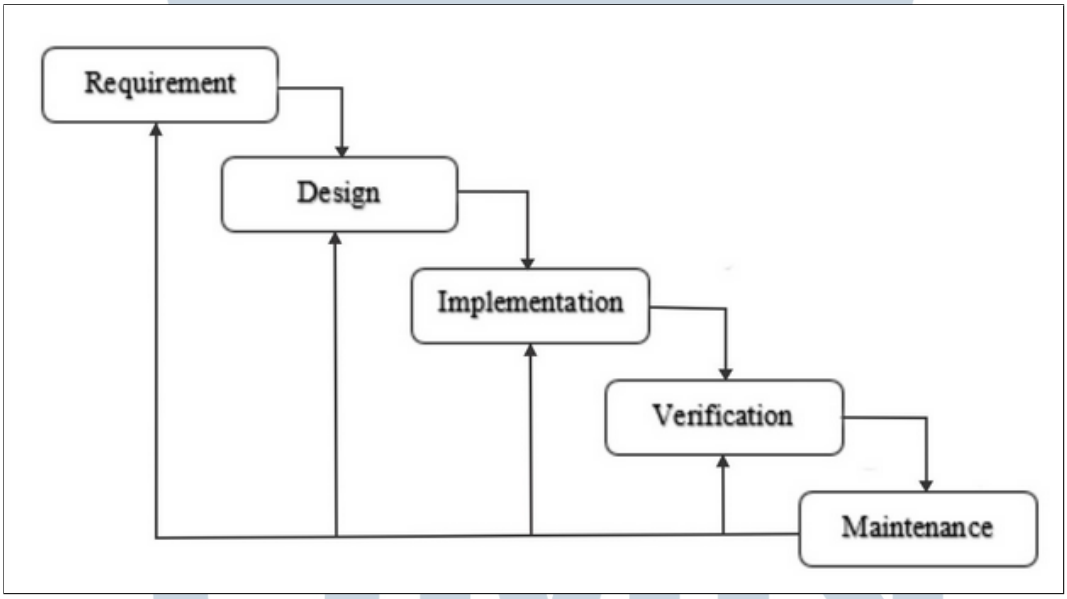
# BAB 2
# LANDASAN TEORI

## 2.1 Metode Waterfall

Metode Waterfall adalah metode pengembangan perangkat lunak atau yang biasa dikenal juga dengan istilah *Software Development Life Cycle* (SDLC). Metode Waterfal merupakan SDLC tertua sebab sifatnya yang natural dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis (*requirement*), desain atau perencanaan, implementasi atau penulisan kode, *testing/verification*, dan *maintenance*.[3]




Gambar 2.1. Tahapan Metode Waterfall

Sumber: Pressman, 2012

1. Analisis Kebutuhan (Requirement)
   Tahap ini merupakan tahap analisis kebutuhan sistem untuk mengetahui dan memahami bagaimana informasi kebutuhan penggguna terhadap sebuah aplikasi. Informasi yang diperoleh lalu diolah dan dianalisa sehingga didapatkan data yang lengkap mengenai kebutuhan atau keperluan pengguna pada perangkat lunak yang akan dikembangkan.

2. Desain
   Pada tahap ini, informasi atau data yang didapat pada tahap analisis kebutuhan

dianalisa kemudian di implementasikan menjadi sebuah rancangan atau desain pengembangan. Perancangan desain bertujuan untuk mendapatkan gambaran lengkap mengenai apa yang harus dikerjakan pada tahap implementasi atau penulisan kode nantinya. Pada tahap ini juga pengembang merencanakan atau menyiapkan kebutuhan hardware dalam pembuatan arsitektur sistem perangkat lunak yang akan dibuat.

3. Implementasi
   Pada tahap ini, hasil dari desain akan direalisasikan dalam bentuk modul-modul berupa kode untuk membentuk sebuah perangkat lunak sesuai dengan rancangan desain sebelumnya. Perangkat lunak yang sudah selesai dibuat akan dilakukan pengujian dan pemeriksaan untuk mengetahui apakah perangkat lunak sudah memenuhi kriteria/spesifikasi yang diinginkan.

