BAB II
LANDASAN TEORI

BAB II
LANDASAN TEORI

2.1 Landasan Teori

Teori-teori yang mendasar sebagai penunjang proses penelitian yaitu sebagai berikut :

2.1.1 Sistem

Sistem adalah suatu jaringan kerja dari beberapa prosedur yang saling berhubungan satu sama lain, yang kemudian berkumpul bersama untuk melakukan suatu kegiatan atau untuk melakukan sasaran yang tertentu[2]. Terdapat penjelasan lain mengenai sistem yaitu suatu jaringan kerja yang terdiri dari beberapa elemen-elemen yang berkumpul bersama-sama agar dapat menyelesaikan tahapan-tahapan yang akan dicapai untuk mencapai suatu tujuan tertentu[3].

2.1.2 Informasi

Informasi (*information)* adalah hasil pengolahan data yang sudah dibentuk dan memiliki arti tertentu. Sedangkan data yaitu suatu kumpulan fakta yang menjadi bahan pengolahan untuk diolah lebih lanjut. Menurut penjelasan lainnya, informasi adalah data yang diolah menjadi sebuah bentuk yang lebih berguna dan lebih berarti bagi seseorang yang menerimanya. Data yang diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakannya disebut juga dengan informasi[4].

2.1.3 Sistem Informasi

Sistem informasi adalah sebuah sistem yang tidak dapat dilepaskan antara elemen satu dengan elemen lainnya dalam pencapaian tujuan sehingga dalam pencapaiannya menjadi suatu perencanaan kegiatan yang terstruktur dan terukur. Sedangkan di dalam penjelasan lainnya, sistem informasi adalah suatu kombinasi antara prosedur kerja, informasi, orang dan teknologi yang dijalankan untuk mencapai suatu tujuan tertentu[5].

2.1.4 Pembukuan

Pembukuan adalah suatu proses pencatatan yang dilakukan secara teratur, dan digunakan untuk mengumpulkan data dan informasi keuangan, pengertian tersebut sesuai dengan UU Nomor 28 tahun 2007 pasal 28 [6]. Pembukuan sangat dibutuhkan di dalam koperasi, karena untuk membuat sebuah laporan akhir tahun akan dibutuhkan berbagai data keuangan Jika ada pembukuan maka akan memudahkan pengurus dalam mencari data yang dibutuhkan

2.1.5 Koperasi

Koperasi secara etimologi berasl dari kata “co-operation” yang memiliki arti Kerjasama. Co yang memiliki arti Bersama dan operation yaitu bekerja. Jadi cooperation memiliki arti bekerja bersama-sama atau usaha bersama untuk kepentingan bersama. Koperasi merupakan badan yang beranggotakan orang seorang atau badan hukum koperasi yang melandaskan kegiatannya berdasarkan prinsip koperasi sekaligus sebagai gerakan ekonomi rakyat yang berdasar atas asas kekeluargaan [7].

