

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Beberapa penelitian terdahulu yang dijadikan acuan dalam penelitian ini adalah

1. Kristanto (2020) meneliti tentang Analisa Dan Perancangan Sistem Informasi E-Booking Property Berbasis Android. Banyaknya pengguna android smartphone, merupakan peluang yang sangat besar bagi para developer untuk memberikan solusi dalam segala permasalahan bisnis yang saat ini sedang berjalan demi mendukung perkembangan bisnis yang sangat cepat dan tepat. Saat ini masih banyak kita jumpai hampir semua konsumen dalam melakukan pencarian rumah dan pemesanannya diharuskan untuk mendatangi langsung ke tempat lokasi, hal ini dikarenakan belum adanya teknologi yang digunakan untuk mempermudah pelanggan dalam mencari informasi mengenai perumahan. Rumah merupakan kebutuhan primer yang bersifat primer selain kebutuhan pangan dan sandang. Dari tingginya minat konsumen akan memiliki suatu rumah, hal ini juga merupakan kesempatan besar bagi para pengusaha properti untuk membangun rumah yang layak dan terjangkau yang kemudian ditawarkan keberbagai kalangan publik. Untuk mengatasi masalah tersebut, dibutuhkan analisa dan perancangan aplikasi berbasis android agar bisnis dapat berjalan dengan lebih lancar. Aplikasi E-Booking Property merupakan aplikasi berbasis android yang dirancang untuk mampu menyediakan informasi yang lebih lengkap dan akurat kepada konsumen tentang perumahan dan ketersediaannya. Aplikasi

ini juga disematkan fitur booking agar dapat diakses dimanapun dan kapanpun oleh konsumen tanpa harus datang ke tempat lokasi, sehingga tidak memakan banyak waktu. Selain menguntungkan bagi konsumen, aplikasi ini juga sangat membantu bagi pengusaha properti dalam meningkatkan penjualan serta memudahkan dalam membuat laporan booking di setiap transaksinya karena sistem sudah terkomputerisasi dan terintegrasikan dengan baik. Perancangan sistem informasi ini menggunakan metode SDLC sedangkan analisa proses bisnisnya menggunakan SWOT.

2. Oktavia *et al.* (2020) meneliti tentang Aplikasi Booking Online Untuk Meningkatkan Kepuasan Pelanggan Dan Mengoptimalkan Customer Relationship Management (CRM). Untuk memudahkan pemesanan pada K-cube family karaoke maka penulis membangun program aplikasi website yang diharapkan dapat memberikan dampak yang sangat penting dalam mendapatkan informasi yang akurat dan cepat dengan meningkatkan kinerja aplikasi pemesanan dalam mewujudkan visi misi K-cube family karaoke. Metodologi yang penulis gunakan dalam pembuatan aplikasi Booking Online adalah waterfall yang meliputi analisa kebutuhan, desain sistem, penulisan kode program, pengujian program, penerapan program dan pemeliharaan atau pengembangan versi pemesanan dapat diharapkan membantu dalam perancangan sistem ini serta desain penelitian dengan menggunakan metode observasi dan wawancara. Alat yang digunakan untuk menggambarkan model sistem adalah use case diagram, activity diagram, class diagram, serta perancangan dalam basis data menggunakan MySQL database. Dengan adanya sistem informasi Booking Online maka

diharapkan dapat meningkatkan kepuasan pelanggan dan mengoptimalkan customer relationship management di K-cube family karaoke memberikan informasi yang tepat mengenai data room, kritik dan saran, testimoni yang di update oleh customer sebelumnya.

