

**RANCANG BANGUN SISTEM PEMINJAMAN *COWORKING SPACE*
BERBASIS *WEBSITE* MENGGUNAKAN *MERN STACK TECHNOLOGY*
PADA GEDUNG LEMBAGA KEMAHASISWAAN
FISIP UNIVERSITAS LAMPUNG**

(Skripsi)

Oleh

YOVANTA ANJELINA

NPM 1915061046



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

**RANCANG BANGUN SISTEM PEMINJAMAN *COWORKING SPACE*
BERBASIS *WEBSITE* MENGGUNAKAN *MERN STACK TECHNOLOGY*
PADA GEDUNG LEMBAGA KEMAHASISWAAN
FISIP UNIVERSITAS LAMPUNG**

Oleh

YOVANTA ANJELINA

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

ABSTRAK

RANCANG BANGUN SISTEM PEMINJAMAN *COWORKING SPACE* BERBASIS *WEBSITE* MENGGUNAKAN *MERN STACK TECHNOLOGY* PADA GEDUNG LEMBAGA KEMAHASISWAAN FISIP UNIVERSITAS LAMPUNG

Oleh

YOVANTA ANJELINA

Coworking Space Gedung Lembaga Kemahasiswaan FISIP Universitas Lampung merupakan fasilitas yang menyediakan tempat bagi civitas akademik untuk melakukan berbagai aktivitas seperti bekerja, berdiskusi, dan berkumpul dalam satu lokasi. Fasilitas yang tersedia mencakup ruang rapat, tempat duduk, dan tempat penyimpanan. Saat ini, proses peminjaman ruangan masih dilakukan secara manual dengan melakukan pengecekan ketersediaan ruangan secara langsung dan mencatat data peminjaman. Proses tersebut dapat berpotensi adanya duplikasi data dan jadwal peminjaman yang tumpang tindih. Penelitian ini bertujuan untuk membangun sistem peminjaman *coworking space* dalam memudahkan pengguna mengecek ketersediaan ruangan dan peminjaman ruangan, serta memudahkan *admin* dalam mengelola data peminjaman ruangan secara digital. Pengembangan sistem ini menggunakan metode pengembangan *Personal Extreme Programming* (PXP) yang dilakukan sebanyak 3 kali iterasi berdurasi waktu 60 hari dan menghasilkan 3 sistem yaitu *backend server*, *web admin* dan *web customer*. Sistem ini dibangun menggunakan teknologi *MERN Stack* yang terdiri dari MongoDB, ExpressJS, ReactJS, dan NodeJS, sehingga pembangunan sistem menjadi lebih mudah dan interaktif. Sistem ini diuji menggunakan metode *Black-box testing* pada 15 fitur dengan 57 kasus uji, dan didapatkan hasil yang sesuai dengan *output* yang diinginkan. Kemudian hasil pengujian dari *User Experience Questionnaire* (UEQ) yang melibatkan 20 responden menyatakan bahwa dari 5 kategori yaitu *attractiveness*, *perspicuity*, *dependability*, *stimulation* dan *novelty* mendapatkan nilai *above average* atau diatas rata-rata dan kategori *efficiency* mendapatkan nilai *good* atau baik.

Kata kunci : Sistem peminjaman, *Coworking Space*, metode PXP, *MERN Stack*, UEQ.

ABSTRACT

DESIGN AND DEVELOPMENT A COWORKING SPACE BOOKING SYSTEM BASED ON MERN STACK TECHNOLOGY AT THE STUDENT ORGANIZATION BUILDING, FACULTY OF SOCIAL AND POLITICAL SCIENCES, UNIVERSITY OF LAMPUNG

By

YOVANTA ANJELINA

The Coworking Space of the Student Organization Building, Faculty of Social and Political Sciences, University of Lampung, is a facility that provides a space for academic community members to engage in various activities such as work, discussions, and gatherings in one location. The available facilities include meeting rooms, seating areas, and storage spaces. Currently, the room booking process is still carried out manually by directly checking the room availability and recording booking data. The process can potentially result in data duplication and overlapping booking schedules. This study aims to develop a coworking space booking system to facilitate users in checking room availability and making room reservations, as well as assisting administrators in managing booking data digitally. The development of this system uses the Personal Extreme Programming (PXP) development method, which involves three iterations lasting 60 days and results in three systems: the backend server, web admin, and web customer. The system is built using the MERN Stack technology, which consists of MongoDB, ExpressJS, ReactJS, and NodeJS, making the system development easier and more interactive. The system is tested using the Black-box testing method on 15 features with 57 test cases, and the results align with the desired output. Furthermore, the User Experience Questionnaire (UEQ) testing involving 20 respondents indicates that out of the five categories: attractiveness, perspicuity, dependability, stimulation, and novelty, all received above-average scores. The efficiency category received a good score.

Keywords: Booking system, Coworking Space, PXP Method, MERN Stack, UEQ.

Judul Skripsi : **RANCANG BANGUN SISTEM PEMINJAMAN
COWORKING SPACE BERBASIS WEBSITE
MENGUNAKAN MERN STACK TECHNOLOGY
PADA GEDUNG LEMBAGA KEMAHASISWAAN
FISIP UNIVERSITAS LAMPUNG**

Nama mahasiswa : **Yovanta Anjefina**

Nomor Pokok Mahasiswa : **1915061046**

Program Studi : **Teknik Informatika**

Fakultas : **Teknik**



1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping


Ir. Meizano Ardhi Muhammad, S.T., M.T.
NIP. 198105282012121001


Ir. Trisya Septiana, S.T., M.T., IPM.
NIP. 199009212019032025

2. Mengetahui

Ketua Jurusan Teknik Elektro

Ketua Program Studi
Teknik Informatika

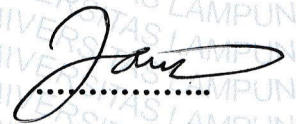

Herlinawati, S.T., M.T.
NIP. 197103141999032001


Mona Arif Muda, S.T., M.T.
NIP. 199009212019032025

MENGESAHKAN

1. Tim Penguji

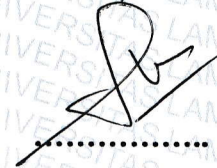
Ketua : **Ir. Meizano Ardhi Muhammad, S.T., M.T.**



Sekretaris : **Ir. Trisya Septiana, S.T., M.T., IPM.**



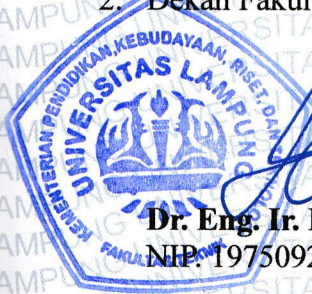
Penguji : **Wahyu Eko Sulistiono, S.T., M.Sc.**



2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP. 19750928 200112 1 002



Tanggal Lulus Ujian Skripsi : **01 Agustus 2023**

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi saya dengan judul “Rancang Bangun Sistem Peminjaman *Coworking Space* Berbasis *Website* Menggunakan *MERN Stack Technology* Pada Gedung Lembaga Kemahasiswaan FISIP Universitas Lampung” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 01 Agustus 2023
Pembuat pernyataan,




Yovanta Anjelina
NPM. 1915061046

RIWAYAT HIDUP



Penulis dilahirkan di Way Kanan, pada tanggal 10 Oktober 2001. Penulis merupakan anak kedua dari dua bersaudara dari pasangan Bapak Yakub Basuki dan Ibu Sumiarsih.

Penulis menyelesaikan pendidikannya di MI An-Nuur Negara Ratu pada tahun 2013, SMP Negeri 2 Sungkai Utara pada tahun 2016 dan SMA Negeri 3 Kotabumi pada tahun 2019. Pada tahun 2019 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan, antara lain:

1. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Pendidikan dan Pengembangan Diri, Divisi Kerohanian periode 2019/2020 dan anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Sosial dan Kewirausahaan, Divisi Sosial periode 2020/2021.
2. Menjadi asisten Laboratorium Teknik Komputer Universitas Lampung tahun 2021.
3. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil Kelas Pengembang *Front-End Web* dan *Back-End* di Dicoding Indonesia pada tahun 2021.
4. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil Kelas React JS *Front-End Engineer* di Alterra Academy pada tahun 2022.
5. Melaksanakan Kuliah Kerja Nyata pada bulan Januari sampai dengan Februari 2022 di Desa Bumi Mandiri, Kecamatan Abung Barat, Kabupaten Lampung Utara, Provinsi Lampung.
6. Melakukan Kerja Praktik di PT Gawai Mutu Digital pada bulan Agustus sampai September tahun 2022 dengan membuat Sistem Informasi Katalog Kupu-Kupu.

MOTTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya”

(Q.S Al-Baqarah : 286)

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah kamu berharap”

(Q.S Al-Insyirah : 6-8)

“Apapun yang menjadi takdirmu, akan mencari jalannya menemukanmu”

(Ali bin Abi Thalib)

“The best way to get started is to quit talking and begin doing”

(Walt Disney)

“Never regret a day in your life. Good days give happiness, bad days give experiences, the worst days give lessons, and the best day give memories”

(Anonymous)

PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji syukur kepada Allah SWT. atas segala Rahmat dan karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. shalawat serta salam teriring kepada Nabi Muhammad SAW sebagai suri tauladan *akhlakul karimah*.

KUPERSEMBAHKAN KARYA INI KEPADA:

“Kedua Orang Tuaku Tercinta

Yang senantiasa selalu memberikan yang terbaik, mengorbankan begitu banyak hal dan melantunkan doa yang tak henti-hentinya untukku. Kuucapkan terima kasih sebesar-besarnya karena telah mendidik dan membesarkanku menjadi pribadi yang baik dengan kasih sayang, kecukupan dan kebahagiaan yang akan selalu aku syukuri seumur hidupku. Semoga dengan ilmu yang kudapatkan dari hasil jerih payah kalian menyekolahkanku akan menjadi amal jariyah bagi kalian”

“Terima kasih untuk kakakku Novita Anjelina yang selalu mendukungku dan memberikan motivasi untuk menjadi pribadi yang lebih baik. Semoga kelak kita bisa menjadi saudara yang selalu bersama dan membahagiakan kedua orang tua kita”

“Diriku sendiri yang telah berjuang sampai akhir. Terima kasih sudah bertahan sampai ditahap ini dan maaf untuk malam-malam yang selalu memikirkan hal-hal yang begitu berat serta air mata yang harus dikeluarkan ketika sedang tidak baik-baik saja. Kamu hebat. Semoga sukses dan bahagia selalu”

“Seluruh Keluarga Besar Teknik Elektro 2019, terima kasih telah menemani dan berjuang bersama di bangku perkuliahan. Begitu banyak cerita yang telah dilalui bersama baik itu susah, senang, tangis yang akan selalu kukenang. Semoga kita bisa menjadi orang-orang yang sukses di kemudian hari”

“Almamater tercinta, Universitas Lampung dan Jurusan Teknik Elektro”

SANWACANA

Puji syukur penulis panjatkan kepada kehadirat Allah SWT. yang telah memberikan rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul “Rancang Bangun Sistem Peminjaman *Coworking Space* Berbasis *Website* Menggunakan *MERN Stack Technology* Pada Gedung Lembaga Kemahasiswaan FISIP Universitas Lampung”. Selama masa penelitian penulis mendapatkan banyak bantuan, bimbingan dan dukungan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Kedua orang tua tercinta dan keluarga penulis yang selalu memberikan dukungan, doa, kasih sayang tiada akhir dan mengingatkan penulis untuk bangkit dalam menyelesaikan penelitian ini;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Bapak Mona Arif Muda, S.T.,M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan Dosen Pembimbing Akademik yang telah membantu proses kelancaran pengerjaan penelitian dan perkuliahan;
5. Bapak Ir. Meizano Ardhi Muhammad, S.T., M.T. selaku Pembimbing Utama yang selalu meluangkan waktunya untuk memberikan bimbingan dan dukungan serta memudahkan penulis dalam menyelesaikan penelitian ini;
6. Ibu Ir. Trisya Septiana, S.T., M.T., IPM., selaku Pembimbing Pendamping yang selalu memberikan dukungan serta bimbingan agar menjadi lebih baik;

7. Bapak Wahyu Eko Sulistiono, S.T., M.Sc., selaku Penguji yang telah memberikan banyak saran dan masukan terhadap penelitian ini;
8. Pihak FISIP Unila yang telah membantu penulis selama melakukan penelitian;
9. Mbak Rika selaku *Admin* Program Studi Teknik Informatika yang telah banyak membantu penulis dalam urusan *administrasi* selama perkuliahan dan penelitian;
10. Alfiyah Widiyaningsih, Dwi liliyawati, Husniatun Aini, Meilika Dwi Putri, Reistha Ramadhanty, Selvia Eldina, Silvia Naim dan segenap teman-teman TI 2019 yang telah menjadi rumah bagi penulis selama masa perkuliahan. Menemani, membantu dan memberikan pundak bagi penulis dikala susah dan sedih;
11. Yola Ajeng Priyanto sebagai teman semasa SMA sampai sekarang yang selalu mendengarkan keluh kesah dari awal perkuliahan hingga di titik ini dengan memberikan semangat dan motivasi;
12. Dwi Ajeng Rahayu, Nimas Ayu Annisa, Nur Aini Fadilah dan segenap teman-teman KKN yang selalu mendukung dan membersamai penulis sampai saat ini, walaupun kita dipertemukan dipenghujung masa perkuliahan namun kalian selalu menemani dan membantu penulis dikala susah dan sedih;
13. Keluarga besar Teknik Elektro Angkatan 2019 yang telah menjadi teman seperjuangan sejak mahasiswa baru. Terima kasih telah mewarnai masa perkuliahan penulis dan menulis banyak cerita bersama;

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang Teknik Informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, 01 Agustus 2023
Penulis,

Yovanta Anjelina

DAFTAR ISI

	Halaman
DAFTAR GAMBAR	vii
DAFTAR TABEL	xi
I. PENDAHULUAN	1
1.1 Latar Belakang dan Masalah	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	4
1.5 Sistematika Penulisan Skripsi	5
II. TINJAUAN PUSTAKA	6
2.1 <i>Coworking Space</i> Gedung Lembaga Kemahasiswaan FISIP Unila	6
2.2 Sistem Informasi Peminjaman.....	7
2.3 <i>Website</i>	7
2.4 Javascript	8
2.5 <i>MERN Stack</i>	8
2.6 MongoDB.....	9
2.7 ExpressJS.....	9
2.8 ReactJS	10
2.9 NodeJS.....	11
2.10 <i>Unified Modelling Language (UML)</i>	12

2.11	<i>Entity Relationship Diagram (ERD)</i>	13
2.12	Metode Pengembangan Sistem <i>Personal Extreme Programming (PXP)</i> ...	14
2.13	Jest	17
2.14	<i>Black-box Testing</i>	17
2.15	<i>User Experience Questionnaire (UEQ)</i>	18
2.16	Penelitian Terkait	20
III.	METODOLOGI PENELITIAN.....	23
3.1	Waktu dan Tempat	23
3.2	Alat dan Bahan	24
3.3	Tahapan Penelitian	25
3.4	Tahap Pengembangan Sistem.....	26
3.4.1	Tahap <i>Requirements</i> (Pengumpulan Kebutuhan).....	27
3.4.2	Tahap <i>Planning</i> (Perencanaan)	33
3.4.3	Tahap <i>Iteration Initialization</i> (Inisialisasi Iterasi)	33
3.4.4	Tahap <i>Design</i> (Perancangan)	34
3.4.5	Tahap <i>Implementation</i> (Implementasi)	46
3.4.6	Tahap <i>System Testing</i> (Pengujian)	46
3.4.7	Tahap <i>Retrospective</i>	46
3.4.8	Tahap <i>Deployment</i>	47
IV.	HASIL DAN PEMBAHASAN.....	48
4.1	Hasil.....	48
4.1.1	Iterasi 1	48
4.1.2	Iterasi 2.....	64
4.1.3	Iterasi 3.....	92

4.1.4	<i>Deployment</i>	117
4.2	Pembahasan	121
4.2.1	Implementasi Metode <i>Personal Extreme Programming</i> (XP).....	121
4.2.2	Kegunaan Sistem Peminjaman <i>Coworking Space</i> pada Gedung Lembaga Kemahasiswaan FISIP Unila	122
V.	KESIMPULAN DAN SARAN	123
5.1	Kesimpulan.....	123
5.2	Saran.....	124
	DAFTAR PUSTAKA	125
	LAMPIRAN.....	128

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Ruang Rapat dengan konsep <i>Coworking Space</i> di Gedung Lembaga Kemahasiswaan FISIP Unila.....	7
Gambar 2.2 Konsep MERN <i>Stack Technology</i>	9
Gambar 2.3 Konsep MVC (<i>Model, View, Controller</i>).....	10
Gambar 2.4 <i>Trend library</i> ReactJS dengan <i>framework</i> Vue dan Angular.....	10
Gambar 2.5 Alur metode PXP	15
Gambar 2.6 Daftar Pertanyaan UEQ.....	19
Gambar 3.1 Tahapan penelitian	26
Gambar 3.2 Sistem yang sedang berjalan	28
Gambar 3.3 Sistem yang diusulkan.....	29
Gambar 3.4 <i>Use case diagram</i> dari Sistem Peminjaman <i>Coworking Space</i>	35
Gambar 3.5 <i>Activity Diagram Admin</i> Kelola Data <i>Booking</i>	37
Gambar 3.6 <i>Activity Diagram Admin</i> Kelola Data Ruangan	38
Gambar 3.7 <i>Activity Diagram Admin</i> Kelola Data <i>Customer</i>	39
Gambar 3.8 <i>Activity Diagram Admin</i> Kelola Data <i>Review</i>	40
Gambar 3.9 <i>Activity Diagram Admin</i> <i>Edit Profile</i>	40
Gambar 3.10 <i>Activity Diagram Register Customer</i>	41
Gambar 3.11 <i>Activity Diagram Customer</i> <i>Search Room</i>	42
Gambar 3.12 <i>Activity Diagram Customer</i> <i>See Detail Room</i>	42
Gambar 3.13 <i>Activity Diagram Customer</i> <i>View All Room dan Booking Room</i>	43

