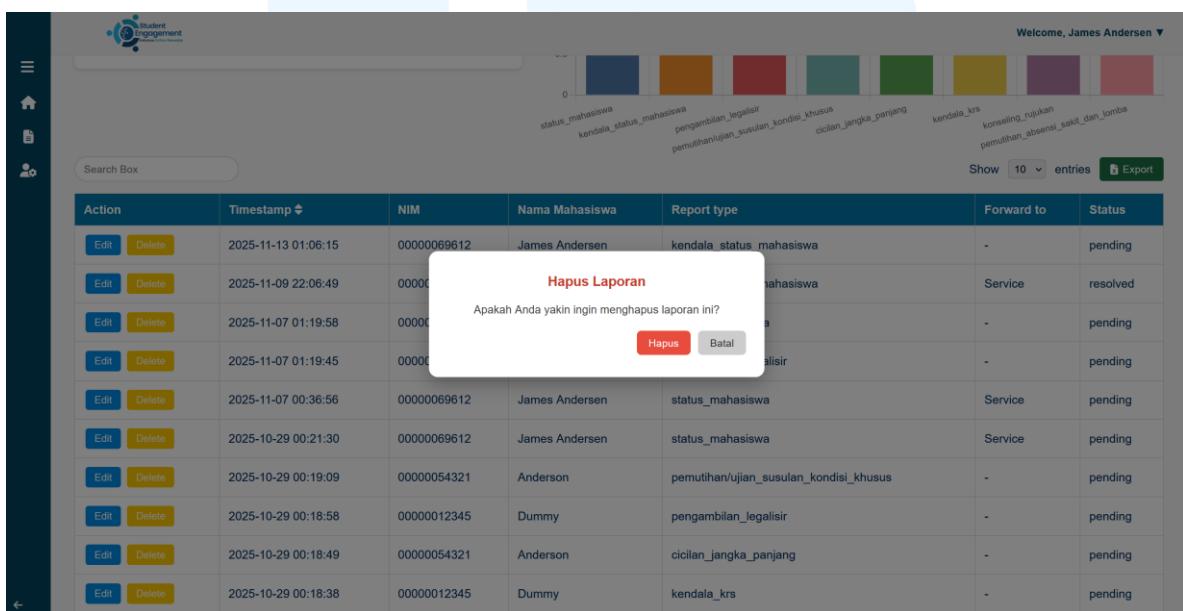
Sistem juga menyesuaikan tampilan tabel berdasarkan peran pengguna. Misalnya, pengguna dari *Student Service* hanya melihat laporan yang masuk ke divisinya, begitu pula *Student Support* dan *Student Development*. Sementara itu, admin (*part-timer*) memiliki akses untuk melihat seluruh laporan dari semua divisi, meskipun tetap dalam batasan kewenangan yang diberikan.

Action	Timestamp	NIM	Status	Forward to	Status
Edit Delete	2025-11-13 01:06:15	0000000000000000	Pending	Mahasiswa	- pending
Edit Delete	2025-11-09 22:06:49	0000000000000000	Pending	Mahasiswa	Service resolved
Edit Delete	2025-11-07 01:19:58	0000000000000000	Pending	Mahasiswa	- pending
Edit Delete	2025-11-07 01:19:45	0000000000000000	Pending	Mahasiswa	- pending
Edit Delete	2025-11-07 00:36:56	0000000000000000	Pending	Mahasiswa	Service pending
Edit Delete	2025-10-29 00:21:30	0000000000000000	Pending	Mahasiswa	Service pending
Edit Delete	2025-10-29 00:19:09	00000054321	Anderson	pemutihan_ujian_susulan_kondisi_khusus	- pending
Edit Delete	2025-10-29 00:18:58	00000012345	Dummy	pengambilan_legalisir	- pending
Edit Delete	2025-10-29 00:18:49	00000054321	Anderson	cicilan_jangka_panjang	- pending
Edit Delete	2025-10-29 00:18:38	00000012345	Dummy	kendala_krs	- pending

Gambar 3.15 Tombol fitur *edit* laporan

Pada gambar 3.15 diatas, fitur *edit* laporan merupakan sistem untuk menampilkan sebuah jendela *pop-up* ketika pengguna memilih tombol *Edit* pada salah satu baris laporan. Melalui *pop-up* ini, *staff* dapat memperbarui informasi laporan tanpa harus berpindah ke halaman lain, sehingga proses pengelolaan menjadi lebih cepat. Beberapa elemen yang tersedia di dalamnya mencakup pilihan status laporan, *Forward to* untuk meneruskan laporan ke divisi lain, serta kolom catatan tambahan yang bisa diisi sesuai kebutuhan. Status laporan ditampilkan dalam bentuk *dropdown* yang berisi tahapan

pemrosesan, seperti *pending*, *assign*, *resolved*, hingga *done*. Begitu juga untuk *Forward to*, sistem menyediakan daftar sub divisi terkait seperti studentservice atau studentsupport. Sementara itu, kolom catatan berfungsi sebagai tempat memberikan keterangan tindak lanjut agar proses penanganan dapat dilacak dengan jelas. Di bagian bawah modal, tersedia tombol *Batal* jika pengguna ingin membatalkan perubahan dan tombol *Simpan* untuk memperbarui data. Desain fitur ini dibuat ringkas agar *staff* bisa melakukan *update* secara praktis dan nyaman.



The screenshot shows a web-based application interface. At the top, there is a navigation bar with a logo, a search box, and a user welcome message: "Welcome, James Andersen". Below the navigation bar is a horizontal bar with colored squares and labels: "status\_mahasiswa" (blue), "kendala\_status\_mahasiswa" (orange), "pengambilan\_legalisir" (red), "permuthian\_ujian\_susulan\_kondisi\_khusus" (green), "cicilan\_jangka\_panjang" (light green), "kendala\_krs" (yellow), "konseling\_nujukan" (purple), and "permuthian\_abserei\_sakit\_dan\_lomba" (pink). A progress bar is visible above the main content area. The main content is a table with the following columns: Action, Timestamp, NIM, Name, Report type, Forward to, and Status. The table lists several entries, each with an "Edit" and "Delete" button. A modal dialog box is overlaid on the table, titled "Hapus Laporan". It contains the question "Apakah Anda yakin ingin menghapus laporan ini?". It has two buttons: "Hapus" (red) and "Batal" (gray). The table data is as follows:

Action	Timestamp	NIM	Nama Mahasiswa	Report type	Forward to	Status
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-13 01:06:15	00000069612	James Andersen	kendala_status_mahasiswa	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-09 22:06:49	00000069612	James Andersen	kendala_status_mahasiswa	Service	resolved
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-07 01:19:58	00000069612	James Andersen	kendala_status_mahasiswa	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-07 01:19:45	00000069612	James Andersen	kendala_status_mahasiswa	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-11-07 00:36:56	00000069612	James Andersen	status_mahasiswa	Service	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 00:21:30	00000069612	James Andersen	status_mahasiswa	Service	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 00:19:09	00000054321	Anderson	permuthian_ujian_susulan_kondisi_khusus	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 00:18:58	00000012345	Dummy	pengambilan_legalisir	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 00:18:49	00000054321	Anderson	cicilan_jangka_panjang	-	pending
<a href="#">Edit</a> <a href="#">Delete</a>	2025-10-29 00:18:38	00000012345	Dummy	kendala_krs	-	pending

Gambar 3.16 Tombol fitur *delete* laporan

Pada gambar 3.16 diatas, fitur *delete* laporan digunakan untuk menghapus laporan yang sudah tidak diperlukan lagi, misalnya laporan yang salah input atau duplikat. Saat pengguna menekan tombol *Delete*, sistem akan menampilkan jendela konfirmasi agar tidak terjadi kesalahan penghapusan. Pesan yang muncul akan menanyakan apakah pengguna benar-benar ingin menghapus data tersebut. Di bawah pesan tersebut, sistem menyediakan tombol *Hapus* yang diberi penanda warna merah sebagai peringatan bahwa tindakan ini bersifat permanen, serta tombol *Batal* untuk kembali ke halaman sebelumnya. Jika pengguna memilih untuk melanjutkan penghapusan, sistem tidak hanya menghapus data laporan utama, tetapi juga seluruh catatan terkait di tabel

*report\_notes* dan *activity\_log*. Dengan begitu, tidak ada data tersisa yang menggantung dan integritas database tetap terjaga secara menyeluruh.

