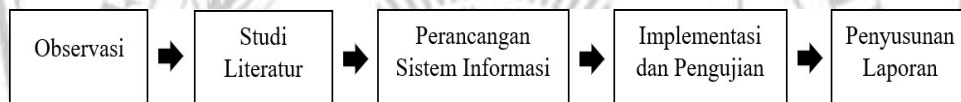


BAB III

METODE PENELITIAN

Penelitian ini dimulai dengan melakukan observasi pada unit perlengkapan pelayanan peminjaman Auditorium Universitas Muhammadiyah Malang untuk mengetahui kebutuhan dari sistem peminjaman yang akan dibangun. Setelah melakukan observasi, peneliti mulai melakukan studi literatur untuk mendukung penelitian yang akan dilakukan. Selanjutnya peneliti mulai melakukan perancangan sistem dengan menggunakan metode *Personal Extreme Programming* (PXP). Setelah sistem peminjaman berhasil dibangun, maka peneliti akan melakukan penerapan dan pengujian sistem peminjaman di unit perlengkapan pelayanan peminjaman Auditorium Universitas Muhammadiyah Malang yang kemudian hasil dari pengujian tersebut akan disajikan dalam bentuk laporan. Rangkaian metode penelitian diatas akan dijelaskan melalui penelitian pada Gambar 3.1.



3.1 Observasi

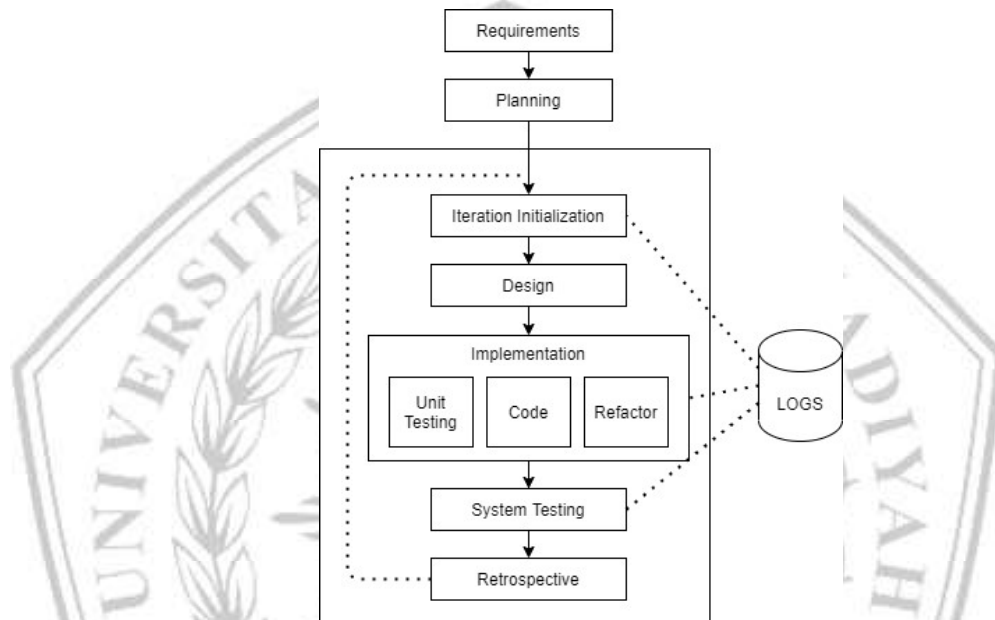
Observasi dilakukan dengan mendatangi langsung unit pelayanan peminjaman kunci ruangan yaitu Perlengkapan samping BAU. Observasi ini dilakukan untuk melihat kebutuhan sistem peminjaman yang akan dibangun dan diimplementasikan pada unit pelayanan peminjaman Auditorium agar sistem peminjaman ini menjadi tepat guna baik bagi unit pelayanan maupun bagi mahasiswa, pegawai, dan orang luar yang menggunakannya.

3.2 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan dan membaca literatur terkait rancang bangun sistem informasi, sistem informasi peminjaman, penerapan metode *Personal Extreme Programming* dalam pembuatan sistem informasi, dan sistem informasi berbasis web agar penelitian yang dilakukan dapat sesuai arah dan tujuan berdasarkan referensi yang telah didapatkan sehingga menghasilkan sistem peminjaman yang tepat guna.