Gambar 3.14 <i>Activity Diagram Customer Edit Profile</i>	44
Gambar 3.15 <i>Activity Diagram Customer Melihat History, Pengecekan Status dan Review dari Booking</i>	45
Gambar 4.1 <i>Entity Relationship Diagram (ERD) Sistem Peminjaman Coworking Space</i>	49
Gambar 4.2 Struktur <i>Folder Proyek Backend Server</i>	50
Gambar 4.3 Kode program pengujian unit API <i>get room by id</i>	51
Gambar 4.4 Hasil <i>unit testing</i> dengan Jest pada API.....	51
Gambar 4.5 Kode Program Konfigurasi MongoDB	52
Gambar 4.6 Tampilan <i>Dashboard</i> pada MongoDB Atlas	52
Gambar 4.7 Kode Program Fungsi Register dan Login.....	54
Gambar 4.8 Kode Program <i>Create dan Update Room</i>	55
Gambar 4.9 Kode Program <i>Get dan Delete Room</i>	56
Gambar 4.10 Kode Program <i>Get dan Delete Booking</i>	57
Gambar 4.11 Kode Program <i>Create dan Update Booking</i>	59
Gambar 4.12 Kode Program <i>Get dan Delete Customer</i>	60
Gambar 4.13 Kode program <i>Update Customer dan setAdmin</i>	61
Gambar 4.14 Kode Program <i>Create dan Delete Review</i>	62
Gambar 4.15 Kode Program <i>Get Review</i>	63
Gambar 4.16 Desain <i>Wireframe</i> Halaman <i>Web Admin</i>	66
Gambar 4.17 Desain <i>High Fidelity</i> Halaman <i>Web Admin</i>	66
Gambar 4.18 Struktur <i>Folder</i> Proyek <i>Sipenting Web Admin</i>	67
Gambar 4.19 Kode program pengujian unit pada komponen <i>form login admin</i>	69
Gambar 4.20 Hasil <i>unit testing</i> dengan Jest pada komponen halaman <i>web admin</i>	70
Gambar 4.21 Kode Program API <i>Forgot Password dan Verifikasi Email</i>	71
Gambar 4.22 Kode Program Halaman <i>Login Admin</i>	73
Gambar 4.23 Tampilan Halaman <i>Login Admin</i>	73
Gambar 4.24 Kode Program Halaman <i>Forgot dan Reset Password</i>	74
Gambar 4.25 Tampilan Halaman <i>Forgot Password dan Reset Password</i>	75
Gambar 4.26 Kode Program Halaman <i>Dashboard Admin</i>	76

Gambar 4.27 Kode Program Integrasi Halaman <i>Dashboard Admin</i>	76
Gambar 4.28 Kode Program <i>Custom Hook useFetch()</i> dan <i>useFetchPrivate()</i>	77
Gambar 4.29 Kode Program <i>Custom Hook useAxiosPrivate()</i> dan <i>useRefreshToken()</i>	79
Gambar 4.30 Tampilan Halaman <i>Dashboard Admin</i>	79
Gambar 4.31 Kode Program <i>Admin</i> Melakukan <i>Create</i> dan <i>Update Room</i>	81
Gambar 4.32 Kode Program <i>Admin Delete Room</i>	82
Gambar 4.33 Tampilan Halaman <i>Kelola Room</i>	82
Gambar 4.34 Tampilan Halaman <i>Create Room</i>	83
Gambar 4.35 Tampilan Halaman <i>Update Room</i>	83
Gambar 4.36 Kode Program <i>Get, Update</i> dan <i>Delete Data Booking</i>	84
Gambar 4.37 Tampilan Halaman <i>Kelola Booking</i>	85
Gambar 4.38 Tampilan Halaman <i>Update Booking</i>	85
Gambar 4.39 Kode Program <i>Get, Delete</i> dan <i>Set As Admin Data Customer</i>	86
Gambar 4.40 Tampilan Halaman <i>Kelola Customer</i>	87
Gambar 4.41 Kode Program <i>Get</i> dan <i>Delete Data Review</i>	88
Gambar 4.42 Tampilan Halaman <i>Kelola Review</i>	88
Gambar 4.43 Kode Program <i>Edit Profile Admin</i>	90
Gambar 4.44 Tampilan Halaman <i>Edit Profile Admin</i>	90
Gambar 4.45 Desain <i>Wireframe</i> Halaman <i>Web Customer</i>	94
Gambar 4.46 Desain <i>High Fidelity</i> Halaman <i>Web Customer</i>	95
Gambar 4.47 Struktur <i>Folder</i> Proyek <i>Sipenting Web Customer</i>	96
Gambar 4.48 Kode program pengujian unit pada komponen <i>register</i>	97
Gambar 4.49 Hasil <i>unit testing</i> dengan <i>Jest</i> pada komponen <i>customer web</i>	98
Gambar 4.50 Kode Program <i>Admin</i> Konfirmasi Persetujuan <i>Booking</i>	99
Gambar 4.51 Kode Program <i>Approve</i> dan <i>Reject Booking</i>	100
Gambar 4.52 Halaman <i>Booking Admin</i> Konfirmasi Persetujuan <i>Booking</i>	100
Gambar 4.53 Kode Program <i>Get Room Popular</i> dan <i>Type Room</i>	101
Gambar 4.54 Tampilan Halaman <i>Landing Page</i>	102
Gambar 4.55 Kode Program Halaman <i>Search Room</i>	103
Gambar 4.56 Tampilan Halaman <i>Search Room</i>	103

Gambar 4.57 Tampilan Halaman Tentang	104
Gambar 4.58 Tampilan Halaman <i>All Room</i>	104
Gambar 4.59 Kode Program <i>Detail Room</i>	105
Gambar 4.60 Tampilan Halaman <i>Detail Room</i>	106
Gambar 4.61 Tampilan Halaman <i>Detail Room Image Modal</i>	106
Gambar 4.62 Kode Program <i>handleRegister</i>	107
Gambar 4.63 Tampilan Halaman <i>Register Customer</i>	108
Gambar 4.64 Kode Program Halaman Verifikasi <i>Email</i>	108
Gambar 4.65 Kode Program <i>Login Customer</i>	109
Gambar 4.66 Tampilan Halaman <i>Login Customer</i>	110
Gambar 4.67 Kode Program Halaman <i>Booking Customer</i>	111
Gambar 4.68 Tampilan Halaman <i>Booking Room</i>	111
Gambar 4.69 Kode Program <i>Customer Logout</i>	112
Gambar 4.70 Tampilan Halaman <i>Profile</i>	113
Gambar 4.71 Tampilan Halaman <i>Edit Profile Customer</i>	113
Gambar 4.72 Kode Program Halaman Riwayat Peminjaman Dan <i>Review</i> Ruangan	115
Gambar 4.73 Tampilan Halaman Riwayat Peminjaman dan <i>Review</i> Ruangan	116
Gambar 4.74 <i>Diagram</i> Karakteristik Responden.....	118
Gambar 4.75 UEQ Benchmark	120

DAFTAR TABEL

	Halaman
Tabel 2.1 Komponen penyusun <i>Use case diagram</i>	12
Tabel 2.2 Komponen penyusun <i>Activity Diagram</i>	13
Tabel 2.3 <i>Benchmark</i> interval untuk skala UEQ.....	20
Tabel 3.1 Tabel waktu penelitian	23
Tabel 3.2 Alat dan Bahan.....	24
Tabel 3.3 Tabel Pertanyaan Wawancara.....	27
Tabel 3.4 Tabel Jawaban Wawancara.....	27
Tabel 3.5 <i>User story</i> sistem.....	30
Tabel 3.6 Kebutuhan Fungsional Sistem	32
Tabel 3.7 Kebutuhan Non Fungsional Sistem	32
Tabel 3.8 Daftar Tugas.....	33
Tabel 3.9 Definisi Aktor	35
Tabel 3.10 Definisi <i>Use case</i>	36
Tabel 4.1 Iterasi 1 Pengembangan Sistem Peminjaman <i>Coworking Space</i>	48
Tabel 4.2 <i>Retrospective</i> iterasi pertama	64
Tabel 4.3 Iterasi 2 Pengembangan Sistem Peminjaman <i>Coworking Space</i>	65
Tabel 4.4 <i>Retrospective</i> iterasi kedua.....	92
Tabel 4.5 Iterasi 3 Pengembangan Sistem Peminjaman <i>Coworking Space</i>	93
Tabel 4.6 <i>Retrospective</i> iterasi ketiga	117
Tabel 4.7 Distribusi Jawaban Responden	119
Tabel 4.8 Skala UEQ (<i>Mean</i> dan <i>Variance</i>)	119
Tabel 4.9 UEQ Benchmark.....	120

I. PENDAHULUAN

1.1 Latar Belakang dan Masalah

Perkembangan teknologi informasi saat ini telah membawa perubahan yang besar dalam berbagai aspek kehidupan termasuk cara manusia dalam mengelola informasi pada suatu organisasi. Transformasi ini melibatkan penggunaan teknologi yang lebih efisien dan terintegrasi guna mengelola informasi dengan lebih baik. Salah satu pemanfaatan teknologi tersebut adalah sistem informasi untuk mengelola data dan pemesanan pada *coworking space*. *Coworking space* merupakan konsep ruangan bersama yang dapat digunakan sebagai tempat berkerja sekaligus untuk berkumpulnya komunitas dalam bekerja sama dan berbagi ide atau gagasan (Sia *et al.*, 2019).

Saat ini, terdapat sistem yang dapat digunakan untuk mencari dan menyewa *coworking space* seperti XWORK dan WEWORK. XWORK merupakan perusahaan yang berdiri sejak 2016 dan memiliki ruangan yang tersedia diseluruh Indonesia dengan berbagai tipe ruangan seperti ruang *meeting*, ruang acara, *coworking space*, ruang kantor, serta *virtual office* (Xwork, 2023). XWORK meningkatkan keefektifan penyewa dengan memasarkan produk sewaanannya secara *online* melalui *website* dan aplikasi (Gautama *et al.*, 2019). Sedangkan WEWORK merupakan perusahaan yang menyediakan ruang kerja global dan didirikan pada tahun 2010 yang berpusat di New York City. WEWORK menyediakan sistem untuk mencari dan menyewa *coworking space* yang dibutuhkan perusahaan rintisan. WEWORK telah memperluas jangkauan ke 724 lokasi di seluruh dunia (WeWork, 2023). Salah satu lokasi di Indonesia yaitu di Jakarta dengan 4 gedung yang dapat disewakan.

Selain perusahaan, penerapan sistem peminjaman *coworking space* juga dapat diimplementasikan di lingkungan universitas. Saat ini, banyak universitas yang menyadari pentingnya kolaborasi dan kreativitas dalam lingkungan akademik. Seperti pada Fakultas Ilmu Sosial dan Ilmu Politik Universitas Lampung (FISIP Unila), telah dibangun Gedung Lembaga Kemahasiswaan yang didesain dengan konsep *coworking space*. Tujuan gedung ini adalah memberikan tempat bagi civitas akademik dalam menjalankan berbagai aktivitas seperti bekerja, berdiskusi dan berkumpul di satu tempat. Fasilitas yang tersedia pada gedung ini meliputi ruang rapat, tempat duduk dan loker untuk penyimpanan barang.

Namun pada *coworking space* di Gedung Lembaga Kemahasiswaan FISIP Unila, mekanisme dan proses peminjaman ruangan saat ini masih dilakukan secara manual. Mahasiswa harus memeriksa ketersediaan ruangan secara langsung, kemudian mendatangi pihak *admin* yang bertugas di ruangan khusus berdekatan dengan *coworking space* untuk mencatat data peminjaman. Dengan alur proses tersebut, terdapat beberapa masalah yang dihadapi, antara lain bentroknnya jadwal peminjaman ruangan, adanya duplikasi data serta kesalahan dalam mengecek ketersediaan ruangan. Hal tersebut dapat memperlambat proses peminjaman ruangan sehingga menyebabkan ketidaknyamanan bagi pengguna yang membutuhkan ruangan pada waktu yang sama.

Dengan adanya permasalahan tersebut, kualitas pelayanan yang diberikan memiliki hubungan yang erat dengan kepuasan pelanggan. Perkembangan teknologi dan perubahan perilaku konsumen telah mendorong perusahaan-perusahaan untuk beradaptasi dan memanfaatkan peluang tersebut sebagai cara efektif dalam menyampaikan informasi kepada pengguna (Davrin and Hidayat, 2021). Untuk meningkatkan kepuasan atas kualitas pelayanan, pihak pengelola *coworking space* FISIP Unila dapat memberikan kemudahan pelayanan bagi setiap pengguna, mulai dari memberikan informasi terkait fasilitas ruangan hingga mempermudah proses peminjaman *coworking space* dengan mengubah proses peminjaman secara konvensional menjadi lebih *modern* dengan sistem peminjaman digital.

Dengan membangun sistem peminjaman ruangan berbasis *website* pada *coworking space* di Gedung Lembaga Kemahasiswaan FISIP Unila dapat membantu pihak pengelola dalam mengoptimalkan penggunaan dan meningkatkan efektivitas dan efisiensi peminjaman *coworking space*. Sistem ini dibangun menggunakan teknologi *MERN Stack* yang terdiri dari MongoDB, ExpressJS, ReactJS, dan NodeJS. Untuk memenuhi kebutuhan pengguna terhadap peminjaman *coworking space*, proses pengembangan sistem menggunakan metode *Personal Extreme Programming* (PXP). Metode PXP merupakan salah satu metode *agile* yang iteratif, fleksibel dan responsif terhadap suatu perubahan. Penerapan metode PXP dapat digunakan untuk mempercepat proses pengembangan sistem dengan berorientasi kepada kebutuhan klien dan hasil produk. (W, Wahyuni and Nuryasin, 2023).

1.2 Perumusan Masalah

Berdasarkan latar belakang, kajian masalah yang mendasari penelitian ini adalah:

1. Bagaimana mengembangkan sebuah Sistem Peminjaman *Coworking Space* yang sesuai dengan kebutuhan pengguna menggunakan metode pengembangan *Personal Extreme Programming* (PXP)?
2. Bagaimana membangun Sistem Peminjaman *Coworking Space* yang dapat membantu peminjaman ruangan dan mengelola data ruangan secara *online* pada *Coworking Space* Gedung Lembaga Kemahasiswaan FISIP Unila?
3. Bagaimana mengevaluasi fungsionalitas dan pengalaman pengguna (*user experience*) dari Sistem Peminjaman *Coworking Space* menggunakan metode *Black-box Testing* dan *User Experience Questionnaire* (UEQ)?

1.3 Tujuan Penelitian

Tujuan adalah suatu usaha yang hendak dicapai dari awal terfokusnya sebuah penelitian. Tujuan dari dilakukannya penelitian ini adalah:

1. Mengembangkan sebuah Sistem Peminjaman *Coworking Space* yang sesuai dengan kebutuhan pengguna dengan menerapkan metode pengembangan *Personal Extreme Programming (PXP)*.
2. Membangun Sistem Peminjaman *Coworking Space* berbasis *web* pada *Coworking Space* Gedung Lembaga Kemahasiswaan FISIP Unila untuk memudahkan peminjaman ruangan, pengelolaan data dan melakukan penjadwalan ruangan guna mencegah terjadinya konflik jadwal peminjaman.
3. Mengevaluasi fungsionalitas dan pengalaman pengguna (*user experience*) terhadap Sistem Peminjaman *Coworking Space* menggunakan *Black-box Testing* dan *User Experience Questionnaire (UEQ)* untuk membantu dalam meningkatkan pengembangan sistem.

1.4 Batasan Masalah

Batasan dalam penelitian ini adalah sebagai berikut:

1. Sistem yang dibangun adalah sistem peminjaman ruangan dan pengelolaan data peminjaman ruangan dengan hak akses *user* sebagai *admin* dan *customer*.
2. Sistem yang dibangun berbasis *website* menggunakan *MERN Stack Technology* yang terdiri dari MongoDB, ExpressJS, ReactJS dan NodeJS dengan bahasa pemrograman utama yaitu Javascript.
3. Sistem ini ditujukan kepada civitas akademik Fakultas Ilmu Sosial dan Ilmu Politik Universitas Lampung (FISIP Unila) untuk melakukan peminjaman *Coworking Space* pada Gedung Lembaga Kemahasiswaan FISIP Unila.

1.5 Sistematika Penulisan Skripsi

Sistematika penulisan skripsi ini terdiri dari 5 bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat latar belakang serta motivasi dari penyusunan skripsi ini, serta permasalahan penelitian yang ada. Perumusan masalah, tujuan penelitian, batasan masalah dan sistematika penulisan juga terdapat di bab ini.

BAB II TINJAUAN PUSTAKA

Bab ini memuat dasar-dasar teori yang menjadi landasan dalam penelitian terkait *Coworking Space* Gedung Lembaga Kemahasiswaan FISIP Unila, *website*, Javascript, *MERN Stack*, MongoDB, ExpressJS, ReactJS, NodeJS, Metode Pengembangan Sistem *Personal Extreme Programming (XP)*, *Unified Modelling Language (UML)*, *Entity Relationship Diagram (ERD)*, Jest, *Black-box testing*, *User Experience Questionnaire (UEQ)* dan penelitian terkait.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan yaitu metode *Personal Extreme Programming (XP)*, waktu dan tempat penelitian, tahapan penelitian secara umum, dan tahapan penelitian secara terperinci.

BAB IV HASIL DAN PEMBAHASAN

Bab ini memuat hasil rancangan dari sistem dengan tahapan metode XP yaitu *iteration initialization*, *design*, *implementation*, *testing* dan *retrospective* dan hasil dari pengujian UEQ.

BAB V KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan dari kegiatan yang dilakukan pada penelitian ini. Saran-saran mengenai perbaikan dan pengembangan lebih lanjut juga terdapat di bab ini.

II. TINJAUAN PUSTAKA

2.1 *Coworking Space* Gedung Lembaga Kemahasiswaan FISIP Unila

Coworking space pada Gedung Lembaga Kemahasiswaan yang terletak di FISIP Unila merupakan konsep ruangan yang menyediakan ruang rapat untuk memfasilitasi civitas akademik FISIP Unila dalam menjalankan aktivitas, seperti rapat bersama yang bertujuan untuk bertukar ide atau gagasan. *Coworking space* berfungsi untuk menyediakan ruang kerja yang dapat menciptakan lingkungan untuk mendukung kolaborasi, inovasi, fleksibilitas dan berbagi pengalaman serta pengetahuan antar para penggunanya (Yoga, 2018).