3. Fauzi, Rahmatuloh and Resdiana (2021) meneliti tentang Rancang Bangun Aplikasi Online Booking Pada Dankie Barbershop Berbasis Website Menggunakan Web Framework Dan Payment Gateway. Barbershop khusus pria saat ini sedang menjadi trend dan sudah mulai banyak ditemukan diberbagai daerah, banyak pria yang mendatangi Barbershop khusus pria yang dapat menyediakan potongan rambut terbaru yang cocok untuk para pria. Dankie barbershop selalu mengedepankan layanan jasa yang berkualitas kepada konsumen, namun sayangnya sistem bookingnya masih manual, sehingga customer diharuskan menunggu dengan waktu yang cukup lama jika saat ada antrian. Implementasi aplikasi online booking dankie barbershop ini diterapkan berbasis web dengan tujuan agar operasional dankie barbershop dapat berjalan lebih efektif, aman, cepat, dan akurat. Dengan aplikasi ini data transaksi barbershop menjadi terkomputerisasi, sehingga data dapat tersimpan dengan baik. Aplikasi ini dikembangkan menggunakan framework codeigniter dengan bahasa pemrograman PHP dan menggunakan database MySQL. Dari hasil pengujian menggunakan metode blackbox dapat disimpulkan bahwa aplikasi online booking barbershop ini dapat membantu proses booking pada dankie barbershop.

4. Saputra and Pratama (2021) meneliti tentang Sistem Booking Foto, Video, Dan Undangan Digital Berbasis Web Pada Recollection Project Semarang. Seiring berkembangnya teknologi saat ini hampir seluruh negara sudah mengikutiperkembangan zaman, penggunaan teknologi informasi menjadi sangat penting bagi masyarakat diseluruh dunia. Internet sangat efektif sekali untuk menyampaikan informasi apapun tidak terkecuali di bidang-bidang usaha seperti jasa fotografer. Recollection Project Semarang adalah sebuah badan usaha yang bergerak dalam penyediaan jasa foto, prewedding, wedding, videoshooting, dan undangan digital. Metode pengembangan yang digunakan dalam penelitian ini adalah Metode Waterfall dan Penelitian yang dilakukan menghasilkan sistem booking foto, video dan undangan digital berbasis web pada Recollection Project Semarang yang memberikan kemudahan dalam pengelolaan transaksi pemesanan jasa foto dan memudahkan pemesanan secara online.
5. Kurnia (2020) meneliti tentang Rancang Bangun Aplikasi E-Booking Futsal Center Kota Bukittinggi. Aplikasi E-Booking ini bertujuan untuk mengelola penyewaan lapangan futsal dikota Bukittinggi yang nantinya akan memberikan kemudahan kepada pelanggan dalam pemesanan lapangan futsal yang ada di Kota Bukittinggi. Sistem yang dirancang ini juga bekerjasama dengan pengelola lapangan futsal seluruh Kota Bukittinggi yang akan memudahkan pengelola lapangan futsal dalam mengontrol pemesanan lapangan futsal karena report yang dihasilkan telah lengkap. Aplikasi E-Booking dirancang menggunakan bahasa pemrograman Php

dengan memanfaatkan database MySQL sebagai media penyimpanan dengan tingkat keamanan yang cukup baik.

Perbedaan penelitian dengan peneliti yaitu pada sistem yang dibangun dapat mengelola transaksi jasa pemesanan jahit kebaya berbasis *mobile* dengan metode pengembangan sistem menggunakan *prototype* dan pengujian sistem menggunakan *ISO 25010*

2.2. Sistem Informasi

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan (Tohari, 2017).

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Kristanto, 2018).

2.3. Pengertian *Booking*

Booking yaitu pemesanan fasilitas yang diantaranya akomodasi, meal, seat pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, night club, *discoutegue* dan sebagainya (Suartana, 2017).

Pemesanan adalah proses, perbuatan, cara memesan (tempat, barang, dsb) kepada orang lain (Sentosa and Thamrin, 2015).

Berdasarkan pengertian para ahli diatas dapat disimpulkan bahwa pemesanan adalah suatu aktifitas yang dilakukan oleh konsumen sebelum membeli atau melakukan perbaikan untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik.

2.2.1. Jenis Booking

Pemesanan dibagi menjadi dua jenis yaitu pemesanan online dan pemesanan offline :

1. Pemesanan *Online*.

Kemajuan teknologi saat ini mengakibatkan sistem pemesanan juga mengalami perkembangan kearah sistem pemesanan online. Pemesanan online bisa diakses oleh siapapun dan dimanapun mereka berada yang memiliki akses internet.

