



BAB II
DASAR
TEORI

BAB II DASAR TEORI

2.1 Landasan Teori

2.2.1 Sistem

Sistem adalah serangkaian komponen yang saling berinteraksi untuk mencapai tujuan tertentu. Komponen-komponen tersebut bekerja sama melalui tiga tahapan: input, proses, dan output. Sistem juga dapat dipandang sebagai kumpulan sub sistem dan komponen yang bekerja bersama-sama untuk menghasilkan output sesuai dengan tujuan yang telah ditetapkan sebelumnya [1].

2.2.2 Metode Pengembangan Sistem *Rapid Application Development (RAD)*

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Rapid Application Development (RAD)*. Metode RAD menitikberatkan pada siklus pengembangan yang singkat, padat, dan cepat. Pengembangan aplikasi dilakukan dengan pendekatan iteratif untuk membangun sistem secara efektif yang terdiri dari empat tahap seperti berikut:

a. Rencana Kebutuhan (*Requirements Planning*)

Pada tahap ini pelapor dan penulis saling bertemu untuk meneliti dan memecahkan masalah yang sedang terjadi, menentukan apa saja yang dibutuhkan untuk membuat sistem aplikasi, karena tahap ini merupakan langkah awal keberhasilan pembuatan sistem serta dapat menghindari kesalahan komunikasi antara pelapor dan penulis.

b. Desain Pelapor (*User Design*)

Tahap membuat rancangan yang akan diusulkan agar sesuai dengan kebutuhan, berjalan sesuai rencana dan diharapkan dapat mengatasi masalah yang sedang terjadi. Pada penelitian ini, desain sistem yang digambarkan menggunakan *Tools Unified Modeling Language (UML)*.

c. Pengembangan

Tahap ini adalah tahap memulai membuat sistem yang sudah direncanakan. Memulai menyusun suatu kode program atau biasa disebut *coding*, untuk mengubah desain sistem yang telah dibuat menjadi sebuah aplikasi yang telah direncanakan agar dapat digunakan.

d. Implementasi

Tahap ini adalah pengujian keseluruhan sistem yang dibangun semua komponen perlu diuji secara menyeluruh yang berfokus pada spesifikasi fungsional dari perangkat lunak [2].

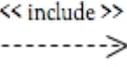
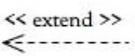
2.2.3 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa yang digunakan untuk menspesifikasi, memvisualisasikan, membangun, dan mendokumentasikan informasi terkait pembuatan perangkat lunak, baik berupa model, deskripsi, atau perangkat lunak sistem. UML menggunakan konsep orientasi objek dan menyediakan notasi-notasi yang membantu pemodelan sistem dari berbagai perspektif [3].

2.2.4 Use Case Diagram

Use Case Diagram adalah representasi visual dari perilaku sistem yang diinginkan dan dibutuhkan oleh pelapor. Diagram ini digunakan untuk mengidentifikasi fungsi-fungsi yang ada dalam sistem serta menentukan aktor-aktor yang memiliki hak untuk menggunakan fungsi-fungsi tersebut. Simbol-simbol *Use Case Diagram* dapat dilihat pada Tabel 2.1 [4].

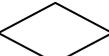
Tabel 2. 1 Simbol *Use Case Diagram*

No	Simbol	Nama Simbol	Keterangan
1		<i>Actor</i>	<i>Actor</i> digunakan untuk menjelaskan entitas atau individu yang terlibat dalam interaksi dengan sistem tersebut.
2		<i>Use Case</i>	<i>Use Case</i> digunakan untuk mengilustrasikan cara kerja sistem tanpa harus membeberkan struktur internal sistem tersebut.
3		<i>Association</i>	Komunikasi antara aktor dan kasus pelapor yang terlibat dalam <i>Use Case</i> atau memiliki interaksi dengan aktor.
5		<i>Include</i>	Penambahan relasi pada sebuah <i>Use Case</i> terjadi ketika ada kebutuhan untuk menjalankan fungsi tertentu atau ketentuan khusus yang harus dipenuhi agar <i>Use Case</i> tersebut dapat dijalankan.
6		<i>Extend</i>	Penambahan relasi ke sebuah <i>Use Case</i> mengacu pada kemampuan <i>Use Case</i> tersebut untuk beroperasi secara mandiri, tanpa bergantung pada <i>Use Case</i> tambahan.

2.2.5 Flowchart

Flowchart adalah representasi visual dari langkah-langkah pemecahan masalah dengan menggunakan simbol-simbol khusus. Flowchart juga menggambarkan aliran atau urutan algoritma. Meskipun memiliki keunggulan visual, tetapi bisa menjadi rumit apabila algoritma yang disusun terlalu panjang dan tidak efisien. Berikut ini simbol-simbol dan kegunaan dari simbol-simbol yang sering digunakan untuk menggambarkan suatu algoritma dalam bentuk diagram alir dapat dilihat pada Tabel 2.2 [5].