Pada tahun 2022, gedung tersebut diresmikan oleh Rektor Unila yang terdiri dari 3 lantai. Lantai pertama digunakan sebagai *Food Court* dengan konsep *self service*, lantai kedua digunakan sebagai *Coworking Space* untuk ruang rapat atau belajar dan lantai ketiga digunakan sebagai ruang sejarah FISIP. Pada lantai kedua ini memiliki 8 ruangan yang dapat digunakan, yaitu *Coworking Space* 1 sampai dengan 4 dan *Meeting Room* 1 sampai dengan 4. Selain itu, terdapat juga loker untuk menyimpan barang dan ruangan *admin* untuk melakukan peminjaman ruangan.

Coworking Space ini terdapat berbagai model ruangan dengan fasilitas dan desain interior yang beragam sehingga dapat memberikan suasana yang santai dan nyaman. Salah satu contoh model ruangan yang mengusung konsep *coworking space* ini dapat dilihat pada gambar berikut.



Gambar 2.1 Ruang Rapat dengan konsep *Coworking Space* di Gedung Lembaga Kemahasiswaan FISIP Unila

2.2 Sistem Informasi Peminjaman

Sistem informasi merupakan sekumpulan dari aktivitas manusia dan penggunaan teknologi seperti *hardware*, *software*, jaringan komputer dan *database* dalam mengumpulkan, mengelola dan menyebarkan suatu informasi (Ariyanti *et al.*, 2020). Sistem peminjaman merupakan salah satu penerapan dari sistem informasi yang bertujuan untuk memudahkan proses peminjaman barang, tempat atau jasa dengan lebih cepat dan mengurangi *human error*. Sistem informasi peminjaman umumnya menyediakan informasi mengenai ketersediaan dan detail informasi peminjaman dari barang, tempat atau jasa.

2.3 Website

Website atau situs merupakan tempat penyimpanan data dan informasi yang berisi topik tertentu. Situs *web* diumpamakan sebuah buku yang berisikan sebuah topik tertentu yang terdiri dari halaman-halaman *web* yang saling berkaitan di dalam *web* tersebut. *Web* dibuat dengan beberapa bahasa pemrograman seperti HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheet*), Javascript (Sari and Hidayat, 2022).

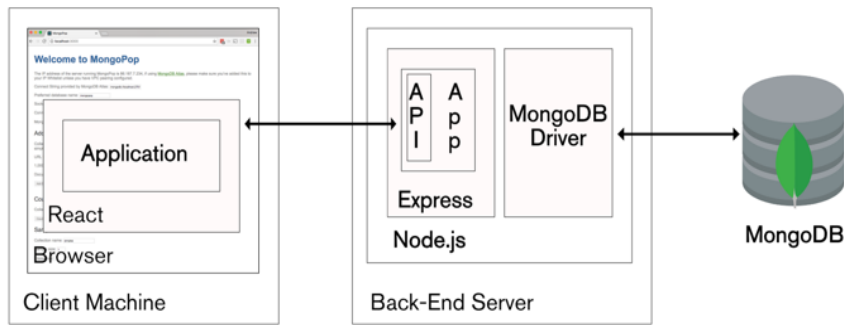
Website terbagi menjadi 2 jenis yaitu *website* statis dan *website* dinamis. Keduanya memiliki fungsi yang sama yaitu untuk menampilkan informasi yang dibungkus dalam halaman *website*. *Website* statis merupakan *website* yang memiliki konten dan data yang bersifat tetap dimana konten tersebut sudah ditampilkan ketika *website* dibuka dan tidak terhubung ke *database*. Sedangkan *website* dinamis merupakan *website* yang bersifat fleksibel dimana konten dan data dari halaman *web* ini dapat menyesuaikan dengan *request* dari pengguna secara *real-time* dan terhubung ke *database*.

2.4 Javascript

Javascript merupakan bahasa pemrograman tingkat tinggi yang mudah dipahami oleh manusia untuk membangun sebuah *website*. Javascript dikembangkan oleh Brendan Eich di Netscape pada tahun 1995 (Sianipar, 2015). Javascript membuat *website* menjadi lebih dinamis dan interaktif dengan menyediakan fungsionalitas pada elemen yang berinteraksi dengan pengguna. Javascript juga bersifat fleksibel untuk diuji, oleh karena itu Javascript memiliki beberapa *framework* atau *library* seperti ReactJS, VueJS, AngularJS dan sebagainya. Javascript berjalan di sisi *client* atau *browser* sehingga Javascript dapat digunakan pada banyak *browser* seperti Internet Explorer, Firefox, Chrome, Opera, Safari dan lainnya.

2.5 MERN Stack

MERN *Stack* atau singkatan dari MongoDB, ExpressJS, ReactJS, NodeJS merupakan kombinasi dari teknologi berbasis Javascript yang digunakan dalam pengembangan sebuah *website*. Konsep dari MERN *Stack* adalah sisi *client* / *web browser* ditangani oleh *library* ReactJS dan akan melakukan komunikasi data ke *Backend Server* dengan NodeJS. Komunikasi tersebut ditangani oleh *framework* ExpressJS yang digunakan untuk membuat API dan mengirim *response* dari *request* yang diterima. Kemudian NodeJS menggunakan *library* mongoose untuk menghubungkan *backend server* ke *database* NoSQL MongoDB untuk mengambil data.



Gambar 2.2 Konsep MERN *Stack Technology*
 Sumber : (Morgan, 2017)

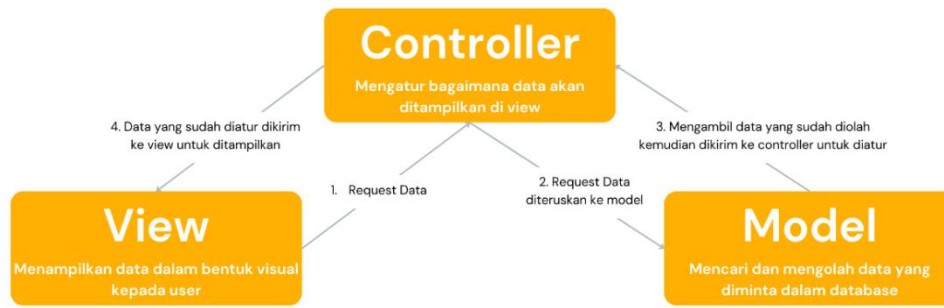
2.6 MongoDB

MongoDB merupakan NoSQL (*Not only Structured Query Language*) *database* yang dapat menyimpan data dalam bentuk yang fleksibel seperti dokumen JSON. MongoDB adalah *document-oriented database* dimana data yang disimpan berbeda dengan *database* relasional yang menggunakan tabel dan relasi, pada *database* ini mengganti tabel dengan sebuah model berupa dokumen JSON yang menyimpan hierarki dan relasi didalamnya (Handoyo, Santoso and Setiawan, 2019).

MongoDB Atlas merupakan layanan dari MongoDB untuk layanan *multi-cloud database* yang dapat menjalankan aplikasi secara bersamaan dengan *Amazon Web Services* (AWS), Google Cloud, atau Microsoft Azure. MongoDB Atlas dapat memudahkan pengelolaan *database* secara gratis dengan batas penyimpanan 512 MB.

2.7 ExpressJS

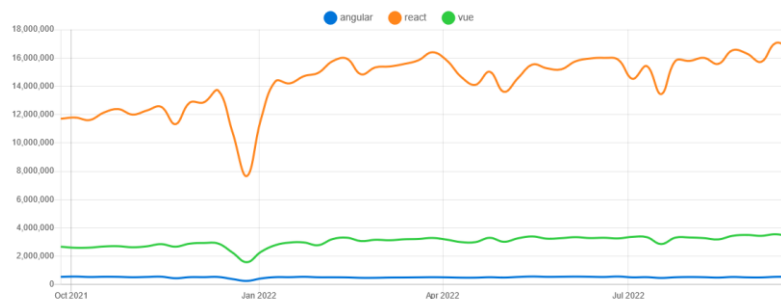
ExpressJS merupakan *framework server-side* untuk membangun aplikasi dengan NodeJS. *Framework* ini digunakan untuk mengatur fungsionalitas sistem seperti membuat *routing* API, penanganan *error*, mengatur *middleware* untuk pertukaran data melalui *request* dan *response* (Handoyo, Santoso and Setiawan, 2019). ExpressJS memiliki arsitektur MVC (*Model View Controller*). *Model* akan mengatur data yang diolah, *View* akan mengatur tampilan UI dan *Controller* untuk menangani logika program dengan perintah berupa *function* untuk menghubungkan *Model* dan *View*.



Gambar 2.3 Konsep MVC (*Model, View, Controller*)

2.8 ReactJS

ReactJS merupakan *library* dari bahasa pemrograman Javascript yang bersifat *open source* yang dibuat oleh Facebook untuk membangun *User Interface* (UI) dari *website*. ReactJS merupakan *library* dikarenakan ReactJS memberikan kebebasan kepada *developer* dalam memanggil *library* kapan dan dimana untuk menggunakannya di dalam kode program. ReactJS hanya mengatur hal yang berkaitan dengan tampilan dan logika yang ada disekitarnya. Menurut situs *npm trends.com*, ReactJS menjadi teknologi yang banyak digunakan oleh para *developer* untuk mengembangkan *website* dibandingkan dengan *framework* lainnya seperti Angular dan Vue.



Gambar 2.4 *Trend library* ReactJS dengan *framework* Vue dan Angular

Berdasarkan situs resmi ReactJS, keuntungan menggunakan ReactJS adalah *declarative* (pembuatan program yang befokus pada apa yang dicapai), *component-based* (satu halaman yang dipecah menjadi beberapa komponen), *component reusable* (komponen yang dapat digunakan kembali di halaman yang berbeda), dan *learn once, write*

anywhere (konsep dari ReactJS sama dengan React Native). ReactJS sudah teruji dengan banyaknya aplikasi yang dibangun seperti Facebook, Netflix, Instagram, Traveloka, AirBnB dan lainnya.

ReactJS diaplikasikan sebagai *View (V)* pada *Model-View-Controller (MVC)* yaitu bagian yang mengatur tampilan yang diberikan terhadap pengguna (Panjaitan and Pakpahan, 2021). ReactJS mengabstraksi *Document Object Model (DOM)*, sehingga pengembangan aplikasi dapat sederhana, berkinerja dan kuat. Komponen pada ReactJS sebagian besar dimuat (*render*) di sisi *server* sehingga dapat menerapkan *Single Page Application (SPA)* lalu setelah selesai semua hasil *render* akan dikirim ke sisi *client*.

Untuk melakukan instalasi ReactJS dapat menggunakan NPM (*Node Package Manager*). NPM adalah *package manager* yang digunakan untuk mengelola *package* atau *library* yang dibutuhkan untuk membangun *website* secara otomatis yang digunakan oleh Javascript *developer*.

2.9 NodeJS

NodeJS merupakan Javascript *runtime environment* yang bersifat *open-source* dan *cross-platform* sehingga dapat menjalankan kode Javascript dimana saja tanpa lingkungan *browser* (Handoyo, Santoso and Setiawan, 2019). NodeJS memiliki banyak *package/module* untuk membantu pengembangan *website*. Untuk mengelola *package* pada NodeJS dapat menggunakan *npm (node package manager)*. Berikut beberapa *package* yang ada pada NodeJS :




1. *Mongoose* : *package* yang digunakan untuk menghubungkan antara NodeJS dengan MongoDB dengan mendefinisikan model skema dokumen untuk menyimpan data.
2. *Bcrypt* : *package* yang digunakan untuk melakukan *hash password* dan token ketika menambahkan data ke *database* dan dapat melakukan proses *compare* pada data *hash* untuk mendapatkan data asli.
3. *JsonWebToken* : *package* yang digunakan untuk membuat token dengan JSON berbentuk *string* acak yang biasanya digunakan untuk melakukan autentikasi.


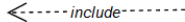
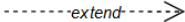
4. Nodemon : *package* yang digunakan untuk membantu dalam pengembangan sistem dengan melakukan *refresh* secara otomatis ketika terjadi perubahan pada file.
5. Cors : *package* yang digunakan untuk mengizinkan *browser* pada halaman *web* untuk mendapatkan akses ke *server* dengan domain yang berbeda.
6. Nodemailer : *package* yang digunakan untuk mengirim email dalam melakukan verifikasi akun.

2.10 Unified Modelling Language (UML)

Unified Modelling Language (UML) merupakan pemodelan yang digunakan untuk menggambarkan sebuah sistem dengan konsep yang berorientasi pada objek. UML dikembangkan oleh *Object Management Group* (OMG) yang merupakan kerja sama antara James Rumbaugh, Grady Booch dan Ivar Jacobson pada tahun 1997. UML gabungan dari metode OMT (*Object Modelling Technique*), *Booch* dan OOSE (*Object-Oriented Software Engineering*) (Fowler, 2004). Pemodelan UML berfokus pada representasi pengguna dan visualisasi struktur sistem dalam model (Putra, 2019). UML terdiri dari beberapa *diagram* yang digunakan dalam mendesain suatu sistem yaitu *use case diagram* dan *activity diagram*. *Use case diagram* digunakan untuk menggambarkan interaksi dan hubungan antara pengguna sistem dengan sistem itu sendiri. Komponen penyusun *Use case Diagram* terdiri dari *use case*, aktor, *system boundary* dan relasi asosiasi (Dennis, Wixom and Roth, 2012). Komponen *use case diagram* dapat dilihat pada Tabel 2.1 berikut.



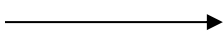

Tabel 2.1 Komponen penyusun *Use case diagram*

Nama	Simbol	Keterangan
Aktor		Aktor mewakili peran pengguna sistem ketika berkomunikasi dengan sistem.
<i>Use case</i>		<i>Use case</i> merupakan abstraksi dari interaksi antara aktor dan sistem
<i>System boundary</i>		<i>System boundary</i> merupakan ruang lingkup sistem yang menggambarkan bagian eksternal dan internal dari sistem.

Relasi asosiasi		Relasi asosiasi adalah representasi hubungan yang terjadi pada sistem yang menghubungkan interaksi antara aktor dengan <i>use case</i> .
Dependency		<i>Include</i> merupakan relasi yang menunjukkan bahwa satu <i>use case</i> secara langsung mencakup (<i>include</i>) fungsionalitas yang diperlukan dari <i>use case</i> lainnya. <i>Use case</i> yang termasuk <i>include</i> berfungsi sebagai bagian dari <i>use case</i> yang mencakup <i>use case</i> tersebut.
		<i>Extend</i> merupakan relasi yang menunjukkan bahwa suatu <i>use case</i> dapat memperluas (<i>extend</i>) fungsionalitas yang menyediakan fungsionalitas tambahan dengan membuat <i>use case</i> lainnya.

Sedangkan *activity diagram* merupakan teknik untuk menggambarkan logika prosedural, proses bisnis dan alur kerja (Fowler, 2004). *Activity diagram* digunakan untuk menggambarkan aktivitas yang terjadi dari awal hingga akhir pada proses menjalankan sistem. Berikut merupakan komponen penyusun *Activity Diagram*.

Tabel 2.2 Komponen penyusun *Activity Diagram*

Nama	Simbol	Keterangan
<i>Initial node</i>		<i>Initial node</i> merupakan titik awal aktivitas pada sistem dimulai.
<i>Action</i>		<i>Action</i> adalah aksi dari aktivitas yang terjadi pada sistem.
Garis Konstruksi		Garis konstruksi adalah penghubung antara proses <i>start</i> ke <i>action</i> dan <i>action</i> lainnya hingga mencapai titik akhir.
<i>Activity final</i>		<i>End</i> merupakan tujuan akhir atau titik akhir dari aktivitas yang ada pada sistem.

2.11 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah *diagram* yang digunakan untuk melakukan perancangan sebuah *database* pada suatu sistem. *Diagram* ini akan memperlihatkan suatu hubungan antar entitas atau objek beserta atributnya sehingga perancangan *database* dapat dilakukan dengan mudah dan terstruktur dikarenakan tabel dan *field* terdefinisi melalui objek dan atributnya (Dzulqori and Zaelani, 2021).

ERD terdiri dari 4 komponen utama (Dennis, Wixom and Roth, 2012), yaitu:

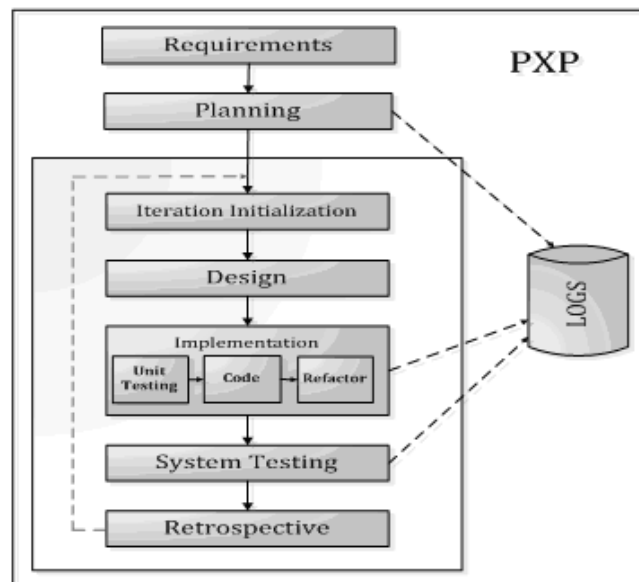
1. Entitas merupakan sekumpulan objek yang ada di dalam dunia nyata dan diidentifikasi secara unik.
2. Atribut, di dalam entitas terdapat beberapa atribut yang berfungsi untuk mendeskripsikan karakteristik dari suatu entitas. Entitas memiliki satu atribut unik yang menjadi atribut kunci (*key attribute*) sebagai data yang penting atau unik.
3. Relasi merupakan hubungan yang terjadi antara satu atau lebih entitas yang ditunjukkan dengan garis yang menghubungkan antar entitas.
4. Kardinalitas merupakan properti pada relasi yang menunjukkan jumlah maksimum suatu entitas yang dapat berelasi dengan entitas lain. Terdapat 3 jenis kardinalitas dari suatu relasi yaitu:
 - a. *Relasi One to One*, yaitu setiap entitas hanya dapat memiliki satu relasi dengan entitas lain.
 - b. *Relasi One to Many*, yaitu satu entitas dapat memiliki relasi dengan banyak entitas, begitupun sebaliknya.
 - c. *Relasi Many to Many*, yaitu setiap entitas dapat memiliki relasi dengan banyak entitas lain, begitu juga sebaliknya.

