

## **BAB III**

### **METODE PENELITIAN**

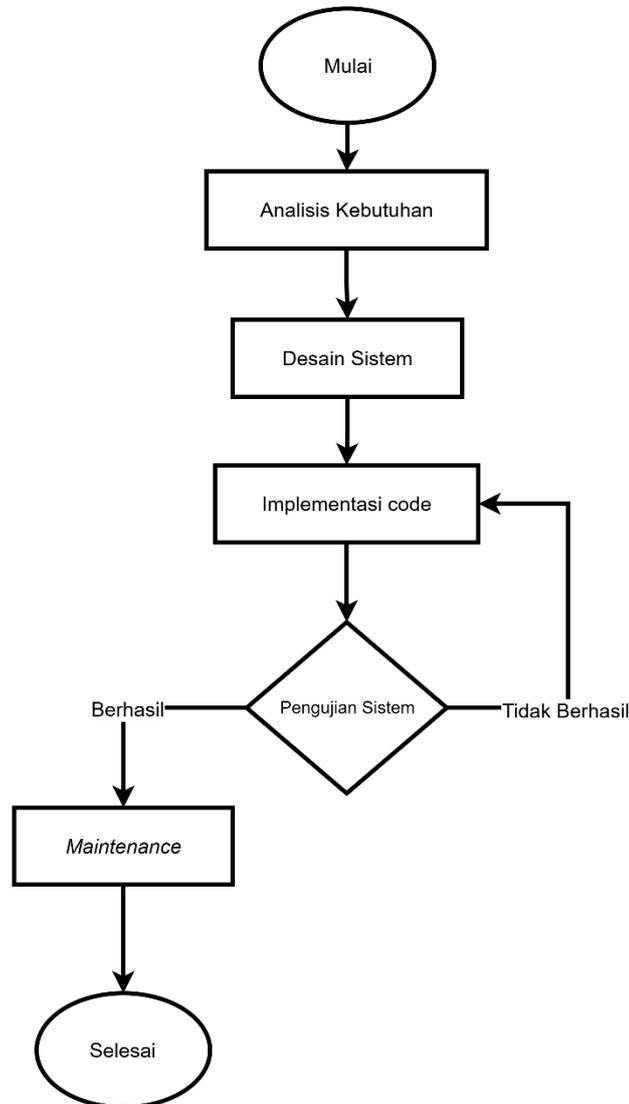
#### **3.1 Jenis Penelitian**

Jenis penelitian yang diterapkan dalam penelitian ini adalah *Research and Development* (R&D). Menurut Borg and Gall (1983), pendekatan R&D bertujuan untuk menghasilkan serta memverifikasi keefektifan suatu produk melalui tahapan yang terstruktur (Karmarkar dkk., 2024). Proses tersebut mencakup kajian pustaka, identifikasi kebutuhan, perancangan awal, pengembangan prototipe, uji coba terbatas, perbaikan, hingga tahap implementasi akhir (Gustiani, 2019)

Dalam penelitian ini, metode R&D digunakan untuk merancang dan membangun aplikasi pengelolaan stok barang yang terintegrasi dengan *Barcode Scanner* (Relvas & Ramos, 2021). Aplikasi ini dikembangkan guna mendukung efisiensi dan akurasi dalam manajemen gudang di toko Havimahesa. Dengan melalui serangkaian tahap pengujian dan penyempurnaan, produk akhir diharapkan mampu memenuhi kebutuhan operasional toko secara optimal.

#### **3.2 Alur Penelitian**

Penelitian ini dilaksanakan melalui rangkaian tahapan yang terstruktur untuk mencapai tujuan utama yang telah ditetapkan. Tahapan penulisan ini tersusun pada gambar 3.1. Diagram tersebut menggambarkan alur berurutan yang dimulai dari identifikasi kebutuhan hingga evaluasi akhir sistem. Setelah kebutuhan dianalisis, proses berlanjut ke tahap desain untuk merumuskan struktur dan aplikasi antarmuka, lalu ke implementasi kode sebagai realisasi rancangan. Hasil implementasi diuji secara menyeluruh apabila pengujian belum memenuhi kriteria, dilakukan *maintenance* dan penyesuaian hingga fungsionalitas tercapai. Ketika pengujian dinyatakan berhasil, kegiatan penelitian ditutup pada tahap selesai, menandai terpenuhinya tujuan yang telah dirumuskan di awal.



