

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam, penelitian ini, menggunakan beberapa penelitian sebelumnya untuk mendukung penelitian yang sedang dilakukan. Berikut merupakan penelitian sebelumnya yang dapat dijadikan tinjauan pustaka pada penelitian ini, dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Tinjauan Pustaka

No	Peneliti, Tahun	Judul	Metode	Hasil
1	(Mamay Syani and Nindi Werstantia, 2018)	Perancangan Aplikasi Pemesanan Catering Berbasis Mobile Android	<i>Extreme Programming</i>	Aplikasi yang dihasilkan memberikan kemudahan dalam melakukan proses pemesanan dan pengelolaannya sesuai dengan kebutuhan penyedia catering.
2	(Pratiwi, Suhartanto dan	Sistem Informasi Pemesanan Produk Catering <i>Weding Attire</i> Berbasis Web	<i>Extreme Programming</i>	Aplikasi yang dihasilkan memfasilitasi proses

	Firmansyah, 2021)	Pada AE.ESTHETIC.ID		penjadwalan untuk memaksimalkan jalinan hubungan konsumen melalui pemasaran dan pelayanan order jasa <i>Wedding Attire</i>
3	(Saing, Alam dan Ika, 2021)	Perancangan Aplikasi E-Catering Pada Usaha Rabila Catering Berbasis Web Menggunakan Notifikasi WhatsApp Gateway	<i>Extreme Programming</i>	Aplikasi yang dihasil mampu mengoptimalkan sistem sebelumnya dimana promosi dan pemesanan bisa menggunakan berbasis web dan whatsapp gateway.
4	(Koswoyo and Mauludin, 2021)	Sistem Informasi Pemesanan Katering Berbasis Web di CV Deqiko I.D.E Cemerlang	<i>Waterfall</i>	Mempermudah pelanggan dalam mencari informasi mengenai menu- menu yang

				tersedia pada penyedia catering. Selain itu, pada sistem ini pelanggan dapat memesan sesuai keinginannya sendiri makanan yang diinginkan.
5	(Bagaskoro, 2019)	Rancang Bangun Sistem Informasi Pemesanan Catering Berbasis Website Pada Rahayu Catering Madiun	<i>Waterfall</i>	Mempermudah proses pemesanan dan pembayaran catering secara online.

Penelitian diatas adalah dasar-dasar penelitian sebelumnya yang dapat dijadikan tinjauan pustaka pada penelitian ini. Berikut merupakan penjelasan tinjauan pustaka diantaranya:

Dalam penelitian yang dilakukan oleh (Mamay Syani dan Nindi Werstantia, 2018) yang menerapkan *Extreme Programming* pada perancangan aplikasi pemesanan catering berbasis *mobile android*. Berawal dari proses pemesanan yang masih manual yaitu dengan datang langsung ke kantor atau menelepon pihak Cimahi *Catering* hingga adanya pelanggan yang mengeluhkan proses pemesanan *catering* tersebut dikarenakan memakan waktu, tenaga dan biaya lebih. Selain itu, dalam pencatatan data juga masih bersifat konvensional sehingga dirasa kurang efektif. Hal ini beresiko terjadinya kesalahan pendataan. Sebagai solusi dari permasalahan tersebut dibangunlah sebuah aplikasi pemesanan catering berbasis *mobile android* untuk membantu dalam proses pemesanan catering oleh pelanggan serta pengelolaan data pemesanan catering oleh

pihak Cimahi Catering. Pada pembuatan aplikasi ini menggunakan basis pemrograman android yaitu Java dan *MySQL* sebagai tempat penyimpanan datanya. Aplikasi ini dibuat untuk mengefisiensikan waktu, tenaga dan menghasilkan informasi yang akurat.

Dalam penelitian yang dilakukan oleh (Pratiwi, Suhartanto dan Firmansyah, 2021) yang menerapkan metode *Extreme Programming* pada pembuatan sistem informasi pemesanan produk *wedding attire* berbasis web pada AE.STHETIC.ID. AE.STHETIC.ID merupakan salah satu jasa penyelenggara persiapan pernikahan (*wedding*) yang beralamatkan di Tanjung Karang Pusat, Bandar Lampung. Jasa yang ditawarkan oleh AE.STHETIC.ID salah satunya adalah *wedding attire*. Berdasarkan hasil wawancara yang dilakukan ditemukan masalah yaitu pada proses pemesanan yang dilakukan masih secara manual yaitu dengan mencatat ke dalam nota atau buku besar dan pelanggan harus datang langsung ke tempat jasa wedding atau melalui telepon untuk melakukan pemesanan. Selain itu, dari segi pemasaran pun masih menggunakan *banner* atau melalui media sosial seperti *Instagram* atau *Facebook*. Untuk mengatasi masalah tersebut, dibuatlah sebuah sistem informasi pemesanan *wedding attire* untuk mempromosikan produk/jasa sehingga masyarakat yang berminat tidak mengalami kesulitan dalam mendapatkan informasi mengenai jasa *wedding attire* yang ditawarkan oleh AE.STHETIC.ID. Dengan sistem informasi ini pelanggan dapat langsung melakukan proses pemesanan jasa *wedding attire* secara online sehingga dengan begitu diharapkan dapat menghemat waktu, tenaga dan biaya. Sistem informasi ini dibuat menggunakan Bahasa pemrograman PHP dan *MySQL* sebagai media penyimpanan data atau *database*.