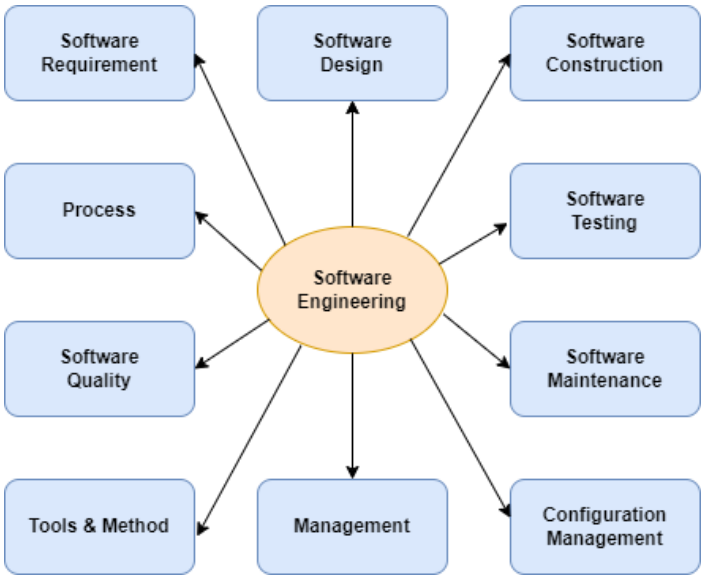
2.1.6 Koperasi Simpan Pinjam

Koperasi simpan pinjam adalah salah satu jenis bentuk usaha koperasi. Koperasi simpan pinjam adalah lembaga keuangan dengan kegiatan usaha menerima simpanan dan memberikan pinjaman uang kepada para anggotanya. Pengertian lain mengenai koperasi simpan pinjam yaitu, koperasi yang menjalankan usaha simpan pinjam sebagai salah satu usaha. Yang bertujuan untuk meningkatkan kesejahteraan anggota pada khususnya dan masyarakat pada umumnya[8].

2.1.7 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (software) adalah sebuah instruksi dalam program komputer, ketika dijalankan oleh pengguna, akan menghasilkan sebuah fungsi dan kinerja yang diharapkan. Ini mengindikasikan bahwa perangkat lunak ini dirancang untuk mengarahkan komputer agar beroperasi secara efisien sesuai dengan keinginan pengguna yang memberikan instruksi [9].

Pada Gambar 2.1 adalah gambaran ruang lingkup dari Rekayasa Perangkat Lunak (RPL) dapat digambarkan sebagai berikut [10]:



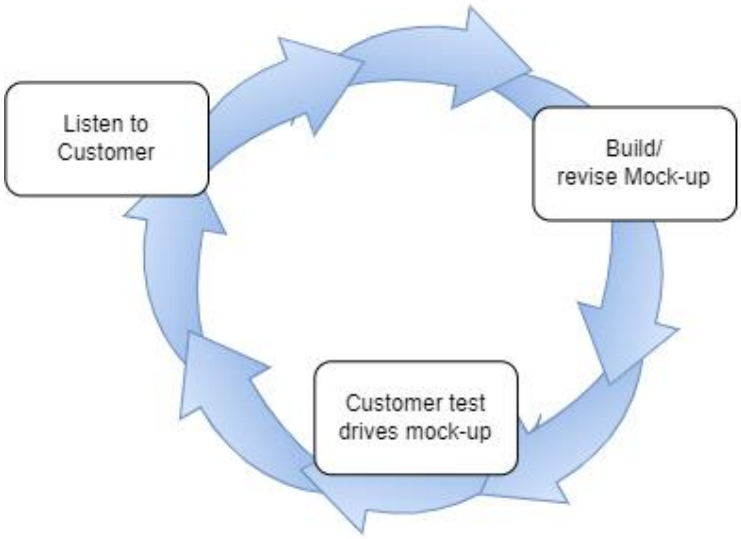
Gambar 2. 1 Ruang Lingkup Rekayasa Perangkat Lunak (RPL)

1. *Software Requirements* adalah spesifikasi dan persyaratan yang harus dipenuhi oleh sebuah perangkat lunak, termasuk fitur, fungsionalitas, kinerja, dan keterbatasan yang ditetapkan untuk memenuhi kebutuhan pengguna dan tujuan bisnis.
2. *Software design* melibatkan proses merancang arsitektur, komponen, antarmuka, dan karakteristik lain dari sebuah perangkat lunak. Ini mencakup perencanaan bagaimana perangkat lunak akan dibangun, bagaimana komponen akan berinteraksi satu sama lain, dan bagaimana antarmuka pengguna akan dirancang untuk memberikan pengalaman yang optimal kepada pengguna.
3. *Software construction* berkaitan dengan proses detail dalam pengembangan perangkat lunak, seperti pembuatan algoritma, pengkodean, pengujian, dan identifikasi kesalahan.
4. *Software testing* mencakup pengujian terhadap perilaku keseluruhan dari sebuah perangkat lunak.
5. *Software maintenance* adalah serangkaian upaya perawatan yang dilakukan setelah perangkat lunak dioperasikan.
6. *Software configuration management* (SCM) berkaitan dengan pengelolaan perubahan dalam konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu.
7. *Software engineering management melibatkan* pengelolaan dan pengukuran Rekayasa Perangkat Lunak (RPL), termasuk perencanaan proyek perangkat lunak.
8. *Software engineering tools and methods* melibatkan kajian teoritis tentang alat dan metode yang digunakan dalam Rekayasa Perangkat Lunak (RPL).
9. *Software engineering process* berkaitan dengan definisi, implementasi, pengukuran, pengelolaan, perubahan, dan perbaikan proses dalam Rekayasa Perangkat Lunak (RPL).
10. *Software quality* menekankan pada kualitas dan siklus hidup perangkat lunak.