2.12 Metode Pengembangan Sistem *Personal Extreme Programming* (XP)

Metode *Personal Extreme Programming* (XP) merupakan gabungan antara model *Personal Software Process* (PSP) dan model *Extreme Programming* (XP) yang dimodifikasi karena pemrograman pasangan tidak dapat dicapai dan dapat menjalankan proses pengembangan sistem secara individu. Pada metode PSP diambil prinsip pengerjaan sistem secara individu untuk meningkatkan kinerja, memperkirakan dan merencanakan pekerjaan sebelum dimulai. Sedangkan pada metode XP yang memiliki 12 *core practice* dimana 6 *practice* dapat dipraktikkan pada metode XP yaitu *continuous integration*, *simple design*, *small releases*, *refactoring*, *test driven development* dan *spike solution* (Dzhurov, Krasteva and Ilieva, 2009). Kedua metode tersebut berfokus pada pengembangan produk berkualitas dan berorientasi pada klien.

PXP merupakan salah satu model dari metode *Agile* yang melakukan memiliki karakteristik *iterative* yaitu melakukan beberapa siklus perulangan untuk diselesaikan. PXP digunakan untuk memecahkan masalah dalam pengembangan sistem yang berskala kecil dan metode ini memungkinkan pengembangan sistem menjadi lebih *flexible* dan *responsive* ketika menghadapi perubahan dari kebutuhan pengguna sesuai yang diinginkan. (W, Wahyuni and Nuryasin, 2023)

Tahapan-tahapan pada pengembangan sistem menggunakan metode PXP yang terdiri dari *requirements*, *planning*, *iteration initialization*, *design*, *implementation* dengan 3 fase yaitu *unit testing*, *code*, dan *refactor*, lalu *system testing* dan *retrospective*. Tahapan PXP dapat dilihat pada Gambar 2.5 berikut.



Gambar 2.5 Alur metode PXP
Sumber : (Dzhurov, Krasteva and Ilieva, 2009)

Berdasarkan pada Gambar 2.5, berikut penjelasan dari tahapan pada metode PXP:

1. *Requirements* (Kebutuhan)

Requirements merupakan tahap awal dari metode PXP dengan melakukan pengumpulan daftar kebutuhan dengan cara eksplorasi, observasi dan wawancara bersama klien. Pada tahap ini akan membuat kebutuhan fungsional dan non-fungsional dari sistem yang akan dibangun.

2. *Planning* (Perencanaan)

Planning adalah tahapan untuk menentukan daftar tugas dari daftar kebutuhan yang telah didapatkan pada tahap sebelumnya. Dari daftar tersebut, dilakukan penyusunan tugas untuk dikerjakan selama pengembangan sistem dan setiap tugas tersebut akan diestimasikan waktu pengerjaannya. Selain itu, tahap ini menentukan keputusan desain utama, bahasa pemrograman, *framework*, *database* dan lainnya.

3. *Iteration Initialization* (Inisialisasi Iterasi)

Iteration Initialization merupakan tahapan awal untuk setiap memulai iterasi. Pada tahap ini dimulai dengan memilih tugas yang akan menjadi fokus dari iterasi yang akan dilakukan. Panjang waktu iterasi dapat dilakukan 1 hingga 3 minggu tergantung pada proyek yang dikerjakan. Setiap iterasi yang dilakukan akan menghasilkan kandidat rilis produk agar dapat dilakukan uji coba.

4. *Design* (Perancangan)

Design merupakan tahapan untuk pemodelan sistem yang akan diimplementasikan selama iterasi berlangsung. Pada tahap ini disarankan untuk menggunakan *tools* sesederhana mungkin dalam pemodelan sistemnya. Perancangan sistem ini harus bertujuan untuk membuat sistem yang memenuhi persyaratan klien tanpa perlu mencoba menebak kebutuhan di masa yang akan datang.

5. *Implementation* (Implementasi)

Implementation merupakan tahapan untuk mengimplementasikan perancangan model sistem kedalam kode program. Pada tahap ini melakukan *unit testing* dengan membuat, menjalankan dan menguji fungsi-fungsi dasar yang sesuai dengan *user story* yang telah dibuat. Setelah itu dilanjutkan dengan tahap *code* dengan menuliskan kode program secara keseluruhan lalu melakukan *refactoring* untuk membersihkan kode program yang *redundant* atau tidak digunakan lagi.

6. *System Testing* (Pengujian)

Pengujian terhadap fitur yang telah dikembangkan untuk mengevaluasi fungsionalitas dari keseluruhan sistem apakah telah sesuai dengan kebutuhan yang diinginkan. Tujuan pengujian ini adalah untuk menemukan dan memperbaiki *bug* atau *error* yang muncul saat sistem dijalankan.

7. *Retrospective*

Retrospective merupakan akhir dari proses iterasi. Pada tahap ini, dilakukan verifikasi terhadap estimasi waktu tugas dengan yang sebenarnya guna menemukan alasan penundaan atau keterlambatan, sehingga dapat mencegah ketidaksesuaian estimasi waktu pada iterasi selanjutnya. Tahap ini dapat berakhir dengan melakukan perilisan produk. Setelah tahap ini berakhir dapat memulai kembali iterasi baru dengan berpindah ke tahap *Iteration Initialization* atau pengembangan sistem dapat berakhir pada tahap ini apabila semua *requirement* dari sistem telah terpenuhi.

2.13 Jest

Jest merupakan *framework* pengujian kode program Javascript yang dibuat oleh Facebook dengan berfokus pada kesederhanaan dalam menulis pengujian unit (*unit testing*) (da Costa, 2021). Jest digunakan untuk melakukan pengujian unit secara otomatis dengan membuat skenario pengujian dan memastikan kebenaran dari basis kode program pada komponen maupun fungsi. Menurut situs resmi dari Jest yaitu *jestjs.io*, Jest dapat digunakan pada *project* Babel, Typescript, Node, React, Angular dan Vue.

2.14 *Black-box Testing*

Black-box testing adalah metode pengujian yang dilakukan dengan mengeksekusi data uji secara fungsional pada sistem untuk mengecek apakah sistem bekerja dengan baik atau tidak, tanpa mengetahui struktur dari kode programnya (Fakhrun and Gumilang, 2018). Pengujian *Black-box* bertujuan untuk mencari kesalahan yang mungkin terdapat dalam sistem seperti fungsi yang salah, kesalahan antarmuka, kesalahan struktur data atau akses *database*, kesalahan performa dan kesalahan inisialisasi atau terminasi (Putra, 2019). Pengujian *Black-box* dapat dilakukan dengan menjalankan kode program dan menguji fungsionalitas sistem yang ditampilkan secara langsung.

2.15 *User Experience Questionnaire (UEQ)*

UEQ adalah metode pengujian *user experience* dari sebuah sistem. Pengujian ini dilakukan untuk menguji sistem apakah sistem yang dibuat telah memenuhi kebutuhan dari *end user* (Wahyudi, 2021). Pengujian sistem menggunakan UEQ dapat membantu pengembang untuk mengetahui kekurangan yang ada pada sistem sehingga nantinya dapat melakukan perbaikan terhadap sistem yang telah dibuat. Pengujian UEQ menggunakan ketentuan skala *seven-stage* untuk mengurangi bias tendensi pada butir pertanyaan yang diajukan, contoh:

attractive o o o o o o o unattractive

Setiap pertanyaan memiliki 7 lingkaran yang menandakan penilaian responden terhadap sistem dengan skala +3 ke -3. Penilaian +3 menandakan jawaban yang positif, 0 menandakan jawaban netral, dan -3 menandakan jawaban yang negatif.

Pengujian UEQ berisi 6 aspek kategori dengan 26 pertanyaan (Schrepp, 2019), yakni :

1. *Attractiveness* (daya tarik), kesan pengguna terhadap produk.
2. *Perspiciuity* (kejelasan), kemudahan pengguna saat menggunakan produk.
3. *Effeciency* (efesiensi), interaksi antar pengguna dan produk yang dilakukan dengan cepat dan efisien.
4. *Dependability* (ketepatan), perasaan pengguna bahwa pengguna memiliki kendali saat berinteraksi dengan produk.
5. *Stimulation* (stimulasi), ketertarikan dan kesenangan yang didapatkan saat menggunakan produk.
6. *Novelty* (kebaruan), kreativitas dan inovasi pada produk.

Berikut 26 pertanyaan yang diajukan kepada responden untuk pengujian UEQ.

	1	2	3	4	5	6	7			
menyusahkan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menyenangkan	1	Scale
tak dapat dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat dipahami	2	Daya tarik
kreatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	monoton	3	Kejelasan
mudah dipelajari	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sulit dipelajari	4	Kebaruan
bermanfaat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	kurang bermanfaat	5	Kejelasan
membosankan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mengasyikkan	6	Stimulasi
tidak menarik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menarik	7	Stimulasi
tak dapat diprediksi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat diprediksi	8	Stimulasi
cepat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lambat	9	Ketepatan
berdaya cipta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	konvensional	10	Efisiensi
menghalangi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mendukung	11	Kebaruan
baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	buruk	12	Ketepatan
rumit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sederhana	13	Daya tarik
tidak disukai	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menggembirakan	14	Kejelasan
lazim	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	terdepan	15	Daya tarik
tidak nyaman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nyaman	16	Kebaruan
aman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak aman	17	Daya tarik
memotivasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak memotivasi	18	Ketepatan
memenuhi ekspektasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak memenuhi ekspektasi	19	Stimulasi
tidak efisien	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efisien	20	Ketepatan
jelas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	membingungkan	21	Efisiensi
tidak praktis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	praktis	22	Kejelasan
terorganisasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	berantakan	23	Efisiensi
atraktif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak atraktif	24	Daya tarik
ramah pengguna	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak ramah pengguna	25	Daya tarik
konservatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inovatif	26	Kebaruan

Gambar 2.6 Daftar Pertanyaan UEQ
Sumber : (Kharis, Santosa and Winarno, 2016)

Hasil jawaban dari responden pada setiap pertanyaan akan dikalkulasikan dan ditentukan nilai rata-ratanya (*mean*). Selanjutnya, nilai *mean* dari setiap pertanyaan akan diklasifikasikan berdasarkan 6 aspek kategori dan dihitung kembali nilai *mean* dari setiap kategori tersebut. Interpretasi standar untuk nilai evaluasi berdasarkan nilai-nilai *mean* adalah, jika nilai *mean* >0.8 maka diklasifikasikan sebagai nilai evaluasi positif, jika nilai *mean* diantara -0.8 dan 0.8 maka diklasifikasikan sebagai nilai evaluasi normal dan jika nilai <-0.8 maka diklasifikasikan sebagai nilai evaluasi negatif (Schrepp, 2019).

Kemudian hasil *mean* yang telah didapatkan dari setiap skala kategori direpresentasikan kedalam *diagram Benchmark*. Untuk memahami makna dari nilai yang dihasilkan dari setiap skala UEQ, dilakukan perbandingan nilai hasil perhitungan skala tersebut dengan nilai yang tercantum dalam Tabel 2.3 (Schrepp, Hinderks and Thomaschewski, 2017).

Tabel 2.3 *Benchmark interval* untuk skala UEQ

	<i>Att.</i>	<i>Eff.</i>	<i>Per.</i>	<i>Dep.</i>	<i>Sti.</i>	<i>Nov.</i>
<i>Excellent</i>	≥ 1.75	≥ 1.78	≥ 1.9	≥ 1.65	≥ 1.55	≥ 1.4
<i>Good</i>	≥ 1.52	≥ 1.47	≥ 1.56	≥ 1.48	≥ 1.31	≥ 1.05
	< 1.75	< 1.78	< 1.9	< 1.65	< 1.55	< 1.4
<i>Above average</i>	≥ 1.17	≥ 0.98	≥ 1.08	≥ 1.14	≥ 0.99	≥ 0.71
	< 1.52	< 1.47	< 1.56	< 1.48	< 1.31	< 1.05
<i>Below average</i>	≥ 0.7	≥ 0.54	≥ 0.64	≥ 0.78	≥ 0.5	≥ 0.3
	< 1.17	< 0.98	< 1.08	< 1.14	< 0.99	< 0.71
<i>Bad</i>	< 0.7	< 0.54	< 0.64	< 0.78	< 0.5	< 0.3

2.16 Penelitian Terkait

Penelitian yang terkait dengan peminjaman ruangan melibatkan berbagai teknologi yang digunakan, baik metode pengembangan sistem maupun bahasa pemrograman. Penelitian terkait dijadikan sebagai panduan ataupun contoh untuk penelitian yang dilakukan. Berikut adalah jurnal atau penelitian yang relevan dengan penelitian ini:

Pada penelitian yang dilakukan oleh (Supriady and Ginasta, 2022) membangun sebuah *website* yang digunakan untuk membantu *entrepreneur* atau *startup* dalam menyewa tempat kerja dengan konsep *coworking space* secara online melalui *web*. Pada aplikasi *web* ini juga memiliki fitur *booking tour* yang dapat melihat tempat kerja yang akan disewa untuk disesuaikan dengan kebutuhan dari penyewa. Aplikasi *web* ini dibangun menggunakan *framework* Laravel yang berbasis bahasa pemrograman PHP dan menggunakan metode *Incremental* untuk pengembangan sistemnya.

Penelitian yang dilakukan oleh (Herlambang *et al.*, 2020) membuat sistem informasi manajemen ruang rapat. Penelitian ini menggunakan metode pengembangan *V-Model* yaitu metode pengembangan sistem dari variasi model *waterfall* dengan tahapan digambarkan dalam bentuk V. Sistem ini dibangun menggunakan bahasa pemrograman

PHP dengan HTML dan Bootstrap serta basis data menggunakan MySQL. Hasil pengujian dari sistem ini menggunakan *Basis Path Testing* dan *Validation Testing* menunjukkan hasil yang *valid* dengan fungsionalitas sistem yang berjalan dengan baik.

Penelitian selanjutnya yang menggunakan teknologi *MERN Stack* adalah penelitian yang dilakukan oleh (Handoyo, Santoso and Setiawan, 2019) bertujuan untuk membangun *website* yang dapat digunakan untuk membantu *user* dalam melakukan pembuatan *diagram* BPMN dan dapat berkolaborasi dengan beberapa *user* dengan fitur yang ada seperti *sharing system*, *real-time diagram*, dan *real-time project*. Penelitian ini menggunakan teknologi *MERN Stack* untuk membantu dalam pembuatan sistem yang *realtime* antara *client* dan *server* sehingga perubahan data dapat dilakukan tanpa memuat ulang halaman *website* dengan menggunakan *library* WebSocket. Hasil penilaian terhadap sistem yang telah dibuat, bernilai standar 3 (cukup) dengan acuan yaitu sistem telah berjalan dengan baik dan memenuhi kebutuhan klien.

Penelitian lainnya yang menggunakan Teknologi *MERN Stack* oleh (Afif, 2021) membangun *website* agar dapat membantu *user* dalam menemukan tempat menginap saat berlibur dan memudahkan pemilik penginapan untuk mempromosikan tempatnya secara *online*. Dengan menggunakan teknologi *MERN Stack* *website* yang dibuat dapat lebih cepat dalam mengolah dan pertukaran data antar *client* dan *server*. Metode *Extreme Programming* (XP) dipilih supaya dapat dikerjakan secara tim dan dapat menyesuaikan kebutuhan *user*. Hasil dari pengujian *Black-box* terhadap sistem menunjukkan hasil yang sesuai dengan acuan terpenuhinya kebutuhan pengguna.

Penelitian selanjutnya yang menggunakan metode PXP dilakukan oleh (Ardianzah, Nuryasin and Wiyono, 2022) dengan membangun sistem pengelolaan peminjaman auditorium yang bertujuan untuk mempermudah dalam proses peminjaman. Sistem ini dibangun menggunakan *framework* Laravel dan MySQL sebagai basis data. Metode PXP dipilih agar dapat digunakan pada *programmer* tunggal supaya pengerjaan sistem lebih cepat dan mudah serta sesuai dengan kebutuhan yang ada. Hasil dari penelitian ini

adalah PXP telah menghasilkan sistem yang sesuai dengan kebutuhan klien walaupun terdapat perubahan pada saat proses pengembangannya.

Penelitian lainnya dengan metode PXP dilakukan oleh (Kurniawan and Eriana, 2022) membangun sistem informasi yang digunakan sebagai media promosi serta reservasi restoran. Penelitian ini menggunakan bahasa pemrograman Java dan PHP dengan MYSQL sebagai basis data. Metode pengembangan sistem yang dipilih adalah PXP yang memiliki kelebihan dalam efisiensi waktu pembuatan sistem yang sesuai dengan kebutuhan *user*. Hasil dari pengujian sistem menggunakan *Black-box testing* adalah sistem dapat berjalan dengan baik dan tidak ada kesalahan pada program sistem tersebut.

Berdasarkan penelitian (Ardianzah, Nuryasin and Wiyono, 2022) dan (Kurniawan and Eriana, 2022) menyatakan bahwa metode pengembangan PXP adalah metode yang tepat untuk pengembang tunggal sehingga dapat mengefisiensikan waktu dalam pembuatan sistem dan menghasilkan sistem yang sesuai dengan kebutuhan klien. Kemudian pada penelitian (Handoyo, Santoso and Setiawan, 2019) dan (Afif, 2021) menyatakan bahwa dengan menggunakan teknologi *MERN Stack*, pembuatan sistem dapat lebih cepat dalam mengolah dan pertukaran data antar *client* dan *server*. Teknologi tersebut menggunakan bahasa pemrograman Javascript dalam pengembangan *full stack* baik pengembangan *client* dan *server* dalam mengelola *database*. Oleh karena itu, penelitian ini yang hanya dilakukan oleh satu orang pengembang serta mengembangkan sisi *client* dan *server*, diputuskan menggunakan metode pengembangan PXP dan teknologi *MERN Stack* untuk membuat Sistem Peminjaman *Coworking Space* pada Gedung Lembaga Kemahasiswaan FISIP Unila.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

Penelitian ini dilakukan di laboratorium Teknik Komputer Universitas Lampung dan Gedung Lembaga Kemahasiswaan Fakultas Ilmu Sosial dan Ilmu Politik Universitas Lampung (FISIP Unila). Waktu penelitian dimulai dari bulan Desember 2022 sampai dengan bulan Juni 2023.

