

BAB II

LANDASAN TEORI

2.1 Landasan Teori

Landasan teori berisikan rangkuman teori-teori yang relevan dengan masalah yang dibahas, dan menjadi landasan penting dalam penyusunan laporan untuk pemahaman yang lebih baik.

2.1.1 Sistem Informasi

Sistem merupakan prosedur yang saling bekerjasama untuk menyelesaikan sasaran. Sistem juga merupakan kumpulan data yang terkait dan terorganisir dan berinteraksi agar mencapai tujuan. [1]

Sistem Informasi dapat diartikan sebagai suatu sistem didalam sebuah organisasi yang mempertemukan pengolahan transaksi yang mendukung operasi organisasi dengan kegiatan dari suatu organisasi agar dapat menyediakan laporan yang dibutuhkan oleh pihak luar.

2.1.2 Konseling

Konseling seringkali diartikan sebagai suatu proses membantu klien untuk menuntaskan masalah yang dimilikinya. Dapat diartikan secara sederhana konseling merupakan hubungan antara konselor dengan seorang klien dengan tujuan untuk membantu klien untuk menemukan solusi serta mengkaji masalahnya untuk mengambil sebuah keputusan berdasarkan informasi yang diberikan oleh konselor profesional.[2]

Tujuan konseling secara umum adalah [3]:

- 1.Menyediakan Fasilitas untuk Perubahan Perilaku: Konselor berperan sebagai fasilitator untuk membantu individu melakukan perubahan perilaku, karena beberapa orang memerlukan dukungan dari orang lain untuk melakukan perubahan tersebut.
- 2.Meningkatkan Keterampilan untuk Menghadapi Tekanan: Konselor berfungsi sebagai pelatih yang membantu konseli meningkatkan keterampilan dalam menghadapi situasi yang mereka anggap sebagai stres atau tekanan.
- 3.Meningkatkan Kemampuan dalam Pengambilan Keputusan: Membuat keputusan secara mandiri memerlukan keterampilan tertentu. Beberapa individu membutuhkan pandangan dari orang lain yang netral untuk memastikan keputusan yang diambilnya benar. Konselor adalah salah satu pihak yang dapat memberikan pandangan netral tersebut.

4. Meningkatkan Hubungan antar Individu: Konflik interpersonal sering kali menghambat perkembangan individu. Oleh karena itu, diperlukan keterampilan untuk menyelesaikan konflik ini. Konselor berperan sebagai pelatih untuk membantu konseli meningkatkan keterampilan dalam menyelesaikan konflik antar pribadi.

5. Menyediakan Fasilitas untuk Pengembangan Kemampuan Konseli: Lingkungan yang kurang mendukung dapat menghambat pengembangan bakat dan potensi individu. Konselor dapat berperan sebagai mediator untuk membantu menciptakan perubahan lingkungan yang lebih kondusif bagi perkembangan konseli.

2.1.3 Konseling Individu

Konseling individu adalah proses pembelajaran melalui hubungan pribadi khusus dalam wawancara terapeutik antara seorang konselor dan seorang konseli. Pendekatan ini biasanya digunakan ketika konseli menghadapi masalah pribadi yang tidak bisa diselesaikan sendiri, sehingga ia mencari bantuan dari konselor profesional. Konseling ini ditujukan kepada individu normal yang mengalami kesulitan dalam bidang pendidikan, pekerjaan, dan sosial, di mana mereka tidak bisa membuat pilihan dan keputusan sendiri. Oleh karena itu, konseling hanya diberikan kepada individu yang sudah sadar akan kehidupan pribadinya. Dalam konseling individual, terdapat hubungan yang dinamis dan istimewa, karena dalam interaksi ini, konseli merasa diterima dan dipahami oleh konselor. [3]

2.1.4 Psikologi

Psikologi adalah ilmu yang mempelajari perilaku dan pikiran, baik pada manusia maupun makhluk hidup lainnya, serta meneliti proses-proses mental yang mendasarinya. Dalam perkembangannya, psikologi tidak hanya menjadi ilmu tentang pikiran, tetapi juga berkembang lebih luas menjadi ilmu tentang perilaku serta proses-proses internal yang mendasarinya, termasuk pikiran. [4]

