BAB II
LANDASAN TEORI

2.1 Landasan Teori

2.1.1 Sistem Informasi

Sistem informasi adalah sebuah kombinasi dari berbagai komponen teknologi informasi yang terintegrasi dengan baik dan saling bekerjasama untuk memproses, mengelola, dan menyebarkan informasi. Komponen-komponen ini meliputi perangkat keras, perangkat lunak, data, prosedur, dan manusia yang berinteraksi untuk menghasilkan informasi yang berguna. Informasi yang dihasilkan digunakan untuk mendukung pengambilan keputusan, koordinasi, kontrol, analisis, dan visualisasi dalam berbagai konteks. Selain itu, sistem informasi memungkinkan terjadinya komunikasi yang lebih efektif dan efisien dalam suatu organisasi atau kelompok, sehingga memfasilitasi aliran informasi yang lebih lancar dan konsisten di seluruh unit atau departemen yang ada[4].

2.1.2 Bank Sampah

Bank sampah merupakan suatu kegiatan yang bersifat *social engineering* dimana masyarakat belajar untuk memilah sampah serta menumbuhkan kesadaran masyarakat dalam pengolahan sampah secara bijak dan pada gilirannya akan mengurangi sampah yang diangkut ke TPA. Bank Sampah merupakan salah satu *alternative* yang memberikan pengajaran kepada masyarakat agar dapat memanfaatkan kembali sampah yang dikenal dengan istilah 3R (*Reduce*, *Reuse* dan *Recycle*)[5].

2.1.3 *Website*

Website adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi[6]. *Web* merupakan salah satu aplikasi yang berisikan dokumen – dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (*hypertext transfer protokol*) dan untuk mengakses menggunakan perangkat lunak yang disebut *browser*".

2.1.4 *Framework Laravel*

Laravel adalah sebuah *framework* PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). *Laravel* adalah pengembangan *website* berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaksis yang ekspresif, jelas dan menghemat waktu[7].

2.1.5 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah praktik pengembangan perangkat lunak yang bertujuan untuk menciptakan produk perangkat lunak yang bernilai ekonomi dan bermanfaat bagi pelanggan serta dapat digunakan secara efisien oleh pengguna. Ini melibatkan penggunaan prinsip dan konsep

rekayasa untuk memastikan keandalan, biaya, dan waktu pengembangan yang optimal serta memenuhi kriteria seperti dapat dipelihara, dapat diandalkan, efisien dalam penggunaan sumber daya, dan mudah digunakan sesuai kebutuhan[8]. Berikut ini metode dan *tools* pembangunan rekayasa perangkat lunak yang digunakan :

a. Metode Pengembangan Sistem

Metode pengembangan yang akan digunakan dalam penelitian ini adalah metode *prototyping*. Model *prototyping* atau Metode *Prototype* adalah metode pengembangan sistem perangkat lunak (SLDC) di mana prototipe dibangun, diuji dan kemudian dikerjakan ulang seperlunya sampai hasil yang dapat diterima dicapai dari sistem atau produk yang lengkap dapat dikembangkan. Sistem ini menggunakan metode *prototype* dikarenakan dalam metode ini terdapat komunikasi antara pengembang dengan klien yang bertujuan agar pengembangan sistem ini dapat berlangsung dengan baik dan sesuai dengan kebutuhan dari klien. Adapun penjelasan dari tahapan metode *Prototyping*, yaitu:

1. Pengumpulan Kebutuhan
   Pengumpulan kebutuhan dilakukan dengan mendefinisikan secara rinci terkait sistem yang akan dibuat. Klien dan tim pengembang harus bertemu untuk mendiskusikan detail sistem yang akan dibuat untuk memenuhi kebutuhan para pengguna aplikasi.

2. Membangun *Prototyping*
   Tahap selanjutnya membangun *prototype*. Pada tahap membangun *prototype* langkah yang dilakukan yaitu membuat rancangan UML yang terdiri dari Use case diagram, *activity* diagram, *sequence* diagram, *class* diagram, *flowchart*, dan *mockup*.

3. Evaluasi *Prototyping*
   Evaluasi *prototyping* merupakan kegiatan untuk mengecek apakah *prototype* yang sudah dibangun sesuai dengan keinginan pengguna.

4. Mengkodekan Sistem
   Mengkodekan sistem adalah tahapan untuk mengubah *prototype* kedalam bahasa pemrograman. Pada sistem ini menggunakan Bahasa pemrograman PHP dengan dibantu *framework Laravel* serta menggunakan *database MySql*. Sistem akan dibuat dalam basis *website*.

