

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 digunakan yaitu:

Literatur 1 : (Janis *et al.*, 2020), dengan judul “Rancang Bangun Aplikasi Online Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi”, pada penelitian ini adapun permasalahan yang dibahas adalah bagaimana membuat sebuah aplikasi online sistem pemesanan jasa tukang bangunan untuk mengatasi permasalahan masyarakat dalam mencari jasa tukang bangunan. Pada penelitian ini menggunakan Android Studio sebagai platform yang membuat aplikasi pada perangkat smartphone android dengan jenis ekstensi APK dan XAPK, menggunakan *Firestore* sebagai manajemen *database*, dan menggunakan metode *Model Rapid Application Development (RAD)*. Penelitian ini menghasilkan sebuah aplikasi online pemesanan jasa tukang bangunan berbasis lokasi yang dapat membantu mempermudah masyarakat dalam mencari jasa tukang bangunan dan penyedia jasa tukang bangunan untuk mendapatkan pekerjaan.

Literatur 2 : (Ramadhan *et al.*, 2019), dengan judul “Rancang Bangun Aplikasi Pemesanan Jasa Tukang Bangunan Berbasis Android (Kabupaten Manggarai Barat)”, pada penelitian ini permasalahan yang dibahas adalah bagaimana mengatasi kesusahan masyarakat di kabupaten Manggarai Barat

dalam mencari jasa tukang bangunan, dimana masyarakat tersebut melakukan pencarian secara manual dengan pergi kesuatu tempat untuk menanyakan jasa tuakng bangunan. Tujuan dari penelitian ini adalah untuk membantu masyarakat di kabupaten Manggarai Barat dengan cepat dan efisien melalui sebuah aplikasi berbasis *Android* yang dapat dilakukan melalui smartphone. Metode yang digunakan dalam penelitian sistem ini adalah metode *waterfall*. Hasil penelitian ini berupa sebuah aplikasi pemesanan jasa tukang bangunan berbasis *android* yang berguna untuk membantu masyarakat di kabupaten Manggarai Barat dalam mencari jasa tukang bangunan.

Literatur 3 : (Wijaksono *et al.*, 2019), dengan judul “Aplikasi Golek Tukang untuk Pencarian Jasa Perbaikan Rumah di Daerah Surabaya Berbasis Android”. Pada penelitian ini permasalahan yang dibahas adalah masyarakat Kota Surabaya masih menggunakan sistem manual dengan cara mencari informasi dari mulut ke mulut sehingga masyarakat kota Surabaya mengalami kesusahan untuk dapat memenuhi kebutuhan perbaikan rumah yang sesuai dengan masalah yang terjadi. Tujuan dari penelitian ini adalah supaya masyarakat Kota Surabaya dapat mencari tenaga jasa perbaikan rumah dengan mudah, cepat, dan terpercaya melalui sebuah aplikasi yang berbasis *android*. Aplikasi ini dibuat menggunakan bahasa aplikasi *Android Studio*, pemrograman PHP dan Java, database yang digunakan adalah MySQL, dan library yang digunakan adalah Retrofit. Hasil penelitian ini berupa sebuah aplikasi pencarian jasa perbaikan rumah berbasis *android* yang dinamakan “Golek Tukang”.

Literatur 4 : (Setiawan, 2022), dengan judul “Perancangan Sistem Aplikasi Pemesanan Jasa Kerja Online Berbasis Android”. Pada penelitian ini permasalahan yang dibahas adalah banyak penyedia pihak jasa masih bersifat konvensional, belum banyak sistem pemesanan jasa berbasis *mobile* atau *android*, sehingga dapat diartikan banyak konsumen yang mengalami kesusahan mencari jasa pekerjaan. Tujuan dari penelitian ini adalah untuk membantu mempermudah pengguna dalam mencari penyedia jasa pekerjaan sesuai kebutuhan. Aplikasi ini dirancang menggunakan bahasa pemrograman *android* dengan database *mysql* dan metode *waterfall*. Hasil penelitian ini berupa sebuah aplikasi pemesanan jasa kerja online berbasis *android*.

Literatur 5 : (Hidayatullah, 2021), dengan judul “ Pengembangan Aplikasi Pemesanan Jasa Tukang Berbasis Android”. Pada penelitian ini permasalahan yang dibahas mengenai masyarakat yang berada di daerah perkotaan masih mengalami kesulitan dalam menemukan pekerja tukang yang mereka perlukan dikarenakan belum banyak aplikasi yang menghubungkan antara konsumen dan penyedia jasa tukang. Tujuan dari penelitian ini adalah untuk mengembangkan aplikasi yang bernama TukangPedia. Metode penelitian yang digunakan dalam penelitian ini adalah penelitian pengembangan dengan model prototype. Dalam pembuatan aplikasi ini penulis menggunakan basis pemrograman android. Hasil dari penelitian ini yaitu berupa prototype aplikasi TukangPedia dan aplikasi TukangPedia.

2.2 Sistem Informasi

Menurut (Canggih Ajika Pamungkas, 2017) dalam *ebook* pengantar dan implementasi basis data, sistem informasi adalah suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang, fasilitas, teknologi, media, prosedur, dan kontrol yang digunakan untuk memelihara jalur komunikasi yang kritis, menangani beberapa jenis urusan tertentu secara rutin, memberikan informasi kepada manajemen, dan lainnya terhadap peristiwa yang berkaitan dengan faktor internal dan eksternal yang penting dalam memberikan dasar yang informatif untuk pengambilan keputusan.

2.3 Tukang Bangunan

Menurut (Janis *et al.*, 2020) dalam jurnal yang berjudul "*Rancang Bangun Aplikasi Online Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi*", definisi dari tukang bangunan adalah pekerja yang ahli dalam pembangunan toko, rumah, dan bangunan lainnya. Pada umumnya tukang bangunan disebut tukang batu. Tergantung pada kesepakatan antara tukang dan orang yang menggunakan jasanya, upah tukang bangunan dapat ditentukan, baik harian maupun borongan. Pada saat ini, pekerjaan seorang tukang bangunan tidak hanya membangun sebuah bangunan. Mereka juga dapat merenovasi bangunan yang sudah terbangun, mengukir bangunan, dan mengecat bangunan yang sudah dibangun. Pelaksana bertanggung jawab atas pembangun dan berfungsi sebagai direktur dan manajer pekerjaan proyek. Ada beberapa bagian berbeda dalam tukang bangunan, yaitu :

1. Tukang Batu adalah orang yang bekerja pasang batu bata, pasang pondasi batu kali dan pekerjaan cor (Untuk pekerjaan kasar).
2. Tukang Plaster Aci adalah tukang yang bekerja merapikan pasangan batu bata dengan campuran semen dan pasir (Untuk pekerjaan halus).
3. Tukang Pembesian adalah tukang yang bekerja merangkai besi di proyek, biasanya bila tukang spesialis ini tidak bisa untuk pekerjaan pasangan batu bata, plaster dan aci.
4. Tukang Profil adalah tukang yang bekerja membuat motif pada tampak depan, biasanya tukang ini mempunyai tingkat keahlian yang lebih tinggi dibanding tukang batu.
5. Tukang Keramik adalah yang bekerja untuk memasang keramik, tukang ini kebanyakan hanya menguasai bidang ini saja.
6. Tukang Batu Alam adalah tukang yang bekerja untuk memasang batu alam biasanya spesialis dan tidak bisa mengerjakan pekerjaan yang lain.
7. Tukang Marmer adalah tukang yang ahli dalam pemasangan marmer atau granit biasanya mempunyai keahlian di bidang ini saja dan tidak dimiliki tukang yang lain pada bagian dalam tukang bangunan yang lain.

2.4 Aplikasi Mobile

Dalam sebuah artikel jurnal yang berjudul “Aplikasi Mobile Zagiyan (Zaringan Digital Nelayan) Dalam Menunjang Produktivitas Dan Keselamatan, Dan Kesehatan Nelayan (Studi Kasus Kelompok Nelayan Percut)” Buyens (2001) menyatakan bahwa istilah aplikasi mobile berasal

dari kata "*application*" dan "mobile". *Mobile* dapat didefinisikan sebagai perpindahan dari satu lokasi ke lokasi lain, sedangkan aplikasi atau *application* adalah program siap pakai yang dirancang untuk melakukan fungsi bagi pengguna atau aplikasi lain dan dapat digunakan oleh target yang dituju. Menurut Purnama (2010), aplikasi mobile merujuk pada aplikasi yang berjalan pada sebuah perangkat *mobile device* karena kata "mobile" berarti "bergerak" atau "berpindah" dimana didalamnya terdapat hiburan, bisnis, *study*, pekerjaan kantor, *browsing*, dan sebagainya dengan memanfaatkan aplikasi *mobile*. Pengguna ponsel lebih memilih menggunakan aplikasi *mobile* untuk hiburan karena memudahkan pengguna untuk bersenang-senang dan menikmati hiburan setiap saat dengan memanfaatkan fitur pemutar video, pemutar musik dan permainan. Dalam hal ukuran dan desain, perangkat seluler memiliki banyak variasi, tetapi sangat berbeda dari sistem desktop. (Harumy & Amrul, 2018)

2.5 Kodular

Menurut (Pramana, 2020) pengertian kodular adalah situs web yang menawarkan sumber daya untuk membangun aplikasi Android menggunakan pemrograman blok yang sebanding dengan MIT App Inventor. Dengan kata lain, membuat aplikasi Android pengguna tidak perlu memasukkan kode komputer secara manual. Kodular Store dan Kodular Extension IDE, dua fitur yang dapat mempermudah pengembang untuk mengirimkan aplikasi Android ke Kodular Store dan membuat blok program ekstensi IDE sesuai dengan kebutuhan pengembang, merupakan salah satu keunggulan Kodular.

2.6 Firebase Database

Definisi Firebase menurut (Rosyana Fitria Purnomo, Onno W. Purbo, n.d.) dalam buku yang berjudul “Firebase: Membangun Aplikasi Berbasis Android” merupakan layanan milik Google yang membantu pengembang aplikasi membuat aplikasi mereka dengan lebih mudah. Google menyediakan alat seperti Backend as a Service dan Firebase Cloud Service Provider untuk memudahkan pengembang membuat aplikasi seluler dan online. Dengan Firebase, pengembang aplikasi dapat berkonsentrasi untuk membuat aplikasi daripada harus menghabiskan banyak waktu untuk membuat infrastruktur dan fungsionalitas backend dari awal. Ini memungkinkan pengembang untuk membuat aplikasi berkualitas tinggi dengan cepat.

2.7 Extreme Programming

Menurut (Project Management Institute, 2017) dalam buku yang berjudul “Agile Practice Guide” , pengertian dari *extreme programming* merupakan teknik pengembangan perangkat lunak yang lebih cepat dibandingkan dengan teknik yang lain sehingga menghasilkan kualitas perangkat lunak yang lebih baik, lebih responsif terhadap perubahan permintaan klien, dan rilis yang lebih sering dalam waktu yang lebih singkat.

2.8 Unified Modeling Language (UML)






Menurut (Hendini, 2016), Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang digunakan untuk dokumentasi, spesifikasi, dan pengembangan perangkat lunak. Selain sebagai alat untuk memfasilitasi

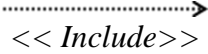
pengembangan sistem, UML menyediakan teknik untuk membuat sistem berbasis objek. Dalam UML terdapat tipe-tipe diagram UML yaitu :

1. *Use Case Diagram*

Use case diagram merupakan pemodelan yang digunakan untuk membuat perilaku sistem informasi dalam menentukan fungsionalitas mana yang tersedia dalam sistem informasi dan siapa yang berwenang untuk menggunakannya. Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada tabel 2.1 di bawah ini.

