

BAB II

LANDASAN TEORI

2.1 Landasan Teori

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

2.1.1 Sistem Informasi

sistem informasi adalah kumpulan dari beberapa sistem di dalam suatu organisasi yang mengumpulkan, memproses, menyimpan dan mendistribusikan informasi sebagai pendukung pengambilan keputusan dan pengendalian dalam organisasi.[1]

2.1.2 Sistem Informasi Akademik

Sistem informasi akademik dapat didefinisikan sebagai suatu sistem yang dirancang untuk memenuhi kebutuhan user yang menginginkan layanan pendidikan komputer untuk meningkatkan efisiensi, kualitas layanan, daya saing dan kualitas sumber daya manusia yang dihasilkan. Sistem informasi akademik sangat membantu dalam pengelolaan data nilai peserta didik, mata pelajaran, data staf pengajar, dan laporan hasil belajar yang sifatnya masih manual untuk dikerjakan dengan bantuan software agar lebih efektif dan efisien sehingga dapat pula menekan biaya operasional.[2]

2.1.3 Flowchart

Flowchart adalah representasi grafis dari langkah-langkah atau proses dalam suatu sistem atau algoritma. Flowchart menggunakan simbol-simbol standar yang dihubungkan dengan garis untuk menggambarkan urutan langkah dan hubungan antar langkah dalam suatu proses. Flowchart membantu dalam memvisualisasikan dan memahami alur kerja, serta dalam mendokumentasikan, menganalisis, dan mengembangkan sistem atau proses.

Flowchart dapat dikategorikan ke dalam beberapa jenis berdasarkan tujuan dan penggunaannya, antara lain:



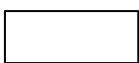


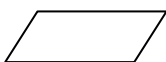
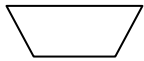
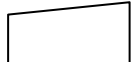
- a. Flowchart Sistem
Menggambarkan alur kerja secara keseluruhan dari suatu sistem.
- b. Flowchart Program
Digunakan untuk mendeskripsikan logika program atau algoritma.
- c. Flowchart Proses
Menunjukkan langkah-langkah dalam suatu proses bisnis atau produksi.

d. Flowchart Dokumen

Menggambarkan alur dari dokumen atau informasi dalam suatu sistem.

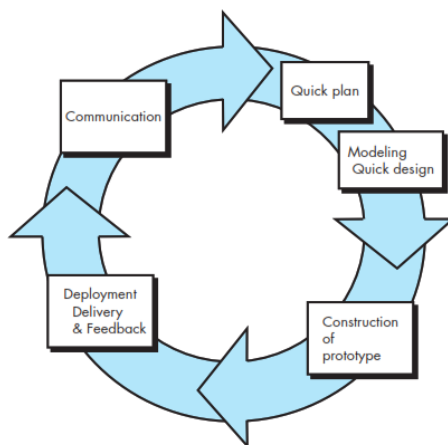
Simbol – simbol flowchart dapat dilihat pada Tabel 2.1

Tabel 2. 1 Simbol Flowchart

No	Simbol	Nama	Keterangan
1		Flow Direction	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini juga disebut connecting line
2		Terminator	Simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan
3		Processing	simbol yang menunjukkan pengolahan yang dilakukan oleh komputer
4		Decision	Simbol pemilihan proses berdasarkan kondisi yang ada
5		Document	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output yang dicetak ke kertas
6		Input - Output	Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
7		Manual Operation	Simbol yang menunjukan pengolahan yang tidak dapat dilakukan oleh komputer
8		Manual Input	Simbol yang digunakan untuk memasukan data secara manual dengan menggunakan online keyboard

2.1.4 Prototype

Prototype adalah salah satu metode dalam pengembangan perangkat lunak yang berfokus pada pembuatan model awal (*prototipe*) yang kemudian diperbaiki berdasarkan masukan dari pengguna. Pendekatan ini digunakan untuk mengatasi ketidakpastian dan memastikan kebutuhan pengguna terpenuhi dengan efektif. Metode *Prototype* memungkinkan pengembang dan pengguna bekerja sama dalam memvisualisasikan dan menguji fitur-fitur produk sebelum diimplementasikan sepenuhnya. Menurut Pressman tahapan-tahapan yang digunakan dalam metode *prototype* dapat dilihat pada gambar 2.1



Gambar 2. 1 Metode Prototype

Berikut penjelasan tentang tahapan-tahapan yang digunakan dalam metode *prototype* menurut Pressman: [3]

1. Komunikasi (*Communication*)

Tahap awal dari model *prototype* yang bertujuan untuk mengidentifikasi masalah-masalah yang ada serta mengumpulkan informasi lain yang dibutuhkan untuk membangun sistem.

