PENGEMBANGAN APLIKASI PEMINJAMAN RUANGAN BERBASIS WEB DI BANDUNG CREATIVE HUB

TUGAS AKHIR

Disusun sebagai salah satu syarat untuk kelulusan
Program Strata 1, Program Studi Teknik Informatika,
Universitas Pasundan Bandung

oleh :

Yudha Permana
Nrp. 20.304.0101

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PASUNDAN BANDUNG
DESEMBER 2024

LEMBAR PENGESAHAN

LAPORAN TUGAS AKHIR

Telah disetujui dan disahkan Laporan Tugas Akhir, dari :

Nama : Yudha Permana
Nrp : 20.304.0101

Dengan Judul :

PENGEMBANGAN APLIKASI PEMINJAMAN RUANGAN BERBASIS WEB DI BANDUNG CREATIVE HUB

Bandung, 10 Desember 2024

Menyetujui,

Pembimbing Utama

(Ade Sukendar, S.T., M.T.)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa :

1. Tugas akhir ini adalah benar-benar asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Pasundan Bandung maupun di Perguruan Tinggi lainnya.
2. Tugas akhir ini merupakan gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing.
3. Dalam tugas akhir ini tidak terdapat karya atau pendapat orang lain, kecuali bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan karya ilmuah, serta disebutkan dalam Daftar Pustaka pada tugas akhir ini.
4. Kakas, perangkat lunak, dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung.

Apabila di kemudian hari ditemukan seluruh atau sebagian laporan tugas akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sangsi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan, serta perundang-undangan lainnya.

Bandung, 10 Desember 2024

Yang membuat pernyataan,

(Yudha Permana)

NRP. 20.304.0101

ABSTRAK

Bandung Creative Hub merupakan pusat kegiatan kreatif di Kota Bandung yang menyediakan berbagai ruangan untuk mendukung kegiatan kreatif masyarakat. Namun, sistem peminjaman pada saat ini masih menghadapi beberapa kendala, seperti pengajuan kegiatan yang memakan waktu karena harus dilakukan secara langsung. Selain itu, berkas pengajuan masih menggunakan media kertas, yang menyulitkan petugas dalam mencari berkas lama yang diperlukan untuk rekap dan penyusunan laporan, serta potensi bentrok dalam penjadwalan oleh petugas.

Penelitian ini bertujuan mengembangkan aplikasi peminjaman ruangan berbasis *Web*. Aplikasi yang akan dibangun ini menggunakan teknologi *React.JS* dan *Express.JS* dengan database MySQL. Metode yang digunakan pada pengembangan aplikasi ini menggunakan metode *Waterfall* dengan alat bantu Unified Modeling Language (UML) untuk menggambarkan aplikasi yang dibangun. Beberapa fitur utama pada aplikasi ini yaitu pendaftaran dan verifikasi peminjam, pengelolaan data ruangan, pengajuan dan verifikasi peminjaman ruangan, pengelolaan jadwal dan pembuatan laporan.

Hasil penelitian ini adalah aplikasi berbasis *Web* yang dapat membantu petugas dalam mengelola data pengajuan ruangan dan mempermudah peminjam untuk melakukan peminjaman ruangan. Dengan demikian, aplikasi ini dapat membantu operasional Bandung Creative Hub dalam menyediakan layanan peminjaman yang lebih baik dan terorganisir.

Kata Kunci : Peminjaman Ruangan, Aplikasi Berbasis *Web*, *Waterfall*

ABSTRACT

Bandung Creative Hub is a creative activity center in Bandung City that provides various rooms to support creative activities of the community. However, the current loan system still faces several obstacles, such as time-consuming activity submissions because they must be done in person. In addition, the application files still use paper media, which makes it difficult for officers to find old files needed for recapitulation and report preparation, as well as potential clashes in scheduling by officers.

This study aims to develop a Web-based room loan application. The application to be built uses React.JS and Express.JS technology with a MySQL database. The method used in developing this application uses the Waterfall method with the Unified Modeling Language (UML) tool to describe the application being built. Some of the main features of this application are borrower registration and verification, room data management, room loan submission and verification, schedule management and report generation.

The results of this study are a Web-based application that can help officers manage room submission data and make it easier for borrowers to borrow rooms. Thus, this application can help Bandung Creative Hub operations in providing better and more organized loan services.

Keywords: Room Lending, Web-Based Application, Waterfall

KATA PENGATAR

Segala puji bagi Allah SWT yang telah memberikan rahmat dan karunia-Nya, sehingga penulis dapat membuat dan menyelesaikan Laporan Tugas Akhir dengan judul "Pengembangan Aplikasi Peminjaman Ruangan Berbasis *Web* Di Bandung Creative Hub".

Adapun penulisan Laporan Tugas Akhir ini adalah bertujuan untuk menjadi salah satu syarat keikutsertaan mata kuliah TUGAS AKHIR (IF21W0805), di Program Studi Teknik Informatika Universitas Pasundan.

Dengan dukungan dan bantuan dari berbagai pihak penulis dapat mewujudkan dan menyelesaikan Tugas Akhir ini. Maka pada kesempatan kali ini penulis ingin menyampaikan rasa terimakasih atas segala dukungan dan bantuan yang penulis terima sehingga penulis pun dapat menyelesaikan laporan ini, kepada:

1. Pembimbing, yaitu Bapak Ade Sukendar, S.T., M.T.
2. Kedua orang tua dan keluarga besar yang selalu memberikan doa dan dukungan terbaik dalam pelaksanaan kuliah sampai dengan Tugas Akhir ini.
3. Koordinator Tugas Akhir dan Ketua Program Studi serta seluruh civitas akademika Teknik Informatika Universitas Pasundan Bandung yang telah memberikan bekal ilmu selama penulisan Proposal Tugas Akhir.
4. Teman-teman seperjuangan Teknik Informatika UNPAS angkatan 2020 dan teman- teman lainnya yang tidak bisa disebutkan satu persatu.

Penulis pun menyadari masih banyak kekurangan dalam penulisan Proposal Tugas Akhir ini oleh karena itu penulis mengharapkan kritik dan saran yang membangun dari semua pihak untuk perbaikan dimasa yang akan datang. Semoga penulisan Laporan Tugas Akhir ini dapat bermanfaat bagi penulis dan perkembangan ilmu teknologi yang akan datang.

Bandung, 10 Desember 2024

Penulis

DAFTAR ISI

ABSTRAK i
ABSTRACT ii
KATA PENGATAR iii
DAFTAR ISI iv
DAFTAR ISTILAH vi
DAFTAR TABEL vii
DAFTAR GAMBAR x
DAFTAR LAMPIRAN xi
DAFTAR SIMBOL xii
BAB 1 PENDAHULUAN 1-1
1.1. Latar Belakang 1-1
1.2. Identifikasi Masalah 1-2
1.3. Tujuan Tugas Akhir 1-2
1.4. Lingkup Tugas Akhir 1-2
1.5. Metodologi Pengerjaan Tugas Akhir 1-3
1.6. Sistematika Penulisan 1-3
BAB 2 LANDASAN TEORI DAN PENELITIAN TERDAHULU 2-1
2.1. Teori Yang Digunakan 2-1
2.1.1. Pengembangan Aplikasi 2-1
2.1.2. Metode Waterfall 2-1
2.1.3. Aplikasi 2-3
2.1.4. Website 2-3
2.1.5. Aplikasi Web 2-4
2.1.6. Arsitektur Aplikasi Web 2-4
2.1.7. Peminjaman Ruangan 2-5
2.1.8. React.js 2-6
2.1.9. JSX 2-7
2.1.10. Node.js 2-7
2.1.11. Express.js 2-8
2.1.12. Application Programming Interface (API) 2-8
2.1.13. REST 2-8
2.1.14. UML (Unified Modeling Language) 2-8
2.1.15. Unit Testing 2-9
2.2. Penelitian Terdahulu 2-9
BAB 3 SKEMA PENELITIAN 3-1
3.1. Alur Penyelesaian Tugas Akhir 3-1

3.2. Perumusan Masalah 3-2
3.2.1. Analisis Persoalan 3-3
3.3. Kerangka Pemikiran Teoritis 3-3
3.4. Profil Tempat Penelitian 3-4
3.4.1. Struktur Organisasi 3-5
BAB 4 ANALISIS DAN PERANCANGAN 4-1
4.1. Analisis Sistem Saat Ini (*Current System*) 4-1
4.2. Analisis Kebutuhan 4-4
4.2.1. Analisis Wawancara 4-4
4.2.2. Menentukan Kebutuhan Perangkat Lunak 4-4
4.2.3. Analisis Pengguna 4-5
4.2.4. Pemodelan Berbasis Skenario 4-5
4.2.5. Pemodelan Berbasis Pelaku 4-28
4.2.6. Pemodelan Data 4-38
4.2.7. Pemodelan Berbasis Kelas 4-40
4.3. Perancangan Perangkat Lunak 4-41
4.3.1. Perancangan Arsitektural 4-41
4.3.2. Perancangan Antarmuka Pengguna 4-41
BAB 5 IMPLEMENTASI 5-1
5.1. Kontruksi Perangkat Lunak 5-1
5.1.1. Kakas dan Kebutuhan Perangkat Lunak 5-1
5.1.2. Pengkodean 5-1
5.2. Pengujian 5-17
5.2.1. Unit Testing 5-17
BAB 6 KESIMPULAN DAN SARAN 6-1
6.1. Kesimpulan 6-1
6.2. Saran 6-1
DAFTAR PUSTAKA 1
LAMPIRAN A-1

DAFTAR ISTILAH

No.	Istilah	Keterangan
1.	*Library*	Kumpulan modul atau fungsi-fungsi yang telah dikodekan sebelumnya dan dapat digunakan oleh program-program lain untuk mempercepat proses pengembangan.
2.	*Framework*	Sebuah kerangka kerja yang digunakan untuk mengembangkan website. Framework ini diciptakan untuk membantu web developer dalam menulis baris kode.
3.	*Database*	Kumpulan data yang dikelola sedemikian rupa berdasarkan ketentuan tertentu yang saling berhubungan sehingga mudah dalam pengelolaannya. dapat memperoleh kemudahan dalam mencari informasi, menyimpan informasi dan membuang informasi
4.	*Fishbone*	Merupakan metode pemetaan masalah dan Solusi.
5.	*Login*	Proses autentikasi pengguna dengan menggunakan kredensial tertentu (misalnya, nama pengguna, email dan kata sandi) untuk mengakses sistem atau aplikasi.
6.	*Requirement*	Adalah gambaran dari layanan (services) dan batasan bagi sistem yang akan dibangun.
7.	*Design*	Diterapkan untuk membuat struktur, ekosistem, antarmuka, produk, layanan, fitur, dan proses.
8.	*Development*	Merupakan proses atau serangkaian kegiatan yang bertujuan untuk meningkatkan atau memperbaiki suatu hal.
9.	*Testing*	Merupakan proses uji coba untuk mengidentifikasi sebuah sistem agar mendapatkan hasil yang diharapkan.
10.	*Infinite event loop*	Adalah permainan bisnis yang sangat condong terhadap manajemen, berbasis kolaborasi real-time, dan komunikasi real-time.
11.	*Event driven*	Adalah salah satu Teknik pemrograman yang konsep kerjanya tergantung dari kejadian atau event tertentu
12.	*Asynchronous*	Yaitu komunikasi yang secara tertunda atau tidak langsung dengan memakai media
13.	*Server*	Merupakan suatu sistem komputer yang memiliki layanan khusus berupa penyimpanan data.
14.	*webserver*	Adalah perangkat lunak yang menyediakan layanan berbasis data dan menerima permintaan dari HTTP atau HTTPS.
15.	*platform*	Adalah serangkaian teknologi yang digunakan sebagai pondasi dalam mengembangkan software atau hardware.

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu 2-9
Tabel 3.1 Alur Penyelesaian Tugas Akhir..... 3-1
Tabel 3.2 Kerangka Pemikiran Teoritis 3-4
Tabel 4.1 Analisis Sistem Saat Ini Jenis Ruangan 1 4-2
Tabel 4.2 Analisis Sistem Saat Ini Jenis Ruangan 2 4-3
Tabel 4.3 Daftar Aktor 4-4
Tabel 4.4 Analisis Kebutuhan Pengguna 4-4
Tabel 4.5 Kebutuhan Fungsional..... 4-5
Tabel 4.6 Daftar Pengguna..... 4-5
Tabel 4.7 Use Case Diagram Peminjam..... 4-6
Tabel 4.8 Use Case Diagram Petugas 4-7
Tabel 4.9 Deskripsi Use Case Peminjam 4-7
Tabel 4.10 Deskripsi Use Case Petugas 4-8
Tabel 4.11 Use Case Mendaftar 4-8
Tabel 4.12 Use Case Melihat Profil 4-9
Tabel 4.13 Use Case Mengubah Profil..... 4-9
Tabel 4.14 Use Case Melihat Informasi Gedung 4-10
Tabel 4.15 Use Case Mengajukan Peminjaman Ruangan..... 4-10
Tabel 4.16 Use Case Melihat Riwayat Pengajuan 4-11
Tabel 4.17 Use Case Membatalkan Pengajuan 4-11
Tabel 4.18 Use Case Mengelola Peminjam..... 4-11
Tabel 4.19 Use Case Memverifikasi Peminjam 4-12
Tabel 4.20 Use Case Mengelola Ruangan..... 4-12
Tabel 4.21 Use Case Mengelola Jadwal Ruangan..... 4-13
Tabel 4.22 Use Case Mengelola Pengajuan 4-14
Tabel 4.23 Use Case Memverifikasi Pengajuan..... 4-15
Tabel 4.24 Use Case Mengelola Laporan 4-15
Tabel 4.25 Activity Diagram pada Use Case UCP-01 4-16
Tabel 4.26 Activity Diagram pada Use Case UCP-02 4-17
Tabel 4.27 Activity Diagram pada Use Case UCP-03 4-17
Tabel 4.28 Activity Diagram pada Use Case UCP-04 4-18
Tabel 4.29 Activity Diagram pada Use Case UCP-05 4-19
Tabel 4.30 Activity Diagram pada Use Case UCP-06 4-20
Tabel 4.31 Activity Diagram pada Use Case UCP-07 4-20
Tabel 4.32 Activity Diagram pada Use Case UCA-01 4-21
Tabel 4.33 Activity Diagram pada Use Case UCA-02 4-22
Tabel 4.34 Activity Diagram pada Use Case UCA-03 4-23
Tabel 4.35 Activity Diagram pada Use Case UCA-04 4-24
Tabel 4.36 Activity Diagram pada Use Case UCA-05 4-25
Tabel 4.37 Activity Diagram pada Use Case UCA-06 4-26
Tabel 4.38 Activity Diagram pada Use Case UCA-07 4-27
Tabel 4.39 Diagram Sekuens Mendaftar..... 4-28
Tabel 4.40 Diagram Sekuens Melihat Profil..... 4-28
Tabel 4.41 Diagram Sekuens Mengubah Profil 4-29
Tabel 4.42 Diagram Sekuens Melihat Informasi Gedung..... 4-29
Tabel 4.43 Diagram Sekuens Mengajukan Peminjaman Ruangan 4-30
Tabel 4.44 Diagram Sekuens Melihat Riwayat Pengajuan 4-30
Tabel 4.45 Diagaram Sekuens Membatalkan Pengajuan 4-31
Tabel 4.46 Diagaram Sekuens Mengelola Peminjam..... 4-32
Tabel 4.47 Diagram Sekuens Memverifikasi Peminjam..... 4-33
Tabel 4.48 Diagram Sekuens Mengelola Ruangan 4-34
Tabel 4.49 Diagram Sekuens Mengelola Jadwal Ruangan 4-35
Tabel 4.50 Diagram Sekuens Mengelola Pengajuan..... 4-36
Tabel 4.51 Diagram Sekuens Memverifikasi Pengajuan 4-37

Tabel 4.52 Diagram Sekuens Mengelola Laporan 4-37
Tabel 4.53 Diagram Kelas Perancangan..... 4-38
Tabel 4.54 Objek Data..... 4-38
Tabel 4.55 Atribut Data..... 4-39
Tabel 4.56 Kelas-Kelas Analisis 4-40
Tabel 4.57 Rancangan Antarmuka Halaman Informasi 4-42
Tabel 4.58 Rancangan Antarmuka Halaman Informasi Ruangan 4-42
Tabel 4.59 Rancangan Antarmuka Halaman Informasi Lokasi..... 4-43
Tabel 4.60 Rancangan Antarmuka Halaman Login 4-43
Tabel 4.61 Rancangan Antarmuka Halaman Register..... 4-44
Tabel 4.62 Rancangan Antarmuka Halaman Register 2..... 4-44
Tabel 4.63 Rancangan Antarmuka Halaman Detail Ruangan..... 4-45
Tabel 4.64 Rancangan Antarmuka Halaman Jadwal Ruangan..... 4-45
Tabel 4.65 Rancangan Antarmuka Halaman Pengajuan 4-46
Tabel 4.66 Rancangan Antarmuka Halaman Pengajuan 2 4-46
Tabel 4.67 Rancangan Antarmuka Halaman Riwayat Pengajuan 4-47
Tabel 4.68 Rancangan Antarmuka Halaman Profil..... 4-47
Tabel 4.69 Rancangan Antarmuka Halaman Dashboard..... 4-48
Tabel 4.70 Rancangan Antarmuka Halaman Daftar Pengguna 4-48
Tabel 4.71 Rancangan Antarmuka Halaman Tambah Pengguna 4-49
Tabel 4.72 Rancangan Antarmuka Halaman Detail Pengguna..... 4-49
Tabel 4.73 Rancangan Antarmuka Halaman Daftar Pengajuan 4-50
Tabel 4.74 Rancangan Antarmuka Halaman Tambah Pengajuan 4-50
Tabel 4.75 Rancangan Antarmuka Halaman Tambah Pengajuan 2 4-51
Tabel 4.76 Rancangan Antarmuka Halaman Detail Pengajuan..... 4-51
Tabel 4.77 Rancangan Antarmuka Halaman Daftar Ruangan..... 4-52
Tabel 4.78 Rancangan Antarmuka Halaman Tambah Ruangan..... 4-52
Tabel 4.79 Rancangan Antarmuka Halaman Daftar Jadwal Ruangan..... 4-53
Tabel 4.80 Rancangan Antarmuka Halaman Tambah Jadwal Ruangan..... 4-53
Tabel 4.81 Rancangan Antarmuka Halaman Laporan..... 4-54
Tabel 4.82 Rancangan Antarmuka Halaman Laporan 2..... 4-54
Tabel 5.1 Spesifikasi Perangkat Keras 5-1
Tabel 5.2 Perangkat lunak yang digunakan dalam pembangunan..... 5-1
Tabel 5.3 Daftar Nama Folder dan File Backend..... 5-2
Tabel 5.4 Daftar Nama Folder dan File Frontend..... 5-3
Tabel 5.5 Daftar Halaman Pada Aplikasi Web..... 5-4
Tabel 5.6 Antarmuka Halaman Informasi 5-9
Tabel 5.7 Antarmuka Halaman Informasi Ruangan 5-9
Tabel 5.8 Antarmuka Halaman Informasi Lokasi 5-9
Tabel 5.9 Antarmuka Halaman Login 5-10
Tabel 5.10 Antarmuka Halaman Register 5-10
Tabel 5.11 Antarmuka Halaman Register 2 5-10
Tabel 5.12 Antarmuka Halaman Detail Ruangan..... 5-11
Tabel 5.13 Antarmuka Halaman Jadwal Ruangan 5-11
Tabel 5.14 Antarmuka Halaman Pengajuan 5-11
Tabel 5.15 Antarmuka Halaman Pengajuan 2 5-12
Tabel 5.16 Antarmuka Halaman Riwayat Pengajuan..... 5-12
Tabel 5.17 Antarmuka Halaman Profil..... 5-12
Tabel 5.18 Antarmuka Halaman Dashboard 5-13
Tabel 5.19 Antarmuka Halaman Daftar Pengguna..... 5-13
Tabel 5.20 Antarmuka Halaman Tambah Pengguna..... 5-13
Tabel 5.21 Antarmuka Halaman Detail Pengguna 5-14
Tabel 5.22 Antarmuka Halaman Daftar Pengajuan..... 5-14
Tabel 5.23 Antarmuka Halaman Tambah Pengajuan 5-14
Tabel 5.24 Antarmuka Halaman Tambah Pengajuan 2 5-15
Tabel 5.25 Antarmuka Halaman Detail Pengajuan 5-15

Tabel 5.26 Rancangan Antarmuka Halaman Daftar Ruangan 5-15
Tabel 5.27 Antarmuka Halaman Tambah Ruangan 5-16
Tabel 5.28 Antarmuka Halaman Daftar Jadwal Ruangan 5-16
Tabel 5.29 Antarmuka Halaman Tambah Jadwal Ruangan 5-16
Tabel 5.30 Antarmuka Halaman Laporan 5-17
Tabel 5.31 Kode Unit Test 5-17
Tabel 5.32 Hasil Unit Test 5-18

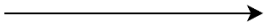
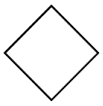
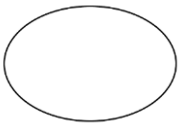
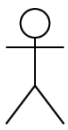
DAFTAR GAMBAR

Gambar 1.1 Metodologi Pengerjaan Tugas Akhir 1-3
Gambar 2.1 Metode Waterfall 2-2
Gambar 2.2 Interaksi Aplikasi Web 2-4
Gambar 2.3 Arsitektur Aplikasi Web 2-5
Gambar 2.4 Diagram Arsitektur React 2-7
Gambar 3.1 Analisis Persoalan 3-3
Gambar 3.2 Kerangka Pemikiran Teoritis 3-3
Gambar 3.3 Bandung Creative Hub 3-4
Gambar 3.4 Struktur Organisasi 3-5
Gambar 4.5 Perancangan Arsitektur 4-41
Gambar 5.1 Struktur Folder Backend 5-2
Gambar 5.2 Struktur Folder Frontend 5-3
Gambar 5.3 View Pada Aplikasi 5-4
Gambar 5.4 Controller pada Aplikasi 5-4
Gambar 5.5 Model pada Aplikasi 5-6
Gambar 5.6 Potongan Kode Model User 5-6
Gambar 5.7 Potongan Kode Model Room 5-7
Gambar 5.8 Potongan Kode Model Schedule 5-7
Gambar 5.9 Potongan Kode Model Booking 5-8
Gambar 5.10 Potongan Kode Index 5-8

DAFTAR LAMPIRAN

A.Lampiran Berita Acara Wawancara A-1
B.Lampiran Bukti Foto Dengan Narasumber.....B-1

DAFTAR SIMBOL

Simbol	Nama Simbol	Keterangan
	Activity	Simbol ini digunakan untuk menunjukan aktivitas yang dilakukan oleh aktor yang terlibat.
	Transition	Simbol yang digunakan untuk mempresentasikan alur aktivitas
	Off Page Connector	Simbol yang digunakan untuk kelua/masuk prosedur atau proses dalam lembar/halaman yang lain.
	On page connector	Menghubungkan arus pemrosesan disatu halaman yang sama penggunaaan konektor ini akan menghindari garis-garis yang saling disatu halaman.
	Initial Node	Menandakan awal dari kumpulan aksi atau aktivitas.
	Final Aktivity Node	Untuk menghentikan seluruh control flow atau object flow pada sebuah aktivitas.
	Nary Association	Ada pilihan aktivitas lebih dari satu.
	Join	Penggabungan Dimana ada lebih dari satu aktivitas digabungkan menjadi satu.
	Use case	Menyatakan fungsi lengkap yang dilakukan user.
	Actor	Adalah simbol yang menggambarkan pihak yang berhubungan dengan sistem.

BAB 1
PENDAHULUAN

Berisi garis besar permasalahan yang akan dibahas dan diselesaikan dengan tujuan yang telah dirumuskan seperti latar belakang, identifikasi masalah, tujuan tugas akhir, lingkup tugas akhir, metodologi tugas akhir dan sistematika penulisan tugas akhir.

1.1. Latar Belakang

Teknologi merupakan hal yang tidak dapat dihindarkan dalam kehidupan sehari – hari, perkembangan teknologi dari tahun ke tahun membuat manusia dapat meyesuaikan pembaruan teknlogi yang ada. Saat ini, teknologi sangatlah mempermudah manusia dalam melakukan segala hal. Salah satu teknologi yang banyak digunakan saat ini adalah aplikasi berbasis web.

Gedung Bandung Creative Hub yang berada di Jalan Laswi No. 7 Kota Bandung telah diresmikan oleh Wali Kota Bandung, bapak Ridwan Kamil bersama Wakil Wali Kota Bandung bapak Oded M. Danial pada 28 Desember 2017. Gedung warna warni berarsitektur unik ini dirancang untuk menjadi wadah mengembangkan kreativitas, edukasi, dan laboratorium untuk berbagai sektor industri kreatif. Bangunan ini menyediakan sarana dan fasilitas lengkap agar anak-anak muda Bandung semakin semangat untuk berkreasi [MIF17].

Namun, dengan sudah adanya teknologi tersebut sistem peminjaman ruangan di Bandung Creative Hub saat ini masih belum memanfaatkannya sehingga masih terdapat kendala, beberapa ruangan mengharuskan peminjam untuk terlebih dahulu mengajukan peminjaman secara langsung ke lokasi, yang memakan waktu dan tenaga. Hal ini tidak hanya menyebabkan ketidaknyamanan bagi peminjam, tetapi juga berdampak pada penumpukan berkas pengajuan karena masih menggunakan media kertas. Penggunaan media kertas ini menambah beban kerja petugas, terutama ketika seluruh berkas pengajuan harus direkap untuk pembuatan laporan. Proses ini juga menyulitkan petugas dalam mencari berkas lama yang diperlukan, sehingga menghambat penyusunan laporan. Selain itu, penjadwalan ruangan saat ini hanya dapat dilakukan oleh petugas yang sering kali rentan terhadap kesalahan, seperti terjadinya bentrok jadwal. Bentrok dalam penjadwalan ini terutama disebabkan oleh lonjakan permintaan peminjaman ruangan. Ketika permintaan meningkat secara signifikan, petugas sering kali kesulitan mengelola jadwal, sehingga meningkatkan risiko terjadinya kesalahan dalam penjadwalan.

Berdasarkan uraian diatas, maka dibutuhkan suatu aplikasi peminjaman ruangan berbasis web yang dapat mempermudah peminjam dalam mengajukan peminjaman ruangan tanpa perlu datang langsung ke lokasi. Aplikasi ini juga diharapkan akan membantu petugas untuk mengelola berkas-berkas pengajuan sehingga memudahkan dalam merekap dan mencari berkas pengajuan yang digunakan untuk proses pembuatan laporan, serta mengurangi potensi kesalahan dalam penjadwalan.

1.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka rumusan masalah yang didapat pada tugas akhir ini yaitu bagaimana melakukan pengembangan aplikasi peminjaman ruangan yang dapat mempermudah peminjam dalam melakukan pengajuan peminjaman dan membantu petugas dalam mengelola berkas-berkas pengajuan serta mengurangi potensi kesalahan dalam penjadwalan berbasis web?

1.3. Tujuan Tugas Akhir

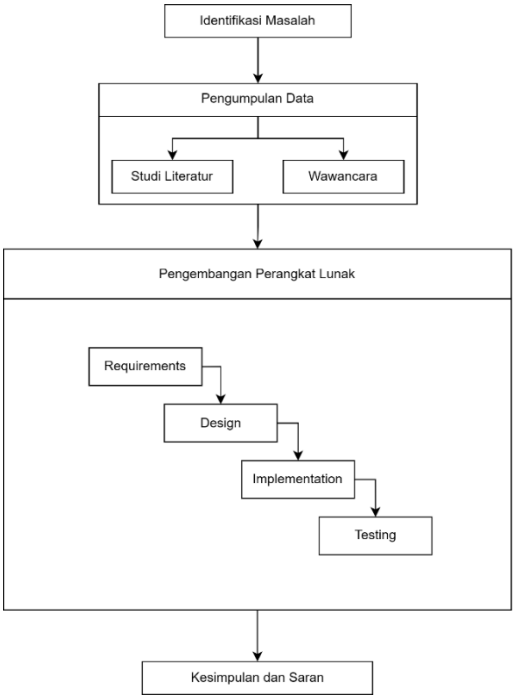
Tujuan yang ingin dicapai dalam pengerjaan tugas akhir ini adalah menghasilkan aplikasi peminjaman ruangan berbasis web yang dapat mempermudah peminjam dalam melakukan pengajuan peminjaman dan membantu petugas dalam mengelola berkas-berkas pengajuan serta mengurangi potensi kesalahan dalam penjadwalan.

1.4. Lingkup Tugas Akhir

Penyelesaian tugas akhir dibatasi sebagai berikut:

1. Pengembangan aplikasi mencakup proses berikut:
   - Pengajuan peminjaman ruangan
   - Pengelolaan pengajuan
   - Verifikasi pengajuan
   - Pengelolaan peminjam
   - Verifikasi peminjam
   - Pengelolaan ruangan
   - Pembuatan laporan.
2. Aplikasi tidak mencakup fitur notifikasi, baik untuk pemberitahuan pengajuan maupun aktivitas lainnya.
3. Aplikasi tidak mencakup penanganan atau pencatatan terkait kerusakan ruangan atau fasilitas yang digunakan.
4. Pengembangan perangkat lunak dilakukan menggunakan metode *Waterfall*, tetapi hanya mencakup tahapan *Requirements, Design, Implementation* dan *Testing*. Tahapan *Maintanance* tidak termasuk dalam lingkup penelitian ini karena tahapan ini melibatkan pemeliharaan jangka panjang yang dilakukan setelah aplikasi diterapkan dalam lingkungan nyata dan membutuhkan pemantauan berkala.

1.5. Metodologi Pengerjaan Tugas Akhir



Gambar 1.1 Metodologi Pengerjaan Tugas Akhir

Gambar 1.1 memvisualisasikan serangkaian langkah yang harus diambil untuk menyelesaikan tugas akhir ini. Tahap awal penyelesaian tugas akhir ini adalah mengidentifikasi permasalahan yang ada di Bandung Creative Hub. Setelah permasalahan diidentifikasi, langkah berikutnya perlu dilakukan pengumpulan data terkait masalah yang ada, seperti studi literatur dan wawancara.

Proses selanjutnya ialah melakukan pengembangan perangkat lunak yang ditujukan untuk menanggulangi permasalahan yang telah diidentifikasi. Metodologi yang digunakan dalam tahap ini hanya pada sampai testing yang mencakup beberapa tahap, seperti *requirements*, *design*, *implementation*, dan *testing*.

Setelah proses pembangunan selesai, langkah terakhir adalah menarik kesimpulan dari hasil yang diperoleh dan juga saran-saran untuk pengembangan lebih lanjut.

1.6. Sistematika Penulisan

Buku Tugas Akhir ditulis dengan mengikuti sistematika sebagai berikut :

Bab 1 : Pendahuluan

Bab ini menjelaskan secara umum tentang usulan penelitian yang dilakukan dalam pengerjaan tugas akhir. Terdapat beberapa elemen yang akan dibahas dalam bab ini, yaitu latar belakang masalah, identifikasi masalah, tujuan tugas akhir, lingkup tugas akhir, metodologi pengerjaan tugas akhir, dan sistematika penulisan laporan tugas akhir.

Bab 2 : Landasan Teori Dan Penelitian Terdahulu

Bab ini membahas definisi, teori-teori, dan konsep yang penting untuk pengerjaan tugas akhir. Selain itu, bab ini juga mengulas jurnal-jurnal ilmiah sebelumnya yang memiliki kesamaan dengan tugas akhir yang sedang dikerjakan.

Bab 3 : Skema Penelitian

Bab ini mencakup penjelasan mengenai alur penyelesaian tugas akhir, analisis persoalan dan manfaat tugas akhir, kerangka pemikiran teoritis, serta profil tempat penelitian.

Bab 4 : Analisis Dan Perancangan

Bab ini membahas tahapan awal dalam proses pembangunan yang terdiri dari analisis dan perancangan. Proses analisis melibatkan pendefinisian perangkat lunak dan analisis kebutuhan dengan menggunakan pemodelan spesifikasi kebutuhan.

Bab 5 : Implementasi Dan Pengujian

Pada bab ini, akan dijelaskan mengenai pengkodean aplikasi dan pengujian hasil analisis dan perancangan yang telah dibahas pada bab sebelumnya.

Bab 6 : Penutup

Bab ini berisi tentang hasil penelitian dan pernyataan yang diperoleh berdasarkan identifikasi masalah yang diajukan, serta hubungan antara semua tahap penelitian. Di bagian ini juga termasuk usulan saran untuk penelitian selanjutnya mengenai prospek penelitian yang lebih lanjut, dan rekomendasi untuk implementasi di perusahaan terkait**.**

BAB 2
LANDASAN TEORI DAN PENELITIAN TERDAHULU

Bab ini membahas definisi, teori-teori, dan konsep yang penting untuk pengerjaan tugas akhir. Selain itu, bab ini juga mengulas jurnal-jurnal ilmiah sebelumnya yang memiliki kesamaan dengan tugas akhir yang sedang dikerjakan.

2.1. Teori Yang Digunakan

Sub bab ini berisi sejumlah teori atau konsep yang relevan digunakan dalam penelitian atau tugas akhir.

2.1.1. Pengembangan Aplikasi

Pengembangan aplikasi kerap disalahartikan sebagai kegiatan seorang programmer melakukan sebuah aktifitas yaitu *coding*, proses menulis kode padahal sebenarnya pengembangan aplikasi lebih dari hanya melakukan proses menulis kode. Pengembangan Aplikasi adalah serangkaian proses yang dilakukan seorang atau sekelompok untuk membuat rangkaian proses dari rancangan awal aplikasi, membuat prototype aplikasi, implementasi, pengujian akhir aplikasi hingga aplikasi selesai dan siap untuk digunakan [MIN21].

Dalam pengembangan aplikasi ada beberapa tahap yang telah dirancang sebagai syarat agar sebuah aplikasi dapat dibuat secara terstruktural dan terancang dengan baik. Tahap tersebut adalah pengumpulan informasi, perencanaan, pengembangan, dan perawatan. Tahap ini merupakan tahap yang penting dalam membangun aplikasi. Dalam pelaksanaannya ada beberapa metode pengembangan aplikasi yang kerap digunakan yaitu seperti metode *waterfall*, *spiral*, *Extreme Programming*, *kanban*, *agile* , *prototype* dan *scrum*.

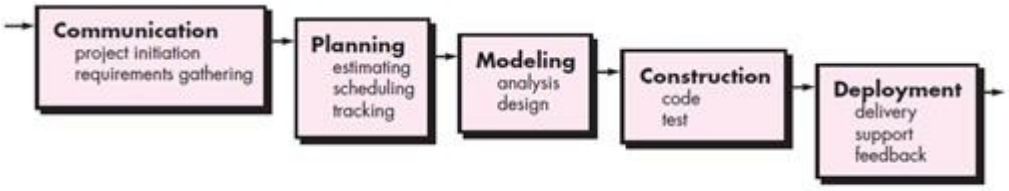
2.1.2. Metode Waterfall

Model Waterfall adalah model pengembangan perangkat lunak dengan metode tahapan berurutan dan sistematis, dimulai dari menganalisis kebutuhan yang dibutuhkan dalam pembuatan aplikasi, membuat desain sistem dari aplikasi, melakukan implmentasi, melakukan testing pada aplikasi, dan proses maintenance pada aplikasi yang sudah dibuat.

Model waterfall pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam Software Engineering (SE). saat ini model waterfall merupakan model pengembangan perangkat lunak yang sering digunakan. Model pengembangan ini melakukan pendekatan secara sistematis dan berurutan. Disebut waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya [WAH20].

2.1.2.1. Tahapan Metode Waterfall

Tahapan peneliti dalam melakukan pembangunan perangkat lunak dari metode waterfall dapat dilihat pada gambar dibawah ini



Gambar 2.1 Metode Waterfall

Sumber : [PRE10]

Model ini memiliki beberapa tahapan yang harus dilalui secara sistematis, yaitu Communication, Planning, Modeling, Construction, dan Deployment.

1. Tahap pertama adalah **Communication (Komunikasi)**. Pada tahap ini, tujuan utamanya adalah mengidentifikasi kebutuhan dan persyaratan proyek. Kegiatan yang dilakukan meliputi *Project Initiation* untuk mendefinisikan tujuan dan ruang lingkup proyek, serta *Requirements Gathering* untuk memahami kebutuhan dari pengguna dan pemangku kepentingan. Hasil dari tahap ini biasanya berupa dokumen kebutuhan (Requirements Document) yang menjadi acuan utama.
2. Tahap kedua adalah **Planning (Perencanaan)**, yang bertujuan untuk menyusun rencana proyek secara rinci. Kegiatan yang dilakukan meliputi *Estimating* untuk memperkirakan waktu, biaya, dan sumber daya yang diperlukan, *Scheduling* untuk menyusun jadwal penyelesaian proyek, dan *Tracking* untuk menentukan mekanisme pemantauan progres proyek. Hasil dari tahap ini adalah dokumen rencana proyek (Project Plan) yang menjadi pedoman pelaksanaan.
3. Tahap ketiga adalah **Modeling (Pemodelan)**, di mana desain sistem dibuat untuk memenuhi kebutuhan yang telah diidentifikasi. Proses ini mencakup *Analysis* untuk membuat spesifikasi sistem yang jelas dan detail, serta *Design* untuk merancang arsitektur sistem, antarmuka pengguna, database, dan alur kerja. Hasil dari tahap ini adalah dokumen desain sistem (System Design Document) yang menjadi panduan pengembangan perangkat lunak.
4. Tahap keempat adalah **Construction (Konstruksi)**, yang melibatkan pengembangan dan pengujian perangkat lunak. Kegiatan dalam tahap ini meliputi *Coding* untuk menulis kode program berdasarkan desain yang telah dibuat, serta *Testing* untuk memastikan sistem berfungsi sesuai dengan kebutuhan. Hasil akhirnya adalah perangkat lunak yang siap untuk diuji secara menyeluruh.
5. Tahap terakhir adalah **Deployment (Penerapan)**. Pada tahap ini, perangkat lunak dikirimkan kepada pengguna akhir dan dukungan diberikan. Proses ini meliputi *Delivery* untuk menyediakan perangkat lunak kepada pengguna, *Support* untuk membantu pengguna selama masa penggunaan awal, serta *Feedback* untuk mengumpulkan masukan guna perbaikan di masa depan. Hasil akhirnya adalah sistem yang digunakan oleh pengguna dan laporan umpan balik yang menjadi bahan evaluasi.

2.1.2.2. Kelebihan Metode Waterfall

Kelebihan dari metode waterfall yaitu :

1. Kualitas dari sistem yang dihasilkan akan baik, karena pelaksanaannya dilakukan secara bertahap.
2. Proses pengembangan model fase one by one, sehingga meminimalis kesalahan yang mungkin akan terjadi.
3. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya.

2.1.2.3. Kekurangan Metode Waterfall

Kekurangan dari metode waterfall yaitu :

1. Waktu pengembangan lama dan biayanya mahal
2. Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
3. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya.
4. Pada kenyataannya, jarang mengikuti urutan sekuensial (runtutan) seperti pada teori. Iterasi (perulangan) sering terjadi menyebabkan masalah baru.

2.1.3. Aplikasi

Aplikasi merupakan perangkat lunak proses data yang berpacu pada sebuah komputasi. Aplikasi berasal dari Bahasa Inggris *application* yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara istilah, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju [SAM20].

Berdasarkan definisi tersebut dapat disimpulkan bahwa aplikasi adalah perangkat lunak yang dirancang untuk menjalankan fungsi tertentu bagi pengguna. Ini dapat berupa program siap pakai yang memenuhi kebutuhan pengguna yang digunakan untuk mencapai tujuan tertentu.

2.1.4. Website

Website (Situs Web) merupakan kumpulan dari halaman-halaman *web* yang berhubungan dengan file-file lain yang terkait. Dalam sebuah *website* terdapat suatu halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi website. Dari *home page*, pengujung dapat mengklik *hyperlink* untuk pindah kehalaman lain yang terdapat dalam *website* tersebut [HEN14].

Website juga salah satu jenis layanan atau fasilitas yang paling banyak digunakan disamping layanan-layanan lainnya. Kelebihan website dibandingkan dengan yang lainnya adalah karena website mampu menyajikan informasi berupa teks, gambar, suara ataupun video yang interaktif.

2.1.5. Aplikasi Web

Aplikasi web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web. Aplikasi web adalah bagian dari client-side yang dapat dijalankan oleh browser web [SIM10].

Aplikasi web merupakan sebuah aplikasi yang dapat diakses melalui internet atau intranet, saat ini aplikasi web lebih banyak digunakan dibandingkan aplikasi desktop.

Interaksi dalam aplikasi web dibagi ke dalam tiga langkah, yaitu:

1. Permintaan

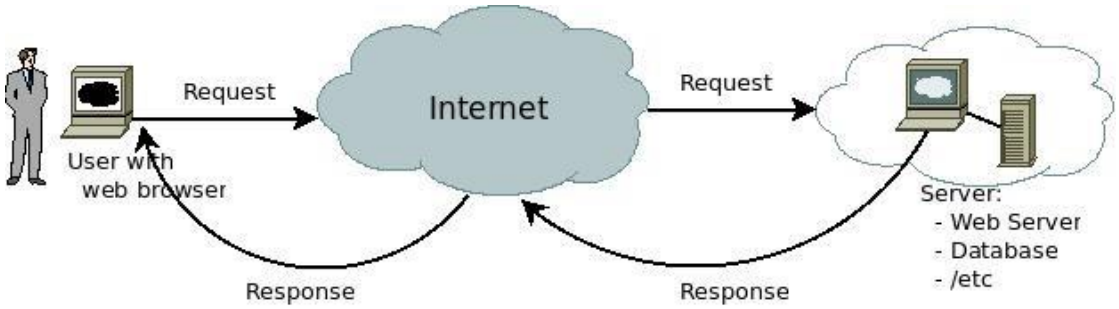
   Pengguna mengirimkan permintaan ke server web, biasanya via halaman web yang ditampilkan pada browser web.

2. Pemrosesan

   Server web menerima permintaan yang dikirimkan oleh pengguna, kemudian memproses permintaan tersebut.

3. Jawaban

   Browser menampilkan hasil dari permintaan pada jendela browser



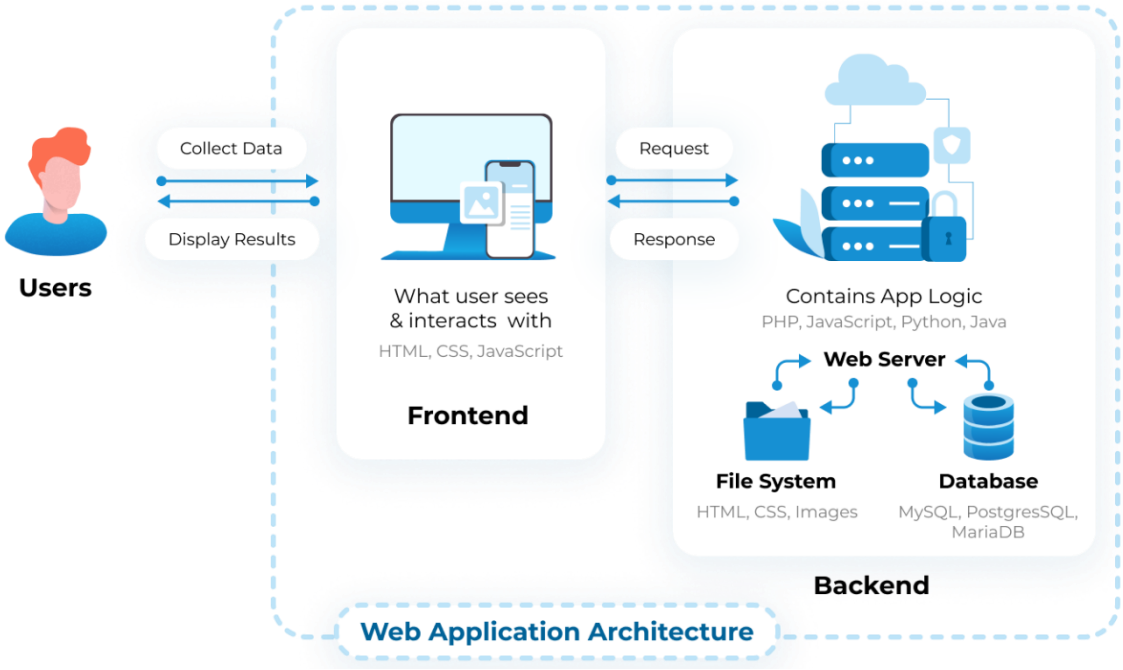
Gambar 2.2 Interaksi Aplikasi Web

Sumber : [PUR10]

2.1.6. Arsitektur Aplikasi Web

Arsitektur aplikasi web adalah kerangka struktural yang mengatur desain dan pengembangan aplikasi web [PET24]. Ini mencakup prinsip-prinsip dan praktik terbaik yang menentukan bagaimana berbagai bagian aplikasi, seperti *frontend, backend,* dan *database,* berinteraksi dengan baik untuk menjalankan fungsi yang diinginkan.

Gambar dibawah merupakan contoh dari arsitektur aplikasi web.



Gambar 2.3 Arsitektur Aplikasi Web

Sumber : [PET24]

Tujuan utama arsitektur aplikasi web adalah memastikan efisiensi, keandalan, dan skalabilitas sistem digital. Ini dilakukan dengan mengatur komponen-komponen frontend untuk memberikan pengalaman pengguna yang mulus dan intuitif, menangani fungsionalitas backend untuk memproses data dan berkomunikasi dengan frontend, serta merancang sistem agar dapat mengatasi pertumbuhan pengguna atau volume data tanpa mengorbankan kinerja.

2.1.7. Peminjaman Ruangan

Peminjaman berasal dari kata dasar pinjam. Definisi kata peminjaman adalah suatu proses kegiatan meminjam atau meminjamkan. Berarti definisi peminjaman ruangan adalah suatu proses kegiatan meminjam atau meminjamkan ruangan untuk digunakan oleh orang yang melakukan peminjaman dalam waktu yang telah ditentukan. Sistem informasi peminjaman ruangan adalah suatu sistem yang memiliki fungsi untuk pemesanan atau peminjaman suatu ruangan dengan menggunakan aplikasi berbasis web. Aplikasi peminjaman ruangan adalah aplikasi yang dirancang untuk membantu kegiatan pengguna dalam melakukan peminjaman ruangan, pengolahan data, dan dapat memberikan informasi jadwal ruangan [DHA20].

Aplikasi ini dapat melakukan peminjaman ruangan secara online dan dapat menampilkan informasi jadwal ruangan. Dengan aplikasi peminjaman ruangan ini pengguna dapat melakukan peminjaman ruangan secara online tidak perlu melakukan surat menyurat atau datang langsung untuk melakukan peminjaman ruangan pada gedung Bandung Creative Hub.

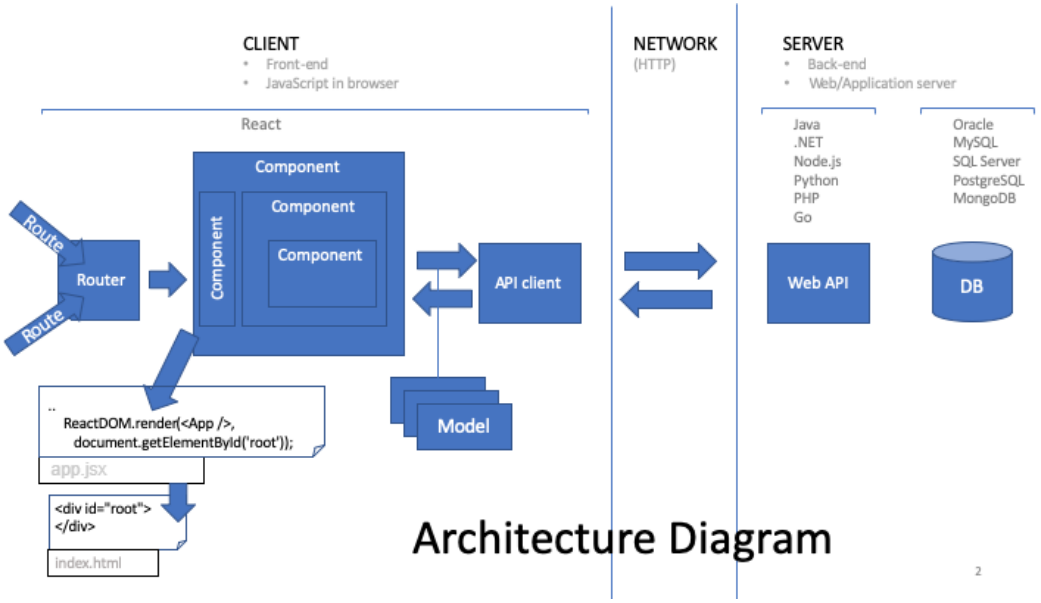
2.1.8. React.js

React biasa dikenal sebagai React atau React.Js, merupakan sebuah *library* JavaScript dengan sifat *opensource* yang digunakan untuk mempermudah dalam membangun tampilan antarmuka pengguna (UI) menjadi lebih responsive dan interaktif. React biasa digunakan untuk mengembangkan tampilan pada *Single Page Application* (SPA) dan *mobile application* [KHU18].

Awalnya, ReactJS adalah *JavaScript Library* yang dikembangkan oleh Jordan Walke sekitar tahun 2013. ReactJS digunakan sebagai salah satu *framework* untuk membuat bagian front-end dari sebuah aplikasi. React JS berfungsi untuk mengatur *layer* pada tampilan untuk *desktop* maupun versi *mobile app*. Dengan React, pengguna bisa membuat *Reusable Component* atau jenis komponen yang bisa digunakan kembali tanpa harus membangunnya dari awal lagi [SUL22].

ReactJS memiliki beberapa kelebihan atau keunggulan diantaranya yaitu :

1. React menggunkan Virtual DOM, yang membuat pembaruan dan rendering yang membuat antarmuka pengguna aplikasi menjadi efisien. Ketika data berubah, React memperbarui perubahan hanya pada bagian-bagian spesifik dari DOM yang dimaksudkan dan tidak perlu memperbarui halaman, yang mneghasilkan peningkatan kinerja aplikasi.
2. Penggunaan komponen yang dapat digunakan kembali (*Reusable Component*) dalam React memudahkan pengembangan dan pemeliharaan aplikasi. Komponen yang sudah dibuat dapat digunakan di berbagai bagian aplikasi tanpa perlu menulis ulang kode.
3. React memiliki ekosistem yang kuat dengan adanya React Native, dimana kita dapat membuat aplikasi mobile dengan hasil build menggunakan komponen dan pengetahuan yang sama dengan ReactJS. Ini memberikan pengembang kemampuan untuk membangun aplikasi antar-platform, yaitu iOS dan Android, lebih mudah dan efisien.
4. React juga memiliki ekosistem yang banyak tersedia banyak pustaka tambahan dan alat bantu., seperti Redux untuk “managing state”, React Router untuk “managing routing”, dan banyak alat lainnya. Ini memudahkan pengembang untuk menambahkan fungsionalitas aplikasi tambahan.



Gambar 2.4 Diagram Arsitektur React

Sumber : [HAN24]

2.1.9. JSX

React.Js memiliki aturan tersendiri dalam menuliskan kode pada setiap komponennya, yaitu dengan menggunakan JSX yang memungkinkan developer untuk menuliskan sintaks HTML di dalam JavaScript. Dengan mengkombinasikan dua hal tersebut antara HTML dan JavaScript, penulisan kode menjadi lebih sederhana dan ringkas dalam pendeskripsian tampilan antarmuka [AKH22].

JSX sendiri merupakan ekstensi sintaks untuk ECMAScript tanpa ada definisi semantik tertentu. Pada dasarnya React.Js bisa digunakan tanpa JSX, namun tanpa JSX penulisan kode menjadi lebih panjang dan sulit. Sedangkan dengan menggunakan JSX dapat memudahkan pengembang dalam penulisan kode dan membantu saat melakukan error debugging [KHU18].

2.1.10. Node.js

Node.js adalah sebuah platform yang dirancang untuk *webserver*. Aplikasi ini ditulis dalam bahasa javascript dan berbasis pada *event* (*event driven*). Node.js dieksekusi sebagai aplikasi *server*. Node.js memiliki efisiensi memori yang lebih baik apabila bekerja dalam keadaan muatan yang banyak. Aplikasi ini terdiri dari V8 Javascript *engine* buatan google dan beberapa modul bawaan yang terintegrasi (Nodejs.org). Di dalam Node.js terdapat operasi-operasi yang sifatnya *synchronous* dan *asynchronous*, tetapi keunggulan utama Node.js terletak pada kemampuannya menjalankan operasi asynchronous atau non-blocking I/O. Ini berarti bahwa Node.js tidak perlu menunggu satu operasi selesai sebelum melanjutkan ke operasi berikutnya, sehingga dapat menangani banyak permintaan secara bersamaan dengan efisien. Node.js menggunakan pendekatan *event-driven* berbasis *infinite event loop* dalam satu *thread* untuk mengurangi jumlah memori [SHA20].

2.1.11. Express.js

Express.JS merupakan sebuah Node.JS framework yang berguna untuk mempermudah pembuatan aplikasi berbasis Node.JS dengan menggunakan design pattern yang dapat disesuaikan dan sangat fleksibel. Selain itu Express.JS juga merupakan framework yang sangat ringan dan cocok untuk membuat web aplikasi dan API [FAJ17].

2.1.12. Application Programming Interface (API)

Application Programming Interface (API) adalah antarmuka yang dibangun oleh pengembang sistem sehingga beberapa atau seluruh fungsi sistem dapat diakses secara terprogram [AFR23]. API berisi sekumpulan data dan fungsi untuk memfasilitasi interaksi antara program komputer dan memungkinkan mereka untuk bertukar informasi [MAS12].

Tujuan penggunaan dari API adalah untuk saling berbagi data antar aplikasi yang berbeda tersebut, Tujuan penggunaan API lainnya yaitu untuk mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah fungsi yang terpisah sehingga para developer tidak perlu lagi membuat fitur yang serupa [KUR20]. API memiliki beberapa arsitektur, ada REST (*Representational State Transfer*), Graphql, SOAP (*Simple Object Access Protocol*), dll.

2.1.13. REST

Representational State Transfer atau biasa disingkat dengan REST merupakan standar dari arsitektur berbasis web. REST menggunakan protokol HTTP untuk komunikasi data. Pada arsitektur REST, di sisi server menyediakan *resources* atau data dan di sisi klien mengakses dan menampilkan sumber daya tersebut. Pada setiap *resource* diidentifikasi melalui URIs (*Universal Resource Identifiers*) atau global ID. JSON atau XML digunakan untuk merepresentasikan sumber daya pada REST [LUM21]. REST dikembangkan oleh Roy Fielding yang merupakan co-founder dari Apache HTTP Server Project [KUR20]. Arsitektur REST umumnya diterapkan pada desain API untuk web *services* modern. Web API yang sesuai dengan gaya arsitektur REST adalah REST API [MAS12].

REST API memanfaatkan metode HTTP standar seperti GET, POST, PUT, DELETE untuk memanipulasi *resource* (sumber daya) yang ada diserver.

2.1.14. UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah sebuah bahasa pemodelan perangkat lunak yang telah distandardisasi sebagai media penulisan cetak biru (blueprints) perangkat lunak (Pressman). UML bisa saja digunakan untuk visualisasi, spesifikasi, kontruksi dan dokumentasi beberapa bagian-bagian dari system yang ada dalam perangkat lunak. Dalam kata lain, seperti halnya seorang arsitek dalam membuat dokumen cetak biru yang digunakan oleh perusahaan konstruksi untuk membangun sebuah bangunan, arsitek perangkat lunak membuat diagram-diagram UML untuk membantu programmer/ developer membangun perangkat lunak. Untuk selanjutnya, semakin kita mengetahui beberapa kosakata yang digunakan UML, kita akan semakin mudah dalam memahami spesifik [SUM21].

2.1.15. Unit Testing

Unit testing merupakan tahap dasar dalam pengujian. Unit testing berfokus pada pengujian building block yang lebih kecil daripada program atau sistem yang diuji. Pengujian ini mengeksekusi setiap fitur atau fungsionalitas untuk memastikan masing-masing fitur atau fungsionalitas tersebut berfungsi sesuai dengan yang diharapkan. Pada definisi lainnya, unit testingmerupakan pengujian yang mencakup pengujian sepotong atau sebagian kode. Dengan demikian, pengujian ini dapat digunakan untuk menguji kelas atau modul tertentu. Pada implementasinya, unit testing berupa sebuah kode program yang dibuat oleh tim pengembang untuk menguji sebuah fitur atau fungsionalitas pada sebuah sistem. Hal tersebut dilakukan untuk membuktikan apakah kode program yang telah dikembangkan sudah sesuai yang diharapkan atau tidak. Unit testing ini sangat efektif bagi tim pengembang, karena akan mempermudah tim pengembang dalam memperbaiki desain kode program dan mengurangi waktu saat melakukan debugging [HAS21].

2.2. Penelitian Terdahulu

Berikut ini merupakan penelitian terdahulu yang dijadikan sebagai referensi untuk tugas akhir atau penelitian yang hamper sama dengan topik tugas akhir.

Tabel 2.1 Penelitian Terdahulu

No	Judul Penelitian	Peneliti	Hasil Penelitian
1	Rancang Bangun Aplikasi Peminjaman Ruangan Pada Laboratorium Komputer Universitas Dinamika	Triska Wahyudiningkasih, 2021	Aplikasi Peminjaman Ruangan Pada Laboratorium Komputer Universitas Dinamika berbasis Web
2	Rancang Bangun Aplikasi Booking Gedung Berbasis Web	Roro Nawang Wulan, Ade Dwi Putra Ade	Aplikasi Booking Gedung Berbasis Web
3	Rancang Bangun Aplikasi Peminjaman Ruangan Berbasis Web dan SMS Gateway	Rudi Setiawan , Indirana Soraya	Aplikasi Peminjaman Ruangan Berbasis Web dan SMS Gateway

1. Literatur pertama dan kedua digunakan untuk memahami bagaimana alur atau tahapan pembuatan aplikasi dan membantu dalam proses pengembangan aplikasi.
2. Literatur Ketiga, digunakan untuk memahami penggunaan notifikasi dalam aplikasi.

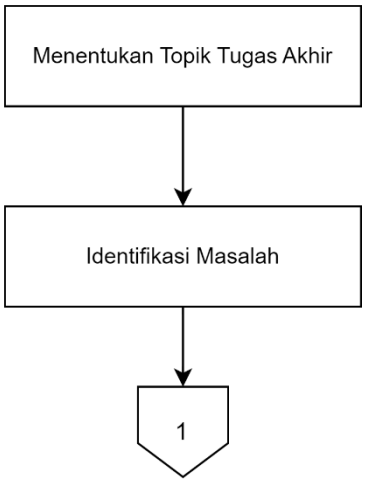
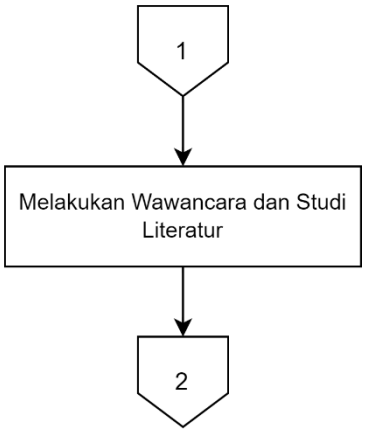
BAB 3
SKEMA PENELITIAN

Bab ini berisi penjelasan alur penyelesaian tugas akhir, perumusan masalah, kerangka pemikiran teoritis, dan profile tempat penelitian. Skema penelitian berfungsi sebagai panduan awal bagi pembaca untuk memahami tujuan, ruang lingkup, dan landasan teoritis penelitian yang akan dilakukan. Dengan menyediakan informasi yang jelas mengenai alur penyelesaian tugas akhir, perumusan masalah, kerangka pemikiran teoritis, dan profil tempat penelitian, skema penelitian membantu pembaca untuk memahami konteks dan relevansi penelitian secara lebih baik.

3.1. Alur Penyelesaian Tugas Akhir

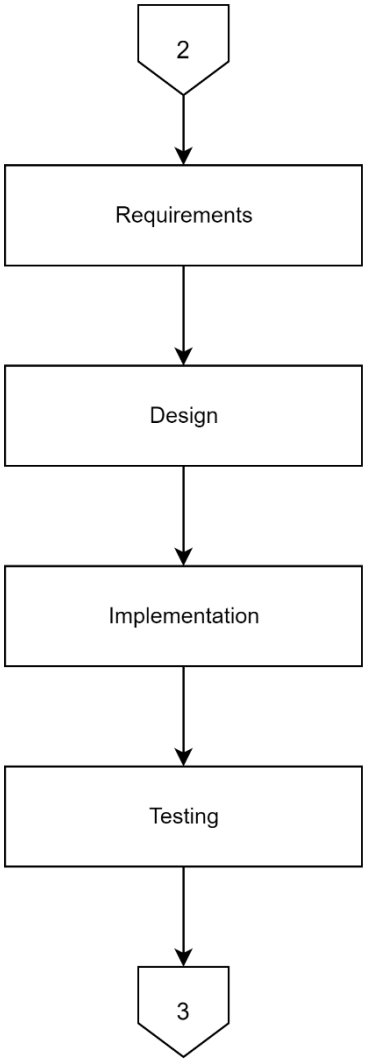
Alur penyelesaian tugas akhir ini dilakukan berdasarkan digaram alur penyelesaian. Berikut ini merupakan alur penyelesaian tugas akhir di jelaskan pada tabel 3.1:

Tabel 3.1 Alur Penyelesaian Tugas Akhir

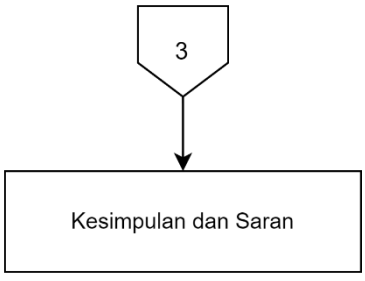
Tahap dan Hasil	Langkah Penelitian	Literatur dan Referensi
Tahap 1: Identifikasi Masalah Hasil: Mendapatkan topik TA dan masalah yang teridentifikasi Kontribusi: Berguna untuk pengumpulan data		1. Latar belakang 2. Identifikasi Masalah 3. Tujuan TA 4. Lingkup TA 5. [MIF17] 6. [ATT23]
Tahap 2: Pengumpulan data dengan melakukan wawancara dan studi literatur berdasarkan masalah yang ditemukan Hasil: Data dan informasi mengenai ruangan dan proses pengajuan peminjaman Kontribusi: Berguna dalam mendukung proses membangun perangkat lunak berdasarkan kebutuhan pengguna.		[MIF17] [DHA20] [SAM20] [HEN14] [SIM10] [PET24] [KHU18] [SUL22] [AKH22]

<table>
<tr>
<td>Tahap 3:
Pembangunan perangkat lunak yang terdiri dari Requirements, Design, Implementation dan Testing.

Hasil:
Menghasilkan pemahaman analisis, design perangkat lunak, coding menggunakan React dan Express, pengujian dengan metode unit testing..

Kontribusi:-</td>
<td> 
</td>
<td>[MIN21]
[WAH20]
[SAM20]
[HEN14]
[SIM10]
[PET24]
[DHA20]
[KHU18]
[AKH22]
[SUL22]
[PRE10]
[SHA20]
[FAJ17]
[AFR23]
[MAS12]
[KUR20]
[HAS21]
[SUM21]</td>
</tr>
<tr>
<td>Tahap 4:
Kesimpulan dan saran

Hasil:
Kesimpulan dan saran berdasarkan hasil yang telah dicapai dalam pengembangan perangkat lunak yang telah selesai.

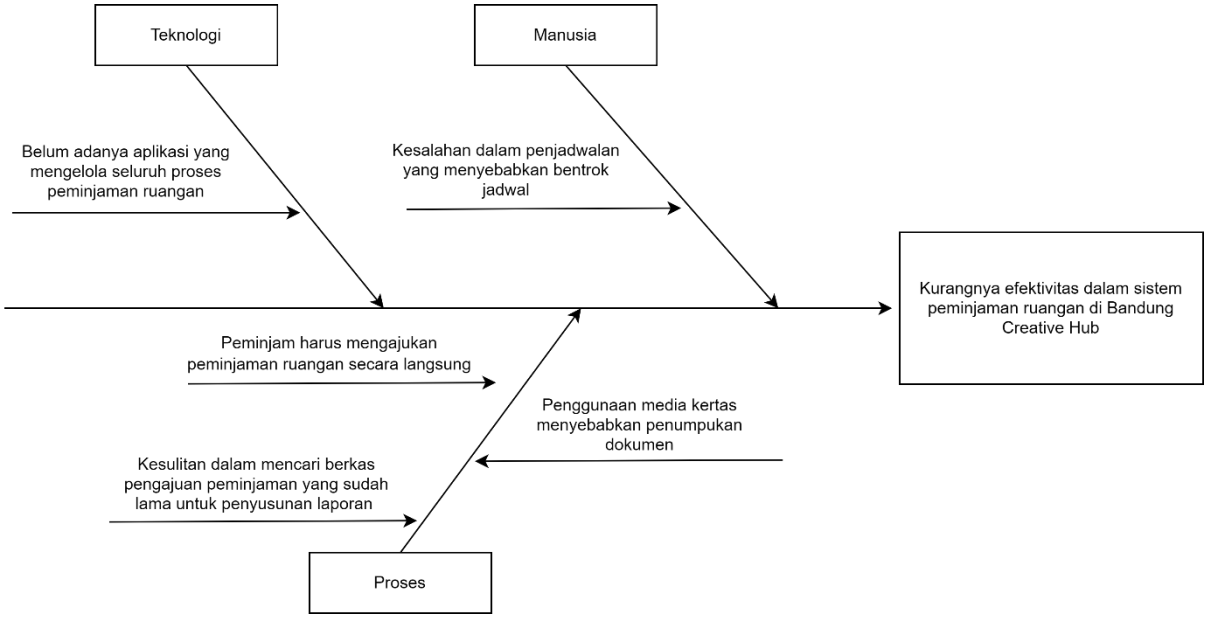
Kontribusi:-</td>
<td> 
</td>
<td></td>
</tr>
</table>

3.2. Perumusan Masalah

Perumusan masalah bertujuan untuk mengidentifikasi suatu permasalahan yang terdapat pada penelitian tugas akhir ini. Pada tahap ini masalah dirumuskan menggunakan diagram sebab akibat atau biasa disebut dengan fishbone diagram.

3.2.1. Analisis Persoalan

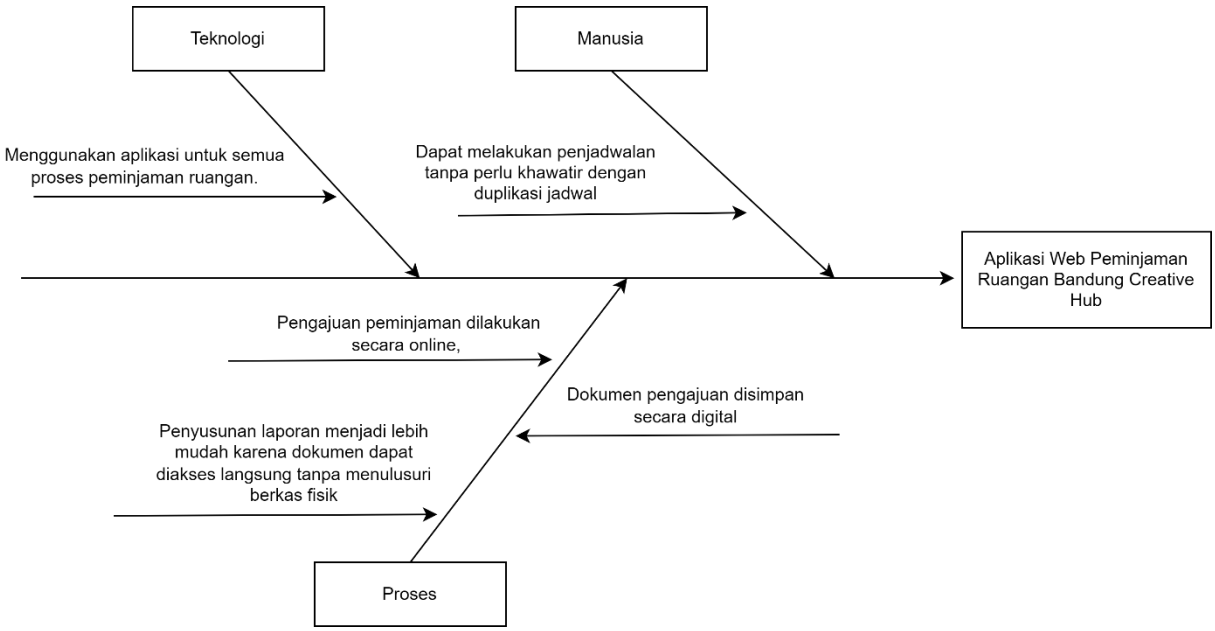
Analisis permasalahan dilakukan dengan menggunakan diagram fishbone sebagai representasi persoalan yang ada dalam tugas akhir Pengembangan Aplikasi Peminjaman Ruangan Berbasis Web Di Bandung Creative Hub. Analisis persoalan terdapat pada gambar 3.1 berikut



Gambar 3.1 Analisis Persoalan

3.3. Kerangka Pemikiran Teoritis

Pada tahap kerangka pemikiran teoritis ini akan dijelaskan mengenai tahapan yang dilakukan dalam pembangunan aplikasi peminjaman ruangan.



Gambar 3.2 Kerangka Pemikiran Teoritis

Berikut ini merupakan penjelasan detail dari kerangka pemikiran teoritis pada tugas akhir ini yang dijelaskan melalui tabel 3.2.

Tabel 3.2 Kerangka Pemikiran Teoritis

No	Komponen Utama	Komponen Detail	Solusi
1.	Teknologi	Menggunakan satu aplikasi, untuk memudahkan peminjam dan petugas dalam melakukan proses pengajuan peminjaman.	Aplikasi Web Peminjaman Ruangan Bandung Creative Hub
2.	Proses	Pengajuan bisa dilakukan secara online, sehingga memudahkan saat pengajuan peminjaman. Dokumen pengajuan yang disimpan secara digital memudahkan petugas dalam penyusunan laporan.	
3.	Manusia	Dengan adanya aplikasi dapat melakukan penjadwalan tanpa perlu khawatir dengan duplikasi jadwal.	

3.4. Profil Tempat Penelitian

Bandung Creative Hub merupakah sebuah gedung yang berada di Jalan Laswi No. 7 Kota Bandung. Gedung ini telah diresmikan pada hari Kamis 28 Desember 2017 oleh Wali Kota dan Wakil Wali Kota Bandung yang menjabat pada saat itu yaitu M. Ridwan Kamil dan Oded M. Danial.

Gedung Bandung Creative Hub merupakan pusat kreativitas dan inovasi. Gedung ini merupakan tempat bagi para pelaku industri kreatif untuk berkumpul, berkolaborasi, dan mengembangkan ide-ide kreatif mereka. Dengan desain arsitektur yang modern dan fasilitas yang lengkap, Gedung Bandung Creative Hub menjadi tempat yang ideal bagi para seniman, desainer, musisi, dan pelaku industri kreatif lainnya untuk menghasilkan karya-karya yang inspiratif. Selain itu, Gedung Bandung Creative Hub juga sering mengadakan berbagai acara dan workshop yang bertujuan untuk memperluas pengetahuan dan keterampilan para pelaku industri kreatif. Dengan demikian, Gedung Bandung Creative Hub menjadi pusat yang vital bagi perkembangan industri kreatif di Kota Bandung.

Gambar 3.3 Bandung Creative Hub

3.4.1. Struktur Organisasi

Gambar 3.4 Struktur Organisasi

BAB 4
ANALISIS DAN PERANCANGAN

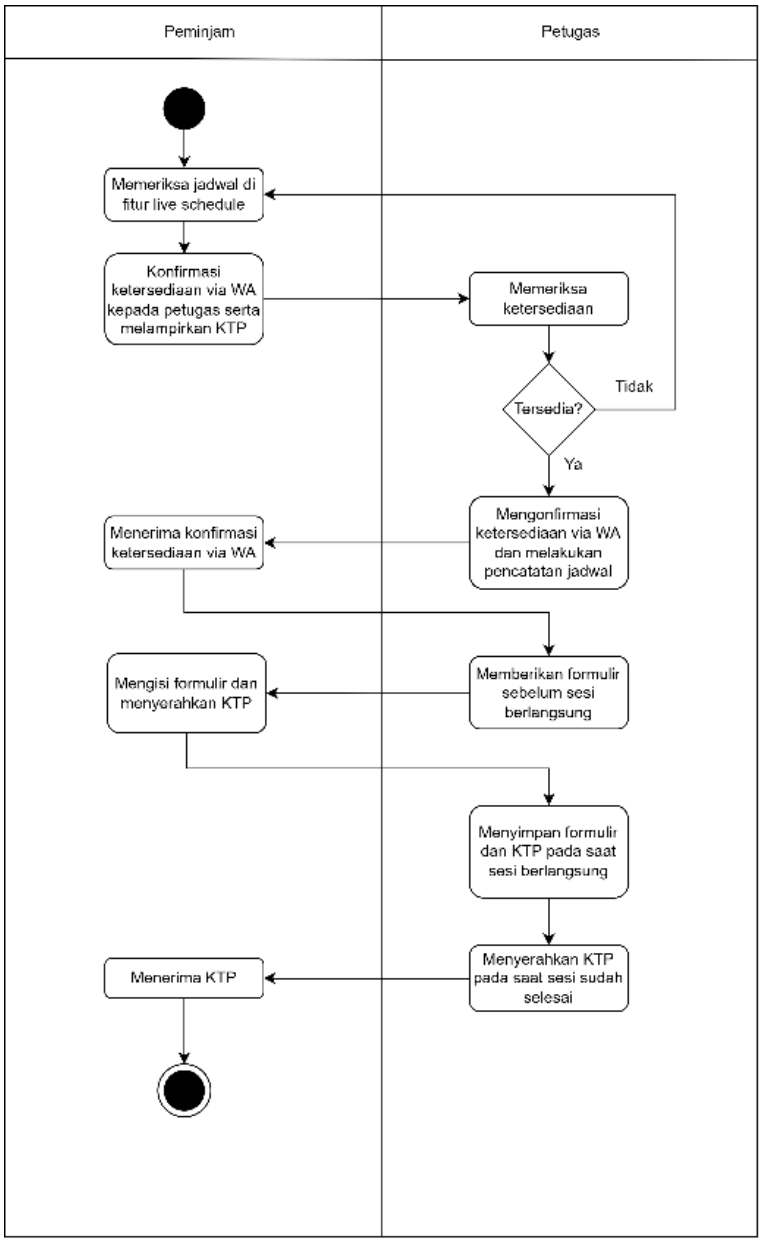
Bab Analisis dan Perancangan ini berisi mengenai proses analisis dan perancangan. Proses analisis terdiri dari pendefinisian perangkat lunak, pendefinisian kebutuhan yang harus dipenuhi perangkat lunak, dan analisis terhadap kebutuhan dengan menggunakan pemodelan spesifikasi kebutuhan dan pemodelan kebutuhan, sedangkan untuk proses perancangan perangkat lunak terdiri dari perancangan data, perancangan arsitektural, dan perancangan antarmuka.

4.1. Analisis Sistem Saat Ini (*Current System*)

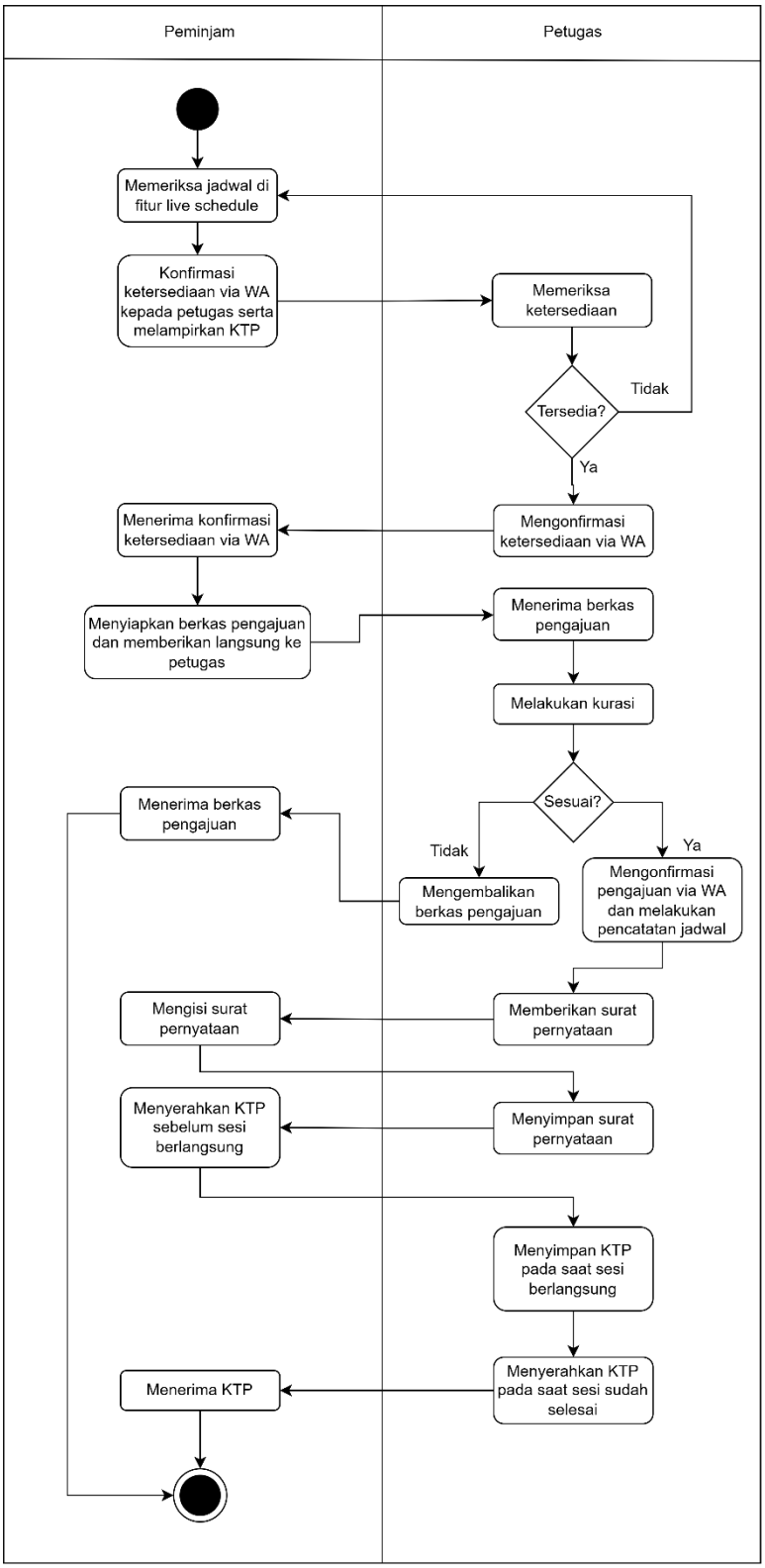
Saat ini, Bandung Creative Hub memiliki fitur live schedule menggunakan spreadsheet untuk menampilkan jadwal ruangan, yang dapat diakses melalui aplikasi media sosial. Proses pengajuan peminjaman ruangan dibedakan berdasarkan kapasitas, yaitu jenis ruangan 1 dengan kapasitas tidak lebih dari 15 peserta dan jenis ruangan 2 dengan kapasitas lebih dari 15 peserta. Peminjam bisa menghubungi petugas via WhatsApp untuk mengonfirmasi ketersediaan ruangan.

Untuk ruangan dengan kapasitas kurang dari 15 peserta, peminjam hanya perlu mengajukan via WhatsApp dengan melampirkan KTP Kota Bandung. Namun, untuk ruangan dengan kapasitas lebih dari 15 peserta, peminjam harus datang langsung ke lokasi dengan membawa berkas pengajuan. Alur aktivitas dapat dilihat pada tabel 4.1. dan 4.2.

Tabel 4.1 Analisis Sistem Saat Ini Jenis Ruangan 1

Current Activity Diagram

Deskripsi
Diagram *current system* ini menjelaskan aktivitas yang dilakukan peminjam dan petugas untuk menjelaskan proses yang sedang berjalan pada sistem pengajuan peminjaman jenis ruangan 1 dengan melakukan pengajuan menggunakan hotline WA.

Tabel 4.2 Analisis Sistem Saat Ini Jenis Ruangan 2

Current Activity Diagram

Deskripsi
Diagram *current system* ini menjelaskan aktivitas yang dilakukan peminjam dan petugas untuk menjelaskan proses yang sedang berjalan pada sistem pengajuan peminjaman jenis ruangan 2 dengan melakukan pengajuan menggunakan hotline WA dan peminjam harus memberikan berkas pengajuan dengan mendatangi langsung ke lokasi.

4.2. Analisis Kebutuhan

Pada tahap analisis kebutuhan dalam tugas akhir Pengembangan Aplikasi Peminjaman Ruangan Berbasis Web Di Bandung Creative Hub adalah dengan melakukan interaksi dengan calon pengguna, yaitu dengan peminjam dan petugas Bandung Creative Hub, yang nantinya akan menggunakan sistem tersebut.

4.2.1. Analisis Wawancara

Analisis wawancara dilakukan untuk membantu mengumpulkan data sebagai bahan analisis dalam penyusunan penelitian tugas akhir Pengembangan Aplikasi Peminjaman Ruangan Berbasis Web Di Bandung Creative Hub.

4.2.2. Menentukan Kebutuhan Perangkat Lunak

Pada tahap analisis kebutuhan, harus ditentukan kebutuhan atau kondisi yang harus dicapai oleh sistem atau aplikasi yang akan dikembangkan. Hal-hal apa saja yang harus ada dan bisa dilakukan oleh aplikasi.

4.2.2.1. Analisis Aktor

Analisis Aktor merupakan proses mengidentifikasi baik perorangan maupun kelompok yang akan mempengaruhi atau mempunyai peran secara langsung atau tidak langsung dalam pengembangan sistem. Daftar aktor dapat dilihat pada tabel 4.3.

Tabel 4.3 Daftar Aktor

No	Aktor	Deskripsi
1.	Peminjam	Orang yang melakukan peminjaman ruangan
2.	Petugas	Orang yang bertugas melayani peminjaman ruangan

4.2.2.2. Analisis Kebutuhan Pengguna

Analsis kebutuhan pengguna adalah tahap penting dalam proses analisis dan perancangan suatu sistem. Tujuan dari analisis kebutuhan pengguna adalah untuk memahami dengan baik apa yang dibutuhkan oleh pengguna dalam suatu sistem yang akan dibuat. Analisis kebutuhan pengguna dapat dilihat pada tabel 4.4.

Tabel 4.4 Analisis Kebutuhan Pengguna

No	Kode	User Requirement
1.	UR-01	Peminjam ingin dapat melakukan pendaftaran akun
2.	UR-02	Peminjam ingin dapat melihat profil
3.	UR-03	Peminjam ingin dapat mengubah profil
4.	UR-04	Peminjam ingin dapat melihat informasi gedung
5.	UR-05	Peminjam ingin dapat mengajukan peminjaman ruangan
6.	UR-06	Peminjam ingin dapat melihat riwayat pengajuan
7.	UR-07	Peminjam ingin dapat membatalkan pengajuan
8.	UR-08	Petugas ingin dapat mengelola peminjam
9.	UR-09	Petugas ingin dapat memverifikasi peminjam
10.	UR-10	Petugas ingin dapat mengelola ruangan
11.	UR-11	Petugas ingin dapat mengelola jadwal ruangan
12.	UR-12	Petugas ingin dapat mengelola pengajuan
13.	UR-13	Petugas ingin dapat memverifikasi pengajuan
14	UR-14	Petugas ingin dapat mengelola laporan peminjaman ruangan

4.2.2.3. Analisis Fungsional

Analisis fungsional merupakan tahap untuk mengetahui kebutuhan fungsional dari perangkat lunak yang akan dikembangkan berdasarkan terjemahan dari kebutuhan pengguna yang diubah menjadi kebutuhan fungsional dari aplikasi. Analisis fungsional dapat dilihat pada tabel 4.5.

Tabel 4.5 Kebutuhan Fungsional

No	Kode	Kebutuhan Fungsional
1.	FR-01	Perangkat lunak harus dapat melayani pendaftaran akun peminjam
2.	FR-02	Perangkat lunak harus dapat menampilkan profil peminjam
3.	FR-03	Perangkat lunak harus dapat melayani perubahan data profil peminjam
4.	FR-04	Perangkat lunak harus dapat menampilkan informasi gedung
5.	FR-05	Perangkat lunak harus dapat melayani pengajuan peminjaman ruangan
6.	FR-06	Perangkat lunak harus dapat menampilkan riwayat pengajuan peminjaman ruangan
7.	FR-07	Perangkat lunak harus dapat melayani pembatalan pengajuan peminjaman ruangan
8.	FR-08	Perangkat lunak harus dapat mengelola data peminjam
9.	FR-09	Perangkat lunak harus dapat melayani verifikasi peminjam
10.	FR-10	Perangkat lunak harus dapat mengelola data ruangan
11.	FR-11	Perangkat lunak harus dapat mengelola data jadwal ruangan
12.	FR-12	Perangkat lunak harus dapat mengelola data pengajuan
13.	FR-13	Perangkat lunak harus dapat melayani verifikasi pengajuan peminjaman ruangan
14.	FR-14	Perangkat lunak harus dapat mengelola laporan peminjaman ruangan

4.2.3. Analisis Pengguna

Analisis pengguna merupakan analisis yang digunakan untuk menentukan siapa saja pengguna yang akan menggunakan atau berinteraksi dengan aplikasi yang akan dibangun. Daftar pengguna dapat dilihat pada tabel 4.6.

