

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Pada penelitian ini, peneliti melakukan tinjauan pustaka pada penelitian-penelitian terdahulu sebagai referensi peneliti untuk menunjang dan mempermudah dalam penelitian ini. Di bawah ini merupakan tinjauan pustaka yang telah diteliti sebelumnya dapat dilihat pada Tabel 2.1. berikut :

Tabel 2.1 Tinjauan Pustaka

No.	Detail Jurnal	
1.	Judul	Pengembangan Sistem Pengelolaan <i>Laundry</i> Berbasis <i>Web</i> (Studi Kasus : Arin <i>Laundry</i>)
	Tahun Penerbit	2021
	Penulis	Angga Dwi Putra, Dwi Sakethi, Ardiansyah
	Metode Penelitian	Metode <i>Waterfall</i>
	Hasil	Hasil dari penelitian tersebut adalah dibuatkannya sebuah sistem yang dapat membantu dan mempermudah mengelola aktivitas bisnis seperti melakukan transaksi, memonitoring pesanan pelanggan, bahan baku untuk pencucian, dan laporan pemasukan laba dan rugi.
2.	Judul	Implementasi Sistem Manajemen Dalam Pengolahan Data <i>Laundry</i> Study Kasus “Rica Utami”
	Tahun Penerbit	2019
	Penulis	Kurniawan Wahyu Haryanto, Merlin Wahyu Supriati
	Metode Penelitian	<i>System Development Life Cycle (SDLC)</i>
	Hasil	Hasil dari penelitian tersebut adalah menghasilkan sebuah sistem yang dapat membantu manajemen pengelolaan data <i>Laundry</i> dengan mudah, dimana

Tabel 2.1 Tinjauan Pustaka Lanjutan

No.	Detail Jurnal	
		dengan sistem ini dapat mempermudah pelanggan dalam pemesanan, tagihan pembayaran dan transaksinya, dan untuk pihak <i>Laundry</i> memudahkan pengelolaan data mulai dari transaksi sampai laporan harian hingga bulanan.
3.	Judul	Aplikasi “ <i>Ilufa Laundry</i> ” Online Menggunakan <i>Framework CodeIgniter</i> (Studi Kasus : <i>Ilufa Laundry</i>)
	Tahun Penerbit	2016
	Penulis	M. Solekhudin Ghozali, Naim Rochmawati
	Metode Penelitian	Metode <i>Waterfall</i>
	Hasil	Hasil dari penelitian tersebut adalah dibuatkannya sebuah sistem untuk mempermudah pelanggan melakukan transaksi agar menjadi lebih baik, dan membuat proses pelayanan lebih baik untuk pihak <i>Laundry</i> yang di mana dapat di akses di mana saja melalui internet.
4.	Judul	Sistem Informasi Pelayanan Jasa <i>Laundry</i> Berbasis <i>Web</i> Pada Tira <i>Laundry</i>
	Tahun Penerbit	2021
	Penulis	Eka Permana, Depi Yuniar
	Metode Penelitian	Metode <i>Waterfall</i>
	Hasil	Hasil dari penelitian tersebut adalah menghasilkan sebuah sistem yang mempermudah pihak <i>Laundry</i> untuk mengelola data pelanggan, data karyawan, dan data pembelian pelanggan agar lebih mudah untuk proses penerimaan data pelanggan dan pembuatan laporan.

Tabel 2.1 Tinjauan Pustaka Lanjutan

No.	Detail Jurnal	
5.	Judul	Sistem Informasi Pelayanan Jasa <i>Laundry</i> Pada Rangka <i>Laundry and Dry Cleaning</i> Berbasis <i>Web</i>
	Tahun Penerbit	2021
	Penulis	Dedi, Muchamad Iqbal, Mohamad Fallah Julyanto
	Metode Penelitian	<i>System Development Life Cycle (SDLC)</i>
	Hasil	Hasil dari penelitian tersebut adalah dibuatkannya sebuah sistem untuk pengelolaan data pesanan pada <i>Laundry</i> , dan penggunaan sistem juga membantu pihak <i>Laundry</i> untuk menyelesaikan masalah yang ada seperti ketidaksesuaian data <i>Laundry</i> sebelum dibuatkannya sebuah sistem.

