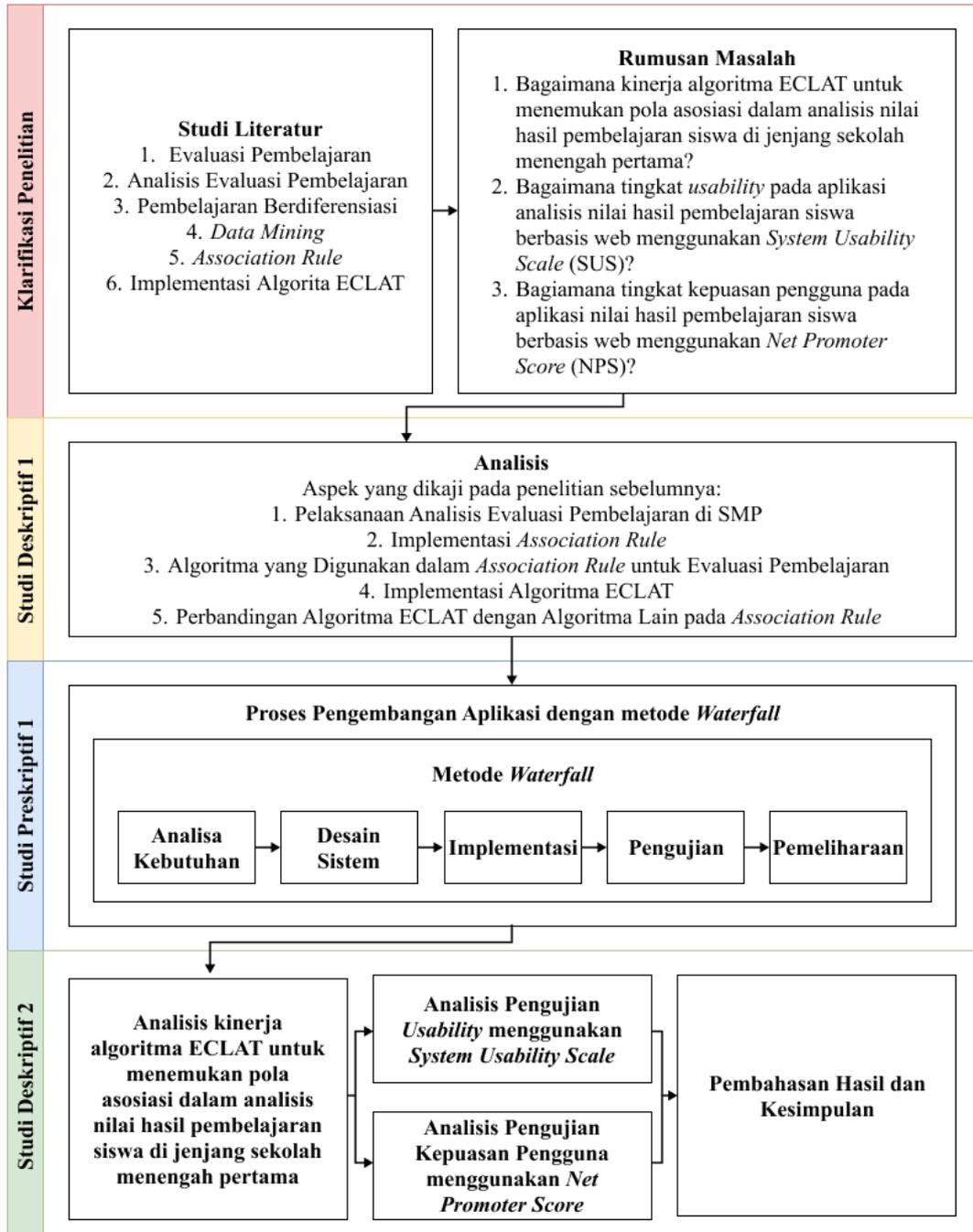


## BAB III METODOLOGI PENELITIAN

### 3.1 Desain Penelitian

Penelitian ini menggunakan metode *Design Research Methodology* (DRM) yang dapat dilihat pada gambar 3.1. DRM berhasil diterapkan untuk melakukan pengembangan aplikasi dengan kerangka kerja terstruktur (Falih dkk., 2021).



