

# 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 sekumpulan elemen atau komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu. Sistem terdiri dari tiga unsur yaitu : *input* (masukan), proses dan *output* (pengeluaran). *Input* merupakan komponen penggerak atau pemberi tenaga di mana sistem itu dioperasikan, sedangkan *output* adalah hasil operasi. Dalam pengertian sederhana *output* berarti yang menjadi tujuan sasaran atau target pengoperasian suatu sistem [3].

##### **2.1.2 Informasi**

Informasi adalah data atau objek yang diolah terlebih dahulu sedemikian rupa sehingga dapat disusun dan diperjelas dengan baik sehingga memiliki makna bagi penerima yang kemudian menjadi pengetahuan bagi penerima tentang suatu hal tertentu yang membantu mengambil keputusan dengan tepat [4].

##### **2.1.3 Sistem Informasi**

Sistem informasi merujuk pada suatu sistem yang dibangun secara umum menggunakan perangkat keras dan perangkat lunak, serta komponen manual, yang memungkinkan untuk mengumpulkan, menyimpan, dan memproses data menggunakan bahasa pemrograman. Tujuannya adalah untuk menyediakan *output* yang dibutuhkan oleh pengguna, baik dalam bentuk informasi maupun laporan. Beberapa komponen yang bekerja sama untuk mencapai tujuan dalam sistem [4], adalah sebagai berikut :

1. Penetapan tujuan sistem

Menentukan arah akhir atau hasil yang ingin dicapai dengan penggunaan sistem tersebut.

2. Pengaturan batasan sistem

Aturan-aturan yang membatasi dan mengatur jalannya sistem. Batasan tersebut dapat berupa aturan organisasi, jangka waktu tertentu, atau alokasi biaya yang diperbolehkan.

3. Pengawasan melalui kontrol sistem

Digunakan untuk mengawasi dan memastikan bahwa sistem berjalan sesuai dengan rencana. Ini melibatkan pengawasan terhadap waktu pelaksanaan, biaya, dan aspek-aspek kinerja lainnya.

#### 4. Masukan (*Input*)

Komponen ini melibatkan data yang diterima oleh sistem sebagai masukan. Masukan tersebut bisa berupa berbagai jenis data dan nilai yang diperlukan untuk pengolahan selanjutnya.

#### 5. Proses

Elemen sistem yang bertugas memproses masukan (input) menjadi informasi yang berguna. Ini melibatkan pengolahan data melalui serangkaian langkah untuk menghasilkan *output* yang diinginkan.

#### 6. Keluaran (*Output*)

Hasil akhir dari proses yang dilakukan oleh sistem. Dalam konteks ini, keluaran dapat berupa laporan atau grafik yang menyajikan informasi yang telah diolah.

#### 7. Umpan balik

Membantu dalam mengevaluasi kinerja sistem dan memperbaiki kekurangan atau ketidaksesuaian dengan kebutuhan organisasi.

### **2.1.4 Laporan**

Laporan merupakan bentuk penyajian fakta mengenai hasil dari suatu kegiatan, yang pada dasarnya berkaitan dengan tanggung jawab. Laporan memiliki peranan penting bagi perusahaan, karena dianggap mampu menjembatani hubungan antara pimpinan dan pegawai dalam perusahaan. Tujuan dari penyusunan laporan adalah untuk memberikan informasi mengenai kinerja terukur suatu kegiatan kepada pimpinan atas hasil yang dicapai. Selain itu, laporan juga dapat digunakan sebagai upaya perbaikan berkelanjutan bagi suatu perusahaan dalam rangka meningkatkan kinerjanya [5].

### **2.1.5 Pendapatan**

Pendapatan dapat didefinisikan dalam berbagai cara, tergantung pada konteksnya. Berikut ini akan dijelaskan beberapa pengertian dari pendapatan. Pendapatan (*revenue*) adalah penghasilan yang diperoleh dalam jangka waktu tertentu. Pendapatan mencakup semua yang diterima dari hasil penjualan barang dan jasa dalam sebuah unit usaha [6]. Pendapatan merupakan unsur yang sangat penting dalam usaha perdagangan, karena setiap usaha pasti ingin mengetahui jumlah pendapatan yang diperoleh selama menjalankan kegiatan usahanya [7].

