

BAB II **LANDASAN TEORI**

2.1 Tinjauan Pustaka

Berikut adalah beberapa penelitian yang diperlukan tinjauan pustaka yang diambil dari beberapa jurnal penelitian yang berkaitan dengan judul penelitian ini dan pokok bahasan, berikut ini merupakan tinjauan pustaka yang diambil yaitu:

1. Penelitian Fitriyana & Sucipto (2020) yang berjudul Sistem Informasi Penjualan Oleh *Sales Marketing* Pada PT Erlangga Mahameru. Proses penjualan buku yang telah diterbitkan ditujukan bagi sekolah-sekolah, masih memiliki beberapa kendala seperti sekolah harus *system* ke perusahaan untuk melakukan pemesanan buku atau melalui jasa, dan kurangnya informasi yang didapatkan oleh pihak-pihak sekolah terhadap buku – buku yang diterbitkan berdasarkan kurikulum yang berlaku di Indonesia. Penelitian ini dilakukan untuk membuat sistem informasi yang dilakukan oleh *sales marketing*. Perancangan sistem ini menggunakan *UML*, *Usecase Diagram*, serta *Activity Diagram*. Sedangkan sistem pemrograman yang digunakan adalah *PHP*. Untuk databasenya menggunakan *MySQL*. Hasil dari penelitian ini adalah sistem informasi yang dapat mempermudah bagian marketing untuk mempromosikan buku. Selain itu konsumen juga dapat melakukan transaksi secara *online*. Sistem informasi ini diuji menggunakan *black box* serta dilakukan pengukuran pengujian menggunakan kuisisioner. Hasil pengujian adalah sebesar 89,58%. Dengan demikian sistem yang dikembangkan ini layak untuk di implementasikan.

2. Penelitian Juniansyah et al (2020) yang berjudul Pembuatan E-Commerce Pemesanan Jasa *Event Organizer* Untuk *Zero Seven Entertainment*. Dengan tujuan agar memudahkan perusahaan untuk promosikan acara atau produk dan jasa mereka. Dan dengan mudah melakukan pemesanan langsung melalui website yang disediakan sehingga pembeli atau konsumen tidak perlu mengantri untuk mendapatkan tempat di *event organizer* tersebut. Aplikasi ini menggunakan *PHP* dan juga menggunakan database *Mysql* ataupun *phpmyadmin*. Metode yang digunakan adalah metode *extreme programming* dan dilakukan pengujian aplikasi menggunakan pengujian ISO 9126. Setelah diuji, aplikasi ini dapat diakses secara *online* melalui internet berupa komputer, laptop dan smartpone. Aplikasi ini mampu memfasilitasi pemasang produk barang atau jasa dalam menawarkan event yang ditawarkan melalui media yang sudah disediakan perusahaan.
3. Penelitian Adrian & Pramono (2017), yang berjudul Rancang Bangun Sistem Informasi Penjualan Barang Pada Toko Distro *Black Outlet* Berbasis Web. Saat ini media promosi yang digunakan pada Toko Distro Black Outlet adalah dengan memasang banner di depan toko. Kemudian untuk mempromosikan produk baru menggunakan *blackberry messenger* yang hanya diketahui oleh kontak yang berteman dengan pihak Toko Distro *Black Outlet*. Permasalahannya informasi pemasaran produk baru tidak diketahui calon konsumen secara luas mengenai produk-produk pada Toko Distro Black Outlet. Akibat dari permasalahan yang ada, Toko Distro Black Outlet membutuhkan peningkatan pada pemasaran produk sehingga dapat menambah jumlah pelanggan dan pendapatan toko. Dengan menggunakan

sistem *e-marketing* konsumen dapat dengan mudah mendapatkan informasi tentang produk-produk baru, mengenai model produk, harga produk, jumlah produk dan stok produk yang ada pada toko ini. Oleh karena itu untuk mengatasi permasalahan pemasaran pada Toko Distro *Black Outlet*, perlu dibuat *E-Marketing*, untuk menyediakan informasi produk. Penelitian ini bertujuan untuk membuat sistem pemasaran dengan pendekatan *user experience design*. Penelitian ini menggunakan perangkat CMS *open cart*.

4. Penelitian Anthony *et al* (2017), yang berjudul Analisis Dan Perancangan Sistem Informasi Penjualan Berdasarkan Stok Gudang Berbasis *Client Server* (Studi Kasus Toko Grosir “Restu Anda”). Sistem penjualan yang berjalan adalah dengan menjual barang kepada *downline* berdasarkan pada berapa banyak barang yang masih tersedia di sistem. Karena proses pencatatan barang masih manual sehingga sistem pemilik toko memeriksa barang-barang di sistem, kadang-kadang jumlah item yang tercatat tidak sesuai dengan yang dilaporkan. Akibatnya saat menerima pesanan pembelian barang dari *downline*, pemilik toko yang melihat pencatatan stok barang dan barang yang diinginkan *downline* tercatat habis atau kosong maka terpaksa pemilik toko menolak dan menawarkan barang lain yang pastinya tidak semua *downline* menerima tawaran tersebut. Maka dari itu untuk menyelesaikan masalah tersebut adalah dengan merancang sistem informasi penjualan berbasis *client-server*. Aplikasi ini dibangun dengan menggunakan teknologi RMI (*Remote Method Invocation*) yang memudahkan pengguna dalam mengelola data dari dua atau lebih komputer yang berbeda. Teknologi ini memudahkan pemilik toko untuk memantau jumlah stok barang yang ada digudang tanpa

