

BAB 4

HASIL PENELITIAN

Berdasarkan metode penelitian, pada tahap ini menjelaskan proses pembuatan sistem informasi peminjaman sarana dan prasarana di FTTI dengan menggunakan metode prototipe.

4.1 COMMUNICATION

Comuunication atau tahapan awal dari metode prototipe yang digunakan pada pengembangan sistem ini dimulai dengan :

1. Wawancara

- a. Hasil Wawancara Wadek 2

pada tahapan ini dilakukan wawancara dengan narasumber yaitu Wakil Dekan 2 FTTI Unjaya dengan hasil wawancara ialah peminjaman sarana dan prasarana di kampus terutama di FTTI Unjaya masih tergolong manual dengan cara surat menyurat hingga menyebabkan terjadinya bentrok ketika peminjaman sarana prasarana.

- b. Hasil Wawancara Opsdik

Wawancara dengan pihak opsdik menghasilkan data barang dan data ruangan serta melihat riwayat peminjaman sarana dan prasarana.

2. Study literatur

tahapan study literatur ini berfokus pada literatur yang memiliki kesamaan dengan judul proposal yang akan dipakai, pada tahapan ini mencari sumber informasi yang ada pada jurnal dan buku yang dimanfaatkan untuk mengembangkan literatur pada pengembangan sistem ini.

3. Rumusan masalah

Berdasarkan hasil wawancara serta study literatur yang telah dilakukan rumusan masalah ini telah dideskripsikan pada latar belakang dengan permasalahan yang dapat diangkat ialah peminjaman atau manajemen sarana dan prasarana di FTTI Unjaya masih tergolong manual yaitu

dengan cara tertulis serta surat menyurat dengan rangkuman pada excel yang terkadang menyebabkan bentrok pada rangkuman peminjaman.

4. Tujuan masalah

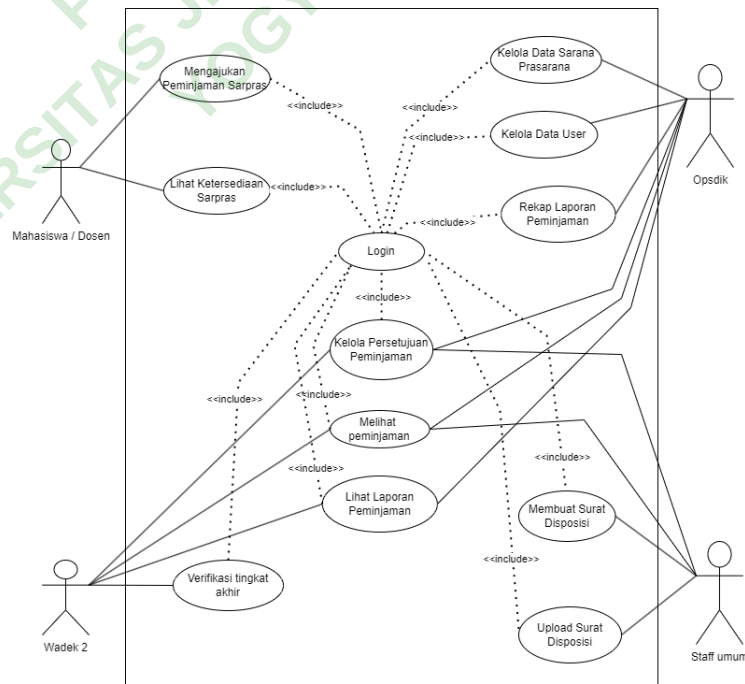
Penelitian ini memiliki tujuan untuk menciptakan sebuah sistem manajemen informasi sarana dan prasarana dengan mengimplementasikan metode prototipe di FTTI Unjaya yang berfokus pada peningkatan pengelolaan infrastruktur dan fasilitas kampus.

4.2 QUICK PLAN

Tahap ini menjelaskan proses dimana rancangan sistem dilakukan dengan menggambarkan proses untuk memecahkan permasalahan pengguna. Berikut adalah perancangan sistem sebagai berikut :

4.2.1 Use Case Diagram

Proses yang dilakukan oleh pengguna didalam sistem digambarkan dalam use case diagram. *Use case diagram* sistem peminjaman sarana dan prasarana dengan implementasi metode *prototype* di Fakultas Teknik dan Teknologi Informasi ditunjukan pada Gambar 4.1.



Gambar 4.1 Use case diagram

Pada Table 4.1 merupakan penjelasan tentang kegiatan aktor pada sistem. Aktor admin merupakan admin opsdik, sedangkan *user* umum adalah mahasiswa dan dosen di Fakultas Teknik dan Teknologi Informasi.

Tabel 4.1 Tabel *use case diagram*

Aktor	Use Case	Deskripsi
Mahasiswa/Dosen, Staff umum, Petugas Opsdik, Wadek 2.		Proses login
Admin (Petugas Opsdik)	Kelola Data Sarana dan Prasarana	Aktivitas yang menggambarkan admin mengelola data sarana dan prasarana
	Kelola Data User	Aktivitas yang menggambarkan admin melakukan pengelolaan data user
	Kelola Persetujuan Peminjaman	Aktivitas yang menggambarkan admin mengelola persetujuan peminjaman
	Lihat Laporan Peminjaman	Aktivitas yang menggambarkan informasi laporan peminjaman
	Rekap Laporan Peminjaman	Aktivitas yang menggambarkan admin merekap laporan data peminjaman
	Melihat Peminjaman	Aktivitas yang menggambarkan admin melihat detail peminjaman
Mahasiswa/Dosen	Mengajukan Peminjaman	Aktivitas yang menggambarkan pengguna mengajukan peminjaman
	Lihat Laporan Peminjaman	Aktivitas yang menggambarkan

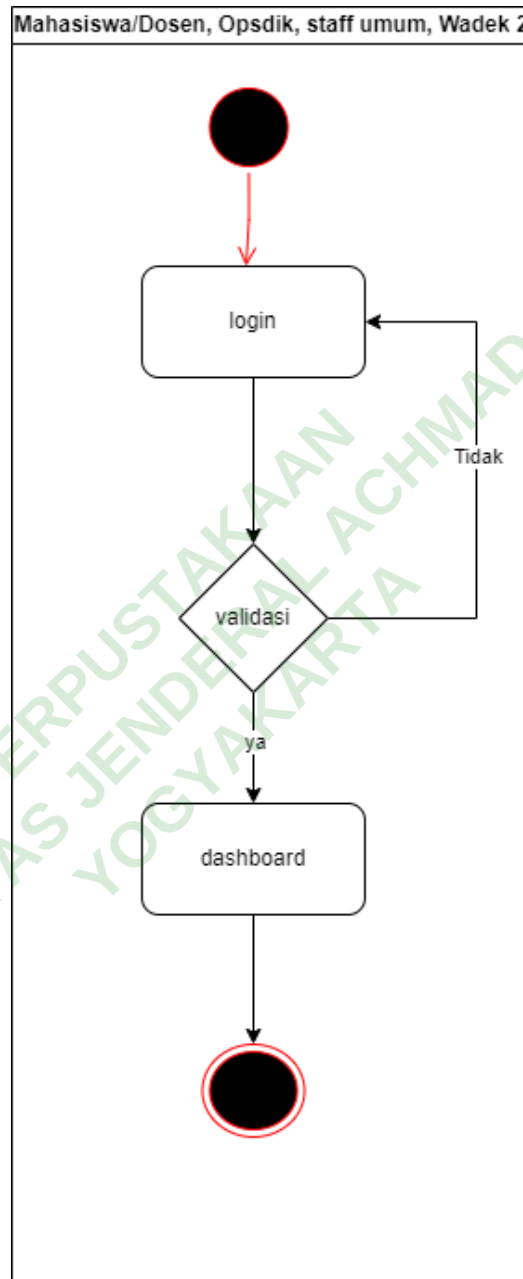
		informasi laporan peminjaman
	Melihat Ketersediaan Barang	Aktivitas yang menggambarkan Mahasiswa/Dosen melihat ketersediaan Sarana dan Prasarana.
	Melihat Peminjaman	Aktivitas yang menggambarkan Mahasiswa/Dosen dapat melihat peminjaman Sarana dan Prasarana
Staff umum	Kelola persetujuan peminjaman	Aktivitas yang menggambarkan staff umum melihat surat peminjamna yang diajukan oleh pengguna.
	Melihat peminjaman	
	Membuat surat disposisi	
Wakil Dekan 2	Melihat Ketersediaan Sarana Prasarana	Aktivitas yang menggambarkan ketersediaan laporan sarana dan prasarana
	Melihat Laporan Peminjaman	Aktivitas yang menggambarkan laporan peminjaman sarana dan prasarana
	Verifikasi Peminjaman	Aktivitas yang menggambarkan persetujuan peminjaman sarana dan prasarana

4.2.2 Activity Diagram

Hubungan antara aktor, logika, proses bisnis, dan alur kerja *use case* digambarkan dalam activity diagram. Bagaimana proses awal aktivitas, *decision* yang terjadi di dalam aktivitas, dan bagaimana alur berakhir Menurut Sari, (2019). Alur aktivitas yang ada dapat dijelaskan sebagai berikut :

1. Activity diagram login

Pada aktivitas diagram login Mahasiswa, Opsdik, Wadek 2, Staff umum melakukan aktivitas seperti yang ditunjukkan pada Gambar 4.2

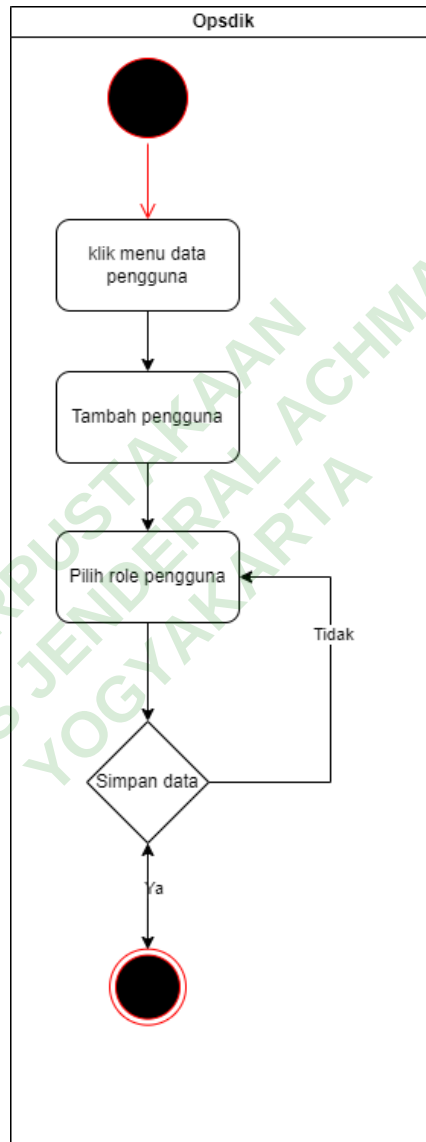


Gambar 4.2 Activity diagram login

Dari Gambar 4.2. admin, Wadek dan pengguna dapat melakukan login ke halaman *dashboard* dengan memasukan *username* dan *password*. Jika *username* dan *password* yang dimasukkan salah, akan Kembali ke halaman login.

