

**RANCANG BANGUN *WEBSITE* PEMINJAMAN RUANGAN  
DENGAN *FRAMEWORK* LARAVEL DI TELKOM  
UNIVERSITY PURWOKERTO**

**Tugas Akhir  
diajukan untuk memenuhi salah satu syarat memperoleh gelar sarjana  
dari Program Studi Teknik Informatika  
Fakultas Informatika  
Universitas Telkom**

**21102247  
Nabila Afifatul Fajri**



**Program Studi Sarjana Teknik Informatika  
Fakultas Informatika  
Direktorat Kampus Universitas Telkom  
Purwokerto  
2025**

**LEMBAR PENGESAHAN**

**RANCANG BANGUN *WEBSITE* PEMINJAMAN RUANGAN  
DENGAN *FRAMEWORK* LARAVEL DI TELKOM  
UNIVERSITY PURWOKERTO**

***DESIGN AND DEVELOPMENT OF A ROOM BOOKING  
WEBSITE USING THE LARAVEL FRAMEWORK AT TELKOM  
UNIVERSITY PURWOKERTO***

**21102247**

**Nabila Afifatul Fajri**

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagai syarat  
memperoleh gelar pada Program Studi S1 Teknik Informatika

Direktorat Kampus Purwokerto

Universitas Telkom

Purwokerto, <Tanggal Bulan Tahun>

Menyetujui

Pembimbing I,

Pembimbing II,

Yohani Setiya Rafika Nur,  
S.Kom., M.Kom.  
NIP: 23950029

Shintia Dwi Alike, S.Pd., M.Pd.  
NIP: 22920011

Penguji I,

Penguji II,

Sarah Astiti, S.Kom., M.MT  
NIP: 23890003

Dany Candra Febrianto, S.Kom.,  
M.Eng.  
NIP: 23920011

Ketua Program Studi  
Sarjana S1 Teknik Informatika,

Aditya Dwi Putro Wicaksono, S.Kom., M.Kom.  
NIP: 17930052

## LEMBAR ORISINALITAS

Dengan ini saya, Nabila Afifatul Fajri, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul “Rancang Bangun *Website* Peminjaman Ruangan dengan *Framework* Laravel di Telkom University Purwokerto” berserta dengan seluruh isinya merupakan hasil karya saya sendiri, dengan tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dengan masyarakat keilmuan, serta produk dari tugas akhir ini bukan merupakan hasil dari *Generative AI*. Saya siap menggugung risiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam Laporan Tugas Akhir, atau jika ada klaim dari pihak lain terhadap keaslian karya.

Purwokerto, 7 April 2025

Yang menyatakan



Nabila Afifatul Fajri

NIM 21102247

## ABSTRAK

Telkom University Purwokerto merupakan salah satu perguruan tinggi di Indonesia yang berfokus pada bidang teknologi informasi dan komunikasi. Namun, pengelolaan peminjaman ruangan masih sering dilakukan secara konvensional, yaitu melalui sistem pemesanan manual yang melibatkan proses administratif yang mengharuskan mahasiswa untuk menemui pihak logistik dan akademik terlebih dahulu. Meskipun proses tersebut relatif cepat, sistem manual memiliki keterbatasan seperti ketergantungan pada jam kerja dan belum adanya dokumentasi historis peminjaman yang terpusat. Selain itu, *user* juga tidak dapat secara langsung mengetahui ketersediaan ruangan tanpa menghubungi pihak terkait. Sebuah *website* diperlukan untuk mengelola peminjaman ruangan di kampus agar dapat terorganisir dengan baik. Sistem peminjaman ruangan yang dapat diakses melalui situs *web* melayani *user* yang ingin menggunakan ruangan untuk tujuan akademik dan non-akademik. Tujuannya adalah merancang *website* peminjaman ruangan yang berfungsi untuk mengelola peminjaman ruang di Telkom University Purwokerto. *Website* peminjaman ruangan ini telah dibuat dengan menggunakan metode *prototype* dan *framework Laravel*. Hasilnya adalah *website* peminjaman ruangan di Telkom University Purwokerto yang memiliki fitur untuk melihat daftar ruangan yang tersedia, proses peminjaman ruangan, serta riwayat dan daftar peminjam ruangan. Berdasarkan hasil pengujian *blackbox testing* didapatkan total presentase keberhasilan fungsi sebesar 96,45%.

**Kata kunci :** *framework Laravel, manajemen ruangan, peminjaman ruangan, prototype, website*

## **ABSTRACT**

*Telkom University Purwokerto is one of the universities in Indonesia that focuses on the field of information and communication technology. However, room booking management is still often done conventionally, namely through a manual booking system that involves an administrative process that requires students to meet the logistics and academic parties first. Although the process is relatively fast, the manual system has limitations such as dependence on working hours and the absence of centralized historical loan documentation. In addition, users also cannot directly find out the availability of the room without contacting the relevant parties. A website is needed to manage room booking on campus so that they can be well organized. The room booking system that can be accessed through the website serves users who want to use the room for academic and non-academic purposes. The goal is to design a room booking website that serves to manage room booking at Telkom University Purwokerto. This room booking website has been created using prototype method and Laravel framework. The result is a room lending website at Telkom Purwokerto University which has features to view a list of available rooms, the room booking process, as well as the history and list of room borrowers. Based on the results of blackbox testing, the total percentage of successful functions is 96.45%.*

***Keywords: framework Laravel, room booking, room management, prototype, website***

## KATA PENGANTAR

Ahamdulillah, puji dan syukur peneliti panjatkan ke hadirat Allah SWT berkat rahmat dan karunia-Nya, peneliti dapat menyelesaikan Laporan Penelitian Tugas Akhir yang berjudul “RANCANG BANGUN *WEBSITE* PEMINJAMAN RUANGAN DENGAN *FRAMEWORK* LARAVEL DI TELKOM UNIVERSITY PURWOKERTO”. Penyusunan tugas akhir ini dilakukan sebagai salah satu syarat untuk memperoleh gelar Sarjana pada program studi Teknik Informatika, Telkom University Purwokerto.

Peneliti menyadari bahwa Laporan Penelitian Tugas Akhir ini masih memiliki kelemahan dan jauh dari kata sempurna. Peneliti sangat terbuka terhadap segala bentuk kritik dan saran yang membangun demi perbaikan di masa mendatang. Pada proses penyusunan Tugas Akhir ini, peneliti mendapatkan banyak bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, peneliti ingin menyampaikan rasa hormat dan juga terima kasih kepada:

1. Dr. Tenia Wahyuningrum, S.Kom., M.T selaku Direktur Telkom University Purwokerto.
2. Aditya Dwi Putro Wicaksono, S.Kom., M.Kom selaku Ketua Program Studi S1 Teknik Informatika.
3. Yohani Setiya Rafika Nur, S.Kom., M.Kom dan Shintia Dwi Alike, S.Pd., M.Pd selaku Dosen Pembimbing yang telah memberikan bimbingan, pengarahan, saran, kritik, dan kontribusi yang membangun pada saat penyusunan Laporan Penelitian Tugas Akhir.
4. Kepada Pihak Akademik Telkom University Purwokerto yang telah memberikan izin untuk melakukan penelitian dan memberikan data yang diperlukan peneliti untuk menyelesaikan penelitian ini.
5. Kepada teman-teman di Telkom University Purwokerto yang telah bersedia meluangkan waktunya untuk menjadi responden pengujian penelitian ini.
6. Kepada keluarga peneliti yang selalu memberikan dukungan, doa, dan saran untuk menyelesaikan penelitian ini.

7. Kepada teman-teman Topaz Genk yaitu, Rizqina Nabila Fitriana, Amalia Suciati, Agustina Mulyaningsih, Feby Setyany Lestari, Annisa Febriana, dan Puspita Kartika Sari yang telah berbagi motivasi, pengalaman, suka dan duka selama di perkuliahan maupun di luar perkuliahan.
8. Kepada teman-teman online yang walaupun terpisah jarak dan belum pernah bertemu secara langsung, namun tetap setia menemani, memberikan semangat, dan menjadi tempat berbagi cerita serta keluh kesah selama masa-masa sulit menyusun penelitian ini. Kehadiran kalian sangat berarti bagi peneliti.
9. Serta semua pihak yang tidak dapat peneliti sebutkan satu per satu, namun telah berjasa dalam penyusunan Laporan Penelitian Tugas Akhir ini.

Akhir kata, peneliti berharap Laporan Penelitian Tugas Akhir ini dapat memberikan manfaat bagi pembaca, khususnya dalam pengembangan sistem informasi berbasis web, serta menjadi referensi bagi penelitian selanjutnya. Semoga Allah SWT senantiasa memberikan kemudahan dalam setiap langkah yang dilalui.

Purwokerto, 7 April 2025



Nabila Afifatul Fajri

## DAFTAR ISI

LEMBAR PENGESAHAN.....	ii
LEMBAR ORISINALITAS .....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xi
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang Masalah .....	1
1.2. Rumusan Masalah .....	3
1.3. Tujuan dan Manfaat.....	3
1.4. Batasan Masalah.....	4
1.5. Metode Penelitian.....	4
BAB 2 TINJAUAN PUSTAKA.....	5
2.1. Tinjauan Pustaka .....	5
2.2. Landasan Teori.....	24
2.2.1. Rancang bangun .....	24
2.2.2. <i>Website</i> .....	24
2.2.3. Peminjaman Ruangan .....	24
2.2.4. Metode <i>Prototype</i> .....	26
2.2.5. <i>Framework</i> Laravel .....	28
2.2.6. <i>Blackbox Testing</i> .....	28



2.2.7. SQL.....	29
2.2.8. UML .....	29
2.2.9. <i>Entity Relationship Diagram</i> .....	34
2.2.10. <i>Software Requirement &amp; User Requirement</i> .....	34
BAB 3 PERANCANGAN SISTEM.....	35
3.1. Desain Perancangan Sistem .....	35
3.2. Diagram Alir .....	35
3.2.1. Pengumpulan data .....	36
3.2.2. <i>Initial Requirement</i> .....	37
3.2.3. Desain .....	40
3.2.4. <i>Prototyping</i> .....	40
3.2.5. <i>Evaluation</i> .....	41
3.2.6. <i>Review &amp; Updating</i> .....	41
3.2.7. <i>Development</i> .....	41
3.2.8. <i>Test</i> .....	41
3.2.9. <i>Maintenance</i> .....	41
3.2.10. Kesimpulan dan Saran .....	42
3.3. Desain sistem .....	42
3.3.1. <i>Entity Relationship Diagram (ERD)</i> .....	42
3.3.2. UML .....	43
3.4. Desain Perangkat Lunak (Iterasi 1) .....	52
3.4.1. Desain <i>Wireframe</i> .....	52
3.4.2. Desain <i>Mockup</i> .....	57
3.5. Evaluasi (Iterasi 1).....	62
3.6. <i>Review &amp; Updating</i> .....	64

3.7. Desain Perangkat Lunak (Iterasi 2) .....	64
3.7.1. Desain <i>Wireframe</i> .....	64
3.7.2. Desain <i>Mockup</i> .....	66
3.8. Evaluasi (Iterasi 2).....	69
<b>BAB 4 HASIL PERCOBAAN DAN ANALISIS .....</b>	<b>72</b>
4.1. Hasil Akhir Sistem .....	72
4.1.1. Perancangan .....	72
4.1.2. <i>Development</i> .....	75
4.2. Analisis .....	86
<b>BAB 5 KESIMPULAN DAN SARAN .....</b>	<b>92</b>
5.1. Kesimpulan .....	92
5.2. Saran .....	92
<b>DAFTAR PUSTAKA .....</b>	<b>94</b>
<b>LAMPIRAN .....</b>	<b>98</b>

## DAFTAR GAMBAR

Gambar 2.1 <i>Flowchart</i> sistem peminjaman ruangan .....	25
Gambar 2.2 Tahapan Metode <i>Prototype</i> [19].....	27
Gambar 3.1 Diagram alir penelitian dengan metode <i>prototype</i> .....	36
Gambar 3.2 <i>Entity Relationship Diagram</i> peminjaman ruangan .....	42
Gambar 3.3 <i>Use Case Diagram</i> peminjaman ruangan .....	44
Gambar 3.4 <i>Class Diagram</i> peminjaman ruangan .....	45
Gambar 3.5 <i>Activity Diagram Login User</i> .....	47
Gambar 3.6 <i>Acitivity Diagram Login Admin</i> .....	48
Gambar 3.7 <i>Activity Diagram</i> Peminjaman Ruangan.....	49
Gambar 3.8 <i>Activity Diagram</i> Manajemen Peminjaman .....	50
Gambar 3.9 <i>Activity Diagram</i> Daftar Peminjaman <i>User</i> .....	51
Gambar 3.10 <i>Activity Diagram</i> Daftar Peminjaman <i>Admin</i> .....	52
Gambar 3.11 <i>Wireframe</i> halaman <i>login</i> (iterasi 1) .....	53
Gambar 3.12 <i>Wireframe</i> halaman <i>register</i> (iterasi 1) .....	53
Gambar 3.13 <i>Wireframe dashboard user</i> (iterasi 1) .....	54
Gambar 3.14 <i>Wireframe</i> daftar ruangan <i>user</i> (iterasi 1) .....	54
Gambar 3.15 <i>Wireframe</i> formulir peminjaman ruangan (iterasi 1) .....	55
Gambar 3.16 <i>Wireframe</i> daftar peminjaman <i>user</i> (iterasi 1).....	55
Gambar 3.17 <i>Wireframe</i> daftar ruangan <i>admin</i> (iterasi 1) .....	56
Gambar 3.18 <i>Wireframe</i> data peminjaman <i>admin</i> (iterasi 1) .....	56
Gambar 3.19 <i>Wireframe</i> daftar peminjaman <i>admin</i> (iterasi 1).....	57
Gambar 3.20 <i>Mockup</i> halaman <i>login</i> (iterasi 1) .....	57
Gambar 3.21 <i>Mockup</i> halaman <i>register</i> (iterasi 1) .....	58
Gambar 3.22 <i>Mockup dashboard user</i> (iterasi 1) .....	58

Gambar 3.23 <i>Mockup</i> daftar ruangan <i>user</i> (iterasi 1) .....	59
Gambar 3.24 <i>Mockup</i> formulir peminjaman ruangan (iterasi 1) .....	60
Gambar 3.25 <i>Mockup</i> daftar peminjaman <i>user</i> (iterasi 1) .....	60
Gambar 3.26 <i>Mockup</i> daftar ruangan <i>admin</i> (iterasi 1) .....	61
Gambar 3.27 <i>Mockup</i> data peminjaman <i>admin</i> (iterasi 1) .....	61
Gambar 3.28 <i>Mockup</i> daftar peminjaman <i>admin</i> (iterasi 1) .....	62
Gambar 4.1 Struktur <i>Models</i> pada sistem .....	72
Gambar 4.2 Struktur folder <i>Views</i> dalam sistem .....	73
Gambar 4.3 Struktur <i>Controller</i> pada sistem .....	74
Gambar 4.4 Halaman <i>login</i> .....	76
Gambar 4.5 Halaman registrasi .....	77
Gambar 4.6 Halaman <i>dashboard user</i> .....	78
Gambar 4.7 Halaman daftar ruangan <i>user</i> .....	79
Gambar 4.8 Formulir peminjaman ruangan .....	79
Gambar 4.9 Halaman daftar peminjaman <i>user</i> .....	81
Gambar 4.10 Halaman daftar peminjaman <i>user</i> setelah dikonfirmasi <i>admin</i> .....	81
Gambar 4.11 Halaman daftar <i>admin</i> .....	82
Gambar 4.12 Halaman daftar mahasiswa .....	82
Gambar 4.13 Halaman daftar ruangan <i>admin</i> .....	83
Gambar 4.14 Halaman daftar peminjaman sementara <i>admin</i> .....	83
Gambar 4.15 Halaman riwayat atau daftar peminjaman <i>admin</i> .....	84
Gambar 4.16 Notifikasi <i>user</i> .....	85
Gambar 4.17 Notifikasi <i>admin</i> .....	85

## DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu .....	9
Tabel 2.2 Simbol <i>Use Case Diagram</i> [27] .....	30
Tabel 2.3 Simbol <i>Class Diagram</i> [27] .....	32
Tabel 2.4 Simbol <i>Activity Diagram</i> [27] .....	33
Tabel 3.1 Daftar ruangan yang dapat dipinjam .....	36
Tabel 3.2 Kebutuhan Fungsional .....	37
Tabel 3.3 Kebutuhan non-fungsional .....	38
Tabel 3.4 Kebutuhan pengguna ( <i>user</i> ) .....	39
Tabel 3.5 Kebutuhan pengguna ( <i>admin</i> ) .....	40
Tabel 3.6 Umpan balik <i>user</i> dan <i>admin</i> pada iterasi 1 .....	63
Tabel 3.7 Perubahan <i>wireframe</i> pada iterasi 2 .....	65
Tabel 3.8 Penambahan <i>wireframe</i> pada iterasi 2 .....	66
Tabel 3.9 Perubahan desain <i>wireframe</i> dan <i>mockup</i> .....	67
Tabel 3.10 Umpan balik <i>user</i> dan <i>admin</i> pada iterasi 2 .....	70
Tabel 4.1 <i>Testcase user</i> .....	86
Tabel 4.2 <i>Testcase Login Admin</i> .....	88
Tabel 4.3 <i>Testcase Edit Admin</i> .....	88
Tabel 4.4 <i>Testcase</i> Daftar Ruangan <i>Admin</i> .....	89
Tabel 4.5 <i>Testcase</i> Daftar Peminjaman <i>Admin</i> .....	89
Tabel 4.6 Rekapitulasi hasil pengujian <i>blackbox</i> .....	90

# **BAB 1**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Telkom University Purwokerto merupakan salah satu perguruan tinggi di Indonesia yang berfokus pada bidang teknologi informasi dan komunikasi. Kampus ini dilengkapi dengan infrastruktur dan fasilitas pendidikan modern, seperti laboratorium komputer, ruang kuliah berbasis teknologi, perpustakaan digital, dan akses internet yang cepat untuk mendukung pembelajaran berbasis teknologi. Seringkali perguruan tinggi dan kampus memiliki banyak ruang yang tersebar di berbagai gedung dan lokasi, salah satunya pada Telkom University Purwokerto. Kegiatan akademik dan non akademik berlangsung di ruangan. Termasuk perkuliahan, seminar, diskusi kelompok, pertemuan organisasi mahasiswa, dan acara kampus lainnya.

Pengelolaan peminjaman ruangan masih sering dilakukan secara konvensional, yaitu melalui sistem pemesanan manual yang melibatkan proses administratif dan mengharuskan mahasiswa untuk menemui pihak logistik dan akademik terlebih dahulu. Berdasarkan hasil wawancara yang dilakukan oleh kepala pelayanan akademik Telkom University Purwokerto, proses ini dapat memakan waktu 10 hingga 15 menit atau bahkan lebih tergantung pada tingkat kecepatan respon dari pihak yang terlibat dalam proses peminjaman ruangan dan juga mengenai ketersediaan ruangnya. Meskipun proses tersebut relatif cepat, sistem manual memiliki keterbatasan seperti ketergantungan pada jam kerja dan belum adanya dokumentasi historis peminjaman yang terpusat. Selain itu, *user* juga tidak dapat secara langsung mengetahui ketersediaan ruangan tanpa menghubungi pihak terkait. Pertama, proses pemesanan manual cenderung memakan waktu dan membingungkan, terutama jika ada tabrakan jadwal atau perubahan mendadak dalam rencana kegiatan. Kedua, jika tidak ada informasi yang jelas tentang ketersediaan ruangan yang memungkinkan akan terjadi konflik jadwal antara pemesanan dan penggunaan ruangan akibat kurangnya informasi yang jelas mengenai ketersediaan ruangan. Ketiga, tanpa adanya sistem yang terpusat pemantauan peminjaman dan penggunaan ruangan akan menjadi susah dilakukan

karena informasi mengenai ketersediaan, pemesanan, dan penggunaan ruangan tidak terorganisir dengan baik, sehingga menyebabkan kebingungan saat akan melakukan peminjaman ruangan.

Telkom University Purwokerto sudah dikenal dengan kemajuan teknologinya terutama sebagai kampus yang lebih berfokus dalam bidang teknologi, maka diperlukan pula sistem yang canggih untuk melakukan peminjaman ruangan. Telkom University Purwokerto terkadang memiliki banyak acara yang menyebabkan dibutuhkan ruangan untuk melakukan kegiatan tersebut sehingga sistem peminjaman ruangan ini diperlukan untuk menghemat waktu. *Website* peminjaman ruangan ini telah dibuat untuk semua program studi yang ada di Telkom University Purwokerto. Mahasiswa dari segala macam program studi serta staf pengelola akademik di Telkom University Purwokerto dapat mengakses *website* peminjaman ruangan ini.

Berdasarkan hasil wawancara singkat yang sudah dilakukan dengan pihak pengelola akademik bahwa di Telkom University Purwokerto memiliki ruangan sebanyak 44 ruangan pada 5 gedung (Gedung DC, Gedung DSP, Gedung IOT, Gedung Rektorat, dan Gedung TT) yang belum memiliki sistem peminjaman ruangan yang efisien, yaitu: DC-201, DC-202, DC-203, DC-301, DC-302, DSP-301, DSP-302, DSP-303, DSP-304, DSP-305, DSP-306, DSP-307, DSP-308, DSP-401, DSP-402, DSP-403, DSP-404, DSP-405, DSP-501, DSP-508, IOT-101, IOT-102, IOT-103, IOT-104, IOT-105, IOT-201, IOT-202, REK-201, REK-202, REK-203, REK-204, REK-205, REK-207, REK-301, REK-302, REK-303, REK-304, REK-305, REK-306, REK-307, TT-102, TT-103, TT-104, TT-105.

Hal ini dibuktikan bahwa masih banyak mahasiswa yang harus menggunakan cara manual dengan melakukan proses administratif yang kompleks yaitu meminjam yang ingin meminjam ruangan harus melewati beberapa tahapan sebelum meminjam ruangan, seperti menemui pihak logistik untuk meminta jadwal ruangan yang kosong, lalu ke pihak akademik dan selanjutnya memastikan apakah ruangan yang akan dipinjam benar kosong atau tidak. Sebuah *website* diperlukan untuk mengelola peminjaman ruangan di kampus agar dapat terorganisir dengan baik. Sistem peminjaman ruangan yang dapat diakses melalui *website* ini melayani

*user* yang ingin menggunakan ruangan untuk keperluan akademik maupun non akademik [1]. *Website* adalah media informasi digital yang dapat diakses dengan mudah dan dibangun menggunakan sintaks atau *script* seperti *HyperText Markup Language (HTML)* dan *Cascading Style Sheets (CSS)* [2]. *Website* peminjaman ruangan ini mempermudah proses peminjaman ruangan dengan memungkinkan peminjaman ruangan, menampilkan data daftar ruangan, mengolah data ruangan, dan daftar peminjaman ruangan [3]. Pengembangan *website* ini akan menggunakan bahasa pemrograman *PHP* dengan *framework* *Laravel*.

*Framework* *Laravel* yang digunakan untuk mengembangkan *website* peminjaman ruangan ini memiliki ciri khas yaitu penekanan pada fleksibilitas desain. Hal tersebut mempermudah dalam proses pembuatan *website* dalam waktu yang singkat dan kode yang rapi [4]. Metode *prototype* digunakan pada pengembangan sistem ini untuk memahami kebutuhan *user*, validasi konsep, dan mendapatkan umpan balik *user*. Proses pengembangan sistem *iterative* yang mengubah kebutuhan menjadi sistem yang terus diperbaiki melalui kerja sama *user* dan analis [5]. Penggunaan metode *prototype* memungkinkan penyesuaian kebutuhan *user*, sekaligus menghemat waktu dan biaya. Hasil uji menggunakan metode *blackbox testing*, yaitu suatu pengujian yang lebih berfokus pada spesifikasi fungsi-fungsi yang dikembangkan. *Blackbox testing* menggunakan perspektif *user*, sehingga fokusnya hanya pada pengujian fungsional sistem [6].

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang di atas, didapatkan rumusan masalah yaitu belum tersedianya sistem terpusat yang dapat membantu *user* dalam mengetahui ketersediaan ruangan, melakukan peminjaman tanpa harus datang langsung, serta mendokumentasikan riwayat peminjaman secara sistematis.