Gambar 3.17 Tombol fitur *detail laporan*

Pada gambar 3.17 diatas, fitur *detail laporan* memungkinkan *staff* melihat informasi lengkap mengenai suatu laporan hanya dengan meng-klik baris data pada tabel utama. Setelah baris dipilih, sebuah *pop-up* akan muncul berisi detail laporan yang tersusun dengan rapi. Informasi yang ditampilkan mencakup waktu laporan dibuat, NIM, nama mahasiswa, jenis laporan, divisi tujuan(*Forward to*), status terbaru, serta deskripsi masalah yang dilaporkan. Selain itu, sistem juga menampilkan riwayat catatan tindak lanjut yang pernah ditambahkan oleh *staff* atau admin, lengkap dengan waktu pencatatan dan isi pesannya. Dengan fitur ini, dapat membantu *staff* memahami informasi laporan secara utuh tanpa harus membuka halaman lain, sehingga pemantauan laporan dapat dilakukan lebih cepat dan efisien. *Pop-up* ini juga dilengkapi tombol *Tutup* untuk kembali ke tampilan tabel seperti semula. Desainnya yang rapi dan terstruktur membuat proses pengecekan laporan menjadi jauh lebih mudah dan nyaman.

Secara keseluruhan, halaman ini dirancang agar tetap nyaman digunakan meskipun menampilkan data dalam jumlah besar. Penempatan kolom yang

konsisten, tombol aksi yang jelas, serta tampilan status yang mudah dibedakan membantu pengguna bekerja lebih cepat dan akurat. Hal ini menjadikan halaman tabel laporan sebagai salah satu komponen inti dalam pemantauan dan pengelolaan laporan mahasiswa di lingkungan *Student Engagement*.

### ***C. Implementation***

Pada tahap implementasi, sistem dikembangkan menggunakan HTML, CSS, PHP dan JavaScript untuk menghasilkan antarmuka dan fungsi yang responsif. Pengelolaan dan koneksi *database* dilakukan melalui PDO (PHP *Data Objects*), sehingga aplikasi PHP berinteraksi dengan berbagai jenis basis data menggunakan struktur API (Antarmuka Pemrograman Aplikasi) yang seragam. Pendekatan ini tidak hanya memberikan fleksibilitas, tetapi juga meningkatkan efisiensi terutama pada eksekusi *query* yang dilakukan secara berulang. Pada tahap ini, sejumlah fitur inti berhasil diimplementasikan, antara lain:

- a. Login & autentikasi pengguna berdasarkan sub divisi.
- b. Formulir pengajuan laporan dengan validasi *input*.
- c. *Dashboard* laporan dengan fitur filter, pencarian, dan *export* data.
- d. Statistik laporan menggunakan Chart.js.
- e. *Export* data ke Excel menggunakan ExcelJS dan FileSaver.js.
- f. Tampilan antarmuka dengan *Font Awesome* untuk ikon interaktif.

Semua *file* dikembangkan dan diatur secara terstruktur seperti db.php, get\_notes.php, get\_chart\_data.php, dan delete\_report.php.

### ***D. Testing***

Pada tahap ini dilakukan proses pengujian untuk memastikan seluruh fitur yang dikembangkan dapat berjalan dengan baik dan sesuai dengan kebutuhan unit *Student Engagement*. Pengujian dimulai dengan memahami kembali fungsionalitas utama sistem, seperti pembuatan tiket laporan, pembaruan status, serta pelacakan informasi pada setiap sub divisi. Setelah itu, dilakukan

serangkaian uji coba dengan memasukkan berbagai jenis data melalui formulir, termasuk data mahasiswa, kategori laporan, serta deskripsi masalah yang berbeda-beda.

Dari setiap percobaan, penulis memeriksa apakah data benar-benar tersimpan di *database* dan apakah perubahan status tiket langsung terlihat pada *dashboard*. Selain itu, beberapa skenario juga disiapkan untuk menguji berbagai kondisi penggunaan, seperti *input* data yang lengkap, data yang belum valid, atau mencoba fitur pencarian dan penyaringan laporan.

Beberapa hal yang diuji meliputi pembuatan tiket laporan baru, perubahan status laporan menjadi *assign* ataupun *resolved*, serta penghapusan tiket melalui halaman *dashboard*. Setiap tindakan kemudian diamati hasilnya untuk memastikan sistem memberikan *respond* yang tepat dan tidak terjadi kegagalan fungsi.

Dari keseluruhan proses pengujian, seluruh fitur utama terbukti dapat berjalan dengan lancar tanpa adanya *error*. Selain itu, integrasi antara antarmuka pengguna dengan *database* MySQL juga berlangsung dengan stabil, sehingga sistem sudah siap digunakan sebagai alat pendukung proses pelaporan mahasiswa di lingkungan Student Engagement UMN.

#### ***E. Deployment***

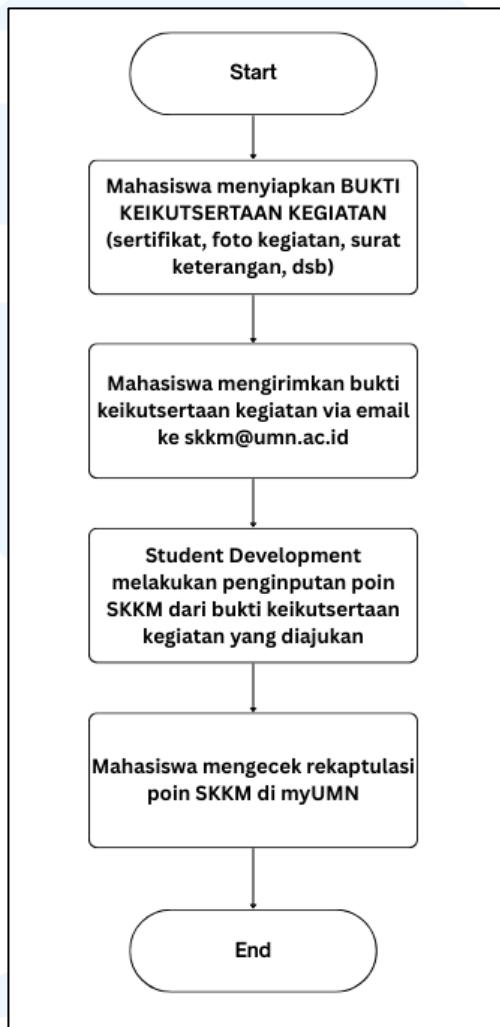
Sistem yang telah diuji kemudian akan *deploy* menggunakan layanan *hosting InfinityFree* dengan alamat berikut: <https://student-engagement.infinityfree.me/>

#### ***F. Maintenance***

Selama melakukan uji coba program, penulis melakukan pemeliharaan terhadap tampilan, optimasi *query*, serta penyesuaian label kategori berdasarkan masukan pengguna. Selain itu, disusun dokumentasi teknis sistem yang mencakup struktur *database*, panduan penggunaan, dan rencana pengembangan selanjutnya.

