

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 studi yang diambil yaitu:

Tabel 2. 1 Tinjauan Pustaka

No.	Nama	Judul	Masalah	Kesimpulan
1.	Handrianto and Sanjaya (2020)	Model Waterfall Dalam Rancang Bangun Sistem Informasi Pemesanan Produk Dan Outlet Berbasis Web	masih menggunakan proses bisnis yang konvensional, dimana setiap adanya pendaftaran outlet baru harus dilakukan pendataan secara manual sehingga membutuhkan proses panjang yang mengakibatkan terjadinya penumpukan data yang mengakibatkan meningkatnya kesalahan pencatatan data sampai hilangnya data transaksi pemesanan outlet oleh pihak toko yang ingin bergabung.	Hasil yang didapat dari penelitian ini adalah sistem informasi pemesanan produk dan outlet dengan menggunakan layanan website menjadi suatu solusi yang dipilih oleh PT. Prima Lakto Sehat. Disamping kemudahan dalam proses transaksi pemesanan produk oleh setiap outlet dan pembukaan outlet yang baru, sistem informasi yang dibangun diharapkan mampu membantu PT. Prima Lakto Sehat dalam mengelola data transaksi keduanya secara cepat dan tepat sehingga memudahkan pihak manajemen perusahaan dalam menentukan suatu pengambilan keputusan. Sistem

No.	Nama	Judul	Masalah	Kesimpulan
			Demikian juga dengan proses pemesanan produk yang dilakukan oleh setiap outlet ke pihak perusahaan, proses yang dilakukan saat ini belum menggunakan suatu sistem yang terintegrasi sehingga setiap pemesanan produk yang diterima oleh perusahaan belum tercatat dengan baik. Melihat	
2.	Lesmono (2018)	Rancang Bangun Sistem Informasi Penjualan Sepatu Berbasis Website Dengan Metode Waterfall	Pada saat ini di masih ada masyarakat yang belum memanfaatkan teknologi informasi sehingga pengelolaan masih dilakukan secara manual	Hasil penelitian pengembangan website sangat penting terutama untuk mengatasi persaingan bisnis sekarang ini khususnya penjualan sepatu, dalam melakukan kegiatan transaksi mulai dari proses order, menyimpan data produk, mengubah data produk serta menghapus data produk sehingga menghasilkan informasi yang cepat, tepat, akurat dan menarik bagi pelanggan
3.	Armalia and Nuryana (2020)	Rancang Bangun Sistem Informasi Penjualan Berbasis Web (Studi Kasus: CV Gilang	Kesulitan dalam melayani dan transaksi penjualan yang masih manual dan dikelola oleh CV tanpa staff	Hasil dari sistem penjualan ini menyediakan informasi produk dan sistem penjualan. Dan dari hasil kusioner Sistem Informasi Penjualan sesuai dengan kebutuhan CV Gilang Sentosa Sidoarjo

No.	Nama	Judul	Masalah	Kesimpulan
		Sentosa Sidoarjo)		
4.	Hidayati (2019)	Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan	Masih menggunakan sistem konvensional seperti dalam pemesanan barang, pengolahan data, penyimpanan data dan pembuatan laporan penjualannya. Hal tersebut mengakibatkan terjadinya kesalahan pencatatan data, pencarian data terlalu lama dan keterlambatan dalam pembuatan laporan serta data yang dihasilkan tidak akurat.	Hasil dari penelitian ini anchang bangun sistem informasi penjualan untuk dapat diterapkan di Toko tersebut, dengan menggunakan Netbeans dandatabase MySQL. Metode yang digunakan dalam rancang bangun sistem ini menggunakan Waterfall. Dengan penggunaan metode ini, maka akan lebih mempermudah dalam pembuatan rancang bangun sistem penjualan terutama pada Toko Fadhil Genteng Bogor, sehingga permasalahan yang dihadapinya dapat teratas
5	Utama <i>et al</i> (2019)	Rancang Bangun Sistem Informasi Pemesanan Custom Furniture Berbasis Website	Mempunyai permasalahan pada proses bisnis mulai dari penempatan dokumen yang tidak terstruktur, pembuatan dokumen yang kurang cepat, jadwal pengukuran yang kurang disiplin.	Sistem informasi pemesanan yang terkomputerisasi untuk mengatasi masalah yang terjadi agar mampu mengolah data serta memberikan informasi yang cepat, akurat dan tepat waktu saat dibutuhkan.