## **1.3. Tujuan dan Manfaat**

Tujuannya adalah untuk merancang sebuah *website* peminjaman ruangan yang berfungsi untuk mengelola peminjaman ruangan di Telkom University Purwokerto. Berikut ini merupakan beberapa manfaat yang didapatkan:

- a. Mahasiswa dan pihak akademik dapat menghemat lebih banyak waktu



dengan melakukan peminjaman ruangan melalui *website*.

- b. *Website* memberikan informasi yang jelas mengenai daftar ruangan yang dapat dipinjam untuk membantu *user* merencanakan kegiatan dengan baik.

#### **1.4. Batasan Masalah**

Sebagai upaya untuk memastikan pembahasan lebih terfokus sesuai dengan rumusan masalah di atas, berikut adalah lingkup batasan masalah yang akan dijelaskan dalam pembahasan ini:

1. Fokus pada penggunaan *website* untuk manajemen peminjaman ruangan di lingkungan Telkom University Purwokerto.
2. Pengguna *website* peminjaman ruangan dapat dilakukan oleh seluruh mahasiswa dan staf pengelola ruangan di Telkom University Purwokerto.
3. Eksekusi perancangan *website* dibatasi dengan menggunakan bahasa pemrograman *PHP* yang dikombinasikan dengan *framework* Laravel dan metode *prototype*.

#### **1.5. Metode Penelitian**

Metode penelitian ini menggunakan metode kualitatif melalui observasi dan wawancara kepada pihak akademik. Data diperoleh melalui wawancara mengenai sistem peminjaman ruangan yang ada di Telkom University Purwokerto. Data ruangan juga diperoleh untuk keperluan pendataan dalam sistem mengenai ruangan mana saja yang dapat dipinjam. Metode pengembangan yang digunakan adalah *prototype* sebagai pendekatan utamanya. Pendekatan ini memberikan gambaran umum yang komprehensif tentang konsep, memfasilitasi eksperimen dengan desain, mengidentifikasi masalah, dan mencari solusi untuk masalah tersebut. Metode *prototype* yang digunakan bertujuan untuk mendapatkan representasi dari pemodelan situs *web* yang telah dikembangkan. Tahapan metode *prototype* menggunakan model *prototyping Software Development Life Cycle (SDLC)* meliputi pengumpulan kebutuhan, membangun *prototype*, mengevaluasi *prototype*, meninjau dan memperbaiki, pengembangan sistem, pengujian sistem, evaluasi sistem, dan penggunaan sistem.

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Penting memperkuat dan mengkontekstualisasikan kontribusi yang diusulkan, serta memanfaatkan acuan studi sebelumnya saat merancang pembahasan ini. Sebelumnya sudah dijelaskan mengenai *website* peminjaman ruangan pada suatu instansi. Sebagai hasilnya, ini bertujuan untuk meningkatkan kerangka pengetahuan yang ada dan memperjelas kontribusi uniknya dibandingkan dengan studi sebelumnya dengan menawarkan gambaran rinci tentang temuan dari sebelumnya.

Hasil [7] peminjaman ruangan untuk area Dinas di Kota Salatiga dengan menggunakan *framework* Laravel dan metode *Waterfall* menghasilkan aplikasi berbasis *web* yang mampu menangani proses peminjaman ruangan secara digital, mulai dari pengelolaan data ruangan, pengajuan peminjaman, hingga pencatatan riwayat. Tahapan yang dilakukan meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian menggunakan *blackbox testing*, hingga pemeliharaan. Pengujian menunjukkan seluruh fitur berjalan sesuai fungsinya, sehingga sistem ini dinilai mampu meningkatkan efisiensi dan akurasi dalam pengelolaan ruangan secara terstruktur dan terdokumentasi dengan baik.

Tujuan [8] untuk merancang dan membangun sistem informasi manajemen ruangan pada Program Studi Teknik Informatika dengan menggunakan *framework* Laravel dan metode pengembangan sistem *Waterfall*. Proses pengembangan dilakukan melalui tahapan analisis kebutuhan, perancangan dengan diagram UML, implementasi sistem berbasis *web*, dan pengujian menggunakan *blackbox testing*. Hasilnya adalah sebuah aplikasi yang mempermudah proses peminjaman dan pengelolaan ruangan oleh dosen maupun mahasiswa, serta membantu staf dalam mengatur jadwal dan administrasi ruangan. Sistem yang dibangun terbukti dapat meningkatkan efisiensi, keakuratan informasi, dan keteraturan dalam pengelolaan ruangan secara digital.

Metode pengembangan sistem yang digunakan pada *website* untuk pemesanan laboratorium dengan *framework* Laravel dan *respons native* untuk basis *mobile* [9] adalah *Waterfall*, yang mencakup tahapan identifikasi masalah, studi literatur, wawancara dengan teknisi laboratorium, analisis sistem, perancangan, implementasi, dan pengujian. Hasilnya adalah sebuah sistem informasi berbasis *web* dan aplikasi Android yang memfasilitasi proses peminjaman laboratorium oleh mahasiswa, serta memudahkan *admin* dalam mengelola data laboratorium, pengguna, dan persetujuan peminjaman.

Tujuan [10] adalah untuk mengembangkan Sistem Informasi Manajemen Ruang Rapat dengan menggunakan *V-Model* sebagai metode pengembangan. Model ini dipilih karena mampu mengintegrasikan tahapan pengembangan sistem secara berurutan dengan proses pengujian yang terstruktur. Tahapan yang dilakukan meliputi identifikasi kebutuhan, perancangan sistem, implementasi, dan pengujian. Hasilnya adalah sistem berbasis *web* yang memudahkan peminjaman ruang rapat secara *real-time*, mengurangi risiko bentrok jadwal. Pengujian sistem menggunakan *Basis Path Testing* menunjukkan hasil yang memuaskan dengan semua kasus uji berhasil dijalankan, menandakan sistem memenuhi kebutuhan fungsional yang diharapkan.

Implementasi sistem registrasi ruangan berbasis web di Amikom *Resource Centre* [1] dengan fokus khusus pada faktor-faktor yang meningkatkan kegunaan, meminimalkan pengeluaran waktu, dan memberikan hasil yang lebih baik bagi mahasiswa dan staf, dibuat menggunakan *Model View Controller (MVC)* sebagai metodenya. Metode MVC efektif dalam menghasilkan sistem yang terstruktur, efisien, dan mudah untuk dikembangkan serta dipelihara. Penerapan metode ini memungkinkan pengelolaan data peminjaman ruangan yang lebih baik, kemudahan dalam pemeliharaan sistem, dan peningkatan kecepatan pengembangan yang berdampak langsung pada efisiensi operasional sistem peminjaman ruangan di *Resource Centre* Universitas Amikom Yogyakarta.

Aplikasi yang dirancang pada Direktorat Jenderal Hak Asasi Manusia [3] ini dibangun menggunakan bahasa pemrograman *PHP*, *JavaScript*, *HTML*, *CSS*, dan *MySQL* dengan metode pengembangan *prototype* yang memungkinkan

pembuatan *prototype* sistem secara cepat dan iteratif. Aplikasi ini meningkatkan efisiensi proses peminjaman ruangan, mengurangi ketergantungan pada surat menyurat, dan mempermudah pegawai dalam mengelola jadwal ruangan.

Pengembangan *Website Administrator Peminjaman Ruangan Laboratorium* di Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta [11] yang menggunakan *framework Vue.js* dan *database Firebase* mempermudah administrasi dalam mengelola peminjaman ruangan laboratorium, dengan fitur *login* admin, penerimaan peminjaman, pendaftaran pengguna, dan penyimpanan riwayat peminjaman. Metode pengembangan yang digunakan adalah *Waterfall* yang mencakup tahapan perencanaan, analisis, desain, implementasi, dan pengujian. Hasilnya adalah sistem yang efisien dan terstruktur dalam mengelola peminjaman ruangan laboratorium.

Sistem aplikasi berbasis *web* yang dibuat [12] bertujuan untuk mendukung manajemen fakultas perguruan tinggi dalam mengawasi sistem peminjaman ruangan. Sistem ini dibangun menggunakan metode *Waterfall* yang diterapkan melalui tahapan perencanaan, analisis kebutuhan, desain sistem, implementasi, serta pengujian. Sistem *database* secara efisien mengatur semua data transaksi, memungkinkan staf perlengkapan untuk mengelola informasi, membuat, dan menyampaikan laporan manajemen secara efektif.

Pengembangan aplikasi peminjaman ruangan dan gedung berbasis *web* di Universitas Mercu Buana Kampus D Jatisampurna [13] dibangun menggunakan *PHP* dan *MySQL*, dengan algoritma *Bubble Sort* untuk mengurutkan data peminjaman. Pengembangan aplikasi menggunakan metode *SDLC (System Development Life Cycle)* yang mencakup tahapan perencanaan, analisis, desain, implementasi, dan pemeliharaan. Hasilnya adalah sistem yang mempermudah pengelolaan peminjaman ruangan dan gedung, meningkatkan efisiensi, dan mengurangi kesalahan administrasi.

Pengembangan Sistem Informasi Peminjaman Ruangan berbasis *web* menggunakan *PHP* dan *MySQL* [14] dengan metode pengembangan *Waterfall*. Sistem ini melalui lima tahapan yaitu komunikasi, perencanaan, pemodelan, konstruksi, dan penyebaran. Pengujian kelayakan dilakukan melalui *Alpha Testing*

oleh ahli media dan *Beta Testing* oleh pengguna, dengan hasil menunjukkan skor tinggi yang menandakan sistem sangat layak digunakan. Aplikasi ini meningkatkan efisiensi dan transparansi dalam pengelolaan peminjaman ruangan.

Pengembangan Sistem Manajemen Peminjaman Ruangan berbasis *web* menggunakan metode *Rapid Application Development (RAD)* [15] dibangun dengan framework *PHP CodeIgniter* dan *database MySQL*, serta menerapkan pola desain MVC. Proses pengembangan melibatkan pembuatan *prototype* dan pengujian berkelanjutan berdasarkan umpan balik pengguna. Fitur utama mencakup peminjaman ruangan, pengelolaan jadwal, dan laporan statistik peminjaman. Hasil pengujian menunjukkan bahwa sistem ini efektif dalam meningkatkan efisiensi dan transparansi pengelolaan peminjaman ruangan.

Pembahasan ini yang berjudul Rancang Bangun *Website* Peminjaman Ruangan dengan *Framework* Laravel di Kampus Telkom University Purwokerto memiliki perbedaan dengan penelitian terdahulu. *Website* peminjaman ruangan ini menggunakan metode *prototype* dengan *framework* Laravel serta memiliki beberapa fitur yang berbeda. Penelitian terdahulu memiliki fitur yang mencakup peminjaman ruangan, melihat riwayat peminjaman ruangan, dan laporan peminjaman sedangkan untuk pembahasan kali ini, menggunakan fitur mengelola data pengguna dan fitur riwayat peminjaman ruangan beserta tambahan notifikasi peminjam ruangan. Berikut ini, merupakan tabel penelitian terdahulu seperti yang disebutkan pada tabel 2.1:

Tabel 2.1 Penelitian terdahulu

No.	Nama Penulis, Tahun	Judul	Masalah Penelitian	Metode	Hasil Penelitian
1.	1. Denny Askar Kurniawan, 2. Sri Dianing Asri Tahun 2020 [13]	Aplikasi Peminjaman Ruangan Dan Gedung Pada Universitas Mercu Buana Kampus D Jatisampurna Berbasis <i>Web</i> .	Universitas Mercu Buana masih melakukan peminjaman gedung atau ruangan dengan cara yang manual sehingga menyebabkan tidak efisien waktu.	<i>System Development Life Cycle (SDLC)</i> .	Aplikasi dibangun menggunakan <i>PHP</i> dan <i>MySQL</i> dengan algoritma <i>Bubble Sort</i> untuk mengurutkan data peminjaman. Pengembangan aplikasi menggunakan metode <i>SDLC (System Development Life Cycle)</i> yang mencakup tahapan perencanaan, analisis, desain, implementasi, dan pemeliharaan. Hasilnya adalah sistem yang mempermudah

					pengelolaan peminjaman ruangan dan gedung, meningkatkan efisiensi, dan mengurangi kesalahan administrasi.
2.	1. Admaja Dwi Herlambang, 2. Aditya Rachmadi 3. Azri Putri Rahmatika, 4. Dinar Indah Dwi Utami, 5. Safira Widya Hapsari Tahun 2020 [10].	<i>V- Model</i> untuk Pengembangan Sistem Informasi Manajemen Ruang Rapat.	PT. PJB Unit Bisnis Jasa Operation & Maintenance PLTU Paiton Baru Unit 9 masih belum memiliki sistem informasi untuk mengelola peminjaman ruang rapat. Akibatnya, perusahaan masih melakukan proses peminjaman ruang rapat secara manual dan tidak memiliki riwayat peminjaman. Divisi	<i>V-Model</i>	Pengembangan Sistem Informasi Manajemen Ruang Rapat dengan menerapkan model pengembangan perangkat lunak <i>V-Model</i> . Model ini dipilih karena mampu mengintegrasikan tahapan pengembangan sistem secara berurutan dengan proses pengujian yang terstruktur. Tahapan yang dilakukan meliputi identifikasi kebutuhan,

			<p>sekretaris bidang umum tidak tahu apa yang diperlukan untuk mengadakan pertemuan.</p>		<p>perancangan arsitektur sistem, desain komponen, implementasi kode, serta pengujian sistem.</p> <p>Hasilnya adalah sebuah sistem informasi berbasis <i>web</i> yang memungkinkan pengguna untuk mengakses informasi peminjaman ruang rapat secara <i>real-time</i>, sehingga mengurangi risiko kesalahan informasi dan bentrok jadwal.</p>
--	--	--	--	--	--



3.	1. Upi Resti Wahyuni, 2. Fauziah, 3. Eri Mardiani Tahun 2020 [15].	<i>Web-Based Room Loan Management Services Using Web Based RAD Method</i>	Mengizinkan mahasiswa untuk meminjam ruang adalah salah satu layanan yang ditawarkan oleh manajemen Universitas Nasional. Termasuk memberikan sarana dan prasarana seperti peminjaman ruangan atau perizinan ruang untuk mendukung kegiatan universitas. Penggunaan sangat penting karena sistem berbasis <i>web</i> membantu	<i>Rapid Application Development (RAD)</i>	Pengembangan Sistem Manajemen Peminjaman Ruangan berbasis <i>web</i> menggunakan metode <i>Rapid Application Development (RAD)</i> . RAD dipilih karena memungkinkan pengembangan aplikasi secara cepat dengan iterasi dan umpan balik pengguna yang intensif. Aplikasi ini dibangun menggunakan framework <i>PHP CodeIgniter</i> dan database <i>MySQL</i> . Fitur utama aplikasi
----	---	---	---	--	--

			administrator dan pengguna mengelola ruang.		mencakup peminjaman ruangan, pengelolaan jadwal, dan laporan statistik peminjaman. Hasil pengujian menunjukkan bahwa sistem ini efektif dalam meningkatkan efisiensi dan transparansi pengelolaan peminjaman ruangan di lingkungan kampus.
4.	1. Nabiih Al Daffa Dhanardana, 2. Tri Rahayu Tahun 2020 [3].	Aplikasi Peminjaman Ruangan Berbasis <i>Web</i> Pada Direktorat Jenderal Hak Asasi Manusia.	Selama ini, proses peminjaman ruangan di Direktorat Jenderal Hak Asasi Manusia masih dilakukan secara manual dengan mengandalkan	<i>Prototype</i>	Aplikasi ini dibangun menggunakan bahasa pemrograman <i>PHP</i> , <i>JavaScript</i> , <i>HTML</i> , dan <i>CSS</i> , serta <i>database MySQL</i> . Metode pengembangan yang

			<p>korespondensi antara bagian umum dan sub bagian.</p>		<p>digunakan adalah <i>Prototype</i> yang memungkinkan pembuatan <i>prototype</i> sistem secara cepat dan iteratif. Adanya aplikasi ini, proses peminjaman ruangan menjadi lebih efisien, mengurangi ketergantungan pada surat menyurat, dan mempermudah pegawai dalam melakukan kegiatan peminjaman ruangan.</p>
--	--	--	---	--	---

5.	1. Jethro Reyhan Naldo, 2. Pascal Aldwin Hernando, 3. Ing. Artambo Benjamin Pangaribuan Tahun 2021 [11].	Pembuatan <i>Website</i> Admininstrator Peminjaman Ruangan Laboratorium Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta.	Aplikasi yang dibutuhkan oleh pihak administrasi dapat diintegrasikan ke dalam proses penerimaan atau penolakan permintaan peminjaman ruangan untuk mempercepat proses seleksi peminjaman tersebut.	<i>Waterfall</i>	Aplikasi ini dibangun menggunakan <i>framework Vue.js</i> untuk <i>front-end</i> dan <i>Firebase</i> sebagai <i>database</i> <i>backend</i> . Metode pengembangan yang digunakan adalah <i>Waterfall</i> yang mencakup tahapan perencanaan, analisis, desain, implementasi, dan pengujian. Hasilnya adalah aplikasi berbasis <i>web</i> yang memudahkan pihak administrasi dalam mengelola peminjaman ruangan laboratorium secara
----	---	---	---	------------------	---

					efisien dan terstruktur
6.	1. Afdal, 2. Novri, 3. Dwiny Meidelfi, 4. Tri Lestari, Tahun 2021 [9].	Aplikasi Peminjaman Laboratorium Pada Fakultas Teknologi Pertanian Universitas Andalas	Mengingat pentingnya fungsi laboratorium, maka manajemen laboratorium yang baik diperlukan. Memungkinkan kampus untuk lebih mudah mendapatkan data aktual tentang penggunaan laboratorium dan aktifitas mahasiswa.	<i>Waterfall</i>	Sistem ini menggunakan <i>framework</i> Laravel untuk pengembangan web dan <i>React Native</i> untuk aplikasi Android. Metode pengembangan sistem yang digunakan adalah <i>Waterfall</i> . Hasilnya adalah sebuah sistem informasi berbasis <i>web</i> dan aplikasi Android yang memfasilitasi proses

					peminjaman laboratorium oleh mahasiswa, serta memudahkan admin dalam mengelola data laboratorium, pengguna, dan persetujuan peminjaman.
7.	1. Fuad Fadlila Surenggana, 2. Fitri Bimantoro, 3. Rival Biasrori Tahun 2021 [8]	Sistem Informasi Manajemen Ruangan Program Studi Teknik Informatika Menggunakan <i>Framework</i> Laravel	Universitas Mataram memiliki program studi informatika yang membutuhkan beberapa ruangan untuk mengelola kegiatan perkuliahan. Data ruangan masih disimpan secara manual, hal ini dapat	<i>Waterfall</i>	Sistem ini dirancang menggunakan <i>framework</i> Laravel dan metode pengembangan sistem <i>Waterfall</i> . Hasilnya adalah sebuah aplikasi yang mempermudah proses peminjaman dan pengelolaan ruangan oleh dosen maupun

			menyebabkan kesalahan seperti lupa jadwal kegiatan di suatu ruangan atau terbenturnya jadwal antar kegiatan, yang pada gilirannya mengurangi efisiensi pelayanan.		mahasiswa, serta membantu staf dalam mengatur jadwal dan administrasi ruangan.
8.	1. Doni Oktaviana, 2. Mustofa Abi Hamid Tahun 2021 [14].	<i>Room Loan Information System Website Based Using PHP And MYSQL With Waterfall Model.</i>	Pendataan dan pengarsipan peminjaman ruangan di Fakultas Keguruan dan Ilmu Pendidikan Universitas Sultan Ageng Tirtayasa masih dilakukan secara konvensional atau manual. Membuatnya	<i>Waterfall</i>	Pengembangan Sistem Informasi Peminjaman Ruangan berbasis web menggunakan <i>PHP</i> dan <i>MySQL</i> dengan metode pengembangan <i>Waterfall</i> . Pengujian kelayakan dilakukan melalui <i>Alpha Testing</i> oleh ahli media dan

			sulit bagi peminjam untuk mendapatkan informasi tentang jadwal peminjaman dan kondisi ruangan.		<i>Beta Testing</i> oleh pengguna, dengan hasil menunjukkan skor tinggi yang menandakan sistem sangat layak digunakan.
9.	1. Kadek Dika Sukmana, 2. Ketut Queena Fredlina, 3. Putu Trisna Hady Permana Tahun 2022 [12].	Model Aplikasi Peminjaman Ruangan Berbasis <i>Web</i> Pada Tingkat Fakultas di Perguruan Tinggi	Proses pengelolaan perizinan saat ini masih dilakukan secara manual, sehingga mungkin ada kesalahan dalam penulisan data perizinan peminjaman ruangan. Selain itu, mahasiswa atau orang lain yang ingin meminjam ruangan harus menanyakan	<i>Waterfall</i>	Aplikasi ini dibangun menggunakan <i>framework</i> Laravel dan bertujuan untuk meningkatkan efisiensi serta transparansi dalam pengelolaan peminjaman ruangan. Metode <i>Waterfall</i> diterapkan melalui tahapan perencanaan, analisis kebutuhan, desain sistem,



			jadwal secara langsung dengan mengunjungi staf perlengkapan.		implementasi, serta pengujian. Pengujian sistem menggunakan <i>blackbox testing</i> menunjukkan bahwa seluruh fitur berfungsi dengan baik dan sesuai spesifikasi
10.	1. Arif Sudharno Putro, 2. Fitri Ismuharyanti, 3. Anggit Mulat Saktiwi Tahun 2022 [1]	Analisis pemanfaatan sistem peminjaman ruangan berbasis <i>web</i> di <i>Resource Centre</i> Universitas Amikom Yogyakarta	Teknologi informasi harus digunakan untuk membantu pustakawan dan pengguna berkomunikasi di <i>Amikom Resource Centre</i> . Diperlukan penerapan teknologi informasi yaitu sistem peminjaman ruangan.	<i>Model View Controller (MVC)</i>	Pengembangan situs <i>web</i> peminjaman ruangan ini menggunakan metode <i>Model-View-Controller (MVC)</i> dalam pengembangan sistem. Metode MVC efektif dalam menghasilkan sistem yang terstruktur, efisien, dan mudah untuk dikembangkan serta dipelihara. Penerapan

					metode ini memungkinkan pengelolaan data peminjaman ruangan yang lebih baik, kemudahan dalam pemeliharaan sistem, dan peningkatan kecepatan pengembangan yang berdampak langsung pada efisiensi operasional sistem peminjaman ruangan di <i>Resource Centre</i> Universitas Amikom Yogyakarta.
11.	1. Timothy Wijaya, 2. Ramos Somya Tahun 2022 [7]	Perancangan dan Implementasi Aplikasi Peminjaman Ruangan Kedinasan Kota Salatiga menggunakan <i>Framework</i> Laravel	Peminjaman ruangan di area kedinasan Kota Salatiga masih dilakukan secara konvensional dengan cara pencatatan	<i>Waterfall</i>	Aplikasi ini menggunakan <i>framework</i> Laravel dan metode pengembangan sistem <i>Waterfall</i> . Tahapan yang

			<p>manual dalam buku, yang sangat tidak akurat sehingga pencarian informasi peminjaman memakan waktu.</p>		<p>dilakukan meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian menggunakan <i>blackbox testing</i>, hingga pemeliharaan. Hasilnya adalah aplikasi berbasis <i>web</i> yang mampu menangani proses peminjaman ruangan secara digital, mulai dari pengelolaan data ruangan, pengajuan peminjaman, hingga pencatatan riwayat.</p>
--	--	--	---	--	---

Berdasarkan penelitian terdahulu pada Tabel 2.1 dapat disimpulkan bahwa sebagian besar instansi pendidikan maupun pemerintahan

masih menggunakan sistem manual dalam pengelolaan peminjaman ruangan, yang berdampak pada kurangnya efisiensi, rawan kesalahan data, serta keterbatasan akses informasi. Solusi yang diusulkan dalam berbagai penelitian adalah pengembangan aplikasi peminjaman ruangan berbasis *web* dengan metode pengembangan sistem seperti *Waterfall*, *V-Model*, *RAD*, *SDLC*, hingga *Prototype*. Hasil dari pengembangan sistem tersebut menunjukkan peningkatan signifikan dalam proses administrasi, kemudahan pengguna dalam melakukan peminjaman dan pengecekan jadwal, serta mendukung pengelolaan data yang lebih terstruktur dan transparan. Secara umum, sistem berbasis *web* terbukti efektif dalam menyelesaikan permasalahan peminjaman ruangan yang sebelumnya dilakukan secara manual.