Gambar 3. 1 Alur Penelitian

### 3.2.1 Analisis Kebutuhan

Tahapan ini bertujuan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional dari sistem yang akan dikembangkan. Kegiatan dalam tahap ini meliputi observasi lapangan, wawancara dengan pemilik toko Havimahesa, serta pengumpulan data terkait proses pengelolaan stok barang yang sedang berjalan. Hasil dari tahap ini menjadi dasar dalam merancang fitur-fitur utama aplikasi.

### 3.2.2 Desain Sistem

Pada tahap ini dilakukan perancangan struktur sistem berdasarkan hasil analisis kebutuhan. Desain mencakup arsitektur sistem, perancangan aplikasi antarmuka pengguna (*user interface*), serta model alur data. Tujuan dari tahap ini adalah untuk memastikan sistem memiliki rancangan yang terstruktur, mudah dipahami, dan siap untuk diimplementasikan.

### 3.2.3 Implementasi Kode

Tahap selanjutnya adalah implementasi atau pengkodean sistem. Pengembangan aplikasi dilakukan menggunakan Microsoft Excel sebagai *platform* utama, dengan *Visual Basic for Applications* (VBA) sebagai bahasa pemrograman. Pada tahap ini, prinsip *Object-Oriented Programming* (OOP) diterapkan untuk merancang modul-modul sistem seperti pengelolaan data stok barang, pemindaian *Barcode*, dan pencatatan transaksi, yang semuanya akan dikelola menggunakan VBA untuk interaksi dengan data dan aplikasi antarmuka pengguna.

### 3.2.4 Pengujian Sistem

Tahap pengujian dilakukan untuk memastikan bahwa sistem berjalan sesuai dengan kebutuhan dan bebas dari kesalahan. Pengujian dilakukan dengan metode *black-box testing* untuk melihat apakah setiap fitur berfungsi sebagaimana mestinya dan *System Usability Scale* (SUS) untuk mengukur kemudahan aplikasi.

### 3.2.5 Maintenance

Setelah sistem dinyatakan berjalan dengan baik, tahap akhir adalah pemeliharaan. Kegiatan ini mencakup perbaikan bug, pembaruan fitur, serta penyesuaian terhadap kebutuhan operasional yang mungkin berubah di masa mendatang. Tahap ini penting untuk menjaga agar sistem tetap optimal dan relevan dalam mendukung aktivitas gudang toko Havimahesa.

## 3.3 Perancangan Sistem

Pada tahap ini, disajikan apa saja kebutuhan-kebutuhan yang akan digunakan dalam penelitian seperti kebutuhan *Hardware*

### 3.3.1 Lingkungan Pengembangan

#### 3.3.1.1 Perangkat Keras

Perangkat Keras atau *Hardware* yang digunakan pada penelitian ini adalah sebagai berikut.

Tabel 3.1 *Hardware* yang digunakan

No	Komponen	Spesifikasi
1	<i>Processor</i>	AMD Ryzen 7 4800H
2	RAM	16 GB
3	SSD	1T
4	<i>Graphic Card</i>	NVIDIA GeForce GTX 1650
5	<i>Scanner</i>	Ultron 1808 1D USB