2. Activity diagram registrasi akun pengguna

Pengelolaan data pendaftaran pengguna merupakan kegiatan yang dilakukan ketika pengguna belum memiliki akun . Untuk melakukan pendaftaran akun pengguna, admin harus melakukan aktivitas yang di gambarkan pada Gambar 4.3

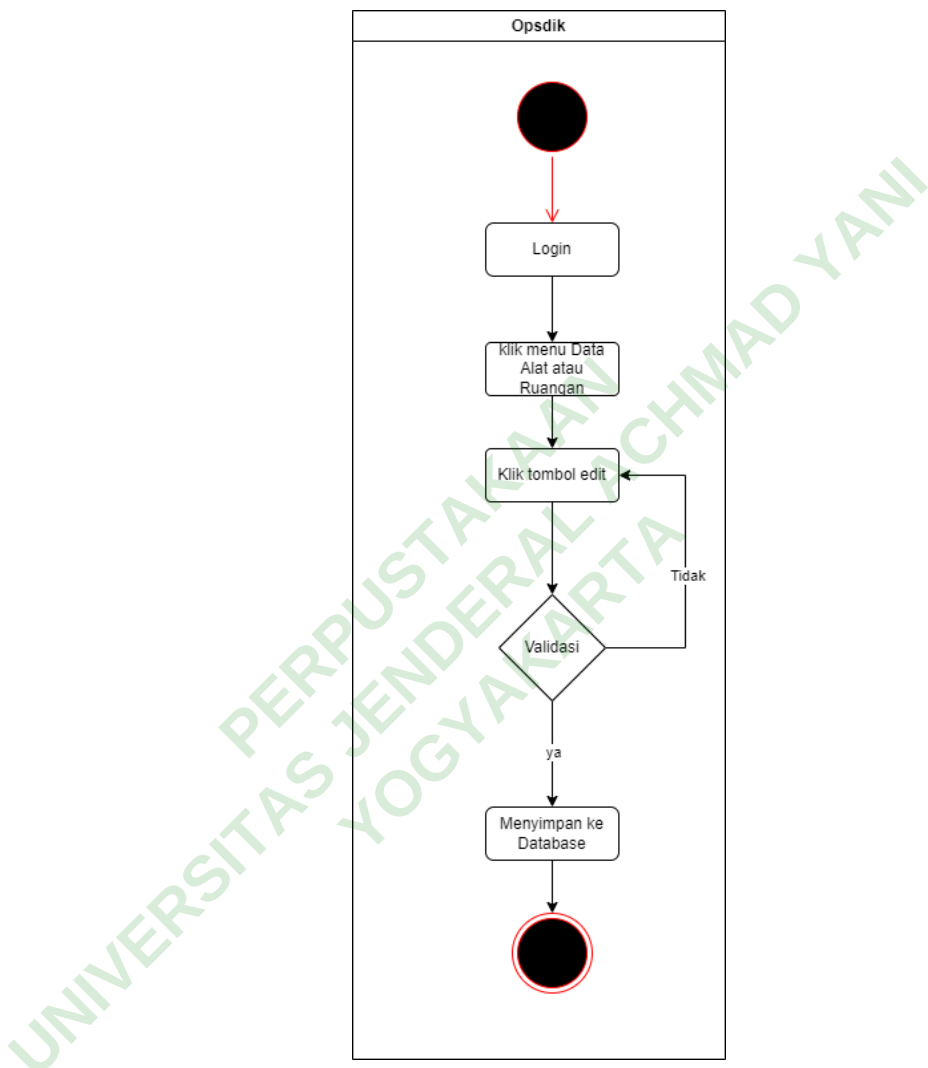


Gambar 4.3 Activity diagram registrasi

Dari diagram aktivitas Gambar 4.3, Opsdik melakukan registrasi akun pengguna agar dapat login. Apabila sudah melakukan registrasi maka sistem akan memvalidasi dan jika validasi gagal maka otomatis Kembali ke form pendaftaran.

3. Activity diagram manajemen data sarana dan prasarana

Pengelolaan data sarana prasarana merupakan kegiatan yang dilakukan oleh admin opsdik. Untuk melakukan update data sarana dan prasarana, admin harus melakukan aktivitas yang di gambarkan pada Gambar 4.4



Gambar 4.4 Activity diagram manajemen data sarpras

Dari diagram aktivitas Gambar 4.4, admin harus login terlebih dahulu kemudian mengakses menu sarana dan prasarana. Di halaman menu sarana dan prasarana, admin bisa melihat data sarana dan prasarana. Ketika data sarana dan prasarana akan diperbarui maka admin harus klik tombol edit dan nantinya akan menampilkan form data saat ini untuk melakukan pembaruan. Apabila data yang di masukkan tidak valid maka akan menampilkan pesan error berwarna merah dan

akan kembali ke halaman sarana dan prasarana, sedangkan jika data yang di masukkan valid maka akan menampilkan pesan berhasil berwarna hijau dan data terbaru akan tersimpan ke dalam database.

4. Activity diagram staff umum

Pada aktivitas Gambar 4.5 menunjukan informasi aktivitas yakni staff umum.

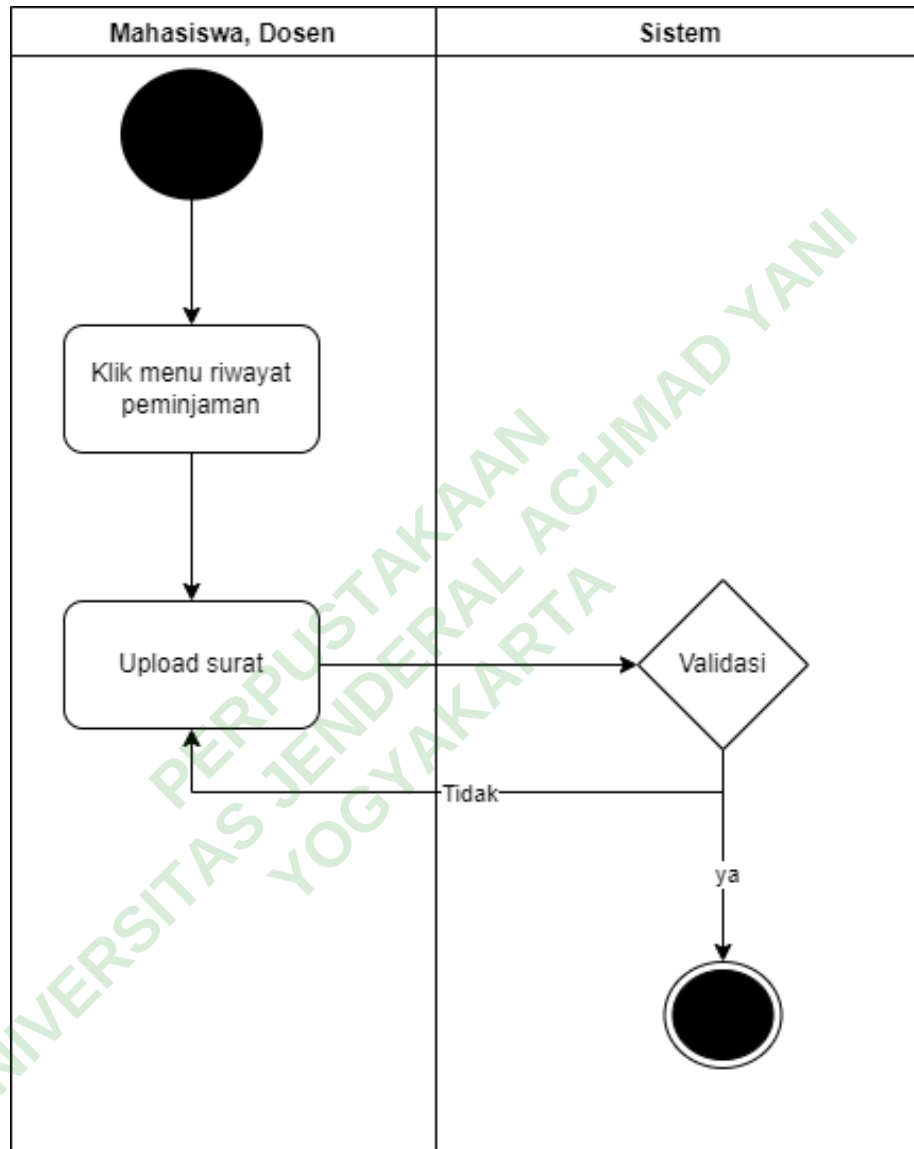


Gambar 4.5 Activity diagram verifikasi opsdik

Dari diagram atau Gambar 4.5 diatas menjelaskan staff umum melakukan verifikasi berkas tahap 1 dengan cara menuju ke halaman riwayat peminjaman yang kemudian dapat melihat informasi detail peminjam serta melakukan konfirmasi peminjaman.

5. Activity diagram upload surat peminjaman tahap 2

Aktivitas tahap 2 atau kegiatan upload surat sebagai syarat wajib pengguna untuk melakukan peminjaman ditunjukkan pada Gambar 4.6 dibawah ini.

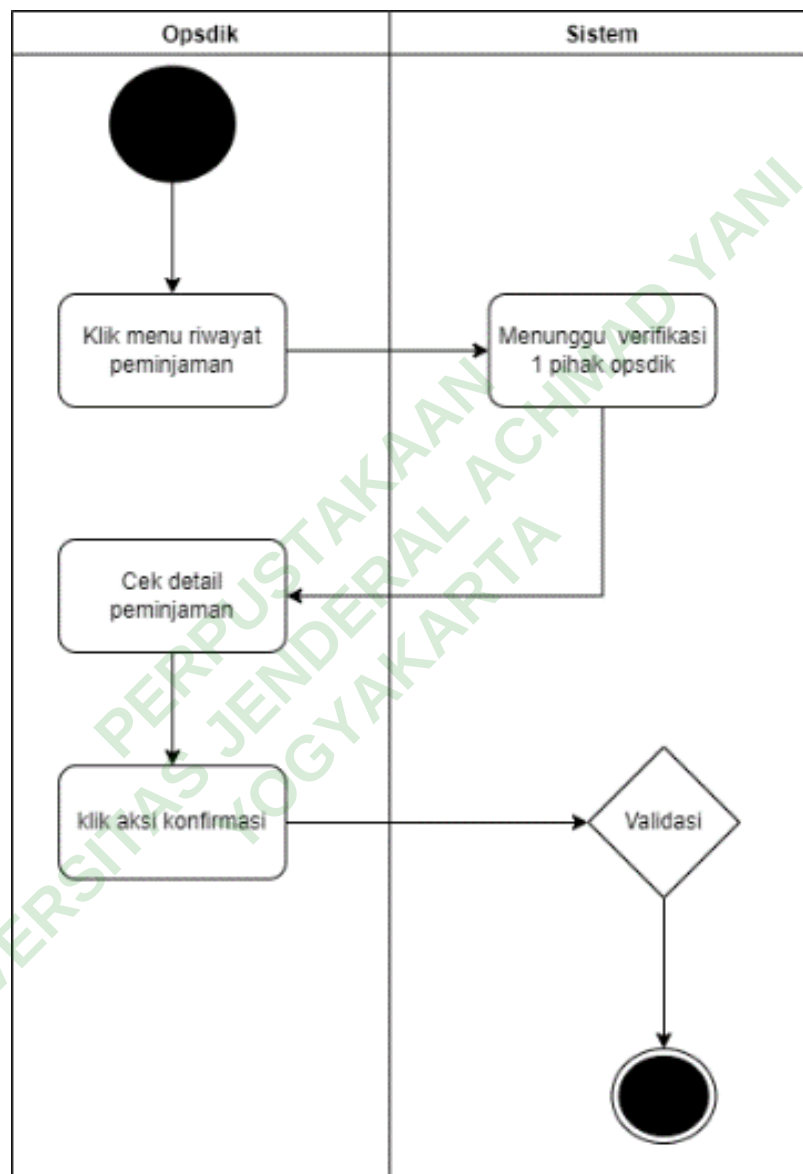


Gambar 4.6 Activity diagram upload surat tahap 2

Dari Gambar 4.6 atau diagram aktivitas tersebut menunjukkan pengguna harus mengunggah berkas surat peminjaman terlebih dahulu ketika ingin melakukan peminjaman alat atau ruangan yang tersedia.

6. Activity diagram verifikasi berkas tahap 3 Wakil Dekan 2

Aktivitas tahap 3 pada Gambar 4.7 tersebut menunjukkan wakil dekan melakukan verifikasi yang dimana terdapat informasi detail peminjam dan konfirmasi surat.

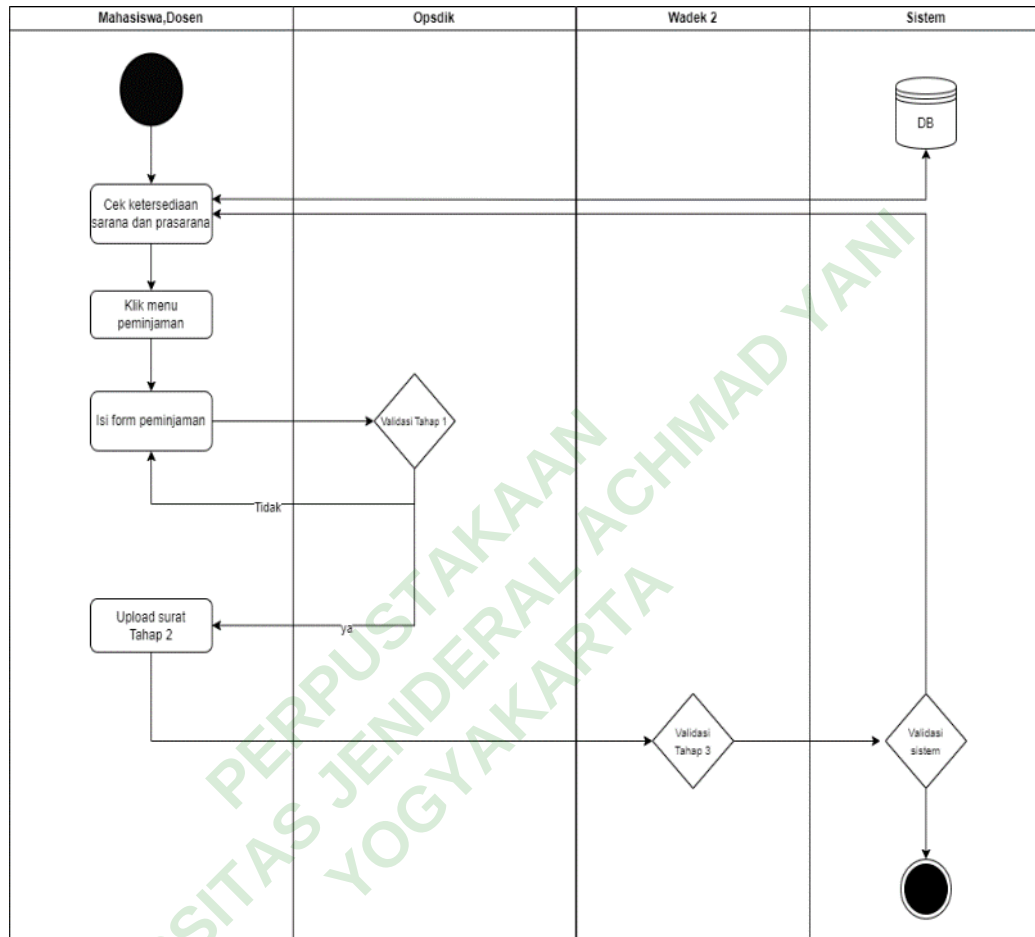


Gambar 4.7 Activity diagram verifikasi berkas tahap 3

Dari Gambar 4.7 atau diagram aktivitas verifikasi berkas tahap 3 ini menunjukkan informasi verifikasi yang dilakukan oleh wakil dekan 2 dengan cara menuju ke halaman riwayat peminjaman untuk mengetahui detail peminjaman yang telah diajukan oleh pengguna.

7. Activity diagram peminjaman sarana prasarana

Pengelolaan data peminjaman sarana dan prasarana pada sistem ditunjukkan pada Gambar 4.8 seperti berikut :

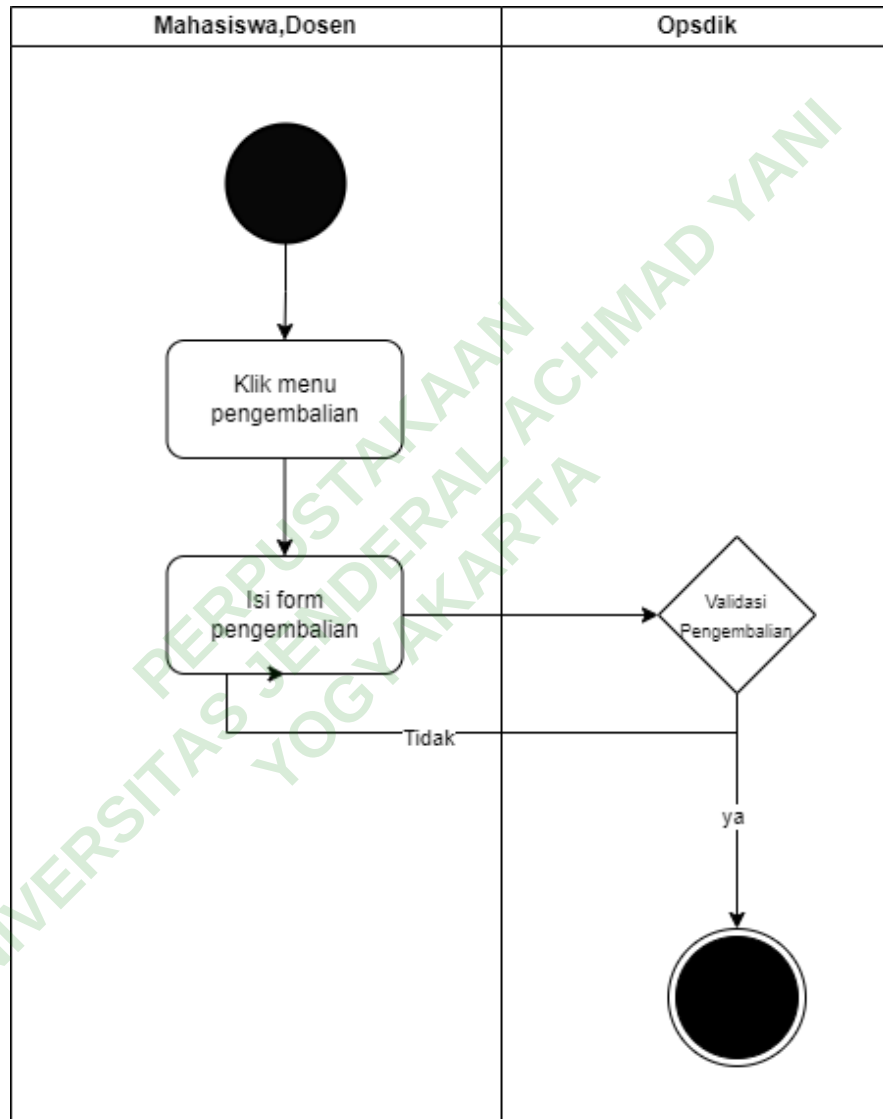


Gambar 4.8 Activity diagram peminjaman data sarana dan prasarana

Aktivitas tahapan pada Gambar 4.8 tersebut menunjukkan bahwa ketika mahasiswa atau dosen ingin meminjam sarana prasarana maka menuju menu peminjaman yang dimana nanti terdapat form peminjaman yang harus di isi kemudian akan di validasi oleh petugas opsdik. Jika sudah valid maka mahasiswa atau dosen melampirkan surat peminjaman pada form yang telah di sediakan yang kemudian akan di validasi atau di cek oleh Wakil Dekan 2.

8. Activity diagram pengembalian sarana dan prasarana

Pengelolaan data pengembalian sarana dan prasarana pada sistem ditunjukkan pada Gambar 4.9 seperti berikut :

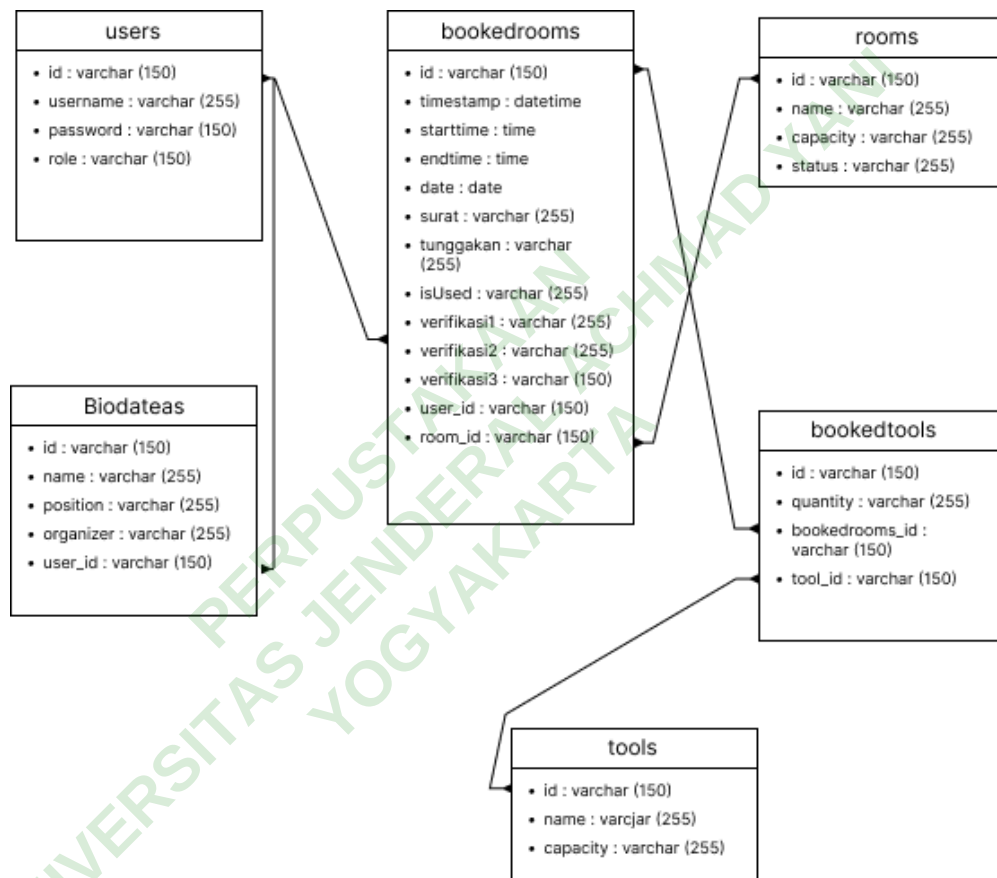


Gambar 4.9 Activity diagram pengembalian data sarpras

Aktivitas tahapan pada Gambar 4.9 tersebut menunjukkan bahwa ketika mahasiswa atau dosen ingin mengembalikan sarana prasarana maka menuju menu pengembalian yang dimana nanti akan ada form pengembalian yang harus di isi kemudian akan di validasi oleh petugas opsdik.

4.2.3 Class Diagram

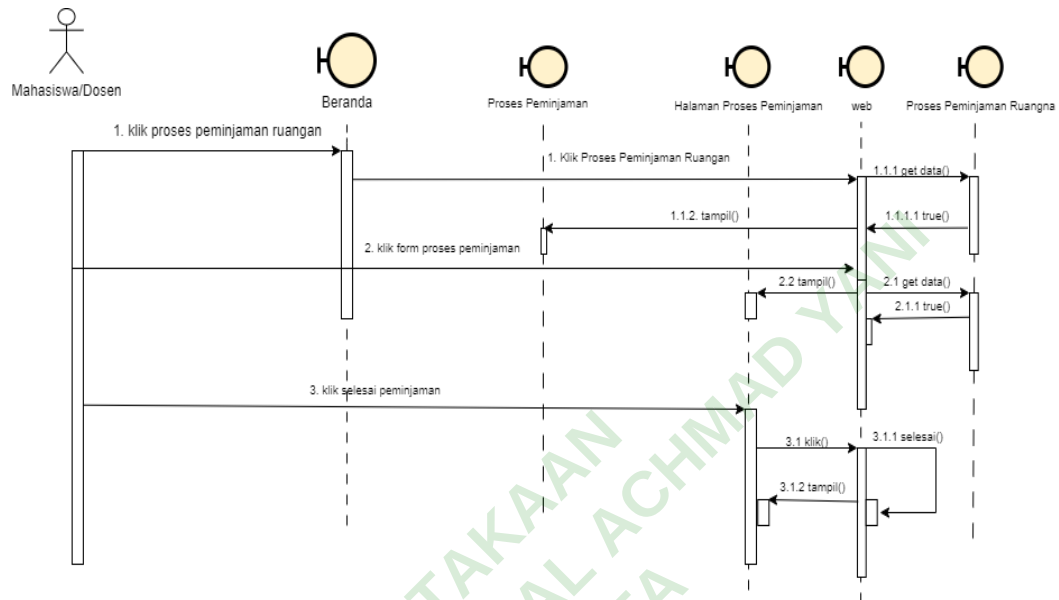
Class Diagram adalah gambaran dari pertemuan objek dengan properti, perilaku dan koneksi yang sama. *Class diagram* dapat memberikan perspektif yang lebih luas pada suatu sistem. *Class diagram* pada implementasi metode prototipe dalam pengembangan sistem informasi sarana dan prasarana di Fakultas Teknik dan Teknologi Informasi ditunjukkan pada Gambar 4.10



Gambar 4.10 Class diagram

4.2.4 Sequence diagram

Sequence Diagram adalah sebuah diagram yang menggambarkan kolaborasi dari objek – objek yang saling berinteraksi antar elemen dari suatu *class*.