## **2.2. Landasan Teori**

### **2.2.1. Rancang bangun**

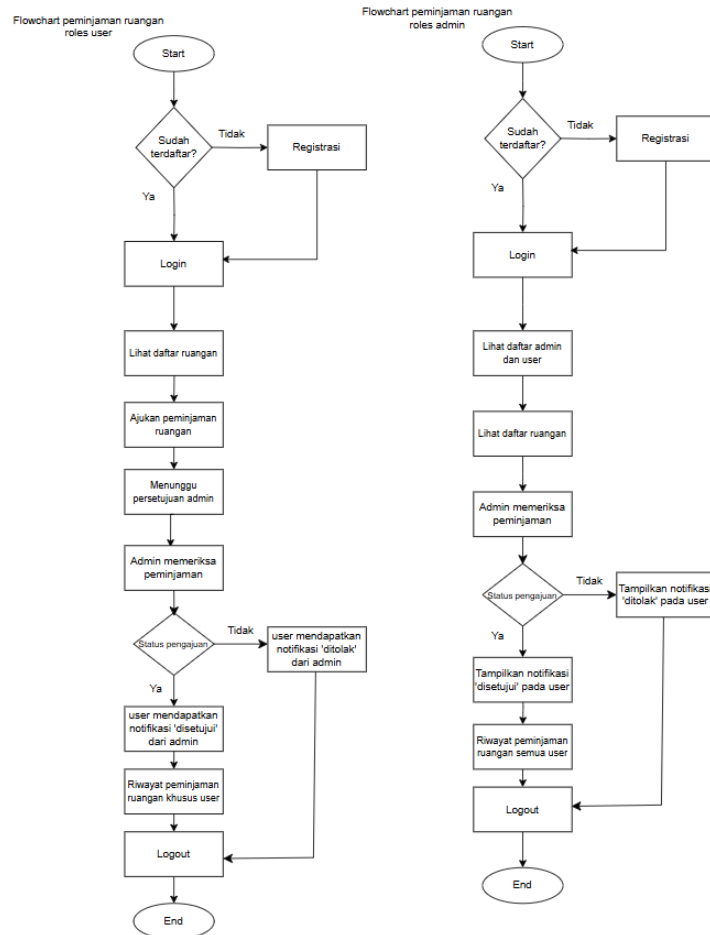
Proses desain melibatkan serangkaian langkah yang bertujuan untuk menerjemahkan hasil analisis sistem ke dalam bahasa pemrograman, sementara juga menawarkan penjelasan rinci tentang fungsionalitas yang terkait dengan setiap komponen sistem. Proses pembangunan melibatkan konstruksi komprehensif dari sistem baru atau modifikasi dan perbaikan sistem yang sudah ada secara keseluruhan. Proses rancang dan bangun melibatkan transformasi hasil analisis ke dalam perangkat lunak, yang kemudian mengarah pada pembangunan atau peningkatan sistem [16].

### **2.2.2. Website**

Situs *web* adalah sekumpulan halaman digital yang berisi informasi termasuk teks, animasi, gambar, suara, dan video. Halaman-halaman ini terhubung ke internet, jadi siapa pun yang memiliki akses internet dapat melihatnya. Sebuah situs *web* memiliki beberapa bahasa pemrograman seperti *HTML*, *PHP*, *CSS*, dan *JAVASCRIPT* [17]. Biasanya, situs *web* dapat diakses melalui *HTTP*, protokol untuk mengirimkan data dari *server web*. Situs *web* dapat diklasifikasikan ke dalam dua kategori statis dan dinamis.

### **2.2.3. Peminjaman Ruangan**

Peminjaman melibatkan tindakan menyediakan sumber daya kepada pihak lain. Peminjaman ruangan melibatkan proses peminjaman atau penyediaan ruangan kepada peminjam untuk jangka waktu tertentu. Dapat terjadi di banyak tempat, seperti sekolah, kampus, perkantoran, atau tempat umum seperti gedung pertemuan atau pusat komunitas [3]. Kegiatan kampus, baik akademik maupun non akademik, membutuhkan pemanfaatan akomodasi, khususnya di Telkom University Purwokerto. Gambar 2.1 di bawah ini menggambarkan diagram alir sistem peminjaman ruangan baik untuk *user* maupun *admin*:



Gambar 2.1 *Flowchart* sistem peminjaman ruangan

a) *Flowchart* peminjaman ruangan pada *roles user*:

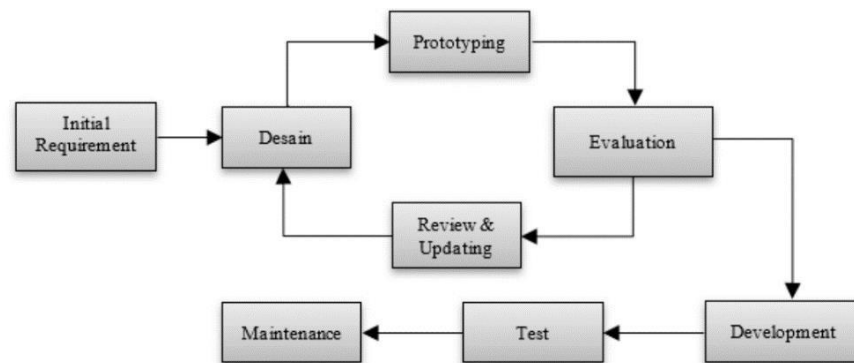
- 1) *Start*: proses peminjaman dimulai.
- 2) Cek apakah akun sudah terdaftar:  
Jika belum, *user* harus melakukan registrasi.  
Jika sudah, *user* dapat segera login.
- 3) Setelah login, *user* dapat:  
Melihat daftar ruangan yang tersedia.  
Mengajukan peminjaman ruangan yang diinginkan.  
Menunggu persetujuan dari *admin*.
- 4) Admin memeriksa pengajuan, jika pengajuan telah disetujui, maka *user* akan mendapatkan notifikasi bahwa pengajuan disetujui, dan sistem akan mencatat riwayat peminjaman khusus untuk *user* tersebut. Namun, jika pengajuan ditolak, maka *user* akan mendapatkan notifikasi bahwa pengajuan ditolak.

- 5) *Logout* dan proses selesai.
- b) *Flowchart* peminjaman ruangan pada *roles admin*:
  - 1) *Start*: proses peminjaman dimulai.
  - 2) Cek apakah akun sudah terdaftar:  
Jika belum, *admin* harus melakukan registrasi.  
Jika sudah, *admin* dapat segera login.
  - 3) Setelah melakukan *login*, *admin* dapat melihat daftar permintaan peminjaman ruangan yang telah dipinjam oleh *user*.
  - 4) *Admin* memeriksa pengajuan peminjaman, jika pengajuan disetujui, maka sistem akan menampilkan notifikasi “disetujui” kepada *user* dan mencatatnya dalam daftar peminjaman seluruh *user*. Namun, jika tidak disetujui, sistem akan menampilkan notifikasi “ditolak” kepada *user*.
  - 5) *Logout* dan proses selesai.

#### **2.2.4. Metode *Prototype***

Metode *prototype* berfungsi sebagai tahap awal dari sistem perangkat lunak, memberikan gambaran umum yang komprehensif tentang konsep, memfasilitasi eksperimen desain, mengungkap berbagai masalah, dan menentukan solusi potensial untuk tantangan-tantangan tersebut. Tujuan dari metode *prototype* yang digunakan dalam sistem ini adalah untuk mendapatkan representasi dari pemodelan situs *web* yang akan dibuat.

Awalnya, rancangan *website* dibuat dalam bentuk *prototype* yang dapat dievaluasi oleh *user*. Setelah *prototype* dievaluasi oleh *user*, *prototype* menjadi referensi bagi pengembang *software* untuk merancang *website* [18]. Fase-fase dari metode *prototype* menggunakan paradigma pembuatan *prototype Software Development Life Cycle (SDLC)*. Gambar 2.2 di bawah ini mengilustrasikan teknik *prototype*, yang merinci tahapan-tahapan selanjutnya:



Gambar 2.2 Tahapan Metode *Prototype* [19]

Tahapan pada metode *prototype* :

- a. Pengumpulan kebutuhan  
Struktur semua perangkat lunak, spesifikasi, dan garis besar sistem yang akan dikembangkan dibuat melalui proses kolaboratif yang melibatkan *user*.
- b. Membangun *prototyping*  
Membuat desain awal yang menekankan pada presentasi *user*.
- c. Evaluasi *prototyping*  
Penilaian dilakukan oleh *user*, dan jika semuanya memuaskan, tindakan selanjutnya akan dijalankan. Jika tidak ada pendekatan alternatif, proses pembuatan *prototype* akan mengalami revisi dengan meninjau kembali langkah-langkah sebelumnya.
- d. Peninjauan dan pembaruan  
Proses evaluasi dan perbaikan yang penting untuk memastikan keberlanjutan, relevansi, dan kualitas produk, layanan, atau proses.
- e. Pengembangan sistem  
Proses merancang, membangun, dan mengimplementasikan sistem baru atau meningkatkan sistem yang ada untuk memenuhi kebutuhan dan tujuan tertentu.
- f. Pengujian sistem



Setelah sistem sistem dikonversi menjadi perangkat lunak, prosedur pengujian dilakukan untuk menilai fungsionalitas aplikasi.

g. Evaluasi sistem

*User* akan menentukan apakah perangkat lunak telah diselesaikan dengan baik dan sesuai dengan harapan. Jika dianggap sesuai, maka akan dilanjutkan ke tahap berikutnya. Jika tidak, maka akan kembali ke tahap sebelumnya.

h. Penggunaan sistem

Menggunakan sistem yang telah diuji dan disetujui *user*.

### **2.2.5. Framework Laravel**

*Framework* Laravel berfungsi sebagai alat pengembangan *web* yang memanfaatkan arsitektur *Model View Controller (MVC)* dan dikembangkan dengan menggunakan bahasa pemrograman *PHP*. Kerangka kerja MVC meningkatkan kualitas perangkat lunak dengan menurunkan biaya pengembangan dan pemeliharaan awal, sementara juga meningkatkan pengalaman pengguna melalui penggunaan aplikasi yang menampilkan sintaksis yang jelas dan ekspresif, yang pada akhirnya mengarah pada penghematan waktu [20]. Aplikasi dalam MVC terdiri dari tiga komponen utama yaitu *Model*, *View*, dan *Controller*.

*Model* berfungsi untuk mengelola data sekaligus melakukan interaksi dengan *database*. *Controller* berfungsi sebagai mediator antara *model* dan *view*, mengatur transfer data dan penerapan logika bisnis, sedangkan *view* bertugas untuk presentasi dan interaksi pengguna. Laravel, dibuat oleh Taylor Otwell pada tahun 2011, disusun untuk menyederhanakan pengembangan aplikasi *web* yang menggunakan pola MVC [21]. *Framework* Laravel telah merilis versi terbaru yaitu Laravel 11 pada tanggal 12 Maret 2024 yang nantinya akan ada pembaruan mengenai penyederhanaan dalam memulai suatu proyek baru serta meningkatkan kemampuan *debugging*.

### **2.2.6. Blackbox Testing**

Pengujian *blackbox* adalah metode yang menilai fungsionalitas perangkat lunak berdasarkan spesifikasi fungsional, tanpa mempelajari desain atau kode

program. Metode ini bertujuan untuk menentukan apakah input, output, dan fungsionalitas perangkat lunak memenuhi spesifikasi yang telah ditetapkan [22]. Penilaian ini dilakukan di akhir proses pengembangan perangkat lunak untuk mengetahui kemampuan operasional perangkat lunak.

#### 2.2.7. SQL

*Standard Query Language (SQL)* berfungsi sebagai media standar untuk mengakses dan mengelola data. SQL berfungsi sebagai elemen integral dalam semua Sistem Manajemen Basis Data (DBMS), seperti *MySQL* [23]. *MySQL* berfungsi sebagai program server *database* yang mampu menerima dan mengirimkan data secara efisien. *MySQL* mendukung akses untuk banyak pengguna melalui pemanfaatan perintah SQL standar. *MySQL* digunakan secara luas dalam berbagai aplikasi termasuk dalam pengembangan *website*, karena menyediakan kemudahan penggunaan dan kinerja yang baik [24]. Pengembang dapat menggunakan SQL untuk membuat, mengelola, dan mengakses data dalam basis data dengan mudah.


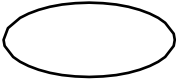


#### 2.2.8. UML

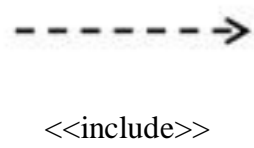
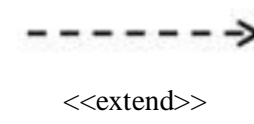
Komponen sistem perangkat lunak yang ada direpresentasikan, didefinisikan, dikembangkan, dan dicatat melalui penggunaan *Unified Modelling Language (UML)*. UML distandarisasi sebagai metode untuk mengembangkan cetak biru perangkat lunak [25]. *Use Case Diagram*, *Class Diagram*, dan *Activity Diagram*, sebagai jenis diagram yang digunakan dalam perancangan sistem yang akan dibuat.

##### a) *Use Case Diagram*

*Use Case Diagram* memberikan gambaran yang jelas tentang alur kerja sistem, menyoroti fungsi-fungsi intinya dan berbagai jenis *user* yang akan terlibat dengan sistem [26]. *Use Case Diagram* menspesifikasikan tindakan atau aktivitas yang dilakukan oleh aktor dalam sistem. Tabel 2.2 di bawah ini menyajikan penjelasan rinci tentang simbol-simbol yang digunakan dalam diagram *use case*:

Tabel 2.2 Simbol *Use Case Diagram* [27]

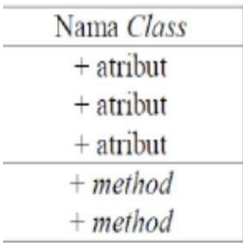




Simbol	Nama	Deskripsi
	<i>Actor</i>	Berfungsi sebagai individu yang terlibat dengan sistem.
	<i>Use case</i>	Pengamatan terhadap perilaku sistem dengan tetap menjaga kerahasiaan struktur internalnya.
	<i>Association</i>	Hubungan antara aktor dan kasus penggunaan.
	<i>Use case generalization</i>	Hubungan antara kasus penggunaan umum dan kasus penggunaan yang lebih spesifik, di mana kasus penggunaan yang lebih spesifik mewarisi atribut dan memasukkan fitur tambahan.


Simbol	Nama	Deskripsi
	<i>Include</i>	Penggabungan perilaku ke dalam kasus penggunaan dasar yang menjelaskan peningkatan.
	<i>Extend</i>	Penggabungan perilaku ke dalam kasus penggunaan dasar yang tidak teridentifikasi.

*b) Class Diagram*

*Class diagram* menggambarkan struktur kelas, atribut, dan objeknya, beserta hubungan yang ada. Diagram kelas menawarkan perspektif yang komprehensif pada sebuah sistem. Hal ini secara efektif membantu dalam memahami struktur kelas dari sebuah sistem. Diagram kelas berfungsi untuk mengilustrasikan berbagai jenis objek sistem dalam pemodelan sistem berorientasi objek, serta hubungan yang ada [27]. Penjelasan rinci mengenai simbol-simbol yang digunakan dalam diagram kelas, seperti yang diuraikan dalam Tabel 2.3 di bawah ini:

Tabel 2.3 Simbol *Class Diagram* [27]





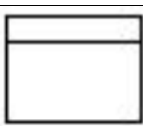
Simbol	Nama	Deskripsi
	<i>Class</i>	Sekumpulan item yang memiliki fungsi yang sama namun memiliki karakteristik yang beragam.
	<i>Association</i>	Hubungan antar kategori yang memiliki arti luas dan biasanya terkait dengan berbagai macam <i>instance</i> .
	<i>Directed Association</i>	Hubungan antar kelas yang menunjukkan bahwa satu kelas memanfaatkan fungsionalitas atau properti dari kelas lain.
	<i>Aggregation</i>	Relasi mencakup semua dimensi hubungan.
	<i>Composition</i>	Komposisi terkait dengan kelas yang menjadi sandarannya.

	<i>Dependency</i>	Mendemonstrasikan fungsionalitas kelas yang berinteraksi dengan kelas lain
---	-------------------	--

c) *Activity Diagram*

*Activity Diagram* menggambarkan alur kerja atau aktivitas dari sistem, proses bisnis, atau menu perangkat lunak saat ini. Diagram aktivitas berfokus pada aktivitas dalam sebuah sistem daripada tindakan aktor yang terlibat [27]. Bagian selanjutnya memberikan definisi rinci tentang simbol-simbol yang digunakan dalam diagram aktivitas, seperti yang diuraikan dalam Tabel 2.4 di bawah ini:

Tabel 2.4 Simbol *Activity Diagram* [27]

Simbol	Nama	Deskripsi
	<i>Initial</i>	Menunjukkan titik awal alur kerja.
	<i>Final</i>	Menunjukkan kesimpulan alur kerja.
	<i>Action</i>	Fase yang terlibat dalam suatu proses.
	<i>Decision</i>	Menunjukkan titik-titik di mana keputusan akan ditentukan.
	<i>Swimlane</i>	Mengatur kegiatan sesuai dengan aktor

		yang terlibat.
--	--	----------------

### **2.2.9. Entity Relationship Diagram**

*Entity Relationship Diagram (ERD)* menunjukkan entitas, atribut, dan hubungan antar entitas dalam struktur data sistem basis data dengan menunjukkan hubungan *one-to-many* (1:M), yang menunjukkan bahwa satu entitas dapat berhubungan dengan banyak entitas lainnya. ERD digunakan dalam perancangan basis data untuk menjamin struktur data yang efisien dan mudah dipahami [25]. ERD menempatkan setiap entitas dalam tabel. Memungkinkan pengembang untuk menjelaskan hubungan antar data dengan jelas, baik untuk aplikasi kecil maupun besar, dan memastikan bahwa data antar tabel tetap konsisten dalam sistem informasi yang kompleks.

### **2.2.10. Software Requirement & User Requirement**

*Software requirement* atau kebutuhan perangkat lunak merupakan dokumen atau daftar yang menjelaskan fungsi, fitur, dan batasan sistem perangkat lunak yang harus dimiliki untuk memenuhi kebutuhan *user*. Kebutuhan ini menjadi dasar dalam perancangan, pengembangan, dan pengujian sistem perangkat lunak. *User requirement* merupakan kebutuhan *user* terhadap sistem yang akan dibuat, yang menjelaskan apa yang harus dilakukan sistem dari perspektif *user* [29].

## **BAB 3**

### **PERANCANGAN SISTEM**

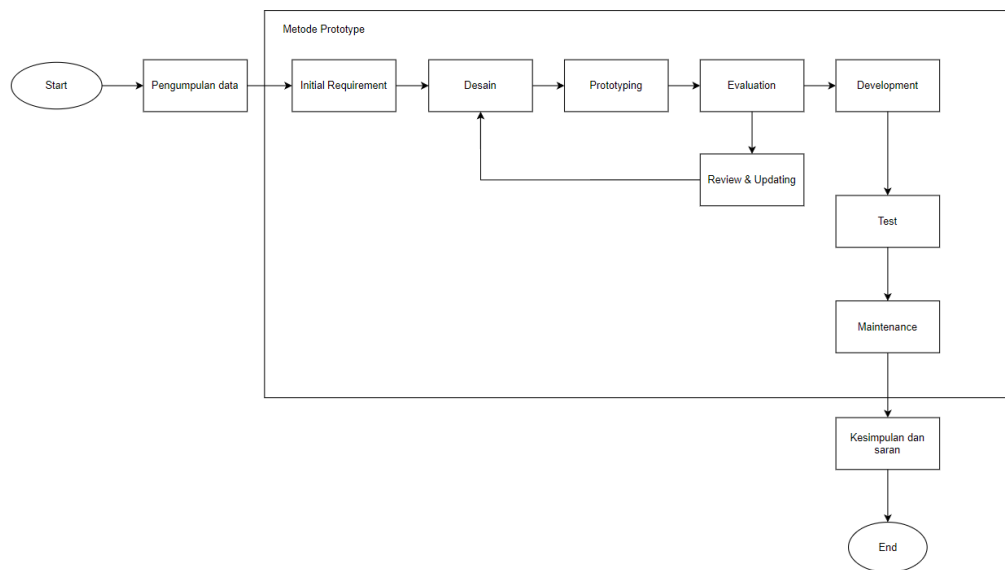
#### **3.1. Desain Perancangan Sistem**

*Website* peminjaman ruangan ini membantu mahasiswa dan staf untuk meminjam dan mengelola ruangan di kampus. Metode *prototype* digunakan untuk mengembangkan sistem ini, sehingga proses pengembangannya melibatkan model awal yang diuji dan disesuaikan berdasarkan umpan dari pengguna. *Website* ini, yang dibangun dengan *framework* Laravel, yang memiliki beberapa fitur penting, yaitu tersedianya daftar ruangan yang dapat dipinjam, formulir peminjaman, daftar atau riwayat peminjaman dari *user*, dan *dashboard admin* yang digunakan untuk mengelola permintaan peminjaman. Sistem ini dirancang untuk meningkatkan efektivitas dan transparansi pengelolaan fasilitas kampus melalui antarmuka yang responsif dan ramah pengguna.

#### **3.2. Diagram Alir**

Bagian ini menjelaskan tahapan metode *prototype* yang akan dilakukan, yaitu mengacu pada tahapan metode *prototype* yang dilakukan oleh studi yang berjudul “Perancangan Sistem Informasi Peminjaman Ruangan Berbasis *Web* (Studi Kasus: BAAK Universitas Logistik dan Bisnis Internasional” tahun 2023 [28] menggunakan metode *prototype* untuk membangun sistem informasi peminjaman ruangan secara berbasis *web* seperti pada Gambar 3.1 dibawah ini:





Gambar 3.1 Diagram alir penelitian dengan metode *prototype*

### 3.2.1. Pengumpulan data

Data yang dikumpulkan berupa data wawancara singkat dengan Kepala Urusan Pelayanan Akademik Telkom University Purwokerto mengenai sistem peminjaman ruangan manual yang ada di Telkom University Purwokerto. Data ruangan juga didapatkan untuk kelengkapan perancangan sistem. Berikut ini merupakan data ruangan yang dapat dipinjam di Telkom University Purwokerto dengan total jumlah ruangan sebanyak 44 ruangan (tidak termasuk laboratorium):

Tabel 3.1 Daftar ruangan yang dapat dipinjam

No.	Nama Gedung	Nama ruangan
1.	Gedung DC	DC-201, DC-202, DC-203, DC-301, DC-302
2.	Gedung DSP	DSP-301, DSP-302, DSP-303, DSP-304, DSP-305, DSP-306, DSP-307, DSP-308, DSP-401, DSP-402, DSP-403, DSP-404, DSP-405, DSP-501, DSP-508
3.	Gedung IOT	IOT-101, IOT-102, IOT-103, IOT-104, IOT-105, IOT-201, IOT-202
4.	Gedung Rektorat	REK-201, REK-202, REK-203, REK-204, REK-205, REK-207, REK-301, REK-302,

No.	Nama Gedung	Nama ruangan
		REK-303, REK-304, REK-305, REK-306, REK-307
5.	Gedung TT	TT-102, TT-103, TT-104, TT-105

### 3.2.2. Initial Requirement

*Initial requirement* merupakan tahap awal dari proses *prototype*, Tahap analisis kebutuhan menggambarkan struktur perangkat lunak, mengidentifikasi semua komponen yang diperlukan, dan menentukan sistem yang akan dikembangkan. Pada tahap ini, dibagi menjadi 2, yaitu:

#### a. Software Requirement

*Software Requirement* merupakan deskripsi khusus tentang fungsi, kinerja, dan keterbatasan perangkat lunak yang harus dipenuhi oleh perangkat lunak untuk memenuhi kebutuhan dan harapan *user*.

#### a) Kebutuhan fungsional

Kebutuhan fungsional merupakan deskripsi mengenai fungsi-fungsi utama yang harus disediakan oleh sistem agar dapat digunakan sebagaimana mestinya.