4. Verifikasi
   Setelah perangkat lunak diuji di tahap implementasi, akan dilakukan proses integrasi pada sistem. Kemudian dilaksanakan pemeriksaan dan pengujian sistem dan perangkat lunak secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan pada perangkat lunak.

5. Maintenance
   Pada tahap ini, perangkat lunak yang sudah jadi akan dilakukan pemeliharaan. Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan, peningkatan, dan penyesuaian sistem sesuai dengan kebutuhan pengguna.

Kelebihan dari metode Waterfall adalah sebagai berikut [3]:

1. Penerapan sistem tidak berbelit-belit.
2. Meminimalisir terjadinya kesalahan karena pada proses pengembangannya dilakukan secara urut berstruktur.
3. Dokumentasi yang lengkap dan mudah untuk dipahami semua orang.

Kekurangan dari metode Waterfall adalah sebagai berikut [3]:

1. Dalam proses pengembangannya membutuhkan waktu yang lama.
2. Apabila terjadi kesalahan pada suatu tahap/proses akan sulit untuk melakukan pengulangan dan dapat berakibat pada tahap selanjutnya.

## 2.2 Android

Android merupakan operasi sistem yang dipakai khusus untuk perangkat ponsel atau *smartphone*. Android sendiri merupakan operasi sistem yang bersifat terbuka (*open source*) sehingga banyak orang dapat bereksperimen untuk mengembangkan Android.[4] Operasi sistem ini juga sudah terintegrasi dengan banyak perangkat dan sistem, sehingga memudahkan pengguna dalam mengembangkan sekaligus memakai operasi sistem tersebut.

## 2.3 Retrofit

Retrofit adalah *library* untuk bahasa pemrograman android yang berfungsi untuk menangani semua hal yang berkaitan dengan koneksi data dari Android ke internet dan atau sebaliknya. Kelebihan dari *library* ini adalah mudahnya untuk digabung dengan *library* lain serta dapat dikostumisasi sesuai kebutuhan, sehingga bisa retrofit sangat membantu untuk mempercepat pembuatan project.[5]

## 2.4 Firebase

Firebase adalah platform pengembangan aplikasi yang membantu dalam membangun dan mengembangkan aplikasi dan game multi platform. Firebase didukung oleh Google dan telah dipercaya oleh jutaan bisnis di seluruh dunia. Firebase sendiri memiliki banyak fitur dan fungsi yang dapat membantu pengguna dalam mengembangkan aplikasinya khususnya dalam hal *back-end* seperti Firebase Authentication dan Firebase Firestore Database.[6]

### 2.4.1 Firebase Authentication

Firebase Authentication adalah fitur dari Firebase yang berfungsi untuk membantu pengguna dalam hal autentikasi pengguna pada aplikasi yang dibangun. Firebase Authentication telah menyediakan banyak pilihan untuk autentikasi pengguna seperti *login* dan *register* menggunakan alamat *e-mail*, nomor telepon, akun Google, dan masih banyak lagi. Dilengkapi dengan berbagai fitur otomatis seperti *forget password*, pengaturan unik 1 *email* per akun, dan lain-lain telah membuat Firebase Authentication menjadi pilihan utama yang dipakai banyak pengembang aplikasi di seluruh dunia.[7]

### 2.4.2 Firebase Firestore Database

Firebase Firestore Database atau bisa disebut juga Cloud Firestore merupakan *database* untuk dokumen NoSQL yang dapat digunakan untuk menyimpan, menyinkronkan, dan membuat kueri data dengan mudah dan cepat di aplikasi seluler dan web dalam skala global. Firebase Firestore Database ini juga telah terintegrasi dengan banyak bahasa pemrograman dan platform sehingga memudahkan pengguna untuk mengembangkan aplikasinya.[8]

