

PROPOSAL PENELITIAN

**PENGEMBANGAN FITUR RESERVASI RUANG
DISKUSI ITTP *LIBRARY* BERBASIS WEB DENGAN
METODE *EXTREME PROGRAMMING* (XP)**



MUHAMMAD NAZAR ALWI

20104038

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM
PURWOKERTO**

2023

PROPOSAL PENELITIAN

**PENGEMBANGAN FITUR RESERVASI RUANG
DISKUSI ITTP *LIBRARY* BERBASIS WEB DENGAN
METODE *EXTREME PROGRAMMING* (XP)**



MUHAMMAD NAZAR ALWI

20104038

Dimas Fanny Hebrasianto Permadi, S.ST. M.Kom. (0731039201)

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023**

HALAMAN PERSETUJUAN PEMBIMBING

**PENGEMBANGAN FITUR RESERVASI RUANG
DISKUSI ITTP *LIBRARY* BERBASIS WEB DENGAN
METODE *EXTREME PROGRAMMING* (XP)**

Dipersiapkan dan Disusun oleh
Muhammad Nazar Alwi
20104038

Usulan penelitian Tugas Akhir telah disetujui pada tanggal
27 Juni 2023

Pembimbing Utama,



Dimas Fanny Hebrasianto Permadi, S.ST., M.Kom.

NIDN. 0731039201

HALAMAN PENGESAHAN

PENGEMBANGAN FITUR RESERVASI RUANG DISKUSI ITTP *LIBRARY* BERBASIS WEB DENGAN METODE *EXTREME PROGRAMMING* (XP)

Dipersiapkan dan Disusun oleh

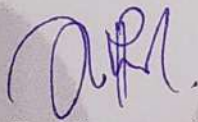
Muhammad Nazar Alwi

20104038

Usulan penelitian Tugas Akhir telah disetujui pada tanggal

26 Juli 2023

Penguji,



Yogo Dwi Prasetyo, S.Si., M.Si.

NIDN. 0101088702

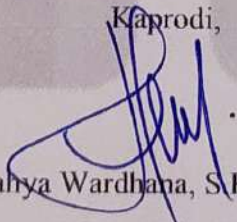
Pembimbing,



Dimas Fanny Hebrasianto Permadi, S.ST., M.Kom.

NIDN. 0731039201

Kaprodi,



Ariq Cahya Wardhana, S.Kom., M.Kom.

NIK. 22930007

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini,

Nama Mahasiswa : Muhammad Nazar Alwi
NIM : 20104038
Program Studi : Rekayasa Perangkat Lunak

Menyatakan bahwa Tugas Akhir dengan judul berikut:

**PENGEMBANGAN FITUR RESERVASI RUANG DISKUSI ITTP *LIBRARY*
BERBASIS WEB DENGAN METODE *EXTREME PROGRAMMING* (XP)**

Dosen Pembimbing Utama : Dimas Fanny Hebrasianto Permadi, S.ST. M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Institut Teknologi Telkom Purwokerto maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan, dan penelitian Saya Sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing.
3. Dalam Karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggungjawab Saya, bukan tanggungjawab Institut Teknologi Telkom Purwokerto.
5. Pernyataan ini Saya buat dengan sesungguhnya, apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka Saya bersedia menerima Sanksi Akademik dengan pencabutan gelar yang

sudah diperoleh serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Purwokerto, 17 Agustus 2023,
Yang Menyatakan,

(Materai 10000 dan tanda tangan)

(Muhammad Nazar Alwi)

KATA PENGANTAR

DAFTAR ISI

Table of Contents

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
ABSTRAK	xi
ABSTRACT.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah.....	4
1.3 Pertanyaan Penelitian.....	4
1.4 Tujuan Penelitian	4
1.5 Batasan Masalah	4
1.6 Manfaat Penelitian.....	5
BAB II LANDASAN TEORI.....	6
2.1 Tinjauan Pustaka.....	6
2.2 Landasan Teori.....	13
BAB III METODOLOGI PENELITIAN	31
3.1 Objek dan Subjek Penelitian.....	31
3.2 Alat dan Bahan Penelitian	31
3.3 Diagram Alir Penelitian	33
3.4 Jadwal Penelitian.....	44
DAFTAR PUSTAKA	47
LAMPIRAN	49
LAMPIRAN 1. SURVEI PENILAIAN PENGGUNA TERHADAP SISTEM RESERVASI RUANG DISKUSI ITTP LIBRARY	49
LAMPIRAN 2. REQUIREMENTS	52

LAMPIRAN 3. USER STORIES.....	55
LAMPIRAN 4. ACCEPTANCE TEST CRITERIA.....	59
LAMPIRAN 5. WIREFRAME	62

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian-Penelitian Terkait.....	9
Tabel 2.2 Elemen DFD	20
Tabel 2.3 Elemen <i>Use Case</i>	22
Tabel 2.4 Elemen <i>Activity Diagram</i>	24
Tabel 3.1 Spesifikasi Perangkat Keras.....	31
Tabel 3.2 Daftar Perangkat Lunak	32
Tabel 3.3 Bahan Penelitian	32
Tabel 3.4 Jadwal Penelitian.....	44

DAFTAR GAMBAR

Gambar 2.1 Alur XP [19].....	18
Gambar 3.1 Alur Penelitian.....	34
Gambar 3.2 DFD Sistem Reservasi Ruang Diskusi.....	36
Gambar 3.3 Case Sistem Reservasi Ruang Diskusi	37
Gambar 3.5 AD – Melakukan Reservasi Ruang Diskusi.....	39
Gambar 3. 6 AD – Menandai Jadwal yang Terpesan.....	40
Gambar 3.7 AD – Menciptakan Kode Batang	40
Gambar 3.8 AD – Mencatat Daftar Pesanan.....	41
Gambar 3.9 AD – Mengelola Daftar Pemesanan.....	41
Gambar 3.10 AD – Menampilkan Jadwal yang Masih Tersedia	42

ABSTRAK

Perpustakaan Institut Teknologi Telkom Purwokerto atau yang biasa disebut sebagai ITTP *Library* memiliki fasilitas ruangan yang bernama Ruang Diskusi. Seluruh anggota perpustakaan dapat menggunakan fasilitas ruang diskusi. Untuk menggunakan fasilitas tersebut, anggota perpustakaan wajib melakukan reservasi melalui Google Form dengan memasukkan data diri dan memilih jadwal. Sebelum memilih jadwal, anggota perpustakaan wajib memeriksa jadwal yang tersedia pada Spreadsheet. Perlu dilakukan pengembangan web khusus reservasi ruang diskusi agar proses reservasi menjadi lebih efisien. Web ini akan terintegrasi dengan web ITTP Digital Library atau *D-Library*. Web *D-Library* dikembangkan menggunakan perangkat lunak bernama SLiMS. SLiMS mendukung sistem *plugin*. Dengan adanya sistem *plugin*, penambahan fitur dapat dilakukan tanpa mengubah kode utama web *D-Library*. Oleh karena itu, peneliti akan mengembangkan web reservasi ruang diskusi menggunakan perangkat lunak SLiMS dan menjadikannya sebagai *plug-in* web *D-Library*. Metode penelitian yang digunakan adalah metode *research and development* (R&D). Metode pengembangan perangkat lunak yang digunakan adalah metode *Extreme Programming* (XP). Hasil dari penelitian ini berupa web reservasi ruang diskusi yang sudah terintegrasi dengan web *D-Library*.

Kata Kunci: web, SLiMS, *plugin*, R&D, *Extreme Programming*

ABSTRACT

Telkom Purwokerto Institute of Technology Library or commonly referred to as the ITTP Library has a room facility called the Discussion Room. All members of the library can use the discussion room facilities. To use this facility, library members are required to make a reservation via Google Form by entering personal data and choosing a schedule. Before choosing a schedule, library members must check the schedule available in the Spreadsheet. It is necessary to develop a website specifically for discussion room reservations so that the reservation process becomes more efficient. This website will be integrated with the ITTP Digital Library or D-Library website. D-Library website was developed using SLiMS software. SLiMS supports a plugin system. With the plugin system, researcher didn't need to change the main code for adding new features. Therefore, researcher will develop a discussion room reservation website using SLiMS software and make it a D-Library website plug-in. The research method used is the research and development (R&D) method. The software development methodology used is the Extreme Programming method. The result of this research is a discussion room reservation website that has been integrated with the D-Library website.

Keyword: *website, SLiMS, plugin, R&D, Extreme Programming*

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perpustakaan Institut Teknologi Telkom Purwokerto (ITTP) atau biasa disebut sebagai ITTP *Library* menyediakan berbagai fasilitas yang memadai untuk para mahasiswa. Fasilitas-fasilitas yang tersedia di dalam Perpustakaan ITTP meliputi buku-buku, loker, meja baca, pojok baca, ruang diskusi, ruang laktasi, serta ruang audio visual. Fasilitas buku yang tersedia menawarkan beragam pilihan, mencakup buku referensi nasional dan internasional, novel, komik, jurnal, serta skripsi mahasiswa ITTP. Fasilitas loker digunakan untuk menyimpan barang bawaan pengunjung. Fasilitas meja baca dapat digunakan untuk membaca atau mengerjakan tugas [1]. Fasilitas pojok baca disediakan untuk orang-orang yang ingin membaca secara lesehan. Fasilitas ruang diskusi digunakan untuk pengunjung yang ingin berdiskusi di forum tertutup. Fasilitas ruang diskusi sangat mendukung kegiatan diskusi karena terdapat layar yang dapat digunakan untuk presentasi, meja yang dilengkapi dengan stopkontak, ruangnya yang kedap suara, dan lainnya. Fasilitas ruang laktasi dapat digunakan oleh ibu menyusui. Fasilitas ruang audio visual dapat digunakan sebagai tempat untuk pengunjung yang ingin rehat sejenak.

Untuk menggunakan fasilitas ruang diskusi, mahasiswa wajib mendaftar atau melakukan reservasi terlebih dahulu. Proses reservasi dilakukan melalui Google Form. Sebelum melakukan pengisian formulir reservasi, mahasiswa wajib untuk melihat jadwal ruang diskusi yang tersedia. Hal ini dilakukan untuk menghindari kemungkinan melakukan reservasi pada jadwal yang sudah dipesan oleh mahasiswa lain. Jika sudah

melakukan reservasi, mahasiswa diperkenankan untuk memakai ruang diskusi maksimal selama dua jam. Setelah menggunakan Ruang Diskusi, mahasiswa wajib mencatatkan nama mereka pada formulir daftar kehadiran.

Sayangnya proses reservasi dan pencatatan anggota yang menggunakan ruang diskusi dilakukan secara manual. Reservasi dilakukan menggunakan Google Form dan pengecekan jadwal dilihat melalui Spreadsheet. Penggunaan Google Form sebagai formulir pendaftaran memiliki kekurangan yaitu pengunjung tidak dapat melihat dan memilih secara langsung jadwal yang tersedia. Oleh karena itu, peneliti ingin mengembangkan sebuah web yang memiliki fitur pengecekan jadwal secara langsung.

Web reservasi ruang diskusi akan terintegrasi dengan web ITTP *Library* atau yang biasa disebut dengan web *D-Library*. Web *D-Library* dibangun menggunakan perangkat lunak bernama SLiMS. SLiMS merupakan sebuah program manajemen perpustakaan yang dikembangkan oleh komunitas Senayan Developers Community (SDC) [2]. Oleh karena itu, peneliti akan mengembangkan web reservasi ruang diskusi menggunakan SLiMS.

Salah satu keunggulan dari SLiMS yaitu adanya sistem *plugin*. Sistem *plugin* memfasilitasi pengembang SLiMS dalam menambahkan fitur baru pada aplikasi SLiMS tanpa harus melakukan perubahan pada kode utama SLiMS [3]. Oleh karena itu, peneliti akan mengembangkan web reservasi ruang diskusi sebagai *plugin* SLiMS.

Di penelitian ini web reservasi ruang diskusi akan menampilkan formulir yang akan diisi oleh pengunjung. Fitur lainnya juga terdapat pemilihan jadwal yang tersedia secara langsung. Tetapi, hanya akan menampilkan jadwal-jadwal yang tersedia. Sehingga tidak ada kemungkinan pengunjung memilih jadwal yang sudah tidak tersedia.

Sebelum melakukan pengembangan web tersebut, peneliti perlu memilih siklus pengembangan perangkat lunak atau SDLC (*Software Development Lifecycle*) yang tepat. SDLC juga bisa disebut sebagai metode pengembangan perangkat lunak. Pada dokumen penelitian ini, SDLC akan dirujuk sebagai metode pengembangan perangkat lunak.

Secara umum, metode pengembangan perangkat lunak dibagi menjadi dua yaitu metode non-Agile dan metode Agile. Metode non-Agile mencakup metode Waterfall, Iterative, Spiral, Rapid Application Development (RAD), dan V-Model. Sementara itu, metode Agile meliputi metode Scrum, Extreme Programming (XP), Lean, dan Kanban [4].