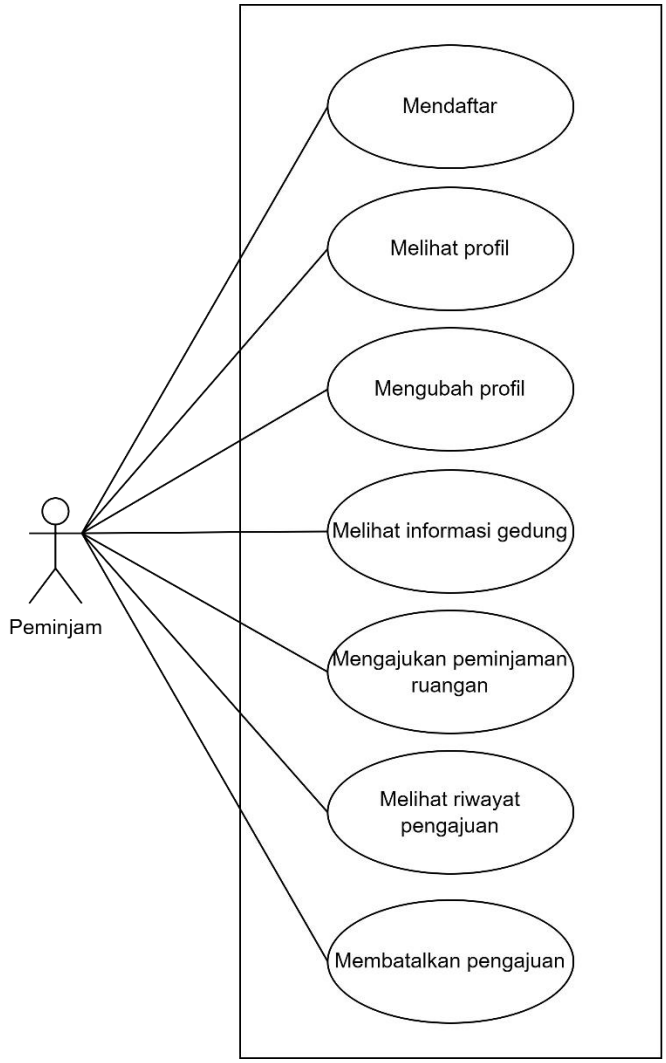
Tabel 4.6 Daftar Pengguna

No	Pengguna	Deskripsi
1.	Peminjam	Orang yang melakukan peminjaman ruangan
2.	Petugas	Orang yang bertugas melayani peminjaman ruangan

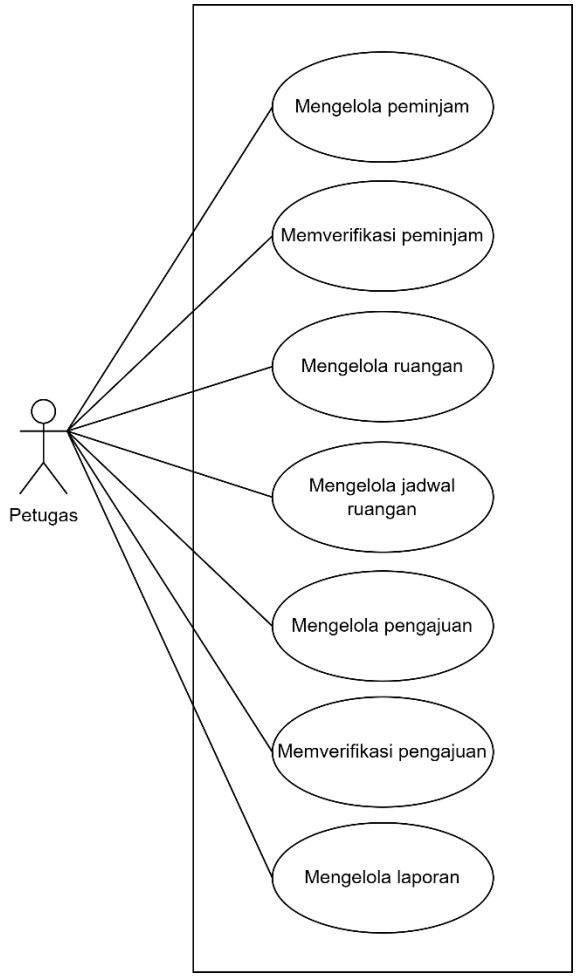
4.2.4. Pemodelan Berbasis Skenario

Pemodelan berbasis skenario menggambarkan interaksi yang terjadi anatara pengguna dengan sistem atau perangkat lunak. Penggunaan UML (*Unified Modeling Language*) sebagai pemodelan spesifikiasi kebutuhan pengguna umumnya dimulai dengan pembuatan skenario-skenario dalam belum *use case*, *activity diagram*, *swimline diagram*. Berikut ini adalah *use case* yang dibangun untuk aplikasi pengajuan peminjaman berbasis web. *Use case* diagram dapat dilihat pada gambar 4.3. dan gambar 4.4.

Tabel 4.7 *Use Case* Diagram Peminjam

***Use Case* Diagram**

Deskripsi
Pada *use case* diagram di atas menjelaskan representasi dari setiap *use case* perangkat lunak yang akan dibangun. Terdapat aktor yaitu peminjam yang berinteraksi dengan setiap *use case* perangkat lunak diatas. Terdiri dari 7 *use case*, diantaranya peminjam dapat mendaftar, melihat profil, mengubah profil, melihat informasi gedung , mengajukan peminjaman ruangan, melihat riwayat pengajuan dan membatalkan pengajuan

Tabel 4.8 *Use Case* Diagram Petugas

***Use Case* Diagram**

Deskripsi
Pada *use case* diagram di atas menjelaskan representasi dari setiap *use case* perangkat lunak yang akan dibangun. Terdapat aktor yaitu petugas yang berinteraksi dengan setiap *use case* perangkat lunak diatas. Terdiri dari 7 *use case*, diantaranya petugas dapat mengelola peminjam, memverifikasi peminjam, mengelola ruangan, mengelola jadwal ruangan, mengelola pengajuan, memverifikasi pengajuan dan mengelola laporan.

4.2.4.1. Deskripsi Use Case

Diagram *use case* digunakan untuk menggambarkan interaksi yang terjadi antara pengguna dengan sistem perangkat lunak yang dikembangkan pada gambar 4.3 dan 4.4 dapat dijelaskan lebih detail. Penjelasan *use case* pada gambar 4.3 dan gambar 4.4 dapat dilihat pada tabel 4.9 dan tabel 4.10.

Tabel 4.9 Deskripsi *Use Case* Peminjam

No	Kode	*Use Case*	Deskripsi
1.	UCP-01	Mendaftar	*Use case* mendaftar digunakan untuk peminjam pada saat melakukan pendaftaran
2.	UCP-02	Melihat profil	*Use case* melihat profil digunakan untuk memperlihatkan detail profil peminjam
3.	UCP-03	Mengubah profil	*Use case* mengubah profil digunakan untuk mengubah data profil peminjam
4.	UCP-04	Melihat informasi gedung	*Use case* melihat informasi gedung digunakan untuk memperlihatkan informasi-informasi gedung seperti profil gedung, informasi ruangan, jadwal ruangan, prosedur pengajuan, dan kontak.
5.	UCP-05	Mengajukan peminjaman ruangan	*Use case m*engajukan peminjaman ruangan digunakan untuk melaksanakan proses pengajuan peminjaman

6.	UCP-06	Melihat riwayat pengajuan	*Use case* melihat riwayat pengajuan digunakan untuk memperlihatkan riwayat pengajuan peminjaman yang sudah diajukan
7.	UCP-07	Membatalkan pengajuan	*Use case* membatalkan pengajuan digunakan untuk melakukan pembatalan pengajuan peminjaman yang sudah dilakukan

Tabel 4.10 Deskripsi Use Case Petugas

No	Kode	*Use Case*	Deskripsi
1.	UCA-01	Mengelola peminjam	*Use case* mengelola peminjam digunakan untuk melakukan pengelolaan data peminjam
2.	UCA-02	Memverifikasi peminjam	*Use case* memverifikasi peminjam digunakan untuk melakukan verifikasi data peminjam
3.	UCA-03	Mengelola ruangan	*Use case* mengelola ruangan digunakan untuk melakukan pengelolaan data ruangan
4.	UCA-04	Mengelola jadwal ruangan	*Use case* mengelola jadwal ruangan digunakan untuk melakukan pengelolaan data jadwal ruangan
5.	UCA-05	Mengelola pengajuan	*Use case* mengelola pengajuan digunakan untuk melakukan pengelolaan data pengajuan
6.	UCA-06	Memverifikasi pengajuan	*Use case* memverifikasi pengajuan digunakan untuk melakukan verifikasi data pengajuan peminjam
7.	UCA-07	Mengelola laporan	*Use case* mengelola laporan digunakan untuk melakukan pengelolaan data laporan penggunaan ruangan

4.2.4.2. Skenario Use Case

Skenario *use case* digunakan untuk menggambarkan berjalannya alur proses dari sisi sistem dan aktor berdasarkan *use case* yang telah dibuat. Skenario *use case* aplikasi peminjaman di Bandung Creative Hub dapat dilihat pada tabel 4.11 sampai dengan tabel 4.24.

Tabel 4.11 *Use Case* Mendaftar

Identifikasi		
No	UCP-01	
Nama	Mendaftar	
Tujuan	Untuk mendaftarkan peminjam ke sistem	
Deskripsi	Mendaftar adalah *use case* yang digunakan untuk mendaftarkan data peminjam ke sistem	
Aktor	Peminjam	
Skenario Utama		
Kondisi Awal	Halaman pendaftaran sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor berada pada halaman pendaftaran		2. Sistem menampilkan form pendaftaran
3. Aktor mengisi form pendaftaran		4. Sistem memvalidasi input data untuk memastikan semua field terisi dengan benar dan sesuai format
5. Aktor menyimpan pendaftaran		6. Sistem memeriksa apakah data yang dimasukkan sudah ada
		7. Sistem menyimpan data pendaftaran
		8. Sistem menampilkan pesan sukses
		9. Sistem mengarahkan aktor ke halaman utama
Kondisi Akhir	Sistem berhasil menyimpan data pendaftaran	
Skenario Alternatif		
Kondisi Awal	Halaman pendaftaran sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor berada pada halaman pendaftaran		2. Sistem menampilkan form pendaftaran
3. Aktor mengisi form pendaftaran		4. Sistem memvalidasi input data untuk memastikan semua field terisi dengan benar dan sesuai format
		5. Sistem menemukan kesalahan pada input data
		6. Sistem menampilkan pesan kesalahan
7. Aktor memperbaiki data yang salah		
8. Aktor menyimpan pendaftaran		9. Sistem memeriksa apakah data yang dimasukkan sudah ada
		10. Sistem menyimpan data pendaftaran
		11. Sistem menampilkan pesan sukses
		12. Sistem mengarahkan aktor ke halaman utama
Kondisi Akhir	Sistem berhasil menyimpan data pendaftaran	
Skenario Alternatif		
Kondisi Awal	Halaman pendaftaran sudah terbuka	

Aksi Aktor	Reaksi Sistem
1. Aktor berada pada halaman pendaftaran	2. Sistem menampilkan form pendaftaran
3. Aktor mengisi form pendaftaran	4. Sistem memvalidasi input data untuk memastikan semua field terisi dengan benar dan sesuai format
5. Aktor menyimpan pendaftaran	6. Sistem memeriksa apakah data yang dimasukkan sudah ada
	7. Sistem menemukan bahwa data (misalnya, email) sudah terdaftar
	8. Sistem menampilkan pesan bahwa email sudah digunakan
9. Aktor memperbaiki data yang salah	
10. Aktor menyimpan pendaftaran	11. Sistem memeriksa apakah data yang dimasukkan sudah ada
	12. Sistem menyimpan data pendaftaran
	13. Sistem menampilkan pesan sukses
	14. Sistem mengarahkan aktor ke halaman login
Kondisi Akhir	Sistem berhasil menyimpan data pendaftaran

Tabel 4.12 *Use Case* Melihat Profil

Identifikasi	
No	UCP-02
Nama	Melihat profil
Tujuan	Untuk dapat menampilkan detail profil peminjam
Deskripsi	Melihat profil adalah u*se case* yang digunakan untuk menampilkan detail profil peminjam jika peminjam sudah login
Aktor	Peminjam
Skenario Utama	
Kondisi Awal	Halaman utama sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu profil pada *navbar*	
	2. Sistem menampilkan halaman detail profil
Kondisi Akhir	Sistem berhasil menampilkan halaman detail profil

Tabel 4.13 *Use Case* Mengubah Profil

Identifikasi	
No	UCP-03
Nama	Mengubah profil
Tujuan	Untuk mengubah data profil
Deskripsi	Mengubah profil adalah u*se case* yang digunakan untuk mengubah data profil peminjam
Aktor	Peminjam
Skenario Utama	
Kondisi Awal	Halaman detail profil sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ubah data	2. Sistem menampilkan form ubah data profil
3. Aktor mengubah data pada form	
4. Aktor menyimpan data profil	5. Sistem memvalidasi data yang diubah untuk memastikan tidak ada kesalahan
	6. Sistem menyimpan perubahan data
	7. Sistem menampilkan pesan data berhasil diubah
Kondisi Akhir	Sistem berhasil menyimpan perubahan data profil
Skenario Alternatif	
Kondisi Awal	Halaman detail profil sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor menekan icon ubah data	2. Sistem menampilkan form ubah data profil
3. Aktor mengubah data pada form	
4. Aktor menyimpan data profil	5. Sistem memvalidasi data yang diubah untuk memastikan tidak ada kesalahan
	6. Sistem menemukan bahwa ada data yang tidak valid
	7. Sistem menampilkan pesan kesalahan
8. Aktor memperbaiki data yang salah	
9. Aktor menyimpan perubahan data	10. Sistem memvalidasi data yang diubah untuk memastikan tidak ada kesalahan
	11. Sistem menyimpan perubahan data
	12. Sistem menampilkan pop up data berhasil diubah
Kondisi Akhir	Sistem berhasil menyimpan perubahan data profil

Tabel 4.14 *Use Case* Melihat Informasi Gedung

Identifikasi		
No	UCP-04	
Nama	Melihat informasi gedung	
Tujuan	Untuk menampilkan informasi-informasi terkait gedung	
Deskripsi	Melihat informasi gedung adalah *use case* yang digunakan untuk menampilkan informasi-informasi gedung seperti profil gedung, informasi ruangan, jadwal ruangan, prosedur pengajuan, dan kontak.	
Aktor	Peminjam	
Skenario Utama		
Kondisi Awal	Halaman utama sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih informasi yang ingin dilihat melalui *navbar*		
		2. Sistem menampilkan informasi yang telah dipilih
Kondisi Akhir	Sistem menampilkan halaman informasi yang dipilih	

Tabel 4.15 *Use Case* Mengajukan Peminjaman Ruangan

Identifikasi	
No	UCP-05
Nama	Mengajukan peminjaman ruangan
Tujuan	Untuk melakukan proses pengajuan peminjaman ruangan
Deskripsi	Mengajukan peminjaman ruangan adalah *use case* yang digunakan oleh peminjam untuk melaksanakan proses pengajuan peminjaman ruangan
Aktor	Peminjam
Skenario Utama	
Kondisi Awal	Halaman pengajuan peminjaman sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor berada pada halaman pengajuan	2. Sistem menampilkan form pengajuan
3. Aktor mengisi form pengajuan	4. Sistem memvalidasi input data untuk memastikan semua field terisi dengan benar dan sesuai format.
5. Aktor menyimpan pengajuan peminjaman	6. Sistem mengecek ketersediaan ruangan
	7. Sistem menyimpan pengajuan peminjaman
	8. Sistem menampilkan pop up berhasil mengajukan peminjaman
	9. Sistem mengarahkan aktor ke halaman riwayat pengajuan
Kondisi Akhir	Sistem berhasil menyimpan data pengajuan peminjaman ruangan
Skenario Alternatif	
Kondisi Awal	Halaman pengajuan peminjaman sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor berada pada halaman pengajuan	2. Sistem menampilkan form pengajuan
3. Aktor mengisi form pengajuan	4. Sistem memvalidasi input data untuk memastikan semua field terisi dengan benar dan sesuai format.
	5. Sistem menemukan adanya field yang tidak valid
	6. Sistem menampilkan pesan kesalahan
7. Aktor memperbaiki data yang salah	
8. Aktor menyimpan pengajuan peminjaman	9. Sistem mengecek ketersediaan ruangan
	10. Sistem menyimpan pengajuan peminjaman
	11. Sistem menampilkan pop up berhasil mengajukan peminjaman
	12. Sistem mengarahkan aktor ke halaman riwayat pengajuan
Kondisi Akhir	Sistem berhasil menyimpan data pengajuan peminjaman ruangan
Skenario Alternatif	
Kondisi Awal	Halaman pengajuan peminjaman sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor berada pada halaman pengajuan	2. Sistem menampilkan form pengajuan
3. Aktor mengisi form pengajuan	4. Sistem memvalidasi input data untuk memastikan semua field terisi dengan benar dan sesuai format.
5. Aktor menyimpan pengajuan peminjaman	6. Sistem mengecek ketersediaan ruangan
	7. Sistem menemukan bahwa ruangan yang dipilih tidak tersedia
	8. Sistem menampilkan pesan kesalahan
9. Aktor merubah ruangan atau memilih waktu lain	
10. Aktor menyimpan pengajuan peminjaman	11. Sistem mengecek ketersediaan ruangan
	12. Sistem menyimpan pengajuan peminjaman
	13. Sistem menampilkan pop up berhasil mengajukan

		peminjaman
		14. Sistem mengarahkan aktor ke halaman riwayat pengajuan
Kondisi Akhir	Sistem berhasil menyimpan data pengajuan peminjaman ruangan	

Tabel 4.16 *Use Case* Melihat Riwayat Pengajuan

Identifikasi		
No	UCP-06	
Nama	Melihat riwayat pengajuan	
Tujuan	Untuk melihat riwayat pengajuan peminjaman ruangan yang diajukan oleh peminjam	
Deskripsi	Melihat status pengajuan adalah *use case* yang digunakan untuk melihat status dan detail dari pengajuan peminjaman yang sudah dilakukan oleh peminjam	
Aktor	Peminjam	
Skenario Utama		
Kondisi Awal	Halaman utama sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu riwayat pengajuan pada *navbar*		
		2. Sistem menampilkan halaman riwayat pengajuan
Kondisi Akhir	Sistem menampilkan halaman status pengajuan	

Tabel 4.17 Use Case Membatalkan Pengajuan

Identifikasi		
No	UCP-07	
Nama	Membatalkan pengajuan	
Tujuan	Untuk digunakan peminjam membatalkan pengajuan peminjaman ruangan yang telah diajukan	
Deskripsi	Membatalkan pengajuan adalah *use case* yang digunakan oleh peminjam untuk membatalkan permintaan peminjaman ruangan yang telah diajukan.	
Aktor	Peminjam	
Skenario Utama		
Kondisi Awal	Halaman riwayat pengajuan sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih pengajuan peminjaman yang ingin dibatalkan dari daftar riwayat pengajuan.		2. Sistem menampilkan konfirmasi untuk membatalkan pengajuan
3. Aktor mengkonfirmasi pembatalan		4. Sistem menyimpan perubahan status pengajuan menjadi dibatalkan
		5. Sistem menampilkan pesan pengajuan berhasil dibatalkan
Kondisi Akhir	Sistem berhasil membatalkan pengajuan peminjaman ruangan dan memperbarui status pengajuan.	

Tabel 4.18 *Use Case* Mengelola Peminjam

Identifikasi		
No	UCA-01	
Nama	Mengelola peminjam	
Tujuan	Untuk pengelolaan data peminjam	
Deskripsi	Mengelola peminjam adalah *use case* yang digunakan oleh petugas untuk melakukan pengelolaan data peminjam, termasuk melihat, menambah, mengubah, dan menghapus data peminjam	
Aktor	Petugas	
Skenario Utama		
Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu pengguna di *sidebar*		2. Sistem menampilkan halaman data peminjam secara keseluruhan
3. Aktor memilih data peminjam yang akan dilihat		4. Sistem menampilkan halaman detail data peminjam yang dipilih
Kondisi Akhir	Sistem berhasil menampilkan halaman detail data peminjam yang dipilih	
Skenario Alternatif		
Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu pengguna di *sidebar*		2. Sistem menampilkan halaman data peminjam secara keseluruhan
3. Aktor memilih menu tambah pengguna		4. Sistem menampilkan form inputan
5. Aktor mengisi form inputan		
6. Aktor menyimpan data		7. Sistem memvalidasi input data
		8. Sistem menyimpan data peminjam jika valid
		9. Sistem menampilkan pesan data berhasil ditambahkan
Kondisi Akhir	Sistem berhasil menyimpan data peminjam	

Skenario Alternatif		
Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu pengguna di *sidebar*		2. Sistem menampilkan halaman data peminjam secara keseluruhan
3. Aktor memilih data peminjam yang akan diubah		4. Sistem menampilkan form inputan untuk ubah data
5. Aktor mengisi form inputan		
6. Aktor menyimpan perubahan data		7. Sistem memvalidasi input data
		8. Sistem menyimpan perubahan data peminjam jika valid
		9. Sistem menampilkan pesan data berhasil diubah
Kondisi Akhir	Sistem berhasil menyimpan perubahan data peminjam	
Skenario Alternatif		
Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu pengguna di *sidebar*		2. Sistem menampilkan halaman data peminjam secara keseluruhan
3. Aktor memilih data peminjam yang akan dihapus		4. Sistem menampilkan konfirmasi dengan pesan "Apakah yakin ingin menghapus data ini"
5. Aktor mengkonfirmasi penghapusan data		6. Sistem menghapus data peminjam yang dipilih
		7. Sistem menampilkan pesan data berhasil dihapus
Kondisi Akhir	Sistem berhasil menghapus data peminjam yang dipilih	

Tabel 4.19 Use Case Memverifikasi Peminjam

Identifikasi	
No	UCA-02
Nama	Memverifikasi peminjam
Tujuan	Untuk verifikasi data identitas peminjam
Deskripsi	Memverifikasi peminjam adalah *use case* yang digunakan untuk melakukan verifikasi data identitas peminjam
Aktor	Petugas
Skenario Utama	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu di *sidebar*	2. Sistem menampilkan halaman data peminjam secara keseluruhan
3. Aktor memilih peminjam yang akan dilihat	4. Sistem menampilkan detail data peminjam
5. Aktor menekan tombol status *Terima*	6. Sistem menyimpan perubahan status verifikasi peminjam
	7. Sistem menampilkan pesan data berhasil diverifikasi
Kondisi Akhir	Sistem berhasil menyimpan perubahan status verifikasi peminjam
Skenario Alternatif	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu *pengguna* di *sidebar*	2. Sistem menampilkan halaman data peminjam secara keseluruhan
3. Aktor memilih peminjam yang akan dilihat	4. Sistem menampilkan detail data peminjam
5. Aktor menekan tombol status *Tolak*	6. Sistem menampilkan input untuk alasan penolakan
7. Aktor mengisi input alasan penolakan	
8. Aktor menyimpan data	9. Sistem menyimpan perubahan status verifikasi peminjam
	10. Sistem menampilkan pesan data berhasil diverifikasi
Kondisi Akhir	Sistem berhasil menyimpan perubahan status verifikasi peminjam

Tabel 4.20 *Use Case* Mengelola Ruangan

Identifikasi	
No	UCA-03
Nama	Mengelola ruangan
Tujuan	Untuk pengelolaan data ruangan
Deskripsi	Mengelola ruangan adalah *use case* yang digunakan untuk melakukan pengelolaan data ruangan
Aktor	Petugas
Skenario Utama	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu data ruangan	4. Sistem menampilkan halaman data ruangan secara keseluruhan

Kondisi Akhir	Sistem berhasil menampilkan halaman data ruangan
Skenario Alternatif	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu data ruangan	4. Sistem menampilkan halaman data ruangan secara keseluruhan
5. Aktor menu tambah ruangan	6. Sistem menampilkan form inputan
7. Aktor mengisi form inputan	
8. Aktor menyimpan data ruangan	9. Sistem memvalidasi input data
	10. Sistem menyimpan data ruangan jika valid
	11. Sistem menampilkan pesan data berhsil ditambahkan
Kondisi Akhir	Sistem berhasil menyimpan data ruangan
Skenario Alternatif	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu data ruangan	4. Sistem menampilkan halaman data ruangan secara keseluruhan
5. Aktor memilih data ruangan yang akan diedit	6. Sistem menampilkan form inputan untuk ubah data
7. Aktor mengisi form inputan	
8. Aktor menyimpan perubahan data	9. Sistem memvalidasi input data
	10. Sistem menyimpan perubahan data ruangan jika valid
	11. Sistem menampilkan pesan data berhasil diubah
Kondisi Akhir	Sistem berhasil menyimpan perubahan data ruangan
Skenario Alternatif	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu data ruangan	4. Sistem menampilkan halaman data ruangan secara keseluruhan
5. Aktor memilih data ruangan yang akan dihapus	6. Sistem menampilkan konfirmasi dengan pesan “Apakah yakin ingin menghapus data ini”
7. Aktor mengkonfirmasi penghapusan data	8. Sistem menghapus data ruangan yang dipilih
	9. Sistem menampilkan pesan data berhasil dihapus
Kondisi Akhir	Sistem berhasil menghapus data ruangan yang dipilih

Tabel 4.21 *Use Case* Mengelola Jadwal Ruangan

Identifikasi	
No	UCA-04
Nama	Mengelola jadwal ruangan
Tujuan	Untuk pengelolaan data jadwal ruangan
Deskripsi	Mengelola jadwal ruangan adalah *use case* yang digunakan untuk melakukan pengelolaan data jadwal ruangan
Aktor	Petugas
Skenario Utama	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu jadwal ruangan	4. Sistem menampilkan halaman data jadwal ruangan secara keseluruhan
Kondisi Akhir	Sistem berhasil menampilkan halaman jadwal ruangan
Skenario Alternatif	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu jadwal ruangan	4. Sistem menampilkan halaman data jadwal ruangan secara keseluruhan
5. Aktor memilih menu tambah jadwal	6. Sistem menampilkan form inputan
7. Aktor mengisi form inputan	
8. Aktor menyimpan data jadwal ruangan	9. Sistem memvalidasi input data
	10. Sistem menyimpan data jadwal ruangan jika valid
	11. Sistem menampilkan pesan data berhasil ditambahkan
Kondisi Akhir	Sistem berhasil menyimpan data jadwal ruangan
Skenario Alternatif	

Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu ruangan		2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu jadwal ruangan		4. Sistem menampilkan halaman data jadwal ruangan secara keseluruhan
5. Aktor memilih jadwal ruangan yang akan diedit		6. Sistem menampilkan form inputan ubah data
7. Aktor mengisi form inputan		
8. Aktor menyimpan perubahan data		9. Sistem memvalidasi input data
		10. Sistem menyimpan perubahan data jadwal ruangan jika valid
		11. Sistem menampilkan pesan data berhasil diubah
Kondisi Akhir	Sistem berhasil menyimpan perubahan data jadwal ruangan	
Skenario Alternatif		
Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu ruangan		2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu jadwal ruangan		4. Sistem menampilkan halaman data jadwal ruangan secara keseluruhan
5. Aktor memilih jadwal ruangan yang akan dihapus		6. Sistem menampilkan konfirmasi dengan pesan "Apakah yakin ingin menghapus data ini"
7. Aktor mengkonfirmasi penghapusan data		8. Sistem menghapus data ruangan yang dipilih
		9. Sistem menampilkan pesan data berhasil dihapus
Kondisi Akhir	Sistem berhasil menghapus jadwal ruangan yang dipilih	

Tabel 4.22 *Use Case* Mengelola Pengajuan

Identifikasi		
No	UCA-05	
Nama	Mengelola pengajuan	
Tujuan	Untuk pengelolaan data pengajuan	
Deskripsi	Mengelola pengajuan adalah *use case* yang digunakan untuk melakukan pengelolaan data pengajuan	
Aktor	Petugas	
Skenario Utama		
Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu ruangan		2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu pengajuan ruangan		4. Sistem menampilkan halaman data pengajuan secara keseluruhan
5. Aktor memilih data pengajuan yang akan dilihat		6. Sistem menampilkan halaman detail data pengajuan yang dipilih
Kondisi Akhir	Sistem berhasil menampilkan halaman detail pengajuan yang dipilih	
Skenario Alternatif		
Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu ruangan		2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu pengajuan ruangan		4. Sistem menampilkan halaman data pengajuan secara keseluruhan
5. Aktor memilih menu tambah pengajuan		6. Sistem menampilkan form inputan
7. Aktor mengisi form inputan		
8. Aktor menyimpan data pengajuan		9. Sistem memvalidasi input data
		10. Sistem menyimpan data pengajuan jika valid
		11. Sistem menampilkan pesan data berhasil ditambahkan
Kondisi Akhir	Sistem berhasil menyimpan data pengajuan ruangan	
Skenario Alternatif		
Kondisi Awal	Halaman dashboard sudah terbuka	
Aksi Aktor		**Reaksi Sistem**
1. Aktor memilih menu ruangan		2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu pengajuan ruangan		4. Sistem menampilkan halaman data pengajuan secara keseluruhan
5. Aktor memilih pengajuan ruangan yang akan diedit		6. Sistem menampilkan form inputan ubah data
7. Aktor mengisi form inputan		
8. Aktor menyimpan perubahan data		9. Sistem memvalidasi input data
		10. Sistem menyimpan perubahan data pengajuan jika valid
		11. Sistem menampilkan pesan data berhasil diubah
Kondisi Akhir	Sistem berhasil menyimpan perubahan data pengajuan ruangan	
Skenario Alternatif		
Kondisi Awal	Halaman dashboard sudah terbuka	

Aksi Aktor	Reaksi Sistem
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu pengajuan ruangan	4. Sistem menampilkan halaman data pengajuan secara keseluruhan
5. Aktor memilih pengajuan ruangan yang akan dihapus	6. Sistem menampilkan konfirmasi dengan pesan "Apakah yakin ingin menghapus data ini"
7. Aktor mengkonfirmasi penghapusan data	8. Sistem menghapus data pengajuan yang dipilih
	9. Sistem menampilkan pesan data berhasil dihapus
Kondisi Akhir	Sistem berhasil menghapus pengajuan ruangan yang dipilih

Tabel 4.23 Use Case Memverifikasi Pengajuan

Identifikasi	
No	UCA-06
Nama	Memverifikasi pengajuan
Tujuan	Untuk verifikasi data pengajuan peminjam
Deskripsi	Memverifikasi pengajuan adalah *use case* yang digunakan untuk melakukan verifikasi data pengajuan peminjam
Aktor	Petugas
Skenario Utama	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu pengajuan ruangan	4. Sistem menampilkan halaman data pengajuan secara keseluruhan
5. Aktor memilih pengajuan yang akan dilihat	6. Sistem menampilkan detail pengajuan
7. Aktor menekan tombol status *Terima*	8. Sistem menyimpan perubahan status verifikasi pengajuan
	9. Sistem menampilkan pesan data berhasil diverifikasi
Kondisi Akhir	Sistem berhasil menyimpan perubahan status pengajuan
Skenario Alternatif	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu ruangan	2. Sistem menampilkan sub menu ruangan
3. Aktor memilih sub menu pengajuan ruangan	4. Sistem menampilkan halaman data pengajuan secara keseluruhan
5. Aktor memilih pengajuan yang akan dilihat	6. Sistem menampilkan detail pengajuan
7. Aktor menekan tombol status *Tolak*	8. Sistem menampilkan input untuk alasan penolakan
9. Aktor mengisi input alasan penolakan	
10. Aktor menyimpan data	11. Sistem menyimpan perubahan status verifikasi pengajuan
	12. Sistem menampilkan pesan data berhasil diverifikasi
Kondisi Akhir	Sistem berhasil menyimpan perubahan status pengajuan

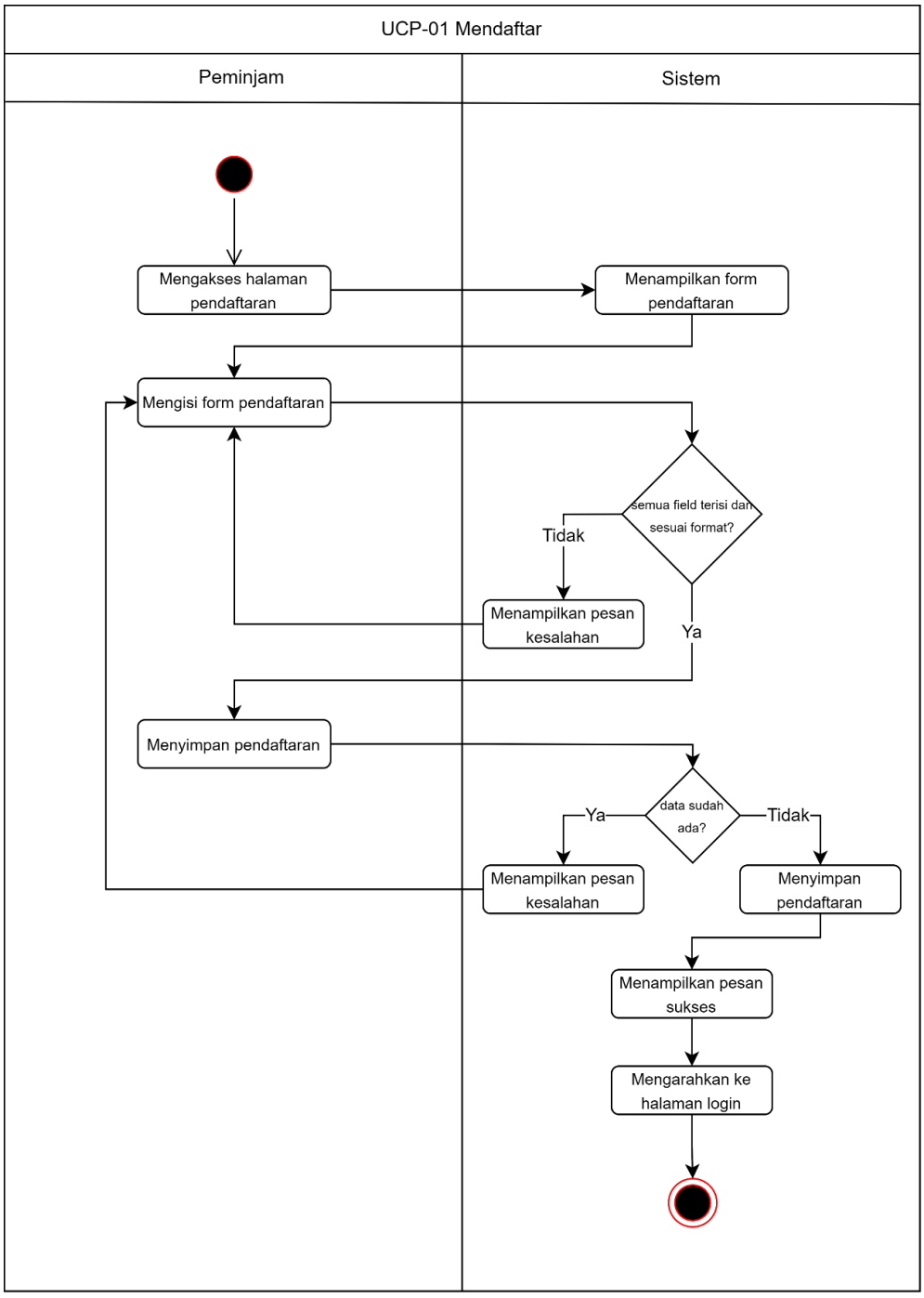
Tabel 4.24 *Use Case* Mengelola Laporan

Identifikasi	
No	UCA-07
Nama	Mengelola laporan
Tujuan	Untuk pengelolaan data laporan
Deskripsi	Mengelola laporan adalah *use case* yang digunakan untuk melakukan pengelolaan data laporan
Aktor	Petugas
Skenario Utama	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu laporan	2. Sistem menampilkan halaman laporan
3. Aktor memilih filter laporan untuk ditampilkan	4. Sistem menampilkan data laporan sesuai dengan filter yang dipilih
Kondisi Akhir	Sistem berhasil menampilkan halaman data laporan yang dipilih
Skenario Alternatif	
Kondisi Awal	Halaman dashboard sudah terbuka
Aksi Aktor	**Reaksi Sistem**
1. Aktor memilih menu laporan	2. Sistem menampilkan halaman laporan
3. Aktor memilih filter laporan untuk ditampilkan	4. Sistem menampilkan data laporan sesuai dengan filter yang dipilih
5. Aktor menekan tombol download laporan	6. Sistem memulai proses unduhan
	7. Sistem menghasilkan file laporan
Kondisi Akhir	Sistem berhasil menghasilkan data laporan

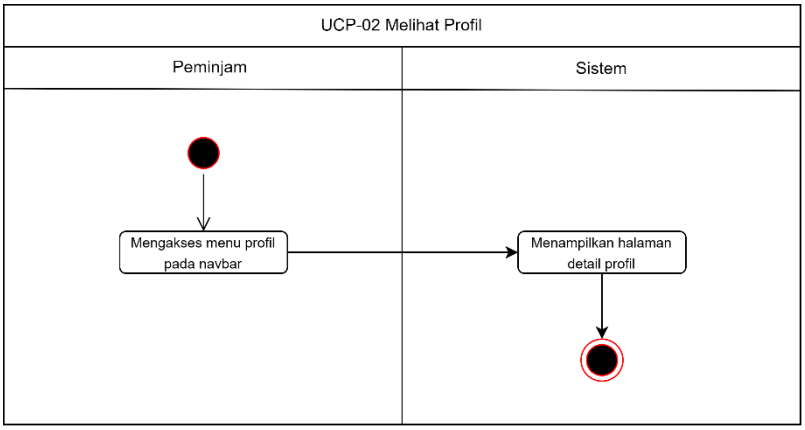
4.2.4.3. Aliran Aktivitas

Aliran Aktivitas yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. Activity diagram merupakan pengembangan dari *use case* yang memiliki alur aktivitas. Aliran aktivitas dapat dilihat pada tabel 4.25 sampai dengan tabel 4.38.

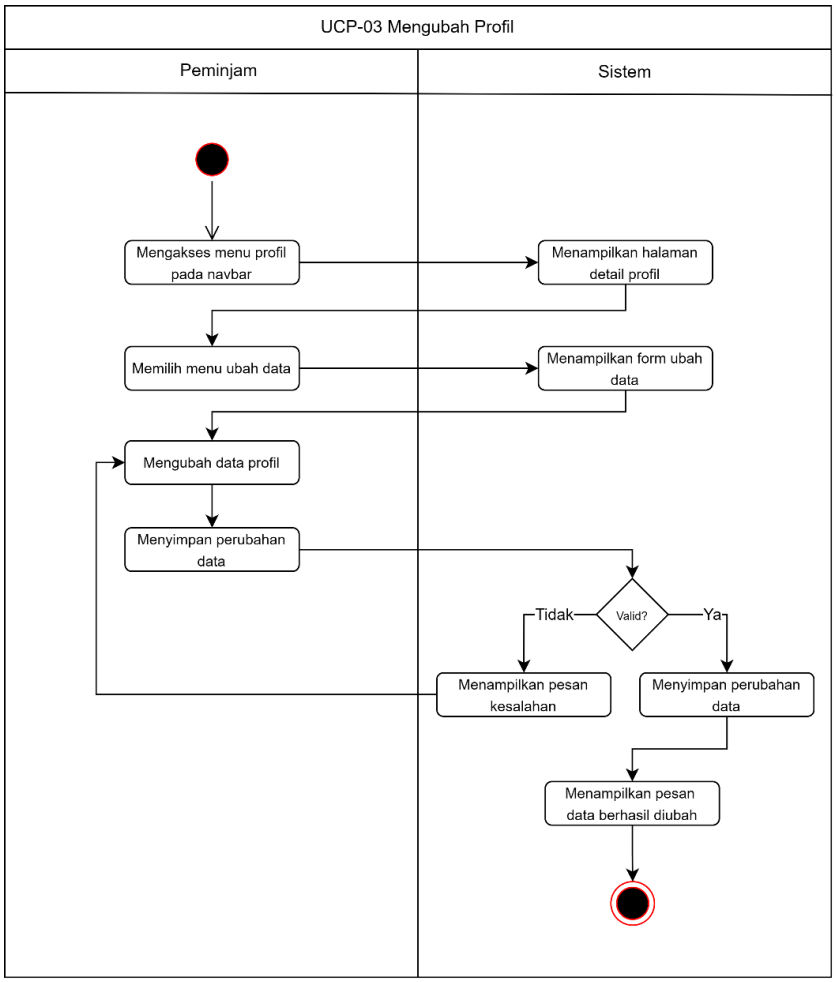
Tabel 4.25 Activity Diagram pada Use Case UCP-01

***Activity Diagram* pada *Use Case* UCP-01**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas mendaftar akun yang dilakukan oleh peminjam. Peminjam bisa mengakses halaman pendaftaran dan mengisi form pendaftaran lalu sistem akan memvalidasi apakah data sudah valid dan belum terdaftar sebelumnya jika valid, sistem akan menyimpan pendaftaran dan mengarahkan ke halaman login

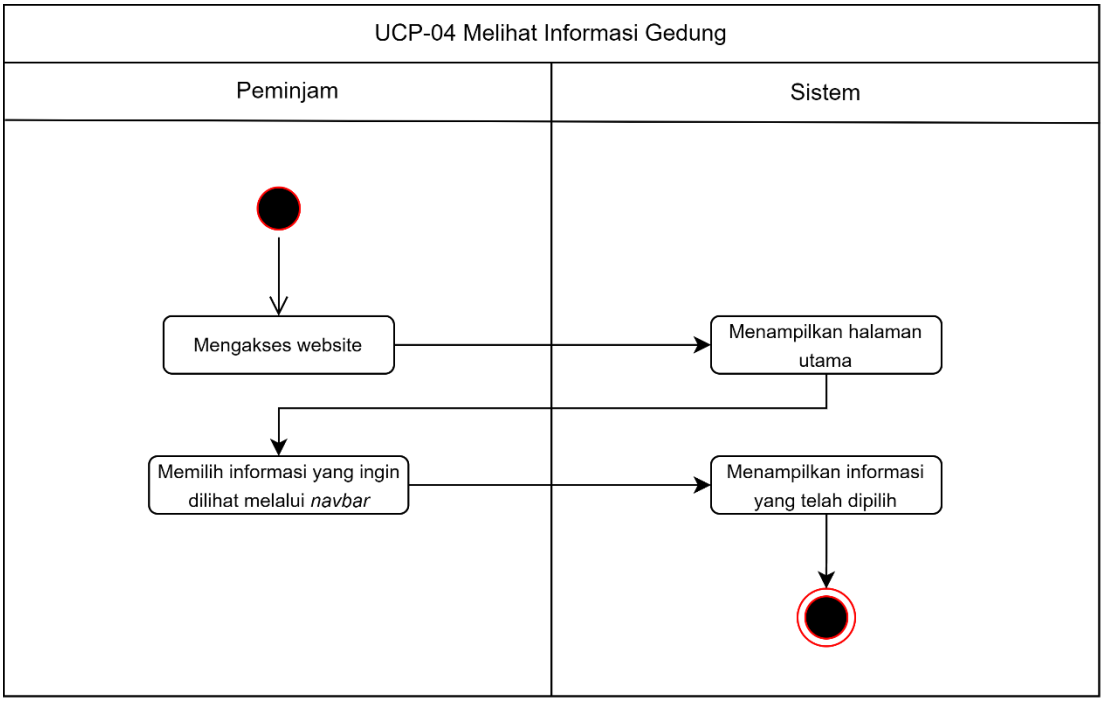
Tabel 4.26 Activity Diagram pada Use Case UCP-02

***Activity Diagram* pada *Use Case* UCP-02**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas melihat profil yang dilakukan oleh peminjam. Peminjam bisa mengakses menu profil pada *navbar*

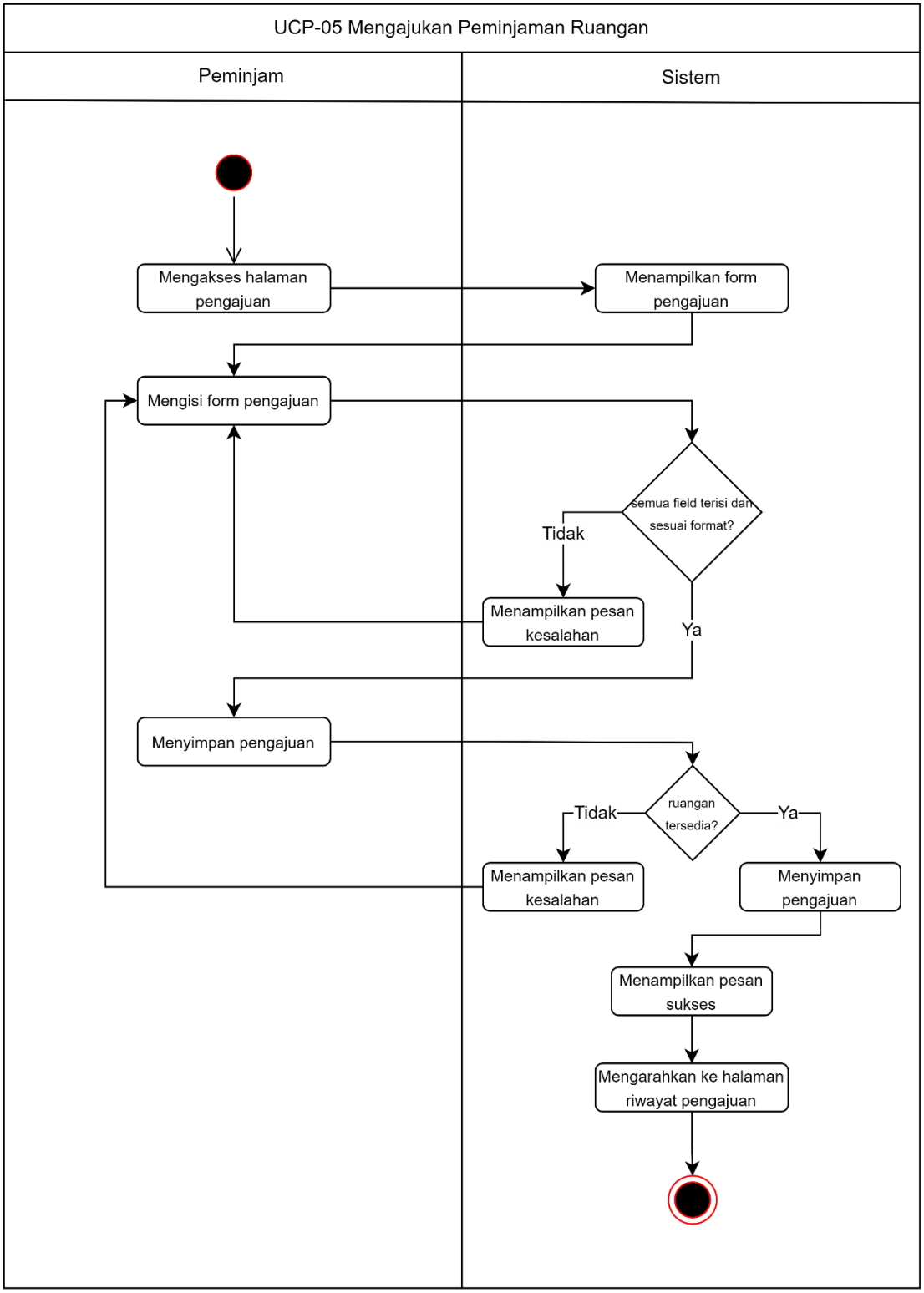
Tabel 4.27 Activity Diagram pada Use Case UCP-03

***Activity Diagram* pada *Use Case* UCP-03**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas mengubah profil yang dilakukan oleh peminjam. Peminjam bisa memilih menu ubah data pada halaman detail profil, setelah selesai sistem akan memvalidasi perubahan data tersebut.