A. Metode Pengembangan Sistem

Metode pengembangan sistem adalah suatu proses yang melibatkan perencanaan, desain, pengembangan, dan implementasi sistem baru untuk menggantikan sistem yang ada atau memperbarui sistem yang sudah ada. Pada penelitian ini pengembangan sistem menggunakan model *prototyping* menurut *Roger S. Pressman Ph.D.*

Metode *prototype* adalah sebuah pendekatan dalam pengembangan sistem di mana hasil analisis bagian-bagian sistem diterapkan secara langsung ke dalam model tanpa harus menunggu keseluruhan sistem selesai[11]. Berikut pada Gambar 2.2 adalah metode dari *prototype*.



Gambar 2. 2 Model Prototype

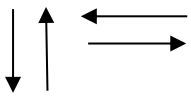
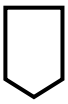
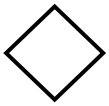
1. Mendengarkan Pelanggan/ Pengguna
   Tahap ini melibatkan pengumpulan kebutuhan sistem melalui pendengaran langsung terhadap keluhan dan keinginan dari pelanggan. Untuk memastikan sistem yang dibangun sesuai dengan kebutuhan, penting untuk memahami sistem yang sedang berjalan dan mengidentifikasi masalah yang ada.
2. Membangun dan Memperbaiki *Prototype*
   Pada langkah ini, dilakukan perancangan dan pembuatan prototipe sistem. Prototipe ini dibuat dengan mempertimbangkan kebutuhan yang telah dikumpulkan sebelumnya dari pelanggan atau pengguna.
3. Uji Coba
   Prototipe sistem diujicobakan oleh pelanggan atau pengguna pada tahap ini. Setelah itu, dilakukan evaluasi terhadap kekurangan yang mungkin muncul dari perspektif pelanggan. Pengembangan selanjutnya melibatkan kembali pendengaran terhadap keluhan pelanggan untuk memperbaiki prototipe yang telah ada.

B. Alat Bantu dan Tools

1. *Flowchart*

Flowchart adalah representasi visual dalam bentuk gambar atau diagram yang menunjukkan urutan dan hubungan antara proses beserta instruksinya. Dalam flowchart, setiap simbol melambangkan suatu proses tertentu, dan hubungan antara proses tersebut digambarkan dengan menggunakan garis penghubung [12]. Simbol-simbol yang terdapat pada *flowchart* terdapat pada Tabel 2.1 berikut ini:

Tabel 2. 1 Simbol-simbol Flowchart

No.	Simbol	Nama	Keterangan
1.		*Flow*	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.
2.		*Terminator*	Simbol untuk permulaan (*start*) atau akhir (*stop*) dari suatu kegiatan.
3.		*On Page Connector*	Simbol untuk keluar – masuk atau penyambungan proses dalam lembar / halaman yang sama.
4.		*Off Page Connector*	Simbol untuk keluar– masuk atau penyambungan proses dalam lembar / halaman yang berbeda.
5.		*Process*	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer
6.		*Manual Operation*	Simbol yang menunjukkan pengolahan yang tidak lakukan oleh computer.
7.		*Decision*	Simbol pemilihan proses berdasarkan kondisi yang ada,
8.		*Input - Output*	Simbol yang menyatakan proses input dan output tanpa terhubung dengan jenis peralatannya.
9.		*Manual Input*	Simbol memasukkan data secara manual *on-line keyboard.*
10.		*Document*	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