Setiap metode pengembangan memiliki kelebihan dan kekurangan. Pemilihan suatu metode disesuaikan dengan karakteristik proyek dan kondisi pengembang. Peneliti memilih metode XP sebagai metode pengembangan perangkat lunak. Yang menjadi pertimbangan peneliti dalam memilih metode XP yaitu karena metode ini adaptif terhadap perubahan dan cocok untuk pengembang tunggal. Metode ini akan memudahkan peneliti dalam mengerjakan proyek walaupun ada perubahan persyaratan di tengah proses pengembangan. Selain itu kelebihan lain pada metode XP yaitu diwajibkannya menambahkan *unit test* atau pengujian unit yang dapat memastikan kode berjalan dengan baik dan tidak terjadi regresi [4].

Dari masalah reservasi yang telah disebutkan, maka tujuan dari penelitian ini adalah merancang sebuah web reservasi ruang diskusi dengan metode pengembangan XP. Hasil dari penelitian ini diharapkan dapat membantu pustakawan dalam melakukan pencatatan anggota perpustakaan yang menggunakan ruang diskusi. Juga dapat membantu anggota perpustakaan agar dapat melakukan reservasi pada platform yang memiliki fitur yang dapat memenuhi kebutuhan mereka.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah disebutkan di atas, dapat diketahui bahwa ITTP *Library* memiliki masalah pada sistem reservasi ruang diskusi. Sistem tersebut menggunakan Google Form sebagai formulir pendaftarannya, sehingga proses pendaftaran tidak efektif. Selain itu, sebelum pengunjung melakukan reservasi, pengunjung terlebih dahulu harus mengakses halaman Spreadsheet untuk melihat jadwal yang tersedia.

1.3 Pertanyaan Penelitian

Berdasarkan uraian di atas, maka peneliti merumuskan pertanyaan yang akan dibahas pada penelitian ini yaitu bagaimana cara merancang dan membangun web berbasis *plugin* yang dapat terintegrasi dengan web D-*Library* sehingga memudahkan mahasiswa untuk melakukan reservasi ruang diskusi?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disebutkan dapat diketahui tujuan dari penelitian ini adalah merancang dan membangun web yang dapat memudahkan mahasiswa untuk melakukan reservasi ruang diskusi.

1.5 Batasan Masalah

Berdasarkan rumusan masalah dan tujuan penelitian, maka untuk mewujudkan penelitian yang sesuai dengan masalah yang ada diperoleh batasan batasan masalah penelitian sebagai berikut:

1. Perangkat lunak yang dihasilkan hanya platform web yang berbasis *plugin* SLiMS.
2. Sistem reservasi ruang diskusi dirancang hanya untuk meningkatkan kemudahan proses reservasi ruang diskusi di ITTP *Library*.

1.6 Manfaat Penelitian

Adapun manfaat yang diharapkan dengan adanya penelitian ini adalah sebagai berikut:

1. Memudahkan mahasiswa dalam melakukan reservasi ruang diskusi.
2. Digitalisasi fasilitas perpustakaan Institut Teknologi Telkom Purwokerto.
3. Memudahkan pustakawan untuk mendapatkan data anggota perpustakaan yang telah melakukan reservasi ruang diskusi.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang bertujuan untuk mengembangkan web berbasis SLiMS sudah banyak diterapkan di berbagai perpustakaan di Indonesia. Pada penelitian yang sudah dilakukan sebelumnya menunjukkan bahwa SLiMS memiliki fitur *plugin* memungkinkan para pengembang untuk menambahkan fitur pada SLiMS tanpa mengubah kode yang sudah ada.

Pada penelitian yang berjudul “Model Pengembangan *Plug-In* SLiMS pada Komunitas SLiMS Kudus” yang dilakukan oleh Zaid Abdurrahman dan Thoriq Tri Prabowo menjelaskan tentang apa saja peran komunitas SLiMS Kudus pada pengembangan *plugin* SLiMS. Dalam tulisannya, Zaid dan Thoriq menggunakan model pengembangan *open-source* [5] untuk membantu mereka dalam mendeskripsikan pengembangan *plugin* SLiMS. Metode penelitian yang mereka gunakan adalah metode penelitian kualitatif. Sementara itu, teknik pengumpulan data yang mereka gunakan antara lain, teknik observasi terstruktur, wawancara tidak terstruktur, dan dokumentasi. Untuk metode analisis data, mereka menggunakan model Miles and Huberman. Kesimpulan dari penelitian ini adalah Komunitas SLiMS Kudus berperan sebagai sumber inspirasi dan wadah dalam penyebaran *plugin* SLiMS [3].

Pada penelitian kedua yang berjudul “Pengembangan Sistem Informasi Perpustakaan Berbasis SLiMS Akasia 8” yang dilakukan oleh Haidir Rahman dkk bertujuan untuk mengembangkan sistem informasi perpustakaan SMK Negeri 2 Banjarmasin. Mereka menggunakan Sistem informasi yang dikembangkan akan menggunakan SLiMS Akasia 8. Sistem

informasi perpustakaan yang dulunya menggunakan metode konvensional dikembangkan ke metode digital. Tujuan penelitian adalah yaitu sebagai prosedur pengembangan sistem informasi perpustakaan berbasis SLiMS Akasia 8 dan sebagai pengembangan sistem informasi perpustakaan. Penelitian ini menggunakan jenis penelitian *research and development* (R&D) dengan menggunakan model pengembangan Alessi & Trollip yang sudah dimodifikasi. Hasil penelitian dan pengembangan sistem informasi perpustakaan ini dinyatakan layak setelah mendapat validasi dari ahli sistem dan ahli perpustakaan, maka direkomendasikan pengembangan ini agar bisa diterapkan ke lembaga pendidikan lainnya yang mempunyai sistem informasi perpustakaan untuk mengevaluasi lebih dalam lagi [6].

Pada penelitian ketiga yang berjudul “Virtualisasi Katalog Senayan Library Management System (SLiMS) Berbasis 3D” bertujuan untuk menghasilkan aplikasi virtualisasi katalog SLiMS untuk penataan buku di rak dan pencarian buku dalam bentuk 3D serta mengetahui keakuratan dan performa. Metode dilakukan diawali membuat struktur *vertex* dari buku dan rak dengan menggunakan data polihedron dari objek kubus. Data polihedron dapat dibaca dengan aplikasi Blender 3D, kemudian penulis melakukan analisis untuk mendapatkan rumus pembentuk polihedron serta mendapatkan urutan indeks dari *vertex*. Data masukkan yang digunakan untuk membentuk polihedron buku adalah panjang, lebar dan tebal buku, sedangkan untuk polihedron rak adalah panjang, lebar, tinggi dan tebal papan. Data masukkan disimpan di basis data agar dapat diakses dan dikonversi menjadi 3D dengan pemrograman web [7].

Pada penelitian keempat yang berjudul “*E-Library* Interaktif dengan SLiMS Bulian Menggunakan Metode RAD” bertujuan untuk merancang dan membangun perpustakaan digital berbasis web menggunakan SLiMS menjadi lebih interaktif dengan menambahkan sarana forum diskusi. Metode yang digunakan untuk pengumpulan data yaitu observasi ke perpustakaan sekolah, wawancara pihak pengelola perpustakaan, dan studi

literatur untuk memperdalam pemahaman masalah. Adapun metode pengembangan perangkat lunak yang digunakan adalah *Rapid Application Development (RAD)* dengan memanfaatkan *Unified Modelling Language (UML)* untuk menggambarkan rancangan sistem yang dikembangkan. Metode pengujian yang digunakan adalah pengujian *black-box* dan *System Usability Scale (SUS)* [8].

Pada penelitian kelima yang berjudul “Implementasi SLiMS di Perpustakaan Perguruan Tinggi” bertujuan untuk mengetahui kendala, tantangan, dan manfaat dalam implementasi SLiMS di perpustakaan perguruan tinggi. Selain itu dalam penelitian ini juga dideskripsikan kendala dan tantangan serta manfaat yang diperoleh melalui implementasi SLiMS di perpustakaan perguruan tinggi. Penelitian ini menggunakan metode *Systematic Literature Review (SLR)* dengan pendekatan kualitatif. Adapun metode pengumpulan dan analisis data yang digunakan terdiri dari beberapa tahapan seperti tahap perencanaan, pelaksanaan, dan mensintesis terhadap hasil temuan pencarian. Temuan pada penelitian ini menunjukkan bahwa terdapat enam belas artikel jurnal terseleksi yang mengulas terkait implementasi SLiMS di perpustakaan perguruan tinggi, yang terdiri dari sepuluh artikel penelitian di perpustakaan universitas, tiga penelitian di perpustakaan sekolah tinggi, satu penelitian di perpustakaan politeknik dan dua penelitian di perpustakaan institut. Setelah melakukan tahap tinjauan literatur terdapat dua artikel jurnal terseleksi yang mengulas sistem SLiMS sebagai kendala atau tantangan serta terdapat dua belas artikel jurnal terseleksi yang mendeskripsikan manfaat implementasi SLiMS di perpustakaan yang terkait dengan pelayanan perpustakaan. Adapun kendala yang dihadapi yaitu terkait dengan sistem SLiMS, sumber daya manusia dan fasilitas. Kemudian manfaat dari implementasi SLiMS di perpustakaan perguruan tinggi diantaranya yaitu terkait pelayanan perpustakaan, pengolahan bahan pustaka dan pengembangan koleksi perpustakaan [9].

Tabel 2.1 Perbandingan Penelitian-Penelitian Terkait

No.	Peneliti	Judul	Metode	Hasil	Perbedaan	Persamaan
1.	Zaid Abdurrahman dan Thoriq Tri Prabowo	Model Pengembangan Plug-In SLiMS pada Komunitas SLiMS Kudus	Kualitatif	Informasi tentang peranan komunitas SLiMS Kudus sebagai sumber inspirasi dan wadah pengembangan <i>plugin</i> SLiMS.	- Metode penelitian yang digunakan - Hasil yang disimpulkan	- Membahas mengenai pengembangan <i>plugin</i> SLiMS.
2.	Haidir Rahman, Hamsi	Pengembangan Sistem Informasi Perpustakaan	Penelitian dan Pengembangan	Sistem informasi perpustakaan yang sudah	- Membahas pengembangan SLiMS secara umum. Tidak	- Metode penelitian yang digunakan. - Membahas mengenai pengembangan sistem informasi perpustakaan berbasis SLiMS.

No.	Peneliti	Judul	Metode	Hasil	Perbedaan	Persamaan
	Mansur, dan Adrie Satrio	Berbasis SLiMS Akasia 8		divalidasi oleh ahli sistem dan ahli perpustakaan.	membahas pengembangan <i>plugin</i> SLiMS.	
3.	Sopingi dan Mei Purweni	Virtualisasi Katalog Senayan Library Management System (SLiMS) Berbasis 3D	<i>Interactive Multimedia System Design and Development (IMSDD)</i>	Menghasilkan aplikasi virtualisasi katalog SLiMS untuk penataan buku di rak dan pencarian buku dalam bentuk 3D.	- Metode penelitian yang digunakan. - Tidak secara khusus membahas pengembangan <i>plugin</i> SLiMS.	- Membahas mengenai pengembangan sistem informasi perpustakaan berbasis SLiMS.
4.	Arief Ichwani dan Muhammad Visal Bainuri	<i>E-Library</i> Interaktif dengan SLiMS Bulian	<i>Rapid Application Development (RAD)</i>	Menghasilkan <i>e-library</i> yang dilengkapi dengan fitur	- Metode penelitian yang digunakan.	- Menambahkan fitur baru yang ada hubungannya dengan forum diskusi.

No.	Peneliti	Judul	Metode	Hasil	Perbedaan	Persamaan
		Menggunakan Metode RAD		forum sehingga dapat mendorong dan budaya literasi bagi siswa.	- Tidak secara khusus membahas pengembangan <i>plugin</i> SLiMS.	
5.	Iskandar dan Luki Wijayanti	Implementasi SLiMS di Perpustakaan Perguruan Tinggi	<i>Systematic Literature Review</i> (SLR) dengan pendekatan kualitatif	Berupa kesimpulan bahwa penelitian terkait implementasi SLiMS di perpustakaan dilakukan di empat jenis perpustakaan	- Metode penelitian yang digunakan. - Tidak secara khusus membahas pengembangan <i>plugin</i> SLiMS.	- Membahas tentang implementasi SLiMS.

No.	Peneliti	Judul	Metode	Hasil	Perbedaan	Persamaan
				perguruan tinggi seperti perpustakaan universitas, perpustakaan sekolah tinggi, perpustakaan politeknik dan perpustakaan institut.		