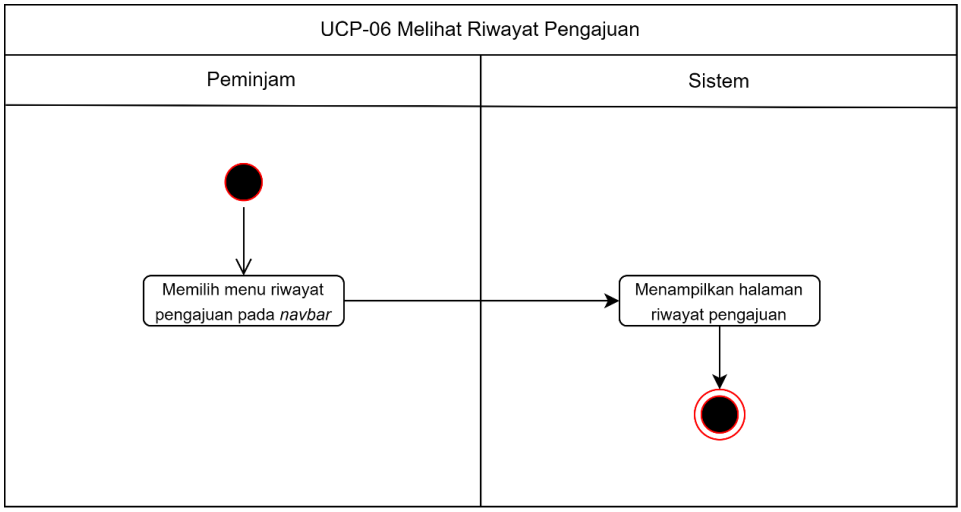
Tabel 4.28 Activity Diagram pada Use Case UCP-04

***Activity Diagram* pada *Use Case* UCP-04**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas melihat informasi gedung yang dilakukan oleh peminjam. Peminjam bisa memilih informasi yang ingin dilihat melalui *navbar*

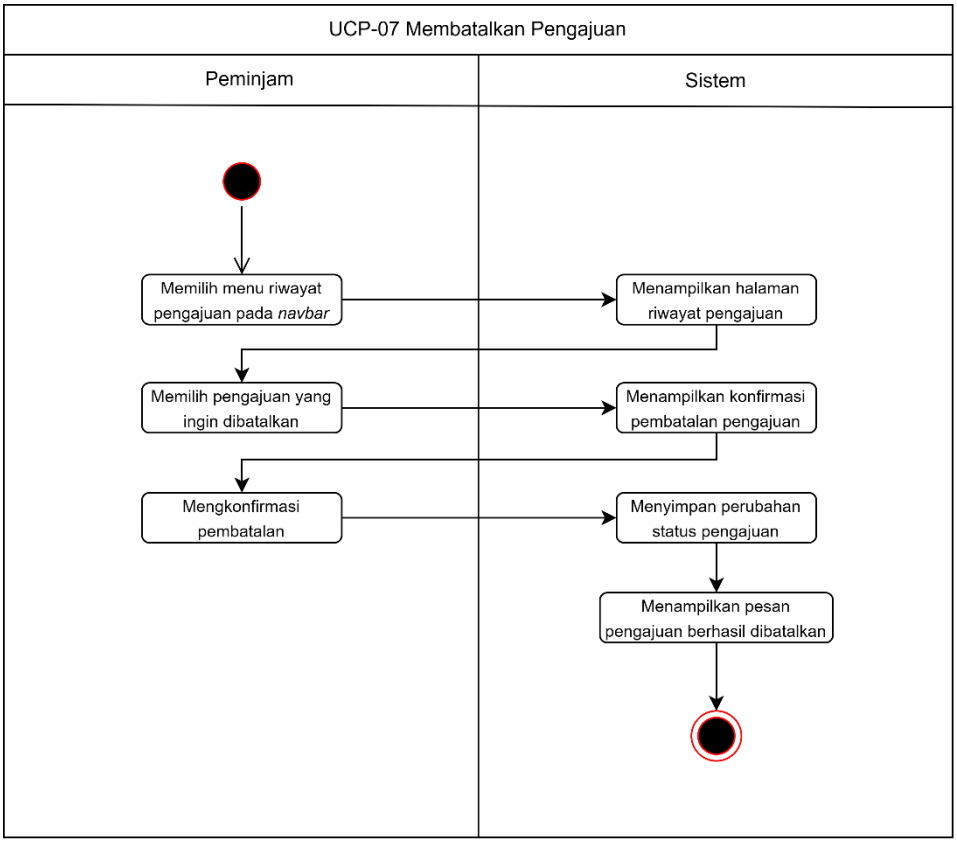
Tabel 4.29 Activity Diagram pada Use Case UCP-05

***Activity Diagram* pada *Use Case* UCP-05**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas mengajukan peminjaman ruangan yang dilakukan oleh peminjam. Peminjam bisa mengisi form pengajuan pada halaman pengajuan, setelah itu sistem akan memvalidasi apakah data sudah valid dan ruangan tersedia. Jika valid sistem akan menyimpan pengajuan dan mengarahkan ke halaman riwayat pengajuan

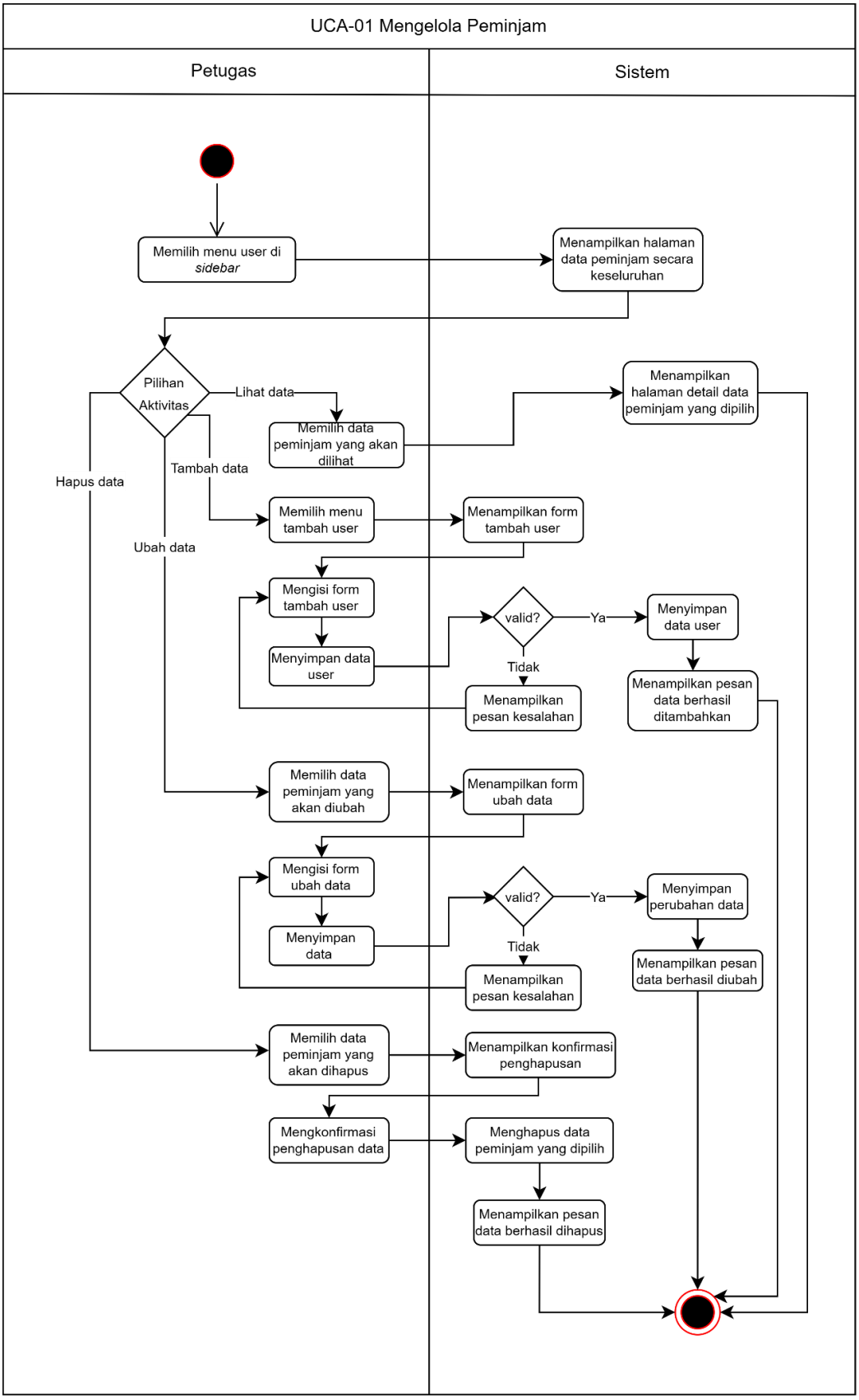
Tabel 4.30 Activity Diagram pada Use Case UCP-06

***Activity Diagram* pada *Use Case* UCP-06**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas melihat riwayat pengajuan yang dilakukan oleh peminjam. Peminjam bisa memilih menu riwayat pengajuan pada *navbar*

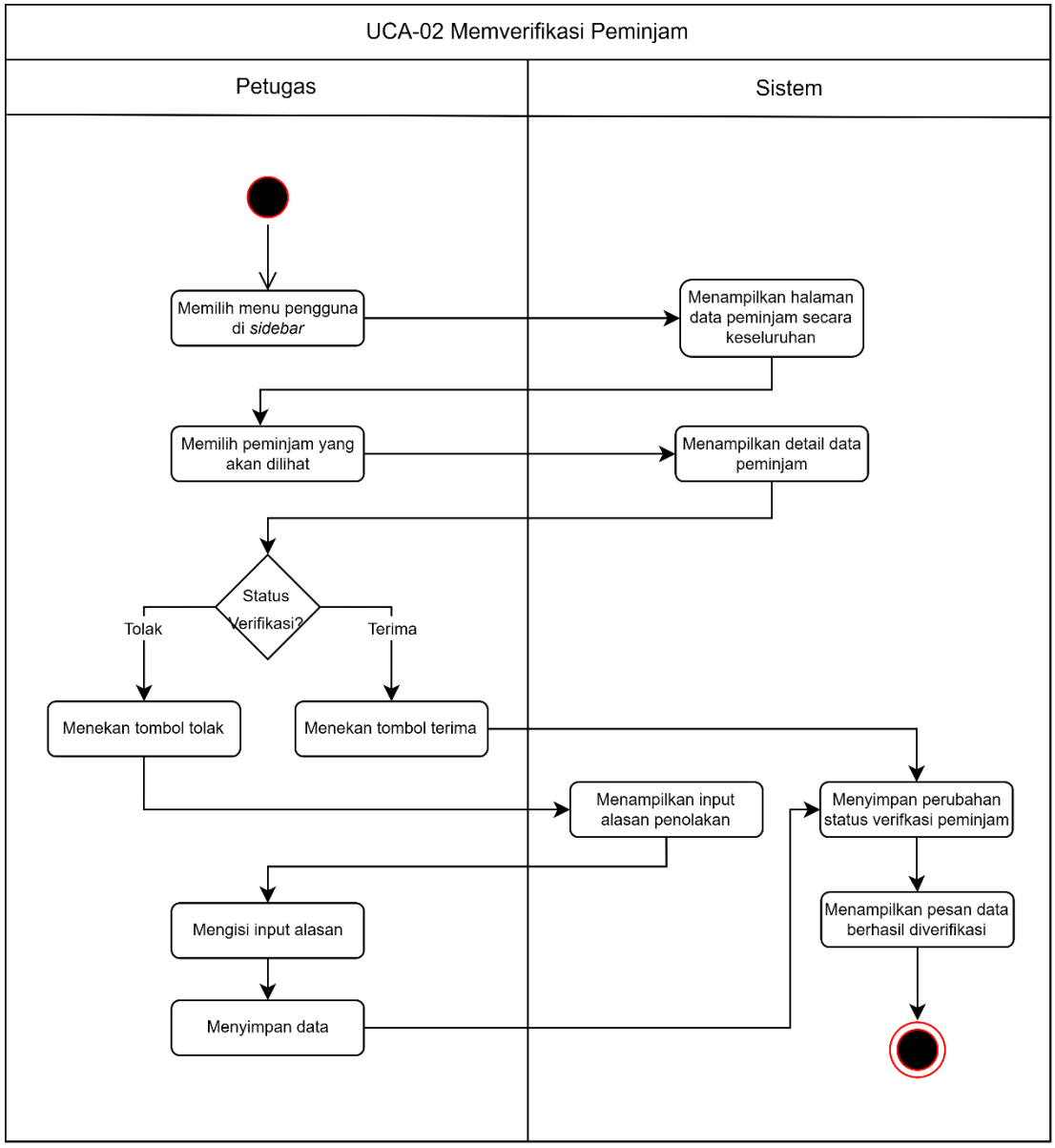
Tabel 4.31 Activity Diagram pada Use Case UCP-07

***Activity Diagram* pada *Use Case* UCP-07**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas membatalkan pengajuan yang dilakukan oleh peminjam. Peminjam bisa memilih pengajuan yang ingin dibatalkan pada halaman riwayat pengajuan, setelah itu sistem akan menyimpan perubahan status pengajuan menjadi dibatalkan

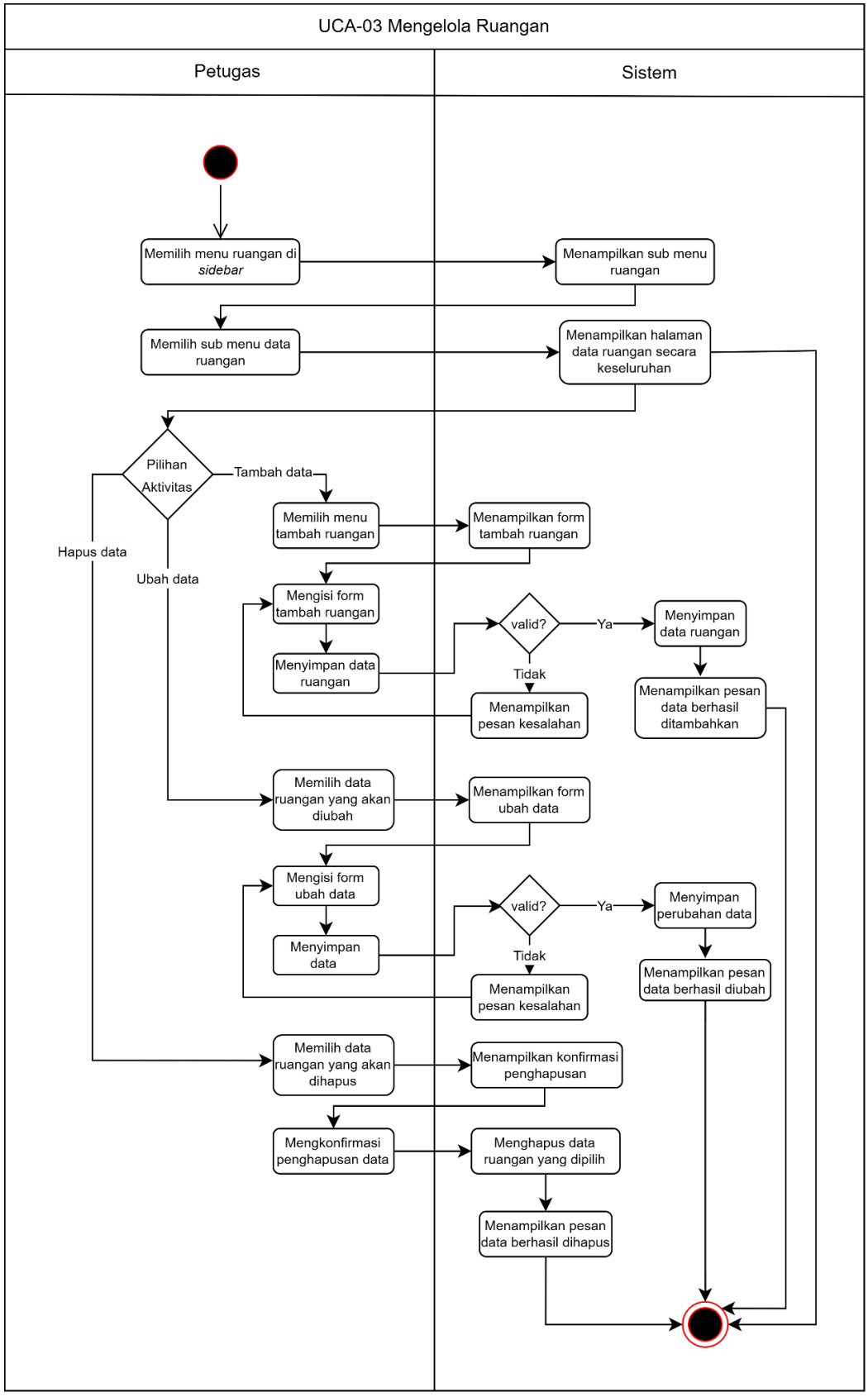
Tabel 4.32 Activity Diagram pada Use Case UCA-01

***Activity Diagram* pada *Use Case* UCA-01**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas yang dilakukan oleh petugas ketika akan mengolah data peminjam yang terdapat 4 aktivitas yaitu lihat data, tambah data, ubah data dan hapus data

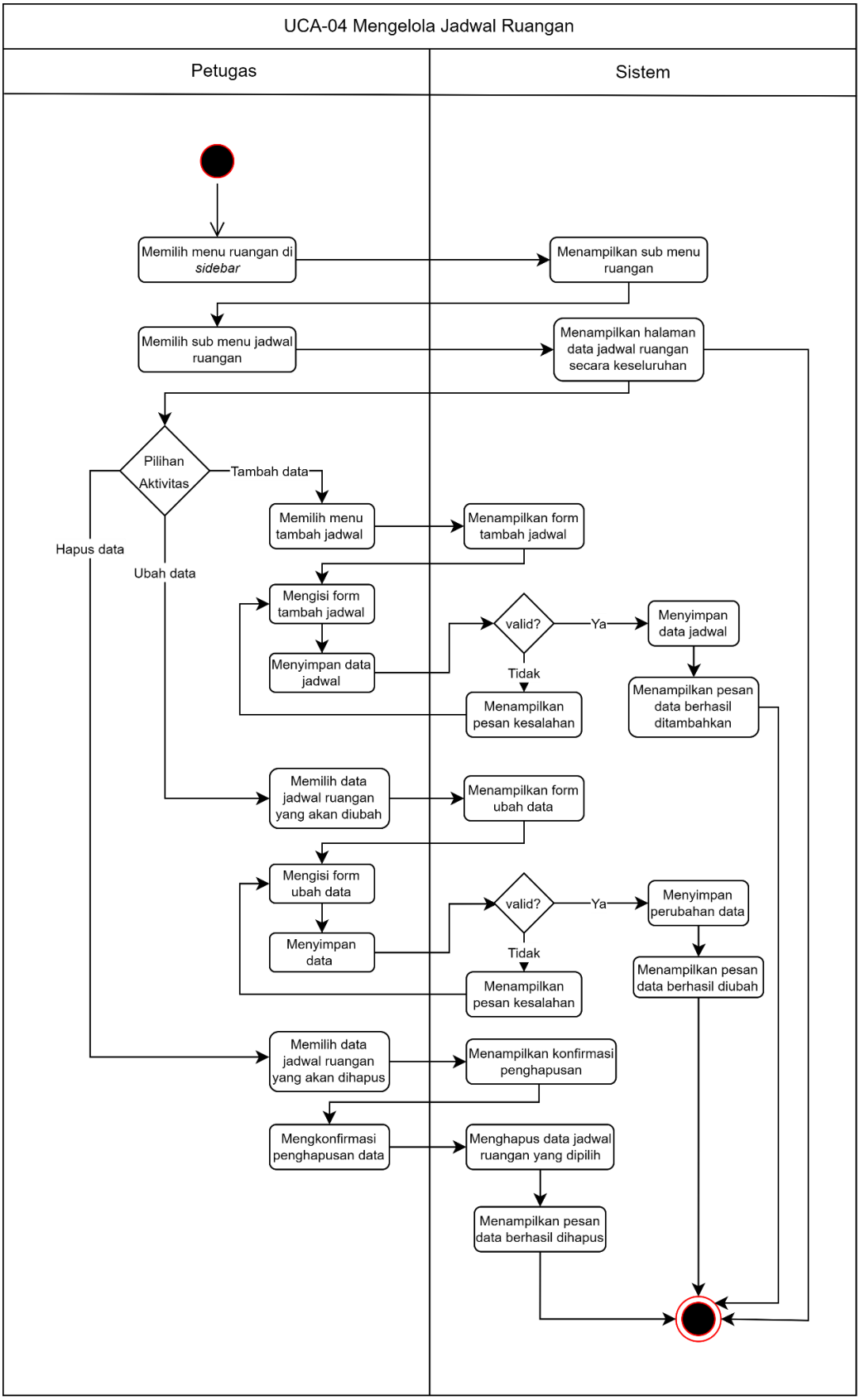
Tabel 4.33 Activity Diagram pada Use Case UCA-02

***Activity Diagram* pada *Use Case* UCA-02**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas memverifikasi peminjam yang dilakukan oleh petugas. Petugas bisa memverfikasi data peminjam pada halaman data peminjam, setelah itu sistem akan menyimpan perubahan status verifikasi peminjam

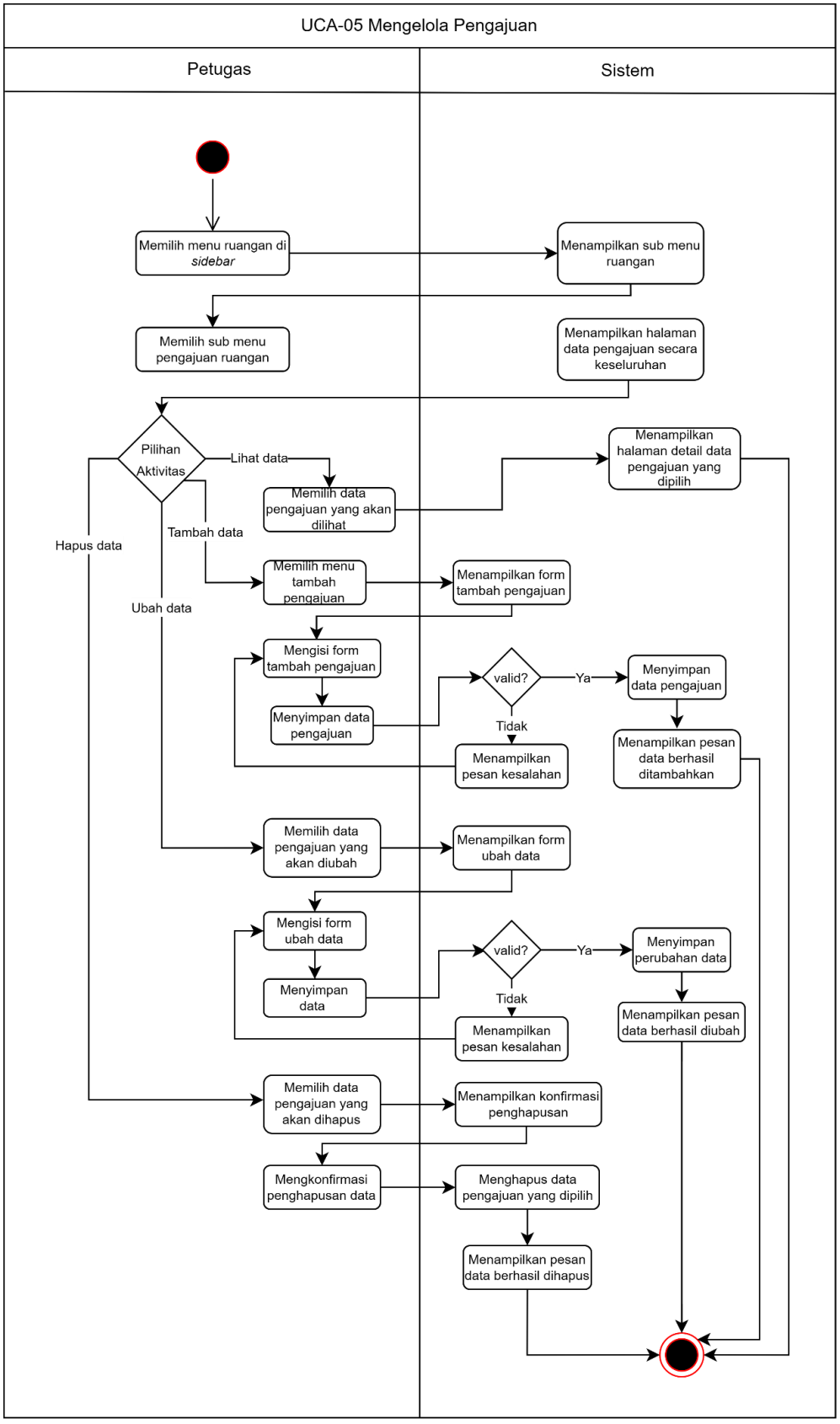
Tabel 4.34 Activity Diagram pada Use Case UCA-03

***Activity Diagram* pada *Use Case* UCA-03**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas yang dilakukan oleh petugas ketika akan mengelola data ruangan yang terdapat 4 aktivitas yaitu lihat data, tambah data, ubah data dan hapus data

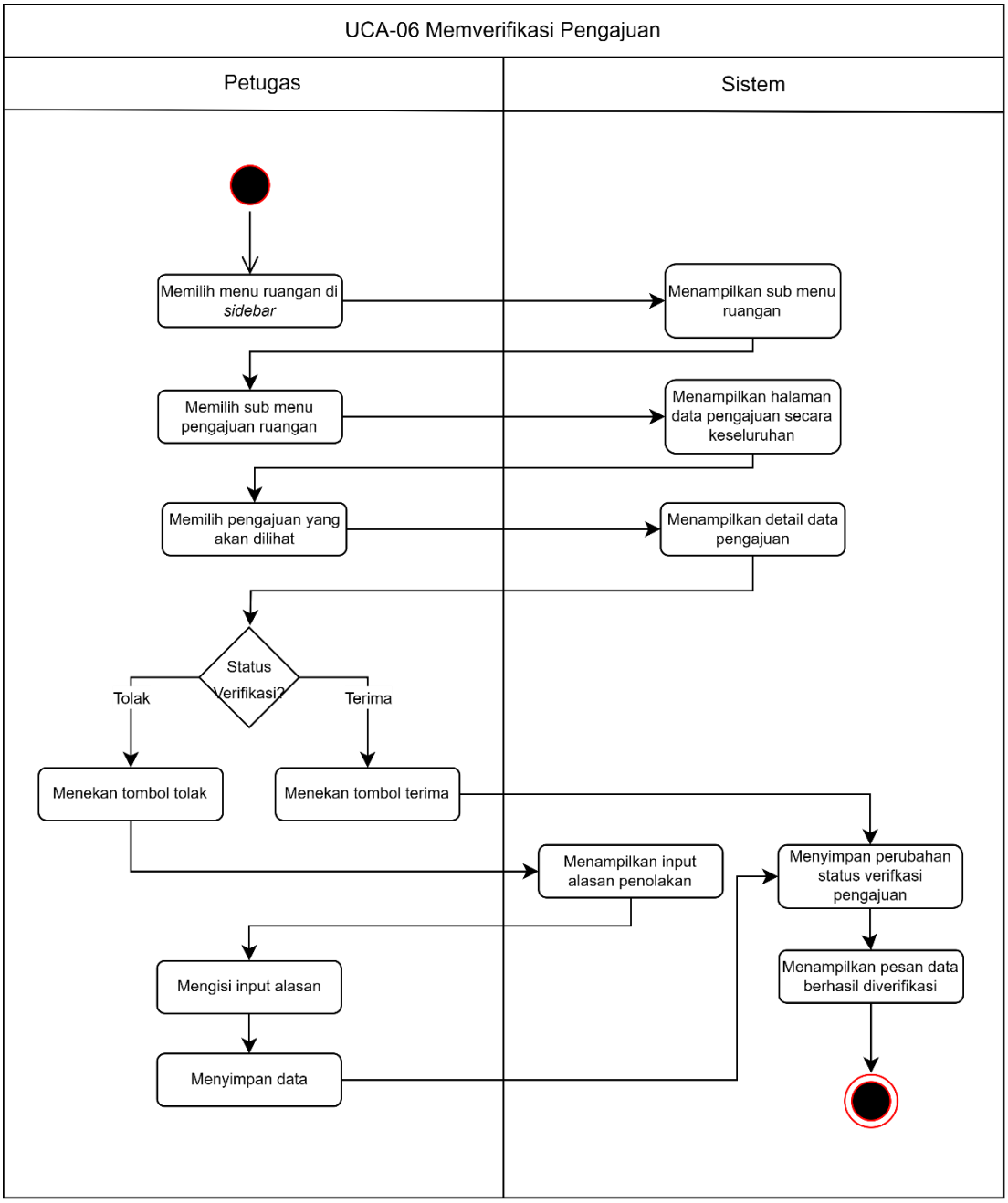
Tabel 4.35 Activity Diagram pada Use Case UCA-04

***Activity Diagram* pada *Use Case* UCA-04**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas yang dilakukan oleh petugas ketika akan mengelola data jadwal ruangan yang terdapat 4 aktivitas yaitu lihat data, tambah data, ubah data dan hapus data

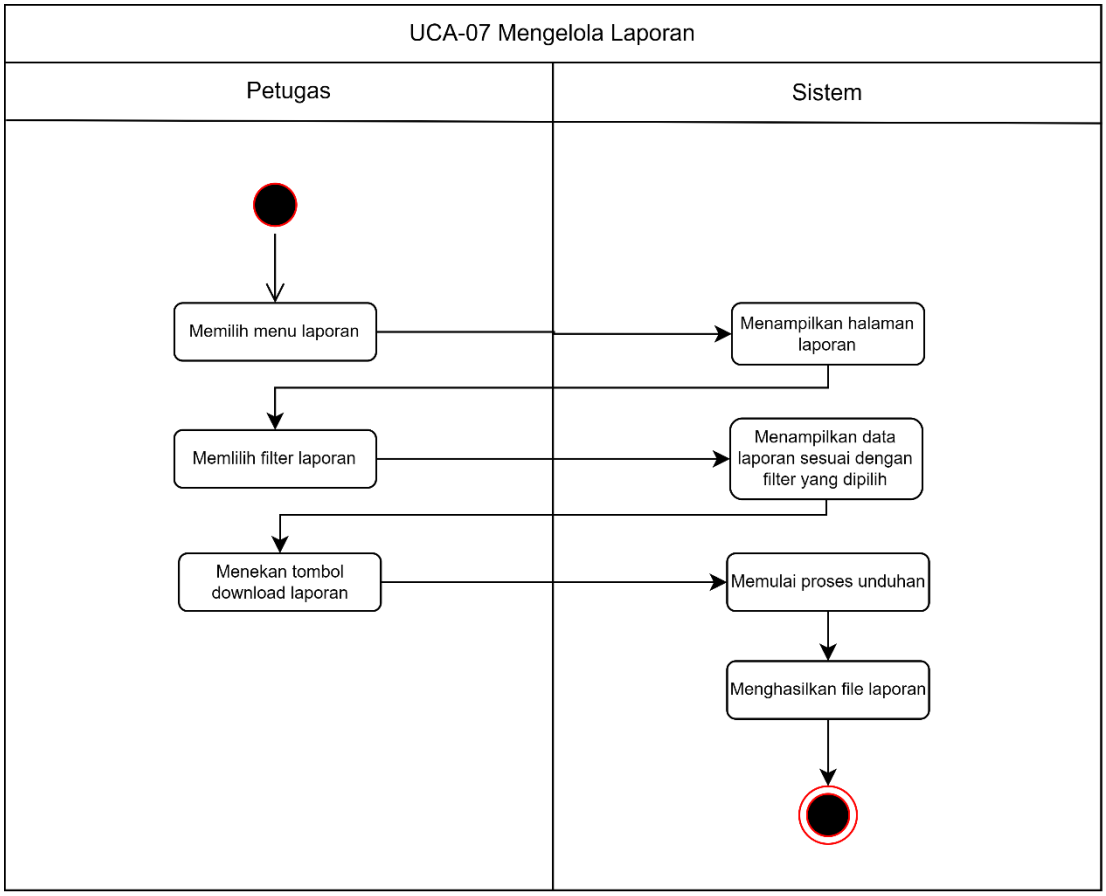
Tabel 4.36 Activity Diagram pada Use Case UCA-05

***Activity Diagram* pada *Use Case* UCA-05**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas yang dilakukan oleh petugas ketika akan mengelola data pengajuan yang terdapat 4 aktivitas yaitu lihat data, tambah data, ubah data dan hapus data

Tabel 4.37 Activity Diagram pada Use Case UCA-06

***Activity Diagram* pada *Use Case* UCA-06**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas memverifikasi pengajuan yang dilakukan oleh petugas. Petugas bisa memverfikasi data peminjam pada halaman data peminjam, setelah itu sistem akan menyimpan perubahan status verifikasi pengajuan

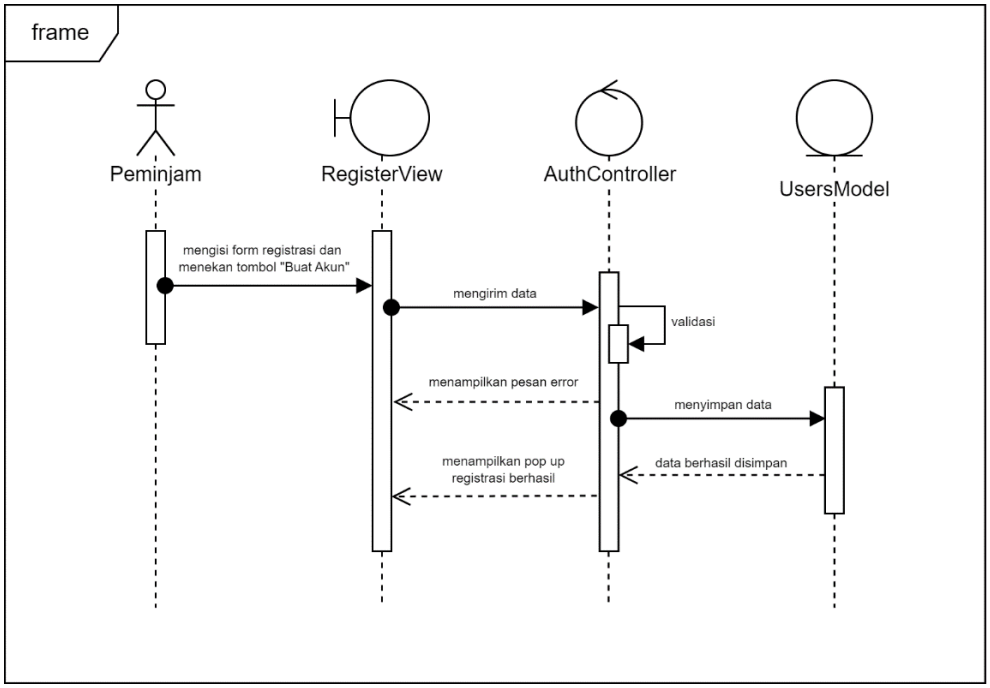
Tabel 4.38 Activity Diagram pada Use Case UCA-07

***Activity Diagram* pada *Use Case* UCA-07**

Deskripsi
Diagram aktivitas ini menjelaskan aliran aktivitas yang dilakukan oleh petugas ketika akan melihat dan mengunduh laporan penggunaan ruangan yang diperlukan. Petugas bisa memilih tanggal mulai dan akir laporan yang diiginkan

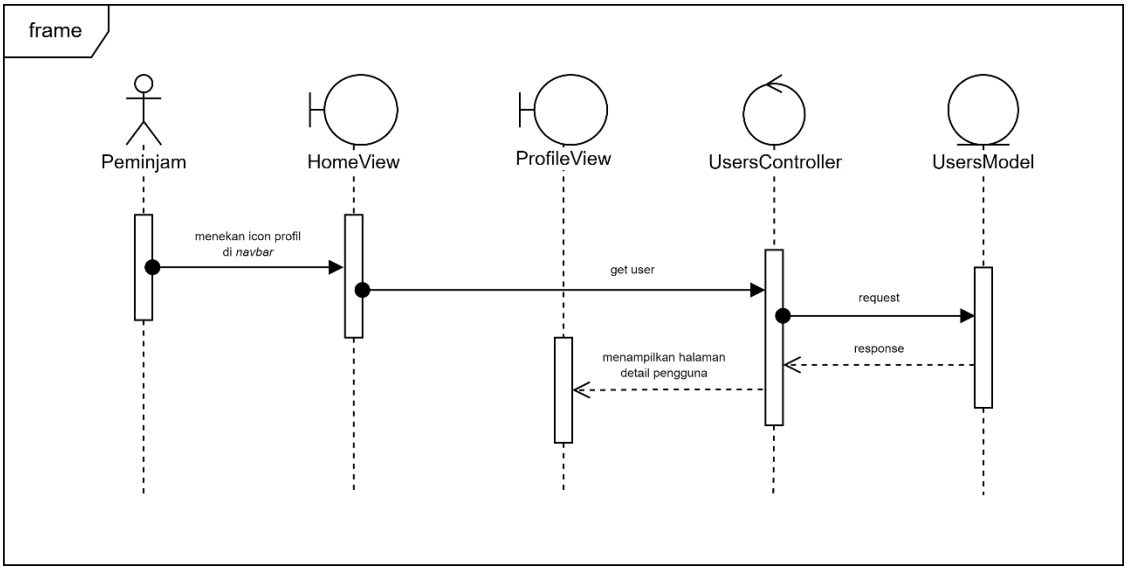
4.2.5. Pemodelan Berbasis Pelaku

Pemodelan berbasis pelaku akan digambarkan dengan *sequence* diagram. Dapat dilihat pada gambar 4.39 sampai tabel 4.52.