Gambar 4.11 *Sequence diagram*

4.3 MODELLING QUICK DESIGN

Pada tahap ini akan dibuat sebuah desain *mockup* yang merupakan tampilan yang berhubungan langsung dengan pengguna. Tujuan dari *mockup* ini ialah untuk memberikan informasi *User Interface* kepada pengguna agar terlihat lebih bagus sebelum diimplementasikan menggunakan Flask, berikut merupakan desain *mockup* pada sistem :

4.3.1 Halaman login

Pada desain mockup admin selaku petugas opsdik, Wakil Dekan 2, serta Mahasiswa dapat mengakses halaman login. Form input disediakan di halaman ini untuk memasukkan username dan password. Desain halaman login ditunjukkan pada gambar 4.12

Gambar 4.12 Desain mockup halaman login

4.3.2 Halaman data pengguna (Petugas Opsdik)

Halaman data pengguna ini terdapat informasi data pengguna yang wajib ketika ingin meminjam sarana dan prasarana di lingkungan kampus serta terdapat tambah user untuk admin atau petugas opsdik yang ditunjukkan pada Gambar 4.8.

No	NPM/NIP	NAMA	PRODI	STATUS	JABATAN	AKSI
1	202102027	C. Nata	Informatika	Mahasiswa	HMIF	Edit Delete

Gambar 4.13 Desain mockup data pengguna

4.3.3 Halaman tambah data pengguna (Petugas Opsdik)

Desian halaman tambah pengguna pada sistem ini akan terdapat formulir yang berisi nama lengkap, program studi, role, jabatan, username, serta password yang dapat dilihat pada Gambar 4.14.

The mockup shows a web interface for adding a user. On the left is a sidebar with the FTI UNJAYA logo and a menu: Dashboard, Data Pengguna, Data Ruangan, Data Alat, Riwayat Peminjaman, and Riwayat Pengembalian. The main content area is titled 'Tambah User' and contains a form with the following fields: 'Nama Lengkap' (text input), 'Program Studi' (text input), 'Role' (text input), 'Jabatan' (text input), 'username (NPP/NIP)' (text input), and 'Password' (text input). To the right of the form is a 'Delete' button. At the bottom right of the form is a 'Tambah' button. Below the form, it says 'showing 1 to 3 of 3 entries' and has 'Previous', '1', and 'Next' pagination controls.

Gambar 4.14 Desain mockup tambah data pengguna

4.3.4 Halaman data ruangan (Petugas Opsdik)

Desian halaman data ruangan ini terdapat informasi ruangan, status, hingga edit atau hapus ruangan seperti yang ditunjukkan pada Gambar 4.15.

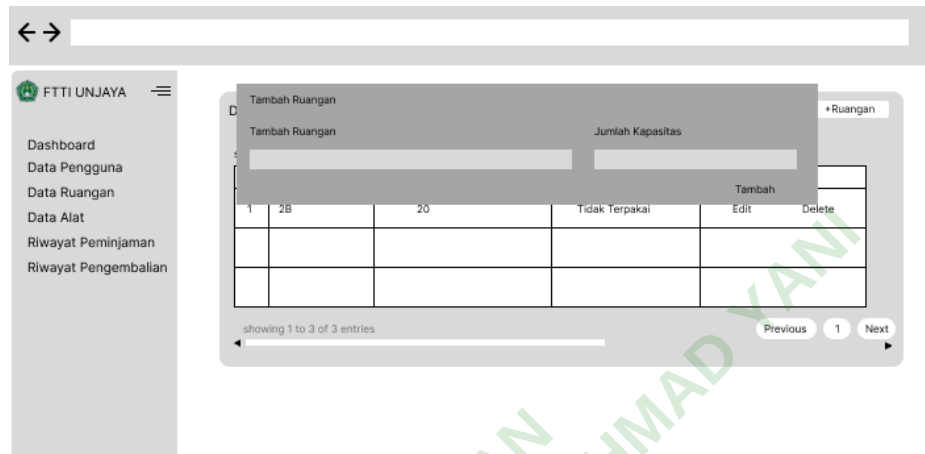
The mockup shows a web interface for viewing room data. On the left is the same sidebar as in Gambar 4.14. The main content area is titled 'Data Ruangan' and contains a table with the following columns: 'No', 'RUANGAN', 'KAPASITAS', 'STATUS', and 'AKSI'. The first row of data shows: No 1, RUANGAN 2B, KAPASITAS 20, STATUS Tidak Terpakai, and AKSI with 'Edit' and 'Delete' buttons. Below the table, it says 'showing 1 to 3 of 3 entries' and has 'Previous', '1', and 'Next' pagination controls.

No	RUANGAN	KAPASITAS	STATUS	AKSI
1	2B	20	Tidak Terpakai	Edit Delete

Gambar 4.15 Desain mockup data ruangan

4.3.5 Halaman tambah data ruangan (Petugas Opsdik)

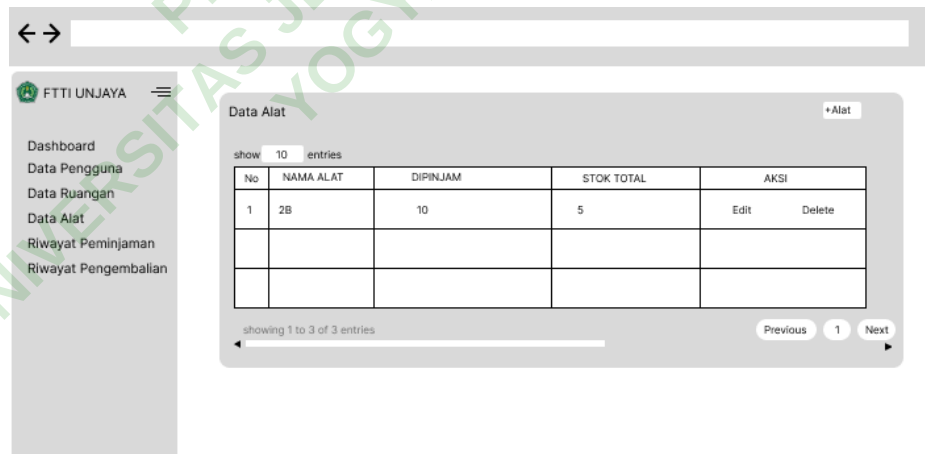
Desian halaman tambah data ruangan ini terdapat tambah informasi ruangan dan jumlah kapasitas ruangan seperti yang ditunjukkan pada Gambar 4.16.



Gambar 4.16 Desain mockup tambah data ruangan

4.3.6 Halaman data alat (Petugas Opsdik)

Desian halaman data alat terdapat tabel nama alat, stok alat, serta aksi yang dapat merubah atau menghapus data alat seperti yang ditunjukkan pada Gambar 4.17.



Gambar 4.17 Desain mockup data alat

4.3.7 Halaman tambah data alat (Petugas Opsdik)

Desian halaman tambah data alat terdapat form tambah alat serta jumlah alat atau stok alat yang tersedia seperti yang ditunjukkan pada Gambar 4.18.

Gambar 4.18 Desain mockup tambah data alat

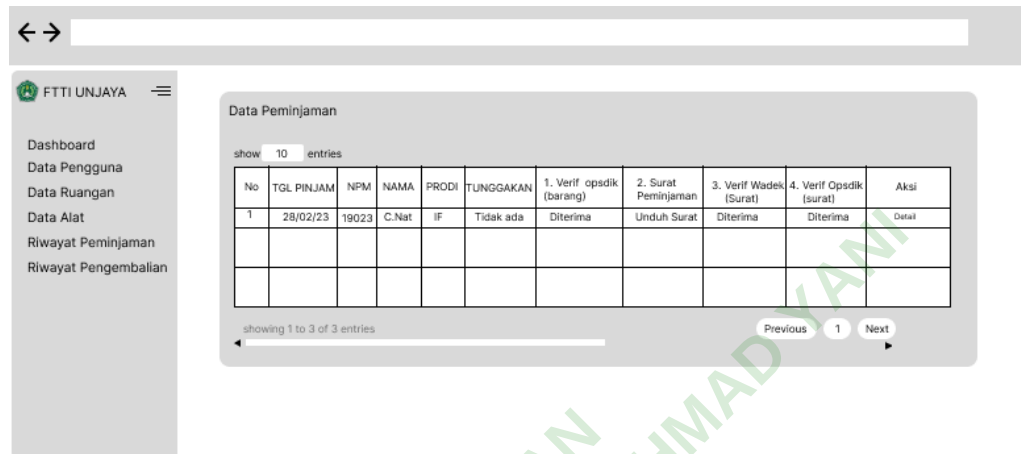
4.3.8 Halaman riwayat peminjaman (Petugas Opsdik)

Desian halaman riwayat peminjaman terdapat tabel yang memberikan informasi peminjaman dari tanggal peminjaman, nama peminjam, verifikasi dari pihak terkait hingga dapat mengunduh surat peminjaman yang ditunjukkan pada Gambar 4.19.

Gambar 4.19 Desain mockup riwayat peminjaman

4.3.9 Halaman riwayat pengembalian (Petugas Opsdik)

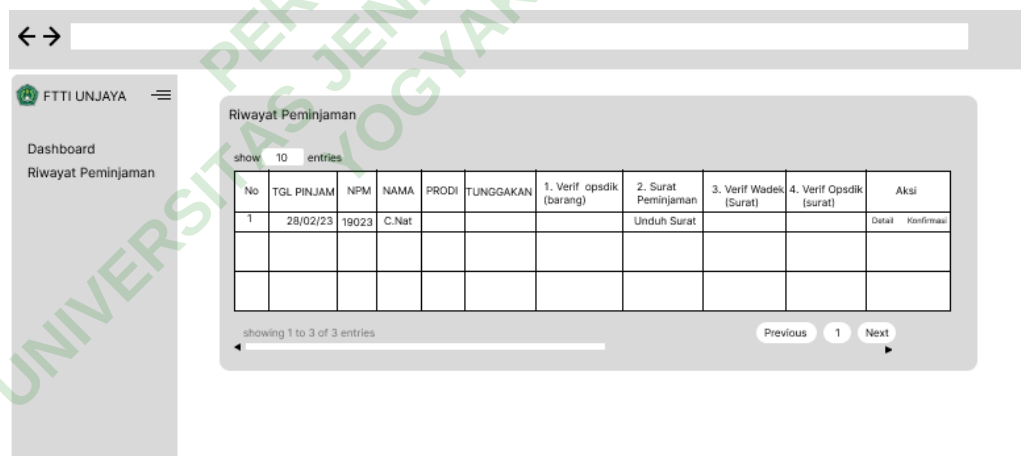
Desian halaman riwayat pengembalian terdapat tabel informasi peminjaman seperti pada desain halaman sebelumnya yang ditunjukkan pada Gambar 4.20.



Gambar 4.20 Desain mockup riwayat peminjaman

4.3.10 Halaman riwayat peminjaman (Wakil Dekan 2)

Desian halaman riwayat peminjaman terdapat tabel informasi peminjaman seperti pada desain halaman sebelumnya yang ditunjukkan pada Gambar 4.21.



Gambar 4.21 Desain mockup riwayat peminjaman

4.3.11 Halaman detail riwayat peminjaman (Wakil Dekan 2)

Desian halaman detail riwayat peminjaman terdapat tabel informasi peminjaman seperti pada desain halaman sebelumnya yang ditunjukkan pada Gambar 4.22.

Gambar 4.22 Desain mockup riwayat peminjaman

4.3.12 Halaman peminjaman (Mahasiswa, Dosen)

Pada halaman peminjaman sarana dan prasarana terdapat form yang harus diisi sesuai dengan kebutuhan pengguna, terdapat form pilih ruangan, jam peminjaman, jam akhir peminjaman, tanggal peminjaman, pilih alat, serta jumlah peminjaman yang ditunjukkan pada Gambar 4.23.

Gambar 4.23 Desain mockup peminjaman sarana dan prasarana

4.3.13 Halaman form pengembalian (Mahasiswa, Dosen)

Desian halaman form Pengembalian terdapat tabel informasi peminjaman seperti pada desain halaman sebelumnya yang ditunjukkan pada Gambar 4.24.