## 2.5 Node JS

Node JS adalah *runtime environment* untuk JavaScript yang bersifat *open-source* dan *cross-platform*. Dengan Node JS pengguna dapat menjalankan kode JavaScript di mana saja tanpa terbatas pada lingkungan *browser*, Node JS menjalankan V8 JavaScript engine (yang juga merupakan inti dari Google Chrome) di luar browser menyebabkan Node JS memiliki performa yang sangat tinggi. Node JS juga menyediakan banyak *library/module* JavaScript yang membantu menyederhanakan pengembangan aplikasi web seperti React JS dan Express JS.[9]

### 2.5.1 React JS

React JS merupakan sebuah *library* JavaScript yang digunakan untuk membangun tampilan sebuah web atau aplikasi web. React sendiri dirilis oleh facebook(sekarang meta) guna membangun sebuah tampilan web yang kaya dan interaktif. React JS juga merupakan *library* JavaScript yang bersifat *open source* sehingga isinya akan terus berkembang karena semua orang bebas memodifikasi kode di dalamnya. Dengan 2 fitur unggulan yang menjadi *signature* dari ReactJS yaitu JSX dan Virtual DOM. JSX adalah sebuah *extension syntax* JavaScript yang memungkinkan *programmer* untuk memodifikasi Document Object Model (DOM) dengan kode bergaya HTML, DOM adalah *application programming interface*(API) yang berfungsi untuk mengatur atau memodifikasi struktur halaman web dengan mudah. Virtual DOM adalah salinan dari DOM asli yang ingin diperbarui/modifikasi, Virtual DOM ini berguna untuk melihat bagian dari DOM asli yang berubah sehingga *programmer* tidak perlu melakukan *reload* satu halaman web untuk melihat perubahan yang terjadi.[10]

### 2.5.2 Express JS

Express JS adalah *framework open source* aplikasi web untuk Node JS yang ditulis dengan bahasa pemrograman JavaScript dan diciptakan oleh TJ Holowaychuk pada tahun 2010. Express Js adalah *framework backend* yang bertanggung jawab untuk mengatur fungsionalitas web, seperti pengelolaan *routing* dan *session*, permintaan HTTP, penanganan error, serta pertukaran data di server. Express JS menggunakan pendekatan *unopinionated* dalam proses pengembangannya dimana pengguna punya kebebasan dalam menentukan metode yang akan digunakan untuk mengeksekusi suatu perintah serta dapat menentukan sendiri model arsitektur yang akan dikembangkan.[11]

### 2.5.3 Express Rate Limiter

*Express Rate Limiter* merupakan sebuah *library/module* yang dipakai untuk membatasi penggunaan *Application Programming Inteface* (API) oleh pengguna/*ip address* secara berlebihan dalam kurun waktu yang singkat atau ditentukan pengembang aplikasi. API merupakan interface yang dapat menghubungkan satu aplikasi dengan aplikasi lain. API bisa digunakan untuk komunikasi dengan berbagai bahasa pemrograman sehingga mempermudah pengembang dalam mengembangkan sebuah platform.[12]

### 2.5.4 Crypto

Crypto adalah sebuah *library/module* yang mendukung teknik kriptografi. *Library* ini menyediakan beberapa fungsi kriptografi yang mencakup satu atau sekumpulan set seperti fungsi *SSL's hash HMAC, cipher, decipher, sign, & verify* pada suatu data dan atau kunci rahasia.[13]

## 2.6 Metode One-Time Password (OTP)

### 2.6.1 One-Time Password

*One Time Password* (OTP) merupakan algoritma yang yang menciptakan kata sandi satu kali pakai. OTP dianggap lebih aman daripada kata sandi yang biasa digunakan karena kata sandi OTP terus berubah sehingga kata sandi OTP tidak rentan terhadap serangan ulangan ataupun kata sandi yang dicuri. Kata sandi OTP

juga biasa digunakan sebagai mekanisme otentikasi tambahan atau sering disebut juga sebagai *two-factor authentication*.[14] Terdapat dua cara untuk mendapatkan kode OTP ini yaitu:

**A Token perangkat lunak**

Seperti Google Authenticator, dimana aplikasi atau web yang terintegrasi akan menampilkan kode OTP untuk digunakan pada formulir yang dibutuhkan.