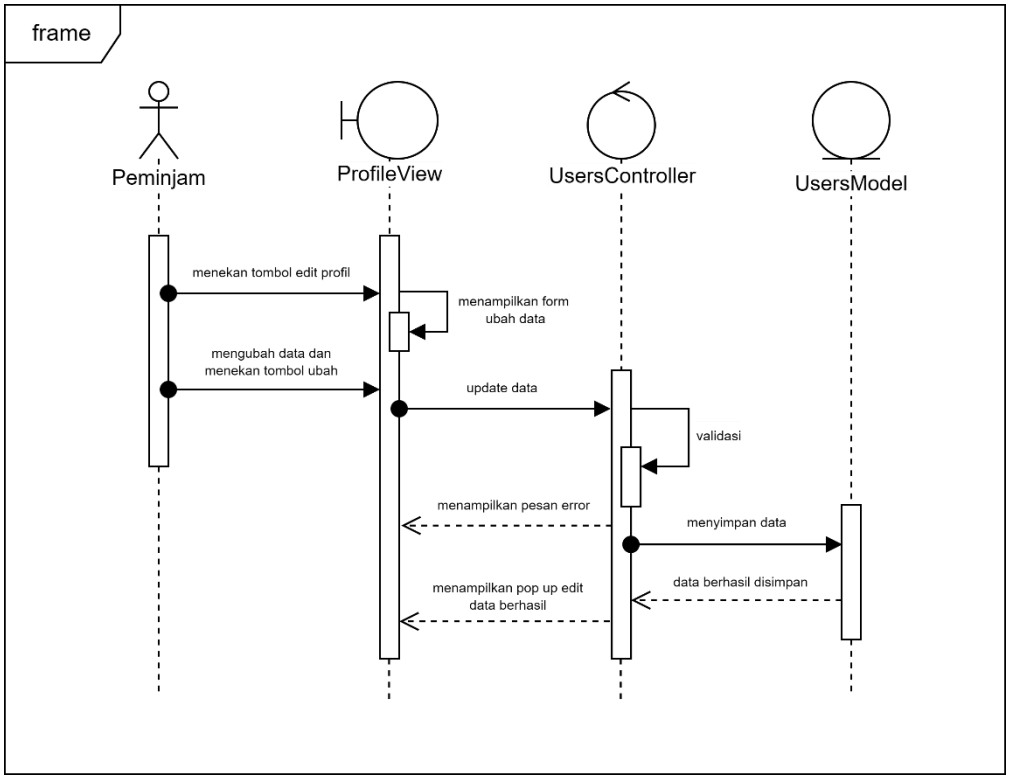
Tabel 4.39 Diagram *Sekuens* Mendaftar

Mendaftar

Deskripsi
Diagram ini menggambarkan bagaimana peminjam melakukan pendaftaran, dimulai dari peminjam mengisi form sampai sistem menyimpan data peminjam

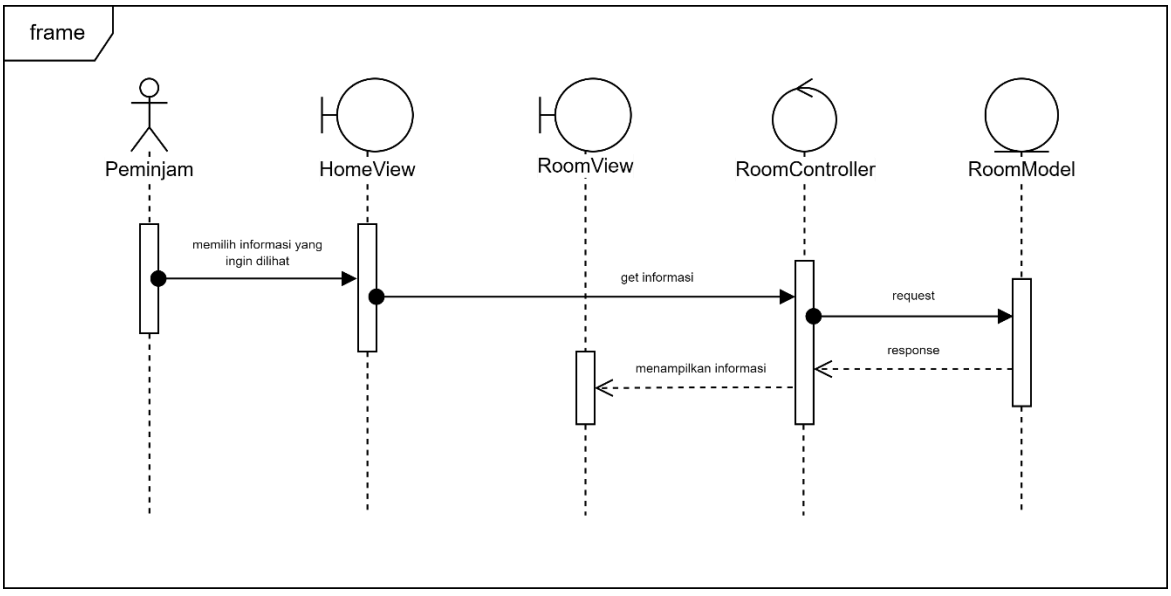
Tabel 4.40 Diagram *Sekuens* Melihat Profil

Melihat Profil

Deskripsi
Diagram ini menggambarkan bagaimana peminjam melihat profil, yaitu dimulai dari peminjam menekan icon profil pada *navbar* sampai sistem menampilkan detail profil dari peminjam

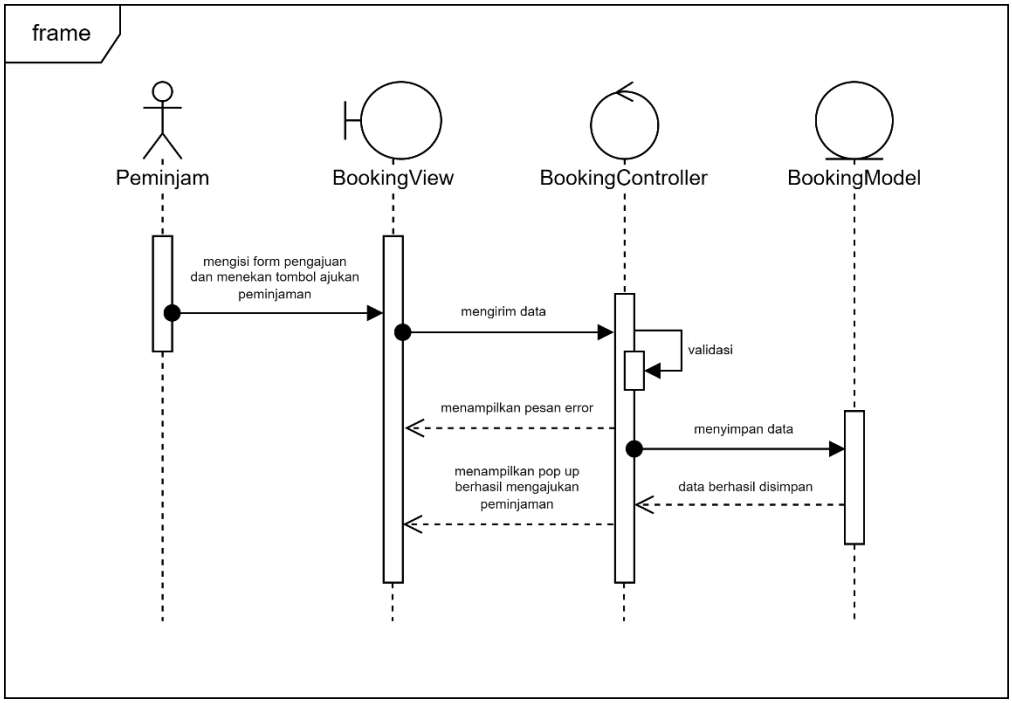
Tabel 4.41 Diagram *Sekuens* Mengubah Profil

Mengubah Profil

Deskripsi
Diagram ini menggambarkan bagaimana peminjam melakukan perubahan profil atau data diri, dimulai dari peminjam menekan tombol edit profil sampai sistem menampilkan pop up “edit data berhasil”

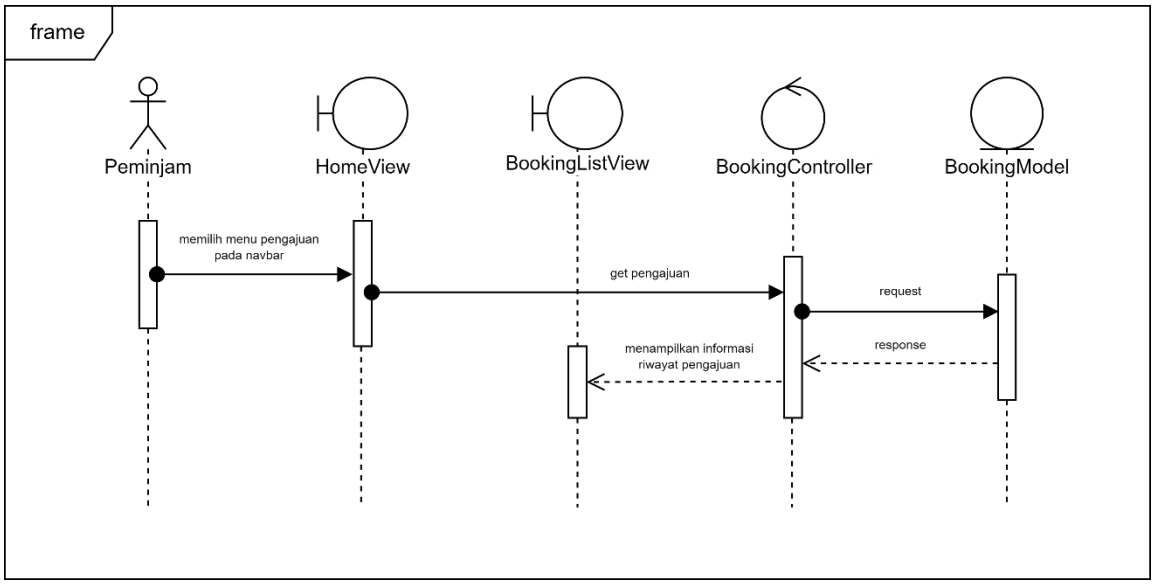
Tabel 4.42 Diagram *Sekuens* Melihat Informasi Gedung

Melihat Informasi Gedung

Deskripsi
Diagram ini menggambarkan bagaimana peminjam ingin melihat informasi gedung, dimulai dari peminjam memilih informasi yang ingin dilihat pada *navbar* sampai dengan sistem menampilkan informasi yang pilih

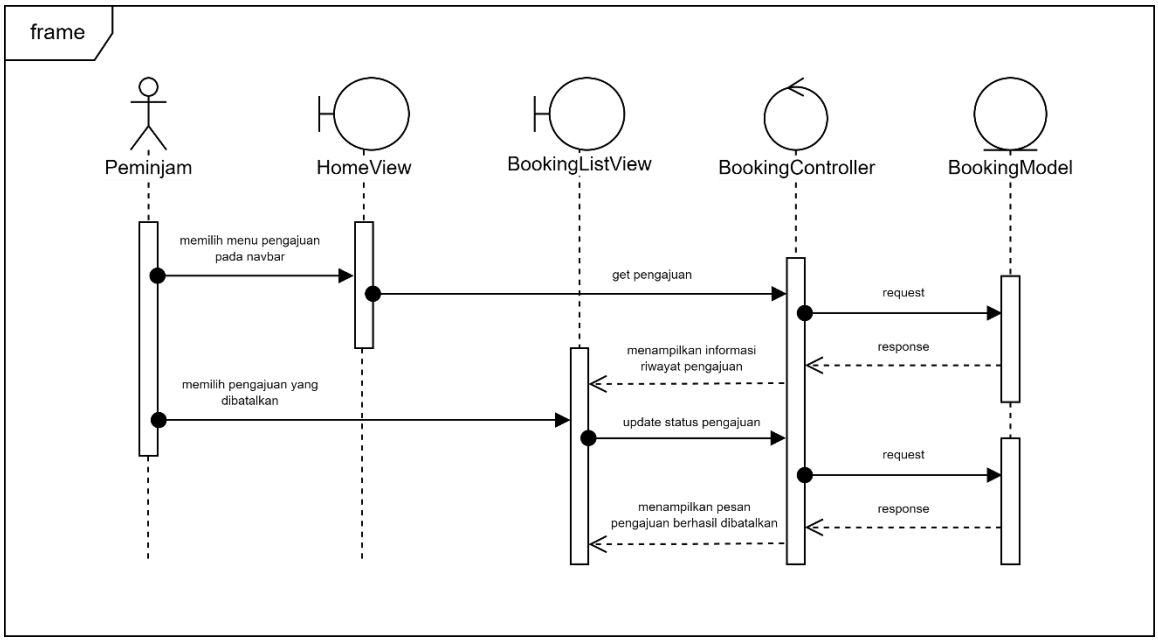
Tabel 4.43 Diagram *Sekuens* Mengajukan Peminjaman Ruangan

Melakukan Pengajuan Peminjaman

Deskripsi
Diagram ini menggambarkan bagaimana peminjam melakukan pengajuan peminjaman dimulai dari peminjam mengisi form pengajuan, sampai dengan sistem menyimpan data pengajuan dan menampilkan pop up berhasil mengajukan peminjaman

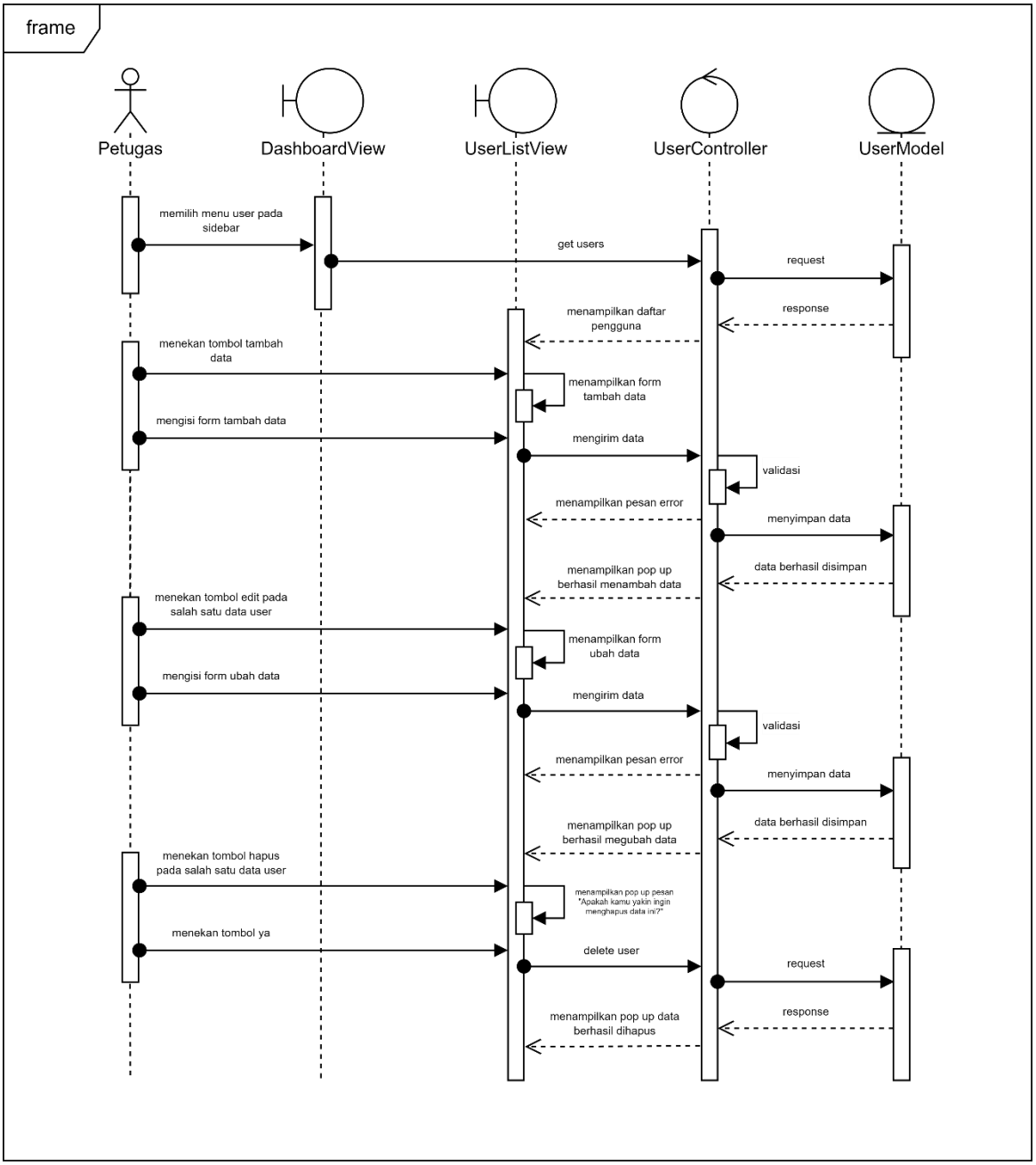
Tabel 4.44 Diagram *Sekuens* Melihat Riwayat Pengajuan

Melihat Status Pengajuan

Deskripsi
Diagram ini menggambarkan bagaimana peminjam melihat riwayat pengajuan yang telah diajukan, dimulai dari memilih menu pengajuan pada *navbar* sampai sistem menampilkan informasi riwayat pengajuan

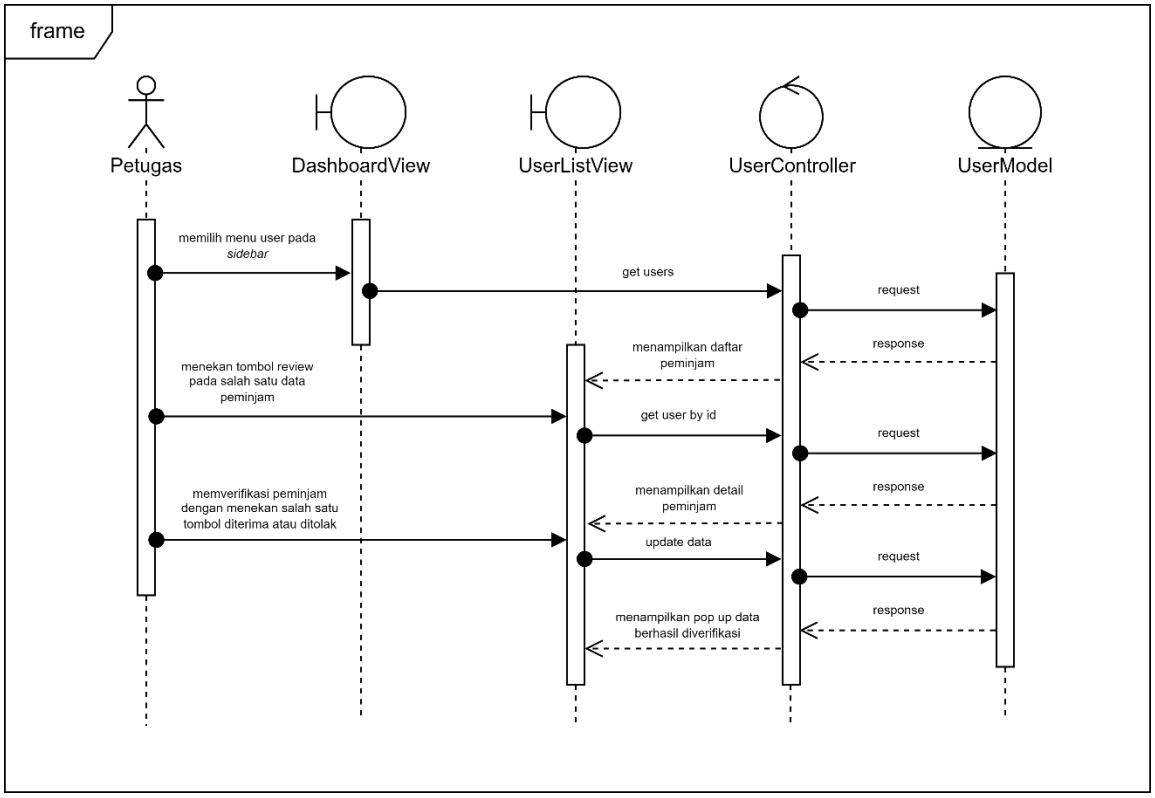
Tabel 4.45 Diagaram *Sekuens* Membatalkan Pengajuan

Membatalakan Pengajuan

Deskripsi
Diagram ini menggambarkan bagaimana peminjam membatalkan pengajuan yang telah diajukan, dimulai dari memilih menu pengajuan pada *navbar* lalu sistem menampilkan informasi riwayat pengajuan, setelah itu peminjam bisa memilih pengajuan yang akan dibatalkan

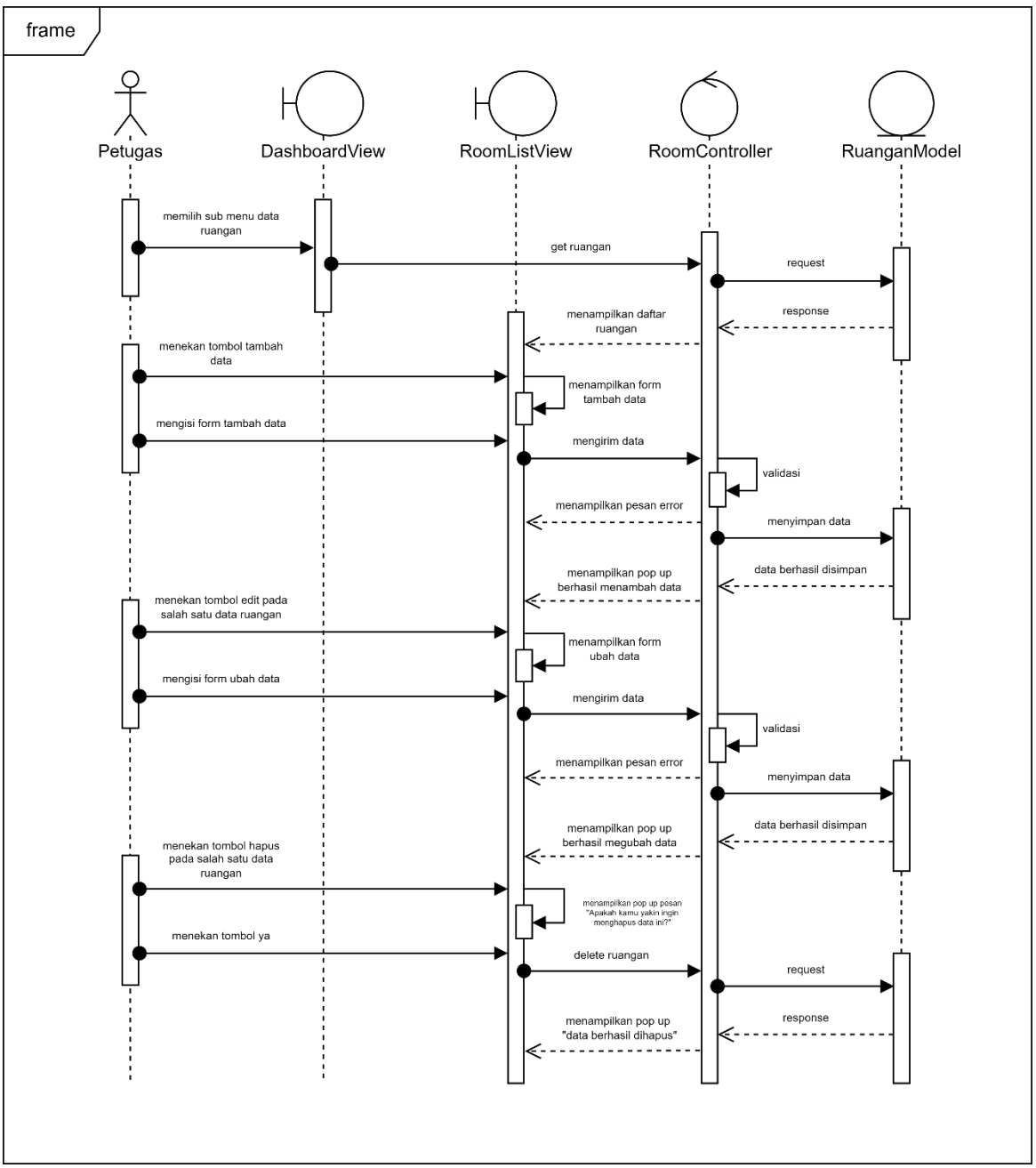
Tabel 4.46 Diagaram *Sekuens* Mengelola Peminjam

Mengelola Peminjam

Deskripsi
Diagram ini menggambarkan bagaimana petugas mengelola data peminjam, yaitu melihat data, menambah data, mengubah data, dan menghapus data

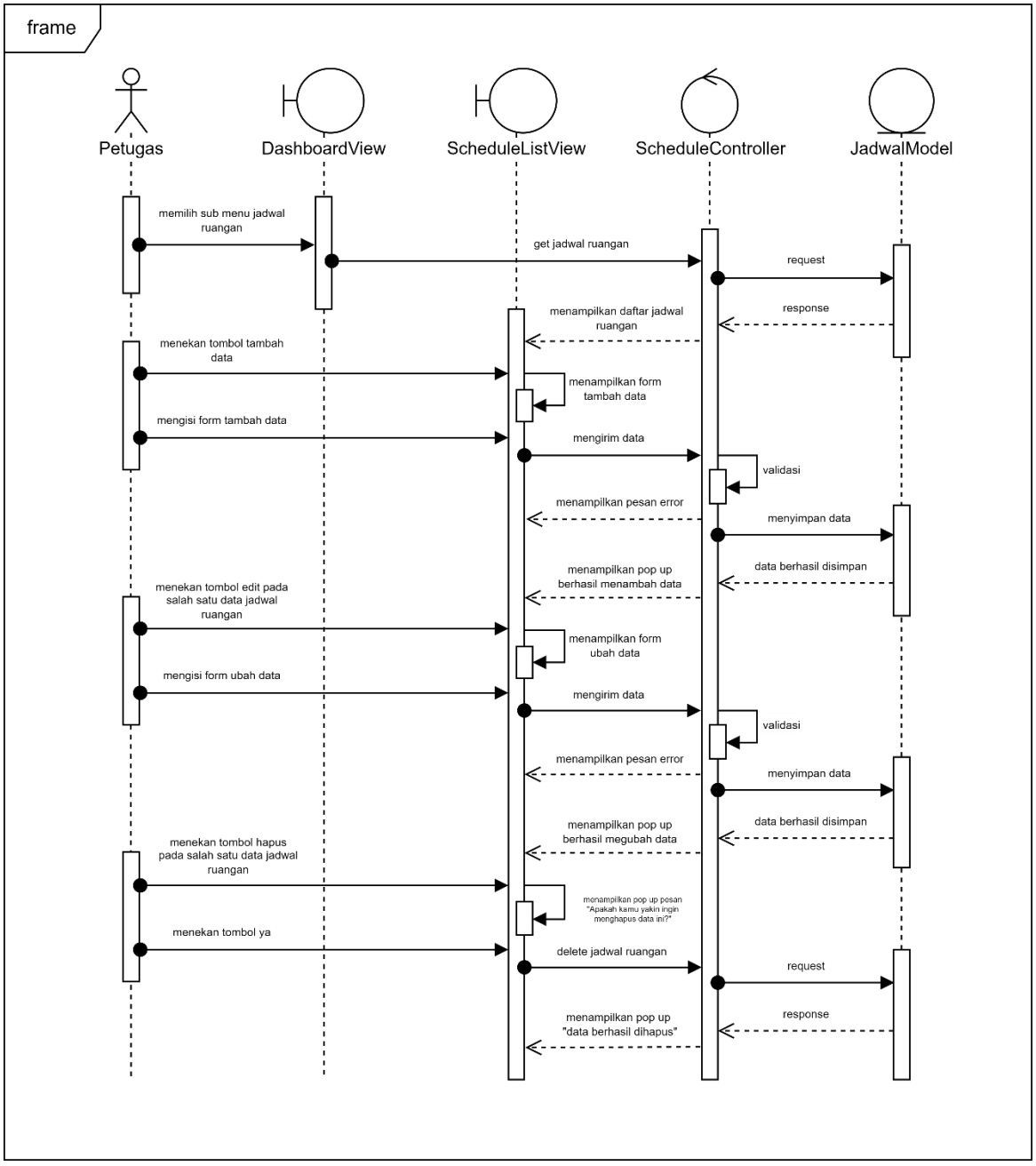
Tabel 4.47 Diagram *Sekuens* Memverifikasi Peminjam

Memverifikasi Peminjam

Deskripsi
Diagram ini menggambarkan bagaimana petugas melakukan verifikasi data peminjam, dimulai dari peminjam memilih menu user, lalu sistem akan menampilkan daftar peminjam, selanjutnya petugas memilih salah satu data peminjam dan sistem akan menampilkan detail data peminjam yang dipilih, setelah itu petugas akan melakukan verifikasi dengan memilih salah tombol diterima atau ditolak lalu sistem akan mengubah status dari pengajuan dan menampilkan pop up data berhasil di verifikasi

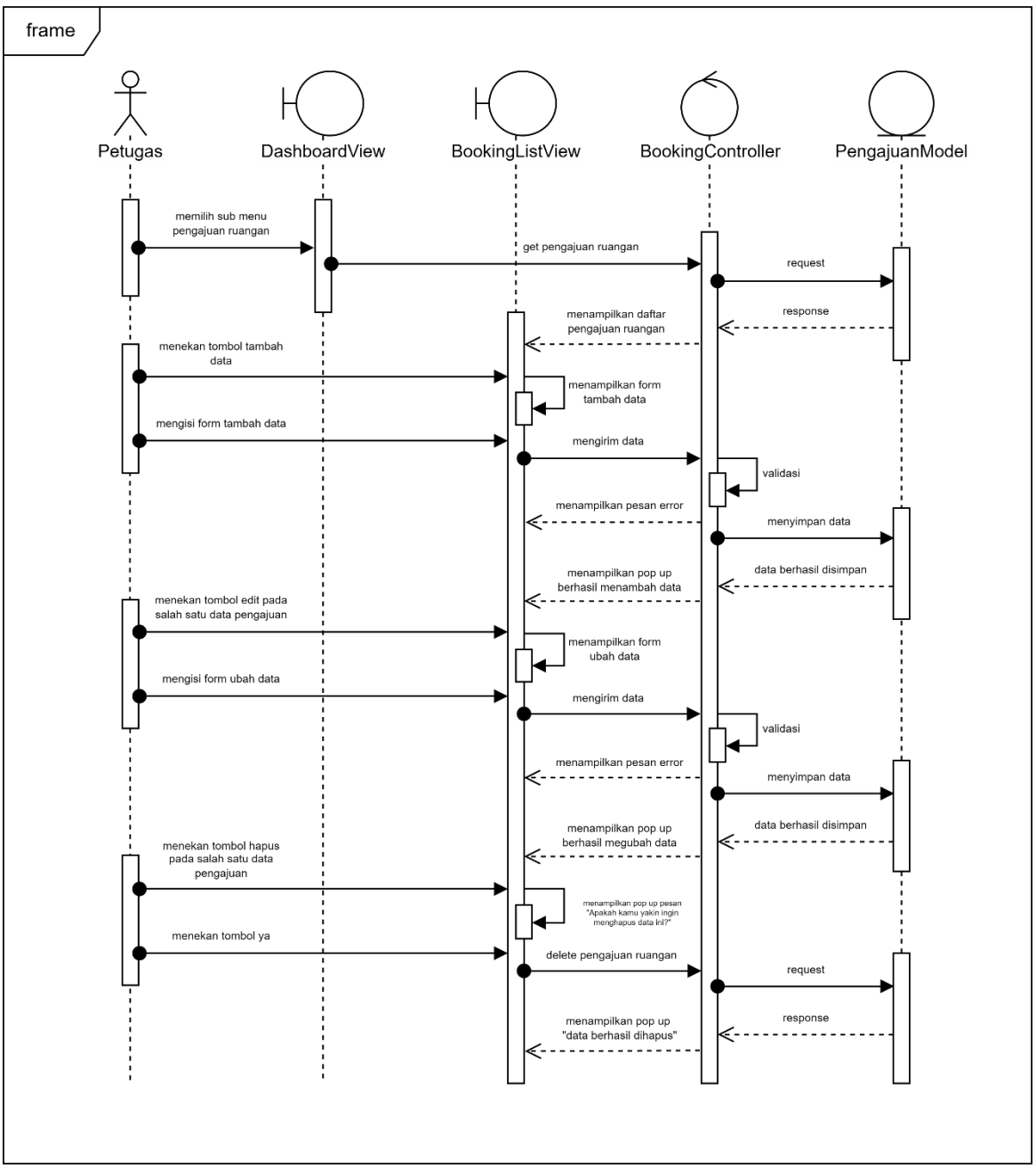
Tabel 4.48 Diagram *Sekuens* Mengelola Ruangan

Mengelola Ruangan

Deskripsi
Diagram ini menggambarkan bagaimana petugas mengelola data ruangan, yaitu petugas melihat data, menambah data, mengubah data, dan menghapus data

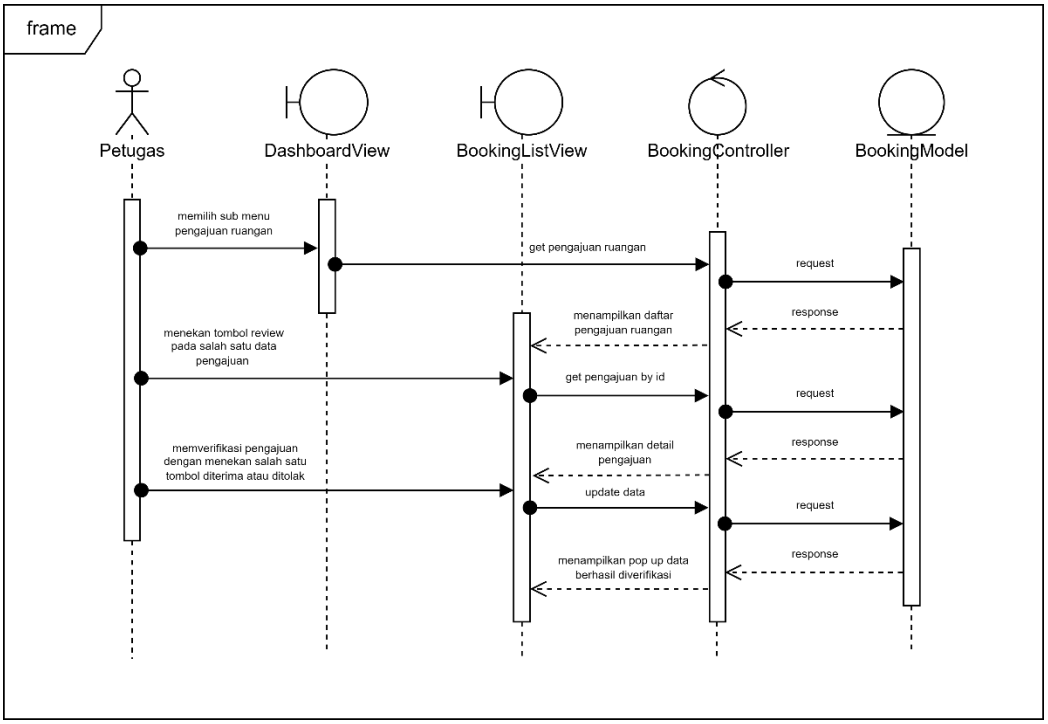
Tabel 4.49 Diagram *Sekuens* Mengelola Jadwal Ruangan

Mengelola Jadwal Ruangan

Deskripsi
Diagram ini menggambarkan bagaimana petugas mengelola data jadwal ruangan, yaitu petugas melihat data, menambah data, mengubah data, dan menghapus data

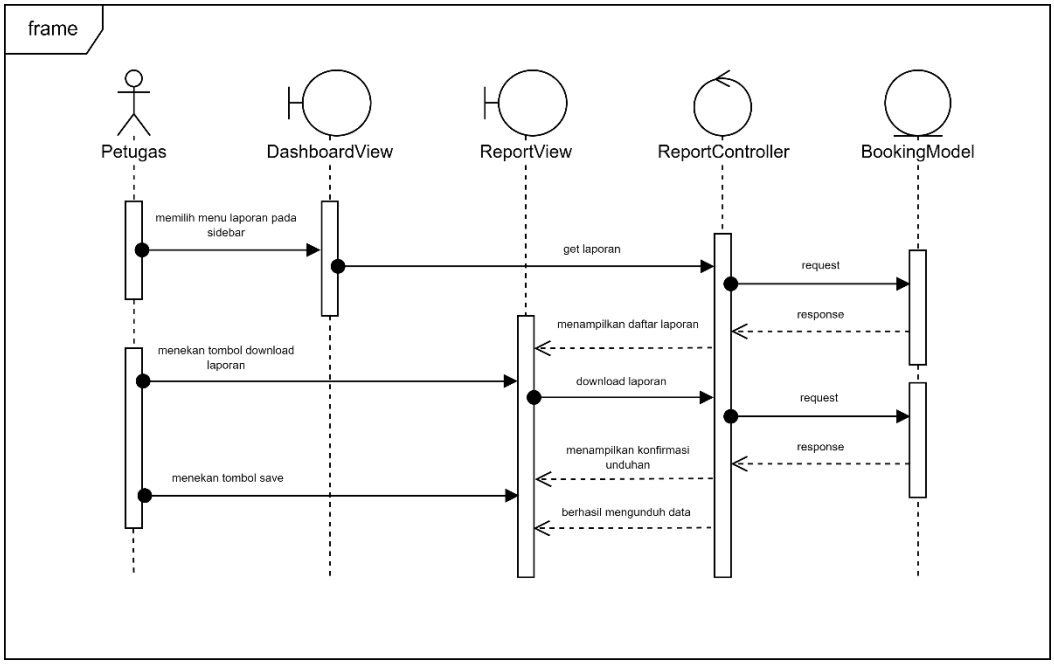
Tabel 4.50 Diagram *Sekuens* Mengelola Pengajuan

Mengelola Pengajuan

Deskripsi
Diagram ini menggambarkan bagaimana petugas mengelola data pengajuan, yaitu petugas melihat data, menambah data, mengubah data, dan menghapus data

Tabel 4.51 Diagram *Sekuens* Memverifikasi Pengajuan

Memverifikasi Pengajuan

Deskripsi
Diagram ini menggambarkan bagaimana petugas melakukan verifikasi pengajuan, dimulai dari petugas memilih sub menu pengajuan ruangan, lalu sistem akan menampilkan daftar pengajuan, selanjutnya petugas memilih salah satu data pengajuan dan sistem akan menampilkan detail pengajuan yang dipilih, setelah itu petugas akan melakukan verifikasi dengan memilih salah tombol diterima atau ditolak lalu sistem akan mengubah status dari pengajuan dan menampilkan pop up data berhasil di verifikasi

Tabel 4.52 Diagram *Sekuens* Mengelola Laporan

Mengelola Laporan

Deskripsi
Diagram ini menggambarkan bagaimana petugas mengelola data laporan, yaitu petugas melihat data, menambah data, mengubah data, dan mengunduh data

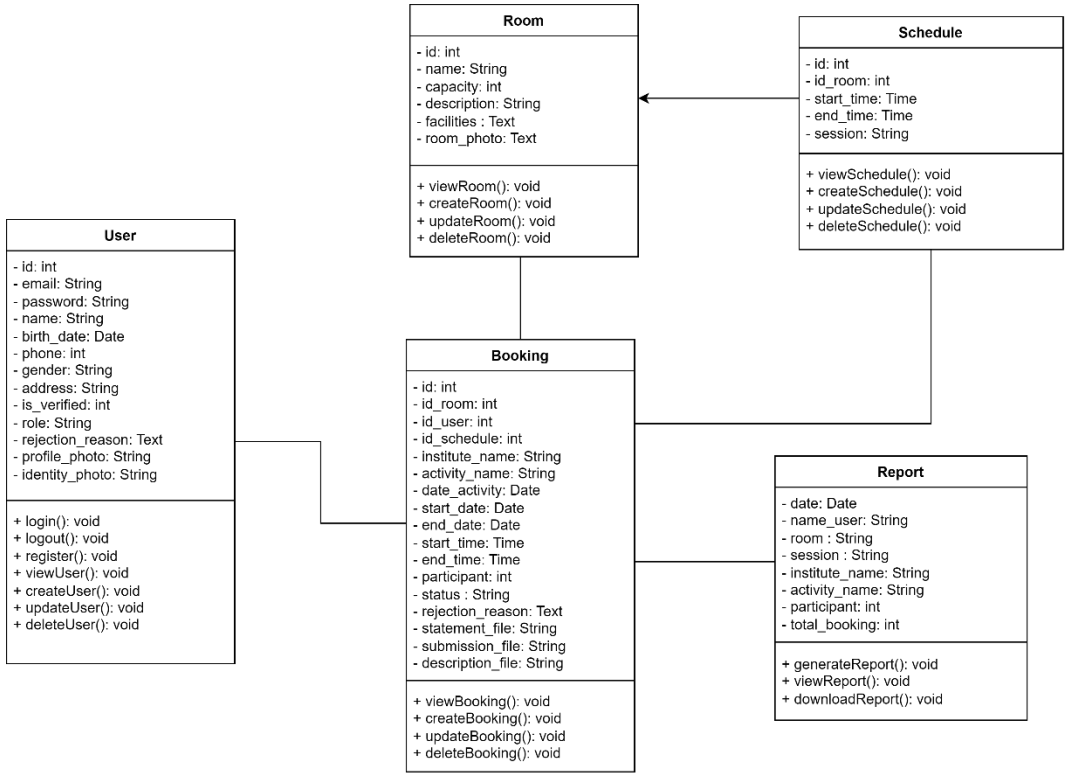
4.2.6. Pemodelan Data

Pemodelan data adalah proses-proses yang digunakan untuk data yang nantinya akan diterapkan pada sistem atau perangkat lunak yang dibuat.

4.2.6.1. Diagram Kelas Perancangan

Diagram kelas perancangan untuk menggambarkan dengan jelas kelas-kelas beserta atribut, fungsi, dan hubungan dari setiap objek. Berikut ini adalah diagram kelas dari Website Peminjaman Ruangan di Bandung Creative Hub dapat dilihat pada tabel 4.53.

Tabel 4.53 Diagram Kelas Perancangan

Diagram Kelas Perancangan Website Peminjaman Ruangan

Deskripsi
Diagram ini merupakan diagram kelas yang sudah dirancang dari mulai kelas-kelas sampai penentuan fungsi-fungsi. Rancangan untuk peminjam dan petugas Bandung Creative Hub

4.2.6.2. Objek Data

Objek data merupakan kumpulan berbagai elemen-elemen yang mungkin mempunyai suatu tipe data tertentu. Objek data juga adalah representasi dari informasi yang harus dipahami oleh perangkat lunak. Objek data yang akan dibuat dapat dilihat pada tabel 4.54.

Tabel 4.54 Objek Data

No.	Nama Objek	Deskripsi
1.	User	Objek ini digunakan untuk menyimpan data diri peminjam
2.	Room	Objek ini digunakan untuk menyimpan data tentang ruangan
3.	Schedule	Objek ini digunakan untuk menyimpan data tentang jadwal yang ada pada ruangan
4.	Booking	Objek ini digunakan untuk menyimpan data tentang pengajuan peminjaman
5.	Report	Objek ini digunakan untuk menyimpan data mengenai laporan peminjaman ruangan

4.2.6.3. Atribut Data

Atribut data merupakan properti atau karakteristik yang dipunyai oleh suatu objek data. Daftar atribut data dari objek data yang sudah dibuat sebelumnya dapat dilihat pada tabel 4.55.