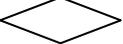
Tabel 2. 2 Simbol Flowchart

No.	Simbol	Nama simbol	Keterangan
1		<i>Terminator</i>	Awal atau akhir suatu program (prosedur).
2		<i>Output/Input</i>	Simbol ini menunjukkan masukan atau keluaran dari proses.
3		<i>Process</i>	Simbol ini menunjukkan langkah atau proses yang dilakukan dalam alur.
4		<i>Decision</i>	Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan, ya atau tidak.
5		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus kerja dalam suatu proses. Simbol arus ini sering disebut juga dengan <i>connecting line</i> .
6		<i>Document</i>	Simbol ini digunakan untuk merepresentasikan langkah di mana pelapor memasukkan data atau informasi ke dalam sistem.
7		<i>Manual symbol</i>	Menyatakan suatu proses yang tidak dilakukan komputer

2.2.6 Entity-Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan suatu representasi visual yang menggambarkan detail informasi mengenai sebuah model data. Tujuannya adalah untuk mempermudah proses pengembangan sistem serta membantu dalam memenuhi kebutuhan analisis sistem. Berikut adalah beberapa simbol ERD yang dapat dilihat pada Tabel 2.3 [6].

Tabel 2. 3 Simbol Entity Relationship Diagram (ERD)

No.	Simbol	Nama Simbol	Keterangan
1		Entitas	Merupakan data inti yang akan disimpan pada tabel basis data. Penamaan entitas biasanya mengarah kata benda dan belum merupakan nama tabel.
2		Atribut	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3		Relasi	Relasi yang menghubungkan antar entitas biasanya diawali kata kerja.
4		<i>Link</i>	Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya

Simbol ERD juga memuat kardinalitas yang menandakan jumlah maksimum entitas yang dapat terhubung dengan entitas di himpunan entitas lainnya. Kardinalitas relasi antara dua himpunan entitas bisa berupa satu ke satu (*one to one*), satu ke banyak (*one to many*), atau banyak ke banyak (*many to many*). Simbol-simbol kardinalitas ERD terdapat pada Tabel 2.4 di bawah ini.

Tabel 2. 4 Simbol Kardinalitas *Entity Relationship Diagram (ERD)*

No.	Simbol	Nama	Keterangan
1		<i>Association</i>	<i>Association</i> merupakan penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian (kardinalitas).
2		Relasi Satu ke Satu (<i>One to One</i>)	Relasi satu ke satu merupakan relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya.
3		Relasi Satu ke Banyak (<i>One to Many</i>)	Relasi satu ke banyak merupakan relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak tapi tidak sebaliknya.

2.2.7 Database Management System (DBMS)

Database Management System (DBMS) adalah sebuah aplikasi perangkat lunak yang dibuat untuk mempermudah pengelolaan serta pemanfaatan data dalam jumlah besar. DBMS terdiri dari sejumlah program yang berhubungan dengan data, yang bertugas mengakses informasi yang tersimpan dalam sebuah basis data. Poin penting dari DBMS adalah menyediakan fasilitas untuk menyimpan dan mengambil data dengan cepat dan efisien dari basis data tersebut [7].

2.2.8 MySQL

MySQL adalah sebuah sistem manajemen basis data SQL yang merupakan perangkat lunak sumber terbuka dan saat ini tengah digemari. Sistem basis data MySQL mendukung berbagai fitur seperti kemampuan *multithreaded*, *multi-user*, serta merupakan SQL DBMS. Basis data ini didesain untuk memenuhi kebutuhan sistem basis data yang cepat, andal, dan mudah digunakan oleh pelapor. MySQL adalah basis data multi-user yang menggunakan bahasa *Structured Query Language (SQL)* [1].

2.2.9 Bahasa Pemrograman PHP

PHP (*Hypertext Preprocessor*) ialah suatu bahasa pemrograman yang bertugas mengubah baris-baris kode program menjadi kode mesin yang bisa dimengerti oleh komputer, terutama pada bagian server, serta dapat dimasukkan ke dalam HTML. Sebagai salah satu bahasa pemrograman web, PHP dibuat oleh dan untuk para pengembang web [1].

2.2.10 Framework

Framework merupakan struktur yang memberikan kerangka kerja bagi pengembang untuk menyusun kode dengan lebih mudah. Dengan kata lain, *framework* adalah sekumpulan fungsi atau pustaka yang dapat langsung digunakan oleh programmer [8].

2.2.11 Laravel

Laravel adalah suatu rangkaian kerja web yang menggunakan bahasa pemrograman PHP sebagai dasarnya, yang memiliki karakteristik sebagai perangkat lunak sumber terbuka dan gratis. Dikembangkan oleh Taylor Otwell, Laravel ditujukan untuk membangun aplikasi web yang mengadopsi pola MVC. Dalam Laravel, terdapat sistem routing yang menghubungkan permintaan dari pelapor dengan pengontrol (*controller*), sehingga pengontrol tidak langsung menerima permintaan tersebut. [9].