2. UML *(Unified Modeling Language)*

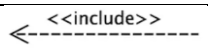
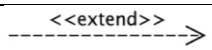
UML *(Unified Modeling Language)* adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artifact dari sistem perangkat lunak. Artifact ini mencakup model, deskripsi, atau perangkat lunak yang digunakan atau dihasilkan dalam proses pembuatan perangkat lunak, termasuk pemodelan bisnis dan sistem non-perangkat lunak lainnya[13]. UML memiliki berbagai diagram yang digunakan untuk memodelkan berbagai aspek dari sistem perangkat lunak. Berikut adalah penjelasan mengenai beberapa diagram yang umum digunakan dalam UML:

a. *Use Case Diagram*

Diagram *Use Case* menggambarkan interaksi antara aktor-aktor eksternal dengan sistem, mewujudkan kegunaan yang diharapkan dari sistem tersebut. Setiap *use case* dalam diagram

ini mencerminkan sebuah tindakan atau fungsi yang dilakukan oleh sistem, yang biasanya terjadi antara aktor dan sistem.
Berikut tabel 2.2 adalah simbol-simbol yang ada pada *use case* diagram :

Tabel 2. 2 Simbol-Simbol Use Case Diagram

No.	Simbol	Nama	Keterangan
1.		*Actor*	Menguraikan peran yang pengguna mainkan saat berinteraksi dengan *use case*
2.		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
3.		*Associtation*	Apayang menghubungkan antara objek satu dengan objek lainnya.
4.		*Include*	Menspesifikasikan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
5.		*Extend*	Menspesifikasikan bahwa *use case* target memperluas prilaku dari *use case*
6.		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

2.1.8 Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek adalah pendekatan baru dalam penyelesaian masalah dengan menggunakan komputer, di mana setiap objek mewakili entitas tunggal yang memiliki gabungan data dan fungsi tertentu. Objek dapat berupa orang, tempat, benda, peristiwa, atau konsep yang signifikan dalam dunia nyata, dan penting dalam konteks aplikasi tertentu, seperti benda fisik seperti mesin, bangunan, komputer, mobil, dan lainnya[14].

a. *Class*

Kelas atau *class* adalah representasi dari sekumpulan objek yang memiliki atribut yang serupa. Kelas mirip dengan tipe data dalam pemrograman non-objek, namun lebih komprehensif karena mencakup struktur dan karakteristik. Kelas baru bisa dibentuk lebih spesifik dari kelas yang ada pada umumnya. Kelas adalah inti dari pemrograman berorientasi objek dan memainkan peran penting dalam pengembangan perangkat lunak modern.

b. Objek

Objek adalah teknik yang efektif untuk menyelesaikan masalah yang sering muncul dalam pengembangan perangkat lunak. Teknik ini membantu menemukan cara yang tepat untuk membangun sistem dan menjadi metode yang paling umum digunakan oleh para pengembang perangkat lunak. Orientasi objek adalah teknik pemodelan sistem nyata yang berbasis objek. Objek merupakan entitas yang memiliki atribut, karakter, dan kondisi tertentu.

c. Pewarisan

Pewarisan atau *inheritance* adalah konsep di mana suatu entitas atau objek dapat memiliki entitas atau objek turunan. Dengan konsep *inheritance*, sebuah kelas (class) dapat memiliki

kelas turunan. Pewarisan dalam Pemrograman Berorientasi Objek (PBO) juga dikenal dengan istilah *parent class* atau *base class* dan *subclass* atau *child class*. *Subclass* atau *child class* mewarisi semua data dan perilaku dari *parent class* atau *base class*, sehingga *subclass* atau *child class* dapat dianggap sebagai perluasan dari *parent class* atau *base class*. Konsep *inheritance* ini memungkinkan adanya hierarki kelas yang memudahkan pengorganisasian kode dan memungkinkan penggunaan kembali kode yang lebih efisien.