Dalam penelitian yang dilakukan oleh (Saing, Alam dan Ika, 2021) yang mana menerapkan metode *Extreme Programming* pada perancangan aplikasi *e-catering* pada usaha Rabila Catering menggunakan notifikasi *WhatsApp Gateway*. Rabila Catering merupakan salah satu catering yang ada di Kota Pinrang sebagai penyedia makanan dan jasa untuk sebuah acara atau pesta. Berawal dari proses transaksi yang terjadi antara pemilik catering dengan pelanggannya masih bersifat konvensional yaitu dengan datang langsung ke kantor ataupun telepon. Selain itu kebebasan pelanggan dalam memilih menu masih minim dan pelanggan baru sulit mengetahui keberadaan letak

tempat Rabila Catering. Sebagai solusi dari permasalahan tersebut, dibuatlah sebuah aplikasi *e-catering* yang harapannya dapat mengoptimalkan sistem sebelumnya dimana promosi dan pemesanan bisa menggunakan berbasis web dan *whatsapp gateway*. Dengan aplikasi ini, diharapkan dapat mempermudah pelanggan dalam mencari informasi mengenai menu, paket, promo dan pemesanan sehingga pelanggan tidak perlu lagi datang ke kantor untuk bisa melakukan pemesanan catering. Aplikasi dibuat menggunakan Bahasa pemrograman VB.NET dan ASP.Net dan *Microsoft SQL Server* untuk pengelolaan database.

Penelitian yang dilakukan oleh (Koswoyo dan Mauludin, 2021) membuat sebuah sistem informasi pemesanan catering berbasis web. Penelitian ini menerapkan metode *Waterfall* dalam pengembangan sistem informasi. Beberapa masalah yang diangkat pada penelitian ini adalah proses penghitungan jumlah pembayaran pesanan yang masih manual. Tentunya hal ini sangat merepotkan pegawai apalagi pada saat permintaan pemesanan sedang banyak-banyaknya. Selain itu, pada Resto Deqiko belum adanya fasilitas bagaimana menangani pemesanan dari calon pelanggan yang ingin mengetahui secara detail tentang layanan yang tersedia seperti menu, harga, dan lain-lain. Dari sistem yang telah dibuat dapat dihasilkan sebuah sistem informasi yang memudahkan calon pelanggan dalam mencari informasi mengenai menu-menu yang ditawarkan pada Resto Deqiko untuk kebutuhan pelanggan. Selain itu, memungkinkan pelanggan untuk memilih menu sesuai keinginannya sehingga proses pemesanan menjadi lebih fleksibel.

Penelitian yang dilakukan oleh (Bagaskoro, 2022) membuat sebuah sistem informasi pemesanan catering berbasis website. Pada penelitian ini menggunakan metode *waterfall* dalam pengembangan sistem informasi. Beberapa masalah yang diangkat dalam penelitian ini adalah kegiatan pendataan pesanan masih dengan cara menulis di buku catatan, keadaan ini meningkatkan resiko seperti kesalahan dalam jumlah orderan, kesalahan pengiriman dan kehilangan data pesanan yang masuk, selain itu belum adanya media promosi tentang layanan catering untuk menjangkau pasar yang lebih luas. Dengan adanya sistem informasi yang dirancang ini dapat mempermudah pelanggan dalam mencari informasi mengenai layanan yang ada pada

penyedia catering tempat penelitian dan juga memungkinkan pemesanan dilakukan secara online.

2.2 Keaslian Peneliti

Adapun beberapa hal yang menjadi pembeda antara penelitian yang dilakukan penulis dengan penelitian yang telah dilakukan sebelumnya, sebagaimana terlampir pada tabel tinjauan pustaka, diantaranya adalah:

1. Pendekatan yang digunakan oleh penulis dalam mengembangkan sistem menggunakan metode *Extreme Programming* (XP).
2. Pengembangan sistem penulis menggunakan arsitektur *client-server*, yakni dengan memisahkan antara *Backend* dan *Frontend*.
3. Pengembangan REST API menggunakan *Framework* Laravel.
4. Tahapan pengembangan *User Interface* menggunakan React JS.

2.3 Sistem Informasi

Sistem informasi merupakan suatu asosiasi terdiri dari beberapa modul yang saling terintegrasi dimana menyajikan informasi informasi dan pengolahan data untuk disajikan sesuai dengan kebutuhan *user*, *database*, *source code*, dan *model design* mewakili dari untaian suatu rancangan sistem informasi berguna untuk mempermudah dalam pengembangan dan maintenance (Hidayat and Muttaqin, 2018).

Sistem informasi adalah kumpulan sumber daya yang sama halnya manusia dan peralatan yang diatur untuk mengubah data menjadi informasi yang berguna bagi organisasi untuk mencapai tujuan. Informasi ini dikomunikasikan kepada beragam pengambilan keputusan yang dibutuhkan apakah penggunaan secara manual atau terkomputerisasi (Trianasari, 2022).