2.2. Sistem

Sistem merupakan sebuah kumpulan dari komponen yang saling berkaitan dan terhubung satu sama lain untuk mencapai sebuah tujuan dalam menjalankan fungsi-fungsi utama dalam sebuah perusahaan (Permana & Yuniar, 2021). Sistem adalah sekumpulan elemen yang saling berhubungan dan bekerja sama untuk memproses masukan (*input*) yang diarahkan pada sistem tersebut dan mengolah masukan (*input*) untuk menghasilkan sebuah keluaran (*output*) sesuai dengan apa yang diinginkan (Simangunsong, 2018). Jadi dapat disimpulkan bahwa, sistem adalah suatu komponen atau elemen yang saling terhubung dengan tahapan - tahapan yang saling berinteraksi untuk saling bekerja sama sehingga mempunyai ketertarikan yang sama untuk mencapai satu tujuan yang telah ditentukan.

2.3. Informasi

Informasi adalah hasil pengolahan data, tetapi tidak semua hasil dari pengolahan dapat menjadi informasi, jika hasil pengolahan data yang tidak memberikan makna atau tidak berarti serta tidak bermanfaat bagi seseorang maka bukanlah sebuah informasi bagi orang tersebut (Simangunsong, 2018). Informasi merupakan sebuah data yang didapat lalu diolah untuk menjadi suatu bentuk yang dapat berguna dan berarti bagi penerimanya (Simargolang & Nasution, 2018). Maka dari itu dapat disimpulkan bahwa, informasi adalah suatu fakta atau data yang dikelola untuk menghasilkan sebuah bentuk pengetahuan ataupun keputusan yang berguna bagi yang menerima.

2.4. Sistem Informasi

Sistem Informasi merupakan suatu kumpulan dari komponen yang terdapat pada sebuah organisasi yang berfungsi sebagai pengolahan untuk menghasilkan suatu laporan yang disajikan kepada pihak tertentu (Destiningrum & Adrian, 2017). Sistem informasi adalah suatu sistem didalam sebuah organisasi yang saling berinteraksi secara teratur yang dapat menyediakan laporan-laporan yang diperlukan untuk mendukung pembuatan keputusan (Faulina, 2016). Jadi dapat disimpulkan bahwa, sistem informasi adalah keterhubungan antara komponen atau elemen untuk menjalankan kegiatan atau proses oprasional untuk mengelola data sesuai dengan yang dibutuhkan agar mendukung keperluan tertentu yang menghasilkan sebuah informasi ataupun data yang dibutuhkan.

2.5. *Laravel*

Laravel adalah *framework* untuk membangun sebuah perangkat lunak *web* dengan mengadaptasi bahasa pemrograman PHP dan menggunakan struktur MVC. MVC memisahkan fungsi berdasarkan komponen utama dalam pengembangan aplikasi, seperti pemrosesan data, *user interface* dan kontrol sistem aplikasi (Pandowo H, Suhasto R, Kirowati D, 2020). Oleh karena itu dapat disimpulkan bahwa, *Laravel* adalah suatu *framework* yang mempermudah pengembangan aplikasi dengan menggunakan model MVC untuk membangun sebuah perangkat lunak, MVC merupakan suatu model aplikasi yang memisahkan komponen -

komponen data dan tampilan pada aplikasi agar aplikasi yang dihasilkan lebih dinamis dan tersusun dengan baik.

2.6. *MySQL*

MySQL merupakan sebuah bentuk dari RDBMS (*Relational Database Management System*), yang telah banyak diaplikasikan secara luas di tingkat perusahaan dan sudah banyak dipakai di perusahaan bisnis kelas atas setara dengan *Microsoft SQL Server, Oracle, Sybase, SAP* dan lain-lain (Ghozali & Rochmawati, 2016). Oleh karena itu dapat disimpulkan bahwa, *MySQL* adalah perangkat lunak untuk memproses manajemen basis data dengan perintah SQL, dan merupakan salah satu dari sistem manajemen basis data yang masuk ke dalam jenis RDBMS (*Relational Database Management System*), yang di mana telah banyak diaplikasikan pada bisnis *high-end* dan dapat dikatakan setara dengan perangkat lunak manajemen basis data lain, contohnya *Oracle, Microsoft SQL Server, Sybase* dan lain-lain.

