

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Landasan Teori**

##### **2.1.1. Sistem Informasi**

Sistem Informasi (SI) adalah sekumpulan komponen yang saling terikat untuk mengumpulkan atau mengambil, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan pengendalian dalam suatu organisasi [1]. Sistem informasi merupakan suatu sistem buatan manusia yang terdiri atas sekumpulan komponen berbasis manual dan komputer yang dibangun untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi kepada pengguna [2]. Sistem informasi dapat mempermudah pekerjaan manusia dalam mengolah data, menyimpan data, melakukan perhitungan, dan mendapatkan informasi yang diinginkan. Kualitas informasi ditentukan oleh hal-hal berikut [3]:

- a. Akurasi  
Akurasi menyatakan derajat kebenaran terhadap sebuah informasi. Informasi yang benar-benar bebas dari kesalahan dikatakan sangat akurat. Akurasi dapat ditingkatkan melalui ketelitian yang lebih tinggi dalam mengumpulkan dan memproses data.
- b. Relevan  
Relevan berarti informasi tersebut memiliki keterkaitan langsung dan memberikan manfaat bagi pemakainya.
- c. Waktu Penyajian Informasi  
Ketepatan waktu menyatakan usia data yang sesuai dengan upaya pengambilan keputusan. Artinya, informasi tersebut tidak usang atau kadaluwarsa ketika sampai ke penerima, sehingga masih ada waktu untuk menggunakan informasi tersebut sebagai bahan pengambilan Keputusan.

##### **2.1.2. Pemesanan**

Pemesanan merupakan proses permintaan suatu produk kepada pihak layanan jasa yang telah disediakan. Pemesanan merupakan suatu proses yang dilakukan pelanggan untuk memesan makanan atau minuman yang tersedia di daftar menu [4]. Pemesanan adalah proses, cara, atau perbuatan memesan atau memesankan. Prosedur pemesanan bisa dilakukan secara langsung maupun tidak langsung, definisi ini mengacu pada Kamus Besar Bahasa Indonesia (KBBI). Pemesanan umumnya terjadi ketika seseorang ingin memperoleh barang atau layanan dan perlu menghubungi pihak pemasok atau penjual untuk mendapatkan informasi lebih lanjut mengenai barang atau layanan tersebut.

##### **2.1.3. Katering**

Katering berasal dari kata *cater* yang artinya menyiapkan dan menyajikan makanan, biasanya katering dibutuhkan dalam sebuah acara [5]. Katering pada dasarnya merupakan usaha dibidang penyediaan jasa makanan berdasarkan permintaan pelanggan yang dikelola oleh sekelompok tim. Menurut Sjahmien Moehyi katering adalah jenis penyelenggaraan makanan yang tempat memasak makanan berbeda dengan tempat menghidangkan makanan. Makanan diangkut ke tempat lain untuk dihidangkan, misalnya ke tempat penyelenggaraan pesta, pertemuan, rapat, kantin atau kafetaria pada pusat industri [6].

##### **2.1.4. Website**

*Website* merupakan sebuah halaman atau Kumpulan beberapa halaman yang terangkum dalam sebuah *domain* atau *subdomain*, sebuah *website* berada didalam *World Wide Web* (WWW) dan diakses secara daring [7]. Sebuah *website* terdiri dari elemen seperti teks, gambar, suara, dan

animasi. Hal tersebut membuat *website* menjadi media informasi yang menarik untuk diakses dan dikunjungi. *Website* dapat diakses menggunakan alamat URL (*Uniform Resource Locator*) selama terkoneksi ke jaringan internet. *Website* terbagi menjadi 3 jenis, yaitu [8]:

- a. *Web Statis*  
*Web* statis merupakan *website* yang mempunyai halaman yang tidak berubah. Perubahan suatu halaman dilakukan secara manual dengan mengedit *code* yang menjadi struktur dari *website*.
- b. *Web Dinamis*  
*Web* Dinamis merupakan *website* yang secara terstruktur diperuntukan untuk diperbarui sesering mungkin. Biasanya disediakan halaman *backend* untuk melakukan perubahan konten dari *website* tersebut.
- c. *Web Interaktif*  
*Web* interaktif merupakan *web* yang berinteraksi antara penggunanya. Biasanya berupa forum diskusi maupun blog. Dimana adanya moderator sebagai pengatur alur diskusi.

### 2.1.5. Database

*Database* adalah sebuah aplikasi untuk mengelola data dan informasi secara terkomputerisasi dengan menyimpan, memproses, dan menyajikannya dalam satu atau lebih tabel, setiap tabel memiliki kolom dan fungsi penyimpanan masing-masing serta terdapat relasi antar tabel [2]. *Database* menyimpan data dengan menggunakan sistem tertentu yang saling berhubungan, sehingga memudahkan pengelolanya. Tanpa *database* data hanya tersimpan di dalam komputer dan tidak dapat diakses oleh pihak lain. *Database* merupakan tempat penyimpanan untuk menyimpan data, *database* dapat diumpamakan sebagai sebuah tempat penyimpanan data yang terstruktur agar dapat diakses dengan cepat dan mudah [9].

#### A. Query

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

DDL atau *Data Definition Language* merupakan sekumpulan perintah yang digunakan untuk menentukan skema dalam basis data. DDL memungkinkan pengguna untuk membuat, mengubah, menghapus objek-objek dalam basis data. Berikut adalah perintah-perintah yang dapat digunakan dalam DDL :

- (a) *Create*, perintah ini digunakan untuk membuat objek. Berikut adalah contoh penggunaannya :  
`CREATE TABLE pelanggan;`
- (b) *Alter*, perintah ini digunakan untuk mengubah struktur objek. Berikut adalah contoh penggunaannya :  
`ALTER TABLE pelanggan ADD Alamat VARCHAR(100);`
- (c) *Drop*, perintah ini digunakan untuk menghapus objek. Berikut adalah contoh penggunaannya :  
`DROP TABLE pelanggan;`

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


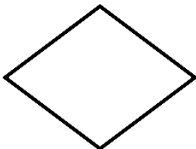
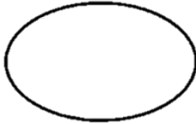
DML atau *Data Manipulation Language* merupakan Kumpulan perintah yang berkaitan dengan manipulasi data dalam basis data, biasanya meliputi sebagian besar pernyataan SQL yang digunakan untuk menambah, menghapus, mengubah, dan mengambil data dari tabel dalam basis data. Berikut adalah perintah-perintah yang dapat digunakan dalam DML :

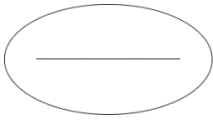

- (a) *Insert*, perintah ini digunakan untuk menambah data baru. Berikut adalah contoh penggunaannya :
- ```
INSERT INTO users ('id', 'username', 'password') VALUES ('1', 'ghinapertiwi', 'ghina123');
```
- (b) *Update*, perintah ini digunakan untuk mengubah data yang sudah ada. Berikut adalah contoh penggunaannya :
- ```
UPDATE users SET password = 'ghinapertiwi123' WHERE id = '1';
```
- (c) *DELETE*, perintah ini digunakan untuk menghapus data. Berikut adalah contoh penggunaannya :
- ```
DELETE FROM users WHERE id = '1';
```
- (d) *Select*, perintah ini digunakan untuk menampilkan atau mengambil data dari tabel. Berikut adalah contoh penggunaannya :
- ```
SELECT * FROM users;
```

### B. *Entity Relationship Diagram (ERD)*

ERD atau *Entity Relationship Diagram* merupakan diagram struktural yang digunakan dalam perancangan *database* untuk menggambarkan relasi antara entitas serta atribut-atributnya. Hubungan antara kedua tabel tersebut dapat diilustrasikan melalui relasi yang digambarkan dalam *Entity Relationship Diagram (ERD)* [2]. Tujuan dari ERD yaitu untuk merancang sebuah database dengan jelas dan menggambarkan data yang akan disimpan dalam sistem beserta batasannya [10]. Berikut adalah gambar dari simbol-simbol pada ERD:

**Tabel 2. 1** Simbol ERD

No	Nama	Simbol	Keterangan
1	<i>Entity</i>		<i>Entity</i> atau entitas merujuk pada objek atau konsep untuk menyimpan data. Dalam <i>database</i> entitas digunakan untuk mengacu pada tabel yang digunakan untuk menyimpan informasi terkait.
2	<i>Relationship</i>		<i>Relationship</i> atau relasi merupakan hubungan yang menggambarkan koneksi atau interaksi dalam <i>database</i> .
3	<i>Attribute</i>		<i>Attribute</i> merupakan karakteristik atau properti yang mendefinisikan entitas dalam <i>database</i> . <i>Attribute</i> membantu memperinci detail

			tentang data yang disimpan dalam sistem.
4	<i>Primary Key, Attribute</i>		<i>Primary Key</i> merupakan <i>attribute</i> yang digunakan untuk mendefinisikan nilai unik dalam sebuah entitas.
5	<i>Line / Connector</i>		<i>Line / Connector</i> merupakan garis yang menghubungkan <i>entity, attribute, dan relationship</i> antara <i>entity</i> dalam diagram.

### 2.1.6. PHP

PHP berasal dari kata *Hypertext Preprocessor* yaitu bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML [11]. PHP sebagai sekumpulan skrip atau bahasa pemrograman memiliki fungsi utama, yaitu mengumpulkan dan mengevaluasi hasil survey atau bentuk apa pun ke *server database*. PHP merupakan pemrograman *script server-side* yang didesain untuk pengembangan *web*. PHP disebut bahasa pemrograman *server-side* karena diproses pada komputer server [12]. Dalam hal ini PHP menawarkan konektifitas yang baik dengan beberapa basis data antara lain *Oracle, Sybase, MySQL, dan PostgreSQL* [13].

### 2.1.7. Laravel

Laravel adalah sebuah kerangka kerja *web* berbasis PHP yang *open-source* dan tidak berbayar, yang dikembangkan oleh Taylor Otwell dan ditujukan untuk pengembangan aplikasi *web* yang mengikuti pola MVC (*Model View Controller*) [14]. Laravel merupakan *framework* PHP yang memanfaatkan komponen kerja yang berbeda untuk membantu dalam pengembangan aplikasi *web* dengan desain *model-view-controller* (MVC). Laravel dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, sama seperti *framework-framework* yang lain, laravel dibangun dengan konsep MVC, kemudian laravel dilengkapi juga *command line tool* yang bernama “Artisan” yang bisa digunakan untuk *packaging bundle* dan *instalasi bundle* melalui *command prompt* [15]. Fungsi pada *framework* laravel menggabungkan fitur dasar dari kerangka PHP lainnya seperti Codeigniter dan Yii, serta bahasa pemrograman lain seperti Ruby on Rails. Struktur pola MVC pada laravel sedikit berbeda, dengan adanya routing yang menghubungkan permintaan dari pengguna ke *controller*.

### 2.1.8. Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek, disebut juga *Object Oriented Programming* merupakan pendekatan pemrograman yang fokus pada penggunaan objek. Pemrograman berorientasi objek adalah suatu cara untuk menghadapi masalah-masalah yang akan diatasi dengan bantuan komputer [16]. Dalam pemrograman ini, semua data dan fungsi terkait dikemas dalam kelas-kelas atau objek-objek. Setiap objek memiliki kemampuan untuk menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi informasi. Sekumpulan objek-objek ini saling berinteraksi untuk bertukar informasi [17]. Dengan demikian, PBO memberikan kerangka kerja

yang kuat untuk mengorganisir dan mengelola kode pemrograman. Dalam pemrograman berorientasi objek (PBO) memiliki karakteristik utama, yaitu:

- a. Kelas (*class*)  
Kelas adalah kategori yang mengelompokkan objek dengan karakteristik serupa. Kelas mendefinisikan atribut dan operasi objek serta memungkinkan pembuatan banyak objek dengan struktur dan perilaku serupa.
- b. Objek (*Object*)  
Objek adalah entitas dalam lingkungan yang memiliki *attribute* dan *method* yang dimiliki kelasnya. Objek juga memiliki kemampuan untuk melakukan tindakan dan bisa berinteraksi dengan objek lain dalam sistem yang lebih besar. Contoh dari objek adalah truk, bis, mobil. Objek merupakan entitas yang memiliki identitas (nama) dan biasanya menyimpan informasi tentang dirinya sendiri dan objek lain.
- c. Metode (*method*)  
Metode adalah fungsi atau prosedur yang didefinisikan oleh seorang programmer di dalam sebuah kelas. Dalam metode memiliki izin akses seperti *private*, *public*, dan *protected*. Sebuah kelas dapat memiliki beberapa *method* dengan nama yang sama, asalkan memiliki parameter masukan yang berbeda. Hal ini memungkinkan kompiler atau interpreter untuk mengenali *method* mana yang akan dipanggil.
- d. Atribut (*attribute*)  
Atribut merupakan informasi terkait objek. Atribut merupakan karakteristik atau property objek yang mempengaruhi perilaku dan keadaan objek. Dalam pemrograman atribut, atribut merupakan variable yang dideklarasikan di dalam kelas dan setiap objek memiliki Salinan atribut yang unik. Atribut juga memiliki hak akses seperti *method* yaitu *private*, *public*, dan *protected*.
- e. Enkapsulasi (*encapsulation*)  
Enkapsulasi merupakan proses membatasi ruang lingkup program terhadap data yang sedang diproses. Konsep pengkapsulan adalah menyembunyikan informasi. Data dan prosedur dikemas secara bersamaan dalam bungkus suatu objek sehingga data atau prosedur dari luar tidak dapat mengaksesnya.

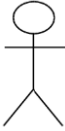


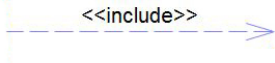
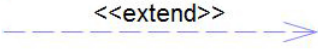
### 2.1.9. UML (*Unified Modeling Language*)

UML atau *Unified Modeling Language* merupakan salah satu bahasa yang umum digunakan dalam industri untuk merumuskan kebutuhan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman *object oriented* [18]. UML adalah bahasa untuk menspesifikasi, memvisualisasikan, serta mengonstruksi bangunan dasar sistem perangkat lunak, termasuk melibatkan pemodelan aturan-aturan bisnis [16]. UML digunakan sebagai sarana untuk memberikan pengetahuan tentang sistem aplikasi yang akan dibangun dari satu pengembang ke pengembang lain. UML juga berperan sebagai penghubung antara pengembang sistem dengan pengguna. Berikut adalah UML yang akan digunakan, yaitu :

#### A. *Use Case Diagram*

*Use Case* merupakan sarana untuk mengilustrasikan kebutuhan suatu sistem, yaitu sistem yang seharusnya digunakan. Komponen dari *use case* terdiri dari *actor*, *use case*, dan subjek (sistem) [19]. Berikut adalah simbol-simbol pada *use case* :


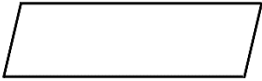

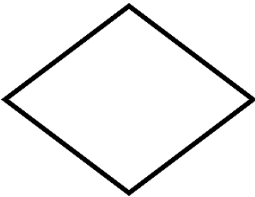
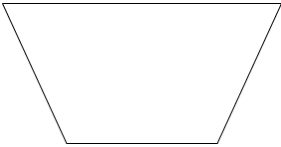

Tabel 2. 2 Simbol *Use Case*

No	Nama	Simbol	Keterangan
1	<i>Actor</i>		<i>Actor</i> adalah entitas luar atau pengguna yang berinteraksi dengan sistem. Dalam diagram <i>use case</i> , <i>actor</i> biasanya digambarkan sebagai manusia, perangkat lunak, atau sistem lain yang terlibat dalam skenario yang digambarkan dalam <i>use case</i> sistem yang akan dikembangkan.
2	<i>Use Case</i>		<i>Use case</i> merupakan aktivitas atau tindakan yang dilakukan oleh sistem. <i>Use case</i> menjelaskan interaksi antara pengguna dengan sistem dan biasanya <i>use case</i> digambarkan dalam bentuk <i>oval</i> atau <i>elips</i> .
3	Sistem		Sistem adalah entitas atau perangkat lunak yang akan dikembangkan. Sistem ini bertugas untuk menerima <i>input</i> dari <i>actor</i> dan memberikan <i>output</i> kembali kepada <i>actor</i> .
4	<i>Include</i>		<i>Include</i> merupakan hubungan antara dua <i>use case</i> yaitu suatu <i>use case</i> bagian dari <i>use case</i> yang lain. Contohnya yaitu, <i>login include register</i> .
5	<i>Extend</i>		<i>Extend</i> merupakan perluasan dari <i>use case</i> tergantung pada kondisi tertentu yang ditetapkan. <i>Extend</i> dilakukan untuk menambah fungsionalitas pada suatu <i>use case</i> .

## B. *Flowchart*

*Flowchart* adalah representasi dari langkah-langkah dan urutan prosedur dari suatu program, yang membantu analisis dan programmer dalam memecahkan masalah [20]. *Flowchart* digunakan untuk menggambarkan langkah-langkah suatu proses secara jelas dan mudah dipahami. Dalam diagram alir, setiap langkah direpresentasikan oleh masing-masing simbol yang berbeda. Berikut adalah simbol-simbol yang ada pada *flowchart* :

**Tabel 2. 3** Simbol *Flowchart*

No	Nama	Simbol	Keterangan
1	<i>Terminator</i>		<i>Terminator</i> digunakan untuk menyatakan awal atau akhir dari suatu proses.
2	<i>Input/Output</i>		<i>Input/output</i> untuk menyatakan sebuah proses <i>input</i> atau <i>output</i> dari suatu program yang berjalan.
3	<i>Process</i>		<i>Process</i> digunakan untuk menyatakan proses yang dilakukan sebuah program untuk menghasilkan <i>input</i> .
4	<i>Decision</i>		<i>Decision</i> digunakan untuk menyatakan suatu keputusan yang dibuat berdasarkan kondisi tertentu.
5	<i>Manual Operation</i>		<i>Manual Operation</i> digunakan untuk menggambarkan langkah-langkah yang tidak dilakukan oleh komputer.
6	<i>Flow</i>		<i>Flow</i> digunakan untuk menghubungkan antara satu simbol dengan simbol yang lainnya

			untuk menyatakan jalanya arus dari proses yang dibuat.
--	--	--	--