2.4 Catering

Catering adalah suatu usaha di bidang jasa dalam hal menyediakan / melayani permintaan makanan, untuk berbagai macam keperluan (Baso, Rindengan dan Sengkey, 2020). Dalam penyelenggaraan acara, catering dapat membantu mengurangi

beban pemilik acara dalam menyiapkan makanan dan minuman, sehingga pemilik acara dapat lebih fokus mengatur kebutuhan acara lainnya. Banyak layanan yang disediakan oleh usaha catering seperti layanan prasmanan dan nasi kotak yang dapat memberikan layanan kepada langganan dalam memenuhi kebutuhan makanan dan minuman untuk acara seperti pernikahan, aqiqah, dan lain-lain. Tidak hanya menyediakan makanan dan minuman, catering juga menyediakan peralatan-peralatan yang biasanya digunakan untuk mendukung keberlangsungan acara seperti halnya dekorasi yang membuat acara menjadi lebih rapih dan menarik.

2.5 Metode Agile

Agile adalah sebuah metode manajemen proyek yang menggunakan siklus pengembangan yang singkat, atau bisa disebut juga “*sprint*” untuk focus pada peningkatan berkelanjutan dalam pengembangan suatu produk atau layanan (Lutfiani *et al.*, 2020). Metode Agile menekankan pada pengiriman produk dalam waktu singkat dan terus menerus melakukan perbaikan berdasarkan umpan balik dari pengguna. Penerapan metode ini juga menekankan pada pengerjaan proyek dalam waktu yang singkat dan sederhana, namun tetap menghasilkan produk yang berkualitas tinggi. Berikut beberapa metode Agile yang berkembang saat ini:

1. Extreme Programming (XP)
2. Adaptive Software Development (ASD).
3. SCRUM
4. Dynamic Systems Development Method (DSDM)
5. Feature Driven Development (FDD)
6. Crystal Light Methodology Family
7. Pragmatic Programming
8. Open Source Software Development (OSSD).

2.6 Metode Extreme Programming (XP)

Extreme Programming (XP) merupakan suatu pendekatan atau model pengembangan perangkat lunak yang bertujuan untuk menyederhanakan langkah-

langkah dalam proses pengembangan tersebut agar menjadi lebih adaptif dan fleksibel. *Extreme Programming* adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan (Anita *et al.*, 2020). Berikut ini merupakan nilai-nilai utama dari metode *Extreme Programming*:

1. Komunikasi

Metode *Extreme Programming* mendorong komunikasi yang intensif dan efektif antara semua anggota tim pengembangan perangkat lunak. Komunikasi yang baik membantu memastikan pemahaman yang jelas tentang persyaratan proyek, perubahan yang diperlukan, dan progress yang dicapai.

2. Keberanian

XP menekankan pentingnya keberanian dalam menghadapi tantangan dan masalah, serta mengambil keputusan yang tepat untuk mencapai tujuan.

3. Kesederhanaan

XP mempromosikan desain dan implementasi perangkat lunak yang sederhana dan mudah dipahami, sehingga perangkat lunak dapat dikembangkan sesuai dengan kebutuhan dan harapan.

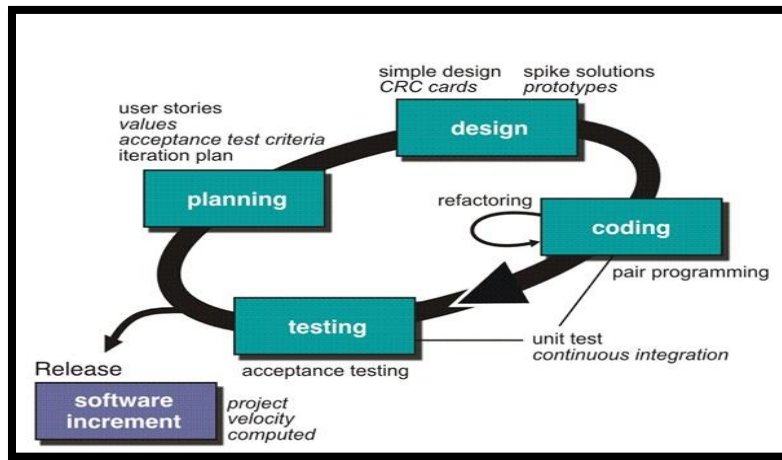
4. Umpan Balik

XP mempromosikan umpan balik yang cepat dan terus-menerus dalam setiap tahap pengembangan, baik dari pengembang maupun pelanggan, sehingga perangkat lunak dapat dikembangkan sesuai dengan kebutuhan dan harapan.

5. Respek

XP mendorong adanya rasa saling menghargai dan percaya di antara pengembang dan pelanggan, sehingga memungkinkan terjadinya kolaborasi yang produktif dan positif.

2.7 Tahapan Metode *Extreme Programming*



Gambar 2. 1 Tahapan Penelitian

Sumber: (Fatoni dan Dwi, 2016)

Dalam pengembangan perangkat lunak menggunakan metode XP, ada beberapa tahapan yang harus dilalui pengembang yaitu:

1. Perencanaan

Tahap perencanaan pada XP melibatkan pengembang dan pelanggan dalam merencanakan proyek dan menentukan tujuan proyek. Beberapa aktivitas yang dilakukan dalam tahap ini antara lain:

a. Menentukan *User Stories*

Pengembang dan pelanggan membuat daftar cerita pengguna (*user stories*) yang mewakili kebutuhan pengguna. Setiap cerita pengguna menjelaskan fitur atau fungsionalitas yang diinginkan oleh pengguna.

b. Menentukan Values

Setelah *user story* ditentukan, pengembang akan menentukan prioritas untuk setiap *user story* yang disebut *values*. *Values* ditentukan berdasarkan nilai bisnis yang dihasilkan, kompleksitas pengembangan, dan kebutuhan pengguna.

c. Menentukan *Acceptance Test Criteria*

Dari hasil penentuan *values* tersebut maka pengembang akan menentukan *acceptance test criteria* yaitu menentukan kriteria apa saja yang nantinya sebagai acuan terhadap sistem yang akan diuji.

d. *Membuat Iteration Plan*

Pada tahap ini, pengembang akan membuat rencana untuk setiap iterasi. Rencana harus mencakup daftar tugas yang harus dilakukan, perkiraan waktu yang dibutuhkan untuk menyelesaikan setiap tugas, dan risiko yang mungkin terjadi selama iterasi.