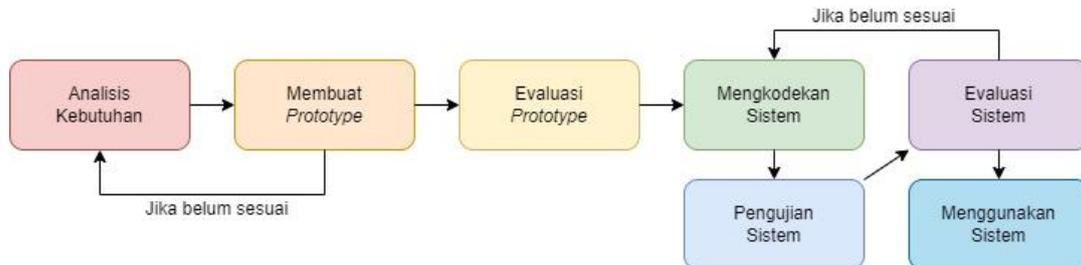
2.1.5 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak (RPL) merupakan bidang ilmu yang mencakup seluruh tahapan produksi perangkat lunak, mulai dari penentuan spesifikasi awal sistem hingga pemeliharaan sistem setelah digunakan. Tujuan utama dari pembuatan perangkat lunak adalah untuk memastikan bahwa proyek tersebut selesai tepat waktu, sesuai dengan anggaran yang tersedia, meningkatkan kinerja sistem, dan menjalankan prosedur sistem dengan benar[5]. Berikut ini adalah metode dan tools yang digunakan:

A. Metode Pengembangan Sistem

Metode yang digunakan dalam penelitian ini adalah metode *prototype*. *Prototype* model adalah pendekatan dalam rekayasa perangkat lunak yang secara langsung menunjukkan cara

kerja perangkat lunak atau komponennya dalam lingkungan aslinya sebelum tahap konstruksi sebenarnya dimulai. Pendekatan ini memungkinkan pengguna untuk memiliki pemahaman awal tentang perangkat lunak yang akan dibuat dan melakukan uji coba sebelum perangkat lunak tersebut diluncurkan. Pemodelan *prototype* dapat dilihat pada gambar 2.1 dibawah. [6]



Gambar 2. 1 Pemodelan *prototype*

Metode *prototype* memiliki 7 tahapan, yaitu [6]:

1. Analisis Kebutuhan : Identifikasi dan definisi semua kebutuhan perangkat lunak yang akan dibuat.
2. Membuat *prototype* : Membuat rancangan sementara sistem dengan fokus pada alur perangkat lunak serta metode dan algoritma yang digunakan.
3. Evaluasi *prototype* : Tinjauan awal terhadap sistem untuk memastikan apakah *prototype* sudah mencerminkan produk akhir dan sesuai dengan harapan klien.
4. Pengkodean sistem : Konstruksi sistem menggunakan bahasa pemrograman yang sesuai.
5. Pengujian sistem : Menguji kinerja perangkat lunak berdasarkan fungsinya menggunakan metode seperti *whitebox testing*, *blackbox testing*, atau metode lainnya.
6. Evaluasi sistem : Peninjauan oleh pengguna untuk memastikan bahwa perangkat lunak sudah sesuai dengan harapan. Jika tidak, ulangi tahapan mengkodekan sistem dan pengujian sistem.
7. Implementasi sistem : Perangkat lunak yang telah diuji dan disetujui siap untuk digunakan oleh pengguna.

B. Tools/ Alat Bantu Penelitian

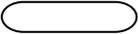
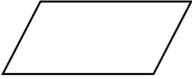
1. Flowchart

Flowchart adalah metode visual yang digunakan untuk menggambarkan beberapa aspek dari sistem informasi secara jelas, singkat, dan logis. *Flowchart* mencatat bagaimana proses bisnis dilakukan dan pergerakan dokumen dalam suatu organisasi.[7]

Berikut adalah jenis-jenis *flowchart* :

- a. *Flowchart* Sistem: gambaran alur kerja keseluruhan dalam sistem..
- b. *Flowchart* Dokumen: melacak alur formulir dan laporan dalam sistem.
- c. *Flowchart* Skematik: Mirip dengan *flowchart* sistem, tetapi menggunakan simbol tambahan untuk peralatan atau komputer.
- d. *Flowchart* Program: Deskripsi rinci dari langkah-langkah dalam sistem.
- e. *Flowchart* Proses: Teknik untuk menggambarkan langkah-langkah dalam prosedur industri untuk studi dan pengembangan.

