



## TUGAS AKHIR

# REENGINEERING APLIKASI TAPATUPA BERBASIS MOBILE

11421004

**GABRIEL IGNATIUS SITUMEANG**

11421018

**YUDI FRANDIANTO SARAGIH**

11421048

ANITA LUCIANA MUNTHER

**PROGRAM STUDI SARJANA TERAPAN**  
**TEKNOLOGI REKAYASA PERANGKAT LUNAK**  
**FAKULTAS VOKASI**  
**INSTITUT TEKNOLOGI DEL**

2025



## **TUGAS AKHIR**

### ***REENGINEERING APLIKASI TAPATUPA BERBASIS MOBILE***

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar  
Sarjana Terapan Komputer (S.Tr.Kom)**

**11421004                      GABRIEL IGNATIUS SITUMEANG**

**11421018                      YUDI FRANDIANTO SARAGIH**

**11421048                      ANITA LUCIANA MUNTHER**

**PROGRAM STUDI SARJANA TERAPAN  
TEKNOLOGI REKAYASA PERANGKAT LUNAK  
FAKULTAS VOKASI  
INSTITUT TEKNOLOGI DEL**

**2025**

## PERNYATAAN ORISINALITAS

Tugas akhir ini adalah hasil karya kami sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah kami nyatakan dengan benar.

NAMA : Gabriel Ignatius Situmeang


NIM : 11421004

TANDA TANGAN : 

TANGGAL : 2 Juli 2025

NAMA : Yudi Frandianto Saragih

NIM : 11421018

TANDA TANGAN : 

TANGGAL : 2 Juli 2025

NAMA : Anita Luciana Munthe

NIM : 11421048

TANDA TANGAN : 

TANGGAL : 2 Juli 2025

## HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh:

- |                   |  |
|-------------------|--|
| 1. Nama           | : Gabriel Ignatius Situmeang                             |
| NIM               | : 11421004   |
| 2. Nama           | : Yudi Fradianto Saragih                                 |
| NIM               | : 11421018   |
| 3. Nama           | : Anita Luciana Munthe                                   |
| NIM               | : 11421048   |
| Program Studi     | : Sarjana Terapan Teknologi Rekayasa Lunak               |
| Judul Tugas Akhir | : <i>Reengineering Aplikasi Tapatupa Berbasis Mobile</i> |

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Terapan Komputer (S.Tr.Kom) pada Program Studi Sarjana Terapan Teknologi Rekayasa Perangkat Lunak, Fakultas Vokasi, Institut Teknologi Del.

## DEWAN PENGUJI

Pembimbing : Ardiles Sinaga, S.T., M.T.

()

Penguji 1 : Oppir Hutapea, S.Tr.Kom., M.Kom.

()

Penguji 2 : Yohanssen Pratama, S.Si., M.T.

()

Ditetapkan di : Laguboti  
Tanggal : 2 Juli 2025

## PERNYATAAN PERSETUJUAN PUBLIKASI

Sebagai civitas akademik Institut Teknologi Del, kami yang bertanda tangan dibawah ini:

- |         |                              |
|---------|------------------------------|
| 1. Nama | : Gabriel Ignatius Situmeang |
| NIM     | : 11421004                   |
| 2. Nama | : Yudi Frandianto Saragih    |
| NIM     | : 11421018                   |
| 3. Nama | : Anita Luciana Munthe       |
| NIM     | : 11421048                   |

Program Studi	: Sarjana Terapan Teknologi Rekayasa Perangkat Lunak
Fakultas	: Vokasi
Jenis Karya	: Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Del **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah penulis yang berjudul:

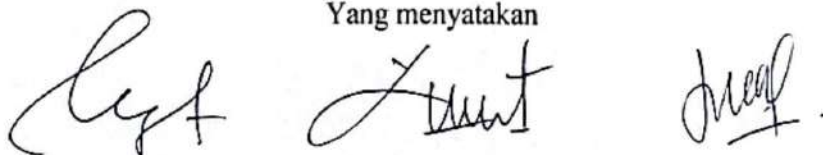
***Reengineering Aplikasi Tapatupa berbasis Mobile***

beserta seluruh perangkat yang menyertainya. Dengan Hak Bebas Royalti Noneksklusif ini, Institut Teknologi Del berhak menyimpan, mengalih/media format dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir kami selama tetap mencantumkan nama kami sebagai penulis/ dan Hak Cipta.

Demikian pernyataan ini kami buat dengan sebenarnya.

Laguboti, 2 Juli 2025

Yang menyatakan



(Gabriel Ignatius Situmeang) (Yudi Frandianto Saragih) (Anita Luciana Munthe)

## ABSTRAK

Nama : 1. Gabriel Ignatius Situmeang  
2. Yudi Frandianto Saragih  
3. Anita Luciana Munthe  
Program Studi : Sarjana terapan Teknologi Rekayasa Perangkat Lunak  
Judul : *Reengineering* Aplikasi Tapatupa Berbasis *Mobile*

Meningkatnya kebutuhan digitalisasi dalam tata kelola pemerintahan daerah mendorong pengembangan sistem informasi, salah satunya dalam pengelolaan aset tanah milik daerah. Aplikasi Tapatupa dikembangkan sebagai solusi administrasi digital bagi masyarakat penyewa aset tanah di Kabupaten Tapanuli Utara. Namun, versi awal aplikasi menghadapi berbagai tantangan terkait keterpakaian (*usability*), struktur antarmuka, serta kesesuaian fungsionalitas dengan kebutuhan pengguna. Untuk melakukan perbaikan hal tersebut, dilakukan *reengineering* aplikasi Tapatupa versi *mobile* berbasis pendekatan *User-Centered Design* (UCD) guna meningkatkan efektivitas pengelolaan aset dan memenuhi ekspektasi pengguna.

Pendekatan yang digunakan mencakup lima tahapan *reengineering* yaitu *inventory analysis*, *document restructuring*, *code refactoring*, *data restructuring*, dan *forward engineering*. Proses pengembangan dilakukan secara iteratif menggunakan kerangka kerja Agile Scrum. Setelah sistem *direengineering*, dilakukan evaluasi kepuasan pengguna melalui kuesioner *usability* yang disebarkan kepada responden di tiga kecamatan. Hasil evaluasi menunjukkan nilai rata-rata sebesar 79,19% dengan kriteria layak, menandakan adanya peningkatan dari segi kemudahan akses, kejelasan informasi, dan kepuasan pengguna. Instrumen evaluasi juga telah diuji validitas dan reliabilitasnya dengan nilai *Cronbach's Alpha* sebesar 0,71, menunjukkan konsistensi internal yang tinggi.

Meski demikian, masih ditemukan beberapa keterbatasan dalam cakupan pengujian dan integrasi sistem. Oleh karena itu, penelitian lanjutan disarankan untuk memperluas instrumen evaluasi serta melanjutkan pengujian secara kolaboratif bersama pemangku kepentingan seperti BKAD, guna menjamin keberlanjutan dan adaptasi sistem di masa mendatang.

### Kata Kunci:

Aplikasi *Mobile*, *Reengineering*, Sistem Informasi Pemerintahan Digital, *Usability*, *User-Centered Design*

## ABSTRACT

Name : 1. Gabriel Ignatius Situmeang  
2. Yudi Frandianto Saragih  
3. Anita Luciana Munthe  
Study Program : Bachelor of Applied Software Engineering Technology  
Title : *Reengineering Aplikasi Tapatupa Berbasis Mobile*

The increasing need for digitalization in regional governance has driven the development of information systems, including the management of regional land assets. One such initiative is the Tapatupa application, developed to support digital administration for land assets leased to the public. However, the initial version of this mobile-based application faced several challenges, particularly in terms of usability, interface structure, and functional alignment with user needs. This final project addresses these issues by conducting a reengineering process on the mobile-based Tapatupa application in a more structured and user-centered manner, aiming to improve the effectiveness of land asset management and meet user expectations as lessees and taxpayers.

The approach used includes five stages of reengineering: inventory analysis, document restructuring, code refactoring, data restructuring, and forward engineering. The development was conducted iteratively using the Agile Scrum framework and the principles of User-Centered Design (UCD). After the system was completed, user satisfaction evaluation was carried out using a usability questionnaire distributed to respondents from three sub-districts. This evaluation aimed to measure the level of usability in terms of interface and system functionality.

The final solution produced a mobile version of the Tapatupa application with an improved interface, a structured request flow, integrated digital agreement features, and real-time payment using virtual account integration. The overall evaluation results showed a score of 79.19% with acceptable criteria, reflecting improvements in clarity, ease of access, and user satisfaction with the application. The evaluation instrument was also tested for validity and reliability, yielding Cronbach's Alpha value of 0.71, indicating high internal consistency. Nevertheless, some limitations were still identified. Therefore, further research is recommended to add additional evaluation instruments and continue iterative testing with stakeholders such as BKAD to ensure the sustainability and relevance of the system in the future.

### **Keywords:**

User-Centered Design, Reengineering, Mobile Application, Usability, Government Digital Information System



## KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir dengan judul “*Reengineering Aplikasi Tapatupa berbasis Mobile.*” Penulisan Tugas Akhir ini dilakukan untuk memenuhi salah satu syarat mencapai gelar Sarjana Terapan Fakultas Vokasi Institut Teknologi Del. Penulis menyadari bahwa Tugas Akhir ini dapat disusun hingga selesai berkat bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Orang tua dan keluarga penulis yang selalu membimbing, membiayai, serta memberikan kasih sayang dan dukungan yang tiada henti kepada penulis.
2. Bapak Ardiles Sinaga, S.T., M.T. selaku dosen pembimbing yang telah memberikan waktu, tenaga, dan pikiran untuk mengarahkan Penulis dalam penyusunan Tugas Akhir ini.
3. Bapak Oppir Hutapea, S.Tr.Kom, M.Kom dan Yohanssen Pratama, S.Si, M.T selaku dosen penguji yang telah memberikan saran, masukan, dan kritik yang membangun selama pengerjaan Tugas Akhir.
4. Sahabat, teman, rekan seperjuangan, dan seluruh pihak yang tidak dapat disebutkan satu per satu yang telah membantu dan mendukung penulis dalam menyelesaikan Tugas Akhir.

Penulis menyadari bahwa masih terdapat kekurangan dan kesalahan dalam penyusunan Tugas Akhir ini. Oleh karena itu, penulis meminta maaf yang sedalam-dalamnya dan mengharapkan kritik dan saran untuk perbaikan di masa yang akan datang. Akhir kata, Semoga Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu.

Laguboti, 2 Juli 2025



(Gabriel Ignatius Situmeang)



(Yudi Frandianto Saragih)



(Anita Luciana Munthe)



## DAFTAR ISI

<b>PERNYATAAN ORISINALITAS.....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iii</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI.....</b>	<b>iv</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR TABEL .....</b>	<b>xii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan .....	4
1.4 Lingkup .....	5
1.5 Pendekatan .....	6
1.6 Metode Pengembangan : Agile Scrum Terintegrasi dengan <i>User - Centered Design</i> 7	
1.6.1 Alasan Pemilihan Scrum .....	8
1.6.2 Struktur Tim Scrum .....	8
1.6.3 Artefak Scrum .....	9
1.7 Sistematika Penyajian .....	10
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>11</b>
2.1 Software Reengineering .....	11
2.2 Aplikasi berbasis <i>Mobile</i> .....	16
2.3 Agile Scrum .....	17
2.4 Flutter .....	19
2.5 Aplikasi Sejenis.....	20
2.6 Kuesioner .....	26
2.6.1 Uji Kelayakan Kuesioner .....	27
2.6.1.1 Uji Validitas Kuesioner.....	27
2.6.1.2. Uji Reliabilitas Kuesioner .....	29
2.6.2 Evaluasi <i>Usability</i> menggunakan Custom Usability <i>Questionnaires</i> ..	30
2.6.3 Metode Evaluasi Sistem Berdasarkan Perhitungan USE Questionnaire .....	31

2.6.4 Skala Likert .....	32
2.7 Stratified Sampling.....	34
2.8 Metode <i>Slovin</i> .....	35
2.9 Responden .....	36
2.10 ISO/IEC 25010 dan Kualitas Perangkat Lunak .....	37
2.10.1 <i>Functional Completeness</i> .....	38
2.10.2 <i>Functional Correctness</i> .....	38
2.10.3 <i>Functional Appropriateness</i> .....	38
2.10.4 Penerapan ISO / IEC 25010 dalam Evaluasi Sistem.....	39
2.11 User Interface & User Experience .....	39
2.11.1 Pentingnya evaluasi UI/UX.....	39
2.11.2 Dampak Positif Implementasi UI/UX.....	40
2.11.3 Desain UI/UX sebagai acuan .....	40
2.12 Konsep Pengelolaan Aset Daerah .....	41
2.12.1 Regulasi dan Kebijakan Pengelolaan Aset Daerah .....	43
2.12.2 Transformasi Pengelolaan Aset Tanah Kabupaten Tapanuli Utara ...	43
2.13 Gambaran Umum Aplikasi Tapatupa.....	44
2.14 Penelitian Terdahulu .....	45
2.15 Kesimpulan .....	49
<b>BAB III ANALISIS DAN DESAIN .....</b>	<b>50</b>
3.1 Analisis Kebutuhan Sistem .....	50
3.1.1 <i>Understand and Specify the Context of Use</i> .....	50
3.1.2 Identifikasi <i>Stakeholder</i> dan Pengguna.....	51
3.1.2.1 User Characteristics .....	52
3.1.2.2 Use Case Target System.....	53
3.1.3 Analisis Sumber Data dan Studi Pendukung .....	54
3.1.3.1. Studi Aplikasi Sejenis .....	55
3.1.3.2. Studi Aplikasi Sebelumnya (Tapatupa versi Awal).....	57
3.1.4. <i>Current System Analysis</i> .....	59
3.1.5. <i>Target System Specification</i> .....	60
3.1.6. <i>Requirement Mapping</i> Berdasarkan Regulasi.....	62
3.1.7. <i>Specify the User Requirements</i> .....	63
3.1.8. <i>Producing Design Solutions to Meet User Requirements</i> .....	65
3.1.9. <i>Evaluating the Designs Against Requirements</i> .....	67
3.2. Analisis Tahapan <i>Software Reengineering</i> terhadap Aplikasi Tapatupa ....	68

3.2.1.	<i>Inventory Analysis</i> .....	71
3.2.2.	<i>Document Restructuring</i> .....	71
3.2.3.	<i>Code Refactoring</i> .....	72
3.2.4.	<i>Data Restructuring</i> .....	72
3.2.5.	<i>Forward Engineering</i> .....	72
3.3.	Kesimpulan .....	73
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>74</b>
4.1.	Lingkungan Implementasi .....	74
4.1.1.	Perangkat Lunak .....	74
4.1.2.	Perangkat Keras .....	74
4.2.	Batasan Implementasi .....	75
4.3.	Implementasi Sprint .....	75
4.3.1.	Pembagian Iterasi <i>Reengineering</i> .....	76
4.3.2.	<i>Sprint Backlog</i> per Iterasi .....	77
4.4.	Tahapan Implementasi .....	78
4.4.1.	Implementasi <i>Inventory Analysis</i> .....	78
4.4.1.1.	<i>Requirement Analysis</i> .....	78
4.4.1.2.	<i>Legacy</i> Fitur .....	80
4.4.1.3.	Komponen Antarmuka .....	81
4.4.2.	Implementasi <i>Document Restructuring</i> .....	82
4.4.3.	Implementasi <i>Code Refactoring</i> .....	84
4.4.4.	Implementasi <i>Data Restructuring</i> .....	89
4.4.5.	Implementasi <i>Forward Engineering</i> .....	90
4.4.5.1.	Hasil Iterasi 1 .....	90
4.4.5.1.1.	Hasil Implementasi Fungsi Iterasi 1 .....	91
4.4.5.1.2.	Hasil Implementasi Pengujian Iterasi 1 .....	91
4.4.5.2.	Hasil Iterasi 2 .....	92
4.4.5.2.1.	Desain Antarmuka .....	92
4.4.5.2.2.	Hasil Implementasi Fungsi Iterasi 2 .....	92
4.4.5.2.3.	Hasil Implementasi Pengujian Iterasi 2 .....	93
4.4.5.3.	Hasil Iterasi 3 .....	94
4.4.5.3.1.	Penyempurnaan Fungsi dan Antarmuka .....	94
4.4.5.3.2.	Hasil Implementasi Pengujian Iterasi 3 .....	94
4.4.5.3.3.	Evaluasi Kepuasan Pengguna Melalui kuesioner .....	95
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>96</b>

5.1.Kesimpulan .....	96
5.2.Saran.....	96
<b>DAFTAR PUSTAKA.....</b>	<b>xiv</b>
<b>LAMPIRAN.....</b>	<b>xix</b>

## DAFTAR TABEL

Tabel 2. 1 Analisis Website SiapSEWA .....	21
Tabel 2. 2 Analisis Website Amanah .....	23
Tabel 2. 3 Analisis Aplikasi SIWA .....	24
Tabel 2. 4 Interval Reliabilitas Cronbach's Alpha .....	30
Tabel 2. 5 Kategori Kelayakan Angka (%) .....	32
Tabel 2. 6 Kriteria Pengukuran Skala Likert .....	33
Tabel 2. 7 Design patterns and subelements .....	41
Tabel 2. 8 Similar Research .....	45
Tabel 3. 1 User Characteristics Current System .....	53
Tabel 3. 2 Objek Perbandingan .....	56
Tabel 3. 3 Aspek Yang Diadopsi .....	56
Tabel 3. 4 Functional Specification versi Awal .....	57
Tabel 3. 5 Fitur Utama Sistem Baru .....	61
Tabel 3. 6 Tasbel Mapping Kebutuhan Sistem Berdasarkan Regulasi .....	62
Tabel 3. 7 Pemetaan Fungsi Sistem Berdasarkan Masalah Pengguna .....	65
Tabel 3. 8 Komponen Antarmuka Pengguna .....	66
Tabel 3. 9 Perbaikan Fitur .....	66
Tabel 3. 10 Analisis Tahapan Software Reengineering .....	68
Tabel 4. 1 Spesifikasi Software .....	74
Tabel 4. 2 Spesifikasi Hardware .....	74
Tabel 4. 3 Pembagian Iterasi Reengineering .....	76
Tabel 4. 4 Sprint Backlog per Iterasi .....	77
Tabel 4. 5 Daftar Legacy Fitur .....	80
Tabel 4. 6 Hasil Perbaikan Antarmuka Pengguna .....	81
Tabel 4. 7 Aspek Dokumen Restrukturisasi .....	82
Tabel 4. 8 Hasil Aspek Dokumen Restrukturisasi .....	83
Tabel 4. 9 Evaluasi Refactoring Fitur Permohonan .....	85
Tabel 4. 10 Evaluasi Refactoring Fitur Tagihan .....	87
Tabel 4. 11 Evaluasi Refactoring Fitur Perjanjian .....	88
Tabel 4. 12 Ringkasan Evaluasi Keseluruhan Sistem .....	88
Tabel 4. 13 Hasil Aspek Data Restrukturisasi .....	90
Tabel 4. 14 Implementasi fungsi iterasi I .....	91
Tabel 4. 15 Pengujian Implementasi Fungsi Iterasi I .....	91
Tabel 4. 16 Implementasi Fungsi Iterasi II .....	93
Tabel 4. 17 Pengujian Implementasi Fungsi Iterasi II .....	94

## DAFTAR GAMBAR

Gambar 1. 1 Tahapan UCD.....	7
Gambar 2. 1 Model Proses Software Reengineering .....	12
Gambar 2. 2 Tampilan Beranda Website SiapSEWA.....	21
Gambar 2. 3 Gambar Tampilan Detail Website SiapSEWA .....	22
Gambar 2. 4 Tampilan Hak Aset Website SiapSEWA .....	22
Gambar 2. 5 Tampilan Beranda Website Amanah.....	23
Gambar 2. 6 Tampilan Pembayaran.....	23
Gambar 2. 7 Tampilan Beranda .....	24
Gambar 2. 8 Tampilan Halaman Pengajuan .....	25
Gambar 2. 9 Tampilan Halaman Keranjang Pengguna.....	26
Gambar 3. 1 Use Case Diagram setelah Reengineering .....	54

# **BAB I**

## **PENDAHULUAN**

Pada bab pendahuluan ini, akan diuraikan mengenai latar belakang, permasalahan penelitian, hasil yang diharapkan, tahapan penelitian dan sistematika penulisan laporan tugas akhir.

### **1.1 Latar Belakang**

Aset tanah daerah merupakan bagian dari Barang Milik Daerah (BMD) yang memiliki peran strategis dalam mendukung pembangunan dan pengelolaan keuangan pemerintah daerah. Aset ini dapat dimanfaatkan sebagai sumber pendapatan, penyediaan layanan publik, dan peningkatan nilai ekonomi melalui investasi. Menurut Santoso et al. [1], pengelolaan aset tanah daerah yang baik dapat meningkatkan akurasi pencatatan serta memperkuat transparansi dan akuntabilitas pemerintahan daerah. Pengelolaan aset tanah daerah harus dilakukan secara optimal, transparan, dan bertanggung jawab agar dapat memberikan manfaat maksimal bagi Masyarakat serta pendapatan daerah. Namun, pengelolaan aset tanah daerah masih menghadapi tantangan seperti ketidakakuratan data, keterlambatan proses administrasi, dan keterbatasan akses terhadap data aset. Oleh karena itu, diperlukan sistem pengelolaan aset yang lebih modern dan berbasis teknologi untuk mengatasi permasalahan ini.

Studi oleh Priyono Budi dalam "Pengelolaan Aset Daerah Berupa Tanah dan Bangunan sebagai Upaya Optimalisasi Pendapatan Asli Daerah (PAD)" [2] menyoroti bahwa di Kota Depok, aset daerah berupa tanah dan bangunan dimanfaatkan melalui tiga bentuk utama: sewa, pinjam pakai, dan kerja sama pemanfaatan. Pemanfaatan aset melalui sewa dapat meningkatkan pendapatan asli daerah jika dikelola dengan baik. Namun, pencatatan yang masih dilakukan secara manual menyebabkan ketidakakuratan data dan hambatan dalam pengambilan keputusan berbasis data. Menurut Melly Nur Qurani dan Maulidah Narastri [3] juga mengatakan bahwa keterbatasan sumber daya manusia yang kompeten serta infrastruktur digital yang belum memadai semakin memperburuk permasalahan ini. Hal ini menunjukkan pentingnya inovasi teknologi dalam pengelolaan aset daerah. Regulasi pemerintah daerah memegang peran penting dalam memastikan pengelolaan aset tanah daerah yang transparan dan akuntabel. Salah satu contohnya



mengacu pada Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024 [4], yang mengatur pengelolaan barang milik daerah berupa tanah dan bangunan dan tata cara penerapan tarif sewa barang milik daerah berupa tanah dan / atau bangunan. Regulasi ini bertujuan untuk memastikan aset dikelola secara tepat dan memberikan manfaat maksimal bagi pemerintah daerah dan masyarakat. Di sisi lain, pemanfaatan aset dapat dilakukan melalui penyewaan atau kemitraan dengan sektor swasta. Selain itu, tanggung jawab pemeliharaan aset ditanggung oleh penyewa, yang membantu mengurangi biaya pendanaan untuk pemerintah daerah. Namun, implementasi di lapangan masih menghadapi kendala, terutama dalam sistem pencatatan yang belum terintegrasi secara digital. Oleh karena itu, solusi berbasis teknologi diperlukan untuk meningkatkan keakuratan pengelolaan aset.

Sebagai respons terhadap permasalahan pengelolaan aset, berbagai pemerintah daerah telah mulai mengadopsi teknologi informasi, termasuk GIS dan sistem pencatatan berbasis web maupun mobile(GIS) [5] Dirgantara menunjukkan bahwa GIS mendukung pemetaan aset yang lebih akurat dan mempercepat pengambilan keputusan data, sistem pencatatan aset berbasis *web* dan *mobile*, seperti E-BMD (*Electronic* Barang Milik Daerah) juga yang telah diterapkan dalam tata kelola aset di beberapa instansi pemerintahan seperti yang dilakukan dalam penelitian Nur et al [6]. Sistem ini memungkinkan pencatatan aset secara *real-time* dan meningkatkan akurasi data yang telah dibuat pada penelitian analisis dan efek audit legal oleh S. Hasanah et al [7]. Namun, sistem ini masih memiliki keterbatasan dalam fleksibilitas dan aksesibilitas bagi pengguna di lapangan. Untuk itu, diperlukan solusi yang lebih inovatif dan sesuai dengan kebutuhan pemerintah daerah.

Salah satu kabupaten yang telah menerapkan sistem pengelolaan aset berbasis teknologi adalah Kabupaten Tapanuli Utara, yang terletak di Provinsi Sumatera Utara, merupakan daerah yang kaya akan potensi sumber daya alam dan memiliki sektor pertanian serta pariwisata yang berkembang. Kabupaten ini dikenal dengan Danau Toba dan sektor agrarisnya yang kuat, terutama dalam produksi kopi dan hortikultura. Dalam rangka meningkatkan efisiensi administrasi aset daerah, Kabupaten Tapanuli Utara dengan aplikasi Tapatupa. Aplikasi ini dikembangkan oleh peneliti Ardiles et al [8] sebagai upaya untuk mengatasi kendala dalam

implementasi sistem pengelolaan aset secara manual, Pemerintah Kabupaten Tapanuli Utara menginisiasi peluncuran aplikasi Tapatupa. Tapatupa memungkinkan pencatatan aset secara digital, pengelolaan kontrak penyewaan, serta pemantauan tagihan dan pembayaran sewa secara *real-time*.

Kabupaten Tapanuli Utara memanfaatkan Tapatupa dalam pengelolaan aset tanah di berbagai desa sebagai langkah modernisasi administrasi aset daerah. Aplikasi Tapatupa bertujuan untuk mendokumentasikan serta mengelola aset daerah secara lebih sistematis. Beberapa fitur utama yang direncanakan dalam aplikasi ini meliputi pengajuan permohonan penyewaan aset, unggah dokumen pendukung oleh wajib retribusi, penampilan detail tagihan sewa, serta proses pembayaran retribusi yang telah terintegrasi langsung dengan bank daerah. Dengan adanya sistem ini, diharapkan proses pengelolaan aset dapat lebih transparan dan terstruktur. Meskipun menawarkan potensi kemudahan, implementasi Tapatupa masih menghadapi beberapa tantangan, seperti antarmuka pengguna yang masih perlu ditingkatkan dalam hal kemudahan navigasi dan responsivitas, fitur pencarian aset yang terbatas, aplikasi belum sepenuhnya terintegrasi dengan sistem pemerintah daerah, pencatatan masih memerlukan proses manual yang dapat memperpanjang waktu pemrosesan serta kendala dalam unggah dokumen yang belum berfungsi dengan baik dalam fitur permohonan seharusnya dapat diidentifikasi dengan jelas [9].

Untuk mengatasi kendala tersebut, diperlukan *reengineering* aplikasi berbasis *mobile* guna meningkatkan pengalaman pengguna dan fungsionalitas sistem. Langkah ini mencakup perancangan ulang antarmuka, peningkatan pengajuan permohonan, tagihan sewa, pembayaran yang sudah terintegrasi, fitur pencarian aset, serta optimalisasi fitur unggah dokumen. Hal ini didukung dengan hasil survei terhadap pengguna Tapatupa di tiga kecamatan sebagai sampel penelitian yang menunjukkan bahwa aplikasi Tapatupa perlu perbaikan.

Seiring dengan proses pengembangan, pendekatan *User-Centered Design* (UCD) yang digunakan dalam penelitian ini membantu tim untuk terus berfokus pada kebutuhan pengguna nyata, yaitu wajib retribusi sebagai pengguna utama sistem. Melalui proses iteratif, ditemukan bahwa pengguna memiliki kebutuhan spesifik seperti kemudahan dalam mengajukan permohonan penyewaan aset, mengunggah

dokumen pendukung, serta melihat detail tagihan dan melakukan pembayaran secara *real-time* melalui perangkat digital. Kebutuhan-kebutuhan ini semakin teridentifikasi secara jelas pada iterasi ketiga, mencerminkan kondisi nyata dan sesuai dengan kebutuhan pengguna dalam mengakses layanan secara mandiri.

Sebagai tahap akhir, dilakukan evaluasi terhadap sistem yang telah dikembangkan dengan menyebarkan kuesioner kepuasan pengguna. Kuesioner ini digunakan untuk mengukur sejauh mana aplikasi hasil *reengineering* telah memenuhi harapan pengguna, baik dari aspek antarmuka maupun fungsionalitas. Hasil evaluasi tersebut memberikan umpan balik kuantitatif yang mendukung kesimpulan bahwa proses *reengineering* berhasil meningkatkan kualitas sistem. Temuan ini memberi gambaran pentingnya penerapan metode *Agile Scrum* yang adaptif serta pendekatan *User-Centered Design* (UCD) yang berorientasi pada kebutuhan nyata pengguna. Dengan proses *reengineering* ini, Tapatupa diharapkan dapat berkembang menjadi sistem pengelolaan aset digital yang terstruktur dan sesuai dengan harapan masyarakat sebagai pengguna utama layanan.

## 1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah diidentifikasi, pertanyaan penelitian yang akan menjadi pedoman serta tujuan utama dalam penelitian ini adalah bagaimana merancang ulang aplikasi Tapatupa berbasis *mobile* agar dapat mendukung proses pengelolaan aset tanah secara lebih terstruktur dan sesuai dengan kebutuhan pengguna melalui pendekatan yang berpusat pada pengguna (*User-Centered Design*)?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah melakukan proses perancangan ulang pada aplikasi Tapatupa berbasis *mobile* dalam pengelolaan aset tanah dengan pendekatan desain iteratif dan berpusat pada pengguna, dengan menggunakan pendekatan *User-Centered Design* dan desain iteratif, yang dibuktikan melalui pengembangan prototipe, dokumentasi perancangan, serta evaluasi desain berdasarkan kuesioner dan umpan balik pengguna.

## 1.4 Lingkup

Ruang lingkup penelitian aplikasi pengelolaan aset tanah berbasis *mobile* mencakup:

1. Pengguna yang menjadi target utama dalam sistem ini adalah masyarakat Kabupaten Tapanuli Utara yang telah memiliki Nomor Pokok Wajib Pajak (NPWP) dan ingin mengajukan sewa aset tanah milik daerah.
2. Penelitian ini difokuskan pada proses *reengineering* aplikasi Tapatupa berbasis *mobile*, dengan tujuan untuk meningkatkan kualitas antarmuka pengguna dan menyempurnakan fitur utama yang digunakan oleh masyarakat sebagai wajib retribusi. Fitur yang menjadi ruang lingkup pengembangan mencakup: pengajuan permohonan sewa aset tanah, akses informasi objek dan tarif retribusi, pengelolaan perjanjian, tagihan, serta integrasi pembayaran melalui sistem *virtual account* dari bank daerah. Pengembangan diarahkan untuk meningkatkan keterpakaian (*usability*), kemudahan navigasi, serta transparansi informasi dalam pengelolaan aset daerah di Kabupaten Tapanuli Utara.
3. Tahapan *reengineering* yang dilakukan adalah *inventory analysis*, *document restructuring*, *code refactoring*, *data refactoring* dan *forward engineering*.
4. Aplikasi dikembangkan sebagai aplikasi berbasis *mobile* (Android) dengan *framework* Flutter, baik pada versi sebelum maupun sesudah *reengineering*. Meskipun tidak terjadi perubahan *framework*, proses *reengineering* difokuskan pada penyempurnaan struktur kode (*code refactoring*), peningkatan desain antarmuka, dan alur kerja pengguna (*user flow*), yang semuanya dilakukan secara iteratif menggunakan pendekatan *User-Centered Design* (UCD).
5. Evaluasi dilakukan melalui *usability testing* terhadap prototipe dan sistem hasil *reengineering* menggunakan metode *custom usability questionnaire* dan perhitungan tingkat kepuasan menggunakan kuesioner berbasis *USE Questionnaire*.

6. Objek kajian mencakup seluruh wilayah administratif sebanyak 15 kecamatan di Kabupaten Tapanuli Utara, dengan 3 kecamatan dipilih sebagai sampel untuk pengumpulan data guna mewakili kebutuhan dan permasalahan pengguna.

### 1.5 Pendekatan

Pendekatan penelitian yang digunakan dalam proyek reengineering aplikasi Tapatupa adalah *User-Centered Design* (UCD). Pendekatan ini dipilih karena menempatkan pengguna sebagai pusat dalam proses analisis, perancangan, pengembangan, dan evaluasi sistem, yang sejalan dengan kebutuhan peningkatan keterpakaian (*usability*) dan kepuasan pengguna. Menurut standar ISO 9241-210:2010, terdapat empat aktivitas utama dalam pendekatan UCD [10] :

#### 1. *Understand and Specify the Context of Use*

Aktivitas ini bertujuan untuk memahami siapa pengguna sistem, apa tujuan mereka, dalam situasi seperti apa sistem digunakan, serta kendala yang mereka hadapi.

Pada tahap ini, identifikasi konteks dilakukan melalui observasi terhadap aplikasi Tapatupa versi sebelumnya, analisis alur kerja lama, dan studi terhadap dokumen resmi, khususnya Peraturan Bupati No. 12 Tahun 2024 tentang pengelolaan retribusi aset. Selain itu, dilakukan analisis terhadap aplikasi sejenis guna membandingkan fitur dan pendekatan layanan terhadap pengguna.

#### 2. *Specify the User Requirements*

Berdasarkan pemahaman terhadap konteks penggunaan, tahap ini bertujuan untuk menentukan kebutuhan eksplisit pengguna terhadap sistem baru. Penentuan kebutuhan dilakukan melalui analisis terhadap sistem Tapatupa versi sebelumnya, termasuk alur kerja pengguna, kelemahan pada fitur-fitur lama, serta studi terhadap regulasi resmi seperti Peraturan Bupati No. 12 Tahun 2024 yang mengatur dokumen pendukung, proses sewa, dan alur pembayaran.

#### 3. *Produce Design Solutions to Meet User Requirements*

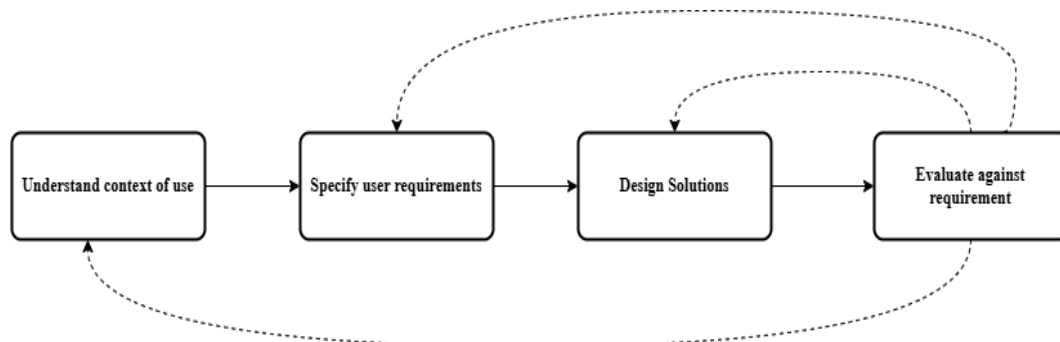
Setelah kebutuhan pengguna ditentukan, solusi desain dikembangkan untuk memenuhi kebutuhan tersebut.

Dalam penelitian ini, desain diwujudkan dalam bentuk *wireframe* dan prototipe menggunakan Figma, serta disiapkan untuk implementasi ulang (*reengineering*) terhadap fitur yang belum memenuhi ekspektasi pengguna berdasarkan hasil evaluasi sebelumnya. Proses ini juga mencakup pengembangan fitur baru seperti integrasi *virtual account*, dan pengelolaan perjanjian digital yang sebelumnya belum tersedia. Tahap ini menjadi titik awal dimulainya pembangunan sistem (*development*) dengan memperhatikan hasil desain yang tervalidasi.

#### 4. Evaluate the Designs Against Requirements

Evaluasi dilakukan untuk menguji apakah solusi yang dikembangkan telah sesuai dengan kebutuhan dan ekspektasi pengguna.

Evaluasi dalam proyek ini dilakukan dengan *usability testing* terhadap prototipe dan sistem berjalan menggunakan metode *custom usability questionnaire* dengan menghitung dan menginterpretasi skor kuesioner likert. Evaluasi dilakukan melalui *usability testing* terhadap prototipe dan sistem akhir menggunakan kuesioner berbasis skala Likert. Nilai rata-rata dari setiap indikator digunakan untuk menilai keberhasilan antarmuka dan fungsi sistem, serta menjadi dasar perbaikan pada iterasi berikutnya.



Gambar 1. 1 Tahapan UCD

### 1.6 Metode Pengembangan : Agile Scrum Terintegrasi dengan User - Centered Design

Metode pengembangan sistem yang digunakan dalam proyek *reengineering* aplikasi Tapatupa mengadopsi *Agile Scrum* yang terintegrasi dengan praktik *User-Centered Design* (UCD). Kombinasi ini dipilih karena keduanya mendukung

pengembangan sistem secara iteratif, bertahap, dan berbasis kebutuhan pengguna. Pendekatan ini memungkinkan evaluasi dan penyesuaian sistem dilakukan secara terus-menerus sepanjang siklus pengembangan [11] [12].

### 1.6.1 Alasan Pemilihan Scrum

Scrum merupakan salah satu kerangka kerja Agile yang menekankan pada iterasi singkat (sprint), kolaborasi tim yang erat, dan responsif terhadap perubahan. Penggunaan Scrum dalam proyek ini didasari oleh beberapa alasan:

1. Adanya kebutuhan perubahan berdasarkan *feedback* pengguna secara berkala.
2. Kompleksitas modul dan fitur yang perlu disempurnakan secara bertahap.
3. Kebutuhan akan kolaborasi erat antara tim pengembang dan pemangku kepentingan.
4. Perlunya fleksibilitas dalam penyesuaian desain dan fungsi sistem dalam waktu yang terbatas.

Castillo et al. dan Alhammad & Moreno menyatakan bahwa Scrum dapat berjalan lebih optimal jika digabungkan dengan praktik UX seperti UCD atau Lean UX. Beberapa poin penting yang ditekankan dalam integrasi ini meliputi:

1. Aktivitas desain dan validasi antarmuka dilakukan sejak awal sprint, bukan hanya di akhir.
2. Umpan balik pengguna digunakan sebagai dasar prioritas backlog.
3. Validasi desain dilakukan dalam setiap iterasi [11], [12].

Dengan mempertimbangkan hal tersebut, integrasi Scrum dan UCD dipilih dalam penelitian ini untuk memastikan sistem dibangun secara *user-centric*, responsif, dan adaptif terhadap masukan selama proses *reengineering* berlangsung.

### 1.6.2 Struktur Tim Scrum

Struktur tim dalam proyek ini disesuaikan dengan *framework Scrum* dan keterbatasan jumlah personel. Tim terdiri dari tiga orang dengan pembagian peran sebagai berikut:

1. *Product Owner*: Diwakili oleh instansi pengguna akhir, yaitu Badan Pengelola Keuangan dan Aset Daerah (BPKAD). Meskipun mereka tidak



terlibat langsung dalam tim teknis, kebijakan dan kebutuhan operasional mereka menjadi acuan utama dalam pengambilan keputusan fitur sistem.

2. *Scrum Master*: Salah satu anggota tim bertindak sebagai fasilitator proses Scrum, mengatur alur kerja *sprint*, memastikan *backlog* berjalan sesuai prioritas, serta menjaga koordinasi antar anggota.
3. *Development Team*: Seluruh anggota tim merangkap sebagai pengembang, dengan pembagian fokus pada desain UI/UX, implementasi *frontend/backend*, serta pengujian dan dokumentasi teknis.

**Tabel 1. 1** Struktur Tim Scrum

Peran	Anggota
Product Owner	BPKAD (Badan Pengelola Keuangan dan Aset Daerah)
Scrum Master	Anita Luciana Munthe
Tim Pengembang	Mahasiswa Peneliti

Pendekatan ini sejalan dengan prinsip *Agile for Small Teams* yang memungkinkan adaptasi *framework Scrum* dalam tim berskala kecil secara efektif [12].

### 1.6.3 Artefak Scrum

Beberapa artefak utama dalam Scrum yang digunakan dalam pengembangan sistem Tapatupa meliputi:

1. *Product Backlog*

Merupakan daftar prioritas yang berisi kebutuhan sistem, termasuk fitur-fitur utama seperti permohonan aset, pengelolaan tagihan, perjanjian digital, serta sistem autentikasi. Backlog disusun berdasarkan kebutuhan pengguna dan diperbarui secara dinamis sesuai hasil evaluasi selama iterasi.

2. *Sprint Backlog*

Berisi tugas-tugas spesifik yang dipilih dari Product Backlog untuk dikerjakan dalam satu sprint. Setiap sprint difokuskan pada pengembangan kelompok fitur tertentu, seperti:

- a. Sprint 1: Pembuatan MVP yang mencakup fitur permohonan, tagihan, dan pembayaran.

- b. Sprint 2: Penambahan fitur informasi retribusi, autentikasi, profil, dan perjanjian digital.
- c. Sprint 3: Penyempurnaan berdasarkan hasil evaluasi *usability* dengan mengukur kepuasan pengguna melalui kuesioner dan validasi akhir.

### 3. *Increment*

Hasil nyata dari setiap *sprint*, berupa sistem yang dapat digunakan sebagian atau seluruhnya. Contohnya, pada akhir *sprint* pertama, fitur permohonan dan tagihan dapat digunakan secara fungsional dan diuji secara langsung oleh pengguna internal. Evaluasi dilakukan untuk memastikan sistem sesuai dengan tujuan *reengineering*.

## 1.7 Sistematika Penyajian

Dokumen Laporan Tugas Akhir ini terdiri dari lima bab dan disusun menurut sistematika penyajian berikut.

1. Bab I Pendahuluan, bab ini berisi penjelasan mengenai latar belakang pembangunan sistem, tujuan pengerjaan, lingkup pengerjaan, pendekatan yang digunakan, dan sistematika penyajian laporan Tugas Akhir.
2. Bab II Tinjauan Pustaka, bab ini berisi penjelasan mengenai tinjauan pustaka yang berisi landasan teori yang relevan serta metode penelitian yang digunakan dalam pelaksanaan penelitian.
3. Bab III Analisis dan Desain, bab ini berisi penjelasan mengenai analisis terhadap objek kaji sesuai dengan metode yang telah ditetapkan dan rancangan desain aplikasi yang telah dibuat.
4. Bab IV Hasil dan pembahasan, berisikan penjelasan terkait hasil dan pembahasan yang diperoleh dari pelaksanaan Tugas Akhir.
5. Bab V Kesimpulan dan Saran, bab ini berisikan penjelasan terkait hasil dan pembahasan yang diperoleh dari pelaksanaan Tugas Akhir.

## BAB II

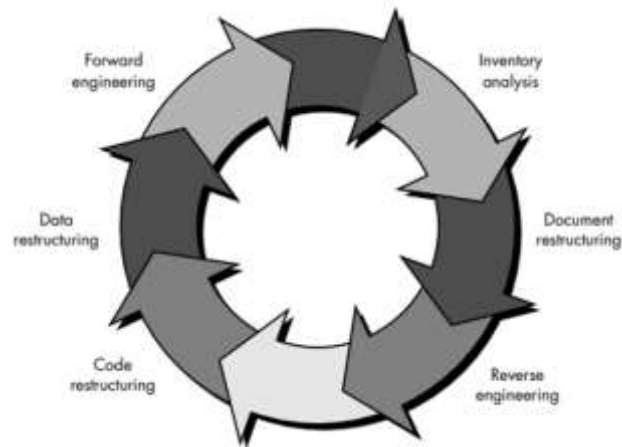
### TINJAUAN PUSTAKA

Pada bab ini dijelaskan tinjauan pustaka pada penelitian sebelumnya yang sesuai dengan topik tugas akhir yang akan digunakan dalam penelitian ini. Bab ini juga akan menjelaskan dasar-dasar teori yang akan menjadi rujukan dalam penelitian ini.

#### 2.1 Software Reengineering

*Software reengineering* merupakan kegiatan atau proses pembangunan kembali suatu perangkat lunak untuk meningkatkan kualitas [13]. Dalam konteks perangkat lunak, *software reengineering* bertujuan untuk membangun ulang suatu sistem agar memiliki kualitas yang lebih baik, meningkatkan fungsionalitas, serta mempermudah proses pemeliharaannya. Proses ini dilakukan secara sistematis, dimulai dengan memahami dan mengidentifikasi kondisi sistem yang telah ada, termasuk aspek spesifikasi, desain, dan implementasi. Melalui proses implementasi ulang, peneliti dapat mengevaluasi apakah sistem memerlukan peningkatan dalam hal fungsionalitas dan kinerja, atau cukup mempertahankan fitur yang ada sambil menambahkan komponen baru yang lebih berdampak dibandingkan versi sebelumnya. Untuk mencapai tujuan tersebut, *reengineering* mengikuti tahapan-tahapan tertentu dalam prosesnya seperti pada Gambar 2.1, yang meliputi *inventory analysis* untuk identifikasi komponen sistem, *document restructuring* untuk pembaruan dokumentasi, *reverse engineering* untuk memahami struktur dan perilaku sistem yang ada, *code refactoring* guna meningkatkan kualitas internal kode, *data restructuring* untuk penyesuaian struktur data, serta *forward engineering*

sebagai tahapan pembangunan kembali sistem berdasarkan desain yang telah direkayasa ulang [14] [15].



**Gambar 2. 1** Model Proses Software Reengineering

Untuk membangun kembali sistem perangkat lunak secara efektif, proses *software reengineering* terdiri dari enam tahapan utama. Setiap tahapan memiliki peran penting dalam memastikan bahwa sistem lama dapat ditransformasi menjadi sistem baru yang lebih berkualitas, dan sesuai kebutuhan terkini. Adapun tahapan-tahapan tersebut adalah sebagai berikut:

### **1. Analisis Inventaris (*Inventory Analysis*)**

Tahap awal dalam proses ini adalah Analisis Inventaris, yang berfokus pada pengumpulan informasi mendetail mengenai sistem yang akan dikembangkan. Informasi yang dikaji mencakup elemen-elemen seperti nama sistem dan versi saat ini, *platform* perangkat lunak dan perangkat keras yang digunakan, modul dan komponen yang ada beserta relasinya, pihak-pihak yang terlibat dalam pengembangan, pemeliharaan, dan penggunaan sistem, serta ketergantungan sistem terhadap teknologi tertentu. Dengan analisis yang menyeluruh, bagian-bagian yang memerlukan rekayasa ulang dapat diidentifikasi dengan lebih mudah, Tujuan dari analisis ini adalah untuk menentukan bagian-bagian mana yang layak untuk dipertahankan, dihapus, atau ditingkatkan. Hasil dari tahap ini akan menjadi dasar dalam menentukan ruang lingkup dan prioritas dari proses *reengineering*.

### **2. Dokumentasi Restrukturisasi (*Document Restructuring*)**

Pada tahap ini, fokus diarahkan pada evaluasi dan perbaikan dokumentasi sistem.

Sistem *legacy* sering kali memiliki dokumentasi yang tidak lengkap, tidak akurat, atau tidak sesuai dengan implementasi sebenarnya. Oleh karena itu, dokumentasi yang ada harus diperbarui atau bahkan dibuat ulang. Tahapan ini bertujuan untuk memperbaiki serta meningkatkan dokumentasi sistem yang akan mengalami rekayasa ulang. Aktivitas yang dilakukan antara lain:

- a. Meninjau dokumentasi yang tersedia seperti spesifikasi kebutuhan, diagram desain, dan manual pengguna.
- b. Membandingkan dokumentasi dengan sistem aktual untuk menemukan ketidaksesuaian.
- c. Melakukan restrukturisasi dokumentasi agar sesuai dengan struktur dan alur sistem yang baru.
- d. Menerapkan standar dokumentasi modern seperti UML untuk diagram sistem dan BPMN untuk alur proses bisnis.

Langkah ini penting untuk memastikan bahwa dokumentasi dapat menjadi acuan dalam pengembangan sistem yang telah direkayasa ulang. Dokumentasi dari sistem sebelumnya dapat direstrukturisasi berdasarkan kebutuhan penelitian sistem yang baru.

### **3. Rekayasa Balik (*Reverse Engineering*)**

*Reverse engineering* dilakukan untuk memahami kembali struktur dan perilaku sistem yang sudah ada, terutama ketika dokumentasi tidak tersedia atau tidak dapat diandalkan. Proses ini bertujuan untuk mengekstrak desain sistem dari kode sumber dan artefak lainnya. Sebelum memasuki tahap rekayasa maju (*forward engineering*), dilakukan proses rekayasa balik. Tujuan utama dari langkah ini adalah memahami kembali sistem yang sudah ada dengan mengidentifikasi komponen serta hubungan di antara komponen tersebut. Beberapa alasan mengapa *reverse engineering* diperlukan antara lain:

- a. Tidak adanya dokumentasi desain atau spesifikasi yang lengkap.
- b. Dokumentasi yang usang, tidak sesuai, atau bahkan hilang.
- c. Kompleksitas sistem yang terus meningkat.
- d. Struktur kode yang tidak tertata dengan baik.
- e. Migrasi kode ke bahasa pemrograman yang berbeda.

Aktivitas yang dilakukan dalam tahapan ini meliputi:

Analisis struktur kode dan logika kontrol.

- a. Identifikasi *class*, modul, dan fungsionalitas utama.
- b. Pembuatan ulang diagram desain seperti *class diagram*, *sequence diagram*, dan *use case diagram*.
- c. Penggunaan alat bantu seperti *static code analyzer*, *visual UML extractor*, dan *dependency graph*.

Tahap ini memberikan wawasan yang sangat penting bagi tim pengembang untuk melanjutkan ke proses perbaikan dan peningkatan sistem.

#### **4. Refaktorisasi Kode (*Code Refactoring*)**

Refaktorisasi kode dilakukan untuk menata ulang struktur kode sumber tanpa mengubah fungsionalitasnya. Langkah ini diawali dengan memahami fungsi-fungsi yang terdapat dalam kode sistem sebelumnya. Setelah pemahaman diperoleh, perubahan dilakukan untuk menyederhanakan struktur kode, meningkatkan efisiensi pemeliharaan, serta mempermudah penelitian di masa mendatang.

Kegiatan yang umum dilakukan dalam refaktorisasi antara lain :

- a. Penggabungan fungsi-fungsi duplikat atau serupa.
- b. Pemisahan fungsi-fungsi besar menjadi fungsi-fungsi kecil yang lebih spesifik.
- c. Penghilangan kode mati (*dead code*) atau tidak digunakan lagi.
- d. Penerapan prinsip desain perangkat lunak seperti SOLID dan DRY (*Don't Repeat Yourself*).
- e. Modularisasi dan enkapsulasi fungsi atau kelas untuk mengurangi ketergantungan.

Metode evaluasi *refactoring* dapat dilakukan menggunakan metrik desain seperti:

- a. WMC (*Weighted Methods per Class*) - menunjukkan jumlah dan kompleksitas metode dalam suatu kelas.
- b. CBO (*Coupling Between Objects*) - menunjukkan tingkat ketergantungan antar kelas.

- c. LCOM (*Lack of Cohesion in Methods*) - mengukur sejauh mana metode dalam kelas berhubungan erat.

*Refactoring* yang baik dapat menghasilkan struktur kode yang lebih bersih dan siap untuk dikembangkan lebih lanjut.

## 5. Data Restrukturisasi (*Data Restructuring*)

Restrukturisasi data merupakan proses analisis dan modifikasi struktur data dalam sistem agar lebih efisien dan penting untuk menyesuaikan struktur data sistem agar optimal dengan desain dan kebutuhan baru. Basis data pada sistem *legacy* sering kali tidak lagi relevan atau tidak sesuai dengan standar modern. Dalam beberapa kasus, tahap ini diawali dengan teknik *Reverse Engineering* untuk mengidentifikasi elemen data seperti objek dan atributnya. Setelah itu, struktur data dievaluasi serta dioptimalkan guna meningkatkan kualitas sistem di masa depan.

Langkah - langkah dalam tahapan ini meliputi :

- a. Analisis struktur tabel, relasi, indeks, dan dependensi data.
- b. Normalisasi atau denormalisasi tabel untuk meningkatkan efisiensi.
- c. Identifikasi elemen data, termasuk objek, atribut, dan tipe data.
- d. Penerapan skema data baru berdasarkan kebutuhan yang diperoleh dari tahap *reverse engineering*.
- e. Migrasi data dari struktur lama ke struktur baru, dengan tetap menjaga integritas dan konsistensi data.

Data *restructuring* juga dapat melibatkan integrasi dengan sistem eksternal, peningkatan keamanan data, dan penerapan enkripsi jika dibutuhkan, misalnya untuk data sensitif seperti identitas pengguna atau kata sandi.

## 6. Rekayasa Maju (*Forward Engineering*)

Tahap akhir dalam proses *reengineering* adalah rekayasa maju, yaitu proses membangun sistem baru berdasarkan hasil dari tahapan-tahapan sebelumnya. Ini mencakup implementasi, integrasi, serta validasi sistem baru yang bertujuan untuk meningkatkan kualitas sistem secara keseluruhan.

Aktivitas utama pada tahap ini :

- a. Penerjemahan ulang artefak desain ke dalam implementasi aktual



menggunakan bahasa pemrograman *modern*.

- b. Pengembangan modul-modul baru dan peningkatan modul lama berdasarkan kebutuhan pengguna.
- c. Integrasi modul dengan sistem lain (jika ada), misalnya layanan web atau sistem ERP.
- d. Pengujian unit, integrasi, dan sistem secara menyeluruh untuk memastikan kesesuaian sistem terhadap spesifikasi.
- e. Penyusunan dokumentasi akhir serta pelatihan pengguna terhadap sistem baru.

Pada tahap ini, tidak hanya fungsi-fungsi yang lama diperbarui, tetapi juga fitur-fitur baru ditambahkan agar sistem menjadi lebih optimal dan relevan dengan kebutuhan bisnis yang berkembang. Tahap ini merupakan hasil akhir dari keseluruhan proses reengineering yang bertujuan untuk memberikan solusi perangkat lunak yang lebih berkualitas, berkelanjutan, dan mudah dikembangkan ke depan.

## **2.2 Aplikasi berbasis *Mobile***

Aplikasi *mobile* adalah perangkat lunak yang berjalan pada perangkat *mobile* seperti *smartphone* atau tablet PC. kasi ini dapat diunduh oleh pengguna melalui internet dan memiliki fungsi tertentu yang menambah fungsionalitas perangkat *mobile* itu sendiri [16].

Dalam kontlayanan pemerintahan, aplikasi *mobile* digunakan untuk meningkatkan interaksi dan efisiensi pelayanan publik. Studi yang dilakukan oleh Zhu dan Hou pada tahun 2021 [17] menunjukkan bahwa aplikasi *mobile* dalam layanan pemerintahan dapat meningkatkan kredibilitas pemerintah dan tata kelola dengan menyediakan *platform* interaktif yang lebih baik bagi warga negara. Dengan meningkatnya jumlah pengguna internet *mobile*, aplikasi pemerintahan menjadi sarana utama bagi masyarakat untuk mengakses layanan publik secara lebih mudah dan cepat.

Selain dalam sektor pemerintahan, aplikasi berbasis *mobile* juga menjadi bagian integral dalam dunia bisnis modern. Perkembangan teknologi *mobile* memungkinkan aplikasi untuk memberikan berbagai manfaat yang tidak hanya

meningkatkan efisiensi bisnis, tetapi juga memperbaiki pengalaman pengguna secara keseluruhan. Beberapa keunggulan utama dari aplikasi *mobile* meliputi aksesibilitas tinggi yang memungkinkan pengguna mengakses aplikasi kapan saja dan di mana saja, serta efisiensi operasional yang membantu perusahaan mengotomatiskan proses bisnis, mengurangi biaya operasional, dan meningkatkan produktivitas karyawan. Selain itu, aplikasi *mobile* memungkinkan komunikasi yang lebih cepat melalui notifikasi *push*, meningkatkan kepuasan pelanggan, serta memanfaatkan fitur perangkat seperti GPS, kamera, dan sensor biometrik untuk pengalaman yang lebih interaktif dan aman. Keunggulan lainnya adalah konektivitas dan integrasi dengan berbagai *platform* dan layanan *cloud*, sehingga pengguna dapat mengakses data dengan lebih mudah di berbagai perangkat. Dengan berbagai manfaat tersebut, aplikasi berbasis *mobile* terus berkembang dan menjadi solusi utama dalam berbagai bidang industri, memberikan dampak positif bagi efisiensi operasional serta membuka peluang bagi bisnis dan pemerintahan untuk beradaptasi dengan kebutuhan pasar yang dinamis.

### 2.3 Agile Scrum

Metode *Agile* merupakan pendekatan pengembangan perangkat lunak yang berfokus pada pengiriman perangkat lunak secara iteratif dan inkremental, di mana kerja sama tim, keterlibatan pengguna, dan kemampuan beradaptasi terhadap perubahan menjadi hal utama [18]. *Agile* memungkinkan tim untuk merespons perubahan dengan lebih cepat dan fleksibel, serta menghasilkan perangkat lunak bernilai tinggi secara bertahap.

*Scrum* adalah salah satu kerangka kerja dalam metode *Agile* yang banyak digunakan dalam praktik industri. *Scrum* menyediakan struktur kerja yang terdiri atas peran (*roles*), artefak (*artifacts*), dan serangkaian kegiatan (*events*) yang terorganisasi dalam satuan waktu tertentu yang disebut *sprint* [19]. Berikut adalah tahapan - tahapan utama dalam metode Scrum:

#### 1. Perumusan *Product Backlog*

*Product Backlog* merupakan daftar kebutuhan produk (*product requirements*) yang disusun dan dikelola oleh *Product Owner*. Daftar ini bersifat dinamis dan dapat mengalami perubahan sesuai dengan

perkembangan produk maupun kebutuhan pengguna. Setiap item dalam *backlog* dapat berupa fitur fungsional, kebutuhan teknis, perbaikan, atau dokumentasi. Tugas utama *Product Owner* adalah menetapkan prioritas *backlog* berdasarkan nilai bisnis dan urgensi pengguna.

## 2. *Sprint Planning*

*Sprint Planning* adalah kegiatan perencanaan yang dilakukan pada awal setiap iterasi (*sprint*), dengan melibatkan seluruh anggota tim Scrum, termasuk *Scrum Master*, *Product Owner*, dan *Development Team*. Tujuan dari sesi ini adalah untuk menentukan pekerjaan yang akan diselesaikan selama satu *sprint*, menetapkan *sprint goal*, serta menyusun *sprint backlog* yang berisi daftar tugas yang akan dikerjakan.

## 3. Eksekusi *Sprint*

*Sprint* adalah siklus iteratif dengan durasi tetap, umumnya antara satu hingga empat minggu, di mana tim pengembang bekerja menyelesaikan item yang telah direncanakan dalam *sprint backlog*. Selama pelaksanaan *sprint*, perubahan ruang lingkup dihindari agar fokus tim tidak terganggu. Kegiatan pengembangan dilakukan secara intensif dan terstruktur untuk mencapai tujuan *sprint* yang telah ditetapkan.

## 4. *Daily Scrum*

Daily Scrum merupakan pertemuan singkat harian yang dihadiri oleh tim pengembang (*Development Team*). Pertemuan ini bertujuan untuk mengevaluasi progres harian, merencanakan aktivitas berikutnya, serta mengidentifikasi kendala yang mungkin menghambat pencapaian *sprint goal*. Diskusi difokuskan pada tiga pertanyaan utama:

- a. Apa yang telah diselesaikan sejak pertemuan sebelumnya.
- b. Apa yang akan dikerjakan hari ini.
- c. Apakah ada hambatan yang dihadapi.

## 5. *Sprint Review*

*Sprint Review* adalah sesi evaluasi yang dilakukan pada akhir setiap *sprint* untuk meninjau hasil kerja tim. Dalam kegiatan ini, tim mempresentasikan *increment* atau fitur produk yang telah selesai kepada para pemangku kepentingan (*stakeholder*), dan menerima umpan balik yang akan digunakan

untuk penyempurnaan produk. Masukan yang diperoleh dapat menghasilkan perubahan atau penambahan pada *product backlog*.

#### 6. *Sprint Retrospective*

*Sprint Retrospective* adalah kegiatan refleksi internal yang dilakukan oleh tim Scrum setelah *sprint review*. Tujuan utamanya adalah untuk mengevaluasi proses kerja selama *sprint*, mengidentifikasi hal-hal yang berjalan baik maupun hambatan yang muncul, serta menyusun rencana perbaikan untuk sprint selanjutnya. Dengan adanya kegiatan ini, tim dapat terus meningkatkan kualitas kolaborasi kerja.

Scrum memiliki sejumlah keunggulan yang telah terbukti secara praktis dalam pengembangan perangkat lunak. Salah satu keunggulan utamanya adalah kemampuannya dalam memfasilitasi pengiriman nilai bisnis secara bertahap dan berkelanjutan. Pendekatan iteratif yang digunakan memungkinkan tim untuk menyesuaikan prioritas dan merespons perubahan kebutuhan pengguna dengan lebih fleksibel. Selain itu, Scrum mendorong kolaborasi tim yang efektif melalui komunikasi rutin, seperti dalam kegiatan *daily scrum*, yang membantu meningkatkan transparansi dan keterlibatan setiap anggota tim. Melalui proses inspeksi dan adaptasi yang dilakukan secara berkala, kualitas perangkat lunak dapat terus ditingkatkan seiring berjalannya waktu. Scrum juga dianggap mampu memberikan budaya kerja yang adaptif, terbuka, dan berorientasi pada hasil, sehingga menjadikannya sebagai pendekatan yang relevan dalam lingkungan proyek yang dinamis dan kompleks.

## 2.4 Flutter

*Flutter* adalah *framework* yang dikembangkan oleh Google untuk mempermudah pembuatan aplikasi lintas *platform* dengan satu kode sumber. Dengan *flutter*, peneliti dapat membangun aplikasi yang dapat berjalan di sistem operasi android maupun IOS tanpa perlu menulis kode secara terpisah. Menurut S. A. Bhagat [20], *flutter* menjadi solusi yang banyak digunakan karena kemampuannya dalam

meningkatkan efisiensi waktu dan biaya penelitian serta integrasi yang lebih baik dengan berbagai layanan.

*Flutter* menggunakan bahasa pemrograman *dart* dan menawarkan sejumlah fitur unggulan yang membantu dalam penelitian aplikasi *mobile*. Beberapa fitur utama dari *flutter* antara lain adalah *hot reload*, memungkinkan peneliti melihat perubahan kode secara instan tanpa harus memulai ulang aplikasi, kode tunggal untuk berbagai *platform*, mengurangi kebutuhan pengkodean ulang untuk sistem yang berbeda, struktur berbasis *widget*, yang membuat antarmuka lebih fleksibel dan dapat disesuaikan dengan mudah, serta kinerja optimal, karena kode *flutter* dikompilasi langsung ke dalam bahasa mesin, menghindari hambatan yang terjadi pada *framework* lain.

Menurut Patil dalam journal internasional [21] mengatakan bahwa penelitian kode di *Flutter* lebih cepat dan efisien dibandingkan dengan *platform* lain, serta mendukung penelitian lintas *platform*, yang menyederhanakan tugas peneliti. Kinerja aplikasi yang dikembangkan dengan *Flutter* juga mendekati aplikasi *native*, sehingga pengguna sering kali tidak dapat membedakan antara keduanya. Selain itu, *Flutter* memiliki kemampuan untuk mengembangkan aplikasi *web* tanpa perlu mengubah kode, yang menambah fleksibilitas dalam penelitian. Antarmuka pengguna yang ramah dan fitur lintas *platform* juga menjadi nilai tambah, memungkinkan peneliti untuk menciptakan aplikasi untuk berbagai sistem operasi secara bersamaan, sehingga mengurangi biaya dan usaha.

## 2.5 Aplikasi Sejenis

Dalam upaya meningkatkan pengelolaan penyewaan aset milik pemerintah secara digital, berbagai daerah di Indonesia telah mengembangkan sistem informasi sejenis dengan tujuan memberikan transparansi, dan kemudahan akses bagi masyarakat. Beberapa aplikasi tersebut, meskipun berbeda dalam pendekatan dan fitur, memiliki kesamaan fungsi utama dengan Tapatupa, yaitu sebagai *platform* untuk penyewaan aset tanah milik pemerintah daerah. Studi terhadap aplikasi sejenis ini penting untuk dijadikan rujukan dalam proses *reengineering*, agar pengembangan aplikasi Tapatupa dapat mengadopsi praktik baik dan menghindari kekurangan yang ditemukan pada sistem lain. Adapun aplikasi yang dijadikan

referensi dalam penelitian ini meliputi ASET Bali, SIAPSEWA Jawa Timur, Amanah Kota Bandung, dan SIWA Bali.


Adapun pengamatan terhadap aplikasi sejenis dilakukan terhadap 3 sistem yang dipakai oleh daerah masing masing, yaitu *website* Siap Sewa dari Jawa Timur, *website* Amanah dari Bandung, dan SIWA App dari Bali dengan hasil sebagai berikut.

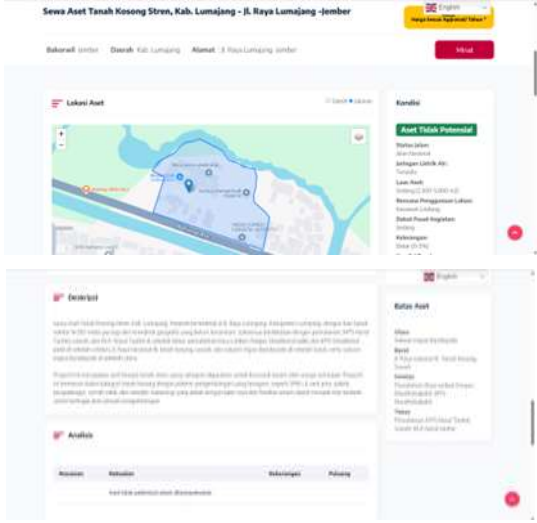

#### 1. *Website* SIAPSEWA Aset Jawa Timur

*Platform* digital yang dikelola oleh Pemerintah Provinsi Jawa Timur untuk memberikan layanan penyewaan Barang Milik Daerah (BMD) kepada masyarakat atau investor. *Platform* ini bertujuan untuk meningkatkan pemanfaatan aset pemerintah dengan skema penyewaan yang transparan dan legal. *Website* ini memberikan informasi terkait:

- a. Aset yang disewakan, termasuk lokasi dan deskripsi properti.
- b. Kategori aset, seperti perkantoran, tanah kosong, penginapan, *open space*, *outdoor*, dan sewa ruangan.
- c. Harga sewa, yang ditampilkan berdasarkan hasil *appraisal* per tahun.
- d. Alur dan Informasi penyewaan, sehingga pengguna dapat memahami prosedur dalam menyewa aset.

**Tabel 2. 1 Analisis Website SiapSEWA**

No	Tampilan Aplikasi	Keterangan
1	 <p>The screenshot shows the homepage of the SIAPSEWA website. At the top, it says 'Kategori BMD Aset Jawa Timur' and 'tentang informasi prosedur transaksi kategori'. Below this, there are six blue hexagonal buttons with icons and text: 'Tanah Kosong' (14 aset), 'Perkantoran' (17 aset), 'Penginapan' (6 aset), 'Open Space' (9 aset), 'Outdoor Space' (2 aset), and 'Perdagangan' (1 aset). At the bottom, it says 'BMD Aset Jawa Timur tiap Kabupaten/ Kota' and 'tentang informasi prosedur transaksi kategori'. There are also 'View More' buttons in red.</p> <p><b>Gambar 2. 2 Tampilan Beranda Website SiapSEWA</b></p>	Tampilan halaman utama <i>website</i> yang menampilkan setiap kategori BMD asset Jawa Timur.

No	Tampilan Aplikasi	Keterangan
2	 <p><b>Gambar 2. 3 Gambar Tampilan Detail Website SiapSEWA</b></p>	<p>Tampilan disamping menampilkan gambar aset, Alamat, data video aset, deskripsi property, dan analisis terhadap aset yang akan dipinjamkan.</p>
3	 <p><b>Gambar 2. 4 Tampilan Hak Aset Website SiapSEWA</b></p>	<p>Tampilan disamping menunjukkan tombol minat bagi pengguna yang ingin menyewa tempat tersebut.</p>



## 2. Website Amanah Bandung

Amanah.bandung.go.id adalah *platform* yang dikelola oleh Pemerintah Kota Bandung untuk pengelolaan aset daerah secara transparan dan efisien. Aplikasi ini dirancang untuk memfasilitasi inventarisasi aset, termasuk tanah, bangunan, dan fasilitas lainnya yang dimiliki pemerintah daerah. Fitur utamanya mencakup pencatatan aset yang terstruktur, visualisasi data berbasis peta, serta sistem pengajuan dan manajemen penyewaan aset. Selain itu, platform ini mendukung transparansi publik dengan



menyediakan informasi terkait status aset yang dapat diakses oleh masyarakat umum.

**Tabel 2. 2 Analisis Website Amanah**


No	Tampilan Aplikasi	Keterangan
1	 <p><b>Gambar 2. 5 Tampilan Beranda Website Amanah</b></p>	<p>Tampilan halaman utama ini menampilkan menu fitur pembayaran, pelaporan, pelayanan, dan peta tematik. Pembayaran membantu pengguna untuk mengelola pembayaran sewa tanah, termasuk informasi tagihan, status pembayaran, dan metode yang disediakan. Pada menu pelaporan, pengguna dapat melaporkan aktivitas atau masalah yang berkaitan dengan sewa aset. Pelayanan berisi pengajuan perpanjangan kontrak, informasi terkait aset.</p>
2	 <p><b>Gambar 2. 6 Tampilan Pembayaran</b></p>	<p>Pada tampilan di samping, secara detail fitur pembayaran berisi informasi penyewa, lokasi, kategori, dan daftar tagihan. Menu pembayaran menyediakan informasi no. kontrak sewa aset yang akan disewakan.</p>


### 3. Aplikasi SIWA Bali

*Platform* atau aplikasi yang dikembangkan oleh Pemerintah Provinsi Bali untuk mengelola data aset daerah. Tampilan beranda pada aplikasi SIWA dirancang untuk memudahkan pengguna dalam menemukan dan menyewa aset secara digital, pada bagian Rekomendasi Aset, pengguna dapat melihat aset yang direkomendasikan, seperti tanah area maupun bangunan. Di bagian Aset Bangunan, pengguna dapat menemukan informasi spesifik tentang aset tertentu, seperti Harga dari aset tersebut.

**Tabel 2. 3** Analisis Aplikasi SIWA

No	Tampilan Aplikasi	Keterangan
1	 <p style="text-align: center;"><b>Gambar 2. 7</b> Tampilan Beranda</p>	<p>Pada tampilan disamping terdapat halaman utama pada aplikasi SIWA. Informasi utama bagian “rekomendasi aset” menampilkan informasi harga dan ketersediaan tanah dengan visual yang mendukung, seperti gambar dan label harga estimasi. Fitur pencarian yang mudah pada kolom pencarian di bagian atas memungkinkan pengguna mencari aset secara langsung. Pada fitur tipe aset memberikan pengelompokan yang jelas antara aset tanah dan aset bangunan yang membantu pengguna memahami pilihan yang tersedia.</p>

No	Tampilan Aplikasi	Keterangan
2	 <p><b>Gambar 2. 8</b> Tampilan Halaman Pengajuan</p>	<p>Halaman disamping menampilkan pengajuan sewa tanah dengan mengisi formulir pengajuan yang berisi no.telepon, umur, pekerjaan, Alamat, lama menyewa, dan file sketsa aset. Setelah pengguna mengisi formulir pengajuan, maka pengajuan dapat disimpan dan masuk dalam fitur keranjang.</p>

No	Tampilan Aplikasi	Keterangan
3	 <p style="text-align: center;"><b>Gambar 2. 9</b> Tampilan Halaman Keranjang Pengguna</p>	<p>Pada tampilan disamping, fitur keranjang menampilkan aset yang akan disewakan sebelumnya. Setelah itu pengguna dapat mengklik <i>button</i> sewa.</p>

## 2.6 Kuesioner

Kuesioner merupakan salah satu alat evaluasi yang paling umum digunakan untuk mengukur kepuasan pengguna terhadap suatu sistem atau layanan. Dalam konteks rekayasa ulang sistem berbasis teknologi informasi, kuesioner berfungsi untuk mengumpulkan data secara sistematis dan terstruktur mengenai persepsi, pengalaman, dan tingkat kenyamanan pengguna dalam menggunakan aplikasi. Dalam penelitian ini, kuesioner digunakan setelah proses *reengineering* aplikasi Tapatupa selesai dilakukan. Tujuan penyebaran kuesioner pada tahap ini adalah untuk mengevaluasi hasil dari perancangan ulang antarmuka dan fungsionalitas sistem berdasarkan persepsi dan pengalaman pengguna akhir. Menurut Carpino, C., et al. [22], meskipun awalnya digunakan dalam konteks pengumpulan data energi bangunan tempat tinggal, kuesioner terbukti sebagai instrumen yang fleksibel untuk

mengukur persepsi subjektif pengguna, asalkan disusun dengan metode dan terminologi yang terstandar agar hasilnya valid dan reliabel.

Dalam penggunaannya untuk mengukur kepuasan pengguna, kuesioner umumnya memuat serangkaian pernyataan yang berkaitan dengan aspek kegunaan (*usability*), kemudahan penggunaan, kenyamanan navigasi, dan kualitas informasi dalam sistem. Pernyataan-pernyataan tersebut dijawab oleh responden menggunakan skala tertentu, seperti skala Likert, yang memungkinkan analisis kuantitatif terhadap tingkat kepuasan yang dirasakan.

Seperti dijelaskan oleh Hasibuan [23], bentuk pertanyaan dalam kuesioner dapat berupa pertanyaan tertutup, di mana responden memilih jawaban yang telah disediakan (misalnya, sangat puas hingga sangat tidak puas), pertanyaan terbuka yang memungkinkan responden memberi masukan bebas, atau pertanyaan campuran. Dalam evaluasi kepuasan pengguna, pertanyaan tertutup dengan skala Likert lebih banyak digunakan karena memungkinkan pengolahan statistik yang lebih terukur.

### **2.6.1 Uji Kelayakan Kuesioner**

Penggunaan kuesioner sebagai alat pengumpulan data sangat umum dalam penelitian sosial, kesehatan, dan epidemiologi, tak terkecuali pada proses penelitian untuk penelitian perangkat lunak. Guna memastikan data yang nantinya dikumpulkan dapat dinyatakan valid dan dapat diandalkan, penting adanya untuk melakukan uji kelayakan pada kuesioner. Sebelum disebarkan ke responden, kuesioner terlebih dahulu melalui proses uji kelayakan untuk memastikan bahwa item-item pertanyaan yang disusun dapat mengukur apa yang memang ingin diukur. Dua uji utama yang dilakukan adalah uji validitas dan uji reliabilitas.

#### **2.6.1.1 Uji Validitas Kuesioner**

Uji validitas merupakan suatu prosedur statistik yang digunakan untuk mengukur sejauh mana suatu instrumen pengukuran (kuesioner) mampu mengukur apa yang seharusnya diukur. Dalam penelitian ini, validasi kuesioner diuji dengan menggunakan teknik korelasi *Pearson Product Moment*, yang mengukur hubungan antara skor tiap item dengan skor total kuesioner.

Pengujian validitas dilakukan dengan menghitung nilai korelasi ( $r$  hitung) antara masing-masing item pertanyaan dan total skor keseluruhan responden. Hasil perhitungan tersebut kemudian dibandingkan dengan nilai kritis dalam tabel *r Pearson* ( $r$  tabel) pada taraf signifikansi tertentu (misalnya  $\alpha = 0,05$ ) dan derajat kebebasan ( $df$ ) =  $n - 2$ , di mana  $n$  merupakan jumlah responden. Apabila nilai  $r$  hitung lebih besar dari  $r$  tabel, maka item dalam kuesioner dinyatakan valid, karena menunjukkan adanya hubungan yang signifikan antara item dengan keseluruhan konstruk pengukuran.

Sebaliknya, jika  $r$  hitung lebih kecil dari  $r$  tabel, maka item dinyatakan tidak valid dan dapat dipertimbangkan untuk dihapus atau direvisi.

Uji validitas dalam penelitian ini bertujuan untuk memastikan bahwa setiap item dalam kuesioner benar-benar merepresentasikan konstruk yang diukur, serta mampu memberikan hasil yang baik dalam pengumpulan data penelitian.

Validitas digunakan untuk memastikan bahwa kuesioner yang akan digunakan dalam penelitian benar-benar sah. Kuesioner dianggap valid jika dapat mengukur apa yang memang ingin diukur. Sebuah tes dikatakan memiliki validitas tinggi jika tes tersebut memberikan hasil yang akurat dan tepat. Sebaliknya, tes dengan validitas rendah menghasilkan data yang tidak relevan dengan tujuan pengukuran. Pada pengujian validitas kuesioner, terdapat dua jenis validitas yang perlu diperhatikan, yaitu validitas faktor dan validitas item.

- a. Validitas faktor diukur ketika item-item yang disusun melibatkan lebih dari satu faktor, di mana antara faktor satu dengan faktor lainnya terdapat kemiripan. Validitas faktor ini diukur dengan cara mengkorelasikan skor faktor (penjumlahan item dalam satu faktor) dengan skor total faktor (penjumlahan seluruh faktor).
- b. Validitas item menunjukkan adanya hubungan atau dukungan antara item dengan skor total (skor keseluruhan). Perhitungan validitas item dilakukan dengan cara mengkorelasikan skor item dengan skor total item. Jika terdapat lebih dari satu faktor, maka pengujian validitas item dilakukan dengan mengkorelasikan skor item dengan skor faktor, dan selanjutnya mengkorelasikan skor item dengan skor total faktor (hasil penjumlahan dari beberapa faktor).

Selanjutnya, angka korelasi yang diperoleh dibandingkan dengan angka kritis pada tabel korelasi nilai  $r$ . Tabel lebih besar sama dengan  $r$ . Hitung dengan taraf signifikansi 5%, maka pernyataan tersebut dianggap valid. Sebaliknya, jika nilai  $r$ .sTabel hitung berada di bawah angka negatif, maka pernyataan tersebut dianggap tidak valid.

#### 2.6.1.2. Uji Reliabilitas Kuesioner

Uji reliabilitas kuesioner adalah langkah penting dalam penelitian untuk memastikan bahwa alat ukur yang digunakan konsisten dan dapat diandalkan. Makna kata reliabilitas adalah ukuran konsistensi suatu instrumen penelitian, seperti kuesioner, dalam menghasilkan hasil yang sama ketika diulang dalam kondisi yang sama. Ini penting untuk memastikan bahwa data yang dikumpulkan dapat dipercaya dan digunakan untuk analisis lebih lanjut menurut Taherdoost, H. [24].

Dalam pelaksanaannya, terdapat beberapa metode dalam pengujian atas taraf reliabilitas pada suatu kuesioner, yakni :

##### 1. Cronbach's Alpha

Menurut Wojujutari, A., et al [25], Metode ini merupakan metode yang paling umum digunakan untuk mengukur reliabilitas internal dari kuesioner. Nilai *Cronbach's Alpha* yang lebih tinggi menunjukkan konsistensi internal yang lebih baik. Uji reliabilitas dinyatakan reliabel apabila *Cronbach's alpha*  $> .0.6$  atau *Cronbach's alpha*  $> r$  tabel.

##### 2. Test-Retest Reliability

Melalui Cabanas-Sánchez, V., et al [26], Dikatakan bahwa metode ini melibatkan pengujian instrumen yang sama pada dua waktu yang berbeda untuk menilai konsistensi hasil dari waktu ke waktu. Misalnya, dalam penelitian tentang *Youth Leisure-time Sedentary Behavior Questionnaire*, reliabilitas *test-retest* menunjukkan kesepakatan yang moderat hingga substansial.

##### 3. Intraclass Correlation Coefficient (ICC)

Dikutip melalui Kumlien, C., et al [27], metode ini digunakan untuk menilai

reliabilitas *test-retest*, seperti yang diterapkan pada *Vascular Quality of Life Questionnaire-6*, yang menunjukkan koefisien korelasi intrakelas yang dapat diterima.

Berdasarkan metode dalam pengujian atas taraf reliabilitas, peneliti menggunakan metode *Cronbach's Alpha* untuk mengetahui dan menilai taraf reliabilitas. Nilai reliabel dan nilai validitas diukur berdasarkan kategori skala likert yang akan dibagikan kedalam kuesioner.

Agar dapat mengetahui tinggi rendahnya reliabilitas instrumen dari suatu kuesioner dapat dikategorikan ke dalam Tabel 2.4.

**Tabel 2. 4** Interval Reliabilitas *Cronbach's Alpha*

Interval Reliabilitas	Kategori
$0,80 < \text{reliabel} \leq 1,00$	Reliabilitas Sangat Tinggi
$0,60 < \text{reliabel} \leq 0,80$	Reliabilitas Tinggi
$0,40 < \text{reliabel} \leq 0,60$	Reliabilitas Sedang
$0,20 < \text{reliabel} \leq 0,40$	Reliabilitas Rendah
$0,00 < \text{reliabel} \leq 0,20$	Tidak Reliabel

### 2.6.2 Evaluasi *Usability* menggunakan *Custom Usability Questionnaires*

Evaluasi usabilitas telah menjadi bagian penting dalam proses pengembangan perangkat lunak, terutama ketika tujuan utama sistem adalah memberikan pengalaman pengguna yang intuitif dan memuaskan. Salah satu pendekatan yang paling banyak digunakan untuk mengevaluasi aspek ini adalah penggunaan *custom usability questionnaires*. Teknik ini mengandalkan serangkaian pertanyaan terstruktur yang dirancang untuk mencerminkan pengalaman dan persepsi pengguna terhadap sistem, biasanya dalam format skala Likert.

Sebuah studi pemetaan sistematis yang dilakukan oleh Paz dan Pow-Sang menunjukkan bahwa metode survei atau kuesioner merupakan teknik evaluasi usabilitas yang paling banyak diterapkan dalam konteks rekayasa perangkat lunak, dengan proporsi mencapai 26,26% dari total metode yang ditinjau dalam 215 publikasi akademik yang relevan [28]. Hasil ini menunjukkan bahwa penggunaan kuesioner bukan hanya lazim, tetapi juga dinilai efektif karena mampu menjangkau



umpan balik langsung dari pengguna dalam jumlah besar, tanpa membutuhkan sumber daya evaluasi yang kompleks.

Keunggulan lain dari pendekatan ini adalah fleksibilitasnya. Kuesioner dapat disesuaikan dengan konteks penggunaan, jenis aplikasi, hingga karakteristik pengguna sasaran. Oleh karena itu, jenis kuesioner ini sering kali menjadi pilihan utama dalam proyek pengembangan berbasis pendekatan *user-centered design*, baik untuk aplikasi *mobile*, web, maupun sistem informatif lainnya. Dalam banyak kasus, kuesioner juga digunakan bersamaan dengan metode lain seperti *user testing* atau *heuristic evaluation* untuk memperoleh gambaran evaluasi yang lebih menyeluruh dan kaya konteks.

### 2.6.3 Metode Evaluasi Sistem Berdasarkan Perhitungan USE Questionnaire

Evaluasi *usability* terhadap aplikasi Tapatupa dilakukan melalui penyebaran kuesioner kepada pengguna setelah sistem selesai direkayasa ulang. Meskipun kuesioner yang digunakan disusun secara khusus atau *custom usability questionnaire* - berdasarkan indikator desain antarmuka pengguna dan fungsionalitas sistem - namun metode pengolahan datanya mengacu pada pendekatan perhitungan USE *Questionnaire*.

USE *Questionnaire* adalah metode evaluasi *usability* yang dikembangkan oleh Lund, dan mengukur persepsi pengguna terhadap sistem dari empat dimensi utama, yaitu *usefulness*, *ease of use*, *ease of learning*, dan *satisfaction*. Dalam penelitian ini, meskipun dimensi tersebut tidak diadopsi secara utuh dalam penyusunan item pertanyaan, namun pendekatan kuantitatifnya tetap diadaptasi untuk menghitung tingkat kepuasan pengguna secara menyeluruh.

Kuesioner disusun dalam format Likert 5 poin, dengan rentang nilai 1 (sangat tidak setuju) hingga 5 (sangat setuju). Setiap responden memberikan penilaian terhadap 10 pernyataan yang terdiri dari dua indikator: indikator UI dan indikator fitur. Hasil penilaian kemudian diolah menggunakan rumus perhitungan berikut, sebagaimana diadopsi dari penelitian-penelitian sebelumnya yang mengimplementasikan USE *Questionnaire* [29]:

$$K (\%) = \frac{\text{Jumlah Skor Aktual}}{\text{Jumlah Skor Maksimal}} \times 100\% \quad (2.1)$$

Keterangan:

1. K (%): Tingkat kepuasan pengguna dalam persen
2. Jumlah Skor Aktual: Total skor jawaban seluruh responden terhadap semua pernyataan
3. Jumlah Skor Maksimal: Skor maksimum, yaitu hasil dari  $s \times y \times d$ 
  - a. s: Skor maksimal skala (5)
  - b. y: Jumlah pernyataan
  - c. d: Jumlah responden

Nilai yang diperoleh kemudian diinterpretasikan ke dalam kategori berikut :

**Tabel 2. 5** Kategori Kelayakan Angka (%)

Rentang Nilai	Interpretasi Tingkat Kepuasan
0% - 20%	Sangat Kurang
21% - 40%	Kurang
41% - 60%	Cukup
61% - 80%	Puas
81% - 100%	Sangat Puas

#### 2.6.4 Skala Likert

Skala Likert merupakan salah satu instrumen pengukuran yang umum digunakan dalam penelitian kuantitatif, terutama dalam bidang ilmu sosial dan perilaku. Skala ini dirancang untuk mengukur sikap, persepsi, atau tingkat persetujuan responden terhadap suatu pernyataan dengan memberikan pilihan jawaban yang terstruktur secara ordinal. Skala ini pertama kali diperkenalkan oleh seorang psikolog asal Amerika Serikat, Rensis Likert, pada tahun 1932, sebagai alat untuk mengukur tingkat persetujuan atau ketidaksetujuan terhadap suatu pernyataan.

Skala Likert biasanya terdiri dari lima hingga tujuh poin jawaban, dengan kategori seperti: "Sangat Tidak Setuju", "Tidak Setuju", "Netral", "Setuju", dan "Sangat Setuju". Format ini memudahkan peneliti untuk mengkuantifikasi opini subjektif responden secara sistematis. Selain itu, skala ini juga fleksibel dan dapat dimodifikasi menjadi skala 4 poin (tanpa pilihan netral), 7 poin, atau bahkan 10 poin, tergantung pada kebutuhan dan konteks penelitian.

Meskipun penggunaannya sangat luas dalam berbagai disiplin ilmu seperti psikologi, pendidikan, manajemen, dan pemasaran, skala Likert juga menimbulkan diskusi akademik terkait validitas statistiknya, terutama dalam pemilihan jenis analisis data yang sesuai. Namun demikian, karena kemudahan dalam penyusunan, distribusi, serta interpretasi data, skala ini tetap menjadi salah satu alat ukur paling populer dalam penelitian yang melibatkan persepsi dan kepuasan pengguna.

Dengan menggunakan skala ini, responden akan menyatakan seberapa setuju atau tidak setuju mereka terhadap berbagai pernyataan yang terkait dengan objek atau kasus yang sedang diteliti, hal ini dapat dilihat dalam buku Sugiyono [30]. Penelitian ini menggunakan skala likert 5 dengan titik respon seperti yang ditunjukkan pada Tabel 2.6, guna untuk mengukur setiap pernyataan yang disajikan pada kuesioner.

**Tabel 2. 6** Kriteria Pengukuran Skala Likert

Singkatan	Kategori	Nilai
STS	Sangat Tidak Setuju	1
TS	Tidak Setuju	2
N	Netral	3
S	Setuju	4
SS	Sangat Setuju	5

Setiap pernyataan pada skala likert akan diberi nilai kriteria pengukuran skala likert. Jika mayoritas responden memilih sangat setuju, maka nilai akan mendekati angka 5. Sebaliknya, jika mayoritas responden memilih sangat tidak setuju, maka nilai ukur akan mendekati angka 1.

Berdasarkan rumus skala interval, dapat dibuat kriteria interpretasi skala likert berdasarkan nilai interval yaitu :

- 1,00 - 1,80 = Sangat Tidak Setuju / Sangat Tidak Puas
- 1,80 - 2,60 = Tidak Setuju / Tidak Puas
- 2,60 - 3,40 = Netral / Cukup

- 3,40 - 4,20 = Setuju / Baik  
 4,20 - 5,00 = Sangat Setuju / Sangat Baik

## 2.7 Stratified Sampling

*Stratified Sampling* atau pengambilan sampel berstrata merupakan salah satu metode probabilistik dalam teknik sampling yang digunakan untuk menjamin keterwakilan subkelompok dalam populasi secara proporsional. Dalam penelitian ini, metode ini digunakan untuk menyebarkan kuesioner evaluasi kepuasan pengguna setelah proses *reengineering* aplikasi Tapatupa selesai dilaksanakan. Tujuan utama dari penggunaan teknik ini adalah memastikan bahwa seluruh lapisan pengguna yang memiliki karakteristik berbeda dapat terlibat secara adil dalam proses evaluasi sistem.

Menurut penelitian [31], *stratified sampling* sangat tepat diterapkan ketika populasi memiliki tingkat heterogenitas yang tinggi, karena dapat meningkatkan akurasi dan validitas hasil evaluasi. Dalam konteks aplikasi Tapatupa, pengguna berasal dari tiga kecamatan di Kabupaten Tapanuli Utara, yaitu Siborongborong, Muara, dan Tarutung. Masing-masing wilayah ini memiliki karakteristik yang berbeda dalam hal usia, akses digital, dan intensitas penggunaan layanan sewa aset. Oleh karena itu, responden dibagi ke dalam strata berdasarkan wilayah tempat tinggal, dengan pengambilan sampel dilakukan secara proporsional untuk menjaga representasi yang adil.

Penerapan *stratified sampling* dalam penyebaran kuesioner ini juga sejalan dengan pendekatan kuantitatif yang digunakan dalam penelitian. Teknik ini memastikan bahwa evaluasi tingkat kepuasan pengguna terhadap aplikasi yang telah di *reengineering* tidak hanya dilakukan secara acak, tetapi mempertimbangkan representasi pengguna secara menyeluruh. Hasil evaluasi dari tiap strata kemudian dapat dianalisis untuk memperoleh pemahaman yang lebih mendalam mengenai fungsionalitas dan antarmuka setelah *reengineering*.

Dengan demikian, penggunaan metode stratified sampling pada tahap evaluasi sistem melalui kuesioner tidak hanya memperkuat validitas hasil penelitian, tetapi juga mendukung prinsip *User-Centered Design* dengan melibatkan beragam kelompok pengguna dalam proses penilaian akhir aplikasi.

## 2.8 Metode Slovin

Metode *Slovin* merupakan salah satu teknik yang digunakan dalam menentukan ukuran sampel dari suatu populasi yang diketahui, khususnya ketika informasi mengenai varians populasi tidak tersedia. Rumus *Slovin* digunakan untuk memperkirakan jumlah responden minimum yang diperlukan dalam suatu penelitian agar hasilnya tetap representatif dengan tingkat kesalahan tertentu.

Rumus *Slovin* dinyatakan sebagai berikut:

$$n = \frac{N}{1 + N e^2} \quad (2.2)$$

Keterangan:

$n$  = jumlah sampel,

$N$  = jumlah populasi,

$e$  = batas toleransi kesalahan (*margin of error*)

*Margin of error* ( $e$ ) adalah nilai yang menunjukkan batas toleransi kesalahan antara hasil dari sampel dengan parameter sebenarnya dari populasi. Dalam konteks penelitian kuantitatif, nilai ini menyatakan seberapa jauh hasil pengamatan dari sampel dapat menyimpang dari keadaan sebenarnya dalam populasi. Semakin kecil nilai *margin of error* yang ditetapkan, semakin tinggi tingkat akurasi dan semakin besar ukuran sampel yang dibutuhkan. Sebaliknya, semakin besar *margin of error*, maka jumlah sampel dapat lebih kecil, tetapi dengan konsekuensi akurasi hasil yang lebih rendah.

Dalam penelitian ini, metode *Slovin* diterapkan untuk menentukan jumlah responden yang digunakan dalam pengujian aplikasi Tapatupa. Dengan mempertimbangkan keterbatasan waktu dan sumber daya yang tersedia, maka ditetapkan tingkat kesalahan sebesar 10% atau  $e = 0,10$ . Nilai ini dipilih untuk menjaga pelaksanaan pengumpulan data, sekaligus tetap mempertahankan tingkat representativitas data yang memadai terhadap populasi yang diteliti.

Meskipun tingkat kesalahan 10% lebih longgar dibandingkan dengan tingkat konvensional 5%, penggunaan nilai ini masih diterima secara metodologis dalam

kondisi di mana penelitian bersifat eksploratif, atau saat peneliti menghadapi kendala teknis seperti waktu, biaya, atau jumlah enumerator yang terbatas [32].

Metode *Slovin* menjadi solusi praktis dalam pengambilan sampel yang terstruktur namun fleksibel, terutama dalam konteks pengembangan aplikasi yang melibatkan partisipasi masyarakat seperti aplikasi Tapatupa. Dengan memasukkan nilai  $N$  sebesar 526 (jumlah populasi pada 3 kecamatan yang sudah memiliki NPWP atau terdaftar sebagai penyewa wajib retribusi), maka jumlah sampel yang diperoleh dihitung sebagai berikut:

$$\begin{aligned} n &= \frac{N}{1 + N e^2} \\ n &= \frac{526}{1 + 526 \times (0,1)^2} \\ &= \frac{526}{1 + 5,26} \\ &= \frac{526}{6,26} \approx 83,98 \end{aligned}$$

Hasil perhitungan tersebut kemudian dibulatkan menjadi 84 responden. Oleh karena itu, metode *Slovin* dinilai tepat dalam membantu proses pengambilan keputusan terkait jumlah responden secara terstruktur khususnya dalam konteks pengembangan aplikasi yang membutuhkan keterlibatan pengguna secara langsung seperti aplikasi Tapatupa.

## 2.9 Responden

Penelitian ini menggunakan pendekatan kuantitatif-kualitatif dengan memerlukan responden untuk menganalisis 9 pernyataan menggunakan skala likert dan 1 pertanyaan kualitatif terbuka. Landasan teoretis *The central limit* dalam penelitian Hogg dan Tanis [33] menunjukkan bahwa jumlah sampel ini memenuhi syarat minimal untuk analisis psikometrik. Untuk instrumen dengan kurang dari 10 item seperti dalam penelitian ini, ukuran sampel minimal 30 responden, distribusi sampel akan tetap normal dan sudah mampu memberikan stabilitas statistik yang memadai dalam mengukur validitas dan reliabilitas.

Field [34] memperkuat pendekatan ini dengan menunjukkan bahwa kombinasi 9 item dengan skala likert dan 1 pertanyaan kualitatif menawarkan keunggulan

komprehensif. Data kuantitatif dari skala likert memungkinkan kuantifikasi sikap pengguna secara sistematis, sementara respons kualitatif memberikan kedalaman pemahaman tentang alasan di balik skor yang diberikan. Memastikan validitas dan keandalan data antara kedua jenis data ini meningkatkan validitas temuan secara keseluruhan. Dalam konteks *reengineering* Tapatupa, pendekatan ini memungkinkan identifikasi tidak hanya pada apa yang menjadi masalah (dari analisis skor likert), tetapi juga mengungkap mengapa masalah tersebut muncul (dari analisis kualitatif).

Beberapa pertimbangan metodologis perlu diperhatikan. Pertama, untuk analisis kuantitatif, uji normalitas distribusi skor perlu dilakukan mengingat jumlah sampel yang berada pada batas minimal. Kedua, dalam analisis kualitatif, meskipun dengan satu pertanyaan terbuka, peneliti harus memastikan pertanyaan dirumuskan secara efektif untuk mendorong respons yang bermakna. Ketiga, integrasi kedua jenis data harus dilakukan secara sistematis, di mana temuan kualitatif berfungsi untuk memperkaya dan menjelaskan pola yang terlihat dalam data kuantitatif.

## **2.10 ISO/IEC 25010 dan Kualitas Perangkat Lunak**

ISO/IEC 25010 merupakan standar internasional yang digunakan untuk mengevaluasi kualitas sistem dan perangkat lunak. Standar ini merupakan bagian dari keluarga *SQuaRE* (*System and Software Quality Requirements and Evaluation*), yang dirancang untuk mendukung proses rekayasa perangkat lunak melalui penentuan dan pengukuran kualitas. ISO/IEC 25010 membagi kualitas perangkat lunak ke dalam dua model utama, yaitu *Product Quality* dan *Quality in Use*. Dalam konteks rekayasa ulang aplikasi (*reengineering*), aspek *Product Quality* menjadi penting karena berfokus pada karakteristik internal dan eksternal perangkat lunak, termasuk perilaku fungsional yang dapat diamati selama penggunaan aplikasi [35].

Salah satu karakteristik utama dalam model *Product Quality* adalah *Functional Suitability*, yang menggambarkan sejauh mana suatu produk perangkat lunak menyediakan fungsi-fungsi yang sesuai dan dapat memenuhi kebutuhan pengguna. Karakteristik ini terdiri dari tiga subkarakteristik penting, yaitu *Functional Completeness*, *Functional Correctness*, dan *Functional Appropriateness*.

### **2.10.1 *Functional Completeness***

*Functional Completeness* didefinisikan sebagai sejauh mana semua fungsi yang dibutuhkan oleh pengguna tersedia dalam perangkat lunak. Dalam implementasinya, subkarakteristik ini berkaitan erat dengan kebutuhan fungsional yang telah didefinisikan sebelumnya. Ketika suatu sistem tidak menyediakan semua fungsi yang diharapkan, pengguna mungkin mengalami kesulitan dalam menyelesaikan tugasnya atau harus menggunakan solusi tambahan di luar sistem. Oleh karena itu, dalam proses reengineering, identifikasi kekurangan fungsional menjadi langkah awal yang krusial dalam memastikan kelengkapan sistem. Estdale dan Georgiadou menekankan pentingnya functional completeness sebagai indikator awal dari efektivitas layanan digital yang disediakan perangkat lunak.

### **2.10.2 *Functional Correctness***

Subkarakteristik *Functional Correctness* menggambarkan sejauh mana fungsi yang tersedia dalam perangkat lunak dapat menghasilkan keluaran yang benar dan sesuai dengan harapan pengguna. Kesesuaian ini mencakup ketepatan perhitungan, validitas data, serta keakuratan proses bisnis yang diterapkan dalam aplikasi. Dalam penelitian sebelumnya, *functional correctness* menjadi aspek penting yang berdampak langsung terhadap kepuasan pengguna serta kepercayaan terhadap sistem. Dalam konteks aplikasi Tapatupa, validasi terhadap alur proses permohonan, penyewaan, dan integrasi pembayaran menjadi bagian dari upaya memastikan fungsi bekerja dengan benar setelah dilakukan perbaikan sistem.

### **2.10.3 *Functional Appropriateness***

*Functional Appropriateness* mengacu pada kemampuan fungsi dalam mendukung tugas pengguna secara efisien dan efektif. Fungsi dianggap sesuai apabila mampu mempercepat proses kerja, mengurangi kompleksitas langkah, serta memudahkan pengguna dalam mencapai tujuannya. Dalam ISO/IEC 25010, subkarakteristik ini menjadi representasi langsung dari kegunaan fungsi dalam konteks nyata. Proses *reengineering* yang baik harus memperhatikan tidak hanya ketersediaan dan ketepatan fungsi, tetapi juga relevansi dan nilai guna fungsi terhadap konteks operasional pengguna. Perbaikan antarmuka, otomatisasi proses, dan integrasi



sistem pembayaran pada aplikasi Tapatupa menjadi contoh konkret dari upaya peningkatan *functional appropriateness*.

#### **2.10.4 Penerapan ISO / IEC 25010 dalam Evaluasi Sistem**

Studi yang dilakukan oleh Estdale dan Georgiadou (2018) menunjukkan bahwa penerapan ISO/IEC 25010 dalam pengembangan dan evaluasi perangkat lunak memungkinkan pengembang untuk melakukan penilaian kualitas yang lebih terstruktur dan terukur. Dalam penelitian tersebut, model kualitas digunakan untuk menilai layanan digital berbasis aplikasi, dengan penekanan pada kualitas fungsionalitas yang terlihat langsung oleh pengguna. Oleh karena itu, penerapan ISO/IEC 25010 dalam proses *reengineering* aplikasi Tapatupa diharapkan dapat membantu pengembang dalam mengidentifikasi, merancang, serta mengevaluasi ulang fungsi-fungsi utama sistem berdasarkan kebutuhan dan ekspektasi pengguna.

### **2.11 User Interface & User Experience**

*User Interface* (UI) dan *User Experience* (UX) merupakan dua aspek krusial dalam penelitian perangkat lunak yang berfokus pada pengalaman pengguna dalam berinteraksi dengan suatu sistem. UI berkaitan dengan desain visual, termasuk tata letak, warna, ikon, dan elemen interaktif lainnya. Sementara itu, UX lebih luas cakupannya karena mencakup bagaimana perasaan pengguna saat berinteraksi dengan suatu aplikasi atau *website*. Desain UI/UX yang baik bertujuan untuk meningkatkan kepuasan pengguna dengan membuat sistem lebih mudah digunakan serta meningkatkan efektivitas dalam mencapai tujuan pengguna [36]. Dalam konteks bisnis, UI/UX yang optimal dapat memberikan keuntungan kompetitif karena mampu meningkatkan loyalitas pengguna dan mendorong mereka untuk terus menggunakan aplikasi atau layanan digital [37].

#### **2.11.1 Pentingnya evaluasi UI/UX**

Evaluasi UI/UX menjadi tahap penting untuk memastikan bahwa desain yang dibuat benar-benar memenuhi kebutuhan pengguna. Beberapa metode yang sering digunakan dalam evaluasi UI/UX adalah *Custom Usability Questionnaires*, *System Usability Scale* (SUS), *heuristic evaluation*, dan *usability testing*. Metode SUS memungkinkan pengukuran kepuasan pengguna secara kuantitatif dengan

memberikan skor berdasarkan pengalaman mereka terhadap sebuah sistem [38]. Pengujian usability juga diperlukan untuk mengidentifikasi hambatan atau masalah dalam interaksi pengguna dengan sistem, sehingga perbaikan dapat dilakukan secara tepat sasaran [39].

### **2.11.2 Dampak Positif Implementasi UI/UX**

Implementasi UI/UX yang baik telah terbukti memberikan dampak positif terhadap kepuasan pengguna dalam berbagai studi. Misalnya, penelitian pada sistem reservasi Amadeus menunjukkan bahwa peningkatan UI/UX mampu meningkatkan loyalitas pengguna serta mendorong mereka untuk merekomendasikan sistem tersebut kepada orang lain. Studi lain pada *website Arttrash* menemukan bahwa penerapan metode *Design Thinking* dalam desain UI/UX secara signifikan meningkatkan keterlibatan pengguna dan memberikan pengalaman interaksi yang lebih baik [40]. Oleh karena itu, dapat disimpulkan bahwa desain UI/UX yang berkualitas sangat berpengaruh terhadap kesuksesan suatu produk digital. Dengan menerapkan metode yang tepat dalam penelitian dan evaluasi, sebuah sistem dapat lebih responsif terhadap kebutuhan pengguna, sehingga meningkatkan kepuasan serta loyalitas mereka terhadap layanan digital yang digunakan.

### **2.11.3 Desain UI/UX sebagai acuan**

Desain UI yang intuitif, mudah digunakan, dan efisien merupakan salah satu kunci keberhasilan dalam penelitian perangkat lunak. Mengacu pada tabel "*Design Patterns and Sub Elements*" dari studi literatur yang dilakukan oleh Punchoojit & Hongwarittorn [41], terdapat beberapa elemen yang mendukung terciptanya UI yang optimal. Komposisi halaman yang terstruktur dengan baik, seperti penggunaan menu, navigasi, dan tata letak yang jelas, memungkinkan pengguna menemukan informasi dengan mudah. Tampilan informasi yang disusun melalui daftar, klasifikasi, dan pengurutan yang logis juga meningkatkan keterbacaan serta pemahaman pengguna. Kontrol dan konfirmasi yang diberikan melalui umpan balik visual saat pengguna melakukan tindakan penting, seperti konfirmasi transaksi atau input data, sangat membantu dalam menghindari kesalahan.

Selain itu, penggunaan *pop-up* atau daftar hierarki untuk menampilkan informasi tambahan tanpa mengganggu alur utama pengguna menjadi elemen penting. Akses

lateral melalui tab dan pagination memungkinkan navigasi yang lebih fleksibel dan efisien, sementara link serta sistem navigasi yang intuitif mempermudah perpindahan antar halaman atau fitur. Tombol yang mudah ditekan dengan ukuran dan posisi yang sesuai, penggunaan ikon yang familiar, serta fitur *zoom*, *filter*, dan pencarian untuk membantu menemukan informasi yang relevan, semuanya berkontribusi pada pengalaman pengguna yang optimal. Terakhir, desain input area, *keyboard* virtual, dan *form* yang mempermudah pengguna dalam memasukkan data juga menjadi faktor penting dalam menciptakan UI yang mudah digunakan. Dengan mengacu pada elemen-elemen tersebut, desain UI yang intuitif, mudah digunakan, dan mudah digunakan dapat meningkatkan efektivitas sistem, mengurangi kesalahan pengguna, serta meningkatkan tingkat kepuasan dan loyalitas pengguna terhadap aplikasi atau layanan digital.

**Tabel 2. 7 Design Patterns and Subelements**

No	<i>Design Patterns</i>	<i>Subelements</i>
1	<i>Page composition</i>	<i>Menu</i>
2	<i>Display of information</i>	<i>List, classify, order</i>
3	<i>Control and confirmation</i>	<i>Sign on, confirmation, time-out</i>
4	<i>Revealing more information</i>	<i>Pop-up, prompt, hierarchical list</i>
5	<i>Lateral access</i>	<i>Tab, pagination</i>
6	<i>Navigation</i>	<i>Link, navigation</i>
7	<i>Button</i>	<i>No subelements</i>
8	<i>Icon</i>	<i>No subelements</i>
9	<i>Information control</i>	<i>Zoom, search, filter</i>
10	<i>Input mode and selection</i>	<i>Gesture, keyboard, input area, form</i>

## 2.12 Konsep Pengelolaan Aset Daerah

Pengelolaan aset daerah adalah suatu proses sistematis yang mencakup perencanaan, pemanfaatan, pemeliharaan, serta pengawasan terhadap aset milik pemerintah daerah. Aset tersebut meliputi tanah, bangunan, dan infrastruktur yang berfungsi sebagai pendukung penyelenggaraan layanan publik serta pembangunan ekonomi daerah [42]. Pengelolaan aset daerah harus dilakukan secara tertib, efektif, efisien, transparan, dan akuntabel sesuai dengan ketentuan dalam Peraturan Pemerintah Nomor 27 Tahun 2014 [43]. Dengan adanya sistem pengelolaan yang

baik, pemerintah daerah dapat memastikan bahwa aset yang dimiliki memberikan manfaat optimal serta menghindari potensi penyalahgunaan.

Tujuan utama dari pengelolaan aset daerah adalah untuk memastikan optimalisasi pemanfaatan aset guna mendukung pembangunan yang berkelanjutan serta meningkatkan pendapatan asli daerah (PAD). Prinsip efisiensi diterapkan agar aset yang tersedia digunakan secara maksimal tanpa adanya pemborosan sumber daya. Transparansi dalam pengelolaan aset didukung dengan penerapan sistem informasi manajemen aset berbasis teknologi digital, seperti Sistem Informasi Manajemen Daerah (SIMDA), yang memungkinkan pemantauan aset secara lebih akurat dan terbuka. Selain itu, akuntabilitas dalam pengelolaan aset daerah dapat ditingkatkan melalui audit rutin yang dilakukan oleh Badan Pemeriksa Keuangan (BPK) dan Badan Pengawasan Keuangan dan Pembangunan (BPKP).

Aset daerah memiliki peran strategis dalam mendukung layanan publik serta mendorong pertumbuhan ekonomi daerah. Pengelolaan yang baik dapat meningkatkan efisiensi pemanfaatan aset dan mencegah idle asset, yakni aset yang tidak digunakan secara optimal sehingga berpotensi menyebabkan pemborosan anggaran. Selain itu, aset daerah dapat dimanfaatkan melalui skema kerja sama dengan pihak swasta guna meningkatkan nilai ekonominya, misalnya melalui *Public-Private Partnership* (PPP) yang memungkinkan pengelolaan bersama untuk kepentingan publik. Dengan adanya strategi yang tepat, aset daerah dapat menjadi faktor pendorong utama dalam pembangunan berkelanjutan serta peningkatan kesejahteraan masyarakat.

Meskipun demikian, berbagai tantangan masih dihadapi dalam pengelolaan aset daerah, seperti kurangnya pencatatan aset yang akurat, risiko penyalahgunaan, serta potensi kebocoran pendapatan. Data aset yang tidak lengkap sering kali menyebabkan ketidaksesuaian dalam proses administrasi dan pengelolaan, sehingga aset menjadi tidak teridentifikasi dengan baik. Selain itu, lemahnya pengawasan dapat meningkatkan risiko penyalahgunaan aset oleh pihak yang tidak bertanggung jawab. Untuk mengatasi tantangan ini, diperlukan penguatan regulasi, peningkatan kapasitas sumber daya manusia yang menangani aset daerah, serta pemanfaatan teknologi informasi guna menciptakan sistem pengelolaan yang lebih transparan.

### **2.12.1 Regulasi dan Kebijakan Pengelolaan Aset Daerah**

Pengelolaan aset daerah di Indonesia memiliki landasan hukum yang diatur dalam berbagai regulasi, seperti Undang-Undang Nomor 23 Tahun 2014 tentang Pemerintahan Daerah dan Peraturan Pemerintah Nomor 27 Tahun 2014 mengenai Pengelolaan Barang Milik Negara/Daerah. Setiap pemerintah daerah biasanya menetapkan peraturan kepala daerah sebagai pedoman teknis untuk tata kelola aset, sebagaimana terlihat dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024 [3]. Kebijakan pengelolaan aset daerah mencakup beberapa aspek penting, yaitu inventarisasi, pemanfaatan, pengamanan, dan penghapusan aset. Proses inventarisasi bertujuan untuk mendata dan memastikan keakuratan informasi mengenai aset daerah, sementara optimalisasi pemanfaatan aset difokuskan untuk mendukung program pembangunan daerah. Dalam hal pengamanan, upaya yang dilakukan meliputi pengawasan dan sertifikasi guna menghindari potensi sengketa maupun kehilangan aset.

Adapun penghapusan aset dilakukan terhadap barang yang tidak lagi memiliki nilai manfaat atau ekonomis. Meskipun kerangka regulasi sudah tersedia, implementasinya kerap menghadapi berbagai kendala, seperti keterbatasan sumber daya manusia yang berkompeten, kurangnya sistem informasi yang memadai, serta hambatan administratif lainnya. Oleh karena itu, perlu dilakukan penguatan kapasitas aparatur pemerintah daerah serta penerapan teknologi modern untuk mendukung pengelolaan aset yang lebih terintegrasi. Secara umum, regulasi dan kebijakan pengelolaan aset daerah bertujuan untuk mendukung pembangunan berkelanjutan melalui tata kelola aset yang efisien dan efektif.

Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024 merupakan salah satu bentuk komitmen pemerintah daerah dalam meningkatkan pengelolaan barang milik daerah sesuai prinsip tata kelola pemerintahan yang baik. Diharapkan kebijakan ini dapat menjadi fondasi yang kokoh untuk menciptakan pengelolaan aset yang lebih transparan, profesional, dan akuntabel.

### **2.12.2 Transformasi Pengelolaan Aset Tanah Kabupaten Tapanuli Utara**

Pada akhirnya, pengelolaan aset tanah adalah mengenai cara terbaik bagi daerah untuk bangun dan memajukan wilayahnya. Aset tanah memainkan perannya dalam

memperbaiki kondisi sosial-ekonomi wilayah, dan pengelolaan aset tanah menjadi pekerjaan penting pemerintah. Sebagaimana disebutkan dalam Undang-undang Nomor 32 tahun 2004; Pengelolaan aset daerah adalah bagian dari reformasi pemerintahan untuk meningkatkan kesejahteraan masyarakat dan pendapatan asli daerah daripada hanya kursi panas. Transformasi ini adalah peluang yang dapat menerima mengurus, menerima, dan mengawasi. Baik itu di daerah administrasi publik atau peradilan pemerintah daerah, dalam penelitian Halim [44] memperhatikan bahwa *good governance* adalah solusi untuk memecahkan semua perbedaan interaksi dan masalah. Reformasi di bidang pengelolaan aset tanah membuahkan hasil yang positif bagi wilayah. Itu membantu pemerintah membangun fasilitas-fasilitas publik dan tempat seperti sekolah, rumah sakit, dan jalan yang lebih baik, mengurangi pemakaian tanah yang tidak terencana untuk tahap pertumbuhan ekonomi daerah, meningkatkan sektor ekonomi utama wilayah, desa dan masyarakat, masih banyak pembangunan pertanian dan daerah lain yang menguntungkan, gejala alam sekitar yang lebih baik dan lebih baik, dan lebih banyak pajak, retribusi, dan pendapatan sewa tanah. Transformasi ini merupakan langkah berharga dalam pencarian pembangunan berkelanjutan.

### **2.13 Gambaran Umum Aplikasi Tapatupa**

Pada akhirnya, pengelolaan aset tanah adalah mengenai cara terbaik bagi daerah untuk bangun dan memajukan wilayahnya. Aset tanah memainkan perannya dalam memperbaiki kondisi sosial-ekonomi wilayah, dan pengelolaan aset tanah menjadi pekerjaan penting pemerintah. Sebagaimana disebutkan dalam Undang-undang Nomor 32 tahun 2004; Pengelolaan aset daerah adalah bagian dari reformasi pemerintahan untuk meningkatkan kesejahteraan masyarakat dan pendapatan asli daerah daripada hanya kursi panas. Transformasi ini adalah peluang yang dapat menerima mengurus, menerima, dan mengawasi. Baik itu di daerah administrasi publik atau peradilan pemerintah daerah, dalam penelitian Halim [44] memperhatikan bahwa *good governance* adalah solusi untuk memecahkan semua perbedaan interaksi dan masalah. Reformasi di bidang pengelolaan aset tanah membuahkan hasil yang positif bagi wilayah. Itu membantu pemerintah membangun fasilitas-fasilitas publik dan tempat seperti sekolah, rumah sakit, dan

jalan yang lebih baik, mengurangi pemakaian tanah yang tidak terencana untuk tahap pertumbuhan ekonomi daerah, meningkatkan sektor ekonomi utama wilayah, desa dan masyarakat, masih banyak pembangunan pertanian dan daerah lain yang menguntungkan, gejala alam sekitar yang lebih baik dan lebih baik, dan lebih banyak pajak, retribusi, dan pendapatan sewa tanah. Transformasi ini merupakan langkah berharga dalam pencarian pembangunan berkelanjutan.

## 2.14 Penelitian Terdahulu

Dalam penelitian ini, dilakukan analisis terhadap beberapa penelitian terdahulu sebagai acuan dan referensi dalam menjalankan proses - proses penelitian. Tujuan dilakukan analisis terhadap penelitian terdahulu adalah membandingkan hasil penelitian dari studi yang pernah dilakukan dengan penelitian yang sedang dilakukan dalam penelitian ini yaitu aplikasi mobile penyewaan asset tanah Tapanuli Utara. Bab ini membahas penelitian terdahulu (*Related Research*). Pada bab ini, peneliti mengidentifikasi penelitian sebelumnya telah dilakukan oleh ahli, dan berhubungan dengan penelitian yang sedang diteliti oleh peneliti.

**Tabel 2. 8** Similar Research

No.	Penulisan (Tahun)	Judul Penelitian	Deskripsi Umum	Perbedaan	Persamaan
1.	Wijaya Kusuma (2024)	Rancang Bangun Aplikasi Sistem Informasi Manajemen Aset Tanah Berbasis Webgis Di Kabupaten Kayong Utara Provinsi Kalimantan Barat”	Penelitian ini berfokus pada perancangan aplikasi sistem informasi manajemen aset tanah berbasis webgis. Pada penelitian ini memanfaatkan WebGIS untuk pengelolaan aset dengan mengumpulkan, menganalisis	Pada penelitian ini dalam meningkatkan efisiensi pengelolaan aset memanfaatkan WebGIS yang dimana pemetaan WEBGIS nya meliputi informasi instansi pemerintah, luas wilayah, alamat	Penelitian yang dilakukan sama-sama menggunakan metode Penelitian dan Penelitian R&D (Research and Development)

No.	Penulisan (Tahun)	Judul Penelitian	Deskripsi Umum	Perbedaan	Persamaan
			dan mevisualisasikan data secara geografis	berdasarkan sertifikat, nomor sertifikat, dan koordinat.	
2.	Muhammad Syaeful Fajar, Gus Nanang Syaifuddiin, Nisa'ul Hafidhoh, MH.Ramdhani Ismar, Maulana Nur Ivansyah (2023)	Optimizing Asset Management: A Risk-Based Approach with Inventory Outsourcing and Asset Management Information System	Penelitian ini bertujuan untuk meningkatkan Pengelolaan aset daerah dengan menggunakan sistem informasi manajemen aset berbasis web yang menerapkan prinsip Risk- Based Asset Management (RBAM) dan fokus utamanya yaitu Optimalisasi Manajemen Aset, Kategorisasi Risiko, Analisis Data, Efisiensi dan Responsivitas.	Penelitian ini menerapkan Prinsip Risk- Based Asset Management (RBAM) yang dimana metode ini mengategorikan permintaan barang berdasarkan tingkat risiko. Pendekatan yang digunakan pada penelitian ini yaitu Waterfall.	Metode penelitian yang digunakan yaitu Research and Development (R&D) dan bertujuan untuk meningkatkan pengelolaan aset.
3.	Indah Nurfi, Yenni Samri Juliati Nasution, Nur Fadhilah Ahmad	Analysis of Regional Asset Management Information System Application	Penelitian ini bertujuan untuk menganalisis implementasi sistem Informasi Manajemen Aset Daerah	Penelitian ini berfokus untuk menganalisis sistem informasi manajemen aset.	Penelitian yang dilakukan menggunakan metode kualitatif dengan



No.	Penulisan (Tahun)	Judul Penelitian	Deskripsi Umum	Perbedaan	Persamaan
	Hasibuan (2024)	for Medan City Library Assets and Archive Services	(RAMIS) di Perpustakaan dan Arsip Kota Medan		pengumpulan data melalui wawancara, observasi.
4.	Khiati, N., Bouchiha, D., Atig, Y., & Hacene, S. B. (2024).	Reengineering Web applications into pervasive information systems.	Penelitian ini memberikan kontribusi yang signifikan dalam bidang rekayasa ulang aplikasi web ke mobile dengan pendekatan berbasis model. <i>Framework</i> WA2MA yang diusulkan menawarkan solusi otomatis, mengurangi upaya manual, dan mempercepat konversi aplikasi <i>web ke mobile</i> .	Penelitian ini memanfaatkan metode MDE (Model - Driven Engineering)	Penelitian ini memanfaatkan metode <i>prototype</i> untuk tahapan Implementasi di bagian <i>“Forward Engineering”</i>
5.	Yutta Nandiya Putri (2019)	Reengineering Aplikasi Moflus Dengan Proses Model Horseshoe	Penelitian ini Penelitian ini berfokus dalam melakukan reengineering aplikasi Moflus, yang dimana aplikasi moflus memiliki	Penelitian ini berfokus dalam reengineering pada struktur MoFlus yang tidak MVP (Model-View- Presnter) menjadi MVP	Penelitian yang dilakukan sama-sama berfokus untuk melakukan reengineering pada suatu

No.	Penulisan (Tahun)	Judul Penelitian	Deskripsi Umum	Perbedaan	Persamaan
			masalah yaitu sulitnya dalam penelitian aplikasi yang sudah ada dikarenakan struktur kodenya yang sudah sangat lama dan tidak MVP sehingga mengakibatkan sulit dikembangkan kembali.	dan menggunakan metode model proses horseshoe.	aplikasi yang sudah ada .

Berdasarkan tabel penelitian terdahulu, terlihat bahwa sebagian besar studi sebelumnya hanya menerapkan metode pengembangan secara waterfall atau tidak sepenuhnya menempatkan pengguna sebagai pusat dalam siklus pengembangan. Sebagai contoh, Putri (2019) hanya membahas pengembangan ulang berbasis struktural tanpa melibatkan validasi langsung dari pengguna. Berbeda dengan penelitian ini, pendekatan *User-Centered Design* (UCD) tetap digunakan sebagai dasar perancangan sistem dengan fokus pada kebutuhan pengguna yang diidentifikasi dari studi dokumentasi dan observasi. Proses pengembangan dilakukan secara iteratif menggunakan metode Agile Scrum, yang memungkinkan adaptasi terhadap kebutuhan sistem secara bertahap. Evaluasi akhir dilakukan melalui *User Acceptance Test* (UAT) pada fitur pembayaran oleh pihak terkait, serta penyebaran kuesioner untuk menilai kepuasan pengguna terhadap sistem secara menyeluruh. Dengan demikian, penelitian ini memberikan kontribusi pada pengembangan aplikasi layanan publik berbasis reengineering yang lebih adaptif dan responsif terhadap kebutuhan pengguna.

### 2.15 Kesimpulan

Berdasarkan tinjauan pustaka yang telah dibahas, dapat disimpulkan bahwa konsep *reengineering*, metode *Agile Scrum*, dan pendekatan *User-Centered Design* (UCD) menjadi dasar dalam pengembangan ulang aplikasi Tapatupa. Pemanfaatan teknologi *Flutter* dan prinsip desain UI/UX yang sesuai dengan standar ISO 25010 mendukung tercapainya kualitas sistem yang sesuai kebutuhan pengguna. Tinjauan terhadap penelitian terdahulu menunjukkan bahwa pendekatan berbasis pengguna masih jarang diterapkan secara menyeluruh dalam konteks layanan publik daerah, sehingga penelitian ini diharapkan dapat memberikan kontribusi nyata dalam penerapan metode UCD pada sistem layanan digital pemerintah daerah.

## **BAB III**

### **ANALISIS DAN DESAIN**

Pada bab ini akan dijelaskan mengenai analisis dan desain yang akan digunakan sebagai acuan untuk tahap implementasi. Analisis dilakukan untuk mendapat gambaran kebutuhan dari aplikasi yang akan dibangun, sementara desain dibuat sebagai perancangan tahapan *reengineering* yang akan dilaksanakan pada tahap implementasi. Selain itu, pada bab ini dijelaskan tentang metode penelitian yang digunakan dalam penelitian aplikasi Tapatupa. Metode yang digunakan dalam penelitian ini mengacu pada pendekatan *User-Centered Design* (UCD) dengan model pengembangan berbasis *Agile Scrum*.

#### **3.1 Analisis Kebutuhan Sistem**

Analisis ini dilakukan dalam konteks reengineering aplikasi Tapatupa berbasis *mobile*. Penelitian ini menggunakan pendekatan *User-Centered Design* (UCD), yang terdiri dari empat tahap utama, yaitu: (1) memahami dan menentukan konteks penggunaan (*understand and specify the context of use*), (2) menetapkan kebutuhan pengguna (*specify the user requirements*), (3) merancang erat desain (*produce design solutions*), dan (4) mengevaluasi desain berdasarkan kebutuhan pengguna (*evaluate designs against requirements*).

##### **3.1.1 *Understand and Specify the Context of Use***

Tahap ini bertujuan untuk memahami konteks penggunaan sistem oleh pengguna akhir, khususnya dalam proses penyewaan aset tanah yang dikelola oleh Pemerintah Kabupaten Tapanuli Utara. Pemahaman ini menjadi dasar dalam merancang sistem yang lebih sesuai dengan kebutuhan 50 terat dan meningkatkan kualitas pengalaman pengguna.

Analisis dilakukan melalui dua pendekatan utama, yaitu observasi terhadap proses manual dan penelusuran sistem aplikasi Tapatupa versi sebelumnya. Pertama, pada proses manual, ditemukan bahwa pencatatan permohonan dilakukan secara tertulis, verifikasi dokumen fisik masih bergantung pada tatap muka dengan petugas, dan pelaporan pembayaran dilakukan secara konvensional. Hal ini menimbulkan beberapa kendala seperti keterlambatan proses verifikasi, potensi kehilangan

dokumen, serta tidak tersedianya pemberitahuan atau notifikasi kepada pemohon mengenai status permohonan.

Kedua, observasi juga dilakukan terhadap aplikasi Tapatupa versi awal. Penelusuran dilakukan pada alur penggunaan mulai dari login pengguna, pengajuan permohonan, hingga proses pembayaran. Beberapa kelemahan yang ditemukan meliputi navigasi yang membingungkan, tidak adanya fitur unggah dokumen, serta proses pembayaran yang tidak terintegrasi dengan status permohonan, sehingga pengguna harus mengunggah bukti bayar secara manual dan tidak mendapatkan informasi real-time mengenai tagihan dan status pembayaran.

Untuk memperkuat pemahaman terhadap proses dan regulasi yang berlaku, peneliti juga melakukan studi terhadap dokumen resmi, yaitu Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024. Dokumen ini memberikan kerangka hukum mengenai alur administrasi, struktur tarif, persyaratan dokumen, dan peran berbagai pihak dalam proses pemanfaatan aset daerah. Studi ini sangat penting dalam memastikan bahwa sistem yang dirancang nantinya selaras dengan ketentuan formal.

Berdasarkan hasil analisis dari observasi sistem dan studi regulasi, diperoleh kesimpulan bahwa sistem perlu dirancang ulang dengan mempertimbangkan kebutuhan seperti peningkatan akses informasi, penyederhanaan proses permohonan, penambahan fitur unggah dokumen, serta integrasi otomatisasi dalam proses pembayaran. Temuan-temuan ini menjadi dasar dalam pengembangan ulang sistem agar lebih terstruktur, transparan, dan mudah digunakan oleh masyarakat sebagai pengguna utama.

### **3.1.2 Identifikasi *Stakeholder* dan Pengguna**

Tahap ini bertujuan untuk mengidentifikasi kelompok pengguna utama yang menjadi fokus dalam proses reengineering aplikasi Tapatupa. Analisis dilakukan terhadap sistem yang telah ada sebelumnya, yaitu aplikasi *mobile* Tapatupa versi awal. Berdasarkan hasil penelusuran terhadap fitur-fitur yang tersedia, peneliti mengidentifikasi fungsi-fungsi utama yang ditujukan kepada pengguna masyarakat sebagai wajib retribusi. Fitur-fitur tersebut kemudian dianalisis untuk memetakan kebutuhan pengguna secara lebih tepat sasaran dalam pengembangan ulang sistem.

Pengembangan ulang aplikasi Tapatupa menggunakan pendekatan *User-Centered Design* (UCD), yang menekankan pada pemahaman mendalam terhadap kebutuhan pengguna akhir. Dalam konteks ini, masyarakat wajib retribusi merupakan satu-satunya kelompok pengguna utama yang secara langsung berinteraksi dengan sistem. Mereka adalah individu yang mengajukan permohonan penyewaan aset milik pemerintah daerah, seperti tanah atau bangunan, melalui aplikasi.

Pengguna ini memiliki peran penting dalam seluruh siklus penggunaan aplikasi, mulai dari pengajuan permohonan, pengunggahan dokumen pendukung, pengecekan status permohonan, hingga pelunasan pembayaran. Oleh karena itu, seluruh proses *reengineering* diarahkan untuk meningkatkan keterpakaian, kenyamanan, serta kejelasan alur layanan yang diterima oleh pengguna utama ini. Fokus utama sistem yang dikembangkan adalah memastikan bahwa pengguna masyarakat dapat mengakses informasi secara transparan, melakukan proses permohonan dengan mudah, dan memperoleh umpan balik sistem secara langsung. Hal ini dilakukan untuk menjawab berbagai kendala yang ditemukan pada versi aplikasi sebelumnya, serta untuk memenuhi prinsip-prinsip perancangan berbasis pengguna secara iteratif dan berkelanjutan.

### **3.1.2.1 User Characteristics**

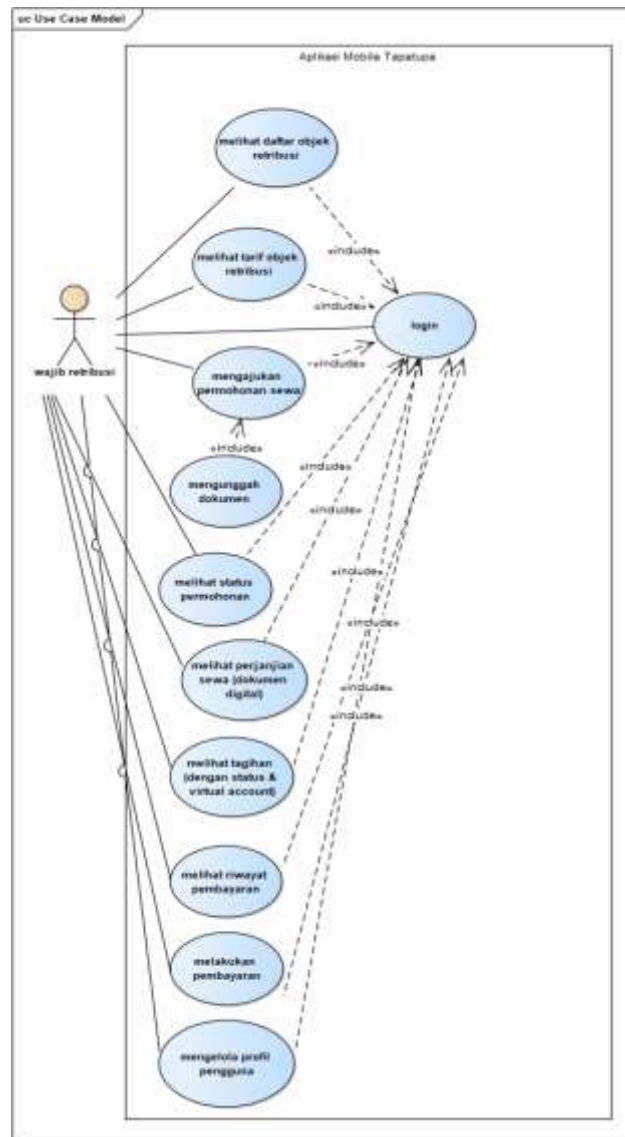
Karakteristik pengguna utama dalam aplikasi *mobile* Tapatupa adalah masyarakat yang tergolong sebagai wajib retribusi atas pemanfaatan aset milik daerah, seperti tanah dan/atau bangunan. Pengguna ini berasal dari berbagai latar belakang, mulai dari individu hingga pelaku usaha yang mengajukan permohonan sewa aset daerah. Berdasarkan hasil observasi dan studi dokumen, pengguna sistem cenderung memiliki keterampilan dasar dalam menggunakan aplikasi *mobile* seperti login, mengisi formulir, serta melakukan pembayaran digital. Oleh karena itu, antarmuka dan alur aplikasi perlu dirancang dengan mempertimbangkan kesederhanaan navigasi, dan kejelasan informasi. Karakteristik pengguna pada aplikasi *mobile* Tapatupa saat ini dijelaskan pada Tabel 3.1.

**Tabel 3. 1** User Characteristics Current System

User Group – Role	Kepentingan Akses	Hak Akses
Group – 1 – Masyarakat Wajib Retribusi	1. Halaman Beranda 2. Halaman Permohonan 3. Halaman Objek Retribusi 4. Halaman Tarif Objek Retribusi 5. Halaman Perjanjian 6. Halaman Tagihan 7. Halaman Pembayaran 8. Halaman Profile	1. Melakukan autentikasi 2. Mengajukan Permohonan 3. Melihat daftar tarif objek 4. Retribusi 5. Melihat Daftar objek 6. Retribusi 7. Melihat status permohonan 8. Melihat perjanjian sewa (dokumen digital) 9. Melihat tagihan (dengan status & <i>virtual account</i> ) 10. Melihat 53terati pembayaran 11. Melakukan pembayaran 12. Mengelola profil pengguna

### 3.1.2.2 Use Case Target System

Untuk menggambarkan interaksi pengguna dengan sistem hasil *reengineering*, berikut disajikan *Use Case Diagram* yang merepresentasikan fungsi-fungsi yang dapat diakses oleh terative sebagai pengguna utama. Diagram ini disusun berdasarkan hasil analisis kebutuhan pengguna dan regulasi daerah yang menjadi dasar sistem, dengan tujuan agar alur aplikasi lebih relevan, dan mudah dipahami oleh masyarakat. *Use case diagram* tersebut dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Use Case Diagram setelah Reengineering

### 3.1.3 Analisis Sumber Data dan Studi Pendukung

Untuk menghasilkan perancangan ulang aplikasi yang lebih terarah dan sesuai dengan kebutuhan terat, tahap awal yang dilakukan adalah menganalisis berbagai sumber data dan studi pendukung. Sumber data ini menjadi dasar dalam memahami konteks penggunaan, kebutuhan fungsional, serta permasalahan yang terdapat pada sistem sebelumnya. Tiga sumber utama yang dijadikan acuan dalam analisis ini adalah dokumen regulasi resmi, aplikasi sejenis yang memiliki fungsi serupa, dan aplikasi Tapatupa versi awal yang telah digunakan sebelumnya.



Analisis terhadap dokumen Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024 bertujuan untuk memastikan bahwa sistem yang dibangun mematuhi ketentuan hukum dan mendukung proses administrasi aset tanah secara formal. Selanjutnya, Studi aplikasi sejenis dilakukan untuk mengidentifikasi fitur utama yang menjadi standar dalam aplikasi pengelolaan aset pemerintah, sehingga dapat digunakan sebagai acuan dalam perancangan ulang. Terakhir, analisis terhadap aplikasi Tapatupa versi sebelumnya dilakukan untuk mengidentifikasi kekurangan, hambatan, serta peluang perbaikan dari sistem yang telah ada.

Dengan menggabungkan ketiga sumber tersebut, proses perancangan ulang dapat dilakukan secara komprehensif dan berbasis data, sehingga menghasilkan sistem yang lebih tepat guna, efisien, dan selaras dengan kebutuhan pengguna serta kebijakan yang berlaku.

#### **3.1.3.1. Studi Aplikasi Sejenis**

Dalam proses *reengineering* aplikasi Tapatupa, dilakukan analisis dengan melakukan pengumpulan informasi terhadap beberapa aplikasi sejenis yang memiliki fungsi serupa dalam pengelolaan aset pemerintah. Tujuan dilakukan pengumpulan data terhadap aplikasi sejenis adalah untuk menemukan fitur yang ditawarkan dan cara aplikasi menyelesaikan masalah pengguna. Pengamatan juga dilakukan untuk mengidentifikasi kelebihan dan kekurangan sehingga perlu dilakukan proses *reengineering*. Selain menemukan fitur - fitur yang dianggap penting, dapat digunakan sebagai acuan untuk mengembangkan fitur.

Setelah melakukan pengamatan terhadap tiga aplikasi Penyewaan Aset Tanah di setiap daerah pada **sub bab 2.5**, penulis menyimpulkan perbandingan terhadap aplikasi tersebut yang dapat dilihat pada Tabel 3.2.

**Tabel 3. 2** Objek Perbandingan

<b>Nama Sistem</b>	<b>Kelebihan</b>	<b>Kekurangan</b>
SiapSEWA	Menampilkan informasi aset secara detail (gambar, alamat, video, deskripsi) dan menarik.	Tidak adanya fitur pembayaran atau pelaporan yang terintegrasi
Amanah Bandung	Fitur pembayaran yang lengkap (tagihan, status pembayaran, metode pembayaran), serta adanya menu pelaporan dan pelayanan yang memudahkan pengelolaan aset	Kurangnya informasi detail tentang aset seperti gambar, 56terat, dan ukuran luas tanah.
SIWA App	Adanya fitur kolom pencarian yang mudah dan pengelompokan aset yang jelas.	Tidak ada informasi detail terhadap informasi tarif objek retribusi.

Setelah melakukan pengamatan pada ketiga sistem penyewaan aset, maka terdapat beberapa aspek pada sistem tersebut yang akan diadopsi pada aplikasi Tapatupa yang akan dijelaskan pada Tabel 3.3.

**Tabel 3. 3** Aspek yang Diadopsi

<b>No</b>	<b>Aspek yang diadopsi</b>	<b>Keterangan</b>
1	<i>Visual Hierarchy</i>	Menggunakan hirarki visual yang jelas untuk menonjolkan informasi penting seperti harga, 56terat, dan ketersediaan aset. Ini dapat dilakukan dengan ukuran <i>font</i> , warna, dan penempatan elemen yang strategis.
2	<i>Feedback dan Interaction</i>	Memberikan umpan balik yang jelas saat pengguna melakukan 56terativ, seperti mengklik dan menyelesaikan pembayaran. Ini meningkatkan kepercayaan pengguna terhadap sistem.
3	Fitur pembayaran terintegrasi	Menyediakan sistem pembayaran yang terintegrasi untuk memudahkan transaksi sewa atau beli aset, seperti menyediakan fitur pembayaran lengkap, termasuk tagihan, status pembayaran, dan metode pembayaran.

No	Aspek yang diadopsi	Keterangan
		Melakukan integrasi dengan gateway pembayaran pada <i>bank local</i> (BPD Sumut) dan memberikan notifikasi status pembayaran

### 3.1.3.2. Studi Aplikasi Sebelumnya (Tapatupa versi Awal)

Aplikasi Tapatupa versi awal dikembangkan sebagai solusi digital dalam menangani permasalahan pengelolaan retribusi sewa tanah dan bangunan di wilayah Kabupaten Tapanuli Utara. Pembangunan aplikasi ini dilatarbelakangi oleh kebutuhan akan sistem yang dapat memfasilitasi proses pengajuan, perpanjangan, dan perpindahan permohonan sewa secara lebih efisien tanpa harus melakukan kontak langsung dengan petugas dinas. Dalam praktiknya, Tapatupa memfasilitasi pembayaran retribusi melalui dua metode, yakni pembayaran tunai yang dilakukan langsung kepada petugas, serta pembayaran non-tunai melalui integrasi dengan sistem *payment gateway* yang tersedia di aplikasi.

Berdasarkan dokumen *Software Technical Specification* dari versi awal aplikasi, sistem tersebut memiliki delapan fungsi utama yang ditujukan kepada pengguna dengan status wajib retribusi. Rincian fungsi tersebut ditampilkan pada Tabel 3.4 yang terdiri dari:

**Tabel 3. 4** Functional Specification versi Awal

No	Fungsi	Deskripsi Fungsi
1	Beranda untuk wajib retribusi	Beranda menampilkan tagihan wajib retribusi
2	Halaman Permohonan untuk wajib retribusi	Halaman Permohonan untuk menampilkan dan mengajukan permohonan sewa dari wajib retribusi.
3	Halaman Objek Retribusi untuk wajib retribusi	Halaman Objek retribusi menampilkan daftar objek retribusi yang ada di Tapanuli Utara.

No	Fungsi	Deskripsi Fungsi
4	Halaman Tarif Objek Retribusi untuk wajib retribusi	Halaman Tarif Objek Retribusi menampilkan berapa tariff dari setiap objek retribusi.
5	Halaman Perjanjian untuk wajib retribusi	Halaman perjanjian untuk menampilkan perjanjian yang telah dibuat oleh wajib retribusi
6	Halaman Tagihan untuk wajib retribusi	Halaman Tagihan menampilkan tagihan yang harus dibayarkan oleh wajib retribusi
7	Halaman Pembayaran untuk wajib retribusi	Halaman Pembayaran menampilkan informasi pembayaran yang belum dibayar oleh wajib retribusi
8	Halaman Profile untuk wajib retribusi	Halaman ini menampilkan data diri wajib retribusi dan dapat mengubah password akun.

Meskipun aplikasi ini telah menyediakan fitur-fitur dasar dalam pengelolaan retribusi, hasil analisis terhadap versi awal menunjukkan adanya sejumlah kekurangan yang teridentifikasi secara terukur. Beberapa temuan tersebut antara lain adalah struktur antarmuka yang belum konsisten, alur penggunaan yang tidak sepenuhnya sesuai dengan urutan proses pengguna, serta keterbatasan fungsi pencarian dan validasi input. Misalnya, pada proses permohonan sewa, sistem belum menyediakan fitur unggah dokumen pendukung sebagaimana disyaratkan oleh regulasi. Selain itu, sistem pembayaran masih memerlukan unggah bukti bayar secara manual, yang berpotensi menimbulkan duplikasi data dan kesalahan verifikasi. Kemudian, tidak ditemukan indikator visual yang memadai untuk menunjukkan status proses yang sedang berjalan, seperti status pembayaran atau keberhasilan pengajuan. Hal ini menyebabkan proses penggunaan aplikasi menjadi kurang transparan dan sulit dilacak secara sistematis oleh pengguna.

Berdasarkan temuan tersebut, dilakukan proses reengineering yang berfokus pada pemenuhan standar fungsional dan kebutuhan pengguna secara lebih terstruktur dan

terukur. Pendekatan yang digunakan mengacu pada prinsip *User-Centered Design* (UCD), sehingga pembaruan yang dilakukan tidak hanya berdasarkan asumsi teknis, tetapi juga merujuk pada data dari studi aplikasi sejenis, regulasi yang berlaku, serta permasalahan nyata pada sistem sebelumnya. Hasil dari reengineering ditargetkan untuk menghasilkan fitur yang mendukung pelacakan proses, validasi input yang lebih kuat, tampilan antarmuka yang konsisten, serta alur penggunaan yang mengikuti urutan proses administrasi yang sebenarnya.

### 3.1.4. *Current System Analysis*

Sebelum dilakukan proses reengineering, penting untuk memahami kondisi sistem yang ada saat ini sebagai dasar perbandingan dan evaluasi. Aplikasi Tapatupa versi awal telah digunakan wajib retribusi Kabupaten Tapanuli Utara untuk mengajukan penyewaan aset tanah milik daerah. Namun, berdasarkan hasil analisis dan observasi terhadap sistem lama, terdapat beberapa kelemahan pada fitur, alur bisnis, serta desain antarmuka yang masih belum optimal. Subbab ini membahas secara mendalam struktur sistem, fungsi utama, serta kendala-kendala yang ditemukan pada sistem sebelumnya.

#### 1. Gambaran Umum Sistem Awal

Nama sistem: Aplikasi Tapatupa versi awal

Teknologi: Flutter (*front-end*)

Fitur utama: Formulir pengajuan permohonan sewa, riwayat permohonan, perjanjian sewa, dan informasi tentang daftar objek dan tarif objek.

Pengguna utama: Masyarakat wajib retribusi

#### 2. Analisis Fungsionalitas

Beberapa kelemahan pada sistem lama:

- a. Tidak tersedianya fitur unggah dokumen pendukung saat pengajuan permohonan, sehingga menyulitkan pengguna dalam memenuhi persyaratan administratif sesuai Peraturan Bupati No. 12 Tahun 2024.
- b. Pencarian objek retribusi tidak dapat dilakukan berdasarkan wilayah atau kategori, sehingga pengguna mengalami kesulitan dalam menemukan objek yang ingin disewa.

- c. Informasi tarif retribusi tidak dapat diakses secara spesifik, menyebabkan ketidakjelasan bagi pengguna dalam mengetahui biaya sewa.
- d. Dokumen perjanjian tidak tersedia dalam bentuk digital, sehingga pengguna tidak dapat langsung mengakses dokumen yang sah setelah proses persetujuan.
- e. Sistem pembayaran belum terintegrasi dengan *virtual account*, menyebabkan pengguna harus mengunggah bukti pembayaran secara manual, yang berpotensi menimbulkan duplikasi data dan kesalahan verifikasi.
- f. Status pembayaran tidak berubah secara otomatis, sehingga pengguna tidak dapat melacak riwayat pembayaran secara sistematis.

### 3. Evaluasi UI/UX

- a. Desain antarmuka dengan navigasi yang menunjukkan alur penggunaan aplikasi masih kurang.
- b. Informasi penting seperti tagihan dan status permohonan tidak ditampilkan secara jelas.
- c. Tidak adanya indikator visual untuk menunjukkan proses yang sedang berjalan, seperti status pembayaran atau keberhasilan pengajuan.

### 4. Kesesuaian dengan Regulasi

- a. Sistem belum sepenuhnya memenuhi ketentuan Peraturan Bupati No. 12 Tahun 2024, khususnya dalam hal dokumentasi persyaratan dan transparansi tarif sewa.
- b. Proses administrasi digital belum sepenuhnya terdokumentasi secara sistematis, sehingga menyulitkan proses verifikasi.

#### 3.1.5. *Target System Specification*

Berdasarkan hasil analisis terhadap sistem lama dan studi regulasi, dirancang target sistem baru yang akan menjadi acuan dalam proses *reengineering*. Sistem baru ditujukan untuk meningkatkan efisiensi administrasi, memperjelas alur bisnis, serta

memberikan pengalaman pengguna yang lebih baik. Subbab ini menjelaskan spesifikasi teknis dan fungsional dari sistem yang direncanakan, termasuk fitur utama, arsitektur sistem, dan standar desain UI/UX yang digunakan.

#### 1. Gambaran Umum Sistem Baru

Nama Sistem : Aplikasi Tapatupa versi reengineering

Teknologi :

*Front - end: Flutter (untuk platform mobile)*

*Back - end: Laravel dengan MySQL*

#### 2. Tujuan Utama:

Memperbaiki fitur sebelumnya dan menambahkan beberapa fitur, serta memastikan target sistem sesuai dengan regulasi daerah.

#### 3. Fitur Utama Sistem Baru

**Tabel 3. 5** Fitur Utama Sistem Baru

No	Fungsi Sistem setelah <i>Reengineering</i>	Tujuan Pengembangan
1	Unggah Dokumen pada fitur Permohonan	Memenuhi syarat administrative sesuai perbup No. 12 tahun 2024 dan meningkatkan validitas permohonan <sup>2</sup>
2	Fiter pencarian pada Objek Retribusi	Mempermudah pencarian objek berdasarkan kecamatan atau nama objek
3	Fiter pencarian pada Tarif Retribusi	Memberikan kemudahan akses terhadap informasi tarif objek tertentu
4	Tautan Dokumen pada Halaman Perjanjian	Menyediakan akses cepat terhadap dokumen perjanjian dalam bentuk digital
5	Tampilan tagihan terintegrasi <i>Virtual Account</i>	Mengotomatisasi proses pembayaran dan menampilkan status tagihan yang terstruktur dan akurat
6	Riwayat Pembayaran Otomatis	Menyediakan pelacakan 61 terati pembayaran dengan status terverifikasi secara otomatis oleh sistem

#### 4. Standar Desain UI / UX

- a. Menggunakan prinsip *User Centered Desain* (UCD)
- b. Antarmuka dirancang dengan sederhana, dan responsif.

- c. Warna dan ikon disesuaikan dengan identitas visual Pemerintah Tapanuli Utara.
- d. Informasi penting seperti tagihan dan status permohonan ditampilkan secara jelas dan terstruktur.

### 3.1.6. *Requirement Mapping* Berdasarkan Regulasi

Salah satu aspek penting dalam pengembangan sistem ini adalah memastikan bahwa seluruh fitur dan alur proses sesuai dengan ketentuan hukum yang berlaku, yaitu Peraturan Bupati Kabupaten Tapanuli Utara No. 12 Tahun 2024 tentang Pengelolaan Aset Tanah Milik Daerah. Untuk itu, dilakukan pemetaan kebutuhan sistem berdasarkan pasal-pasal dalam regulasi tersebut, sehingga sistem yang dikembangkan dapat menjadi alat administrasi yang sah dan dapat dipertanggungjawabkan. Subbab ini menjelaskan bagaimana setiap ketentuan dalam regulasi diimplementasikan ke dalam bentuk kebutuhan fungsional sistem.

#### 1. Dasar Regulasi

Regulasi yang digunakan sebagai dasar *mapping* adalah:

Nama Dokumen: Peraturan Bupati Kabupaten Tapanuli Utara Nomor 12 Tahun 2024

Tentang: Pedoman Pemanfaatan Barang Milik Daerah Berupa Tanah dan / atau Bangunan

Tujuan: Menjadi pedoman dalam pemanfaatan barang milik daerah termasuk tata cara sewa, penetapan tarif, serta mekanisme pengelolaan aset.

**Tabel 3. 6** Mapping Kebutuhan Sistem Berdasarkan Regulasi

Pasal	Isi Ketentuan	Kebutuhan
Pasal 3 Ayat (1)	Calon penyewa harus mengajukan surat permohonan disertai dokumen pendukung	Fitur unggah dokumen wajib saat pengajuan permohonan
Pasal 3 Ayat (2) Huruf a	Surat permohonan harus memuat data calon penyewa	Formulir pengajuan menyertakan input data identitas lengkap pengguna
Pasal 3 Ayat (2) Huruf b	Surat permohonan harus mencantumkan jangka waktu penyewaan	Input durasi sewa dengan maksimal 10 tahun sesuai Pasal 9



Pasal	Isi Ketentuan	Kebutuhan
Pasal 3 Ayat (2) Huruf c	Surat permohonan harus menyebutkan peruntukan sewa	Kolom deskripsi tujuan penggunaan tanah
Pasal 5 Ayat (2)	Permohonan sewa harus dilengkapi dokumen identitas dan surat permohonan	Sistem menyediakan kolom dan form upload dokumen wajib (KTP, surat permohonan)
Pasal 7 Ayat (1)	Penetapan tarif sewa berdasarkan 63terat dan luas tanah	Sistem menampilkan estimasi tarif sewa otomatis berdasarkan input pengguna
Pasal 9	Jangka waktu sewa maksimal 10 tahun	Sistem membatasi durasi sewa maksimal 10 tahun

## 2. Analisis Hasil *Mapping*

Dari total 10 pasal relevan, semua telah diimplementasikan dalam bentuk kebutuhan sistem. Beberapa contoh implementasi langsung dari ketentuan regulasi adalah:

- a. Fitur unggah dokumen pada formulir permohonan untuk memenuhi persyaratan administratif.
- b. Input durasi sewa.
- c. Sistem penagihan untuk mendukung transparansi dan akuntabilitas pembayaran.
- d. Estimasi tarif sewa otomatis berdasarkan lokasi dan luas tanah, sesuai Pasal 7.

*Mapping* ini menjadi dasar untuk merancang solusi desain dan menentukan prioritas pengembangan fitur dalam tahap selanjutnya.

Dengan adanya pemetaan ini, sistem yang dikembangkan tidak hanya memenuhi kebutuhan pengguna dalam hal kemudahan akses dan kenyamanan, tetapi juga sepenuhnya patuh terhadap regulasi daerah, sehingga dapat digunakan sebagai alat administrasi digital yang valid dan dapat dipertanggungjawabkan.

### 3.1.7. *Specify the User Requirements*

Rancangan fungsi sistem merupakan bagian penting dalam proses *reengineering* aplikasi Tapatupa. Fungsi-fungsi yang dirancang berfokus pada peningkatan proses permohonan penyewaan, otomatisasi pembayaran, serta penyampaian informasi

secara lebih terstruktur dan sesuai regulasi. Perancangan ini didasarkan pada hasil analisis terhadap sistem versi awal, studi dokumentasi peraturan daerah, serta perbandingan dengan aplikasi sejenis yang relevan dalam konteks pengelolaan aset daerah.

Fungsi-fungsi tersebut dikembangkan untuk menjawab permasalahan nyata yang ditemukan pada sistem lama, seperti ketiadaan fitur unggah dokumen pendukung saat permohonan, keterbatasan dalam pencarian data objek dan tarif retribusi, serta proses pembayaran yang belum terintegrasi dengan sistem perbankan. Sebagai contoh, berdasarkan ketentuan dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024, permohonan sewa wajib dilengkapi dokumen pendukung. Oleh karena itu, fungsi unggah dokumen kini ditambahkan sebagai input wajib pada fitur permohonan. Selain itu, fitur pencarian objek dan tarif telah dilengkapi dengan filter berbasis nama dan kecamatan, guna mendukung pencarian yang lebih terarah.

Sistem pembayaran juga mengalami pembaruan yang signifikan. Jika sebelumnya pengguna harus mengunggah bukti pembayaran secara manual, maka versi hasil reengineering kini menampilkan informasi *virtual account* secara otomatis dan menyesuaikan status pembayaran secara sistemik. Riwayat pembayaran dan tagihan juga ditampilkan dengan informasi yang lebih lengkap dan dapat dilacak, termasuk nomor perjanjian, tanggal pengesahan, dan jangka waktu pembayaran.

Dengan pembaruan ini, perancangan sistem baru tidak hanya berfokus pada aspek fungsional, tetapi juga memperhatikan kebutuhan pengguna dalam hal transparansi proses, kepatuhan terhadap regulasi, dan pengelolaan informasi yang lebih terukur. Berikut adalah pemetaan antara fungsi sistem yang dikembangkan dengan tujuan serta permasalahan pengguna yang ingin diselesaikan. Berikut adalah tabel pemetaan antara fungsi sistem yang dikembangkan dengan tujuan serta permasalahan pengguna yang ingin diselesaikan, disesuaikan dengan hasil *reengineering* aplikasi Tapatupa:

**Tabel 3. 7** Pemetaan Fungsi Sistem Berdasarkan Masalah Pengguna

No	Fungsi Sistem setelah <i>Reengineering</i>	Permasalahan pada Sistem Lama	Tujuan Pengembangan
1	Unggah Dokumen pada Fitur Permohonan	Tidak tersedia fitur unggah dokumen pendukung saat pengajuan permohonan	Memenuhi syarat administratif sesuai <i>Perbup</i> No. 12 Tahun 2024 dan meningkatkan validitas data permohonan
2	Filter Pencarian pada Objek Retribusi	Sulit mencari objek berdasarkan wilayah atau kategori	Mempermudah pencarian objek berdasarkan kecamatan atau nama objek
3	Filter Pencarian pada Tarif Retribusi	Informasi tarif sulit ditemukan secara spesifik	Memberikan kemudahan akses terhadap informasi tarif objek tertentu
4	<i>Link</i> Dokumen pada Halaman Perjanjian	Pengguna tidak dapat langsung melihat dokumen perjanjian yang sah	Menyediakan akses cepat terhadap dokumen perjanjian dalam bentuk digital
5	Tampilan Tagihan Terintegrasi <i>Virtual Account</i>	Sistem belum menampilkan <i>virtual account</i> secara otomatis dan masih mengharuskan unggah bukti bayar	Mengotomatisasi proses pembayaran dan menampilkan status tagihan yang terstruktur dan akurat
6	Riwayat Pembayaran Otomatis	Status pembayaran tidak langsung berubah dan tidak terdokumentasi secara sistematis	Menyediakan pelacakan otomatis pembayaran dengan status terverifikasi secara otomatis oleh sistem

### 3.1.8. *Producing Design Solutions to Meet User Requirements*

Rancangan antarmuka aplikasi Tapatupa dikembangkan menggunakan pendekatan *User-Centered Design* (UCD) yang bersifat iteratif. Pendekatan ini memungkinkan

perbaikan desain dilakukan secara bertahap melalui prototipe yang dikembangkan menggunakan Figma. Desain tersebut dibuat berdasarkan hasil analisis sistem lama, ketentuan dari dokumen Peraturan Bupati Nomor 12 Tahun 2024, serta referensi dari aplikasi sejenis.

Perbaikan desain dilakukan untuk memenuhi kebutuhan pengguna secara nyata, seperti keterpenuhan dokumen wajib pada saat permohonan, keterlacakan status tagihan, serta penyajian informasi yang lebih terstruktur. Komponen antarmuka yang dirancang merujuk pada indikator dalam **sub bab 2.11.4** tentang Desain UI/UX, yang digunakan sebagai acuan dalam merancang ulang tampilan aplikasi. Komponen-komponen tersebut dirangkum dalam Tabel 3.8 berikut.

**Tabel 3. 8** Komponen Antarmuka Pengguna

No	Komponen Antarmuka
1	Komposisi Menu dan Halaman
2	Tampilan Informasi
3	Navigasi
4	Tombol dan ikon
5	Mode input dan pemilihan
6	Kontrol dan konfirmasi

Setiap perbaikan fitur yang dilakukan dalam proses *reengineering* telah dirancang untuk mencerminkan enam komponen utama antarmuka pengguna, mulai dari komposisi halaman hingga 66terati dan konfirmasi, sebagaimana dirinci dalam Tabel 3.9.

**Tabel 3. 9** Perbaikan Fitur

No	Fitur	Perbaikan Utama
1	Permohonan Aset	Penambahan fitur unggah dokumen wajib seperti KTP dan surat permohonan
2	Tagihan	Penambahan informasi nomor <i>Virtual Account</i> otomatis dari Bank daerah

No	Fitur	Perbaikan Utama
3	Pembayaran	Otomatisasi status pembayaran dan pencatatan ke 67terati transaksi
4	Objek Retribusi	Penambahan fitur pencarian berdasarkan kecamatan atau nama objek
5	Tarif Retribusi	Penambahan fitur pencarian berdasarkan nama atau kategori objek
6	Perjanjian	Penambahan tautan ke dokumen digital serta penekanan informasi penting sewa

### 3.1.9. *Evaluating the Designs Against Requirements*

Evaluasi terhadap rancangan sistem dilakukan secara bertahap dengan pendekatan iteratif, guna memastikan bahwa desain antarmuka dan fungsionalitas aplikasi Tapatupa telah sesuai dengan kebutuhan pengguna dan ketentuan regulasi yang berlaku. Evaluasi ini merupakan bagian dari pendekatan *User-Centered Design* (UCD), yang menempatkan pengguna sebagai pusat dari proses perancangan sistem.

Proses validasi desain dilakukan melalui tiga tahap iterasi sebagai berikut:

#### 1 Iterasi Pertama

Fokus pada rancangan awal fitur permohonan, tagihan, dan pembayaran. Evaluasi dilakukan oleh tim pengembang secara internal berdasarkan hasil analisis aplikasi sebelumnya, kebutuhan pengguna, serta dokumen Peraturan Bupati Nomor 12 Tahun 2024. Peninjauan mencakup struktur menu, urutan alur penggunaan, dan elemen antarmuka awal.

#### 2 Iterasi kedua

Revisi dilakukan terhadap desain prototipe berdasarkan identifikasi kendala yang ditemukan pada iterasi sebelumnya, seperti tidak tersedianya filter pencarian pada fitur objek dan tarif retribusi, serta belum adanya kolom unggah dokumen yang dipersyaratkan. Penyesuaian dilakukan agar

antarmuka lebih mendukung proses yang sesuai peraturan dan mudah diakses oleh pengguna.

### 3 Iterasi Ketiga

Dilakukan penyempurnaan desain pada seluruh fitur utama. Desain fitur diperkuat dengan penambahan komponen informasi seperti link dokumen perjanjian, penyusunan iteratif pembayaran yang lebih lengkap, serta *virtual account* otomatis yang ditampilkan secara langsung pada halaman tagihan. Evaluasi iterasi ini bertujuan memastikan bahwa seluruh aspek desain telah mencerminkan kebutuhan pengguna dan siap untuk masuk ke tahap implementasi sistem.

Evaluasi ini memastikan bahwa sistem yang dirancang telah mempertimbangkan berbagai aspek kebutuhan pengguna, konteks penggunaan, dan ketentuan formal, sehingga desain akhir layak untuk diimplementasikan dan diuji pada tahap pengembangan sistem selanjutnya.

### 3.2. Analisis Tahapan *Software Reengineering* terhadap Aplikasi Tapatupa

Proses *software reengineering* pada aplikasi Tapatupa dilakukan secara bertahap dan sistematis untuk meningkatkan kualitas sistem, baik dari sisi struktur, fungsionalitas, maupun pengalaman pengguna. Pendekatan ini mengikuti alur tahapan *reengineering* yang terdiri dari *inventory analysis*, *document restructuring*, *code refactoring*, *data restructuring*, hingga *forward engineering*. Masing-masing tahapan bertujuan mengidentifikasi dan memperbaiki kekurangan sistem lama serta merancang solusi yang lebih sesuai dengan kebutuhan pengguna berdasarkan analisis terdahulu dan regulasi yang berlaku.

**Tabel 3. 10** Analisis Tahapan *Software Reengineering*

No	Tahapan	Perlu / Tidak Perlu	Keterangan
1	<i>Inventory Analysis</i>	Perlu	<i>Inventory Analysis</i> diperlukan dalam <i>reengineering</i> aplikasi Tapatupa yang bertujuan

No	Tahapan	Perlu / Tidak Perlu	Keterangan
			untuk memahami struktur dan komponen yang ada dalam aplikasi sebelumnya. Melalui tahap ini, peneliti dapat mengidentifikasi dengan jelas kebutuhan pengguna yang belum terpenuhi sepenuhnya.
2	<i>Document Restructuring</i>	Perlu	<i>Document Restructuring</i> dilakukan pada dokumentasi yang baru. Bertujuan untuk mengorganisir dokumentasi sebelumnya guna memahami struktur dan detail. Dengan mengidentifikasi komponen-komponen dalam dokumentasi sebelumnya, peneliti dapat melakukan perbaikan dan pembaharuan yang diperlukan.
3	<i>Reverse Engineering</i>	Tidak Perlu	<i>Reverse Engineering</i> tidak diperlukan karena dokumentasi sistem terdahulu (SWTD) masih tersedia dan dapat diakses dengan baik. Spesifikasi dan rancangan sistem sebelumnya tidak hilang atau rusak, meskipun terdapat kebutuhan untuk dilakukan perbaikan dan penyesuaian agar lebih

No	Tahapan	Perlu / Tidak Perlu	Keterangan
			sesuai dengan kebutuhan terkini.
4	<i>Code Refactoring</i>	Perlu	<i>Code Restructuring</i> merupakan tahapan dalam <i>reengineering</i> yang berfokus pada analisis dan restrukturisasi kode sumber dari sistem terdahulu. Pada tahap ini, kode sumber dianalisis untuk memahami fungsi-fungsi utama yang telah ada. Dalam <i>reengineering</i> aplikasi Tapatupa, proses <i>code restructuring</i> diperlukan dengan tujuan memahami struktur kode dan fungsi yang telah berjalan pada aplikasi sebelumnya, melakukan perapihan ( <i>refactoring</i> ) struktur kode agar lebih terorganisir.
5	<i>Data Restructuring</i>	Perlu	Berfokus untuk merapikan struktur data seperti <i>database</i> agar lebih konsisten, terorganisir, dan mudah diakses. Dengan Menyusun ulang dan relasi antar data dapat mempermudah data restructuring.
6	<i>Forward engineering</i>	Perlu	<i>Forward engineering</i> Perlu Terdapatnya penambahan fitur serta kode program dan <i>database</i> yang dibuat dari



No	Tahapan	Perlu / Tidak Perlu	Keterangan
			awal menyebabkan tahapan ini perlu dilakukan. Dibuat perbaikan fitur menggunakan metode Agile Scrum dalam pengembangan yang mencakup desain UI, implementasi fitur, dan pengujian fungsionalitas menggunakan <i>blackbox</i>

### 3.2.1. *Inventory Analysis*

Tahap pertama adalah *inventory analysis* yang bertujuan untuk memahami struktur dan komponen yang ada dalam aplikasi Tapatupa versi awal. Melalui tahap ini, dilakukan identifikasi terhadap fitur-fitur yang sudah tersedia, sekaligus mengevaluasi kekurangan sistem yang belum memenuhi kebutuhan pengguna, seperti tidak tersedianya fungsi unggah dokumen, keterbatasan akses informasi tagihan, serta alur permohonan yang tidak terdokumentasi dengan baik.

Analisis ini merujuk pada studi dokumentasi internal dan pemahaman sistem lama, yang dijelaskan lebih lanjut pada **Subbab 3.1.3**. Dengan pemetaan yang lebih terstruktur, tahap ini menjadi landasan penting untuk menentukan prioritas fitur yang perlu diperbaiki dalam proses *reengineering*.

### 3.2.2. *Document Restructuring*

Setelah melakukan *inventory analysis*, tahap selanjutnya adalah *document restructuring*, yaitu restrukturisasi terhadap dokumentasi sistem. Proses ini bertujuan untuk memperjelas struktur dan detail teknis dari sistem lama, serta menyusun ulang dokumentasi agar lebih terorganisir.

Dokumentasi yang diperbarui mencakup aspek fungsional, alur proses, dan struktur data, serta ditambahkan komponen baru yang sebelumnya tidak terdokumentasi, seperti kebutuhan unggah dokumen sesuai Peraturan Bupati Nomor 12 Tahun 2024. Pembaruan dokumen ini berperan sebagai pedoman dalam merancang sistem baru

dan memastikan keselarasan antara rancangan sistem dengan kebutuhan nyata pengguna dan regulasi.

### **3.2.3. Code Refactoring**

Tahapan *code refactoring* bertujuan untuk memperbaiki struktur internal kode tanpa mengubah perilaku eksternal sistem. *Refactoring* dilakukan berdasarkan hasil identifikasi terhadap bagian kode yang memiliki kompleksitas tinggi, keterbacaan rendah, atau adanya duplikasi logika.

Pada aplikasi Tapatupa, proses *refactoring* difokuskan pada tiga fitur utama, yaitu:

1. Fitur Permohonan Sewa perbaikan logika alur pengajuan serta integrasi fungsi unggah dokumen.
2. Riwayat Permohonan penyederhanaan tampilan dan pemrosesan data riwayat.
3. Detail Perjanjian pengelolaan data perjanjian agar dapat ditampilkan secara lengkap dan relevan.

*Refactoring* dilakukan secara bertahap untuk memastikan sistem tetap stabil selama pengembangan.

### **3.2.4. Data Restructuring**

*Data restructuring* merupakan penataan ulang skema basis data untuk meningkatkan konsistensi, dan keterkelolaan sistem. Pada aplikasi sebelumnya, ditemukan beberapa masalah seperti penamaan kolom yang tidak seragam, hubungan antar entitas yang tidak jelas, serta kurangnya *foreign key* pada tabel penting seperti user, permohonan, dan tagihan.

Dalam tahap ini, dilakukan penyesuaian skema basis data agar mendukung fitur baru yang dirancang, seperti penyimpanan dokumen, pengelompokan 72terati pembayaran, serta integrasi dengan nomor *virtual account*. Penyesuaian ini dilakukan tanpa mengubah isi data secara substansial, tetapi fokus pada perbaikan struktur untuk mendukung sistem baru.

### **3.2.5. Forward Engineering**

*Forward engineering* adalah tahap implementasi dari seluruh hasil analisis, desain ulang, dan perbaikan sistem. Pada tahap ini, peneliti mengembangkan ulang aplikasi

Tapatupa berdasarkan hasil *inventory*, dokumen baru, serta rancangan antarmuka dan fungsionalitas yang telah melalui iterasi desain.

Proses ini meliputi:

1. Implementasi fitur-fitur hasil *reengineering* seperti unggah dokumen, filter pencarian objek retribusi, dan *virtual account* otomatis.
2. Penerapan antarmuka pengguna (UI) yang lebih iteratif dan terstruktur sesuai prinsip desain UCD.
3. Pengujian awal internal terhadap sistem untuk memastikan bahwa fungsi berjalan sesuai spesifikasi.

Tahap ini juga menjadi dasar untuk masuk ke tahap pengujian lebih lanjut, termasuk *User Acceptance Test* (UAT).

### **3.3. Kesimpulan**

Subbab ini menjelaskan tahapan-tahapan software reengineering yang dilakukan terhadap aplikasi Tapatupa, dimulai dari analisis sistem lama hingga tahap implementasi dan UAT. Pendekatan berbasis *User - Centered Design* yang terintegrasi dengan *Agile Scrum* memastikan bahwa setiap langkah pembangunan sistem dilakukan secara iteratif dan sesuai dengan kebutuhan pengguna. Tahapan-tahapan tersebut menjadi pondasi utama dalam menghasilkan aplikasi yang lebih terstruktur, terukur, dan mudah digunakan.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Lingkungan Implementasi

Tahap implementasi merupakan proses penerapan hasil rancangan sistem ke dalam lingkungan nyata atau mendekati kondisi operasional sebenarnya. Lingkungan implementasi meliputi perangkat lunak (*software*) dan perangkat keras (*hardware*) yang digunakan selama proses pengembangan dan pengujian sistem. Tujuan dari tahap ini adalah memastikan bahwa sistem dapat berjalan secara optimal sesuai dengan spesifikasi dan kebutuhan pengguna.

##### 4.1.1. Perangkat Lunak

Perangkat lunak merupakan komponen penting dalam pengembangan dan implementasi sistem. Spesifikasi perangkat lunak yang digunakan dalam pembangunan aplikasi Tapatupa ditampilkan pada Tabel 4.1

**Tabel 4. 1** Spesifikasi Software

Software	Spesifikasi
<i>Operating System</i>	<i>Windows10 Pro 64-Bit</i>
<i>Programming Language</i>	<i>Dart</i>
<i>Web Browser</i>	<i>Google Chrome, Brave</i>

##### 4.1.2. Perangkat Keras

Spesifikasi perangkat keras merujuk pada minimum *requirement* yang dibutuhkan agar aplikasi dapat berjalan dengan lancar selama proses implementasi dan pengujian. Adapun spesifikasi perangkat keras yang digunakan dalam implementasi ditampilkan pada Tabel 4.2.

**Tabel 4. 2** Spesifikasi Hardware

Hardware	Spesifikasi
<i>Type</i>	<i>Asus VivoBook 14</i>
<i>processor</i>	<i>Intel Core i5 with Iris Xe Graphics</i>
<i>Memory</i>	<i>8 GB RAM</i>

## 4.2. Batasan Implementasi

Batasan implementasi merujuk pada ruang lingkup penggunaan sistem berdasarkan target pengguna, platform teknologi, serta fitur yang direkayasa ulang. Penetapan batasan ini penting untuk mengarahkan proses evaluasi dan pengembangan sistem secara terfokus.

Adapun batasan dalam implementasi aplikasi Tapatupa adalah sebagai berikut:

1. Aplikasi dapat digunakan oleh masyarakat kabupaten Tapanuli Utara yang sudah memiliki NPWP.
2. Aplikasi Tapatupa dikembangkan sebagai aplikasi berbasis *mobile*, yang dapat diakses melalui perangkat *smartphone*.
3. Proses *reengineering* difokuskan pada pengembangan ulang antarmuka pengguna, pengelolaan permohonan sewa, dan pengelolaan pembayaran sewa, sesuai kebutuhan pengguna dan hasil evaluasi sistem sebelumnya.
4. Pengujian dilakukan oleh pengguna terbatas, dan tim internal.

## 4.3. Implementasi Sprint

Implementasi pengembangan aplikasi Tapatupa dilakukan secara bertahap melalui tiga *sprint* utama yang disusun berdasarkan pendekatan Scrum yang terintegrasi dengan metode *User-Centered Design* (UCD). Setiap *sprint* mewakili satu iterasi pengembangan, yang mencakup tahapan analisis kebutuhan, desain antarmuka, implementasi fungsionalitas, serta pengujian.

Pembagian *sprint* ini dirancang untuk memastikan bahwa fitur-fitur yang dikembangkan mampu menjawab kebutuhan pengguna secara bertahap dan terukur. Proses pengembangan dimulai dengan pembuatan *Minimum Viable Product* (MVP) pada sprint pertama. Pada iterasi kedua, pengujian sistem dilakukan oleh pihak Badan dan Keuangan Aset Daerah (BKAD) bersama bank daerah melalui kegiatan *User Acceptance Test* (UAT). UAT ini bertujuan untuk menilai kesesuaian sistem dengan kebutuhan pengguna sebenarnya di lingkungan instansi terkait. Tim pengembang hanya bertindak sebagai pihak pengamat dan pencatat umpan balik yang diberikan oleh pengguna eksternal selama proses pengujian berlangsung, serta pada iterasi ketiga diakhiri dengan penyempurnaan sistem dan evaluasi kepuasan pengguna.

Rincian pembagian iterasi pengembangan dibuat lengkap dengan daftar *backlog* per *sprint* yang dikerjakan tim selama periode Februari hingga Juni 2025.

#### 4.3.1. Pembagian Iterasi *Reengineering*

Pembagian iterasi menjelaskan proses pengembangan aplikasi Tapatupa yang dilakukan selama periode Februari hingga Juni 2025. Setiap iterasi merepresentasikan satu *sprint* pengembangan yang mencakup kegiatan analisis kebutuhan, perancangan antarmuka, implementasi fungsionalitas, serta pengujian. Strategi pembagian ini disusun agar pengembangan sistem dapat dilakukan secara bertahap, terukur, dan selaras dengan kebutuhan aktual pengguna di lapangan. Dengan menerapkan pendekatan Scrum yang terintegrasi dengan *User-Centered Design* (UCD), proses *reengineering* aplikasi Tapatupa dibagi ke dalam tiga iterasi utama, yang masing-masing menghasilkan *increment* berupa fitur atau modul yang dapat digunakan dan dievaluasi. Berikut adalah penjelasan masing-masing iterasi secara lebih mendalam.

**Tabel 4. 3** Pembagian Iterasi Reengineering

Iterasi	Periode	Durasi	Kegiatan Utama
Iterasi 1	1 Februari - 15 Maret 2025	±6 minggu	Pengembangan MVP sistem: form permohonan, unggah dokumen, tagihan sewa, dan integrasi <i>virtual account</i> , validasi desain awal dan pengujian <i>black-box</i> internal.
Iterasi 2	16 Maret - 30 April 2025	±6 minggu	Pengembangan prototipe <i>high-fidelity</i> untuk modul objek retribusi, tarif retribusi, perjanjian, autentikasi, dan profil pengguna. Observasi <i>User Acceptance Test</i> (UAT) oleh pihak BKAD dan bank daerah.
Iterasi 3	1 Mei - 30 Juni 2025	±6 minggu	Penyempurnaan antarmuka dan fungsionalitas sistem,

Iterasi	Periode	Durasi	Kegiatan Utama
			dan penyebaran kuesioner untuk evaluasi kepuasan pengguna terhadap hasil <i>reengineering</i>

#### 4.3.2. *Sprint Backlog* per Iterasi

Setelah menetapkan pembagian iterasi dalam pengembangan aplikasi Tapatupa, setiap *sprint* disusun berdasarkan prioritas fitur yang diambil dari *product backlog*. Daftar tugas yang akan dikerjakan dalam satu *sprint* tersebut disebut sebagai *sprint backlog*. *Sprint backlog* berfungsi sebagai panduan bagi tim dalam menyelesaikan pekerjaan yang terfokus, terukur, dan sesuai dengan tujuan iterasi.

Tabel 4.5 ditetapkan pada masing-masing *sprint*, mencerminkan tugas-tugas utama yang dikerjakan selama tiga iterasi pengembangan aplikasi.

**Tabel 4. 4** Sprint Backlog per Iterasi

Iterasi	Sprint Backlog
Iterasi 1	<ol style="list-style-type: none"> <li>1. Implementasi form permohonan</li> <li>2. Fitur unggah dokumen</li> <li>3. Modul tagihan sewa</li> <li>4. Integrasi nomor <i>virtual account</i> (VA)</li> <li>5. Pengujian fungsi <i>black - box</i></li> </ol>
Iterasi 2	<ol style="list-style-type: none"> <li>1. Modul Daftar Objek Retribusi</li> <li>2. Modul Tarif Retribusi</li> <li>3. Modul Perjanjian Sewa</li> <li>4. Modul Autentikasi pengguna</li> <li>5. Pengujian fungsi <i>black - box</i></li> <li>6. Observasi UAT yang dilakukan oleh BKAD dan bank daerah</li> </ol>
Iterasi 3	<ol style="list-style-type: none"> <li>1. Penyempurnaan UI / UX</li> <li>2. Validasi dan perbaikan fungsi</li> <li>3. Penyempurnaan setiap modul secara menyeluruh</li> <li>4. Pengujian <i>usability</i> dan dokumentasi akhir</li> <li>5. Penyebaran kuesioner sebagai evaluasi uji kepuasan pengguna terhadap hasil <i>reengineering</i></li> </ol>

#### 4.4.Tahapan Implementasi

Bagian ini berisi tahapan implementasi *Software Reengineering* yang dirancang untuk meningkatkan kualitas dan efisiensi sistem perangkat lunak. Proses implementasi ini melibatkan serangkaian langkah yang dimulai dari analisis sistem yang ada, identifikasi area perbaikan, pemodelan ulang struktur, hingga implementasi dan pengujian perubahan. Setiap tahapan ini penting untuk memastikan bahwa perangkat lunak yang direkayasa ulang sehingga dapat memenuhi kebutuhan pengguna dengan lebih baik.

##### 4.4.1. Implementasi *Inventory Analysis*

Tahap awal dalam proses *reengineering* dimulai dengan *inventory analysis*, yang dilakukan oleh tim peneliti internal. Implementasi *inventory analysis* melibatkan serangkaian langkah dimulai dari pemahaman mendalam terhadap aplikasi yang sudah ada, dilanjutkan dengan pengumpulan data hingga identifikasi aset perangkat lunak yang memerlukan rekayasa ulang. Hal ini bertujuan untuk memastikan bahwa analisis yang dihasilkan mencakup pemahaman yang mendalam serta aspek-aspek yang memerlukan proses *reengineering*.

##### 4.4.1.1. *Requirement Analysis*

*Requirement analysis* dilakukan melalui studi dokumen Peraturan Bupati No. 12 Tahun 2024, analisis aplikasi versi awal, dan evaluasi aplikasi sejenis guna merumuskan kebutuhan sistem yang spesifik dan terukur.

##### 1. Analisis aplikasi sejenis

Analisis ini bertujuan untuk memahami fitur dan kelemahan dari aplikasi lain yang memiliki fungsi serupa dengan Tapatupa. Beberapa aplikasi yang dianalisis mencakup *website* Siap Sewa dari Jawa Timur, *website* Amanah dari Bandung, dan SIWA aplikasi dari Bali.

##### 2. Analisis Aplikasi Tapatupa Sebelumnya

Evaluasi terhadap sistem Tapatupa versi lama mengungkap beberapa keterbatasan, antara lain:



- a. Pada fitur permohonan, tidak adanya fungsi unggah dokumen permohonan.
- b. Pembayaran yang masih mengharuskan pengguna mengunggah bukti pembayaran secara manual.
- c. Tagihan sewa memiliki tampilan informasi tagihan masih kurang terstruktur dengan baik. Beberapa informasi dasar seperti jumlah tagihan, tanggal jatuh tempo, dan status pembayaran tidak ada ditampilkan dalam riwayat tagihan. Informasi-informasi tersebut seharusnya ditampilkan secara ringkas dan langsung terbaca tanpa perlu menggulir halaman terlalu jauh.

Perubahan yang dilakukan dalam sistem baru mencakup integrasi pembayaran otomatis, fitur unggah dokumen, sistem pemberitahuan pembayaran, serta merancang sistem notifikasi otomatis berbasis status tagihan.

### 3. Analisis Dokumen Peraturan Bupati

Sebagai sistem yang digunakan untuk administrasi retribusi daerah, Tapatupa harus sesuai dengan regulasi yang berlaku. Analisis dilakukan terhadap Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024, yang mengatur tentang:

- a. Tata cara pelaksanaan sewa barang milik daerah berupa tanah dan bangunan.
- b. Tata cara penetapan tarif sewa barang milik daerah berupa tanah dan / atau bangunan.
- c. Jangka waktu sewa barang milik daerah.
- d. Perjanjian sewa.
- e. Pembayaran Sewa.
- f. Perpanjangan Jangka Waktu Sewa.

Dari hasil analisis ini, sistem Tapatupa diperbarui agar seluruh proses bisnisnya sesuai dengan ketentuan yang ditetapkan oleh Peraturan Bupati.

#### 4.4.1.2 Legacy Fitur

*Legacy* Fitur adalah fitur lama dalam sistem yang masih digunakan tetapi direncanakan untuk dioptimalkan atau digantikan dengan fitur yang lebih *modern*. *Legacy* fitur merupakan fitur-fitur yang telah tersedia pada versi awal aplikasi Tapatupa dan masih digunakan sebagai dasar dalam pengembangan ulang sistem. Meskipun fitur-fitur ini sudah pernah diterapkan sebelumnya, dalam proses *reengineering*, fitur tersebut tidak dihilangkan, melainkan dioptimalkan untuk disesuaikan dengan kebutuhan pengguna, standar antarmuka modern, dan ketentuan regulasi terbaru seperti yang tertuang dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024.

Pemanfaatan kembali *legacy* fitur dilakukan dengan berbagai pendekatan, seperti penambahan fungsi baru, perbaikan tampilan antarmuka (UI), integrasi otomatisasi proses, hingga penyempurnaan logika bisnis. Tujuannya adalah untuk menjaga sistem yang telah dikenal oleh pengguna, dengan tetap menghadirkan peningkatan dan kenyamanan penggunaan pada versi hasil *reengineering*.

Analisis terhadap *legacy* fitur menjadi bagian penting dalam tahapan inventory analysis, karena membantu tim pengembang memahami fitur mana yang masih relevan, fitur mana yang perlu diperbaiki, dan bagaimana transisi ke sistem baru dapat dilakukan secara mulus tanpa menghilangkan nilai dari sistem sebelumnya.

**Tabel 4. 5** Daftar Legacy Fitur

Legacy Fitur	Perubahan / Rekayasa Ulang	Keterangan
Halaman Permohonan	Ditingkatkan menjadi form permohonan + unggah dokumen	Tetap digunakan tapi dioptimasi sesuai Perbup
Halaman Objek Retribusi	Ditambah filter pencarian berbasis kecamatan	Fungsi lama diperluas
Halaman Tarif Retribusi	Ditambah filter kategori objek	Antarmuka diperjelas
Halaman Perjanjian	Ditambah <i>link</i> ke dokumen PDF sah	Dioptimalkan untuk transparansi
Halaman Tagihan	Ditambah integrasi <i>virtual account</i>	Diubah menjadi otomatisasi tagihan
Halaman Pembayaran	Ditambah otomatisasi status bayar	Tidak perlu upload bukti manual

Legacy Fitur	Perubahan / Rekayasa Ulang	Keterangan
Beranda	Menampilkan notifikasi dan ringkasan status	Diubah <i>layout</i> -nya, fungsional tetap

#### 4.4.1.3. Komponen Antarmuka

Perancangan ulang antarmuka dalam aplikasi Tapatupa dilakukan berdasarkan hasil analisis terhadap sistem versi awal, standar desain antarmuka yang baik menurut **sub bab 2.11.3**, serta kesesuaian dengan konteks penggunaan aplikasi oleh masyarakat wajib retribusi. Perbaikan ini tidak langsung berasal dari masukan pengguna di awal, namun diturunkan dari Analisa aplikasi versi awal, dokumen regulasi, dan studi terhadap aplikasi sejenis. Komponen - komponen antarmuka yang mengalami penyesuaian dapat dilihat pada Tabel 4.6.

**Tabel 4. 6** Hasil Perbaikan Antarmuka Pengguna

No	Komponen Antarmuka	Kondisi Sebelumnya	Perbaikan Rancangan	Keterangan
1	Komposisi Menu dan Halaman	Menu utama ditampilkan dua kali, di bagian atas (ikon) dan bawah (navbar), serta membuat sapaan pengguna dan tulisan ajakan membayar retribusi.	Menu atas dikurangi dan difokuskan, sapaan pengguna dan ajakan dihapus.	Untuk menyederhanakan dan memfokuskan tampilan agar lebih mudah digunakan dan dimengerti oleh pengguna.
2	Tampilan Informasi	Informasi tagihan ditampilkan secara menonjol di beranda, dengan teks berwarna merah dan format jumlah tagihan.	Diganti menjadi informasi status permohonan, ditampilkan lebih ringkas dan fokus.	Fokus informasi digeser ke pelacakan proses permohonan, yang lebih sesuai untuk pengguna awal.
3	Navigasi	Navigasi atas dan bawah mengandung menu yang sama sehingga membingungkan.	<i>Navbar</i> dikurangi pada bagian atas.	Penghapusan redundansi dan penyederhanaan akses menu.

No	Komponen Antarmuka	Kondisi Sebelumnya	Perbaikan Rancangan	Keterangan
4	Tombol dan ikon	Terdapat ikon menu dalam bentuk kotak yang tidak sama sehingga membingungkan	Ikon diperbaharui lebih sederhana dan disesuaikan dengan fungsi minimalis.	Untuk memberikan tampilan yang lebih bersih.
5	Mode input dan pemilihan	Tidak ditampilkan pada halaman beranda	Tidak ada perubahan	Mode input hanya muncul pada halaman permohonan atau pembayaran.
6	Kontrol dan konfirmasi	Belum ditampilkan eksplisit pada beranda.	Diperjelas pada bagian status permohonan dengan status proses	Memberikan kepastian dan kontrol informasi kepada pengguna.

#### 4.4.2. Implementasi *Document Restructuring*

Pada implementasi ini peneliti akan melakukan dokumentasi ulang. Peneliti melakukan restrukturisasi pada dokumen *Software Technical Document* (SWTD) yang kemudian akan diperbaiki dan dibuat dalam bentuk Dokumen Penelitian Produk (DPP). Restrukturisasi dilakukan setelah mengidentifikasi perubahan yang diperlukan berdasarkan temuan dari *inventory analysis* sebelumnya. Proses ini melibatkan pembaharuan atau penggantian dokumen sesuai dengan rencana perubahan. Perubahan yang dilakukan oleh peneliti mencakup penyesuaian pada bagian yang signifikan ataupun penambahan dari fungsi sistem. Dokumen teknis yang akan dilakukan perubahan yaitu dokumen aplikasi Tapatupa berbasis *mobile*. Restrukturisasi dilakukan kepada konten maupun format dari dokumen sebelumnya.

**Tabel 4. 7** Aspek Dokumen Restrukturisasi

Aspek	Deskripsi
Peneliti Awal	<ul style="list-style-type: none"> <li>- Ardiles Sinaga</li> <li>- Istas Pratomo Manalu</li> <li>- Frengki Simatupang</li> </ul>

Aspek	Deskripsi
	<ul style="list-style-type: none"> <li>- Johannes Sipayung</li> <li>- Samuel Volder</li> </ul>
Modul Awal	Permohonan, Objek Retribusi, Tarif objek, Perjanjian, Tagihan, Pembayaran, Tagihan
Peneliti Lanjutan	<ul style="list-style-type: none"> <li>- Gabriel Ignatius Situmeang</li> <li>- Anita Luciana Munthe</li> <li>- Yudi Fradianto Saragih</li> </ul>
Modul Tambahan	<i>Login</i> , Autentikasi, unggah dokumen pada permohonan sewa, integrasi nomor <i>virtual account</i> , filter pencarian objek & tarif, status riwayat pembayaran otomatis, tautan dokumen perjanjian,
Dokumentasi yang tersedia	<i>Software Technical Document</i> Dokumen User Manual
Masalah yang Ditemukan	Kurangnya informasi detail mengenai struktur, logika, dan fungsi dari perangkat lunak.
Langkah yang dilakukan	Melakukan <i>document restructuring</i> untuk menghasilkan dokumen penelitian produk.

Berikut hasil aspek dokumen restrukturisasi yang dapat dilihat pada Tabel 4.8.

**Tabel 4. 8** Hasil Aspek Dokumen Restrukturisasi

Aspek Restrukturisasi	Alasan
Reorganisasi Bab/ Subbab	Memindahkan bab atau subbab untuk mengikuti alur yang lebih baik.
Penambahan atau Penghapusan Konten	Menambahkan informasi baru atau menghapus konten yang tidak relevan.
Pemecahan dan Penggabungan Konten	Membagi bagian yang terlalu panjang menjadi bagian-bagian yang lebih kecil dan menggabungkan beberapa bagian yang terkait untuk mengurangi redundansi.
Penambahan Elemen Visual	Menambahkan tabel, grafik, atau gambar untuk membantu menyajikan informasi.
Penyusunan Ulang Daftar Isi	Mengupdate daftar isi agar sesuai dengan perubahan struktur konten.
Penyusunan Ulang Dokumentasi <i>Business Process Model</i> .	Dokumentasi proses bisnis lama diperbaharui menggunakan standar BPMN ( <i>Business Process Model and Notation</i> ), sehingga alur proses menjadi

Aspek Restrukturisasi	Alasan
	lebih jelas, sistematis, dan mudah dipahami oleh tim peneliti dan <i>stakeholder</i> .
Penggambaran Ulang <i>User Flow Diagram</i>	Alur interaksi pengguna ( <i>user flow</i> ) diperbaharui dengan membuat skenario berbasis <i>use case</i> untuk semua fitur utama. Hal ini bertujuan untuk memastikan setiap langkah pengguna dalam aplikasi terekam dengan baik dan mendukung pengalaman pengguna yang lebih baik.

#### 4.4.3. Implementasi *Code Refactoring*

Pada tahap ini dilakukan upaya penyempurnaan terhadap struktur kode sumber aplikasi Tapatupa melalui proses *refactoring*. *Refactoring* dipilih sebagai pendekatan karena bertujuan memperbaiki kualitas internal perangkat lunak tanpa mengubah fungsionalitas yang telah berjalan. Fokus utama dalam tahap ini adalah menyederhanakan logika, memisahkan tanggung jawab modul, serta meningkatkan keterbacaan dan kesiapan sistem untuk dikembangkan di masa mendatang, untuk melihat penggunaan *SonarQube* dengan *code* dapat dilihat pada **lampiran sub bab 4.3.3.**

Pada tahap ini, dilakukan *refactoring* kode terhadap tiga fitur utama dalam aplikasi Tapatupa, yaitu Permohonan, Tagihan, dan Perjanjian. Evaluasi dilakukan menggunakan alat bantu *SonarQube*, yang menghasilkan berbagai metrik kuantitatif seperti *lines of code*, *maintainability*, *reliability issues*, *cyclomatic complexity*, dan *duplication density*. Namun, dalam praktiknya, hasil *refactoring* tidak sepenuhnya selaras dengan harapan sebelumnya. Beberapa metrik justru menunjukkan peningkatan angka yang secara sekilas tampak negatif. Hal ini menjadi bahan pertimbangan yang penting dalam menilai ulang dari proses *refactoring* itu sendiri bahwa perbaikan kualitas sistem tidak selalu tercermin dalam perbaikan angka semata. Namun demikian, hasil analisis menunjukkan bahwa tidak semua indikator metrik mengalami perbaikan kuantitatif secara langsung. Hal ini menandakan bahwa *refactoring* memiliki nilai strategis jangka panjang, yang tidak sepenuhnya tercermin dari metrik statik seperti jumlah baris kode atau persentase duplikasi.

#### 4.4.3.1. Fitur Permohonan

Refactoring pada fitur Permohonan dilakukan dengan memecah logika dari dua *file* utama (permohonan.dart dan buat - permohonan.dart) menjadi tiga *file* yang memiliki tanggung jawab lebih spesifik: permohonan.dart untuk daftar, buat-permohonan-baru.dart untuk form permohonan, dan detail-permohonan.dart untuk detail pengajuan.

*Refactoring* pada fitur ini melibatkan pemisahan modul menjadi tiga *file* yang berbeda, sesuai prinsip *Single Responsibility*. Penambahan duplikasi dan penurunan *maintainability index* terjadi karena kode yang sebelumnya tersentralisasi harus dituliskan ulang secara modular di masing-masing *file*. Meskipun secara metrik terjadi peningkatan duplikasi, dari sisi struktur dan pemeliharaan, kode menjadi lebih terbaca dan dapat diuji secara terpisah.

**Tabel 4. 9** Evaluasi Refactoring Fitur Permohonan

Metrik	Sebelum refactoring	Sesudah Refactoring
<i>Lines of Codes</i> (LOC)	410	615
<i>Reliability Issues</i>	1	6
<i>Maintainability Score</i>	137	119
<i>Duplications</i>	0.0%	25.5%

1. *Lines of Code* adalah jumlah baris kode yang ditulis dalam suatu modul atau program. Metrik ini digunakan untuk mengukur kompleksitas dan skala suatu kode. Pada aplikasi Tapatupa, LOC mengalami peningkatan dari 410 menjadi 615 baris kode. Hal ini terjadi karena pada saat *refactoring* dilakukan penambahan fitur baru, pemecahan fungsi besar menjadi fungsi-fungsi kecil yang lebih modular, serta penambahan komentar dan struktur kode yang lebih terorganisir.
2. *Reliability Issues* merujuk pada potensi kesalahan (*bugs*) atau titik-titik kegagalan yang dapat mempengaruhi keandalan program. Dalam hal evaluasi, jumlah nya meningkat dari 1 menjadi 6. Kenaikan ini disebabkan oleh kompleksitas kode yang bertambah setelah pengembangan fitur baru. Namun demikian, reliabilitas ini

bersifat sementara dan umum terjadi dalam pengembangan fitur baru, dan dapat diperbaiki lebih lanjut pada tahap optimasi dan pengujian lanjutan.

3. *Maintainability Score* adalah indikator seberapa mudah kode diperlihara, diperbaiki, atau dikembangkan Kembali. Setelah *refactoring*, skor mengalami penurunan dari 137 menjadi 119. Nilai ini menunjukkan bahwa walaupun struktur kode sudah ditata ulang, peningkatan jumlah baris kode dan kompleksitas logika turut mempengaruhi tingkat kemudahan pemeliharaan. Skor ini dapat ditingkatkan kembali melalui dokumentasi lebih detail dan pengurangan redundansi pada iterasi berikutnya.
4. *Code Duplication* menunjukkan persentase bagian kode yang ditulis ulang secara identik dari 0% menjadi 25.5%. Persentase ini menunjukkan beberapa fungsi baru memiliki pola atau logika serupa. Pada fase pengembangan berikutnya, tim pengembang perlu melakukan optimalisasi ulang untuk mengurangi duplikasi dan meningkatkan efisiensi struktur program.

Secara keseluruhan, metrik tersebut menunjukkan bahwa proses *refactoring* tidak hanya mencakup perapihan kode, tetapi juga berdampak pada struktur program secara menyeluruh karena adanya pengembangan dan penambahan fitur baru. Evaluasi lanjutan dan iteratif tetap diperlukan agar kualitas kode tetap terjaga dalam jangka panjang.

#### **4.4.3.2. Fitur Tagihan**

Untuk fitur Tagihan, *refactoring* dilakukan tanpa perubahan jumlah *file*, namun fokus pada penataan ulang struktur kode di dalam *file* tagihan-baru.dart, detail - sewa. dart, dan pembayaran.dart. Beberapa fungsi validasi dan alur pembayaran yang sebelumnya bercampur dengan tampilan, kini dipisahkan menjadi fungsi terstruktur.

*Refactoring* pada fitur Tagihan difokuskan pada pemisahan logika visual, validasi pembayaran, serta penyusunan ulang struktur kontrol. Peningkatan duplikasi yang signifikan terjadi karena pola logika pembayaran yang mirip perlu digunakan di



beberapa tempat (misalnya verifikasi status sewa). Namun, secara struktural, pemisahan ini membuat fungsi-fungsi menjadi lebih dapat diuji dan dikelola secara terpisah.

**Tabel 4. 10** Evaluasi Refactoring Fitur Tagihan

Metrik	Sebelum Refactoring	Sesudah Refactoring
LOC -file tagihan-baru.dart	195	225
<i>Duplications</i> - tagihan-baru.dart	41.9%	89.5%
LOC - detail -sewa. dart	305	478
<i>Duplications</i> - detail -sewa. dart	11.9%	98.5%
LOC - pembayaran.dart	207	353
<i>Duplications</i> - pembayaran.dart	0.0%	Tidak berubah (0.0%)

1. File detail tagihan - baru.dart, pada file LOC naik dari 195 menjadi 225 LOC (*Line of Code*) adalah jumlah baris kode dalam sebuah file, menunjukkan adanya penambahan logika / fungsi baru. Namun, *duplications* juga meningkat dari 41.9% menjadi 89.5% yang artinya banyak bagian kode yang perlu di evaluasi lanjutan.
2. File detail - sewa.dart, LOC bertambah dari 305 menjadi 478 dan *duplications* meningkat sehingga menunjukkan perlu *refactoring* lebih lanjut.
3. File pembayaran.dart, LOC meningkatkan dari 207 menjadi 353 karena penambahan fitur, tetapi *duplications* nya tetap 0.0%, yang artinya tidak ada bagian kode yang terduplikasi dan struktur kode masih efisien.

#### 4.4.3.3. Fitur Perjanjian

*Refactoring* dilakukan pada dua *file* yaitu perjanjian.dart dan detail-perjanjian.dart. Beberapa fungsi digabung atau dipecah ulang untuk mengurangi kedalaman logika dan meningkatkan keterbacaan.

Hasil ini menunjukkan perbedaan yang kontras. Pada perjanjian.dart, duplikasi meningkat karena logika dipecah menjadi beberapa blok eksplisit. Sebaliknya, pada

detail-perjanjian.dart, *refactoring* berhasil menghilangkan seluruh duplikasi yang ada. Penurunan *maintainability* pada *file* utama tidak sepenuhnya negatif, karena justru membuat fungsionalitas lebih terfokus.

**Tabel 4. 11** Evaluasi Refactoring Fitur Perjanjian

Metrik	Sebelum Refactoring	Sesudah Refactoring
LOC - perjanjian.dart	216	209
<i>Duplications</i> - perjanjian.dart	36,2%	87.0%
<i>Maintainability</i> - perjanjian.dart	158	102
LOC - detail - perjanjian.dart	249	178
<i>Duplications</i> - detail - perjanjian.dart	15,8%	0.0%

Untuk melihat gambaran secara umum, berikut adalah Tabel 4.13 yang merupakan rekapitulasi perubahan metrik utama dari seluruh sistem:

**Tabel 4. 12** Ringkasan Evaluasi Keseluruhan Sistem

Metrik	Sebelum Refactoring	Sesudah Refactoring
Jumlah <i>Issues</i>	11	73
<i>Maintainability Issues</i>	1,746	3,510
<i>Technical Debt</i>	5d 7h	12d
<i>Debt Ratio</i>	2.4%	2.1%
<i>Duplication Density</i>	18.2%	51.9%
<i>Duplicated Lines</i>	1,154	7,948
<i>Duplicated Blocks</i>	27	22
<i>Duplicate Files</i>	9	20
<i>Total LOC</i>	3,954	9,718
<i>Total Lines</i>	6,358	15,304
<i>Total Files</i>	24	40
<i>Comment Lines</i>	482	1,271
<i>Comment Density</i>	10.9%	11.6%
<i>Cyclomatic Complexity</i>	120	243
<i>Open Code Issues (Total)</i>	1,757	3,583

Dari tabel di atas, terlihat bahwa ukuran sistem meningkat hampir dua kali lipat. Hal ini disebabkan oleh pemecahan logika yang sebelumnya tersentral menjadi

lebih modular. *Technical debt* bertambah dalam angka mutlak (nyata), namun rasio beban terhadap ukuran sistem justru menurun. Dengan kata lain, sistem kini memiliki skala yang lebih besar, tetapi relatif lebih efisien dan jelas dari sisi struktur.

Melalui hasil ini, menjadi jelas bahwa *refactoring* tidak selalu identik dengan perbaikan angka. Pada kondisi tertentu, *refactoring* justru menambah duplikasi atau kompleksitas secara metrik, namun mengurangi beban kognitif pengembang karena struktur kode yang lebih jelas dan terpisah.

Oleh sebab itu, keberhasilan *refactoring* dalam penelitian ini tidak hanya dinilai dari hasil analisis otomatis seperti *SonarQube*, melainkan juga dari perubahan positif dalam struktur, keterbacaan, modularitas, dan kesiapan untuk pengembangan lebih lanjut.

*Refactoring* dianggap berhasil apabila kode menjadi lebih mudah dipahami, diuji, dan dimodifikasi, meskipun konsekuensinya adalah meningkatnya baris kode atau kemunculan pola yang mirip antar *file*.

#### 4.4.4. Implementasi Data Restructuring

Setelah dilakukan analisis struktur data seperti *database* pada aplikasi Tapatupa, ditemukan beberapa permasalahan yang perlu diperbaiki, seperti penamaan kolom yang tidak konsisten ada beberapa menggunakan campuran huruf besar, dan kecil, kemudian ada beberapa tabel yang tidak saling terhubung, sehingga dilakukan implementasi data *restructuring* dengan langkah sebagai berikut.

1. Model proses bisnis lebih terstruktur menggunakan BPMN dapat dilihat pada **Lampiran sub bab 3.2**.
2. Desain *database* menjadi lebih konsisten dan mendukung penelitian siklus baru.
3. Alur pengguna dipetakan lebih detail dengan membuat *scenario* berbasis *use case*.

**Tabel 4. 13** Hasil Aspek Data Restrukturisasi

Aktivitas	Penjelasan
Pembaruan Dokumentasi <i>Database Schema</i> .	Struktur <i>database</i> sebelumnya dianalisis dan disusun ulang. Proses ini menghasilkan <i>Entity Relationship Diagram</i> (ERD) yang diperjelas, lalu dikembangkan menjadi <i>Conceptual Data Model</i> (CDM) dan <i>Physical Data Model</i> (PDM) untuk mendukung penelitian <i>database</i> baru yang lebih terstruktur
Penyesuaian Struktur Data agar Mendukung setiap fitur	Memastikan bahwa data yang disimpan dalam sistem baru memiliki struktur yang optimal untuk mendukung fitur seperti penambahan atau perubahan pada setiap <i>field</i>

#### 4.4.5. Implementasi *Forward Engineering*

*Forward engineering* merupakan tahap realisasi sistem berdasarkan hasil analisis kebutuhan dan perancangan desain yang telah ditentukan pada tahapan sebelumnya. Pada proses ini, sistem dibangun dari awal menggunakan pendekatan Scrum yang terintegrasi dengan *User-Centered Design* (UCD). Setiap iterasi menghasilkan *increment* yang diuji melalui proses *regression testing* internal untuk memastikan stabilitas dan fungsi sistem, serta *User Acceptance Testing* (UAT) khusus pada fitur pembayaran.

*Forward engineering* dibagi ke dalam tiga iterasi utama yang masing-masing mencakup pengembangan fungsional, implementasi antarmuka, dan validasi sistem.

##### 4.4.5.1. Hasil Iterasi 1

Pada iterasi pertama, proses implementasi berfokus pada pengembangan *minimum viable product* (MVP) dari aplikasi Tapatupa. Tujuan utama dari iterasi ini adalah membangun pondasi awal sistem yang mencakup fitur yang dibutuhkan oleh pengguna akhir, yaitu masyarakat wajib retribusi. Setelah proses implementasi, dilakukan pengujian regresi (*regression testing*) terhadap seluruh fitur yang dikembangkan.

#### 4.4.5.1.1. Hasil Implementasi Fungsi Iterasi 1

Bagian ini dijelaskan implementasi fungsi yang dikembangkan dalam iterasi pertama dalam aplikasi Tapatupa. Implementasi fungsi ini mencakup fitur - fitur utama yang memungkinkan penyewa mengakses aplikasi Tapatupa. Fokus iterasi pertama adalah membangun *minimum viable product* (MVP) yang mencakup fitur-fitur utama yang menjadi dasar proses penyewaan aset. Modul yang dikembangkan dapat dilihat pada Tabel 4.15.

**Tabel 4. 14** Implementasi fungsi iterasi I

No	Nama Fungsi	Aplikasi
1.	Form Permohonan Sewa	Aplikasi
2.	Validasi Permohonan dan Dokumen	Aplikasi
3.	Unggah Dokumen Permohonan	Aplikasi
4.	Menu Riwayat Permohonan	Aplikasi
5.	Modul Tagihan Sewa	Aplikasi
6.	Modul Pembayaran dengan <i>virtual account</i>	Aplikasi

#### 4.4.5.1.2. Hasil Implementasi Pengujian Iterasi 1

Pengujian dilakukan melalui metode *regression testing* untuk memastikan bahwa fungsi-fungsi yang diimplementasikan berjalan sesuai dengan skenario yang dirancang. Pengujian dilakukan dan hasilnya menunjukkan bahwa semua fungsi diterima (*passed*). Detail skenario pengujian disajikan pada **Lampiran 5.2.1.1.**

**Tabel 4. 15** Pengujian Implementasi Fungsi Iterasi I

No	Butir Pengujian	Hasil Pengujian
1.	Pengujian mengakses menu riwayat permohonan	Diterima
2.	Pengujian Validasi Tampilan Detail Permohonan	Diterima
3.	Pengisian dan Pengajuan Form Permohonan	Diterima
4.	Pengujian form permohonan jika field Jenis Permohonan tidak diisi	Diterima
5.	Pengujian form permohonan jika dokumen tidak diunggah	Diterima

No	Butir Pengujian	Hasil Pengujian
6.	Pengujian akses halaman tagihan sewa	Diterima
7.	Pengujian klik detail tagihan sewa	Diterima
8.	Pengujian pembayaran tagihan dengan VA aktif - Pembayaran Melalui Teller	Diterima
9.	Pengujian pembayaran tagihan dengan VA aktif - Pembayaran Melalui ATM	Diterima
10.	Pengujian pembayaran tagihan dengan VA aktif - Pembayaran Melalui Mobile Banking	Diterima
11.	Pengujian Kode Bayar tidak ditemukan / sudah dibayarkan	Diterima
12.	Pengujian Saldo Tidak Cukup Saat Pembayaran melalui Teller	Diterima

#### 4.4.5.2. Hasil Iterasi 2

Pada bagian ini dijelaskan hasil dari implementasi tahap iterasi II dari rangkaian proses *reengineering* yang sudah dilakukan.

##### 4.4.5.2.1. Desain Antarmuka

Pada iterasi ini, dilakukan perumusan dan implementasi UI *Styleguide* yang bertujuan menyelaraskan elemen antarmuka pengguna. Styleguide ini menjadi acuan dalam pembuatan desain *high-fidelity* yang diterapkan pada fitur-fitur baru yang dikembangkan. Desain antarmuka dirancang untuk meningkatkan konsistensi dan kenyamanan interaksi pengguna. Desain *styleguide* dan antarmuka dapat dilihat pada lampiran sub bab 4.3.5.2.

##### 4.4.5.2.2. Hasil Implementasi Fungsi Iterasi 2

Pada bagian ini dijelaskan kembali implementasi fungsi yang dikembangkan dalam iterasi kedua dalam aplikasi Tapatupa, perbaikan diterima kemudian diimplementasikan dalam bentuk fungsi yang mencakup fitur yang berhubungan dengan informasi objek retribusi dan perjanjian penyewaan. Fungsi - fungsi yang telah diimplementasikan dalam iterasi kedua dapat dilihat pada Tabel 4.17.

**Tabel 4. 16** Implementasi Fungsi Iterasi II

No.	Nama Fungsi	Aplikasi
1.	Modul Objek Retribusi	Aplikasi
2.	Modul Pencarian & Filter Objek Retribusi	Aplikasi
3.	Modul Tarif Retribusi	Aplikasi
4.	Modul Perjanjian Sewa	Aplikasi
5.	Modul Autentikasi ( <i>Login</i> )	Aplikasi
6.	Modul Profil Pengguna	Aplikasi

#### 4.4.5.2.3. Hasil Implementasi Pengujian Iterasi 2

Pengujian pada iterasi ini dilakukan secara menyeluruh untuk memastikan bahwa seluruh fungsionalitas baru yang ditambahkan tidak mengganggu kestabilan sistem sebelumnya. Pengujian dilakukan melalui dua pendekatan utama. Pertama, dilakukan *regression testing* terhadap semua modul, baik yang baru dikembangkan maupun yang telah ada sejak iterasi sebelumnya. Tujuan dari pengujian ini adalah untuk memastikan bahwa perubahan atau penambahan fitur tidak menimbulkan regresi atau kerusakan pada fungsionalitas yang telah berjalan stabil. Kedua, dilakukan *User Acceptance Testing* (UAT) oleh pihak BKAD dan banj daerah khusus untuk fitur integrasi pembayaran menggunakan *virtual account*. UAT ini bertujuan mengevaluasi apakah sistem dapat menghasilkan nomor VA secara otomatis dan mampu memverifikasi status pembayaran secara *real-time*. Berdasarkan hasil pengujian tersebut, seluruh fitur yang dikembangkan pada iterasi kedua dinyatakan berfungsi dengan baik pada modul baru maupun pada modul yang telah dikembangkan sebelumnya. Hasil ini menjadi dasar untuk melanjutkan ke tahap penyempurnaan sistem pada iterasi berikutnya. Adapun hasil pengujian pada iterasi II dapat dilihat pada Tabel 4.18. pengujian skenario lebih rinci dapat dilihat pada lampiran sub bab 5.2.1.2.

Tabel 4. 17 Pengujian Implementasi Fungsi Iterasi II

No	Butir Pengujian	Hasil Pengujian
1.	Pengujian akses daftar Objek Retribusi	Diterima
2.	Pengujian Filter Objek Retribusi dengan Kata Kunci Tidak Sesuai	Diterima
3.	Pengujian Pencarian Objek dengan Karakter tidak bermakna	Diterima
4.	Pengujian akses tarif objek retribusi	Diterima
5.	Pengujian akses perjanjian sewa untuk wajib retribusi	Diterima
6.	Pengujian Validasi Tampilan Detail Perjanjian	Diterima
7.	Pengujian akses profil	Diterima
8.	Pengujian login Berhasil ( <i>Username</i> dan Password <i>Valid</i> )	Diterima
9.	Pengujian <i>login</i> Gagal ( <i>Password</i> salah)	Diterima

#### 4.4.5.3. Hasil Iterasi 3

Pada bagian ini dijelaskan hasil dari implementasi tahap iterasi III dari rangkaian proses *reengineering* yang sudah dilakukan.

##### 4.4.5.3.1. Penyempurnaan Fungsi dan Antarmuka

Iterasi ketiga berfokus pada perbaikan dan penyempurnaan sistem berdasarkan hasil pengujian sebelumnya. Penyesuaian dilakukan terhadap alur interaksi, struktur informasi, dan konsistensi visual antarmuka. Seluruh modul diperiksa kembali agar antar halaman dan fungsi saling terintegrasi secara konsisten.

##### 4.4.5.3.2. Hasil Implementasi Pengujian Iterasi 3

*Regression testing* kembali dilakukan pada seluruh modul aplikasi, termasuk modul yang telah dikembangkan pada iterasi sebelumnya. Pengujian ini bertujuan untuk memastikan bahwa seluruh sistem dapat berfungsi dengan baik secara keseluruhan tanpa konflik baru. Adapun hasil pengujian pada iterasi III dapat dilihat pada lampiran sub bab 5.2.1.3.



#### 4.4.5.3.3. Evaluasi Kepuasan Pengguna Melalui kuesioner

Pada tahap akhir iterasi ketiga, dilakukan evaluasi sistem melalui penyebaran kuesioner kepuasan pengguna. Tujuan utama evaluasi ini adalah menilai tingkat keberhasilan *reengineering* aplikasi Tapatupa dari perspektif pengguna.

Evaluasi dilakukan menggunakan *custom usability questionnaire* dengan dua indikator utama:

1. Tampilan antarmuka (UI) berdasarkan desain yang diadaptasi dapat dilihat pada **sub bab 2.11.3.** mencakup aspek visual *hierarchy*, navigasi, ikon, *input*, dan *layout*.
2. Fungsionalitas sistem berdasarkan standar ISO/IEC 25010 dapat dilihat pada **sub bab 2.10.** khususnya pada subkarakteristik seperti *functional completeness*, *functional correctness*, dan *functional appropriateness*.

Kuesioner disebarkan kepada 84 responden dengan skala likert 1 - 5. Hasilnya dianalisis untuk mengetahui tingkat kepuasan terhadap aplikasi yang telah dikembangkan. Untuk mendapatkan hasil kepuasan aplikasi Tapatupa setelah di *reengineering* dari penelitian ini, maka pengukuran secara keseluruhan dapat diperhitungkan dengan menggunakan **persamaan (2.1)** didapatkan hasil sebesar 79.17%, menurut tabel 2.5 angka tersebut termasuk kedalam kategori yang layak. Untuk melihat rincian lengkap dari daftar pertanyaan, distribusi skor, serta hasil rekapitulasi penilaian tiap butir pernyataan, dapat dilihat pada **lampiran sub bab 5.3.4.**

## BAB V

### KESIMPULAN DAN SARAN

Bagian ini berisi kesimpulan yang diperoleh dari hasil pelaksanaan penelitian ini serta saran yang diberikan berdasarkan temuan dari hasil yang diperoleh untuk bisa ditingkatkan pada penelitian selanjutnya.

#### 5.1. Kesimpulan

Berdasarkan hasil analisis, implementasi, dan evaluasi terhadap proses perancangan ulang (*reengineering*) aplikasi Tapatupa berbasis *mobile* dalam pengelolaan aset tanah, dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini telah berhasil menerapkan lima tahapan *reengineering*, yaitu *inventory analysis*, *document restructuring*, *code refactoring*, *data restructuring*, dan *forward engineering*. Proses ini menghasilkan sistem dengan struktur yang lebih baik, dokumentasi yang lebih jelas, serta kode program dan pengelolaan data yang lebih mudah dikembangkan.
2. Proses *reengineering* berhasil meningkatkan kualitas sistem, terbukti dari hasil kuesioner yang menunjukkan nilai *usability* sebesar 79.19% yang menunjukkan bahwa hasil pengukuran pada aplikasi Tapatupa setelah di *reengineering* memiliki nilai yang layak.
3. Dibandingkan dengan versi sebelumnya, sistem hasil *reengineering* memiliki cakupan fitur yang lebih lengkap, struktur antarmuka yang disederhanakan, serta alur permohonan dan pembayaran yang lebih terintegrasi. Hal ini menunjukkan adanya perbaikan pada aspek kemudahan penggunaan dan fungsionalitas.

#### 5.2. Saran

Berdasarkan hasil dan temuan selama proses pengembangan, berikut beberapa saran yang dapat dipertimbangkan untuk pengembangan selanjutnya:

1. Meskipun pengujian telah dilakukan melalui UAT dan evaluasi kepuasan pengguna, disarankan agar pengujian sistem ke depan dilakukan secara menyeluruh, termasuk regresi testing.
2. Proses *code refactoring* yang dilakukan dalam penelitian ini menunjukkan adanya duplikasi dan penurunan skor *maintainability*. Oleh karena itu,

disarankan agar proses *refactoring* dilakukan lebih terstruktur dengan memperhatikan prinsip *clean code* dan modularisasi fungsi agar sistem lebih mudah dikembangkan di masa mendatang.

3. Untuk pengembangan selanjutnya, pelibatan pengguna secara langsung melalui uji coba dan observasi dianjurkan agar proses iterasi pengembangan tetap sesuai kebutuhan lapangan.

## DAFTAR PUSTAKA

- [1] Aryanto and T. Santoso, "Pengelolaan Aset Pemerintah Dengan Perspektif Administrasi Jakarta Barat Tahun 2021," *J. Ilmu Adm. Publik*, vol. 2, no. 2, pp. 176-188, 2022, [Online]. Available: <https://ojs.stiami.ac.id/index.php/JUMAIP/article/view/2915/1356>
- [2] B. Priyono, "Pengelolaan Aset Daerah Berupa Tanah Dan Bangunan Sebagai Salah Satu Sumber Pendapatan Asli Daerah Pemerintah Kota Depok Berdasarkan Peraturan Menteri Dalam Negeri Nomor 9 Tahun 2009 Tentang Perubahan Atas Peraturan Daerah Kota Depok Nomor 14 Tahun 2013 T," vol. 5, no. 2, pp. 71-82, 2023.
- [3] Melly Nur Qurani and Maulidah Narastri, "Analisis Optimalisasi Pemanfaatan Aset Daerah (Tanah) Untuk Meningkatkan PAD di Kabupaten Sampang," *J. Ris. Akunt.*, vol. 2, no. 1, pp. 72-81, 2023, doi: 10.54066/jura-itb.v2i1.1319.
- [4] "Peraturan Bupati Tentang Pedoman Pemanfaatan Barang Milik Daerah.pdf."
- [5] G. F. Dirgantara and D. M. Driptufany, "The Management of Land Assets in Dharmasraya Regency Based on WebGIS," *South East Asian J. Adv. Eng. Technol.*, vol. 1, no. 1, pp. 10-14, 2023, doi: 10.62447/sejaet.v1i1.2.
- [6] J. Nur, M. I. Mattalitti, R. G. Ahmad, P. K. Kendari, and U. M. Kendari, "TATA KELOLA ASET BARANG MILIK DAERAH BERBASIS E-BMD PADA," vol. 7, no. 3, pp. 1752-1766, 2024.
- [7] S. Hasanah, E. Sutisman, and S. Ponto, "Analysis the Effect Legal Audit, Control and Supervision on The Optimization of Fixed Asset Management (Land and Buildings) In the Regional Government of Papua Province," *West Sci. Soc. Humanit. Stud.*, vol. 1, no. 06, 2023, doi: 10.58812/wsshs.v1i06.515.
- [8] A. Sinaga, "Analysis and Design of Regional Asset Leasing Management Information System in North Tapanuli Regency," vol. 5, no. 4, pp. 102-110, 2024.
- [9] R. Fajriah and S. Nazar, "Analisa Business Process Reengineering Dalam Pengembangan Sistem Distribusi Produk Lensa Mata Berbasis Mobile Application," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 5, no. 1, p. 1, 2020, doi: 10.24114/cess.v5i1.14353.

- [10] C. D. Farinango, J. S. Benavides, J. D. Cerón, D. M. López, and R. E. Álvarez, "Human-centered design of a personal health record system for metabolic syndrome management based on the ISO 9241-210:2010 standard," *J. Multidiscip. Healthc.*, vol. 11, pp. 21-37, 2018, doi: 10.2147/JMDH.S150976.
- [11] B. M. Amen, S. M. Mahmood, and J. Lu, "Mobile Application Testing Matrix," *Fifth Int. Conf. Comput. Sci. Inf. Technol.*, vol. 14, no. 5, pp. 27-40, 2015, doi: 10.5121/ijsea.2023.14501.
- [12] M. M. Alhammad and A. M. Moreno, "Integrating user experience into Agile," pp. 146-157, 2022, doi: 10.1145/3510456.3514156.
- [13] R. Baxter, *Software engineering is software engineering*. 2006. doi: 10.1049/ic:20040411.
- [14] J. Singh, K. S. Dhindsa, and J. Singh, "Software quality improvement and validation using reengineering," *J. Eng. Res.*, vol. 9, no. 4 A, pp. 59-73, 2021, doi: 10.36909/jer.9481.
- [15] A. Sahoo, D. Kung, and S. Gupta, "An agile methodology for reengineering object-oriented software," *Proc. Int. Conf. Softw. Eng. Knowl. Eng. SEKE*, vol. 2016-Janua, pp. 638-643, 2016, doi: 10.18293/SEKE2016-227.
- [16] S. Zihra Fidela, M. Putri Azizah, and S. Rizka Hidayah, "Tren Pengembangan Aplikasi Mobile: Sebuah Tinjauan Literatur," *J. Tek. Mesin, Ind. Elektro Dan Inform.*, vol. 2, no. 4, pp. 30-48, 2023, [Online]. Available: <https://doi.org/10.55606/jtmei.v2i4.2848>
- [17] J. Zhu and H. Hou, "Research on User Experience Evaluation of Mobile Applications in Government Services," *IEEE Access*, vol. 9, pp. 52634-52641, 2021, doi: 10.1109/ACCESS.2021.3070365.
- [18] K. Bhavsar\*, D. V. Shah, and D. S. Gopalan, "Scrum Challenges: An Agile Process Reengineering in Software Engineering," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 4, pp. 1233-1241, 2020, doi: 10.35940/ijitee.d1567.029420.
- [19] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile software development: Methodologies and trends," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 11, pp. 246-270, 2020, doi: 10.3991/ijim.v14i11.13269.
- [20] S. A. Bhagat, "Review on Mobile Application Development Based on Flutter Platform," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 1, pp. 803-809,

2022, doi: 10.22214/ijraset.2022.39920.

- [21] V. P. Patil, D. Jagtap, S. Khodke, O. Jagdale, and A. Shitole, "Flutter-Modern and Easy Technology to Build Applications," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 2, pp. 458-461, 2023, doi: 10.22214/ijraset.2023.49035.
- [22] C. Carpino, D. Mora, and M. De Simone, "On the use of questionnaire in residential buildings. A review of collected data, methodologies and objectives," *Energy Build.*, vol. 186, pp. 297-318, 2019, doi: 10.1016/j.enbuild.2018.12.021.
- [23] Z. A. Hasibuan, "Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi," *Konsep, Tek. Dan Apl.*, no. Universitas Indonesia, p. 194, 2007.
- [24] H. Taherdoost, "Validity and Reliability of the Research Instrument; How to Test the Validation of a Questionnaire/Survey in a Research," *SSRN Electron. J.*, vol. 5, no. 3, pp. 28-36, 2018, doi: 10.2139/ssrn.3205040.
- [25] A. K. Wojujutari, E. S. Idemudia, and L. E. Ugwu, "The evaluation of the General Health Questionnaire (GHQ-12) reliability generalization: A meta-analysis," *PLoS One*, vol. 19, no. 7 July, pp. 1-16, 2024, doi: 10.1371/journal.pone.0304182.
- [26] V. Cabanas-Sánchez, D. Martínez-Gómez, I. Esteban-Cornejo, J. Castro-Piñero, J. Conde-Caveda, and Ó. L. Veiga, "Reliability and validity of the Youth Leisure-time Sedentary Behavior Questionnaire (YLSBQ)," *J. Sci. Med. Sport*, vol. 21, no. 1, pp. 69-74, 2018, doi: 10.1016/j.jsams.2017.10.031.
- [27] C. Kumlien, J. Nordanstig, M. Lundström, and M. Pettersson, "Validity and test retest reliability of the vascular quality of life Questionnaire-6: A short form of a disease-specific health-related quality of life instrument for patients with peripheral arterial disease," *Health Qual. Life Outcomes*, vol. 15, no. 1, pp. 1-12, 2017, doi: 10.1186/s12955-017-0762-1.
- [28] F. Paz and J. A. Pow-Sang, "A systematic mapping review of usability evaluation methods for software development process," *Int. J. Softw. Eng. its Appl.*, vol. 10, no. 1, pp. 165-178, 2016, doi: 10.14257/ijseia.2016.10.1.16.
- [29] A. Ningtiyas, S. N. Faizah, M. Mustikasari, and I. Bastian, "Pengukuran Usability Sistem Menggunakan USE Questionnaire pada Aplikasi OVO

- Pendahuluan,” *Ilm. KOMPUTASI*, vol. 20, pp. 101-107, 2021, [Online]. Available: <https://ejournal.jakstik.ac.id/files/journals/1/articles/Vol20No1Mar2021/2701/submission/prooff/2701-1-1633-1-10-20210522.pdf>
- [30] Sugiyono, *Metodologi Penelitian Kuantitatif, Kualitatif dan R & D*. 2020.
- [31] “Designing Survey Research: Recommendation for Questionnaire Development, Calculating Sample Size and Selecting Research Paradigms,” no. February, p. 2019, 2019.
- [32] M. S. Ummah, “No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title,” *Sustain.*, vol. 11, no. 1, pp. 1-14, 2019, [Online]. Available: [http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regsciurbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484\\_SISTEM\\_PEMBETUNGAN\\_TERPUSAT\\_STRATEGI\\_MELESTARI](http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regsciurbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SISTEM_PEMBETUNGAN_TERPUSAT_STRATEGI_MELESTARI)
- [33] I. Sommerville, “TENTH edition Tenth Edition,” 2016.
- [34] A. Field, “Discovering Statistic Using IBM SPSS Statistic 5th,” *Dk*, vol. 53, no. 9, pp. 1689-1699, 2017.
- [35] J. Estdale and E. Georgiadou, “Applying the ISO/IEC 25010 Quality Models to Software Product,” *Commun. Comput. Inf. Sci.*, vol. 896, no. December, pp. 492-503, 2018, doi: 10.1007/978-3-319-97925-0\_42.
- [36] T. F. M. Syafei and A. Hidayatullah, “Analisis Penerapan UI/UX Dalam Meningkatkan Pengalaman Pengguna Pada Sistem Reservasi Amadeus,” *JUSTINFO | J. Sist. Inf. dan Teknol. Inf.*, vol. 1, no. 1, pp. 1-8, 2023, doi: 10.33197/justinfo.vol1.iss1.2023.1252.
- [37] M. A. Prastiyo and J. Sundari, “Analisis dan Rancangan UI/UX pada PT. Sherindo Cargo dengan Metode Design Thinking dan SUS,” *J. Masy. Inform.*, vol. 14, no. 2, pp. 131-145, 2023, doi: 10.14710/jmasif.14.2.57117.
- [38] S. N. Islami and M. Dody Firmansyah, “Evaluasi Ui/Ux Dari Aplikasi Ikmas Dengan Menggunakan Metode Design Thinking Dan Pengujian Pengguna,” *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 9, no. 1, pp. 29-38, 2023, doi: 10.36341/rabit.v9i1.4116.
- [39] K. Zaini, T. Marlianti, and M. Maryani, “Design UI/UX e-commerce photography services using the Lean User Experience Model and System

Usability Scale (SUS) Assessment,” *J. Nas. Ilmu Komput.*, vol. 4, no. 4, pp. 45-62, 2023, doi: 10.47747/jurnalnik.v4i4.1421.

- [40] A. E. Agnestisia, M. B. Wenas, and P. Pratiwi, “Perancangan UI/UX pada website Arttrash menggunakan metode Design Thinking,” *Aiti*, vol. 21, no. 1, pp. 14-28, 2024, doi: 10.24246/aiti.v21i1.14-28.
- [41] L. Punchoojit and N. Hongwarittorn, “Usability Studies on Mobile User Interface Design Patterns: A Systematic Literature Review,” *Adv. Human-Computer Interact.*, vol. 2017, 2017, doi: 10.1155/2017/6787504.
- [42] Sholly Salam Sitompul, Sadu Wasistiono, and Fernandes Simangunsong, “Manajemen Strategis Pemerintah Daerah Dalam Pengelolaan Aset Tanah Dan Bangunan Kabupaten Kotawaringin Timur Provinsi Kalimantan Tengah,” *VISIONER J. Pemerintah. Drh. di Indones.*, vol. 13, no. 3, pp. 607-620, 2021, doi: 10.54783/jv.v13i3.479.
- [43] S. Wahyuni *et al.*, “Jurnal Manajemen Aset dan Penilaian OF ASSET MANAGEMENT DALAM PEDOMAN KEBIJAKAN ASET DAERAH,” vol. 1, no. 1, pp. 9-16, 2021.
- [44] E. Maria and A. Halim, “E-goverment dan korupsi: studi di pemerintah daerah, Indonesia dari perspektif teori keagenan,” *Ekuitas J. Ekon. dan Keuang.*, vol. 5, no. 1, pp. 40-58, 2021, doi: 10.24034/j25485024.y2021.v5.i1.4789.



**LAMPIRAN**

**Lampiran 1** Dokumen Pengembangan Produk (DPP)

**LAPORAN PENGEMBANGAN PRODUK TUGAS AKHIR**

***REENGINEERING* APLIKASI TAPATUPA BERBASIS**

***MOBILE***

**DISUSUN OLEH:**

<b>11421004</b>	<b>:</b>	<b>GABRIEL IGNATIUS SITUMEANG</b>
<b>11421018</b>	<b>:</b>	<b>YUDI FRANDIANTO SARAGIH</b>
<b>11421048</b>	<b>:</b>	<b>ANITA LUCIANA MUNTHER</b>

**PROGRAM STUDI SARJANA TERAPAN TEKNOLOGI  
REKAYASA PERANGKAT LUNAK  
FAKULTAS VOKASI  
INSTITUT TEKNOLOGI DEL  
2025**

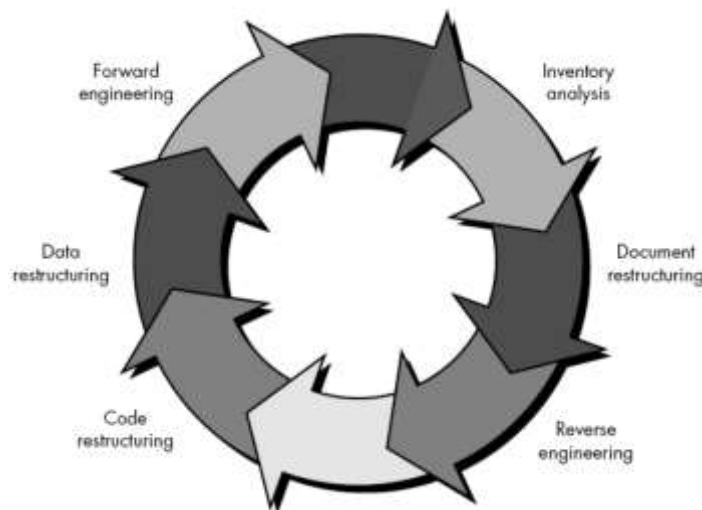
# BAB I

## PRODUCT REQUIREMENT SPECIFICATION (SPESIFIKASI KEBUTUHAN PRODUK)

### 1.1. PENDAHULUAN

Dokumen Pengembangan Produk disusun sebagai bentuk lampiran dari Tugas Akhir yang berjudul “*Reengineering* Aplikasi Tapatupa Berbasis *Mobile*”, yang merupakan panduan menyeluruh dalam proses *reengineering* yang bertujuan untuk mendokumentasikan seluruh spesifikasi kebutuhan, tahapan perancangan ulang, sampai siklus produk yang dikembangkan guna memenuhi kebutuhan pengguna. Berikut adalah tahapan - tahapan dalam siklus proses *reengineering*, yang mencakup:

1. *Inventory analysis* adalah tahapan awal dalam proses *reengineering* yang bertujuan untuk mengidentifikasi fungsi pada aplikasi sebelumnya.
2. *Document Restructuring* adalah tahapan ini bertujuan untuk memperbaiki dan meningkatkan dokumentasi pada sistem aplikasi yang di *reengineering*.
3. *Reverse Engineering* adalah tahapan ini bertujuan untuk menganalisis sistem yang sudah ada untuk diidentifikasi komponen dan hubungannya, dan membuat representasi sistem dalam bentuk lain pada level yang lebih tinggi.
4. *Code Restructuring* adalah tahapan ini berfokus pada perbaikan kualitas kode sumber tanpa mengubah fungsionalitas.
5. *Data Restructuring* adalah tahapan ini bertujuan untuk melakukan modifikasi struktur data dalam suatu sistem agar lebih mudah untuk dipahami.
6. *Forward Engineering* adalah tahapan ini merupakan kebalikan dari *reverse engineering*, dimana proses ini menjadi pengembangan perangkat lunak dari awal tahapan merancang sampai mengimplementasikan ke sistem yang baru.



**Gambar 1. 1 Tahapan *Reengineering***

#### **1.1.1. Tujuan Penulisan Dokumen**

Dokumen ini disusun untuk menyajikan kebutuhan sistem aplikasi Tapatupa yang akan dikembangkan ulang sebagai hasil dari proses evaluasi sistem sebelumnya. Dokumen ini memuat daftar kebutuhan sistem secara rinci, termasuk kebutuhan fungsional, non-fungsional, batasan teknis, serta ruang lingkup pengembangan sistem. Seluruh informasi ini dikumpulkan melalui analisis alur proses bisnis pada aplikasi sebelumnya, aplikasi sejenis, dan dokumen Peraturan Bupati No. 12 tahun 2024, sehingga dapat digunakan sebagai referensi utama dalam proses perancangan, pengembangan, dan pengujian aplikasi.

Tujuan dari Dokumen Pengembangan Produk (DPP) untuk proyek *Reengineering* Aplikasi Tapatupa berbasis *mobile* adalah sebagai berikut:

1. Mendokumentasikan kebutuhan sistem secara sistematis dan rinci, berdasarkan hasil analisis proses bisnis dan evaluasi akhir dari kuesioner pengguna akhir, sehingga dapat ditelusuri kembali secara jelas saat proses pengembangan.
2. Menyediakan acuan teknis terstandar yang digunakan oleh tim pengembang untuk mendesain dan membangun ulang sistem sesuai spesifikasi.

3. Menjamin keselarasan pemahaman antara tim teknis dan pemangku kepentingan, melalui penyusunan kebutuhan sistem dalam bentuk yang eksplisit dan tervalidasi.
4. Menjadi dasar pengujian akhir (final testing) untuk memverifikasi bahwa sistem yang dibangun telah memenuhi seluruh kebutuhan pengguna yang ditentukan dalam dokumen ini.

### **1.1.2. Latar Belakang Produk**

Aset tanah merupakan salah satu komponen penting dari Barang Milik Daerah (BMD) yang memiliki peran dalam mendukung pelayanan publik dan menghasilkan pendapatan sewa bagi pemerintah daerah. Agar proses pengelolaan berjalan sesuai dengan prinsip akuntabilitas dan transparansi, dibutuhkan sistem pencatatan dan pemanfaatan aset yang terdigitalisasi. Pemerintah Kabupaten Tapanuli Utara telah menetapkan kebijakan melalui Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024, yang mengatur pelaksanaan pengelolaan aset daerah secara elektronik, termasuk proses permohonan dan pembayaran sewa oleh masyarakat.

Sebagai bentuk pelaksanaan dari kebijakan tersebut, dikembangkan aplikasi Tapatupa untuk mendukung proses pencatatan, permohonan sewa, dan pemantauan kontrak aset tanah daerah. Namun, berdasarkan hasil analisis terhadap versi awal aplikasi, observasi langsung terhadap pengguna, dan perbandingan dengan aplikasi serupa, ditemukan beberapa kendala yang menghambat fungsionalitas sistem. Antarmuka pengguna dinilai membingungkan karena tidak menyediakan navigasi yang jelas dan konsisten. Fitur unggah dokumen belum tersedia secara digital sehingga pengguna masih harus menyerahkan dokumen fisik, yang menyebabkan keterlambatan dalam verifikasi permohonan. Sistem pembayaran belum terhubung langsung dengan bank daerah, sehingga petugas harus melakukan verifikasi secara manual. Selain itu, fitur pencarian aset tidak mendukung kata kunci tertentu, dan status permohonan tidak selalu ditampilkan secara *real-time*, menyebabkan pengguna kesulitan dalam melacak progres sewa.

Permasalahan tersebut berdampak pada peningkatan waktu penyelesaian permohonan, serta tingginya ketergantungan pada proses manual, yang berpotensi menyebabkan keterlambatan pelayanan dan kesalahan data. Ketidakhadiran sistem yang terintegrasi juga menyulitkan petugas dalam melakukan *monitoring* dan pencatatan sesuai prosedur. Jika tidak diperbaiki, maka sistem ini berisiko tidak memenuhi ketentuan administratif dalam Peraturan Bupati, dan tidak mendukung target modernisasi pengelolaan aset sebagaimana diharapkan.

Sebagai respons terhadap temuan tersebut, dilakukan proses *reengineering* terhadap aplikasi Tapatupa dengan pendekatan pengembangan berbasis *mobile*. *Reengineering* ini mencakup perancangan ulang antarmuka agar lebih mudah digunakan berdasarkan pola interaksi umum pengguna *smartphone*, penambahan fitur unggah dokumen agar seluruh berkas permohonan bisa dikirimkan secara digital, serta integrasi sistem pembayaran langsung dengan *Virtual account* (VA) dari bank daerah guna mempercepat proses verifikasi pembayaran. Dengan perubahan ini, sistem diharapkan dapat meminimalkan keterlambatan proses, mengurangi ketergantungan terhadap administrasi manual, dan menyediakan informasi status sewa secara langsung bagi pengguna.

### **1.1.3. Tujuan Produk**

Tujuan dari pengembangan produk ini adalah melakukan proses membangun ulang terhadap aplikasi Tapatupa berbasis *mobile* dalam konteks pengelolaan aset tanah di Kabupaten Tapanuli Utara. Proses ini dilakukan untuk menjawab berbagai tantangan dalam implementasi aplikasi sebelumnya serta untuk menghadirkan solusi digital yang lebih terstruktur dan berorientasi pada kebutuhan pengguna (*user-oriented*).

Secara khusus, tujuan dari proses *reengineering* ini meliputi:

1. Merancang ulang antarmuka pengguna (*user interface*) agar lebih informatif, responsif, dan sesuai dengan kebutuhan pengguna. Tujuannya adalah untuk meningkatkan pengalaman pengguna (*user experience*) dalam mengakses informasi dan menjalankan fungsi-fungsi utama aplikasi dengan mudah.

2. Menyempurnakan fitur permohonan penyewaan tanah, termasuk proses pengajuan, unggah dokumen, dan pelacakan status permohonan agar lebih selaras dengan kebutuhan pengguna dan ketentuan dalam regulasi, seperti yang tertuang dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024. Dengan penyempurnaan ini, proses administrasi penyewaan diharapkan menjadi lebih cepat, transparan, dan terdokumentasi dengan baik.
3. Meningkatkan sistem pembayaran dan pengelolaan tagihan melalui integrasi notifikasi status pembayaran, penyajian informasi tagihan yang lebih rinci dan akurat, serta otomatisasi dalam pengelolaan nomor *Virtual account*. Upaya ini bertujuan untuk mengurangi kesalahan input manual, meningkatkan proses pembayaran, serta memperkuat aspek dalam pelaporan retribusi.

#### **1.1.4. Ruang Lingkup Produk**

Ruang lingkup pengembangan aplikasi pengelolaan aset tanah berbasis *mobile* ini mencakup seluruh aspek yang berkaitan dengan pemanfaatan tanah milik daerah di 15 kecamatan Kabupaten Tapanuli Utara, sesuai dengan Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024. Berikut adalah detail ruang lingkup produk:

1. Penelitian difokuskan pada proses *reengineering* aplikasi Tapatupa berbasis *mobile* bagian desain antarmuka pengguna dan fungsi utama yang termasuk dalam ruang lingkup produk meliputi modul registrasi dan autentikasi pengguna, modul pengajuan permohonan penyewaan aset, modul pengelolaan data aset tanah seperti informasi objek retribusi dan tarif objek retribusi dalam bentuk filter pencarian, modul persetujuan dalam bentuk perjanjian sewa, serta modul tagihan sewa untuk dilakukan pembayaran.
2. Tahapan *reengineering* yang dilakukan adalah *inventory analysis*, *document restructuring*, *code refactoring*, *data refactoring* dan *forward engineering*.
3. Pengembangan ulang aplikasi dilakukan menggunakan *framework Flutter*, yang memungkinkan pengembangan aplikasi *mobile* dengan antarmuka yang responsif dan dapat berjalan diberbagai *platform*. Meskipun *framework* pengembangan yang digunakan sebelum dan sesudah proses

*reengineering* adalah Flutter, terdapat perbedaan signifikan pada struktur internal, arsitektur kode, dan pendekatan desain sistem.

4. Objek kajian mencakup seluruh wilayah administratif sebanyak 15 kecamatan, dengan 3 kecamatan dipilih sebagai sampel untuk pengumpulan data guna mewakili kebutuhan dan permasalahan pengguna.

#### 1.1.5. Manfaat Produk

Manfaat dari pengembangan aplikasi pengelolaan aset tanah berbasis *mobile* ini adalah

1. Membantu pengguna dalam mengakses informasi dan layanan pengajuan permohonan sewa aset tanah melalui antarmuka aplikasi yang lebih jelas, mudah dinavigasi, sehingga mengurangi waktu yang dibutuhkan pengguna untuk menemukan dan menggunakan fitur utama aplikasi.
2. Mempercepat proses permohonan penyewaan tanah, dengan menyediakan fitur digital yang mendukung pengajuan, pengunggahan dokumen, serta pelacakan status permohonan secara langsung, sehingga mengurangi ketergantungan pada proses tatap muka dan dokumen fisik.
3. Mempermudah pengelolaan pembayaran, melalui penyediaan informasi tagihan yang lengkap dan penggunaan nomor *Virtual account*, sehingga mengurangi kesalahan input pembayaran dan mempermudah proses verifikasi transaksi.

#### 1.1.6. Definisi dan Singkatan

Bagian ini memberikan daftar istilah teknis atau singkatan yang digunakan dalam dokumen ini, beserta definisi atau kepanjangannya. Tujuannya adalah untuk memudahkan pembaca memahami isi dokumen dengan konsisten.

**Tabel 1. 1 Daftar Definisi dan Singkatan**

No.	Kata	Deskripsi
1	<i>Developer</i>	Pihak bersangkutan yang telah memberikan penjelasan dan requirement tentang aplikasi yang akan dibangun. 2 -

No.	Kata	Deskripsi
		3 Tim pengembang yang akan mengembangkan aplikasi baru.
2	<i>Current System</i>	Sistem yang berlaku saat ini yang digunakan oleh user pada aplikasi sebelumnya ataupun kegiatan yang berlangsung.
3	<i>Target System</i>	Target atau hasil yang ingin dicapai dalam pembuatan sistem.
4	<i>Requirement</i>	Kebutuhan yang diperlukan pengguna yang harus ada di aplikasi.
5	<i>Deliverable</i>	Hasil akhir dari proses pengerjaan proyek pengembangan aplikasi Tapatupa berbasis <i>mobile</i> .
6	<i>Software</i>	Perangkat keras yang digunakan selama pengembangan aplikasi.
7	<i>Hardware</i>	Perangkat lunak yang digunakan selama pengembangan aplikasi.
8	<i>User / Pengguna</i>	Pihak yang akan menggunakan aplikasi ini yaitu pengguna aplikasi di Kabupaten Tapanuli Utara.
9	<i>Servis Time</i>	Waktu yang dibutuhkan untuk mengakses layanan suatu sistem.
10	<i>Functional Requirement</i>	Fungsi yang harus terdapat pada aplikasi yang dibangun untuk memenuhi kebutuhan pengguna dan untuk mencapai bisnis proses aplikasi.
11	<i>Non-functional Requirement</i>	Kebutuhan yang menentukan aspek kualitas sistem, seperti kinerja, keamanan, dan keandalan.



No.	Kata	Deskripsi
12	<i>Use Case Diagram</i>	Diagram yang menggambarkan interaksi antara pengguna dan sistem dalam berbagai skenario penggunaan.
13	<i>Business Process</i>	Serangkaian langkah atau prosedur yang digunakan untuk mencapai tujuan bisnis tertentu.

Akronim yang digunakan dalam penulisan dokumen aplikasi Tapatupa berbasis *mobile* ini dapat dilihat pada Tabel 1.2.

**Tabel 1. 2 Daftar Akronim**

No	Kata	Deskripsi
1	PiP	<i>Project Implementation</i>
2	ToR	<i>ToR Term of References</i>

Singkatan yang digunakan dalam penulisan dokumen aplikasi Tapatupa berbasis *mobile* ini dapat dilihat pada Tabel 1. 3.

**Tabel 1. 3 Daftar Singkatan**

No	Kata	Deskripsi
1	BPMN	<i>Business Process Modeling Notation</i>
2	ERD	<i>Entity Relationship Diagram</i>
3	TA	Tugas Akhir

#### **1.1.7. Referensi**

Bagian ini berfungsi untuk merujuk penelitian, teknologi, atau solusi yang sesuai dan telah ada sebelumnya, yang menjadi dasar pengembangan produk. Studi literatur mencakup referensi terkait konsep atau teknologi yang digunakan, perbandingan dengan sistem atau produk serupa, identifikasi kelebihan dan kekurangan solusi yang ada, sehingga menjadi dasar untuk inovasi.

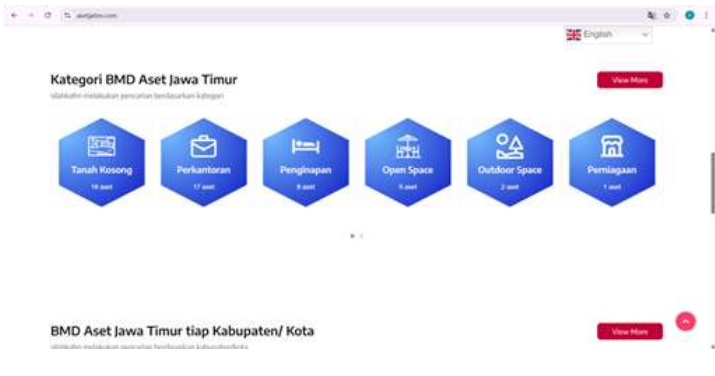
Adapun pengamatan terhadap aplikasi sejenis dilakukan terhadap 3 sistem yang dipakai oleh daerah masing masing, yaitu *website* Siap Sewa dari Jawa Timur, *website* Amanah dari Bandung, dan SIWA App dari Bali dengan hasil sebagai berikut.

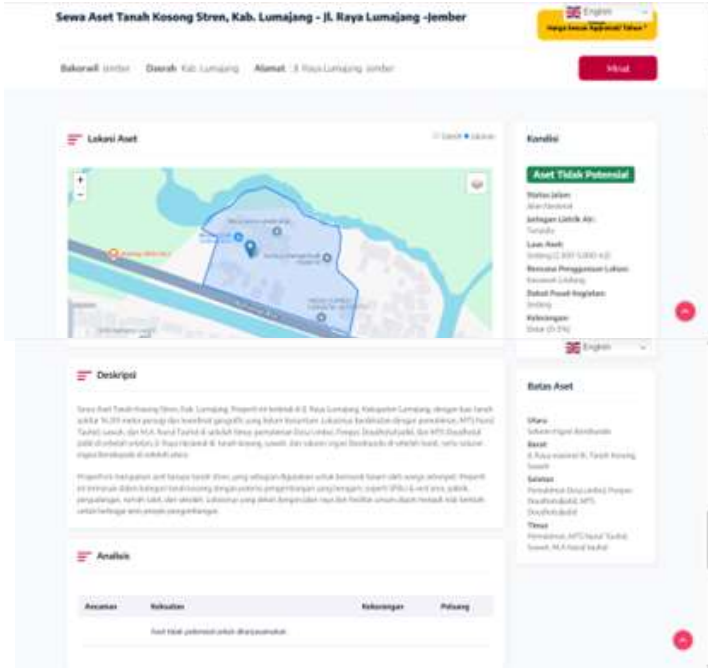

#### 1. Website SIAPSEWA Aset Jawa Timur

*Platform* digital yang dikelola oleh Pemerintah Provinsi Jawa Timur untuk memberikan layanan penyewaan Barang Milik Daerah (BMD) kepada masyarakat atau investor. *Platform* ini bertujuan untuk meningkatkan pemanfaatan aset pemerintah dengan skema penyewaan yang transparan dan legal. *Website* ini memberikan informasi terkait:

- a. Aset yang disewakan, termasuk lokasi dan deskripsi properti.
- b. Kategori aset, seperti perkantoran, tanah kosong, penginapan, open space, outdoor, dan sewa ruangan.
- c. Harga sewa, yang ditampilkan berdasarkan hasil appraisal per tahun.
- d. Alur dan Informasi penyewaan, sehingga pengguna dapat memahami prosedur dalam menyewa aset.

**Tabel 1. 4 Analisis Website SiapSEWA**

No	Tampilan Aplikasi	Keterangan
1	 <p><b>Gambar 1. 2 Tampilan beranda website SiapSEWA</b></p>	Tampilan halaman utama <i>website</i> yang menampilkan setiap kategori BMD asset Jawa Timur.


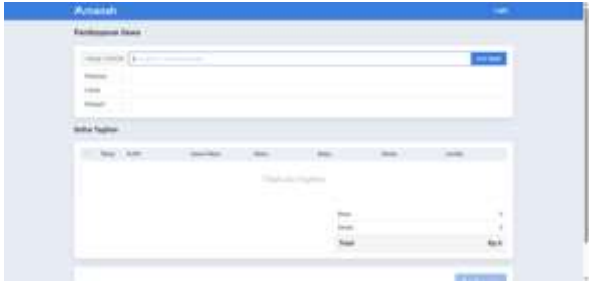
No	Tampilan Aplikasi	Keterangan
2	 <p><b>Gambar 1. 3 Tampilan detail website SiapSEWA</b></p>	Tampilan disamping menampilkan gambar aset, Alamat, data video aset, deskripsi property, dan analisis terhadap asset yang akan dipinjamkan.
3	 <p><b>Gambar 1. 4 Tampilan Hak aset website SiapSEWA</b></p>	Tampilan disamping menunjukkan tombol minat bagi pengguna yang ingin menyewa tempat tersebut.

## 2. Website Amanah Bandung

Amanah.bandung.go.id adalah *platform* yang dikelola oleh Pemerintah Kota Bandung untuk pengelolaan aset daerah secara transparan dan efisien. Aplikasi ini dirancang untuk memfasilitasi inventarisasi aset, termasuk tanah, bangunan, dan fasilitas lainnya yang dimiliki pemerintah daerah. Fitur utamanya mencakup pencatatan aset yang terstruktur, visualisasi data berbasis peta, serta sistem

pengajuan dan manajemen penyewaan aset. Selain itu, platform ini mendukung transparansi publik dengan menyediakan informasi terkait status aset yang dapat diakses oleh masyarakat umum.

**Tabel 1. 5 Analisis Website Amanah**

No	Tampilan Aplikasi	Keterangan
1	 <p><b>Gambar 1. 5 Tampilan beranda website Amanah</b></p>	<p>Tampilan halaman utama ini menampilkan menu fitur pembayaran, pelaporan, pelayanan, dan peta tematik.</p> <p>Pembayaran membantu pengguna untuk mengelola pembayaran sewa tanah, termasuk informasi tagihan, status pembayaran, dan metode yang disediakan.</p> <p>Pada menu pelaporan, pengguna dapat melaporkan aktivitas atau masalah yang berkaitan dengan sewa aset.</p> <p>Pelayanan berisi pengajuan perpanjangan kontrak, informasi terkait aset.</p>
2	 <p><b>Gambar 1. 6 Tampilan pembayaran di dalam website</b></p>	<p>Pada tampilan di samping, secara detail fitur pembayaran berisi informasi penyewa, lokasi, kategori, dan daftar tagihan. Menu pembayaran menyediakan informasi no. kontrak sewa asset yang akan disewakan.</p>


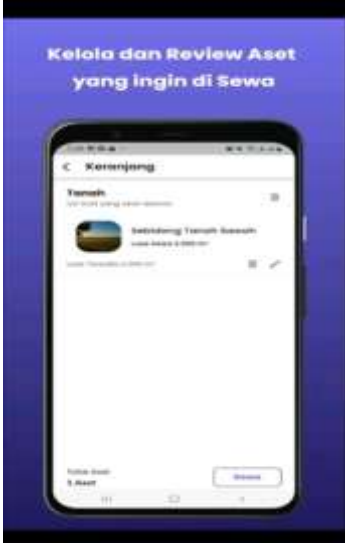
No	Tampilan Aplikasi	Keterangan

### 3. Aplikasi SIWA Bali

*Platform* atau aplikasi yang dikembangkan oleh Pemerintah Provinsi Bali untuk mengelola data aset daerah. Tampilan beranda pada aplikasi SIWA dirancang untuk memudahkan pengguna dalam menemukan dan menyewa aset secara digital, pada bagian Rekomendasi Aset, pengguna dapat melihat aset yang direkomendasikan, seperti tanah area maupun bangunan. Di bagian Aset Bangunan, pengguna dapat menemukan informasi spesifik tentang aset tertentu, seperti Harga dari aset tersebut.

**Tabel 1. 6 Analisis Aplikasi SIWA**

No	Tampilan Aplikasi	Keterangan
1	 <p><b>Gambar 1. 7 Analisis Beranda SIWA</b></p>	<p>Pada tampilan disamping terdapat halaman utama pada aplikasi SIWA. Informasi utama bagian “rekomendasi aset” menampilkan informasi harga dan ketersediaan tanah dengan visual yang mendukung, seperti gambar dan label harga estimasi. Fitur pencarian yang mudah pada kolom pencarian di bagian atas memungkinkan pengguna mencari aset secara langsung. Pada fitur tipe aset memberikan pengelompokan yang jelas antara aset tanah dan aset bangunan yang membantu pengguna memahami pilihan yang tersedia.</p>

No	Tampilan Aplikasi	Keterangan
2	 <p style="text-align: center;"><b>Gambar 1. 8 Tampilan halaman pengajuan</b></p>	<p>Halaman disamping menampilkan pengajuan sewa tanah dengan mengisi formulir pengajuan yang berisi no.telepon, umur, pekerjaan, Alamat, lama menyewa, dan file sketsa aset. Setelah pengguna mengisi formulir pengajuan, maka pengajuan dapat disimpan dan masuk dalam fitur keranjang.</p>
3	 <p style="text-align: center;"><b>Gambar 1. 9 Tampilan keranjang Aplikasi SEWA</b></p>	<p>Pada tampilan disamping, fitur keranjang menampilkan aset yang akan disewakan sebelumnya. Setelah itu pengguna dapat mengklik <i>button</i> sewa.</p>

## 1.2. DESKRIPSI UMUM PRODUK

Aplikasi Tapatupa merupakan sistem informasi pengelolaan aset tanah milik daerah yang dikembangkan untuk mendukung proses administrasi penyewaan tanah di

Kabupaten Tapanuli Utara. Aplikasi ini dirancang untuk menggantikan proses manual dalam pencatatan aset, pengajuan permohonan sewa, pengelolaan perjanjian, dan pencatatan pembayaran retribusi, melalui sistem berbasis *mobile* yang dapat diakses oleh pengguna masyarakat wajib retribusi.

Aplikasi ini dikembangkan ulang melalui pendekatan *reengineering* dengan menyesuaikan kebutuhan teknis berdasarkan hasil evaluasi terhadap versi sebelumnya, serta merujuk pada ketentuan dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024. Proses pengembangan mencakup perancangan ulang antarmuka pengguna berbasis pendekatan *User-Centered Design*(UCD), penambahan modul seperti unggah dokumen dan pencarian aset, serta integrasi sistem pembayaran melalui *Virtual account* (VA).

Produk ini menyediakan akses informasi aset melalui perangkat *mobile*, memungkinkan pengguna untuk melakukan permohonan dan mengunggah dokumen pendukung secara langsung, serta menampilkan status permohonan dan pembayaran secara waktu nyata (*real-time*). Seluruh data permohonan, perjanjian, dan tagihan disimpan secara terpusat dan dapat ditelusuri oleh petugas kecamatan melalui akun pengguna yang terautentikasi.

### **1.2.1. Permasalahan dan Persoalan**

Pengembangan ulang aplikasi Tapatupa dilakukan berdasarkan hasil observasi terhadap aplikasi versi sebelumnya, analisis terhadap aplikasi sejenis, dan penyesuaian terhadap Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024. Aplikasi Tapatupa dirancang untuk mendukung proses administrasi penyewaan tanah milik daerah melalui sistem digital, namun dalam implementasinya ditemukan beberapa kendala yang menghambat efektivitas penggunaannya di lapangan.

Salah satu permasalahan utama adalah tidak tersedianya fitur unggah dokumen secara digital dalam proses pengajuan permohonan sewa. Pengguna tetap diminta membawa berkas fisik ke kantor kecamatan, sehingga proses tidak berjalan secara daring sepenuhnya. Selain itu, sistem tidak menyediakan fungsi pencarian atau filter data aset berdasarkan lokasi atau jenis retribusi, yang menyebabkan pengguna



harus menelusuri seluruh daftar aset tanpa bantuan penyaringan data. Informasi status permohonan dan pembayaran juga tidak ditampilkan secara langsung pada aplikasi, sehingga pengguna perlu menanyakan perkembangan proses secara manual kepada petugas.

Pada struktur antarmuka, susunan menu tidak mengikuti urutan proses kerja pengguna. Formulir pengajuan tidak memberikan notifikasi atau umpan balik setelah data dikirim, sehingga pengguna tidak mengetahui apakah permohonan berhasil diproses atau tidak. Dokumen perjanjian sewa serta rincian tagihan juga tidak tersedia dalam bentuk digital di dalam aplikasi, dan seluruh proses pencetakan dokumen masih dilakukan secara manual oleh petugas.

Permasalahan lain ditemukan pada proses pembayaran retribusi. Aplikasi versi lama tidak memiliki integrasi sistem pembayaran digital. Setelah melakukan transfer, pengguna diminta mengunggah bukti pembayaran melalui form terpisah yang tidak terhubung langsung dengan modul tagihan. Hal ini menyebabkan proses verifikasi pembayaran harus dilakukan secara manual oleh petugas dan meningkatkan risiko keterlambatan atau kekeliruan pencatatan.

Berdasarkan permasalahan tersebut, dibutuhkan sistem yang menyediakan fitur unggah dokumen secara langsung pada formulir permohonan, pencarian aset menggunakan filter berdasarkan lokasi dan jenis retribusi, tampilan status permohonan dan pembayaran yang diperbarui secara waktu nyata, serta integrasi pembayaran melalui *Virtual account* (VA) agar pengguna tidak perlu lagi mengunggah bukti pembayaran. Seluruh data permohonan, kontrak, dan tagihan juga perlu disimpan dan ditampilkan dalam bentuk digital yang dapat diakses oleh pengguna melalui akun masing-masing sesuai peran. Pengembangan ulang ini dilakukan untuk memastikan sistem dapat digunakan secara daring, mendukung transparansi data, serta memenuhi ketentuan teknis yang telah ditetapkan oleh pemerintah daerah.



**Tabel 1. 2 Objek Perbandingan**

<b>Nama Sistem</b>	<b>Kelebihan</b>	<b>Kekurangan</b>
SiapSEWA	Menampilkan informasi aset secara detail (gambar, alamat, video, deskripsi) dan menarik.	Tidak adanya fitur pembayaran atau pelaporan yang terintegrasi
Amanah Bandung	Fitur pembayaran yang lengkap (tagihan, status pembayaran, metode pembayaran), serta adanya menu pelaporan dan pelayanan yang memudahkan pengelolaan aset.	Kurangnya informasi detail tentang aset seperti gambar, dan aset video,
SIWA App	Adanya fitur kolom pencarian yang mudah dan pengelompokan aset yang jelas.	Tidak ada laporan aktivitas yang dapat dilihat oleh pengguna.

Setelah objek perbandingan pada ketiga sistem penyewaan aset dilakukan, maka terdapat beberapa aspek pada sistem tersebut yang akan diadopsi pada aplikasi Tapatupa yang akan dijelaskan pada Tabel 1.9.

**Tabel 1. 3 Aspek yang diadopsi**

<b>No</b>	<b>Aspek yang diadopsi</b>	<b>Keterangan</b>
1	<i>Visual Hierarchy</i>	Menggunakan hirarki visual yang jelas untuk menonjolkan informasi penting seperti harga, lokasi, dan ketersediaan aset. Ini dapat dilakukan dengan ukuran font, warna, dan penempatan elemen yang strategis.

No	Aspek yang diadopsi	Keterangan
2	<i>Feedback dan Interaction</i>	Memberikan umpan balik yang jelas saat pengguna melakukan tindakan, seperti mengklik dan menyelesaikan pembayaran. ini meningkatkan kepercayaan pengguna terhadap sistem.
3	Fitur pembayaran terintegrasi	Menyediakan sistem pembayaran yang terintegrasi untuk memudahkan transaksi sewa atau beli aset, seperti menyediakan fitur pembayaran lengkap, termasuk tagihan, status pembayaran, dan metode pembayaran.  Melakukan integrasi dengan gateway pembayaran pada <i>bank local</i> (BPD Sumut) dan memberikan notifikasi status pembayaran

### 1.2.3. Produk yang akan dibangun

Produk yang akan dibangun merupakan hasil dari proses *reengineering* aplikasi Tapatupa berbasis *mobile*, yang bertujuan untuk mengatasi keterbatasan pada versi sebelumnya. Aplikasi ini dikembangkan dengan pendekatan *User-Centered Design* dan berbasis framework Flutter, agar mendukung akses lintas platform serta mampu menjawab kebutuhan pengguna secara fungsional. Pengembangan diarahkan untuk menyempurnakan proses administrasi penyewaan aset tanah milik daerah di Kabupaten Tapanuli Utara sesuai dengan ketentuan Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024.

Beberapa fitur utama dirancang secara spesifik untuk menyelesaikan masalah yang telah diidentifikasi sebelumnya. Pertama, pada proses pengajuan permohonan,

sistem akan dilengkapi dengan fitur unggah dokumen digital agar pengguna dapat mengirimkan dokumen pendukung secara langsung melalui aplikasi tanpa perlu menyerahkan berkas fisik. Kedua, sistem akan menyediakan fitur filter pencarian objek retribusi, yang memungkinkan pengguna mencari aset berdasarkan nama objek atau lokasi (kecamatan), serta fitur pencarian tarif retribusi berdasarkan jenis objek. Fitur ini dibangun agar pengguna dapat memperoleh informasi spesifik secara cepat dan akurat.

Ketiga, untuk mendukung transparansi dalam perjanjian, sistem akan menampilkan tautan dokumen perjanjian dalam bentuk digital pada halaman perjanjian, yang dapat diakses langsung oleh pengguna. Keempat, sistem akan diintegrasikan dengan *Virtual account* (VA) pada fitur tagihan, sehingga setiap pengguna akan menerima nomor VA unik yang ditampilkan langsung di aplikasi. Dengan integrasi ini, proses pembayaran tidak lagi memerlukan unggah bukti bayar manual.

Selanjutnya, sistem juga akan menyediakan fitur riwayat pembayaran yang menampilkan status pembayaran secara otomatis dan terdokumentasi. Pembayaran yang masuk akan divalidasi langsung oleh sistem melalui notifikasi bank, sehingga status pembayaran akan diperbarui secara waktu nyata dan tidak memerlukan proses verifikasi manual oleh petugas.

Dengan penerapan fitur-fitur tersebut, produk hasil *reengineering* ini diharapkan dapat menyederhanakan proses permohonan sewa aset, meningkatkan akurasi informasi, serta mempercepat siklus administrasi penyewaan tanah milik daerah melalui sistem yang terdigitalisasi dan terintegrasi penuh.

#### **1.2.4. Konteks**

Sub Bab ini menjelaskan konteks sistem yang akan dikembangkan, termasuk pemetaan *stakeholder*, komponen utama aplikasi, serta relasi antar modul dalam sistem. Tujuan dari bagian ini adalah untuk memberikan gambaran awal yang menyeluruh tentang ruang lingkup sistem dalam arsitektur fungsional dan peran aktor-aktor terkait. Adapun *stakeholder* yang terlibat dimulai dari :

1. *User* (Masyarakat wajib retribusi) : Masyarakat wajib retribusi adalah pengguna akhir yang dapat mengakses informasi sewa tanah melalui aplikasi *mobile* seperti mengajukan permohonan, mengakses tagihan sewa,

melakukan pembayaran, mengakses daftar objek retribusi, serta mengakses tarif objek retribusi.

2. *Developer* : *Developer* adalah tim pengembang yang membangun ulang aplikasi Tapatupa berbasis *mobile* dengan menerapkan tahapan *software reengineering* agar sesuai kebutuhan pengguna.

#### **1.2.5. Deskripsi Kebutuhan Produk**

Sistem Tapatupa merupakan aplikasi penyewaan aset tanah milik pemerintah daerah yang dirancang untuk memudahkan masyarakat dalam mengakses informasi aset, mengajukan permohonan penyewaan, serta melakukan pembayaran retribusi secara digital. Aplikasi ini dikembangkan untuk meningkatkan efisiensi administrasi dan transparansi pengelolaan aset daerah melalui sistem yang terintegrasi dan dapat diakses secara daring.

Struktur sistem terdiri atas sejumlah modul utama yang masing-masing memuat fungsi spesifik, serta didukung oleh kebutuhan non-fungsional yang disesuaikan dengan praktik pengembangan aplikasi *mobile*. Aspek keamanan diterapkan melalui autentikasi pengguna sebelum mengakses fitur utama. Keandalan sistem dijaga dengan melakukan pengujian fungsi dasar setiap kali ada pembaruan. Antarmuka pengguna dirancang ulang agar mengikuti alur proses permohonan secara berurutan, sehingga pengguna dapat mengakses fitur-fitur sesuai urutan kerja tanpa kebingungan.

Adapun fungsi utama yang akan diimplementasikan dalam sistem Tapatupa meliputi:

1. Fitur Permohonan
2. Fitur Objek Retribusi
3. Fitur Tarif Objek Retribusi
4. Fitur Perjanjian Sewa
5. Fitur Tagihan Sewa
6. Fitur Pembayaran
7. Fitur *Profile*

Berdasarkan fungsi - fungsi utama yang sudah di kembangkan pada aplikasi versi sebelumnya diatas, pengembang menambahkan beberapa fungsi pada aplikasi dengan tujuan pemenuhan kebutuhan dari aplikasi Tapatupa, sebagai berikut.

#### 1. Fitur Permohonan

Fitur ini digunakan oleh pengguna (wajib retribusi) untuk mengajukan permohonan penyewaan aset tanah milik pemerintah daerah. Fungsi dalam fitur ini meliputi :

- a. Mengisi formulir digital permohonan sewa aset tanah.
- b. Mengunggah dokumen pendukung seperti KTP, surat permohonan, dan dokumen lain yang diperlukan.
- c. Melihat detail permohonan sewa yang sudah diajukan.
- d. Melihat status permohonan dalam riwayat permohonan.

#### 2. Fitur Tagihan Sewa

Setelah permohonan disetujui, fitur ini menyajikan tagihan sewa yang harus dibayarkan oleh penyewa berdasarkan perjanjian sewa yang telah disetujui. Fungsi dalam fitur ini meliputi :

- a. Menampilkan rician tagihan sewa berdasarkan nomor perjanjian sewa, tanggal yang disahkan, jangka waktu, pembayaran, nama wajib retribusi, dan objek retribusi yang disewa.
- b. Menyediakan informasi jatuh tempo pembayaran.

#### 3. Fitur Pembayaran

Fitur ini mendukung proses pembayaran digital retribusi sewa oleh penyewa kepada pemerintah daerah. Sistem akan mengintegrasikan proses pembayaran dengan bank daerah. Pengguna dapat melakukan pembayaran tanpa perlu mengunggah bukti bayar seperti sistem sebelumnya. Fungsi dalam fitur ini meliputi :

- a. Menampilkan instruksi cara pembayaran (melalui bank, *mobile banking*)
- b. Menampilkan status pembayaran apabila sudah bayar.
- c. Menyediakan nomor *Virtual account* untuk setiap transaksi.

#### 4. Fitur Objek Retribusi

Fitur ini menampilkan daftar aset tanah yang tersedia untuk disewa oleh masyarakat.

Fungsi dalam fitur ini meliputi:

- a. Melihat informasi detail setiap objek tanah (lokasi, ukuran, status kepemilikan).
- b. Melakukan pencarian dan filter objek berdasarkan lokasi atau kategori penggunaan.
- c. Melihat gambar/foto aset.

#### 5. Fitur Tarif Retribusi

Fitur ini memberikan informasi mengenai besaran tarif sewa yang dikenakan pada masing-masing objek retribusi sesuai dengan regulasi yang berlaku.

Fungsi dalam fitur ini meliputi:

- a. Menampilkan daftar tarif sewa per objek atau per kategori lahan.
- b. Menyesuaikan tarif berdasarkan durasi sewa, luas lahan, atau fungsi penggunaan.
- c. Memberikan transparansi kepada pengguna tentang besaran biaya sebelum mengajukan permohonan.

#### 6. Fitur Perjanjian Sewa

Fitur ini mengelola dokumen perjanjian sewa yang menjadi dasar hukum hubungan antara penyewa dan pemerintah daerah.

Fungsi dalam fitur ini meliputi:

- a. Menampilkan ringkasan isi perjanjian, termasuk masa berlaku, hak dan kewajiban penyewa.
- b. Menyediakan dokumen perjanjian dalam format digital dalam bentuk tautan yang dapat di klik.
- c. Menampilkan status perjanjian (aktif, berakhir, atau diperpanjang).

#### 7. Fitur Profil

Fitur ini menyimpan data dan identitas pengguna aplikasi berupa NIK, alamat, dan nomor telepon.



### 1.2.6. *Environment Hardware dan Software*

Pada bagian ini akan diuraikan terkait kebutuhan spesifikasi *tools* yang dibutuhkan oleh aplikasi Tapatupa berbasis *mobile* agar sistem yang dibangun dapat berfungsi dengan baik. Kebutuhan ini mencakup kebutuhan perangkat keras (*hardware requirement*) dan kebutuhan perangkat lunak (*software requirement*) serta infrastruktur lainnya sebagai pendukung yang dibutuhkan.

#### 1.2.6.1 Hardware Requirement

Dalam rancangan aplikasi Tapatupa, *hardware requirement* dapat dilihat pada Tabel 1.10.

**Tabel 1. 4 Daftar Hardware Requirement**

Hardware	Spesification
<i>Server</i>	XAMPP
<i>Web server</i>	8 CPU cores, 16 GB RAM
<i>Memory (RAM)</i>	Minimum 1 GB; Recommended 4 GB or above
<i>Hard Drive</i>	Minimum 32 GB; Recommended 64 GB or more

#### 1.2.6.2 S/W Requirement

Spesifikasi perangkat lunak (*software*) dari lingkungan pengoperasian aplikasi Tapatupa dapat dilihat pada Tabel 1.11.

**Tabel 1. 5 Daftar S/W Requirement**

Groups	Components	Spesification
Sistem Operasi	Windows	windows 11
Client	<i>Web Browser</i>	Chrome, Mozilla Firefox
Proses Bisnis	Bizagi Modeler	Untuk pembuatan diagram alur proses bisnis (BPMN) dalam perancangan sistem.
Alat desain UI / UX	Figma	Untuk pembuatan desain antarmuka pengguna (UI) dan prototipe aplikasi berbasis <i>mobile</i> .

Groups	Components	Spesification
Arsitektur pemodelan	<i>Enterprise Architect</i>	Untuk pemodelan sistem seperti ERD, DFD, dan diagram arsitektur sistem lainnya.
<i>Project repositori</i>	<i>Github</i>	Github Untuk kolaborasi implementasi project bersama tim

### 1.2.7. Metodologi dan Tools Pengembangan

Berisi penjelasan tentang metodologi dan tools pengembangan yang digunakan oleh pengembang dalam pengembangan produk.

#### 1.2.7.1. Metodologi

Pengembangan ulang (*reengineering*) aplikasi Tapatupa dilakukan dengan mengintegrasikan pendekatan Agile Scrum dan *User-Centered Design*(UCD). Pendekatan ini dipilih untuk memastikan proses pengembangan berlangsung secara iteratif, fleksibel, serta tetap berfokus pada kebutuhan pengguna yang telah dianalisis melalui sistem versi sebelumnya, studi aplikasi sejenis, dan ketentuan dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024.

Scrum digunakan sebagai kerangka kerja iteratif dalam bentuk sprint, sementara prinsip-prinsip UCD diimplementasikan melalui tahapan yang terstruktur sesuai ISO 9241-210. Tahapan tersebut meliputi:

##### 1. *Understand Context of Use*

Pada tahap ini, pengembang mengidentifikasi konteks penggunaan berdasarkan evaluasi sistem Tapatupa versi sebelumnya, studi terhadap aplikasi serupa, serta analisis ketentuan regulatif. Hasil dari tahap ini digunakan untuk menyusun asumsi awal mengenai kebutuhan pengguna, hambatan dalam proses digitalisasi aset, dan fungsionalitas yang diperlukan.

##### 2. *Specify User Requirements*

Berdasarkan konteks yang telah dipahami, tim merumuskan kebutuhan pengguna dalam bentuk user needs list dan feature definition, yang kemudian diuraikan menjadi daftar backlog untuk pengembangan.

Penekanan diarahkan pada kebutuhan administratif, kemudahan penggunaan, dan dukungan terhadap proses pembayaran digital.

3. *Produce Design Solutions (Low-Fidelity to High-Fidelity Prototype)*

Tim merancang solusi awal dalam bentuk low-fidelity wireframe, yang kemudian disempurnakan menjadi prototipe high-fidelity seiring proses iterasi sprint. Setiap perubahan desain dilakukan berdasarkan hasil evaluasi terhadap fungsi-fungsi sebelumnya dan hasil *User Acceptance Test* (UAT).

4. *Evaluate Against Requirements*

Evaluasi dilakukan secara bertahap melalui uji fungsional internal dan observasi UAT oleh BKAD dan bank daerah. Pada akhir pengembangan, penyebaran kuesioner dilakukan untuk mengevaluasi tingkat kepuasan pengguna terhadap aplikasi hasil *reengineering*.

Proses pengembangan sistem dilaksanakan dalam tiga iterasi (sprint) sebagai berikut:

a. Sprint 1 - Pengembangan MVP Sistem

Fokus pada formulir permohonan, fitur unggah dokumen, tampilan tagihan sewa, dan integrasi informasi *virtual account*. Validasi awal dilakukan melalui pengujian black-box internal.

b. Sprint 2 - Prototipe High-Fidelity dan UAT

Iterasi ini mencakup pengembangan modul objek retribusi, tarif, perjanjian, autentikasi, dan pengelolaan profil pengguna. Validasi dilakukan melalui observasi proses UAT oleh pihak BKAD dan bank daerah.

c. Sprint 3 - Penyempurnaan dan Evaluasi Pengguna

Penyempurnaan antarmuka dan fungsionalitas sistem berdasarkan masukan dari UAT. Evaluasi akhir dilakukan melalui penyebaran kuesioner untuk mengukur kepuasan pengguna terhadap hasil pengembangan.

Dengan integrasi ini, pendekatan Agile Scrum memberikan kerangka kerja iteratif yang memungkinkan penyesuaian backlog secara berkala dalam setiap sprint, sementara UCD memastikan bahwa desain dan fitur dikembangkan berdasarkan analisis kebutuhan pengguna yang diperoleh dari sistem sebelumnya dan regulasi yang berlaku.

### 1.2.7.2. Tools Pengembangan

Spesifikasi *development tools* yang digunakan dalam pengembangan aplikasi *mobile* Tapatupa akan diuraikan pada Tabel 1.12.

**Tabel 1. 6 S/W Development Tools**

Groups	Tools	Spesification
Code Editor	Visual Studio Code	Visual Studio Code Visual studio code adalah sebuah editor kode sumber (source code editor) yang dikembangkan oleh Microsoft. Ini adalah salah satu editor kode sumber yang paling populer yang digunakan oleh pengembang. Visual Studio code yang digunakan Versi terbaru 1.65
Image editor	canva	Canva merupakan platform desain grafis online yang memungkinkan pengguna untuk membuat berbagai jenis materi desain tanpa perlu keterampilan desain yang mendalam. Canva yang digunakan adalah Versi Terbaru :2.155
Design	Diagram Visual paradigm	Diagram Visual paradigm Visual Paradigm adalah penyedia solusi perangkat lunak Transformasi Bisnis dan TI yang terkemuka secara global. Solusi ini membantu organisasi meningkatkan kelincahan bisnis dan TI serta mendorong inovasi melalui standar terbuka yang populer. Visual Paradigm yang digunakan adalah versi Versi 16.5.1
	Bizagi	Bizagi Modeler adalah aplikasi yang digunakan untuk membuat diagram menggunakan BPMN (Business Process Model Notation) pada current system dan target system aplikasi. Bizagi Modeler yang digunakan versi 3.8
Web Server	XAMPP	XAMPP merupakan media atau web server localhost yang bisa digunakan secara offline. Melalui XAMPP, pengguna dapat mengelola database yang berada di localhost tanpa memerlukan akses internet sehingga jika koneksi

Groups	Tools	Spesification
		internet terganggu dan tidak dapat mengakses web server, pengguna tidak lagi perlu khawatir Versi XAMPP yang digunakan adalah versi v3.3.0
Browser	Google Chrome	Google Chrome adalah peramban web lintas platform yang dikembangkan oleh Google. Peramban ini Pertama kali dirilis pada tahun 2008 untuk Microsoft Windows, kemudian diporting ke Android, iOS, Linux, dan mac OS yang menjadikannya sebagai peramban bawaan dalam sistem operasi. versi Google Chrome yang digunakan adalah versi Chrome 109.

## **BAB II**

### **PROJECT PLANNING (PP)**

#### **(PERENCANAAN PENGEMBANGAN PRODUK)**

##### **2.1. PENDAHULUAN**

Pemerintah daerah memiliki tanggung jawab penting dalam mengelola aset milik daerah, termasuk tanah dan bangunan yang tergolong sebagai Barang Milik Daerah (BMD). Aset-aset ini tidak hanya mendukung pelayanan publik, tetapi juga berperan sebagai sumber pendapatan daerah yang perlu dikelola secara efektif dan akuntabel. Di Kabupaten Tapanuli Utara, proses pengelolaan aset tanah sebelumnya dilakukan secara manual tanpa sistem digital terintegrasi. Hal ini menimbulkan sejumlah kendala, antara lain keterlambatan pencatatan, potensi kesalahan administrasi, dan terbatasnya akses data oleh pemangku kepentingan.

Untuk mengatasi permasalahan tersebut, Pemerintah Kabupaten Tapanuli Utara mengembangkan aplikasi Tapatupa sebagai sistem digital pengelolaan aset tanah daerah. Aplikasi ini memiliki cakupan fungsional seperti pencatatan aset, permohonan penyewaan, serta pengelolaan pembayaran retribusi. Namun, hasil observasi terhadap aplikasi versi awal menunjukkan adanya kekurangan dari sisi fungsionalitas, seperti tidak tersedianya fitur unggah dokumen pada saat pengajuan permohonan, tidak adanya tampilan status permohonan secara langsung, dan belum tersedia sistem pembayaran yang terintegrasi dengan *virtual account*.

Selain itu, struktur antarmuka pengguna pada sistem sebelumnya menyebabkan pengguna kesulitan dalam mengakses informasi penting, seperti tarif retribusi dan detail objek aset. Pengguna juga diharuskan mengunggah bukti pembayaran secara manual, karena sistem belum mendukung pelacakan otomatis terhadap status pembayaran melalui *virtual account*.

Berdasarkan temuan tersebut, dilakukan proyek *reengineering* terhadap aplikasi Tapatupa dengan fokus pada pengembangan sistem berbasis *mobile* dan penyesuaian terhadap Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024. Rencana pengembangan ini mencakup pembuatan ulang modul-modul utama yang telah ditentukan, termasuk peningkatan aksesibilitas, integrasi sistem pembayaran,

serta penyempurnaan tampilan dan struktur data yang mendukung kecepatan serta keakuratan proses administrasi.

Dokumen perencanaan proyek ini disusun untuk mendukung pengembangan ulang sistem, dengan menjelaskan ruang lingkup proyek, tujuan, strategi pengembangan, penjadwalan, serta kebutuhan sumber daya yang diperlukan. Perencanaan ini menjadi dasar pelaksanaan *reengineering* aplikasi agar sesuai dengan kebutuhan operasional di lapangan dan mendukung tata kelola aset daerah secara sistematis dan terdokumentasi.

## **2.2. DESKRIPSI PROYEK**

Proyek ini merupakan pengembangan ulang (*reengineering*) aplikasi Tapatupa, yaitu sistem informasi penyewaan aset tanah milik daerah yang digunakan oleh Badan Keuangan dan Aset Daerah (BKAD) Kabupaten Tapanuli Utara. Aplikasi ini sebelumnya dikembangkan untuk memfasilitasi pencatatan aset, pengelolaan permohonan penyewaan, dan pembayaran retribusi secara digital. Berdasarkan hasil observasi terhadap sistem lama, ditemukan sejumlah kekurangan pada fitur utama, alur interaksi pengguna, dan pengelolaan data yang belum sepenuhnya mendukung kebutuhan masyarakat sebagai pengguna layanan.

Proyek ini ditujukan secara khusus kepada masyarakat Kabupaten Tapanuli Utara yang menjadi wajib retribusi atas penyewaan aset tanah daerah. Pengguna aplikasi dapat mengakses layanan melalui perangkat *mobile* untuk mengajukan permohonan sewa, mengunggah dokumen, memantau status pengajuan, serta melakukan pembayaran retribusi secara terintegrasi dengan layanan *virtual account*. Aplikasi ini tidak ditujukan untuk penggunaan oleh petugas internal pemerintah daerah, karena proses validasi dan pengelolaan aset tetap dilakukan melalui sistem *backend* terpisah.

Tujuan dari proyek ini adalah untuk menghadirkan sistem aplikasi berbasis *mobile* yang sesuai dengan ketentuan Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024, serta mengembangkan ulang aplikasi Tapatupa dengan menambahkan fitur unggah dokumen pada modul permohonan, menampilkan *virtual account* secara

langsung dalam halaman tagihan, serta menghilangkan keharusan mengunggah bukti pembayaran secara manual.

Sistem ini juga menyederhanakan navigasi formulir permohonan dan menyediakan filter pencarian pada objek retribusi dan tarif berdasarkan wilayah serta nama objek. Pengembangan dilakukan melalui pendekatan *software reengineering*, dengan tahapan sebagai berikut:

1. *Inventory Analysis*

Analisis sistem lama dilakukan untuk mengidentifikasi kelemahan dari segi fitur, alur permohonan, proses pembayaran, serta hambatan dalam penggunaan di lapangan.

2. *Document Restructuring*

Dokumentasi teknis dari sistem sebelumnya disusun ulang untuk memberikan referensi yang terstruktur bagi pengembangan ulang sistem.

3. *Code Refactoring*

Struktur kode lama diperbaiki agar lebih modular, bersih, dan memenuhi standar pengembangan aplikasi *mobile*.

4. *Data Restructuring*

Struktur basis data didesain ulang untuk mendukung pengelolaan data aset, kontrak, dan pembayaran secara transparan.

5. *Forward Engineering*

Aplikasi dikembangkan kembali menggunakan *framework* Flutter, dengan fokus pada implementasi fitur baru seperti unggah dokumen permohonan, pencarian data aset dan tarif, pembuatan perjanjian digital, serta integrasi pembayaran dengan *virtual account*.

Manajemen proyek dilakukan dengan mengintegrasikan metode Agile Scrum dan pendekatan *User-Centered Design*(UCD). Pengembangan dilaksanakan dalam tiga iterasi, dengan masing-masing iterasi difokuskan pada pembangunan fitur utama, penyempurnaan sistem, dan evaluasi dengan menyebarkan kuesioner pada iterasi ketiga untuk mengukur kepuasan pengguna serta melakukan UAT antara pihak BKAD dengan bank daerah untuk memvalidasi integrasi *virtual account* yang dibuat.



Dengan pendekatan ini, sistem yang dirancang ulang diharapkan mampu menjawab kebutuhan masyarakat secara langsung serta meningkatkan transparansi dalam proses penyewaan aset tanah milik daerah.

### 2.2.1. Project Organization

*Project organization* dalam tugas akhir *Reengineering* Aplikasi Tapatupa berbasis *Mobile* digunakan untuk mendukung kelancaran proses *reengineering*, memastikan pembagian peran dan tanggung jawab serta menjaga komunikasi koordinasi tim selama pengerjaan tugas akhir.

Tugas akhir ini dikembangkan oleh tiga mahasiswa tingkat akhir Program Studi Teknologi ekayasa Perangkat Lunak, Institut Teknologi Del. Masing-masing mahasiswa memiliki peran penting dalam pengembangan produk. Berikut ini adalah rincian pembagian peran dan tanggung jawab tiap anggota tim.



Gambar 2. 1 project Organization

1. Ardiles Sinaga, S.T., M.T. - Dosen Pembimbing

Berperan sebagai pembimbing utama yang memberikan arahan, saran, serta evaluasi terhadap perkembangan proyek. Membantu memastikan proyek berjalan sesuai metodologi yang tepat serta memenuhi standar akademik.

2. Gabriel Ignatius Situmeang

Berperan sebagai *Full Stack Developer* selama pembangunan aplikasi Tapatupa.

3. Yudi Frandianto Saragih

Berperan sebagai desain UI aplikasi Tapatupa.

#### 4. Anita Luciana Munthe

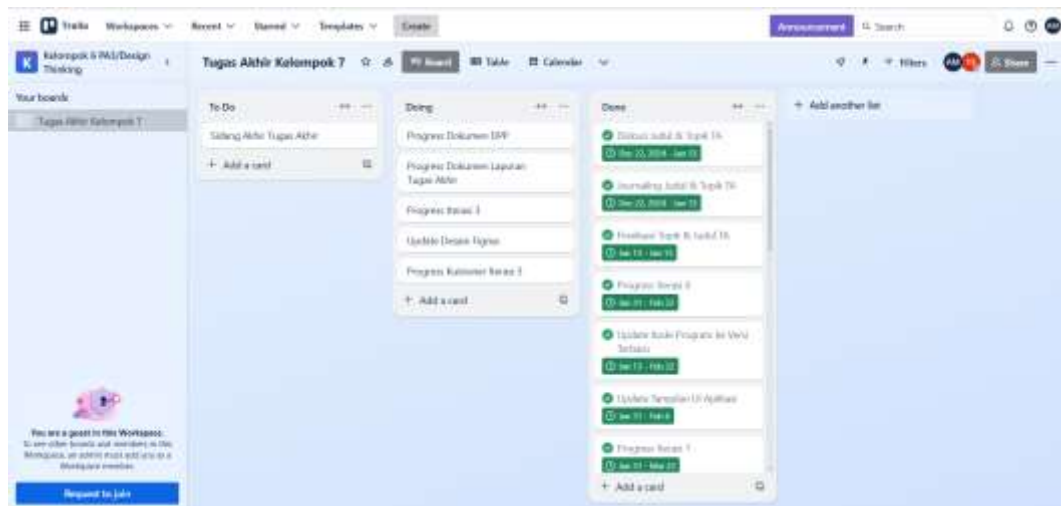
Berperan sebagai Project Manager dari Kelompok 7 TA 2025. Bertanggung jawab dalam perencanaan, koordinasi tugas, serta memastikan proyek selesai tepat waktu dengan hasil yang optimal.

### 2.2.2. Work Breakdown Structure (WBS)

*Work Breakdown Structure* (WBS) adalah metode yang digunakan untuk membagi proyek menjadi bagian-bagian kecil yang lebih terstruktur, sistematis, dan mudah dikelola. WBS berfungsi untuk mempermudah perencanaan, pengelolaan sumber daya, pengendalian waktu, serta monitoring kemajuan proyek.

Penyusunan *work breakdown structure* membantu memperkirakan timeline penyelesaian proyek serta mempermudah koordinasi anggota tim. Struktur *work breakdown structure* dibagi menjadi beberapa komponen utama yang dipecah menjadi sub-komponen dan kemudian dibagi kembali menjadi tugas-tugas kecil yang lebih spesifik. Pada tugas akhir, penerapan *work breakdown structure* menggunakan *tools Trello*.

Dalam proyek *reengineering* aplikasi Tapatupa oleh Kelompok 7 TA 2025, WBS mencakup beberapa aspek utama berikut:



Gambar 2. 2 *Work Breakdown Structure*

### 2.2.3. Budget

*Budget* berfokus pada alokasi anggaran yang dirancang untuk mendukung keseluruhan proses pengembangan Aplikasi Tapatupa berbasis *Mobile*. Anggaran

terbagi ke dalam tiga kategori utama, yaitu biaya pengembangan perangkat lunak, biaya tenaga kerja, serta biaya tak terduga. Pada tahap pengembangan perangkat lunak, dana dialokasikan untuk enam fase utama. Selanjutnya, pada kategori biaya tenaga kerja, sebagian besar tenaga kerja berasal dari internal tim sehingga tidak memerlukan alokasi dana khusus. Dalam proses *reengineering* aplikasi Tapatupa, diperlukan alokasi anggaran untuk mendukung kegiatan teknis dan non-teknis selama masa pengembangan. Budget ini disusun berdasarkan kebutuhan sumber daya manusia, perangkat lunak, perangkat keras, serta kebutuhan operasional lainnya.

**Tabel 2. 1 Budget**

<b>Komponen Biaya</b>	<b>Jumlah</b>	<b>Satuan</b>	<b>Total</b>	<b>Catatan</b>
A. Tahap Pengembangan Perangkat Lunak				
1. <i>Requirement Gathering</i>	1 tahap	Rp 3.000.000	Rp 3.000.000	Estimasi untuk 1 tahap
2. Analisis	1 tahap	Rp 10.000.000	Rp 10.000.000	-
3. Desain	1 tahap	Rp 7.000.000	Rp 7.000.000	-
4. Implementasi	10.000 LOC	Rp 500/LOC	Rp 5.000.000	Berdasarkan estimasi LOC
5. <i>Testing &amp; Debugging</i>	1 tahap	Rp 3.000.000	Rp 3.000.000	-
6. <i>Maintenance</i>	1 tahap	Rp 2.500.000	Rp 2.500.000	-
<b>Total</b>			<b>Rp 30.500.000</b>	
B. Biaya untuk pengembang perangkat lunak				

<b>Komponen Biaya</b>	<b>Jumlah</b>	<b>Satuan</b>	<b>Total</b>	<b>Catatan</b>
1. <i>Project Manager</i>	3 bulan	Rp 10.000.000	Rp 30.000.000	Diperhitungkan untuk per bulan
2. <i>System Analyst</i>	3 bulan	Rp 8.000.000	Rp 24.000.000	Diperhitungkan untuk per bulan
3. <i>Programmer</i>	3 bulan	Rp 7.000.000	Rp 21.000.000	Diperhitungkan untuk per bulan
<b>Total</b>			<b>Rp 75.000.000</b>	
C. Biaya Tak Terduga				
1. Revisi atas perubahan kebutuhan dari pengguna/ <i>client</i>	1 cycle	Rp 3.500.000	Rp 3.500.000	-
2. Perbaikan teknis ( <i>hardware/software</i> )	1 tahap	Rp 2.000.000	Rp 2.000.000	-
<b>Total</b>	Total		<b>Rp 5.500.000</b>	
<b>Total Biaya Proyek</b>			<b>Rp 111.000.000</b>	Belum termasuk margin keuntungan

#### 2.2.4. Tools

Bagian ini menjelaskan alat dan teknologi yang digunakan selama pelaksanaan proyek. Hal-hal yang dijelaskan meliputi perangkat keras dan perangkat lunak pendukung, *tools* manajemen proyek, alat bantu kolaborasi tim, tools untuk simulasi, desain, pengujian, serta alasan pemilihan tools berdasarkan kebutuhan proyek.

**Tabel 2. 2 Tools**

No	Hardware	Software	Tool manajemen proyek	Alat bantu untuk kolaborasi tim	Tools untuk simulasi, desain, pengujian	Alasan pemilihan tools
1.	Laptop	Visual Studio Code, Flutter	ClickUp	Github	Figma, Google Chrome	mendukung prinsip <i>reengineering</i> yaitu meningkatkan kualitas, mempercepat siklus pengembangan, serta menghasilkan aplikasi yang lebih modern, <i>scalable</i> , dan sesuai dengan kebutuhan pengguna dan stakeholder di Kabupaten Tapanuli Utara.

#### **2.2.5. Resiko dan Hambatan**

Dalam pelaksanaan proyek ini, terdapat berbagai potensi risiko dan hambatan yang perlu diantisipasi untuk memastikan kelancaran dan keberhasilan proyek. Oleh karena itu, dilakukan identifikasi terhadap risiko utama yang mungkin terjadi, penilaian terhadap tingkat dampak dan kemungkinan terjadinya, serta penyusunan

strategi mitigasi untuk mengurangi dampak jika risiko tersebut benar-benar terjadi. Berikut ini adalah daftar risiko dan hambatan yang telah diidentifikasi selama pelaksanaan proyek Tugas Akhir.

**Tabel 2. 3 Risiko dan Hambatan**

No	Kategori	Risiko	Dampak	Kemungkinan	Strategi Mitigasi
1	Waktu	Keterlambatan penyelesaian proyek	<i>Sedang</i>	<i>Sedang</i>	<i>Membuat timeline realistis, memonitor progres menggunakan ClickUp secara rutin</i>
2	Teknis	Kesalahan aplikasi (bug, error) saat pengembangan atau saat di hosting	Tinggi	Sedang	Melakukan pengujian berkala di server lokal dan menggunakan GitHub untuk kontrol versi
3	Komunikasi	Perbedaan Pendapat Antar Anggota Tim	Sedang	Rendah	Melakukan diskusi dan mendokumentasikan hasil keputusan agar semua anggota tim memiliki pemahaman yang sama.
4	Dokumentasi	.Produk Tidak Dimengerti oleh Pengguna	Sedang	Sedang	Membuat dokumentasi atau panduan pengguna yang mudah dipahami, melakukan pengujian bersama pengguna sebelum sistem diluncurkan, dan menyediakan sesi

No	Kategori	Risiko	Dampak	Kemungkinan	Strategi Mitigasi
					pelatihan apabila dibutuhkan.

### **BAB III**

#### **PRODUCT DESIGN (PD)**

#### **(DESAIN PENGEMBANGAN PRODUK)**

##### **3.1.PENDAHULUAN**

Bagian ini membahas desain pengembangan aplikasi Tapatupa versi *mobile* yang merupakan hasil dari proses pengembangan ulang (*reengineering*) terhadap sistem pengelolaan aset tanah milik daerah yang telah ada sebelumnya. Produk ini dirancang untuk menggantikan proses permohonan dan pembayaran retribusi yang sebelumnya dilakukan secara manual, dengan dukungan antarmuka *mobile* yang dapat diakses langsung oleh masyarakat sebagai pengguna utama.

Perancangan dilakukan berdasarkan dokumen spesifikasi kebutuhan dan rencana pengembangan yang telah disusun pada tahap sebelumnya, dengan mengacu pada ketentuan dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024. Desain produk mencakup perbaikan alur permohonan, unggah dokumen, penyusunan perjanjian, serta integrasi dengan sistem pembayaran digital, sehingga pengguna dapat melakukan seluruh proses secara *online* melalui perangkat *mobile*.

Pendekatan desain mengintegrasikan prinsip *User-Centered Design*(UCD) yang berfokus pada kebutuhan pengguna, serta praktik pengembangan sistem yang disesuaikan untuk perangkat *mobile*. Setiap fitur dirancang dengan mempertimbangkan hasil observasi sistem sebelumnya dan kebutuhan administratif sesuai regulasi, sehingga sistem dapat mendukung proses penyewaan yang lebih efisien dan terdokumentasi secara digital.

##### **3.2. DESKRIPSI PRODUK DESIGN**

Desain produk untuk aplikasi Tapatupa berbasis *mobile* dikembangkan untuk mendukung proses digitalisasi penyewaan aset tanah milik Pemerintah Kabupaten Tapanuli Utara. Sistem ini dibangun ulang atau *reengineering* fokus pada digitalisasi alur kerja yang sebelumnya dilakukan secara manual atau terbatas melalui sistem web internal, yang berdampak pada keterbatasan akses pengguna dan tingginya potensi kesalahan dalam pencatatan serta proses administrasi.

Proses bisnis utama yang didukung oleh aplikasi ini mencakup enam alur berikut:



1. Registrasi dan Autentikasi Pengguna

Pengguna dapat membuat akun dengan mengisi formulir digital yang telah disesuaikan dengan persyaratan identitas wajib retribusi. Proses autentikasi dilakukan untuk memastikan hanya pengguna terdaftar yang dapat mengakses layanan.

2. Pengajuan Permohonan Penyewaan Aset

Pengguna mengisi formulir permohonan sewa yang mencakup identitas, objek yang dimohonkan, dan mengunggah dokumen administratif sesuai ketentuan Peraturan Bupati Nomor 12 Tahun 2024. Formulir dilengkapi validasi format dan status pengajuan dapat dipantau secara langsung.

3. Pencarian Objek dan Tarif Retribusi

Sistem menyediakan informasi objek retribusi dan tarif dalam format tabel yang dapat disaring berdasarkan wilayah kecamatan atau nama objek. Hal ini memudahkan pengguna untuk menemukan informasi spesifik tanpa harus membaca keseluruhan data.

4. Persetujuan Permohonan dan Penerbitan Perjanjian

Setelah permohonan diterima, petugas pengelola aset meninjau data dan dokumen, kemudian menerbitkan perjanjian sewa digital yang dapat diakses dan diunduh oleh pengguna langsung dari aplikasi.

5. Tagihan dan Pembayaran Menggunakan *Virtual account*

Sistem menampilkan tagihan secara otomatis setelah perjanjian dibuat. Tagihan berisi nominal retribusi dan nomor *Virtual account* (VA) yang terhubung ke sistem bank daerah. Pembayaran dilakukan tanpa perlu unggah bukti bayar, karena sistem mencatat transaksi secara otomatis melalui integrasi dengan sistem bank.

6. Riwayat Pembayaran dan Permohonan

Pengguna dapat melihat daftar transaksi pembayaran dan permohonan sebelumnya lengkap dengan tanggal, status, serta dokumen terkait.

Setiap proses di atas divisualisasikan dalam bentuk diagram *Business Process Model and Notation* (BPMN) agar memudahkan pemahaman teknis, baik oleh tim pengembang, penguji, maupun *stakeholder* lain. Diagram BPMN menggambarkan aktor, alur keputusan, dan sistem yang terlibat secara jelas.

Desain ini disusun menggunakan pendekatan modular. Setiap fitur dikembangkan sebagai komponen independen, seperti modul permohonan, modul objek retribusi, modul pembayaran, dan modul perjanjian. Pendekatan ini mempermudah pemeliharaan sistem di masa depan dan memungkinkan integrasi dengan layanan digital lainnya, seperti sistem informasi keuangan daerah atau sistem verifikasi identitas digital.

Dengan dukungan antarmuka *mobile* yang telah disesuaikan, sistem memungkinkan pengguna untuk mengakses seluruh proses secara langsung dari perangkat ponsel mereka. Hal ini mendukung tujuan peningkatan kecepatan layanan, pengurangan interaksi manual, dan transparansi data pada pengelolaan aset tanah milik daerah..

### **3.2.1. Proses Bisnis Target System**

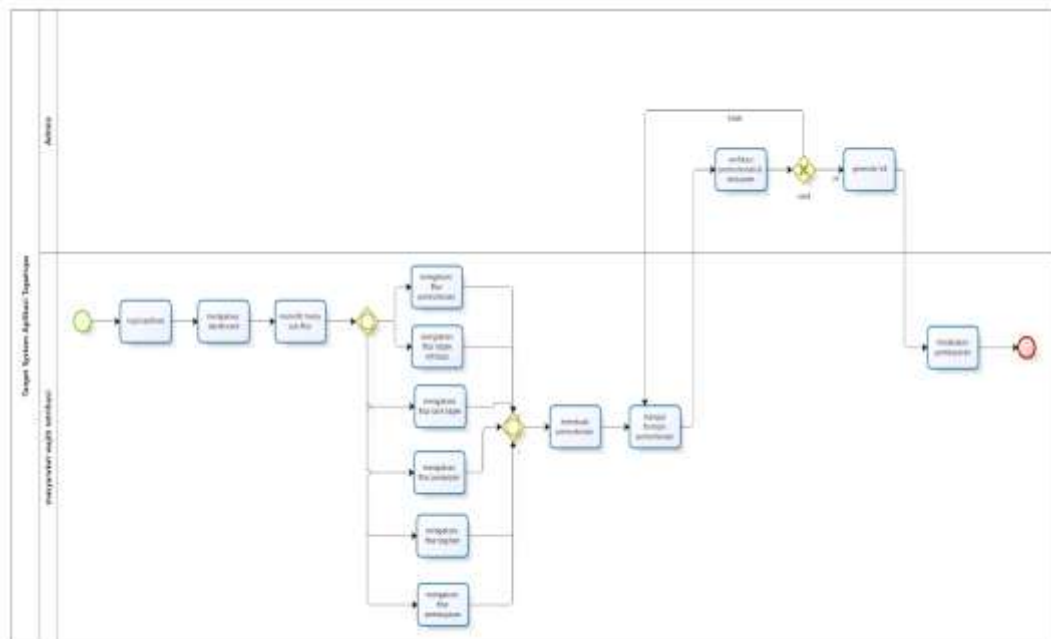
*Target system* yang dirancang merupakan aplikasi Tapatupa berbasis *mobile* yang dikembangkan ulang melalui pendekatan *software reengineering*. Pengembangan ini dilakukan sebagai bagian dari tugas akhir mahasiswa, dengan tujuan memperbaiki sistem sebelumnya agar lebih sesuai dengan kebutuhan masyarakat sebagai wajib retribusi dalam proses penyewaan tanah milik pemerintah daerah. Fokus utama sistem adalah menyediakan alur layanan yang lebih sederhana, mudah diakses melalui perangkat *mobile*, serta sesuai dengan prosedur yang diatur dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024.