Tabel 3.1 Tabel waktu penelitian

No	Aktivitas	2022	2023					
		Des	Jan	Feb	Mar	Apr	Mei	Jun
1	<i>Requirement</i>	■						
2	<i>Planning</i>	■						
3	<i>Iteration Initialization</i>		■	■				
4	<i>Design</i>		■	■				
5	<i>Implementation</i>		■	■	■			
6	<i>Testing</i>		■	■	■			
7	<i>Retrospective</i>		■	■	■			
8	UEQ					■	■	
9	Pelaporan				■	■	■	■

3.2 Alat dan Bahan

Hardware (perangkat keras) dan *software* (perangkat lunak) yang digunakan untuk penelitian ini adalah sebagai berikut:

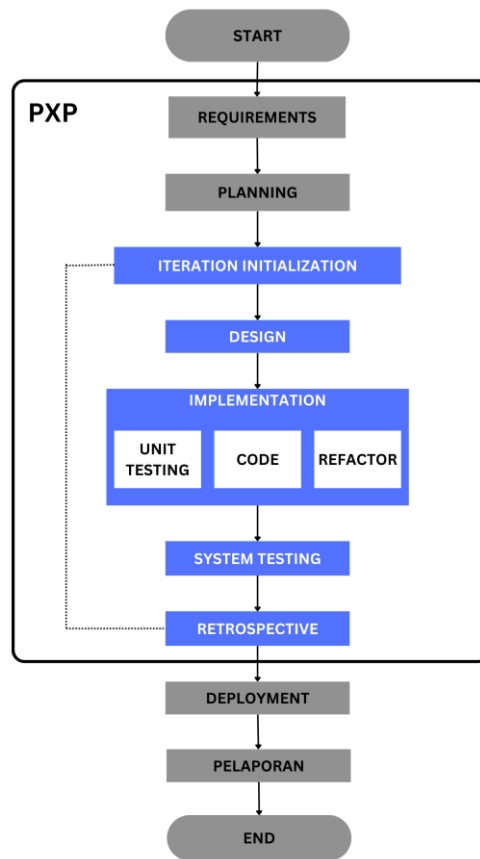
Tabel 3.2 Alat dan Bahan

No	Perangkat	Spesifikasi	Kegunaan
1	Laptop	Intel Celeron, RAM 8GB, HDD 500GB	Perangkat pembuat dan pengujian aplikasi.
2	ReactJS	Versi 18.2.0	<i>Library</i> pemrograman untuk pembuatan bagian <i>frontend website</i> .
3	Javascript	Javascript ES6	Bahasa Pemrograman untuk memberikan logika pada sistem.
4	NodeJS	Versi 16.14.0	<i>Runtime Environment</i> Javascript untuk menjalankan kode Javascript dimana saja.
5	ExpressJS	Versi 4.18.2	<i>Framework</i> NodeJS untuk mengatur <i>routing</i> API dan pertukaran data.
6	MongoDB Atlas	Penyimpanan 512 MB	Layanan <i>multi-cloud database</i> dari MongoDB
7	Cloudinary	Cloudinary <i>Web</i>	Layanan untuk menyimpan data foto
8	Git & Github	Versi 2.33.0 & Github <i>Web</i>	Menyimpan <i>source code</i> dari perubahan sistem selama pembuatan.
9	Node Package Manager (NPM)	Versi 6.14.17	Sebagai <i>Package Manager</i> untuk menginstall <i>library</i> .

10	Visual Studio Code	Versi 1.71.2	Perangkat lunak untuk menyunting kode program sistem
11	Figma	Figma <i>Web</i>	Perangkat lunak untuk mendesain UI sistem
13	Postman	Versi 10.12.4	Perangkat lunak untuk melakukan pengujian API
14	<i>Browser</i> Google Chrome	Versi 105.0.5195.127	Perangkat lunak untuk melakukan pengujian sistem
15	Microsoft Excel	2013 versi 15.0.4569.1506	Perangkat lunak yang digunakan sebagai <i>data analysis tool</i> untuk mengolah hasil pengujian <i>User Experience Questionnaire</i> (UEQ).

3.3 Tahapan Penelitian

Tahapan penelitian yang dilakukan pada penelitian ini terdiri dari beberapa tahapan yaitu, tahapan *requirements, planning, iteration initialization, design, implementation, system testing, retrospective, deployment* dan penulisan laporan. Tahapan pengembangan sistem pada perancangan Sistem Peminjaman *Coworking Space* adalah metode *Personal Extreme Programming* (PXP). Diagram alir penelitian ini dapat dilihat pada Gambar 3.1



Gambar 3.1 Tahapan penelitian

3.4 Tahap Pengembangan Sistem

Metode yang digunakan untuk merancang Sistem Peminjaman *Coworking Space* adalah metode *Personal Extreme Programming* (PXP). Metode ini merupakan salah satu metode dari pendekatan *Agile* yang memiliki 7 tahapan utama yaitu *requirements*, *planning*, *iteration initialization*, *design*, *implementation*, *system testing* dan *retrospective*. Pada metode ini pembangunan sistem dipecah menjadi beberapa proses kecil yang disebut dengan iterasi. Metode ini bersifat fleksibel yaitu dapat menanggapi perubahan yang ditemukan ketika pengerjaan sistem untuk perbaikan pada iterasi selanjutnya. Tahapan yang dilakukan pada pengembangan sistem menggunakan metode PXP adalah sebagai berikut:

3.4.1 Tahap *Requirements* (Pengumpulan Kebutuhan)

Pada tahap ini dilakukan pengumpulan data dengan menggunakan teknik wawancara. Wawancara dilakukan kepada pihak pengelola *Coworking Space* di Gedung Lembaga Kemahasiswaan FISIP Unila agar mengetahui permasalahan yang dialami ketika melakukan peminjaman ruangan.

Tabel 3.3 Tabel Pertanyaan Wawancara

No	Pertanyaan
1	Bagaimana pengelolaan dan alur dari peminjaman ruangan <i>coworking space</i> saat ini?
2	Apa saja kendala yang ditemukan ketika melakukan peminjaman ruangan?
3	Jika membuat sistem informasi apakah nantinya dapat mempermudah peminjaman ruangan?
4	Data apa saja yang dapat dimasukkan untuk pengelolaan ini?

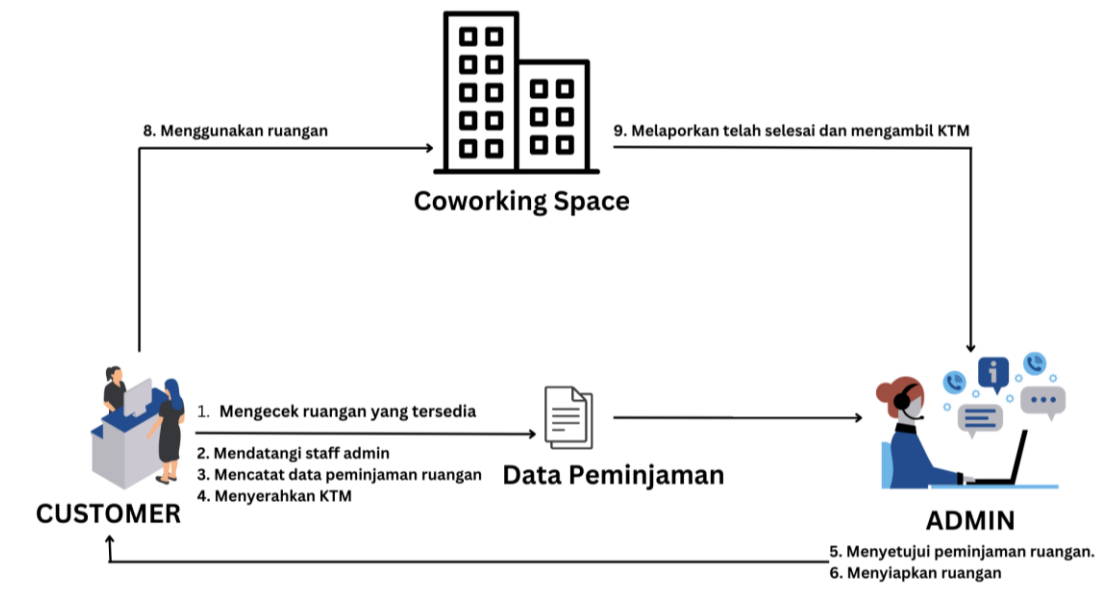
Tabel 3.4 Tabel Jawaban Wawancara

No	Jawaban
1	Saat ini pengelolaan peminjaman ruangan masih menggunakan cara manual dengan menulis dibuku yang telah disediakan untuk data peminjam. Kemudian untuk alur peminjaman mendatangi secara langsung <i>admin</i> yang ada di ruangan.
2	Untuk kendala peminjaman yang sering terjadi adalah bentrok waktu peminjaman sehingga harus melakukan peminjaman ruangan 3 hari sebelum menggunakannya. Namun jika ruangan kosong dapat langsung memakainya.
3	Jika terdapat sistem yang mengelola peminjaman ini saya harap dapat mempermudah dalam peminjaman ruangan karena peminjam dapat mengetahui informasi ruangnya terlebih dahulu dan dapat melakukan peminjaman tanpa harus mendatangi <i>admin</i> secara langsung. Kemudian untuk pengelolaan datanya menjadi lebih mudah jika sudah menjadi sistem.
4	Untuk data yang dibutuhkan seperti data peminjam, ruangan, dan daftar peminjaman

Setelah melakukan wawancara kepada pihak pengelola, selanjutnya dilakukan analisis pada sistem yang sedang berjalan dan sistem yang akan diusulkan, yaitu sebagai berikut:

a. Sistem Yang Sedang Berjalan

Sistem yang sedang berjalan saat ini pada peminjaman *coworking space* adalah *customer* harus mendatangi *admin* yang ada di ruangan dekat *coworking space* dengan memberikan kartu identitas seperti Kartu Tanda Mahasiswa (KTM) untuk meminjam ruangan. Setelah *customer* mencatat data peminjaman, *admin* akan mengkonfirmasi peminjaman dan memberikan kunci dari ruangan tersebut.

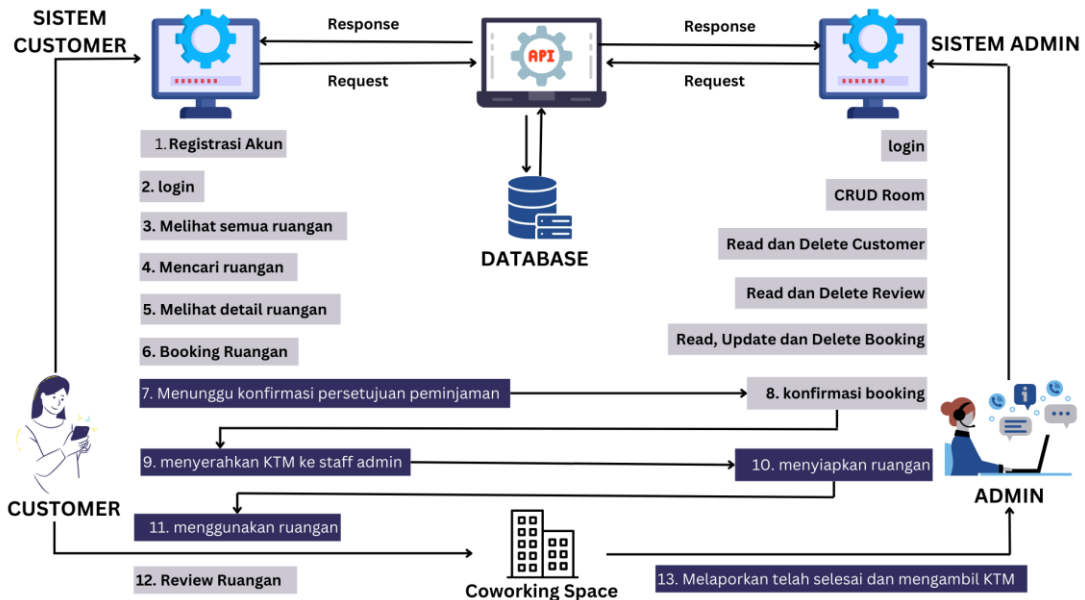


Gambar 3.2 Sistem yang sedang berjalan

b. Sistem Yang Diusulkan

Setelah melakukan analisis terhadap sistem yang sedang berjalan, ditemukan beberapa permasalahan yang perlu diperbaiki. Sebagai solusi, diusulkan untuk membangun sistem yang dapat melakukan pemesanan ruangan dan pengelolaan data secara *online*. Sistem yang diusulkan memiliki 2 aktor yaitu *admin* dan *customer*. Aktor *admin* memiliki hak akses untuk melakukan pengelolaan data ruangan, data *customer*, data *review*, data peminjaman, *login*, dan menyetujui

peminjaman. Sedangkan aktor *customer* memiliki hak akses untuk melakukan *register* dan *login* kemudian melihat semua data ruangan, mencari ruangan, melihat detail ruangan dan membuat pesanan peminjaman ruangan, riwayat peminjaman serta dapat melakukan *review* dari ruangan yang telah dipesan.



Gambar 3.3 Sistem yang diusulkan

Setelah melakukan analisis dari sistem yang sedang berjalan dan memberikan solusi atas permasalahan yang ada dengan mengusulkan Sistem Peminjaman *Coworking Space*, tahap selanjutnya yaitu membuat *user story* dan menentukan kebutuhan fungsional serta non-fungsional sebagai berikut :

1. *User Stories dan Acceptance Criteria*

User story merupakan tahapan awal yang digunakan untuk mendeskripsikan kebutuhan sistem dalam bentuk bahasa natural (manusia) sehingga dapat dipahami oleh *end user* yang tidak memiliki background IT. *Acceptance Criteria* merupakan pembuatan *product* yang sesuai dengan kebutuhan pengguna. Berikut *user stories* dari Sistem Peminjaman *Coworking Space*:

Tabel 3.5 *User story* sistem

Kode	User story	Acceptance Criteria
Admin		
US-01	Sebagai <i>admin</i> , saya ingin melakukan <i>login</i> sehingga saya dapat mengakses fitur-fitur khusus dan mengelola data.	Terdapat halaman <i>login</i> yang dapat diakses oleh <i>admin</i> dengan memasukkan <i>username</i> dan <i>password</i> yang valid.
US-02	Sebagai <i>admin</i> , saya ingin mengakses informasi ruangan sehingga saya bisa mengelola data ruangan.	Terdapat halaman kelola ruangan yang berisi tabel data untuk melihat informasi ruangan, fitur <i>form edit</i> , dan hapus data.
US-03	Sebagai <i>admin</i> , saya ingin mengakses informasi peminjaman sehingga saya bisa melakukan pengelolaan data.	Terdapat halaman kelola <i>booking</i> yang berisi tabel data untuk melihat informasi peminjaman, konfirmasi persetujuan peminjaman, fitur <i>form edit</i> dan hapus data peminjaman.
US-04	Sebagai <i>admin</i> , saya ingin mengakses informasi <i>customer</i> sehingga saya bisa melakukan menghapus data.	Terdapat halaman kelola <i>customer</i> yang berisi tabel data untuk melihat data <i>customer</i> , mengubah hak akses <i>user</i> menjadi <i>admin</i> , fitur hapus data.
US-05	Sebagai <i>admin</i> , saya ingin mengakses informasi <i>review customer</i> sehingga saya bisa menghapus data.	Terdapat halaman kelola <i>review</i> yang berisi tabel data untuk melihat informasi <i>review</i> dari ruangan yang telah di <i>booking</i> , fitur hapus data.
US-06	Sebagai <i>admin</i> , saya ingin dapat mengelola data profil saya.	Terdapat fitur <i>edit profile</i> dengan <i>form</i> yang berisi data <i>user</i> .
Customer		
US-07	Sebagai <i>customer</i> , saya ingin mencari sebuah ruangan sehingga saya dapat dengan mudah menemukan ruangan yang diinginkan.	Terdapat fitur <i>search room</i> untuk mencari ruangan sesuai dengan nama ruangan yang dicari.
US-08	Sebagai <i>customer</i> , saya ingin melihat semua daftar data ruangan yang ada sehingga saya dapat memilih ruangan yang sesuai dengan yang dibutuhkan.	Terdapat halaman ruangan yang berisi semua daftar data ruangan yang ada di dalam gedung.
US-09	Sebagai <i>customer</i> , saya ingin melihat informasi dan fasilitas apa saja yang ada di ruangan sehingga dapat menyesuaikan dengan kebutuhan yang diinginkan.	Terdapat informasi <i>detail</i> ruangan beserta fasilitas, foto dan <i>review</i> dari ruangan yang dipilih.

US-10	Sebagai <i>customer</i> , saya ingin melakukan registrasi akun sehingga data saya dapat tersimpan dan saya dapat dengan mudah mengakses fitur lain yang membutuhkan data pribadi.	Terdapat halaman registrasi yang meminta informasi yang diperlukan seperti nama lengkap, alamat <i>email</i> , pekerjaan, organisasi, nomor handphone, <i>username</i> dan <i>password</i> .
US-11	Sebagai <i>customer</i> , saya ingin melakukan <i>login</i> sehingga saya dapat mengakses fitur-fitur khusus.	Terdapat halaman <i>login</i> yang dapat diakses oleh <i>customer</i> dengan memasukkan <i>username</i> dan <i>password</i> yang valid.
US-12	Sebagai <i>customer</i> , saya ingin melakukan peminjaman ruangan melalui <i>website</i> sehingga saya dapat memakai ruangan sesuai dengan waktu yang dipesan tanpa adanya waktu bentrok.	Terdapat <i>form</i> untuk membuat pesanan peminjaman dengan pemilihan waktu agar terhindar dari bentrok.
US-13	Sebagai <i>customer</i> , saya ingin dapat mengelola data profil saya.	Terdapat fitur <i>edit profile</i> dengan <i>form</i> yang berisi data <i>user</i> .
US-14	Sebagai <i>customer</i> , saya ingin menulis <i>review</i> dari ruangan yang telah saya pesan sehingga saya dapat memberikan umpan balik terhadap ruangan tersebut.	Terdapat <i>form</i> untuk membuat <i>review</i> ruangan dari pesanan yang telah dibuat.
US-15	Sebagai <i>customer</i> , saya ingin melihat riwayat pemesanan	Terdapat fitur daftar riwayat pemesanan ruangan.

2. Analisis Kebutuhan

Pada tahap ini dilakukan pengumpulan kebutuhan dengan menentukan kebutuhan fungsional dan non fungsional dari Sistem Peminjaman *Coworking Space*. Berikut hasil analisis dan didapatkan kebutuhan fungsional dan non-fungsional pada sistem ini:

a. Kebutuhan Fungsional

Kebutuhan fungsional adalah proses atau layanan apa saja yang nantinya disediakan oleh sistem, berikut merupakan kebutuhan fungsional dari Sistem Peminjaman *Coworking Space*:

Tabel 3.6 Kebutuhan Fungsional Sistem

Role	Kebutuhan Fungsional
<i>Admin</i>	Melakukan <i>login</i>
	Mengelola data ruangan
	Mengelola data <i>customer</i>
	Mengelola data peminjaman
	Mengelola data <i>review</i>
	Melakukan <i>edit profile</i>
<i>Customer</i>	Melakukan <i>register</i>
	Melakukan <i>login</i>
	Mencari data ruangan
	Melihat semua data ruangan
	Melihat informasi ruangan
	Melakukan peminjaman ruangan
	Melihat riwayat peminjaman
	Melakukan <i>review</i> ruangan
	Melakukan <i>edit profile</i>

b. Kebutuhan Non-Fungsional

Kebutuhan non fungsional adalah batasan layanan yang ditawarkan oleh suatu sistem, berikut merupakan kebutuhan non fungsional dari Sistem Peminjaman *Coworking Space*:

Tabel 3.7 Kebutuhan Non Fungsional Sistem

Kebutuhan Non Fungsional	Deskripsi
<i>Availability</i>	Sistem dijalankan menggunakan koneksi internet dan berjalan sesuai dengan jam aktivitas pada gedung lembaga kemahasiswaan.
<i>Portability</i>	Sistem ini mampu berjalan diberbagai sistem operasi yang mampu menggunakan <i>web browser</i> .
<i>Security</i>	Pada sistem ini hanya <i>customer</i> yang dapat melakukan peminjaman ruangan, sedangkan pengelolaan data hanya dapat dilakukan oleh <i>admin</i> setelah kedua <i>user</i> tersebut terdaftar di dalam <i>database</i> sistem.
<i>Build</i>	Sistem dibangun menggunakan bahasa pemrograman Javascript dan teknologi MERN <i>Stack</i> yaitu MongoDB, ExpressJS, ReactJS, dan NodeJS

3.4.2 Tahap *Planning* (Perencanaan)

Pada tahap ini menentukan daftar tugas dengan estimasi waktu pengerjaannya berdasarkan daftar kebutuhan yang telah didapatkan. Daftar tugas pada pengembangan Sistem Peminjaman *Coworking Space* dapat dilihat pada Tabel 3.8 berikut.

Tabel 3.8 Daftar Tugas

No	Perencanaan Tugas	Deskripsi	Perkiraan Waktu
1	<i>Backend Server</i>	Membuat autentikasi <i>user</i> dan API <i>Create, Read, Update Delete</i> (CRUD)	2 minggu
2	Halaman <i>Admin</i>	Membuat halaman <i>login, dashboard, kelola ruangan, booking, review, customer, dan edit profile.</i>	3 minggu
3	Halaman <i>Customer</i>	Membuat halaman <i>register, login, landing page, tentang, all room, detail room, booking room, history booking, review booking dan edit profile.</i>	3 minggu

3.4.3 Tahap *Iteration Initialization* (Inisialisasi Iterasi)

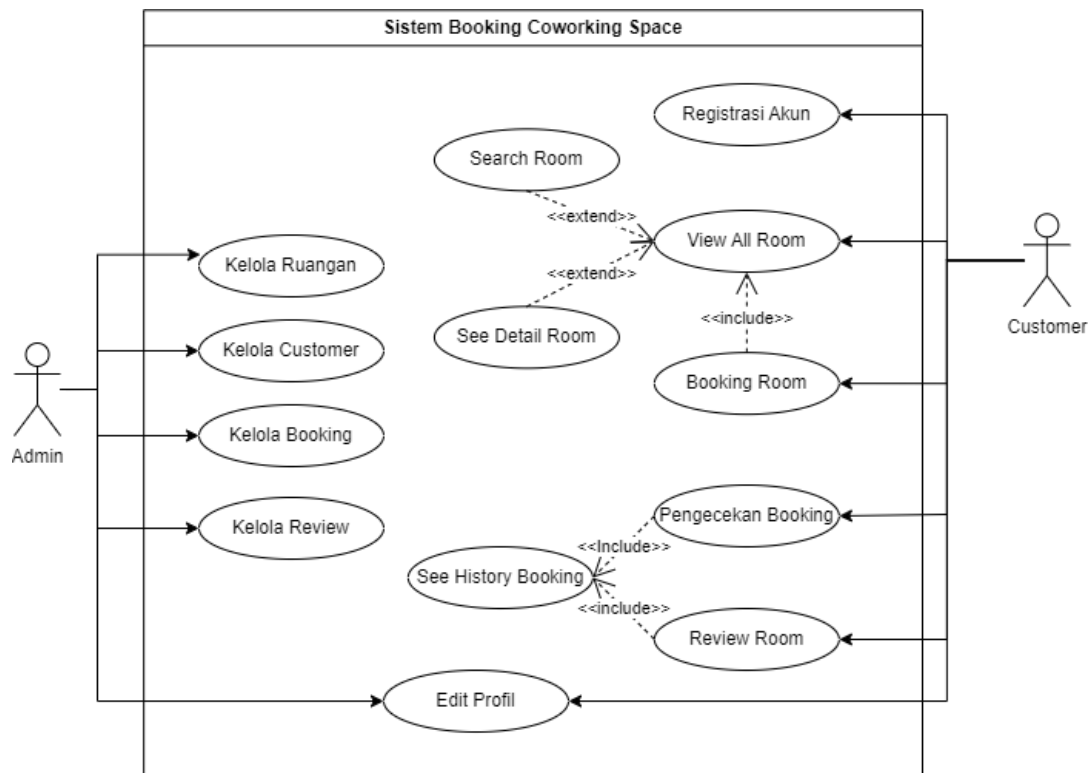
Pada tahap ini merupakan tahap awal dari setiap iterasi dengan melakukan pemilihan fokus tugas dan estimasi waktu yang ditentukan pada iterasi tersebut. Iterasi dilakukan sebanyak 3 kali dengan durasi waktu yang bervariasi. Iterasi pertama membangun *backend server* yaitu API CRUD data dan *authentication*. Kemudian iterasi kedua melakukan *slicing design* dari *design UI* yang diimplementasikan kedalam kode program dan integrasi API pada halaman *admin*. Selanjutnya, iterasi ketiga melakukan *slicing design* dan integrasi API pada halaman *customer*.

3.4.4 Tahap *Design* (Perancangan)

Pada tahap ini melakukan perancangan *diagram* UML, ERD dan perancangan antarmuka dalam Sistem Peminjaman *Coworking Space* menggunakan *tools* Figma untuk membuat tampilan *prototype* sistem yang akan diimplementasikan ke dalam sistem. Rancangan antarmuka ini dibuat berdasarkan pada *diagram* UML dan ERD yang telah dibuat. Tampilan antarmuka dibagi menjadi beberapa *menu*, pada *admin* terdapat *dashboard page*, *customer*, *room*, *booking*, *review*, *profile* dan *login*. Sedangkan pada *customer* terdapat *menu landing page*, *about*, *all room*, *detail room*, *edit profile*, *booking room*, *history booking* dan *review booking*. Berikut perancangan *diagram* UML yang terdiri dari *use case diagram* dan *activity diagram* :

1. *Use case Diagram*

Use case diagram adalah *diagram* yang digunakan untuk mengetahui kebutuhan fungsional yang ada didalam sistem serta menggambarkan interaksi dan hubungan antara aktor dan *use case*. *Diagram* ini membantu *user* dalam memahami inti dasar dari sistem yang dibangun. *Use case diagram* pada Sistem Peminjaman *Coworking Space* terdiri dari 2 aktor yaitu *admin* dan *customer*. Kemudian memiliki *use case* yang menjelaskan aktivitas aktor didalam sistem. Pada *use case view all room* memiliki fungsionalitas tambahan yaitu *search room* dan *see detail room*, sehingga kedua *use case* tersebut berelasi <<*extend*>> ke *use case view all room*. Sedangkan untuk menjalankan *use case booking room* diwajibkan untuk terlebih dahulu menjalankan *use case view all room*, oleh karena itu kedua *use case* ini berelasi <<*include*>>. *Use case diagram* dari Sistem Peminjaman *Coworking Space* dapat dilihat pada Gambar 3.4 berikut.



Gambar 3.4 Use case diagram dari Sistem Peminjaman Coworking Space

Berikut penjelasan mengenai aktor dan use case pada use case diagram Sistem Peminjaman Coworking Space.

Tabel 3.9 Definisi Aktor

Aktor	Deskripsi
<i>Admin</i>	Aktor <i>Admin</i> ditujukan kepada pihak pengelola ruangan yang memiliki hak akses untuk mengelola data ruangan, peminjaman, <i>customer</i> dan <i>review</i> serta <i>edit profile</i> .
<i>Customer</i>	Aktor <i>Customer</i> ditujukan kepada mahasiswa, dosen dan staf di lingkungan Unila yang memiliki hak akses untuk registrasi akun, melihat semua ruangan, melihat detail ruangan, mencari ruangan, <i>booking</i> ruangan, pengecekan status <i>booking</i> , melihat <i>history booking</i> , melakukan <i>review</i> ruangan dan <i>edit profile</i> .

Tabel 3.10 Definisi *Use case*

<i>Use case</i>	Deskripsi
Kelola Ruangan	Aktivitas ini dilakukan oleh <i>admin</i> untuk proses melihat, menambah, mengedit dan menghapus data ruangan. Precondition : telah melakukan autentikasi (<i>login</i>)
Kelola <i>Booking</i>	Aktivitas ini dilakukan oleh <i>admin</i> untuk proses melihat, mengkonfirmasi, mengedit dan menghapus data peminjaman. Precondition : telah melakukan autentikasi (<i>login</i>)
Kelola <i>Customer</i>	Aktivitas ini dilakukan oleh <i>admin</i> untuk proses melihat, mengubah hak akses <i>user</i> menjadi <i>admin</i> dan menghapus data <i>customer</i> . Precondition : telah melakukan autentikasi (<i>login</i>)
Kelola <i>Review</i>	Aktivitas ini dilakukan oleh <i>admin</i> untuk proses melihat dan menghapus data <i>review</i> . Precondition : telah melakukan autentikasi (<i>login</i>)
<i>Edit Profile</i>	Aktivitas ini dilakukan <i>admin</i> dan <i>customer</i> untuk mengubah data akun. Precondition : telah melakukan autentikasi (<i>login</i>)
Registrasi Akun	Aktivitas ini dilakukan oleh <i>customer</i> untuk mendaftar sebagai <i>user</i> tetap.
<i>Search Room</i>	Aktivitas ini dilakukan <i>customer</i> untuk proses mencari ruangan berdasarkan nama.
<i>View All Room</i>	Aktivitas ini dilakukan <i>customer</i> untuk proses melihat semua ruangan yang tersedia.
<i>See Detail Room</i>	Aktivitas ini dilakukan <i>customer</i> untuk melihat <i>detail</i> informasi dari masing-masing ruangan.
<i>Booking Room</i>	Aktivitas ini dilakukan <i>customer</i> dengan proses pengisian <i>form</i> data peminjaman ruangan. Precondition : telah melakukan autentikasi (<i>login</i>)
<i>See History Booking</i>	Aktivitas ini dilakukan <i>customer</i> untuk melihat riwayat peminjaman ruangan yang telah dilakukan. Precondition : telah melakukan autentikasi (<i>login</i>)
Pengecekan <i>booking</i>	Aktivitas ini dilakukan <i>customer</i> untuk melihat status peminjaman ruangan apakah <i>pending</i> , <i>approved</i> atau <i>rejected</i> . Precondition : telah melakukan autentikasi (<i>login</i>)
<i>Review Room</i>	Aktivitas ini dilakukan <i>customer</i> untuk proses <i>review</i> dari ruangan yang telah dipinjam. Precondition : telah melakukan autentikasi (<i>login</i>)

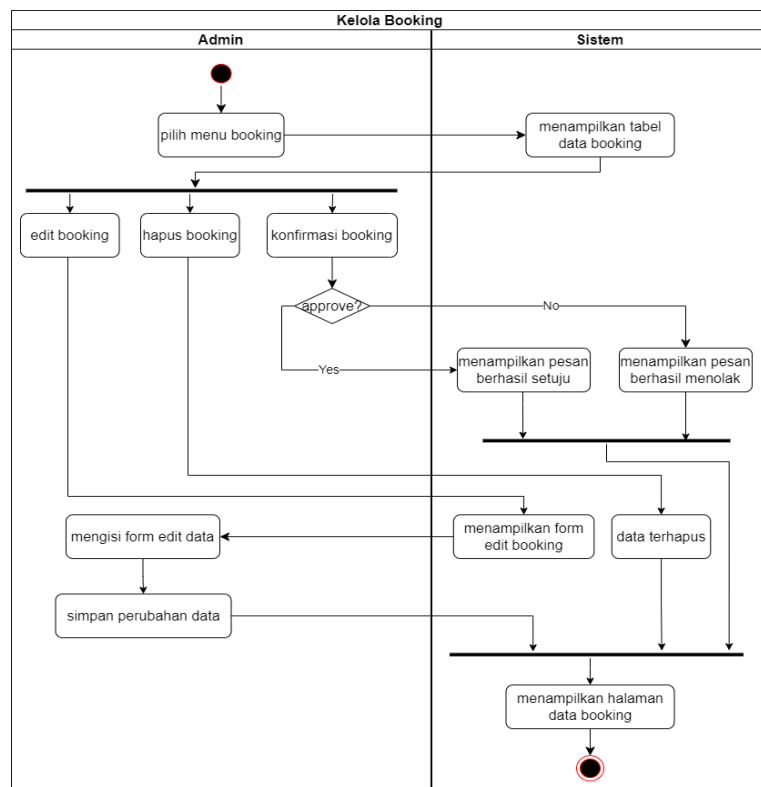
2. Activity Diagram

Activity diagram adalah *diagram* yang digunakan untuk menggambarkan aktivitas aktor dari awal hingga akhir pada suatu proses dalam menjalankan sistem. Berikut merupakan perancangan *activity diagram* pada Sistem Peminjaman *Coworking Space*.

1. Activity Diagram Admin

a. Activity Diagram Kelola Booking

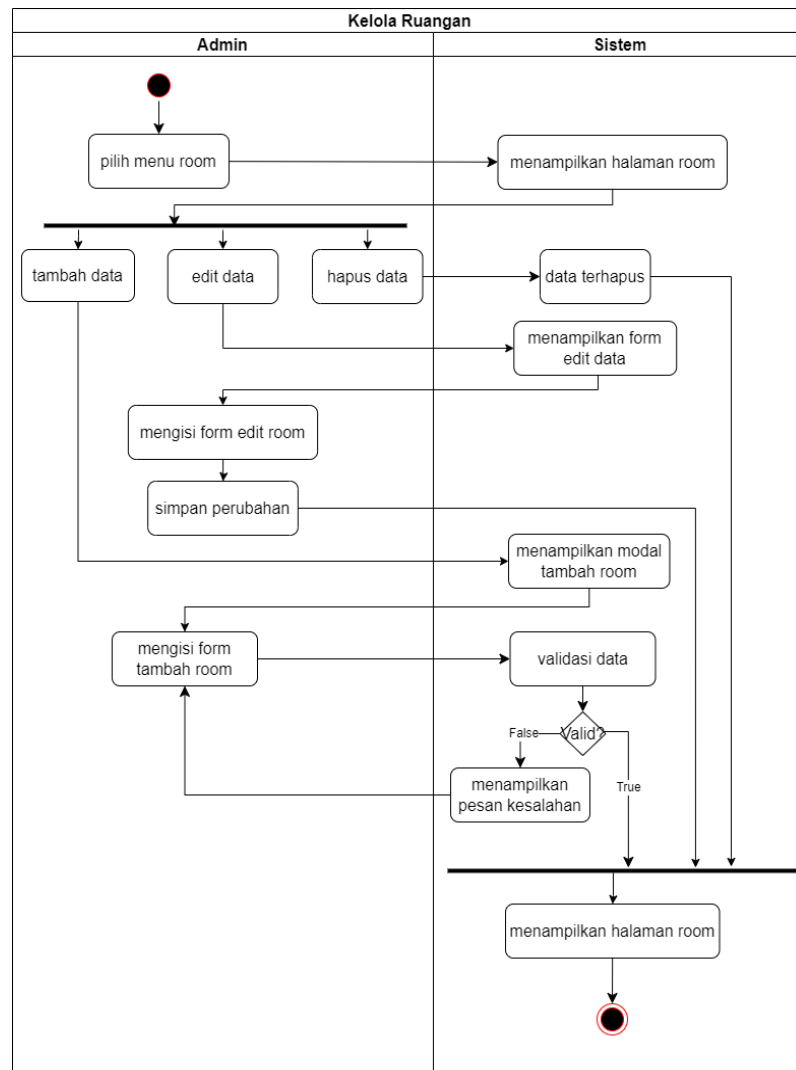
Activity diagram kelola *booking* menjelaskan alur proses aktivitas yang dilakukan sistem saat *admin* melakukan pengelolaan data *booking*. *Admin* dapat melakukan konfirmasi *booking* dengan memilih setuju atau menolak *booking*, jika menolak maka akan menampilkan pesan berhasil menolak dan jika setuju maka akan menampilkan pesan berhasil setuju, lalu keduanya akan kembali menampilkan halaman *booking*. *Admin* juga dapat melakukan *edit* data dengan cara mengisi *form edit* data, lalu menyimpan data tersebut kemudian *admin* juga dapat menghapus data. Setelah berhasil melakukan *edit* atau hapus data maka sistem akan kembali menampilkan halaman data *booking*.



Gambar 3.5 Activity Diagram Admin Kelola Data Booking

b. *Activity Diagram* Kelola Ruangan

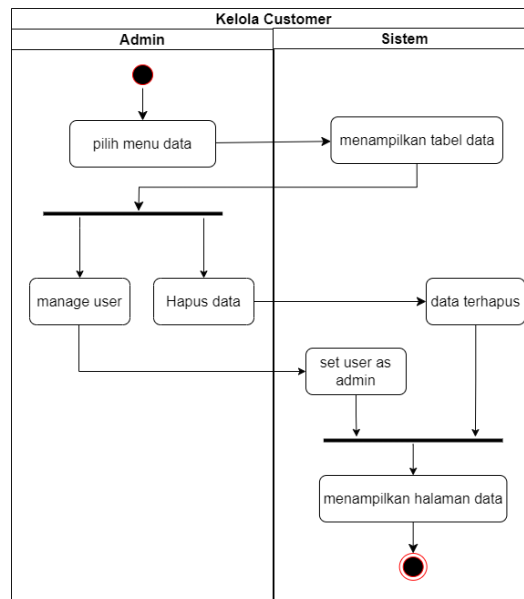
Activity diagram kelola ruangan menjelaskan alur proses aktivitas yang dilakukan sistem saat *admin* melakukan pengelolaan data ruangan. *Admin* dapat melakukan penambahan data dengan menginputkan data dan data akan divalidasi, jika data tidak valid maka menampilkan pesan kesalahan dan jika berhasil akan kembali menampilkan tabel data. *Admin* juga dapat melakukan *edit* data dengan cara mengisi *form edit* data, lalu menyimpan data tersebut kemudian *admin* juga dapat menghapus data. Setelah berhasil melakukan *edit* atau hapus data maka sistem akan kembali menampilkan halaman data ruangan.



Gambar 3.6 *Activity Diagram* Admin Kelola Data Ruangan

c. *Activity Diagram Kelola Customer*

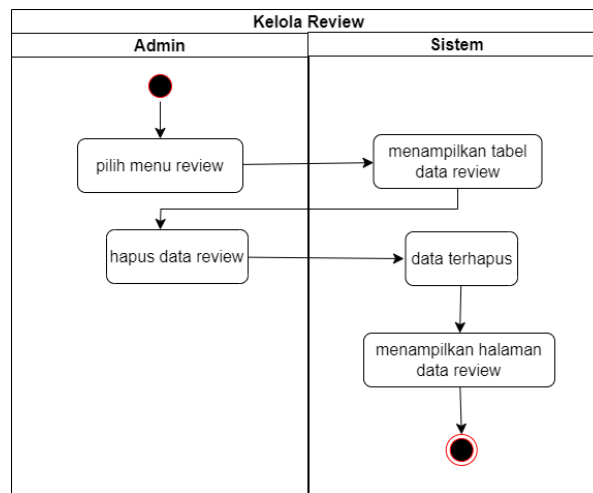
Activity Diagram Kelola Customer menjelaskan alur proses aktivitas yang dilakukan sistem saat *admin* melakukan pengelolaan data *customer*. *Admin* memulai proses dengan memilih menu *customer* kemudian sistem menampilkan tabel *customer*. Selanjutnya *admin* dapat melakukan penghapusan data dengan mengklik *button* hapus, lalu sistem akan menghapus data tersebut dari *database*. *Admin* juga dapat mengubah *customer* menjadi *admin* dengan memilih *manage user* dan sistem akan mengubah *user customer* menjadi *user admin*. Setelah berhasil, keduanya akan menampilkan halaman data *customer*



Gambar 3.7 *Activity Diagram Admin Kelola Data Customer*

d. *Activity Diagram Kelola Review*

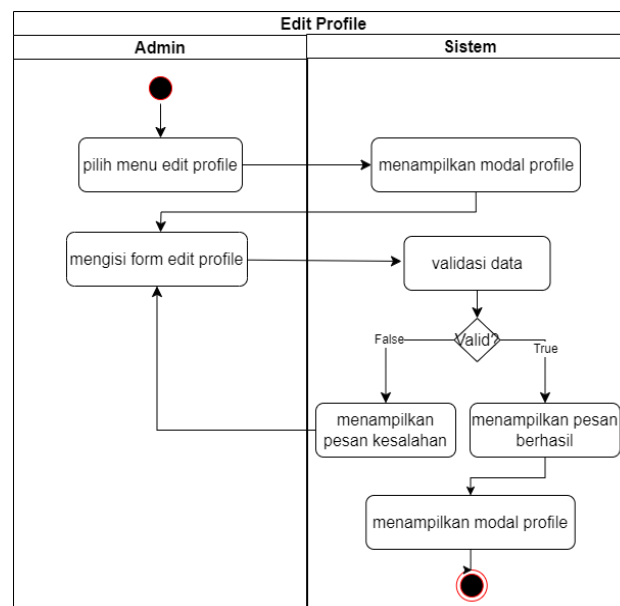
Activity Diagram Kelola Review menjelaskan alur proses aktivitas yang dilakukan sistem saat *admin* melakukan pengelolaan data *review*. *Admin* memulai proses dengan memilih menu *review* kemudian sistem menampilkan tabel *review*. Selanjutnya *admin* dapat melakukan penghapusan data dengan mengklik *button* hapus, lalu sistem akan menghapus data tersebut dari *database* dan setelah berhasil akan menampilkan kembali halaman tabel *review*.



Gambar 3.8 Activity Diagram Admin Kelola Data Review

e. Activity Diagram Edit Profile

Activity diagram edit profile menjelaskan alur proses aktivitas yang dilakukan sistem saat *admin* melakukan *edit profile*. *Admin* memulai proses dengan memilih menu *edit profile* kemudian sistem menampilkan *modal profile*. Selanjutnya *admin* dapat melakukan input data dan data akan divalidasi, jika data tidak valid maka menampilkan pesan kesalahan dan jika berhasil akan kembali menampilkan pesan berhasil *update data*.

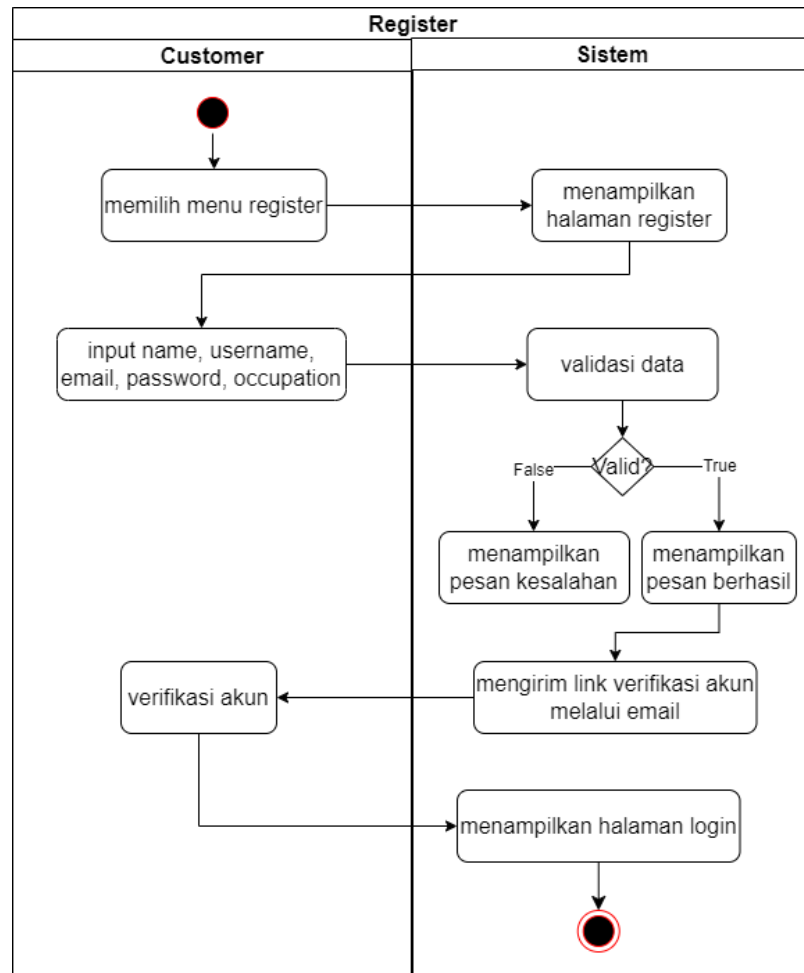


Gambar 3.9 Activity Diagram Admin Edit Profile

2. Activity Diagram Customer

a. Activity Diagram Register

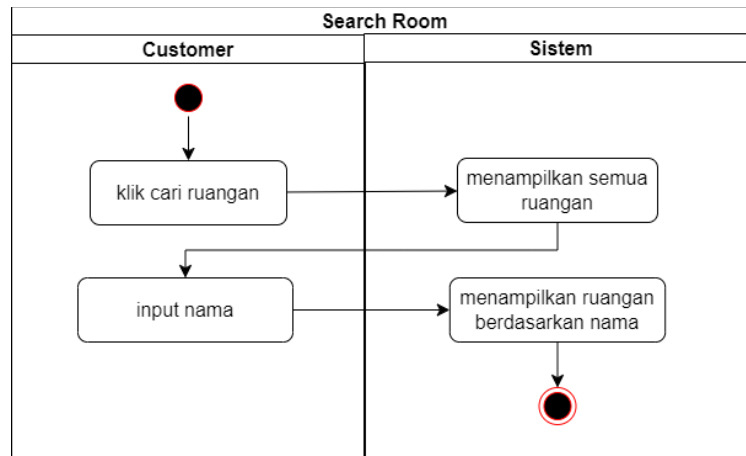
Activity diagram register menjelaskan alur proses aktivitas yang dilakukan sistem saat *customer* melakukan registrasi akun. Dimulai dengan *customer* memilih menu *register* kemudian sistem akan menampilkan halaman *register*. Selanjutnya *customer* dapat menginputkan data diri seperti nama, *username*, *email*, *password* serta *occupation*, dan sistem melakukan validasi data. Jika registrasi akun gagal maka sistem menampilkan pesan kesalahan, namun jika berhasil maka sistem mengirimkan *url* verifikasi akun melalui *email* dan *customer* dapat melakukan verifikasi akun. Setelah berhasil registrasi, sistem menampilkan halaman *login*.



Gambar 3.10 Activity Diagram Register Customer

b. *Activity Diagram Search Room*

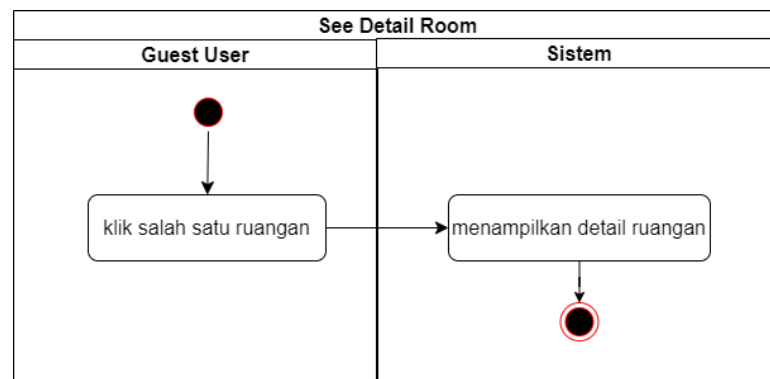
Activity diagram search room menjelaskan alur proses aktivitas yang dilakukan sistem saat *customer* mencari ruangan. Dimulai dengan *customer* memilih menu cari ruangan kemudian sistem menampilkan halaman yang berisi semua ruangan. Selanjutnya *customer* dapat menginputkan nama, lalu sistem menampilkan ruangan yang tersedia berdasarkan nama yang dicari *customer*.



Gambar 3.11 *Activity Diagram Customer Search Room*

c. *Activity Diagram See Detail Room*

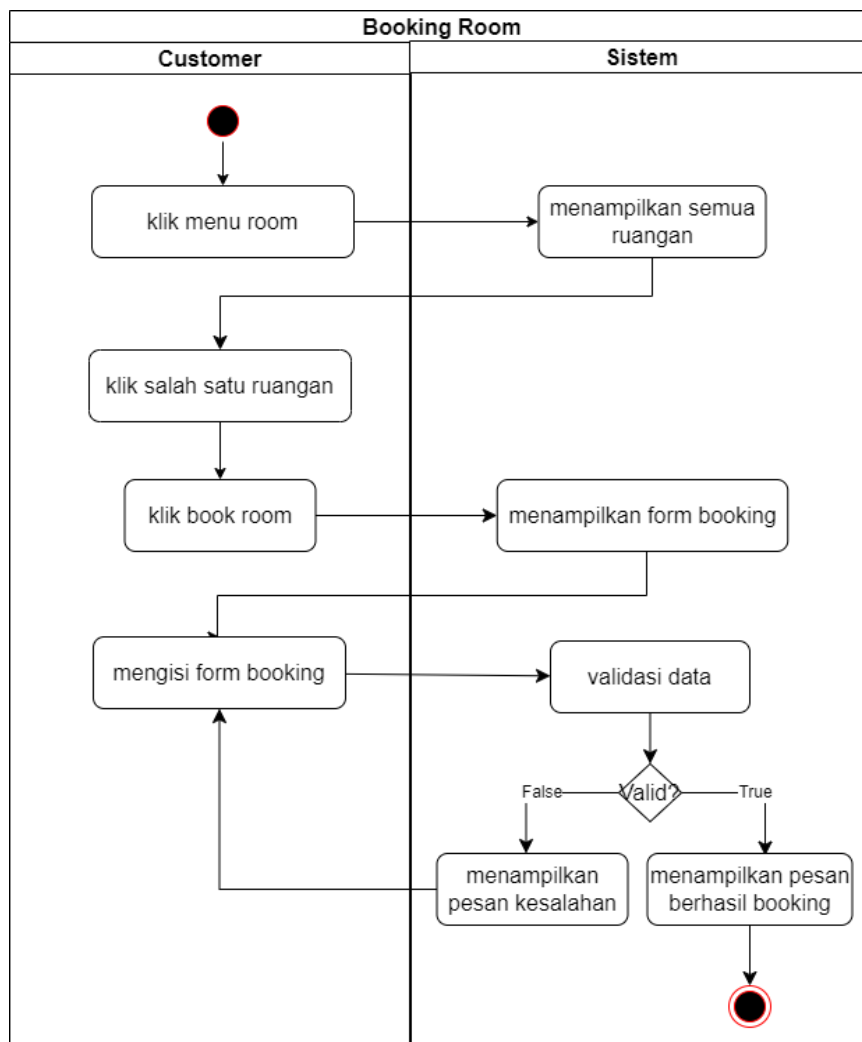
Activity diagram see detail room menjelaskan alur proses aktivitas yang dilakukan sistem saat *customer* melakukan aktivitas melihat detail ruangan. Dimulai dengan *customer* memilih salah satu ruangan, sistem akan menampilkan detail dari ruangan yang dipilih *customer*.



Gambar 3.12 *Activity Diagram Customer See Detail Room*

d. *Activity Diagram View All Room dan Booking Room*

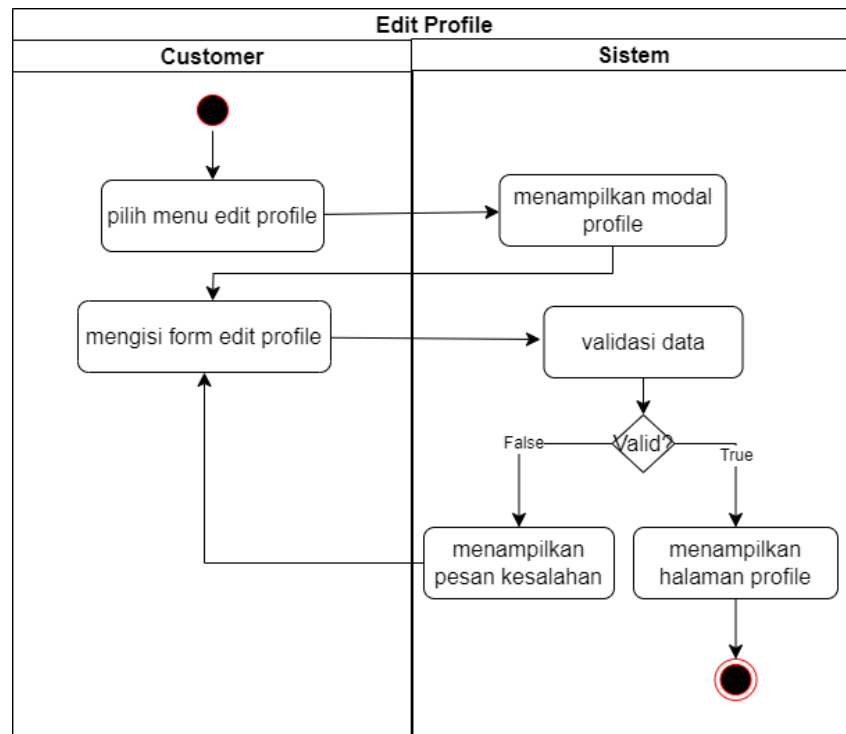
Activity diagram view all room dan booking room menjelaskan alur proses aktivitas yang dilakukan sistem saat *customer* melihat semua ruangan dan melakukan *booking room*. *Customer* dapat mengklik menu *room* dan sistem akan menampilkan semua data ruangan. Selanjutnya *customer* memilih salah satu ruangan dan mengklik *button book room*, lalu sistem menampilkan *form booking* ruangan. Selanjutnya *customer* melakukan *input* data terkait peminjaman ruangan lalu sistem akan melakukan validasi data, jika gagal maka sistem akan menampilkan pesan kesalahan, namun jika berhasil maka sistem akan menampilkan pesan berhasil melakukan *booking* ruangan.



Gambar 3.13 *Activity Diagram Customer View All Room dan Booking Room*

e. *Activity Diagram Edit Profile*

Activity diagram edit profile menjelaskan alur proses aktivitas yang dilakukan sistem saat *customer* melakukan *edit profile*. *Customer* memulai proses dengan memilih menu *edit profile* kemudian sistem menampilkan modal *profile*. Selanjutnya *customer* dapat mengisi form edit profile dan data akan divalidasi, jika data tidak valid maka menampilkan pesan kesalahan dan jika berhasil akan kembali menampilkan pesan berhasil *update* data.

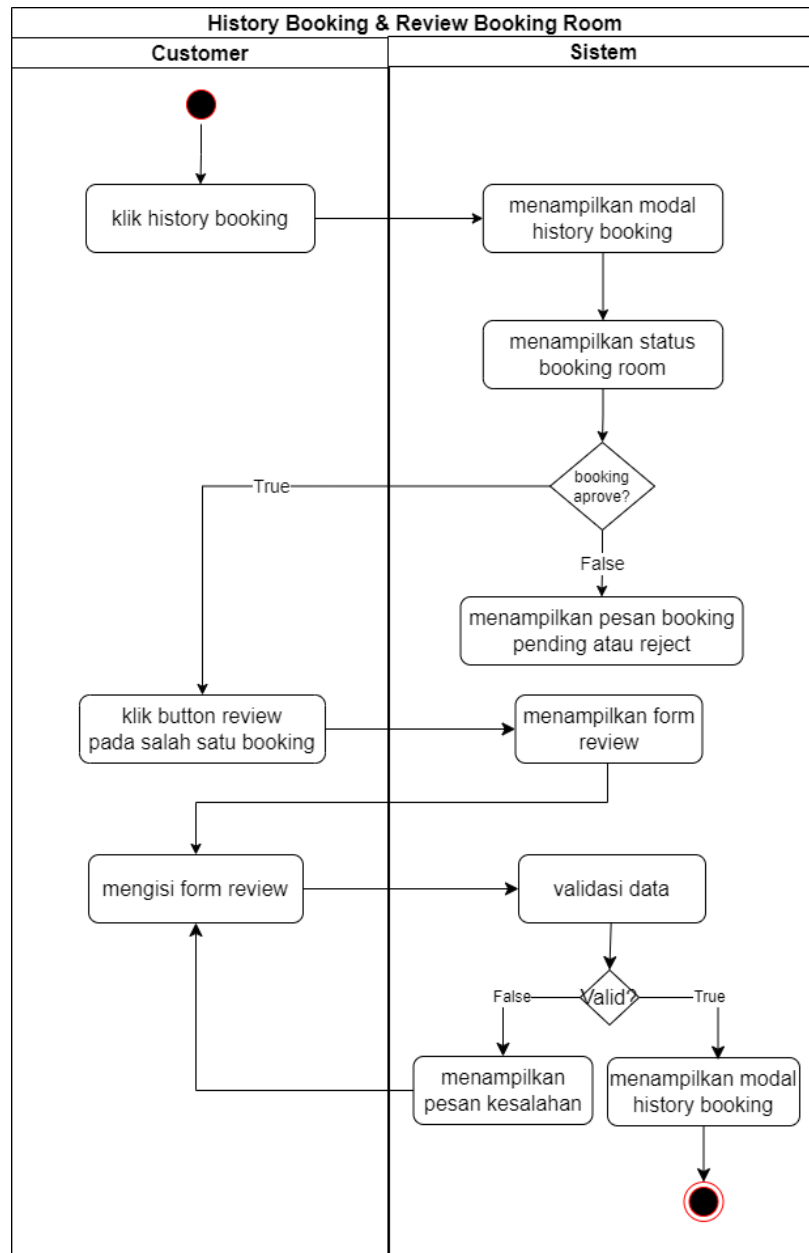


Gambar 3.14 *Activity Diagram Customer Edit Profile*

f. *Activity Diagram View History, Pengecekan dan Review dari Booking Room*

Activity diagram view history booking, pengecekan booking dan review booking room menjelaskan alur proses aktivitas yang dilakukan sistem saat *customer* melihat riwayat peminjaman ruangan. *Customer* memulai proses dengan memilih menu *history booking*, lalu sistem akan menampilkan modal *history booking* dan status dari peminjaman ruangan. Selanjutnya jika *booking* telah disetujui maka *customer* dapat melakukan *review* dengan cara memilih salah satu *booking* dan sistem akan menampilkan *form review*. Kemudian

customer dapat melakukan *input* data *review* dan sistem akan melakukan validasi data, jika data tidak valid maka menampilkan pesan kesalahan dan jika berhasil akan kembali menampilkan kembali *modal history booking*. Akan tetapi jika *booking* dalam status ditolak atau *pending*, maka akan menampilkan pesan *booking pending* atau *reject*.



Gambar 3.15 Activity Diagram Customer Melihat History, Pengecekan Status dan Review dari Booking

3.4.5 Tahap *Implementation* (Implementasi)

Pada tahapan ini yaitu melakukan *unit testing* untuk memastikan setiap komponen berfungsi dengan benar menggunakan *framework* Jest, kemudian pembuatan kode program keseluruhan sesuai dengan perancangan sistem yang telah dibuat dan melakukan *refactoring* kode program untuk membersihkan dan meningkatkan kualitas kode program. Untuk pengembangan sistem ini menggunakan bahasa pemrograman Javascript dengan teknologi MERN *Stack* yang terdiri dari MongoDB, ExpressJS, ReactJS dan NodeJS. Ketika terdapat perubahan pada sistem selama pembuatan, Github digunakan untuk menyimpan *source code* dari setiap kode program yang di *push* ke dalam Github sehingga dapat mempermudah untuk mengidentifikasi *bug*.

3.4.6 Tahap *System Testing* (Pengujian)

Pada tahap ini dilakukan pengujian dari sistem yang telah dibuat menggunakan metode *Black-box testing*. Proses pengujian dilakukan dengan 2 tahapan, yaitu pada bagian *frontend*, sistem diuji secara langsung pada *website*-nya apakah fitur yang dibangun berjalan sesuai dengan fungsi yang diharapkan pada analisis kebutuhan dan berjalan dengan baik atau tidak. Pengujian yang kedua yaitu pada bagian *backend*, sistem diuji menggunakan *tool* Postman apakah API yang telah dibuat telah sesuai dengan *request* dan *response* data yang diharapkan.

3.4.7 Tahap *Retrospective*

Pada tahap akhir dari setiap iterasi ini dilakukan introspeksi dalam pengerjaan iterasi berlangsung seperti kesesuaian estimasi waktu atau penyebab terjadinya keterlambatan pengerjaan sistem sehingga dapat melakukan peningkatan proses iterasi agar dapat lebih efektif pada iterasi berikutnya.

3.4.8 Tahap *Deployment*

Pada tahap ini melakukan perilisian sistem ke lingkungan produksi secara publik agar dapat diakses oleh pengguna sebenarnya. Setelah sistem dapat diakses secara publik, dilakukan pengujian pengalaman pengguna (*user experience*) menggunakan metode *User Experience Questionnaire* (UEQ) untuk mengukur kepuasan dan persepsi pengguna terhadap sistem yang telah dikembangkan. Metode UEQ digunakan untuk mengumpulkan data mengenai pengalaman pengguna melalui kuesioner yang mencakup aspek-aspek seperti daya tarik, kejelasan, efisiensi, ketepatan, stimulasi dan kebaruan terhadap sistem. Dengan melakukan pengujian UEQ, pengembang sistem dapat memperoleh umpan balik langsung dari pengguna yang digunakan sebagai dasar untuk melakukan perbaikan dan penyesuaian pada sistem. Hal ini bertujuan untuk meningkatkan pengalaman pengguna dan memastikan bahwa sistem telah memenuhi kebutuhan pengguna.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang telah didapatkan pada penelitian ini, terdapat kesimpulan yang dapat diambil diantaranya:

1. Proses pengembangan sistem pada penelitian ini menggunakan metode pengembangan PXP (*Personal Extreme Programming*) yang melalui 3 kali iterasi dengan sistem yang dihasilkan yaitu *backend server*, *web customer* dan *web admin*. Berdasarkan perhitungan efisiensi dari estimasi waktu terhadap waktu pengerjaan, metode PXP berhasil mencapai 93%. Hal ini membuktikan bahwa metode PXP efisien dalam merancang sebuah sistem dengan lingkup yang tidak besar dan dapat beranggotakan satu orang yang mengutamakan kebutuhan klien dan hasil produk.
2. Sistem Peminjaman *Coworking Space* berhasil dibangun dengan nama Sipenting (Sistem Informasi Peminjaman Ruang *Meeting*) menggunakan MERN *stack technology*. Sistem ini memiliki fitur pengelolaan data oleh *admin* untuk menghindari duplikasi data dan fitur *booking* ruangan oleh *customer* untuk melakukan penjadwalan penggunaan ruangan.
3. Berdasarkan hasil pengujian menggunakan *Black-box testing* pada 15 fitur dengan 57 *test case* yang diuji, didapatkan bahwa fitur-fitur yang dikembangkan dapat berjalan sesuai dengan kebutuhan pengguna. Kemudian hasil pengujian menggunakan *User Experience Questionnaire* (UEQ) yang melibatkan 20 responden, Sistem Peminjaman *Coworking Space* mendapatkan nilai *above average* (diatas rata-rata) pada kategori *attractiveness*, *perspicuity*, *dependability*, *stimulation* dan *novelty* serta nilai *good* (baik) pada kategori *efficiency*.

5.2 Saran

1. Melakukan pengembangan UI yang lebih menarik dan *responsive* terhadap beberapa *device* agar dapat membantu meningkatkan pengalaman pengguna dalam menggunakan sistem ini.
2. Melakukan penambahan fitur notifikasi ke email *client* yang melakukan *booking* untuk memberikan informasi terkait konfirmasi persetujuan atau penolakan dari peminjaman ruangan.
3. Untuk pengembangan selanjutnya, sistem ini dapat diimplementasikan pada peminjaman ruangan lainnya yang ada di gedung di FISIP Unila agar dapat melakukan peminjaman ruangan secara digital.

DAFTAR PUSTAKA

Afif, S.M. (2021) ‘Rancang Bangun Sistem Informasi Staycation Berbasis Web Dengan Implementasi Teknologi Mern Stack’, *Jurnal Manajemen Informasi*, 12(1), pp. 1–12. Available at: <https://ejournal.unesa.ac.id/index.php/jurnal-manajemen-informatika/article/view/41882>.

Ardianzah, D.H., Nuryasin, I. and Wiyono, B.S. (2022) ‘Pengembangan Sistem Pengelolaan Peminjaman Auditorium Universitas Muhammadiyah Malang Berbasis Web Menggunakan Metode Personal Extreme Programming’, *Jurnal Repositor*, 4(2), pp. 137–146. Available at: <https://doi.org/10.22219/repositor.v4i2.1342>.

Ariyanti, L. *et al.* (2020) ‘Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan’, *Jurnal Teknologi dan Sistem Informasi (JTSI)*, 1(1), pp. 90–96. Available at: <http://jim.teknokrat.ac.id/index.php/sisteminformasi>.

Aryo Kombang Ali (2019) *Penggunaan Port Management System (PMS) Dalam Proses Perencanaan Waktu Bongkar Muat Kapal Untuk Efektivitas Dan Efisiensi Kerja Pada PT. Varia Usaha Bahari Gresik*. Politeknik Ilmu Pelayaran Semarang. Available at: <http://repository.pip-semarang.ac.id/id/eprint/2066>.

da Costa, L. (2021) *Testing JavaScript Applications*. Manning. Available at: <https://books.google.co.id/books?id=JTgzEAAAQBAJ>.

Davrin, D.N. and Hidayat, A.M. (2021) ‘Pengaruh Layanan Reservasi Tiket Online Terhadap Kepuasan Pelanggan Studi Kasus Pt . Kereta Api Indonesia the Effect of Online Ticket Reservation Services on Customer Satisfaction Case Study Pt . Kereta Api Indonesia’, *e-Proceeding of Management*, 8(5), pp. 6280–6288. Available at: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/management/article/view/16419/16130>.

Dennis, A., Wixom, B. haley and Roth, roberta M. (2012) *System Analysis and Design Fifth Edition*. 5th edn, *Thorax*. 5th edn. Edited by B.L. Golub. United States of America: Don Fowley.

Dzhurov, Y., Krasteva, I. and Ilieva, S. (2009) ‘Personal Extreme Programming–An Agile Process for Autonomous Developers’, *International Conference on*

software, services & semantic technologies, (August 2016), pp. 252–259. Available at: <http://hdl.handle.net/10506/647>.

Dzulqhuri, D. and Zaelani, A.U. (2021) ‘Rancang Bangun Sistem Reservasi Ruang Rapat Dengan Menggunakan Framework Codeigniter’, *Journal of Artificial Intelligence and Innovative Applications*, 2(2), pp. 152–161. Available at: <http://openjournal.unpam.ac.id/index.php/JOAIIA/article/view/10997/6846>.

Evanandy, A. (2022) *Implementasi Personal Extreme Programming Dalam Pengembangan Integrasi Sistem Informasi Akademik Dan E-Learning*. Universitas Isam Negeri Maulana Malik Ibrahim Malang. Available at: <http://etheses.uin-malang.ac.id/id/eprint/42942>.

Fakhrun, M.W.R. and Gumilang, S.F.S. (2018) ‘Rancangan Web Service Dengan Metode Rest Api Untuk Integrasi Aplikasi Mobile Dan Website Pada Bank Sampah’, *Konferensi Nasional Sistem Informasi*, pp. 214–219. Available at: <http://jurnal.atmaluhur.ac.id/index.php/knsi2018/article/view/361>.

Fowler, M. (2004) *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley (Addison-Wesley object technology series). Available at: <https://books.google.co.id/books?id=nHZslSr1gJAC>.

Gautama, C.F. *et al.* (2019) ‘Desain Media Promosi Merek XWork untuk Meningkatkan Brand Awareness’, *Rupaka*, 1(1), pp. 1–7. Available at: <https://doi.org/10.24912/rupaka.v1i1.2943>.

Handoyo, R., Santoso, L.W. and Setiawan, A. (2019) ‘Real-Time BPMN Website Menggunakan Teknologi MERN Stack’, *Jurnal Infra*, 7(2), pp. 75–80. Available at: <http://publication.petra.ac.id/index.php/teknik-informatika/article/view/8758>.

Herlambang, A.D. *et al.* (2020) ‘V- Model untuk Pengembangan Sistem Informasi Manajemen Ruang Rapat’, *Jurnal Teknologi Informasi dan Ilmu Komputer*, 7(2), p. 313. Available at: <https://doi.org/10.25126/jtiik.2020721893>.

Kharis, Santosa, P.I. and Winarno, W.W. (2016) ‘Evaluasi *User Experience* pada Sistem Informasi Pasar Kerja Menggunakan *User Experience Questionnaire (UEQ)*’, *The Journal of Educators Online-JEO*, 13(1), pp. 1–7. Available at: <http://citee.ft.ugm.ac.id/2019/download51.php?f=38>. Kharis_Evaluasi UX pada Sistem Informasi Pasar Kerja.pdf.

Kurniawan, A. and Eriana, S. (2022) ‘Strategi Framework E-Marketing Dan Reservasi Tempat Dengan Metode Personal Extreme Programming (Studi Kasus: Resto & Cafe Rimbun Kampung Konservasi)’, *Multimedia dan Sistem Informasi (REKLAMASI)*, 1(1), pp. 1–6. Available at: <https://maklumatika.i-tech.ac.id/index.php/reklamasi>.

Morgan, A. (2017) *Introducing the MEAN and MERN Stacks*, *Mongodb Blog*. Available at: <https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack> (Accessed: 14 April 2023).

- Panjaitan, J. and Pakpahan, A.F. (2021) ‘Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase’, *Jurnal Teknik Informatika dan Sistem Informasi*, 7(1), pp. 20–34. Available at: <https://doi.org/10.28932/jutisi.v7i1.3098>.
- Putra, R.E. (2019) *Pengembangan Sistem Informasi Perpustakaan menggunakan Metode Extreme Programming (Studi pada: SMK 1 Muhammadiyah Malang)*, *Jurnal Pengembangan Teknologi dan Ilmu Komputer*. Universitas Brawijaya. Available at: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5695>.
- Sari, A.S. and Hidayat, R. (2022) ‘Designing website vaccine booking system using golang programming language and framework react JS’, *Journal of Information System, Informatics and Computing*, 6(1), pp. 22–39. Available at: <https://doi.org/10.52362/jisicom.v6i1.760>.
- Schrepp, M. (2019) ‘User Experience Questionnaire Handbook Version 8’, pp. 1–15. Available at: www.ueq-online.org.
- Schrepp, M., Hinderks, A. and Thomaschewski, J. (2017) ‘Construction of a Benchmark for the User Experience Questionnaire (UEQ)’, *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4), p. 40. Available at: <https://doi.org/10.9781/ijimai.2017.445>.
- Sia *et al.* (2019) ‘Implementasi Konsep “ Smart ” pada Interior Co-working Space “ Semarang Digital Kreatif ” di Semarang’, *Jurnal Intra*, 7(2), pp. 196–204. Available at: <https://publication.petra.ac.id/index.php/desain-interior/article/view/8892>.
- Sianipar, R.H. (2015) *Pemrograman Javascript: Teori Dan Implementasi*. Penerbit Informatika.
- Supriady and Ginasta, N.G. (2022) ‘Rancang Bangun Aplikasi Penyewaan Coworking Space Berbasis Web’, *Jurnal Teknik Informatika*, 14(3), pp. 123–127. Available at: <https://ejurnal.poltekpos.ac.id/index.php/informatika/article/view/2384>.
- W, D.M.R., Wahyuni, E.D. and Nuryasin, I. (2023) ‘Rancang Bangun Web Pengelolaan Hutan Rakyat Dengan Metode Personal Extreme Programming’, *Repositor*, 5(201510370311123), pp. 533–542.
- Wahyudi, M. (2021) *Rancang Bangun Aplikasi Pembelajaran Praktikum Sistem Informasi Akuntansi di SMKS Barunawati Surabaya, repository dinamika*. Universitas Dinamika. Available at: <https://repository.dinamika.ac.id/id/eprint/5623/>.
- WeWork (2023) *About WeWork, WeWork Website*. Available at: <https://www.wework.com/id-ID> (Accessed: 24 March 2023).
- Xwork (2023) *About XWork, XWork Website*. Available at: <https://xwork.co/> (Accessed: 24 March 2023).
- Yoga, I.G.A. (2018) *Coworking Space Di Kota Yogyakarta Dengan Pendekatan Prinsip-Prinsip Fleksibilitas Dalam Arsitektur*. Universitas Atma Jaya Yogyakarta. Available at: <http://e-journal.uajy.ac.id/13989/>.