2.7. *Visual Studio Code*

Visual Studio Code merupakan sebuah kode editor yang dikembangkan oleh Microsoft untuk Windows, macOS dan Linux. Dan memiliki beberapa beberapa fitur dukungan contohnya kontrol git yang tersedia secara otomatis, *debugging*, penyorotan sintaksis, melakukan pemfaktoran kode, dan kode penyelesaian kode cerdas. Menjadikan aplikasi ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tampilan tema, pintasan keyboard, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan (Agustini & Joni Kurniawan, 2019).

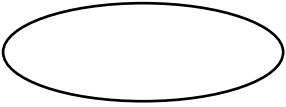
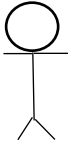


2.8. *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem peranti lunak (Sugiarti, 2018). UML menyediakan beberapa standar model untuk merancang sebuah sistem. Berikut di bawah ini adalah tiga diagram yang memiliki fungsinya masing-masing, sebagai berikut :


1. Use case Diagram

Use case diagram merupakan bentuk grafis yang menggambarkan hubungan antara sistem, sistem eksternal, dan pengguna (Sugiarti, 2018). Diagram *use case* memiliki beberapa simbol-simbol sebagai berikut :

Tabel 2.2 Simbol Use case Diagram

Simbol	Deskripsi
<p><i>Use case :</i></p> 	<p>Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor, yang di mana biasanya dinyatakan dengan menggunakan sebuah kata kerja di awal dengan frasa atau <i>use case</i>.</p>
<p>Aktor :</p> 	<p>Orang, proses, atau sistem lain yang di mana berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor berbentuk atau bergambar orang, tetapi belum tentu merupakan orang, dan biasanya dinyatakan menggunakan kata benda di awal frasa nama aktor.</p>
<p>Asosiasi :</p> 	<p>Merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p><i>Extend :</i></p> <p><<extend>></p> 	<p>Merupakan suatu relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.</p>

Tabel 2.2 Simbol *Use case Diagram Lanjutan*



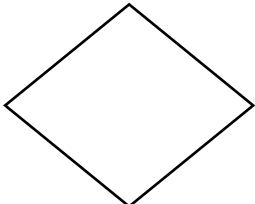
Simbol	Deskripsi
<p><i>Include :</i> <<include>></p> 	<p>Merupakan suatu relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>

Sumber : (Sugiarti, 2018)


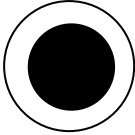
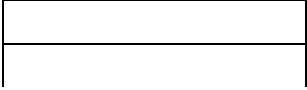
2. Activity Diagram

Activity diagram merupakan suatu penggambaran dari suatu aktivitas atau *workflow* (aliran kerja) yang terjadi pada sebuah sistem atau proses bisnis (Sugiarti, 2018). Berikut ini adalah simbol-simbol yang ada pada penggunaan *activity diagram*, sebagai berikut :

Tabel 2.3 Simbol *Activity Diagram*

Simbol	Deskripsi
<p>Mulai / Start :</p> 	<p>Merupakan sebuah status awal yang dipakai pada sebuah <i>activity diagram</i>.</p>
<p>Aktivitas :</p> 	<p>Merupakan sebuah aktivitas yang akan dilakukan oleh sebuah sistem, di mana dapat diawali dengan sebuah kata.</p>
<p>Asosiasi Percabangan :</p> 	<p>Asosiasi percabangan , adalah sebuah simbol untuk mendefinisikan jika terdapat sebuah pilihan lebih dari satu.</p>

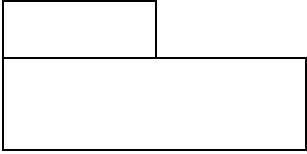
Tabel 2.3 Simbol *Activity Diagram Lanjutan*

Simbol	Deskripsi
Asosiasi Penggabungan : 	Asosiasi penggabungan, merupakan sebuah simbol yang digunakan pada <i>activity diagram</i> untuk menggabungkan lebih dari satu aktivitas yang digabungkan menjadi satu.
Akhir / Finish : 	Merupakan sebuah status akhir yang dipakai pada sebuah <i>activity diagram</i> .
Swimlane : 	Merupakan sebuah pemisah sebuah organisasi bisnis yang di mana bertanggung jawab terhadap aktivitas yang terjadi.