### 3.3.1.2 Perancangan Website Ticketing untuk Pengajuan SKKM Mahasiswa

#### A. Sebelum menggunakan sistem

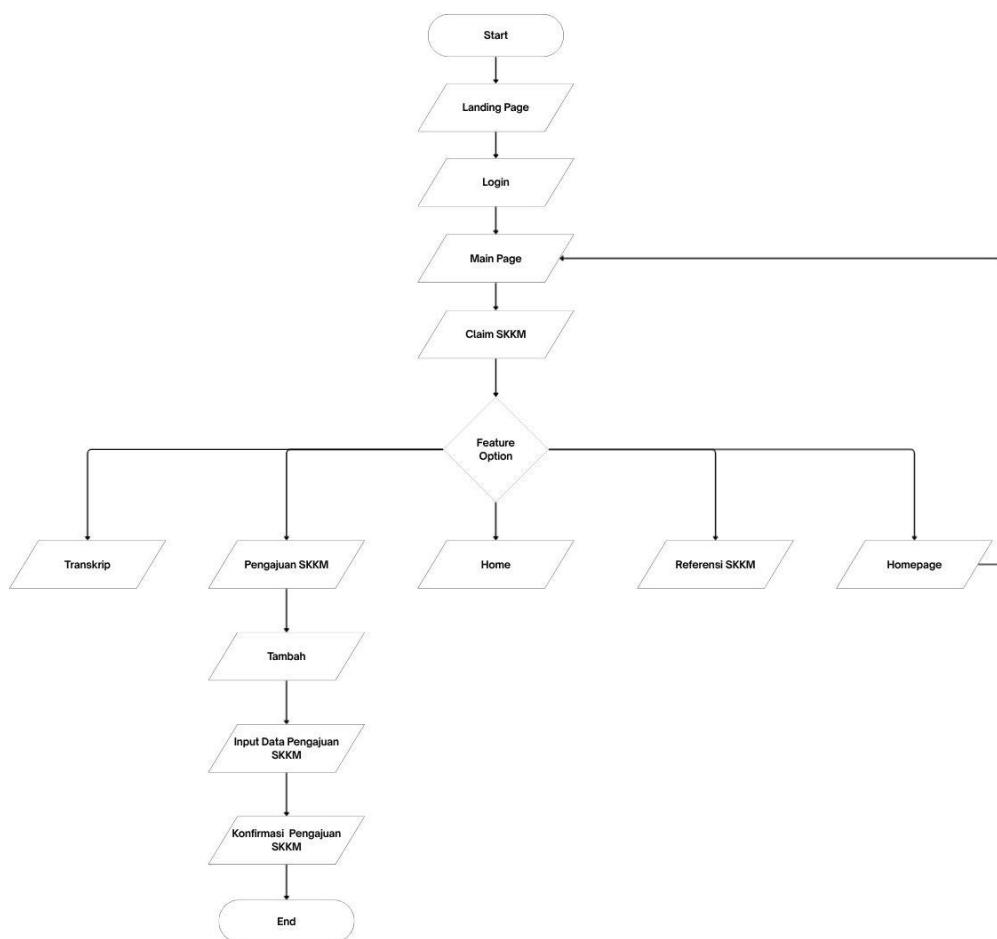


Gambar 3.18 Alur Sebelum Menggunakan Sistem Pengajuan SKKM Mahasiswa

Berdasarkan alur pada gambar 3.18 diatas, proses pengajuan SKKM sebelum menggunakan sistem masih dilakukan secara manual dan bergantung pada komunikasi melalui email. Proses dimulai ketika mahasiswa menyiapkan bukti keikutsertaan kegiatan seperti sertifikat, foto kegiatan, atau surat keterangan pendukung lainnya. Setelah seluruh dokumen disiapkan, mahasiswa mengirimkan bukti tersebut melalui email ke alamat resmi pengelola SKKM.

Selanjutnya, pihak *Student Development* menerima email pengajuan dan melakukan pengecekan dokumen secara manual. Apabila dokumen dinilai valid, *staff* melakukan penginputan poin SKKM ke dalam sistem akademik berdasarkan bukti kegiatan yang diajukan. Setelah proses *input* selesai, mahasiswa diminta untuk melakukan pengecekan rekapitulasi poin SKKM secara mandiri melalui portal *myUMN*. Proses ini tidak menyediakan mekanisme pelacakan status pengajuan secara langsung, sehingga mahasiswa harus menunggu tanpa kepastian waktu pemrosesan.

## B. Sesudah menggunakan sistem



**Gambar 3.19** Flowchart Tampilan Pengajuan SKKM Mahasiswa

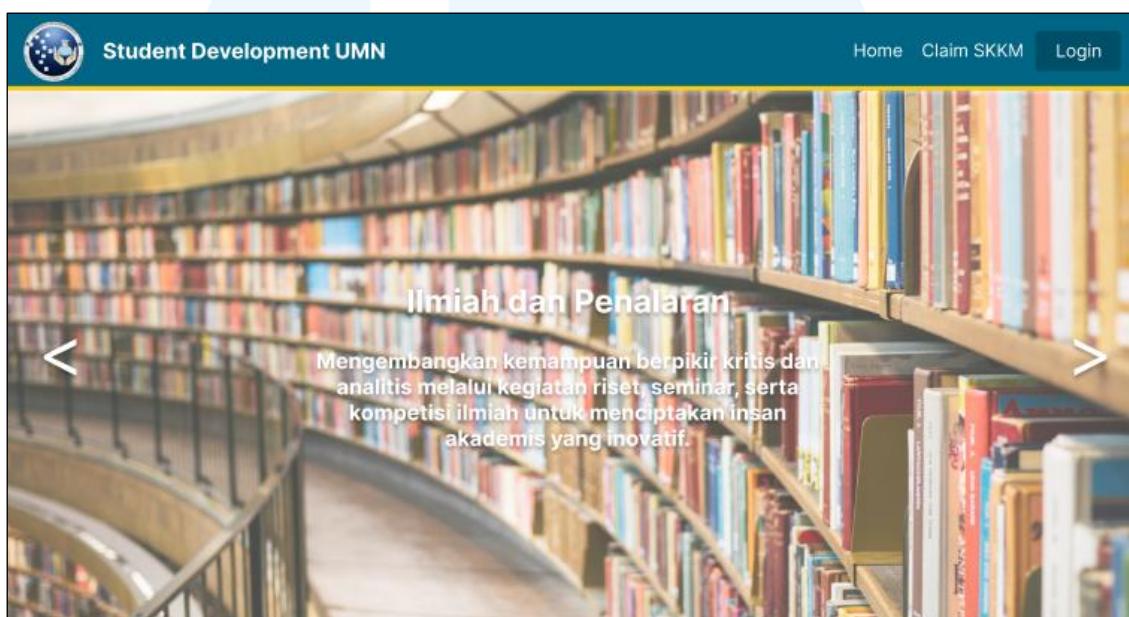
Pada gambar 3.19 diatas, alur proses dimulai ketika pengguna (*user*) mengakses halaman awal. Pengguna akan diarahkan untuk melakukan Login

menggunakan akun yang telah terdaftar. Jika proses autentikasi berhasil, sistem akan menampilkan Main Page atau halaman utama. Apabila login gagal, akses akan ditutup dan pengguna tidak dapat masuk ke dalam sistem.

Setelah berhasil login, pengguna dapat melakukan “Claim SKKM” melalui navigasi bar yang ada dan akan masuk ke dalam page untuk pengajuan SKKM. Terdapat beberapa pilihan fitur yang ada pada *sidebar* seperti *Home*, Pengajuan SKKM, Transkrip, dan Referensi SKKM. Fitur Pengajuan SKKM merupakan inti dari sistem ini, sehingga mahasiswa dapat membuat permohonan validasi kegiatan baru. Di dalam fitur ini, terdapat tombol “Tambah” yang memandu pengguna untuk mengisi formulir dan mengunggah bukti kegiatan. Setelah formulir diajukan, data tersebut akan diproses dalam sistem yang melibatkan pemeriksaan oleh admin. Hasil dari proses ini akan tercermin dalam Status Pengajuan yang nantinya dapat dilihat kembali oleh mahasiswa di halaman transkrip atau daftar tabel pengajuan. Fitur Transkrip memungkinkan mahasiswa untuk melihat rekapitulasi dan riwayat seluruh kegiatan yang telah diajukan beserta status dan bobot SKKM yang telah diperoleh. Fitur Referensi SKKM menyediakan panduan mengenai jenis kegiatan, jabatan, tingkat, dan bobot poin yang dapat diperoleh, serta membantu mahasiswa dalam merencanakan partisipasi kegiatan. Setelah seluruh proses selesai, alur pun berakhir.