Tabel 3.2 Kebutuhan Fungsional

No.	Kebutuhan Fungsional
1.	Sistem menyediakan fitur <i>login</i> untuk <i>user</i> dan admin menggunakan <i>email</i> dan <i>password</i> .
2.	Sistem menampilkan daftar ruangan yang tersedia di Telkom University Purwokerto.
3.	Sistem memungkinkan <i>user</i> (mahasiswa) untuk mengisi formulir peminjaman ruangan.
4.	Sistem memungkinkan <i>admin</i> untuk melihat dan memverifikasi permintaan peminjaman.

5.	Sistem menampilkan status peminjaman (disetujui atau ditolak) kepada <i>user</i> .
6.	Sistem mencatat dan menampilkan riwayat peminjaman untuk masing-masing <i>user</i> .
7.	Sistem menyediakan fitur <i>logout</i> untuk mengakhiri sesi <i>user</i> .

b) Kebutuhan non-fungsional

Kebutuhan non-fungsional merupakan karakteristik sistem secara umum yang tidak berkaitan langsung dengan fungsi utama, tetapi berpengaruh terhadap kualitas sistem secara keseluruhan.

Tabel 3.3 Kebutuhan non-fungsional

No.	Kebutuhan Non-Fungsional
1.	Sistem hanya dapat diakses secara <i>online</i> melalui koneksi internet.
2.	Sistem dapat diakses melalui <i>browser</i> .
3.	Sistem hanya dapat diakses oleh mahasiswa dan pengelola akademik Telkom University Purwokerto.
4.	Sistem harus memiliki antarmuka yang mudah digunakan ( <i>user friendly</i> ).

b. *User Requirement*

*User Requirement* merupakan deskripsi mengenai kebutuhan dan preferensi *user* akhir sistem perangkat lunak. *User Requirement* berfokus pada pengalaman pengguna dan mencakup fitur dan fungsi yang diharapkan serta cara sistem akan digunakan dalam operasional sehari-hari. Berikut ini merupakan kebutuhan pengguna yang dapat dideskripsikan sebagai berikut:

a) Mahasiswa (*User*)

Tabel 3.4 Kebutuhan pengguna (*user*)

No.	Kebutuhan Pengguna ( <i>user</i> )
1.	Mahasiswa dapat melakukan <i>registrasi</i> dan <i>login</i> ke dalam sistem.
2.	Mahasiswa dapat melihat daftar ruangan yang tersedia.
3.	Mahasiswa dapat mengajukan peminjaman ruangan secara <i>online</i> .
4.	Mahasiswa dapat melihat status peminjaman.
5.	Mahasiswa dapat melihat riwayat peminjaman ruangan.

b) *Admin*

Tabel 3.5 Kebutuhan pengguna (*admin*)

No.	Kebutuhan Pengguna ( <i>admin</i> )
1.	<i>Admin</i> dapat melakukan <i>registrasi</i> dan <i>login</i> ke dalam sistem.
2.	<i>Admin</i> dapat melihat daftar <i>user</i> dan ruangan.
3.	<i>Admin</i> dapat melihat dan memverifikasi pengajuan peminjaman ruangan.
4.	<i>Admin</i> dapat menyetujui atau menolak peminjaman ruangan.
5.	<i>Admin</i> dapat melihat riwayat peminjaman semua <i>user</i> .

### 3.2.3. Desain

Tahap ini dilakukan pembuatan desain sederhana yang telah memberi gambaran singkat mengenai sistem yang ingin dibuat. Desain ini dibuat berdasarkan kebutuhan sistem yang telah diidentifikasi sebelumnya diterjemahkan dalam bentuk *prototype* atau model awal dari perangkat lunak. Tujuan utama dari tahap desain ini adalah untuk memberikan representasi awal dari produk yang telah dikembangkan, yang kemudian dapat divalidasi dan diperbaiki melalui umpan balik *user*.

### 3.2.4. Prototyping

Tahap *prototyping* merupakan sebuah model awal atau representasi dari sistem perangkat lunak yang diusulkan dibuat dan diperbaiki melalui iterasi berulang. Tujuan utama dari tahap ini adalah untuk membuat *prototype* yang akan membantu memahami dan mendefinisikan kebutuhan pengguna dengan lebih baik melalui pengujian dan umpan balik langsung.

### **3.2.5. Evaluation**

Tahap evaluasi, *user* dapat menilai *prototype* untuk memastikan bahwa ia memenuhi kebutuhan *user* atau tidak. Tujuan utama dari tahap evaluasi adalah untuk menemukan titik lemah *prototype* dan area yang perlu diperbaiki atau ditingkatkan. Umpan balik yang diterima selama tahap evaluasi sangat penting untuk menyempurnakan dan mengoptimalkan *prototype* sebelum dibawa ke tahap pengembangan akhir atau implementasi.

### **3.2.6. Review & Updating**

Tahap ini *prototype* dievaluasi dan diperbarui berdasarkan hasilnya. Tujuan utama dari tahap ini adalah untuk memastikan bahwa *prototype* terus disempurnakan melalui umpan balik langsung dari *user* sesuai dengan kebutuhan dan keinginan *user* sebelum dikembangkan lebih lanjut.

### **3.2.7. Development**

Tahap *development* merupakan tahap di mana *prototype* yang telah dirancang dan dievaluasi pada tahap sebelumnya dikembangkan lebih lanjut menjadi perangkat lunak yang lebih lengkap dan berfungsi. *Prototype* awal berkembang menjadi versi yang lebih canggih, sehingga mencapai produk akhir yang diinginkan, dan memiliki fitur sertakemampuan yang lebih lengkap.

### **3.2.8. Test**

Tahap selanjutnya yaitu tahap *test* di mana *prototype* yang telah dikembangkan pada tahap sebelumnya diuji secara menyeluruh untuk memastikan bahwa sistem tersebut memenuhi standar kualitas yang diharapkan dan memenuhi kebutuhan *user*. Tujuan utama dari tahap ini adalah untuk menemukan dan memperbaiki *bug*, masalah, atau kekurangan dalam *prototype* sebelum sistem digunakan oleh pengguna. Tahap *testing* ini dilakukan melalui *blackbox testing* yang merupakan pengujian dari sudut pandang *user* tanpa harus mengetahui struktur kode dari perangkat lunak.

### **3.2.9. Maintenance**

Tahap *maintenance*, *prototype* perangkat lunak yang telah dikembangkan

dan diuji akan diperbaiki dan ditingkatkan seiring waktu. Fase ini memainkan peran penting dalam proses pengembangan perangkat lunak karena menjamin fungsionalitas yang berkelanjutan dari sistem yang ada, sesuai dengan kebutuhan *user*, dan beradaptasi dengan persyaratan atau lingkungan baru.

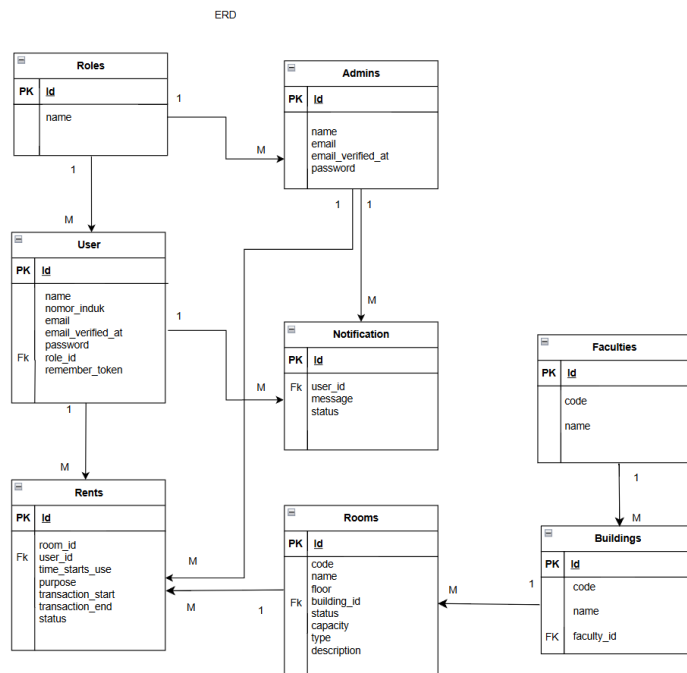
### 3.2.10. Kesimpulan dan Saran

Setelah melakukan penelitian dan uji coba hasil, maka akan dibuat rangkuman mengenai hasil dari penelitian tersebut ke dalam laporan. Kesimpulan ini akan menyajikan rangkuman komprehensif dari penelitian yang dilakukan. Saran merupakan rekomendasi atau tindakan yang disarankan berdasarkan hasil analisis atau penelitian.

## 3.3. Desain sistem

### 3.3.1. *Entity Relationship Diagram (ERD)*

ERD menunjukkan entitas, atribut, dan hubungan antar entitas dalam struktur data sistem basis data dengan menunjukkan hubungan *one-to-many* (1:M), yang menunjukkan bahwa satu entitas dapat berhubungan dengan banyak entitas lainnya. Berikut ini adalah ERD untuk peminjaman ruangan, seperti pada Gambar 3.2:



Gambar 3.2 *Entity Relationship Diagram* peminjaman ruangan

Sistem ini dirancang untuk pengelolaan peminjaman ruangan dalam suatu institusi, dan terdiri dari beberapa entitas utama dengan relasi yang saling berhubungan sebagai berikut:

- a) *Roles* menyimpan jenis peran *user*, seperti *admin* atau mahasiswa. Satu *role* dapat dimiliki oleh banyak *user*.
- b) *User* menyimpan data mahasiswa sebagai pengguna sistem, termasuk *nama*, *email*, dan *password*. Terhubung dengan tabel *Roles*, *Rents*, dan *Notification*.
- c) *Admins* menyimpan data *admin* termasuk *nama*, *email*, dan *password*.
- d) *Rents* adalah tabel yang mencatat peminjaman ruangan oleh *user*. Terhubung dengan *User* dan *Rooms*.
- e) *Rooms* menyimpan data ruangan seperti kode, nama, lantai, kapasitas, tipe, dan status. Berelasi ke *Buildings* dan *Rents*.
- f) *Building* menyimpan informasi gedung tempat ruangan berada. Berelasi ke *Rooms* dan *Faculties*.
- g) *Faculties* berisi informasi fakultas di kampus. Satu fakultas bisa memiliki banyak gedung.
- h) *Notification* mencatat pesan sistem yang ditujukan ke *user*, dengan status apakah sudah dibaca atau belum. Setiap notifikasi dikaitkan dengan satu *user*.

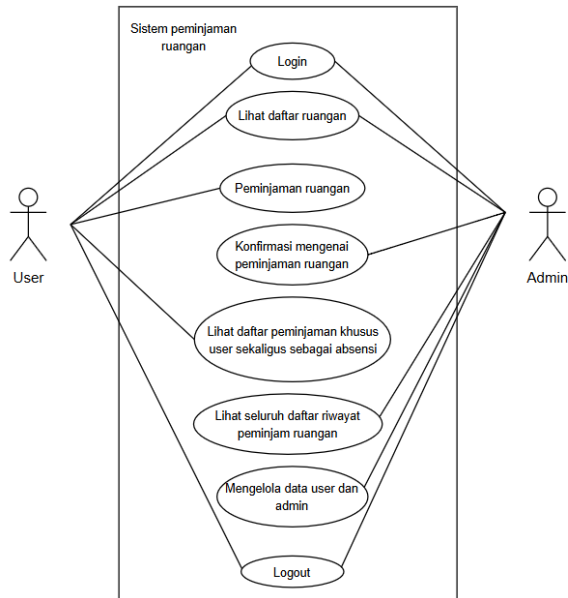
### 3.3.2. UML

Berikut ini merupakan jenis diagram yang akan digunakan dalam perancangan *website* peminjaman ruangan ini:

#### a. Use Case Diagram

Hal ini menunjukkan interaksi antara aktor dan sistem yang akan dibuat. Berikut ini adalah skema yang menunjukkan *use case* peminjaman ruangan, seperti yang terlihat pada Gambar 3.3:



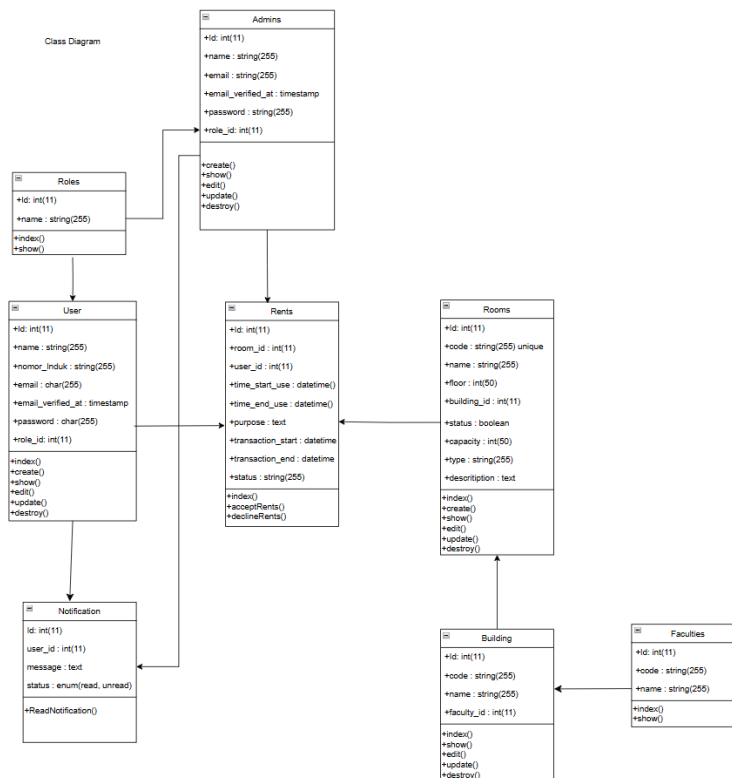


Gambar 3.3 *Use Case Diagram* peminjaman ruangan

Dua karakter dalam *use case diagram* sistem peminjaman ruang ini adalah *user* dan *admin*. Lima *use case* yang terpisah dimiliki oleh aktor *user*: *use case login*, meminjam ruang, daftar peminjaman ruang khusus *user*, dan *logout*. Aktor *admin* terhubung dengan enam *use case* yang terpisah: *use case login*, manajemen daftar ruangan, daftar riwayat peminjam ruangan secara keseluruhan, administrasi data *user* dan *admin*, dan prosedur *logout*.

#### b. *Class Diagram*

Hal ini melibatkan perincian sifat, kategori, dan *instance*, bersama dengan koneksi yang ada. Berikut ini adalah diagram kelas untuk peminjaman ruangan, seperti yang diilustrasikan pada Gambar 3.4:



Gambar 3.4 *Class Diagram* peminjaman ruangan

*Class diagram* diatas menggambarkan struktur sistem peminjaman ruangan termasuk hubungan antar kelas utama yang terlibat. Berikut ini penjelasan mengenai setiap kelasnya:

a) *Class Roles*

Kelas *Roles* merepresentasikan peran *user* dalam sistem, seperti *admin* atau mahasiswa. Kelas ini memiliki *atribut id* dan *name* sebagai nama *role*.

b) *Class User*

Kelas ini merepresentasikan *user* sistem. Atribut yang dimiliki antara lain: *id*, *name*, *nomor\_induk*, *email*, *email\_verified\_at*, *password*, dan *role\_id*. *Method* yang dimiliki adalah *create()*, *show()*, *edit()*, *update()*, dan *destroy()* untuk manajemen data.

c) *Class Admins*

Kelas ini merepresentasikan *admin*. Atribut yang dimiliki antara lain: *id*, *name*, *nomor\_induk*, *email*, *email\_verified\_at*, *password*, dan

*role\_id*. *Method* yang dimiliki adalah *create()*, *show()*, *edit()*, *update()*, dan *destroy()* untuk manajemen data.

d) *Class Rent*

Kelas *Rent* digunakan untuk merekam data transaksi peminjaman ruangan. Atributnya meliputi *room\_id*, *user\_id*, *time\_start\_use* dan *time\_end\_use* untuk menunjukkan periode pemakaian, serta atribut transaksi dan status. *Method* yang tersedia seperti *acceptRents()* dan *declineRents()* untuk memproses pengajuan.

e) *Class Room*

Kelas *Room* berisi informasi tentang ruangan yang dapat dipinjam. Atributnya antara lain *code*, *name*, *floor*, *status*, *capacity*, *type*, dan *description*. *Method* yang digunakan *index()*, *show()*, *edit()*, *update()*, dan *destroy()* untuk manajemen data.

f) *Class Building*

Kelas ini menyimpan data bangunan yang memiliki satu atau lebih ruangan. Atributnya meliputi *id*, *code*, *name*, dan *faculty\_id*. *Method* yang digunakan *index()*, *create()*, *show()*, *edit()*, *update()*, dan *destroy()* untuk manajemen data.

g) *Class Faculties*

Kelas ini menyimpan data fakultas. Atributnya meliputi *id*, *code*, dan *name*. *Method* yang digunakan *index()* dan *show()*.

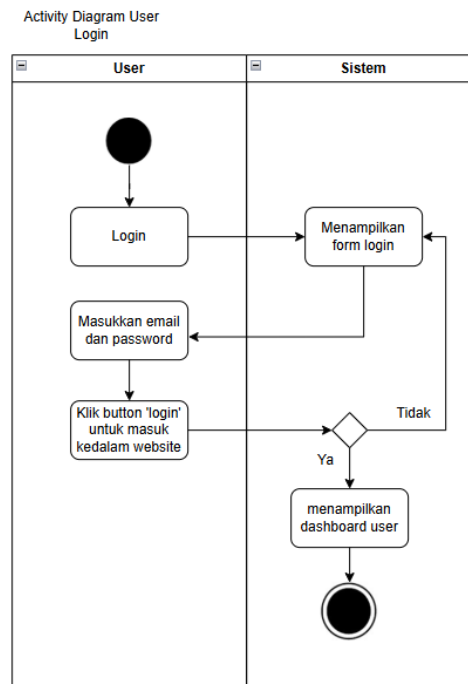
h) *Class Notification*

Kelas *Notification* menyimpan pesan yang dikirimkan kepada *user*. Atribut status menggunakan enumerasi dengan nilai *read* atau *unread*. *Method* *ReadNotification()* digunakan untuk menandai pesan sebagai sudah dibaca.

c. *Activity Diagram*

Tujuannya adalah untuk menguraikan alur kerja atau operasi-operasi dari sebuah sistem, proses bisnis, atau antarmuka perangkat lunak yang ada. Gambar berikut ini menunjukkan skema operasi peminjaman ruang:

### 1) Activity Diagram Login User

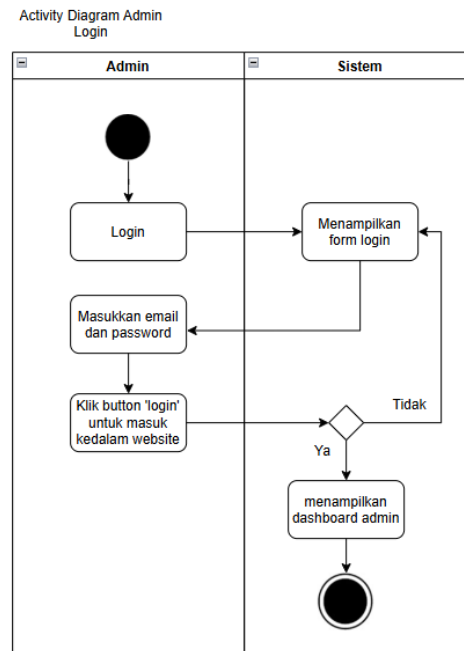


Gambar 3.5 Activity Diagram Login User

Activity diagram login pada Gambar 3.5 menggambarkan secara mendetail mengenai alur *login user* pada *website* peminjaman ruangan:

- Proses 1, *user* masuk ke halaman *login* pada *website*.
- Proses 2, sistem menampilkan form inputan *login* berupa *email* dan *password*.
- Proses 3, *user* mengisi *email* dan *password* yang terdaftar.
- Proses 4, *user* menekan tombol *login* untuk masuk kedalam *website*.
- Proses 5, jika *email* dan *password* dimasukkan dengan benar, sistem akan menampilkan dasbor *user*. Jika *user* memasukkan *email* atau *password* yang salah, sistem akan menampilkan formulir *login* sekali lagi.

## 2) Activity Diagram Login Admin

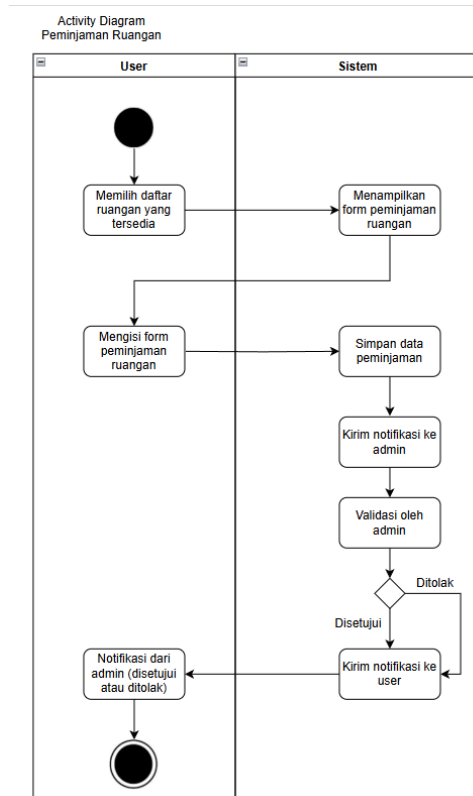


Gambar 3.6 Acitivity Diagram Login Admin

Activity diagram login pada Gambar 3.6 menggambarkan secara mendetail mengenai alur *login admin* pada *website* peminjaman ruangan:

- Proses 1, *admin* masuk ke halaman *login* pada *website*.
- Proses 2, sistem menampilkan form inputan *login* berupa *email* dan *password*.
- Proses 3, *admin* mengisi *email* dan *password* yang terdaftar.
- Proses 4, *admin* menekan tombol *login* untuk masuk kedalam *website*.
- Proses 5, jika *email* dan *password* dimasukkan dengan benar, sistem akan menampilkan dasbor *admin*. Jika *admin* memasukkan *email* atau *password* yang salah, sistem akan menampilkan formulir *login* sekali lagi.

### 3) Activity Diagram Peminjaman Ruangan



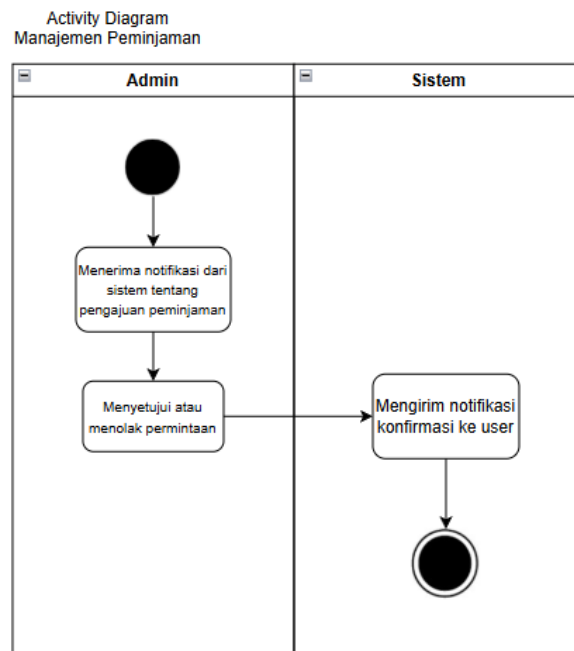
Gambar 3.7 Activity Diagram Peminjaman Ruangan

Activity diagram peminjaman ruangan pada Gambar 3.7 menggambarkan secara mendetail mengenai alur peminjaman ruangan *user* pada *website* peminjaman ruangan:

- a) Proses 1, *user* memilih daftar ruangan yang ingin dipinjam.
- b) Proses 2, sistem menampilkan form peminjaman ruang.
- c) Proses 3, *user* mengisi form peminjaman ruang.
- d) Proses 4, sistem menyimpan data peminjaman.
- e) Proses 5, sistem mengirim notifikasi pada *admin*.
- f) Proses 6, *admin* akan mengonfirmasi peminjaman apakah disetujui atau ditolak.
- g) Proses 7, jika *admin* sudah mengonfirmasi mengenai peminjaman (disetujui atau ditolak) maka notifikasi dari *admin* akan dikirim ke *user*.
- h) Proses 8, *user* akan menerima notifikasi dari *admin*

mengenai konfirmasi peminjaman ruangan.