d. *Method*

Method adalah sekumpulan program yang memiliki nama khusus. *Method* merupakan sarana bagi programmer untuk memecah program menjadi bagian-bagian kecil yang lebih terstruktur, sehingga program menjadi lebih kompleks namun terorganisir dengan baik. Dengan menggunakan *method*, kode dapat digunakan kembali secara berulang-ulang, meningkatkan efisiensi dan kemudahan dalam pengelolaan program. *Method* juga membantu dalam memfasilitasi pemeliharaan dan pengembangan program di masa mendatang.

2.1.9 Website

Website merupakan sebuah halaman yang banyak dikunjungi oleh orang-orang, karena banyak menampilkan berbagai informasi. Halaman web adalah sebuah dokumen yang ditulis dalam format HTML *(Hyper Text Markup Language)* dan dapat diakses melaui HTTP yaitu sebuah protokol yang mengirimkan informasi dari server situs web ke browser sehingga dapat ditampilkan kepada pengguna (*user*)[15].

2.1.10 Framework

Framework adalah suatu perangkat lunak atau aplikasi yang berfungsi sebagai kerangka kerja yang memudahkan para pengembang dalam proses pengembangan aplikasi website. Dapat diibaratkan sebagai landasan atau struktur dasar yang menyediakan berbagai fitur, alat, dan pola desain yang siap digunakan untuk membangun aplikasi dengan efisiensi dan konsistensi yang lebih baik[16].

a. Model-View-Controller (MVC)

Model-View-Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh Trygve Reenskaug, pencipta Smalltalk, yang bertujuan untuk memisahkan komponen-komponen utama dalam pembangunan aplikasi. Konsep ini mengizinkan pengenalan data bersama dengan pemrosesan (model), isolasi proses manipulasi (controller), dan representasi tampilan (view) pada antarmuka pengguna. MVC menjadi salah satu konsep yang populer dalam pengembangan aplikasi web karena memisahkan pengembangan aplikasi berdasarkan komponen-komponen inti seperti manipulasi data, dan antarmuka pengguna[17].

b. Framework Laravel

Framework Laravel adalah sebuah *framework* bahasa pemrograman PHP yang kaya fitur dan memberikan bantuan yang signifikan kepada para pengembang dalam pembangunan aplikasi web. Framework ini dirancang dengan tujuan meningkatkan kualitas aplikasi, mengurangi biaya pengembangan, menyederhanakan proses pemeliharaan, dan meningkatkan produktivitas kerja melalui penggunaan kode program yang terstruktur dan bersih[18].

2.1.11 *Hypertext Preprocessor* (PHP)

PHP merupakan singkatan dari *Hypertext Preprocessor*, merupakan bahasa pemrograman *server-side* yang sangat umum digunakan saat ini, terutama dalam pembuatan *website* yang dinamis. Dalam proses pembuatan *website*, bahasa pemrograman PHP memiliki peran penting, terutama dalam mengelola data yang diterima dari pengunjung situs[19].

2.1.12 Basis Data (*Database)*

Database adalah sebuah kumpulan koleksi data yang berhubungan secara logical, dan sebuah deskripsi dari data tersebut, didesain untuk dapat menemukan sebuah informasi pada suatu instansi. Menurut pengertian lain, database adalah kumpulan data yang disimpan secara sistematis di dalam computer yang dapat diolah atau manipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi[20].

a. DBMS *(Database Management System)*

Database Management System atau DBMS berfungsi untuk mengelola semua aspek yang terkait dengan basis data, termasuk penyimpanan, pengaturan, dan akses data. Ini merupakan platform yang digunakan untuk mengatur dan mengelola informasi dalam suatu sistem basis data. DBMS menyediakan berbagai macam fasilitas,yaitu:

1. DDL (*Data Definition Language)*

   DDL *(Data Definition Language)* adalah serangkaian instruksi yang sering digunakan oleh administrator basis data untuk menetapkan struktur dan sub-struktur dari basis data. Instruksi-instruksi yang termasuk didalamnya mencakup :

   a) *CREATE:* digunakan untuk menciptakan entitas baru dalam basis data, seperti pembuatan database atau tabel baru.
   b) *ALTER :* digunakan untuk memodifikasi struktur dari entitas yang sudah ada, seperti tabel yang sudah ada dalam basis data.
   c) *DROP :* digunakan untuk menghapus entitas dari basis data, baik itu database secara keseluruhan, atau tabel individual di dalamnya.