**B Token perangkat keras**

Seperti Key BCA atau perangkat token lainnya yang dapat secara otomatis menghasilkan kode OTP.

### 2.6.2 Hmac-based One-Time Password (HOTP)

Algoritma *Hmac-based One-Time Password* (HOTP) adalah algoritma yang menggunakan kunci rahasia bersama dan faktor bergerak sebagai hal utama. Kunci rahasia bersama merupakan sebuah string yang digunakan sebagai kunci rahasia yang diketahui kedua belah pihak, sedangkan faktor bergerak adalah bilangan yang selalu bertambah apabila sebuah transaksi atau OTP baru dihasilkan sehingga kata sandi OTP baru yang dihasilkan akan berbeda-beda. Algortima HOTP menggunakan algoritma HMAC-SHA-1 yang merupakan salah satu tipe dari MAC (*Message Authentication Code*) yang berbasis kriptografi hash satu arah. HMAC digunakan untuk memeriksa integritas dan autentikasi data dari message yang dikirimkan. Berikut adalah bentuk umum dari HOTP:

$$HOTP(K,C) = Truncate(HMAC-SHA-1(K,C)) \tag{2.1}$$

Dimana *truncate* adalah fungsi untuk mengkonversi nilai HMAC-SHA-1 menjadi nilai HOTP, K sebagai kunci rahasia bersama dan C adalah faktor bergerak.[15]

### 2.6.3 Time-based One-Time Password (TOTP)

*Time-based One Time Pasword* (TOTP) adalah kata sandi sementara yang memberikan otentikasi untuk akses pengguna. TOTP juga menggunakan kunci rahasia bersama dan faktor bergerak namun pada TOTP faktor bergerak ini akan

terus berganti dalam kurun waktu tertentu biasanya 30-60 detik. TOTP memiliki banyak cara seperti keamanan perangkat keras (token), aplikasi seluler, pesan teks, dan masih banyak lagi lainnya. TOTP biasanya terdiri dari 8 digit kode angka yang akan berlaku selama 30 sampai 60 detik yang akan terus berubah dalam kurun waktu yang ditentukan. TOTP Ini dihasilkan dari kunci rahasia bersama dan Zona Waktu saat ini yang menggunakan HMAC (berbasis Hash Message Authentication Code) sehingga disebut juga sebagai HOTP. Berikut adalah bentuk dari TOTP:

$$TOTP = HOTP(K,T) \tag{2.2}$$

Dimana K adalah kunci rahasia bersama dan T adalah nilai integer yang merepresentasikan jumlah langkah waktu antara waktu penghitung awal ($T_0$) dan waktu saat ini yang dihitung menggunakan fungsi:

$$T = \frac{T_{\text{Current}} - T_0}{X} \tag{2.3}$$

$T_{\text{Current}}$ adalah waktu saat ini dalam satuan detik dan X merupakan waktu yang mendefinisikan berapa lama TOTP akan bisa digunakan biasanya 30 sampai 60 detik.[15]

Keunggulan menggunakan TOTP dibanding metode serupa:

1. Dapat digunakan sebagai *soft token*
   Autentikator TOTP dapat disematkan di token perangkat keras khusus maupun diimplementasikan dalam perangkat lunak, biasanya sebagai aplikasi seluler seperti Google Authenticator. Dengan menerapkannya dalam perangkat lunak, dapat menghindari biaya yang terkait dengan pembuatan, distribusi, inventaris, dan pemeliharaan perangkat keras sehingga dapat menghemat biaya.

2. Tidak memerlukan koneksi internet
   Perangkat yang menghasilkan dan menerima kode TOTP dapat sepenuhnya *offline*. Selama kedua perangkat berbagi kunci rahasia yang sama dan disinkronkan, mereka dapat menghasilkan kode TOTP satu per satu dan membandingkannya satu sama lain.