2. Perancangan

Tahap perancangan pada XP dilakukan dengan pendekatan desain sederhana (*simple design*). Fokus utama pada tahap ini adalah untuk membuat desain perangkat lunak yang mudah dipahami dan tidak kompleks, sehingga memudahkan perubahan dan pengembangan lebih lanjut. Beberapa aktivitas yang dilakukan dalam tahap ini antara lain:

a. *Simple Design*

Pengembang membangun perangkat lunak dengan memulai dengan desain yang sederhana. Ini dilakukan dengan menggunakan diagram UML seperti diagram use case, *sequence* diagram, diagram aktivitas, dan diagram kelas.

b. *CRC Cards*

CRC (Class Responsibility Collaborator) Cards adalah salah satu teknik yang digunakan dalam pengembangan perangkat lunak berbasis objek untuk membantu mengidentifikasi kelas-kelas dalam sebuah sistem perangkat lunak, serta tanggung jawab dan kolaborasi kelas-kelas dalam memenuhi kebutuhan pengguna.

c. *Prototype*

Bagian perancangan berupa *user interface* dalam bentuk *wireframing* untuk mempermudah pengguna melihat bagaimana sistem bekerja.

3. Pengkodean

Tahap pengkodean pada XP dilakukan dengan pendekatan *pair programming*, yaitu dua pengembang bekerja bersama-sama untuk menulis kode program. Beberapa aktivitas yang dilakukan dalam tahap ini antara lain:

a. Menulis kode program

Pengembang menulis kode program berdasarkan rancangan desain yang telah dibuat.

b. Memastikan konsistensi kode program

Setiap kode program yang ditulis harus konsisten dengan standar yang telah ditetapkan.

c. Melakukan *refactoring*

Pengembang melakukan *refactoring* secara terus menerus untuk memastikan bahwa kode program selalu bersih dan mudah dipelihara.

4. Pengujian

Fase pengujian adalah ketika pengguna mencoba sistem untuk memastikannya berfungsi sesuai yang diharapkan. Ini dilakukan dengan melakukan *acceptance test* yang telah disetujui oleh pengembang. Tes ini berfokus pada keseluruhan fitur dan fungsionalitas sistem. Setelah pengguna yakin sistem bekerja seperti yang diharapkan, tahap pengujian selesai.

2.8 UML (Unified Modeling Language)

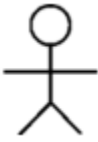


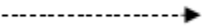
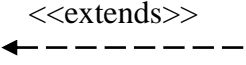
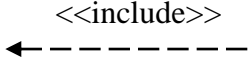
UML adalah bahasa yang telah menjadi standar industri untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak. UML menyediakan standar untuk merancang model sistem. UML juga memudahkan pengembangan perangkat lunak sehingga mengetahui alur yang diharapkan (Aulia *et al.*, 2021). Adapun beberapa diagram pada UML adalah sebagai berikut:

2.8.1 Use Case Diagram

Use Case Diagram adalah jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan fungsionalitas sistem dari perspektif pengguna atau operator. *Use Case Diagram* membantu mengidentifikasi dan memahami

persyaratan fungsional dari suatu sistem dan mengkomunikasikan fungsionalitas tersebut kepada pemangku kepentingan yang relevan. sebuah use case dapat mencakup fungsionalitas use case lain sebagai bagian dari proses internal (Dharwiyanti dan Wahono, 2003). Simbol yang digunakan dalam *use case diagram* dapat dilihat pada tabel 2.2.

Tabel 2. 2 Use Case Diagram

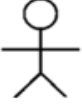


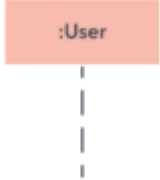



No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran orang, sistem, atau alat dalam berkomunikasi dengan use case
2		Use Case	Abstraksi dan interaksi antara sistem dan aktor
3		Asosiasi	Abstraksi hubungan antara subjek dan use case
4		Generalisasi	Menunjukkan kemampuan aktor untuk berpartisipasi dalam <i>use case</i>
5		Extends	Menunjukkan bahwa <i>use case</i> merupakan tambahan fungsional ke <i>use case</i> lain jika kondisinya benar
6		<i>Include</i>	Menunjukkan bahwa satu <i>use case</i> sepenuhnya bergantung pada <i>use case</i> lain


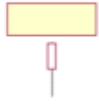
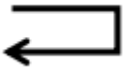
2.8.2 Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu (Dharwiyanti dan Wahono, 2003). Untuk

menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel 2.3 berikut ini.