### **2.1.6 Laporan Keuangan**

Laporan keuangan merupakan laporan yang menunjukkan kondisi keuangan perusahaan pada suatu waktu tertentu atau dalam periode tertentu. Laporan ini berfungsi sebagai alat untuk

memperoleh informasi mengenai posisi keuangan dan kinerja yang telah dicapai oleh perusahaan. Tujuan dari laporan keuangan adalah untuk menyediakan informasi mengenai posisi keuangan, kinerja, serta perubahan posisi keuangan perusahaan yang bermanfaat bagi berbagai pihak dalam pengambilan keputusan [8].

### 2.1.7 Website

*Website* atau situs web adalah sekelompok halaman web yang terhubung satu sama lain dengan topik yang saling terkait. Situs web juga dapat menyertakan berkas-berkas seperti gambar, video, atau berkas lainnya. Situs web merupakan kumpulan dari halaman-halaman web yang telah dipublikasikan di internet dan dapat diakses oleh pengguna internet melalui domain atau URL (*Unified Resource Locator*) dengan mengetikkan alamatnya [9].

### 2.1.8 PHP

*PHP* adalah kependekan dari *PHP Hypertext Preprocessor*, yang berperan sebagai bahasa skrip *server-side* dalam pengembangan *web* dan disematkan dalam dokumen *HTML*. *PHP* merupakan perangkat lunak sumber terbuka yang didistribusikan dan dilisensikan secara gratis. Bahasa pemrograman *PHP* adalah jenis bahasa skrip yang tidak memerlukan proses kompilasi oleh pengguna, dan *web server* yang mendukung *PHP* dapat dijumpai di berbagai tempat, salah satunya adalah Apache. *PHP* adalah bahasa sumber terbuka yang dapat digunakan di berbagai mesin seperti Linux, Unix, Macintosh, dan Windows. Selain itu, *PHP* dapat dijalankan secara runtime melalui konsol dan mampu mengeksekusi perintah-perintah sistem [10].

### 2.1.9 Framework Laravel

*Framework* merupakan suatu komponen yang dimanfaatkan untuk menyederhanakan proses pembuatan kode oleh pengembang. Dengan kata lain, *framework* adalah gabungan fungsi atau pustaka yang telah siap digunakan oleh para pemrogram [11]. *Laravel* adalah salah satu *php framework* paling populer dan paling banyak digunakan untuk membuat *website* saat ini. Dengan menggunakan *laravel* proses pengembangan aplikasi menjadi lebih cepat dan *powerfull*. Dalam pengembangan *website* menggunakan *laravel*, ada 2 *tools* yang akan sering kita pakai. yaitu *composer* dan *artisan*. Kelebihan dari *framework laravel* jika dibandingkan dengan *framework* lain adalah sebagai berikut :

1. Lebih aman

*Laravel* adalah *framework* *PHP* yang memberikan beberapa pilihan yang dapat digunakan agar aplikasi tetap aman. Salah satu sistem keamanan yang dimiliki *Laravel* adalah *PDO* yang bisa mencegah *SQL Injection*.

## 2. Gratis (*Open-Source*)

Laravel merupakan *framework* yang bersifat *open source*, bisa digunakan dengan gratis. *Framework* ini juga banyak digunakan karena *developer* dapat menggunakannya untuk dikembangkan lagi.

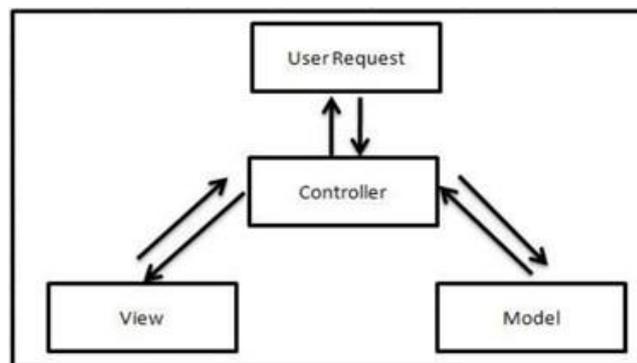
## 3. Menggunakan konsep M-V-C

Laravel merupakan konsep M-V-C (*Hierarchical ModelView-Controller*) yang memungkinkan pemisahan antara layer *application-logic* dan *presentation*.

Menggunakan konsep ini kode PHP, *query* Mysql, Javascript dan CSS dapat saling dipisahkan sehingga ukuran file menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau *maintenance*.

- Model Kode merupakan program (berupa OOP *class*) yang digunakan untuk berhubungan dengan *database* MySQL sekaligus untuk memanipulasinya (*input-edit-delete*).
- View* merupakan kode program berupa *template* atau PHP untuk menampilkan data pada *browser*.
- Controller* merupakan kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol model dan *view*.