2.2 Landasan Teori

Berikut adalah kajian mengenai beberapa teori yang digunakan pada penelitian ini:

2.2.1 SLiMS

SLiMS atau Senayan Library Management System merupakan sistem atau aplikasi otomasi perpustakaan yang dicetuskan pertama kali pada tahun 2006 oleh Hendro Wicaksono dan Arie Nugrah [3]. SLiMS pertama kali digunakan di Perpustakaan Departemen Pendidikan Nasional [2]. Fungsi utama SLiMS sebagai pengelola sumber daya perpustakaan seperti buku, jurnal, dokumen digital, dan material perpustakaan lainnya. Selain itu, SLiMS juga dapat mengelola administrasi perpustakaan seperti sirkulasi koleksi, manajemen koleksi, keanggotaan, penghitungan stok, dan lain-lain [10].

SLiMS bersifat *Open Source Software* (OSS) berbasis web [11]. SLiMS mampu berjalan sempurna di dalam sistem jaringan lokal (intranet) maupun internet. Maka dari itu, pemustaka dapat menelusuri katalog perpustakaan dari mana saja dan kapan saja melalui web perpustakaan yang disediakan [12].

2.2.2 PHP

PHP (*Hypertext Preprocessor*) merupakan salah satu bahasa pemrograman yang berjalan dalam sebuah web server dan berfungsi sebagai pengolah data pada sebuah server. Data yang dikirim oleh *user client* akan diolah dan disimpan pada database web server dan dapat ditampilkan kembali apabila diakses. Untuk menjalankan kode-kode program PHP, file harus diunggah ke dalam server.

Unggah adalah proses mentransfer data atau file dari komputer klien ke dalam web server.

Untuk membuat web yang dinamis dan mudah diperbaharui setiap saat dari *browser*, dibutuhkan sebuah program yang mampu mengolah data dari komputer klien atau dari komputer *server* itu sendiri sehingga mudah dan nyaman disajikan di *browser*. Salah satu program yang dapat dijalankan di *server* dan cukup andal adalah PHP.

PHP bekerja di dalam sebuah dokumen HTML (*Hypertext Markup Language*) untuk dapat menghasilkan isi dari sebuah halaman web sesuai permintaan. Dengan PHP, kita dapat merubah situs kita menjadi sebuah aplikasi berbasis web, tidak lagi hanya sekedar sekumpulan halaman statis yang jarang diperbaharui.

Pada awalnya, PHP dirancang untuk diintegrasikan dengan web *server* Apache. Namun belakangan ini, PHP juga dapat bekerja dengan web *server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*) dan Xitami. Yang membedakan PHP dengan bahasa pemrograman lain adalah adanya tag penentu, yaitu diawali dengan “<?” atau “<?php” dan diakhiri dengan “>”. Jadi kita bebas menempatkan skrip PHP dimanapun dalam dokumen HTML yang telah kita buat [13].

PHP adalah bahasa *scripting server-side* yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi web. *Script* sendiri merupakan sekumpulan instruksi pemrograman yang ditafsirkan pada saat *runtime*. Sedangkan bahasa *scripting* adalah bahasa yang menafsirkan skrip saat *runtime* dan biasanya tertanam ke dalam lingkungan perangkat lunak lain [14].

Karena PHP merupakan bahasa pemrograman bertipe *server-side*, PHP akan diproses oleh *server* yang hasil olahannya akan dikirim kembali ke *browser*. Oleh karena itu, salah satu alat yang harus tersedia sebelum menggunakan PHP adalah *server*. Agar dapat menjalankan PHP di komputer sendiri, yang harus dilakukan pertama kali adalah melakukan instalasi *server* [15].

2.2.3 MySQL

My Structured Query Language (MySQL) merupakan salah satu basis data *open-source* yang banyak digunakan pengembang perangkat lunak. MySQL bekerja menggunakan SQL (*Structure Query Language*). MySQL merupakan standar penggunaan database di dunia untuk pengolahan data [16].

MySQL adalah sebuah DBMS (*Database Management System*) menggunakan perintah SQL yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis web. MySQL dibagi menjadi dua lisensi, pertama adalah *free software* dimana perangkat lunak dapat diakses oleh siapa saja. Dan kedua adalah *shareware* di mana perangkat lunak berpemilik memiliki batasan dalam penggunaannya [14].

2.2.4 XAMPP

XAMPP merupakan *server* yang paling banyak digunakan. Fiturnya lengkap namun tetap mudah untuk digunakan oleh pemrogram PHP pemula karena yang perlu dilakukan hanyalah menjalankan salah satu modul bernama Apache yang dapat memproses PHP [15].

2.2.5 Bootstrap

Bootstrap merupakan sebuah *framework* yang dikembangkan oleh Mark Otto dan Jacob Thornton dari Twitter. *Framework* ini diluncurkan sebagai produk *open source* pada Agustus tahun 2011

di GitHub. Bootstrap memiliki fitur-fitur komponen antarmuka yang bagus seperti Typography, Forms, Buttons, Tables, Navigations, Dropdowns, Alert, Modals, Tabs, Accordion, Carousel, dan lain sebagainya. Bootstrap memudahkan proses pembuatan tata letak web yang responsif dengan mudah. Salah satu kelebihan yang dimiliki Bootstrap adalah *framework* ini berisi kumpulan *tool* yang gratis untuk membuat tata letak web yang fleksibel dan responsif. *Framework* ini juga memiliki komponen antarmuka bagus [17].

2.2.6 Metode Pengembangan *Agile*

Metode Pengembangan *Agile* adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun.

Pendekatan untuk pengembangan perangkat lunak yang cepat karena persyaratan berubah dalam waktu yang relatif singkat merupakan pengertian *Agile*. Pendekatan ini juga populer saat ini karena memberikan fleksibilitas terhadap pengembang ketika melakukan proses pengembangan. Konsep utama pengembangan *Agile* adalah pengembangan aplikasi, kolaborasi, dan komunikasi antar tim. Tim akan fokus pada pengerjaan aplikasi dengan meminimalkan dokumentasi. Bekerja sebagai tim yang terdiri dari dua atau lebih pemrogram yang mengerjakan fitur dan komunikasi intensif antara pemrogram dan pelanggan.

Ada beberapa macam metode *Agile*, di antaranya adalah *Extreme Programming*, *Adaptive Software Development*, *Dynamic Systems Development Method* (DSDM) dan *Scrum*. Terdapat beberapa prinsip yang berlaku untuk mengimplementasikan pengembangan perangkat lunak, yakni kepuasan pelanggan adalah prioritas utama, menerima perubahan persyaratan walaupun di akhir

pengembangan, memberikan hasil/perangkat lunak dalam beberapa minggu hingga bulan, selama proses pengembangan, lingkungan tim yang dapat dipercaya dan memotivasi satu anggota yang lain, mengedepankan komunikasi antar tim dan mengedepankan hasil fungsi dari perangkat lunak tersebut [18].

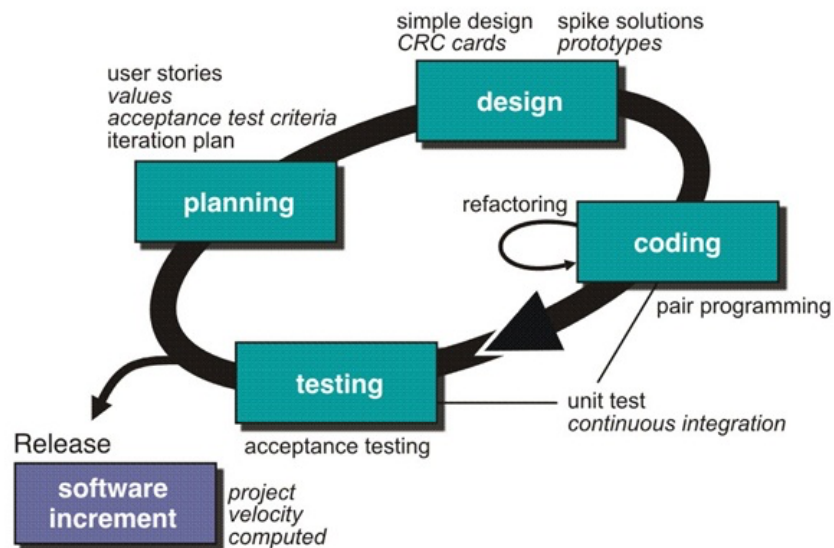
2.2.7 *Extreme Programming (XP)*

Metode XP pertama kali digunakan pada 6 Maret 1996. XP merupakan salah satu metode pengembangan *Agile* yang populer. XP sudah dibuktikan sukses di banyak perusahaan di berbagai industri [19].

XP bisa sukses diterapkan karena mendahulukan kepuasan pelanggan. Alih-alih mengembangkan perangkat lunak dengan banyak kriteria untuk masa yang akan datang. Metode ini mengembangkan perangkat lunak sesuai dengan kebutuhan yang prioritas saja [19].

XP meningkatkan proyek perangkat lunak dalam lima cara; *communication, simplicity, feedback, respect, and courage*. Pengembang yang menerapkan metode ini melakukan komunikasi dengan pelanggan dan sesama pengembang secara berkesinambungan. Mereka menjaga desain mereka agar tetap sederhana dan bersih. Mereka mendapatkan umpan balik dengan menguji perangkat lunak mereka yang dimulai dari hari pertama. Mereka memberikan sistem ke pengguna sedini mungkin dan menerapkan perubahan seperti yang disarankan oleh pengguna. Setiap kesuksesan kecil memperdalam rasa hormat mereka terhadap kontribusi dari setiap anggota tim. Dengan landasan ini, pengembang dapat dengan berani menanggapi perubahan persyaratan dan teknologi [19].

Aspek yang paling unggul dari XP adalah alurnya yang sederhana. Berikut alur pada XP:



Gambar 2.1 Alur XP [20]

Tahap **planning** dimulai dengan membuat *user stories*, *acceptance test criteria*, dan merencanakan iterasi. Pada tahap **design**, XP akan menggunakan *spike solution* di mana pembuatan desain dibuat langsung ke tujuannya. Selanjutnya pada tahap **coding**, XP akan memulainya dengan membangun *unit test* yang kemudian dilanjutkan dengan membuat *production code* yang sesuai dengan *unit test* tersebut. Pada tahap **testing**, dilakukan pengujian *user acceptance test* (UAT). Pengujian ini dilakukan oleh *customer* yang berfokus pada fitur dan fungsi sistem secara keseluruhan [20].

XP sangat mirip dengan permainan *puzzle*. Ada banyak potongan kecil. Secara individual potongan-potongan itu tidak ada artinya, tetapi ketika digabungkan bersama, potongan-potongan itu menjadi gambar yang jelas. Aturan XP mungkin tampak canggung dan bahkan mungkin naif pada awalnya, tetapi didasarkan pada nilai dan prinsip yang sehat [19].

Aturan XP menetapkan harapan di antara anggota tim tetapi bukan tujuan akhir itu sendiri. Aturan ini menentukan lingkungan yang mempromosikan kolaborasi dan pemberdayaan tim. Setelah tercapai, kerja sama tim yang produktif akan berlanjut meskipun aturan diubah agar sesuai dengan kebutuhan khusus perusahaan [19].

Intinya,

2.2.8 GitHub

Github adalah perangkat lunak *hosting* untuk proyek *open source* yang menggunakan sistem kontrol versi Git. Git adalah *tool* untuk melakukan revisi kode, sedangkan Github adalah web *hosting*-nya. Mudahnya Github adalah web *hosting* untuk proyek-proyek perangkat lunak seperti Linux, Atom, dan lain-lain. Dalam bahasa gaul, Github adalah jejaring sosial untuk software developer [21].

2.2.9 Unit Test

Unit test atau pengujian unit merupakan bagian dari *automated test* di mana pengujian yang umpan baliknya cepat, konsisten, dan tidak ambigu. Cepat yang dimaksud adalah sebuah *unit test* harus selesai dijalankan dalam milidetik. Konsisten yang dimaksud adalah *unit test* harus memberikan hasil yang sama pada kode yang sama. Urutan eksekusi *unit test* tidak berpengaruh, sehingga *unit test* dapat dijalankan secara paralel. Tidak ambigu yang dimaksud adalah *unit test* harus secara jelas memberikan informasi tentang *bug* yang terdeteksi [22].

2.2.10 Data Flow Diagram (DFD)


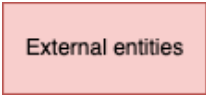

Data Flow Diagram (DFD) merupakan sebuah model data *Requirements Modeling Language* (RML) yang menyediakan representasi grafis dari aliran informasi melalui solusi, berfokus pada bagaimana data ditransformasikan saat dimanipulasi atau digunakan dalam proses. DFD menunjukkan tampilan solusi yang


menyatukan banyak alur proses yang berbeda, alur sistem, atau *use case*. DFD tidak menunjukkan keputusan (*decision*), dan tidak menunjukkan urutan proses. DFD hanya menunjukkan bagaimana data mengalir di antara proses dan diubah oleh proses [23].