2. DML (*Data Manipulation Language)*

   DML adalah singkatan dari *Data Manipulation Language*. Ini merujuk pada kumpulan perintah yang memungkinkan pengguna untuk mengakses dan memanipulasi data dalam sebuah database yang sudah didefinisikan menggunakan DDL (*Data Definition Language*). Dalam DML, pengguna dapat melakukan berbagai operasi seperti penambahan, penghapusan, pembaruan, atau pengambilan data dari database. Dengan menggunakan perintah- perintah dalam DML, pengguna dapat melakukan manipulasi data sesuai dengan model data yang telah ditetapkan sebelumnya. Perintah yang termasuk dalam DML yaitu sebagai berikut :

   a) *INSERT :* Berfungsi untuk menambahkan atau memasukkan data ke dalam tabel.
   b) *SELECT :* Berperan dalam mengambil atau menampilkan data dari satu atau beberapa tabel..
   c) *UPDATE :* Digunakan untuk mengubah data yang sudah ada menjadi data yang baru.
   d) *DELETE:* Berfungsi untuk menghapus data dari tabel.

b. SQL (*Structured Query Language*)

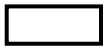
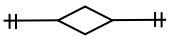
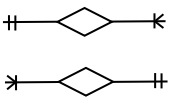
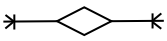
SQL (*Structured Query Language*) adalah bahasa relasional yang terdiri dari pernyataan-pernyataan yang digunakan untuk menambahkan, mengubah, menghapus, memilih, dan melindungi data. SQL bukanlah database itu sendiri, melainkan sebuah bahasa yang digunakan untuk berinteraksi dengan database. Lebih tepatnya, SQL adalah bahasa yang digunakan untuk mengajukan pertanyaan atau melakukan operasi terhadap database oleh pengguna. MySQL adalah jenis RDBMS (*Relational Database Management System*). Oleh karena itu, istilah seperti tabel, baris, dan kolom digunakan dalam MySQL karena MySQL mengadopsi model relasional untuk menyimpan dan mengelola data. Dalam konteks ini, "tabel" merujuk kepada entitas yang menyimpan data dalam format berbaris dan berkolom, di mana setiap baris

mewakili satu entitas atau rekord, dan setiap kolom mewakili atribut atau field dari entitas tersebut.

c. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) adalah sebuah diagram yang dibuat dengan menggunakan simbol-simbol untuk menggambarkan hubungan antara entitas beserta relasinya yang saling terhubung dalam sebuah sistem[21]. Berikut tabel 2.3 adalah simbol-simbol dalam ERD :

Tabel 2. 3 Simbol - Simbol ERD

No.	Simbol	Nama	Keterangan
1.		*Entitas*	Entitas merupakan data inti yang akan disimpan.
2.		*Atribut*	*Field* atau kolom data yang butuh disimpan dalam suatu entitas.
3.		*Relasi*	Relasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja
4.		*Asosiasi*	Penghubung antara relasi dan entitas, kedua ujungnya memiliki *multiplicity* jumlah pemakaian.
5.		*Kardinalitas One to one*	Satu anggota entitas dapat berelasi dengan entitas lain.
6.		*Kardinalitas One to many*	Satu anggota entitas dapat berelasi dengan beberapa anggota entitas lain.
8.		*Kardinalitas Many to many*	Beberapa entitas dapat berelasi dengan beberapa anggota entitas lain.