Berdasarkan penelitian terdahulu terdapat beberapa perbedaan yaitu :

1. Metode yang digunakan menggunakan metode *extreme programming*
2. Sistem yang dibangun dapat melakukan *chating* melalui *whatsapp*
3. Sistem yang dibangun dapat melakukan komentar dan *rating* produk
4. Sistem dapat menampilkan grafik penjualan
5. Pengujian sistem menggunakan ISO 25010

2.2. Rancang Bangun

Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada (Pressman, 2015).

2.3. 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 (Kristanto, 2018).

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).

2.4. Pemesanan

Pemesanan merupakan kegiatan pemesanan barang atau jasa yang sesungguhnya untuk mencapai volume pemesanan yang dikehendaki (Tjendera, 2016)

Pemesanan adalah penerimaan pesanan dari pelanggan terhadap suatu produk. Lanjutan dari pemesanan adalah pengiriman produk sampai ketangan pemesan dengan selamat (Susanti, 2018)

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 Pemesanan

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

1) Pemesanan *Online*.

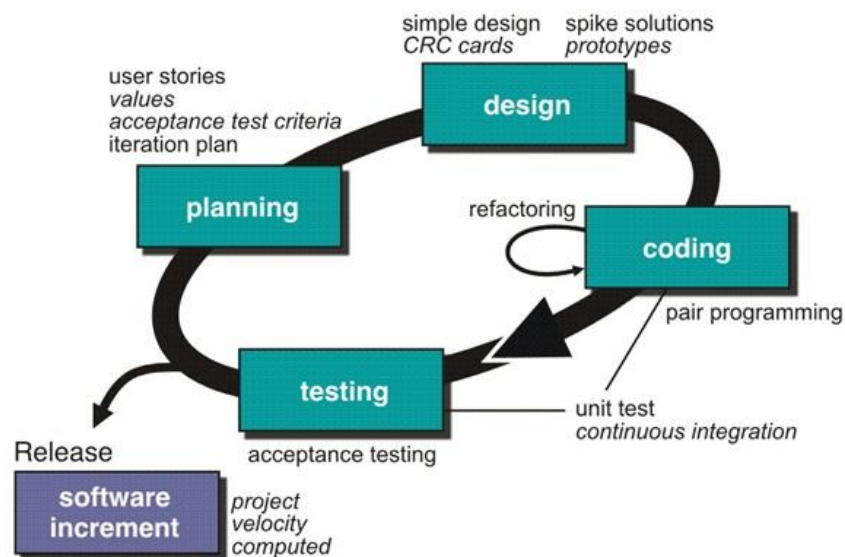
Kemajuan teknologi saat ini mengakibatkan sistem pemesanan juga mengalami perkembangan kearah sistem pemesanan online. Pemesanan online bisa diakses oleh siapapun dan dimanapun mereka berada yang memiliki akses internet.

2) Pemesanan *Offline*

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

2.5. Metode *Extreme Programming*

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 dimana persyaratan pelanggan baru dapat diadopsi. 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* (Pressman, 2012). Dibawah ini adalah gambar tahapan *extreme programming* dapat dilihat pada Gambar 2.1 :



Gambar 2.1 Tahapan *Extreme Programming*
Sumber : (Pressman, 2012)

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 system mengalami kerusakan, maka hal tersebut tidak akan mempengaruhi system 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. *simple design CRC Cards* merupakan gambaran untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

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 kedalam 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 dilakukans ebelumnya.

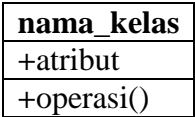



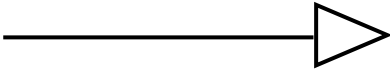
2.6. Perancangan Sistem UML (*Unified Modeling Language*)


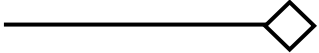
UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa and Shalahudin, 2018)

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. Simbol-simbol yang ada pada diagram kelas pada **Tabel 2.2** di bawah ini:

Tabel 2. 2 Simbol *Class Diagram*


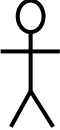
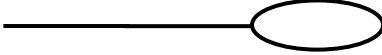
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi 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)


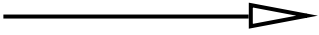

Kebergantungan/ <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2. Use Case Diagram

Menurut Rosa dan Shalahuddin (2018) *Use case diagram* atau diagram *use case* merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol yang ada pada diagram *use case* dapat dilihat pada **Gambar 2.3** di bawah ini

Tabel 2. 3 Simbol diagram use case



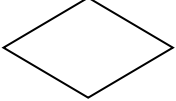
Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan 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
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> << <i>extend</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <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

Diagram Aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak, diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor yaitu aktivitas yang dapat dilakukan oleh sistem. simbol-simbol yang ada pada *sequence* digram dapat dilihat pada **Tabel 2.4** di bawah ini:

Tabel 2. 4 Simbol Activity Diagram

Simbol	Deskripsi
Status Awal 	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>Decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Asosiasi penggabungan dimana lebih dari satu aktivitas

Penggabungan / <i>Join</i>	digabungkan menjadi satu		
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir		
<p><i>Swimlane</i></p> <div data-bbox="373 562 793 902" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;"><i>Name swimlane</i></p> </div> <p style="text-align: center;">Atau</p> <div data-bbox="411 1102 767 1406" style="border: 1px solid black; padding: 5px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; padding: 5px;"><i>Na me swi mla ne</i></td> <td style="width: 85%;"></td> </tr> </table> </div>	<i>Na me swi mla ne</i>		<p>Swimlane adalah elemen visual yang digunakan dalam diagram alir proses yang menggambarkan siapa bekerja pada subset tertentu dari sebuah proses. Swimlane tersebut diatur baik horisontal maupun vertikal dan digunakan untuk mengelompokkan sub-proses sesuai dengan tanggung jawab mereka.</p>
<i>Na me swi mla ne</i>			

2.7. Pengertian MySQL

Gilmore (2015) *MySQL* adalah sebuah relasional *database server* yang menawarkan berbagai macam mekanisme dalam pengolahan data yang dikenal dengan *storage engines*.

Sheldon and Moes (2015) *MySQL* sama seperti sistem *DBMS* lainnya, seperti *Oracel*, *DB2*, dan *SQL Server*. Memungkinkan untuk mengakses,

memanipulasi, melindungi, dan memelihara metadata yang diperlukan untuk mendefinisikan data yang disimpan.

2.8. Web

Website atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau bergerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis ataupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Website sendiri merupakan sebuah kumpulan halaman-halaman situs yang tersimpan dalam sebuah server/hosting, dan teridentifikasi melalui sebuah nama yang disebut juga sebagai domain atau sub domain (Sadeli 2015)

2.9. PHP (*Personal Home Page*)

PHP (*Personal Home Page*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada HTML. PHP merupakan singkatan dari “PHP: *Hypertext Preprocessor*”, dan merupakan bahasa yang disertakan dalam dokumen HTML sekaligus bekerja di sisi *server* (*server-side HTML-embedded scripting*). Artinya sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada halaman HTML biasa, sehingga script-nya tak tampak di sisi client (Sadeli 2015)

2.10. XAMPP

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

2.11. *Sublime Text*

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user (Sadeli, 2015)

2.12. Analisis PIECES

Metode analisis sistem yang digunakan adalah metode PIECES. Menurut (Fatta, 2012), metode ini menggunakan enam variabel evaluasi yaitu :

1. *Performance* (kinerja) memiliki peran penting untuk menilai apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya.
2. *Information* (informasi) menilai apakah informasi mempunyai nilai guna untuk pengguna dalam hal konten, ketepatan waktu, akurasi, dan format informasi.
3. *Economic* (ekonomi) menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau diturunkan biaya penyelenggaraannya.
4. *Control* (pengendalian) menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik dan kemampuannya untuk mendeteksi kesalahan/ kecurangan menjadi semakin baik pula.
5. *Efficiency* (efisiensi) menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi. Kemampuan mengolah data dengan meminimalisasi langkah kerja yang dianggap tidak perlu,
6. *Service* (layanan) menilai apakah layanan sistem dapat diandalkan, fleksibel, dan ditingkatkan kemampuannya.

2.13. Pengujian ISO 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang software engineering (Iqbal, 2016). Product quality ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

1. Kualitas dalam model penggunaan
2. Model kualitas produk
3. Data model kualitas

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) *Usability* 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.
- 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 (Sugiyono, 2017). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga). Skala Likert dapat dilihat pada **Tabel 2.4** di bawah ini.

Tabel 2. 5 Skala Likert

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

Sumber : (Sugiyono, 2017)

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

$$\text{Persentase} = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\% \dots\dots\dots(1)$$

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

Tabel 2. 6 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 : (Sugiyono, 2017)

Keterangan: x = persentase hasil pengujian.