3.3 Perancangan Sistem Peminjaman

Pengembangan pembuatan perangkat lunak ini menggunakan metode *Extreme Programming* (XP) yang bisa dilakukan untuk membuat perangkat lunak dengan pemrograman tunggal, yang kemudian diberi nama *Personal extreme programming* (PXP). Metode PXP ini ada beberapa tahap yakni *requirements*, *planning*, *iteration initialization*, *design*, *implementation*, *sistem testing*, dan *retrospective* [7] akan dijelaskan dan ditampilkan PXP di Gambar 3.2.



Gambar 3. 2 Skema Metode Personal Extreme Programming

3.3.1 Requirements

Pada tahap *Requirement* dilakukan dengan mulai menganalisa kebutuhan penelitian yang meliputi wawancara pada pegawai unit perlengkapan pengelolaan peminjaman Auditorium Universitas Muhammadiyah Malang kampus 3 terkait fitur yang akan ditampilkan dan digunakan pada pembuatan sistem peminjaman ini.

Tabel 3. 1 Story

“Saya menginginkan sistem untuk mengelola peminjaman auditorium berbasis website, yang dapat mengelola peminjaman secara sistem supaya dapat mempermudah pendataan peminjamannya, kalau bisa peminjaman itu dapat membaca nim dan langsung keluar semua data yang diperlukan untuk mempercepat proses peminjaman. Kemudian jangan lupa menambahkan notifikasi bukti peminjaman berupa pesan untuk apa saja yang dipinjam dan waktu peminjamannya ,dan juga tambahkan laporan untuk setiap peminjaman, oh iya satu lagi tambahkan dimana saya selaku ketua perlengkapan bisa melihat apa saja yang sudah ruangan yang sudah dipinjam supaya saya dapat melihat detail peminjaman “

3.1.1.1 User Story

Berdasarkan kebutuhan user maka kebutuhan akan dibuat sebuah *user story* pada setiap permasalahan yang dialami customer. Format penulisan pada *user story* “sebagai <pengguna>, saya menginginkan sistem<tindakan> sehingga<manfaat>” Setelah proses pengumpulan kebutuhan selesai maka dapat ditarik kesimpulan bahwa pengembang akan melakukan

Tabel 3. 2 *User Story*

Kode Stories	User Stories
Story-01	Sebagai <i>user</i> , saya menginginkan sistem ini bisa mengelola ruangan untuk nantinya bisa menambahkan , melihat dan menghapus ruangan yang nantinya ruangan inilah yang akan dipinjam sehingga dapat mempermudah peminjaman.
Story-02	Sebagai <i>user</i> , saya menginginkan sistem dapat mengelola fasilitas yang ada pada setiap ruangan yang sudah ada.
Story-03	Sebagai <i>user</i> , saya menginginkan sistem dapat menyimpan data mahasiswa dan pegawai supaya nantinya mudah dalam peminjaman.
Story-04	Sebagai <i>user</i> Saya ingin sistem dapat melakukan peminjaman ruangan dan fasilitas sesuai kebutuhan peminjam.
Story-05	Sebagai <i>user</i> , saya menginginkan sistem dimana dapat melihat history peminjaman.dan pengembalian
Story-06	Sebagai <i>user</i> , saya menginginkan memiliki fitur laporan untuk pendataan peminjaman dan juga dapat didownload dan dicetak
Story-07	Sebagai <i>user</i> , saya menginginkan sistem dimana ketua perlengkapan bisa melihat history peminjaman dan pengembalian dimana saja.

3.3.2 Planning

Pada tahap *Planning* dilakukan perencanaan *user stories*, menentukan hal utama *user stories*, dan perencanaan iterasi. Kemudian dari tahap ini akan didapatkan hal utama, perkiraan waktu pengerjaan, dan urutan pengerjaan setiap

story dari berbagai *user stories* dan iterasi yang akan dibutuhkan selama pembuatan sistem tersebut. *Story point* didapatkan dengan melihat tingkat kesulitan pembuatan pada setiap *story* yang ada. Semakin tinggi *story point* maka semakin lama pengerjaan fitur tersebut.