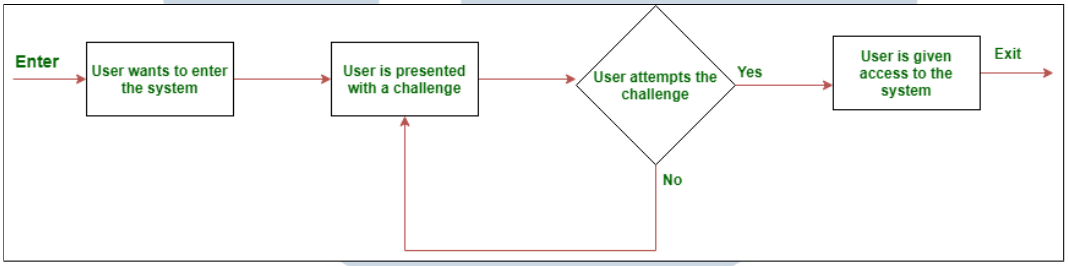
3. Mudah digunakan di seluruh aplikasi dan saluran TOTP dapat digunakan atau berfungsi untuk berbagai jenis aplikasi dan saluran.

4. Proteksi tambahan

   TOTP memiliki proteksi tambahan yaitu kode yang memiliki masa kadaluarsa.

## 2.7 Challenge Response Authentication Mechanism (CRAM)

*Challenge Response Authentication Mechanism* (CRAM) adalah cara yang paling sering digunakan untuk melakukan otentikasi. CRAM adalah sekelompok protokol di mana satu sisi menyajikan tantangan (untuk dijawab) dan sisi lain harus memberikan jawaban yang benar (untuk diperiksa/divalidasi) untuk tantangan agar diautentikasi.[16]




Gambar 2.2. Diagram Alir CRAM

Sumber: GeeksforGeeks

Dalam CRAM, terdapat 2 jenis pertanyaan yaitu[16]:

(a) *Static*, seperti namanya, melibatkan pendekatan statis untuk pemilihan tantangan. Pengguna dapat memilih tantangannya dan mengotentikasi dirinya sendiri dengan kunci atau sandi rahasia yang tidak berganti-ganti atau memiliki jawaban yang tetap. Contoh: Penggunaan fitur *forgot password*, dimana pengguna diminta untuk memasukkan jawaban rahasia yang sudah tetap seperti nama ibu kandung pengguna.

(b) *Dynamic*, melibatkan pendekatan dinamis terhadap seleksi dan otentikasi tantangan. Tantangan dipilih secara acak dengan asumsi bahwa pengguna akan mengetahui jawaban yang *valid* untuk tantangan mengingat pengguna adalah yang sebenarnya. Contoh: Penggunaan *One-Time Password* dimana kode hanya bisa digunakan sekali untuk setiap otentikasi.

Dengan 2 jenis pertanyaan di atas, terdapat beberapa cara dalam mengeksekusi CRAM yaitu[16]:

(a) CAPTCHA, digunakan untuk mencegah spam dan pendaftaran otomatis akun baru untuk email atau situs web yang secara otomatis dijalankan oleh program untuk membedakan antara komputer atau manusia.

(b) SSH *(Secure Shell)*, protokol jaringan kriptografi untuk mengoperasikan layanan jaringan secara aman melalui jaringan yang tidak aman.

(c) *Password*, protokol menggunakan kata sandi yang dikirim ke *server* untuk validasi dengan mencocokkan dengan kata sandi yang benar.

(d) *Salted Challenge Response Authentication Mechanism* (SCRAM), tantangan dengan tambahan *hash* untuk memastikan kata sandi digunakan hanya untuk satu kali saja. *Hash* dikirim ke *server* untuk dicocokkan dengan *hash* kata sandi yang benar untuk kecocokan dan bukan kecocokan kata sandi teks biasa itu sendiri.

(e) *Biometrics*, protokol yang menggunakan bagian tubuh manusia sebagai identikasi seperti retina, sidik jari, wajah, dan lain-lain. Setiap kali pengguna ingin mengautentikasi dirinya sendiri, dia harus menunjukkan kredensial biometriknya ke sistem autentikasi untuk validasi.