5. Menguji Sistem
   Setelah sistem selesai dibuat, kemudian dilakukan pengujian menggunakan metode *black box* testing. Metode *black box* testing adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Sistem ini menggunakan *black box* testing dikarenakan dalam proses pengujian dilakukan tanpa harus memeriksa kode program.

6. Evaluasi Sistem
   Evaluasi sistem adalah tahap yang dilakukan untuk mengevaluasi sistem aplikasi yang telah di buat dengan mengevaluasi keseluruhan dari aplikasi.

7. Menggunakan Sistem

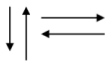
   Tahap terakhir adalah penerapan sistem. Setelah sistem dievaluasi dan mendapat persetujuan dari pengguna, sistem tersebut kemudian diimplementasikan secara langsung.

b. Alat Bantu atau *tools*

1. *Flowchart*

Flowchart merupakan visualisasi tentang bagaimana suatu program berjalan dari satu proses ke proses lainnya. Ini juga membantu dalam memecah rangkaian prosedur agar lebih mudah dimengerti[9]. Dibawah ini adalah simbol yang digunakan :

Tabel 2. 1 Tabel *Flowchart*

No	Simbol	Nama	Keterangan
1		*Flow Directon Symbol*	Digunakan untuk mengaitkan simbol satu dengan yang lainnya.
2		*Terminator Symbol*	Digunakan sebagai awal atau akhir dari suatu aktivitas.
3		*Connector Symbol*	Digunakan sebagai simbol yang menghubungkan proses dalam halaman.
4		*Processing Symbol*	Digunakan untuk menunjukkan pengolahan yang dilakukan oleh komputer.
5		Simbol *Input-Output*	Digunakan untuk menunjukkan proses *input* atau *output*.
6		Simbol *Manual Operation*	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer adalah simbol yang menandakan bahwa proses tersebut dilakukan secara manual oleh manusia atau sistem lainnya.

7		Simbol *Decision*	Simbol yang menunjukkan pemilihan proses berdasarkan kondisi yang ada.

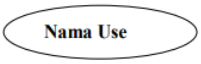
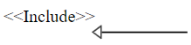
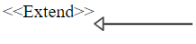
2. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah sebuah bahasa grafis yang digunakan untuk menspesifikasikan dan mendokumentasikan sistem pengembangan perangkat lunak berbasis objek. UML mendukung visualisasi, pembangunan, dan dokumentasi sistem tersebut. UML juga menyediakan standar untuk membuat *blue print* sistem, yang mencakup konsep proses bisnis, penulisan kelas-kelas dalam bahasa pemrograman tertentu, skema *database*, dan komponen lain yang dibutuhkan dalam sistem perangkat lunak[11]. UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya [10] :

a. *Use case Diagram*

Use Case Diagram digunakan untuk mengilustrasikan fungsionalitas yang disediakan oleh keseluruhan organisasi dan sering dimanfaatkan untuk menentukan konteks sistem[12], antara lain:

Tabel 2. 2 Tabel Use case Diagram

No	Simbol	Nama	Keterangan
1		*Actor*	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi dibuat di luar sistem informasi tersebut..
2		*Use Case*	Fungsionalitas yang disediakan oleh sistem saat unit dan aktor saling bertukar pesan..
3		*Association* / Asosiasi	Sesuatu yang menghubungkan satu objek dengan objek lainnya
4		*Include*	Hubungan antara *Use case* tambahan dan *Use case* di mana *Use case* yang ditambahkan memerlukan *Use case* ini untuk menjalankan fungsinya
5		*Extends*	Hubungkan kembali *Use case* tambahan ke *Use case* di mana *Use case* yang ditambahkan dapat independen bahkan tanpa *Use case* tambahan.

2.1.6 MVC (Model View Controller)

Model-View-Controller atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasinya kebanyakan *framework* dalam aplikasi *website* adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen

utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web[11].

a. *Model*
   Model mewakili struktur data. Biasanya model berisi fungsi – fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
b. *View*
   View adalah bagian yang mengatur tampilan ke pengguna. Bisa di katakan berupa halaman web.
c. *Controller*
   Controller merupakan bagian yang menjembatani *model* dan *view*. *controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

2.1.7 PHP

PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman untuk kompilasi Baris kode program dalam kode mesin sisi server yang dapat dimengerti komputer yang dapat ditambahkan ke *HTML*.PHP adalah bahasa pemrograman berbasis web yang ditulis oleh pengembang *web*[12].