Tabel 2. 3 *Sequence Diagram*


No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran orang, sistem, atau alat dalam berinteraksi di dalam dan di luar
2		<i>Activation Box</i>	Merepresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugas tertentu
3		Objek	Mendemonstrasikan bagaimana sebuah objek akan berperilaku dalam sistem
4		<i>Lifelines</i>	Menunjukkan kejadian berurutan yang terjadi pada sebuah objek
5		<i>Synchronous Message</i>	Menunjukkan bahwa pengirim harus menunggu respon pesan sebelum melanjutkan aktivitas
6		<i>Asynchronous Message</i>	Menunjukkan bahwa pengirim tidak memerlukan respon pesan sebelum melanjutkan aktivitas
7		<i>Reply Message</i>	Menunjukkan balasan untuk sebuah panggilan tertentu


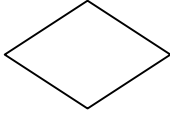


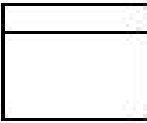
8		<i>Delete Message</i>	Menghapus sebuah objek
9		General	Merepresentasikan entitas tunggal berupa kelas
10		<i>Message Return</i>	Penggambaran hasil dari pengiriman pesan

2.8.3 Activity Diagram

Activity Diagram merupakan salah satu jenis diagram pada UML yang digunakan untuk menggambarkan aliran atau urutan aktivitas atau tindakan yang terjadi dalam sebuah proses bisnis atau sistem. Diagram ini dapat membantu dalam memahami interaksi antara proses, orang, dan sistem. *Activity diagram* dapat digunakan untuk memperjelas alur kerja dalam sistem dan mengidentifikasi aktivitas-aktivitas kritis yang perlu diperhatikan. Activity Diagram juga menggambarkan tahap paralel yang mungkin terjadi pada beberapa eksekusi (Dharwiyanti dan Wahono, 2003). Adapun beberapa simbol yang terdapat pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini.

Tabel 2. 4 *Activity Diagram*


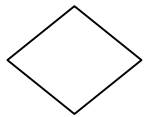
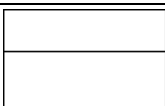

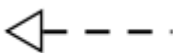
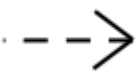
No	Simbol	Nama	Keterangan
1		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

2		Status Awal	Sebuah diagram aktivitas memiliki status awal
3		Percabangan/Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
4		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
5		Penggabungan/Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu
6.		Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

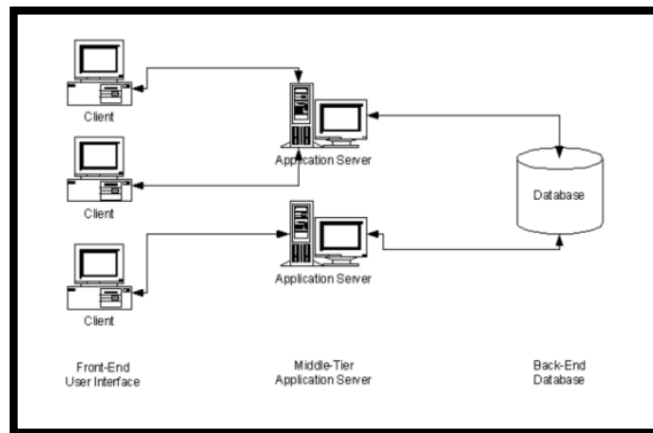
2.8.4 Class Diagram

Class Diagram merupakan salah satu jenis diagram dalam UML yang digunakan untuk merepresentasikan struktur atau *blueprint* dari suatu sistem atau objek yang akan dibangun. *Class Diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Dharwiyanti dan Wahono, 2003). Diagram ini menunjukkan kelas-kelas yang ada dalam sistem beserta atribut-atribut dan metode-metode yang dimiliki oleh masing-masing kelas, serta hubungan antar kelas. *Class Diagram* merupakan sebuah spesifikasi yang jika diinstansikan akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Adapun simbol yang terdapat pada *class diagram* dapat dilihat pada tabel 2.5 di bawah ini.

Tabel 2.5 *Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Association</i>	Menghubungkan satu objek dengan objek yang lainnya
2		<i>Nary Association</i>	Upaya untuk menghindari assosiasi dengan lebih dari 2 objek
3		<i>Class</i>	Kumpulan objek dengan atribut dan operasi yang sama
4		<i>Collaboration</i>	Deskripsi urutan kegiatan yang dilakukan oleh sistem, yang memberikan hasil yang terukur bagi aktor.
5		<i>Realization</i>	Operasi yang sebenarnya dilakukan oleh objek
6		<i>Dependency</i>	Hubungan dimana perubahan pada elemen independen akan mempengaruhi elemen yang bergantung pada elemen yang tidak independen

2.9 Arsitektur *Client-Server*



Gambar 2. 2 Arsitektur *Client Server*

Sumber: (Jasri, 2018)

Client-server merupakan sebuah paradigma dalam teknologi informasi yang merujuk kepada cara untuk mendistribusikan aplikasi dalam dua pihak, pihak klien dan pihak server (Asidiqqi, 2018). Dalam model, ini *client* berfungsi sebagai pengguna akhir (*end user*) yang mengakses sumber daya atau layanan yang diberikan oleh server, sedangkan server berfungsi sebagai penyedia sumber daya atau layanan tersebut. Dalam arsitektur *client-server*, sumber daya atau layanan disimpan di server dan diakses oleh *client* melalui jaringan.

Dalam arsitektur *client-server*, *client* dan *server* berkomunikasi melalui protokol yang telah ditentukan sebelumnya, seperti *Transmission Control Protocol* (TCP) atau *User Datagram Protocol* (UDP). Protokol ini memastikan bahwa data yang dikirim antara *client* dan *server* dapat diterima dan diproses dengan benar. Ketika *client* meminta sumber daya atau layanan dari server, *client* mengirimkan permintaan ke *server* melalui jaringan. *Server* menerima permintaan dan memprosesnya, lalu mengirimkan balasan ke *client* melalui jaringan.

Salah satu keuntungan utama dari arsitektur *client-server* adalah memungkinkan pengembang untuk memperluas aplikasi dan menambahkan fitur baru tanpa mengganggu bagian yang sudah ada. Misalnya, jika suatu aplikasi memiliki lebih banyak pengguna, pengembang dapat menambahkan server tambahan untuk mempercepat kinerja aplikasi.

2.10 Pengertian *Backend*

Backend adalah bagian dari aplikasi atau sistem yang bertanggung jawab untuk pemrosesan dan manajemen data di server. *Backend* terdiri dari berbagai komponen seperti database, server, aplikasi logika bisnis, API, dll. *Backend* digunakan untuk mengelola interaksi antara server dan klien atau sisi pengguna. *Backend* menangani permintaan masuk dari sisi klien, memproses data, menggunakan basis data, mengirim respon, dan memastikan keamanan sistem. *Backend Development* itu sendiri bertanggung jawab untuk sisi *server* dan database (Hasanuddin, Asgar dan Hartono, 2022).

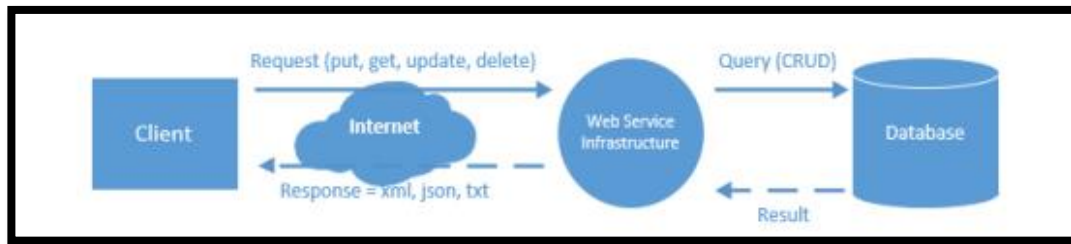
Dalam pengembangan web, backend biasanya ditulis dalam bahasa pemrograman seperti PHP, Python, Ruby, atau Node.js. Backend juga sering dibangun menggunakan framework atau kerangka kerja seperti Laravel, Django, Ruby atau Express.js untuk memudahkan pengembangan dan pengelolaan sistem. Dalam dunia software, sistem backend sangatlah penting karena menangani proses kritis seperti manajemen data, keamanan, dan integrasi sistem.

2.11 Pengertian *Frontend*

Front end, juga dikenal sebagai sisi klien, adalah bagian dari aplikasi atau sistem yang ada di sisi klien atau pengguna. Antarmuka pengguna bertanggung jawab untuk menampilkan informasi dan interaksi antara pengguna dan aplikasi atau sistem. Antarmuka pengguna biasanya terdiri dari layar antarmuka pengguna yang berisi elemen seperti tombol, bentuk, gambar, dan teks. Selain itu, antarmuka pengguna juga memproses masukan pengguna dan meneruskannya ke sistem latar belakang untuk diproses lebih lanjut. Dalam melakukan pertukaran data, *client* tidak dapat mengakses *resource* secara langsung pada *database server* sehingga diperlukan komunikasi antara keduanya agar *client* dapat mengakses *resource* yang ada pada *server* (Safitri dan Putro, 2021).

Dalam pengembangan web, antarmuka pengguna sebagian besar ditulis dalam bahasa pemrograman seperti HTML, CSS, dan JavaScript. Antarmuka pengguna juga dibangun menggunakan kerangka kerja seperti React, Angular, atau Vue.js untuk memfasilitasi pengembangan dan pengelolaan aplikasi. Bagian depan sangat penting dalam pengembangan aplikasi karena merupakan bagian pertama yang dilihat pengguna. Antarmuka pengguna yang baik harus dirancang dengan menarik, mudah digunakan dan dapat mengakses data dengan cepat dan efisien.

2.12 Representational State Transfer API (REST API)



Gambar 2. 3 Arsitektur REST

Sumber: (Chandra dan Irmawan, 2019)

REST API adalah suatu gaya arsitektur perangkat lunak yang sering digunakan dalam pembuatan aplikasi web services. REST API merupakan suatu bentuk implementasi dari arsitektur *client-server* yang digunakan untuk membangun aplikasi web *services* yang dapat digunakan oleh berbagai platform, seperti web, desktop, dan mobile. REST API sangat populer di kalangan pengembang aplikasi web modern. REST API memungkinkan berbagai sistem untuk berkomunikasi dan mengirim/menerima data dengan cara yang sangat sederhana (Choirudin *et al.*, 2019).