#### 3.3.1.2 Perangkat Lunak

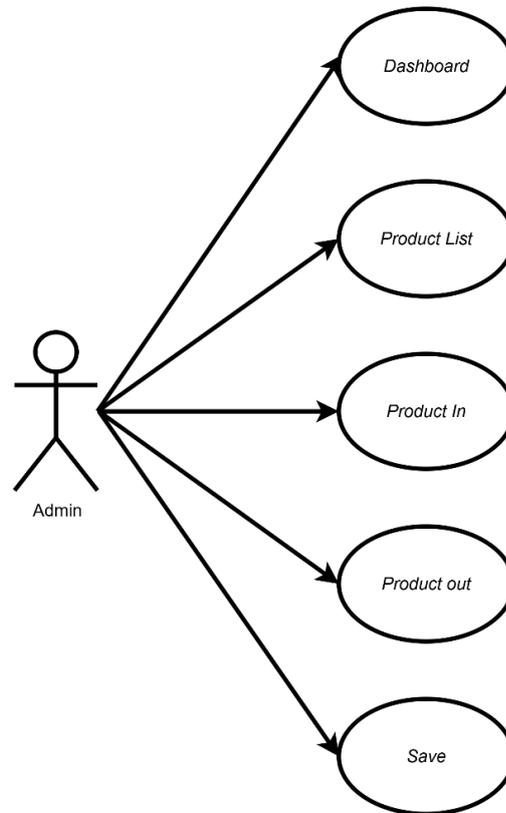
Perangkat lunak atau *Software* yang digunakan pada penelitian ini adalah sebagai berikut.

Tabel 3.2 *Software* yang digunakan

No	<i>Software</i>	Kegunaan
1	<i>Windows 11</i>	Sistem Operasi
2	<i>Microsoft Excel</i>	Menjalankan Aplikasi

#### 3.3.1.3 Desain Kebutuhan Sistem

Desain kebutuhan sistem bertujuan untuk menggambarkan bagaimana sistem yang akan dibangun akan memenuhi kebutuhan pengguna. Pada bagian ini, penulis akan menggunakan *Use Case Diagram* untuk memvisualisasikan interaksi antara aktor dan sistem. *Use Case Diagram* ini menggambarkan fungsi-fungsi utama yang akan disediakan oleh aplikasi pengelolaan stok barang untuk pengguna yang berperan sebagai Admin yang divisualisasikan di gambar 3.2.



Gambar 3. 2 Use Case Diagram

Admin dapat mengakses beberapa fitur utama dalam sistem, seperti *Dashboard*, yang berfungsi sebagai tampilan utama aplikasi, memberikan gambaran umum tentang informasi stok barang dan aktivitas yang sedang berlangsung. Selain itu, Admin juga dapat melihat daftar produk yang ada dalam sistem melalui fitur *Product List*, yang memungkinkan pengelolaan dan pengecekan produk yang tersedia.

Admin memiliki kemampuan untuk mencatat barang yang masuk ke dalam sistem menggunakan fitur *Product In*, yang mencakup pengisian informasi barang baru yang diterima. Salah satu metode yang digunakan untuk menambah produk adalah dengan memindai *Barcode*, yang akan mempercepat dan mempermudah proses pencatatan produk ke dalam sistem. Fitur *Product Return* memungkinkan Admin untuk mencatat barang yang dikembalikan atau tidak terjual. Semua perubahan yang dilakukan pada data produk atau stok harus disimpan, yang

dilakukan melalui fitur *Save*. Fitur ini memastikan bahwa setiap pembaruan yang dilakukan oleh Admin pada sistem selalu tercatat dan tersimpan dengan akurat.

### **3.4 Teknik Pengumpulan Data**

Pada bagian ini, dijelaskan teknik-teknik yang digunakan untuk mengumpulkan data yang dibutuhkan dalam penelitian ini. Pengumpulan data merupakan langkah krusial dalam penelitian, karena data yang dikumpulkan akan menjadi dasar untuk menganalisis dan menarik kesimpulan. Dalam penelitian ini, terdapat beberapa teknik pengumpulan data yang digunakan, yaitu observasi, studi literatur, implementasi, dan pengujian. Berikut adalah penjelasan rinci mengenai masing-masing teknik pengumpulan data yang diterapkan dalam penelitian ini.