2.1.8 *Framework Laravel*

Laravel adalah kerangka kerja berdasarkan bahasa pemrograman PHP, yang dengannya situs web dapat dikembangkan secara optimal. Dengan menggunakan *Laravel website* yang dihasilkan lebih dinamis, dan keuntungan menggunakan *framework Laravel* adalah biaya pembuatan website lebih murah. Hal ini tidak hanya menghemat biaya, tetapi juga lebih hemat waktu. Fitur bawaan lengkap, salah satunya adalah fitur *autentikasi*[13].

2.1.9 *Database*

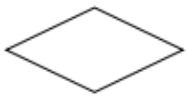
Database adalah kumpulan data terkait yang disimpan secara terstruktur di komputer. Ini juga mencakup serangkaian program yang memungkinkan pengguna mengakses, menyimpan, dan memanipulasi data ini. Tujuan utama dari sistem manajemen basis data (DBMS) adalah untuk menyediakan cara yang sederhana, cepat, dan efisien untuk menyimpan dan mengambil informasi dari basis data. Sistem basis data dirancang untuk menangani informasi dalam jumlah besar, dan data ini sering kali diproses dengan beberapa analisis untuk membantu mengambil keputusan. Basis data memiliki banyak aplikasi dalam kehidupan sehari-hari, termasuk data bisnis, data perbankan, data akademik, dan banyak lagi. Dalam konteks penggunaan sehari-hari, *database* membantu menyimpan, mengelola, dan mengakses informasi yang diperlukan untuk berbagai keperluan seperti manajemen inventaris, manajemen pelanggan, manajemen transaksi, dll[14].

A. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) merupakan sebuah metode dalam pemodelan basis data yang digunakan sebagai kerangka konseptual, di mana berbagai jenis model data dari sistem semantik direpresentasikan. ER menggunakan sistem basis data relasional yang memiliki ciri-ciri *top-down*. Diagram yang digunakan dalam ER adalah gambaran model *Entity-Relationship* yang dikenal sebagai *Entity-Relationship Diagram* (ERD)[15]. Fungsi

ERD digunakan untuk memodelkan struktur data dan hubungan antara entitas dalam suatu sistem informasi. Simbol ERD dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Tabel ERD

No	Simbol	Nama	Keterangan
1		*Entity*	Objek riil yang dapat di bedakan satu dengan yang lain.
2		*Attribute*	Unsur-unsur suatu entitas mempunyai fungsi untuk menjelaskan entitas tersebut.
3		*Relationship*	menggambarkan hubungan atau koneksi yang terjadi antara satu atau lebih entitas dalam suatu sistem informasi..
4		*Link*	Simbol berupa garis ini digunakan sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya

B. Query

1. *DDL (Data Definition Language)*

DDL atau *Data Definition Language* adalah bagian dari bahasa *SQL* yang berkaitan dengan mendefinisikan struktur *database*, khususnya *database* dan tabel. Ini mencakup perintah-perintah untuk membuat, mengubah, dan menghapus objek-objek dalam *database* seperti tabel dan indeks. Dengan menggunakan *DDL*, pengguna dapat mengatur struktur *database* sesuai dengan kebutuhan aplikasi mereka Perintah-perintah utama dalam *DDL* antara lain[16] *:*

(a) *CREATE*: Digunakan untuk membuat objek baru dalam *database*, seperti tabel, indeks, atau tampilan.

(b) *ALTER:* Mengubah struktur objek yang sudah ada dalam *database*, misalnya menambahkan kolom baru ke dalam tabel atau mengubah tipe data kolom.
(c) *DROP:* Menghapus objek dari *database*, seperti tabel, indeks, atau tampilan.

2. *DML*

Data Manipulation Language atau DML adalah bagian dari bahasa *SQL* yang terkait dengan manipulasi atau pengolahan data dalam tabel. Ini meliputi perintah-perintah untuk menambah, mengubah, menghapus, dan mengambil data dari tabel. Dengan menggunakan *DML*, pengguna dapat melakukan operasi – operasi penting seperti menyisipkan data baru, memperbarui nilai-nilai yang ada, menghapus data yang tidak diperlukan, dan mengekstrak informasi dari tabel sesuai dengan kebutuhan aplikasi mereka, adapun perintah – perintahnya antara lain[16] :

(a) *INSERT*: digunakan untuk menyisipkan data baru ke dalam tabel.
(b) *SELECT*: Perintah *SELECT* digunakan untuk mengambil data dari satu atau beberapa tabel.
(c) *UPDATE*: Perintah *UPDATE* digunakan untuk memperbarui nilai-nilai dalam satu atau beberapa baris data dalam tabel.
(d) *DELETE*: Perintah *DELETE* digunakan untuk menghapus satu atau beberapa baris data dari sebuah tabel.