Proyek perancangan *website ticketing* untuk pengajuan Satuan Kredit Kegiatan Mahasiswa (SKKM) ini dilatarbelakangi oleh kebutuhan untuk menciptakan efisiensi dalam proses administrasi kegiatan kemahasiswaan di Universitas Multimedia Nusantara. Selama ini, mahasiswa harus datang secara langsung ke kantor *Student Development* untuk mengajukan validasi kegiatan yang sering kali menimbulkan kendala seperti antrean panjang, keterbatasan waktu, dan potensi hilangnya dokumen atau melalui email namun data yang diberikan kurang lengkap agar pengajuan SKKM valid. Maka dari itu, proyek ini bertujuan untuk mengembangkan sebuah platform terpusat yang memungkinkan mahasiswa dapat mengajukan permohonan validasi SKKM

secara daring tanpa terkendala jarak dan waktu. Fokus proyek ini adalah pada perancangan sistem yang mencakup kemampuan unggah bukti kegiatan secara digital, pelacakan status verifikasi secara *realtime*, dan memberikan notifikasi otomatis. Walaupun proyek ini baru sampai pada tahap perancangan alur kerja dan antarmuka pengguna (UI/UX) tanpa implementasi penuh, hasilnya diharapkan dapat memberikan dasar yang kuat dan alur kerja yang jelas untuk pengembangan sistem *ticketing* SKKM yang lebih terintegrasi di masa depan.



Gambar 3.20 Halaman awal website pengajuan SKKM

Pada gambar 3.20 diatas, rancangan halaman awal untuk sistem *ticketing* SKKM dirancang dengan mempertimbangkan identitas visual dan kebutuhan informasi pengguna. Bagian paling atas halaman (*header*) menampilkan judul situs "Student Development UMN" untuk menegaskan platform sebagai bagian resmi dari universitas. Pada sebelah judul situs, *navigation bar* yang sederhana dan intuitif memuat menu "Home" untuk kembali ke halaman utama, "Claim SKKM" yang berfungsi sebagai pintu masuk utama bagi mahasiswa untuk memulai pengajuan, dan "Login" untuk akses sebagai mahasiswa atau admin yang dimana jika ingin melakukan pengajuan SKKM, diperlukan login terlebih dahulu sebagai validasi data pengguna. Bagian tampilan ini dirancang konsisten sehingga dapat ditampilkan di semua halaman untuk memudahkan navigasi.

Selanjutnya, halaman utama akan menyajikan pengantar mengenai pentingnya pengembangan *softskill*. Terdapat beberapa informasi dari *slide card* tersebut, salah satunya adalah "Ilmiah dan Penalaran", berisi penjelasan mengenai misi untuk mengembangkan kemampuan berpikir kritis dan analitis mahasiswa melalui berbagai kegiatan, seperti riset, seminar, dan kompetisi ilmiah. Tujuannya adalah untuk memberikan konteks dan motivasi kepada mahasiswa mengenai arti penting dari kegiatan-kegiatan yang akan diklaim SKKM-nya, sekaligus memperkuat nilai-nilai yang ingin dibangun oleh universitas.



**Peran Bidang Pengembangan Softskills**

Bidang Pengembangan Softskills di Universitas Multimedia Nusantara berperan dalam mengembangkan kecakapan lunak (soft skills) setiap mahasiswa. Kecakapan lunak merupakan atribut pribadi yang merefleksikan kemampuan emosional (Emotional Intelligence Quotient) yang meliputi:

- 1. Kemampuan kepemimpinan, seperti kemampuan mengambil keputusan dalam situasi sulit dan kritis, kemampuan mengelola dan mengarahkan orang (people management);
- 2. Kemampuan berpindah, bernegosiasi, dan berargumentasi secara persuasif;
- 3. Kemampuan mendengarkan pendapat orang lain secara empatik;
- 4. Kemampuan bekerja sama dalam tim dan bersinergi;
- 5. Kemampuan membagi waktu secara berkualitas;
- 6. Kemampuan kerja yang dapat diandalkan sesuai dengan fungsinya dalam struktur organisasi;
- 7. Kemampuan menerima kritik dan mentransformasi diri; dan
- 8. Kemampuan merencang, melaksanakan, dan membuat kegiatan kahasiswaan secara transparan dan bertanggung jawab.

Poin-poin di atas menjadi indikator dalam melihat perkembangan softskills dari setiap mahasiswa, sekaligus juga menjadi ajang latihan untuk menjadi lulusan yang berwatak baik dan profesional.

**Program Kerja Bidang Pengembangan Softskills**

- 1. Menyelenggarakan Pelatihan Leadership and Teamworking yang wajib diliiki oleh mahasiswa baru pada setiap semester genap.
- 2. Mengelola penyelenggaraan kegiatan mahasiswa yang dilaksanakan oleh organisasi-organisasi kahasiswaan.
- 3. Menggelar Latihan Dasar Kepemimpinan (LDK), dengan model outbound di luar kampus untuk para pengurus BEM, KBM, Himpunan Mahasiswa Prodi, dan UKM yang dilaksanakan satu tahun sekali.
- 4. Pengadaan seminar dengan tema-tema seputar softskills yang dilaksanakan setiap semester.
- 5. Visitasi dan studi banding ke universitas-universitas lain.
- 6. Orientasi Mahasiswa Baru menjelang dimulainya perkuliahan tahun ajaran baru yang dilaksanakan setahun sekali.
- 7. Wisuda sekaligus Dies Natalis UMN yang dilaksanakan setahun sekali.



**Gambar 3.21** Halaman awal website pengajuan SKKM

Pada gambar 3.21 diatas, penjelasan ini diikuti dengan pemaparan peran "Bidang Pengembangan *Softskill*" yang merinci delapan indikator kemampuan yang dikembangkan, seperti kepemimpinan, argumentasi, kerja sama tim, dan

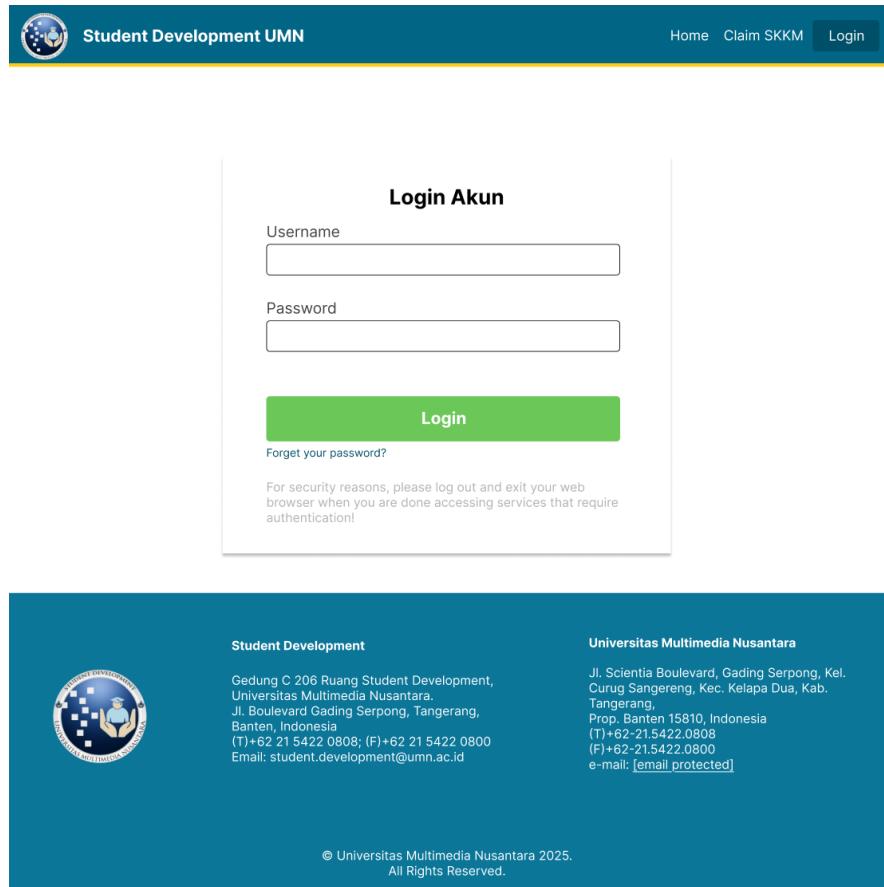
manajemen waktu. Dengan menampilkan informasi ini pada halaman awal, mahasiswa tidak hanya diajak untuk mengurus administrasi tetapi juga diingatkan pada tujuan akhir dari seluruh proses ini, yaitu pembentukan karakter dan profesionalisme.