Gambar 4.24 Desain mockup form peminjaman

4.3.14 Halaman riwayat peminjaman (Mahasiswa, Dosen)

Desian halaman riwayat peminjaman terdapat tabel informasi peminjaman seperti pada desain halaman sebelumnya yang ditunjukkan pada Gambar 4.25.

No	TGL PINJAM	NPM	NAMA	PRODI	TUNGGAKAN	1. Verif opsdik (barang)	2. Surat Peminjaman	3. Verif Wadep (Surat)	4. Verif Opsdik (surat)	Aksi
1	28/02/23	19023	C.Nat		1		Unduh Surat			Detail Konfirmasi

Gambar 4.25 Desain mockup riwayat peminjaman

4.3.15 Halaman detail riwayat peminjaman (Mahasiswa, Dosen)

Desian halaman detail riwayat peminjaman terdapat tabel informasi peminjaman seperti pada desain halaman sebelumnya yang ditunjukkan pada Gambar 4.26.

Gambar 4.26 Desain mockup riwayat peminjaman

4.3.16 Halaman riwayat pengembalian (Mahasiswa, Dosen)

Desian halaman riwayat pengembalian terdapat tabel informasi peminjaman seperti pada desain halaman sebelumnya yang ditunjukkan pada Gambar 4.27.

No	TGL PINJAM	NPM	NAMA	PRODI	TUNGGAKAN	1. Verif opsdik (barang)	2. Surat Peminjaman	3. Verif Wadeti (Surat)	4. Verif Opsdik (surat)	Aksi
1	28/02/23	19023	C.Nat	IF	Tidak ada	Diterima	Unduh Surat	Diterima	Diterima	Detail

Gambar 4.27 Desain mockup riwayat peminjaman

4.4 CONSTRUCTION OF PROTOTYPE

Tahapan *construction of prototype* ini akan untuk pengujian yang akan menggunakan metode *Single Ease Question (SEQ)* adalah jenis pengujian yang memungkinkan menggunakan nilai suatu produk secara cepat dengan menyelesaikan tugas tertentu yang telah di ukur secara efektif menurut pengalaman pengguna. Pengujian ini juga bertujuan untuk memperoleh umpan balik dari responden guna melakukan. Metode

Single Ease Question (SEQ) digunakan dalam pengujian usability yang mengukur tingkat kesulitan yang dihadapi pengguna dalam menyelesaikan skenario yang telah diberikan. Hasil pengujian dan nilai SEQ kemudian ditampilkan dalam Tabel , pada tabel Pengujian *Single Ease Question (SEQ)* dilakukan setelah pengguna menyelesaikan skenario dengan memberikan penilaian pada skala likert dari 1-7.

1. Skala 1 : Sangat Sulit
2. Skala 2 : Sulit
3. Skala 3 : Cukup Sulit
4. Skala 4 : Netral
5. Skala 5 : Cukup Mudah
6. Skala 6 : Mudah
7. Skala 7 : Sangat Mudah

Berikut merupakan hasil dari pengujian *Single Ease Question (SEQ)* yang dapat dilihat pada tabel

Tabel 4.2 Hasil Rekapitulasi Admin

No.	Tugas	Nilai						
		1	2	3	4	5	6	7
1	Melakukan login							1
2	Melihat dashboard							1
3	Melihat riwayat peminjaman dan pengembalian							1
4	Melihat data pengguna							1

Tabel 4.4 Hasil Rekapitulasi Wakil Dekan 2

No.	Tugas	Nilai						
		1	2	3	4	5	6	7
1	Melakukan login							1
2	Melihat dashboard							1
3	Melihat data riwayat peminjaman							1
4	Melihat surat peminjaman							1
5	Melihat surat disposisi							1
6	Membuat Surat Disposisi Tahap 2						1	
7	Melihat detail peminjaman						1	
8	Menyetujui peminjaman							1
Jumlah							2	6
Total Suara		8						

Tabel 4.5 Hasil Rekapitulasi Pengguna

No.	Tugas	Nilai						
		1	2	3	4	5	6	7
1	Melakukan login							1
2	Melihat dashboard							1
3	Melihat riwayat peminjaman dan pengembalian							1
4	Melihat form peminjaman							1
5	Mengisi form peminjaman							1
6	Melihat form pengembalian							1

7	Mengisi form pengembalian							1
8	Melihat riwayat peminjaman							1
9	Unggah surat peminjaman							1
10	Melihat riwayat pengembalian						1	
Jumlah							1	9
Total Suara		10						

4.5 DEPLOYMENT (IMPLEMENTASI, DELIVERY DAN FEEDBACK)

Tahapan ini akan dilakukannya deployment terhadap kontruksi-kontruksi dari bahasa pemrograman python yang telah diterapkan agar dapat dipergunakan secara umum atau dapat diakses secara publik.

4.5.1 Perancangan Database

Selama perancangan *database*, penting untuk memilih tipe data yang sesuai untuk setiap kolom agar cocok dengan jenis data yang akan disimpan. Dengan desain database yang baik, sistem akan menjadi lebih mudah diakses, dikelola, dan dioptimalkan, yang pada akhirnya akan meningkatkan kinerja dan kendala yang menggunakan database tersebut.

1. Tabel *biodatas*

Tabel biodatas berfungsi untuk menyimpan detail dari users atau pengguna pada sistem yang terdiri dari id, *name*, posisi, organisasi yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 Tabel biodatas

Nama Kolom	Tipe Data	Default	Null	Lebar	Kegunaan
id	VARCHAR(255)	NULL	No	255	Identifikasi unik untuk setiap entri
name	VARCHAR(255)	NULL	No	255	Nama individu atau entitas yang bersangkutan
position	VARCHAR(255)	NULL	No	255	Posisi atau jabatan individu dalam organisasi
Organizer	VARCHAR(255)	NULL	No	255	Departemen atau organisasi terkait individu
User_id	VARCHAR(255)	NULL	No	255	Identifikasi unik pengguna yang terhubung dengan entri

2. Tabel bookedrooms

Tabel bookedrooms berfungsi untuk mengatur peminjaman pada sistem dengan struktur id, *timestamp*, *starttime*, *sudtime*, *date*, yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Tabel bookedrooms

Nama Kolom	Tipe Data	Default	Null	Lebar	Kegunaan
id	VARCHAR(255)	NULL	No	255	Identifikasi unik untuk setiap entri
timestamp	DATETIME	NULL	No	-	Waktu pencatatan entri
starttime	TIME	NULL	No	-	Waktu mulai kegiatan atau peminjaman
endtime	TIME	NULL	No	-	Waktu berakhir kegiatan atau peminjaman
date	DATE	NULL	No	-	Tanggal kegiatan atau peminjaman
disposisi	VARCHAR(255)	NULL	Yes	255	Disposisi terkait entri
surat	VARCHAR(255)	NULL	Yes	255	Lokasi file surat yang terkait
tunggakan	VARCHAR(255)	NULL	Yes	255	Status tunggakan terkait peminjaman

isUsed	VARCHAR(3)	NULL	No	3	Indikator apakah entri digunakan
verifikasi1	VARCHAR(255)	NULL	Yes	255	Status verifikasi pertama
komentar1	VARCHAR(255)	NULL	Yes	255	Komentar terkait verifikasi pertama
verifikasi2	VARCHAR(255)	NULL	Yes	255	Status verifikasi kedua
komentar2	VARCHAR(255)	NULL	Yes	255	Komnetar terkait verifikasi kedua
verifikasi3	VARCHAR(255)	NULL	Yes	255	Status verifikasi ketiga
komentar3	VARCHAR(255)	NUL	Yes	255	Komentar terkait verifikasi ketiga
user_id	VARCHAR(255)	NULL	No	255	Identifikasi unik pengguna yang terhubung dengan entri
room_id	VARCHAR(255)	NULL	No	255	Identifikasi unik ruangan yang terhubung degan entri

3. Tabel bookedtools

Tabel bookedtools berfungsi untuk meminjam alat pada sistem dengan struktur dari tabel yakni *id*, *quantity*, *bookedrooms_id* serta *tool_id* yang dapat dilihat pada Tabel 4.8.

Tabel 4.8 Tabel bookedtools

Tab Kolom	Nama	Tipe Data	Default	Null	Lebar	Kegunaan
id		VARCHAR (255)	NULL	No	255	Identifikasi unik untuk setiap entri
quantity		INT	NULL	No	-	Jumlah item yang terkait
bookedrooms_id		VARCHAR (255)	NULL	No	255	Identifikasi unik untuk setiap entri
Tool_id		VARCHAR(255)	NULL	No	255	Identifikasi unik peralatan

4. Tabel rooms

Tabel rooms berfungsi sebagai informasi detail ruangan dengan struktur yakni *id*, *name*, *capacity*, *status*, *default* yang dapat dilihat pada Tabel 4.9.

Tabel 4.9 Tabel rooms

Nama Kolom	Tipe Data	Default	Null	Lebar	Kegunaan
id	VARCHAR	NULL	NO	36	ID unik untuk setiap entri
name	VARCHAR	NULL	NO	255	Nama ruangan
capacity	INT	0	NO	-	Kapasitas ruangan
status	VARCHAR	NULL	NO	255	Status pemakaian ruangan
default	BOOLEAN	FALSE	NO	-	Menunjukkan apakah ruangan adalah default atau tidak

5. Tabel tools

Tabel tools berfungsi sebagai informasi alat dengan struktur yang mirip seperti tabel rooms. Berikut merupakan struktur tabel tools yang dapat dilihat pada Tabel 4.10.

Tabel 4.10 Tabel rooms

Nama Kolom	Tipe Data	Default	Null	Lebar	Kegunaan
id	VARCHAR	NULL	NO	36	ID unik untuk setiap entri
name	VARCHAR	NULL	NO	255	Nama ruangan
capacity	INT	0	NO	-	Kapasitas ruangan
status	VARCHAR	NULL	NO	255	Status pemakaian ruangan
dafult	BOOLEAN	NULL	YES	-	Menandakan apakah ini adalah pengaturan default atau tidak

6. Tabel users

Tabel Users berfungsi untuk mengatur users pada sistem yang terdiri dari informasi akun seperti *id*, *username*, *password*, dan *role* yang dapat dilihat pada Tabel 4.11.

Tabel 4.11 Tabel users

Nama Kolom	Tipe Data	Default	Null	Lebar	Kegunaan
id	VARCHAR	NULL	NO	36	ID unik untuk setiap entri
username	VARCHAR	NULL	NO	255	Nama Pengguna
password	VARCHAR	NULL	NO	255	Kata sandi pengguna yang telah di-hash
role	VARCHAR	NULL	NO	255	Peran pengguna (misalnya Dosen, Mahasiswa, Admin, Staff umum)

7. Tabel disposisi

Tabel disposisi berfungsi untuk menambahkan surat disposisi dan mengatur disposisi yang dapat dilihat pada Tabel 4.12

Tabel 4.12 Tabel disposisi

Nama Kolom	Tipe Data	Default	Null	Lebar	Kegunaan
id	VARCHAR	NULL	NO	150	ID unik untuk setiap entri
Nomor_disposel	VARCHAR	NULL	NO	255	Nomor disposisi
From_disposel	VARCHAR	NULL	NO	255	Sumber disposisi
Nomor_surat	VARCHAR	NULL	NO	255	Nomor surat
Tanggal_surat	DATE	NULL	YES	-	Tanggal surat
Perihal	VARCHAR	NULL	YES	255	Perihal surat
Lampiran	TINYINT	NULL	YES	1	Lampiran
Tanggal_diterima	DATE	NULL	YES	-	Tanggal diterima
Penerima	VARCHAR	-	NO	255	Penerima
Sifat	VARCHAR	NULL	YES	255	Sifat Disposisi
Klasifikasi	VARCHAR	NULL	YES	255	Klasifikasi
Ditunjukkan_wadek1	TINYINT	NULL	YES	1	Ditunjukkan wadek 1
Ditunjukkan_wadek2	TINYINT	NULL	YESS	1	Ditunjukkan wadek 2
Dittembuskan_ka_spm	TINYINT	NULL	YESS	1	Dittembuskan Ka. SPM
Dittembuskan_ka_prodi_ik	TINYINT	NULL	YESS	1	Dittembuskan ka prodi ik
Dittembuskan_ka_prodi_t	TINYINT	NULL	YESS	1	Dittembuskan ka prodi t
Dittembuskan_ka_prodi_teno	TINYINT	NULL	YESS	1	Dittembuskan ka prodi teno
Dittembuskan_ka_informatika	TINYINT	NULL	YESS	1	Dittembuskan ka prodi if
Dittembuskan_ka_gkm	TINYINT	NULL	YESS	1	Dittembuskan ka gkm

