

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Tinjauan pustaka sebuah penelitian sebelumnya yang dapat digunakan sebagai pendukung penelitian yang sedang dilakukan sekarang. Peneliti telah mengumpulkan beberapa tinjauan pustaka yang dapat dilihat pada tabel 2.1 yang ada dibawah.

**Tabel 2.1** Studi Literatur

Nomor	Detail Jurnal	
1	Judul	Rancang Bangun Aplikasi Laporan Aspirasi dengan <i>Firestore Cloud Messaging</i> Berbasis <i>Mobile</i>
	Tahun Terbit	2017
	Penulis	Nurzam, Fajri dan Prabowo
	Hasil	Dalam penelitian yang berjudul, “Rancang Bangun Aplikasi Laporan Aspirasi dengan <i>Firestore Cloud Messaging</i> Berbasis <i>Mobile</i> ”, penelitian ini berguna untuk memudahkan masyarakat dalam menyampaikan aspirasi nya kepada instansi terkait, tidak hanya itu sistem ini juga guna memudahkan pihak instansi dalam mengelola laporan-laporan yang disampaikan

		oleh masyarakat sehingga dapat mengetahui kualitas kinerja instansi tersebut dengan menggunakan <i>Firestore Cloud Messaging</i> Berbasis <i>Mobile</i> .
	Kekurangan	Kekurangan dalam penelitian ini aplikasi tidak dapat memberikan laporan notifikasi ke antar lintas aplikasi seperti terbuang ke website atau aplikasi lainnya, dan juga aplikasi ini hanya dapat berjalan pada android minimal versi 5.0.
2	Judul	Mobilisasi Penyebaran Informasi Kampus Berbasis <i>Firestore Cloud Messaging</i> (FSM)
	Tahun Terbit	2019
	Penulis	Aziz, dan Airlangga
	Hasil	Dalam penelitian yang berjudul, “Mobilisasi Penyebaran Informasi Kampus Berbasis <i>Firestore Cloud Messaging</i> (FSM)”, pada penelitian ini akan menawarkan sebuah aplikasi yang dapat memberikan informasi yang <i>realtime</i> . Informasi yang dikirimkan dapat dilakukan antara admin dengan <i>user</i> begitu juga sebaliknya, aplikasi ini mendukung kebutuhan

		seperti memberikan informasi secara spesifik secara <i>realtime</i> dan memiliki fitur alarm kegiatan.
	Kekurangan	Kekurangan dalam aplikasi ini adalah penelitian yang dibangun berbasis website dimana user admin hanya dapat diakses dengan jaringan local pada kampus tersebut.
3	Judul	Penerapan <i>Firestore Realtime Database</i> pada <i>Prototype</i> Aplikasi Pemesanan Makanan Berbasis Android
	Tahun Terbit	2018
	Penulis	Paraya dan Tanone
	Hasil	Dalam penelitian yang berjudul, “Penerapan <i>Firestore Realtime Database</i> pada <i>Prototype</i> Aplikasi Pemesanan Makanan Berbasis Android”, sistem dapat membantu dalam memberikan informasi tempat makanan dan proses pemesanan makanan yang lebih mudah dan cepat dengan penerapan <i>Firestore Realtime Database</i> berbasis android.

	Kekurangan	Kekurangan pada sistem yang dirancang adalah sistem ini belum menggunakan fitur <i>push notification</i> yang ada pada <i>Firestore</i> yaitu <i>Firestore Cloud Computing</i> untuk
4	judul	<i>Realtime Notification</i> pada Aplikasi Berbasis Web Menggunakan <i>Firestore Cloud Messaging</i> (FCM)
	Tahun Terbit	2019
	Penulis	Faisol and Rahmadianto
	Hasil	Dalam penelitian yang berjudul, “ <i>Realtime Notification</i> pada Aplikasi Berbasis Web Menggunakan <i>Firestore Cloud Messaging</i> (FCM)”, penelitian ini menggunakan <i>Firestore Cloud Messaging</i> (FCM) untuk mengirimkan pesan atau notifikasi secara realtime kepada perangkat <i>client</i> , kinerja sistem manajemen surat dapat lebih efektif dan efisien sebab sistem dapat memberikan <i>notifikasi</i> pada perangkat klien selama <i>browser</i> dalam keadaan aktif atau terhubung dengan internet tanpa mengakses aplikasi.