DFD berasal dari seperangkat teknik analisis yang disebut analisis terstruktur, yang didefinisikan pada tahun 1970-an dan 1980-an (DeMarco 1979). Dalam analisis terstruktur, pengembang perangkat lunak membuat DFD dari diagram konteks dan kemudian menguraikan DFD untuk membuat modul fungsional dari solusi (Yourdon 1986). DFD RML digunakan untuk membantu mengumpulkan persyaratan daripada merancang arsitektur teknis. DFD adalah tampilan berorientasi data dari solusi yang memungkinkan pengembang untuk mengembangkan gambaran besar tentang bagaimana data bergerak melalui solusi .

DFD merupakan diagram visual yang menggunakan 4 tipe elemen, seperti yang tertera pada Tabel 2.2 berikut:

Tabel 2.2 Elemen DFD

Elemen	Arti
	<i>Data stores</i> merupakan tempat di mana data disimpan sementara atau permanen [23].
	<i>External entities</i> dapat berupa orang atau sistem lainnya yang memberi data atau mengambil data dari sistem [23].
	<i>Process</i> merupakan tanda bahwa data dimanipulasi [23].

	<p>Merupakan aliran data. Aliran data harus melewati elemen <i>process</i>, karena <i>external entities</i> dan <i>data stores</i> tidak mengirimkan data antara satu sama lain secara langsung [23].</p>
-----------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.2.11 Use Case Modeling

Ide dasar di balik pemodelan *use case* cukup sederhana yaitu untuk mengetahui inti dari apa yang harus dilakukan sistem. Untuk membuat model *use case*, yang harus dilakukan pertama kali adalah fokus pada siapa (atau apa) yang akan menggunakannya, atau digunakan olehnya. Setelah itu, lihat apa yang harus dilakukan sistem untuk pengguna tersebut agar dapat melakukan sesuatu yang berguna.

Model *use case* mencakup komponen-komponen berikut:

Actors. mewakili orang-orang atau hal-hal yang berinteraksi dalam beberapa cara dengan sistem. *Actors* berada di luar sistem. *Actors* memiliki nama dan deskripsi singkat, dan mereka diasosiasikan dengan *use case* dimana mereka berinteraksi. [24].

Use cases. mewakili hal-hal yang dapat dilakukan sistem untuk para aktornya. *Use cases* bukanlah fungsi atau fitur, dan tidak dapat diuraikan. *Use cases* memiliki nama dan deskripsi singkat. *Use Cases* juga memiliki deskripsi detail yang pada intinya adalah cerita tentang bagaimana *actors* menggunakan sistem untuk melakukan sesuatu yang mereka anggap penting, dan apa yang sistem lakukan untuk memenuhi kebutuhan tersebut.



2.2.12 Use Case Diagram



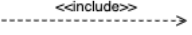
Actors dan *use case* dapat digambarkan pada diagram *use case*. *Actors* diwakili oleh orang tongkat dan kasus penggunaan oleh elips. Panah (mewakili hubungan) menghubungkan aktor dan kasus

penggunaan yang berinteraksi. Panah membantu menunjukkan penggagas interaksi. Tujuan dari diagram adalah untuk meringkas apa yang akan dilakukan sistem. Diagram tidak benar-benar mendeskripsikan sistem—mengira diagram *use case* sebagai model *use case* lengkap adalah kesalahan umum yang dilakukan banyak tim. Diagram memberikan ringkasan, tetapi sebagian besar deskripsi diadakan, sebagai teks, dalam dokumen yang terkait dengan *use case*. Deskripsi *use case* ini memberikan cerita lengkap tentang apa yang terjadi dalam *use case*. Jadi untuk setiap *use case* dalam model *use case*, akan ada dokumen yang menjelaskan bagaimana *actors* dan sistem berkolaborasi untuk memenuhi tujuan yang diwakili oleh *use case*. Dalam karya tulis ini, saat peneliti mengacu pada *use case*, yang kami maksud adalah totalitas *use case*, termasuk representasi ikoniknya, hubungannya, dan, yang paling penting, deskripsi mendetailnya.

Pada Tabel 2.3 dapat dilihat arti dari setiap elemen pada *use case*.

Tabel 2.3 Elemen *Use Case*

Elemen	Arti
	Sebuah <i>actor</i> mendefinisikan peran yang dapat dimainkan pengguna saat berinteraksi dengan sistem. Pengguna dapat berupa individu atau sistem lain [24].
	<i>Use case</i> menggambarkan bagaimana seorang <i>actor</i> menggunakan sistem untuk mencapai tujuan dan apa yang sistem lakukan untuk <i>actor</i> untuk mencapai tujuan tersebut. <i>Use case</i> bercerita tentang bagaimana sistem dan <i>actor</i> -nya berkolaborasi untuk memberikan sesuatu yang bernilai setidaknya

	untuk salah satu <i>actor</i> [24].
	<i>Actor</i> dan <i>use case</i> yang berinteraksi satu sama lain dihubungkan oleh asosiasi komunikasi [24].
	Tanda panah bersifat opsional, tetapi jika digunakan, panah menunjukkan bahwa terdapat suatu elemen yang memulai interaksi. Elemen yang memulai interaksi berada di ujung garis yang tumpul [24].
	Relasi <i>include</i> memberikan kemampuan untuk mengekstrak bagian umum dari dua atau lebih deskripsi <i>use case</i> dan menempatkannya dalam <i>use case</i> terpisah yang dapat dirujuk [24].


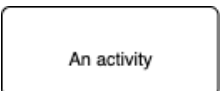
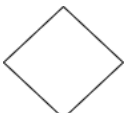


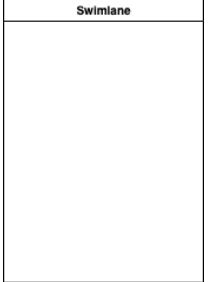
2.2.13 Activity Diagram

Activity diagram digunakan untuk menspesifikasikan bagaimana sebuah sistem dapat mencapai tujuan. *Activity diagram* menampilkan proses yang terjadi di sebuah sistem secara umum. Sebagai contoh, *activity diagram* dapat digunakan untuk menggambarkan proses atau tahap-tahap dalam membuat akun blog [25].

Activity diagram pada dasarnya adalah sebuah *flowchart*, yang menampilkan alur kontrol dari suatu aktivitas ke aktivitas lainnya. Tidak seperti *flowchart* tradisional, *activity diagram* menampilkan konkurensi sebagai cabang kendali. Sebagian besar, *activity diagram* melibatkan pemodelan langkah-langkah sekuensial (dan mungkin saja bersamaan) dalam proses komputasi. Diagram ini juga dapat memodelkan aliran nilai di antara langkah-langkah [26].

Notasi-notasi dalam *activity diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Elemen *Activity Diagram*

Elemen	Arti
	Merepresentasikan status awal atau titik mulai aktivitas [27].
	Merepresentasikan sebuah aktivitas pada sistem. Aktivitas biasanya diawali dengan kata kerja [28].
	Merepresentasikan sebuah pengambilan keputusan (<i>decision</i>) dalam alur logika pemrograman [28].
	Digunakan sebagai penghubung antar elemen [28].
	Merepresentasikan status akhir atau titik akhir aktivitas [28]
	Merupakan wadah dari setiap aktivitas atau proses yang terjadi [28].

2.2.14 Black-Box Testing

Black-box testing memperhatikan perilaku eksternal, spesifikasi, dan hasil akhir yang diinginkan yang dihasilkan oleh suatu perangkat lunak dengan memberikan sekumpulan parameter *input*. Tujuan utama *black-box testing* adalah untuk memverifikasi perangkat lunak dengan menggunakannya tanpa memiliki pengetahuan tentang cara kerja internal sistem yang diuji. *Black-box testing* membantu penguji untuk mengidentifikasi apakah perangkat

lunak memenuhi semua persyaratan yang dinyatakan dan tidak dinyatakan dan berperilaku sesuai perspektif pengguna akhir (*end-user*) [29]. Ada berbagai teknik yang dapat digunakan dalam jenis pengujian ini:

- **Analisis spesifikasi kebutuhan:** Mengkonfirmasi apakah perangkat lunak berperilaku seperti yang ditentukan dalam dokumen spesifikasi kebutuhan. Dapat dijangkau dan tersedia bagi pengguna untuk digunakan, berperilaku konsisten dan akurat. Penguji menyiapkan matriks ketertelusuran (*traceability matrix*), di mana mereka mengonfirmasi bahwa skenario pengujian mereka telah mencakup semua persyaratan yang dinyatakan [29].
- **Pengujian positif dan negatif:** Pengujian positif mengacu pada memvalidasi semua skenario positif atau istilah lainnya *happy-path*. Pengujian positif memverifikasi apakah alur kerja *end-to-end*, atau bagian dari alur kerja, berfungsi seperti yang diharapkan. Pengujian negatif adalah kebalikan dari pengujian positif, di mana tujuannya adalah untuk menunjukkan bahwa aplikasi tidak berperilaku seperti yang diharapkan. Dalam pengujian negatif, penguji harus datang dengan serangkaian parameter *input*, atau kondisi di mana aplikasi tidak akan bertahan dan rusak [29].
- **Boundary-value analysis:** Ketika pengujian dilakukan pada tingkat batas, atau pada batas ekstrim (pinggiran), ini disebut sebagai analisis nilai batas. Teknik ini sangat efektif untuk menemukan cacat. Ini adalah kondisi di mana batasan fungsi aplikasi diidentifikasi dan menambahkan pengujian di sekitar batasan tersebut memberikan hasil positif atau negatif. Jika bekerja di sekitar kondisi tersebut, itu berarti tindakan

pencegahan telah dilakukan oleh pengembang, dan jika tidak, penguji mencatatnya sebagai cacat [29].

Contoh nilai batas adalah membuat daftar kata sandi yang menerima huruf (A-Z) dan angka (0-9) dengan panjang minimum 6 dan maksimum 14 (yaitu, kondisi validasi, jika panjang variabel ≤ 6 dan ≥ 14 , artinya salah). Dalam hal ini, penguji dapat menguji daftar ini dengan membuat kata sandi sebagai berikut:

- ☐ 5 karakter
- ☐ 6 karakter
- ☐ 7 karakter
- ☐ 13 karakter
- ☐ 14 karakter
- ☐ 15 karakter

Ini membantu penguji untuk mengidentifikasi apakah memungkinkan pengguna membuat kata sandi di bawah atau di atas rentang batas yang ditentukan [29].

- ☐ ***Equivalence partitioning:*** Melibatkan pembuatan sekumpulan kecil nilai masukkan yang dapat membantu menghasilkan sekumpulan hasil yang berbeda. Hal ini membantu cakupan pengujian dan mengurangi pekerjaan penguji dengan memverifikasi setiap nilai masukan. Partisi ini dapat terdiri dari himpunan nilai yang sama, nilai yang berbeda, atau himpunan nilai dengan kondisi ekstrim [29].

Misalnya, perusahaan asuransi memiliki tiga jenis penawaran berlangganan berdasarkan usia pengguna: harganya \$100 per bulan jika mereka berusia di bawah 18 tahun, \$250 jika

mereka berusia antara 19-40, dan \$150 jika mereka lebih tua dari 41. Dalam hal ini, kumpulan nilai input dapat terdiri dari tes data pengguna berusia dalam rentang 0-18, 18-20, 19-39, 35-40, 40-42, dan di atas 41. Itu juga dapat memiliki beberapa parameter input yang tidak valid, di mana usia adalah 0, -1, satu set huruf (ABCD), nilai titik desimal (33,45), nilai tiga atau empat digit (333 atau 5.654), dan seterusnya [29].

2.2.15 *White-Box Testing*

White-box testing dilakukan pada kode aplikasi perangkat lunak. *White-box testing* melibatkan verifikasi fungsi, loop, pernyataan, strukturnya, aliran data, hasil keluaran yang diharapkan berdasarkan serangkaian nilai input tertentu, serta desain internalnya. Sebagian darinya dicakup selama proses peninjauan kode dan pengujian unit untuk memastikan cakupan kode sesuai dengan persyaratan yang ditentukan. Cakupan pernyataan, kondisi cakupan jalur, dan cakupan fungsi adalah semua komponen cakupan kode yang membantu peninjau meninjau setiap aspek kode [29]. Dengan bantuan *white-box testing*, peneliti dapat mengidentifikasi hal-hal berikut:

- ☐ Bagian kode yang tidak dapat dijangkau, sebagian besar dibuat menggunakan pernyataan GOTO.
- ☐ Variabel (lokal atau global) yang belum pernah digunakan atau menyimpan nilai yang tidak valid.
- ☐ Kebocoran memori atau *memory leak* di mana alokasi memori dan dealokasi untuk variabel atau *pointer* telah diurus.
- ☐ Apakah suatu fungsi mengembalikan nilai dalam jenis yang benar dan format yang diharapkan.
- ☐ Apakah semua variabel, *pointer*, kelas, dan objek yang diperlukan diinisialisasi seperti yang diharapkan.