2. Perencanaan (*Quick Plan*)

Tahap ini melibatkan penentuan sumber daya, spesifikasi pengembangan berdasarkan kebutuhan sistem, dan penetapan tujuan berdasarkan hasil komunikasi yang telah dilakukan, sehingga pengembangan dapat sesuai dengan harapan.

3. Pemodelan (*Modeling Quick Design*)

Pada tahapan merupakan tahapan di mana sistem yang akan dikembangkan direpresentasikan atau digambarkan melalui proses perancangan menggunakan Unified Modeling Language (UML). *Prototype* yang dibangun dengan desain sementara dievaluasi terhadap kebutuhan pengguna untuk memastikan kesesuaian. Jika *prototype* dianggap sesuai, langkah selanjutnya adalah mengembangkan aplikasi dengan

menerjemahkan rancangan sistem ke dalam bahasa pemrograman menggunakan Framework Laravel yang terintegrasi dengan basis data MySQL.

4. Konstruksi (*Construction of Prototype*)

Tahap ini melibatkan pembangunan prototype dan pengujian sistem yang dibuat. Instalasi dan penyediaan dukungan pengguna juga dilakukan untuk memastikan sistem berjalan dengan baik.

5. Penyerahan (*Deployment Delivery & Feedback*)

Pada tahapan ini diperlukan untuk mendapatkan umpan balik dari pengguna sebagai hasil evaluasi dari tahap sebelumnya dan untuk mengimplementasikan sistem yang telah dikembangkan.

2.1.5 Basis Data

Pangkalan data atau basis data (database) adalah kumpulan data yang terstruktur, biasanya disimpan dan diakses secara elektronik melalui sistem komputer. Seiring dengan meningkatnya kompleksitas pangkalan data, pengembangannya dilakukan dengan menggunakan teknik perancangan dan pemodelan formal. Perangkat lunak yang digunakan untuk mengelola basis data dikenal sebagai sistem manajemen basis data (database management system) atau disingkat DBMS. DBMS adalah perangkat lunak yang dirancang untuk mengatur dan mengelola kumpulan data dalam jumlah besar serta mempermudah manipulasi data. DBMS berfungsi sebagai antarmuka antara pengguna basis data (baik pengguna langsung maupun melalui aplikasi) dengan data yang disimpan.

Salah satu contoh DBMS yaitu MySQL (*Structured Query Language*). MySQL adalah DBMS *opensource* yang tersedia dalam dua jenis lisensi yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik dengan penggunaan terbatas). Sebagai database server gratis dengan lisensi GNU *General Public License* (GPL), MySQL dapat digunakan untuk keperluan pribadi atau komersial tanpa perlu membayar lisensi. Secara umum MySQL mempunyai 2 bahasa yaitu:

1. DDL (*Data Definition Language*)

DDL adalah kumpulan perintah SQL yang digunakan untuk membuat, mengubah, dan menghapus struktur serta definisi metadata dari objek-objek dalam basis data. DDL mencakup perintah-perintah seperti

- a. CREATE untuk membuat tabel atau indeks baru
- b. ALTER untuk memodifikasi struktur tabel yang sudah ada

c. DROP untuk menghapus tabel atau objek database lainnya

Dengan menggunakan DDL, administrator basis data dapat mendefinisikan skema database dan memastikan bahwa semua objek database terstruktur dengan benar sesuai kebutuhan aplikasi.