4) *Activity Diagram* Manajemen Peminjaman (*admin*)

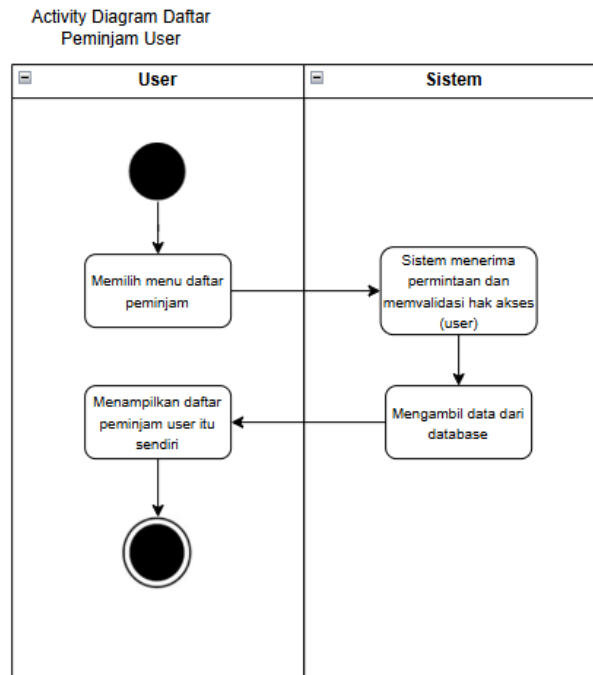


Gambar 3.8 *Activity Diagram* Manajemen Peminjaman

*Activity diagram* manajemen peminjaman pada Gambar 3.8 menggambarkan secara mendetail mengenai alur manajemen peminjaman yang dilakukan oleh *admin* pada *website* peminjaman ruangan:

- a) Proses 1, *admin* menerima notifikasi dari sistem mengenai pengajuan peminjaman.
- b) Proses 2, *admin* dapat menyetujui atau menolak peminjaman ruangan.
- c) Proses 3, sistem akan mengirim notifikasi konfirmasi dari *admin* ke *user*.

5) *Activity Diagram* Daftar Peminjaman *User*



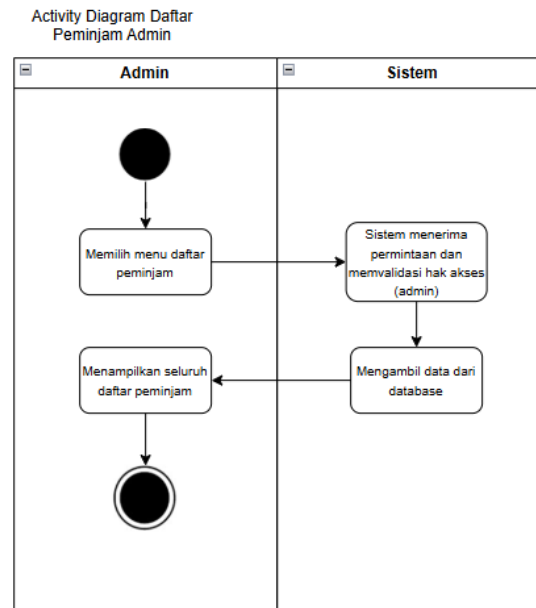
Gambar 3.9 *Activity Diagram* Daftar Peminjaman *User*

*Activity diagram* peminjaman pada Gambar 3.9 menggambarkan secara mendetail mengenai alur daftar peminjaman oleh *user* pada *website* peminjaman ruangan:

- Proses 1, *user* memilih menu daftar peminjaman.
- Proses 2, sistem menerima permintaan dan memvalidasi hak akses *user*.
- Proses 3, sistem akan mengambil data daftar peminjam dari *database*.
- Proses 4, *user* dapat melihat daftar peminjam yang telah dilakukan oleh *user* itu sendiri yang nantinya akan digunakan sebagai bukti peminjam.



## 6) Activity Diagram Daftar Peminjaman Admin



Gambar 3.10 Activity Diagram Daftar Peminjaman Admin

Activity diagram peminjaman pada Gambar 3.10 menggambarkan secara mendetail mengenai alur daftar peminjaman oleh *admin* pada *website* peminjaman ruangan:

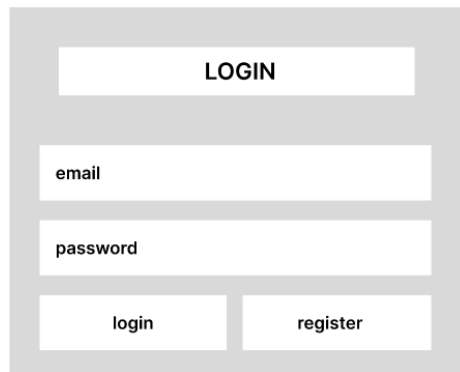
- Proses 1, *admin* memilih menu daftar peminjam.
- Proses 2, sistem menerima permintaan dan memvalidasi hak akses *admin*.
- Proses 3, sistem akan mengambil data daftar peminjam dari *database*.
- Proses 4, *admin* dapat melihat seluruh daftar peminjam yang telah dilakukan oleh *user* yang nantinya akan digunakan sebagai bukti peminjam.

### 3.4. Desain Perangkat Lunak (Iterasi 1)

#### 3.4.1. Desain Wireframe

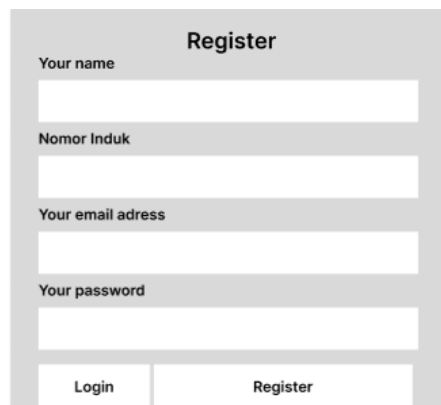
Tahap desain ini merupakan proses pengembangan yang akan berkonsentrasi pada membuat desain awal antarmuka pengguna. Memastikan bahwa struktur, tata letak, dan alur interaksi antarmuka *user* sesuai dengan tujuan dan kebutuhan sistem. Tahap ini, langkah penting adalah membuat *wireframe*, atau

desain *low-fidelity*, yang berfungsi sebagai kerangka visual awal. *Wireframe* ini akan menunjukkan fitur penting antarmuka seperti konten, posisi menu, dan tombol, tanpa memperhatikan detail estetika. Berikut ini desain *wireframe website* peminjaman ruangan:



Gambar 3.11 *Wireframe* halaman *login* (iterasi 1)

Halaman *login* memuat *email* dan *password* yang harus dimasukkan oleh *user* agar dapat melanjutkan ke halaman selanjutnya. Desain pada iterasi pertama masih bersifat sederhana dan fokus pada fungsi dasar *login*, seperti input *email*, input *password*, dan tombol *login* untuk mengirimkan data *login* ke sistem.



Gambar 3.12 *Wireframe* halaman *register* (iterasi 1)

Halaman *register* memungkinkan *user* baru untuk membuat akun agar dapat mengakses sistem. Pada iterasi pertama, tujuan utamanya adalah mengumpulkan informasi dasar dari *user* dengan proses yang sederhana. Komponen pada halaman *register* ini adalah input nama, input nomor induk, input *email*, input *password*, tombol *register* untuk mengirim data ke sistem sekaligus membuat akun, dan

tombol *login* yang nantinya akan kembali ke halaman *login* untuk menginputkan data yang tadi telah dibuat agar dapat masuk kedalam sistem.



Gambar 3.13 Wireframe dashboard user (iterasi 1)

Halaman *dashboard user* merupakan halaman utama yang ditampilkan setelah *user* berhasil *login* yang memiliki fungsi utama untuk menampilkan informasi dan akses cepat ke fitur-fitur inti yang tersedia untuk *user*. Komponen pada halaman ini yaitu *header* teks yang menampilkan teks ‘selamat datang di *website* peminjaman ruangan, foto kampus yang akan dimasukkan kedalam halaman *dashboard*, fitur seperti *home*, data ruangan yang akan menampilkan data ruangan, data peminjaman yang berisi mengenai form peminjaman ruangan yang dapat diisi oleh *user*, fitur peminjaman yang merupakan daftar peminjaman yang telah dilakukan oleh *user*, dan fitur *logout* untuk keluar dari sistem.



Gambar 3.14 Wireframe daftar ruangan user (iterasi 1)

Halaman fitur daftar ruangan yang berisi mengenai daftar ruangan yang dapat dipinjam oleh *user*. Komponen yang ada pada halaman ini mencakup daftar

nama ruangan, tombol pinjam ruang yang akan mengarahkan *user* ke form peminjaman ruangan, dan tombol angka 1, 2, 3 untuk melihat ruangan lainnya selain yang ada pada halaman itu.

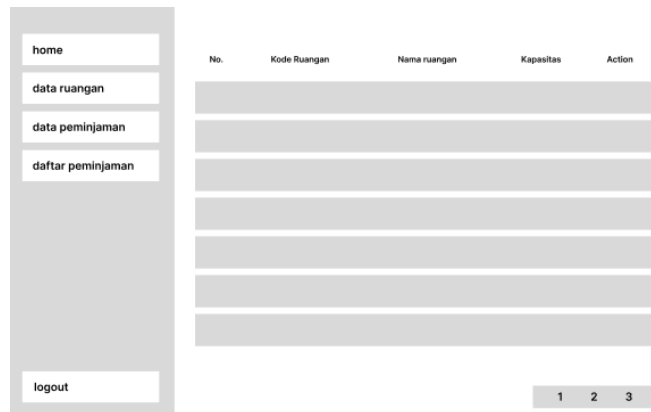
Gambar 3.15 *Wireframe* formulir peminjaman ruangan (iterasi 1)

Halaman yang memuat mengenai form peminjaman ruangan yang dapat diisi oleh *user* yang ingin mengajukan peminjaman ruangan. Desain yang masih bersifat sederhana, yang mana *user* akan mengisi ruangan yang dipinjam, tujuan, mulai pinjam, selesai pinjam, dan tombol ajukan untuk mengajukan formulir peminjaman ruangan.

No.	Kode ruangan	Nama	Mulai pinjam	Selesai pinjam	Tujuan	Status pinjam

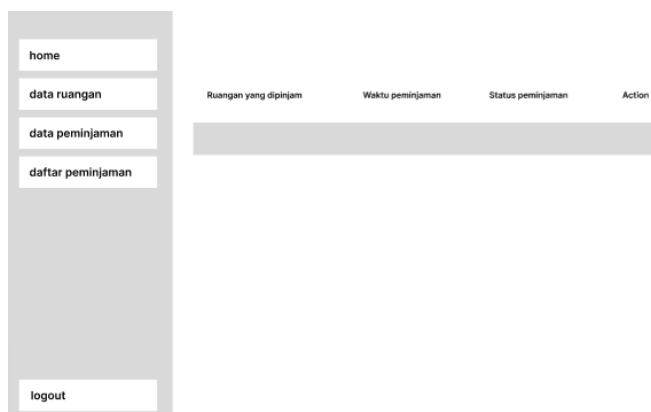
Gambar 3.16 *Wireframe* daftar peminjaman user (iterasi 1)

Halaman daftar peminjaman *user* menampilkan daftar semua permintaan peminjaman ruangan yang telah diajukan oleh *user*. Terdapat tabel daftar peminjaman yang berisi nomor, kode ruangan, nama, mulai pinjam, selesai pinjam, tujuan, dan status pinjam. Data ditampilkan dalam bentuk tabel sederhana.



Gambar 3.17 *Wireframe* daftar ruangan *admin* (iterasi 1)

Halaman daftar ruangan *admin* yang menampilkan seluruh daftar ruangan yang dapat dipinjam. Terdapat tabel daftar ruangan yang berisi nomor, kode ruangan, nama ruangan, kapasitas, dan *action* yang menunjukkan bahwa data ruangan dapat diedit atau dihapus. Data ruangan ditampilkan dalam bentuk tabel sederhana.



Gambar 3.18 *Wireframe* data peminjaman *admin* (iterasi 1)

Halaman data peminjaman pada *admin* menampilkan data mengenai ruangan yang dipinjam, waktu peminjaman, status peminjaman, dan *action* yang mana *admin* dapat menyetujui atau menolak permintaan *user*.

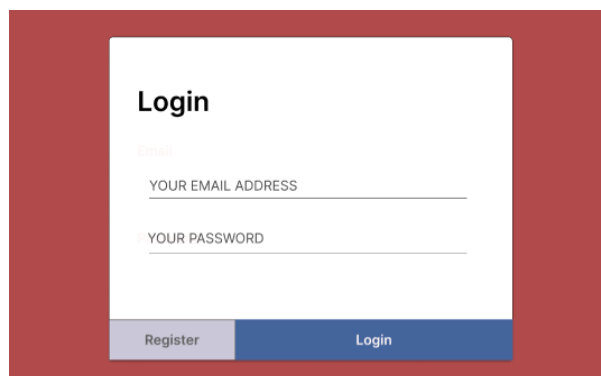


Gambar 3.19 *Wireframe* daftar peminjaman *admin* (iterasi 1)

Halaman daftar peminjaman pada *admin* yang menampilkan daftar peminjam seluruh *user* yang telah mengajukan peminjaman ruangan. Terdapat tabel daftar peminjaman yang berisi nomor, kode ruangan, nama peminjam, waktu pinjam, status pinjam, dan *action* yang mana *admin* dapat menghapus daftar peminjaman yang dilakukan oleh *user* tersebut. Data ditampilkan dalam bentuk tabel sederhana.

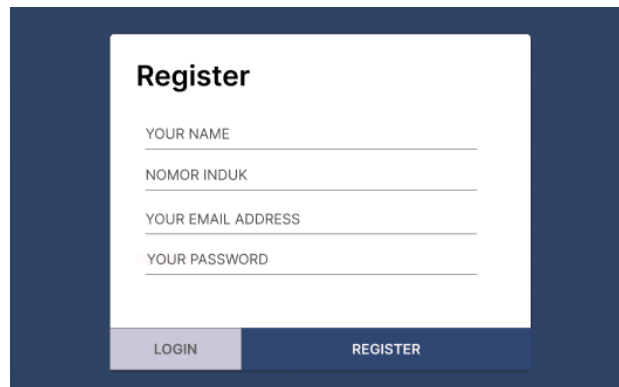
### 3.4.2. Desain *Mockup*

Tahap *prototype*, pengembangan desain akan berkonsentrasi pada membuat rancangan tampilan web versi *high-fidelity*. Versi *high-fidelity* ini merupakan penyempurnaan dari *wireframe* atau desain *low-fidelity* yang telah dibuat pada tahap sebelumnya. Elemen visual seperti interaktivitas, warna, tipografi, dan ikonografi akan ditambahkan dalam versi ini. Berikut ini merupakan *mockup* dari *website* peminjaman ruangan:



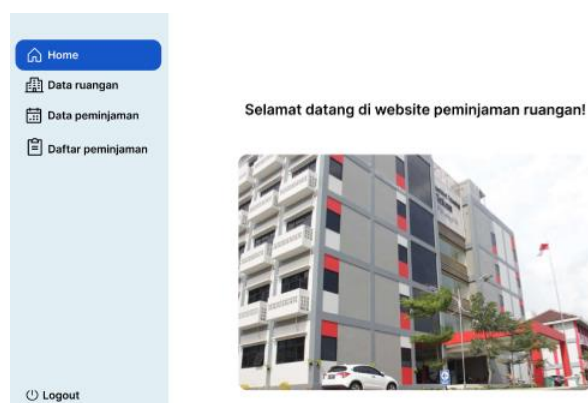
Gambar 3.20 *Mockup* halaman *login* (iterasi 1)

*Mockup* halaman *login* dirancang sebagai representasi awal dari antarmuka user untuk proses autentikasi. Tujuan utama *mockup* ini adalah memberikan gambaran visual mengenai letak dan fungsi elemen-elemen penting pada halaman *login*, sebelum masuk ke tahap pengembangan. Elemen yang terdapat pada halaman *login* berupa formulir *login* yang berisi input *email*, input *password*, dan tombol *login* untuk mengarahkan *user* ke *dashboard*.



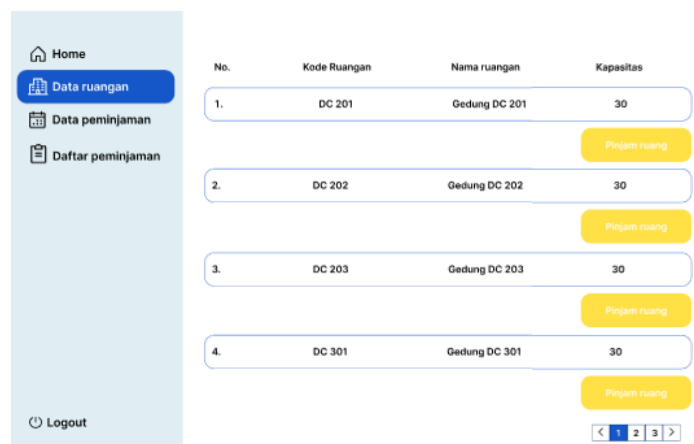
Gambar 3.21 *Mockup* halaman *register* (iterasi 1)

*Mockup* halaman *register* dirancang sebagai rancangan awal antarmuka untuk *user* baru yang ingin membuat akun. Fokus utama adalah menyusun elemen-elemen dasar pendaftaran secara jelas, sederhana, dan mudah dipahami. Formulir *register* berisi inputan nama, nomor induk, *email*, *password*, tombol *register* untuk mendaftarkan akun yang baru dibuat, serta tombol *login* untuk menuju ke halaman *login*. *Mockup* ini masih dalam tahap awal, bertujuan menyiapkan struktur dan alur dasar dari halaman *register*.



Gambar 3.22 *Mockup* dashboard *user* (iterasi 1)

*Mockup* halaman *dashboard user* didesain untuk menjadi pusat navigasi utama bagi *user* setelah berhasil *login*. *Mockup* ini menyusun struktur awal dari tampilan *dashboard* beserta fitur-fitur penting yang dibutuhkan *user* dalam proses peminjaman ruangan. Desain *mockup* ini dengan *sidebar* navigasi (bagian kiri) yang berisi fitur *home* atau beranda, data ruangan yang menampilkan daftar ruangan yang dapat dipinjam, data peminjaman yang berisi formulir peminjaman, daftar peminjam yang berisi tabel riwayat peminjaman yang telah dilakukan oleh *user*, dan *logout* untuk keluar dari sistem.



The mockup shows a user dashboard with a light blue sidebar on the left and a white main content area. The sidebar contains a 'Home' icon, a 'Data ruangan' button (highlighted in blue), 'Data peminjaman', 'Daftar peminjaman', and a 'Logout' link. The main content area displays a table of rooms with columns for 'No.', 'Kode Ruangan', 'Nama ruangan', and 'Kapasitas'. There are four rows of room data, each with a yellow 'Pinjam ruang' button to its right. At the bottom right of the table, there is a pagination control showing '< 1 2 3 >'.

No.	Kode Ruangan	Nama ruangan	Kapasitas
1.	DC 201	Gedung DC 201	30
2.	DC 202	Gedung DC 202	30
3.	DC 203	Gedung DC 203	30
4.	DC 301	Gedung DC 301	30

Gambar 3.23 *Mockup* daftar ruangan *user* (iterasi 1)

*Mockup* halaman daftar ruangan merupakan bagian dari *dashboard user* yang menampilkan informasi lengkap mengenai ruangan-ruangan yang tersedia untuk dipinjam. Tujuan dari *mockup* ini adalah menyusun tampilan awal yang memudahkan *user* dalam melihat dan memilih ruangan yang ingin dipinjam. Berisi tabel yang berisi daftar ruangan dengan rincian kode ruangan, nama ruangan, kapasitas, dan tombol pinjam ruang yang akan mengarahkan *user* ke formulir peminjaman ruangan. *Mockup* pada tahap ini masih berfokus pada fungsi dasar dan struktur informasi.



Gambar 3.24 *Mockup* formulir peminjaman ruangan (iterasi 1)

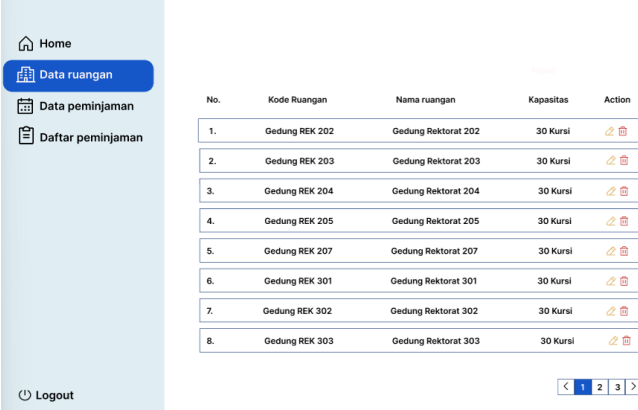
*Mockup* halaman formulir peminjaman ruangan dirancang sebagai sarana bagi *user* untuk mengajukan peminjaman ruangan secara langsung melalui sistem. *Mockup* pada Iterasi 1 difokuskan pada kelengkapan informasi dan kemudahan *user* dalam mengisi data. Formulir di atas berisi inputan mengenai ruangan yang dipinjam, tujuan, mulai pinjam, selesai pinjam, dan tombol ajukan untuk mengajukan peminjaman ruangan. *Mockup* ini berfokus pada kelengkapan fungsional dan alur pengisian data. Estetika dan pengalaman pengguna akan ditingkatkan pada iterasi selanjutnya, berdasarkan umpan balik dari *user*.

No.	Kode ruangan	Nama	Mulai pinjam	Selesai pinjam	Tujuan	Status pinjam
1.	Gedung REK 202	Nabila	2025-03-09 10:00:00	2025-03-09 12:00:00	Rapat	Pending

Gambar 3.25 *Mockup* daftar peminjaman user (iterasi 1)

*Mockup* halaman daftar peminjaman dirancang sebagai tampilan untuk melihat status dan riwayat peminjaman ruangan yang telah diajukan oleh *user*. Tujuannya adalah agar *user* dapat memantau progres pengajuan peminjaman ruangan. Tabel daftar peminjaman yang berisi nomor, kode ruangan, nama, mulai pinjam, selesai pinjam, tujuan, dan status pinjam. *Mockup* masih fokus pada

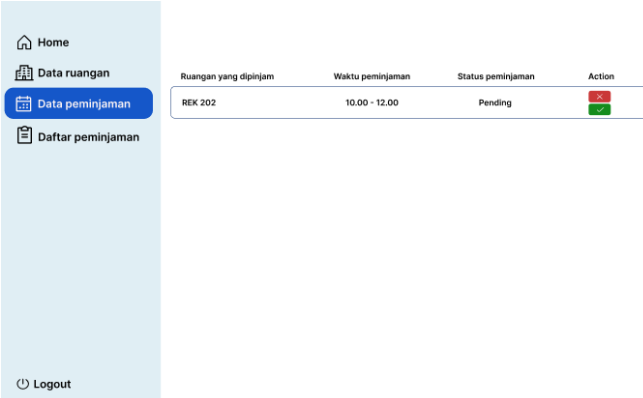
struktur informasi dan alur akses data oleh *user*. Desain akhir akan ditingkatkan pada iterasi berikutnya.



No.	Kode Ruangan	Nama ruangan	Kapasitas	Action
1.	Gedung REK 202	Gedung Rektorat 202	30 Kursi	
2.	Gedung REK 203	Gedung Rektorat 203	30 Kursi	
3.	Gedung REK 204	Gedung Rektorat 204	30 Kursi	
4.	Gedung REK 205	Gedung Rektorat 205	30 Kursi	
5.	Gedung REK 207	Gedung Rektorat 207	30 Kursi	
6.	Gedung REK 301	Gedung Rektorat 301	30 Kursi	
7.	Gedung REK 302	Gedung Rektorat 302	30 Kursi	
8.	Gedung REK 303	Gedung Rektorat 303	30 Kursi	

Gambar 3.26 *Mockup* daftar ruangan *admin* (iterasi 1)

*Mockup* halaman daftar ruangan *admin* dibuat khusus untuk *admin* agar dapat melihat, mengedit, dan menghapus data ruangan yang tersedia dalam sistem. Fokus pada iterasi pertama adalah menyediakan kontrol dasar untuk manajemen data ruangan. Tabel daftar ruangan yang berisi mengenai nomor, kode ruangan, nama ruangan, kapasitas, dan *action* untuk mengedit atau menghapus data ruangan. *Mockup* ini berfungsi untuk menunjukkan fitur utama manajemen data ruangan, khususnya bagi *admin*.

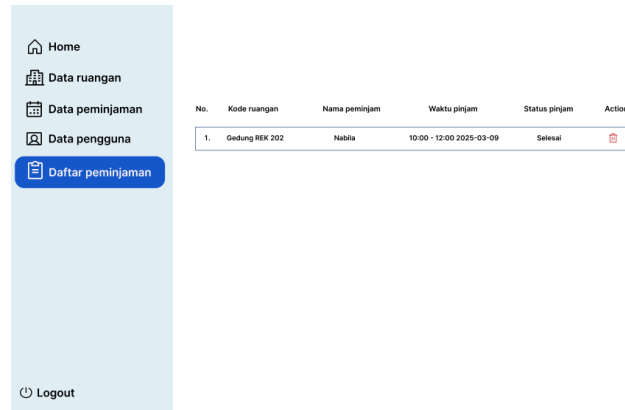


Ruangan yang dipinjam	Waktu peminjaman	Status peminjaman	Action
REK 202	10.00 - 12.00	Pending	

Gambar 3.27 *Mockup* data peminjaman *admin* (iterasi 1)

*Mockup* halaman data peminjaman khusus untuk *admin* ini berfungsi sebagai tempat untuk melihat seluruh pengajuan peminjaman dari *user*, serta memberikan keputusan konfirmasi berupa disetujui atau ditolak. Tabel pada halaman ini berisi mengenai ruangan yang dipinjam, waktu peminjaman, status peminjaman, dan *action* untuk menyetujui (tombol hijau) atau menolak (tombol

merah) peminjaman. *Mockup* ini difokuskan untuk mengatur alur verifikasi peminjaman oleh *admin* secara efisien. Pengembangan visual lebih lanjut dan integrasi notifikasi akan dilakukan di iterasi selanjutnya.



Gambar 3.28 *Mockup* daftar peminjaman *admin* (iterasi 1)

*Mockup* halaman daftar peminjaman *admin* berfungsi sebagai tempat bagi *admin* untuk melihat seluruh riwayat peminjaman ruangan dari semua *user*, baik yang masih berlangsung, sudah selesai, maupun yang pernah ditolak. *Mockup* ini membantu *admin* dalam proses pemantauan dan dokumentasi. Tabel daftar peminjaman *admin* ini berisi nomor, kode ruangan, nama peminjam, waktu pinjam, status pinjam, dan *action* untuk menghapus data peminjaman. Tampilan pada halaman ini masih sederhana dan fokus pada fungsionalitas dasar.

### 3.5. Evaluasi (Iterasi 1)

Tahap evaluasi ini, rancangan *prototype* yang telah dibuat sebelumnya akan ditinjau dan diperbaiki untuk memastikan tampilan dan fungsinya lebih optimal. Tahap ini juga mencakup *review and updating* mengenai hasil dari evaluasi. Salah satu aspek penting dalam evaluasi ini adalah peningkatan daya tarik visual tampilan *web* untuk menjadikannya lebih menarik dan mudah digunakan. Berikut ini merupakan umpan balik dari *user* dan *admin* mengenai desain dari iterasi 1 pada Tabel 3.6:

Tabel 3.6 Umpan balik *user* dan *admin* pada iterasi 1

No.	Nama	Jabatan	Umpan balik
1.	Rizqina Nabila Fitriana	Mahasiswa Telkom University Purwokerto	<ul style="list-style-type: none"> <li>• <i>Dashboard user</i> dapat dibuat agar lebih menarik dan modern tidak terlalu monoton.</li> <li>• Daftar ruangan yang ditampilkan jangan berbentuk tabel seperti itu, buat <i>layout</i> yang lebih menarik untuk dilihat.</li> <li>• Navigasi <i>bar</i> lebih baik diletakkan diatas jangan di sebelah kiri.</li> </ul>
2.	Sa' Baniatun Astia	<i>Student</i> staf akademik Telkom University Purwokerto	<ul style="list-style-type: none"> <li>• Formulir peminjaman lebih baik disertakan dengan deskripsi ruang juga agar lebih lengkap informasinya.</li> <li>• Tambahkan fitur notifikasi saat <i>user</i> meminjam ruangan agar</li> </ul>

			admin dapat mengetahui ada yang meminjam. <ul style="list-style-type: none"> <li>• Tambahkan daftar <i>admin</i> dan daftar <i>user</i> untuk mengelola data lebih baik.</li> </ul>
--	--	--	---

### 3.6. Review & Updating

Berdasarkan hasil evaluasi pada iterasi pertama, sejumlah pembaruan dan perbaikan dilakukan pada *prototype* untuk meningkatkan tampilan dan fungsionalitas sistem. Iterasi 2 dirancang untuk mengatasi kekurangan yang ditemukan pada iterasi pertama dan memastikan bahwa setiap elemen pada sistem memberikan pengalaman yang lebih optimal. Salah satu fokus utama adalah peningkatan daya tarik visual yang bertujuan untuk membuat tampilan lebih menarik dan nyaman digunakan.

Fitur baru juga ditambahkan untuk memperbaiki pengelolaan data, termasuk daftar *admin* dan daftar *user*, yang memungkinkan *admin* untuk lebih mudah mengelola informasi. Selain itu, untuk meningkatkan komunikasi, fitur notifikasi dibuat untuk memberikan pemberitahuan kepada *admin* dan *user* mengenai status peminjaman ruangan, apakah disetujui atau ditolak. Pembaruan ini diharapkan dapat memperbaiki pengalaman pengguna secara keseluruhan dan meningkatkan efisiensi sistem.

### 3.7. Desain Perangkat Lunak (Iterasi 2)

#### 3.7.1. Desain Wireframe

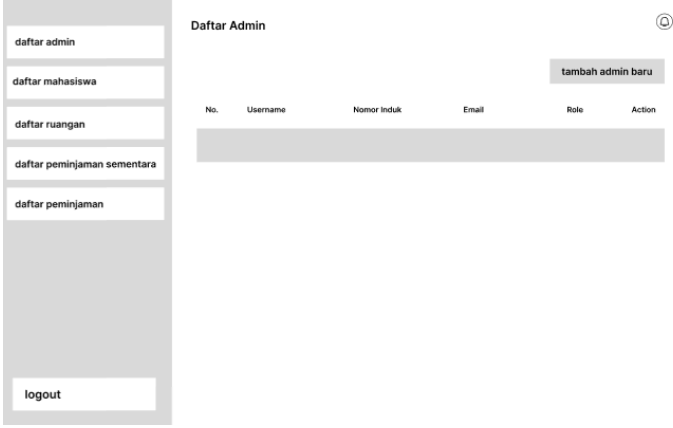
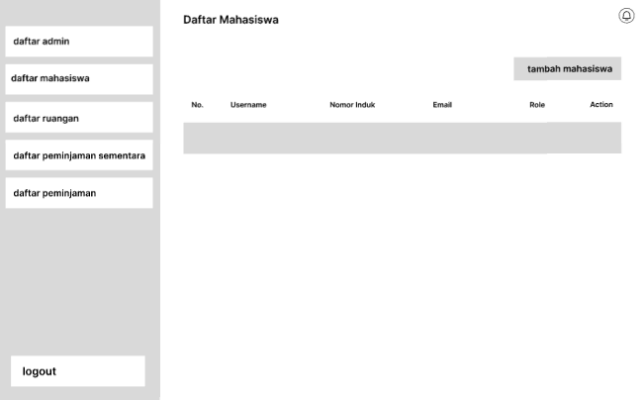
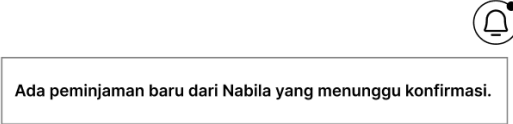
Iterasi 2 ini akan dilakukan perubahan sesuai dengan umpan balik dari *user* mengenai rancangan *low fidelity* atau *wireframe*. Tabel 3.7 menunjukkan perbedaan antara iterasi 1 dan iterasi 2:

Tabel 3.7 Perubahan *wireframe* pada iterasi 2

Sebelum	Sesudah
 <p>Halaman <i>dashboard user</i> iterasi 1</p>	 <p>Halaman <i>dashboard user</i> iterasi 2</p>
 <p>Halaman daftar ruang iterasi 1</p>	 <p>Halaman daftar ruang iterasi 2</p>
 <p>Formulir peminjaman ruang iterasi 1</p>	 <p>Formulir peminjaman ruang iterasi 2</p>

Iterasi 2 ini juga terdapat penambahan halaman seperti fitur daftar admin, daftar user, dan notifikasi seperti yang ditunjukkan pada Tabel 3.8:

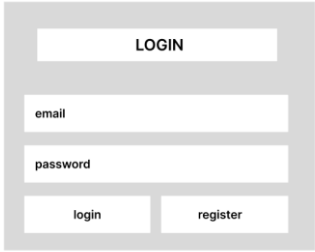
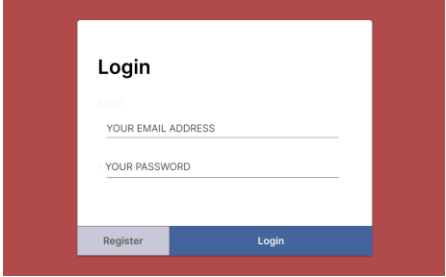

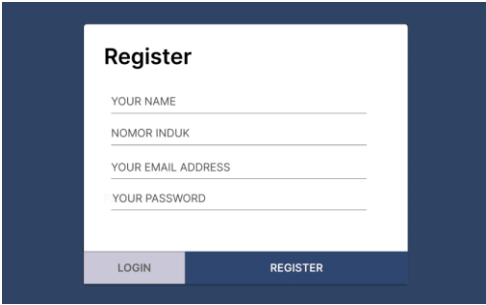
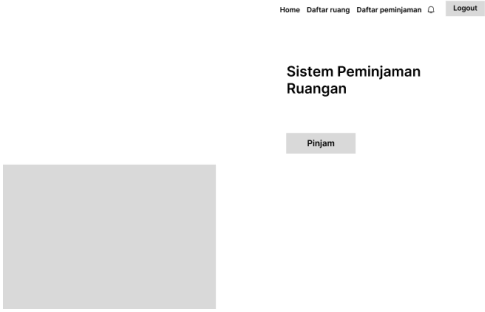



Tabel 3.8 Penambahan *wireframe* pada iterasi 2

No.	Wireframe yang ditambahkan
1.	 <p>Halaman daftar <i>admin</i></p>
2.	 <p>Halaman daftar mahasiswa</p>
3.	 <p>Notifikasi pinjaman ruangan</p>

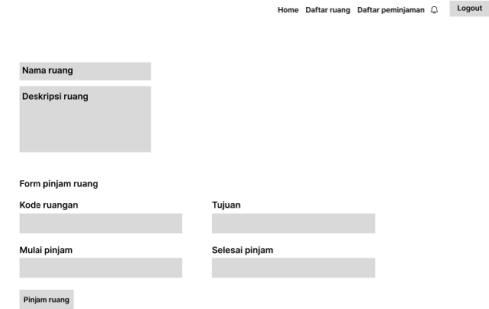
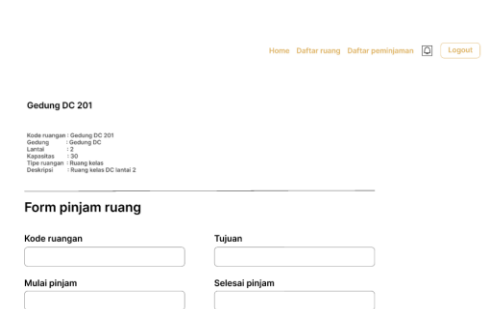
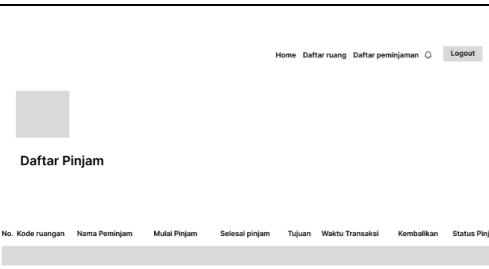

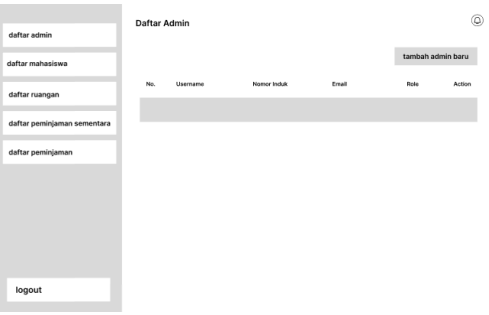
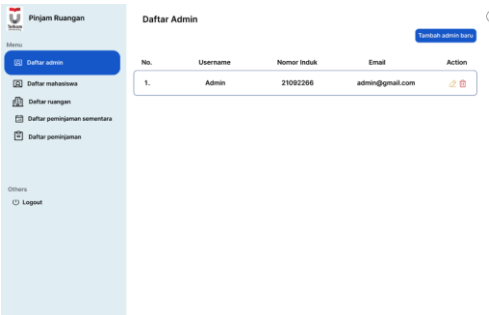
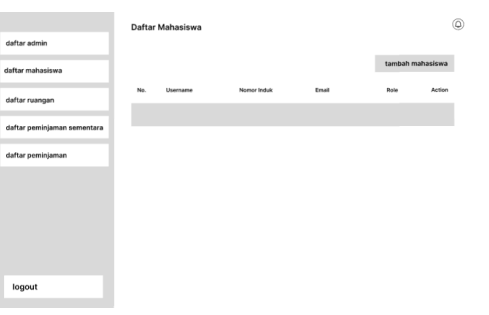

### 3.7.2. Desain Mockup

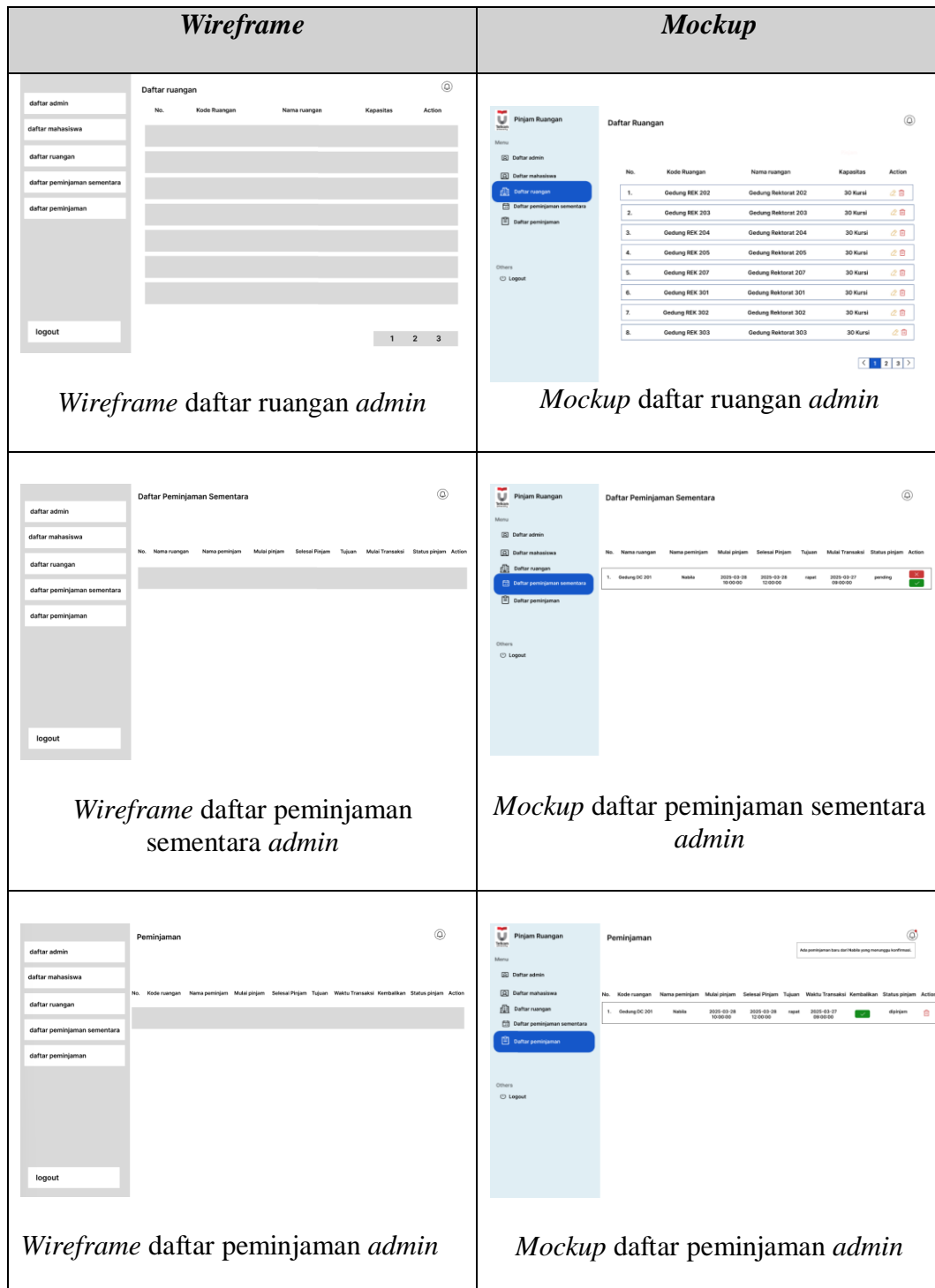
*High fidelity prototype* atau *mockup* dibuat berdasarkan hasil desain terakhir pada iterasi 2. Berikut ini merupakan perubahan desain dari *low fidelity* menuju ke *high fidelity* yang ditunjukkan pada Tabel 3.9:

Tabel 3.9 Perubahan desain *wireframe* dan *mockup*

<i>Wireframe</i>	<i>Mockup</i>
 <p><i>Wireframe halaman login</i></p>	 <p><i>Mockup halaman login</i></p>
 <p><i>Wireframe halaman register</i></p>	 <p><i>Mockup halaman register</i></p>
 <p><i>Wireframe dashboard user</i></p>	 <p><i>Mockup dashboard user</i></p>
 <p><i>Wireframe daftar ruangan user</i></p>	 <p><i>Mockup daftar ruangan user</i></p>



Wireframe	Mockup
 <p>Wireframe formulir peminjaman</p>	 <p>Mockup formulir peminjaman</p>
 <p>Wireframe daftar peminjaman user</p>	 <p>Mockup daftar peminjaman user</p>
 <p>Wireframe daftar admin</p>	 <p>Mockup daftar admin</p>
 <p>Wireframe daftar mahasiswa</p>	 <p>Mockup daftar mahasiswa</p>



### 3.8. Evaluasi (Iterasi 2)

Iterasi 2 telah berhasil mengakomodasi kebutuhan *user* dan *admin* dengan peningkatan signifikan pada tampilan visual dan fungsionalitas. Desain yang lebih modern dan formulir peminjaman yang lebih detail memudahkan *user* dalam

mengajukan peminjaman dengan informasi yang lebih akurat. Fitur notifikasi juga memberikan kemudahan bagi *user* dan *admin* mengenai pemberitahuan peminjaman dan konfirmasi peminjaman dari *admin* ke *user*. Penambahan daftar *admin* dan daftar *user* memperbaiki pengelolaan data, membuat proses lebih efisien bagi *admin*. Secara keseluruhan, iterasi 2 telah berhasil meningkatkan pengalaman *user* dan memberikan sistem yang lebih intuitif. Berikut ini merupakan umpan balik mengenai perbaikan dari iterasi 2 yang ditunjukkan pada Tabel 3.10:

Tabel 3.10 Umpan balik *user* dan *admin* pada iterasi 2

No.	Nama	Jabatan	Umpan balik
1.	Rizqina Nabila Fitriana	Mahasiswa Telkom University Purwokerto	<ul style="list-style-type: none"> <li>• Sudah lebih baik tampilannya dari iterasi 1.</li> <li>• Tata letak fitur juga tidak membingungkan.</li> <li>• Adanya fitur notifikasi sangat membantu <i>user</i> saat menerima notifikasi dari <i>admin</i> mengenai konfirmasi peminjaman.</li> </ul>
2.	Sa' Baniatun Astia	Student staf akademik Telkom University Purwokerto	<ul style="list-style-type: none"> <li>• Adanya fitur daftar <i>admin</i> dan daftar mahasiswa membantu dalam mengelola data <i>user</i> dan <i>admin</i>.</li> </ul>

			<ul style="list-style-type: none"> <li>• Fitur notifikasi membantu <i>admin</i> saat ada mahasiswa yang mengajukan peminjaman.</li> </ul>
--	--	--	---

## BAB 4

### HASIL PERCOBAAN DAN ANALISIS

#### 4.1. Hasil Akhir Sistem

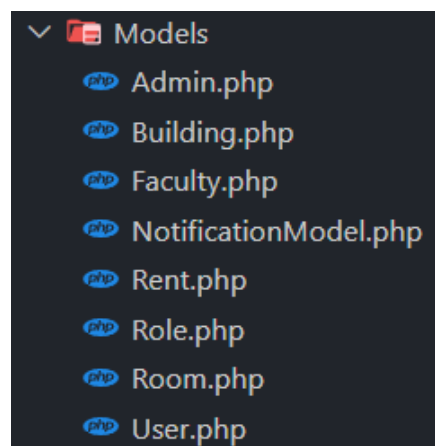
##### 4.1.1. Perancangan

Perancangan sistem diawali dengan proses pengumpulan data yang mencakup jenis ruangan, kapasitas, serta lokasi ruangan. Data ini diperoleh melalui dokumentasi dan observasi langsung. Hal ini dilakukan untuk memastikan bahwa sistem yang dibuat sesuai dengan kebutuhan nyata institusi.

*Framework* Laravel digunakan untuk pengembangan sistem karena mendukung arsitektur *Model-View-Controller (MVC)*, keamanan yang terintegrasi, dan kemudahan pemeliharaan sistem jangka panjang. Selain itu, penggunaan metode *prototype* memungkinkan proses pengembangan berlangsung secara iteratif dan menyesuaikan diri dengan umpan balik *user*. Berdasarkan penggabungan keduanya, sistem diharapkan dapat menyediakan produk digital yang efisien, dinamis, dan tepat guna untuk membantu manajemen peminjaman ruangan secara menyeluruh.

##### a) *Model*

*Model* berfungsi untuk mengelola data sekaligus melakukan interaksi dengan *database*. Berikut ini struktur *Model* yang digunakan pada Gambar 4.1:

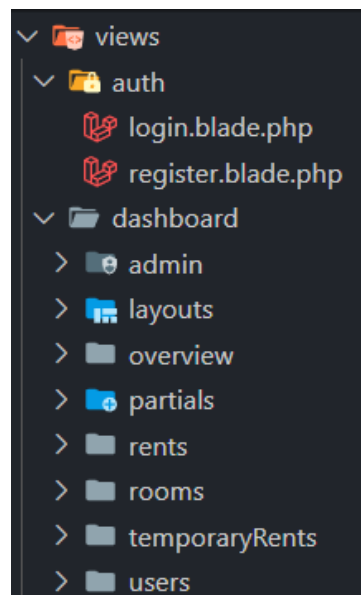


Gambar 4.1 Struktur *Models* pada sistem

- a) *Admin*: merepresentasikan data administrator sistem.
- b) *Building*: digunakan untuk mengelola data gedung.
- c) *Faculty*: berisi data fakultas yang ada pada sistem.
- d) *NotificationModel*: menangani notifikasi yang muncul dalam sistem.
- e) *Rent*: digunakan untuk memproses data peminjaman ruang.
- f) *Role*: menyimpan data peran *user* dalam sistem
- g) *Room*: mengelola data ruangan.
- h) *User*: merepresentasikan data *user* sistem.

b) *View*

*View* merupakan bagian antarmuka yang berfungsi menampilkan data ke *user*. Berikut ini struktur *folder View* yang digunakan pada Gambar 4.2:



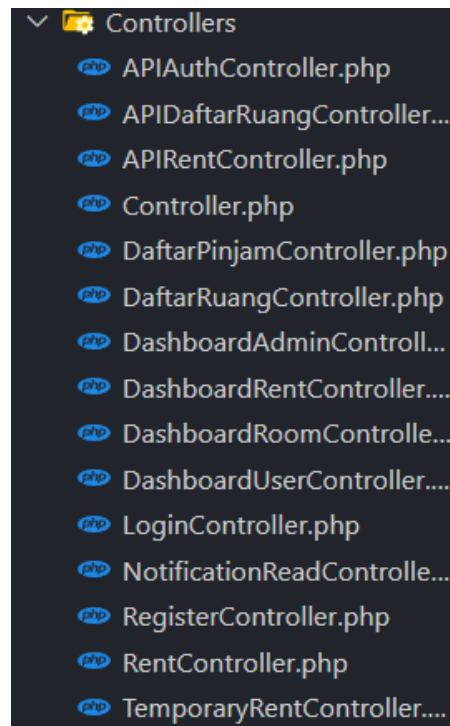
Gambar 4.2 Struktur folder *Views* dalam sistem

- a) *auth/login.blade.php* dan *auth/register.blade.php*: digunakan untuk proses *login* dan registrasi *user*.
- b) *dashboard/layouts* dan *partials*: berisi *layout* utama dan komponen seperti *sidebar* atau *header* yang digunakan secara global.