- ☐ Apakah kode dapat dibaca dan mengikuti konvensi pengkodean organisasi.
- ☐ Apakah kode yang baru ditambahkan berfungsi seperti yang diharapkan dengan bagian kode yang sudah ada.
- ☐ Apakah aliran data berurutan dan akurat.
- ☐ Efisiensi dan kinerjanya untuk mengoptimalkan kode.
- ☐ Pemanfaatan sumber daya.
- ☐ Apakah semua persyaratan konfigurasi telah dipenuhi dan menyertakan semua dependensi untuk menjalankan komponen atau seluruh aplikasi.

2.2.16 Integration Testing

Perangkat lunak apa pun terbuat dari modul atau komponen yang berbeda dan/atau digunakan bersama dengan perangkat lunak lain. Untuk memastikan bahwa dua atau lebih individu dan unit atau komponen independen bekerja sama dengan lancar, pengujian melakukan pengujian integrasi. Ini menegaskan bahwa data di berbagai komponen sistem atau dua sistem terpisah mengalir dengan lancar. Contoh pengujian integrasi adalah situs belanja daring tempat memilih barang yang ingin dibeli dan membayar secara daring menggunakan opsi *internet-banking* [29].

2.2.17 Regression Testing

Poin utama dari pengujian regresi adalah untuk memverifikasi bahwa kode yang baru dikembangkan atau versi kode yang diperbarui tidak memiliki efek buruk pada bagian aplikasi yang ada dan berfungsi. Terkadang, bagian aplikasi atau fitur yang baru dirancang bekerja dengan sempurna tetapi merusak fungsi kerja yang ada. Di sinilah pengujian regresi muncul.

Pengujian regresi sebagian besar dilakukan pada akhir siklus pengujian untuk memastikan bahwa seluruh aplikasi—setelah

membuat perubahan kode beberapa kali karena perbaikan *bug* atau pemutakhiran komponen kode atau basis data apa pun—masih memberikan hasil yang diinginkan. Sebagian besar waktu, penguji menggunakan skrip otomatis untuk melakukan pengujian regresi secara berulang pada aplikasi. Alat seperti HP-UFT, TestComplete, Eggplant, atau Selenium dengan JUnit hingga NUnit sangat berguna untuk jenis pengujian ini [29].

2.2.18 *User Acceptance Testing* (UAT)

Mengonfirmasi apakah produk atau layanan perangkat lunak dapat diterima dan berfungsi sesuai harapan pengguna akhir. Sebagian besar organisasi memiliki pengujian penerimaan pengguna atau *User Acceptance Testing* (UAT) sebagai fase pengujian terpisah, yang umumnya dilakukan oleh sekelompok kecil pengguna akhir atau klien. Tujuannya adalah untuk memverifikasi bahwa produk perangkat lunak berfungsi dan memenuhi kebutuhan pelanggan, aman digunakan, dan tidak memiliki efek buruk pada pengguna akhir. Ini memberi tim pengembangan kesempatan untuk memasukkan fitur yang hilang atau permintaan peningkatan sebelum merilis produk ke khalayak yang lebih luas. Pada tahap ini, klien masih bisa menolak produk atau fiturnya. Saat pengujian dilakukan di dalam organisasi, meniru penyiapan lingkungan dunia nyata, ini disebut sebagai pengujian alfa. Saat uji penerimaan dilakukan oleh pengguna akhir di lingkungannya sendiri, ini disebut sebagai pengujian beta. Dalam jenis pengujian ini, tim pengembangan tidak terlibat dengan pengguna akhir yang sebenarnya. Ini adalah tes yang bagus untuk berbagi versi beta produk dengan kelompok pengguna akhir aktual yang relatif kecil sehingga mereka dapat memverifikasi produk, fungsinya, dan fiturnya [29].

Namun, saat merilis versi beta, penting untuk mencantumkan persyaratan perangkat keras atau perangkat lunak. Bersamaan dengan itu, tim eksekutif pendukung yang berdedikasi harus tersedia untuk menjawab pertanyaan pelanggan. Selain itu, versi perangkat lunak ini dapat tersedia secara gratis untuk waktu yang terbatas (umumnya, dua minggu hingga satu bulan) untuk mendorong lebih banyak orang berpartisipasi dalam pengujian yang sebenarnya [29].

Teknik UAT merupakan pengujian terakhir sebelum sistem dipakai oleh pengguna yang melibatkan pengujian dengan data pengguna sistem. Perhitungan dilakukan menggunakan skala likert. Skala likert merupakan suatu skala penilaian yang menyajikan pilihan skala dengan nilai pada setiap skala untuk mengukur tingkat persetujuan terhadap sesuatu. Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok tentang kejadian atau gejala sosial dimana setiap pertanyaan memiliki bobot nilai. Adapun bobot jawaban yang diberikan adalah sangat baik (5), baik (4), cukup baik (3), kurang baik (2), tidak baik (1). Perhitungan dilakukan dengan rumus berikut [30]:

$$L = \frac{((SB \times 5) + (B \times 4) + (CB \times 3) + (KB \times 2) + (TB \times 1))}{NB} \times 100$$

BAB III

METODOLOGI PENELITIAN

3.1 Objek dan Subjek Penelitian

Berdasarkan latar belakang masalah yang telah diuraikan pada bab 1, subjek yang digunakan dalam penelitian ini adalah fitur reservasi ruang diskusi di ITTP *Library*. Sedangkan objek penelitiannya diambil berdasarkan batas masalah yang sudah dijelaskan pada bab 1, yaitu web reservasi yang berbasis *plugin* SLiMS.

3.2 Alat dan Bahan Penelitian

Dalam melakukan penelitian, peneliti memerlukan alat dan bahan untuk menunjang proses perancangan sistem informasi ini. Alat dan bahan dapat dilihat pada poin-poin sebagai berikut.

3.2.1 Spesifikasi Perangkat Keras

Laptop yang digunakan untuk pengembangan web reservasi memiliki spesifikasi yang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Processor	Processor Intel Core i7 2,6 GHz 6-Core
RAM	RAM 16.00 GB
Graphics	Graphic Card Intel UHD Graphics 630 1536 MB
SSD	512 GB

3.2.2 Perangkat Lunak

Daftar perangkat lunak beserta fungsinya yang digunakan untuk pengembangan web reservasi dapat dilihat pada Tabel 3.2.

Tabel 3.2 Daftar Perangkat Lunak

Perangkat Lunak	Fungsi
MacOS Ventura	Sebagai sistem operasi laptop.
SLiMS	Sebagai basis web reservasi.
PHP	Sebagai bahasa pemrograman.
Bootstrap	Sebagai <i>framework</i> yang berfungsi untuk membangun tampilan web.
Visual Studio Code	Sebagai <i>text editor</i> .
XAMPP	Sebagai <i>local web server</i> yang berfungsi untuk mendukung proses pengembangan web dan sebagai tempat berjalannya kode program PHP.
MySQL	Sebagai basis data yang akan menyimpan data reservasi
Google Chrome	Sebagai <i>browser</i> yang digunakan untuk menampilkan hasil pengembangan sistem

3.2.3 Bahan

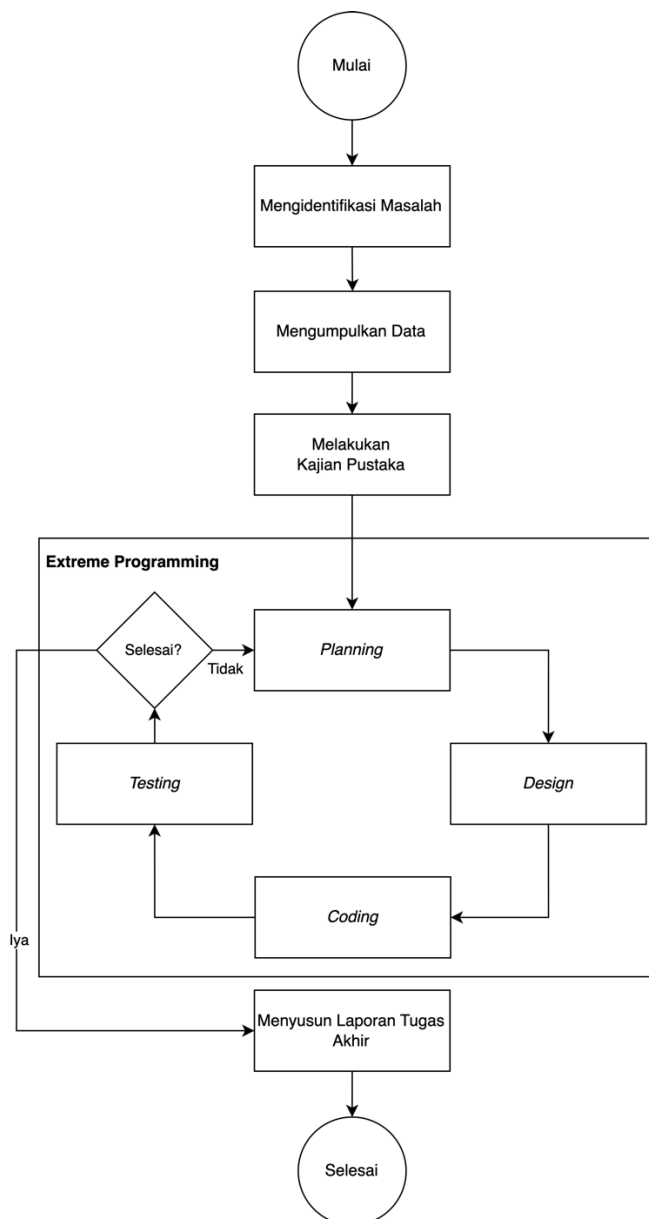
Bahan penelitian yang digunakan untuk pengembangan web reservasi dapat dilihat pada Tabel 3.3.

Tabel 3.3 Bahan Penelitian

Data	Penjelasan
Sampel data hasil survei	Merupakan data hasil survei terkait penilaian pengguna terhadap sistem reservasi Ruang Diskusi ITTP Library
Data anggota perpustakaan	Merupakan data yang berisi nama dan NIM mahasiswa yang telah terdaftar sebagai anggota perpustakaan.
Data reservasi	Merupakan data daftar anggota perpustakaan yang melakukan reservasi. Data ini berisi nama, NIM, program studi, dan tanggal melakukan reservasi.

3.3 Diagram Alir Penelitian

Untuk mempermudah pelaksanaan penelitian, tahapan penelitian disusun berdasarkan alur tahapan SDLC dengan metode XP. Alur penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.2.4 Mengidentifikasi masalah

Pada tahap awal penelitian, langkah pertama yang peneliti lakukan adalah mengamati permasalahan yang ada pada proses reservasi Ruang Diskusi. Peneliti melakukan wawancara semi terstruktur kepada beberapa mahasiswa terkait pengalaman mereka dalam melakukan reservasi Ruang Diskusi.

3.2.5 Mengumpulkan data

Setelah mengidentifikasi masalah, peneliti melakukan survei kepada mahasiswa yang pernah melakukan reservasi. Survei tersebut berisi pertanyaan seputar penilaian mahasiswa terhadap sistem reservasi yang ada saat ini. Survei ini juga berisi pertanyaan terkait kebutuhan akan pengembangan web reservasi. Kesimpulan dari survei ini adalah mahasiswa merasa kesulitan pada saat melakukan reservasi dan perlu adanya pengembangan web reservasi guna memudahkan proses reservasi. Untuk detail hasil survey dapat dilihat pada **Lampiran 1**.

Selain melakukan survei, pengumpulan data dilakukan melalui wawancara semi terstruktur kepada pustakawan dan vendor yang bertanggung jawab dalam mengembangkan situs *D-Library*. Wawancara ini dilaksanakan secara bebas dan tidak menggunakan daftar pertanyaan untuk melakukannya. Tujuan dari wawancara ini untuk mendapatkan wawasan terkait pengembangan web. Dengan ini peneliti hanya perlu mendengarkan dan mencatat informasi yang diberikan oleh informan.

3.2.6 Melakukan kajian pustaka

Pada tahap ini, peneliti mempelajari ilmu dan informasi yang akan mendukung penelitian. Peneliti membaca penelitian terdahulu, buku, jurnal, dokumentasi teknologi terkait, dan literatur lainnya. Hasil dari tahap atau proses ini peneliti tuliskan pada bab 2.

3.2.7 *Planning*

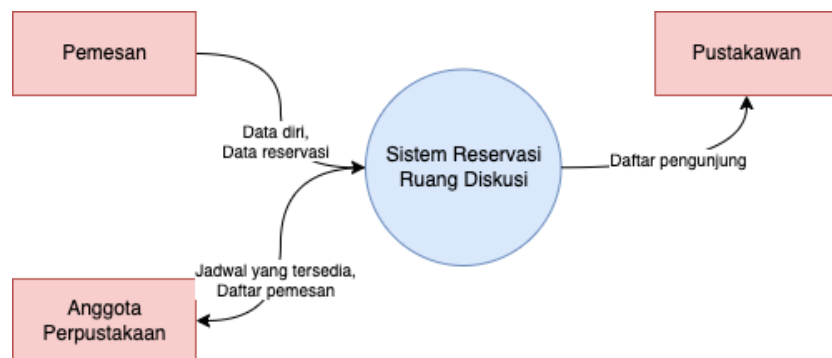
Setelah melakukan kajian pustaka, peneliti mulai melakukan perencanaan untuk mengembangkan web reservasi Ruang Diskusi. Pada tahap ini peneliti membuat dokumen *user stories*, *acceptance criteria*, dan *requirements*. Selain itu, pada tahap ini peneliti juga merencanakan jadwal penelitian.