## 2.8 Github

GitHub merupakan *manajemen project*, sistem pemberian versi kode, sekaligus *platform* jaringan sosial bersifat terbuka (*open source*) bagi para developer seluruh dunia. Github memudahkan pengembang dalam mengembangkan sebuah karya baik secara individual maupun dalam suatu tim.[17]

## 2.9 Vercel

Vercel merupakan salah satu penyedia layanan *hosting* gratis. Vercel dilengkapi dengan berbagai fitur yang mempermudah penggunanya dalam melakukan kegiatan terkait *hosting* di vercel yang telah terintegrasi dengan berbagai macam *framework* dan bahasa pemrograman. Salah satu keunggulan vercel yaitu terdapatnya fitur *deployment* otomatis berdasarkan versi terbaru *repository* github yang dipilih untuk di*hosting*.[18]

## 2.10 Black Box Testing

*Black Box Testing* atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil masukan dan keluaran dari perangkat lunak atau aplikasi tanpa mengetahui struktur kode tersebut. Pada metode ini fokus menguji fungsi-fungsi dari aplikasi dimana metode ini memiliki tujuan untuk memeriksa fungsi dari aplikasi atau perangkat lunak berfungsi dengan baik serta bekerja secara efisien. *Black Box Testing* memiliki banyak sekali keunggulan yaitu [19]:

(a) Pengujian dilakukan dalam sudut pandang pengguna.

(b) Efisien untuk sekmen kode yang besar.

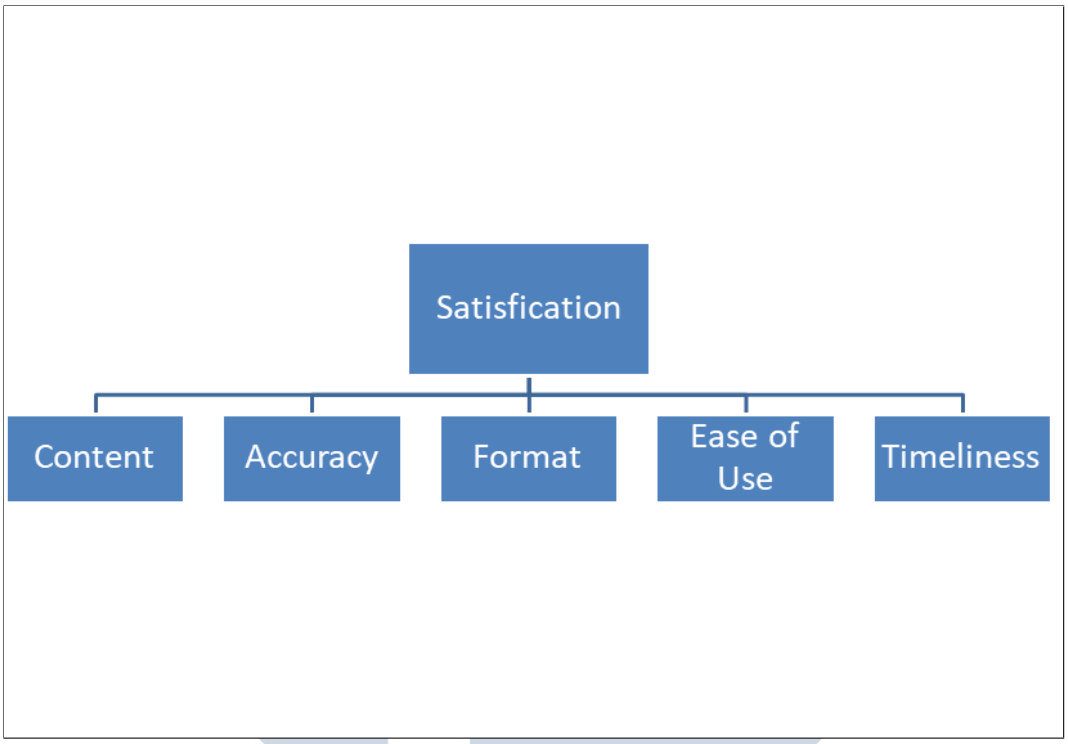
(c) Akses kode tidak diperlukan

(d) Pemisahan antara perspektif pengguna dan pengembang sehingga tidak menganggu proses kerja satu sama lain.