Table 2. 1 Simbol Use Case Diagram




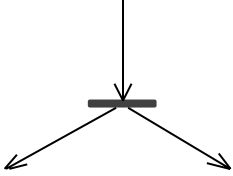
Gambar	Keterangan
<p>Use Case</p> 	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.</p>
<p>Actor</p> 	<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p>
<p>Asosiasi/association</p> 	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.</p>
<p>Generalization</p> 	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
<p>Ekstensi (<i>extend</i>)</p>  <p><<Extend>></p>	<p>Extend, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

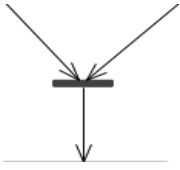
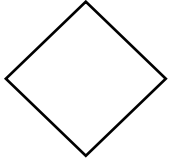

Include 	Include, merupakan di dalam <i>use case</i> lain (required) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
---	---

2. Activity Diagram (Diagram Aktivitas)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam activity Diagram yaitu:

Table 2. 2 Simbol Activity Diagram



Gambar	Keterangan
Status Awal Start Point 	Status awal atau <i>start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
Status Akhir End Point 	Status akhir atau <i>End Point</i> , merupakan akhir aktivitas.
Aktivitas Activities 	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis.
Percabangan Fork 	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.



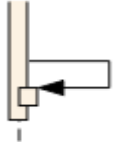


<p>Penggabungan Join</p> 	<p><i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukkan adanya dekomposisi.</p>
<p>Decision Points</p> 	<p><i>Decision points</i>, menggambarkan pilihan untuk pengambilan keputusan, true atau false.</p>
<p>Swimline</p> 	<p><i>Swimlane</i>, pembagian activity diagram untuk menunjukkan siapa melakukan apa.</p>

3. Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Table 2. 3 Simbol Sequence Diagram

Gambar	Keterangan
<p>Entity Class</p> 	<p><i>Entity Class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
<p>Boundary Class</p> 	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.</p>

<p>Control class</p> 	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
<p>Message</p> 	<p><i>Message</i>, simbol mengirim pesan antar class.</p>
<p>Recursive</p> 	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
<p>Activation</p> 	<p><i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.</p>
<p>Lifeline</p> 	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation</p>

4. Diagram Kelas (*Class Diagram*)

Diagram kelas merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalisation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan

visibility, tingkat akses objek eksternal kepada suatu operasi atau atribut.

2.9 Pengujian Standar Kualitas ISO-9126

Menurut (Pamungkas, 2018), ISO 9126 merupakan salah satu *framework* standar internasional untuk menilai kualitas perangkat lunak yang dikembangkan oleh *International Electrotechnical Commission* (IEC) dan *International Organization for Standardization* (ISO). Standar internasional dapat membangun dan menentukan kualitas produk perangkat lunak, termasuk kualitas, karakteristik, model, dan metrik yang terkait. Model ISO 9126 memiliki 6 (enam) faktor dan beberapa subfaktor. Keenam karakteristik model ISO 9126 antara lain :

1. *Functionality* (Fungsionalitas) adalah kemampuan perangkat lunak untuk memberikan fitur sesuai dengan kebutuhan pengguna saat digunakan dalam keadaan tertentu.
2. *Reliability* (*Kehandalan*) adalah kemampuan perangkat lunak untuk tetap berkerja dengan baik saat digunakan dalam keadaan tertentu. Jumlah input yang gagal saat perangkat lunak sedang beroperasi dapat digunakan untuk mengukur keandalannya.
3. *Usability* (*Kebergunaan*) adalah faktor yang menentukan apakah suatu produk dapat berhasil, efisien sesuai dengan kebutuhan pengguna dalam memenuhi tujuan tertentu. Memanfaatkan alat bergaya kuesioner, elemen usability dapat dinilai.
4. *Efficiency* (*Efisiensi*) merupakan faktor yang mengacu pada kapasitas perangkat lunak untuk memberikan kinerja yang efisien sesuai dengan

volume sumber daya yang digunakan.

5. *Maintainability (Pemeliharaan)* merupakan faktor yang berkaitan dengan kemampuan perangkat lunak untuk membangun, melakukan perbaikan, adaptasi terhadap perubahan lingkungan, persyaratan, standar, dan portabilitas (portabilitas mengacu pada kemampuan perangkat lunak untuk berfungsi atau bekerja dalam banyak pengaturan) adalah contoh modifikasi.