3.2.8 *Design*

Setelah membuat perencanaan, peneliti melakukan perancangan sistem informasi guna menjelaskan proses bisnis. Hasil dari proses perancangan sistem adalah metafora sistem yang berupa *Data Flow Diagram* (DFD) dan *Unified Modeling Language* (UML). Berikut metafora yang sudah penulis buat:

1. DFD

Menggambarkan aliran data yang ada pada sistem. Gambar 3.2 merupakan *data flow diagram* dari sistem reservasi ruang diskusi.



Gambar 3.2 DFD Sistem Reservasi Ruang Diskusi

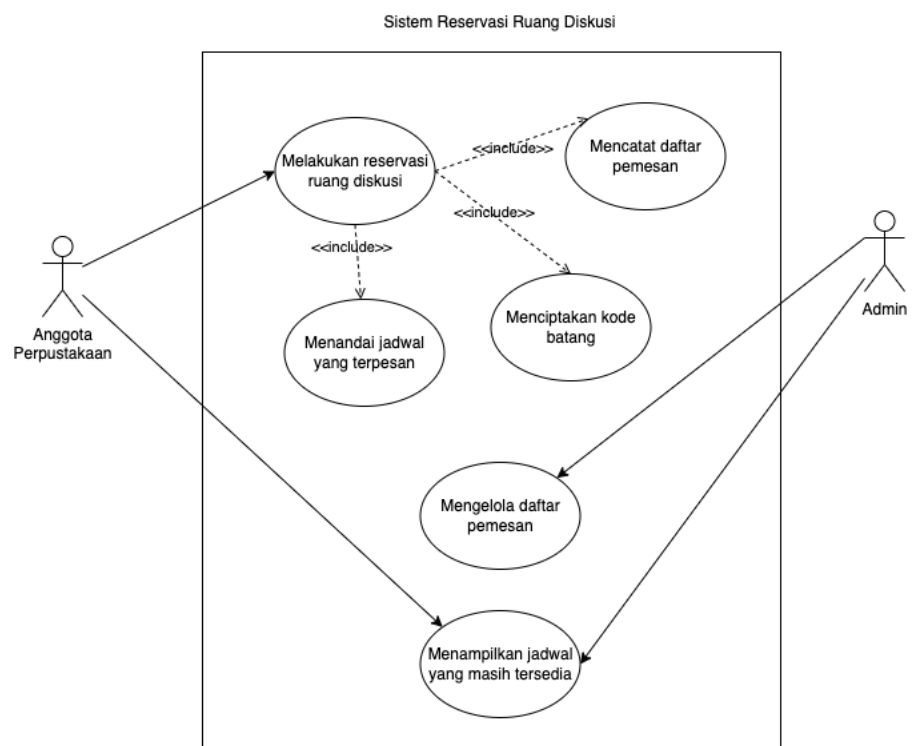
Pemesan ruang diskusi akan memasukkan data diri dan data reservasi. Data diri berupa NIM, nama lengkap, program studi, nomor telepon. Data reservasi berupa tanggal dan jam penggunaan ruang diskusi, jumlah anggota, NIM anggota, Nama anggota, dan jenis kegiatan yang akan dilakukan. Pemesan juga dapat melampirkan surat peminjaman ruang jika ingin menggunakan ruangan selama lebih dari dua jam.

Anggota perpustakaan dapat melihat jadwal ruang diskusi baik itu yang tersedia maupun yang tidak. Anggota perpustakaan juga dapat melihat daftar nama yang memesan ruang diskusi. Di sisi lain, sistem mengelola daftar pemesan ruang diskusi dengan menandai mereka sebagai pengunjung perpustakaan. Pengunjung

perpustakaan akan mendapatkan *barcode* kunjungan. *Barcode* kunjungan digunakan sebagai tanda kehadiran anggota perpustakaan yang nantinya akan di-*scan* oleh pustakawan.

2. Use Case Diagram

Peneliti menggambarkan apa yang dapat dilakukan oleh sistem. Gambar 3.3 merupakan *use case diagram* dari sistem reservasi ruang diskusi.



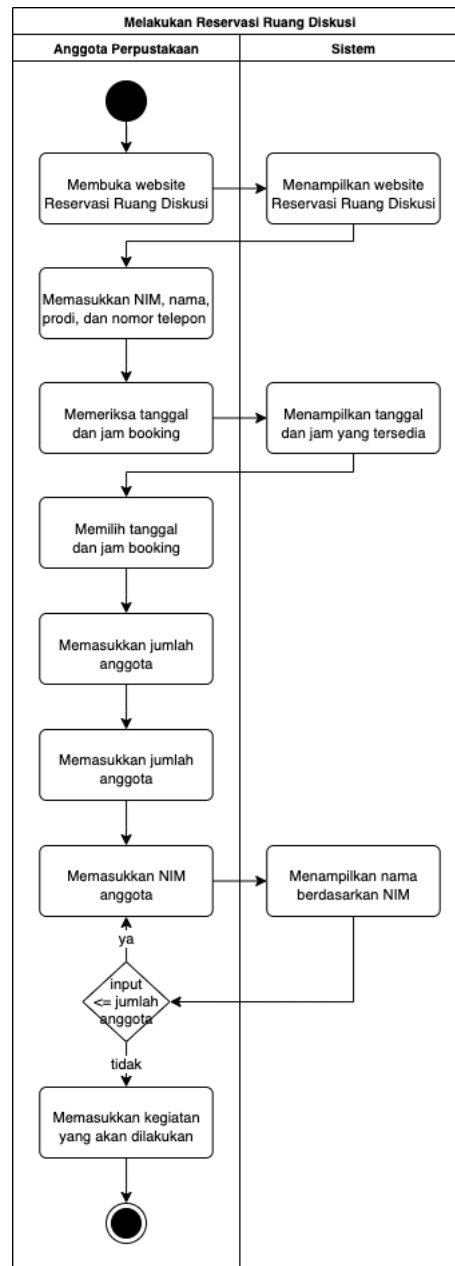
Gambar 3.3 Use Case Sistem Reservasi Ruang Diskusi

Seperti yang tertera pada Gambar 3.3, anggota perpustakaan dapat melakukan reservasi. Setelah melakukan reservasi, sistem melakukan pencatatan daftar pesanan. Kemudian sistem akan menandai jadwal atau tanggal yang telah dipesan dengan tanda *reserved*. Selanjutnya, sistem akan mencetak barcode yang akan digunakan untuk reservasi kunjungan.

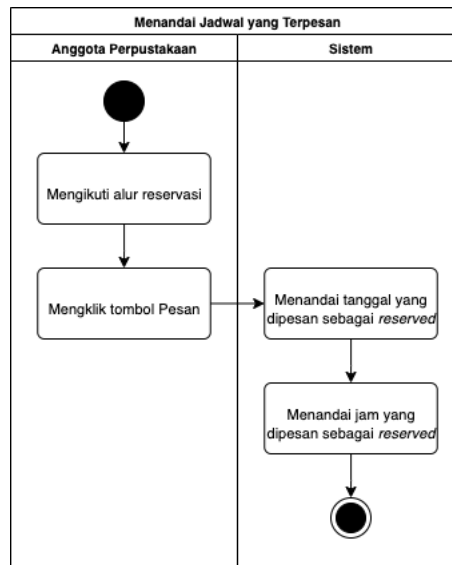
Sistem dapat menampilkan daftar pemesan. Data ini akan dikelola oleh admin atau pustakawan sebagai riwayat pemesanan ruang diskusi. Selain itu, karena sistem menandai tanggal yang telah dipesan sebagai *reserved*, maka sistem dapat menampilkan jadwal yang masih tersedia kepada calon pemesan.

3. Activity Diagram

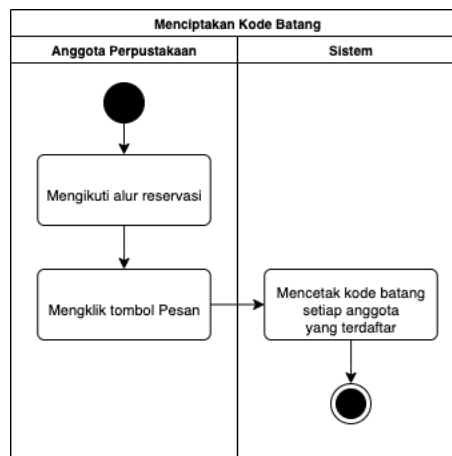
Terdapat 5 activity diagram pada perancangan sistem reservasi ruang diskusi. Gambar 3.4 menggambarkan aktivitas pengguna ketika sedang melakukan reservasi. Gambar 3.5 menggambarkan aktivitas sistem ketika sedang menandai jadwal yang telah dipesan sebagai *reserved*. Gambar 3.6 menggambarkan aktivitas sistem ketika menciptakan kode batang. Gambar 3.7 menggambarkan aktivitas sistem ketika mencatat riwayat daftar pemesan ruang diskusi. Gambar 3.8 menggambarkan aktivitas sistem ketika mengelola data daftar pemesan. Gambar 3.9 menggambarkan aktivitas sistem ketika menampilkan jadwal yang masih tersedia.



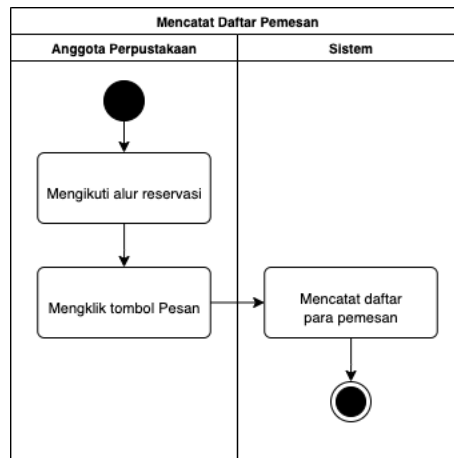
Gambar 3.4 AD – Melakukan Reservasi Ruang Diskusi



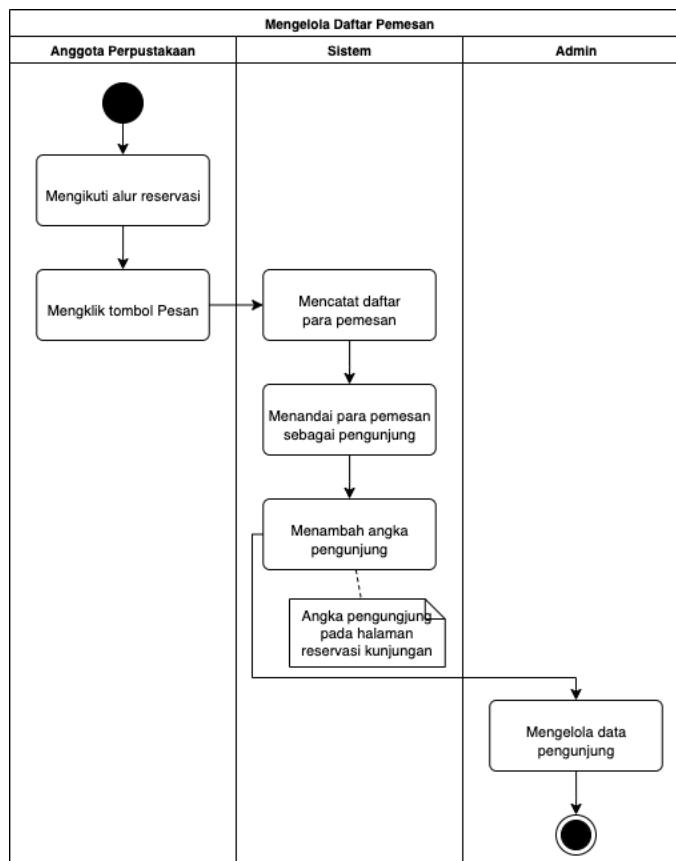
Gambar 3. 5 AD – Menandai Jadwal yang Terpesan



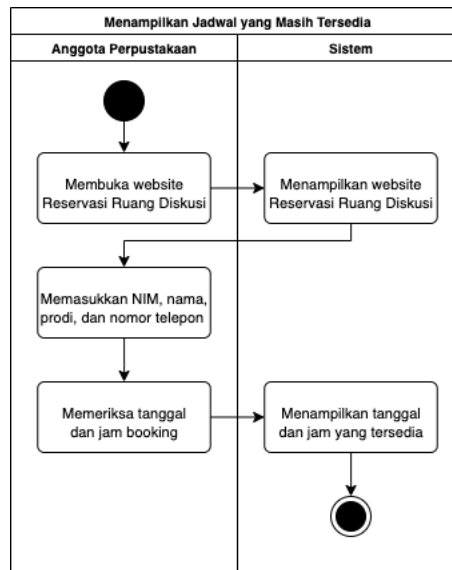
Gambar 3.6 AD – Menciptakan Kode Batang



Gambar 3.7 AD – Mencatat Daftar Pesanan



Gambar 3.8 AD – Mengelola Daftar Pemesanan



Gambar 3.9 AD – Menampilkan Jadwal yang Masih Tersedia

3.2.9 Coding

Pada tahap ini, peneliti akan membuat tugas harian (*tasks*), berdiskusi dengan pihak-pihak terkait secara berkesinambungan, dan melakukan pengkodean. Jika tugas harian (*tasks*) tidak selesai di hari yang sama, maka peneliti akan mengerjakan tugas tersebut di hari berikutnya. Peneliti juga melakukan pengujian dengan jenis pengujian *white-box testing*. Peneliti membuat *unit test* setiap terdapat fitur baru. Peneliti memastikan bahwa *unit test* yang dibuat berhasil dijalankan.