Tabel 4.55 Atribut Data

No.	Nama Objek	Atribut	Deskripsi
1.	User	id	Atribut ini digunakan untuk menyimpan id pengguna dan sebagai *primary key* pada tabel
		email	Atribut ini digunakan untuk menyimpan email
		password	Atribut ini digunakan untuk menyimpan kata sandi
		name	Atribut ini digunakan untuk menyimpan nama
		birth_date	Atribut ini digunakan unruk menyimpan tanggal lahir
		phone	Atribut ini digunakan untuk menyimpan nomer telepon
		gender	Atribut ini digunakan unruk menyimpan jenis kelamin
		address	Atribut ini digunakan untuk menyimpan alamat
		is_verified	Atribut ini digunakan untuk menyimpan status penggua
		role	Atribut ini digunakan untuk menyimpan role pengguna
		rejection_reason	Atribut ini digunakan untuk menyimpan alasan penolakan verifikasi
		profile_photo	Atribut ini digunakan untuk menyimpan foto profile
		identity_photo	Atribut ini digunakan untuk menyimpan foto identitas pengguna
2.	Room	id	Atribut ini digunakan untuk menyimpan id ruangan dan sebagai *primary key* pada tabel
		name	Atribut ini digunakan untuk menyimpan nama ruangan
		capacity	Atribut ini digunakan untuk menyimpan berapa kapasitas ruangan
		description	Atribut ini digunakan untuk menyimpan deskripsi dari ruangan
		facilities	Atribut ini digunakan untuk menyimpan fasilitas dari ruangan
		room_photo	Atribut ini digunakan untuk menyimpan beberapa gambar dari ruangan
3.	Schedule	id	Atribut ini digunakan untuk menyimpan id jadwal ruangan dan sebagai *primary key* pada tabel
		start_time	Atribut ini digunakan untuk menyimpan waktu mulai ruangan
		end_time	Atribut ini digunakan untuk menyimpan waktu selesai ruangan
		session	Atribut ini digunakan untuk menyimpan nama sesi
		room_id	Atribut ini digunakan untuk menyimpan id ruangan
4.	Booking	id	Atribut ini digunakan untuk menyimpan id pengajuan dan sebagai *primary key* pada tabel
		room_id	Atribut ini digunakan untuk menyimpan nama ruangan yang diajukan
		user_id	Atribut ini digunakan untuk menyimpan nama peminjam
		schedule_id	Atribut ini digunakan untuk menyimpan jadwal ruangan
		institute_name	Atribut ini digunakan untuk menyimpan nama instansi atau komunitas
		activity_name	Atribut ini digunakan untuk menyimpan nama kegiatan
		date_activity	Atribut ini digunakan untuk menyimpan tanggal kegiatan
		start_date	Atribut ini digunakan untuk menyimpan tanggal mulai kegiatan
		end_date	Atribut ini digunakan untuk menyimpan tanggal selesai kegiatan
		start_time	Atribut ini digunakan untuk menyimpan waktu mulai kegiatan
		end_time	Atribut ini digunakan untuk menyimpan waktu selesai kegiatan
		participant	Atribut ini digunakan untuk menyimpan total peserta kegiatan
		status	Atribut ini digunakan untuk menyimpan status pengajuan
		rejection_reason	Atribut ini digunakan untuk menyimpan alasan penolakan verifikasi

		submission_file	Atribut ini digunakan untuk menyimpan file surat pengajuan peminjaman
		description_file	Atribut ini digunakan untuk menyimpan file lampiran deskripsi kegiatan
		statement_file	Atribut ini digunakan untuk menyimpan file surat pernyataan
5.	Report	date	Atribut ini digunakan untuk menyimpan tanggal laporan
		name_user	Atribut ini digunakan untuk menyimpan nama peminjam
		room	Atribut ini digunakan untuk menyimpan nama rungan
		session	Atribut ini digunakan untuk menyimpan sesi ruangan
		institute_name	Atribut ini digunakan untuk menyimpan nama grup atau institusi peminjam
		activity_name	Atribut ini digunakan untuk menyimpan nama kegiatan
		participant	Atribut ini digunakan untuk menyimpan total peserta kegiatan
		total_booking	Atribut ini digunakan untuk menyimpan total peminjaman

4.2.7. Pemodelan Berbasis Kelas

Pemodelan berbasis kelas digunakan untuk menggambarkan objek-objek yang nantinya akan dimanipulasi oleh sistem atau perangkat lunak, menggambarkan operasi-operasi yang akan diterapkan pada objek, relasi-relasi antar objek serta kolaborasi yang terjadi antar kelas yang telah didefinisikan.

4.2.7.1. Identifikasi Kelas-Kelas Analisis

Pada proses mengidentifikasi kelas bisa dilakukan dengan cara memeriksa skenario sistem atau perangkat lunak yang akan dikembangkan sebagai acuan untuk membuat kelas-kelas analisis. Kelas-kelas analisis yang akan dibuat bisa dilihat pada tabel 4.56.

Tabel 4.56 Kelas-Kelas Analisis

No.	Nama Kelas	Jenis	Deskripsi
1.	HomeView	*View*	Kelas ini digunakan untuk menampilkan halaman utama *website*
2.	LoginView	*View*	Kelas ini digunakan untuk menampilkan halaman masuk peminjam
3.	RegisterView	*View*	Kelas ini digunakan untuk menampilkan halaman daftar peminjam
4.	ProfileView	*View*	Kelas ini digunakan untuk menampilkan halaman profil pengguna
5.	RoomView	*View*	Kelas ini digunakan untuk menampilkan halaman daftar ruangan yang ada
6.	RoomDetailView	*View*	Kelas ini digunakan untuk menampilkan halaman detail ruangan yang dipilih
7.	ScheduleView	*View*	Kelas ini digunakan untuk menampilkan halaman jadwal ruangan yang ada
8.	BookingView	*View*	Kelas ini digunakan untuk menampilkan halaman formulir pengajuan ruangan
9.	BookingListView	*View*	Kelas ini digunakan untuk menampilkan halaman riwayat pengajuan dari peminjam
10.	DashboardView	*View*	Kelas ini digunakan untuk menampilkan halaman utama admin
11.	UserListView	*View*	Kelas ini digunakan untuk menampilkan halaman pengelolaan peminjam
12.	BookingListView	*View*	Kelas ini digunakan untuk menampilkan halaman pengelolaan pengajuan yang ada
13.	RoomListView	*View*	Kelas ini digunakan untuk menampilkan halaman pengelolaan daftar ruangan
14.	ScheduleListView	*View*	Kelas ini digunakan untuk menampilkan halaman pengelolaan jadwal ruangan
15.	ReportView	*View*	Kelas ini digunakan untuk menampilkan halaman pengelolaan laporan
16.	AuthController	*Controller*	Kelas ini digunakan untuk mengelola *request* autentikasi pengguna
17.	RoomController	*Controller*	Kelas ini digunakan untuk mengelola *request* data ruangan
18.	SchedulelController	*Controller*	Kelas ini digunakan untuk mengelola *request* data jadwal ruangan
19.	BookingController	*Controller*	Kelas ini digunakan untuk mengelola *request* data pengajuan
20.	UserController	*Controller*	Kelas ini digunakan untuk mengelola *request* data peminjam
21.	ReportController	*Controller*	Kelas ini digunakan untuk mengelola *request* data laporan

22.	RoomModel	*Model*	Kelas ini digunakan untuk interaksi dengan database pada tabel Ruangan
23.	ScheduleModel	*Model*	Kelas ini digunakan untuk interaksi dengan database pada tabel Jadwal
24.	BookingModel	*Model*	Kelas ini digunakan untuk interaksi dengan database pada tabel Pengajuan
25.	UserModel	*Model*	Kelas ini digunakan untuk interaksi dengan database pada tabel User

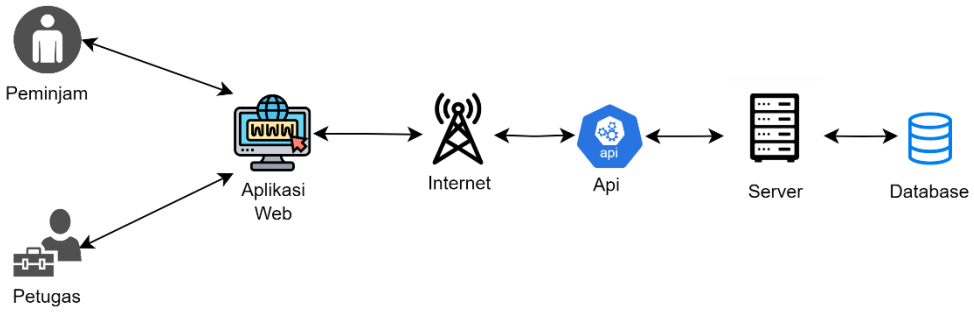
4.3. Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan tahap proses untuk menerapkan berbagai macam teknik dan prinsip untuk merealisasikan bentuk fisik dari perangkat lunak yang akan dikembangkan. Beberapa tahapan dalam perancangan perangkat lunak yang diterapkan yaitu perancangan data, perancangan arsitektural, dan perancangan antarmuka.

4.3.1. Perancangan Arsitektural

Perancangan arsitektural digunakan untuk menggambarkan komunikasi antar komponen-komponen program yang diperlukan untuk mengembangkan sistem atau perangkat lunak. Arsitektur yang dibangun pada penelitian tugas akhir "Pengembangan Aplikasi Peminjaman Ruangan Berbasis Web Di Bandung Creative Hub" ini terdiri dari beberapa komponen utama yaitu aplikasi web untuk peminjam, aplikasi web admin untuk petugas, API, *server* dan *database.*

Peminjam dan petugas mengakses sistem melalui aplikasi web yang dibangun menggunakan React.JS. Komunikasi antara aplikasi web dan server dilakukan melalui API yang dibangun menggunakan Express.JS, yang berfungsi sebagai penghubung antara *client* dan *server*. *Server* ini memproses permintaan dan mengelola data yang tersimpan di *database* menggunakan MySQL.

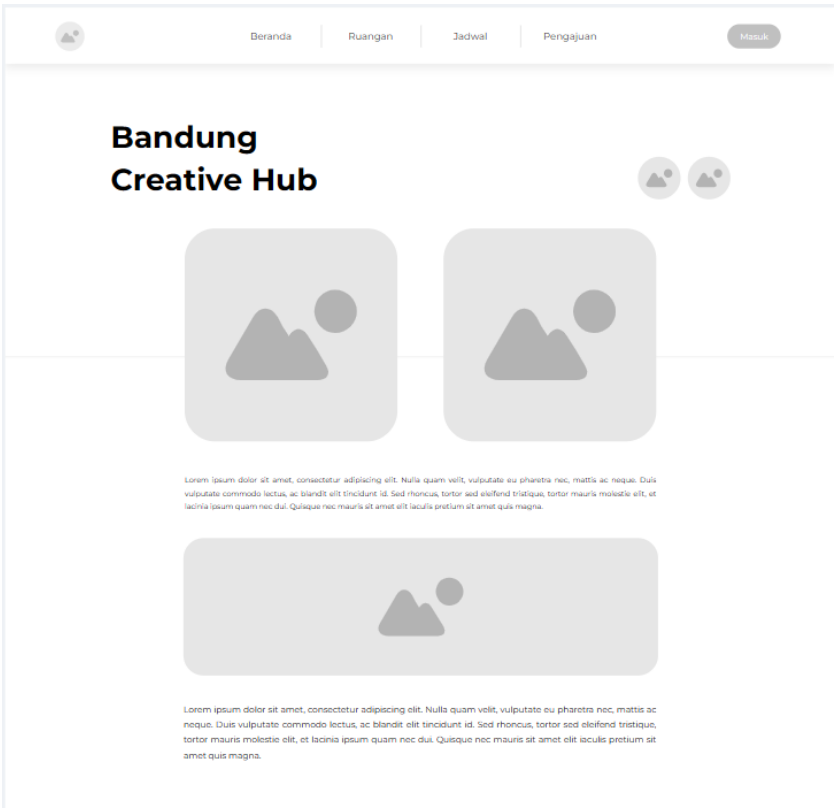


Gambar 4.1 Perancangan Arsitektur

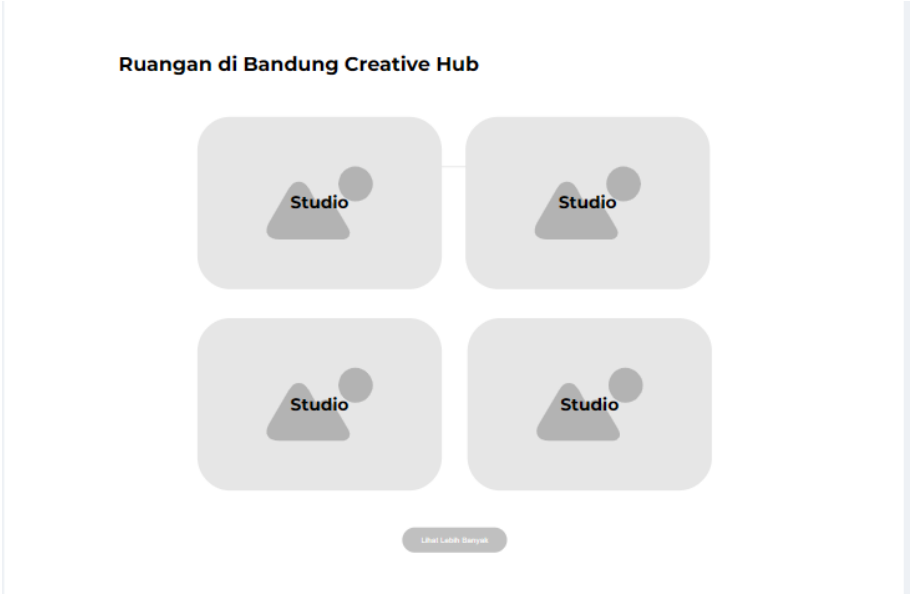
4.3.2. Perancangan Antarmuka Pengguna

Perancangan antarmuka proses untuk membuat penggambaran bagaimana antarmuka sistem atau perangkat lunak dibentuk dengan menciptakan komunikasi yang efektif yang menjembatani interaksi anatara manusia dan komputer. Antarmuka pengguna dari perangkat lunak yang dibuat dapat dilihat pada tabel 4.57 sampai dengan 4.82.

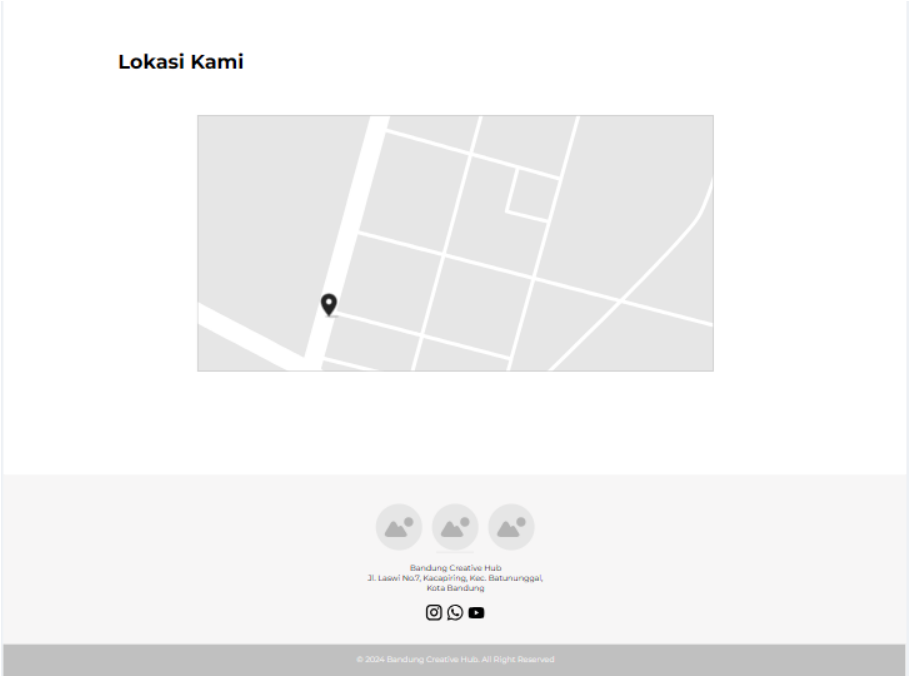
Tabel 4.57 Rancangan Antarmuka Halaman Informasi

Rancangan Antarmuka Halaman Informasi

Deskripsi
Rancangan antarmuka ini merupakan halaman utama dari aplikasi ini. Dibagian atas terdapat menu yang terdiri dari *Beranda, Ruangan, Jadwal,* dan *Pengajuan*, serta tombol *Masuk* yang digunakan untuk ke halaman login. Dibagian utama halaman terdapat beberapa foto dan deskripsi tentang Bandung Creative Hub.

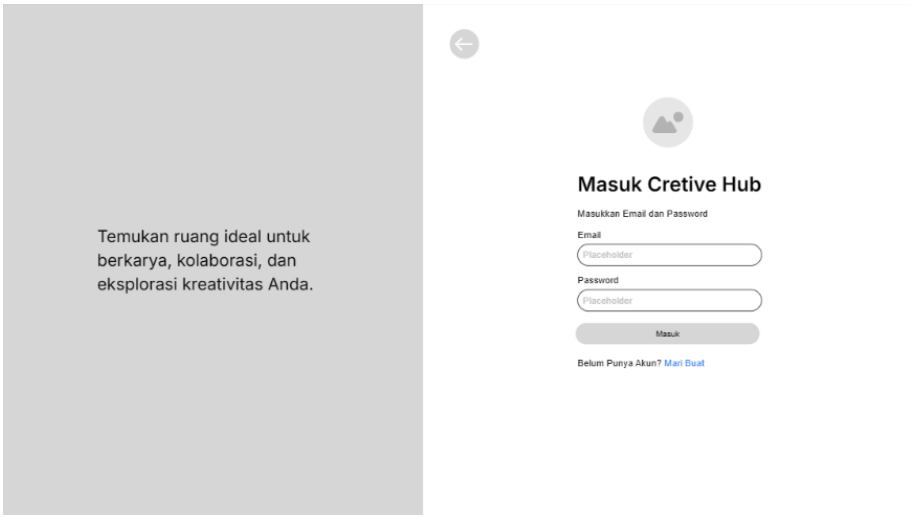
Tabel 4.58 Rancangan Antarmuka Halaman Informasi Ruangan

Rancangan Antarmuka Halaman Informasi Ruangan

Deskripsi
Rancangan antarmuka ini merupakan tampilan lanjutan pada halaman utama. Dibagian ini terdapat beberapa foto ruangan berserta nama ruangannya dan tombol “Lihat Lebih Banyak” yang digunakan untuk mengalihkan ke halaman *Ruangan*.

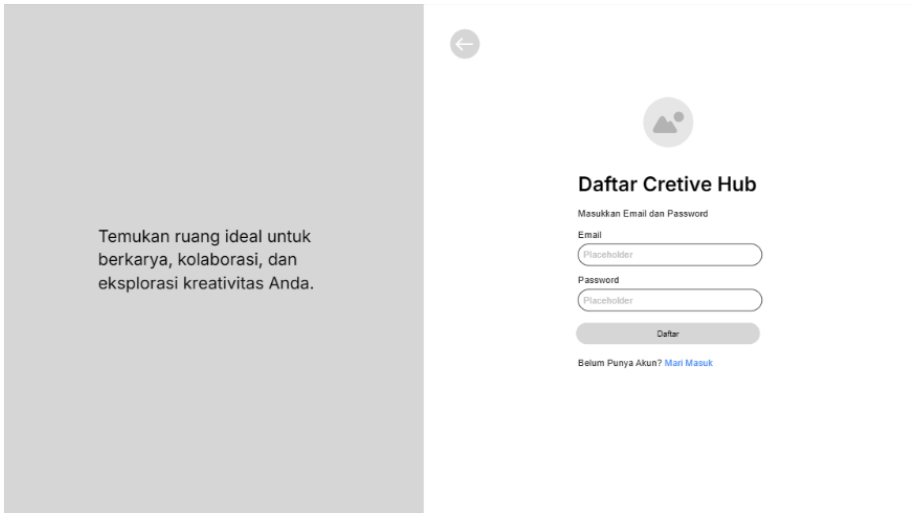
Tabel 4.59 Rancangan Antarmuka Halaman Informasi Lokasi

Rancangan Antarmuka Halaman Informasi Lokasi

Deskripsi
Rancangan antarmuka ini merupakan tampilan lanjutan dari halaman utama. Dibagian ini terdapat peta lokasi dari Bandung Creative Hub dan dibagian bawah terdapat alamat dan sosial media Bandung Creative Hub.

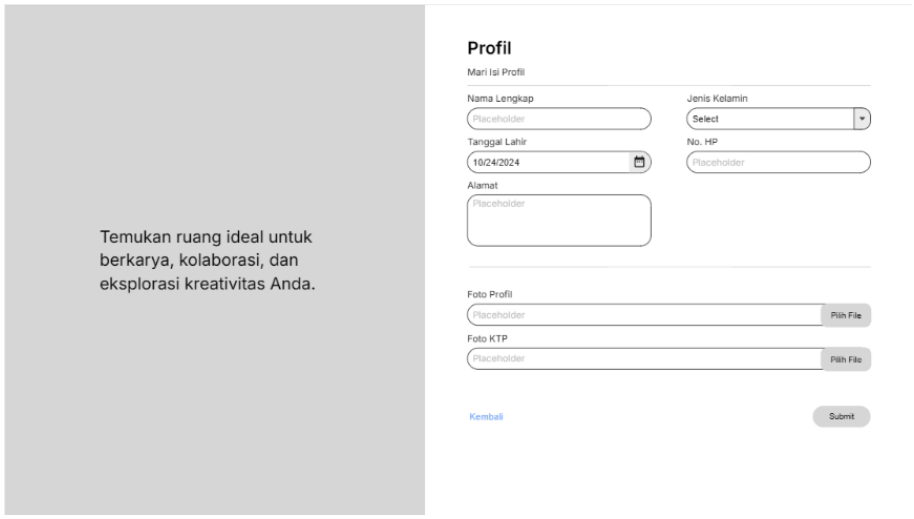
Tabel 4.60 Rancangan Antarmuka Halaman Login

Rancangan Antarmuka Halaman Login

Deskripsi
Rancangan antarmuka ini menampilkan halaman login yang terbagi menjadi dua bagian: kiri sebagai area informasi singkat, sementara dibagian sebelah kanan terdapat form login yang terdapat form input untuk email dan password. Setelah mengisi data, pengguna dapat mengklik tombol “Masuk” untuk melanjutkan. Jika belum memiliki akun, pengguna dapat memilih link “Mari Buat” untuk mengakses halaman pendaftaran.

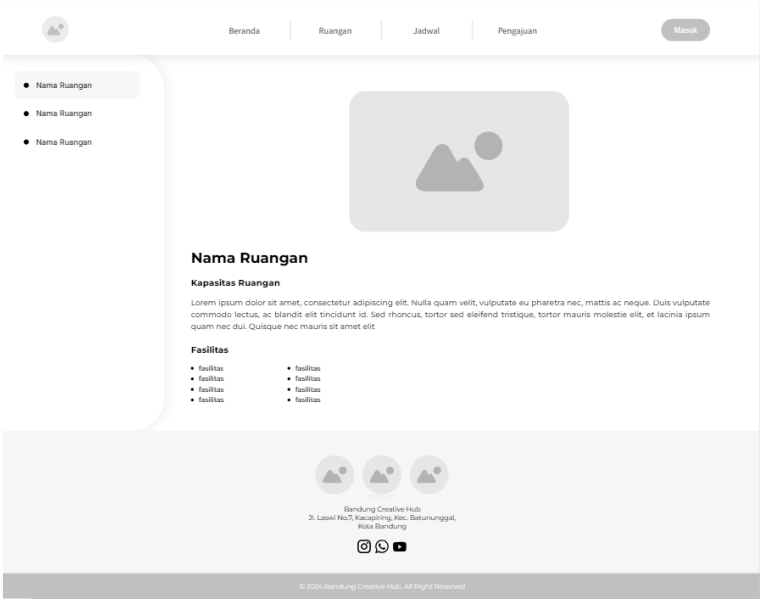
Tabel 4.61 Rancangan Antarmuka Halaman Register

Rancangan Antarmuka Halaman Register

Deskripsi
Rancangan antarmuka ini menampilkan halaman registrasi yang terbagi menjadi dua bagian: kiri sebagai area informasi singkat, sementara dibagian sebelah kanan terdapat form pendaftaran yang terdapat form input untuk email dan password. Setelah mengisi data, pengguna dapat mengklik tombol “Daftar” untuk melanjutkan. Jika sudah memiliki akun, pengguna dapat memilih link “Mari Masuk” untuk mengakses halaman login.

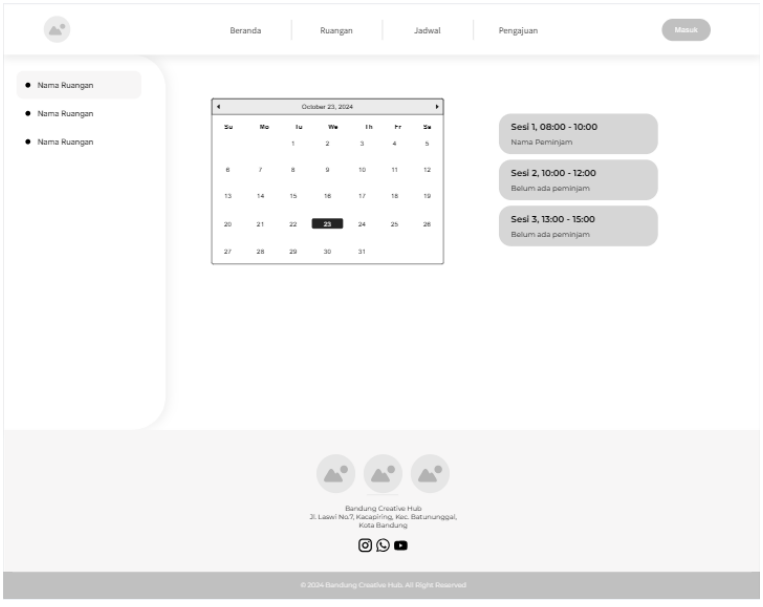
Tabel 4.62 Rancangan Antarmuka Halaman Register 2

Rancangan Antarmuka Halaman Register 2

Deskripsi
Rancangan antarmuka ini menampilkan halaman registrasi yang kedua yang terbagi menjadi dua bagian: kiri sebagai area informasi singkat, sementara dibagian sebelah kanan terdapat form pendaftaran yang lebih lengkap yang terdapat form input untuk nama lengkap, jenis kelamin, tanggal lahir, nomer HP, alamat, foto profil dan foto KTP. Setelah mengisi data, pengguna dapat mengklik tombol “Simpan” untuk melanjutkan atau “Kembali” untuk kembali ke form halaman registrasi sebelumnya.

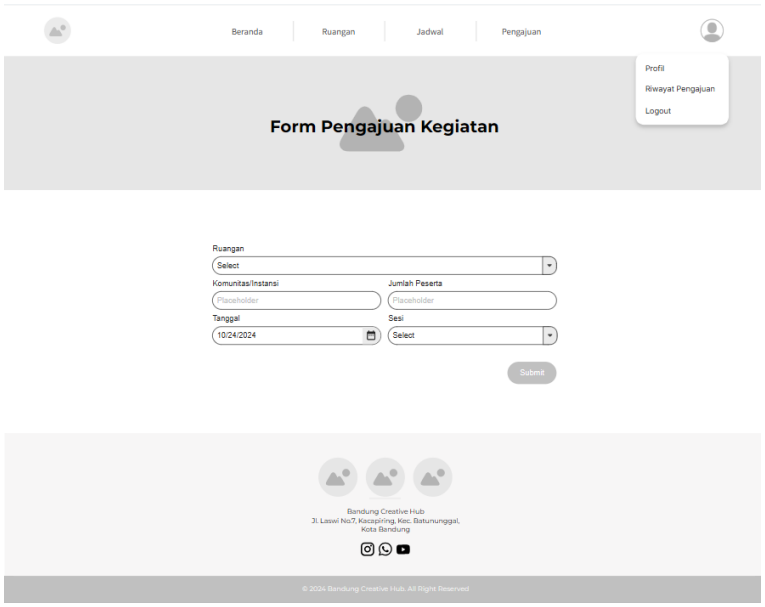
Tabel 4.63 Rancangan Antarmuka Halaman Detail Ruangan

Rancangan Antarmuka Halaman Detail Ruangan

Deskripsi
Rancangan antarmuka ini merupakan halaman informasi ruangan. Disebelah kiri terdapat daftar nama ruangan yang dapat diklik untuk menampilkan detailnya. Dibagian tengah terdapat gambar, nama, kapasitas, deskripsi dan fasilitas ruangan.

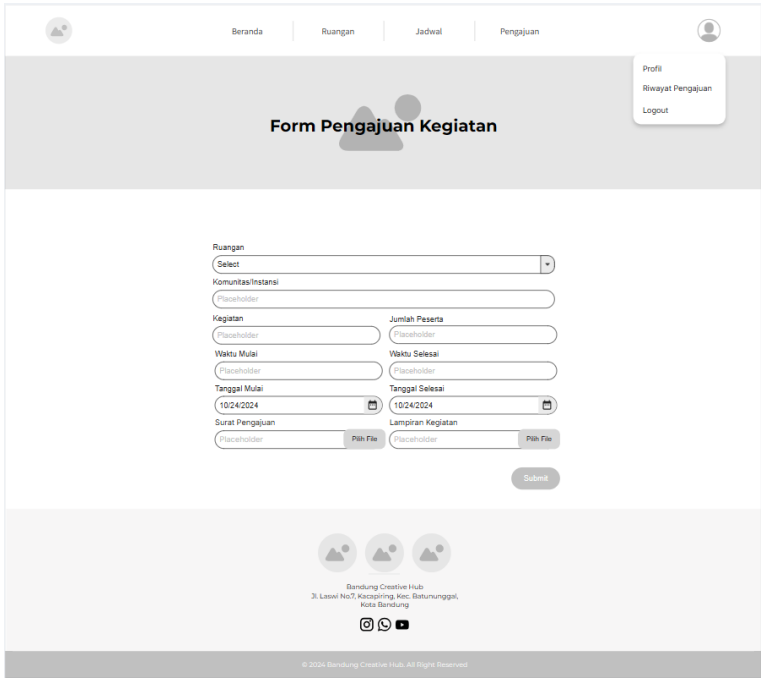
Tabel 4.64 Rancangan Antarmuka Halaman Jadwal Ruangan

Rancangan Antarmuka Halaman Jadwal Ruangan

Deskripsi
Rancangan antarmuka ini merupakan halaman informasi jadwal ruangan. Disebelah kiri terdapat daftar nama ruangan yang dapat diklik untuk menampilkan jadwal dari ruangan yang dipilih. Dibagian tengah terdapat kalender bulanan yang bisa digunakan oleh pengguna memilih tanggal untuk melihat jadwal pada hari tersebut. Di sisi kanan kalender, terdapat sesi waktu ruangan untuk hari yang dipilih, dengan rincian jam dan nama peminjam (jika sudah ada pemesanan). Jika sesi masih kosong, muncul keterangan "Belum ada peminjam."

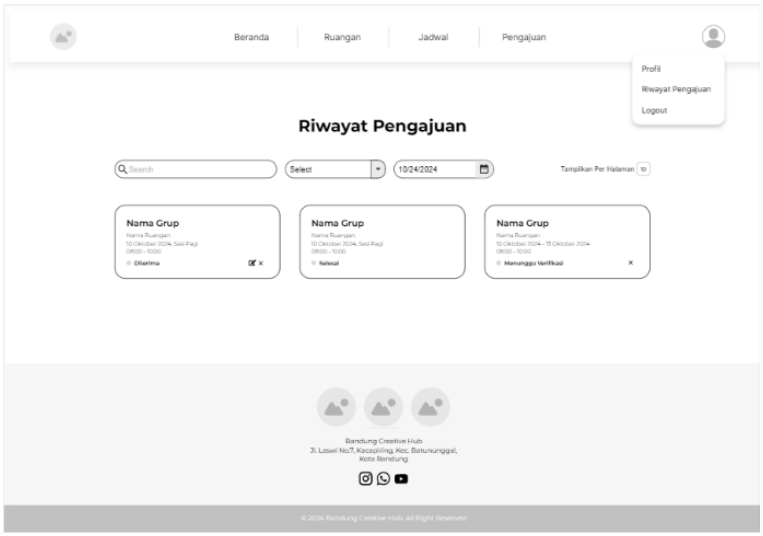
Tabel 4.65 Rancangan Antarmuka Halaman Pengajuan

Rancangan Antarmuka Halaman Pengajuan

Deskripsi
Rancangan antarmuka ini merupakan halaman form pengajuan untuk jenis ruangan yang pertama terdapat form input untuk memilih ruangan, komunitas/instansi, jumlah peserta, tanggal, dan sesi. Jika sudah mengisi data pengguna bisa mengklik tombol “Simpan” untuk menyimpan pengajuan.

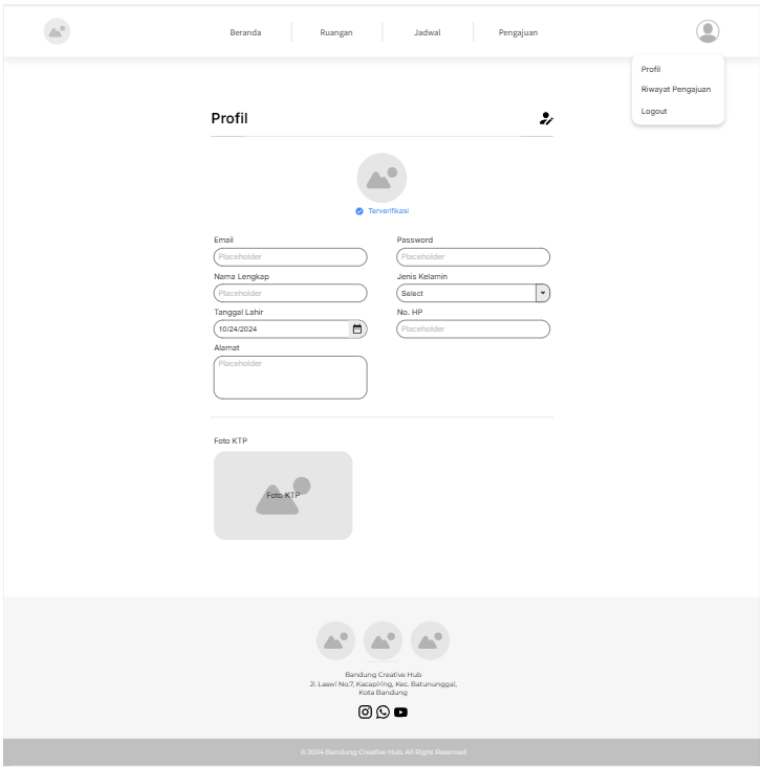
Tabel 4.66 Rancangan Antarmuka Halaman Pengajuan 2

Rancangan Antarmuka Halaman Pengajuan 2

Deskripsi
Rancangan antarmuka ini merupakan halaman form pengajuan untuk jenis ruangan yang kedua terdpat form input untuk memilih ruangan, komunitas/instansi, nama kegiatan, jumlah peserta, waktu mulai, waktu selesai, tanggal mulai, tanggal selesai, surat pengajuan dan lampiran kegiatan. Jika sudah mengisi data pengguna bisa mengklik tombol “Simpan” untuk menyimpan pengajuan.

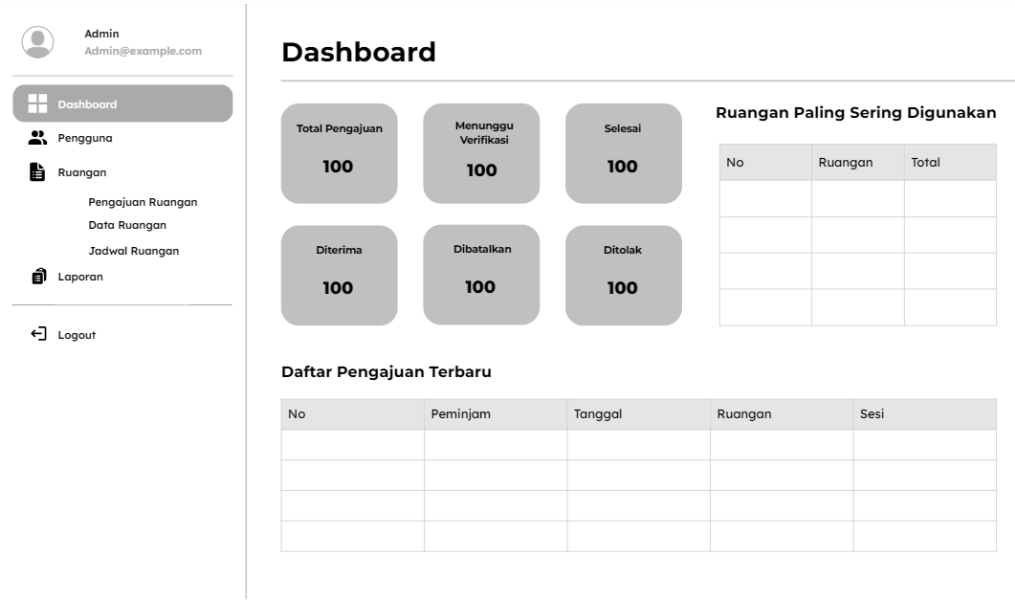
Tabel 4.67 Rancangan Antarmuka Halaman Riwayat Pengajuan

Rancangan Antarmuka Halaman Riwayat Pengajuan

Deskripsi
Rancangan antarmuka ini merupakan halaman riwayat pengajuan. Dihalaman ini terdapat informasi pengajuan ruangan dalam satu pengajuan terdapat informasi nama grup/instansi, ruangan, tanggal, waktu dan status pengajuan. Dibagian atas terdapat fitur pencarian serta filter berdasarkan status pengajuan dan tanggal kegiatan.

Tabel 4.68 Rancangan Antarmuka Halaman Profil

Rancangan Antarmuka Halaman Profil

Deskripsi
Rancangan antarmuka ini merupakan halaman profil dari pengguna dengan form input seperti email, password, nama lengkap, jenis kelamin, tanggal lahir, no HP, alamat dan foto KTP. Di bagian atas terdapat foto profil dengan status verifikasi dan opsi untuk mengedit.

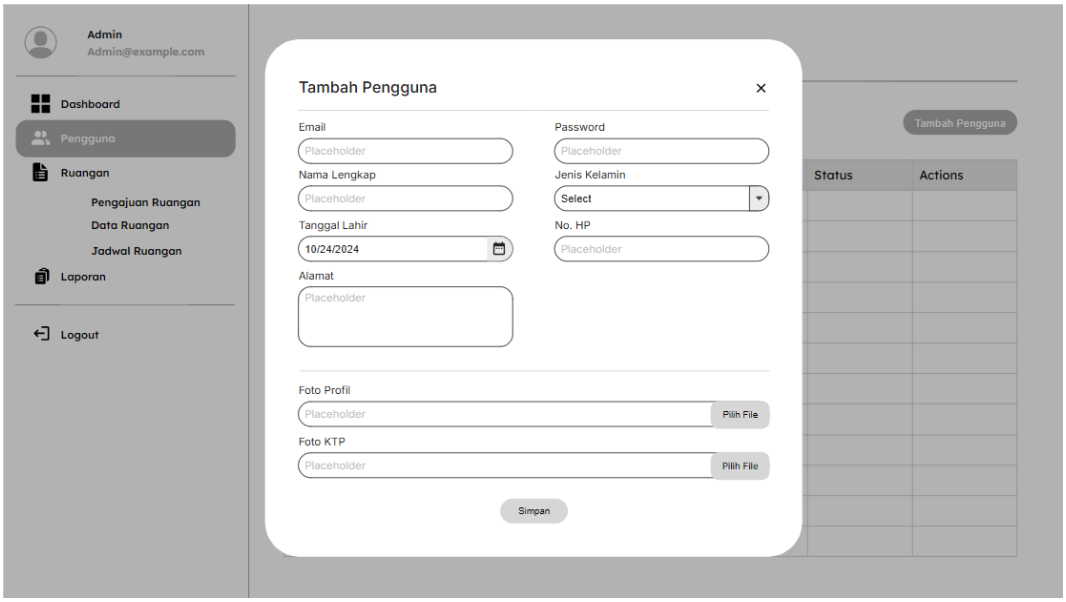
Tabel 4.69 Rancangan Antarmuka Halaman Dashboard

Rancangan Antarmuka Halaman Dashboard

Deskripsi
Rancangan antarmuka ini merupakan halaman dashboard untuk petugas. Dibagian sebelah kiri terdapat beberapa menu yang ditampilkan pada semua halaman petugas seperti *Dashboard, Pengguna, Ruangan* dengan submenu *Pengajuan Ruangan, Data Ruangan* dan Jadwal Ruangan, *Laporan,* serta tombol *Logout.* Bagian utama halaman menampilkan ringkasan statistik dalam empat kotak yaitu total pengajuan, menunggu verifikasi, diterima, dan selesai serta terdapat dua tabel yaitu untuk menampilkan ruangan paling sering digunakan dan daftar pengajuan terbaru.

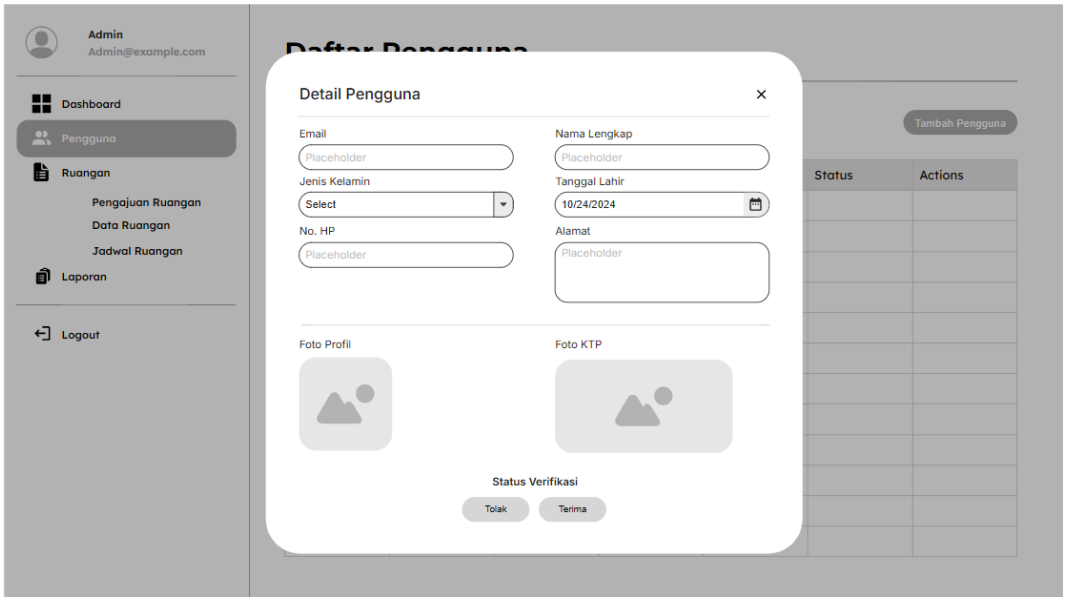
Tabel 4.70 Rancangan Antarmuka Halaman Daftar Pengguna

Rancangan Antarmuka Halaman Daftar Pengguna
Deskripsi
Rancangan antarmuka ini merupakan halaman daftar pengguna. Pada halaman ini terdapat kolom pencarian, tombol “Tambah Pengguna” dan tabel yang menampilkan daftar pengguna.

Tabel 4.71 Rancangan Antarmuka Halaman Tambah Pengguna

Rancangan Antarmuka Halaman Tambah Pengguna

Deskripsi
Rancangan antarmuka ini merupakan tampilan untuk menambah data pengguna, terdapat form inputan untuk email, password, nama lengkap, jenis kelamin, tanggal lahir, no hp, alamat, foto profil dan foto KTP. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

Tabel 4.72 Rancangan Antarmuka Halaman Detail Pengguna

Rancangan Antarmuka Halaman Detail Pengguna

Deskripsi
Rancangan antarmuka ini merupakan tampilan untuk menampilkan detail data dari pengguna yang bisa digunakan petugas untuk melakukan verifikasi data.