Ditembuskan_ka_kabag_uk	TINYINT	NULL	YESS	1	Ditembuskan ka kabag uk
Ditembuskan_ka_kabag_ak	TINYINT	NULL	YESS	1	Ditembuskan ka kabag ak
Ditembuskan_ka_lab	TINYINT	NULL	YESS	1	Ditembuskan ka lab
Ditembuskan_ka_bab	TINYINT	NULL	YESS	1	Ditembuskan ka bab
Ditembuskan_ka_ketgam	TINYINT	NULL	YESS	1	Ditembuskan ka ketgam
Disposel_ditindaklanjuti	TINYINT	NULL	YESS	1	Disposel tindak lanjut
Disposel_alap_ambuntuan	TINYINT	NULL	YESS	1	Disposel alap ambuntuan
Disposel_siap_menghadiri	TINYINT	NULL	YESS	1	Disposel siap hadir
Disposel_koordinasikan	TINYINT	NULL	YESS	1	Disposel koordinasikan
Disposel_wakil_dekan	TINYINT	NULL	YESS	1	Disposel wakil dekan
Disposel_review_feasibility	TINYINT	NULL	YESS	1	Disposel review feasibility
Disposel_perimbangkan	TINYINT	NULL	YESS	1	Disposel perimbangkan
Disposel_fasilitasi	TINYINT	NULL	YESS	1	Disposel fasilitasi
Catatan	VARCHAR	NULL	YESS	255	Catatan
Tanggal_ttd	DATE	NULL	YES	-	Tanggal ttd
File_ttd	VARCHAR	NULL	YES	255	File ttd
Nama_ttd	VARCHAR	NULL	YES	255	Nama ttd
Bookedrooms_id	VARCHAR	-	NO	150	Id bookedrooms

4.5.2 Implementasi desain antar muka

Tahapan implementasi desain antar muka sistem informasi peminjaman sarana dan prasarana.

1. Implementasi Halaman Login

Halaman login digunakan untuk proses pertama pengguna ketika menggunakan sistem. Pada Gambar 4.28 merupakan halaman pertama disaat sistem berhasil diakses. Halaman login ini digunakan oleh admin, mahasiswa, dosen, untuk dapat masuk ke dalam sistem dengan akun yang telah terdaftar.



Gambar 4.28 Implementasi halaman login

Berikut merupakan kode autentikasi yang digunakan untuk mengautentikasi pengguna saat login. Pesan kesalahan akan ditampilkan Ketika npm dan password login tidak sesuai.

```
@auth.route('/login', methods=['GET', 'POST'])
def login():
    try:
        if request.method == 'POST':
            username = request.form.get('username')
            password = request.form.get('password')

            user = User.query.filter_by(username=username).first()
            if user:
                if check_password_hash(user.password, password):
                    login_user(user)
                    if user.role == "admin" :
                        return redirect(url_for('admin.dashboard'))
                    elif user.role == "Mahasiswa" :
```

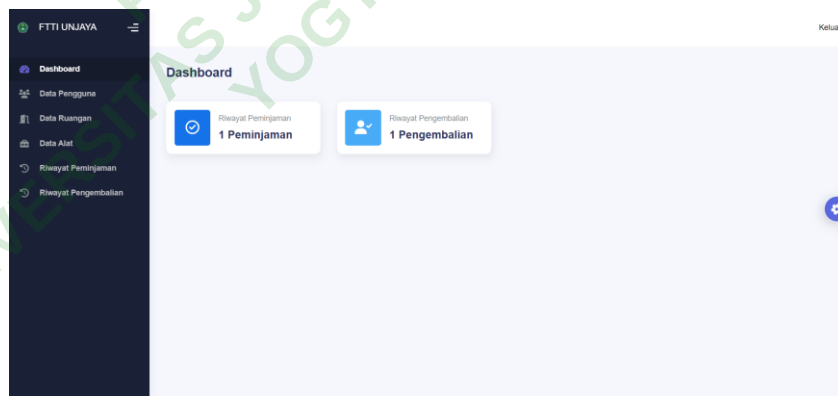
```

return
redirect(url_for('students.dashboard'))
elif user.role == "Dosen" and
user.profile_relationship.position == "Wakil Dekan 2":
return
redirect(url_for('wadek.dashboard'))
elif user.role == "Dosen" and
user.profile_relationship.position == "Dosen":
return
redirect(url_for('students.dashboard'))
flash('NPM atau Password salah.',
category='error')
return redirect(url_for('auth.login'))
return render_template('autentikasi/login.html',
user=current_user)
except Exception as e:
print(e)
abort(404)

```

2. Implementasi halaman dashboard admin

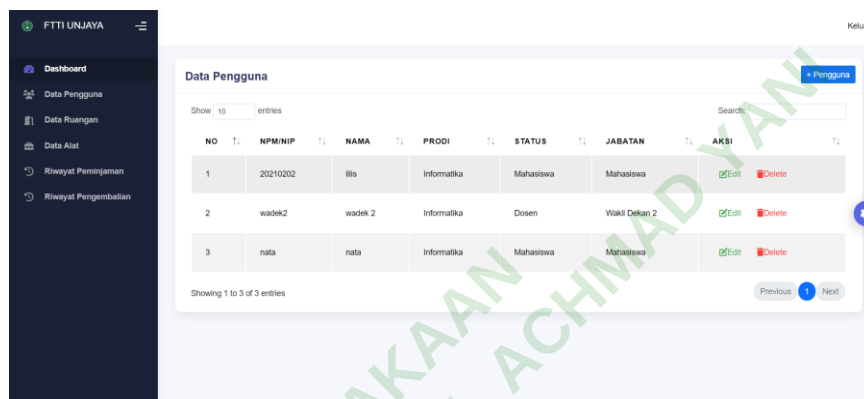
Halaman dashboard admin ditampilkan ketika admin berhasil melakukan proses login. Pada Gambar 4.29 terdapat informasi riwayat peminjaman serta pengembalian sarana dan prasarana. Terdapat angka yang akan bertambah ketika nanti akan bertambahnya peminjaman dan pengembalian sarana dan prasarana.



Gambar 4.29 Implementasi halaman dashboard admin

3. Implementasi halaman data pengguna

Pada halaman data pengguna ditampilkan ketika admin berhasil melakukan login. Pada Gambar 4.30 merupakan halaman data pengguna yang menampilkan keseluruhan data pengguna baik dari mahasiswa atau dosen. Halaman ini memiliki tabel yang terdiri nomor, npm/npp, nama, prodi, status, jabatan, aksi, selain itu juga terdapat tombol untuk melakukan edit data pengguna ataupun hapus data pengguna



Gambar 4.30 Implementasi halaman data pengguna

Berikut merupakan salah satu fungsi kode program untuk menambahkan data pengguna.

```
@admin.route('/add-user', methods=['GET', 'POST'])
@login_required
def add_user():
    if request.method == 'POST':
        nama = request.form.get('nama')
        role = request.form.get('role')
        prodi = request.form.get('prodi')
        jabatan = request.form.get('jabatan')
        username = request.form.get('username')
        password = request.form.get('password')
        if not all([nama, role, prodi, jabatan, username,
password]):
            flash('Semua bidang harus diisi.', 'error')
            return redirect(url_for('admin.user'))
        existing_user =
User.query.filter_by(username=username).first()
        if existing_user:
            flash('NPM sudah terdaftar. Silakan gunakan NPM lain.',
category='error')
            return redirect(url_for('admin.user'))
        else :
            hashed_password = generate_password_hash(password,
method='pbkdf2:sha256')
```

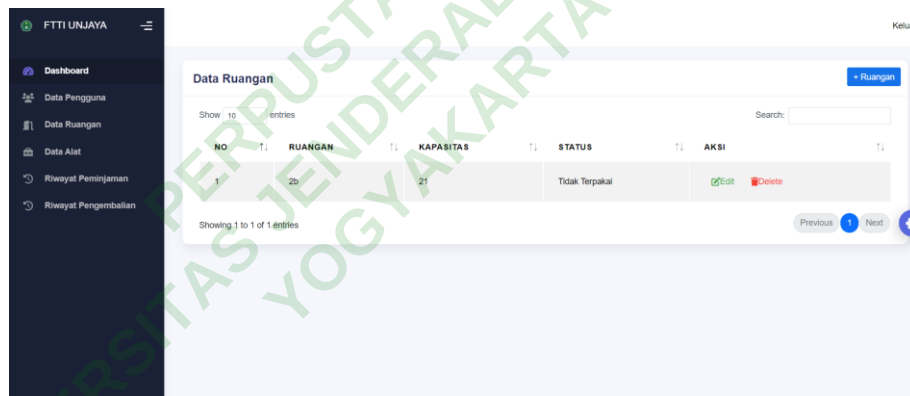
```

        new_user = User(username=username,
password=hashed_password, role=role)
        db.session.add(new_user)
        db.session.commit()
        new_profile = Biodata(name=nama, position=jabatan,
organizer=prodi,user_id=new_user.id)
        db.session.add(new_profile)
        db.session.commit()
        flash('Akun berhasil ditambahkan!', 'success')
        return redirect(url_for('admin.user'))
    return render_template('admin/user.html')

```

4. Implementasi halaman data ruangan

Halaman data ruangan ditampilkan ketika admin sudah melakukan login. Pada Gambar 4.31 merupakan halaman data ruangan yang menampilkan seluruh data ruangan yang tersedia ataupun tidak tersedia serta memiliki tabel yang terdiri dari nomor, ruangan, kapasitas, status, serta aksi yang dapat untuk mengedit dan menghapus data ruangan yang telah.



Gambar 4.31 Implementasi halaman data ruangan

Berikut merupakan fungsi kode program untuk menambahkan data ruangan.

```

@admin.route('/ruangan', methods=['GET', 'POST'])
@login_required
def ruangan():
    room = Room.query.all()
    return render_template('admin/ruangan.html', room=room)
@admin.route('/add-room', methods=['GET', 'POST'])
@login_required
def add_room():
    if request.method == 'POST':
        name = request.form.get('name')
        capacity = request.form.get('capacity')
        if not all([name, capacity]):

```

```

        flash('Semua bidang harus diisi.', 'error')
        return redirect(url_for('admin.ruangan'))
    existing = Room.query.filter_by(name=name).first()
    if existing:
        flash('Ruangan sudah terdaftar.', category='error')
        return redirect(url_for('admin.ruangan'))
    else :
        new = Room(name=name, capacity=capacity, status="Tidak
Terpakai")
        db.session.add(new)
        db.session.commit()
        flash('Akun berhasil ditambahkan!', 'success')
        return redirect(url_for('admin.ruangan'))
    return render_template('admin/user.html')

```

5. Implementasi halaman data alat

Halaman data alat ditampilkan ketika admin sudah melakukan login. Pada Gambar 4.32 merupakan halaman data alat yang menampilkan seluruh data ruangan yang tersedia ataupun tidak tersedia serta memiliki tabel yang terdiri dari nomor, nama alat, dipinjam, stok total, serta aksi yang dapat untuk mengedit dan menghapus data ruangan yang telah.