#### **3.4.1 Observasi**

Observasi merupakan teknik pengumpulan data yang dilakukan dengan cara mengamati langsung interaksi pengguna dengan aplikasi pengelolaan stok barang yang dikembangkan. Observasi ini bertujuan untuk mengidentifikasi bagaimana aplikasi digunakan dalam situasi nyata, mencatat hambatan yang dihadapi pengguna, serta mencatat setiap kekurangan atau kelebihan yang muncul saat aplikasi digunakan di lapangan. Dalam penelitian ini, observasi dilakukan di lingkungan toko yang menggunakan aplikasi untuk mengelola stok barang. Pengamatan dilakukan terhadap pengguna yang berinteraksi dengan sistem, serta terhadap proses transaksi barang masuk dan keluar yang dilakukan melalui aplikasi.

#### **3.4.2 Studi Literatur**

Studi literatur adalah teknik pengumpulan data yang dilakukan dengan cara mencari, membaca, dan menganalisis berbagai referensi yang relevan dengan topik penelitian. Referensi tersebut dapat berupa buku, jurnal, artikel, atau publikasi ilmiah lainnya yang membahas teori-teori yang berkaitan dengan pengelolaan stok barang, pengujian perangkat lunak, serta teknik-teknik yang digunakan dalam penelitian seperti *Black Box Testing* dan *System Usability Scale (SUS)*. Studi literatur ini bertujuan untuk memberikan dasar teoritis bagi penelitian ini, serta

membantu dalam merancang implementasi dan pengujian yang sesuai dengan standar yang telah ada.

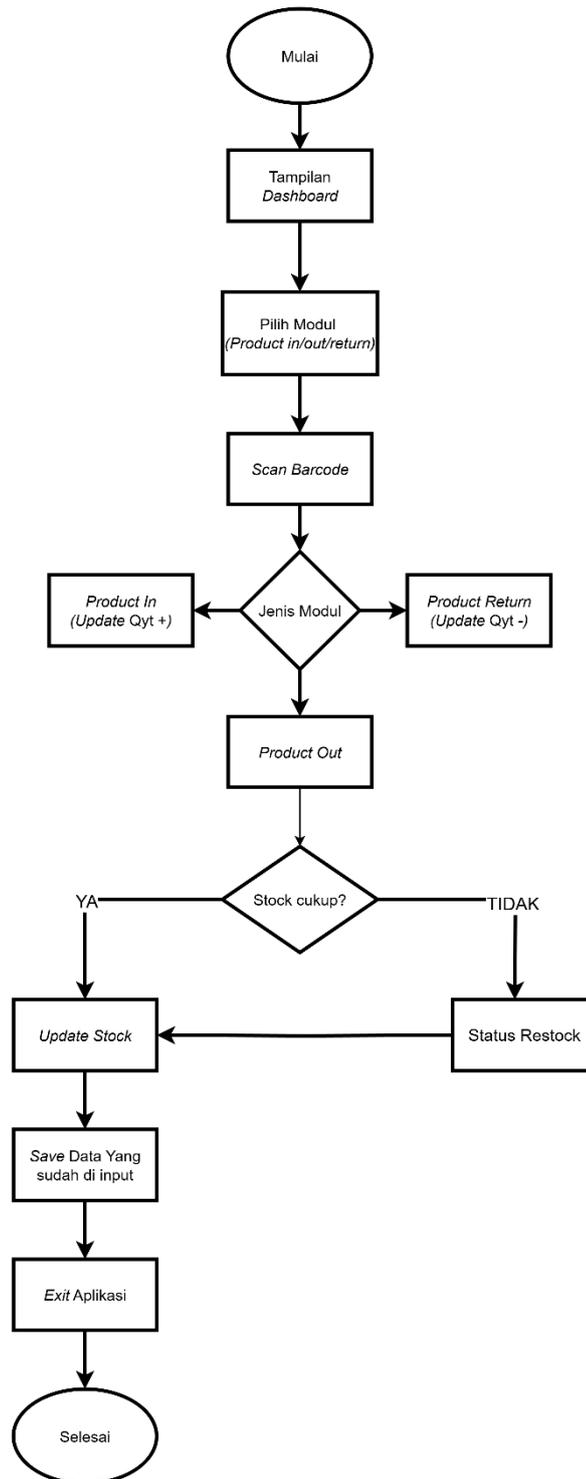
### 3.4.3 Implementasi

Pada tahap implementasi ini, seluruh hasil perancangan sistem pengelolaan stok barang terintegrasi dengan *Barcode Scanner* mulai diterapkan ke dalam aplikasi. Pengembangan aplikasi dilakukan dengan menggunakan Excel sebagai *platform* utama dan VBA (*Visual Basic for Applications*) sebagai bahasa pemrogramannya, yang mengimplementasikan prinsip *Object-Oriented Programming* (OOP) untuk pengelolaan kode yang lebih terstruktur dan efisien. Fitur utama yang dikembangkan mencakup pemindaian *Barcode* untuk pencatatan barang, pengelolaan data stok, dan pencatatan transaksi barang masuk dan keluar. Setiap komponen sistem ini diintegrasikan dengan tujuan untuk meningkatkan efisiensi pengelolaan stok barang di gudang toko Havimahesa. Tahap ini bertujuan untuk memastikan seluruh fitur dapat berjalan secara optimal dan siap untuk diuji pada tahap pengujian lebih lanjut.

### 3.4.4 Pengujian

Pada tahap pengujian, dilakukan untuk memastikan bahwa aplikasi yang telah dikembangkan benar-benar sesuai dengan kebutuhan pengguna dan berfungsi dengan baik. Pada Gambar 3.3, alur aplikasi mulai dari *Dashboard*, pemilihan modul *Product In*, *Product Out*, dan *Product Return*, proses pemindaian *barcode*, hingga pembaruan stok serta penetapan status *reorder* ditelusuri melalui empat lapis uji. Pertama, *unit testing* memverifikasi setiap fungsi inti secara terpisah, termasuk pemanggilan kamera dan metode *updateQtyPlus/updateQtyMinus* pada berbagai skenario valid maupun error. Kedua, *integration testing* memastikan *SKU* hasil *scan* otomatis mengalir ke prosedur pengecekan stok dan, bila jumlahnya di bawah ambang minimum, memicu logika *reorder* tanpa intervensi manual. Setelah integrasi dinyatakan stabil, *system testing* menjalankan skenario *end-to-end* guna menjamin percabangan “Stock cukup?” dan penanganan pesan kesalahan bekerja mulus. Terakhir, *user acceptance testing* melibatkan admin gudang untuk menilai

kecepatan pemindaian, kejelasan notifikasi, serta kemudahan keluar aplikasi setelah data tersimpan. Hasil tiap tahap dicatat melalui metrik keberhasilan skenario, waktu eksekusi, dan umpan balik kegunaan, sehingga keseluruhan pengujian tidak hanya mengonfirmasi kesesuaian spesifikasi teknis, tetapi juga reliabilitas dan efisiensi proses bisnis yang diilustrasikan pada diagram tersebut.



Gambar 3. 3 Alur Aplikasi

### 3.4.1.1 *Black-Box Testing*

*Black Box Testing* merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem dengan memeriksa *input* yang diberikan dan *output* yang dihasilkan, tanpa perlu memahami struktur atau kode internal sistem tersebut. Pengujian ini dilakukan dengan tujuan untuk memastikan bahwa aplikasi berfungsi sesuai dengan persyaratan yang telah ditentukan, dengan mengandalkan aplikasi antarmuka atau tampilan sistem yang terlihat oleh pengguna (Mintarsih, 2023).

Dalam pengujian ini, penguji tidak perlu mengetahui bagaimana kode program bekerja, melainkan hanya perlu memperhatikan bagaimana sistem merespons *input* yang diberikan. Proses pengujian dimulai dengan pembuatan test case untuk menguji fungsi-fungsi utama dalam aplikasi, serta memastikan bahwa alur kerja fungsi tersebut sesuai dengan harapan pengguna. Selain itu, pengujian juga meliputi pencarian *bugs* atau kesalahan pada tampilan aplikasi antarmuka aplikasi, dengan harapan dapat menemukan masalah yang mungkin tidak terdeteksi pada tahap pengembangan (Huda dkk., 2022).

### 3.4.1.2 System Usability Case (SUS)

*System Usability Scale* (SUS) adalah metode pengujian yang digunakan untuk mengukur seberapa mudah dan efektif suatu sistem atau aplikasi digunakan oleh penggunanya. SUS adalah alat yang sering digunakan dalam pengujian kegunaan (*usability testing*) untuk mengevaluasi kualitas pengalaman pengguna dalam menggunakan suatu sistem. Seperti pada tabel SUS terdiri dari 10 pertanyaan yang dinilai dari skala Likert, yang berjumlah 5 nilai seperti *Strongly Agree* hingga *Strongly Disagree* (Zaibintoro dkk., 2024).

Tabel 3.3 Pertanyaan System Usability Scale

Kode	Pertanyaan
P1	“Saya berpikir akan menggunakan sistem ini lagi.”
P2	“Saya merasa sistem ini rumit untuk digunakan.”

Kode	Pertanyaan
P3	“Saya merasa sistem ini mudah digunakan.”
P4	“Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.”
P5	“Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.”
P6	“Saya merasa ada banyak hal yang tidak konsisten pada sistem ini.”
P7	“Saya merasa sistem ini membingungkan.”
P8	“Saya merasa tidak ada hambatan dalam menggunakan sistem ini.”
P9	“Saya merasa sangat percaya diri dalam menggunakan sistem ini.”
P10	“Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.”

### 3.5 Teknik Analisis Data

Pada bagian ini, dijelaskan teknik-teknik analisis data yang digunakan untuk mengevaluasi hasil pengujian yang telah dilakukan terhadap aplikasi pengelolaan stok barang. Teknik analisis data ini bertujuan untuk mendapatkan pemahaman yang lebih mendalam mengenai kinerja aplikasi serta persepsi pengguna terhadap kemudahan dan efektivitas aplikasi tersebut. Dalam penelitian ini, dua teknik analisis utama yang digunakan adalah Analisis *Black-box Testing* dan Analisis *Usability*. Berikut adalah penjelasan lebih rinci mengenai masing-masing teknik analisis yang diterapkan.

#### 3.5.1 Analisis *Black-box Testing*

Analisis *Black-box Testing* dilakukan untuk mengevaluasi apakah aplikasi pengelolaan stok barang dapat berfungsi sesuai dengan yang diharapkan tanpa memperhatikan kode internal aplikasi. Fokus utama dari analisis ini adalah pada *input* dan *output* yang dihasilkan oleh sistem. Pada pengujian ini, seluruh fungsi aplikasi diuji berdasarkan skenario pengujian yang mencakup berbagai jenis fitur aplikasi untuk melihat apakah sistem menghasilkan *output* yang sesuai.

Proses analisis dilakukan dengan memverifikasi setiap skenario pengujian yang telah disiapkan sebelumnya untuk memastikan bahwa aplikasi dapat menangani semua kemungkinan fitur aplikasi dengan benar. Pengujian ini mencakup berbagai fitur aplikasi, seperti pencatatan transaksi barang, pemindaian *Barcode*, pembaruan stok, dan pembuatan laporan. Hasil dari analisis ini akan menunjukkan sejauh mana aplikasi memenuhi spesifikasi fungsional yang telah ditentukan dan membantu mengidentifikasi masalah atau kesalahan yang mungkin ada dalam sistem. Secara keseluruhan, analisis *Black-box Testing* bertujuan untuk menilai ketepatan aplikasi dalam menghasilkan *output* yang diinginkan berdasarkan *input* yang diberikan oleh pengguna. Dari hasil penelitian akan dihitung persentasenya dengan rumus berikut (Rahmawati & Susilowibowo, 2020).

$$\text{Presentase Kelayakan} = \frac{\text{Jumlah skor yang sesuai}}{\text{Total skenario}} \times 100\% \quad (3.1)$$

Setelah didapatkan presentase dari rumus 3.1, nilai kelayakannya dapat dikategorikan dengan tabel kriteria kelayakan di tabel.