2. Pemesanan *Offline*

Sistem pemesanan yang menggunakan pengiriman pemesanan langsung ke tempat dengan media pemesanan seperti telepon, *fax*, *e-mail*, dan *walk in*.

2.4.E-booking

E-booking atau *electronic booking* adalah suatu cara untuk mendokumentasikan proses penjualan dari aktifitas pelanggan tanpa harus mengeluarkan dokumen berharga secara fisik. Semua informasi mengenai

electronic booking disimpan secara digital dalam sistem komputer milik perusahaan (Sentosa and Thamrin, 2015).

2.5. Jasa Menjahit

Jasa menjahit adalah usaha seseorang yang pekerjaannya menjahit dan yang memberikan pelayanan berupa jasa menjahit pakaian seperti kemeja, celana, rok, atau jas, baik untuk laki-laki maupun perempuan (Novianti and Setiawan, 2019).

2.6. Kebaya

Kebaya merupakan pakaian khas perempuan Indonesia bahkan di Era 1940an presiden soekarno memilih kebaya sebagai pakaian nasional. Secara etimologi dilihat dari katanya kebaya berasal dari kata abaya yang dalam bahasa arab berarti pakaian yang menyatakan kebaya berasal dari China lalu menyebar ke Malaka, Jawa, Bali, Sumatera, dan Sulawesi. Setelah akulturasi yang berlangsung ratusan tahun, pakaian itu diterima di budaya dan norma setempat. Dengan semakin berkembangnya jaman dan dunia mode, kebaya bali pun mulai berkembang. Motif dan warna pun tidak selalu monoton atau polos-polos saja tetapi mulai banyak yang melakukan modifikasi. Hal itu dilakukan semata-mata agar si pemakai nyaman menggunakan dan tak terlepas dari unsur manusianya yang ingin tampil menarik. Tidak hanya remaja saja tetapi juga orang-orang tua mulai mencari jenis-jenis kebaya yang dianggap menarik, tanpa mengesampingkan kesopanan di dalam berpakaian (Wahyudi *et al.*, 2016).

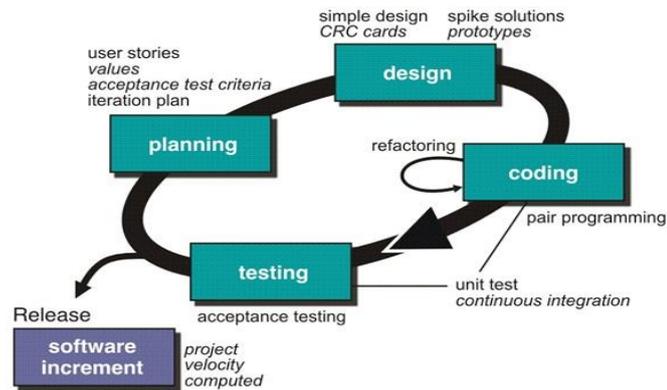
2.7. Metode Pengembangan *Extreme Programming*

XP adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan di mana persyaratan pelanggan baru dapat diadopsi (Pressman, 2012).

Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah testing. Unsur-unsur lain dari *Extreme Programming* meliputi *paired programming* pada tahapan *coding*, unit testing pada semua kode, penghindaran pemrograman fitur kecuali benar-benar diperlukan, struktur manajemen yang datar, kode yang sederhana dan jelas, dan seringnya terjadi komunikasi antara programmer dan pelanggan ketika terjadi perubahan kebutuhan pelanggan seiring berlalunya waktu berlalu.

Metode ini membawa unsur-unsur yang menguntungkan dari praktek rekayasa perangkat lunak tradisional ke tingkat “ekstrem”, sehingga metode ini dinamai *Extreme Programming*. Unsur-unsur yang menjadi karakteristik metodologi adalah kesederhanaan, komunikasi, umpan balik, dan keberanian.

Gambar tahapan XP dapat dilihat pada gambar 2.1 :



Gambar 2. 1 Tahapan Extreme Programming

Sumbe : (Pressman, 2012)

Dibawah ini adalah penjelasan tahapan *Extreme Programming* yaitu :

1. *Planning*

Pada *Planning* berfokus untuk mendapatkan gambaran fitur dan fungsi dari perangkat lunak yang akan dibangun. Aktivitas *planning* dimulai dengan membuat kumpulan gambaran atau cerita yang telah diberikan oleh klien yang akan menjadi gambaran dasar dari perangkat lunak tersebut. Kumpulan gambaran atau cerita tersebut akan dikumpulkan dalam sebuah indeks dimana setiap poin memiliki prioritasnya masing-masing. Tim pengembang aplikasi juga akan menentukan perkiraan waktu serta biaya yang dibutuhkan untuk masing-masing indeks. Setelah semua kebutuhan terpenuhi, tim XP akan menentukan alur dari pengembangan aplikasi sebelum memulai pengembangan tugas.

2. *Design*

Aktivitas *design* dalam pengembangan aplikasi ini, bertujuan untuk mengatur pola logika dalam sistem. Sebuah desain aplikasi yang baik adalah desain yang dapat mengurangi ketergantungan antar setiap proses pada sebuah sistem. Jika salah satu

fitur pada sistem mengalami kerusakan, maka hal tersebut tidak akan mempengaruhi sistem secara keseluruhan.

Tahap Design pada model proses *Extreme Programming* merupakan panduan dalam membangun perangkat lunak yang didasari dari cerita klien sebelumnya yang telah dikumpulkan pada tahap *planning*. Dalam XP, proses design terjadi sebelum dan sesudah aktivitas coding berlangsung. Artinya, aktivitas design terjadi secara terus-menerus selama proses pengembangan aplikasi berlangsung.

3. Coding

Setelah menyelesaikan gambaran dasar perangkat lunak dan menyelesaikan design untuk aplikasi secara keseluruhan, XP lebih merekomendasikan tim untuk membuat modul unit tes terlebih dahulu yang bertujuan untuk melakukan uji coba setiap cerita dan gambaran yang diberikan oleh klien.

Setelah berbagai unit tes selesai dibangun, tim barulah melanjutkan aktivitasnya kepenulisan coding aplikasi. XP menerapkan konsep Pair Programming dimana setiap tugas sebuah modul dikembangkan oleh dua orang programmer. XP beranggapan, 2 orang akan lebih cepat dan baik dalam menyelesaikan sebuah masalah. Selanjutnya, modul aplikasi yang sudah selesai dibangun akan digabungkan dengan aplikasi utama.

4. Testing

Walaupun tahapan uji coba sudah dilakukan pada tahapan coding, XP juga akan melakukan pengujian sistem yang sudah sempurna. Pada tahap coding, XP akan terus mengecek dan memperbaiki semua masalah-masalah yang terjadi walaupun

hanya masalah kecil. Setiap modul yang sedang dikembangkan, akan diuji terlebih dahulu dengan modul unit tes yang telah dibuat sebelumnya.

Setelah semua modul selesai dan dikumpulkan ke dalam sebuah sistem yang sempurna, maka tim XP akan melakukan pengujian penerimaan atau acceptance test. Pada tahap ini, aplikasi akan langsung diuji coba oleh user dan klien agar mendapat tanggapan langsung mengenai penerapan gambaran dan cerita yang telah dilakukan sebelumnya.

2.8. JavaScript

JavaScript dimulai sekitar tahun 1994, pada tahun 1995, Brendan Eich mulai mengembangkan sebuah bahasa pemrograman script dinamakan Mocha. Bahasa Mocha ditujukan untuk client side dan juga server side. JavaScript merupakan bahasascript berbasis objek yang mengijjinkan pengguna untuk mengendalikan banyak aspek interaksi pengguna pada dokumen HTML. Semua objek tersebut memiliki properti yang saling berhubungan dengannya (Saifudin & Setiaji, 2019).