2. DML (*Data Manipulation Language*)

DML adalah sekumpulan perintah yang memungkinkan pengguna untuk mengakses dan memanipulasi data dalam sistem basis data relasional. DML mencakup perintah-perintah seperti

- a. SELECT untuk mengambil data dari tabel
- b. INSERT untuk menambah data baru
- c. UPDATE untuk mengubah data yang sudah ada
- d. DELETE untuk menghapus data.


Dengan menggunakan DML pengguna dapat melakukan berbagai operasi pada data yang tersimpan dalam database, seperti mengelola informasi, melakukan analisis, dan mengelola transaksi dengan efisien.

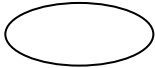
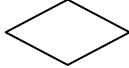


2.1.6 ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) adalah sebuah metode untuk memodelkan data atau sistem dalam sebuah database. ERD berfungsi untuk menggambarkan struktur dan hubungan antar data yang kompleks. Keberadaan ERD sangat penting bagi perusahaan dalam mengelola data mereka secara efektif. Dengan menggunakan ERD, perusahaan dapat memvisualisasikan bagaimana data diorganisir dan dihubungkan satu sama lain, sehingga mempermudah dalam perancangan, implementasi, dan pemeliharaan sistem basis data. ERD membantu dalam memastikan bahwa semua entitas dan hubungan di dalam sistem dapat dipahami dengan jelas, mengurangi kesalahan dalam proses pengembangan dan meningkatkan efisiensi dalam pengelolaan data. [4]

ERD berisikan simbol-simbol yang sesuai dengan hubungan antara masing-masing entitas. Berikut simbol – simbol ERD pada Tabel 2.2

Tabel 2. 2 Entity Relationship Diagram (ERD)

No	Simbol	Nama	Keterangan
1		Entitas	Kumpulan dari objek yang dapat diidentifikasi secara unik

No	Simbol	Nama	Keterangan
2		Atribut	Karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas
3		Relationship	Hubungan yang terjadi antara salah satu lebih entitas. Jenis hubungan antara lain: one to one, one to many, dan many to many
4		Link	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya
5		Associative Entity	Entity yang digunakan pada many-to-many relationship (banyak antar banyak)

2.1.7 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk merancang, mendokumentasikan, dan memvisualisasikan sistem perangkat lunak. UML menyediakan serangkaian notasi grafis yang membantu dalam menggambarkan berbagai aspek dari sistem, seperti struktur, perilaku, dan interaksi.

Terdapat beberapa diagram UML yang digunakan dalam pengembangan sebuah sistem salah satunya yaitu *Use Case Diagram*

a. *Use Case Diagram*

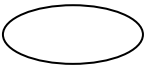



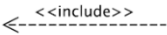
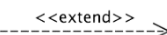
Use-case diagram adalah diagram yang menggambarkan fungsi-fungsi yang diharapkan dari sebuah sistem yang sedang dikembangkan. Fokus utama pada *use-case* diagram adalah "apa" yang dilakukan oleh sistem, bukan "bagaimana" caranya. Sebuah *use-case* merepresentasikan interaksi antara aktor dengan sistem.

Use case mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara garis besar, *use case* digunakan untuk mengidentifikasi fungsi-fungsi yang ada dalam sistem informasi dan menentukan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Use-case diagram sangat bermanfaat dalam menyusun kebutuhan sistem, mengomunikasikan kebutuhan tersebut dengan klien, dan merancang pengujian untuk

semua fitur dalam sistem. Dalam sistem aplikasi berbasis komputer, *use-case* diagram sangat membantu dalam menggambarkan kebutuhan yang sesuai dengan proses bisnis dan cakupan masalah yang ada. Berikut simbol – simbol yang ada pada *use case* diagram pada Tabel 2.3.

Tabel 2. 3 Use-case diagram

No	Simbol	Nama	Keterangan
1		Use Case	Deskripsi dari urutan aksi - aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
2		Association	Menghubungkan antara objek satu dengan objek lainnya
3		System	Pembatas sistem yang terhubung dengan aktor yang menggunakannya
4		Actor	Mengidentifikasi aktor yang berperan dalam use case
5		Include	Mengarahkan bahwa use case sumber secara tersira
6		Extend	Menunjukkan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan