

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu:

**Tabel 2.1** Tinjauan Pustaka

No	Judul	Nama (Tahun)	Hasil
1	Perancangan Sistem Informasi Berbasis Mobile Pada Restoran Lokal di Bandar Lampung	(Agarin and Afnil, 2018)	Sistem yang dihasilkan dapat mengelola data pesanan makanan, booking makanan, mengelola meja makan dan kotak saran. Perbedaan pada penelitian yang dilakukan yaitu penerapan metode yang dilakukan peneliti menggunakan metode <i>extreme programming</i> .
2	Aplikasi Reservasi Menu Restoran Berbasis Web Dan Mobile Android Di Cowek Ireng	(Rasid, Supriyono and Setiawan, 2019)	Sistem yang dibangun meliputi pengelolaan data user seperti pimpinan, kasir, koki, dan pelanggan, pengelolaan pemesanan menu, pengelolaan reservasi menu dan tempat yang dilakukan pelanggan, pengelolaan pembayaran, sampai pada pelaporan penjualan ke pimpinan. Perbedaan pada penelitian yang dilakukan yaitu penerapan metode yang dilakukan peneliti menggunakan metode <i>extreme programming</i> .
3	Aplikasi Pengelolaan Transaksi dan Manajemen Menu di Cafe Yuri Berbasis Web Dan Mobile	(Hadrانيا nsyah <i>et al.</i> , 2020)	Aplikasi yang dihasilkan dapat digunakan untuk mengelola proses pemesanan makanan dari pelanggan ke dapur, mengelola bahan makanan, proses pembayaran, dan menghitung jumlah total dari semua industri. Perbedaan dengan penelitian yang dilakukan yaitu peneliti menambahkan fitur reservasi.

**Tabel 2.1** Tinjauan Pustaka

No	Judul	Nama (Tahun)	Hasil
4	Manajemen Sistem Informasi Restoran Berbasis Client Server	(Sutjahjo, 2021)	Hasil dari sistem yang dikembangkan yaitu Adanya transaksi penjualan menu, pengembalian uang, serta pembuatan laporan yang lebih teliti, cepat dan akurat serta memberikan kemudahan dalam pengolahan data. Perbedaan dengan penelitian yang dilakukan yaitu peneliti menambahkan fitur reservasi.
5	Perancangan Sistem Informasi Manajemen Restoran Pada Kedai In Aja	(Permadi and Mirza, 2022)	Hasil dari pembuatan aplikasi sistem manajemen restoran untuk Kedai in Aja adalah pelanggan dapat melakukan pemesanan secara mandiri melalui smartphone mereka masing-masing. Pemesanan mencakup pemilihan menu, meja, serta pemilihan metode pembayaran. Perbedaan dengan penelitian yang dilakukan yaitu peneliti menambahkan fitur reservasi.

## 2.2 Reservasi

Menurut Sambodo (2020), reservasi adalah mengumpulkan informasi dari calon tamu, agen perjalanan, grup atau penyelenggara konferensi dan pihak lain yang mungkin memesan akomodasi. Peran petugas reservasi yaitu untuk mendapatkan informasi yang diperlukan untuk melakukan pemesanan, mencatat informasi calon tamu ke dalam catatan reservasi manual atau terkomputerisasi, memantau tingkat pemesanan, mengeluarkan segala dokumentasi kepada calon tamu untuk mengkonfirmasi pemesanan (Seperti surat konfirmasi, permintaan jaminan uang muka/Deposito) dan memastikan detail reservasi dikonfirmasi dan tanggal kedatangan yang diharapkan. Sehingga reservasi dapat diartikan sebagai bentuk transaksi pemesanan suatu fasilitas maupun tempat yang dibutuhkan bagi penggunanya.

### **2.3 Restoran**

Menurut Wicaksono and Hamsir (2019), restoran adalah setiap tempat usaha komersial yang ruang lingkup kegiatannya menyediakan hidangan dan minuman untuk umum. Setiap rumah makan atau restoran harus memiliki seseorang yang bertindak sebagai pemimpin restoran yang sehari-hari mengelola dan bertanggung jawab atas pengusahaan restoran atau rumah makan tersebut.

Restoran merupakan usaha penyediaan makanan dan minuman dilengkapi dengan pembuatan, peralatan dan perlengkapan untuk proses penyimpanan dan penyajian, didalam satu tempat tetap yang tidak berpindah-pindah.

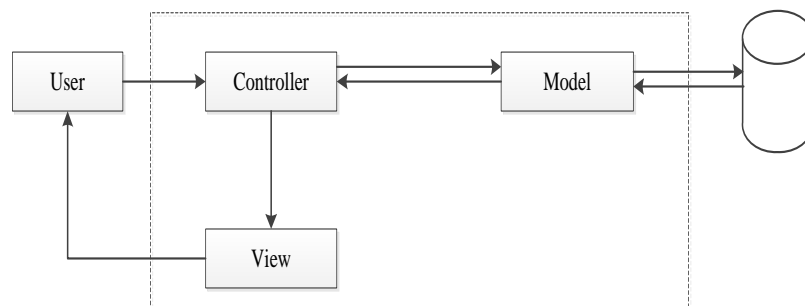
### **2.4 Web Based**

Menurut Urbietta *et al.* (2019), *Web Based* adalah aplikasi yang dibuat berbasis *web* yang membutuhkan *web server* dan *browser* untuk menjalankannya. Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya:

1. Tidak membutuhkan *hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.
2. Server yang dibutuhkan cukup diinstallkan *tools* pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).
4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

### 2.4.1 CodeIgniter

Menurut Raharjo (2018), CodeIgniter adalah Framework untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. CodeIgniter memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat dikembangkan dalam perangkat web, dekstop maupun mobile. *CodeIgniter* memiliki konsep atau pola Model-View-Controller (MVC) sehingga kode-kode dapat di sederhanakan.



**Gambar 2.1** Arsitektur MVC

### 2.4.2 Hypertext Preprocessor (PHP)

Menurut Subagja (2018), PHP adalah bahasa *server-side-scripting* yang menyatudengan HTML untuk membuat halaman *web* yang dinamis.

Menurut Oetomo and Maharginono (2020), berpendapat bahwa *PHP Hypertext Preprocessor* adalah bahasa pemograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server (server side HTML embedded scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang

terbaru/*up to date*. Semua script PHP dieksekusi pada *server* dimana script tersebut dijalankan. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis.

Sehingga PHP merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa HTML dan *script* lainnya.

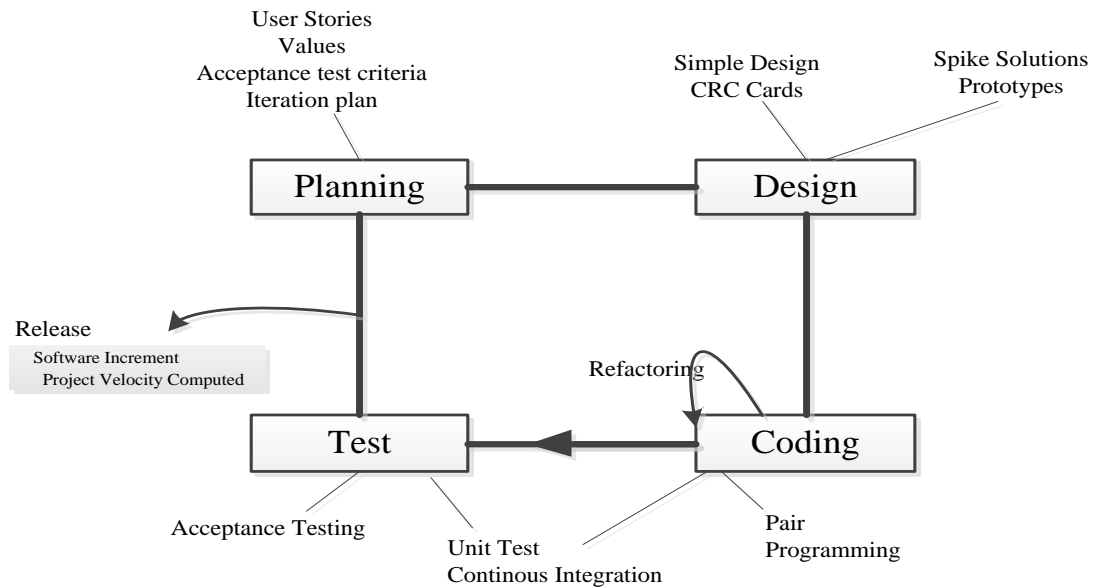
### **2.4.3 MySQL**

Menurut MySQL (2018), *MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data. Menurut Amin (2018) mendefinisikan *mysql* adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

*MySQL* merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

### **2.5 Metode *Extreme Programming***

Menurut Suryantara (2017) *extreme programming* berdasarkan sejarah singkat bahwa pengembangan perangkat lunak banyak digunakan untuk pengembangan yang lebih cepat dengan meliputi tahapan *planning*, *design*, *coding* dan *testing*. Berikut merupakan konsep *Extreme programming*.



**Gambar 2.2** *Extreme Programming*  
 Sumber: (Suryantara, 2017)

### 2.5.1 Kerangka Kerja *Extreme Programming*

Pengembangan yang dilakukan menggunakan XP dengan proses yang lebih cepat dengan tahapan seperti *planning*, *design*, *coding* dan *testing*.

#### 1. *Planning*/Perencanaan

Tahap ini dimulain dengan pemahaman konteks bisnis dari aplikasi dengan mendefinisikan keluaran seperti fitur, fungsi, penentuan waktu dan biaya serta alur pengembangan.

#### 2. *Design*/Perancangan

Tahap perencanaan secara sederhana dengan alat mendesain kartu CRC (*Class Responsibility Collaborator*) yang digunakan untuk pemetaan kelas-kelas yang akan diguanakna pada diagram UML.

#### 3. *Coding*/Pengkodean

Hal utama dalam pengembangan menggunakan XP yaitu *pari programming* (Proses pembuatan program melibatkan 2 atau lebih programmer).

#### 4. *Testing/Pengujian*

Tahap ini fokus pada pengujian fitur pada aplikasi sehingga tidak ada kesalahan dan sesuai dengan proses bisnisnya.

#### **2.5.2 Keuntungan dan Kerugian *Extreme Programming***

Keuntungan pada penerapan metode XP yaitu:

1. Dalam hal XP menjalin komunikasi yang baik dengan klien pada pengembangan aplikasi
2. Saling menghargai antar developer dan meningkatkan komunikasi
3. Dapat menjadi pembelajaran bagi orang lain
4. Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat
5. Dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat
6. Developer tidak berkerja secara berlebihan
7. Dengan XP dapat membuat keputusan yang bersifat teknis

#### **2.5.3 *Class Responsibility Collaboration (CRC) Card***

Menurut Rosa dan Shalahuddin (2019) *Class Responsibility Collaboration (CRC) Card* merupakan tools yang digunakan untuk mendefinisikan behavior dan responsibility dari masingmasing class dan hubungan kolaborasi antara *class-class* tersebut. *CRC Card* dikembangkan dengan interaksi antara analis dan pengguna. Masing-masing orang diminta untuk mendeskripsikan logika yang diperlukan untuk memenuhi suatu *responsibility*, dan informasi apa saja yang dibutuhkan tetapi tidak dimilikinya. *Class* lain yang memiliki informasi yang dibutuhkan akan menjadi *collaborator* untuk *behavior* tersebut.


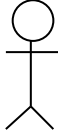

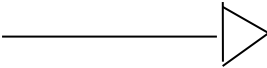
## 2.6 Unified Modelling Language (UML)

Menurut Rosa dan Shalahuddin (2019) *unified Modelling Language* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML.

### 2.6.1 Use Case Diagram

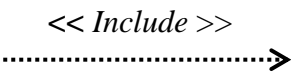
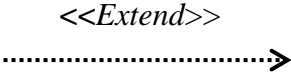
Menurut Rosa dan Shalahuddin (2019) *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2.2** Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Use case</i> Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi ( <i>generalization</i> ) merupakan hubungan (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum



**Tabel 2.2** Simbol *Use Case Diagram*



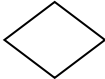

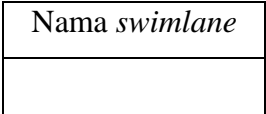

No	Simbol	Deskripsi
5.		Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan.
6.		Ekstensi ( <i>extend</i> ) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.

Sumber : (Rosa dan Shalahuddin, 2019)

### 2.6.2 Activity Diagram

Menurut Rosa dan Shalahuddin (2019) *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.3:

**Tabel 2.3** Simbol *Activity Diagram*

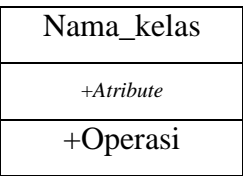


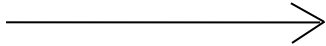
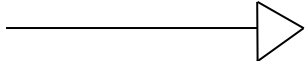
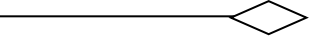
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan ( <i>Decision</i> ) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan ( <i>Join</i> ) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber : (Rosa dan Shalahuddin, 2019)

### 2.6.3 Class Diagram

Menurut Rosa dan Shalahuddin (2019) *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.4.

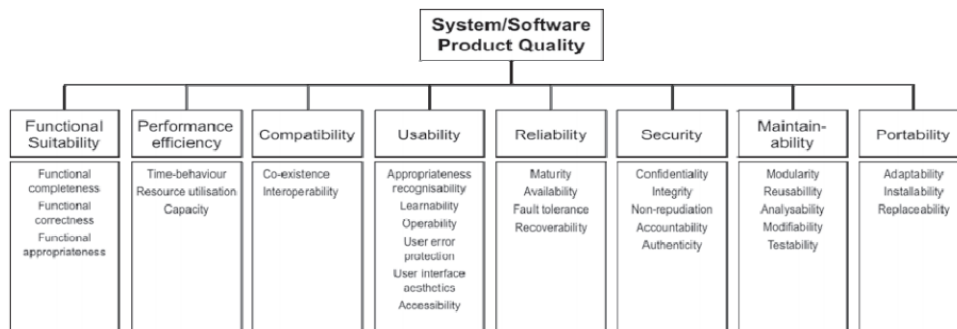
**Tabel 2.4** Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/<i>Interface</i></p>  <p>Nama_<i>Interface</i></p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / <i>Asociation</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / <i>Directed Association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan maksna semua bagian ( <i>whole-part</i> )

**Sumber:** (Rosa dan Shalahuddin, 2019)

## 2.7 Pengujian ISO 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang *software engineering* (*International Organisation for Standardisation*, 2011), dapat dilihat pada gambar berikut:



**Gambar 2.3** ISO 25010

Model kualitas produk terdiri dari delapan karakteristik yang berhubungan dengan sifat statis perangkat lunak dan sifat dinamis dari sistem komputer. Berdasarkan tahapan-tahapan ISO 25010 tersebut maka peneliti menggunakan pengujian terhadap kualitas perangkat lunak berupa aplikasi *web* dapat dinilai dari empat aspek yaitu:

### 1. *Functional Suitability*

Sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan di bawah kondisi tertentu. Sub pengujian tersebut sebagai berikut:

#### a. *Functional Completeness* (Kelengkapan Fungsional)

Sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan

b. *Functional Correctness* (Kebenaran Fungsional)

Sejauh mana suatu produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang dibutuhkan.

Alat pengujian yang digunakan yaitu kuisioner yang memiliki 2 jawaban pertanyaan seperti sukses atau gagal.

2. *Usability*

Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi, dan kepuasan dalam konteks penggunaan tertentu. Sub pengujian tersebut sebagai berikut :

a. *Appropriateness Recognizability*

Sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.

b. *Learnability*

Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran yang ditentukan untuk menggunakan produk atau sistem dengan efektifitas, efisiensi, bebas dari resiko dan kepuasan dengan tujuan tertentu konteks penggunaan.

Alat pengujian yang digunakan yaitu kuisioner yang memiliki 5 jawaban pertanyaan seperti Sangat Setuju, Setuju, Ragu-ragu, Tidak Setuju dan Sangat Tidak Setuju.

### 3. *Portability*

Tingkat efektivitas dan efisiensi dimana sistem, produk atau komponen dapat ditransfer dari satu perangkat keras, perangkat lunak atau lingkungan operasional atau penggunaan lainnya ke yang lain. Berikut sub pengujian tersebut:

#### a. *Adaptability*

Sejauh mana suatu produk atau sistem dapat secara efektif dan efisien diadaptasi untuk perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan yang berbeda atau berkembang.

#### b. *Coexistence*

Sejauh mana sistem dapat bekerja pada perangkat lunak yang ada.

Alat yang digunakan untuk pengujian tersebut berupa *cross browsing compability testing* seperti *chrome, firefox* dan *opera*.

### 4. *Performance eficiency.*

kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi yang dinyatakan, berikut sub pengujian tersebut:

#### a. *Time Behaviour*

Sejauh mana respon dan waktu pemrosesan dan tingkat throughput dari suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan

#### b. *Resource Utilization*

Sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan.

Alat pengujian yang digunakan yaitu dengan melakukan pengecekan performa sistem pada media web test secara online sehingga mampu menampilkan detail performa sistem, adapun link web test yaitu <https://www.webpagetest.org/>