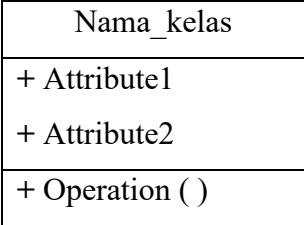
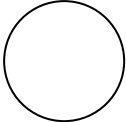



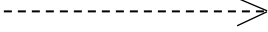
3. Class Diagram

Class diagram merupakan penggambaran dari sebuah struktur sistem dari segi pendefinisian *class-class* yang akan dibuat untuk membangun sebuah sistem (Sugiarti, 2018). *class diagram* memiliki beberapa simbol-simbol, sebagai berikut :

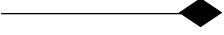
Tabel 2.4 Simbol *Class Diagram*

Simbol	Deskripsi
Package : 	Merupakan sebuah bungkusan dari satu atau lebih <i>class</i> yang dipakai.

Tabel 2.4 Simbol *Class Diagram Lanjutan*

Simbol	Deskripsi
Operasi : 	Merupakan sebuah <i>class</i> yang ada pada struktur sistem dari <i>class diagram</i> .
Antarmuka / <i>Interface</i> : 	Pada simbol <i>interface class diagram</i> , sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi : 	Merupakan suatu antar <i>class</i> dengan makna umum, di mana asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / <i>directed</i> asosiasi : 	Suatu relasi antar <i>class</i> dengan makna <i>class</i> yang satu digunakan oleh <i>class</i> lain, di mana asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Merupakan suatu relasi antar <i>class</i> dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / <i>dependency</i> : 	Merupakan suatu relasi antar <i>class</i> dengan makna kebergantungan antar <i>class</i> .

Tabel 2.4 Simbol *Class Diagram Lanjutan*

Simbol	Deskripsi
Agregasi : 	Merupakan sebuah relasi antar <i>class</i> dengan makna semua-bagian (<i>whole-part</i>).

Sumber : (Sugiarti, 2018)

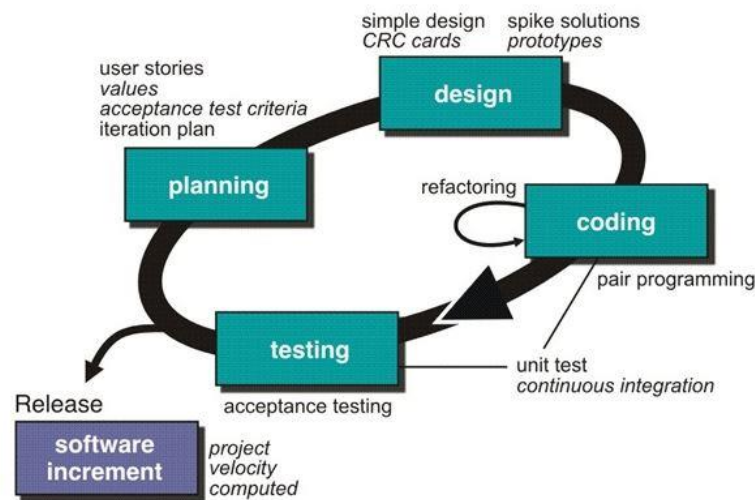
2.9. Metode Pengembangan Sistem

2.9.1. Metode *Extreme Programming* (XP)

Extreme Programming (XP) merupakan salah satu metode pengembangan perangkat lunak yang termasuk ringan dan salah satu agile methods yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. XP masuk ke dalam sebuah metode agile yang cukup banyak digunakan dan menjadi sebuah pendekatan yang sangat populer. Pada penggunaan metode XP sangat cocok pada sebuah tim yang berukuran antara kecil sampai *medium* saja, tidak perlu menggunakan sebuah tim yang besar. Hal ini bertujuan untuk menghadapi *requirements* yang tidak jelas maupun terjadinya perubahan-perubahan *requirements* yang sangat cepat (Widodo & Subekti, 2006).