Tabel 2. 1 Tabel Simbol *Flowchart*

No	Simbol	Keterangan
1	 Anak Panah	Menjelaskan secara terperinci bagaimana serangkaian langkah atau proses dilakukan dari awal hingga akhir, dengan tujuan untuk memahami bagaimana elemen-elemen berinteraksi dan berkontribusi terhadap hasil akhir atau pencapaian tujuan tertentu.
2	 <i>Terminal Points</i>	Simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu proses.
3	 Proses	Simbol yang menunjukkan pengolahan/proses yang dilakukan oleh komputer.
4	 <i>Manual Operation</i>	Simbol yang menunjukkan pengolahan data yang dilakukan secara manual dan tidak melibatkan komputer.
6	 Keputusan	Simbol untuk menggambarkan proses keputusan berdasarkan kondisi yang ada.
7		Simbol yang menunjukkan proses <i>input</i> dan <i>output</i> data yang diproses tanpa tergantung dari jenis peralatannya.

	<i>Input-Output</i>	
6	 Dokumen	Simbol yang menunjukkan <i>input</i> ataupun <i>output</i> yang berasal dari dokumen dalam bentuk kertas.

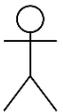
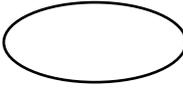
2. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu alat pemodelan yang sangat andal dalam pengembangan sistem berorientasi objek. UML merupakan hasil pengembangan dari beberapa bahasa pemodelan, termasuk *Booch*, *Objects Modeling Technique* (OMT), dan *Object Oriented Software Engineering* (OOSE). Pendekatan ini mengintegrasikan proses analisis dan desain dalam iterasi, dengan langkah-langkah seperti identifikasi kelas dan objek, penentuan semantik hubungan antara objek dan kelas, serta rincian antarmuka dan implementasi[8]. Pemodelan UML meliputi *use case* diagram.

UseCase adalah sebuah diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dengan detail interaksi antara para aktor dan sistem yang sedang dikembangkan. Setiap *usecase* merepresentasikan deskripsi lengkap tentang fungsi yang diinginkan dari sistem, serta memvisualisasikan bagaimana interaksi tersebut terjadi antara aktor dengan sistem yang dibangun. Aktor dalam diagram *usecase* menggambarkan entitas manusia atau sistem yang melakukan tugas yang terlibat dalam sistem yang sedang dibahas, sehingga membentuk sebuah gambaran lengkap tentang keterlibatan dan ketergantungan antara entitas dalam sistem.[9]

Dari pengertian tersebut, dapat disimpulkan bahwa *usecase* adalah cara untuk merepresentasikan fungsi-fungsi yang diinginkan dari suatu sistem secara visual. *Usecase* juga menggambarkan bagaimana interaksi antara aktor dengan sistem tersebut.

Tabel 2. 2 Tabel Simbol *Usecase*

No	Gambar	Nama Gambar	Keterangan
1.		<i>Actor</i>	Mengidentifikasi berbagai peran yang dimainkan oleh pengguna saat berinteraksi dengan kasus penggunaan <i>usecase</i>
2.		<i>Usecase</i>	Penjelasan tentang serangkaian langkah yang dilakukan oleh sistem untuk menghasilkan hasil yang dapat diukur bagi seorang aktor
3.		<i>System</i>	Menentukan paket yang menampilkan sistem dengan cakupan yang terbatas.
4.		<i>Dependency</i>	Hubungan yang merujuk pada dinamika perubahan yang terjadi pada suatu elemen yang memiliki keberadaan independen, yang akan mengakibatkan perubahan pada elemen-elemen yang bergantung padanya, yang dalam konteks ini keberadaan independen yang sama.
5.		<i>Generalization</i>	Hubungan di mana objek yang berada di tingkat hierarki lebih rendah (descendant) memiliki karakteristik dan struktur data yang sama dengan objek yang berada di atasnya dalam hierarki (ancestor).
6.		<i>Clude</i>	Spesifikasi bahwa sumber <i>usecase</i> secara terperinci.
7.		<i>Tend</i>	Menentukan bahwa <i>usecase</i> yang dituju memperluas fungsi dari <i>usecase</i> asal pada titik tertentu.

8.	_____	<i>Assosiation</i>	Membuat koneksi antara satu objek dengan objek lainnya.
----	-------	--------------------	---

2.1.6 Pemrograman Berorientasi Objek

Menurut Andi dalam bukunya tentang analisis dan desain sistem informasi berbasis metodologi berorientasi objek, Pemrograman Berorientasi Objek merupakan pendekatan baru dalam pemikiran dan logika dalam mengatasi masalah dengan bantuan komputer. Dalam pendekatan ini, setiap objek merupakan entitas tunggal yang memiliki kombinasi struktur data dan fungsi khusus. Objek dapat merujuk kepada berbagai hal seperti manusia, tempat, benda, peristiwa, atau kegiatan seperti pembayaran uang pendidikan, registrasi biodata siswa, membaca buku, dan sebagainya. [10]