membutuhkan waktu yang lama, sehingga penjualan barang di toko Restu Anda dapat dijalankan dengan benar. Dan dengan aplikasi ini pemilik toko dapat memprediksikan berapa banyak kebutuhan barang yang harus disediakan untuk dijual kepada *downline* dengan melihat data penjualan yang sudah tersinkronisasi dengan data member.

5. Penelitian Puspita & Irianti (2020), yang berjudul Perancangan Sistem Informasi Pemesanan Dan Pembayaran Pada Percetakan Mahardika. Percetakan adalah sebuah proses industri untuk memproduksi secara massal tulisan dan gambar. Jasa cetak saat ini semakin banyak dibutuhkan masyarakat karena dapat mempermudah dalam penyampaian suatu informasi. Percetakan Mahardika masih menggunakan sistem konvensional dalam transaksinya sehingga masih sering terjadi kesalahan dalam pembuatan laporan. Tujuan dilakukan penelitian adalah merancang sebuah sistem informasi untuk memudahkan transaksi pemesanan dan pembayaran pada percetakan. Metode penelitian yang digunakan adalah Observasi, Wawancara, Studi kepustakaan, Analisis dan Perancangan. Dengan adanya perancangan sistem ini diharapkan dapat memberikan gambaran kepada pihak percetakan dalam proses pengolahan data transaksi pemesanan dan pembayaran pada percetakan sehingga dapat mengolah data dengan lebih cepat dan dapat menghasilkan informasi yang tepat.

2.2 Sistem Informasi

Sistem adalah kumpulan atau himpunan dari unsur atau variabel-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).

2.3 Penjualan

Penjualan adalah kegiatan transaksi yang sesungguhnya untuk mencapai volume penjualan yang dikehendaki (Tjendera, 2016).

Tujuan tersebut akan menjadi kenyataan apabila dilaksanakan dengan kemauan dan kemampuan yang memadai. Selain itu, harus diperhatikan pula faktor-faktor berikut ini.

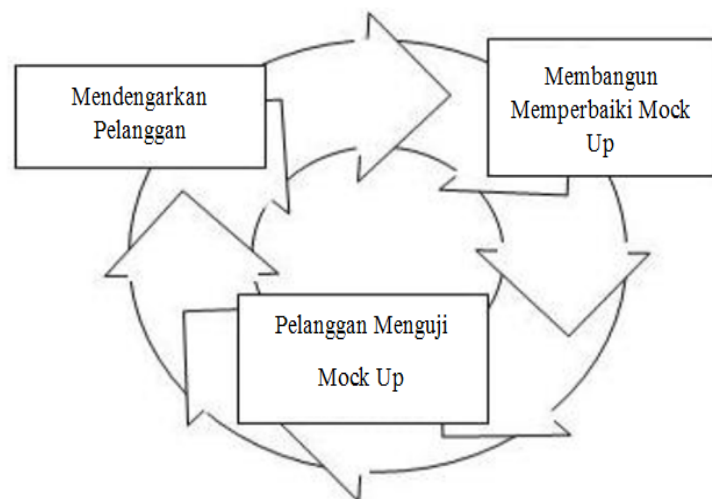
1. Modal yang diperlukan
2. Kemampuan perencanaan dan membuat produk
3. Kemampuan menentukan tingkat harga yang tepat
4. Kemampuan memilih penyalur yang tepat
5. Kemampuan dalam menggunakan cara-cara promosi yang tepat

Pada umumnya, para pengusaha mempunyai tujuan mendapatkan laba tertentu atau semaksimal mungkin, dan mempertahankan atau bahkan berusaha meningkatkannya untuk jangka panjang. Tujuan tersebut dapat direalisasikan apabila

penjualan dapat dilaksanakan seperti yang direncanakan, dengan demikian tidak berarti bahwa barang atau jasa yang terjual selalu akan menghasilkan laba, disinilah faktor-faktor diatas harus mendapat perhatian sepenuhnya oleh pihak perusahaan.

2.4 Metode *Prototype*

Model *prototype* dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa A.S dan M. Shalahuddin, 2018) . Model *prototype* dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Model *Prototype*

Berikut ini terdapat beberapa tahapan dalam proses *prototype* yaitu:

1. Mendengarkan Pelanggan

Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan pelanggan sebagai pengguna sistem perangkat lunak untuk menganalisis serta mengembangkan kebutuhan pengguna.

2. Merancang dan Membuat *Prototype*

Pada tahap ini, dilakukan perancangan dan pembuatan *prototype* sistem yang disesuaikan dengan kebutuhan pengguna.