NO	NAMA ALAT	DIPINJAM	STOK TOTAL	AKSI
1	Kamera	0 0	10	Edit Delete

Gambar 4.32 Implementasi halaman data alat

Berikut merupakan fungsi kode program untuk menambahkan data alat.

```

@admin.route('/alat-alat', methods=['GET', 'POST'])
@login_required
def alat():
    data = Tool.query.all()
    results = []
    for tool in data:
        tool_booked = []
        for booked_tool in tool.tool_booked:
            if booked_tool.booked_relationship.tunggakan == "Ada":

```

```

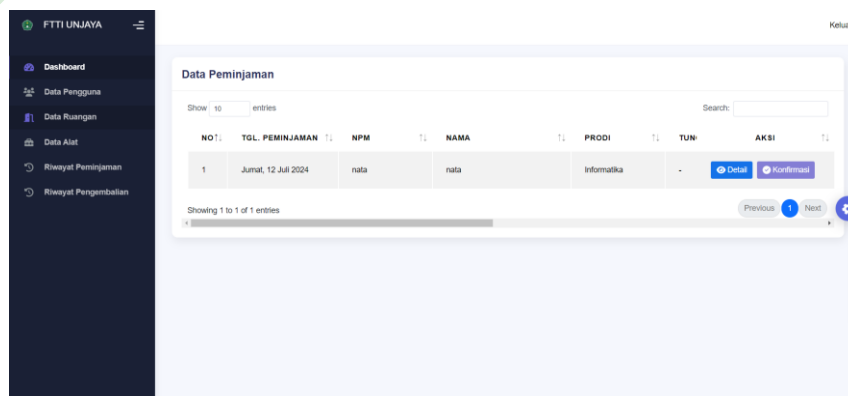
tool_booked.append({'quantity':
booked_tool.quantity})
else:
    tool_booked.append({'quantity': "0"})

results.append({
    'id': tool.id,
    'name': tool.name,
    'capacity': tool.capacity,
    'tool_booked': tool_booked,
})
return render_template('admin/alat.html', data=results)
@admin.route('/add-alat', methods=['GET', 'POST'])
@login_required
def add_alat():
    if request.method == 'POST':
        name = request.form.get('name')
        capacity = request.form.get('capacity')
        if not all([name, capacity]):
            flash('Semua bidang harus diisi.', 'error')
            return redirect(url_for('admin.alat'))
        existing = Tool.query.filter_by(name=name).first()
        if existing:
            flash('Alat sudah terdaftar.', category='error')
            return redirect(url_for('admin.alat'))
        else :
            new = Tool(name=name, capacity=capacity)
            db.session.add(new)
            db.session.commit()
            flash('Data berhasil ditambahkan!', 'success')
            return redirect(url_for('admin.alat'))
    return render_template('admin/alat.html')

```

6. Implementasi halaman riwayat peminjaman

Halaman riwayat peminjaman ini harus diakses admin ketika sudah melakukan login. Pada Gambar 4.33 ini terdapat informasi riwayat peminjaman.



Gambar 4.33 Implementasi halaman riwayat peminjaman

Berikut merupakan kode program untuk menampilkan halaman riwayat peminjaman.

```
@admin.route('/data-peminjaman', methods=['GET', 'POST'])
@login_required
def data_peminjaman():
    booked = BookedRoom.query.filter_by(isUsed="Yes").all()
    results = []
    for data in booked:
        user = data.user_relationship
        biodata = user.profile_relationship
        room = data.room_relationship
        results.append({
            'id': data.id,
            'timestamp': day_and_date(data.timestamp),
            'username': user.username,
            'role': user.role,
            'tunggakan': data.tunggakan,
            'surat': data.surat,
            'verifikasi1': data.verifikasi1,
            'verifikasi2': data.verifikasi2,
            'verifikasi3': data.verifikasi3,
            'starttime': data.starttime,
            'endtime': data.endtime,
            'date': day_and_date(data.date),
            'name': biodata.name if biodata else '',
            'organizer': biodata.organizer if biodata else '',
            'room': room.name,
            'room_id': room.id,
            'tools': [
                {
                    'name': booked_tool.tool_relationship.name,
                    'capacity':
booked_tool.tool_relationship.capacity
                }
                for booked_tool in data.bookedtool_relationship
            ]
        })
    return render_template('admin/datapeminjaman.html',
results=results)
```

7. Implementasi halaman riwayat pengembalian

Halaman riwayat pengembalian ini dapat diakses ketika admin sudah melakukan login pada sistem. Gambar 4.34 menunjukkan tabel yang berisikan informasi pengembalian atau riwayat yang telah digunakan ataupun dalam proses.

NO	TGL PEMINJAMAN	NPM	NAMA	PRODI	TUN	AKSI
1	Jumat, 12 Juli 2024	nata	nata	Informatika	Ada	Detail Konfirmasi
2	Jumat, 05 Juli 2024	nata	nata	Informatika	Tidak	Detail

Gambar 4.34 Implementasi halaman riwayat pengembalian

Berikut merupakan fungsi kode program untuk menampilkan halaman riwayat pengembalian.

```
@admin.route('/data-pengembalian', methods=['GET', 'POST'])
@login_required
def data_pengembalian():
    booked = BookedRoom.query.filter_by(isUsed="No").all()
    results = []
    for data in booked:
        user = data.user_relationship
        biodata = user.profile_relationship
        room = data.room_relationship
        results.append({
            'id': data.id,
            'timestamp': day_and_date(data.timestamp),
            'username': user.username,
            'role': user.role,
            'tunggakan': data.tunggakan,
            'surat': data.surat,
            'verifikasi1': data.verifikasi1,
            'verifikasi2': data.verifikasi2,
            'verifikasi3': data.verifikasi3,
            'starttime': data.starttime,
            'endtime': data.endtime,
            'date': day_and_date(data.date),
            'name': biodata.name if biodata else '',
            'organizer': biodata.organizer if biodata else '',
            'room': room.name,
            'room_id': room.id,
            'tools': [
```

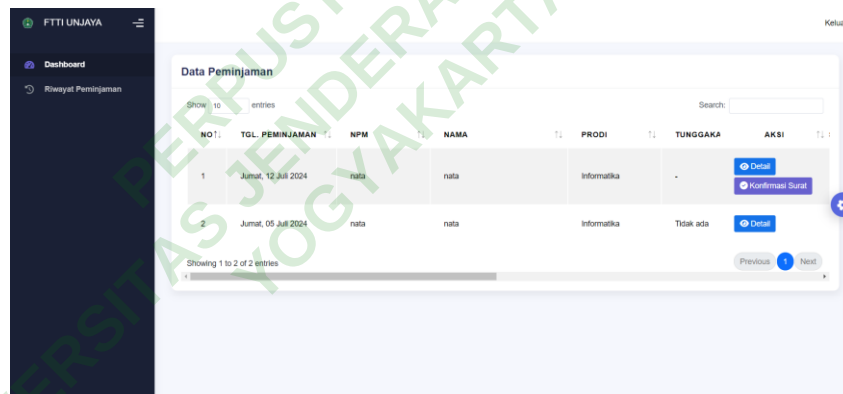
```

        {
            'name': booked_tool.tool_relationship.name,
            'capacity':
booked_tool.tool_relationship.capacity
        }
        for booked_tool in data.bookedtool_relationship
    ]
    })
    return render_template('admin/datapengembalian.html',
results=results)

```

8. Implementasi halaman riwayat peminjaman wakil dekan 2

Halaman riwayat peminjaman dapat diakses oleh wakil dekan 2 dan tidak dapat diakses petugas opsdik atau admin. Pada Gambar 4.35 menunjukkan bahwa informasi data peminjaman dari pengguna dan petugas opsdik ditampilkan. Aktivitas wakil dekan 2 pada sistem ini adalah menyetujui peminjaman serta dapat mengetahui detail peminjaman dan informasi surat yang telah diajukan oleh peminjam.



Gambar 4.35 Implementasi halaman riwayat peminjaman wakil dekan 2

Berikut merupakan fungsi kode program untuk menampilkan halaman riwayat peminjaman wakil dekan 2.

```

@wadek.route('/peminjaman-data', methods=['GET', 'POST'])
def data_peminjaman():
    booked = BookedRoom.query.all()
    results = []
    for data in booked:
        user = data.user_relationship
        biodata = user.profile_relationship
        room = data.room_relationship
        results.append({

```

```

'id': data.id,
'timestamp': day_and_date(data.timestamp),
'username': user.username,
'role': user.role,
'surat': data.surat,
'tunggakan': data.tunggakan,
'starttime': data.starttime,
'endtime': data.endtime,
'verifikasi1': data.verifikasi1,
'verifikasi2': data.verifikasi2,
'verifikasi3': data.verifikasi3,
'date': day_and_date(data.date),
'name': biodata.name if biodata else '',
'organizer': biodata.organizer if biodata else '',
'room' : room.name,
'tools' : [
    {
        'name': booked_tool.tool_relationship.name,
        'capacity':
booked_tool.tool_relationship.capacity
    }
    for booked_tool in data.bookedtool_relationship
]
})
return render_template('wadek_2/datapeminjaman.html',
results=results)

```

9. Implementasi halaman riwayat peminjaman staff umum

Halaman riwayat peminjaman staff umum memberikan informasi detail peminjaman dari surat hingga detail peminjam dengan fitur utama yaitu membuat surat diposisi setelah menyetujui peminjaman pada surat peminjaman yang dapat dilihat pada Gambar 4.36.

No	Tanggal	ID User	Nama User	Departemen	Ruang	Status	Aksi
1	Kamis, 18 Juli 2024	tsr123	Hetai	Informatika	Ada	Unduh Surat ✓	Unduh
2	Jumat, 19 Juli 2024	bagus123	bagus	Informatika	-	Unduh Surat ✓	Unduh
3	Kamis, 25 Juli 2024	asd	asd	Informatika	-	Unduh Surat ✓	Unduh
4	Sabtu, 20 Juli 2024	lily1	lily	Informatika	-	Unduh Surat ✓	Unduh
5	Kamis, 18 Juli 2024	putri	putri	Informatika	Ada	Unduh Surat ✓	Unduh
6	Jumat, 19 Juli 2024	staffumum	staff umum	Informatika	-	Unduh Surat ✓	Unduh
7	Sabtu, 20 Juli 2024	tsr1	tsr1	Informatika	Ada	Unduh Surat ✓	Unduh
8	Senin, 30 Juli 2024	apriadi	apriadi	Informatika	-	Unduh Surat ✓	Unduh
9	Kamis, 25 Juli 2024	siska123	siska	Teknik Industri	-	Unduh Surat ✓	Unduh
10	Kamis, 18 Juli 2024	nata1	cynthia nata videsasti	Informatika	-	Unduh Surat ✓	Unduh

Gambar 4.36 Implementasi halaman riwayat peminjaman staff umum

10. Implementasi surat disposisi staff umum dan wakil dekan 2

Halaman form surat disposisi dapat diakses oleh wakil dekan 2 beserta staff umum yang nantinya akan diberikan kepada pengguna selaku peminjam dapat dilihat pada Gambar 3.7.

The screenshot shows a web browser window displaying a form titled 'SURAT DISPOSISI' from Universitas Kartika Eka Paksi. The form is divided into several sections: 'Form Disposisi' with fields for Name, ID, Email, Password, and a checkbox for 'Lampiran'; 'Form Disposisi' with checkboxes for 'Surat Disposisi', 'Surat Disposisi', and 'Surat Disposisi'; 'Dikirimkan Kepada' with radio buttons for 'Kepada', 'Kepada', and 'Kepada'; and 'Klasifikasi' with checkboxes for 'Surat Disposisi', 'Surat Disposisi', and 'Surat Disposisi'. The form is watermarked with 'UNIVERSITAS KARTIKA EKA PAKSI' and 'FAKULTAS TEKNIK DAN TEKNOLOGI INFORMASI'.

Gambar 4.37 Implementasi form surat disposisi

Berikut potongan kode dari halaman form surat disposisi yang dapat diakses oleh staff umum beserta wakil dekan 2.

```
@wadek.route('/add_disposisi', methods=['POST'])
def add_disposisi():
    if request.method == 'POST':
        id = request.form['id']
        nomor_disposisi = request.form['nomor_disposisi']
        from_disposisi = request.form['from_disposisi']
        nomor_surat = request.form['nomor_surat']
        tanggal_surat =
datetime.strptime(request.form['tanggal_surat'], '%Y-%m-%d') if
request.form['tanggal_surat'] else None
        perihal = request.form['perihal']
        lampiran = True if request.form.get('lampiran') == 'Ada'
        else False
        tanggal_diterima =
datetime.strptime(request.form['tanggal_diterima'], '%Y-%m-%d') if
request.form['tanggal_diterima'] else None
        penerima = request.form['penerima']
        sifat = request.form.get('sifat')
        klasifikasi = request.form.get('klasifikasi')
```

pada potongan kode diatas merupakan endpoint dari penambahan disposisi dari staff umum beserta wakil dekan 2 untuk diberikan kepada peminjam.

11. Implementasi halaman form peminjaman sarana dan prasarana

Halaman form peminjaman sarana dan prasarana dapat diakses pengguna (mahasiswa / dosen) ketika sudah melakukan login. Pada Gambar 4.38 tersebut menampilkan informasi atau form peminjaman sarana dan prasarana.