	Kekurangan	Kekurangan pada sistem yang dibangun adalah sistem belum menggunakan <i>protocol HTTPS</i> dan hanya menggunakan <i>protocol HTTP</i> yang bersifat ( <i>not secure</i> ) untuk pengguna atau user dan data yang diinputkan user sangat rentan terhadap serangan atau <i>inject</i> .
5	Judul	<i>Push Notification Monitoring</i> Sistem Pintu Air Berbasis Android Menggunakan <i>Firestore Cloud Messaging</i>
	Tahun Terbit	2020
	Penulis	Fernando, Arini dan Fahrianto
	Hasil	Dalam penelitian berjudul, “ <i>Push Notification Monitoring</i> Sistem Pintu Air Berbasis Android Menggunakan <i>Firestore Cloud Messaging</i> ”, pada penelitian ini memanfaatkan <i>Firestore Cloud Messaging</i> dalam memonitoring pintu air waduk, dengan adanya <i>notifikasi</i> dan data yang diberikan secara <i>realtime</i> memudahkan <i>user</i> dalam mendapatkan data secara <i>realtime</i> . Informasi yang diberikan yaitu penanggulangan terhadap banjir, solusi yang diberikan, ketinggian air dan informasi lingkungan sekitar

		pintu.
	Kekurangan	Kekurangan dalam penelitian ini adalah sistem yang dibangun belum menerapkan sensor otomatis untuk <i>monitoring</i> tekanan air dan masih menggunakan bantuan manusia dalam <i>memonitoring</i> , jadi <i>push notification</i> akan aktif jika admin menekan <i>button</i> dari aplikasi tersebut

## 2.2 *Makeup Artist*

*Makeup Artist* adalah seniman *professional* yang menggunakan kulit, terutama wajah, sebagai standar karyanya dan produk *makeup* sebagai alatnya. *Makeup Artist* bisa memiliki fokus yang berbeda, misalnya riasan untuk pengantin dan acara formal lainnya, rias tradisional dan adat, *Face and Body painting*, atau *special effect* yang biasa digunakan oleh layar kaca fiksi (Mai *et al.*, 2020).

Bisnis jasa *makeup* atau tata rias yang sekarang lebih dikenal *Makeup Artis* (MUA) merupakan layanan personal yang dapat diukur melalui kinerja pada jasa atau layanan yang diberikan kepada pelanggan, profesi yang disematkan untuk seseorang yang bisa merias wajah dengan menggunakan alat dan produk kosmetik

## 2.3 *Firebase Cloud Messaging*

*Firebase* merupakan suatu layanan yang dimiliki oleh *Google* dan digunakan untuk mempermudah para *developer* aplikasi dalam mengembangkan aplikasinya. *Firebase Cloud*

*Service Provider* dan *Backend as a Service* ini merupakan solusi yang ditawarkan oleh google untuk mempermudah pekerjaan *Developer* dalam mengembangkan aplikasi *mobile* maupun *web*. Dengan adanya *Firebase*, para *developer* aplikasi bisa fokus mengembangkan aplikasi tanpa perlu membangun fitur-fitur yang dibuat pada *backend* dan infrastruktur dari awal sehingga para *developer* bisa fokus mengembangkan aplikasi yang berkualitas tinggi tanpa perlu mengeluarkan upaya yang besar (Purnomo & Purbo, 2020).

*Firebase* merupakan sebuah aplikasi pihak ketiga untuk pengelolaan *database* secara *online*, *database* ini akan dihubungkan atau di autentikasikan kedalam proses *backend*.

*Firebase Cloud Messaging* (FCM) merupakan layanan *cross-platform* untuk berkirim pesan yang disediakan oleh *google* secara gratis. FCM juga menyediakan fungsi untuk melakukan *push notification*, yaitu notifikasi yang muncul di bagian atas layar *smartphone* dan dapat diseret ke bawah, untuk mengakses pesan lengkapnya pengguna cukup menekan pesan yang tampil pada notifikasinya. Penggunaan *Fitur push notification* dengan FCM sangat membantu karena FCM akan mengirimkan notifikasi secara *realtime* (Faisol & Rahmadianto, 2019).