Adapun alur dari program aplikasi berbasis *laravel* yang menggunakan konsep M-V-C ditunjukkan pada Gambar 2.2 berikut :



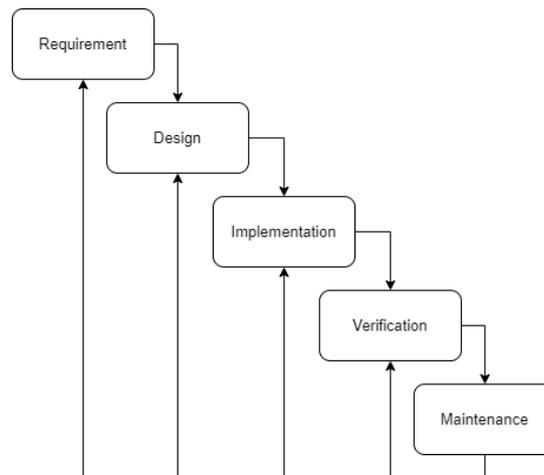
**Gambar 2. 1** Konsep Aliran M-V-P

### 2.1.10 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah suatu kegiatan yang bersifat kreatif dan terstruktur berdasarkan suatu disiplin ilmu yang membangun perangkat lunak berdasarkan pemahaman terhadap aspek masalah tertentu. Proses perangkat lunak merupakan suatu kerangka kerja yang digunakan untuk menjalankan tugas-tugas yang diperlukan dalam pembangunan perangkat lunak dengan standar kualitas yang tinggi. Proses perangkat lunak menentukan pendekatan yang

diterapkan dalam pengembangan perangkat lunak. Pengembangan perangkat lunak juga mencakup penerapan teknologi yang merinci metode-metode, teknis, alat bantu otomatis, dan prosedur-prosedur yang sering disebut sebagai proses [12].

*System Development Life Cycle* (SDLC) adalah suatu proses yang melibatkan pengembangan atau modifikasi sistem perangkat lunak, menggunakan model dan metodologi yang umumnya digunakan oleh para profesional untuk mengembangkan sistem perangkat lunak sebelumnya. Untuk mempermudah dalam proses pengembangan perangkat lunak, salah satunya model *waterfall* menurut *Roger S. Pressman Ph.D.* dapat dilihat pada Gambar 2.2 [2].



**Gambar 2. 2** Tahapan metode waterfall menurut Roger S. Pressman Ph.D

Tahapan-tahapan dari metode ini adalah sebagai berikut :

1. Analisis Kebutuhan (*Requirement Analysis*)

Tahap ini merupakan proses penentuan fitur, kendala, dan tujuan sistem melalui dialog dengan pengguna sistem. Semua elemen tersebut akan dijelaskan secara terperinci dan menjadi panduan dalam proses pembuatan sistem.

2. Desain Sistem (*Design System*)

Proses pengumpulan kebutuhan perangkat lunak bertujuan untuk memahami karakteristik dari program yang sedang dibangun oleh rekayasa perangkat lunak, atau analisis harus memahami fungsi perangkat lunak yang diinginkan oleh pengguna. Pada tahap ini, dilakukan identifikasi dan pembuatan gambaran mengenai abstraksi dasar dari sistem perangkat lunak beserta hubungan-hubungannya.

3. Penulisan Kode Program (*Implementasi*)

Hasil dari desain perangkat lunak akan diimplementasikan melalui serangkaian program atau unit program. Setiap unit akan menjalani pengujian untuk memastikan bahwa telah memenuhi spesifikasinya.

#### 4. Pengujian Program (*Verification*)

Pada fase ini, setiap unit program akan digabungkan dan diuji secara bersama-sama sebagai satu sistem menyeluruh untuk memverifikasi bahwa sistem telah memenuhi semua persyaratan yang telah ditetapkan.

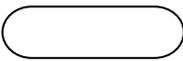
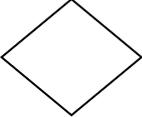
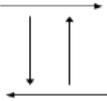
#### 5. Penerapan Program dan Pemeliharaan (*Maintenance*)

Tahap ini mencakup pemasangan sistem pada pengguna, dan juga melibatkan perbaikan kesalahan yang mungkin tidak terdeteksi selama proses pembuatan. Pada tahap ini, terdapat juga potensi pengembangan sistem melalui penambahan fitur dan fungsi baru.