**Gambar 3.22** Halaman awal website pengajuan SKKM

Pada gambar 3.22 diatas pada halaman awal juga akan menampilkan *footer* pada bagian akhir halaman berupa informasi terkait Student Development dan universitas secara jelas. Informasi ini mencakup alamat lengkap, nomor telepon, *faximile*, dan alamat email yang dapat dihubungi jika mahasiswa memerlukan bantuan langsung. Dengan menempatkan informasi kontak yang mudah ditemukan, website ini bertujuan untuk tetap mempertahankan unsur pelayanan dan interaksi, meskipun proses utamanya telah dialihkan ke sistem digital.





Gambar 3.23 Halaman login website pengajuan SKKM

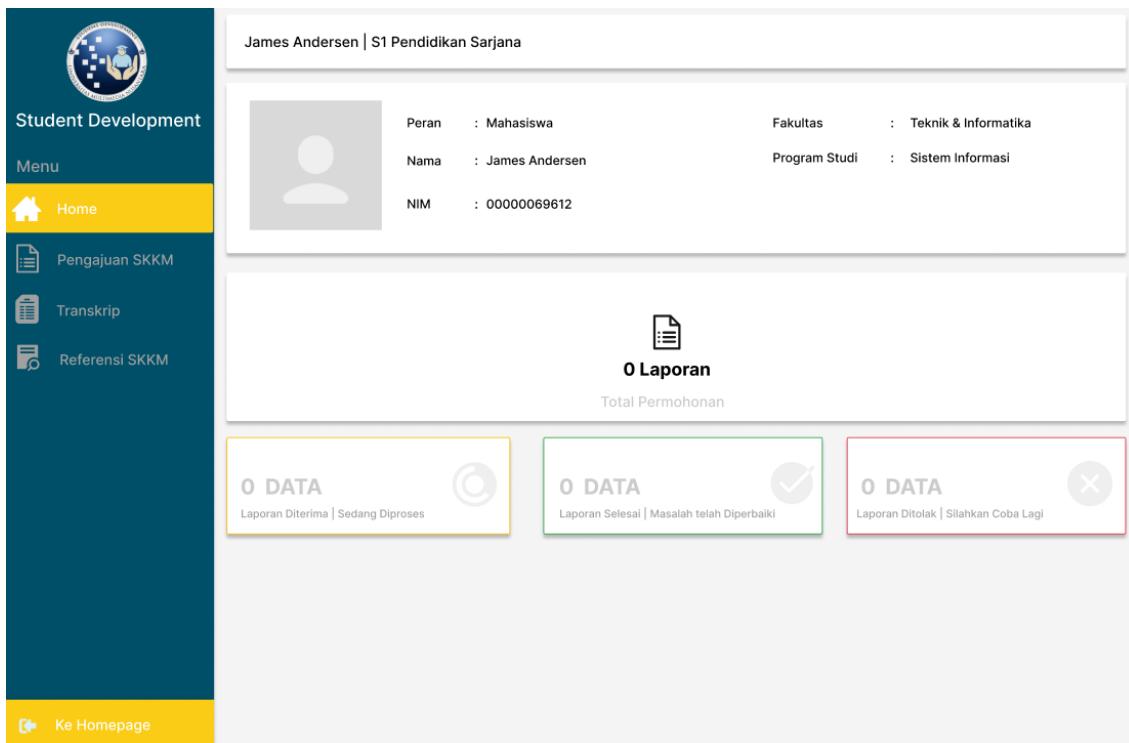
Pada gambar 3.23 diatas, halaman tersebut menampilkan halaman login yang dimana setelah pengguna klik menu "Login" dari halaman awal, pengguna akan diarahkan ke halaman login yang dirancang sebagai identifikasi identitas mahasiswa atau admin dari Universitas Multimedia Nusantara. Halaman ini merupakan gerbang utama untuk mengakses layanan pengajuan SKKM dan memastikan bahwa hanya pengguna yang terdaftar baik mahasiswa maupun admin yang dapat masuk ke dalam sistem. Formulir login hanya meminta dua kredensial utama, yaitu "Username" dan "Password" untuk meminimalisir kerumitan dan mempercepat proses autentikasi. Dalam mengantisipasi kasus lupa kata sandi, disediakan tautan "Forget your password?" yang akan memandu pengguna melalui proses pemulihan akun, sehingga dapat meningkatkan kesadaran akan keamanan siber.



Gambar 3.24 Header halaman awal setelah login

Pada gambar 3.24 diatas, menampilkan navigasi terbaru setelah pengguna berhasil melakukan autentikasi pada halaman login, pengguna akan diarahkan kembali ke halaman utama yang kini menampilkan antarmuka yang telah diperbarui. Perubahan paling signifikan terlihat pada bagian *header* navigasi di bagian atas halaman. Menu "*Claim SKKM*" sekarang dapat digunakan untuk diarahkan ke halaman lain untuk pengajuan yang sebelumnya berpindah ke halaman login terlebih dahulu. Selain itu, sebagai indikasi bahwa sesi telah aktif, *header* kini menampilkan informasi identitas pengguna berupa alamat email dalam contoh ini "[email@umn.ac.id](mailto:email@umn.ac.id)" yang dilengkapi dengan ikon panah *dropdown* yang dimana dapat melakukan *logout* jika selesai digunakan. Modifikasi *header* halaman awal ini tidak hanya berfungsi sebagai penanda visual status pengguna yang telah autentikasi, tetapi juga secara fungsional membuka akses penuh terhadap inti dari sistem *ticketing* SKKM, yaitu fitur pengajuan SKKM yang menjadi tujuan utama dari platform ini.