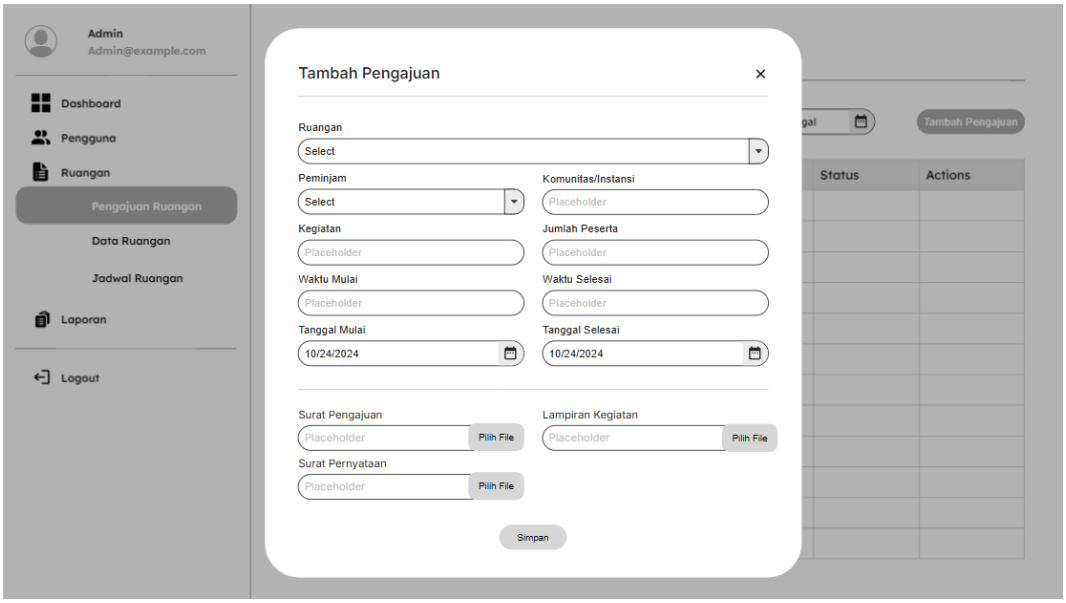
Tabel 4.73 Rancangan Antarmuka Halaman Daftar Pengajuan

Rancangan Antarmuka Halaman Daftar Pengajuan
Deskripsi
Rancangan antarmuka ini merupakan halaman daftar pengajuan. Pada halaman ini terdapat kolom pencarian, filter berdasarkan ruangan, status, tanggal, tombol “Tambah Pengajuan” dan tabel yang menampilkan daftar pengajuan.

Tabel 4.74 Rancangan Antarmuka Halaman Tambah Pengajuan

Rancangan Antarmuka Halaman Tambah Pengajuan
Deskripsi
Rancangan antarmuka ini merupakan tampilan untuk menambah data pengajuan dengan jenis ruangan yang pertama, terdapat form inputan untuk ruangan, peminjam, komunitas/instansi, jumlah peserta, tanggal dan sesi. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

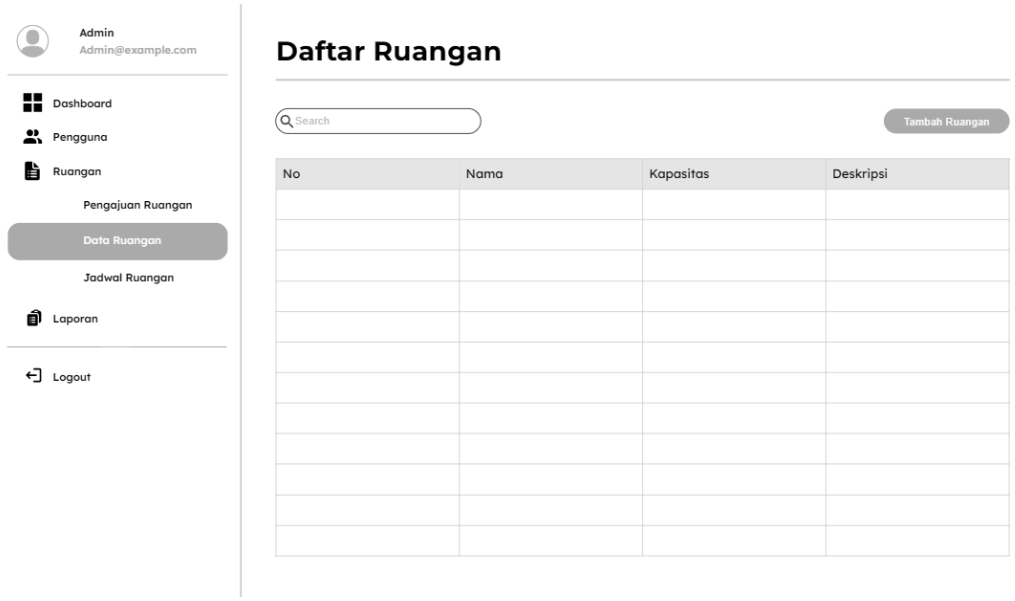
Tabel 4.75 Rancangan Antarmuka Halaman Tambah Pengajuan 2

Rancangan Antarmuka Halaman Tambah Pengajuan 2

Deskripsi
Rancangan antarmuka ini merupakan tampilan untuk menambah data pengajuan dengan jenis ruangan yang kedua, terdapat form inputan untuk ruangan, peminjam, komunitas/instansi, kegiatan, jumlah peserta, waktu mulai, waktu selesai, tanggal mulai, tanggal selesai, surat pengajuan, lampiran kegiatan, dan surat pernyataan. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

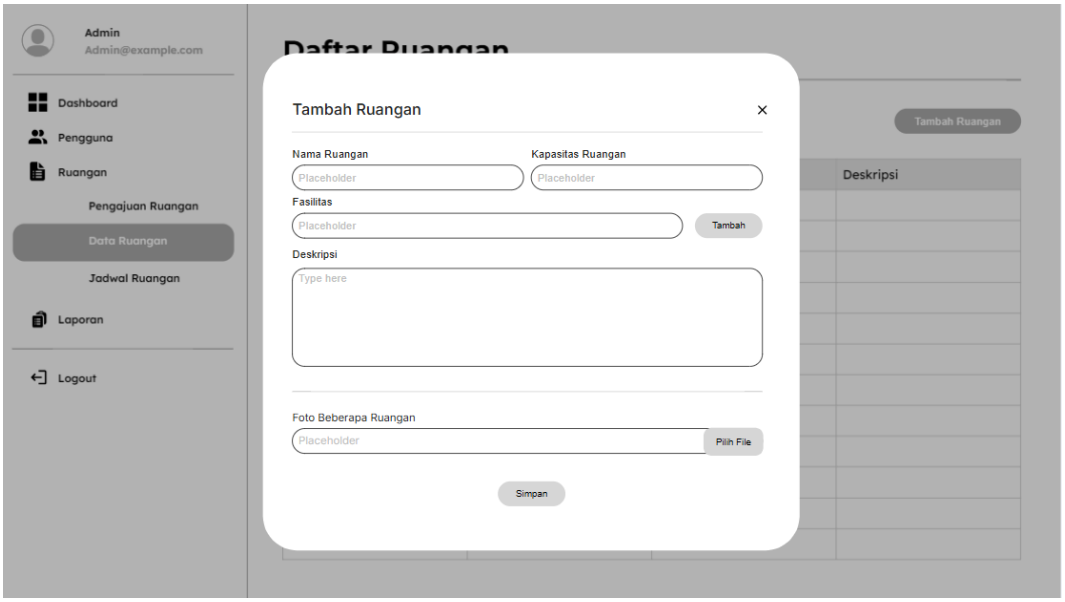
Tabel 4.76 Rancangan Antarmuka Halaman Detail Pengajuan

Rancangan Antarmuka Halaman Detail Pengajuan
Deskripsi
Rancangan antarmuka ini merupakan tampilan untuk menampilkan detail data dari pengajuan yang bisa digunakan petugas untuk melakukan verifikasi data. Petugas bisa memilih tombol “Terima” atau “Tolak” untuk melakukan verifikasi.

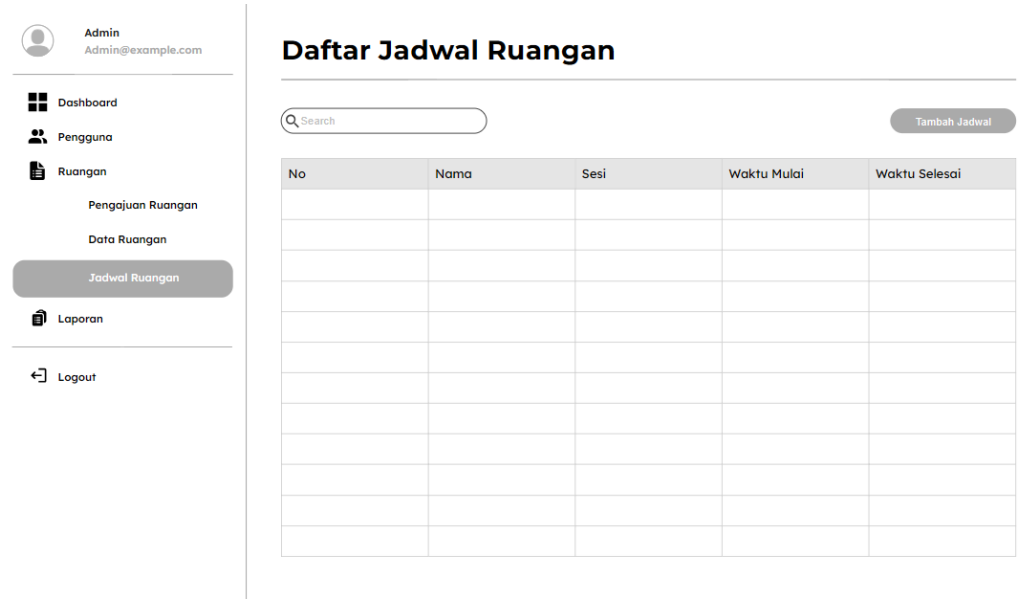
Tabel 4.77 Rancangan Antarmuka Halaman Daftar Ruangan

Rancangan Antarmuka Halaman Daftar Ruangan

Deskripsi
Rancangan antarmuka ini merupakan halaman daftar ruangan. Pada halaman ini terdapat kolom pencarian, tombol “Tambah Ruangan” dan tabel yang menampilkan daftar ruangan.

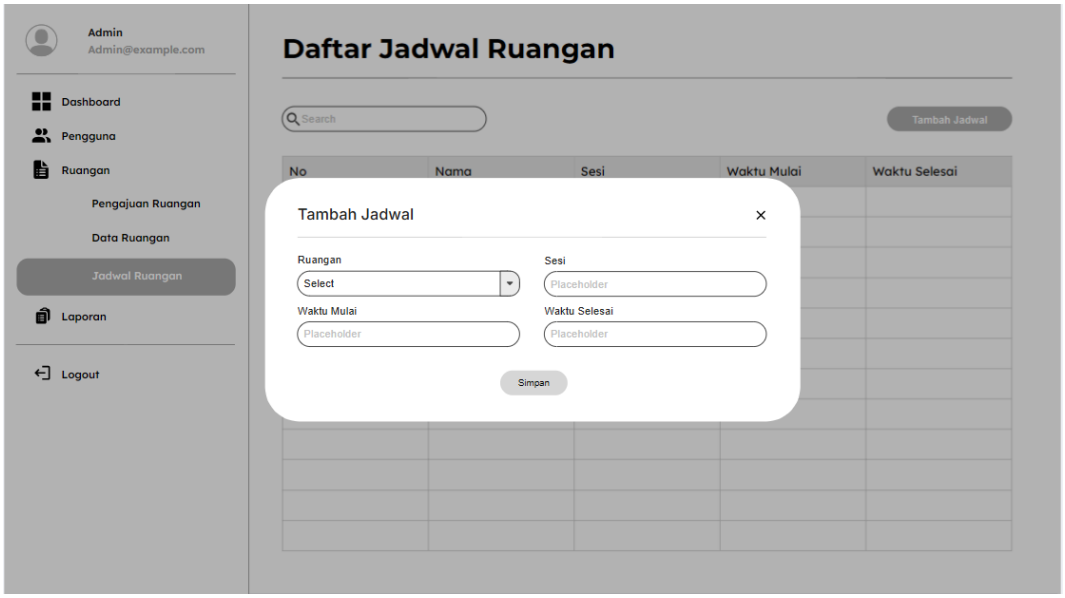
Tabel 4.78 Rancangan Antarmuka Halaman Tambah Ruangan

Rancangan Antarmuka Halaman Tambah Ruangan

Deskripsi
Rancangan antarmuka ini merupakan tampilan untuk menambah data ruangan, terdapat form inputan untuk nama, kapasitas, fasilitas, deskripsi, dan foto ruangan. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

Tabel 4.79 Rancangan Antarmuka Halaman Daftar Jadwal Ruangan

Rancangan Antarmuka Halaman Daftar Jadwal Ruangan

Deskripsi
Rancangan antarmuka ini merupakan halaman daftar jadwal ruangan. Pada halaman ini terdapat kolom pencarian, tombol “Tambah Jadwal” dan tabel yang menampilkan daftar jadwal ruangan.

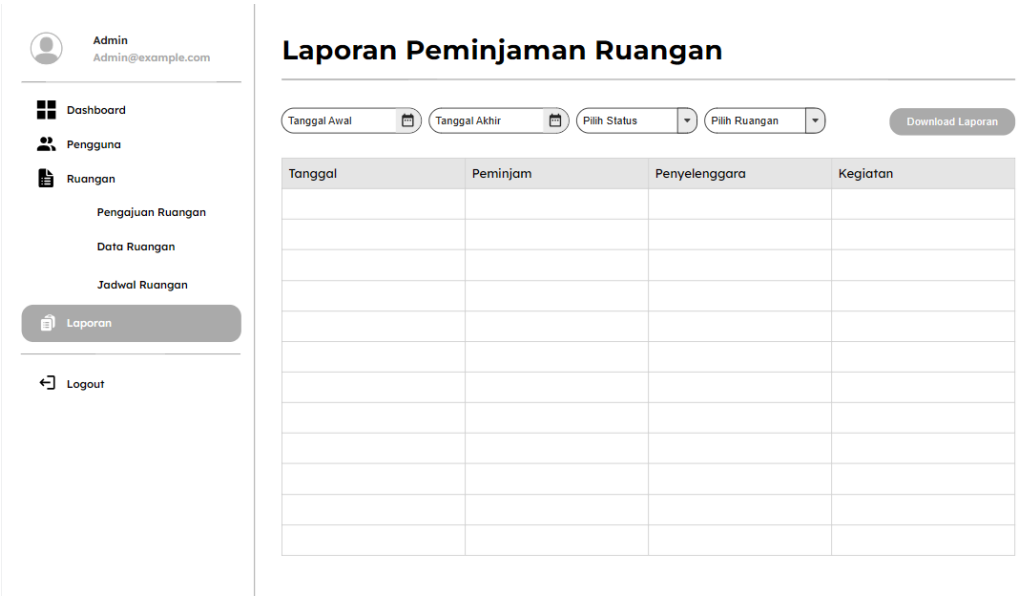
Tabel 4.80 Rancangan Antarmuka Halaman Tambah Jadwal Ruangan

Rancangan Antarmuka Halaman Tambah Jadwal Ruangan

Deskripsi
Rancangan antarmuka ini merupakan tampilan untuk menambah data jadwal, terdapat form inputan untuk ruangan, nama sesi, waktu mulai dan waktu selesai. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

Tabel 4.81 Rancangan Antarmuka Halaman Laporan

Rancangan Antarmuka Halaman Laporan
Deskripsi
Rancangan antarmuka ini merupakan halaman laporan peminjaman ruangan jenis pertama. Pada halaman ini petugas harus memilih tanggal awal, tanggal akhir, status, dan ruangan. Jika petugas sudah mengisi semua data nantinya akan muncul daftar laporan berdasarkan filter yang dipilih. Petugas juga bisa mendowload laporan dengan klik “Download Laporan”

Tabel 4.82 Rancangan Antarmuka Halaman Laporan 2

Rancangan Antarmuka Halaman Laporan 2

Deskripsi
Rancangan antarmuka ini merupakan halaman laporan peminjaman ruangan jenis kedua. Pada halaman ini petugas harus memilih tanggal awal, tanggal akhir, status, dan ruangan. Jika petugas sudah mengisi semua data nantinya akan muncul daftar laporan berdasarkan filter yang dipilih. Petugas juga bisa mendowload laporan dengan klik “Download Laporan”

BAB 5
IMPLEMENTASI

Bab ini menjelaskan proses implementasi pembangunan perangkat lunak. Pada tahap ini terdiri dari proses pembangunan dan pengujian pada fitur perangkat lunak.

5.1. Kontruksi Perangkat Lunak

Kontruksi perangkat lunak mengacu pada pembuatan detail kerja perangkat lunak yang berarti melalui kombinasi kode, integrasi , dan *debugging*. Kontruksi perangkat lunak mengacu pada perancangan-perancangan yang sudah dibuat sebelumnya.

5.1.1. Kakas dan Kebutuhan Perangkat Lunak

Dalam melakukan pembangunan perangkat lunak dibutuhkan beberapa kakas dan kebutuhan perangkat lunak. Berikut ini merupakan kakas dan kebutuhan perangkat lunak yang digunakan:

a. Spesifikasi perangkat keras

Tabel 5.1 Spesifikasi Perangkat Keras

No	Nama Perangkat	Sistem Requirement
1.	Processor	Intel ® core ™ i5
2.	RAM	8 GB
3.	Disk Space	256 GB
4.	Operating System	Windows 10 64-bit

b. Perangkat lunak yang digunakan dalam pembangunan

Tabel 5.2 Perangkat lunak yang digunakan dalam pembangunan

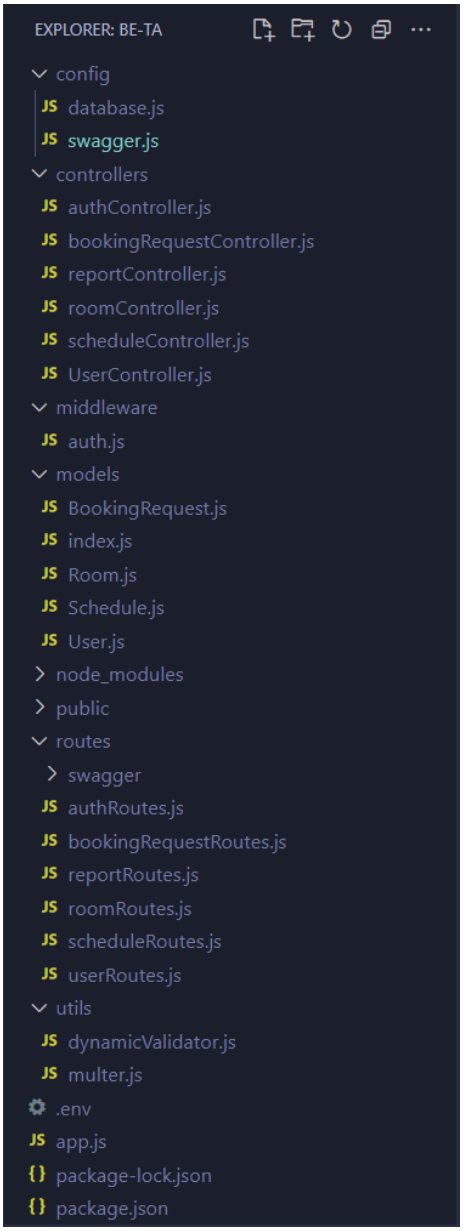
No	Nama Perangkat	Sumber	Deskripsi
1.	Visual Studio Code	https://code.visualstudio.com/	Perangkat lunak yang digunakan sebagai code editor untuk menuliskan code pada saat membangun perangkat lunak.
2.	MySql	https://www.apachefriends.org/	Perangkat lunak yang digunakan sebagai database.
3.	Draw.io	https://app.diagrams.net/	Perangkat lunak yang digunakan untuk membuat diagram pada saat tahap analisis dan perancangan.
4.	Node.js	https://nodejs.org/en	Digunakan untuk membuat server dan mengelola tugas server-side berkomunikasi dengan basis data, dan menangani permintaan HTTP.
5.	Postman	https://www.postman.com/	Perangkat lunak yang digunakan untuk melakukan tes terhadap API yang sedang digunakan.

5.1.2. Pengkodean

Tahap pengkodean merupakan tahap implementasi dari tahap perancangan menjadi kode-kode program. Pada tahap ini menggunakan bahasa Javascript dengan *Library* React.JS untuk frontend dan Node.JS dengan framework Express.JS untuk backend.

5.1.2.1. Struktur Folder Backend

Struktur folder backend pada aplikasi yang dibangun dapat dilihat pada gambar berikut:



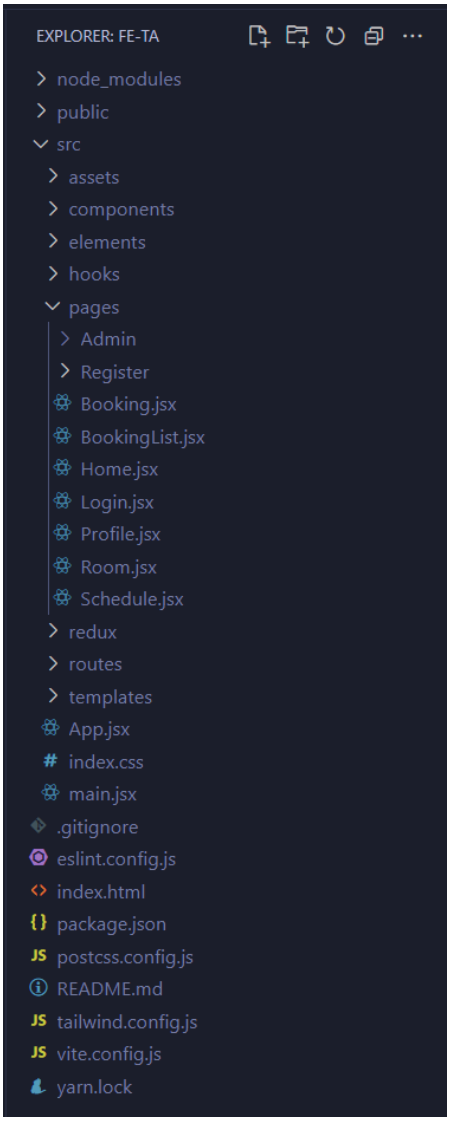
Gambar 5.1 Struktur Folder Backend

Tabel 5.3 Daftar Nama Folder dan File Backend

No.	Nama Folder atau File	Deskripsi
1.	config	Didalam folder ini terdapat file yang digunakan untuk membuat koneksi dengan database dan konfigurasi Swagger untuk dokumentasi API
2.	controllers	Folder ini berisi beberapa file controller yang bertanggung jawab mengelola logika aplikasi,
3.	middleware	Folder ini menyimpan file yang digunakan untuk autentikasi dan validasi sebelum mengakses rute tertentu
4.	models	Folder ini berisi model data yang mendefinisikan struktur dan relasi data di database.
5.	routes	Folder ini digunakan untuk mengelola file rute API yang menghubungkan permintaan HTTP dengan controller terkait.
6.	utils	Folder ini berisi fungsi-fungsi utilitas atau helper yang bisa digunakan di berbagai bagian aplikasi yaitu seperti validasi dinamis dan pengelolaan upload file menggunakan multer
7.	.env	File ini digunakan untuk menyimpan variabel lingkungan (environment variables) seperti kredensial database, API key, dan informasi sensitif lainnya.
8.	app.js	File ini merupakan entry point utama aplikasi yang menginisialisasi server dan konfigurasi rute.

5.1.2.2. Struktur Folder Frontend

Struktur folder frontend pada aplikasi yang dibangun dapat dilihat pada gambar berikut:



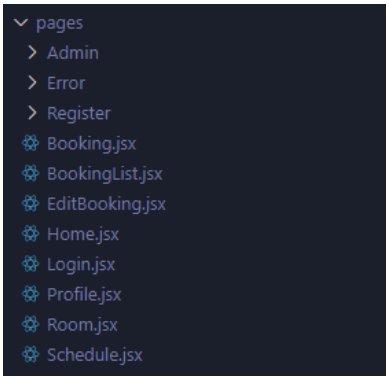
Gambar 5.2 Struktur Folder Frontend

Tabel 5.4 Daftar Nama Folder dan File Frontend

No.	Nama Folder dan File	Deskripsi
1.	assets	Folder ini digunakan untuk menyimpan aset yang digunakan dalam aplikasi seperti gambar dan font.
2.	components	Folder ini digunakan untuk menyimpan file-file komponen React yang dapat digunakan di berbagai halaman
3.	elements	Folder ini digunakan untuk menyimpan elemen UI kecil yang dapat dipakai ulang di beberapa komponen atau halaman
4.	hooks	Folder ini terdapat file yang digunakan untuk membungkus logika tertentu dalam aplikasi agar dapat digunakan ulang di berbagai komponen
5.	pages	Folder ini digunakan untuk menyimpan file berbagai halaman dalam aplikasi
6.	redux	Folder ini digunakan untuk menyimpan file yang berhubungan dengan manajemen state menggunakan Redux (reducers, actions, store).
7.	routes	Folder ini berisi file yang digunakan mengatur navigasi antara halaman-halaman yang berbeda dalam aplikasi.
8.	templates	Folder ini menyimpan file layout yang dapat digunakan ulang diberbagai halaman
9.	App.jsx	File ini digunakan untuk menyimpan rute/navigasi utama antara halaman-halaman di aplikasi

5.1.2.3. View Pada Aplikasi Web

View menerima data dari komponen lain dan menampilkan tampilan atau antarmuka pengguna, tidak adak ada ketentuan pada React,js untuk menyimpan kelas view. Akan tetapi untuk mempermudah dalam mengidentifikasi struktur folder.



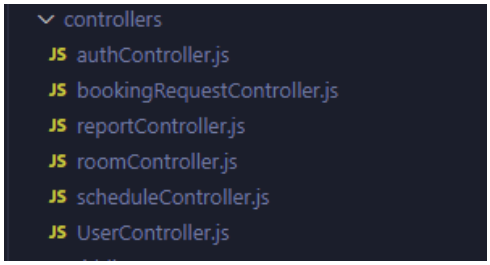
Gambar 5.3 View Pada Aplikasi

Tabel 5.5 Daftar Halaman Pada Aplikasi Web

No.	Halaman	Deskripsi
1.	Admin	Folder Admin berisi file halaman-halaman yang dikhususkan untuk admin.
2.	Error	Folder Error berisi file halaman yang ditampilkan apaila alamat yang dicari tidak ditemukan
3.	Register	Folder Register berisi file halaman-halaman untuk registrasi
4.	Booking.jsx	Menampilkan halaman untuk melakukan pengajuan peminjaman ruangan oleh peminjam
5.	BookingList.jsx	Menampilkan halaman riwayat pengajuan yang sudah dilakukan oleh peminjam
6.	EditBooking.jsx	Menampilkan halaman untuk melakukan perubahan pengajuan
7.	Home.jsx	Menampilkan halaman utama peminjam
8.	Login.jsx	Menampilkan halaman login peminjam
9.	Profile.jsx	Menampilkan halaman profile peminjam
10.	Room.jsx	Menampilkan halaman daftar ruangan
11.	Schedule.jsx	Menampilkan halaman jadwal ruangan

5.1.2.4. Controller pada Aplikasi

Controller pada aplikasi ini memiliki 6 *controller* yaitu, auth, bookingRequest, report, room, schedule, dan user. Setiap controller bertanggung jawab untuk memastikan komunikasi yang lancar antara aplikasi dan database melalui API endpoint, sehingga setiap proses berjalan sesuai kebutuhan.



Gambar 5.4 Controller pada Aplikasi

a. Controller Auth

Controller auth memiliki fungsi utama untuk menangani autentikasi pengguna, Hal ini mencakup:

- login : memverifikasi kredensial pengguna agar dapat mengakses sistem.

- checkEmail : mengecek email pengguna saat membuat akun baru.
- register : membuat akun baru dengan menambahkan data pengguna ke database.
- logout : mengakhiri sesi pengguna yang sedang aktif.

b. Controller User

Controller users bertugas untuk mengelola data penggna. Beberapa fungsinya yaitu :

- getUsers : digunakan untuk mendapatkan semua data pengguna.
- getUserById : digunakan untuk mendapatkan data pegguna berdasarkan id
- createUser : digunakan untuk menambah data pengguna.
- updateUser : digunakan untuk mengubah data pengguna berdasarkan id pengguna.
- deleteUser : digunakan untuk menghapus data pengguna berdasarkan id pengguna.

c. Controller Room

Controller Room bertugas untuk mengelola data ruangan. Beberapa fungsinya yaitu :

- getRooms : digunakan untuk mendapatkan semua data ruangan.
- getRoomById : digunakan untuk mendapatkan data ruangan berdasarkan id
- createRoom : digunakan untuk menambah data ruangan.
- updateRoom : digunakan untuk mengubah data ruangan berdasarkan id ruangan.
- deleteRoom : digunakan untuk menghapus data ruangan berdasarkan id ruangan.

d. Controller Schedule

Controller Schedule bertugas untuk mengelola data jadwal ruangan. Beberapa fungsinya yaitu :

- getSchedules : digunakan untuk mendapatkan semua data jadwal ruangan.
- getScheduleById : digunakan untuk mendapatkan data jadwal ruangan berdasarkan id
- createSchedule : digunakan untuk menambah data jadwal ruangan.
- updateSchedule : digunakan untuk mengubah data jadwal ruangan berdasarkan id jadwal ruangan.
- deleteSchedule : digunakan untuk menghapus data jadwal ruangan berdasarkan id jadwal ruangan.

e. Controller Booking

Controller Booking bertugas untuk mengelola data pengajuan ruangan. Beberapa fungsinya yaitu :

- getBookings : digunakan untuk mendapatkan semua data pengajuan ruangan.
- getBookingById : digunakan untuk mendapatkan data pengajuan ruangan berdasarkan id
- createBooking : digunakan untuk menambah data pengajuan ruangan.
- updateBooking : digunakan untuk mengubah data pengajuan ruangan berdasarkan id pengajuan ruangan.
- deleteBooking : digunakan untuk menghapus data pengajuan ruangan berdasarkan id pengajuan ruangan.

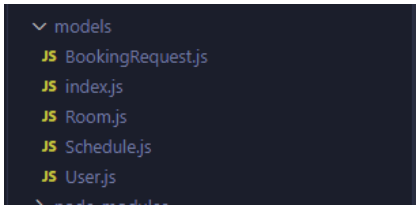
f. Controller Report

Controller Report berfungsi untuk mendapatkan dan mencetak data laporan. Berikut merupakan fungsi dari controller report:

- getBookingReport : digunakan untuk mendapatkan data laporan, berdasarkan tanggal dan ruangan yang dipilih.
- exportBooking : digunakan untuk mengekspor data laporan yang ada ke dalam format tertentu.

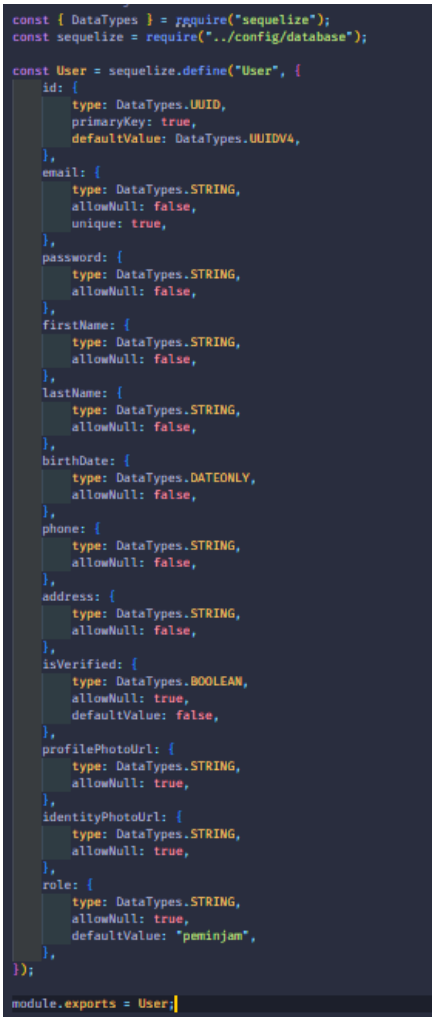
5.1.2.5. Model pada Aplikasi

Model pada aplikasi ini terdapat 4 *model* yaitu, User, BookingRequest, Report, Room, dan Schedule. Sedangkan index merupakan file utama yang mengatur relasi antar model.



Gambar 5.5 Model pada Aplikasi

a. Model User



```
const { DataTypes } = require("sequelize");
const sequelize = require("../config/database");

const User = sequelize.define("User", {
    id: {
        type: DataTypes.UUID,
        primaryKey: true,
        defaultValue: DataTypes.UUIDV4,
    },
    email: {
        type: DataTypes.STRING,
        allowNull: false,
        unique: true,
    },
    password: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    firstName: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    lastName: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    birthDate: {
        type: DataTypes.DATEONLY,
        allowNull: false,
    },
    phone: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    address: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    isVerified: {
        type: DataTypes.BOOLEAN,
        allowNull: true,
        defaultValue: false,
    },
    profilePhotoUrl: {
        type: DataTypes.STRING,
        allowNull: true,
    },
    identityPhotoUrl: {
        type: DataTypes.STRING,
        allowNull: true,
    },
    role: {
        type: DataTypes.STRING,
        allowNull: true,
        defaultValue: "peminjam",
    },
});

module.exports = User;
```

Gambar 5.6 Potongan Kode Model User

b. Model Room

```
const { DataTypes } = require("sequelize");
const sequelize = require("../config/database");

const Room = sequelize.define("Room", {
    id: {
        type: DataTypes.UUID,
        primaryKey: true,
        defaultValue: DataTypes.UUIDV4,
    },
    name: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    capacity: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    description: {
        type: DataTypes.STRING,
        allowNull: false,
    },
    roomPhotoUrl: {
        type: DataTypes.TEXT, // Menggunakan TEXT untuk menyimpan JSON
        allowNull: true,
        get() {
            // Mengonversi dari string JSON ke array saat membaca
            const rawValue = this.getDataValue("roomPhotoUrl");
            return rawValue ? JSON.parse(rawValue) : null;
        },
        set(value) {
            // Mengonversi dari array ke string JSON saat menyimpan
            this.setDataValue("roomPhotoUrl", JSON.stringify(value));
        },
    },
});

module.exports = Room;
```

Gambar 5.7 Potongan Kode Model Room

c. Model Schedule

```
const { DataTypes } = require("sequelize");
const Room = require("./Room")
const sequelize = require("../config/database");

const Schedule = sequelize.define("Schedule", {
    id: {
        type: DataTypes.UUID,
        primaryKey: true,
        defaultValue: DataTypes.UUIDV4,
    },
    session: {
        type: DataTypes.STRING,
        allowNull: true,
    },
    startTime: {
        type: DataTypes.TIME,
        allowNull: false,
    },
    endTime: {
        type: DataTypes.TIME,
        allowNull: false,
    },
    roomId: {
        type: DataTypes.UUID,
        references: {
            model: Room,
            key: "id",
        },
        allowNull: false,
    },
});

module.exports = Schedule;
```

Gambar 5.8 Potongan Kode Model Schedule

d. Model Booking

```
const { DataTypes } = require("sequelize");
const User = require("./User");
const Room = require("./Room");
const Schedule = require("./Schedule");
const sequelize = require("../config/database");

const BookingRequest = sequelize.define("BookingRequest", {
    id: {
        type: DataTypes.UUID,
        primaryKey: true,
        defaultValue: DataTypes.UUIDV4,
    },
    institutionName: {
        type: DataTypes.STRING,
        allowNull: true,
    },
    activityName: {
        type: DataTypes.STRING,
        allowNull: true,
    },
    status: {
        type: DataTypes.STRING,
        allowNull: true,
        defaultValue: "Menunggu",
    },
    dateActivity: {
        type: DataTypes.DATEONLY,
        allowNull: false,
    },
    submissionFileUrl: {
        type: DataTypes.STRING,
        allowNull: true,
    },
    descriptionFileUrl: {
        type: DataTypes.STRING,
        allowNull: true,
    },
    statementFileUrl: {
        type: DataTypes.STRING,
        allowNull: true,
    },
    userId: {
        type: DataTypes.UUID,
        references: {
            model: User,
            key: "id",
        },
        allowNull: false,
    },
    roomId: {
        type: DataTypes.UUID,
        references: {
            model: Room,
            key: "id",
        },
        allowNull: false,
    },
    scheduleId: {
        type: DataTypes.UUID,
        references: {
            model: Schedule,
            key: "id",
        },
        allowNull: false,
    },
});

module.exports = BookingRequest;
```

Gambar 5.9 Potongan Kode Model Booking

e. index

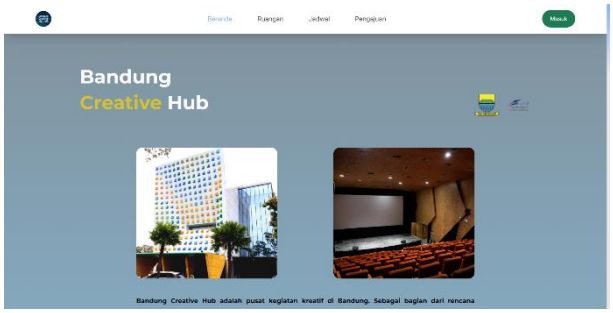
```
const User = require("./User");
const Room = require("./Room");
const Schedule = require("./Schedule");
const BookingRequest = require("./BookingRequest");

// Relasi User dan BookingRequest
User.hasMany(BookingRequest, { foreignKey: "userId" });
BookingRequest.belongsTo(User, { foreignKey: "userId" });

// Relasi Room dan BookingRequest
Room.hasMany(BookingRequest, { foreignKey: "roomId" });
BookingRequest.belongsTo(Room, { foreignKey: "roomId" });

// Relasi Room dan Schedule
Room.hasMany(Schedule, { foreignKey: "roomId" });
Schedule.belongsTo(Room, { foreignKey: "roomId" });

// Relasi Schedule dan BookingRequest
Schedule.hasMany(BookingRequest, { foreignKey: "scheduleId" });
BookingRequest.belongsTo(Schedule, { foreignKey: "scheduleId" });

module.exports = {
    User,
    Room,
    Schedule,
    BookingRequest,
};
```

Gambar 5.10 Potongan Kode Index

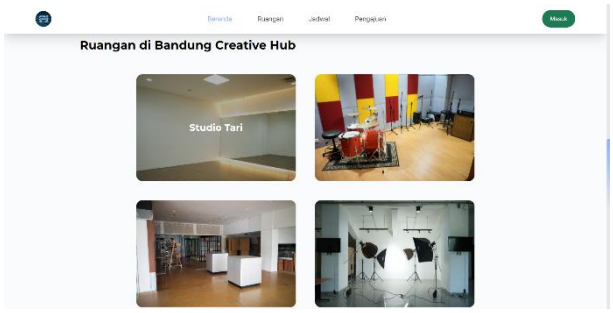
5.1.2.6. Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna merupakan hasil penerapan dari perancangan antarmuka pengguna yang telah disusun sebelumnya. Berikut ini adalah tampilan antarmuka aplikasi web peminjaman ruangan Bandung Creative Hub

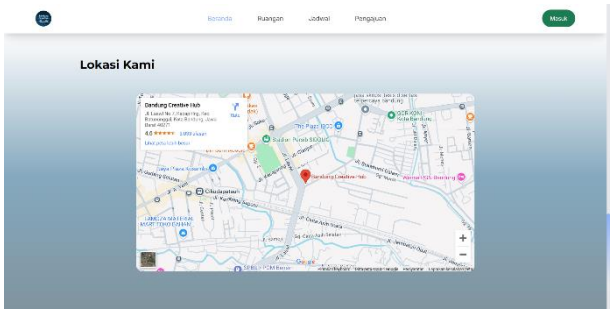
Tabel 5.6 Antarmuka Halaman Informasi

Antarmuka Halaman Informasi

Deskripsi
Ini merupakan halaman utama dari aplikasi ini. Pada bagian utama halaman terdapat beberapa informasi seperti foto dan deskripsi tentang Bandung Creative Hub.

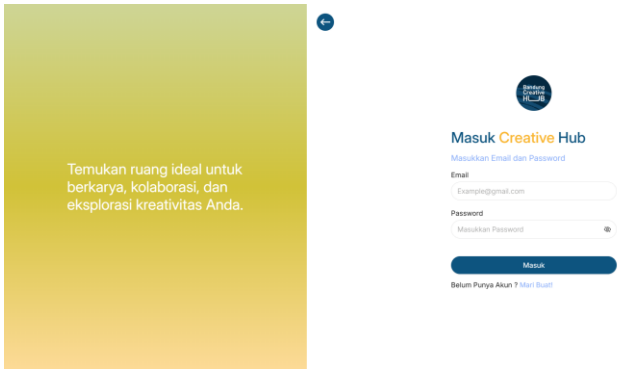
Tabel 5.7 Antarmuka Halaman Informasi Ruangan

Antarmuka Halaman Informasi Ruangan

Deskripsi
Ini merupakan tampilan lanjutan pada halaman utama yang menampilkan beberapa foto ruangan di Bandung Creative Hub

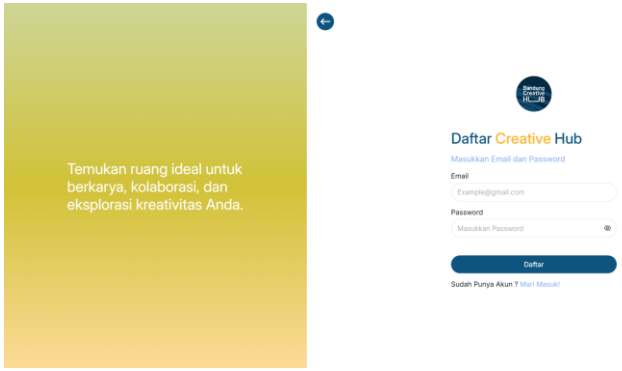
Tabel 5.8 Antarmuka Halaman Informasi Lokasi

Antarmuka Halaman Informasi Lokasi

Deskripsi
Ini merupakan tampilan lanjutan pada halaman utama yang menampilkan informasi lokasi Bandung Creative Hub

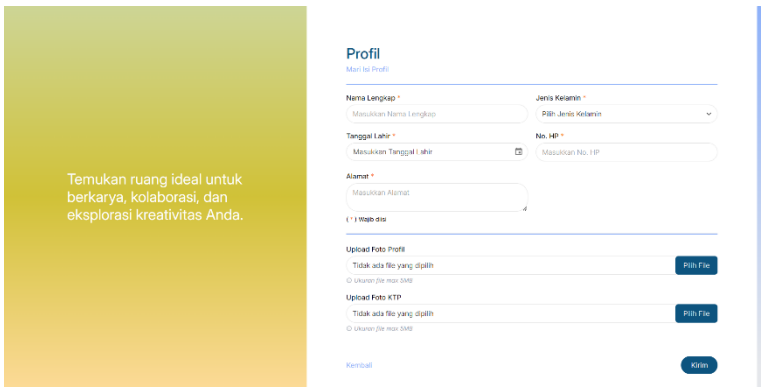
Tabel 5.9 Antarmuka Halaman Login

Antarmuka Halaman Login

Deskripsi
Ini merupakan halaman login yang digunakan oleh pengguna dengan memasukkan email dan password. Jika belum memiliki akun, pengguna dapat memilih link “Mari Buat” untuk mengakses halaman pendaftaran.

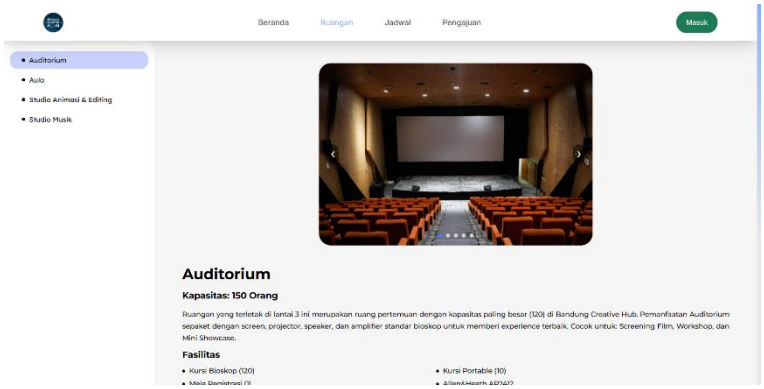
Tabel 5.10 Antarmuka Halaman Register

Antarmuka Halaman Register

Deskripsi
Ini merupakan halaman registrasi yang digunakan oleh pengguna dengan memasukkan email dan password. Jika sudah memiliki akun, pengguna dapat memilih link “Mari Masuk” untuk mengakses halaman login.