3.2.10 Testing

Pada tahap pengujian perangkat lunak, peneliti melakukan *user acceptance test* (UAT) berdasarkan skenario pengujian atau *acceptance test criteria*. Jika pengujian tidak berhasil, artinya UAT gagal. Kegagalan pengujian ini dapat disebabkan oleh *bug* atau perangkat lunak belum menyelesaikan masalah pengguna. Sehingga peneliti perlu kembali ke tahap *planning* dan melakukan iterasi untuk memperbaiki UAT yang gagal. Jika UAT berhasil, maka perangkat lunak tidak terdapat *bug* dan sudah menyelesaikan masalah pengguna. Setelah itu, peneliti akan merilis perangkat lunak.

3.2.11 Penulisan laporan tugas akhir

Setelah pengguna menyetujui perangkat lunak yang telah dikembangkan artinya perangkat lunak sudah menyelesaikan permasalahan yang telah disinggung pada bab 1. Perangkat lunak atau web Reservasi Ruang Diskusi sudah dapat dirilis. Pada tahap ini, peneliti akan menyelesaikan laporan tugas akhir.

3.4 Jadwal Penelitian

Tabel 3.4 Jadwal Penelitian

Jenis Kegiatan	Bulan I Minggu Ke-				Bulan II Minggu Ke-				Bulan III Minggu Ke-				Bulan IV Minggu Ke-				Bulan V Minggu Ke-				Indikator
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
Tahap I – Semester 6																					
1. Mengidentifikasi masalah dan mengumpulkan data																					Menemukan judul tugas akhir.
2. Melakukan kajian pustaka, <i>planning</i> , dan <i>design</i>																					Proposal penelitian disetujui oleh dosen pembimbing.
3. Ujian Proposal																					Seminar Proposal

DAFTAR PUSTAKA

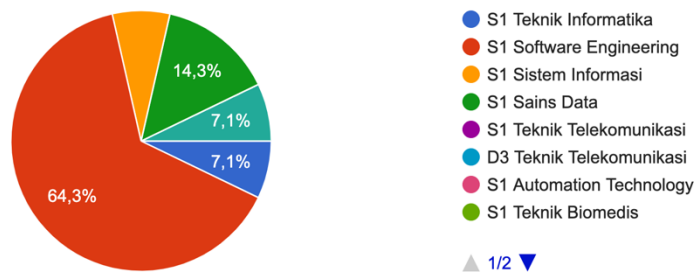
- [1] Pustakawan ITTP Library, *Perpustakaan Digital IT Telkom Purwokerto*. 2023. Accessed: Jul. 12, 2023. [Online]. Available: dlibrary.ittelkom-pwt.ac.id
- [2] SDC (Senayan Developers Community), "History," <https://slims.web.id/sdc/>, Nov. 13, 2022.
- [3] Z. Abdurrahman and T. Prabowo, "Model Pengembangan PLUG-IN SLiMS Pada Komunitas SLiMS Kudus," *Pustakaloka*, vol. 13, pp. 199–222, May 2021, doi: 10.21154/pustakaloka.v13i2.3158.
- [4] A. Stellman and J. Greene, *Learning Agile*, First edition. Beijing: O'Reilly, 2014.
- [5] I. Haddad and B. Warner, "Understanding The Open Source Development Model," <http://www.ibrahimatlinux.com/uploads/6/3/9/7/6397792/00.pdf>, 2011.
- [6] H. Rahman, H. Mansur, and A. Satrio, "Pengembangan Sistem Informasi Perpustakaan Berbasis SLiMS Akasia 8," *Journal of Instructional Technology*, vol. 2, no. 2, p. 8, 2022, doi: 10.20527/j-instech.v2i2.3820.
- [7] S. Sopingi and M. Purweni, "Virtualisasi Katalog Senayan Library Management System (SLiMS) Berbasis 3D," *Telematika*, vol. 11, no. 2, pp. 12–26, 2018, doi: 10.35671/telematika.v11i2.700.
- [8] A. Ichwani, "E-Library Interaktif dengan SLiMS Bulian Menggunakan Metode RAD," *Jurnal Komputasi*, vol. 10, no. 2, pp. 97–111, 2022, doi: 10.23960/komputasi.v10i2.3170.
- [9] I. Iskandar and L. Wijayanti, "Implementasi SLiMS di Perpustakaan Perguruan Tinggi," *JURNAL ILMU PERPUSTAKAAN (JIPER)*, vol. 4, no. 2, 2022, doi: 10.31764/jiper.v4i2.9225.
- [10] SDC (Senayan Developers Community), "About SLiMS," <https://slims.web.id/web/pages/about/>, 2023.
- [11] R. RUDIANTO, "Aplikasi Teknologi Informasi SLiMS di Perpustakaan UIN Sulthan Thaha Saifuddin Jambi," *Baitul 'Ulum: Jurnal Ilmu Perpustakaan dan Informasi*, Dec. 2017, doi: 10.30631/baitululum.v1i0.16.
- [12] M. Azwar, "Membangun Sistem Otomasi Perpustakaan dengan Senayan Library Management System (SLiMS)," *Khazanah al-Hikmah Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan*, vol. 1, pp. 7–18, Jun. 2013, doi: 10.24252/v1i1a3.
- [13] A. H. Mubarak, "Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (Unified Modeling Language) dan Bahasa Pemrograman PHP (PHP Hypertext Preprocessor) Berorientasi Objek," *JIKO (Jurnal Informatika dan Komputer)*, 2019.
- [14] N. Husin, "Peminjaman Buku Online dengan Bootstrap untuk Dinas Perpustakaan dan Kearsipan Jakarta," *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi dan Sistem Komputer*, vol. 6, no. 1, pp. 46–50, May 2022, doi: 10.55886/infokom.v6i1.455.
- [15] J. Enterprise, *PHP Edisi Lengkap: Semua ilmu dasar untuk menguasai PHP ada di buku ini*. Elex Media Komputindo, 2021.

- [16] M. Handayani, A. Z. Mardiansyah, and Jatradi, "Sistem Informasi Perpustakaan SMAN 1 BAYAN Berbasis Website Dengan PHP MYSQL," *Jurnal Begawe Teknologi Informasi (JBegaTI)*, vol. 2, Mar. 2021, doi: 10.29303/jbegati.v2i1.138.
- [17] M. Mandasari and E. Tampubolon, "Perancangan Sistem Informasi Perpustakaan Berbasis Web dengan Metode Rapid Application Development (RAD) dan Framework CSS Bootstrap," 2020.
- [18] D. J. K. Putra and P. F. Tanaem, "Perancangan Aplikasi Pembukuan Menggunakan Metode Agile Scrum," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 3, pp. 509–521, 2022, doi: 10.28932/jutisi.v8i3.5060.
- [19] D. Wells, "Extreme Programming: A Gentle Introduction." 2013. [Online]. Available: <http://www.extremeprogramming.org/>
- [20] M. Y. Handita and E. Setiawan, "Sistem Informasi Penerimaan Pegawai dengan Metode Agile Extreme Programming," *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, vol. 5, no. 2, pp. 154–164, 2022, doi: 10.37792/jukanti.v5i2.571.
- [21] S. Hidayatulloh, "Optimalisasi GitHub untuk Software Project Management dengan Memanfaatkan Notifikasi SMS," *Jurnal Informatika*, vol. 2, Mar. 2016, doi: 10.31311/ji.v2i1.64.
- [22] J. Reid, *iOS unit testing by example: XCTest tips and techniques using Swift*. Raleigh, North Carolina: The Pragmatic Bookshelf, 2020.
- [23] J. Beatty and A. Chen, *Visual models for software requirements*. Redmond, Wash.: Microsoft Press, 2012.
- [24] K. Bittner and I. Spence, *Use case modeling*. Boston, MA: Addison Wesley, 2003.
- [25] R. Miles and K. Hamilton, *Learning UML 2.0*, 1st ed. Beijing ; Sebastopol, CA: O'Reilly, 2006.
- [26] G. Booch, J. Rumbaugh, and I. Jacobson, "Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series)," *J. Database Manag.*, vol. 10, Jan. 1999.
- [27] N. R. A. Prasetyo, "Perancangan Sistem Informasi Audit Mutu Internal di Institut Teknologi Telkom Purwokerto dengan Metode Scrum," 2022. [Online]. Available: <http://repository.ittelkom-pwt.ac.id/id/eprint/7535>
- [28] D. Sudrjat, "PET PARADISE : Aplikasi Adopsi Hewan Peliharaan Berbasis Android Menggunakan Convolutional Neural Network," 2021. [Online]. Available: <http://repository.ittelkom-pwt.ac.id/id/eprint/6419>
- [29] A. Atar, *Hands-on test management with Jira: end-to-end test management with Zephyr, synapseRT, and Jenkins in Jira*. Birmingham, UK: Packt Publishing, 2019.
- [30] A. R. Yusmita, H. Anra, and H. Novriando, "Sistem Informasi Pelatihan pada Kantor Unit Pelaksana Teknis Latihan Kerja Industri (UPT LKI) Provinsi Kalimantan Barat," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 8, no. 2, p. 160, 2020, doi: 10.26418/justin.v8i2.36797.

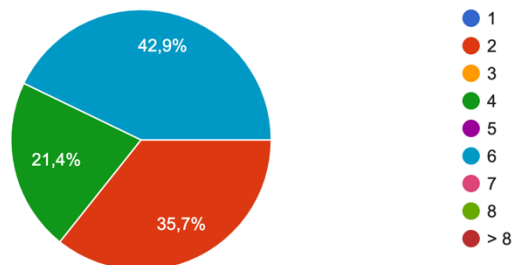
LAMPIRAN

LAMPIRAN 1. SURVEI PENILAIAN PENGGUNA TERHADAP SISTEM RESERVASI RUANG DISKUSI ITTP LIBRARY

Jurusan
14 jawaban

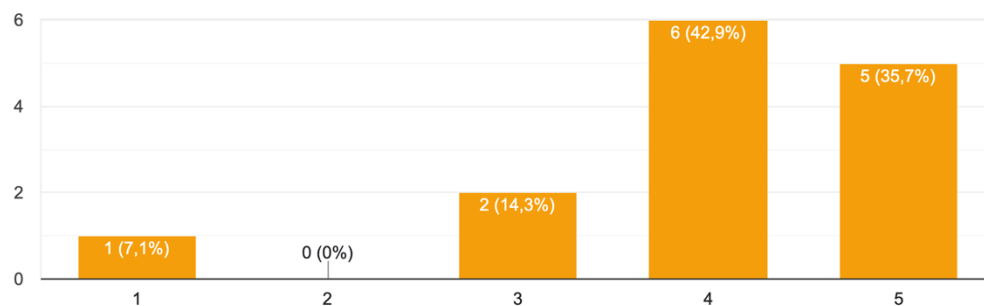


Semester
14 jawaban



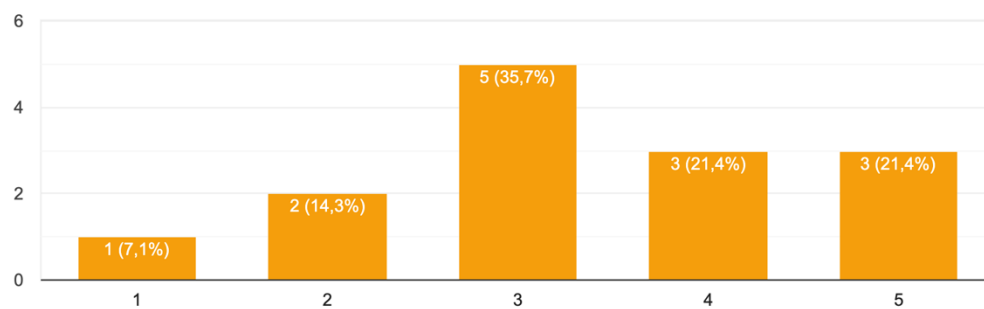
Seberapa sering Anda mengunjungi ITTP Library?