- c) *dashboard/rooms* dan *dashboard/rents*: menampilkan data ruangan dan riwayat peminjaman ruangan.
- d) *dashboard/temporaryRents* dan *dashboard/users*: mengelola data peminjaman sementara serta *user*.

### c) *Controller*

*Controller* berfungsi sebagai mediator antara *model* dan *view*. *Controller* menerima permintaan (*request*) dari *user*, memproses logika yang dibutuhkan (misalnya mengambil data dari *database* melalui *Model*), dan kemudian meneruskan hasilnya ke *View* untuk ditampilkan. Berikut ini struktur *controller* yang digunakan pada Gambar 4.3:



Gambar 4.3 Struktur *Controller* pada sistem

- a) *LoginController* dan *RegisterController*: menangani proses autentikasi *user*.
- b) *DaftarRuangController*: menampilkan daftar ruangan yang tersedia.

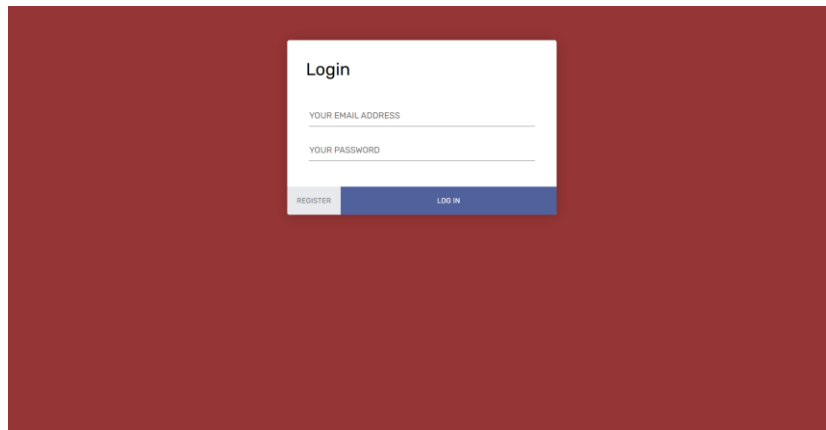
- c) *RentController*, *TemporaryRentController*, dan *DaftarPinjamController*: mengatur proses peminjaman ruang secara permanen maupun sementara.
- d) *DashboardAdminController*, *DashboardUserController*, *DashboardRoomController*, dan *DashboardRentController*: menangani tampilan dashboard sesuai dengan peran *user* dan modul yang diakses.
- e) *APIRentController*, *APIAuthController*, dan *APIDAftarRuangController*: *controller* khusus untuk mengatur *endpoint* API.
- f) *NotificationReadController*: menangani proses pembacaan atau update status notifikasi oleh *user*.

#### **4.1.2. Development**

Hasil dari tahap pengembangan (perancangan dan desain) ini adalah sebuah *website* peminjaman ruangan yang berfungsi dan memiliki fitur utama yang dirancang untuk memenuhi kebutuhan *user* dan *admin*. Tampilan dan fiturnya telah disesuaikan sesuai dengan rancangan awal, termasuk halaman *login* dan *register*, daftar ruangan, formulir peminjaman, daftar peminjaman atau riwayat peminjaman, pengelolaan data oleh *admin*, persetujuan peminjaman oleh *admin*, dan *logout*. Berikut ini merupakan tampilan antarmuka (UI) dari *website* peminjaman ruangan yang telah dibuat:



a) Halaman *login*

The image shows a login page with a solid red background. In the center, there is a white rectangular box containing the login form. The form has the title "Login" at the top. Below the title, there are two input fields: "YOUR EMAIL ADDRESS" and "YOUR PASSWORD". At the bottom of the form, there are two buttons: "REGISTER" on the left and "LOG IN" on the right. The "LOG IN" button is highlighted in blue.

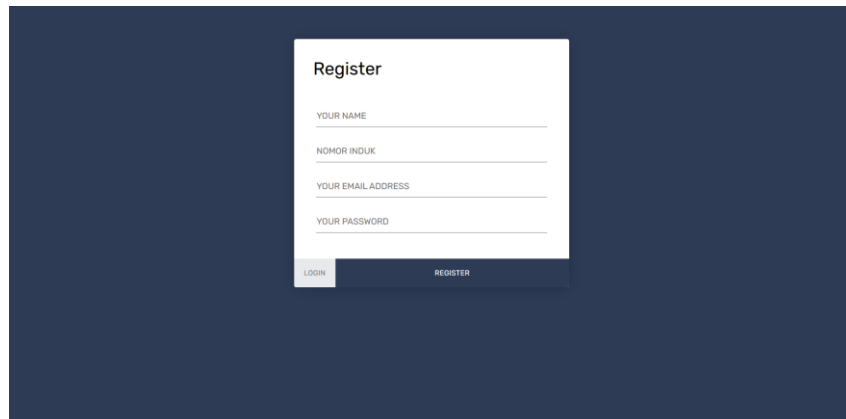
Gambar 4.4 Halaman *login*

Halaman *login website* untuk *user* yang sudah memiliki akun dapat langsung mengakses sistem dengan mengisi *email* dan *password* pada form *login* yang disediakan. Setelah informasi yang benar dimasukkan, *user* akan langsung diarahkan ke halaman utama *website* untuk menggunakan layanan yang tersedia. Halaman *login* ini memudahkan *user* yang sudah terdaftar untuk masuk ke dalam sistem dengan cepat dan efisien.

```
$validator = Validator::make($request->all(), [  
    'name'=>'required',  
    'nomor_induk' => 'required|min:8|unique:users,nomor_induk',  
    'email'=>'required|email',  
    'password'=>'required',  
]);  
  
if($validator->fails()){  
    return response()->json([  
        'succes'=>false,  
        'message'=>'Validasi Gagal',  
        'data'=>$validator->errors()  
    ]);  
}
```

Kode di atas merupakan bagian dari proses autentikasi, tepatnya saat *user* melakukan pendaftaran atau *login*. Validasi ini memastikan bahwa data yang dikirim *user* sesuai dengan aturan yang ditentukan sebelum diproses lebih lanjut.

b) Halaman registrasi

The image shows a web registration form titled "Register" centered on a dark blue background. The form is a white rectangle containing four text input fields, each with a label above it: "YOUR NAME", "NOMOR INDUK", "YOUR EMAIL ADDRESS", and "YOUR PASSWORD". Below these fields are two buttons: a light gray "LOGIN" button on the left and a dark blue "REGISTER" button on the right.

Gambar 4.5 Halaman registrasi

Halaman registrasi ini memungkinkan *user* yang belum memiliki akun untuk mendaftarkan diri dengan mengisi formulir yang tersedia. Formulir tersebut mencakup data seperti nama lengkap, *email*, dan *password* yang akan digunakan untuk *login* ke dalam sistem. Setelah semua informasi diisi dengan benar dan proses pendaftaran selesai, *user* dapat langsung menggunakan akun tersebut untuk mengakses *website* dan fitur-fitur yang disediakan. Halaman ini dirancang agar proses pembuatan akun menjadi lebih cepat dan mudah dipahami.

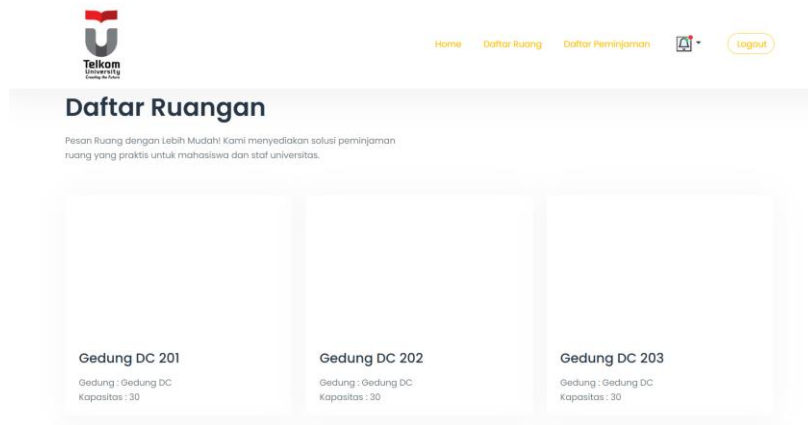
c) Halaman beranda atau *dashboard user*



Gambar 4.6 Halaman *dashboard user*

*Dashboard user* merupakan halaman utama setelah *user* berhasil *login*, yang menampilkan beberapa fitur navigasi penting seperti *home*, daftar ruangan, daftar peminjaman, notifikasi peminjaman, tombol pinjam, dan *logout*. Melalui fitur *home* atau beranda, *user* dapat melihat ringkasan atau informasi umum terkait sistem. Fitur daftar ruangan digunakan untuk melihat seluruh ruangan yang dapat dipinjam, daftar peminjaman untuk menampilkan riwayat serta status peminjaman yang pernah dilakukan, dan notifikasi untuk memberitahu mengenai konfirmasi dari *admin* terkait persetujuan peminjaman ruangan. Tombol pinjam yang akan mengarahkan *user* menuju menu daftar ruangan dan fitur *logout* digunakan untuk keluar dari akun *user* dan mengakhiri sesi penggunaan sistem.

d) Halaman daftar ruangan *user*



Gambar 4.7 Halaman daftar ruangan *user*

Halaman daftar ruangan menampilkan seluruh ruangan yang tersedia untuk dipinjam oleh *user*. Setiap ruangan ditampilkan dengan informasi yang jelas, seperti nama ruangan, lokasi, kapasitas, serta deskripsi singkat mengenai ruangan tersebut. Tampilan pada fitur ini dirancang agar mudah dibaca dan menarik secara visual, sehingga *user* dapat dengan cepat memahami detail ruangan sebelum melakukan peminjaman. *User* juga dapat langsung memilih ruangan yang diinginkan untuk melihat informasi lebih lanjut atau melanjutkan ke proses peminjaman. Fitur ini mempermudah *user* dalam mengeksplorasi pilihan ruangan yang sesuai dengan kebutuhan.

e) Formulir peminjaman ruangan

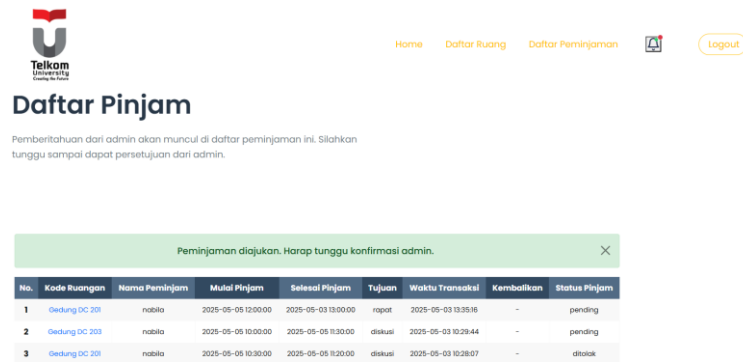
Gambar 4.8 Formulir peminjaman ruangan

Halaman ini memuat *user* yang telah memilih ruangan yang ingin dipinjam dan akan diarahkan untuk mengisi formulir peminjaman ruangan yang tersedia. Formulir ini mencakup informasi penting seperti tujuan peminjaman, waktu mulai, dan waktu selesai peminjaman. Pengisian data dilakukan secara lengkap dan sesuai agar permohonan dapat diproses dengan baik. Jika seluruh informasi telah diisi, *user* perlu menekan tombol pinjam ruang untuk mengirim formulir tersebut kepada *admin*. Permohonan yang diajukan kemudian akan ditinjau oleh *admin* untuk menentukan apakah peminjaman dapat disetujui atau ditolak, bergantung pada ketersediaan ruangan. Berikut merupakan kode program mengenai validasi formulir peminjaman:

```
$validatedData = $request->validate([
    'room_id' => 'required',
    'user_id' => 'required',
    'time_start_use' => 'required|date',
    'time_end_use' => 'required|date',
    'purpose' => 'required|max:250',
]);
```

Validasi data formulir dilakukan untuk memastikan bahwa input yang diterima sesuai dengan aturan yang ditentukan, seperti wajib diisi atau dalam format tanggal, sehingga mencegah kesalahan saat penyimpanan data ke dalam basis data.

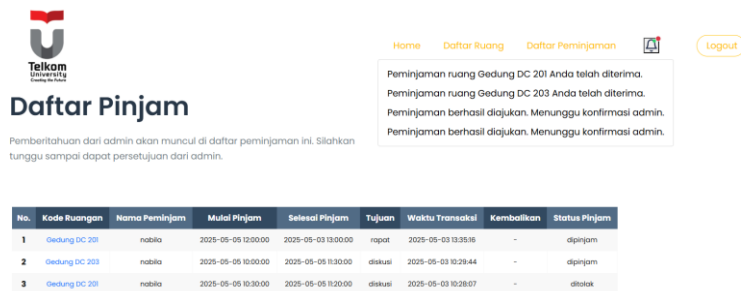
f) Daftar peminjaman *user*



No.	Kode Ruangan	Nama Peminjam	Mulai Pinjam	Selesai Pinjam	Tujuan	Waktu Transaksi	Kembalikan	Status Pinjam
1	Gedung DC 201	nabila	2025-05-05 12:00:00	2025-05-03 13:00:00	rapat	2025-05-03 13:35:16	-	pending
2	Gedung DC 203	nabila	2025-05-05 10:00:00	2025-05-05 11:30:00	diskusi	2025-05-03 10:29:44	-	pending
3	Gedung DC 201	nabila	2025-05-05 10:30:00	2025-05-05 11:20:00	diskusi	2025-05-03 10:28:07	-	ditolak

Gambar 4.9 Halaman daftar peminjaman *user*

Halaman ini menampilkan seluruh riwayat peminjaman ruangan yang telah dilakukan oleh *user*. *User* dapat melihat data peminjaman secara lengkap, seperti nomor, kode ruangan, nama peminjam, tanggal dan waktu peminjaman, tujuan peminjaman, serta status peminjaman apakah masih menunggu konfirmasi, disetujui, atau ditolak oleh *admin*. Tabel yang digunakan disusun secara rapi agar mudah dibaca dan dipahami, dengan tampilan yang lebih modern. Melalui halaman ini, *user* dapat memantau seluruh aktivitas peminjaman yang pernah dilakukan, serta memastikan apakah permohonan sudah diproses oleh *admin* atau belum.



No.	Kode Ruangan	Nama Peminjam	Mulai Pinjam	Selesai Pinjam	Tujuan	Waktu Transaksi	Kembalikan	Status Pinjam
1	Gedung DC 201	nabila	2025-05-05 12:00:00	2025-05-03 13:00:00	rapat	2025-05-03 13:35:16	-	dipinjam
2	Gedung DC 203	nabila	2025-05-05 10:00:00	2025-05-05 11:30:00	diskusi	2025-05-03 10:29:44	-	dipinjam
3	Gedung DC 201	nabila	2025-05-05 10:30:00	2025-05-05 11:20:00	diskusi	2025-05-03 10:28:07	-	ditolak

Gambar 4.10 Halaman daftar peminjaman *user* setelah dikonfirmasi *admin*

Konfirmasi dari *admin* terhadap peminjaman ruangan akan langsung ditampilkan pada halaman daftar peminjaman milik *user*. Status peminjaman akan diperbarui menjadi disetujui atau ditolak. Tampilan yang terstruktur dan mudah dibaca, halaman ini membantu pengguna memantau

proses peminjaman secara *real-time* dan mengetahui hasil keputusan admin tanpa harus melakukan pengecekan manual.

g) Halaman daftar *admin*



Gambar 4.11 Halaman daftar *admin*

Halaman daftar *admin* ini memungkinkan *admin* untuk menambahkan, mengedit, maupun menghapus *admin*. *Admin* dapat menambahkan *admin* baru ke dalam daftar *admin* dengan mengisi formulir pada tombol tambah *admin* baru. Fitur ini mempermudah pengelolaan data *admin* yang terlibat dalam sistem, memberikan kontrol penuh kepada *admin* dalam mengatur siapa saja yang dapat mengakses dan mengelola *website*.

h) Halaman daftar mahasiswa



Gambar 4.12 Halaman daftar mahasiswa

Halaman daftar mahasiswa ini memungkinkan *admin* untuk menambahkan, mengedit, maupun menghapus data mahasiswa yang terdaftar. *Admin* dapat menambahkan mahasiswa baru ke dalam daftar dengan mengisi formulir yang tersedia pada tombol tambah mahasiswa baru. Fitur ini mempermudah pengelolaan data mahasiswa yang terlibat dalam sistem, memberikan kontrol penuh kepada *admin* untuk mengatur

siapa saja yang terdaftar dan memastikan data mahasiswa terkelola dengan baik.

i) Halaman daftar ruangan *admin*



No.	Kode Ruangan	Nama Ruangan	Kapasitas	Action
1	Gedung DC 201	Gedung DC 201	30 Kursi	
2	Gedung DC 202	Gedung DC 202	30 Kursi	
3	Gedung DC 203	Gedung DC 203	30 Kursi	
4	Gedung DC 301	Gedung DC 301	30 Kursi	
5	Gedung DC 302	Gedung DC 302	30 Kursi	
6	Gedung DSP 301	Gedung DSP 301	30 Kursi	
7	Gedung DSP 302	Gedung DSP 302	30 Kursi	
8	Gedung DSP 303	Gedung DSP 303	30 Kursi	
9	Gedung DSP 304	Gedung DSP 304	30 Kursi	
10	Gedung DSP 305	Gedung DSP 305	30 Kursi	

Gambar 4.13 Halaman daftar ruangan *admin*

Halaman daftar ruangan ini memungkinkan *admin* untuk mengedit maupun menghapus ruangan yang terdaftar. Fitur ini mempermudah pengelolaan data ruangan yang tersedia, memberikan kontrol penuh kepada *admin* untuk mengatur dan memperbarui informasi mengenai ruangan yang dapat dipinjam oleh *user*. *Admin* dapat memastikan data ruangan selalu *ter-update* dan siap digunakan.

j) Daftar peminjaman sementara *admin*



No.	Nama Ruangan	Nama Peminjam	Mulai Pinjam	Selesai Pinjam	Tujuan	Mulai Transaksi	Status Pinjam	Action
1	Gedung DC 201	nabila	2025-05-05 12:00:00	2025-05-03 13:00:00	rapat	2025-05-03 13:35:16	pending	

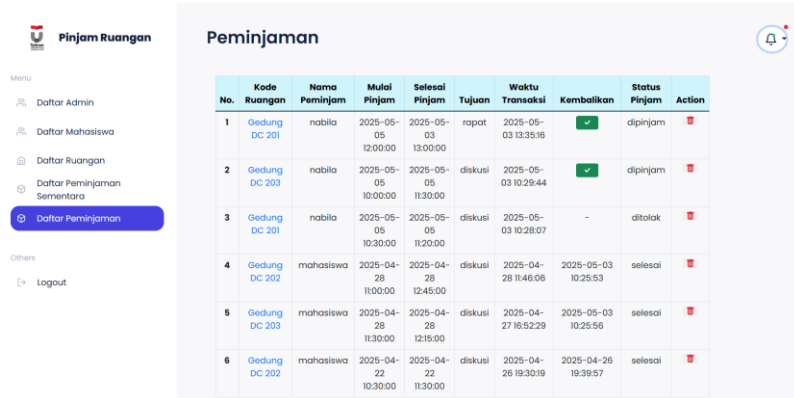
Gambar 4.14 Halaman daftar peminjaman sementara *admin*

Halaman ini mencakup peminjaman sementara jika ada *user* yang mengajukan peminjaman. *Admin* akan mengecek terlebih dahulu mengenai ketersediaan ruangan dan jika memang kosong maka *admin* akan menyetujui peminjaman, namun jika ruangan sedang dipakai atau sedang



tidak dapat dipinjam, maka permintaan peminjaman ruangan akan ditolak oleh *admin*.

k) Riwayat atau daftar peminjaman keseluruhan pada *admin*



No.	Kode Ruangan	Nama Peminjam	Mulai Pinjam	Selesai Pinjam	Tujuan	Waktu Transaksi	Kembalikan	Status Pinjam	Action
1	Gedung DC 201	nabila	2025-05-05 12:00:00	2025-05-03 13:00:00	rapat	2025-05-03 13:35:16	<input checked="" type="checkbox"/>	dipinjam	
2	Gedung DC 203	nabila	2025-05-05 10:00:00	2025-05-05 11:30:00	diskusi	2025-05-03 10:29:44	<input checked="" type="checkbox"/>	dipinjam	
3	Gedung DC 201	nabila	2025-05-05 10:30:00	2025-05-05 11:20:00	diskusi	2025-05-03 10:28:07	-	ditolak	
4	Gedung DC 202	mahasiswa	2025-04-28 11:00:00	2025-04-28 12:45:00	diskusi	2025-04-28 11:46:06	2025-05-03 10:25:53	selesai	
5	Gedung DC 203	mahasiswa	2025-04-28 11:30:00	2025-04-28 12:15:00	diskusi	2025-04-27 16:52:29	2025-05-03 10:25:56	selesai	
6	Gedung DC 202	mahasiswa	2025-04-22 10:30:00	2025-04-22 11:30:00	diskusi	2025-04-26 19:30:19	2025-04-26 19:39:57	selesai	

Gambar 4.15 Halaman riwayat atau daftar peminjaman *admin*

Halaman ini memuat seluruh daftar peminjaman atau riwayat peminjaman ruangan yang telah dilakukan oleh *user*, memberikan gambaran lengkap mengenai status peminjaman yang sedang berlangsung maupun yang telah selesai. *Admin* dapat memantau setiap peminjaman ruangan dengan mudah dan melihat detail terkait waktu peminjaman serta tujuan penggunaan ruangan. Jika *admin* merasa bahwa *user* telah selesai menggunakan ruangnya, maka *admin* dapat menandai peminjaman tersebut dengan status 'kembalikan', menandakan bahwa ruangan tersebut sudah tersedia untuk dipinjam kembali oleh *user* lainnya. *Admin* juga memiliki kemampuan untuk menghapus daftar peminjaman yang sudah tidak relevan atau diperlukan lagi, menjaga sistem tetap rapi dan teratur. Fitur ini memberikan kontrol penuh kepada *admin* dalam mengelola dan memastikan bahwa proses peminjaman ruangan berjalan dengan efisien dan terkelola dengan baik.