Tabel 3.4 Kriteria Kelayakan

Penilaian	Kriteria Kelayakan
0% - 20%	Sangat tidak layak
21% - 40%	Tidak layak
41% - 60%	Cukup layak
61% - 80%	Layak
81% - 100%	Sangat layak

### 3.5.2 Analisis Usability

Dalam pengujian *Usability*, salah satu alat yang digunakan untuk mengukur kualitas pengalaman pengguna adalah *System Usability Scale* (SUS). SUS menggunakan skala Likert, yang terdiri dari sepuluh pernyataan yang mencakup

berbagai aspek pengalaman pengguna dengan aplikasi atau sistem yang diuji. Pengguna diminta untuk memberikan penilaian berdasarkan skala Likert, yang mengukur sejauh mana mereka sangat setuju hingga sangat tidak setuju dengan setiap pernyataan yang diberikan.

Skala Likert digunakan untuk mengumpulkan secara objektif mengenai persepsi pengguna terhadap berbagai aspek dari aplikasi, seperti kemudahan penggunaan, kesenangan dalam menggunakan aplikasi, serta potensi kesulitan atau kebingungannya. Selain itu peneliti dapat memperoleh skor yang menunjukkan sejauh mana pengguna merasa aplikasi tersebut mudah digunakan, serta memberikan gambaran umum tentang tingkat kepuasan terhadap aplikasi tersebut mereka (Najib dkk., 2023).

Metode SUS menilai kecenderungan responden terhadap setiap pernyataan menggunakan skala Likert lima poin. Responden menyatakan tingkat persetujuan atau ketidaksetujuan mereka, dan rentang nilainya tercantum pada Tabel 3.5.

Tabel 3. 5 Skor Skala Likert

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

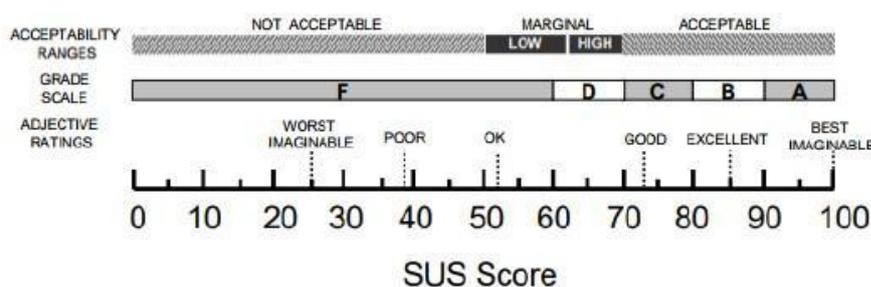
Untuk mendapatkan skor rata-rata SUS, jawaban dari responden perlu diolah terlebih dahulu menggunakan perhitungan pada persamaan

$$\text{SUS Skor (Individu)} : \{(P_1-1) + (5- P_2) + (P_3-1) + (5- P_4) + (P_4-1) + (5- P_6) + (P_7-1) + (5- P_8) + (P_9-1) + (5- P_{10})\} *2.5 \quad (3.2)$$

Setelah didapatkan total skor dari jawaban setiap responden, selanjutnya dilakukan perhitungan hasil akhir rata-rata untuk hasil SUS yang dapat dilihat pada persamaan berikut.

$$\begin{array}{l} \text{Skor Rata-} \\ \text{Rata} \end{array} : \frac{\text{Total Skor Individu}}{\text{Jumlah Responden}} \quad (3.3)$$

Setelah mendapatkan skor rata-rata akhir, nilai tersebut kemudian disesuaikan dengan rentang penilaian yang akan menentukan hasil dari evaluasi SUS. Rentang penilaian yang digunakan sebagai acuan untuk penilaian SUS adalah sebagai berikut.



Gambar 3.1 Penilaian Pengujian SUS

Berdasarkan Gambar 3.4, terdapat tiga jenis penilaian dalam metode SUS, yaitu *Acceptability Range*, *Grade Scale*, dan *Adjective Ratings*. Ketiga indikator ini memungkinkan penilaian untuk diinterpretasikan lebih mendalam, mencakup kategori penerimaan, tingkat kepuasan pengguna, serta kecenderungan mereka untuk merekomendasikan aplikasi kepada orang lain.