Gambar 3.1 Desain Penelitian

### 3.1.1 Klarifikasi Penelitian (*Research Clarification*)

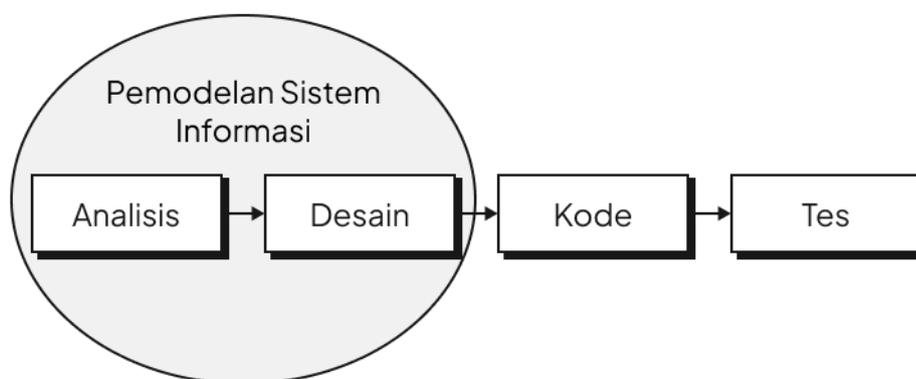
Pada tahap awal klarifikasi penelitian di DRM bertujuan untuk menentukan tujuan atau *goals* dari penelitian yang akan dilakukan. Di tahap ini dilakukan pencarian permasalahan dengan melakukan studi literatur sebagai justifikasi terhadap tujuan dari penelitian ini. Pada tahap ini, akan dilakukan studi literatur terkait evaluasi pembelajaran, analisis evaluasi pembelajaran, dan pembelajaran berdiferensiasi untuk mengetahui permasalahan yang ada dan solusi yang dapat diberikan untuk mengatasi permasalahan yang ditemukan. Selain itu, studi terhadap *data mining*, *association rule*, dan implementasi algoritma ECLAT dilakukan untuk memahami kajian terbaru yang dapat diterapkan sebagai solusi dari permasalahan yang ditemukan. Pada tahap ini dilakukan juga wawancara dengan beberapa guru sekolah menengah pertama (SMP) di Bandung Timur untuk memperkuat justifikasi masalah. Hasil studi literatur ini akan digunakan untuk merumuskan masalah dan tujuan penelitian.

### 3.1.2 Studi Deskriptif 1 (*Descriptive Study*)

Tahap studi deskriptif 1 dalam metode DRM bertujuan untuk memperdalam kajian ilmiah terkait dengan permasalahan yang diangkat pada penelitian. Pada tahap ini dilakukan analisis terhadap data empirik, seperti yang terdapat pada penelitian ilmiah sebelumnya. Penelitian ilmiah sebelumnya akan dijadikan sebagai acuan atau bahan referensi untuk mengembangkan penelitian ini. Aspek yang akan dikaji lebih dalam pada penelitian ini, yaitu pelaksanaan analisis evaluasi pembelajaran di jenjang SMP, implementasi *association rule*, algoritma yang digunakan dalam *association rule* untuk evaluasi pembelajaran, implementasi algoritma ECLAT, dan perbandingan algoritma ECLAT dengan algoritma lain pada *association rule*. Penelitian-penelitian sebelumnya yang membahas aspek tersebut akan ditelaah dan dibandingkan satu sama lain terkait implementasi, alat atau media yang digunakan, metode yang digunakan, dan hasil pengujian yang dapat digunakan untuk mengidentifikasi potensi-potensi perbaikan sehingga dapat meningkatkan efektivitas atau efisiensi dari hasil penelitian yang akan dilakukan. Oleh karena itu, studi deskriptif ini bertujuan untuk memperluas dan memperdalam pemahaman tentang konteks yang relevan dengan isu-isu tersebut, serta menyusun dasar pengetahuan yang kokoh untuk pengembangan penelitian.