3.3.2.1 Value Story

Sebelum memberikan *value stories* pada fase ini menentukan estimasi waktu yang dibutuhkan pada tiap bagian *user stories*. Estimasi waktu akan dipilah menjadi sebagian poin tergantung lama pembuatan semakin besar maka poin dari estimasi waktu yang dibutuhkan akan semakin besar, akan halnya hasil dari estimasi waktu yang didapat bisa dilihat pada tabel berikut:

Tabel 3. 3 Hasil Estimasi Waktu Setiap User Stories

Kode User Stories	StoryPoints	Estimasi
Story-01	3	3
Story-02	3	3
Story-03	2	2
Story-04	3	3
Story-05	2	2
Story-06	2	2
Story-07	3	3

Tabel diatas adalah hasil dari penentuan estimasi waktu yang akan dikerjakan oleh pengembang untuk setiap *user stories*. Estimasi pada setiap *story* berbeda, karena pada setiap *story* memiliki tingkat kesulitan masing-masing. Oleh karena itu, pengembang hanya memperkirakan estimasi waktu pengerjaan pada setiap *story*.

Setelah menentukan *story point* pada setiap *user stories*, pengembang dan klien akan berdiskusi untuk menentukan setiap prioritas pada setiap *user stories* yang ada. Kemudian penentuan prioritas akan dilakukan penyortiran terhadap *user stories* sesuai dengan *value* dan *risk*. Value dari *user stories* ini dapat ditentukan

dari nilai bisnis dari *user stories*. Ada 3 nilai value diantaranya, *critical*, *significant business value*, dan *nice to have*[24].

1. *Critical*, adalah *user stories* Ketika tidak adanya *user stories* tersebut di dalam sistem yang akan dibangun, maka sistemnya tidak ada artinya. Terdapat 7 *user stories* yang termasuk dalam kategori ini.
 - a. *Story-01. User stories* ini bertujuan agar sistem dapat mengelola ruangan dan fasilitas.
 - b. *Story-02. User stories* ini bertujuan agar sistem dapat mengelola fasilitas seperti menambah jumlah fasilitas.
 - c. *Story-03. User Stories* ini bertujuan agar sistem dapat mengelola data mahasiswa dan pegawai supaya nantinya Ketika meminjam ruangan akan lebih cepat untuk meminjamnya.
 - d. *Story-04. User stories* ini bertujuan agar sistem melakukan peminjaman ruangan dan fasilitas sesuai kebutuhan peminjam
 - e. *Story-05. User stories* ini bertujuan agar sistem dapat melihat history peminjaman dan pengembalian
 - f. *Story-06. User stories* ini bertujuan agar sistem memiliki fitur laporan untuk pendataan peminjaman dan juga dapat didownload dan dicetak
 - g. *Story-07. User stories* ini bertujuan agar sistem sistem dimana ketua perlengkapan bisa melihat history peminjaman dan pengembalian dimana saja
2. *Significant business value*, adalah *user stories* yang sama sekali tidak masuk kedalam *critical*, tetapi mempunyai nilai bisnis yang signifikan. Dari rencananya yang telah dibuat, tidak terdapat *user stories* yang termasuk dalam kriteria ini.
3. *Nice to have*, adalah *user stories* yang tidak mempunyai pengaruh terhadap proses bisnis didalam sistem. Dari rencana yang telah dibuat, tidak terdapat *user stories* yang termasuk dalam kriteria ini.

Tabel 3. 4 Hasil Penentuan Value User Stories

Kode User Stories	Value
<i>Story-01</i>	Critical
<i>Story-02</i>	Critical
<i>Story-03</i>	Critical
<i>Story-04</i>	Critical
<i>Story-05</i>	Critical
<i>Story-06</i>	Critical
<i>Story-07</i>	Critical

Hasil dari *value* pada tabel diatas akan dilanjutkan dengan penentuan dari *risk* ditentukan dari *risk index* berdasarkan *completeness*, *volatylity*, dan *complexity*.

Mengenai penentuan *completeness* ini ditentukan oleh pengembang yakni berlandaskan dari rincian *user stories* tersebut. Kemudian *Volatility* ini ditentukan dapat di tentukan apakah ada kemungkinan terjadinya perubahan pada *user stories*. Dan yang terakhir adalah *complexity* yakni didtentukan dari tingkat kesulitan pembuatan pada *user stories* yang ada[4].

Dari beberapa indeks tersebut menentukan adanya tingkatan dari *risk* tiap-tiap *user stories* yang terdiri dari *low* dengan nilai 0 sampai 1, *medium* dengan nilai 2 sampai 4, dan *high* dengan nilai 5 sampai 6[24]. Hasil dari perencanaan iterasi tersebut dapat dilihat pada Tabel 3.5

Tabel 3. 5 Hasil Perencanaan Iterasi

<i>Kode User Stories</i>	<i>Risk</i>			
	<i>Completeness</i>	<i>Volatility</i>	<i>Complexity</i>	<i>Total</i>
<i>Story-</i>	2	1	1	(4) <i>Medium</i>

01				
Story-02	2	2	0	(4)Medium
Story-03	2	1	1	(4)Medium
Story-04	2	3	1	(5)High
Story-05	1	1	1	(3)Medium
Story-06	2	1	1	(4)Medium
Story-07	2	0	1	(3)Medium

a. *CRC Card*

Class Responsibilities Collaboration (CRC-Card) merupakan sebuah rancangan sistem dan untuk mengetahui interaksi antara suatu objek yang terdapat pada sistem yang telah dibuat. *CRC-Card* dibagi menjadi 3 bagian yakni nama *class*, *Responsibilities*, dan *Collaborator* yang dipergunakan untuk pada kelas yang saling berhubungan[25]. *CRC-Card* ini dibuat berdasarkan *Story* yang telah dibuat. Berikut adalah gambar *CRC-Card* yang sudah dibuat dan juga penjelasannya.

Class : Admin	
Responsibility	Collaborator
<ul style="list-style-type: none"> - Menambah Fasilitas, dan Menghapus Fasilitas - Menambah, Melihat , dan Menghapus Mahasiswa - Menambah, Melihat , dan Menghapus Pegawai - Menambah Peminjaman Mahasiswa - Menambah Peminjaman Pegawai 	Admin Perlengkapan

<ul style="list-style-type: none"> - Menambah Peminjaman Umum - Menambah, Melihat , dan Menghapus Ruangan 	
---	--

Gambar 3. 3 *Crc card Admin*

3.3.3 Iteration Initialization

Pada tahap *iteration initialization* adalah tahapan pengembangan yang harus menyelesaikan sistem sebelum bertemu dengan client untuk memperlihatkan hasil dari pengerjaan sistem tersebut , kemudian pada tahap ini pengembang dan client telah melakukan diskusi sebelumnya untuk panjang iterasi bervariasi tergantung pada proyek yang akan dikerjakan. Karena, sistem ini termasuk proyek skala kecil untuk setiap fitur yang diinginkan. Pada tahap selanjutnya sistem yang diperlihatkan pada iterasi pertama dilakukan lebih dari satu kali dimana pertemuan ini untuk membahas analisa sistem, desain sistem dan pengujian sistem.

3.3.4 Design

Pada tahap *Design* dilakukan pembuatan sesuai database menggunakan dan desain antar muka sesuai dengan kebutuhan yang sudah ada dengan iterasi yang sedang dijalankan. Pengembang harus merancang sistem sesuai dengan pada fase *requirement* yang sudah ada.

3.3.5 Implementation

Pada tahap *Implementasi* ini Seluruh data pada sistem informasi ini disimpan pada *database* MySQL yang didukung dengan web yang dibangun dengan menggunakan *framework* Laravel dan bahasa pemrograman PHP. Laravel dipilih karena berdasarkan beberapa penelitian terdahulu yang sejenis *framework* ini dapat mudah menyesuaikan versi PHP hingga pada versi terbaru sehingga dapat mengoptimalkan kualitas perangkat lunak yang dibangun[26] karena memiliki gabungan fitur terbaik dari beberapa *framework* terdahulunya [21]. *Framework* ini juga diimplementasikan pada sistem informasi maupun media pembelajaran terkait akademik karena mudahnya pengembangan sistem di masa mendatang [16,[28].

Kemudian mengenai tahap ini yang terdiri dari *unit testing*, *code*, dan *refactor*. *Unit testing* adalah pengujian terhadap *user story* yang terdapat pada

masing-masing *iterasi* kemudian menganalisa hasil dari keluaran yang telah dihasilkan.

Selanjutnya ada *code* yakni perbaikan seluruh *code* apabila semua *unit testing* telah diuji sesuai dengan *user story* yang sudah ada dan berhasil. Dan pada tahap terakhir ada *refactor* merupakan proses memperbaiki struktur dalam sebuah sistem perangkat lunak tanpa harus mengubah fungsionalitas yang ada pada sistem. Artinya, pada proses *refactoring* dilakukan perubahan program untuk memperbaiki struktur, meminimalisasi program, atau untuk membuatnya mudah dimengerti[29]. Pada tahap *implementation* dilakukan sampai tidak ada kekeliruan atau kesalahan, setelah itu akan berlanjut ke tahap *system testing*.

3.3.6 System Testing

Pada tahap *System Testing* dengan teknik *User Acceptance Testing*. Pada saat pengujian *sistem testing*

a. User Acceptance Testing

User Acceptance Testing (UAT) merupakan pengujian yang dikerjakan oleh end-user dimana pengguna tersebut merupakan karyawan dari sebuah perusahaan yang secara langsung berinteraksi dengan sistem dan dilakukan pencocokan apakah sistem sudah berfungsi sesuai kebutuhan[30]. Contoh *User Acceptance Test* yang akan digunakan dapat dilihat pada tabel dibawah ini:

Tabel 3. 6 Contoh User Acceptance Test

<i>User Stories</i>	Aktivitas	Hasil Uji
1		Disetujui /Ditolak
2		Disetujui /Ditolak

3.3.7 Retrospective

Tahapan ini merupakan tahapan pengambilan kesimpulan untuk mencatat apabila kekurangan-kekurangan pada sistem maka akan dilakukan pengulangan pada tahapan iteration initialization.

3.4 Implementasi dan Pengujian Sistem Informasi

Pengujian sistem informasi yang dilakukan pada penelitian ini untuk menemukan tingkat keberhasilan pembuatan sistem ini yaitu dengan mengujikan masing-masing fitur yang ada pada sistem informasi.

3.4.1 Implementasi

Pada tahap ini implementasi dilakukan penerapan perancangan perangkat lunak ataupun basis data kedalam Bahasa pemrograman. Akan halnya kebutuhan dari implementasi adalah sebagai berikut :

Operating Sistem (OS)	: Windows 10
Memory	: 6 gb
Bahasa Pemrograman	: PHP
Database	: Mysql

3.4.2 Pengujian Sistem

Pada tahap pengujian sistem ini adalah tahap untuk uji coba sistem, dimana perangkat lunak akan di uji dan dijalankan. Berikut adalah tujuan dari pengujian perangkat lunak yang sudah dibangun.

- a. Membuat dan menyelesaikan sistem yang telah direncanakan pada tahap perancangan.
- b. Pengujian dan penulisan dokumentasi program atau prosedur dari pembuatan sistem yang telah dilakukan
- c. Memastikan bahwa pihak perlengkapan bisa menggunakan sistem dengan baik dan benar.
- d. Mengoreksi kembali apakah sistem yang telah dibuat ini kebutuhan dari pihak perlengkapan.

3.5 Penyusunan Laporan dan Kesimpulan

Laporan disusun berdasarkan kegiatan penelitian yang dilakukan dalam pengembangan sistem informasi pengelolaan peminjaman Auditorium Universitas Muhammadiyah Malang yang diantaranya yaitu kegiatan observasi, analisa kebutuhan, perancangan dan pembuatan sistem informasi, pengujian serta penerapan sistem informasi pada unit perlengkapan Universitas Muhammadiyah Malang serta kesimpulan yang dapat diambil dari seluruh kegiatan penelitian yang telah dilakukan.