2.1.7 Basis Data

Basis data adalah kumpulan data terkait yang disimpan bersama dengan pengurangan redundansi yang terkontrol, dirancang untuk melayani aplikasi secara efisien. Data tersusun sedemikian rupa sehingga tidak bergantung pada program tertentu. Akses untuk menambah, mengubah, dan mengambil data diatur secara terkontrol. Ini juga merupakan sistem penyimpanan file elektronik yang terstruktur.[11]

MySQL adalah sebuah perangkat lunak sistem manajemen basis data yang menggunakan bahasa SQL (*Structured Query Language*), atau DBMS. Secara keseluruhan, MySQL dapat diklasifikasikan sebagai jenis database server yang termasuk dalam kategori RDBMS (*Relational Database Management System*).[12]

DBMS adalah perangkat lunak yang digunakan untuk mengatur basis data dengan tanggung jawab dalam penyimpanan, pengambilan, pembaruan, dan penghapusan data di dalamnya. DBMS juga menyediakan antarmuka serta perangkat untuk mengatur struktur basis data, memanipulasi data, dan menjaga keamanan serta integritas data.[11]

DBMS terdiri dari tiga komponen utama, yaitu Bahasa Definisi Data (DDL), Bahasa Manipulasi Data dan Fasilitas *Query* (DML/SQL), serta perangkat lunak untuk mengatur akses ke basis data.

1.) *Data Definition Language* (DDL)

DDL merupakan segmen dari bahasa *query database* yang bertugas untuk mengatur struktur dan skema basis data. DDL digunakan untuk pembuatan, modifikasi, dan penghapusan objek dalam basis data seperti tabel, *indeks*, tampilan, dan konstrain. Melalui perintah-perintah

DDL, pengguna dapat menentukan entitas dan relasi antara entitas dalam basis data. Berikut merupakan perintah umum DDL:

- a. *CREATE*: perintah *CREATE* digunakan untuk menciptakan objek *database* baru seperti tabel, *indeks*, atau tampilan.
- b. *ALTER*: perintah *ALTER* digunakan untuk mengubah struktur objek *database* yang telah ada, seperti menambahkan kolom baru ke tabel.
- c. *DROP*: perintah *DROP* digunakan untuk menghapus objek *database* seperti tabel atau indeks dari dalam database.

2.) *Data Manipulation Language (DML)*

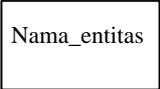
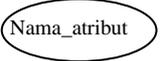
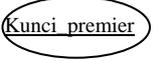
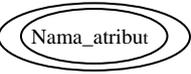
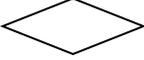
DML merupakan bagian dari bahasa *query database* yang bertugas untuk mengelola data dalam tabel. DML berperan dalam menyisipkan (*insert*), memperbarui (*update*), menghapus (*delete*), dan mengambil (*select*) data dari tabel. DML memberikan kemampuan kepada pengguna untuk memanipulasi data dengan mengubah nilai-nilai dalam tabel, mengambil subset data yang diinginkan, dan menjalankan operasi terkait data lainnya yang telah dibuat. Berikut merupakan beberapa perintah umum DML:

- a. *INSERT* : perintah *INSERT* digunakan untuk menambahkan data baru ke dalam tabel.
- b. *UPDATE* : perintah *UPDATE* digunakan untuk mengubah data yang telah ada dalam tabel.
- c. *DELETE* : perintah *DELETE* digunakan untuk menghapus data dari tabel.
- d. *SELECT* : perintah *SELECT* digunakan untuk mengambil data tertentu dari tabel berdasarkan kriteria yang ditentukan.

2.1.8 *Entity Relationship Diagram (ERD)*

ERD adalah suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubpeungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan entity lainnya[7].

Tabel 2. 3 Tabel Simbol ERD

No.	Nama	Simbol	Keterangan Fungsi
1.	Entitas		Data inti yang akan disimpan, dengan menggunakan kata benda untuk menamai entitas yang belum diubah menjadi nama tabel.
2.	Atribut		Kolom yang akan disimpan dalam entitas.
3.	Atribut Kunci Primer		Kolom yang akan disimpan dalam entitas dan digunakan sebagai <i>primary key</i> .
4.	Atribut <i>multivalue</i>		Kolom yang akan disimpan dalam entitas yang memiliki nilai dua atau lebih.
3.	Relasi		Simbol untuk menghubungkan antar entitas, yang biasanya diawali dengan kata kerja.
4.	Asosiasi		Merupakan simbol enghubung antara relasi, dan entitas.