Metode *Extreme Programming* sering juga disingkat dengan metode XP. Merupakan metode ini dibuat oleh Kent Beck, Ward Cunningham dan Ron Jeffries yang merupakan seorang pakar *software engineering*. *Extreme Programming* adalah suatu model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien, adaptif dan fleksibel (Fatoni & Dwi, 2016).

Dalam metode ini terdapat empat tahapan yang harus dilakukan oleh pengembang sebelum mengerjakan perangkat lunak, berikut ini adalah empat tahapan tersebut :



Gambar 2.1 *Extreme Programming (XP)*

1. *Planning* (Perencanaan)

Pada tahapan ini merupakan awal dari suatu pengembangan sistem dengan menggunakan metode XP, dalam tahapan ini menjadi tahapan paling penting karena pengembang harus merencanakan sebuah sistem secara baik, mulai dari kebutuhan sampai dengan masalah yang dialami *user*, yang di mana diharuskan mengidentifikasi permasalahannya lalu menganalisis kebutuhan yang diperlukan dari sebuah permasalahan yang ada untuk dapat diselsaikan dengan sebuah sistem, dan membuat jadwal pelaksanaan perancangan sistem sesuai dengan yang sudah ditetapkan.

2. *Design* (Perancangan)

Setelah tahapan pada perencanaan selesai, selanjutnya masuk ke dalam tahapan perancangan. Pada tahapan perancangan ini pengembang melakukan beberapa rancangan pemodelan pada hasil analisa kebutuhan yang sudah didapat, yang dimulai pada pemodelan sistem, lalu pemodelan arsitektur dan terakhir adalah pemodelan basis data.

3. *Coding* (Pengkodean)

Setelah tahapan pada perancangan selesai, selanjutnya masuk ke dalam tahapan pengkodean. Pada tahapan ini pengembang melakukan implementasi pada

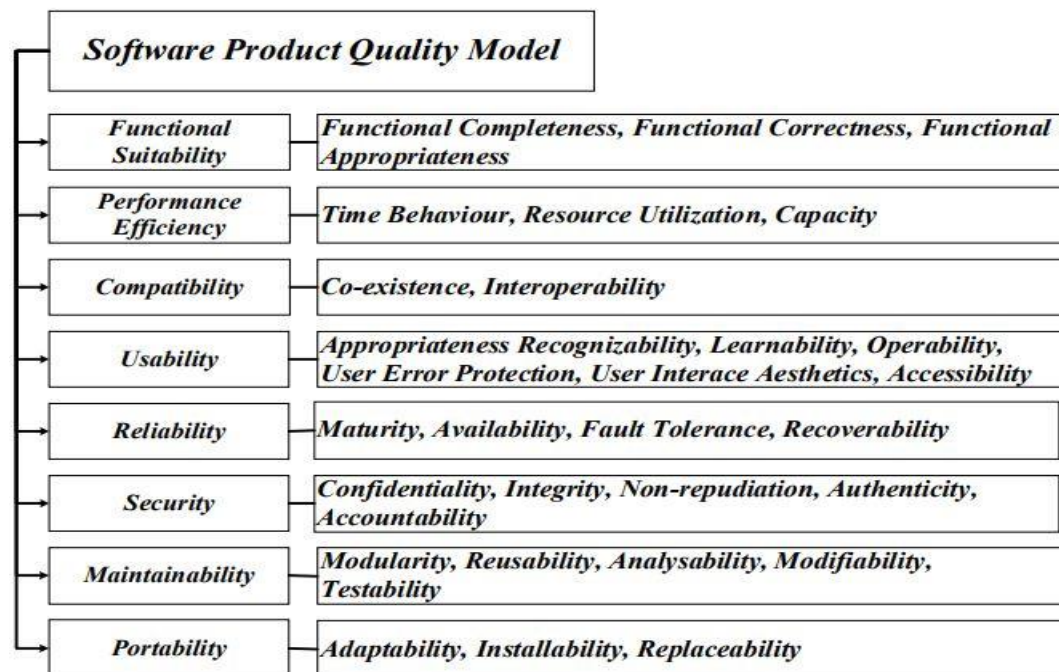
pemodelan sistem yang telah selesai dibuat, dengan melakukan pengkodean bahasa pemrograman agar dapat ditampilkan menjadi *user interface*.

4. *Testing* (Pengujian)

Tahapan ini adalah tahapan terakhir setelah selesai melakukan tahapan pengkodean. Tahapan ini merupakan tahapan pengembang melakukan pengujian terhadap sistem yang telah dibangun untuk mendapatkan apakah fungsionalitas pada sistem sudah sesuai dengan yang dibutuhkan atau masih terdapat kekurangan.

2.10. Pengujian ISO 25010

Pengujian ISO 25010 merupakan sebuah pengujian yang banyak digunakan untuk menjadi sebuah pedoman dalam mengevaluasi perangkat lunak yang merupakan pengembangan dari versi lanjutan yaitu versi ISO 9126, di mana dengan pengembangan versi ini membuat karakteristik dari kualitas model pengujian menjadi dapat dipertimbangkan ketika melakukan sebuah evaluasi pengujian perangkat lunak (Mulyawan et al., 2021). Pengujian ISO 25010 memiliki karakteristik model dalam melakukan pengujian, yang ditunjukkan pada gambar di bawah ini :



Gambar 2. 2 Model ISO 25010

Berdasarkan gambar diatas, maka dijelaskan mengenai delapan karakteristik model tersebut, adalah sebagai berikut :

1. *Functional Suitability*, merupakan sebuah karakteristik yang melakukan pengukuran sampai mana sistem atau produk menyediakan fungsi yang memenuhi kebutuhan ketika digunakan dalam kondisi tertentu.
2. *Performance Efficiency*, merupakan karakteristik untuk mengukur kinerja dari sistem atau produk terhadap sumber daya yang digunakan dalam suatu kondisi tertentu pada sebuah sistem.
3. *Compatibility*, merupakan karakteristik yang mengukur sampai mana suatu sistem dapat melakukan pertukaran informasi dengan sistem lain serta melakukan suatu fungsi yang disyaratkan saat melakukan koneksi antara perangkat keras atau perangkat lunak yang sama.
4. *Usability*, merupakan karakteristik yang mengukur sampai mana sistem dapat digunakan oleh pengguna dalam mencapai sebuah tujuan yang telah di tentukan dengan efisiensi, efektivitas dan kepuasan dalam konteks produk mudah dimengerti, mudah digunakan, dan menarik untuk digunakan.
5. *Reliability*, merupakan karakteristik untuk mengukur sampai mana sebuah sistem dapat melakukan fungsi dalam kondisi dan waktu tertentu untuk mempertahankan kinerjanya.
6. *Security*, merupakan karakteristik yang mengukur sistem atau produk dalam menjalankan proteksi terhadap data dan informasi, sehingga sistem atau produk menjadi memiliki tingkat akses data yang baik.
7. *Maintainability*, merupakan karakteristik untuk melakukan pengukuran terhadap tingkat efektivitas dan efisiensi pada suatu sistem atau produk dalam proses modifikasi yang meliputi perbaikan dan penyesuaian pada lingkungan oprasional.
8. *Portability*, merupakan karakteristik yang mewakili tingkat dari efektivitas dan efisiensi sistem atau produk dalam perpindahan dari satu perangkat ke perangkat yang lain.

2.11. Skala *Likert*

Skala *likert* merupakan skala penelitian untuk dapat mengukur sebuah sikap ataupun pendapat dari seseorang maupun sekelompok orang untuk mendapatkan jawaban tentang fenomenal *social* yang sedang diteliti. Agar dapat mengukur pendapat ataupun sikap dari suatu objek, subjek atau kejadian tertentu terdapat dua bentuk pernyataan yaitu setuju atau tidak setuju. Namun peneliti dapat menyesuaikan bentuk pilihannya sesuai dengan kebutuhan dari penelitian.