3. Melihat atau Menguji *Mockup*

Pada tahap ini, dilakukan pengujian *prototype* sistem oleh pengguna kemudian dilakukan evaluasi sesuai dengan kekurangan-kekurangan dari kebutuhan pelanggan. Jika sistem sudah sesuai dengan *prototype*, maka sistem akan diselesaikan sepenuhnya. Namun, jika masih belum sesuai kembali ke tahap pertama.

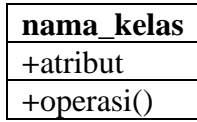



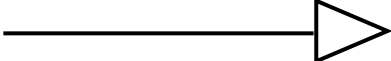

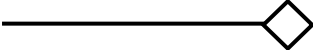
2.5 Perancangan Sistem UML (*Unified Modeling Language*)

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 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 2.1 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)
Kebergantungan/ <i>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Sumber : (Rosa and Shalahudin, 2018).

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 Tabel 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
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 actor
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



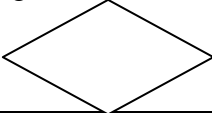


Sumber : (Rosa and Shalahudin, 2018).

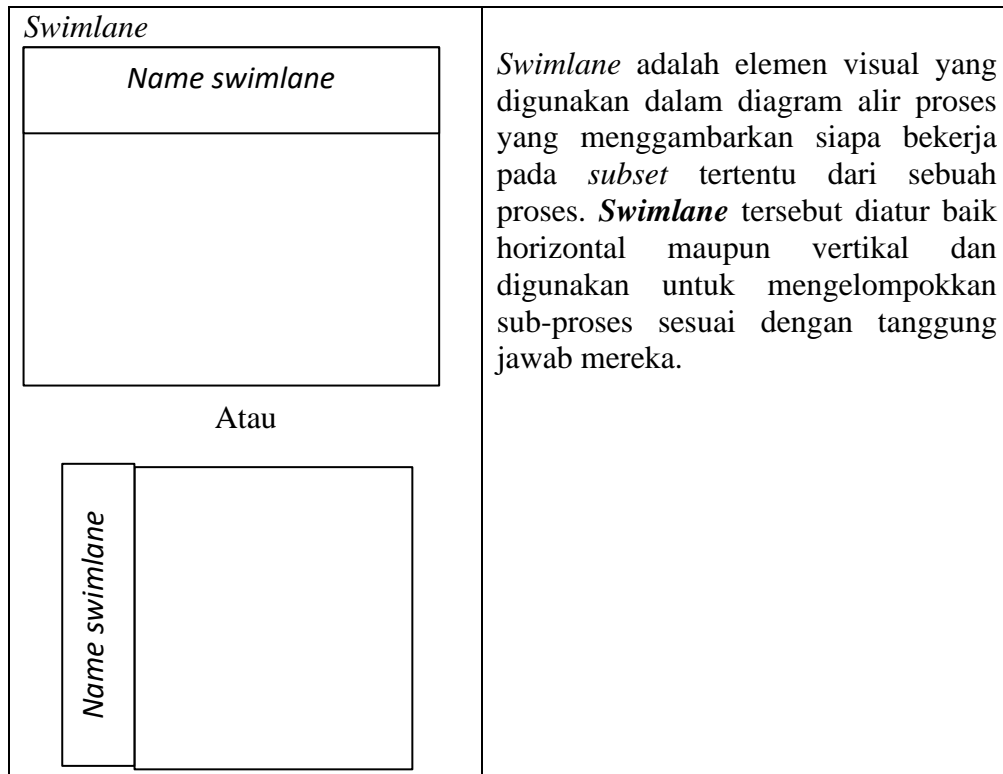
3. Activity Diagram

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 *activity* digram 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 oleh 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
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir



Sumber : (Rosa and Shalahudin, 2018).

2.6 Pengertian MySQL

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

Menurut Sheldon and Moes (2015) *MySQL* sama seperti sistem *DBMS* lainnya, seperti *Oracle*, *DB2*, dan *SQL Server*. Memungkinkan untuk mengakses, memanipulasi, melindungi, dan memelihara *metadata* yang diperlukan untuk mendefinisikan data yang disimpan.

2.7 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).

2.8 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).

2.9 XAMPP

Menurut Supono and Putratama (2016) mengemukakan bahwa XAMPP adalah suatu program yang digunakan sebagai server untuk mengeksekusi fungsi yang ada dalam halaman *website*.

XAMPP merupakan *software* yang digunakan sebagai server untuk mengeksekusi fungsi *website* tanpa jaringan internet. XAMPP terdiri dari *apache*

web server, MySQL, PHP, Perl, FTP *server* dan phpMyAdmin (Palit, Rindengan and Lumenta, 2019).

2.10 Sublime Text

Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai *platform operating system* dengan menggunakan teknologi *Python* 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 pemrograman 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 di *download* sesuai kebutuhan *user* (Sadeli, 2015).

2.11 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* 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 tingkatan 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.

2.11 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 variabel 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. 4 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. 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 : (Sugiyono, 2017)

Keterangan: x = persentase hasil pengujian.