## 2.11 End User Computing Satisfaction

*End User Computing Satisfaction* adalah metode untuk mengukur atau evaluasi secara keseluruhan dari para pengguna sistem informasi yang berdasarkan pengalaman mereka dalam menggunakan sistem tersebut. Evaluasi dengan menggunakan model ini lebih menekankan kepuasan pengguna terhadap aspek teknologi, dengan isi, keakuratan, *format*, waktu dan kemudahan penggunaan dari sistem.[20]

UNIVERSITAS MULTIMEDIA NUSANTARA




Gambar 2.3. Model Evaluasi *End User Computing Satisfaction*

Berikut adalah beberapa penjelasan singkat dari tiap dimensi yang diukur dengan metode *End User Computing Satisfaction* [20]:

(a) Dimensi *Content*, mengukur kepuasan pengguna dari sisi isi dari suatu sistem seperti fungsi dan modul yang dapat digunakan oleh pengguna. Dimensi ini juga mengukur apakah sistem menghasilkan informasi yang sesuai dengan kebutuhan penggunanya.

(b) Dimensi *Accuracy*, mengukur kepuasan dari sisi keakuratan data. Keakuratan sistem diukur dengan melihat seberapa sering sistem melakukan kesalahan atau mengalami kendala dalam *output* yang dikeluarkan.

(c) Dimensi *Format*, mengukur kepuasan pengguna dari sisi tampilan dan estetika dari antarmuka sistem atau aplikasi. Dimensi ini mengukur apakah tampilan dari aplikasi ini menarik dan mudah digunakan pengguna atau tidak.

(d) Dimensi *Ease of Use*, mengukur kepuasan pengguna dari sisi kemudahan pengguna dalam menggunakan sistem atau aplikasi seperti pada proses memasukan, mengolah, dan atau mencari data yang dibutuhkan.

(e) Dimensi *Timeliness*, mengukur kepuasan pengguna dari sisi ketepatan waktu sistem dalam menampilkan data dan informasi yang dibutuhkan pengguna. Dimensi ini juga memperhatikan setiap permintaan yang dilakukan pengguna apakah akan langsung diproses atau ditampilkan secara cepat tanpa menunggu lama.

## 2.12 Skala Likert

Skala Likert merupakan suatu skala yang dapat digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang suatu gejala atau fenomena sosial. Skala ini memuat item yang diperkirakan sama dalam beban nilainya, subjek merespon dengan berbagai tingkat intensitas berdasarkan skala antara dua sudut berlawanan yang mempunyai gradasi dari sangat positif sampai dengan sangat negatif yang dapat berupa kata-kata seperti Sangat Setuju-Sangat Tidak Setuju.[21]

Tabel 2.1. Kriteria Skala Likert

| **Kategori** | **Kriteria** | **Syarat** |
|---|---|---|
| SS | Sangat Setuju | $P \geq 80\%$ |
| S | Setuju | $60\% \leq P < 80\%$ |
| N | Netral | $40\% \leq P < 80\%$ |
| TS | Tidak Setuju | $20\% \leq P < 80\%$ |
| STS | Sangat Tidak Setuju | $0\% \leq P < 80\%$ |

Bedasarkan Tabel Kriteria Skala Likert diatas, Rumus yang digunakan untuk menghitung presentase skor adalah sebagai berikut [21]:

$$T = \frac{(SS*5)+(S*4)+(N*3)+(TS*2)+(STS*1)}{5+n} * 100\% \quad (2.4)$$

Dengan variabel yang diketahui sebagai:

P = presentase skor skala likert

SS = jumlah jawaban sangat setuju

N = jumlah jawaban netral

TS = jumlah jawaban tidak setuju

STS = jumlah jawaban sangat tidak setuju

n = jumlah responden

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA