

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Adapun perbedaan penelitian dengan penelitian terdahulu dapat dilihat pada Tabel

2.1 :

Tabel 2. 1 Tinjauan Pustaka

No	Nama (Tahun)	Judul	Kemahan	Metode	Kelebihan
1	Mariana dan Wijayanti (2018)	Sistem Informasi Penjualan Berbasis Web Parfum Non Alkohol Pada Toko Maju Pontianak	Toko Maju ini banyak di kunjungi pelanggan pada hari libur. Dengan kondisi bangunan yang sempit, sehingga pelanggan berdesakan, terkadang sampai harus menunggu antrian dengan pelanggan lainnya. Penyewaan	RAD	Proses yang dihasilkan didapat secara cepat dan tepat. Keuntungan memiliki web atau situs toko online sendiri dapat meningkatkan kesan profesional dan kepercayaan konsumen. Nama
2	Syafriani dan Suendri (2024)	Aplikasi Penjualan Parfum Menggunakan Metode Economic Order Quantity (EOQ) Perfume Sales Application using the	Proses penjualan, data dicatat di form penjualan kemudian diinput menggunakan aplikasi Microsoft Excel. Proses pencatatan seperti ini	RAD	Maju Jaya Parfum dalam penjualannya bisa lebih tinggi atau lebih rendah tergantung atas permintaan masing-masing penjual, akibatnya muncul beberapa

No	Nama (Tahun)	Judul	Kemahan	Metode	Kelebihan
		Economic Order Quantity	kurang efektif karena banyaknya jumlah buku dan pesanan yang harus diinput, sehingga sering terjadi kekeliruan dan kesalahan dalam penginputan dan pencarian data yang dapat menyita waktu dalam pembuatan laporannya. Pengelolaan		kesulitan dan kesalahan dalam pencatatan laporan pembelian dan persediaan barang.
3	Annas Susanto <i>et al.</i> (2022)	Sistem Informasi Penjualan Pewangi Pakaian berbasis Website di Toko Parfum Rika Utami Bangil	Masalah yang terjadi pada Toko Parfum Laundry Rica Utami kesulitan bagi pelanggan harus datang ke toko untuk memesan dan mengetahui informasi produk, pembuatan rekap data-data penjualan pada toko masih manual serta area penjualan belum meluas	<i>Watrefall</i>	Hasil dari pembuatan sistem informasi penjualan pewangi pakaian pada toko Parfum Laundry Rica Utami, para pelanggan dapat mencari informasi dan pemesanan produk melalui internet serta bagi pemilik toko dan karyawan toko dapat mengakses dan

No	Nama (Tahun)	Judul	Kemahan	Metode	Kelebihan
					mengolah data dengan mudah.
4	Prayuda <i>et al.</i> (2020)	Rancang Bangun Sistem Informasi Penjualan Pada Pt. Duta Perfume Berbasis Web Menggunakan Metode Sekuensial Linier	Transaksi yang masih menggunakan nota untuk penjualan, nota ini yang membuat pekerjaan semakin lambat selain itu nota penjualan tersebut akan terjadinya kehilangan, pencacatan stok barang masih dengan manual, pemesanan barang sandal masih manual dengan menggunakan media social	<i>Prototype</i>	Hasil dari penelitian ini dapat membantu penjualan PT. Duta Perfume dalam memanfaatkan online marketing (pemasaran online) dengan menggunakan teknologi e-commerce
5	Putra, Padilah dan Carudin (2022)	Perancangan Sistem Informasi Penjualan Produk Parfum Berbasis Web Pada Toko Rinas Mashel Bekasi	Transaksi penjualannya yang masih manual sehingga membuat pelanggan kesulitan ketika memilih produk parfum yang sesuai	<i>Waterfall</i>	sistem informasi penjualan produk parfum yang bertujuan untuk memudahkan proses transaksi, meningkatkan penjualan serta meminimalisir terjadinya faktor kesalahan human error dalam

No	Nama (Tahun)	Judul	Kemahan	Metode	Kelebihan
					pembuatan laporan penjualan

2.2. Sistem Informasi

Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware dan software yang saling berinteraksi sebagai kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama (Maniah dan Haminidin, 2017).

Sistem adalah sekumpulan unsur atau elemen dan komponen-komponen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu (Fernanda dan Fernando, 2017)

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Wati, Rahmanto dan Fernando, 2019)

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 ialah suatu sistem yang menyajikan informasi yang bertujuan untuk mengambil suatu keputusan dan menjalankan kegiatan operasional suatu organisasi dan memberikan suatu keunggulan kompetitif (Awaludin, Bahri dan Muslih, 2019)

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi (Adrian dan Pramono, 2017)

2.3. Penjualan

Penjualan adalah proses dimana sang penjual memuaskan segala kebutuhan dan keinginan pembeli agar dicapai manfaatnya bagi yang penjual maupun sang pembeli yang berkelanjutan dan yang menguntungkan bagi kedua belah pihak. Penjualan juga hasil yang dicapai sebagai imbalan jasa – jasa yang diselenggarakan yang dilakukannya perniagaan transaksi dunia usaha (Ahmad dan Hasti, 2018).

2.3.1. Model Transaksi

Jenis transaksi dalam menerapkan sistemnya pada penelitian yang dilakukan oleh peneliti menggunakan *Business to Business (B2B)*. Berikut ini Menurut (Laudon dan Traver, 2017) Jenis-jenis transaksi di antaranya sebagai berikut:

1. *Collaborative Commerce (C-Commerce)* yaitu kerjasama secara elektronik antara rekan bisnis. Kerjasama ini biasanya terjadi antara rekan bisnis yang berada pada jalur penyediaan barang (*Supply Chain*).
2. *Business to Business (B2B)* tipe ini meliputi transaksi antar organisasi yang dilakukan di *electronic market*. *Business to Business* memiliki karakteristik:
 - a. *Trading partners* yang sudah diketahui dan umumnya memiliki hubungan yang cukup lama. Informasi hanya dipertukarkan dengan *partner* tersebut. Dikarenakan sudah mengenal rekan komunikasi,

jenis informasi yang dikirimkan dapat disusun sesuai dengan kebutuhan dan kepercayaan.

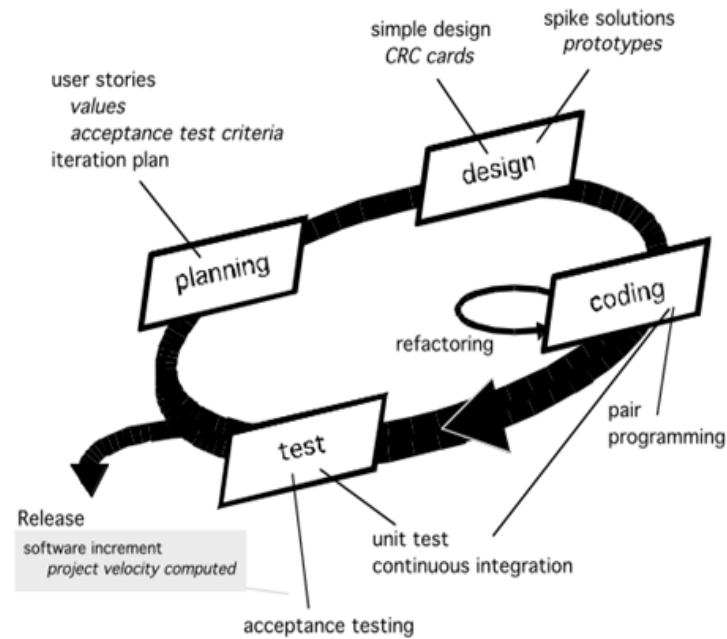
- b. Pertukaran data berlangsung berulang-ulang dan secara berkala, misalnya setiap hari dengan format data yang sudah disepakati bersama. Dengan kata lain, servis yang digunakan sudah tertentu. Hal ini memudahkan pertukaran data untuk dua *entity* yang menggunakan standar yang sama.
 - c. Salah satu pelaku dapat melakukan inisiatif untuk mengirimkan data, tidak harus menunggu partnernya.
 - d. Model yang umum digunakan adalah *peer to peer*, dimana *processing intelligence* dapat didistribusikan pada pelaku bisnis.
3. *Business to Consumers (B2C)* yaitu penjual adalah suatu organisasi dan pembeli adalah individu. *Business to Consumers* memiliki karakteristik sebagai berikut:
- a. Terbuka untuk umum, dimana informasi disebarakan ke umum.
 - b. *Service* yang diberikan bersifat umum (*generic*).
 - c. *Service* diberikan berdasarkan permohonan. *Consumer* melakukan inisiatif dan produsen harus memberikan respon sesuai permohonan.
4. *Consumer to Business (C2B)* yaitu *Consumer to Business* memberitahukan kebutuhan atas suatu produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen.

5. *Costumer to Customer (C2C)* yaitu konsumen menjual secara langsung ke konsumen lain atau mengiklankan jasa pribadi di internet. Dalam *Costumer to Costumer costumer* menjual produk atau jasa ke costumer lain. Dapat juga disebut sebagai pelanggan ke pelanggan, yaitu orang yang menjual produk dan jasa ke satu sama lain.

2.4. Metode Pengembangan pada *Extreme Programming*

Menurut Supriyatna (2018) *Extreme Programming (XP)* merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan-perubahan *requirement* yang sangat cepat.

Sedangkan menurut Lubis (2016) *Extreme Programming (XP)* dikenal dengan metode atau “*technical how to*” bagaimana suatu tim teknis mengembangkan perangkat lunak secara efisien melalui berbagai prinsip dan teknik praktis pengembangan perangkat lunak. XP menjadi dasar bagaimana tim bekerja sehari- hari. Tahapan *Extreme Programming* dapat dilihat pada **Gambar 2.1**



Gambar 2. 1 Model *Extreme Programming (XP)*
Sumber : Lubis (2016)

Berikut ini adalah penjelasan tahapan *Extreme Programming* yaitu :

1. *Planning* (Perencanaan)

Kegiatan Perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan-kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan *fungsionalitas*.