Dalam REST API, setiap sumber daya (*resource*) direpresentasikan oleh sebuah URL, atau yang dikenal dengan istilah *endpoint*. Setiap *endpoint* pada REST API menyediakan satu atau lebih operasi yang terkait dengan sumber daya yang direpresentasikan. Data yang dikirim dan diterima REST API umumnya berbentuk JSON atau XML.

RESTful API bekerja dengan dimulai dari pengiriman data atau permintaan oleh klien melalui HTTP *Request*, yang kemudian direspon oleh server melalui HTTP *Response*. Berikut ini merupakan komponen utama dari REST API:

1. *Resource*

Resource merupakan objek atau sumber daya yang dapat diakses melalui REST API, contohnya adalah data pengguna, produk, atau artikel. Setiap *resource* memiliki sebuah identifikasi yang unik, yaitu URI (*Uniform Resource Identifier*).

2. *HTTP Methods*

HTTP Methods digunakan untuk mengelola *resource* pada REST API. Metode yang paling umum digunakan mencakup GET (untuk mengambil data),

POST(untuk mengirimkan data), PUT(untuk mengubah data), dan DELETE (untuk menghapus data).

3. *Representations*

Representations adalah format data yang digunakan untuk memindahkan data dari server ke client dan sebaliknya. Format yang umum digunakan adalah JSON (*Javascript Object Notation*) dan XML (*eXtensible Markup Language*).

4. Kode Status

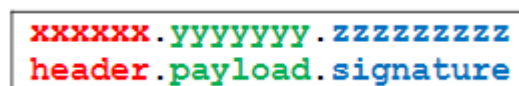
Kode status digunakan untuk memberikan informasi tentang status permintaan yang dikirim oleh *client*. Contohnya, kode status 200 menandakan bahwa permintaan sukses dilakukan, sementara kode status 404 menandakan bahwa resource tidak ditemukan.

5. *Authentication and Authorization*

Authentication and Authorization digunakan untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses resource tertentu. *Authentication* digunakan untuk memverifikasi identitas pengguna, sementara *authorization* digunakan untuk memverifikasikan hak akses pengguna.

2.13 JSON Web Token (JWT)

JWT (*JSON Web Token*) adalah standar industri yang digunakan untuk mengirimkan informasi terenkripsi melalui jaringan komputer. JWT sering digunakan sebagai mekanisme otentikasi pada aplikasi web dan mobile. JWT ini adalah sebuah *token* berbentuk *string* JSON yang sangat padat (ukurannya), informasi mandiri yang gunanya sendiri untuk melakukan sistem autentikasi dan pertukaran informasi (Gunawan dan Rahmatulloh, 2019).



Gambar 2. 4 Struktur JWT

Sumber : (Rahmatulloh, Sulastris dan Nugroho, 2018)

JWT dapat digunakan untuk mengamankan akses pada suatu API. Saat pengguna berhasil login ke aplikasi, server akan memberikan JWT kepada pengguna.

Setiap permintaan selanjutnya dari pengguna yang memerlukan otentikasi harus menyertakan JWT. Server kemudian akan memeriksa apakah JWT tersebut valid dan memiliki akses yang diperlukan.

JWT terdiri dari tiga bagian yaitu *header*, *payload*, dan *signature*. Header menyediakan informasi tentang tipe token dan algoritma yang digunakan untuk membuat *signature*. *Payload* merupakan bagian di mana informasi pengguna (*user*) disimpan dalam bentuk *key-value pair*. *Signature* digunakan untuk memastikan bahwa pesan tidak telah diubah selama pengiriman.

2.14 Payment Gateway

Payment Gateway adalah sebuah sistem yang memungkinkan transaksi keuangan dilakukan secara online antara pembeli dan penjual. Secara sederhana, *payment gateway* berfungsi sebagai perantara antara pembeli, penjual dan lembaga keuangan (seperti bank) untuk memfasilitasi pembayaran melalui kartu kredit, *debit card*, *transfer bank*, atau metode pembayaran elektronik lainnya. *Payment gateway* adalah pembayaran online yang fungsinya mendeskripsikan dan mengesahkan informasi pada sebuah transaksi sesuai dengan kebijakan yang telah diatur oleh para provider

Payment gateway juga bisa dianggap sebagai titik akses dalam jaringan perbankan nasional, karena semua transaksi pembayaran online harus dilakukan melalui *payment gateway* agar dapat diproses. Dalam konsepnya, *payment gateway* merupakan penghubung antara pemilik situs web dan lembaga keuangan untuk melakukan transaksi pembayaran. Kemudian, *payment gateway* juga memiliki fungsi menjadi saluran yang terenkripsi, artinya secara aman mengirimkan detail transaksi dari pembeli dalam hal ini menggunakan personal komputer ke bank untuk dilakukan persetujuan, selanjutnya untuk mendapatkan persetujuan, *payment gateway* mengirimkan kembali informasi kepada pemilik website untuk menyelesaikan order dan menyediakan verifikasi (Islami dan Sulistyowati, 2016).

2.15 Midtrans Payment Gateway

Midtrans adalah salah satu *payment gateway* yang memfasilitasi kebutuhan para pebisnis *online* dengan memberikan pelayanan dengan berbagai metode pembayaran (Alfian, Sokibi and Magdalena, 2020). *Midtrans* menghubungkan bisnis dengan berbagai metode pembayaran elektronik, termasuk kartu kredit, *debit card*, *transfer bank*, dan metode pembayaran digital seperti *e-wallet*. Dengan menggunakan *Midtrans*, bisnis dapat menerima pembayaran secara *online* dengan mudah dan efisien. Dengan *Midtrans* dapat melakukan pembayaran untuk belanja *online*, donasi, produk berlangganan, dan penarikan uang dengan mudah, cepat, dan aman (Nisrina *et al.*, 2019).