### 3.1.3 Studi Preskriptif (*Prescriptive Study*)

Pada studi preskriptif ini, akan dilakukan pengembangan aplikasi untuk menguji solusi dari permasalahan yang telah ditentukan pada tahap sebelumnya. Tahap awal dimulai dengan melakukan analisis kebutuhan untuk memahami kebutuhan fungsional dan non-fungsional. Hasil analisis kebutuhan tersebut kemudian dirancang untuk mengembangkan aplikasi sesuai dengan kebutuhan. Setelah itu, rancangan tersebut dikembangkan dan diimplementasikan sebagai solusi dari permasalahan dan tujuan yang ditentukan. Aplikasi yang telah berhasil dikembangkan kemudian diuji untuk memastikan aplikasi fungsional dan memenuhi kebutuhan yang telah dirancang. Pengembangan aplikasi ini menggunakan metode *waterfall* untuk memudahkan proses pengembangan yang lebih terstruktur dan terencana. Mengutip dari Nurseptaji dkk. (2021), metode *waterfall* merupakan sebuah metode klasik yang memiliki sifat sistematis secara terurut dalam membangun perangkat lunak. Metode ini menyediakan kerangka kerja terstruktur dengan beberapa tahap yang dimulai pada dengan perencanaan tahap analisis hingga implementasi, menekankan dokumentasi yang kuat untuk memastikan pemahaman terhadap kebutuhan sistem dan hasil akhir yang sesuai harapan (Wardanu dan Aryanto, 2023). Pada metode ini terdapat 4 tahap utama dalam proses pengembangan aplikasi, yaitu analisis, desain, implementasi, dan pengujian. Pada gambar 3.2 menggambarkan keseluruhan dari tahapan yang ada di metode *waterfall* dimulai dari tahap analisis hingga tahap pengujian atau testing.



Gambar 3.2 Metode *Waterfall*

## 1. Analisis

Analisis merupakan tahap pertama dalam metode waterfall. Pada tahap ini analisis yang dilakukan bertujuan untuk mengetahui kebutuhan dalam perancangan dan pengembangan aplikasi. Analisis dilakukan dengan intensif untuk menentukan spesifikasi kebutuhan aplikasi yang dibutuhkan oleh pengguna (Rosa dan Shalahuddin, 2018). Terdapat beberapa kebutuhan yang harus dipenuhi sebagai spesifikasi bagi aplikasi atau perangkat lunak yang akan dikembangkan, seperti kebutuhan fungsional dan non-fungsional. Pengumpulan data dengan wawancara dapat diterapkan untuk menentukan kebutuhan fungsional dan non-fungsional aplikasi (Zikri, 2021). Dalam menentukan kebutuhan aplikasi yang dikembangkan, penelitian ini menggunakan data hasil wawancara yang telah dilakukan di SMP Laboratorium Percontohan UPI Cibiru. Selain itu, hasil kajian pustaka sebelumnya juga dijadikan sebagai bahan untuk menentukan kebutuhan aplikasi.

## 2. Desain

Desain atau perancangan merupakan salah satu tahap penting dalam metode *waterfall* yang berguna dalam proses konstruksi sistem untuk memenuhi spesifikasi yang dibutuhkan (Rosa dan Shalahuddin, 2018). Pada tahap desain ini, hasil dari analisis kebutuhan yang telah dilakukan sebelumnya dijadikan acuan dalam merancang aplikasi. Spesifikasi yang telah ditentukan untuk mengembangkan aplikasi mulai dirancang sebagai solusi dalam menjawab permasalahan penelitian. Proses desain atau perancangan ini terdiri dari beberapa proses perancangan, yaitu merancang diagram *use case*, *flowchart*, dan desain antarmuka pengguna. Proses perancangan ini menggunakan beberapa *tools* yang berguna untuk mempermudah dalam membuat rancangan, seperti Draw.io untuk merancang diagram dan Figma untuk merancang desain antarmuka pengguna. Pada masing-masing proses di tahap ini menghasilkan sebuah luaran berupa rancangan atau desain berdasarkan spesifikasi aplikasi.