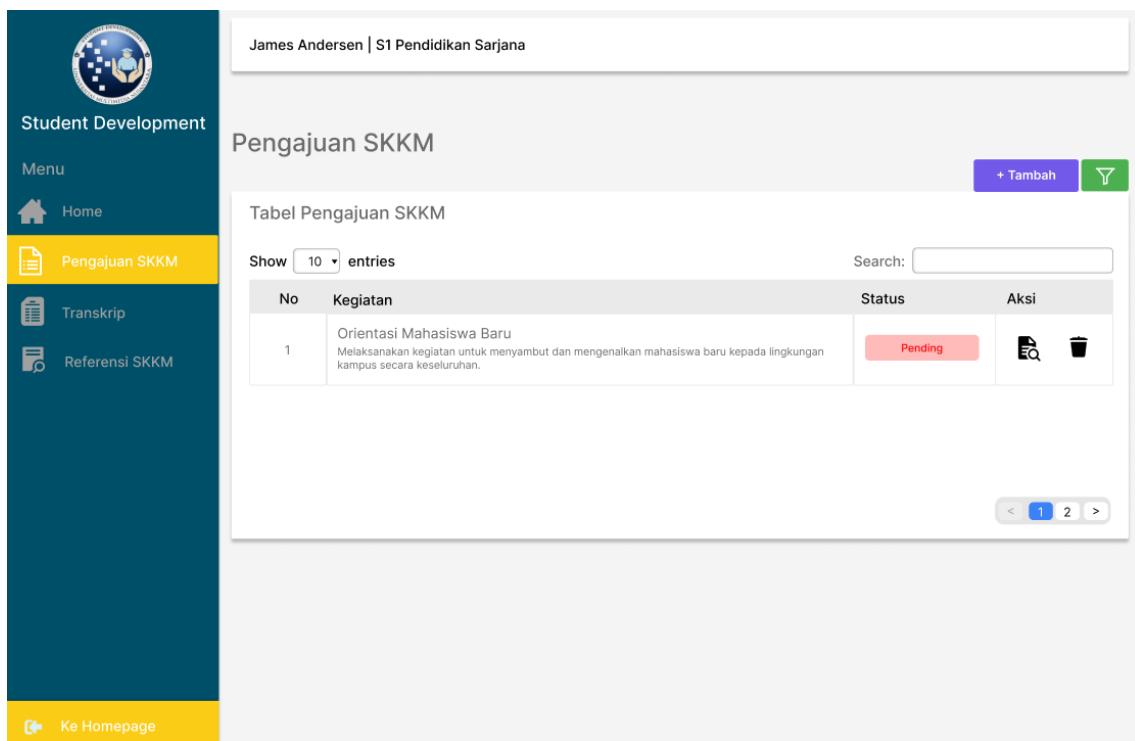




Gambar 3.25 Halaman *home* Claim SKKM

Pada gambar 3.25 diatas, pengguna yang telah login klik menu "Claim SKKM" pada *header* akan menampilkan halaman *dashboard* pribadi yang berfungsi sebagai pusat kendali utama untuk seluruh proses pengajuan Satuan Kredit Kegiatan Mahasiswa (SKKM). Halaman ini dirancang untuk menyajikan informasi yang rapi dan terstruktur, dimulai dengan sidebar menu yang memuat navigasi utama, yaitu "Home", "Pengajuan SKKM", "Transkrip", dan "Referensi SKKM", sehingga pengguna dengan mudah beralih antara fitur-fitur dalam sistem. Bagian profil menampilkan identitas pengguna secara detail, termasuk nama, program studi, dan NIM yang berfungsi untuk memastikan bahwa semua aktivitas pengajuan terhubung dengan identitas akademik yang benar. Elemen inti dari halaman ini adalah panel statistik yang menyajikan ringkasan visual status seluruh permohonan pengguna yang dikategorikan ke dalam empat kelompok: "Total Permohonan", "Laporan diterima", "Laporan diproses", "Laporan selesai", dan "Laporan Ditolak". Desain ini memungkinkan mahasiswa untuk memantau perkembangan pengajuan mereka secara *realtime* dalam satu tampilan yang terpusat, sehingga meningkatkan transparansi proses

dan memudahkan dalam mengambil keputusan lebih lanjut, seperti mengajukan ulang untuk klaim yang ditolak atau menunggu verifikasi untuk klaim yang masih diproses.



No	Kegiatan	Status	Aksi
1	Orientasi Mahasiswa Baru Melaksanakan kegiatan untuk menyambut dan mengenalkan mahasiswa baru kepada lingkungan kampus secara keseluruhan.	Pending	

**Gambar 3.26** Halaman Pengajuan SKKM

Pada gambar 3.26 diatas, menampilkan setelah pengguna memilih menu "Pengajuan SKKM" dari *sidebar* navigasi, sistem akan menampilkan halaman yang berfungsi sebagai pusat *monitoring* dan manajemen seluruh riwayat pengajuan Satuan Kredit Kegiatan Mahasiswa (SKKM). Halaman ini didominasi oleh komponen utama berupa tabel yang menampilkan daftar kegiatan yang telah diajukan oleh mahasiswa, dalam contoh ini menampilkan kegiatan "Orientasi Mahasiswa Baru" beserta deskripsi singkatnya. Untuk memudahkan navigasi dalam daftar yang potensial panjang, disediakan kontrol "Show entries" yang memungkinkan pengguna menentukan jumlah data yang ditampilkan per halaman, serta fitur *search* untuk menemukan pengajuan tertentu dengan cepat. Setiap entri dalam tabel juga dilengkapi dengan kolom "Status" yang memberikan informasi *realtime* tentang proses pengajuan seperti *pending*, *assign*, atau *reject* dan kolom "Aksi" yang berisi tombol-tombol untuk

melihat detail dan menghapus permohonan. Pada halaman ini, fokus utama pada fungsionalitas tabel untuk memastikan pengguna dapat melacak dan mengelola semua pengajuan SKKM secara efisien dalam satu antarmuka yang terorganisir.

The screenshot shows a web-based application for managing SKKM (Student Knowledge and Skill) applications. The main page is titled "Tambah Pengajuan SKKM" (Add SKKM Application). The form contains the following fields:

- Required Fields:** Nama Kegiatan\*, Bidang Kegiatan\*, Jenis SKKM\*, Tingkat\*, Jabatan/Status\*, Tanggal Pelaksanaan\*, Nomor SK/Sertifikat\*, and Sertifikat\*.
- Optional Fields:** Jumlah Peserta, Keterangan\*, and Scan Sertifikat\*.
- File Upload:** A field for "Scan Sertifikat\*" with a "Pilih Berkas..." (Select File) button. A note below states: "File hanya bisa menemima format pdf/image, maksimal besar file ≤2MB".
- Buttons:** "Kirim Pengajuan" (Send Application) and "Kembali" (Back).

The sidebar on the left is titled "Student Development" and includes a "Menu" with "Home", "Pengajuan SKKM" (which is highlighted in yellow), "Transkrip", and "Referensi SKKM". A "Ke Homepage" button is also present.

Gambar 3.27 Halaman permohonan pengajuan SKKM

Pada gambar 3.27 diatas, ketika pengguna klik tombol untuk membuat pengajuan baru, sistem akan menampilkan halaman formulir "Tambah Pengajuan SKKM" yang dirancang secara komprehensif untuk menampilkan seluruh informasi yang diperlukan dalam proses validasi kegiatan. Formulir ini terbagi dalam beberapa bagian untuk memandu pengguna dalam mengisi data secara sistematis. Bagian pertama berisi *field* seperti "Nama Kegiatan", "Bidang Kegiatan", dan "Jenis SKKM" yang berfungsi untuk mengklasifikasikan kegiatan secara tepat. Bagian kedua menangkap detail spesifik kegiatan termasuk "Tingkat" (lokalisasi kegiatan), "Jabatan/Status" peran mahasiswa

dalam kegiatan, "Tanggal Pelaksanaan", "Jumlah Peserta" (opsional), serta informasi sertifikasi seperti nomor dan tanggal sertifikat. Bagian penting dari formulir ini adalah bukti fisik kegiatan melalui *field* "Scan Sertifikat" yang dilengkapi dengan petunjuk teknis jelas mengenai format *file (pdf/image)* dan batasan ukuran maksimal 2MB untuk memastikan bahwa dokumen yang diunggah memenuhi standar sistem. Di bagian bawah formulir, tombol "Kirim Pengajuan" dan "Kembali" memberikan opsi yang jelas untuk menyelesaikan proses atau membatalkan pengisian. Desain *form* yang terstruktur dengan *field* wajib (ditandai bintang) ini tidak hanya memandu pengguna dalam melengkapi data secara lengkap tetapi juga meminimalisir kesalahan *input* dan memastikan kelengkapan dokumen pendukung sejak tahap pengajuan.



James Andersen | S1 Pendidikan Sarjana

**Hasil Kegiatan Mahasiswa**

**I. Kegiatan Wajib**

**A. Dari Universitas**

Status	No	Jenis Kegiatan	Kegiatan	Detail	Bobot
Diterima	1	Orientasi Mahasiswa Baru (OMB)	Mengikuti rangkaian kegiatan untuk pengenalan lingkungan kampus secara keseluruhan.		1.0
<b>Belum memenuhi bobot, minimal 20 Poin</b>				<b>Total</b>	1.0

**II. Kegiatan Tidak Wajib**

**A. Ilmiah dan Penalaran**

Status	No	Jenis Kegiatan	Kegiatan	Detail	Bobot
Diproses	1	Peserta Seminar/Diskusi	Mengikuti Seminar: "Bring AI Into Teaching And Learning"		1.0
<b>Telah memenuhi bobot, minimal 0 Poin</b>				<b>Total</b>	1.0

**B. Bakat dan Minat**

Status	No	Jenis Kegiatan	Kegiatan	Detail	Bobot
Ditolak	1	Lomba Video/Film/Fotografi	Mengikuti lomba pembuatan poster untuk kampus		2.0
<b>Telah memenuhi bobot, minimal 0 Poin</b>				<b>Total</b>	2.0

**C. Organisasi dan Pengembangan Kepribadian**

Status	No	Jenis Kegiatan	Kegiatan	Detail	Bobot
Diterima	1	Orientasi Mahasiswa Baru (OMB)	Mengikuti rangkaian kegiatan untuk pengenalan lingkungan kampus secara keseluruhan.		1.0
<b>Belum memenuhi bobot, minimal 6 Poin</b>				<b>Total</b>	1.0

**D. Pengabdian Masyarakat**

Status	No	Jenis Kegiatan	Kegiatan	Detail	Bobot
Belum Ada Kegiatan				<b>Total</b>	
<b>Telah memenuhi bobot, minimal 0 Poin</b>				<b>Total</b>	