14 jawaban



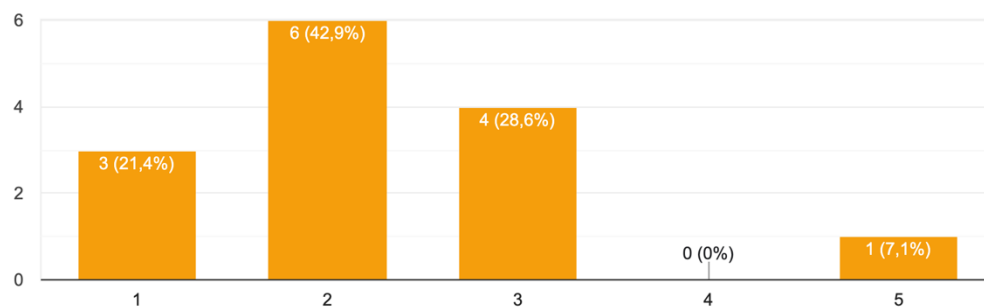
Seberapa sering Anda menggunakan fasilitas Ruang Diskusi yang ada di ITTP Library?

14 jawaban



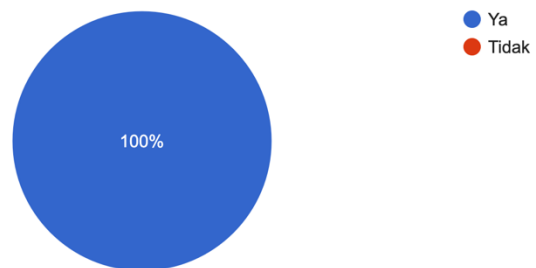
Bagaimana penilaian Anda terhadap sistem reservasi yang ada saat ini?

14 jawaban



Apakah website tersebut akan dibutuhkan sebagai pembaruan sistem reservasi ruang diskusi?

14 jawaban



LAMPIRAN 2. *REQUIREMENTS*

Functional Requirement

Sistem harus mampu menerima *input* data anggota perpustakaan yang melakukan reservasi:

- ☐ Pengguna dapat menginput NIM
- ☐ Pengguna dapat menginput Nama Lengkap
- ☐ Pengguna dapat menginput Program Studi
- ☐ Pengguna dapat menginput Nomor Telepon

Sistem harus mampu menerima *input* data detail reservasi:

- ☐ Pengguna dapat memilih Tanggal Booking
- ☐ Pengguna dapat memilih Jam Booking
- ☐ Pengguna dapat menginput surat peminjaman ruang (jika lebih dari 2 jam)
 - Cari template surat peminjaman ruang
 - Pikirkan skenario dimana anggota perpustakaan yang hendak memesan ruang diskusi lebih dari 2 jam tetapi bentrok dengan jadwal yang sudah dipesan. Misal, ada yang ingin memesan selama sehari penuh untuk kebutuhan penting, tetapi di hari itu sudah ada beberapa kelompok yang memesan.
- ☐ Pengguna dapat menginput Jumlah Anggota
- ☐ Pengguna dapat menginput NIM masing-masing anggota
- ☐ Pengguna dapat menginput Nama masing-masing anggota
- ☐ Pengguna dapat menginput Kegiatan yang akan dilakukan

Non-Functional Requirement

Acceptance:

- ☐ Sistem hanya akan mengizinkan pemesanan pada kelompok yang beranggotakan minimal 5 orang.
- ☐ Sistem hanya akan mengizinkan pemesanan pada kelompok yang beranggotakan maksimal 10 orang.

Reliability:

- ☐ Sistem hanya mengizinkan pengguna untuk memilih jadwal yang tersedia.
- ☐ Sistem harus mengizinkan pengguna untuk mendaftar kapanpun.
- ☐ Sistem harus menandai pengguna yang mendaftar sebagai pengunjung perpustakaan.
 - Setiap anggota perpustakaan yang ingin mengunjungi perpustakaan harus melakukan reservasi kehadiran terlebih dahulu.
- ☐ Sistem harus dapat menampilkan nama anggota perpustakaan berdasarkan NIM mereka.
 - Fitur *autocomplete*
- ☐ Perhatikan bagaimana jika ada anggota perpustakaan yang mendaftarkan nama tanpa seizin orangnya.
- ☐ Sistem harus dapat mendeteksi hari libur.
- ☐ Sistem harus mencetak barcode kehadiran anggota yang terdaftar.
 - Kemungkinan akan ada bentrok dengan use case reservasi kehadiran, di mana anggota yang telah melakukan reservasi kehadiran, barcode-nya akan expired 30 menit setelah daftar.

Security:

- ☐ Sistem hanya mengizinkan pengguna untuk melakukan reservasi jika pengguna sudah menyetujui kebijakan.

Catatan:

- ☐ Perlu diperhatikan *use case* jika dosen atau selain mahasiswa yang mendaftar.
- ☐ Perlu diperhatikan spamming pendaftar.

LAMPIRAN 3. *USER STORIES*

Terminologi

Reservasi kunjungan → reservasi yang wajib dilakukan seluruh anggota perpustakaan yang ingin mengunjungi perpustakaan

Reservasi ruang diskusi → reservasi yang dilakukan anggota perpustakaan yang ingin menggunakan ruang diskusi

User Stories

Story - Kebutuhan akan web reservasi ruang diskusi

Sebagai anggota perpustakaan

Saya ingin sebuah web yang dapat memudahkan proses reservasi ruang diskusi

Sehingga saya dapat melakukan reservasi dengan lebih cepat dan efektif

Story - Kebutuhan akan jadwal ruang diskusi

Sebagai anggota perpustakaan

Saya ingin web reservasi ruang diskusi menampilkan jadwal yang tersedia

Sehingga saya hanya dapat melakukan reservasi pada jadwal yang tersedia

Story - Reservasi ruang diskusi == reservasi kunjungan

Sebagai anggota perpustakaan

Saya ingin melakukan reservasi ruang diskusi sekaligus reservasi kunjungan

Sehingga saya tidak perlu lagi melakukan reservasi kunjungan pada situs yang berbeda

Story - Menambahkan anggota ruang diskusi

Sebagai anggota perpustakaan yang melakukan reservasi ruang diskusi

Saya ingin web reservasi ruang diskusi mengizinkan saya untuk mendaftarkan teman saya sebagai pengunjung ruang diskusi

Sehingga teman saya tidak perlu mendaftarkan dirinya sendiri

Story - Menambahkan anggota ruang diskusi (autocomplete)

Sebagai anggota perpustakaan yang melakukan reservasi ruang diskusi

Ketika saya mengetikkan NIM (valid) anggota perpustakaan

Saya ingin web reservasi ruang diskusi menampilkan data nama anggota perpustakaan

Sehingga saya tidak perlu mengetikkan nama mereka secara manual

Story - Sistem men-generate QR code kunjungan perpustakaan

Sebagai anggota perpustakaan yang melakukan reservasi ruang diskusi

Saya ingin web reservasi ruang diskusi men-generate QR code kunjungan untuk seluruh anggota yang didaftarkan

Sehingga seluruh anggota yang didaftarkan dapat melakukan reservasi kunjungan perpustakaan

Story - Sistem men-generate QR code kunjungan perpustakaan

Sebagai anggota perpustakaan yang didaftarkan sebagai pengunjung ruang diskusi

Saya ingin web reservasi ruang diskusi men-generate QR code kehadiran saya

Sehingga saya tidak perlu lagi melakukan reservasi kunjungan pada situs yang berbeda

Story Edge Cases

Story - Sistem cancelling

Sebagai anggota perpustakaan yang telah melakukan reservasi ruang diskusi

Saya ingin agar web reservasi ruang diskusi memungkinkan saya untuk membatalkan jadwal yang direservasi

Sehingga saya dapat membatalkan jadwal yang telah direservasi

Story - Pengunjung dosen

Sebagai dosen

Saya ingin agar web reservasi ruang diskusi memungkinkan saya untuk melakukan reservasi ruang diskusi

Sehingga saya dapat melakukan reservasi

Admin Story Edge Cases

Story - Sistem cancelling

Sebagai pustakawan

Saya ingin agar dashboard admin web reservasi ruang diskusi memungkinkan saya untuk membatalkan jadwal reservasi

Sehingga saya dapat membatalkan jadwal yang telah direservasi berdasarkan permintaan pengunjung

Story - Sistem pencatatan pengunjung ruang diskusi

Sebagai pustakawan

Saya ingin agar dashboard admin web reservasi ruang diskusi memungkinkan saya untuk membuka daftar anggota perpustakaan yang telah menggunakan fasilitas ruang diskusi

Sehingga daftar anggota tersebut dapat disimpan sebagai data

Story - Sistem reservasi melalui admin

Sebagai pustakawan

Saya ingin agar dashboard admin web reservasi ruang diskusi mengizinkan saya untuk melakukan reservasi

Sehingga saya dapat mereservasi sesuai dengan permintaan

LAMPIRAN 4. *ACCEPTANCE TEST CRITERIA*

User Stories

Story - Kebutuhan akan web reservasi ruang diskusi

GIVEN anggota perpustakaan

WHEN ingin sebuah web yang dapat memudahkan proses reservasi ruang diskusi

THEN dibuatkan web agar dapat melakukan reservasi dengan lebih cepat dan efektif

Story - Kebutuhan akan jadwal ruang diskusi

GIVEN anggota perpustakaan

WHEN ingin web reservasi ruang diskusi menampilkan jadwal yang tersedia

THEN anggota perpustakaan hanya dapat melakukan reservasi pada jadwal yang tersedia

Story - Reservasi ruang diskusi == reservasi kunjungan

GIVEN anggota perpustakaan

WHEN ingin melakukan reservasi ruang diskusi sekaligus reservasi kunjungan

THEN anggota perpustakaan tidak perlu lagi melakukan reservasi kunjungan pada situs yang berbeda

Story - Menambahkan anggota ruang diskusi

GIVEN anggota perpustakaan yang melakukan reservasi ruang diskusi

WHEN ingin web reservasi ruang diskusi mengizinkan ia untuk mendaftarkan temannya sebagai pengunjung ruang diskusi

THEN temannya tidak perlu mendaftarkan dirinya sendiri

Story - Menambahkan anggota ruang diskusi (autocomplete)

GIVEN anggota perpustakaan yang melakukan reservasi ruang diskusi

AND anggota perpustakaan mengetikkan NIM (valid)

WHEN ingin web reservasi ruang diskusi menampilkan data nama anggota perpustakaan

THEN anggota perpustakaan tidak perlu mengetikkan nama mereka secara manual

Story - Sistem men-generate QR code kunjungan perpustakaan

GIVEN anggota perpustakaan yang melakukan reservasi ruang diskusi

WHEN ingin web reservasi ruang diskusi men-*generate* QR code kunjungan untuk seluruh anggota yang didaftarkan

THEN seluruh anggota yang didaftarkan dapat melakukan reservasi kunjungan perpustakaan

Story - Sistem men-generate QR code kunjungan perpustakaan

GIVEN anggota perpustakaan yang didaftarkan sebagai pengunjung ruang diskusi

WHEN ingin web reservasi ruang diskusi men-*generate* QR code kehadirannya

THEN anggota perpustakaan tidak perlu lagi melakukan reservasi kunjungan pada situs yang berbeda

Story Edge Cases

Story - Sistem cancelling

GIVEN anggota perpustakaan yang telah melakukan reservasi ruang diskusi

WHEN ingin agar web reservasi ruang diskusi mengizinkan ia untuk membatalkan jadwal yang direservasi

THEN anggota perpustakaan dapat membatalkan jadwal yang telah direservasi

Story - Pengunjung dosen

GIVEN dosen

WHEN ingin agar web reservasi ruang diskusi mengizinkan ia untuk melakukan reservasi ruang diskusi

THEN dosen dapat melakukan reservasi

Admin Story Edge Cases

Story - Sistem cancelling

GIVEN pustakawan

WHEN ingin agar dashboard admin web reservasi ruang diskusi mengizinkan ia untuk membatalkan jadwal reservasi

THEN pustakawan dapat membatalkan jadwal yang telah direservasi berdasarkan permintaan pengunjung

Story - Sistem pencatatan pengunjung ruang diskusi

GIVEN pustakawan

WHEN ingin agar dashboard admin web reservasi ruang diskusi mengizinkan ia untuk membuka daftar anggota perpustakaan yang telah menggunakan fasilitas ruang diskusi

THEN daftar anggota tersebut dapat disimpan sebagai data

Story - Sistem reservasi melalui admin

GIVEN pustakawan

WHEN ingin agar dashboard admin web reservasi ruang diskusi mengizinkan ia untuk melakukan reservasi

THEN pustakawan dapat mereservasi sesuai dengan permintaan

LAMPIRAN 5. *WIREFRAME*