Gambar 4.38 Implementasi halaman form peminjaman sarpras

Berikut merupakan fungsi kode program untuk menampilkan form peminjaman sarana dan prasarana.

```
@students.route('/form-peminjaman', methods=['GET', 'POST'])
@login_required
def form_peminjaman():
    rooms = Room.query.filter_by().all()
    data = Tool.query.all()
    tools = []
    for tool in data:
        total_booked = sum(int(booked_tool.quantity) for
booked_tool in tool.tool_booked if
booked_tool.booked_relationship.tunggakan == "Ada")
        remaining_stock = int(tool.capacity) - total_booked
        tools.append({
            'id': tool.id,
            'name': tool.name,
            'capacity': tool.capacity,
            'remaining_stock': remaining_stock,
        })
    isUsed = BookedRoom.query.filter_by(isUsed='Yes', tunggakan="-", user_id=current_user.id).first()
    isUsed1 = BookedRoom.query.filter_by(isUsed='Yes', tunggakan="Ada", user_id=current_user.id).first()
    isUsed2 = BookedRoom.query.filter_by(isUsed='No', tunggakan="Ada", user_id=current_user.id).first()
    return render_template('mahasiswa/peminjaman.html',
rooms=rooms, tools=tools, isUsed=isUsed, isUsed1=isUsed1,
isUsed2=isUsed2)
```

12. Implementasi halaman form pengembalian sarana dan prasarana

Halaman form pengembalian sarana dan prasarana dapat diakses pengguna (mahasiswa / dosen) ketika sudah melakukan login. Pada Gambar 4.39 tersebut menampilkan informasi atau form pengembalian sarana dan prasarana.

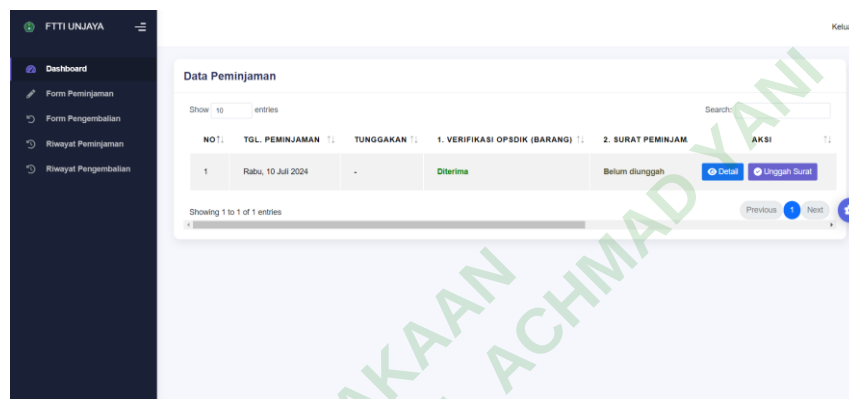
Gambar 4.39 Implementasi halaman form pengembalian sarpras

Berikut merupakan fungsi kode program untuk menampilkan form peminjaman sarana dan prasarana.

```
@students.route('/form-pengembalian', methods=['GET', 'POST'])
@login_required
def form_pengembalian():
    booked_room_with_tunggakan =
    BookedRoom.query.filter_by(isUsed="Yes", tunggakan="Ada",
    verifikasi3="Diterima", user_id=current_user.id).first()
    booked_room1 = BookedRoom.query.filter_by(isUsed="No",
    tunggakan="Ada", verifikasi3="Diterima",
    user_id=current_user.id).first()
    booked_room2 = BookedRoom.query.filter_by(isUsed="Yes",
    tunggakan="Tidak ada", user_id=current_user.id).first()
    booked_room3 = BookedRoom.query.filter_by(isUsed="No",
    tunggakan="Tidak ada", user_id=current_user.id).first()
    booked_tools = []
    tools = []
    if booked_room_with_tunggakan:
        booked_tools =
        BookedTool.query.filter_by(bookedrooms_id=booked_room_with_tungga
        kan.id).all()
        tool_ids = [booked_tool.tool_id for booked_tool in
        booked_tools]
        tools = Tool.query.filter(Tool.id.in_(tool_ids)).all()
```

13. Implementasi halaman Riwayat peminjaman

Halaman riwayat pengembalian dapat diakses ketika mahasiswa atau dosen telah melakukan login. Pada Gambar 4.40 menampilkan informasi riwayat peminjaman yang telah dilakukan. Gambar tersebut menampilkan fitur yaitu detail untuk melakukan cek peminjaman sarana dan prasarana serta unggah surat sebagai syarat wajib ketika ingin melakukan peminjaman sarana dan prasarana.



Gambar 4.40 Implementasi halaman riwayat peminjaman

Berikut merupakan fungsi kode program untuk menampilkan form peminjaman sarana dan prasarana.

```
@students.route('/peminjaman-saya', methods=['GET', 'POST'])
@login_required
def data_peminjaman():
    booked = BookedRoom.query.filter_by(user_id=current_user.id,
isUsed="Yes").all()
    results = []
    for data in booked:
        user = data.user_relationship
        biodata = user.profile_relationship
        room = data.room_relationship
        results.append({
            'id': data.id,
            'timestamp': day_and_date(data.timestamp),
            'username': user.username,
            'role': user.role,
            'surat': data.surat,
            'tunggakan': data.tunggakan,
            'verifikasi1': data.verifikasi1,
            'verifikasi2': data.verifikasi2,
            'verifikasi3': data.verifikasi3,
            'starttime': data.starttime,
            'endtime': data.endtime,
            'date': day_and_date(data.date),
```

```

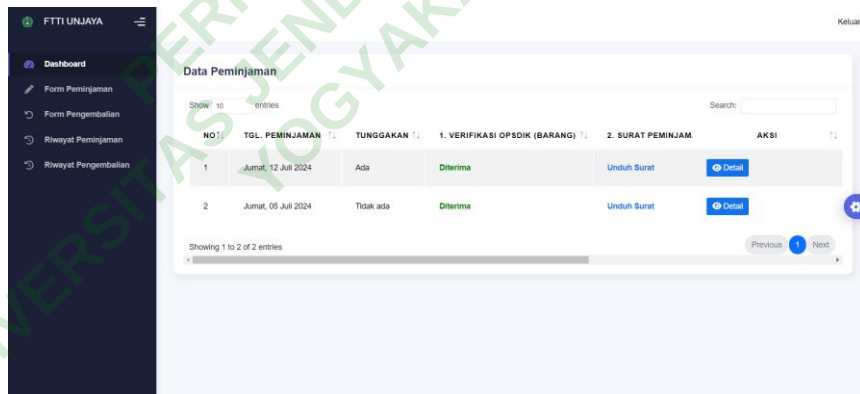
'name': biodata.name if biodata else '',
'organizer': biodata.organizer if biodata else '',
'room' : room.name,
'tools' : [
    {
        'name': booked_tool.tool_relationship.name,
        'capacity':
booked_tool.tool_relationship.capacity
    }
    for booked_tool in data.bookedtool_relationship
]
})

return render_template('mahasiswa/datapeminjaman.html',
results=results)

```

14. Implementasi halaman riwayat pengembalian

Halaman riwayat pengembalian dapat diakses ketika mahasiswa atau dosen sebagai pengguna telah melakukan login. Pada Gambar 4.41 tersebut menampilkan hasil riwayat pengembalian sarana dan prasarana yang telah di pinjam serta memiliki fitur aksi yakni untuk mengetahui detail peminjaman sarana dan prasarana sebelumnya.



Gambar 4.41 Implementasi halaman riwayat pengembalian

Berikut merupakan fungsi kode program untuk menampilkan form peminjaman sarana dan prasarana.

```

@students.route('/data-pengembalian-mahasiswa', methods=['GET',
'POST'])
@login_required
def data_pengembalian():
    booked = BookedRoom.query.filter_by(isUsed="No",
user_id=current_user.id).all()

```

```

results = []
for data in booked:
    user = data.user_relationship
    biodata = user.profile_relationship
    room = data.room_relationship
    results.append({
        'id': data.id,
        'timestamp': day_and_date(data.timestamp),
        'username': user.username,
        'role': user.role,
        'tunggakan': data.tunggakan,
        'surat': data.surat,
        'verifikasi1': data.verifikasi1,
        'verifikasi2': data.verifikasi2,
        'verifikasi3': data.verifikasi3,
        'starttime': data.starttime,
        'endtime': data.endtime,
        'date': day_and_date(data.date),
        'name': biodata.name if biodata else '',
        'organizer': biodata.organizer if biodata else '',
        'room' : room.name,
        'tools' : [
            {
                'name': booked_tool.tool_relationship.name,
                'capacity':
booked_tool.tool_relationship.capacity
            }
        ]
    })
    for booked_tool in data.bookedtool_relationship
return render_template('mahasiswa/datapengembalian.html',
results=results)

```

4.5.3 Delivery

Pada tahapan *delivery* atau pengiriman ini merupakan tahapan yang akan diuji oleh pakar, admin beserta pengguna. Pengujian ini memiliki dua tahap yaitu, tahap satu ialah *usability testing* dan tahap ke dua *black box testing*. Berikut tahapan dari pengujian sistem:

1. Tahap satu *usability testing*

Pada tahapan ini memberikan informasi kepada pakar, admin, beserta pengguna untuk melakukan *testing* pada *prototype* figma dengan bantuan *tools* maze. *Tools* maze merupakan tools yang bisa digunakan secara *online* untuk melakukan tahapan *usability testing* berikut merupakan hasil dari *usability testing* menggunakan *tools* maze :

a. Rekap hasil pengujian *usability testing*

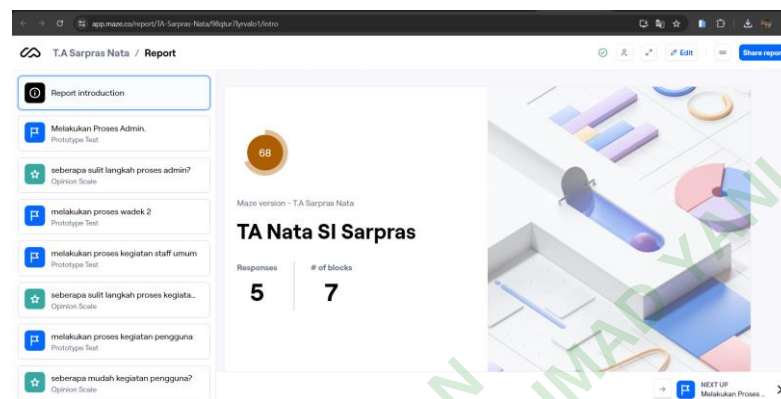
Pada Gambar 4.42 tersebut menjelaskan hasil dari *usability testing* yang dilakukan secara remote atau online menggunakan *tools* maze dengan hasil *direct success* ialah 100% kepada 5 responden dengan rata-rata waktu setiap block penguji ialah 25 detik.



Gambar 4.42 Implementasi halaman riwayat pengembalian

b. Hasil *score usability testing*

Pada Gambar 4.43 menjelaskan hasil skor dari *usability testing* sistem peminjaman sarana dan prasarana dengan 5 responden yaitu Pakar 2, Admin 1, Staff Umum 1, dan Pengguna 1.



Gambar 4.43 *Score usability testing*

2. Tahap dua *black box testing*

Black box testing merupakan pengujian tahap ke 2 dengan uji fungsional pada sistem. Berbeda dengan tahap 1 yang berfokus dengan pengujian *prototype* pada tahapan pengujian ini akan fokus kepada sistem yang sudah jadi. Berdasarkan pengujian yang dilakukan, diperoleh hasil pengujian yang selengkapnya dapat dilihat di bagian lampiran serta dapat disimpulkan bahwa fitur pada sistem berjalan dengan lancar sesuai dengan tujuan.

4.5.4 *Feedback*

Berdasarkan hasil *usability testing* dengan maze, sistem mendapatkan skor *usability* sebesar 68 dari 100 (5 Responden). Beberapa pengguna mengalami kesulitan menemukan tombol pada *prototype* saat dilakukannya uji test menggunakan tools maze dan membutuhkan waktu sekitar kurang lebih 30 detik. Dengan hasil *feedback* yakni pengeditan tugas, validasi input dan peningkatan desain antarmuka. Serta untuk pengujian *black box testing* menunjukkan bahwa fungsionalitas dasar berfungsi dengan baik sesuai ekspektasi. Validasi input berfungsi baik. Dengan beberapa perbaikan desain antarmuka dan menambahkan fitur yang diusulkan