**Gambar 3.28** Halaman transkrip SKKM

Pada gambar 3.28 diatas, halaman tersebut menampilkan tabel yang mengelompokkan berbagai kegiatan mahasiswa ke dalam kategori terstruktur. Tampilan ini terbagi menjadi dua bagian utama, yaitu "Kegiatan Wajib" dan "Kegiatan Tidak Wajib" yang masing-masing terbagi lagi menjadi sub-kategori lebih detail seperti "Ilmiah dan Penalaran", "Bakat dan Minat", serta "Organisasi dan Pengembangan Kepribadian". Setiap kategori menampilkan tabel yang berisi detail lengkap setiap kegiatan termasuk status verifikasi ("Diterima", "Ditolak", atau "Diproses"), jenis kegiatan, deskripsi aktivitas, dan bobot SKKM yang diperoleh. Fitur kunci dari halaman ini adalah tampilan

progress *tracking* yang jelas melalui indikator "Telah memenuhi bobot" atau "Belum memenuhi bobot" disertai informasi target minimal poin yang harus dicapai pada setiap kategori. Sebagai contoh, pada kategori "Kegiatan Wajib - Dari Universitas" terlihat bahwa mahasiswa telah mengumpulkan 1.0 poin dari OMB namun masih belum memenuhi target minimal 20.0 poin, sementara pada kategori "Bakat dan Minat" status "Ditolak" untuk lomba pembuatan poster memberikan transparansi mengenai kegiatan yang tidak disetujui beserta bobot yang tidak diakui. Desain laporan terstruktur ini memungkinkan mahasiswa tidak hanya melihat sejarah kegiatan mereka, tetapi juga melakukan perencanaan strategis untuk memenuhi persyaratan SKKM yang belum tercapai, sekaligus menjadi alat *monitoring* yang transparan bagi kemajuan perkembangan *softskills* mahasiswa selama masa studi.

Hasil Kegiatan Mahasiswa					
I. Kegiatan Wajib					
A. Dari Universitas					
Status	No	Jenis Kegiatan	Kegiatan	Detail	Bobot
Diterima	1	Orientasi Mahasiswa Baru (OMB)	Mengikuti rangkaian kegiatan untuk pengenalan lingkungan kampus secara keseluruhan.		1.0
Total 1.0					
Belum memenuhi					
II. Kegiatan Tidak Wajib					
A. Ilmiah dan Penelitian					
Status	No	Detail Laporan			
Diproses	1	Nama Kegiatan : ##### Bidang Kegiatan : ##### Jenis SKKM : ##### Tingkat : ##### Jabatan/Status : ##### Tanggal Pelaksanaan : ##### Jumlah Peserta : ##### Keterangan : #####           #####			
Total 1.0					
Tutup					
B. Bakat dan Minat					
Status	No	Jenis Kegiatan	Kegiatan	Detail	Bobot
Ditolak	1	Lomba Video/Film/Fotografi	Mengikuti lomba pembuatan poster untuk kampus		2.0
Total 2.0					
Telah memenuhi bobot, minimal 0 Poin					

Gambar 3.29 Halaman transkrip detail laporan SKKM

Pada gambar 3.29 diatas, ditampilkan *pop-up* atau *modal* yang muncul ketika pengguna menekan tombol ikon pada kolom "Detail" di tabel transkrip. Fungsi utama dari halaman ini adalah untuk menyajikan informasi lengkap dan

mendetail mengenai suatu kegiatan yang telah diajukan yang hanya menampilkan jenis kegiatan dan bobotnya. Fitur ini sangat penting untuk memastikan transparansi karena memungkinkan mahasiswa untuk meninjau ulang semua data yang mereka ajukan dan memverifikasi kebenaran informasinya. Selain itu, dengan menampilkan data yang sama yang dilihat oleh admin saat proses verifikasi, fitur ini juga membantu mahasiswa dalam memahami dasar persetujuan atau penolakan suatu pengajuan.



James Andersen | S1 Pendidikan Sarjana

Student Development

Menu

 Home

 Pengajuan SKKM

 Transkrip

 Referensi SKKM

 Ke Homepage

## Referensi SKKM

### Daftar Bobot SKKM

Show  entries

Search:

No	Kegiatan	Jabatan	Tingkat	Bobot
1	Pengurus Organisasi	Peserta	Universitas	1.0
2	Pengurus Organisasi	Ketua	Universitas	2.0
3	Seminar/Sosialisasi	Peserta	Universitas	1.0
4	Lomba GEMASTIK	Peserta	Nasional	2.0

**Gambar 3.30** Halaman referensi SKKM

Pada gambar 3.30 diatas, halaman referensi SKKM berfungsi sebagai panduan bagi mahasiswa untuk memahami skema dan bobot poin yang dapat diperoleh dari berbagai jenis kegiatan. Setelah pengguna mengakses menu "Referensi SKKM" dari *sidebar* navigasi, akan ditampilkan sebuah tabel referensi yang terstruktur dengan kolom-kolom meliputi "Kegiatan", "Jabatan",

"Tingkat", dan "Bobot". Tabel ini menyajikan daftar resmi yang memuat berbagai aktivitas yang diakui oleh universitas, contohnya seperti "Pengurus Organisasi", "Seminar/Sosialisasi", dan "Lomba GEMASTIK", disertai dengan variabel penilaian seperti peran atau jabatan dalam kegiatan (peserta atau ketua) serta tingkat penyelenggarannya (universitas atau nasional). Setiap kombinasi dari variabel-variabel ini memiliki bobot poin yang telah ditetapkan, seperti terlihat pada contoh peran menjadi "Ketua" pada "Pengurus Organisasi" tingkat "Universitas" memberikan bobot 2.0, sementara peran "Peserta" untuk kegiatan yang sama hanya memberikan 1.0 poin. Keberadaan halaman ini sangat penting untuk membantu mahasiswa dalam melakukan perencanaan strategis partisipasi kegiatan, untuk memprioritaskan jenis kegiatan yang dapat memberikan kontribusi optimal terhadap pemenuhan total poin SKKM yang disyaratkan. Dengan menyediakan informasi yang transparan dan terpusat, fitur ini tidak hanya memandu pengajuan klaim tetapi juga mendorong mahasiswa untuk terlibat aktif dalam kegiatan yang sesuai dengan minat dan perkembangan *softskill*.

### **3.3.2 Kendala yang Ditemukan**

Selama mengerjakan dua proyek utama, yaitu pengembangan sistem *ticketing Student Engagement* dan perancangan *website* pengajuan SKKM, penulis menemui beberapa kendala yang muncul secara alami dalam proses pengembangan sistem informasi. Kendala-kendala ini tidak bersifat menyalahkan pihak mana pun, melainkan menjadi bagian dari proses pembelajaran dan penyesuaian terhadap kebutuhan unit kerja.

#### **C. Kendala pada Proyek 1 (Sistem Ticketing Student Engagement)**

Dalam pengembangan sistem ticketing internal, terdapat beberapa hal yang menjadi tantangan selama proses implementasi:

##### **1. Pembaruan Data pada *Dashboard* Tidak Bersifat *Realtime***

Salah satu kendala utama adalah pembaruan data pada *dashboard* yang tidak terjadi secara langsung. Informasi seperti *Key Performance Indicator*

(KPI) maupun status laporan baru akan muncul setelah halaman *refresh*, sehingga perubahan yang dilakukan pengguna terkadang belum terlihat secara instan, terutama saat laporan dipindahkan antar sub divisi. Hal ini membuat proses pemantauan laporan perlu sedikit waktu tambahan.

## 2. Optimalisasi *Input Data* Bagi Pengguna *Part-Timer*

Dari sisi penggunaan, beberapa bagian formulir dan alur *input* yang ditujukan untuk pengguna *part-timer* masih belum sepenuhnya terintegrasi dengan baik. Tampilan dan akses fitur yang digunakan belum benar-benar mendukung kebutuhan kerja yang bersifat tidak penuh waktu (*fulltime*). Hal ini menyebabkan beberapa proses seperti pengisian data, pengecekan laporan, dan pembaruan status tiket, terasa kurang efisien bagi mereka. Maka diperlukan penyederhanaan tampilan serta penyesuaian alur kerja agar *part-timer* dapat menjalankan tugas administrasi secara lebih mudah dan konsisten meskipun tidak berada di sistem setiap hari.