### 2.1.11 Flowchart

*Flowchart* merupakan suatu diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Pada Tabel 2.1 [13] merupakan simbol-simbol *flowchart* :

**Tabel 2. 1** Simbol-simbol *flowchart*

No	Simbol	Nama	Keterangan
1		<i>Terminator Symbol</i>	Simbol yang menyatakan awal atau akhir suatu program.
2		<i>Input-Output</i>	Memasukkan data maupun menunjukkan hasil dari suatu proses tanpa tergantung dengan jenis pendataannya.
3		<i>Processing Symbol</i>	Menyatakan suatu proses yang dilakukan oleh komputer
4		<i>Decision symbol</i>	Menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya atau tidak.
5		<i>Flow symbol</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan <i>connecting line</i> .
5		<i>Manual symbol</i>	Menyatakan suatu proses yang tidak dilakukan komputer
6		<i>Document Symbol</i>	Merupakan simbol untuk data yang terbentuk informasi.

### 2.1.12 Unified Modeling Language (UML)

Salah satu standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur disebut dengan Unified Modeling Language (UML). UML menyederhanakan permasalahan-permasalahan yang kompleks sehingga dapat dipelajari dan dipahami dengan mudah. UML dapat menggambarkan hubungan antar kelas dengan garis lurus [13]. Terdapat beberapa diagram UML yang digunakan dalam pengembangan sebuah sistem yaitu :

#### 1. Use Case Diagram

*Use case diagram* merupakan permodelan untuk melakukan sistem informasi yang akan dibuat. *Use case diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [13]. Tabel 2.2 menunjukkan simbol-simbol yang ada pada use case diagram.

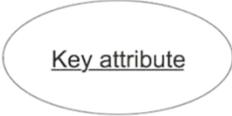
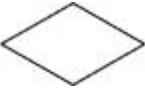
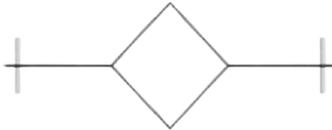
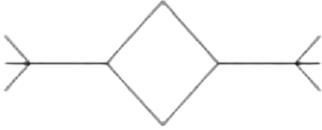
**Tabel 2. 2** Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
2		<i>Actor</i>	Mewakili peran orang, sistem yang lain, atau alat Ketika berkomunikasi dengan <i>use case</i> .
3		<i>Assosiation</i>	Menghubungkan antara objek satu dengan objek lainnya.
4		<i>Include</i>	Menspesifikan bahwa <i>use case</i> sumber segala eksplisit.
5		<i>Extend</i>	Menspesifikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber suatu titik yang diberikan.
6		<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.

## 2. Entity Relationship Diagram (ERD)

ERD adalah sebuah gambaran konseptual yang menggambarkan keterkaitan antar penyimpanan data. ERD berguna untuk merancang struktur data dan hubungan antar data. Model ERD dapat diuji dengan cara mengabaikan proses yang terlibat. Berikut simbol-simbol yang terdapat pada entity relationship diagram dapat dilihat pada Tabel 2.3 [14].

**Tabel 2. 3** Simbol Entity Relationship Diagram

Notasi	Komponen	Keterangan
	Entitas	Individu yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain.
	Atribut	Properti yang dimiliki oleh suatu entitas, dimana dapat mendeskripsikan karakteristik dari entitas tersebut.
	Atribut Kunci	Properti yang berperan sebagai kunci dari suatu entitas dan memiliki sifat yang unik.
	Relasi	Menunjukkan hubungan diantara sejumlah entitas yang berbeda.
	<i>One to One</i>	Relasi yang menunjukkan bahwa setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada himpunan entitas kedua.
	<i>One to Many</i>	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak tetapi tidak sebaliknya.
	<i>Many to Many</i>	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak atau sebaliknya. Setiap entitas dapat berelasi dengan banyak entitas pada himpunan entitas yang lain.

Metode pemodelan data dengan menggunakan *Entity-Relationship Diagram* (ERD) digunakan untuk mengidentifikasi objek data dan hubungannya dengan menggunakan notasi grafis dalam perancangan perangkat lunak. Atribut dari masing-masing objek data digambarkan dengan menggunakan deskripsi objek data. ERD hanya berfokus pada data dengan menunjukkan jaringan data yang ada untuk suatu sistem yang diberikan.

### 2.1.13 Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek merupakan pemrograman berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam *class-class* atau *object-object*. Syarat sebuah bahasa pemrograman bisa digolongkan sebagai berorientasi objek adalah bila bahasa pemrograman tersebut memiliki fitur untuk mengimplementasikan ke 4 (empat) konsep berorientasi objek yaitu abstraksi, *encapsulation*, *polymorphison*, dan *inheritance* [15]. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi.