### a. *Use Case Diagram*

*Use Case Diagram* adalah suatu teknik pemodelan terhadap kelakuan atau *behavior* dari aplikasi yang dikembangkan (Rosa dan Shalahuddin, 2018). Rancangan yang dihasilkan dari *use case diagram* berupa deskripsi dari masing-masing fungsi yang terjadi dalam interaksi antara aktor dengan sistem atau aplikasi

yang dikembangkan. Perancangan *use case diagram* dilakukan menyesuaikan dengan kebutuhan yang telah ditentukan pada tahap sebelumnya. Kebutuhan tersebut kemudian dirancang untuk mengetahui fungsi apa saja yang dimiliki oleh aktor ketika menggunakan aplikasi. Hasil dari perancangan *use case diagram* ini juga digunakan sebagai dasar dalam perancangan *flowchart* untuk menggambarkan lebih jelas alur yang dapat dilakukan oleh aktor ketika menggunakan fungsi-fungsi yang ada.

#### b. *Flowchart Diagram*

Diagram *flowchart* digunakan untuk menggambarkan alur kerja dari aplikasi yang dikembangkan. Hasil dari perancangan diagram *flowchart* digunakan sebagai dasar dalam mengembangkan alur dari setiap proses aplikasi sehingga pengembangan aplikasi dapat dilakukan dengan terstruktur. Perancangan *flowchart* dilakukan dengan menentukan seperti alur yang terjadi ketika pengguna menggunakan aplikasi. Alur yang dirancang dalam *flowchart* terdiri dari beberapa proses, seperti pendaftaran akun, masuk akun, halaman dan proses analisis, serta halaman untuk melihat hasil analisis.

#### c. *Activity Diagram*

*Activity diagram* atau diagram aktivitas merupakan diagram yang menggambarkan aktivitas yang terdapat dari sebuah sistem atau proses bisnis (Rosa dan Shalahuddin, 2018). Alur yang digambarkan pada diagram ini merupakan alur dari aktivitas yang dapat dilakukan oleh sistem dan bukan alur yang dilakukan oleh aktor.

#### d. *User Interface/Antarmuka Pengguna*

Hasil dari *user interface* digunakan untuk menggambarkan tampilan dari aplikasi. Proses perancangan antar muka pengguna menggunakan aplikasi Figma dilakukan dengan merancang terlebih dahulu *wireframe* (Hannah dkk., 2024). Perancangan *wireframe* dilakukan berdasarkan *flowchart* aplikasi untuk merancang beberapa halaman, seperti halaman autentikasi akun, halaman utama analisis, halaman riwayat, serta halaman kelas. Setelah *wireframe* berhasil dirancang, hasil akhir dari perancangan antar muka pengguna adalah *high fidelity prototype*

(Rusanty dkk., 2019). Antarmuka aplikasi yang dihasilkan merupakan halaman dari setiap fitur utama yang terdapat di dalam aplikasi analisis nilai hasil pembelajaran.

### 3. Kode/Implementasi

Setelah semua kebutuhan ditentukan dan rancangan aplikasi berhasil dirancang, tahap selanjutnya untuk mengembangkan aplikasi dalam metode *waterfall* ini adalah kode atau implementasi. Pada tahap implementasi, semua rancangan aplikasi yang telah dibuat sebelumnya diimplementasikan. Proses implementasi dalam membangun aplikasi berbasis web ini menggunakan *source code editor* Visual Studio Code. Bahasa pemrograman yang digunakan dalam mengembangkan model adalah Python. Implementasi python pada pengembangan aplikasi berbasis web ini diintegrasikan menggunakan *application programming interface* (API) FastAPI. FastAPI merupakan API yang dirancang untuk implementasi *web service* (Azhari, 2022). FastAPI ini digunakan untuk mengembangkan API dengan versi minimum Python 3.6 (Voron, 2023). Selain itu, pada implementasi python, digunakan juga beberapa *library* pendukung untuk mendukung proses pengembangan model, seperti PyEclat.

### 4. Pengujian/*Testing*

Tahap akhir pada metode ini adalah melakukan pengujian aplikasi yang telah diimplementasi atau dikembangkan. Pengujian dilakukan menggunakan metode pengujian *black box*. Pada penelitian Mukrodin dan Sugianto (2020), menemukan bahwa metode *black box* efektif untuk menguji hasil pengembangan aplikasi dengan implementasi metode pengembangan *waterfall*. Implementasi pengujian menggunakan *black box* dilakukan dengan menentukan terlebih dahulu *use case scenario* yang ingin diuji. Masing-masing *use case* diuji sesuai dengan langkah-langkah dan hasil yang diharapkan. Pengujian *black box* difokuskan pada fitur utama yang terdapat pada aplikasi, seperti analisis nilai hasil belajar, riwayat analisis, menambahkan dan melihat kelas, serta autentikasi akun.

#### 3.1.4 Studi Deskriptif 2 (*Descriptive Study*)

Pada tahap studi deskriptif akhir memiliki tujuan untuk melakukan evaluasi dan analisis terhadap rumusan masalah berdasarkan aplikasi yang telah dikembangkan. Evaluasi yang dilakukan adalah dengan melakukan pengujian

terhadap model dan aplikasi untuk mengetahui hasil implementasi algoritma ECLAT. Hasil implementasi dari algoritma ECLAT akan digunakan dalam pengujian terhadap pengembangan model. Pengujian model dilakukan dengan menggunakan metrik *lift ratio* untuk mengukur kinerja dari algoritma yang diterapkan. Nilai yang diperoleh dari pengujian tersebut kemudian dianalisis untuk mengetahui apakah model telah berjalan dengan menghasilkan *output* sesuai harapan. Selain itu, evaluasi juga dilakukan untuk mengetahui tingkat *usability* dari aplikasi menggunakan *system usability scale* dan tingkat kepuasan pengguna menggunakan *net promoter score*. Hasil dari ketiga evaluasi tersebut akan dianalisis dan dibahas untuk mendapatkan simpulan, rekomendasi, dan saran yang komprehensif.

### 3.2 Populasi dan Sampel

Populasi yang digunakan pada penelitian ini adalah guru di SMP Laboratorium Percontohan UPI di Cibiru. Populasi guru pada sekolah tersebut sebanyak 30 guru. Penentuan sampel pada penelitian ini menggunakan metode *simple random sampling* (SRS). Metode SRS digunakan untuk penentuan sampel dalam penelitian kuantitatif, memiliki model perhitungan dengan semua individu pada populasi memiliki kesempatan yang sama untuk berpartisipasi (Noor dkk., 2022). Penggunaan metode SRC ini cocok digunakan dalam menentukan sampel dengan populasi yang relatif homogen dan menghasilkan representasi yang tidak bias dari populasi (Lakhre dan Mishra, 2024). Berikut merupakan formula untuk menentukan sampel menggunakan formula 3.1.

$$n = \frac{N}{1 + N \cdot e^2}$$

Formula 3.1 Perhitungan sampel menggunakan *simple random sampling*

Keterangan:

n = Ukuran sampel

N = Jumlah populasi

e = *Margin of error*

$$n = \frac{30}{1 + 30 \cdot (0,1)^2} = \frac{30}{1,30} \approx 23$$

Berdasarkan hasil perhitungan dengan menggunakan formula 3.1 dengan *margin of error* 10% atau 0.1 dan dengan jumlah populasi yang digunakan pada penelitian ini sebanyak 30 guru, maka sampel penelitian yang akan digunakan adalah 23 guru.

### 3.3 Instrumen Penelitian

#### 3.3.1 *Lift Ratio*

Model yang telah dikembangkan kemudian diuji menggunakan nilai *lift*. Pembagian *dataset* dilakukan dengan membagi data menjadi data latih dan data uji (Li dan Rosas, 2023). Rasio data latih dan data uji sebesar 80% dan 20% (Dhawan, 2023). Berdasarkan 904 *dataset*, 723 data digunakan sebagai data latih dan 181 data digunakan sebagai data uji. Pembagian *Lift ratio* merupakan metrik yang digunakan untuk memvalidasi hasil dari *association rule*. Penelitian yang dilakukan oleh Hassan dkk. (2023) menggunakan metrik ini dengan membagi *confidence* dengan *expected confidence*.

$$\text{Expected Confidence} = \frac{\text{Jumlah transaksi yang mengandung } Y}{\text{Jumlah transaksi}}$$

Formula 3.2 Perhitungan *expected confidence*

Formula 3.2 merupakan formula untuk menghitung *expected confidence* yang bertujuan untuk menentukan hasil dari hubungan antar *itemset* yang diharapkan, sedangkan untuk menghitung nilai *confidence* secara aktual dapat menggunakan formula 2.3 yang bertujuan untuk menemukan nilai *confidence* atau kepastian dari masing-masing pola yang ditemukan. Setelah nilai dari *confidence* dan *expected confidence* dihasilkan, perhitungan untuk menemukan nilai *lift* dapat dilakukan dengan menggunakan formula 3.3. Nilai *lift* yang ditemukan dapat menjadi wawasan dalam mengetahui kekuatan hubungan yang ditemukan.

$$\text{Lift Ratio} = \frac{\text{Confidence}}{\text{Expected Confidence}}$$

Formula 3.3 Perhitungan *lift ratio*

Analisis Nilai *Lift*:

1.  $Lift > 1.0$  menyatakan bahwa kemunculan *antecedent* dan *consequent* yang lebih sering dengan yang diharapkan. Hasil tersebut menunjukkan keterkaitan yang kuat antara *antecedent* dan *consequent* pada *rule*.
2.  $Lift < 1.0$  menyatakan bahwa kemunculan *antecedent* dan *consequent* yang lebih sedikit dengan yang diharapkan. Hasil tersebut menunjukkan keterkaitan yang lemah antara *antecedent* dan *consequent* pada *rule*.
3.  $Lift \approx 0$  menyatakan bahwa kemunculan *antecedent* dan *consequent* yang hampir selalu bersamaan dengan yang diharapkan. Hasil tersebut menunjukkan keterkaitan yang tidak signifikan antara *antecedent* dan *consequent* pada *rule*.

### 3.3.2 System Usability Scale (SUS)

*Usability* adalah ukuran seberapa mudah dan nyamannya sebuah aplikasi dapat digunakan oleh pengguna untuk mencapai tujuan tertentu. *Usability* dapat diukur dengan beberapa faktor, seperti kemudahan penggunaan, efisiensi, dan kepuasan pengguna (Saputra, 2019). *Usability* dapat didefinisikan sebagai suatu pendekatan sistematis dalam merancang, mengembangkan, dan mengevaluasi sistem interaktif dengan tujuan untuk meningkatkan kemudahan penggunaan sistem tersebut. (Sidik, 2018). Kualitas *usability* dalam suatu produk dapat diukur dan diuji untuk diketahui tingkat kegunaannya. Pengujian *usability* merupakan pengujian yang menekankan pada responden dalam proses pengujiannya (Ependi dkk., 2019). Agar dapat mengetahui apakah aplikasi yang disediakan bermanfaat bagi penggunanya, perlu dilakukan evaluasi terhadap pandangan pengguna untuk mengukur tingkat manfaatnya. (Ardhana, 2022). Salah satu metode yang dapat digunakan untuk mengevaluasi *usability* adalah *System Usability Scale*. Menurut Maryati dkk. (2022) mengemukakan bahwa SUS merupakan salah satu metode penilaian kegunaan pengguna yang populer digunakan. SUS berguna untuk melakukan pengukuran dengan cepat terkait persepsi pengguna terhadap kemudahan penggunaan aplikasi yang digunakan (Firmansyah, 2021). Berikut merupakan 10 pertanyaan dalam SUS yang terdapat pada tabel 3.1, berisikan pertanyaan yang ditujukan kepada pengguna setelah menggunakan aplikasi sehingga dapat mengetahui tingkat *usability*.

Tabel 3.1 Pertanyaan SUS

No	Pertanyaan SUS
1	Saya akan kembali menggunakan aplikasi ini.
2	Saya pikir aplikasi ini sulit digunakan.
3	Saya pikir aplikasi ini mudah dan sederhana untuk digunakan.
4	Saya membutuhkan bimbingan dari orang lain untuk menggunakan aplikasi ini.
5	Saya merasa fitur dalam aplikasi ini berfungsi dengan semestinya.
6	Saya pikir banyak inkonsistensi dalam aplikasi ini.
7	Saya beranggapan orang lain akan dengan mudah memahami cara menggunakan aplikasi ini.
8	Saya pikir aplikasi ini sangat membingungkan ketika digunakan.
9	Saya merasa tidak ada kesulitan dalam menggunakan aplikasi ini.
10	Saya perlu beradaptasi terlebih dahulu dalam mengoperasikan aplikasi ini.

Berdasarkan tabel 3.1, SUS menggunakan skala 1–5, dimana nilai 1 berarti sangat tidak setuju dan nilai 5 berarti sangat setuju. Sebagaimana Saputra (2019), *System Usability Scale* mengukur kegunaan suatu sistem dengan menggunakan lima (5) skala, yaitu dari yang sangat setuju hingga sangat tidak setuju. Rincian klasifikasi skala nilai dari SUS dapat dilihat pada tabel 3.2.

Tabel 3.2 Klasifikasi Nilai SUS

Klasifikasi	Skor
Sangat Tidak Setuju	1
Tidak Setuju	2
Ragu-ragu	3
Setuju	4
Sangat Setuju	5

Setelah data kuesioner dikumpulkan, hasil tersebut kemudian dikonversi berdasarkan skor yang telah ditentukan. Berikut ini merupakan mekanisme konversi tanggapan responden:

- a. Untuk pertanyaan ganjil (1,3,5,7, dan 9), skor yang diberikan oleh responden tidak sama dengan skor yang diberikan. Skor yang diberikan harus dikurangi 1. Perhitungan untuk pertanyaan ganjil dilakukan menggunakan formula 3.4.

$$\text{Skor pertanyaan ganjil} = \sum Px - 1$$

Formula 3.4 Perhitungan skor SUS pertanyaan ganjil

Keterangan:  $Px$  = Jumlah pertanyaan ganjil.

- b. Pada pertanyaan genap (2,4,6, dan 8), skor yang diberikan oleh responden harus dikonversi menjadi skor yang lebih rendah dengan cara mengurangi 5. Perhitungan skor pada pertanyaan genap dilakukan menggunakan formula 3.5.

$$\text{Skor pertanyaan genap} = \sum 5 - Pn$$

Formula 3.5 Perhitungan skor SUS pertanyaan genap

Keterangan:  $Pn$  = Jumlah pertanyaan genap.

- c. Setelah dilakukan konversi skor tersebut, formula 3.6 digunakan untuk menghitung jumlah dari keseluruhan skor ganjil dan genap yang telah dikonversi.

$$(\sum \text{skor ganjil} + \sum \text{skor genap}) \times 2.5$$

Formula 3.6 Perhitungan keseluruhan skor ganjil dan genap pada SUS

- d. Untuk mendapatkan rata-rata skor, kita perlu menjumlahkan semua skor yang diperoleh berdasarkan formula 3.6, lalu membagi hasilnya dengan jumlah responden. Formula 3.7 digunakan untuk mendapatkan rata-rata skor tersebut.

$$\bar{X} = \frac{\sum x}{n}$$

Formula 3.7 Perhitungan rata-rata skor SUS

Keterangan:

$\bar{X}$  = Skor rata-rata

$\sum x$  = Jumlah skor SUS

$n$  = Jumlah responden

Langkah terakhir adalah dengan menentukan klasifikasi penilaian berdasarkan hasil akhir atau rata-rata yang telah didapatkan sebelumnya. Terdapat lima (5) kategori penilaian dalam SUS. Rincian klasifikasi penilaian tersebut terdapat pada tabel 3.3.

Tabel 3.3 Kategori Penilaian SUS

<i>Grade</i>	<i>Score Range</i>	<i>Adjective Rating</i>
<b>A</b>	>80.3	<i>Excellent</i>
<b>B</b>	68–80.3	<i>Good</i>
<b>C</b>	68	<i>OK</i>
<b>D</b>	51–67	<i>Poor</i>
<b>F</b>	<51	<i>Awful</i>

### 3.3.3 Net Promoter Score (NPS)

*Net Promoter Score* adalah metrik yang mengukur loyalitas pengguna dengan menanyakan kepada mereka apakah mereka akan merekomendasikan situs web kepada orang lain (Wisnu dkk., 2023). Jawaban pengguna dengan angka antara 0–10 dapat menunjukkan seberapa besar loyalitas dan kepuasan mereka (Puspitaningtias dkk., 2020). Loyalitas produk/layanan adalah sikap pelanggan yang menyukai dan puas dengan produk atau layanan tertentu, sehingga mereka bersedia untuk terus menggunakan produk atau layanan tersebut (Wisnu dkk., 2023). Loyalitas pelanggan menurut NPS dapat dikelompokkan menjadi tiga jenis, yaitu *Detractor*, *Netral* atau *Passive*, dan *Promoter* (Supriyadi dkk., 2022). Sebagaimana Dewantoro dan Ismail (2023) dalam penelitiannya dengan judul “*Pengujian Aplikasi PNJ Bergerak Menggunakan Metode System Usability Scale dan Net Promoter Score*” menjelaskan bahwa pengguna yang memberikan skor 9 atau 10 adalah pengguna yang loyal dan cenderung merekomendasikan aplikasi kepada orang lain. Pengguna yang memberikan skor 7 atau 8 adalah pengguna yang puas, tetapi berpotensi beralih ke aplikasi lain. Pengguna yang memberikan skor rendah (0-6) pada aplikasi menunjukkan ketidakpuasan mereka dan kemungkinan besar tidak akan menggunakan aplikasi tersebut lagi. Hal ini menunjukkan bahwa kepuasan pengguna merupakan faktor penting yang dapat mempengaruhi

keberhasilan suatu produk atau layanan. Dalam NPS hanya terdapat satu pertanyaan, berikut merupakan pertanyaan dalam NPS yang terdapat pada tabel 3.4.

Tabel 3.4 Pertanyaan NPS

No	Pertanyaan
1	Berapa besar kemungkinan Anda akan merekomendasikan produk/aplikasi ini kepada teman-teman Anda?

Tanggapan yang diberikan responden diklasifikasikan dengan ketentuan kategori yang terdapat pada tabel 3.5.

Tabel 3.5 Kategori Penilaian NPS

Kategori	Rentang Nilai
<i>Detractor</i>	0–6
<i>Passive</i>	7–8
<i>Promoter</i>	9–10

Setelah itu, untuk menentukan tingkat kepuasan pengguna maka digunakan persamaan yang terdapat pada formula 3.8.

$$NPS = \%Promoter - Detractor$$

Formula 3.8 Perhitungan nilai NPS

Hasil yang telah didapatkan menggunakan persamaan pada formula 3.8 diklasifikasikan kembali berdasarkan skor yang dihasilkan. Klasifikasi skor NPS dapat dilihat pada tabel 3.6. Klasifikasi skor NPS terdiri dari empat kategori, yaitu tidak valid, perlu perbaikan, direkomendasikan, dan sangat direkomendasikan. Proses klasifikasi ini dilakukan untuk mendapatkan gambaran yang lebih jelas tentang tingkat kepuasan pelanggan terhadap produk atau layanan yang ditawarkan. Dengan mengetahui klasifikasi skor NPS, dapat dilakukan pengambilan langkah-langkah untuk meningkatkan kepuasan pelanggan. Langkah-langkah tersebut dapat berupa perbaikan produk atau layanan, peningkatan kualitas pelayanan, atau penambahan fitur-fitur baru.

Tabel 3.6 Klasifikasi Nilai NPS

<b>Skor</b>	<b>Hasil</b>
<0	Tidak Valid
0–49	Perlu Perbaikan
50–75	Direkomendasikan
76–100	Sangat Direkomendasikan

NPS berkisar dari -100 (semua pelanggan tidak puas) hingga 100 (semua pelanggan sangat senang) dan ditampilkan dalam bentuk bilangan bulat bukan persentase (Koladycz dkk., 2018).

### 3.4 Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian ini adalah seperti yang terdapat pada tabel 3.7, meliputi perangkat keras dan perangkat lunak yang diperlukan untuk pelaksanaan dan analisis data.

Tabel 3.7 Alat dan Bahan

<b>Perangkat</b>	<b>Rincian</b>
<b>Perangkat Keras</b>	<ul style="list-style-type: none"> <li>a. Intel(R) Core(TM) i5-5300U CPU @ 2.30GHz <i>Processor</i></li> <li>b. RAM 8.00 GB DDR3</li> <li>c. SSD 512 GB</li> <li>d. GPU Intel HD Graphics 5500</li> </ul>
<b>Perangkat Lunak</b>	<ul style="list-style-type: none"> <li>a. Windows 11 Pro <i>Operation System</i></li> <li>b. Microsoft Office</li> <li>c. Microsoft Edge</li> <li>d. Mendeley Reference Manager</li> <li>e. Harzing Publish or Perish</li> <li>f. Figma</li> <li>g. Visual Studio Code</li> <li>h. Github Desktop</li> </ul>