Pada perencanaan terdapat *user stories values* yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, *acceptance test criteria iteration plan* melakukan perhitungan kecepatan project selama development, customer dapat menambah story, merubah value, membagi story atau menghapusnya.

2. *Design* (Perancangan)

Perancangan yang simple, menarik, dan sederhana selalu memberikan hasil yang lebih disukai daripada gambaran-gambaran yang lebih kompleks. Perancangan XP memberikan panduan implementasi untuk suatu cerita ketika ditulis, tidak kurang, tidak lebih.

Terdapat *simple design CRC Cards* untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

3. *Coding* (Pengkodean)

Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke pengkodean.

Pada tahapan *pair programming* melakukan kerja sama untuk membuat *code* dari satu *story*. Dan *refactoring* adalah proses restrukturisasi kode program komputer yang ada tanpa mengubah perilaku eksternalnya.

4. *Pengujian* (Pengujian)

Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali.

Pada tahapan pengujian yaitu *unit test continuous integration* yaitu tahapan pengujian *code* yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh *customer* dan *focus* pada keseluruhan dan fungsional

sistem, dan *acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari *software realease*.

Selanjutnya terdapat tahapan *software increment project velocity computed* yaitu tahapan yang telah diimplementasikan dari *software realease* yang nantinya akan diterapkan dalam suatu sistem.

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




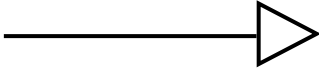


UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa A.S dan M. Shalahuddin, 2018)

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. simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.2 di bawah ini:

Tabel 2. 2 Simbol *Class Diagram*

No	Simbol	Deskripsi
1.	<div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: fit-content;"> <p>nama_kelas</p> <hr/> <p>+atribut</p> <hr/> <p>+operasi()</p> </div>	Kelas pada struktur sistem

No	Simbol	Deskripsi
2.	<p>Antarmuka/<i>Interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	<p>Asosiasi/<i>asociation</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Kebergantungan/<i>dependecy</i></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7.	<p>Agregasi/<i>agregation</i></p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)


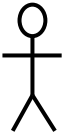


Sumber: (Rosa A.S dan M. Shalahuddin, 2018)



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 A.S dan M. Shalahuddin, 2018), simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.3 di bawah ini:

Tabel 2. 3 Simbol diagram *use case*

No	Simbol	Deskripsi
1.	<p><i>Use Case</i></p> 	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>
2.	<p>Aktor/<i>actor</i></p> 	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 actor
3.	<p>Asosiasi/<i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4.	<p>Ekstensi/<i>extend</i></p> <p><<<i>extend</i>>></p> 	Relasi <i>use case</i> tambahan kesebuah <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



No	Simbol	Deskripsi
5.	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.
6.	Menggunakan/ <i>Include/uses</i> <code><<include>></code> 	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



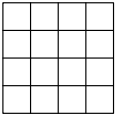


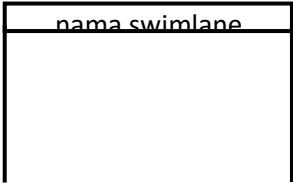
Sumber: (Rosa A.S dan M. Shalahuddin, 2018)

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. simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2. 4 Simbol Activity Diagram

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
6.	Dokumen 	Menunjukkan dokumen sumber atau laporan
7.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
8.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: (Rosa A.S dan M. Shalahuddin, 2018)

2.6. MySQL

Gilmore (2017) MySQL adalah server database relasional yang menawarkan berbagai mekanisme untuk memproses data yang dikenal sebagai mesin penyimpanan.

Sheldon and Moes (2016) MySQL sama seperti sistem DBMS lainnya, seperti Oracle, DB2, dan SQL Server. Memungkinkan untuk mengakses, memanipulasi, melindungi, dan memelihara metadata yang diperlukan untuk menentukan data yang disimpan.

Jadi dapat disimpulkan bahwa *MySQL* adalah sebuah relasional *database server* suatu bahasa yang digunakan untuk mengelola data pada *Relation DBMS*.

2.7. 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.

PHP dirancang untuk dapat bekerja sama dengan database server dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses database menjadi begitu mudah. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi dimana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di *server* (Betha & Husni, 2016)

2.8. XAMPP

Menurut Sheldon and Moes (2015) XAMPP merupakan paket program web lengkap yang dapat Anda gunakan untuk mempelajari pemrograman web khususnya PHP dan MySQL.

Keunggulan Xampp adalah performa tinggi, stabil, banyak fitur, mudah dikonfigurasi, hanya menggunakan sedikit resource di server, tidak bergantung pada thread untuk melayani klien.

Kekurangan Xampp adalah tidak support IPV6, update / patch terbaru sudah lama, Fast-CGI-nya tidak berfungsi maksimal, penggunaanya tidak sebanyak aktivitas paket program lainnya.

2.9. 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.

Dalam pengujian perangkat lunak ini penulis menggunakan suatu metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak yang dibangun dengan metode yang diambil adalah metode pengujian *ISO 25010* dengan aspek *functionlaity* dan *usability*.

2.10. 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 2.5** 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

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

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada **Tabel 2.6** 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.