## 3. Sistem Login Belum Menggunakan *Single Sign-On* (SSO)

Proses autentikasi login pada sistem saat ini masih menggunakan mekanisme tradisional berupa *username* dan *password* lokal. Sistem belum terintegrasi dengan *Single Sign-On* UMN seperti MyUMN atau layanan autentikasi kampus. Walaupun secara fungsi sudah berjalan, penggunaan autentikasi terpusat akan jauh lebih memudahkan terutama untuk mengelola akses dan keamanan pengguna. Hal ini akan menjadi tantangan apabila ke depannya diharapkan ada integrasi dengan sistem kampus yang lebih luas.

## 4. Perbedaan Alur Kerja Antar Sub divisi

Setiap sub divisi memiliki karakteristik laporan yang berbeda-beda. Menyatukan seluruh kebutuhan ke dalam satu sistem memerlukan penyesuaian terhadap alur kerja internal, sehingga beberapa fitur harus diatur ulang berdasarkan masukan dari masing-masing sub divisi. Selama proses pengembangan, penulis juga perlu menyesuaikan desain tampilan dengan kebutuhan tiap sub divisi. Setiap sub divisi memiliki karakteristik laporan yang berbeda, sehingga penyatuan alur kerja ke dalam satu sistem

memerlukan beberapa revisi tampilan dan struktur fitur agar sesuai dengan kebutuhan pengguna.

## B. Kendala pada Proyek 2 (Perancangan *Website Pengajuan SKKM*)

Pada proyek perancangan *website* SKKM, kendala yang ditemui lebih berkaitan dengan proses perancangan dan cakupan pengembangan:

## 1. Perancangan Lebih Terfokus pada Pengguna Mahasiswa

Rancangan awal lebih banyak berfokus pada alur mahasiswa untuk pengajuan SKKM. Selain itu, kebutuhan admin untuk memeriksa, memvalidasi, atau menolak pengajuan belum sepenuhnya terdeskripsikan dalam perancangan. Hal ini terjadi karena prioritas awal diarahkan pada penyelesaian desain bagi pengguna utama terlebih dahulu.

## 2. Variasi Format Bukti Kegiatan Mahasiswa

Selama proses analisis kebutuhan, ditemukan bahwa mahasiswa memiliki beragam jenis format bukti kegiatan (sertifikat, dokumentasi, surat tugas). Hal ini menimbulkan tantangan dalam menentukan format unggahan dan penyimpanan yang paling sesuai jika nantinya sistem akan dibangun.

### 3. Kebutuhan Validasi Berjenjang

Proses pengajuan SKKM idealnya memerlukan validasi bertahap dari berbagai pihak. Pada tahap perancangan, alur ini cukup kompleks sehingga membutuhkan analisis tambahan agar dapat dirancang dalam *flowchart* dan UI dengan tepat.

### 3.3.3 Solusi atas Kendala yang Ditemukan

Selama proses penggerjaan kedua proyek tersebut, berbagai kendala yang muncul dihadapi dengan langkah-langkah penyelesaian yang bertahap dan disesuaikan dengan kebutuhan sistem serta arahan dari pembimbing lapangan. Setiap kendala menjadi kesempatan untuk melakukan evaluasi dan menyesuaikan pengembangan agar sistem dapat berfungsi secara optimal dan dapat digunakan secara efektif oleh setiap sub divisi di bawah *Student*

*Engagement*. Berikut merupakan solusi yang ditempuh berdasarkan kendala yang ditemukan.

### **A. Solusi Kendala pada Proyek 1 (Sistem Ticketing Student Engagement)**

#### **1. Pembaruan *Dashboard* yang Tidak *Realtime***

Dalam mengatasi tampilan *dashboard* yang belum menampilkan data secara *realtime*, langkah awal yang dilakukan adalah melakukan evaluasi terhadap alur pembaruan data di sisi *front-end* dan *back-end*. Dilakukan penyesuaian pada beberapa bagian *query* dan struktur pemanggilan data agar dapat mempercepat pembaruan informasi. Meskipun belum seluruhnya diterapkan, rancangan solusi ini telah disiapkan sebagai referensi pengembangan berikutnya agar *dashboard* dapat memuat indikator KPI dan status laporan secara lebih dinamis.

#### **2. Optimalisasi dan Integrasi Fitur untuk Pengguna *Part-Timer***

Saat mengatasi alur kerja *part-timer* yang belum sepenuhnya efisien, dilakukan peninjauan ulang terhadap tampilan *form* dan navigasi fitur yang digunakan oleh admin *part-time*. Beberapa komponen antarmuka dirancang ulang agar lebih sederhana dan mudah dipahami, terutama pada bagian *input* data dan pembaruan status. Selain itu, disusun rekomendasi desain lanjutan untuk integrasi fitur yang lebih seragam antara pengguna *full-time* dan *part-time* sehingga alur kerjanya lebih konsisten. Rekomendasi ini diharapkan dapat menjadi panduan bagi tim Student Engagement dalam pengembangan berikutnya.

#### **3. Sistem Login yang Belum Terintegrasi SSO**

Untuk mengantisipasi kebutuhan integrasi dengan *Single Sign-On* UMN ke depannya, disusun *draft* rencana integrasi autentikasi dengan standar SSO kampus, termasuk alur login umum dan kebutuhan token autentikasi. Selain itu, mekanisme *login* saat ini juga telah disesuaikan agar mudah dikembangkan ke sistem autentikasi terpusat jika sewaktu-waktu proses

integrasi ingin dilakukan oleh tim IT UMN. Solusi ini memungkinkan sistem tetap dapat berjalan dan membuka peluang pengembangan integrasi keamanan di masa depan.

#### **4. Penyesuaian Alur Kerja untuk Tiga Sub divisi**

Dalam mengatasi kebutuhan yang berbeda dari *Student Service*, *Student Support*, dan *Student Development*, dilakukan penyempurnaan desain tampilan agar fleksibel dan dapat menyesuaikan kebutuhan masing-masing sub divisi. Selama proses pengembangan, penulis berkonsultasi dengan pembimbing lapangan dan *staff* terkait untuk menyesuaikan kategori laporan, struktur *form*, serta alur distribusi tiket. Penyelarasan ini menghasilkan tampilan yang lebih konsisten namun tetap responsif terhadap kebutuhan masing-masing sub divisi, sehingga meminimalkan kebingungan dalam penggunaan sistem.

### **B. Solusi Kendala pada Proyek 2 (Perancangan Website Pengajuan SKKM)**

#### **1. Fokus Perancangan yang Dominan pada Pengguna Mahasiswa**

Untuk menyeimbangkan kebutuhan mahasiswa dan admin, ditambahkan alur kerja tambahan pada rancangan *flowchart*, khususnya untuk proses verifikasi, peninjauan bukti kegiatan, dan validasi oleh admin *Student Development*. Tampilan UI pada bagian admin juga dirancang secara terstruktur agar dapat mendukung proses pengecekan status, riwayat pengajuan, serta tindak lanjut. Dengan demikian, rancangan sistem mencakup kebutuhan kedua pihak dan dapat dijadikan acuan pengembangan sistem yang lebih lengkap.

#### **2. Variasi Format Bukti Kegiatan**

Sebagai solusi terhadap keberagaman jenis berkas bukti kegiatan, dirancang fitur *upload* yang fleksibel dan mampu menerima beberapa format file umum seperti PDF, JPG, dan PNG. Selain itu, disiapkan konsep

pemisahan penyimpanan *file* berdasarkan kategori kegiatan agar mempermudah proses validasi. Rancangan teknis ini diharapkan dapat membantu pengembang di tahap berikutnya untuk memastikan sistem mampu menangani beragam jenis dokumen tanpa mengganggu performa aplikasi.

### **3. Kompleksitas Validasi Berjenjang**

Dalam mengatasi tantangan pada proses validasi yang melibatkan beberapa pihak, disusun alur *workflow* yang lebih terstruktur melalui diagram *flowchart* yang menggambarkan langkah validasi dari mahasiswa hingga admin. Perancangan ini dirinci secara bertahap agar setiap tahap persetujuan dapat terekam dengan baik. Solusi ini menyediakan dasar yang lebih jelas bagi pengembang sistem di masa depan sehingga proses validasi dapat diimplementasikan secara berurutan, akurat, dan terdokumentasi.