## **2.4 Push Notification**

*Push notification* adalah sebuah layanan yang banyak digunakan untuk keperluan pemberitahuan melalui pesan pendek yang ada di *smartphone*. Dengan adanya layanan *Push notification* tersebut, pengguna dapat terbantu dalam hal yang bersifat pemberitahuan secara singkat. Pada implementasinya *Push notification* dapat dimanfaatkan dalam berbagai keperluan sehari-hari misalnya, untuk monitoring absensi, update berita terbaru, dan sebagainya. Implementasi dari salah satu fitur yang dimiliki oleh *Firebase Cloud Messaging* ini mencakup 2 komponen utama untuk mengirim dan menerima pesan, komponen yang pertama adalah

lingkungan terpercaya seperti *Cloud Functions for Firebase* atau *server* aplikasi yang akan digunakan untuk membuat, menargetkan, dan mengirim pesan. Komponen yang kedua adalah aplikasi *client* iOS, Android, atau Web (JavaScript) yang menerima pesan (Siddik and Nasution, 2018).

*Push notification* juga disebut sebagai salah satu fitur yang ada dalam sebuah aplikasi baik itu android maupun aplikasi berbasis yang lainnya. *Push notification* ini sangat bermanfaat untuk sebuah aplikasi, dikarenakan fitur ini dapat membantu user dalam mengelola atau mendapatkan pemberitahuan bahwa dalam aplikasi tersebut telah terjadi suatu aktifitas yang melibatkan user tersebut.

## 2.5 Android

*Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan *software*. *android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka, dengan perkembangan yang semakin pesat *android* telah memiliki banyak fitur didalamnya untuk mempermudah aktivitas aktivitas pengguna *android*. Dari tahun 2008 android telah mengalami banyak perkembangan dimulai dari android versi 1.1 dengan alpha beta nya kemudian merilis versi 1.5 (*cupcake*) hingga sekarang dengan nama *android* 11. Sampai saat ini *android* telah memiliki 19 versi, versi tersebut dapat dilihat pada tabel 2.2.

**Tabel 2.2** Versi Android

Versi	Nama
Android 1.0	Alpha
Android 1.1	Beta
Android 1.5	Cupcake

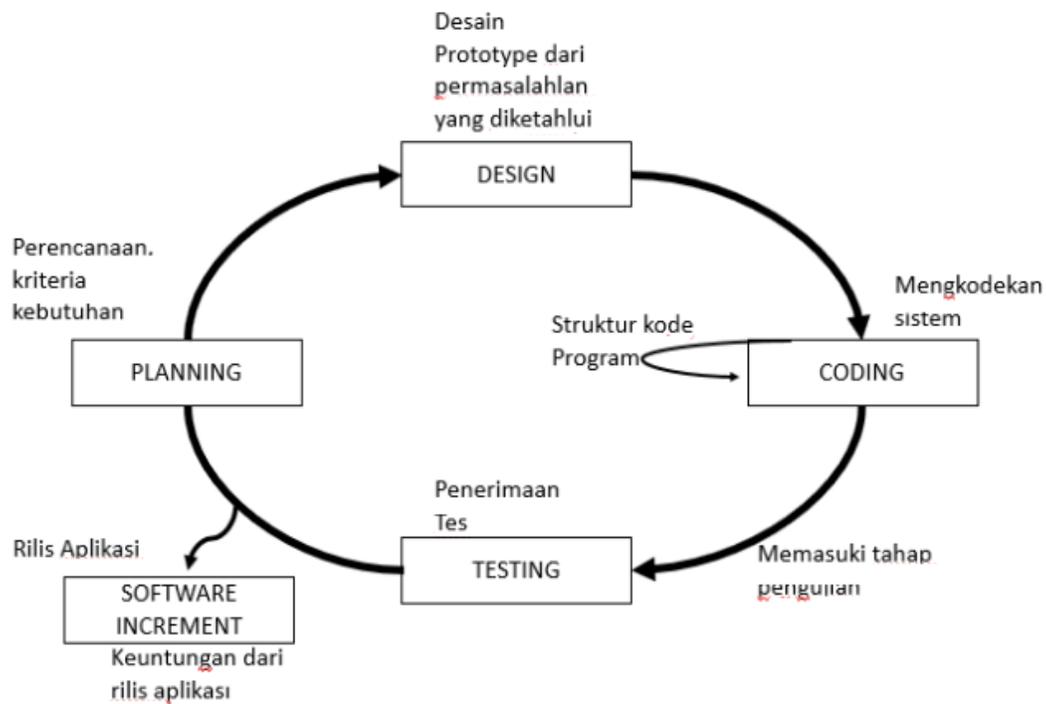
Android 1.6	Donut
Android 2.0 – 2.1	Éclair
Android 2.2	Frozen Yoghurt – Froyo
Android 2.3	Gingerbread
Android 3.0 – 3.2	Honeycomb
Android 4.0	Ice Cream Sandwich
Android 4.1 – 4.3	Jelly Bean
Android 4.4	KitKat
Android 5.0	Lollipop
Android 6.0	Marshmallow
Android 7.0 – 7.1	Nougat
Android 8.0 – 8.1	Oreo
Android 9	Pie
Android 10	Android Q
Android 11	Android 11
Android 12	Snow Cone

## 2.6 *Extreme Programming*

*Extreme Programming* (XP) merupakan metodologi yang digunakan untuk pengembangan perangkat lunak yang ditujukan dalam meningkatkan kualitas perangkat lunak terhadap perubahan serta kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini

dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pro pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Ada beberapa tahapan yang ada pada *Extreme Programming* yaitu terdiri dari Perencanaan (Planning) seperti memahami kriteria pengguna dan perencanaan pengembangan, designing seperti perancangan prototype dan tampilan, pengkodean juga termasuk dalam pengintegrasian, terakhir adalah testing (Lisa Ariyanti, 2020).

Dapat simpulkan juga *Extreme Programming* adalah suatu metode pengembangan perangkat lunak yang digunakan untuk menyederhanakan tahapan saat proses pengembangan, sehingga menjadi lebih fleksibel, adaptif, dan dapat dikerjakan oleh satu atau dua orang. Pada metode ini terdapat empat tahapan yang harus dilakukan oleh penulis sebelum mengerjakan sebuah perangkat lunak, empat tahapan tersebut adalah sebagai berikut:



**Gambar 2.1** *Extreme Programming*

(Rusman et al., 2019)

#### 1. Perencanaan

Tahapan ini merupakan tahapan yang diperlukan sebelum pengembang membuat sistem. Tahapan ini penting karena dalam membuat suatu perangkat lunak atau sebuah sistem harus direncanakan dan dianalisis kebutuhan-kebutuhan apa saja yang diperlukan user atau pengguna terlebih dahulu. Dengan cara mengidentifikasi permasalahan, kemudian menganalisis kebutuhan yang diperlukan seperti fungsi utama dan fitur-fitur lainnya, serta menetapkan jadwal untuk melaksanakan pembuatan sistem.

#### 2. Desain

Setelah selesai pada tahapan perencanaan, maka tahapan selanjutnya adalah perancangan desain sistem atau software. Pada tahapan ini penulis melakukan perancangan dengan membuat sebuah pemodelan, yang dimulai dari pemodelan sistem, kemudian pemodelan arsitektur, dan yang terakhir adalah pemodelan basis data

#### 3. Pengkodean Sstem

Setelah tahap perancangan selesai, maka tahapan selanjutnya yaitu mengkodekan sistem. Tahapan ini merupakan tahapan untuk menerapkan pemodelan yang sudah dirancang di tahapan perancangan yang sudah dibuat ke dalam bentuk user interface dan menggunakan bahasa pemrograman.

#### 4. Pengujian

Selanjutnya masuk pada bagian tahap akhir, setelah melakukan pengkodean sistem maka selanjutnya adalah tahapan testing atau pengujian sistem atau software. Pada tahapan ini penulis melakukan pengujian terhadap sistem yang sudah dibuat untuk mengetahui apakah ada kesalahan

atau yang sering disebut Bug pada sistem saat sistem tersebut dijalankan, dan untuk memeriksa sistem yang telah dibuat sudah sesuai dengan kebutuhan pengguna atau belum.

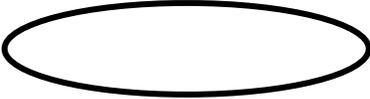
## 2.7 Unified Modelling Language

*Unified Modelling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia pengembangan sistem untuk mendefinisikan requirement, membuat analisis & design, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Josi, 2017).

## 2.8 Use Case Diagram

*User Case Diagram* merupakan pemodelan untuk perilaku sistem informasi yang akan dibuat, *use case* bekerja dengan mendeskripsikan tipikal interaksi antara user sebuah sistem dengan sistem itu sendiri melalui sebuah cerita bagaimana sistem itu dipakai. Berikut simbol atau node yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.3 berikut:

**Table 2.3** Simbol *Use Case Diagram*

Simbol	Keterangan
	<i>Use Case</i> Fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>Frase</i> nama <i>Use Case</i>
	Aktor Aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem yang akan dibuat, diluar sistem informasi. biasanya dinyatakan menggunakan

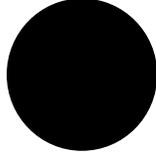
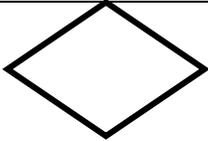
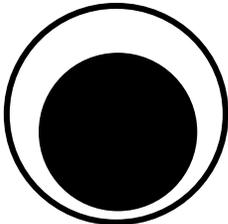
	kata benda
	<p>Asosiasi</p> <p>Asosiasi merupakan komunikasi antara aktor dengan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau secara singkat <i>use case</i> memiliki interaksi dengan aktor.</p>
	<p>Generalisasi</p> <p><i>Generalization</i> merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum.</p>
<p>&lt;&lt; Include &gt;&gt;</p> 	<p><i>Include</i></p> <p><i>Include</i> merupakan sebuah <i>use case</i> tambahan yang dimana <i>use case</i> yang dituju harus melalui sebuah proses.</p>
<p>&lt;&lt; Extend &gt;&gt;</p> 	<p><i>Extend</i></p> <p><i>Extend</i> merupakan sebuah <i>use case</i> tambahan yang dimana <i>use case</i> yang dituju berdiri sendiri tanpa harus dilalui oleh sebuah proses.</p>

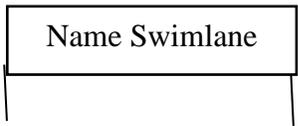
(Winda & Umi , 2016)

## 2.9 Activity Diagram

*Activity diagram* adalah *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.4 berikut ini:

**Tabel 2.4** Activity Diagram

Simbol	Keterangan
	Status Awal  Sebuah diagram aktivitas yang memiliki status awal.
	Aktivitas, yang dilakukan oleh sistem, yang biasanya diawali dengan kata kerja.
	Decision  asosiasi percabangan, dimana jika ada pilihan aktivitas lebih dari satu.
	Join  asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Status akhir, sebuah diagram aktivitas memiliki sebuah status akhir.

	<p>Swimlane</p> <p>memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>
-----------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------

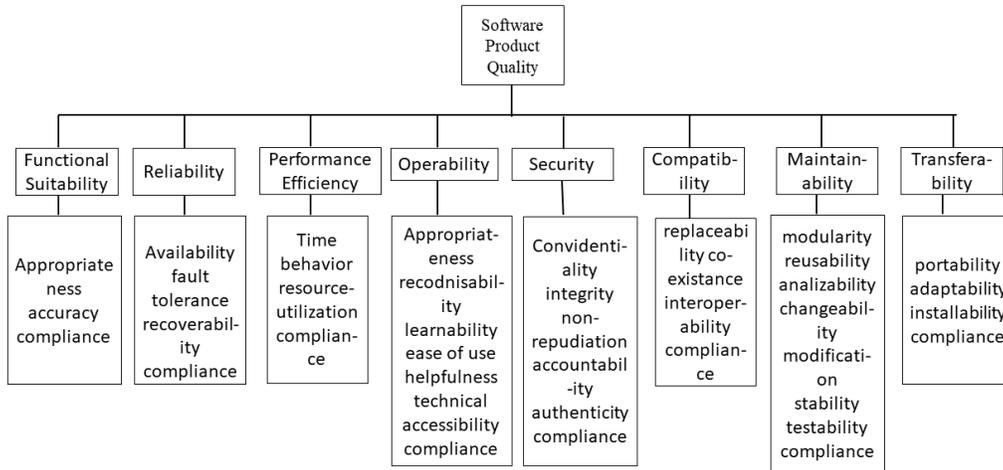
(Winda & Umi , 2016)

## 2.10 Pengujian ISO 25010

ISO 25010 merupakan bagian dari Systems and Software Quality Requirements and Evaluation (SQuaRE) yang merupakan versi lanjutan dari ISO 91261, yang telah direvisi secara teknis dengan menambahkan beberapa struktur dan bagian dari standar model kualitas. Tujuan dari penggunaan kualitas ini adalah untuk mengukur sejauh mana produk atau sistem tersebut bisa digunakan oleh pengguna untuk memenuhi kebutuhan dalam mencapai tujuan yang diinginkan dengan efisiensi, efektivitas, kepuasan dalam konteks penggunaan yang spesifik, dan bebas dari resiko (Wattiheluw, Rochimah and Fatichah, 2019).

Jadi dapat disimpulkan juga bahwa pengujian ISO 25010 ini merupakan pengembangan dari pengujian sebelumnya yaitu pengujian ISO 91261, dan merupakan standar pengujian yang digunakan untuk pembuatan karya ilmiah,

Menurut (Harun, 2018) “terdiri dari delapan karakteristik yang dibagi menjadi beberapa bagian yang berhubungan dengan sifat-sifat statis perangkat lunak dan sifat dinamis dari sistem komputer, yang dapat ditunjukkan pada gambar dibawah ini:”



**Gambar 2. 2** Model ISO 25010

(Harun, 2018)

Berdasarkan gambar diatas, dapat dijelaskan mengenai delapan karakteristik tersebut, sebagai berikut:

1. *Functional Suitability*, merupakan sistem atau produk yang memberikan fungsional untuk memenuhi kebutuhan saat sistem atau produk tersebut digunakan pada keadaan tertentu.
2. *Reliability*, merupakan tingkat dimana suatu sistem dapat mempertahankan kinerjanya pada level tertentu ketika digunakan pada keadaan tertentu.
3. *Performance Efficiency*, merupakan tingkat dimana sistem menyediakan performa yang baik dengan sejumlah *resource* yang akan digunakan pada sistem atau produk.
4. *Usability*, merupakan tingkat dimana pada suatu sistem atau produk mudah dimengerti, mudah dipakai, dan menarik untuk digunakan.
5. *Security*, merupakan tingkat dimana pada suatu sistem menyediakan layanan untuk melindungi akses, ataupun pengungkapan yang berbahaya.
6. *Compatibility*, merupakan kemampuan pada suatu komponen atau sistem untuk bertukar informasi.

7. *Maintainability*, merupakan tingkat dimana pada suatu sistem dapat dimodifikasi, yang meliputi perbaikan, pengembangan untuk menyesuaikan dengan lingkungan, modifikasi pada kriteria, dan spesifikasi fungsi.
8. *Portability*, merupakan tingkat dimana pada suatu sistem dapat dipindahkan dari satu ruang ke ruang lainnya.

### 2.11 Skala Likert

Menurut (Pranatawijaya *et al.*, 2019), Skala Likert adalah “ukuran yang digunakan untuk mengukur pendapat seseorang atau kelompok mengenai sebuah peristiwa atau kejadian. Pertanyaan yang digunakan disebut dengan *variable* penelitian.”

Skala Likert mempunyai kriteria nilai jawaban. Kriteria tersebut digambarkan dalam bentuk tabel seperti dibawah ini

**Tabel 2.5** Kriteria Nilai

Bobot	Keterangan
5	Sangat Setuju (SS)
4	Setuju (S)
3	Netral (N)
2	Tidak Setuju (TS)
1	Sangat Tidak Setuju (STS)

(Pranatawijaya, et al., 2019)

Skala Likert juga mempunyai sebuah interval penilaian. Interval Penilaian tersebut digambarkan dalam bentuk tabel seperti dibawah ini

**Tabel 2.6** Interval Nilai

Indeks	Kriteria
0% - 19%	Sangat tidak Baik
20% - 39%	Tidak Baik
40% - 59%	Cukup
60% - 79%	Baik
80% - 100%	Sangat Baik

(Pranatawijaya, et al., 2019)

Pada penggunaan Skala Likert terdapat suatu rumus penilaian. Berikut rumus yang digunakan dalam perhitungan skala likert:

$$\text{Hasil} = \frac{\text{Skor Diperoleh}}{\text{Skor Maksimal}} \times 100\%$$