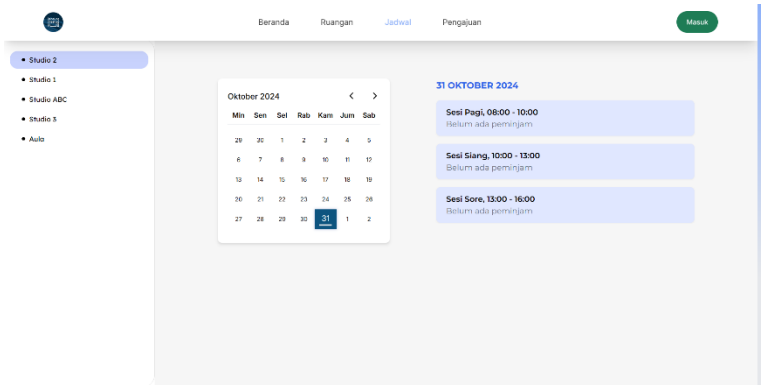
Tabel 5.11 Antarmuka Halaman Register 2

Antarmuka Halaman Register 2

Deskripsi
Ini merupakan halaman registrasi yang kedua yang dimana pengguna harus mengisi form pendaftaran yang lebih lengkap seperti nama lengkap, jenis kelamin, tanggal lahir, nomer HP, alamat, foto profil dan foto KTP. Setelah mengisi data, pengguna dapat mengklik tombol “Kirim” untuk melanjutkan atau “Kembali” untuk kembali ke form halaman registrasi sebelumnya.

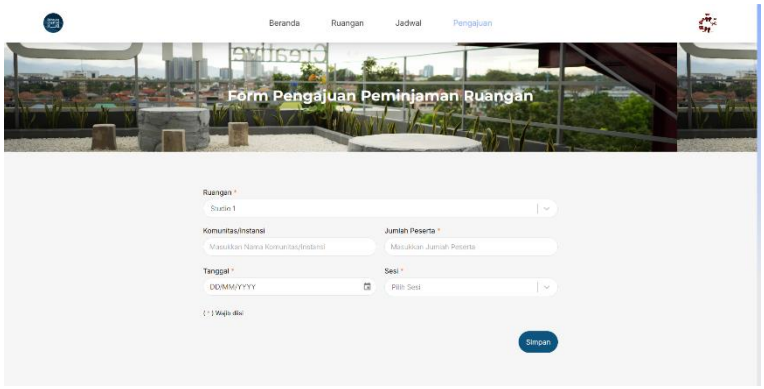
Tabel 5.12 Antarmuka Halaman Detail Ruangan

Antarmuka Halaman Detail Ruangan

Deskripsi
Ini merupakan halaman detail ruangan yang bisa digunakan oleh pengguna untuk melihat informasi detail ruangan.

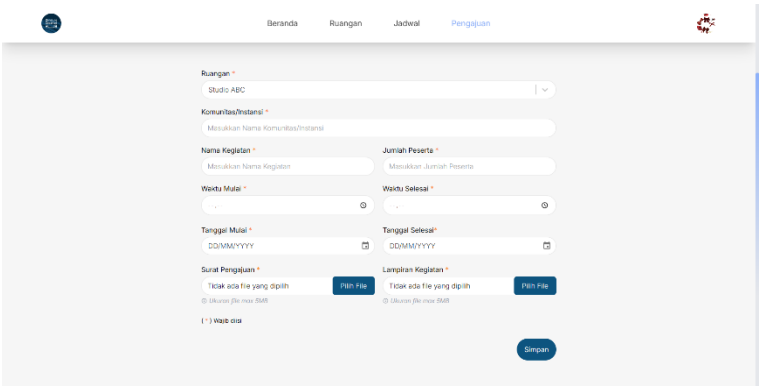
Tabel 5.13 Antarmuka Halaman Jadwal Ruangan

Antarmuka Halaman Jadwal Ruangan

Deskripsi
Ini merupakan halaman jadwal ruangan yang bisa digunakan oleh pengguna untuk melihat informasi ketersediaan ruangan.

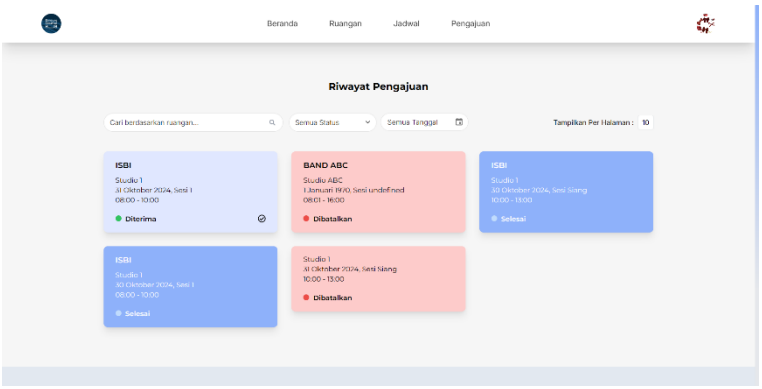
Tabel 5.14 Antarmuka Halaman Pengajuan

Antarmuka Halaman Pengajuan

Deskripsi
Ini merupakan halaman form pengajuan untuk jenis ruangan yang pertama terdapat form input untuk memilih ruangan, komunitas/instansi, jumlah peserta, tanggal, dan sesi. Jika sudah mengisi data pengguna bisa mengklik tombol “Simpan” untuk menyimpan pengajuan.

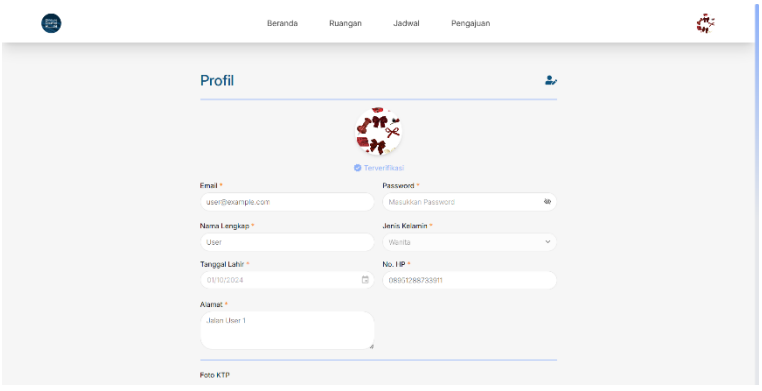
Tabel 5.15 Antarmuka Halaman Pengajuan 2

Antarmuka Halaman Pengajuan 2

Deskripsi
Ini merupakan halaman form pengajuan untuk jenis ruangan yang kedua terdapat form input untuk memilih ruangan, komunitas/instansi, nama kegiatan, jumlah peserta, waktu mulai, waktu selesai, tanggal mulai, tanggal selesai, surat pengajuan dan lampiran kegiatan. Jika sudah mengisi data pengguna bisa mengklik tombol “Simpan” untuk menyimpan pengajuan.

Tabel 5.16 Antarmuka Halaman Riwayat Pengajuan

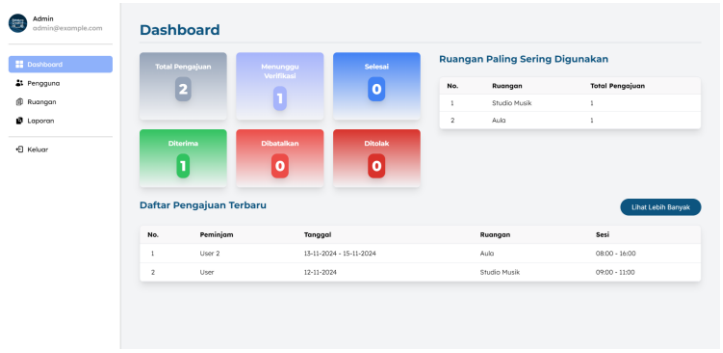
Antarmuka Halaman Riwayat Pengajuan

Deskripsi
Ini merupakan halaman riwayat pengajuan. Dihalaman ini terdapat informasi pengajuan ruangan yang sudah dilakukan dalam satu pengajuan terdapat informasi nama grup/instansi, ruangan, tanggal, waktu dan status pengajuan. Dibagian atas terdapat fitur pencarian serta filter berdasarkan status pengajuan dan tanggal kegiatan.

Tabel 5.17 Antarmuka Halaman Profil

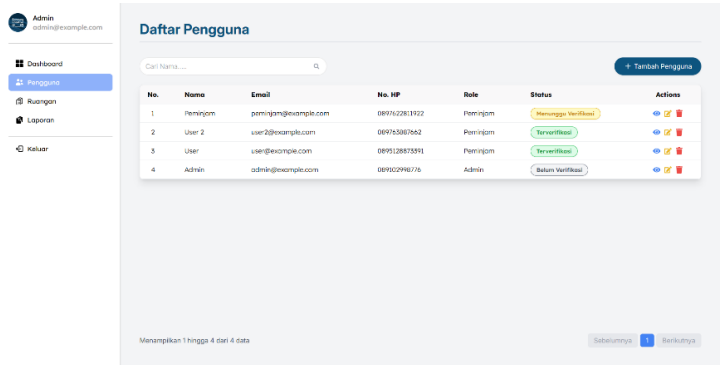
Rancangan Antarmuka Halaman Profil

Deskripsi
Ini merupakan halaman profil dari pengguna dengan form input seperti email, password, nama lengkap, jenis kelamin, tanggal lahir, no

HP, alamat dan foto KTP. Di bagian atas terdapat foto profil dengan status verifikasi dan opsi untuk mengedit.

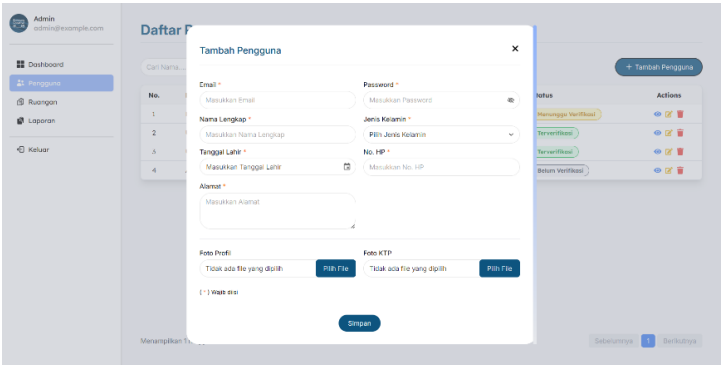
Tabel 5.18 Antarmuka Halaman Dashboard

Antarmuka Halaman Dashboard

Deskripsi
Ini merupakan halaman dashboard untuk petugas. Dibagian sebelah kiri terdapat beberapa menu yang ditampilkan pada semua halaman petugas seperti *Dashboard, User, Ruangan* dengan submenu *Pengajuan Ruangan, Data Ruangan* dan Jadwal Ruangan, *Laporan,* serta tombol *Logout.* Bagian utama halaman menampilkan ringkasan statistik pengajuan serta terdapat dua tabel yaitu untuk menampilkan ruangan paling sering digunakan dan daftar pengajuan terbaru.

Tabel 5.19 Antarmuka Halaman Daftar Pengguna

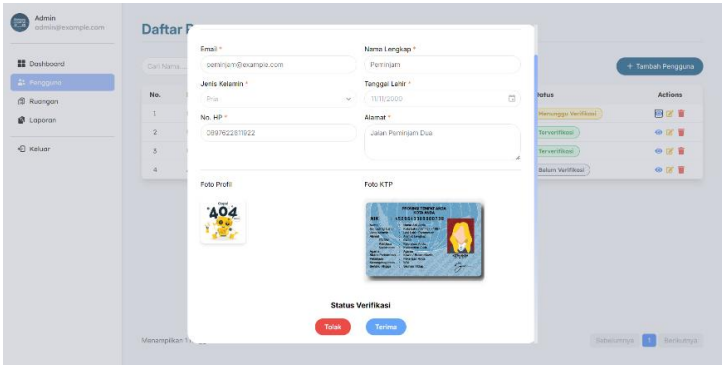
Antarmuka Halaman Daftar Pengguna

Deskripsi
Ini merupakan halaman daftar pengguna. Pada halaman ini terdapat kolom pencarian, tombol “Tambah Pengguna” dan tabel yang menampilkan daftar pengguna.

Tabel 5.20 Antarmuka Halaman Tambah Pengguna

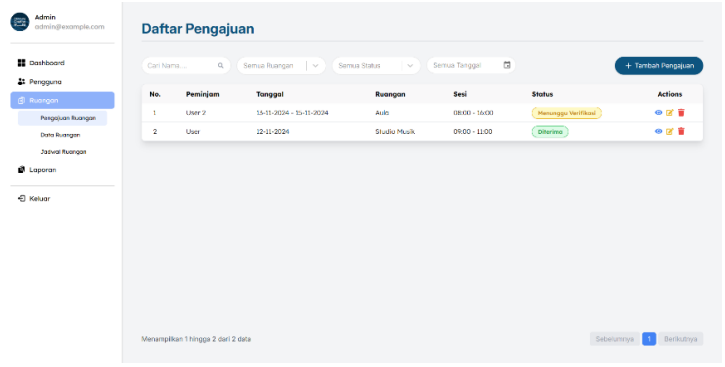
Antarmuka Halaman Tambah Pengguna

Deskripsi
Ini merupakan tampilan untuk menambah data pengguna, terdapat form inputan untuk email, password, nama lengkap, jenis kelamin,

tanggal lahir, no hp, alamat, foto profil dan foto KTP. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

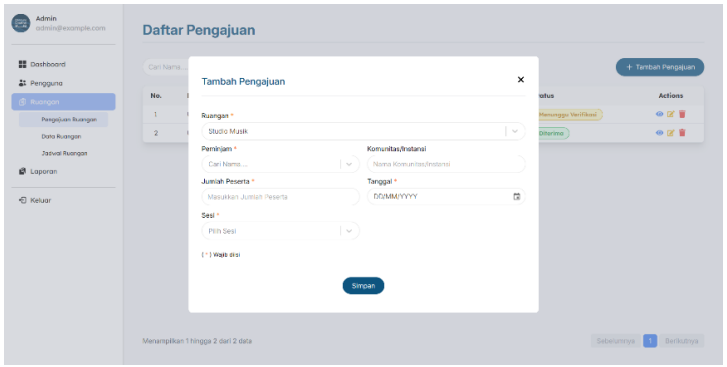
Tabel 5.21 Antarmuka Halaman Detail Pengguna

Antarmuka Halaman Detail Pengguna

Deskripsi
Ini merupakan tampilan untuk menampilkan detail data dari pengguna yang bisa digunakan petugas untuk melakukan verifikasi data.

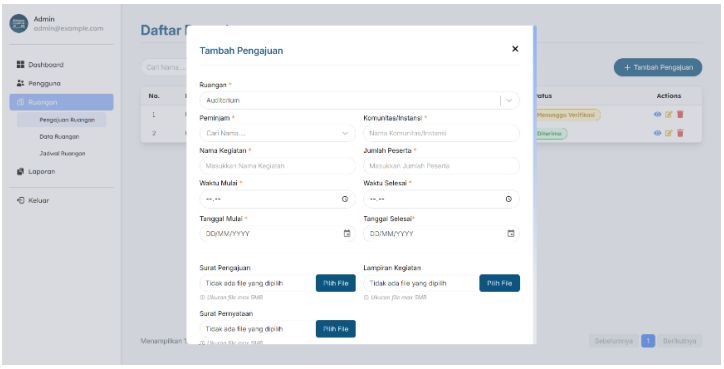
Tabel 5.22 Antarmuka Halaman Daftar Pengajuan

Antarmuka Halaman Daftar Pengajuan

Deskripsi
Ini merupakan halaman daftar pengajuan. Pada halaman ini terdapat kolom pencarian, filter berdasarkan ruangan, status, tanggal, tombol “Tambah Pengajuan” dan tabel yang menampilkan daftar pengajuan.

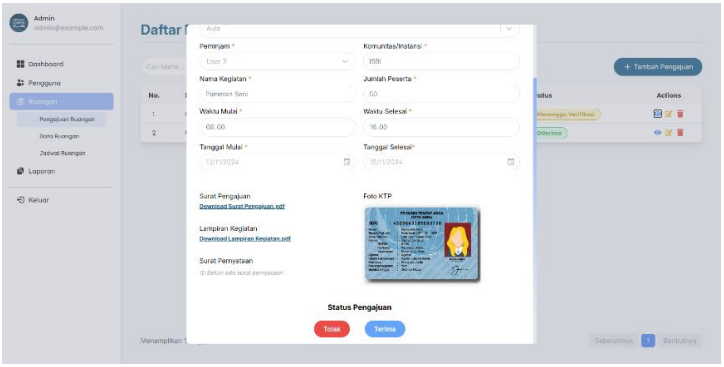
Tabel 5.23 Antarmuka Halaman Tambah Pengajuan

Antarmuka Halaman Tambah Pengajuan

Deskripsi
Ini merupakan tampilan untuk menambah data pengajuan dengan jenis ruangan yang pertama, terdapat form inputan untuk ruangan, peminjam, komunitas/instansi, jumlah peserta, tanggal dan sesi. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

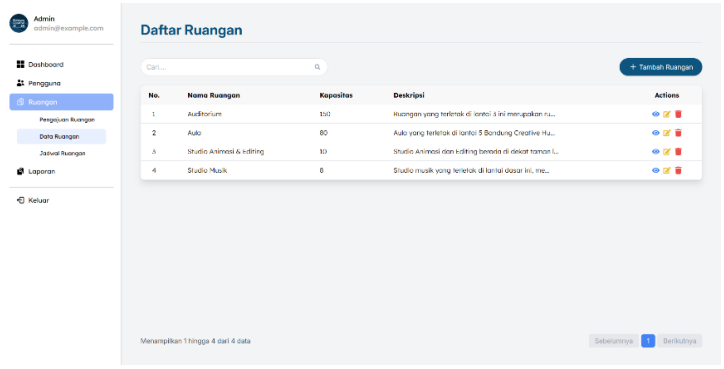
Tabel 5.24 Antarmuka Halaman Tambah Pengajuan 2

Antarmuka Halaman Tambah Pengajuan 2

Deskripsi
Ini merupakan tampilan untuk menambah data pengajuan dengan jenis ruangan yang kedua, terdapat form inputan untuk ruangan, peminjam, komunitas/instansi, kegiatan, jumlah peserta, waktu mulai, waktu selesai, tanggal mulai, tanggal selesai, surat pengajuan, lampiran kegiatan, dan surat pernyataan. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

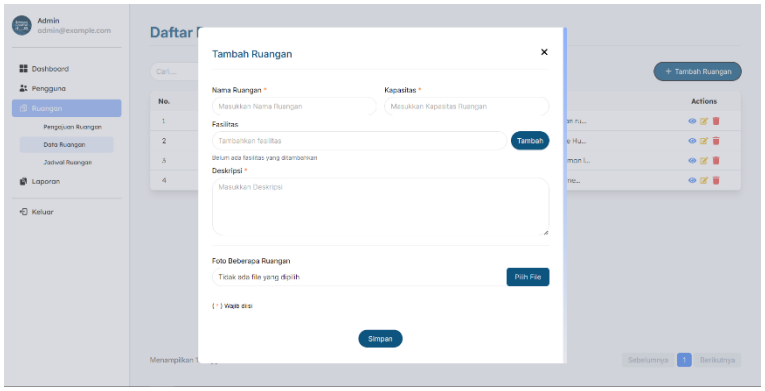
Tabel 5.25 Antarmuka Halaman Detail Pengajuan

Antarmuka Halaman Detail Pengajuan

Deskripsi
Ini merupakan tampilan untuk menampilkan detail data dari pengajuan yang bisa digunakan petugas untuk melakukan verifikasi data. Petugas bisa memilih tombol “Terima” atau “Tolak” untuk melakukan verifikasi.

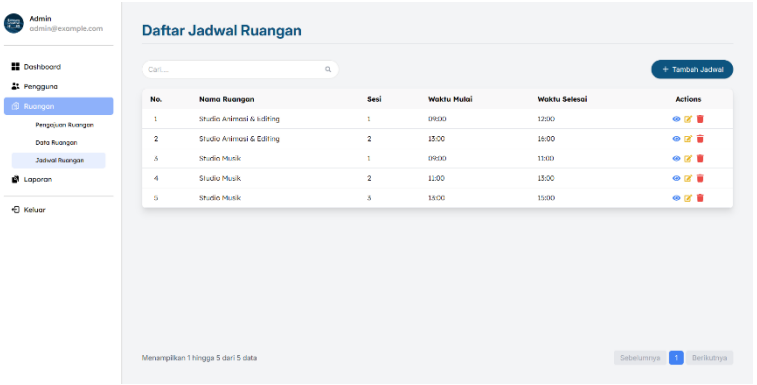
Tabel 5.26 Rancangan Antarmuka Halaman Daftar Ruangan

Antarmuka Halaman Daftar Ruangan

Deskripsi
Ini merupakan halaman daftar ruangan. Pada halaman ini terdapat kolom pencarian, tombol “Tambah Ruangan” dan tabel yang menampilkan daftar ruangan.

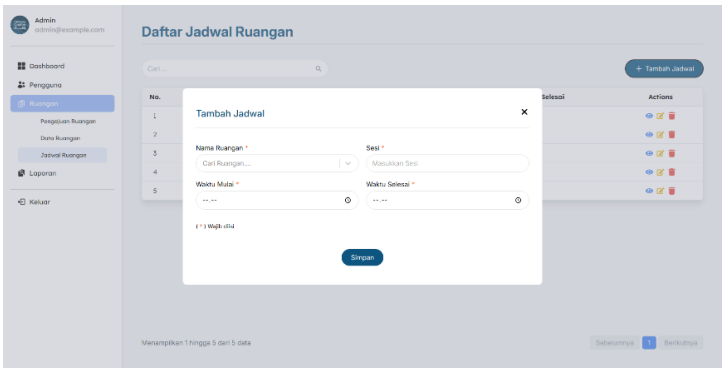
Tabel 5.27 Antarmuka Halaman Tambah Ruangan

Antarmuka Halaman Tambah Ruangan

Deskripsi
Ini merupakan tampilan untuk menambah data ruangan, terdapat form inputan untuk nama, kapasitas, fasilitas, deskripsi, dan foto ruangan. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

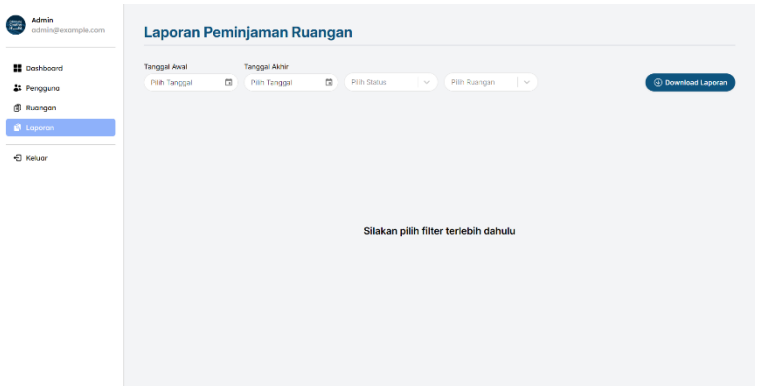
Tabel 5.28 Antarmuka Halaman Daftar Jadwal Ruangan

Antarmuka Halaman Daftar Jadwal Ruangan

Deskripsi
Ini merupakan halaman daftar jadwal ruangan. Pada halaman ini terdapat kolom pencarian, tombol “Tambah Jadwal” dan tabel yang menampilkan daftar jadwal ruangan.

Tabel 5.29 Antarmuka Halaman Tambah Jadwal Ruangan

Antarmuka Halaman Tambah Jadwal Ruangan

Deskripsi
Ini merupakan tampilan untuk menambah data jadwal, terdapat form inputan untuk ruangan, nama sesi, waktu mulai dan waktu selesai. Jika petugas sudah mengisi semua data bisa mengklik tombol “Simpan” untuk menyimpan data.

Tabel 5.30 Antarmuka Halaman Laporan

Antarmuka Halaman Laporan

Deskripsi
Ini merupakan halaman laporan peminjaman ruangan. Pada halaman ini petugas harus memilih tanggal awal, tanggal akhir, status, dan ruangan. Jika petugas sudah mengisi semua data nantinya akan muncul daftar laporan berdasarkan filter yang dipilih. Petugas juga bisa mendowload laporan dengan klik “Download Laporan”

5.2. Pengujian

5.2.1. Unit Testing

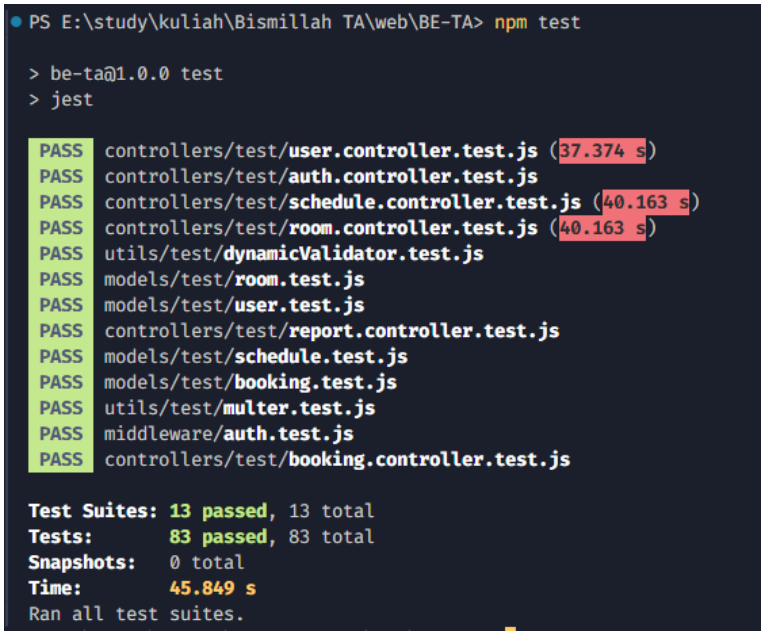
Unit testing merupakan metode pengujian kode yang dibuat, akan diuji dalam keaadan berhasil dan gagal dijalankan. Berikut merupakan kode dan hasil dari unit testing, dapat dilihat pada tabel 5.31

Tabel 5.31 Kode Unit Test

Kode Unit Test
Deskripsi
Gambar diatas merupakan salah satu kode saat melakukan unit testing untuk menguji endpoint POST /booking/booking-requests, Terdapat dua skenario pengujian yaitu membuat permintaan booking berhasil dan menangani konflik booking. Pengujian ini membantu memastikan bahwa sistem dapat menangani pembuatan booking baru dengan benar, baik ketika tersedia maupun ketika ada konflik jadwal.

```
describe("POST /booking/booking-requests", () => {
    it("harus membuat permintaan booking", async () => {
        Booking.findOne.mockResolvedValue(null);
        Booking.create.mockResolvedValue(mockBookingData);

        const response = await request(app)
            .post("/booking/booking-requests")
            .send({
                userId: 1,
                roomId: 1,
                scheduleId: 1,
                dateActivity: "2024-11-04",
                startDate: "2024-11-04",
                endDate: "2024-11-05",
            });

        expect(response.status).toBe(201);
        expect(response.body).toHaveProperty("id", mockBookingData.id);
    });

    it("harus mengembalikan error jika booking bentrok dengan booking yang sudah ada", async () => {
        Booking.findOne.mockResolvedValue(mockBookingData);

        const response = await request(app)
            .post("/booking/booking-requests")
            .send({
                userId: 1,
                roomId: 1,
                scheduleId: 1,
                dateActivity: "2024-11-04",
                startDate: "2024-11-04",
                endDate: "2024-11-05",
            });

        expect(response.status).toBe(400);
        expect(response.body).toHaveProperty(
            "error",
            "Ruangan tidak tersedia"
        );
    });
});
```

Tabel 5.32 Hasil Unit Test

Hasil Unit Test

Deskripsi
Gambar di atas menunjukkan hasil unit test menggunakan Jest, di mana seluruh test suite (13) dan test case (83) berhasil dijalankan dengan status **PASS**, ditandai dengan warna hijau. Waktu total eksekusi adalah 45,849 detik.

Hasil pengujian pada tabel di atas menunjukkan bahwa seluruh 13 suite pengujian telah berhasil dijalankan dengan total 83 pengujian sukses tanpa ada kegagalan atau. Pengujian ini meliputi berbagai modul dalam aplikasi, seperti pengujian pada controller (user.controller.test.js, auth.controller.test.js, schedule.controller.test.js, room.controller.test.js, dan booking.controller.test.js), utilitas (dynamicValidator.test.js, multer.test.js), middleware (auth.test.js), serta model (user.test.js, report.controller.test.js, schedule.test.js, dan booking.test.js).

Pengujian berhasil diselesaikan dalam waktu total 45,849 detik, dengan beberapa pengujian membutuhkan waktu lebih lama, seperti schedule.controller.test.js dan room.controller.test.js, yang masing-masing memakan waktu sekitar 40 detik. Hasil ini menunjukkan bahwa sistem telah diuji secara menyeluruh untuk memastikan fungsionalitas setiap bagian aplikasi berjalan sesuai dengan yang diharapkan.

BAB 6
KESIMPULAN DAN SARAN

Bab ini menjelaskan kesimpulan yang dapat diambil dari hasil penelitian yang dilakukan serta saran yang diberikan untuk pengembangan selanjutnya

6.1. Kesimpulan

Berdasarkan hasil dari tugas akhir ini, dapat disimpulkan bahwa pengembangan aplikasi peminjaman ruangan berbasis web untuk Bandung Creative Hub berhasil dilakukan sesuai dengan tujuan awal yang dirumuskan, berikut adalah poin-poin kesimpulan yang dapat disampaikan :

1. **Mempermudah Proses Pengajuan Peminjaman Ruangan**

   Aplikasi yang dikembangkan memberikan kemudahan bagi peminjam untuk melakukan pengajuan peminjaman ruangan secara online tanpa harus datang langsung ke lokasi.

2. **Mendukung Pengelolaan Berkas Secara Terstruktur**

   Fitur pengelolaan data pengguna dan peminjaman yang terintegrasi dalam aplikasi membantu petugas mengelola berkas-berkas pengajuan dengan lebih terstruktur dan terorganisir, sehingga mengurangi beban kerja manual yang sebelumnya dilakukan secara konvensional menggunakan media kertas.

3. **Mengurangi Potensi Kesalahan dalam Penjadwalan Ruangan**

   Aplikasi ini mengurangi potensi bentrok jadwal ruangan melalui sistem yang mampu memvalidasi dan mengelola penjadwalan secara otomatis.

4. **Mempercepat Proses Pembuatan Laporan**

   Dengan fitur digitalisasi pengelolaan data, aplikasi ini mempercepat proses pembuatan laporan yang sebelumnya memakan banyak waktu karena harus merekap data secara manual.

Dengan demikian, aplikasi ini dapat memberikan pengalaman yang lebih baik bagi para peminjam dan petugas Bandung Creative Hub.

6.2. Saran

Berdasarkan kesimpulan dari penelitian ini, berikut adalah saran yang diajukan untuk pengembangan aplikasi peminjaman ruangan berbasis web agar dapat memberikan manfaat yang lebih optimal:

1. **Penambahan Fitur Notifikasi**

   Untuk meningkatkan kenyamanan pengguna dan efisiensi kerja petugas, disarankan menambahkan fitur notifikasi. Fitur ini dapat memberikan informasi kepada peminjam terkait status pengajuan, dan pengingat waktu peminjaman melalui email atau notifikasi dalam aplikasi. Hal ini selaras dengan tujuan aplikasi untuk mengurangi kebutuhan interaksi langsung, sekaligus memastikan peminjam mendapatkan informasi secara real-time tanpa harus memeriksa status secara manual.

2. **Pencatatan Masalah Fasilitas dan Feedback Pengguna**

Mengingat pentingnya pemeliharaan fasilitas, disarankan menambahkan fitur pencatatan masalah atau kerusakan pada ruangan dan fasilitas yang dapat diakses oleh petugas maupun peminjam. Fitur ini akan membantu mengidentifikasi kebutuhan perbaikan lebih cepat, sehingga menjaga kualitas pelayanan ruangan. Selain itu, sistem feedback pengguna juga dapat diterapkan untuk mendapatkan masukan terkait pengalaman penggunaan aplikasi dan fasilitas ruangan, yang dapat menjadi dasar pengembangan lebih lanjut.

3. **Tahapan Maintenance**

   Meskipun tahap maintenance tidak termasuk dalam lingkup penelitian ini, disarankan agar dilakukan pemeliharaan dan perbaikan berkala setelah aplikasi diterapkan. Tahapan maintenance akan memastikan bahwa aplikasi tetap berfungsi dengan baik, memperbaiki bug yang mungkin muncul, dan menyesuaikan aplikasi dengan kebutuhan pengguna di masa mendatang.

Dengan aaran-saran ini diharapkan dapat menjadi dasar pengembangan lebih lanjut, sehingga aplikasi peminjaman ruangan berbasis web ini dapat memberikan manfaat yang lebih besar, baik bagi peminjam maupun pengelola atau petugas Bandung Creative Hub.

DAFTAR PUSTAKA

[AFR23] Afriansyah, R., Sholeh, M., & Andayanti, D., "Perancangan Aplikasi Pemrograman Antarmuka Berbasis Web Menggunakan Gaya Arsitektur Representasi Untuk Sistem Presensi Sekolah", Jurnal SCRIPT, Volume 11, No 1, Juni 2023

[AKH22] Akhdani, A.R., "Implementasi React. Js Pada Pengembangan Frontend Sistem Informasi Manajemen Kader Partai", 2022

[DHA20] Dhanardana, N.A.D., Rahayu, T., "Aplikasi Peminjaman Ruangan Berbasis Web Pada Direktorat Jenderal Hak Asasi Manusia", Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA), Agustus 2020

[FAJ17] Fajrin, R., "Pengembangan Sistem Informasi Geografis Berbasis Node.JS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia", Jurnal Komputer Terapan, Vol. 3, No. 1, Mei 2017

[HAN24] https://handsonreact.com/docs/architecture, 2024

[HAS21] Hasibuan, A.N., Dirgahayu, T., "Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna", journal.uii.ac.id/automata, Volume 2, No. 1, Januari 2021

[HEN14] Hendrianto, D.E., "Pembuatan Sistem Informasi Perpustakaan Berbasis Website Pada Sekolah Menegah Pertama Negeri 1 Donorojo Kabupaten Pacitan", IJNS – Indonesian Journal on Networking and Security, Volume 3, No 4, 2014

[KHU18] Khuat, Tung., "Developing a frontend application using ReactJS and Redux.", Laurea University of Applied Sciences, Leppävaara, 2018

[KUR20] Kurniawan, I., Humaira, & Rozi, F., "REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android", Jurnal Ilmiah Teknologi Sistem Informasi, Volume 1, No 4, Desember 2020

[LUM21] Lumba, E., "Pertukaran Data Pada Aplikasi Android Menggunakan Java Script Object Notation (Json) Dan Rest Api Dengan Retrofit 2", Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST), Maret 2021

[MAS12] Massé, M., "REST API Design Rulebook", O'Reilly Media, Sebastopol, 2012

[MIF17] Miftah, "Bandung Creative Hub, Surga Bagi Insan Kreatif" tersedia : Desember 29 2017, https://www.bandung.go.id/news/read/4339/bandung-creative-hub-surga-bagi-insan-kreatif , diakses : 21 Desember 2023

[MIN21] Mintoharjo, A., Ikhsan, F., "Buku Pedoman Teknis Pengelolaan Aplikasi Dan Dashboard Perhitungan Demand MPK Dan Nilai Capaian TKDN Berbasis Web Melalui SIMPK", Direktorat Kelembagaan dan Sumber Daya Konstruksi, Jakarta, 2021

[PET24] Petrenko, V., “Understanding Contemporary Web Application Architecture: Key Components, Best Practices, and Beyond”, tersedia : April 29 2024, https://litslink.com/blog/web-application-architecture , diakses : 7 Mei 2024

[PRE10] Pressman, R. S., “Software Engineering: A Practitioner's Approach, Seventh Edition”, McGraw-Hill, New York, 2010

[PUR10] Purnomosidi, B., “Model Interaksi Aplikasi Web”, tersedia : September 6 2010, https://bambangpdp.wordpress.com/2010/09/06/model-interaksi-aplikasi-web/

[SAM20] Samania, N., Nirsal, & Riska Yanu Fa’rifah., "Rancang Bangun Aplikasi E-Voting Pemilihan Ketua Umum Himpunan Mahasiswa Informatika (Hmti) Universitas Cokroaminoto Palopo Berbasis Website”, Jurnal Ilmiah d’Computare, Volume 10, Januari 2020

[SHA20] Shaf’at, A., Retnoningsih, D., & Khusnuliawati, H., “Sistem Pengelolaan Kendaraan Dinas Di Pemerintah Kota Salatiga”, Gaung Informatika, Volume 13, Nomor 2, Juli-Desember 2020

[SIM10] Simarmata, J., “Rekayasa Web” Andi, Yogyakarta, 2010

[SUL22] Sulistyorini, T., Sova, E., & Ramadhan, R., “Pemantauan Kasus Penyebaran Covid-19 Berbasis Website Menggunakan Framework React Js Dan Api”, Jurnal Jukim, Volume 1, No 4, Juli 2022

[SUM21] Sumiati, M., Abdillah, R., & Cahyo, A., “Pemodelan UML untuk Sistem Informasi Persewaan Alat Pesta”, Jurnal Fasilkom, Volume 11, No. 2, Agustus 2021

[SUR17] Surahman, S., Setiawan, E.B., “Aplikasi *Mobile Driver Online* Berbasis Android Untuk Perusahaan Rental Kendaraan”, ULTIMA InfoSys, Vol. 8, No. 1, Juni 2017

[WAH20] Wahid, A.A., “Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi”, Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK, 2020

LAMPIRAN

LAMPIRAN

A. Lampiran Berita Acara Wawancara

UNIVERITAS PASUNDAN

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INFORMATIKA

Jl. Dr. Setiabudi No.193, Gegerkalong, Kec. Sukasari, Kota Bandung, Jawa Barat 40153

DEKANAT 022-2019435, (HUNTING) 2019433, 2019407 Fax. 022 - 2019329

BERITA ACARA WAWANCARA

Pada hari ini,

Telah dilaksanakan wawancara yang berkaitan dengan penelitian yang akan dilakukan untuk memenuhi Tugas Akhir Strata Satu (SI).

Nama Narasumber	:	Eka Maulila lestari
Jabatan	:	Tim kreatif BCH
Tempat	:	Bandung Creative Hub
Alamat	:	Jl. Laswi No.7, Kacapiring, Kec. Batununggal, Kota Bandung, Jawa Barat 40271

Peneliti melakukan wawancara dengan narasumber yang berkaitan dengan penelitian di Bandung Creative Hub. Kemudian narasumber memberikan jawaban terkait dengan pertanyaan yang diajukan. Adapun daftar pertanyaan yang diajukan dan hasil wawancara terlampir.

Bandung, 13 Juni 2024

Peneliti	Narasumber,
Yudha Permana	(..Eka Maulila.....)
NRP. 203040101	

Dipindai dengan CamScanner

UNIVERITAS PASUNDAN

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Dr. Setiabudi No.193, Gegerkalong, Kec. Sukasari, Kota Bandung, Jawa Barat 40153
DEKANAT 022-2019435, (HUNTING) 2019433, 2019407 Fax. 022 - 2019329

Daftar Pertanyaan

1. Bisakah menjelaskan bagaimana proses pengajuan peminjaman ruangan saat ini?
2. Berapa lama waktu yang dibutuhkan untuk menyelesaikan satu pengajuan peminjaman ruangan dari awal hingga akhir?
3. Apa kendala terbesar yang ditemui dengan sistem pengajuan peminjaman ruangan saat ini?
4. Seberapa sering terjadi kesalahan penjadwalan atau kehilangan data formulir peminjaman?
5. Bagaimana cara menangani kesalahan dalam proses peminjaman?
6. Bagaimana cara menginformasikan ketersediaan ruangan kepada pengguna?
7. Bagaimana menyimpan dan mengelola data peminjaman ruangan pada saat ini?
8. Apakah ada kesulitan dalam mencari data historis atau membuat laporan penggunaan ruangan?
9. Apa yang bisa diperbaiki dalam sistem pengajuan peminjaman ruangan saat ini?
10. Apakah Bandung Creative Hub telah memanfaatkan kemudahan teknologi melalui sebuah website dalam pengelolaan data?
11. Jika beralih ke sistem digital, apa fitur yang paling penting untuk ada dalam sistem tersebut?

Dipindai dengan CamScanner

UNIVERITAS PASUNDAN
FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Dr. Setiabudi No.193, Gegerkalong, Kec. Sukasari, Kota Bandung, Jawa Barat 40153
DEKANAT 022-2019435, (HUNTING) 2019433, 2019407 Fax. 022 - 2019329

Jawaban

1. Untuk proses peminjaman ruangan 2 macam, dibawah 15 bisa chat ke hotline, kalo lebih dari 15 orang harus mengajukan surat. Prosesnya itu tanya dulu live schedule, chat dulu ke hotline tapi harus lihat dulu jadwal. Setiap akhir bulan. Milih tanggalnya langsung di chat ke hotline. Sebelum mengajukan dapat tanya hotline dahulu,
2. Sebelumnya memang karena dinas, jadi senin sampai jumat. 3 hari kerja harus selesai di kurasi, tapi kendalanya dikurator. Setelah pengajuan kita harus dipanggil kembali.
3. Kendalanya, beberapa masyarakat yang kurang teredukasi. Ada infografis diinstagram. Info grafis harus lebih detail
4. Hampir jarang kehilangan proposal, tidak sering tapi pernah ada, menjadikan jadwal bentrok. Karena adanya lonjakan pengguna, ada miss
5. Mengenai langkah cek langsung, agar tidak terjadi betrok.
6. Informasi mengenai ketersediaan. Live schedule cukup mudah didapatkan dan di kenali
7. Cara verifikasi, mengkurasi jadwal, setiap proposal yang masuk ke front office, dikurasi awalcocok dan masuk ke syarat langsung masuk ke penjadwalan. Jika cocok maka langsung bisa booking, jika tidak booking tapi langsung dating maka akan disarankan booking dahulu di jadwal kosong, kurasi awal di fo lal ke team curator
8. Setiap awal bulan diminta laporan, agak lama harus bagi waktu, untuk merekap surat masuk dan pembuatan laporan. Kendala di waktu mencari data yang sudah lama
9. Yang bisa diperbaiki, sebagai fasilitator. masih ada pengguna yang malas baca, sudah ada artikel dan hotline yang dapat menjelaskan.
10. Sepertinya sudah memanfatkan kemudahan teknologi. Kalau full aplkikasi tuh belum bisa, di satu sisi pakai teknologi, sisi lainya masih manual. Masih menjaga asset yang ada, masih menggunakan kertas. Lalu masyarakatnya juga belum teredukasi, tidak terlalu suka dengan Gedung pemerintahan
11. Fitur yang penting khusus studio harus bisa mengisi jadwal sendiri. Masih ada yang mengakali jadwal.

Dipindai dengan CamScanner

B. Lampiran Bukti Foto Dengan Narasumber