- a. Abstraksi adalah sebuah prinsip yang digunakan untuk merepresentasikan dunia nyata yang kompleks ini menjadi sebuah model yang sederhana dengan menghiraukan aspek - aspek lainnya yang tidak sesuai dengan permasalahan.
- b. *Encapsulation* adalah bagaimana membungkus data dan method yang menyusun class hingga class dipandang sebagai suatu modul.
- c. *Polymorphison* adalah sesuatu yang memiliki banyak bentuk, diartikan sebagai modul yang memiliki kesamaan nama, namun behaviour (tingkah laku yang berbeda) sehingga listing kode implementasinya berbeda.
- d. *Inheritance* (Penurunan sifat) adalah proses pewarisan data dan method dari suatu class kepada class yang lain.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi data dan fungsi. Sehingga, objek dapat memiliki kelakuan yang berbeda, apabila diperbarui implementasi dari objek tersebut sesuai dengan karakteristiknya [15].

### 2.1.14 Basis Data

Basis data merupakan koleksi dari data yang terorganisasi dengan cara sedemikian rupa sehingga data tersebut mudah disimpan dan dimanipulasi. Disamping berisi atau menyimpan data, setiap basis data juga mengandung atau menyimpan definisi struktur [14].

**a) Database Management System (DBMS)**

*Database Management System* (DBMS) atau dalam Bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. Dalam penggunaan DBMS dibutuhkan komponen-komponen antara lain :

- (1) *Query processor*, komponen yang mengubah bentuk *query* dalam bentuk instruksi ke dalam *database manager*.
- (2) *Database manager*, menerima *query*, menguji eksternal dan konseptual untuk menentukan apakah *record-record* tersebut dibutuhkan untuk memenuhi permintaan kemudian hari dari *database manager* dengan memanggil *file manager* untuk menyelesaikan permintaan.
- (3) *File manager*, memanipulasi penyimpanan file dan mengatur alokasi ruang penyimpanan disk.
- (4) *Data manipulation language processor*, modul yang mengubah perintah DML yang ditempelkan ke dalam program aplikasi dalam bentuk fungsi-fungsi.
- (5) *Data definition language compiler*, mengubah statement DDL menjadi kumpulan tabel atau *file* yang berisi data *dictionary* atau meta data.
- (6) *Dictionary manager*, mengatur akses dan memelihara data *dictionary*.

Salah satu *software* yang tergolong ke dalam DBMS adalah *MySQL*. *MySQL* merupakan salah satu *Relational Database Management System* (RDBMS) yang saat ini sedang banyak diminati.

**b) Structural Query Language (SQL)**

*Structural Query Language* (SQL) adalah Bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. Secara umum SQL terdiri dari 2 (dua) bahasa yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML).

**(1) Data Definition Language (DDL)**

*Data Definition Language* (DDL) merupakan suatu perintah yang berfungsi untuk mendefinisikan atribut atribut basis data, tabel, atribut serta hubungan antar tabel. DDL berfungsi lebih ke dalam memanipulasi struktur *database*. DDL digunakan untuk membuat tabel atau menghapus tabel, membuat key atau indeks, membuat relasi antar tabel. Berikut sintaks yang ada di dalam DDL :

- (a) *Create*

Perintah *create* digunakan untuk membuat objek baru, baik berupa *database*, tabel, indeks atau prosedur yang tersimpan.

(b) *Alter*

Perintah *alter* digunakan untuk memodifikasi atau mengubah objek pada *database*, seperti *indeks*, dan lokasi.

(c) *Drop*

Perintah *drop* digunakan untuk menghilangkan atau menghapus objek pada *database*.

**(2) *Data Manipulation Language (DML)***

*Data Manipulation Language (DML)* merupakan kelompok perintah yang berfungsi untuk melakukan proses insert, update atau delete ke dalam suatu *database*. Berikut sintaks yang ada didalam DML :

(a) *Select*

Perintah *select* digunakan untuk mengambil atau menampilkan data dari suatu tabel atau beberapa tabel dalam *database*.

(b) *Insert*

Perintah *insert* untuk memasukan atau menambahkan data baru dalam *database*.

(c) *Update*

Perintah *update* digunakan untuk memperbarui data lama menjadi data terkini di dalam *database*.

(d) *Delete*

Perintah *delete* digunakan untuk menghapus data dari tabel dalam *database*.