2.16 Laravel

Laravel adalah sebuah *framework* PHP yang bersifat *open source* untuk membangun aplikasi web dengan mudah dan cepat. Dalam konteks pengembangan web, sebuah *framework* adalah sekumpulan alat, aturan, dan struktur kerja yang digunakan untuk memudahkan pengembangan aplikasi. Laravel dirancang untuk memudahkan pengembangan aplikasi web dengan menyediakan berbagai fitur dan kemudahan seperti routing, controller, model, view, template engine, dan lain-lain.

Laravel dikembangkan oleh Taylor Otwell pada tahun 2011 sebagai alternatif untuk *framework* PHP yang sudah ada saat itu. Otwell merancang Laravel dengan tujuan membuat *framework* yang mudah digunakan, intuitif, dan menyenangkan untuk digunakan. Laravel pertama kali diluncurkan pada bulan Juni 2011 dan sejak saat itu terus berkembang dan menjadi salah satu *framework* PHP paling populer di dunia.

Laravel merupakan kerangka kerja pengembangan web MVC yang meningkatkan kualitas perangkat lunak dengan meningkatkan produktivitas tenaga dengan sintaks pengembangan dan perbaikan serta secara signifikan mengurangi waktu implementasi (Luthfi, 2017).

2.17 ReactJS

React adalah *open-source library* Javascript deklaratif, efisien dan fleksibel untuk membangun antarmuka pengguna (Nasution and Iswari, 2021). ReactJS dikembangkan oleh Facebook dan dirilis pada tahun 2013. ReactJS digunakan untuk menangani lapisan tampilan dalam aplikasi satu halaman dan aplikasi seluler. React berusaha untuk memberikan kecepatan, kesederhanaan, dan skalabilitas (Nasution and Iswari, 2021).

ReactJS bertujuan untuk memberikan kecepatan, kesederhanaan, dan skalabilitas. React memungkinkan untuk membuat *user interface* yang kompleks dengan set kode kecil yang terisolasi yang disebut komponen (Nasution and Iswari, 2021). Hal ini juga memudahkan developer untuk melakukan perawatan ketika terjadi kesalahan, karena developer langsung ke komponen yang bermasalah tanpa mengganggu komponen lain yang lebih mudah dan cepat.

2.18 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) *open source* yang paling banyak digunakan di dunia. MySQL dibuat dan dikembangkan oleh perusahaan Swedia bernama MySQL AB dan sekarang dimiliki oleh Oracle Corporation. Sebagai salah satu sistem manajemen basis data relasional paling populer di dunia, MySQL memberikan solusi yang andal dan efektif untuk mengelola data dalam aplikasi web. Dalam pengalaman saya sebagai pengembang web, MySQL memungkinkan pengembang untuk membuat aplikasi yang cepat, efisien, dan mudah diakses oleh pengguna.

2.19 Pengujian ISO 25010

Model ISO 25010 merupakan sebagai standar internasional dalam pengujian perangkat lunak yang berlaku saat ini (Lamada *et al.*, 2020). Standar ini membantu untuk mengidentifikasi kriteria kualitas perangkat lunak yang harus dipenuhi untuk memenuhi kebutuhan pengguna dan organisasi yang menggunakan perangkat lunak tersebut. Standar 25010 terdiri dari delapan karakteristik, yaitu kompatibilitas,

keandalan, efisiensi, kemudahan penggunaan, keamanan, keterukuran, portabilitas, dan keterjangkauan.



Gambar 2. 5 Karakteristik ISO 25010

(Sumber: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>)

Berikut ini merupakan penjelasan dari delapan karakteristik Model ISO 25010 yang terdapat pada gambar diatas:

1. Fungsionalitas (Functionality) yaitu kemampuan perangkat lunak untuk memenuhi kebutuhan pengguna dengan akurasi dan kelengkapan.
2. Keandalan (Reliability) yaitu kemampuan perangkat lunak untuk memberikan kinerja yang konsisten dan terpercaya dalam berbagai situasi.
3. Kemudahan penggunaan (Usability) yaitu kemudahan penggunaan dan kepuasan pengguna dalam mengoperasikan perangkat lunak.
4. Efisiensi kinerja (Performance Efficiency) yaitu kemampuan perangkat lunak untuk melakukan tugas dengan cepat dan dengan menggunakan sumber daya yang minimal.
5. Kompatibilitas (Compatibility) yaitu kemampuan perangkat lunak untuk beroperasi dan berintegrasi dengan sistem dan perangkat lainnya.
6. Keamanan (Security) yaitu kemampuan perangkat lunak untuk melindungi data dan sistem dari akses yang tidak sah atau kerusakan.
7. Keterukuran (Maintainability) yaitu kemampuan perangkat lunak untuk diperbaiki, dimodifikasi dan dikembangkan dengan mudah.
8. Portabilitas (Portability) yaitu kemampuan perangkat lunak untuk diimplementasikan dan dioperasikan pada lingkungan yang berbeda.