

## BAB II

### KAJIAN LITERATUR

#### 2.1 Sistem Informasi

Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya. Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi di antara unsur-unsur tersebut dengan lingkungan. Jadi dapat disimpulkan bahwa sistem adalah suatu kesatuan yang terdiri dari elemen-elemen yang saling terkait, baik secara konseptual maupun fisik, dan berada dalam hubungan timbal balik antar bagian-bagian tersebut serta dengan lingkungan sekitarnya. Sistem ini mencakup entitas atau proses yang bersatu untuk mencapai tujuan tertentu, menunjukkan adanya ketergantungan dan interaksi yang kompleks di dalamnya. Selain itu, sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terpadu dan terorganisir, saling berinteraksi dan saling ketergantungan satu sama lain, misalnya sistem komputer yang terdiri dari *hardware* dan *software* [15]. Dengan kata lain, sistem mencerminkan ide integrasi unsur-unsur dalam suatu ketergantungan saling berguna untuk mencapai suatu fungsi atau tujuan tertentu [16].

Informasi adalah hasil dari pengolahan data yang telah dikelola untuk memberikan arti dan meningkatkan kegunaannya. Informasi ini memiliki nilai lebih bagi penerimanya, karena telah diubah menjadi bentuk yang lebih berguna dan bermakna. Selain itu, informasi juga dapat menggambarkan kejadian-kejadian nyata, dan fungsinya sebagai alat bantu dalam proses pengambilan keputusan. Dengan demikian, informasi bukan hanya sekadar data, tetapi merupakan produk dari proses pengolahan data yang memiliki nilai tambah dan relevansi untuk mendukung pengambilan keputusan [16].

Sistem informasi merupakan kombinasi dari teknologi, manusia, dan proses yang dirancang untuk mengumpulkan, memproses, menyimpan dan menyebarkan informasi guna mendukung pengambilan keputusan, koordinasi, analisis, dan kontrol dalam suatu organisasi. Sistem informasi berfungsi untuk mengelola data mentah menjadi informasi yang berguna dan bernilai bagi pengguna [17].

Definisi lainnya mengatakan bahwa sistem informasi adalah suatu rangkaian aktivitas yang meliputi proses pengumpulan, penyimpanan, serta analisis informasi dengan tujuan yang telah ditentukan. Sistem informasi ini terdiri dari data yang dimasukkan (*input*) dan menghasilkan laporan sebagai keluaran (*output*), sehingga informasi tersebut dapat

diproses oleh sistem lainnya. Selain itu, sistem informasi juga mendukung kegiatan strategis dalam suatu organisasi dengan memberikan panduan untuk tindakan atau keputusan. Seiring dengan meningkatnya kebutuhan akan informasi yang cepat dan akurat, pengembangan sistem informasi menjadi hal yang sangat penting. Proses pengembangan ini bertujuan untuk merancang, membangun, serta mengimplementasikan sistem yang mampu memenuhi kebutuhan pengguna secara efektif dan efisien [16].

## 2.2 Pengembangan Sistem Informasi

Terdapat beberapa definisi pengembangan sistem informasi dari beberapa sumber, antara lain sebagai berikut.

1. Pengembangan sistem informasi merupakan tahapan yang melibatkan pemanfaatan teknologi informasi untuk meningkatkan efektivitas dan efisiensi pekerjaan manusia. Dengan berbagai metode dan platform, seperti *mobile*, *web*, atau *desktop*, sistem informasi dirancang untuk mengumpulkan, memproses, menyimpan, dan menganalisis informasi guna mendukung pengambilan keputusan dalam bisnis atau perusahaan [18].
2. Pengembangan sistem informasi adalah langkah-langkah yang digunakan untuk merancang dan mengembangkan sistem informasi yang efektif dan efisien. Proses ini melibatkan tujuan pengembangan, analisis kebutuhan pengguna, perancangan sistem, implementasi, uji coba, evaluasi, dan pemeliharaan sistem informasi [16].
3. Pengembangan sistem informasi adalah proses merancang, mengembangkan, dan mengimplementasikan sistem informasi menggunakan pendekatan, strategi, dan alat tertentu [19].

Berdasarkan definisi-definisi yang telah diuraikan sebelumnya, maka dapat disimpulkan bahwa pengembangan sistem informasi adalah suatu proses terstruktur yang mencakup perencanaan, perancangan, implementasi, dan pemeliharaan sistem informasi dengan memanfaatkan teknologi informasi guna meningkatkan efisiensi, efektivitas, serta mendukung pengambilan keputusan dalam organisasi atau perusahaan. Proses ini melibatkan analisis kebutuhan pengguna, pemilihan platform yang sesuai (seperti *web*, *mobile*, atau *desktop*), serta penggunaan metode dan alat tertentu agar sistem yang dikembangkan mampu memenuhi tujuan dan kebutuhan pengguna secara optimal.

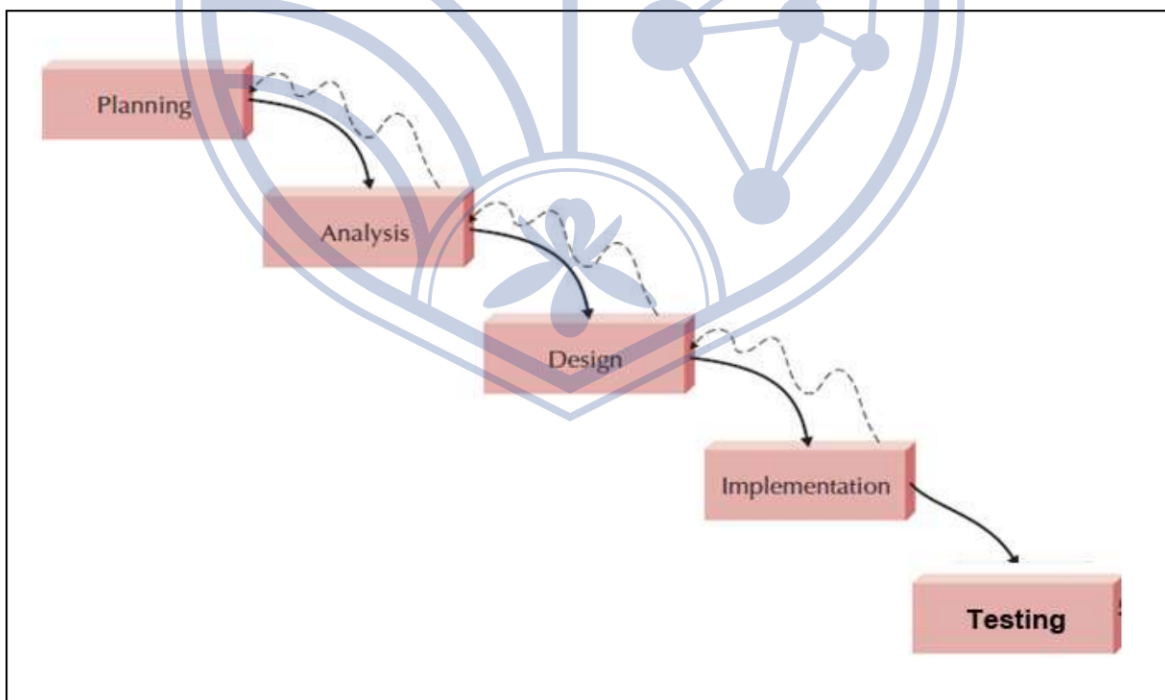
## 2.3 Metodologi Pengembangan Sistem *Waterfall Model*

Tahapan pengembangan sistem adalah serangkaian proses yang dilakukan dalam mengembangkan sistem informasi atau perangkat lunak dari awal hingga selesai. Terdapat

beberapa metode pengembangan sistem, namun yang umum digunakan adalah metode *Waterfall*. Pengembangan sistem informasi sering juga disebut dengan *Software Development Life Cycle* (SDLC) yaitu siklus yang digunakan dalam pembuatan atau pengembangan sistem informasi untuk menghasilkan sistem berkualitas tinggi yang sesuai dengan keinginan pelanggan atau tujuan dibuatnya sistem tersebut [20]. Dalam penelitian ini, aplikasi penyedia jasa renovasi berbasis *web* yang akan dibangun dikembangkan dengan menggunakan tahapan pengembangan sistem dari metode *Waterfall Model* [21].

*Waterfall* merupakan model pengembangan tradisional yang bersifat linier dan berurutan, di mana setiap tahapan seperti analisis, desain, implementasi, dan pengujian dilakukan secara bertahap dan tidak dapat kembali ke tahap sebelumnya. Keunggulan *Waterfall* terletak pada dokumentasi yang jelas dan struktur proyek yang rapi, yang sangat sesuai untuk proyek dengan kebutuhan tetap dan pengawasan ketat. Oleh karena itu, model ini banyak digunakan dalam proyek yang memerlukan pelacakan formal serta verifikasi terhadap setiap tahapan, seperti dalam sistem yang terkait dengan regulasi atau audit. Namun, model ini kurang fleksibel dalam menghadapi perubahan yang sering kali terjadi di tengah jalan, terutama bila terdapat pembaruan regulasi atau penyesuaian kebutuhan pengguna [22].

Metode *Waterfall* sering disebut pengembangan air terjun karena bergerak maju dari fase ke fase dengan cara yang sama seperti air terjun. Gambar 2.1 berikut ini menunjukkan tahapan pengembangan sistem dengan metode *Waterfall Model* [21].



Gambar 2.1 Tahapan Metode *Waterfall Model* [17]

Berikut ini akan diuraikan tahapan-tahapan dari metode *Waterfall Model* yang ditunjukkan pada Gambar 2.1, yaitu [21]:

1. *Planning*/Perencanaan

Di dalam tahapan ini, persyaratan potensial dari aplikasi dianalisis ke dalam dokumen spesifik yang berfungsi sebagai dasar untuk semua pengembangan di masa mendatang. Proses ini akan menghasilkan dokumen persyaratan yang menentukan apa yang harus dilakukan aplikasi, bukan bagaimana cara melakukannya.

2. *Analysis*/Analisis

Selama tahap kedua, sistem akan dianalisis untuk menghasilkan model dan logika bisnis yang akan digunakan dalam aplikasi.

3. *Design*/Desain

Tahap ketiga ini biasanya mencakup kepentingan desain teknis, seperti bahasa pemrograman, lapisan data, layanan, dan sebagainya. Spesifikasi desain biasanya akan dibuat untuk menguraikan bagaimana logika bisnis yang tercakup dalam analisis akan diimplementasikan secara teknis.

4. *Implementation*/Implementasi

Dari tahap ketiga dan kedua didapatkan sebuah rancangan sistem secara teknis, kemudian akan dilakukan implementasi dengan membangun aplikasi menggunakan bahasa-bahasa pemrograman yang dipilih.

5. *Testing*/Pengujian

Tahap terakhir adalah melakukan *testing* terhadap aplikasi yang telah dibangun dan diimplementasikan agar dapat diperbaiki *bug* ataupun *error* yang terdapat pada aplikasi.

## 2.4 Aplikasi Web

*Web* atau yang biasa disebut *website* atau singkatan dari “*World Wide Web*” adalah sistem informasi global yang terdiri dari jaringan komputer yang saling terhubung secara global melalui protokol internet. *World Wide Web* yaitu media informasi global dimana pengguna dapat membaca dan menulis melalui komputer yang terhubung ke internet. Pada dasarnya sebuah sistem server Internet yang mendukung dokumen yang diformat secara khusus. Dokumen diformat dalam bahasa markup yang disebut HTML (*Hypertext Markup Language*) yang mendukung tautan ke dokumen lain, serta *file* grafik, *audio*, dan *video*. Singkatnya, *World Wide Web* (WWW) adalah kumpulan halaman teks, foto digital, *file* musik, *video*, dan animasi yang dapat Anda akses melalui internet. Halaman *web* pada dasarnya merupakan dokumen teks yang diformat dan dianotasi dengan *Hypertext Markup*

*Language* (HTML). Selain teks yang diformat, halaman web mungkin berisi gambar, video, dan komponen perangkat lunak yang ditampilkan di browser *web* pengguna sebagai halaman konten multimedia yang koheren. Istilah internet dan *World Wide Web* sering digunakan tanpa banyak perbedaan, namun keduanya tidak sama [23].

*Web* merupakan salah satu komponen utama dari internet yang memungkinkan pengguna untuk mengakses, mencari, dan berbagi berbagai jenis informasi, data, gambar, video, dan lainnya dengan mudah melalui *browser web*. *Web* adalah media yang sangat luas dan memiliki berbagai jenis dan fungsi yang beragam. Berikut ini adalah beberapa jenis *web* dan fungsi utamanya [24]:

1. *Web* informasi.

Jenis *web* ini bertujuan untuk menyediakan informasi kepada pengguna. *Web* ini bisa berupa situs berita, ensiklopedia *online*, portal informasi pemerintah, dan lain sebagainya. Tujuannya adalah memberikan akses cepat dan mudah ke informasi.

2. *Web E-commerce*.

*Web e-commerce* adalah tempat untuk membeli dan menjual produk dan jasa secara *online*. *Web* ini mencakup toko *online*, pasar elektronik seperti Amazon, dan platform lelang seperti eBay. Fungsi utamanya adalah memungkinkan transaksi bisnis *online*.

3. *Web* sosial.

*Web* sosial adalah platform yang memungkinkan pengguna berinteraksi, berbagi konten, dan terhubung dengan orang lain secara *online*. Contoh termasuk Facebook, Twitter, Instagram, dan LinkedIn. Tujuannya adalah memfasilitasi komunikasi sosial dan berbagi informasi.

4. *Web* pendidikan.

*Web* pendidikan menyediakan akses ke pembelajaran *online*, kursus, materi pelajaran, dan sumber daya pendidikan lainnya. *Web* ini memungkinkan siswa dan pembelajar dewasa untuk mengakses pengetahuan dan keterampilan secara *online*.

5. *Web* hiburan.

*Web* hiburan menawarkan hiburan dalam bentuk video, musik, *game online*, dan banyak lagi. Layanan *streaming* video seperti Netflix, platform musik seperti Spotify, dan situs *web game online* adalah contoh dari jenis *web* ini.

6. *Web* berbagi konten.

*Web* ini memungkinkan pengguna untuk mengunggah, berbagi, dan mengakses konten seperti gambar, video, musik, dan dokumen. Contoh termasuk YouTube (video), Instagram (gambar), dan SlideShare (presentasi).

7. *Web* bisnis.

*Web* bisnis adalah platform yang digunakan oleh perusahaan untuk berkomunikasi dengan pelanggan, mengelola operasi bisnis, dan menjalankan layanan seperti pemesanan *online*. *Web* ini bisa berupa situs *web* perusahaan, portal pelanggan, atau platform manajemen inventaris.

8. *Web* berita dan *blog*.

*Web* berita dan *blog* adalah tempat di mana individu atau organisasi dapat memublikasikan berita, artikel, pandangan, dan opini. *Web* ini adalah sumber berita dan informasi yang luas.

9. *Web* pemerintah.

*Web* pemerintah digunakan oleh pemerintah untuk memberikan informasi kepada warga negara, menerima laporan, dan memberikan layanan pemerintah *online* seperti perpajakan, pendaftaran kendaraan, dan perizinan.

Pada penelitian ini, jenis *web* yang akan dibangun yaitu *web e-commerce* dikarenakan penelitian ini bertujuan untuk mengembangkan sebuah *web marketplace* penyedia jasa renovasi.

## 2.5 Unified Modeling Language (UML)

Pada bagian ini akan dibahas mengenai *Unified Modeling Language* (UML) yang merupakan salah satu alat bantu utama dalam proses perancangan sistem berbasis *Object-Oriented*. UML digunakan untuk memvisualisasikan, merancang, serta mendokumentasikan struktur dan perilaku sistem secara lebih terstruktur dan mudah dipahami.

### 2.5.1 Pengertian Unified Modeling Language (UML)

*Unified Modeling Language* (UML) merupakan sebuah bahasa pemodelan berbasis grafik yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, serta mendokumentasikan sistem pengembangan perangkat lunak berbasis *Object-Oriented* (OO). UML tidak hanya berfungsi sebagai bahasa pemrograman visual, tetapi juga dapat diintegrasikan dengan berbagai bahasa pemrograman seperti Java, C++, Visual Basic, bahkan dengan basis data berorientasi objek (*object-oriented database*) [25].

UML menyediakan notasi grafis untuk memvisualisasikan elemen-elemen penting dalam sistem, seperti aktor, kelas, objek, aliran data, serta hubungan antar komponen UML tidak hanya terbatas pada dokumentasi statis, tetapi juga mampu menggambarkan dinamika dan perilaku sistem secara menyeluruh. Karena kemampuannya dalam menyatukan berbagai

pendekatan pemodelan sebelumnya, UML menjadi alat penting dalam seluruh siklus hidup pengembangan perangkat lunak, dari tahap *requirement* hingga *deployment* [26].

Salah satu tahapan penting dalam *Software Development Life Cycle* (SDLC) adalah tahap desain. Tahap ini bertujuan agar perangkat lunak yang dikembangkan mampu memenuhi kebutuhan pengguna (*user*) serta memiliki kinerja yang handal. Oleh karena itu, desain menjadi bagian yang sangat krusial dalam proses pengembangan perangkat lunak. Dalam tahap ini, kebutuhan pengguna baik secara fungsional maupun non-fungsional ditransformasikan ke dalam model yang menggambarkan bagaimana sistem akan dibangun dan dijalankan [25].

### 2.5.2 Jenis-Jenis Diagram *Unified Modeling Language* (UML)

Terdapat beberapa jenis-jenis diagram UML beserta penjelasannya [25]:

1. *Use Case Diagram* yaitu diagram yang digunakan untuk menggambarkan interaksi antara sistem dengan pengguna (*user*) atau sistem lain. Diagram ini sangat berguna untuk memetakan kebutuhan (*requirement*) dari suatu sistem.
2. *Activity Diagram* yaitu diagram yang digunakan untuk menggambarkan alur aktivitas secara sekuensial maupun paralel dalam sistem. Diagram ini digunakan untuk menunjukkan logika proses atau alur kerja (*workflow*) suatu sistem.
3. *Class Diagram* yaitu diagram yang digunakan untuk menggambarkan kelas (*class*), tipe, *interface*, serta hubungan antar ketiganya. Diagram ini merupakan inti dari pemodelan berorientasi objek.
4. *Object Diagram* yaitu diagram yang digunakan untuk menggambarkan *instance* dari objek-objek yang telah didefinisikan pada *class diagram*, serta memperlihatkan hubungan antar objek tersebut.
5. *Sequence Diagram* yaitu diagram yang digunakan untuk menggambarkan interaksi antar objek di dalam sistem, dengan menekankan pada urutan pesan atau interaksi yang terjadi dari waktu ke waktu.
6. *Communication Diagram* yaitu diagram yang digunakan untuk menggambarkan cara objek berinteraksi satu sama lain serta koneksi yang dibutuhkan untuk melakukan interaksi tersebut.
7. *Timing Diagram* yaitu diagram yang digunakan untuk menggambarkan interaksi antar objek dengan menekankan aspek waktu sebagai faktor yang penting dalam proses komunikasi.

8. *Component Diagram* yaitu diagram yang digunakan untuk menggambarkan komponen-komponen penting dalam sistem serta *interface* yang digunakan untuk saling berinteraksi antar komponen.
9. *Package Diagram* yaitu diagram yang digunakan untuk menggambarkan struktur hierarki dan pengelompokan dari sekumpulan *class* atau *component* dalam sistem.
10. *State Machine Diagram* yaitu diagram yang digunakan untuk menggambarkan berbagai keadaan (*state*) dari suatu objek selama masa aktifnya, serta peristiwa (*event*) yang dapat menyebabkan perubahan keadaan tersebut.
11. *Deployment Diagram* yaitu diagram yang digunakan untuk menggambarkan bagaimana sistem diimplementasikan (*deployed*) di lingkungan nyata, termasuk perangkat keras (*hardware*) dan hubungan antar komponennya.

Pada penelitian ini, jenis diagram UML yang digunakan adalah *Use Case Diagram* dikarenakan penelitian ini merupakan pengembangan aplikasi yang memerlukan pemodelan dari fungsionalitas sistem usulan yang akan dikembangkan pada penelitian ini.

### 2.5.3 Use Case Diagram

UML menggunakan beberapa grafik untuk mewakili sistem dari dua sudut pandang yaitu secara perilaku dan struktural. Salah satu diagram yang dikualifikasikan kedalam aspek perilaku adalah *Use Case Diagram*. Dalam *Use Case Diagram*, setiap perilaku dijelaskan secara detail. Dokumen tekstual, yang termasuk skenario kasus penggunaan, menggunakan spesifikasi kasus, atau menggunakan deskripsi kasus, digunakan untuk mengidentifikasi mereka [27].

*Use Case Diagram* adalah salah satu alat pemodelan sistem yang digunakan untuk menggambarkan interaksi antara pengguna dan sistem. Penggunaan Diagram *Use Case* dapat memberikan gambaran yang efektif dalam menjelaskan konteks suatu sistem dengan jelas, sehingga memperlihatkan batasan sistem tersebut dengan jelas. Adapun, fungsi dari *Use Case Diagram* sebagai berikut [25]:

1. Berguna memperlihatkan proses aktivitas secara urut dalam sistem.
2. Mampu menggambarkan proses bisnis, bahkan menampilkan urutan aktivitas pada sebuah proses.
3. Sebagai *bridge* atau jembatan antara pembuat dengan konsumen untuk mendeskripsikan sebuah sistem.

*Use Case Diagram* akan sangat membantu kita dalam menyusun kebutuhan-kebutuhan (*requirement analisis*) sebuah sistem. *Use case diagram* akan digunakan untuk


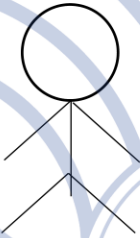

mengkomunikasikan rancangan sistem dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem [28] .

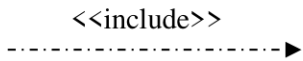
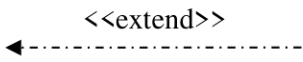

Manfaat dari *Use Case* di antaranya [25]:

1. Menggunakannya sebagai kebutuhan verifikasi.
2. Menjadi gambaran *interface* dari sebuah sistem karena setiap sistem yang dibangun haruslah memiliki *interface*.
3. Mengidentifikasi siapa saja orang yang dapat berinteraksi dengan sistem, serta apa yang dapat dilakukan oleh sistem.
4. Memberikan kepastian mengenai kebutuhan sistem, sehingga tidak membingungkan.
5. Memudahkan proses komunikasi antara *domain expert* dan *end user*.

Terdapat beberapa simbol-simbol dari *Use Case Diagram* seperti yang ditunjukkan pada Tabel 2.1 berikut [25].

Tabel 2.1 Simbol *Use Case Diagram* [25]

Nama	Simbol	Deskripsi
<i>Use Case</i>		Simbol yang menggambarkan fungsi dari sistem umumnya diekspresikan dalam bentuk kata kerja di awal setiap frasa.
Aktor		Simbol yang mewakili pengguna yang terlibat dalam <i>Use Case</i> biasanya menyatakan orang-orang yang berinteraksi dengan sistem.
Asosiasi		Simbol yang menghubungkan antara <i>Use Case</i> dan aktor menggambarkan komunikasi antara keduanya. Asosiasi digunakan untuk menggambarkan hubungan statis antara elemen-elemen tersebut.

<i>Include</i>		Simbol ini menggambarkan bahwa <i>Use Case</i> secara keseluruhan merupakan bagian dari fungsionalitas <i>Use Case</i> lainnya.
<i>Extend</i>		Simbol yang serupa dengan " <i>include</i> ", namun menunjukkan bahwa sebuah <i>Use Case</i> tambahan dapat berdiri sendiri tanpa membutuhkan <i>Use Case</i> yang ditambahkan.
Sistem		Simbol yang membatasi bahwa setiap interaksi dengan simbol lain harus berada di dalamnya.

## 2.6 E-commerce

*E-commerce* yang merupakan singkatan dari *Electronic Commerce* atau perdagangan elektronik, adalah suatu bentuk perdagangan yang melibatkan pembelian dan penjualan barang atau jasa melalui internet atau platform elektronik lainnya. Dalam konteks ini, transaksi dilakukan secara elektronik, tanpa kehadiran fisik pembeli dan penjual di lokasi yang sama [29].

Terdapat beberapa klasifikasi model bisnis *web e-commerce* sebagai berikut [30]:

### 1. *Business to Business* (B2B).

Model bisnis *e-commerce* yang pertama adalah *Business to Business* (B2B) yang merupakan salah satu model bisnis terbanyak dan terbesar penggunaannya, meliputi semua transaksi yang dilakukan antarperusahaan bisnis di berbagai jenis usaha. *Business to Business* (B2B) merupakan salah satu transaksi terbaik bagi perusahaan penjual barang dagang dan juga perusahaan jasa secara elektronik. Hal ini disebabkan badan atau perusahaan yang menjalankan bisnis, bukan konsumen individual perorangan. Umumnya model bisnis *e-commerce* perusahaan seperti ini menawarkan hal-hal terkait inventarisasi bisnis, misalnya seperti mesin pabrik, peralatan kantor, dan sebagainya, meskipun juga ditawarkan ke konsumen individual sebagai pelaku bisnis.

### 2. *Business to Consumer* (B2C).

Model bisnis *Business to Consumer* (B2C) adalah model usaha yang menjual produk langsung kepada konsumen akhir melalui internet, sering disebut juga *e-tailing*. Berbeda

dengan B2B, B2C tidak memerlukan strategi pemasaran yang kompleks karena konsumen cenderung cepat dalam mengambil keputusan pembelian. Produk dan layanan yang ditawarkan sangat beragam, mulai dari kebutuhan sehari-hari hingga barang dan jasa seperti peralatan rumah tangga, buku, *fashion*, layanan perbankan, *travel*, hingga informasi kesehatan. Contoh perusahaan B2C global yang sukses adalah Amazon, yang melayani pembelian berbagai produk secara *online* melalui situsnya.

3. *Business to Business to Consumer (B2B2C)*.

Model bisnis *Business to Business to Consumer (B2B2C)* melibatkan dua perusahaan sebelum produk sampai ke konsumen akhir. Perusahaan pertama (B1) menjual produk ke perusahaan kedua (B2), lalu B2 menjualnya kembali atau memberikannya kepada individu seperti pelanggan atau karyawan mereka. Contoh B2B2C adalah *godiva.com*, dimana Godiva menjual cokelat ke perusahaan lain, lalu perusahaan tersebut mengirimkan cokelat ke penerima akhir, misalnya sebagai hadiah, dengan mencantumkan nama pengirim perusahaan.

4. *Consumer to Business (C2B)*.

Model *Consumer to Business (C2B)* adalah jenis *e-commerce* yang memberikan layanan kepada individu atau konsumen yang memiliki produk atau jasa dan menjualnya kepada pelaku usaha yang membutuhkan. Seseorang akan menggunakan internet untuk menawarkan produk atau jasanya kepada *customer* individu atau *customer* perusahaan. Contoh seperti *web* yang dikelola individu profesional yang membuka layanan untuk mendesain *web* ataupun mendesain logo yang dibuat, kemudian akan dikirim untuk dijual kepada perusahaan yang membutuhkan.

5. *Consumer to Consumer (C2C)*.

Model bisnis *e-commerce Consumer to Consumer* atau sering juga disebut *Citizen to Citizen* yang mana disingkat C2C merupakan salah satu bentuk perdagangan elektronik yang paling umum dimana agen mentransaksikan barang atau jasa dari satu konsumen ke konsumen lainnya. Dalam model bisnis C2C ini terbagi atas 2 saluran, yakni:

a. *Classified*.

Model *Classified Consumer to Consumer (C2C)* adalah model bisnis *e-commerce* yang sederhana dan cocok untuk negara berkembang seperti Indonesia. Situs *web* dalam model ini hanya menghubungkan penjual dan pembeli tanpa memfasilitasi perdagangan *online*, seperti pembayaran elektronik. Penjual individu bebas berjualan kapan saja dan di mana saja, dengan transaksi yang umumnya dilakukan secara

langsung, seperti metode *Cash on Delivery* (COD). Contoh platform *classified C2C* adalah Kaskus dan OLX.

b. *Marketplace*.

Model bisnis *C2C marketplace* adalah bentuk *e-commerce* dimana situs tidak hanya mempromosikan produk, tetapi juga memfasilitasi pembayaran *online*. Ciri utamanya adalah semua transaksi didukung oleh platform dan dapat digunakan oleh setiap penjual. Pengguna bisa berperan sebagai penjual maupun pembeli dalam satu platform. Contoh populer di Indonesia adalah Tokopedia, Bukalapak, dan Shopee. Untuk metode pembayaran *Cash on Delivery* (COD), Shopee dikenal sebagai *marketplace* yang mendukung skema ini.

6. *E-government*.

Dalam model *e-commerce* berbentuk *e-government* dimana pemerintah membeli atau menyediakan layanan jasa atau informasi ke pelaku bisnis. *E-government* disebut juga dengan *Government to Business* (G2B). Terdapat 2 model turunan atas G2B yakni:

a. *Business to Administration* (B2A).

*Business to Administration* atau disingkat B2A merupakan salah satu model bisnis dari *e-commerce* dimana pelayanan transaksi kepada perusahaan ditawarkan oleh lembaga bagian pelayanan administrasi publik atau sebuah lembaga masyarakat secara *online*.

b. *Consumer to Administration* (C2A).

Bentuk turunan bisnis dari G2B adalah *Consumer to Administration* (C2A) yang secara proses tidak berbeda jauh dengan model bisnis B2A. Perbedaan hanya terdapat pada prosesnya, yakni pemberian pelayanan informasi.

7. *Intrabusiness E-commerce*.

*Intrabusiness (organizational) e-commerce* adalah kegiatan yang mencakup seluruh aktivitas internal suatu organisasi melalui internet untuk bertukar barang, jasa, informasi, menjual produk perusahaan kepada karyawan, dan lainnya. Aktivitas internal ini bisa bermacam-macam, mulai dari menjual produk korporat kepada karyawan hingga aktivitas pelatihan *online* bagi karyawan.

8. *Mobile Commerce* (*M-commerce*).

*Mobile commerce* (*m-commerce*) merupakan bentuk dari *e-commerce*, hanya saja fokus pada penggunaan sarana ponsel *smartphone* dan *tablet*. Sejumlah *marketplace* menyediakan sarana aplikasi *mobile* untuk proses pemasaran dan penjualan yang lebih intensif.

## 9. *Online to Offline (O2O)*.

Model bisnis *Online to Offline (O2O)* adalah model bisnis yang prosesnya menggunakan dua jalur, yaitu baik *online* dan *offline*. Pada model bisnis ini, produsen akan selalu melakukan program promosi atau pemasaran, mencari konsumen dan menarik konsumen dengan cara membuat mereka lebih mengenal produk atau jasa yang ditawarkan secara *online*.

Pada penelitian ini, *web* aplikasi penyedia jasa renovasi yang akan dibangun diimplementasikan dalam model bisnis *e-commerce Consumer to Consumer (C2C)* dengan saluran *marketplace*.

## 2.7 *Google Maps*

*Google Maps* adalah layanan pemetaan digital berbasis *web* yang dikembangkan oleh Google untuk memberikan informasi lokasi, rute, citra satelit, dan navigasi. Teknologi ini menggabungkan data geospasial dari berbagai sumber seperti satelit, kendaraan *Street View*, mitra pemetaan, serta *input* pengguna dan memvisualisasikannya dalam bentuk peta interaktif. *Google Maps* menggunakan format peta vektor sehingga tampilannya responsif, hemat data, dan dapat di-*zoom* tanpa kehilangan kualitas. *Google Maps* juga memanfaatkan kecerdasan buatan dan *machine learning* untuk menganalisis kondisi lalu lintas *real-time*, memprediksi kepadatan kendaraan, dan memperkirakan waktu tempuh. Sistemnya dirancang untuk diperbarui secara terus menerus sehingga data peta tetap akurat, termasuk penambahan jalan baru, perubahan arah jalan, dan informasi tempat bisnis [31].

Terdapat beberapa cara kerja dari *Google Maps* antara lain [32]:

### 1. Pengumpulan Data.

*Google Maps* mengumpulkan data dari citra satelit, foto udara, *Street View*, *Global Positioning System (GPS)* dari pengguna *smartphone*, serta informasi dari lembaga pemerintah dan komunitas lokal. Data *crowdsourcing* dari pengguna juga membantu memperbarui informasi seperti ulasan tempat, foto, jam buka, dan laporan kemacetan atau jalan ditutup.

### 2. Pemrosesan dan Analisis Data.

Data yang dikumpulkan diproses menggunakan algoritma dan kecerdasan buatan. *Google* memanfaatkan *deep learning* untuk memprediksi lalu lintas berdasarkan kecepatan kendaraan pengguna lain, pola perjalanan historis, dan faktor geografis. Teknologi OCR (*Optical Character Recognition*) digunakan untuk membaca nama jalan dan tanda dari citra *Street View*.

### 3. Penyajian Peta dan Navigasi.

Setelah diproses, peta ditampilkan dalam format vektor. Pengguna menerima informasi berupa peta interaktif, rute tercepat, estimasi waktu perjalanan (ETA), dan navigasi langkah demi langkah. *Google Maps API* menyediakan antarmuka bagi pengembang untuk mengintegrasikan fitur peta ke dalam aplikasi atau *website*.

### 4. Pembaharuan Berkelanjutan.

*Google Maps* melakukan pembaruan rutin untuk memperbaiki kesalahan peta, menambah lokasi baru, memperbaiki rute, dan memperbarui data lalu lintas. Pembaruan *traffic* dilakukan hampir *real-time* berdasarkan sensor perangkat dan data *crowdsourcing* dari jutaan pengguna di lapangan.

Dalam penerapan nyata, permasalahan *shortest path* tidak hanya diselesaikan secara teoretis menggunakan algoritma seperti *Dijkstra*, *Bellman-Ford*, atau *Floyd-Warshall*, tetapi juga telah diimplementasikan dalam berbagai layanan pemetaan digital, salah satunya adalah *Google Maps* melalui *Google Routes API*. Integrasi *Google Maps* berperan dalam beberapa tahapan proses, seperti penentuan titik lokasi renovasi, perhitungan estimasi jarak, hingga pengecekan area layanan. Sedangkan, *Google Routes API* menyediakan kemampuan untuk menghitung jarak dan waktu tempuh terpendek antara dua atau lebih titik lokasi berdasarkan kondisi aktual di jalan raya. Dengan demikian, konsep graf berbobot pada teori graf diimplementasikan secara langsung pada data spasial dunia nyata, di mana setiap titik lokasi (*vertex*) merepresentasikan posisi geografis, sedangkan setiap ruas jalan (*edge*) memiliki bobot berupa jarak atau estimasi waktu tempuh [33].

*Google Routes API* menggunakan pendekatan berbasis *weighted directed graph*, di mana bobot setiap sisi tidak hanya ditentukan oleh jarak geometris, tetapi juga memperhitungkan berbagai faktor dinamis seperti kondisi lalu lintas (*real-time traffic*), batas kecepatan, jenis jalan, dan arah lalu lintas satu arah (*one-way direction*). Dengan kata lain, bobot yang digunakan bukanlah nilai statis, melainkan hasil perhitungan dinamis yang terus diperbarui sesuai kondisi aktual jaringan jalan. Perhitungan lintasan terpendek dilakukan dengan mengombinasikan algoritma pencarian rute efisien seperti *Dijkstra's Algorithm* atau *A\** (*A-star*) dengan data spasial yang tersimpan dalam infrastruktur *Google Maps* [34] [35].

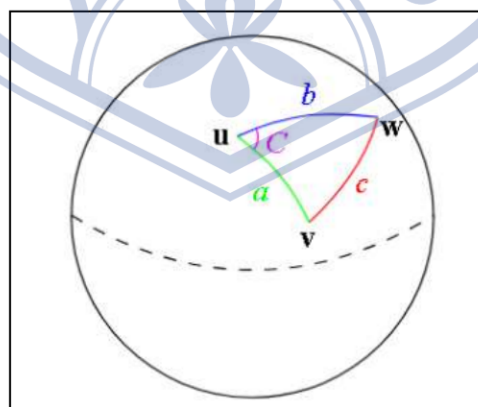
Dalam proses perhitungan jarak, *Google Routes API* mengubah koordinat geografis asal dan tujuan menjadi simpul (*node*) pada graf, kemudian menghubungkannya melalui sisi-sisi yang mewakili ruas jalan. Setiap sisi memiliki atribut seperti panjang jalan (dalam meter), estimasi waktu tempuh, dan tingkat kemacetan. Selanjutnya, algoritma akan menelusuri semua kemungkinan lintasan dan memilih jalur dengan total bobot minimum,

baik berdasarkan jarak terpendek maupun waktu tempuh tercepat. Oleh karena itu, hasil yang diperoleh dapat bervariasi tergantung parameter yang digunakan, misalnya mode perjalanan (mengemudi, berjalan kaki, bersepeda), waktu keberangkatan, serta kondisi lalu lintas pada saat permintaan dilakukan [34], [35].

Selain menghitung jarak dan waktu tempuh antara dua titik, *Google Routes API* juga mendukung perhitungan *multi-destination routing* yang sejalan dengan konsep *Travelling Salesman Problem (TSP)* pada teori graf. Fitur ini memungkinkan sistem untuk menentukan urutan kunjungan optimal ke beberapa lokasi dengan total jarak atau waktu tempuh minimal. Dengan demikian, implementasi *Google Routes API* menjadi contoh nyata penerapan teori graf dan permasalahan *shortest path* dalam dunia nyata yang memanfaatkan data spasial, algoritma optimasi, serta komputasi berbasis jaringan global [19], [36].

## 2.8 Metode Haversine

Metode *Haversine* merupakan metode perhitungan jarak yang digunakan untuk menentukan jarak terpendek antara dua titik di permukaan bumi berdasarkan koordinat lintang (latitude) dan bujur (*longitude*). Metode ini mempertimbangkan kelengkungan bumi dengan menggunakan pendekatan model bumi berbentuk bola, sehingga mampu menghasilkan estimasi jarak yang lebih mendekati kondisi sebenarnya dibandingkan perhitungan jarak pada bidang datar. Oleh karena itu, metode Haversine banyak digunakan dalam sistem informasi geografis dan layanan berbasis lokasi (*Location Based Service*) untuk perhitungan jarak antar lokasi [37]. Gambar 2.2 memperlihatkan bentuk pola perhitungan jarak menggunakan *Haversine* formula, di mana jarak dihitung berdasarkan sudut pusat antara dua titik pada permukaan bola [38].



Gambar 2.2 Bentuk Pola Metode Haversine [38]

Berikut ini adalah rumus dari *Haversine* formula antara lain [38]:

$$\text{hav}\left(\frac{d}{r}\right) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1) \quad (1)$$

Dimana:

1. Hav adalah fungsi *harvesine*

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos \theta}{2} \quad (2)$$

2.  $d$  adalah jarak antara 2 titik
3.  $r$  adalah radius lingkaran
4.  $\varphi_1, \varphi_2$ : lintang pada poin 1 and lintang pada poin 2, dalam radians
5.  $\lambda_1, \lambda_2$ : bujur pada poin 1 and bujur pada poin 2, dalam radians

Pada persamaan, simbol  $\frac{d}{r}$  merupakan sudut pusat yang diukur dalam satuan radian. Untuk mendapatkan nilai  $d$  dapat diterapkan rumus kebalikan *haversine* atau dengan menggunakan arcsine (inverse sinus) fungsi: [38]

$$d = r \text{hav}^{-1}(h) = 2r \arcsin(\sqrt{h}) \quad (3)$$

Dimana  $h$  adalah  $\text{hav}\left(\frac{d}{r}\right)$

$$\begin{aligned} d &= 2r \arcsin\left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)}\right) \\ &= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \end{aligned} \quad (4)$$

Dalam penerapannya, metode Haversine umum digunakan pada aplikasi pemetaan digital, sistem pencarian lokasi terdekat, serta layanan berbasis web dan mobile. Metode ini sering dimanfaatkan sebagai tahap awal dalam proses penentuan kedekatan lokasi karena memiliki tingkat akurasi yang memadai untuk skala wilayah lokal hingga regional serta efisien dari sisi komputasi [39].

Keunggulan metode Haversine terletak pada kesederhanaan perhitungan dan kebutuhan data yang relatif minimal, yaitu hanya berupa koordinat geografis tanpa memerlukan informasi jaringan jalan. Dengan karakteristik tersebut, metode ini sesuai digunakan sebagai mekanisme penyaringan awal (initial filtering) sebelum dilakukan perhitungan rute perjalanan yang lebih kompleks dan mempertimbangkan kondisi jalan secara aktual [40].

Pada penelitian ini, metode Haversine digunakan untuk menghitung jarak antara lokasi pelanggan dan lokasi penyedia jasa renovasi sebagai dasar penentuan radius layanan, yaitu 5 km, 10 km, dan 15 km. Hasil perhitungan jarak tersebut digunakan untuk menyaring penyedia jasa yang berada dalam jangkauan lokasi pelanggan, sebelum dilanjutkan dengan perhitungan rute perjalanan aktual menggunakan layanan Google Maps. Dengan demikian,

metode Haversine berperan dalam meningkatkan efisiensi dan relevansi pencarian lokasi pada sistem yang dikembangkan.

## 2.9 Blackbox Testing

*Blackbox Testing* atau sering disebut dengan pengujian fungsional adalah metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program. Tujuan utama pengujian ini adalah untuk memastikan bahwa perangkat lunak telah berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Dalam metode ini, penguji hanya mengetahui *input* dan *output* dari sistem, tanpa perlu mengetahui bagaimana proses di dalam sistem tersebut dijalankan [41].

Berikut beberapa metode yang digunakan untuk melakukan pengujian *Black Box*, yaitu [42]:

1. *Equivalence Partitioning*.

*Equivalence Partitioning* adalah teknik pengujian yang membagi himpunan *input* yang mungkin menjadi beberapa kelas ekuivalen (*equivalence class*), sehingga setiap anggota kelas dianggap memberikan hasil yang sama. Tujuannya adalah mengurangi jumlah kasus uji yang diperlukan tanpa mengurangi cakupan pengujian. Misalnya, jika sistem menerima *input* angka 1–100, maka rentang ini dapat dibagi menjadi kelas valid (1–100) dan kelas tidak valid ( $<1$  atau  $>100$ ). Dari setiap kelas, cukup diambil satu contoh sebagai *test case*.

2. *Boundary Value Analysis* (BVA).

*Boundary Value Analysis* merupakan pengembangan dari *Equivalence Partitioning* dengan fokus pada nilai batas (*boundary values*) dari suatu *input*. Hal ini karena kesalahan pada program cenderung terjadi di sekitar batas minimal, maksimal, atau transisi antar kelas *input*. Contohnya, jika *input* yang valid adalah 1–100, maka *test case* akan difokuskan pada nilai 1, 2, 99, 100, serta nilai di luar batas 0 dan 101.

3. *Syntax Testing*.

*Syntax Testing* adalah metode yang digunakan untuk menghasilkan kasus uji dari domain *input* yang telah didefinisikan secara formal, biasanya menggunakan *meta-language* seperti *Backus–Naur Form* (BNF). Metode ini memanfaatkan *Abstrak Syntax Tree* (AST) untuk memetakan struktur *input*. *Test case* yang valid dirancang untuk menutupi seluruh cabang AST, sedangkan *test case* yang tidak valid sengaja dibuat dengan kesalahan pada terminal atau *node* nonterminal AST. Pendekatan ini sangat

efektif untuk menguji sistem yang menerima *input* terstruktur, misalnya *compiler* atau *parser*.

4. *Random Testing*.

*Random Testing* adalah metode pengujian di mana *input* dihasilkan secara acak dari domain *input* yang mungkin. Tujuannya adalah menemukan kasus-kasus anomali yang tidak terpikirkan oleh penguji secara manual. Kelebihan metode ini adalah dapat menemukan *bug* tersembunyi dalam kondisi ekstrem atau tidak biasa, namun kekurangannya adalah tidak menjamin cakupan pengujian yang sistematis.

5. *Non-prescriptive Approaches to Blackbox Testing*.

Metode *non-prescriptive* ini bersifat lebih fleksibel dan tidak mengikuti aturan atau teknik formal tertentu. Pengujian dilakukan berdasarkan intuisi penguji, pengalaman sebelumnya, dan pemahaman terhadap sistem. Pendekatan ini bermanfaat ketika spesifikasi formal tidak tersedia atau sistem sangat kompleks, tetapi hasilnya cenderung lebih sulit diulang secara konsisten.

6. *Test Catalogues, Test Categories, dan Test Matrices*.

Metode ini melibatkan pendokumentasian kasus uji secara sistematis dalam bentuk katalog, kategori, atau matriks. Dengan cara ini, setiap fungsi sistem, kombinasi *input*, dan jalur logika dapat dicatat, sehingga memudahkan penguji dalam menelusuri, menjalankan, dan melacak *test case*. Matriks pengujian biasanya memetakan *input* terhadap *output* yang diharapkan, sehingga memudahkan evaluasi hasil pengujian.

7. *Combinatorial Test Methods*.

Metode ini fokus pada pengujian kombinasi *input* yang berbeda-beda untuk memastikan bahwa sistem tetap berjalan dengan benar dalam berbagai kondisi *input* yang mungkin. Teknik ini sangat berguna ketika suatu sistem memiliki banyak parameter, karena jumlah kombinasi *input* bisa sangat besar, sehingga *combinatorial testing* membantu memprioritaskan kombinasi yang paling kritis untuk diuji.

8. *Category Partition Method (CPM)*.

*Category Partition Method* digunakan untuk memformalkan dokumentasi kasus uji *Blackbox* dalam bahasa yang disebut *Test Specification Language (TSL)*. Metode ini membagi *input* ke dalam kategori dan subkategori, lalu menetapkan *test case* berdasarkan kombinasi kategori tersebut. CPM memberikan cara yang terstruktur untuk menentukan cakupan pengujian dan menghasilkan dokumen *test case* yang lengkap dan dapat direproduksi.

9. *Classification Trees*.

*Classification Trees* adalah metode visual untuk merancang *test case*. *Input* dibagi ke dalam pohon klasifikasi, di mana setiap cabang mewakili kategori atau atribut *input*, dan kombinasi cabang membentuk *test case*. Keuntungan dari metode ini adalah mempermudah pemahaman hubungan antar *input*, mendukung *coverage* yang lebih baik, dan memudahkan pengelolaan kasus uji secara sistematis.

Pada penelitian ini, metode *Blackbox Testing* yang digunakan adalah *Non-prescriptive Approaches to Blackbox Testing* dikarenakan pendekatan ini lebih fleksibel dan memungkinkan pengujian berfokus pada skenario fungsional atau *Use Case* sistem tanpa harus mengikuti aturan atau teknik formal tertentu. Metode ini sangat sesuai untuk memastikan bahwa setiap alur pengguna atau fungsi utama sistem berjalan sesuai spesifikasi dan kebutuhan, terutama ketika sistem memiliki berbagai kondisi *input* yang kompleks atau tidak sepenuhnya terdokumentasi secara formal.

## **2.10 Penyedia Jasa Renovasi**

Pada bagian ini akan dijelaskan mengenai penyedia jasa renovasi yang memiliki peran penting dalam kegiatan pembangunan maupun perbaikan bangunan. Pembahasan ini mencakup pengertian penyedia jasa renovasi, jenis-jenis layanan yang ditawarkan, serta peranan dan tanggung jawabnya dalam suatu proyek. Dengan memahami konsep penyedia jasa renovasi, diharapkan pembaca dapat mengetahui bagaimana proses renovasi dijalankan secara profesional dan terorganisir.

### **2.10.1 Pengertian Penyedia Jasa Renovasi**

Penyedia jasa renovasi adalah individu, kelompok, atau perusahaan yang menyediakan layanan perbaikan, pembangunan ulang, atau penataan ulang suatu bangunan, baik rumah tinggal, kantor, maupun fasilitas komersial. Layanan ini tidak hanya mencakup pelaksanaan fisik renovasi, tetapi juga dapat meliputi konsultasi desain, perencanaan proyek, pengadaan material, serta manajemen pengerjaan agar hasil sesuai dengan keinginan pelanggan [43].

Selain itu, penyedia jasa renovasi dapat dibedakan berdasarkan skala dan bidang keahliannya. Ada penyedia jasa yang menangani proyek kecil, seperti perbaikan rumah atau interior, hingga penyedia jasa berskala besar yang mampu menangani renovasi gedung, kantor, atau proyek komersial. Dengan semakin berkembangnya teknologi dan kebutuhan masyarakat, penyedia jasa renovasi kini juga mulai memanfaatkan platform digital, seperti

aplikasi berbasis *web*, untuk mempermudah proses pemesanan, komunikasi, dan manajemen proyek [44].

Dengan kata lain, penyedia jasa renovasi berfungsi sebagai penghubung antara kebutuhan pelanggan dan keahlian teknis tenaga kerja, sekaligus sebagai pihak yang menjamin kualitas, keamanan, dan efisiensi proyek renovasi. Pemahaman tentang pengertian ini menjadi dasar penting dalam pengembangan aplikasi berbasis *web* untuk mempermudah interaksi antara penyedia jasa dan pelanggan [45].

### 2.10.2 Jenis-Jenis Penyedia Jasa Renovasi

Penyedia jasa renovasi dapat dikategorikan berdasarkan bidang keahlian mereka. Berdasarkan data yang ada, jenis-jenis penyedia jasa renovasi antara lain [44]:

1. **Tukang AC.**

Penyedia jasa ini spesialis dalam pemasangan, perbaikan, dan perawatan pendingin udara (AC). Layanan yang ditawarkan mencakup instalasi unit AC baru, perbaikan kerusakan, serta maintenance rutin untuk memastikan AC berfungsi optimal.

2. **Tukang CCTV.**

Penyedia jasa yang fokus pada pemasangan dan perawatan sistem kamera pengawas (CCTV). Layanan ini meliputi instalasi perangkat, konfigurasi sistem keamanan, dan pemeliharaan perangkat agar dapat beroperasi secara efektif.

3. **Tukang elektronik.**

Penyedia jasa yang menangani perbaikan dan instalasi perangkat elektronik, seperti televisi, peralatan rumah tangga, atau sistem elektronik lainnya. Keahlian mereka membantu pelanggan memperbaiki perangkat yang rusak atau melakukan instalasi baru.

4. **Tukang listrik.**

Penyedia jasa yang spesialis dalam instalasi, perbaikan, dan pemeliharaan sistem kelistrikan bangunan. Layanan mencakup pemasangan kabel, panel listrik, lampu, saklar, dan memastikan instalasi listrik aman dan sesuai standar.

5. **Tukang bangunan.**

Penyedia jasa ini menangani pekerjaan konstruksi dan renovasi bangunan secara umum, termasuk pembangunan dinding, lantai, atap, plafon, dan perbaikan struktural. Tukang bangunan bertugas memastikan struktur bangunan aman, kokoh, dan sesuai desain yang direncanakan. Mereka juga dapat bekerja sama dengan tukang spesialis lainnya untuk menyelesaikan proyek renovasi secara menyeluruh.

### 2.10.3 Proses Bisnis Jasa Renovasi

Proses bisnis penyedia jasa renovasi secara umum melibatkan beberapa tahap, yaitu [44]:

1. Permintaan pelanggan.

Pelanggan mengajukan permintaan renovasi, bisa berupa konsultasi awal atau pengajuan desain dan estimasi biaya.

2. Estimasi biaya dan waktu pengerjaan.

Penyedia jasa menilai kebutuhan renovasi, menghitung biaya material dan tenaga kerja, serta menentukan lama pengerjaan.

3. Kesepakatan kontrak.

Kedua pihak menyetujui harga, jadwal, dan spesifikasi pekerjaan melalui kontrak resmi untuk menghindari sengketa.

4. Pelaksanaan renovasi.

Tim penyedia jasa melakukan pengerjaan renovasi sesuai dengan kontrak yang telah disepakati, termasuk koordinasi dengan subkontraktor atau penyedia material.

5. Evaluasi dan penyerahan hasil.

Setelah selesai, pekerjaan dievaluasi untuk memastikan kualitas sesuai standar. Kemudian, hasil renovasi diserahkan kepada pelanggan dan dilakukan pembayaran akhir.

Proses ini menjadi dasar alur fitur pada aplikasi berbasis *web* yang akan dikembangkan, seperti sistem estimasi biaya, *tracking proyek*, dan konfirmasi pekerjaan.

### 2.10.4 Permasalahan pada Penyedia Jasa Renovasi Konvensional

Penyedia jasa renovasi konvensional menghadapi beberapa permasalahan yang menyebabkan layanan tidak efisien dan kurang transparan. Berdasarkan kondisi di Kota Medan dan pengalaman masyarakat, permasalahan utama antara lain [5] [46]:

1. Sulit menemukan tenaga profesional terpercaya.

Pelanggan sering kali kesulitan mencari tukang atau penyedia jasa yang memiliki keahlian sesuai kebutuhan. Rekomendasi biasanya masih bergantung pada jaringan pribadi atau iklan daring, sehingga waktu pencarian menjadi lebih lama dan kualitas tenaga kerja tidak selalu terjamin.

2. Perbedaan harga dan transparansi biaya yang minim.

Biaya renovasi sering kali tidak transparan dan dapat berbeda secara signifikan antara satu penyedia jasa dengan penyedia jasa lain. Hal ini menimbulkan kebingungan bagi

pelanggan dalam menentukan pilihan, sekaligus meningkatkan risiko pembengkakan biaya selama proyek berjalan.

3. Komunikasi yang kurang efektif.

Selama proses renovasi, komunikasi antara pelanggan dan penyedia jasa sering tidak lancar. Hal ini menyebabkan keterlambatan, perbedaan hasil yang diharapkan, dan ketidaksesuaian antara rencana awal dengan hasil akhir.

4. Kurangnya pemantauan dan kontrol proyek.

Sebagian besar renovasi konvensional tidak menyediakan sistem pemantauan pekerjaan secara *real-time*. Pelanggan harus mengawasi langsung di lokasi, yang menambah beban waktu dan tenaga serta menurunkan efisiensi pengerjaan proyek.

5. Risiko kualitas pekerjaan yang tidak konsisten.

Tanpa adanya mekanisme verifikasi penyedia jasa, pelanggan berisiko mendapatkan hasil yang tidak sesuai kesepakatan atau standar yang diinginkan. Hal ini menurunkan tingkat kepercayaan terhadap penyedia jasa dan membuat pengalaman renovasi menjadi kurang memuaskan.

6. Kurangnya jaminan layanan dan perlindungan pelanggan.

Layanan konvensional biasanya tidak menyediakan garansi atau perlindungan terhadap kerusakan setelah pekerjaan selesai. Jika terjadi masalah, pelanggan harus menghubungi penyedia jasa secara manual dan seringkali tidak mendapatkan tanggapan yang memadai.

Dengan permasalahan-permasalahan tersebut, jelas bahwa penyedia jasa renovasi konvensional belum mampu memenuhi kebutuhan pelanggan secara efisien, transparan, dan terukur. Hal inilah yang mendorong perlunya pengembangan aplikasi berbasis *web* yang dapat menjembatani interaksi antara pelanggan dan penyedia jasa dengan lebih profesional, terkontrol, dan terpercaya

### 2.10.5 Peluang Digitalisasi Layanan Jasa Renovasi

Digitalisasi layanan renovasi memberikan peluang signifikan bagi penyedia jasa maupun pelanggan, antara lain [44]:

1. Kemudahan pencarian dan pemesanan layanan.

Pelanggan dapat menemukan penyedia jasa sesuai kebutuhan, lokasi, dan *rating* melalui aplikasi *web*.

2. Efisiensi proses transaksi.

Estimasi biaya, jadwal pengerjaan, dan pembayaran dapat dilakukan secara *online* sehingga mengurangi waktu dan tenaga.

3. Peningkatan kepercayaan pelanggan.

Fitur *rating*, *review*, dan portofolio pekerjaan memungkinkan pelanggan memilih penyedia jasa yang sudah terbukti kualitasnya.

4. Ekspansi pasar.

Digitalisasi memungkinkan penyedia jasa menjangkau lebih banyak pelanggan di luar wilayah lokal.

5. Integrasi sistem manajemen proyek.

Aplikasi *web* dapat menyediakan *dashboard* untuk *tracking proyek*, manajemen material, dan laporan pengerjaan secara *real-time*.

Selain itu, digitalisasi layanan renovasi membuka peluang bagi pengembangan platform berbasis *web* dan *e-commerce* yang lebih terstruktur. Dengan memanfaatkan teknologi web, penyedia jasa tidak hanya dapat menampilkan profil, layanan, dan portofolio mereka secara *online*, tetapi juga mengelola pemesanan, pembayaran, dan komunikasi dengan pelanggan secara terintegrasi. Pendekatan ini memungkinkan terciptanya ekosistem renovasi yang efisien, transparan, dan mudah diakses, sehingga pelanggan dapat memperoleh layanan berkualitas tanpa harus datang langsung ke lokasi, sementara penyedia jasa dapat mengoptimalkan manajemen proyek dan ekspansi pasar secara lebih profesional.

*Web* penyedia jasa renovasi merupakan platform digital yang memungkinkan pelanggan mencari, memilih, dan memesan layanan renovasi secara *online*. Berdasarkan latar belakang sebelumnya, banyak masyarakat mengalami kesulitan dalam menemukan tenaga profesional yang terpercaya, menghadapi perbedaan harga yang signifikan, serta kurangnya transparansi dan pemantauan proyek. *Web* penyedia jasa renovasi hadir sebagai solusi untuk mengatasi masalah-masalah tersebut.

Melalui *web*, pelanggan dapat dengan mudah menelusuri profil penyedia jasa, melihat portofolio pekerjaan, membaca ulasan dari pengguna sebelumnya, dan memanfaatkan fitur *rating* untuk menilai kualitas layanan. Selain itu, estimasi biaya, jadwal pengerjaan, dan sistem pembayaran dapat dilakukan secara *online*, sehingga mengurangi waktu dan tenaga yang sebelumnya dibutuhkan untuk koordinasi secara langsung. Fitur notifikasi dan *tracking proyek* memungkinkan pelanggan memantau progres renovasi secara *real-time*, sehingga komunikasi antara pelanggan dan penyedia jasa lebih efektif dan risiko kesalahpahaman dapat diminimalkan.

Dengan kata lain, *web* penyedia jasa renovasi tidak hanya mempercepat proses pencarian dan pemesanan layanan, tetapi juga meningkatkan kepercayaan pelanggan serta efisiensi pengerjaan proyek. Platform ini menjadi jembatan digital yang mengintegrasikan kebutuhan pelanggan dan kemampuan penyedia jasa, sesuai dengan kebutuhan yang muncul akibat pertumbuhan populasi, meningkatnya permintaan renovasi, dan tantangan layanan konvensional yang telah diuraikan dalam latar belakang