2.9. Android

Android adalah sistem operasi untuk telpon seluler berbasis Linux sebagai karnelnya. Android menyediakan *platform* terbuka (*open source*) bagi *developer* untuk menciptakan aplikasinya sendiri. Android Inc. adalah pendatang baru yang bergerak dalam pembuatan piranti lunak untuk ponsel. (Sasmito & Hadiansah, 2015)

Android adalah *mobile Operating System* (OS) yang dikembangkan oleh Google. OS Android berbasis pada OS Linux Kernel. Android bersifat open source, artinya pengembang bisa memodifikasi dan menyesuaikan OS untuk

setiap ponsel. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk di gunakan oleh bermacam-macam kegunaan. Android beberapa kali melakukan pembaruan versinya, kebanyakan nama di setiap versinya adalah nama makanan.



Gambar 2. 2 Logo Android

Android memiliki empat karakteristik sebagai berikut :

1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android merupakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan *open source*, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. *Platform* ini akan terus berkembang untuk membangun aplikasi *mobile* yang inovatif.

2. Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (*third-party application*). Semua aplikasi dapat

dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari web dengan data pada ponsel seseorang seperti kontak pengguna, kalender atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan tools yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Komersial pertama, Android 1.0, dirilis pada September 2008. Sejak April 2009, versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan makanan manis. Masing-masing versi dirilis sesuai urutan alfabet, yaitu :

1. *Cupcake (1.5)*
2. *Donut (1.6)*
3. *Eclair (2.0–2.1)*
4. *Froyo (2.2–2.2.3)*
5. *Gingerbread (2.3–2.3.7)*
6. *Honeycomb (3.0–3.2.6)*

7. *Ice Cream Sandwich (4.0–4.0.4)*
8. *Jelly Bean (4.1–4.3)*
9. *KitKat (4.4+)*
10. *Lollipop (5.0-5.1)*
11. *Marshmallow (6.0)*
12. *Nougat (7.0)*
13. *Oreo (8.0)*
14. *Pie (9.0)*
15. *Android 10.*

2.10. MySQL

MySQL merupakan *Database* yang menghubungkan script php menggunakan perintah *query* dan *escaps character* yang sama dengan php. MySQL mempunyai tampilan *client* yang mempermudah anda dalam mengakses *database* dengan kata sandi untuk mengizinkan proses yang bisa anda lakukan (Sadeli, 2014).

2.11. Xampp

XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL (Batubara, 2015)

XAMPP adalah perangkat lunak *opensource* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi seperti *windows, linux, solaris, dan mac* (Buana, 2014).

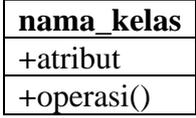
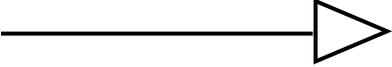
2.12. Perancangan Sistem UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa dan Shalahuddin, 2018).

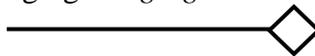
1. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa dan Shalahuddin, 2018). simbol-simbol yang ada pada diagram kelas pada tabel class diagram 2.1 di bawah ini:

Tabel 2. 1 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

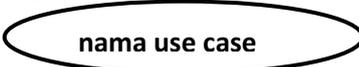
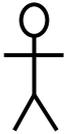
Tabel 2.1 Simbol *Class Diagram* (Lanjutan)

Simbol	Deskripsi
Kebergantungan/ <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2. Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa dan Shalahuddin, 2018). simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.2 di bawah ini:

Tabel 2. 2 Simbol Diagram *Use Case*

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi/ <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan
<< <i>extend</i> >>	dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip

Tabel 2.2 Simbol Diagram *Use Case* (Lanjutan)

Simbol	Deskripsi
	<i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>Include/uses</i> << <i>include</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

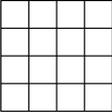
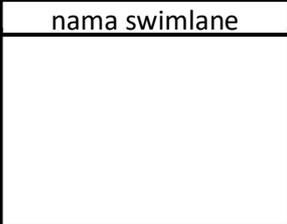
3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2018). simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2. 3 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

Tabel 2.3 Simbol *Activity Diagram* (Lanjutan)