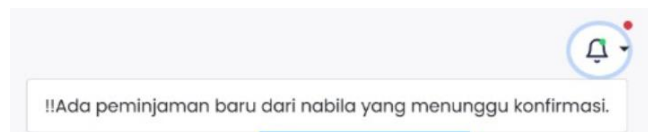
l) Notifikasi *user* saat mengajukan peminjaman kepada *admin*



Gambar 4.16 Notifikasi *user*

Fitur notifikasi untuk *user* akan muncul setiap kali terdapat pembaruan terkait status peminjaman. Setelah *user* mengajukan permohonan peminjaman ruangan, *user* akan menerima notifikasi mengenai konfirmasi peminjaman, apakah permohonan disetujui atau ditolak oleh *admin*. Notifikasi ini membantu *user* untuk mendapatkan informasi terkini mengenai peminjaman, memastikan *user* tahu mengenai status pengajuan tanpa harus memeriksa secara manual, serta meningkatkan pengalaman *user* dalam mengelola peminjaman ruangan.

m) Notifikasi *admin* saat menerimaajuan peminjaman dari *user*



Gambar 4.17 Notifikasi *admin*

Fitur notifikasi untuk *admin* akan muncul setiap kali ada pengajuan peminjaman baru dari *user*. *Admin* akan menerima pemberitahuan terkait permohonan peminjaman ruangan yang mencakup informasi penting seperti nama ruangan, tanggal dan waktu peminjaman, serta tujuan penggunaan. Setelah *admin* meninjau permohonan, notifikasi juga akan dikirimkan ke *user* untuk mengonfirmasi bahwa status peminjaman telah diperbarui, apakah disetujui atau ditolak. Fitur notifikasi ini mempermudah *admin* untuk memantau setiap permohonan peminjaman secara *real-time*, memastikan tidak ada pengajuan yang terlewat, serta meningkatkan efisiensi dalam pengelolaan ruangan.

#### 4.2. Analisis

Setelah tahap desain dan pengembangan, tahap pengujian dilaksanakan untuk menjamin bahwa sistem berfungsi sebagaimana mestinya. Teknik pengujian yang digunakan adalah *blackbox testing*, yaitu metode evaluasi fungsionalitas sistem tanpa memperhatikan struktur pengujian dari sudut pandang *user*, dan tanpa perlu mengetahui struktur kode perangkat lunak. Pada pengujian ini, diambil 20 responden mahasiswa sebagai *user* untuk menguji *website* peminjaman ruangan. Berikut ini berisi hasil dari *blackbox testing user*:

Tabel 4.1 *Testcase user*

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba <i>login</i> dengan <i>email</i> dan <i>password</i> yang valid	<i>User</i> berhasil <i>login</i> dan sistem menampilkan <i>dashboard user</i> .	✓	-
2.	Mencoba <i>login</i> dengan <i>email</i> dan <i>password</i> yang salah	Muncul pesan <i>error</i> “Masukkan <i>Email &amp; Password</i> Dengan Benar”.	✓	-
3.	Mencoba <i>login</i> tanpa mengisi <i>email</i> atau <i>password</i>	Muncul pesan <i>error</i> “Harap isi bidang ini”.	✓	-
4.	Mencoba <i>register</i> dengan mengisi nama, nomor induk, <i>email</i> , dan <i>password</i>	<i>User</i> berhasil melakukan <i>register</i> dan diarahkan ke halaman <i>login</i> dengan menampilkan pesan “Berhasil <i>Register</i> , Silahkan <i>Login</i> ”.	✓	-

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
5.	Mencoba mengakses <i>dashboard user</i>	<i>User</i> berhasil mengakses menu <i>dashboard</i> atau <i>home</i>	✓	-
6.	Mencoba mengakses menu daftar ruang	Mengakses menu daftar ruangan dan mengisi formulir peminjaman jika ingin mengajukan peminjaman ruangan.	✓	-
7.	Mencoba melihat daftar peminjaman ruangan khusus <i>user</i>	<i>User</i> dapat melihat daftar peminjaman yang telah dilakukan, serta status peminjamannya. Selain itu, daftar peminjaman ini akan menjadi riwayat dari peminjaman ruangan yang dilakukan oleh <i>user</i> .	✓	-
8.	Mencoba <i>logout</i>	<i>User</i> berhasil keluar dari <i>website</i> .	✓	-

Setelah dilakukan pengujian terhadap *user*, maka pengujian terhadap *admin* dilakukan. Pengujian untuk *admin* hanya dilakukan oleh satu orang karena dalam praktiknya, sistem ini memang hanya digunakan oleh satu staf akademik yang memiliki wewenang sebagai *admin*. Meskipun jumlahnya terbatas, pengujian tetap valid karena dilakukan oleh *admin* yang benar-benar menjalankan peran tersebut

dalam operasional harian. Berdasarkan dari hasil pengujian oleh *admin* didapatkan hasil seperti pada tabel dibawah ini:

Tabel 4.2 *Testcase Login Admin*

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba <i>login</i> dengan <i>email</i> dan <i>password</i> yang valid	<i>Admin</i> berhasil <i>login</i> dan sistem menampilkan halaman <i>admin</i> .	✓	-
2.	Mencoba <i>login</i> dengan <i>email</i> dan <i>password</i> yang salah	Muncul pesan <i>error</i> “Masukkan <i>Email &amp; Password</i> Dengan Benar”	✓	-
3.	Mencoba <i>login</i> tanpa mengisi <i>email</i> atau <i>password</i>	Muncul pesan <i>error</i> “Harap isi bidang ini”	✓	-

Tabel 4.3 *Testcase Edit Admin*

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba mengedit <i>admin</i> pada daftar <i>admin</i>	<i>Admin</i> berhasil mengedit form daftar <i>admin</i> yang berisi nama lengkap, nomor induk, dan <i>email</i>	✓	-
2.	Mencoba menambahkan data <i>admin</i> baru	<i>Admin</i> mengisi form yang berisi nama lengkap, nomor induk, <i>email</i> , dan <i>password</i>	✓	-

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
		setelah itu akan muncul pesan “Data <i>admin</i> berhasil ditambahkan”		

Tabel 4.4 *Testcase* Daftar Ruangan *Admin*

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba mengedit ruangan pada daftar ruangan.	<i>Admin</i> dapat mengelola data ruangan dengan menghapus atau mengedit ruangan.	✓	-

Tabel 4.5 *Testcase* Daftar Peminjaman *Admin*

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba menyetujui atau menolak permintaan peminjaman ruangan.	<i>Admin</i> dapat menyetujui atau menolak permintaan peminjaman ruangan yang dilakukan oleh <i>user</i> .	✓	-
2.	Mencoba menandai ruangan yang telah selesai digunakan dengan tanda ‘kembalikan’.	<i>Admin</i> dapat menandai pada daftar peminjaman dengan status ‘kembalikan’ saat waktu peminjaman ruangan	✓	-

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
		yang seharusnya telah selesai digunakan.		
3.	Mencoba menghapus <i>user</i> dan <i>admin</i> dari daftar peminjaman.	<i>Admin</i> dapat menghapus <i>user</i> maupun <i>admin</i> yang ada pada daftar peminjaman.	✓	-

Hasil pengujian pada *blackbox* yang melibatkan 20 mahasiswa sebagai *user* dengan masing-masing *user* melakukan 8 pola pengujian dan 1 *admin* yang melakukan 9 pola pengujian, didapatkan hasil perhitungan *blackbox testing* seperti pada Tabel 4.6 dibawah ini:

Tabel 4.6 Rekapitulasi hasil pengujian *blackbox*

Role	Jumlah responden	Pola pengujian	Total uji coba	Berhasil	Tidak berhasil
Mahasiswa ( <i>user</i> )	20	8	160	154	6
Staf akademik ( <i>admin</i> )	1	9	9	9	0
<b>Total</b>	<b>21</b>	<b>17</b>	<b>169</b>	<b>163</b>	<b>6</b>

Dari rekapitulasi tersebut, lalu dihitung menggunakan teknik analisis deskriptif, yaitu:

$$\text{Presentase kelayakan} = \frac{\text{Total uji coba berhasil}}{\text{Total uji coba}} \times 100\%$$

$$\text{Presentase kelayakan} = \frac{163}{169} \times 100\% = 96,45\%$$

Berdasarkan hasil perhitungan *blackbox testing* diatas didapatkan total presentase keberhasilan fungsi sebesar 96,45%.



## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan hasil perancangan sistem, telah berhasil dikembangkan sebuah *website* peminjaman ruangan yang dirancang untuk meminjam ruangan yang ada di Telkom University Purwokerto. *User* dapat melakukan proses peminjaman secara *online* dan melihat informasi ketersediaan ruangan secara *real-time* melalui sistem ini. Sistem ini juga membantu pengelola mengatur jadwal penggunaan ruang secara lebih terorganisir.

*Website* ini telah berhasil menerapkan *framework* Laravel dan metode *prototype* secara optimal. Sementara Laravel memudahkan pengembangan sistem berbasis arsitektur MVC serta mendukung keamanan dan skalabilitas sistem, metode *prototype* memungkinkan pengembangan berulang dengan melibatkan *user* melalui pengujian awal, yang memungkinkan sistem disesuaikan dengan kebutuhan nyata. Hasilnya adalah *website* peminjaman ruangan di Telkom University Purwokerto yang memiliki fitur untuk melihat daftar ruangan yang tersedia, proses peminjaman ruangan, serta riwayat dan daftar peminjam ruangan. Berdasarkan hasil pengujian *blackbox testing* didapatkan total presentase keberhasilan fungsi sebesar 96,45%.

#### **5.2. Saran**

Berdasarkan hasil pada rancang bangun *website* peminjaman ruangan di Telkom University Purwokerto menunjukkan adanya beberapa kekurangan. Saran-saran berikut disampaikan sebagai masukan untuk pengembangan sistem di masa mendatang agar dapat menjadi lebih optimal dan sesuai dengan kebutuhan *user*:

- a. Pengembangan ke depan dapat mencakup integrasi dengan sistem akademik atau kalender kampus, agar data jadwal ruangan yang sudah terdaftar dapat terhubung otomatis dan meminimalkan risiko bentrok penggunaan ruangan.
- b. Penambahan *dashboard* statistik *user* ruangan agar *admin* dapat melihat data statistik penggunaan ruangan, seperti frekuensi

peminjaman tiap ruangan, pengguna paling aktif, dan waktu paling sibuk, dalam bentuk grafik yang informatif.

## DAFTAR PUSTAKA

- [1] A. S. Putro, F. Ismuharyanti, and A. M. Saktiwi, "Analisis pemanfaatan sistem peminjaman ruangan berbasis *web* di *Resource Centre* Universitas Amikom Yogyakarta," *Pustaka Karya: Jurnal Ilmiah Ilmu Perpustakaan Dan Informasi*, vol. 10, no. 2, pp. 104, December 2022.
- [2] A. Apriliando, "Implementasi *Framework* Laravel pada Rancang Bangun *Website* IAKN Palangka Raya dengan Metode *Prototype*," *Jurnal Sains Komputer Dan Teknologi Informasi*, vol. 3, no. 2, pp. 87-96, May 2021.
- [3] N. A. D. Dhanardana, and T. Rahayu, "Aplikasi Peminjaman Ruangan Berbasis *Web* Pada Direktorat Jenderal Hak Asasi Manusia," *Seminar Nasional Mahasiswa Bidang Ilmu Komputer Dan Aplikasi (SENAMIKA)*, vol. 1, no. 2, pp. 211–221, August 2020.
- [4] A. Nurhidayah, and S. Kosasi, "Perancangan Perangkat Lunak Penjualan Berbasis *Website* Dengan *Framework* Laravel Pada Emiracase," *E-Jurnal JUSITI (Jurnal Sistem Informasi Dan Teknologi Informasi)*, vol. 11, no. 1, pp. 24–35, April 2022.
- [5] N. Nabilla, and A. Ichwani, "Sistem Informasi Layanan E-Konseling Psikologi Untuk Mahasiswa Berbasis *Website* Dengan Metode *Prototype*," *Jurnal Mnemonic*, vol. 5, no. 2, pp. 191–198, September 2022.
- [6] N. L. G. P. Suwirmayanti, I. K. A. A. Aryanto, I. G. A. N. W. Putra, N. K. Sukerti, and R. Hadi, "Penerapan *Helpdesk System* dengan Pengujian *Blackbox Testing*," *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, vol. 2, no. 2, pp. 55-64, November 2020.
- [7] T. Wijaya, and R. Somya, "Perancangan dan Implementasi Aplikasi Peminjaman Ruangan Kedinasan Kota Salatiga menggunakan *Framework* Laravel," *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, vol. 9, no. 4, pp. 3095–3107, December 2022.
- [8] F. F. Surenggana, F. Bimantoro, and R. Biasrori, "Sistem Informasi Manajemen Ruangan Program Studi Teknik Informatika Menggunakan *Framework* Laravel," *Jurnal Begawe Teknologi Informasi*, vol. 2, no. 2, pp.

203–213, September 2021.

- [9] A., N., D. Meidelfi, D., and T. Lestari, “Aplikasi Peminjaman Laboratorium Pada Fakultas Teknologi Pertanian Universitas Andalas,” *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 2, no. 2, pp. 42–47, June 2021.
- [10] A. D. Herlambang, A. Rachmadi, A. P. Rahmatika, D. I. D. Utami, and S. W. Hapsari, “V- Model untuk Pengembangan Sistem Informasi Manajemen Ruang Rapat,” *Jurnal Teknologi Informasi Dan Ilmu Komputer*, vol. 7, no. 2, pp. 313, April 2020.
- [11] J. R. Naldo, P. A. Hernando, and A. B. Pangaribuan, “Pembuatan Website Admininstrator Peminjaman Ruangan Laboratorium Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta”, *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA)*, vol. 2, no. 1, pp. 306–314, April 2021.
- [12] K. D. Sukmana, K. Q. Fredlina, and P. T. H. Permana, “Model Aplikasi Peminjaman Ruangan Berbasis Web Pada Tingkat Fakultas di Perguruan Tinggi,” *Jutisi : Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, vol. 11, no. 2, pp. 393, August 2022.
- [13] D. A. Kurniawan, and S. D. Asri, “Aplikasi Peminjaman Ruangan dan Gedung Pada Universitas Mercu Buana Kampus D Jatisampurna Berbasis Web,” *Jurnal Ilmu Teknik Dan Komputer*, vol. 3, no. 2, pp. 128, July 2020.
- [14] D. Oktaviana, and M. A. Hamid, “Room Loan Information System Website Based Using PHP And MYSQL With Waterfall Model,” *Journal of Vocational Education Studies*, vol. 4, no. 2, pp. 210–218, November 2021.
- [15] U. R. Wahyuni, F. Fauziah, and E. Mardiani, “Web-Based Room Loan Management Services Using Web Based RAD Method: Web-Based Room Loan Management Services Using Web Based RAD Method”, *Mantik*, vol. 4, no. 1, pp. 452-459, May 2020.
- [16] N. A. Putri, and Amaliyah, “Rancang Bangun Sistem Reservasi Ruang Rapat Berbasis Website,” *IKRAITH-Teknologi*, vol. 7, no. 2, pp. 22–34, July 2023.

- [17] A. O. Sari, A. Abdilah, and Sunarti, *Web Programming*, edisi ke-1, Yogyakarta: Graha Ilmu, 2020.
- [18] N. L. A. M. R. Dewi, R. S. Hartati, and Y. Divayana, “Penerapan Metode *Prototype* dalam Perancangan Sistem Informasi Penerimaan Karyawan Berbasis *Website* pada Berlian Agency,” *Majalah Ilmiah Teknologi Elektro*, vol. 20, no. 1, pp. 147, January-June 2021.
- [19] Arizal, A. N. Puteri, F. Zakiyabarsi, and D. F. Priambodo, “Metode *Prototype* pada Sistem Informasi Manajemen Tugas Akhir Mahasiswa Berbasis *Website*,” *Jurnal Teknologi Informasi Dan Komunikasi (TIKomSiN)*, vol. 10, no. 1, April 2022.
- [20] A. Yunisa, and R. Amalia, “Perancangan Sistem Informasi Manajemen Jasa Fotografi Berbasis *Website* Menggunakan *Framework* Laravel (Studi Kasus: Haydey Moment)”, *Jurnal Informatika MULTI*, vol. 1, no. 1, pp. 25–36, January 2023.
- [21] D. P. Sari, and R. Wijanarko, “Implementasi *Framework* Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus di Rumah Kamera Semarang),” *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, vol. 2, no. 1, pp. 32, March 2020.
- [22] M. Syarif, and E. B. Pratama, “Analisis Metode Pengujian Perangkat Lunak *Blackbox Testing* Dan Pemodelan Diagram Uml Pada Aplikasi Veterinary Services Yang Dikembangkan Dengan Model *Waterfall*,” *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 5, no. 2, pp. 253–258, July 2021.
- [23] I. Arwani, “Optimasi Proses Klasterisasi di *MySQL* DBMS dengan Mengintegrasikan Algoritme MIC-Kmeans Menggunakan Bahasa SQL dalam *Stored Procedure*”, *Jurnal Teknologi Informasi Dan Ilmu Komputer*, vol. 7, no.2, pp. 391–398, April 2020.
- [24] M. Ahmadar, P. Perwito, and C. Taufik, C, “Perancangan Sistem Informasi Penjualan Berbasis *Web* Pada Rahayu Photo Copy Dengan *Database MySQL*”, *Dharmakarya*, vol. 10, no. 4, pp. 284, December 2021.

- [25] M. Sumiati, R. Abdillah, and A. Cahyo, "Pemodelan UML Untuk Sistem Informasi Persewaan Alat Pesta", *Jurnal Fasikom*, vol. 11, no. 2, pp. 79–86, August 2021.
- [26] A. Voutama, "Sistem Antrian Cucian Mobil Berbasis *Website* Menggunakan Konsep CRM dan Penerapan UML", *Komputika : Jurnal Sistem Komputer*, vol.11, no. 1, pp. 102–111, April 2022.
- [27] Suharni, E. Susilowati, and F. Pakusadewa, "Perancangan *Website* Rumah Makan Ninik Sebagai Media Promosi Menggunakan Unified Modelling Language," *Rekayasa Informasi*, vol. 12, no. 1, pp. 1–12, April 2023.
- [28] V. E. Pratama, S. Balqis, Z. P. Drajatnita, and M. I. Choldun, "Perancangan Sistem Informasi Peminjaman Ruangan Berbasis *Web* (Studi Kasus: BAAK Universitas Logistik dan Bisnis Internasional)," *IMPROVE*, vol. 15, no. 2, pp. 46–49, Sep 2023.
- [29] S. Ditta and Y. Kurniawan, "*Software Requirement Specification* Sistem Informasi Terintegrasi Akademik SMA Katolik Santo Thomas Aquino," *Jurnal Basicedu*, vol. 6, no. 5, pp. 7738–7748, Nov. 2022.

# LAMPIRAN

## Lampiran 1. Surat Permohonan Pengambilan Data



Nomor : 069/AKD09/KP-WD1/2024  
Lampiran : -  
Perihal : Permohonan Penelitian Tugas Akhir

Kepada Yth,  
Kepala Bagian Pelayanan Akademik Pusat Telkom University Purwokerto  
Telkom University Purwokerto  
Jl. DI Panjaitan No.128, Karangreja, Purwokerto Kidul, Kec. Purwokerto Sel., Kabupaten  
Banyumas, Jawa Tengah 53147

Dengan Hormat,

Kami informasikan bahwa mahasiswa kami atas nama:

NO	NIM	NAMA	PRODI	FAKULTAS
1	21102247	Nabila Afifatul Fajri	S1 Teknik Informatika - Kampus Purwokerto	Fakultas Direktorat Kampus Purwokerto

Dengan topik:

RANCANG BANGUN WEBSITE PEMINJAMAN RUANGAN DENGAN FRAMEWORK LARAVEL DI KAMPUS INSTITUT TEKNOLOGI TELKOM PURWOKERTO

Bermaksud melakukan pengambilan data untuk **Tugas Akhir** di perusahaan/lembaga yang bapak/ibu pimpin terkait dengan **penelitian** sesuai topik yang sedang dilakukan.

Oleh karena itu, kami mohon bapak/ibu berkenan memberikan izin kepada yang bersangkutan.

Demikian surat permohonan yang kami sampaikan. Atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Bandung, 01 November 2024  
Kepala Bagian Pelayanan Akademik Pusat Telkom University Purwokerto



Nia Annisa Ferani Tanjung, S.Si., M.Sc  
NIP. 20920034-1

Telkom University Learning Center Building - Bandung Technoplex | Jl. Telekomunikasi, Terusan Buah Batu, Bandung 40257, West Java, Indonesia  
t: 62-22-756 4108 f: 62-22 756 5200 e: info@telkomuniversity.ac.id

[www.telkomuniversity.ac.id](http://www.telkomuniversity.ac.id)

## Lampiran 2. Hasil data wawancara

### Hasil Wawancara

#### Pengumpulan Data

Nama Narasumber : Rizpan Ilham

Tanggal wawancara : 30 April 2025

Lokasi wawancara : Ruang Akademik Telkom University Purwokerto (REK-401)

Jabatan : Kepala Urusan Pelayanan Akademik Telkom University Purwokerto

Pertanyaan	Jawaban
1. Bagaimana proses peminjaman ruangan selama ini?	Dari mahasiswa harus membuat surat pengajuan terlebih dahulu yang nantinya setelah mendapatkan izin dari akademik, kita cek ketersediaan ruangan terlebih dahulu. Kemudian saat kita sudah memberi izin, maka lanjut ke logistic untuk izin peminjaman ruangnya.
2. Apakah website peminjaman ruangan diperlukan untuk menghemat waktu dan mempermudah untuk mengelola daftar peminjaman ruangan?	Sekarang memang belum ada, tapi kalau nantinya akan dibuat itu akan menghemat waktu dan meringkas proses bisnis yang ada.
3. Fitur apa saja yang dibutuhkan untuk pembuatan website ini?	Kalau dari sisi kebutuhan kita sebagai penyedia ruangan selain akses dari admin, adanya fitur pengecekan ruangan secara otomatis. Namun, jika memang masih belum ada setidaknya dapat meringkas atau mempermudah dari yang sudah ada.
4. Proses peminjaman ruangan biasanya memakan waktu berapa lama?	Peminjaman ruangan sih gak lama ya, sekitar 10 sampai 15 menit bisa lebih cepat tergantung ketersediaan ruangan. Tapi, kalo semisal ada ruangan yang tidak tersedia itu yang agak sulit sih, karena kita harus cari ruangan lain yang tersedia.
5. Dengan adanya website ini, apakah sistem peminjaman ruangan menjadi lebih efektif?	Jika sistem nantinya sudah mengakomodasi sesuai kebutuhan sih, harusnya dapat mempermudah proses peminjaman ruangan.

Purwokerto, 30 April 2025

Kepala Urusan Pelayanan Akademik Telkom University Purwokerto

  
Rizpan Ilham



### Lampiran 3. Dokumentasi wawancara





Lampiran 5. Pengujian *Blackbox*



Lampiran 6. Validator Uji *Blackbox*

Testcase Login Admin

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba login dengan email dan password yang valid	Admin berhasil login dan sistem menampilkan halaman admin.	✓	
2.	Mencoba login dengan email dan password yang salah	Muncul pesan error "Masukkan Email & Password Dengan Benar"	✓	
3.	Mencoba login tanpa mengisi email atau password	Muncul pesan error "Harap isi bidang ini"	✓	

### Testcase Edit Admin

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba mengedit admin pada daftar admin	Admin berhasil mengedit form daftar admin yang berisi nama lengkap, nomor induk, dan email	✓	
2.	Mencoba menambahkan data admin baru	Admin mengisi form yang berisi nama lengkap, nomor induk, email, dan password setelah itu akan muncul pesan "Data admin berhasil ditambahkan"	✓	



Testcase Daftar Ruangan Admin

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba mengedit ruangan pada daftar ruangan.	Admin dapat mengelola data ruangan dengan menghapus atau mengedit ruangan.	✓	

Testcase Daftar Peminjaman Admin

No.	Pola pengujian	Hasil yang diharapkan	Berhasil	
			Ya	Tidak
1.	Mencoba menyetujui atau menolak permintaan peminjaman ruangan.	Admin dapat menyetujui atau menolak permintaan peminjaman ruangan yang dilakukan oleh user.	✓	
2.	Mencoba menandai ruangan yang telah selesai digunakan dengan tanda	Admin dapat menandai pada daftar peminjaman dengan status 'kembalikan' saat waktu	✓	

	'kembalikan'.	peminjaman ruangan yang seharusnya telah selesai digunakan.		
3.	Mencoba menghapus user dan admin dari daftar peminjaman.	Admin dapat menghapus user maupun admin yang ada pada daftar peminjaman.	✓	

Purwokerto, 13 Maret 2025

Student Staf Akademik Telkom University Purwokerto



Sa' Baniatun Astia