Simbol	Deskripsi
<p>Tabel</p> 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
<p>Dokumen</p> 	Menunjukkan dokumen sumber atau laporan
<p>Status akhir</p> 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<p><i>Swimlane</i></p> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.13. Pengujian ISO 25010

Model ISO-25010 merupakan bagian dari *Software product Quality Requirements and Evaluation (SQuaRE)*, yang merupakan pengembangan dari model kualitas perangkat lunak sebelumnya yaitu ISO-9126. Dalam model ISO-25010 ini digunakan untuk melihat kualitas suatu perangkat lunak yang digunakan oleh perusahaan, instansi ataupun organisasi. Metode ISO 25010 ini dapat digunakan untuk mengevaluasi kualitas sistem perangkat lunak secara spesifik berdasarkan dua dimensi umum, yaitu dimensi *product quality*, dimana prosesnya mengacu pada karakteristik intrinsik dari sebuah produk perangkat lunak, memiliki beberapa elemen antara lain meliputi *functional suitability*, *reliability*, *operability*, *performance efficiency*, *security*, *compatibility*, *maintainability* dan

transferability. *Quality in use* dan *product quality*. Sedangkan pada *dimensi quality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user* antara lain *Usability in use*, *Flexibility in use*, dan *Safety* Adapun untuk mengetahui gambaran kualitas *system*, penulis melakukan analisis berdasarkan model ISO-25010 yang terdiri dari dua dimensi umum, yaitu dimensi *product quality* dan dimensi *quality in use* (Abran et al., 2008) Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya :

1. *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk Merupakan tingkatan dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi spesifik tertentu dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna.
2. *Reliability* Merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik dalam hal ini perangkat lunak dapat beroperasi dan siap ketika dibutuhkan untuk digunakan dan juga dapat bertahan pada tingkat kemampuan tertentu terhadap kegagalan, kesalahan serta perangkat lunak kembali pada tingkat tertentu dalam mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak.
3. *Performance efficiency* Merupakan tingkatan dimana perangkat lunak dapat memberikan kinerja terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu dalam hal ini *performance efficiency*

dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi dan perangkat lunak dapat menggunakan sejumlah sumber daya ketika melakukan aksi dari sebuah fungsi.

4. *Operability* Perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik pengguna ketika digunakan dalam hal ini perangkat lunak mudah dipelajari oleh pengguna, perangkat lunak dapat digunakan dan dioperasikan oleh pengguna, perangkat lunak dapat memberikan bantuan ketika pengguna membutuhkan panduan, perangkat lunak dapat menarik perhatian pengguna, perangkat lunak memenuhi kebutuhan pengguna yang memiliki keterbatasan dan perangkat memungkinkan untuk dianalisis oleh pengguna apakah perangkat lunak sudah memenuhi kebutuhan mereka.
5. *Security* Merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman atau keganjalan dalam hal ini perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna dan merupakan dari kelengkapan, ketepatan dari sejumlah *asset* yang telah dijaga sehingga aksi atau tindakan yang dilakukan telah terbukti dan hal tersebut tidak dapat ditolak.
6. *Compability* Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.

7. *Maintainability* Merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan , kebutuhan dan fungsionalitas yang spesifik. Selain itu perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak untuk mengidentifikasi bagian yang dapat dimodifikasi.
8. *Transferability*. Merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan satu ke lingkungan yang lain dalam hal ini perangkat lunak dapat beradaptasi dengan cepat pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.

2.14. Skala Pengukuran

Skala pengukuran yang digunakan adalah skala Likert, skala yang didasarkan pada penjumlahan sikap responden dalam merespon pernyataan berkaitan indikator-indikator suatu konsep atau variable yang sedang diukur (Sanusi, 2012). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga).Skala Likert apat dilihat pada Tabel 2.4.

Tabel 2. 4 Skala Likert

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Sumber : (Sanusi, 2012)

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat dibawah ini

$$\text{Persentase} = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\%$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
$90 \leq x$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

Sumber : (Sanusi, 2012)

Keterangan: x = persentase hasil pengujian.