Sistem ini akan menggantikan beberapa proses manual menjadi berbasis digital, sehingga pengguna dapat menjalankan layanan penyewaan tanah secara mandiri melalui aplikasi. Proses dimulai ketika pengguna melakukan *login* ke dalam aplikasi dan diarahkan menuju halaman *dashboard*. Di dalam *dashboard*, pengguna dapat memilih fitur layanan yang tersedia, termasuk melihat informasi objek retribusi, tarif retribusi, perjanjian, tagihan, serta status pembayaran. Untuk mempermudah akses terhadap informasi, sistem menyediakan fitur filter pencarian berdasarkan kategori seperti kecamatan, jenis objek, atau nama objek tertentu.

Setelah mendapatkan informasi yang dibutuhkan, pengguna dapat membuat permohonan sewa dengan mengisi formulir digital dan mengunggah dokumen

pendukung sesuai ketentuan. Permohonan tersebut akan diproses oleh sistem dan diverifikasi oleh petugas untuk memastikan kelengkapan dan kesesuaian data. Jika data dinyatakan valid, maka sistem akan secara otomatis menghasilkan nomor *Virtual account* (VA) sebagai tujuan pembayaran. Pengguna kemudian dapat melakukan pembayaran langsung melalui bank yang telah ditentukan, tanpa perlu melakukan unggah bukti pembayaran secara manual.

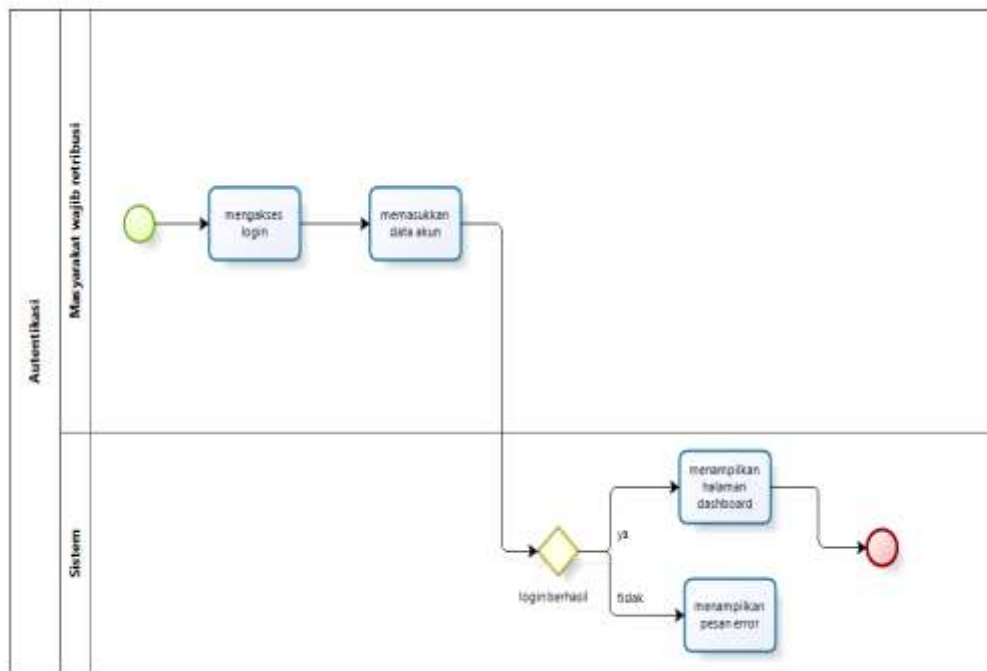
Seluruh proses tersebut divisualisasikan dalam diagram *Business Process Model and Notation* (BPMN) yang ditampilkan pada Gambar 3.1. Proses bisnis ini dirancang untuk membantu pengguna dalam mengakses layanan penyewaan tanah secara langsung, tepat sesuai tahapan prosedur yang berlaku, dan dapat dijalankan dari berbagai lokasi melalui perangkat *mobile*. Dengan sistem ini, proses administrasi yang sebelumnya dilakukan secara manual dapat dilakukan secara digital dalam satu *platform* terintegrasi.



**Gambar 3. 1 Helicopter view target system**

### 3.2.1.1. Business Process Login

Pada proses *login*, pengguna (masyarakat wajib retribusi) harus *login* kedalam aplikasi untuk dapat mengakses aplikasi tersebut. *Login* dilakukan agar masyarakat dapat melakukan pengajuan permohonan sewa aset tanah pada aplikasi tersebut, *business process* dapat dilihat pada gambar dibawah ini.



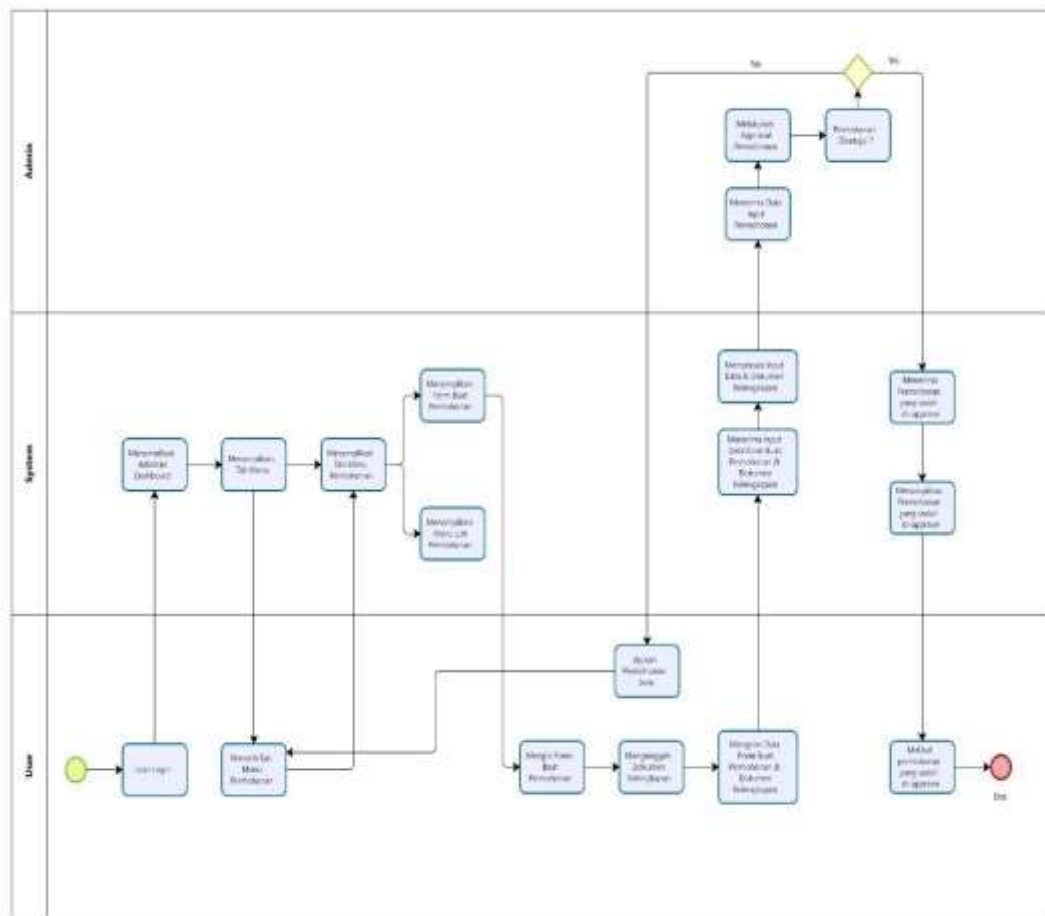
**Gambar 3. 2 Business Process Login**

#### **3.2.1.1.1. Service Time**

*Service time* yang diperlukan oleh masyarakat dalam melakukan *login*  $\pm 25$  detik.

#### **3.2.1.2. Business Process Permohonan**

Pada *business process* permohonan, pengguna (masyarakat wajib retribusi) melakukan *login* terlebih dahulu. Kemudian, memilih menu permohonan dan mengajukan permohonan. Pengguna akan diarahkan untuk mengisi formulir yang dibutuhkan serta mengunggah dokumen yang diperlukan. Sistem akan menerima dan mengirim kepada admin untuk melakukan pengecekan dan persetujuan dari admin, apabila permohonan disetujui maka, sistem akan menampilkan permohonan yang sudah di setujui dan pengguna dapat melihat informasi setuju permohonan.

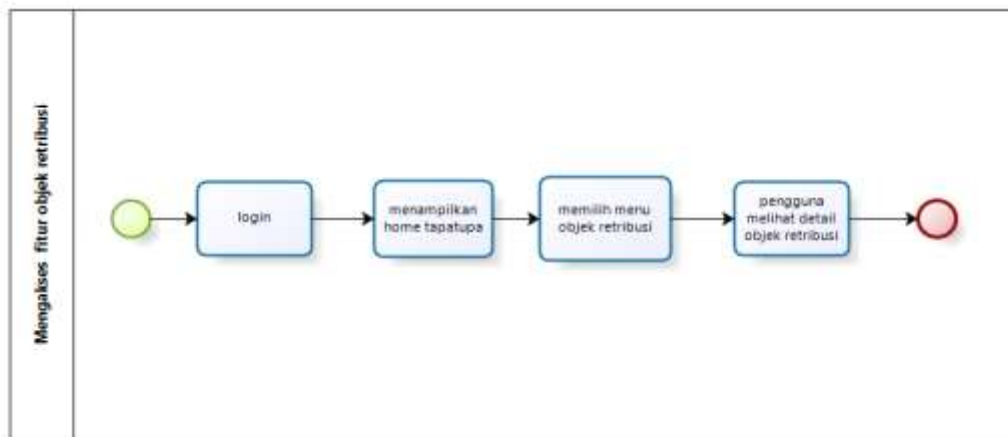


#### 3.2.1.2.1 Service Time

*Service time* yang diperlukan oleh masyarakat dalam mengajukan permohonan, mengisi formulir permohonan, melampirkan dokumen, serta mengecek ulang data biasanya membutuhkan waktu  $\pm 7$  menit tergantung kompleksitas formulir dan koneksi internet.

### 3.2.1.3 Business Process Fitur Objek Retribusi

Pada *business process* objek retribusi, pengguna melakukan *login* terlebih dahulu, kemudian memilih menu objek retribusi untuk dapat melihat detail objek retribusi.



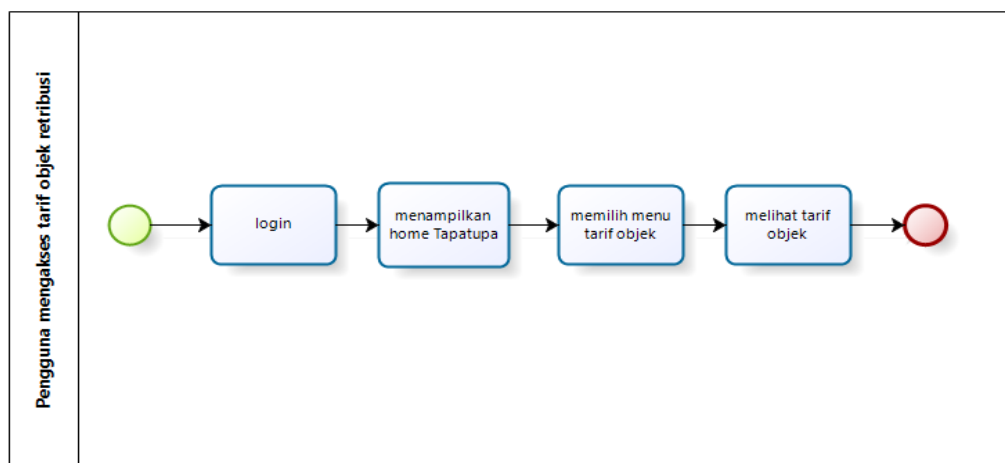
**Gambar 3. 4 Business Process Objek Retribusi**

#### **3.2.1.3.1 Service Time**

*Service time* yang diperlukan oleh masyarakat dalam mengakses objek retribusi, dimulai dari *login*, memilih menu objek retribusi, sampai sistem menampilkan detail objek retribusi  $\pm 25$  detik.

#### **3.2.1.4. Business Process Fitur Tarif Objek**

Pada *business process* tarif objek retribusi, pengguna melakukan *login* terlebih dahulu, kemudian memilih menu tarif objek retribusi untuk dapat melihat detail tarif objek retribusi.



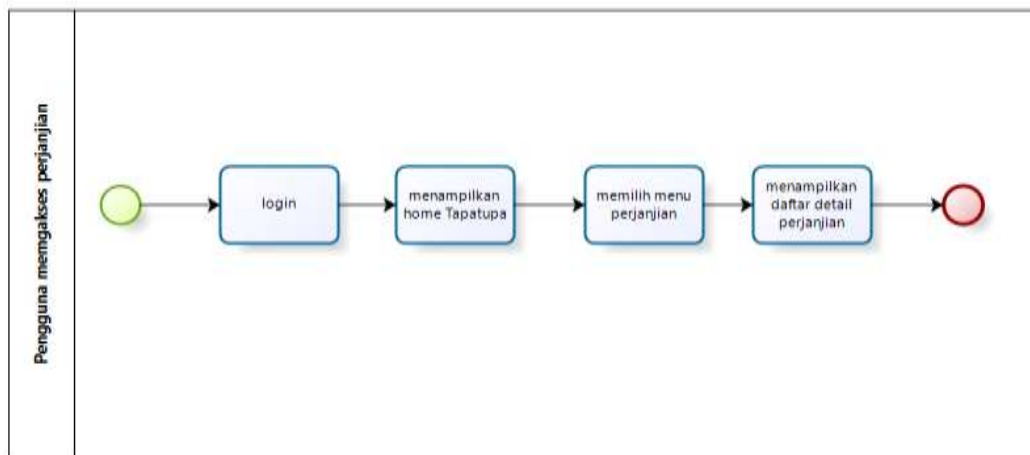
**Gambar 3. 5 Business Process Tarif Objek Retribusi**

#### 3.2.1.4.1. Service Time

*Service time* yang diperlukan oleh masyarakat dalam mengakses tarif objek retribusi, dimulai dari *login*, memilih menu tarif objek retribusi, sampai sistem menampilkan detail tarif objek retribusi  $\pm 25$  detik.

#### 3.2.1.5. Business Process Fitur Perjanjian

Pada *business process* perjanjian, pengguna melakukan *login* terlebih dahulu, kemudian memilih menu perjanjian untuk dapat melihat detail perjanjian.



Gambar 3. 6 Business Process Perjanjian

#### 3.2.1.5.1. Service Time

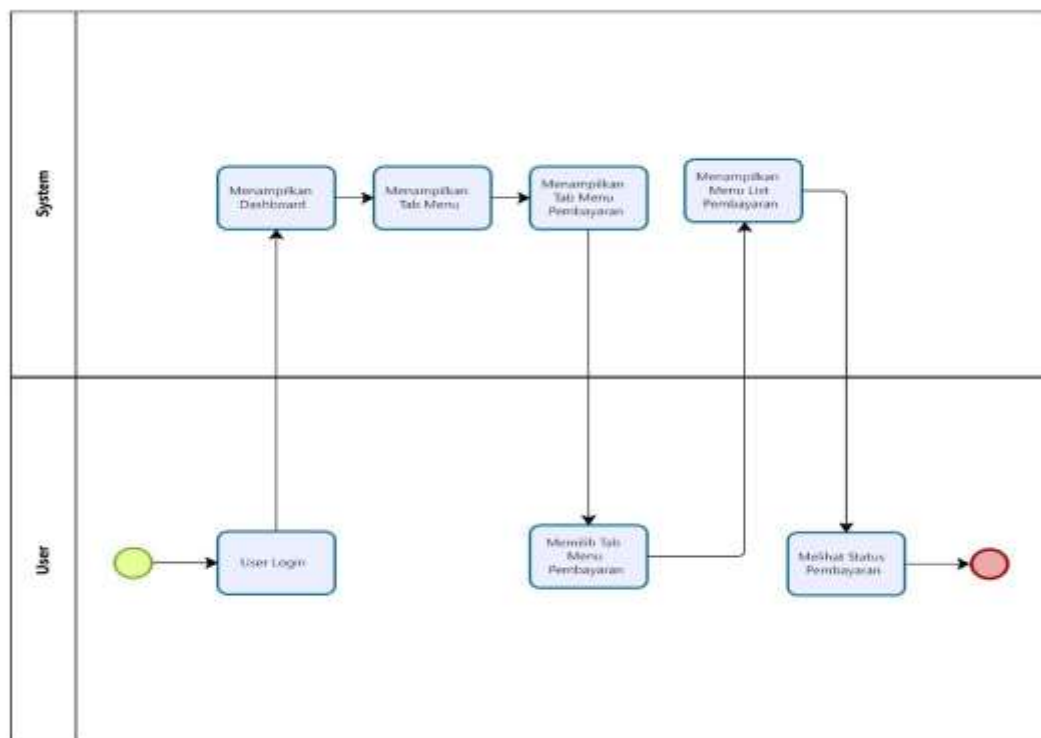
*Service time* yang diperlukan oleh masyarakat dalam mengakses perjanjian, dimulai dari *login*, memilih menu perjanjian, sampai sistem menampilkan detail perjanjian  $\pm 25$  detik.

#### 3.2.1.6. Business Process Fitur Tagihan

Pada *business process* tagihan, pengguna (masyarakat wajib retribusi) melakukan *login* terlebih dahulu. Kemudian, pengguna memilih menu fitur tagihan untuk melihat *list* tagihan penyewaan, sistem akan menampilkan tagihan. Pengguna dapat memilih tagihan apabila *list* tagihan belum dibayar. Sistem akan menampilkan detail pembayaran, nomor *virtual account*, dan petunjuk pembayaran. Apabila pengguna sudah melakukan pembayaran, maka sistem akan menampilkan status sudah bayar.







**Gambar 3. 8 Business Process Pembayaran**

#### 3.2.1.7.1 Service Time

*Service time* yang diperlukan oleh masyarakat mulai dari melakukan *login*, melihat detail *list* pembayaran, dan status pembayaran mencapai  $\pm 15 - 25$  detik. Estimasi ini mempertimbangkan waktu *respons* sistem dalam melakukan autentikasi, pengambilan data tagihan, serta proses verifikasi status pembayaran melalui integrasi dengan sistem *virtual account* yang bersifat *asynchronous*.

#### 3.2.2. Use Case Diagram

Pada bab ini dijelaskan deskripsi fungsi-fungsi yang terdapat pada aplikasi Tapatupa sebelum dan sesudah dilakukan proses *reengineering*, yang digambarkan melalui *use case* diagram dan *use case scenario*. Diagram ini berperan penting dalam merancang dan memahami kebutuhan sistem dari perspektif pengguna, serta sebagai dasar untuk pengembangan fitur dan antarmuka yang sesuai dengan proses bisnis. Spesifikasi fungsi sebelum dilakukan *reengineering* merepresentasikan kondisi awal sistem yang masih memiliki sejumlah keterbatasan, sedangkan spesifikasi fungsi setelah *reengineering* menunjukkan perbaikan dan penyesuaian berdasarkan kebutuhan pengguna yang telah diidentifikasi melalui hasil observasi

pada aplikasi lama, aplikasi sejenis, dan dokumen Peraturan Bupati No. 12 tahun 2024.

Dokumentasi spesifikasi fungsi ini menjadi acuan penting dalam tahapan pengujian, karena setiap *requirement* yang telah diidentifikasi dan diperbaiki harus dapat dibuktikan melalui pengujian bahwa sistem hasil *reengineering* benar-benar telah memenuhi kebutuhan pengguna. Spesifikasi fungsi sebelum dan sesudah *reengineering* ditampilkan dalam bentuk gambar *use case* diagram dan disertai *use case scenario* pada setiap fungsi yang sudah di *reengineering*.

#### **3.2.2.1. Aplikasi Tapatupa sebelum di *reengineering***

*Use case* diagram sebelum *reengineering* menggambarkan interaksi antara aktor utama, yaitu Wajib Retribusi, dengan sistem Tapatupa versi awal. Berdasarkan hasil observasi terhadap sistem yang telah berjalan, aplikasi hanya menyediakan fungsi-fungsi dasar untuk mendukung penyewaan aset tanah milik pemerintah daerah. Namun, sejumlah keterbatasan teridentifikasi dalam alur layanan dan fitur sistem.

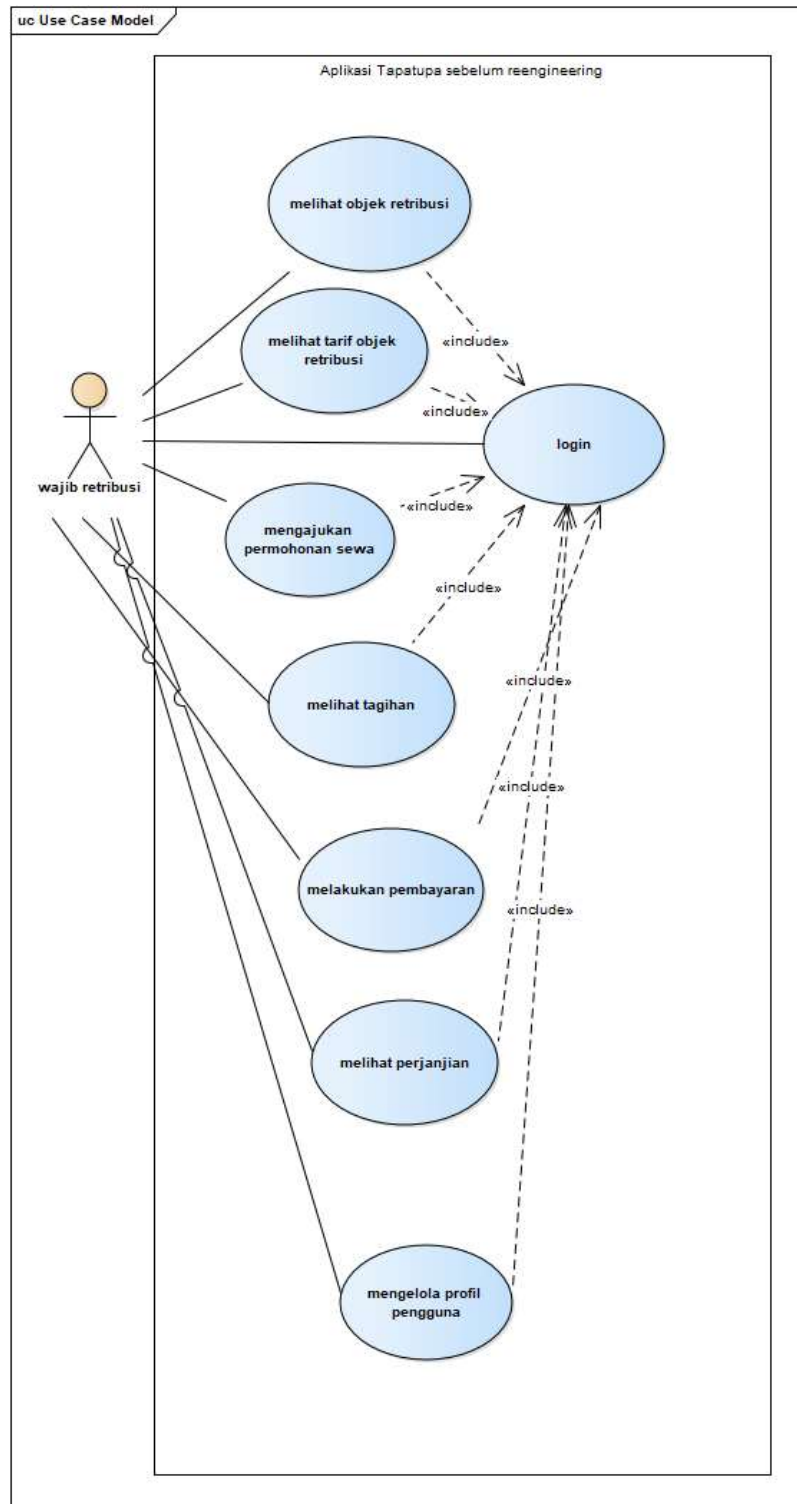
Pengguna dapat melakukan login dan mengakses berbagai menu informasi dasar seperti melihat objek retribusi dan melihat tarif retribusi. Saat mengajukan permohonan, pengguna hanya mengisi formulir tanpa menyertakan dokumen pendukung, karena sistem belum menyediakan fitur unggah berkas. Proses melihat tagihan juga tidak menyertakan status pembayaran maupun nomor *virtual account*, sehingga pembayaran dilakukan melalui saluran eksternal dan mewajibkan pengguna untuk mengunggah bukti bayar secara manual.

Tidak tersedia halaman status permohonan, sehingga pengguna tidak dapat melacak progres permohonannya. Selain itu, riwayat permohonan dan riwayat pembayaran belum disediakan. Sistem juga memungkinkan pengguna untuk melihat perjanjian dan mengelola profil, namun tanpa fitur tambahan seperti dokumen digital atau pembaruan informasi otomatis.

Adapun fungsi sistem sebelum *reengineering* adalah sebagai berikut.

1. Login, autentikasi pengguna wajib retribusi kedalam sistem.
2. Melihat objek retribusi, menampilkan daftar objek tanpa filter pencarian.

3. Melihat tarif retribusi, menampilkan informasi tanpa pencarian per wilayah.
4. Mengajukan permohonan, pengguna mengisi form tanpa unggah dokumen.
5. Melihat tagihan, dengan menampilkan nominal tagihan tanpa status atau *virtual account*.
6. Melakukan pembayaran dengan manual, sistem tidak mencatat status.
7. Melihat perjanjian, dengan menampilkan informasi perjanjian tanpa dokumen digital.
8. Mengelola profil, mengubah data pengguna secara terbatas.



Gambar 3. 9 Usecase sebelum *reengineering*

### 3.2.2.2. Aplikasi Tapatupa setelah di *reengineering*

Setelah dilakukan proses *reengineering*, sistem Tapatupa dikembangkan ulang menjadi aplikasi berbasis *mobile* dengan fungsionalitas yang lebih lengkap dan

sesuai dengan kebutuhan wajib retribusi. Pengembangan ini didasarkan pada hasil analisis terhadap sistem sebelumnya, referensi aplikasi sejenis, serta ketentuan teknis dalam Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024.

Dalam versi terbaru, seluruh proses utama mulai dari permohonan hingga pembayaran diintegrasikan secara digital. Pengguna tetap melakukan login sebagai autentikasi awal, hampir seluruh fitur utama seperti mengakses daftar objek retribusi, melihat tarif retribusi, dan melakukan pembayaran, hanya dapat dilakukan setelah pengguna berhasil login. Maka, login juga memiliki relasi *<include>* terhadap semua fitur tersebut karena autentikasi pengguna merupakan prasyarat sebelum interaksi dengan sistem dapat dilakukan. Namun, fitur-fitur lain diperluas untuk menjawab permasalahan pada sistem lama. Misalnya, pada saat mengajukan permohonan sewa, pengguna kini dapat langsung mengunggah dokumen pendukung, yang akan diverifikasi oleh petugas. Pengguna juga dapat melihat status permohonan secara berkala di dalam sistem.

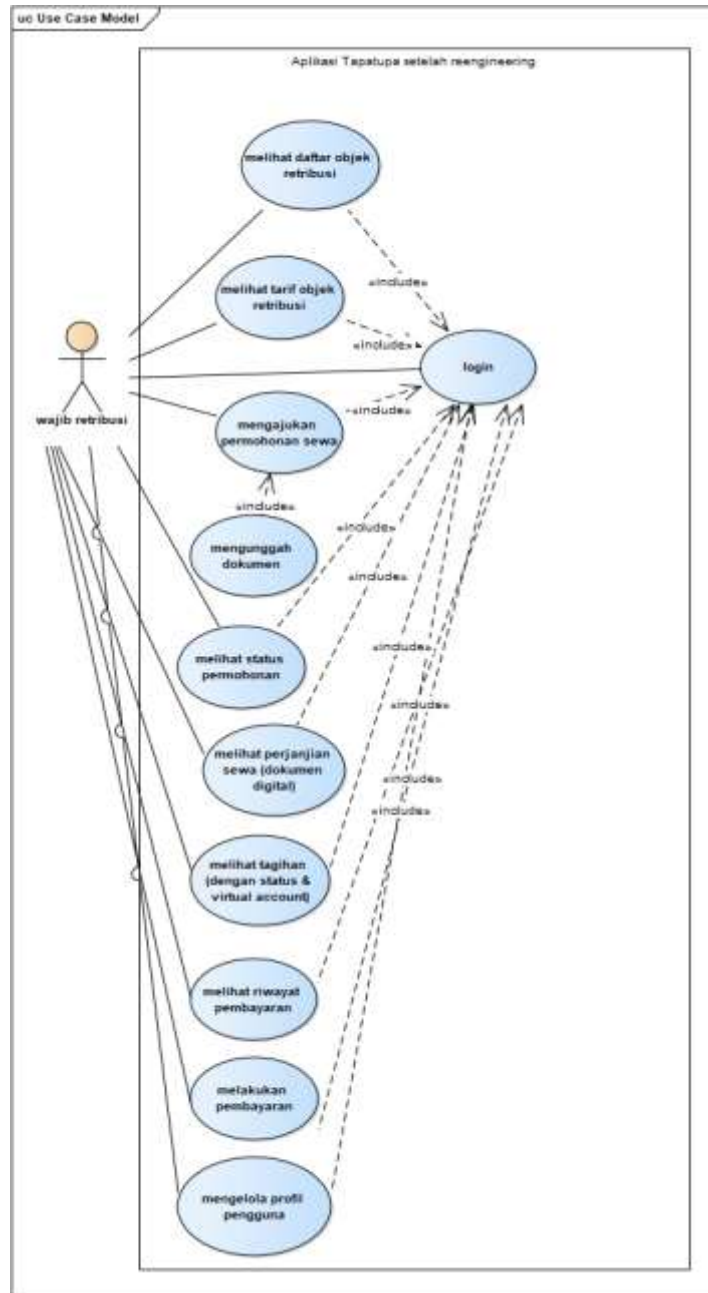
Untuk pembayaran, sistem akan otomatis menghasilkan nomor *virtual account* (VA) setelah permohonan dinyatakan valid, dan status tagihan ditampilkan secara real-time. Fitur riwayat pembayaran dan riwayat permohonan kini tersedia agar pengguna dapat melacak proses sebelumnya. Selain itu, sistem memungkinkan akses langsung terhadap dokumen perjanjian digital dan profil pengguna yang dapat memperbarui informasi profil seperti nama, nomor telepon, dan alamat secara langsung melalui aplikasi tanpa perlu bantuan admin.

Adapun fungsi utama sistem setelah *reengineering* adalah sebagai berikut.

1. Login: Autentikasi pengguna wajib retribusi.
2. Melihat daftar objek retribusi: Daftar objek ditampilkan dengan opsi pencarian per kecamatan atau nama objek.
3. Melihat tarif objek retribusi: Pengguna dapat melihat tarif berdasarkan kategori dan objek tertentu.
4. Mengajukan permohonan sewa: Pengguna mengisi form permohonan, Dalam sistem Tapatupa, hubungan *<include>* digunakan untuk menggambarkan aktivitas yang selalu menjadi bagian dari proses utama.

Sebagai contoh, pada saat pengguna melakukan pengajuan permohonan penyewaan tanah, sistem secara otomatis menyertakan proses unggah dokumen sebagai tahapan wajib. Dokumen yang diunggah dapat berupa KTP, surat permohonan resmi, atau dokumen pendukung lainnya sesuai persyaratan dari instansi. Oleh karena itu, proses unggah dokumen tidak berdiri sendiri, melainkan terintegrasi sebagai bagian dari permohonan.

5. Melihat status permohonan: Pengguna dapat memantau apakah permohonan sedang diproses, ditolak, atau disetujui.
6. Melihat perjanjian sewa (dokumen digital): Akses dokumen perjanjian dalam format PDF atau digital lainnya.
7. Melihat tagihan: Status tagihan dan *virtual account* ditampilkan secara otomatis setelah permohonan disetujui.
8. Melihat riwayat pembayaran: Riwayat transaksi disimpan dan dapat diakses kembali oleh pengguna.
9. Melakukan pembayaran: Pengguna membayar menggunakan *virtual account* yang diberikan.
10. Mengelola profil pengguna: Data akun dapat diperbarui sesuai kebutuhan.



Gambar 3. 10 Usecase setelah *reengineering*

### 3.2.3. Use Case Scenario

Deskripsi singkat dari setiap *use case* yang terdapat pada use case diagram pada gambar di atas dapat dilihat pada sub bab berikut.

### 3.2.3.1. Use Case Scenario Melakukan Login

Fungsi ini digunakan oleh pengguna untuk melakukan *login*, agar pengguna dapat mengakses aplikasi Tapatupa, *use case scenario* dari *login* dapat dilihat pada Tabel 3.1.

**Tabel 3. 1 usecase scenario Login**

<i>Use Case ID</i>	UC-01	
<i>Use Case Name</i>	Melakukan <i>Login</i>	
<i>Brief Description</i>	<i>Use Case</i> ini menjelaskan bagaimana <i>User</i> dapat melakukan <i>Login</i> untuk dapat mengakses Aplikasi Tapatupa	
<i>Primary Actor</i>	Wajib retribusi	
<i>Pre - condition</i>	<ol style="list-style-type: none"><li>1. Aplikasi Tapatupa telah diinstal pada perangkat android milik penyewa</li><li>2. Penyewa sudah membuka aplikasi <i>mobile</i> Tapatupa</li></ol>	
<i>Post condition</i>	<ol style="list-style-type: none"><li>1. Setelah proses registrasi akun selesai, penyewa dapat menggunakan nama pengguna dan kata sandi untuk proses <i>login</i>.</li><li>2. Penyewa dapat mengakses semua fitur yang ada pada aplikasi Tapatupa.</li></ol>	
<i>Basic Flow Events</i>	<i>Actor's Action</i>	<i>System Response</i>
	<ol style="list-style-type: none"><li>1. Penyewa mengakses Aplikasi Tapatupa yang telah di-<i>instal</i></li></ol>	
		<ol style="list-style-type: none"><li>2. System menampilkan halaman <i>login</i> Aplikasi Tapatupa</li></ol>
	<ol style="list-style-type: none"><li>3. Penyewa mengisi <i>form login</i> dengan <i>email</i> dan <i>password</i> yang telah dimiliki</li></ol>	
		<ol style="list-style-type: none"><li>4. System menerima data <i>login</i> dan</li></ol>



		menampilkan halaman beranda
	5. Penyewa dapat mengakses halaman beranda Aplikasi Tapatupa	
<i>Alternative Flow Of Events</i>	Jika data tidak valid, sistem menampilkan pesan “ <i>Username atau password error</i> ”	

### 3.2.3.2. Use Case Scenario Mengajukan Permohonan

Fungsi ini digunakan oleh penyewa

untuk mengajukan permohonan, agar penyewa dapat melakukan penyewaan aset tanah, *use case scenario* dari mengajukan permohonan dapat dilihat pada Tabel 3.2.

**Tabel 3. 2 Use case Mengajukan Permohonan Sewa**

<i>Use Case ID</i>	UC-02	
<i>Use Case Name</i>	Mengajukan Permohonan Sewa	
<i>Brief Description</i>	Pengguna mengajukan permohonan penyewaan aset tanah melalui aplikasi Tapatupa dengan mengisi data yang dibutuhkan dan mengunggah dokumen persyaratan.	
<i>Primary Actor</i>	Wajib retribusi	
<i>Pre - condition</i>	<ul style="list-style-type: none"> <li>- Pengguna telah memiliki akun dan berhasil login ke aplikasi Tapatupa</li> <li>- Data lahan yang ingin disewa tersedia di dalam sistem.</li> </ul>	
<i>Post condition</i>	Permohonan penyewaan berhasil dibuat.	
<i>Basic Flow Events</i>	<i>Actor's Action</i>	<i>System Response</i>
	1. Penyewa masuk ke aplikasi dan memilih menu "Permohonan Baru".	
		2. System menampilkan

		menu halaman permohonan
	3. Penyewa mengklik tombol “tambah” untuk mengajukan permohonan.	
		4. Sistem menampilkan detail aset dan tombol "Ajukan Permohonan".
	5. Penyewa mengisi formulir permohonan (jenis permohonan, objek retribusi, jenis peruntukan sewa.	
	6. mengunggah dokumen <i>&lt;include&gt;</i> pendukung sesuai persyaratan (KTP, Surat Permohonan, dsb)	
	7. Pengguna menekan tombol "Kirim Permohonan".	
		8. Sistem memvalidasi isian dan dokumen.
<i>Alternative Flow Of Events</i>	Jika dokumen tidak lengkap / tidak diunggah, aplikasi menampilkan pesan <i>error</i> “mohon tambahkan satu dokumen”	

### 3.2.3.3. Use Case Scenario Melakukan Pembayaran

Fungsi ini digunakan oleh penyewa untuk melakukan pembayaran setelah mengajukan permohonan, *use case scenario* dari melakukan pembayaran dapat dilihat pada Tabel 3.3.

**Tabel 3. 3 UseCase Scenario Melakukan Pembayaran**

<i>Use Case ID</i>	UC-03	
<i>Use Case Name</i>	Melakukan Pembayaran	
<i>Brief Description</i>	<i>Use Case</i> ini menjelaskan penyewa melakukan pembayaran biaya sewa aset tanah yang telah disetujui melalui <i>Virtual account</i> yang disediakan oleh sistem	
<i>Primary Actor</i>	Wajib retribusi	
<i>Pre - condition</i>	<ul style="list-style-type: none"> <li>- Permohonan penyewaan sudah disetujui oleh petugas lapangan.</li> <li>- Sistem telah menghasilkan tagihan dan nomor <i>Virtual account</i> untuk pembayaran.</li> </ul>	
<i>Post condition</i>	<ul style="list-style-type: none"> <li>- Pembayaran berhasil diterima oleh sistem.</li> <li>- Status permohonan berubah menjadi “sudah bayar”</li> <li>- Bukti pembayaran tersedia dan notifikasi dikirimkan ke pengguna.</li> </ul>	
<i>Basic Flow Events</i>	<i>Actor Action</i>	<i>System Response</i>
	1. Pengguna login ke aplikasi Tapatupa.	
	2. Penyewa membuka menu “tagihan”	
	3. Penyewa mengklik tagihan sewa	
		4. Sistem menampilkan detail sewa tagihan.
	5. Penyewa mengklik “bayar”	
		6. Sistem menampilkan detail tagihan, jumlah pembayaran, dan nomor <i>Virtual</i>

		<i>account</i>
	7. Penyewa melakukan pembayaran melalui bank atau kanal pembayaran yang tersedia.	
		8. Sistem menerima notifikasi dari sistem pembayaran
		9. sistem memperbarui status menjadi “sudah bayar” dan menampilkan bukti pembayaran
		10. mengirim notifikasi pembayaran sudah berhasil.
<i>Alternative Flow Of Events</i>	-	

#### 3.2.3.4. Use Case Scenario melihat Riwayat Pembayaran

Fungsi ini digunakan oleh penyewa untuk melihat riwayat pembayaran setelah selesai melakukan pembayaran, *use case scenario* dari melihat riwayat pembayaran dapat dilihat pada Tabel 3.4.

**Tabel 3. 4 Usecase Scenario melihat riwayat pembayaran**

<i>Use Case ID</i>	UC-04
<i>Use Case Name</i>	Melihat Riwayat Pembayaran
<i>Brief Description</i>	<i>Use Case</i> ini menjelaskan <i>user</i> melihat histori pembayaran sewa aset tanah yang telah dilakukan.
<i>Primary Actor</i>	Wajib retribusi

<i>Pre - condition</i>	<ul style="list-style-type: none"> <li>- Permohonan penyewaan sudah disetujui oleh petugas lapangan.</li> <li>- Sistem telah menghasilkan tagihan dan nomor <i>Virtual account</i> untuk pembayaran.</li> <li>- Pengguna sudah melakukan pembayaran.</li> </ul>	
<i>Post condition</i>	<ul style="list-style-type: none"> <li>- Pembayaran berhasil diterima oleh sistem.</li> <li>- Status permohonan berubah menjadi “sudah bayar”</li> <li>- Riwayat ditampilkan.</li> </ul>	
<i>Basic Flow Events</i>	<i>Actor Action</i>	<i>System Response</i>
	1. Pengguna login ke aplikasi Tapatupa.	
	2. Penyewa membuka menu “pembayaran”	
	3. Penyewa mengklik pembayaran	
		4. Sistem menampilkan menu “Riwayat Pembayaran”
		5. Sistem menampilkan detail tagihan dan status pembayaran
<i>Alternative Flow Of Events</i>	-	

### 3.2.3.5. Use Case Scenario Melakukan Unggah Dokumen Kelengkapan

Fungsi ini digunakan oleh penyewa untuk melakukan unggah dokumen untuk kelengkapan permohonan aset, *use case scenario* dari melakukan unggah dokumen dapat dilihat pada Tabel 3.5.

**Tabel 3. 5 UseCase Scenario melakukan unggah dokumen**

<i>Use Case ID</i>	UC-05
--------------------	-------

<i>Use Case Name</i>	Melakukan unggah dokumen kelengkapan	
<i>Brief Description</i>	Use Case ini menjelaskan bagaimana <i>User</i> dapat melakukan unggah dokumen untuk kelengkapan saat proses pengajuan Permohonan yang baru	
<i>Primary Actor</i>	Wajib retribusi	
<i>Precondition</i>	Pengguna telah mengisi data permohonan dan siap melanjutkan ke tahap unggah dokumen.	
<i>Postcondition</i>	Dokumen berhasil diunggah dan disimpan di sistem.	
<i>Basic Flow Events</i>	<i>Actor's Action</i>	<i>System Response</i>
	1. Pengguna <i>login</i> ke aplikasi Tapatupa.	
	2. Penyewa mengklik menu “permohonan”	
	3. Penyewa memilih menu “buat permohonan” dan mengisi data permohonan.	
		4. System menampilkan formulir permohonan
	5. Penyewa mengklik tambah dokumen kelengkapan	
		6. Sistem menampilkan formulir dokumen
	7. Penyewa mengisi formulir dokumen sesuai kebutuhan	
	8. Penyewa mengklik “unggah dokumen”	
		9. Sistem

		menampilkan pesan bahwa permohonan telah dikirim
<i>Alternative Flow Of Events</i>	Jika dokumen tidak diunggah, aplikasi menampilkan pesan error “mohon tambahkan satu dokumen”	

### 3.2.3.6. Use Case Scenario Melihat Perjanjian Sewa

Fungsi ini digunakan oleh penyewa untuk mengakses perjanjian sewa yang telah diajukan sebelumnya, *use case scenario* dari melihat perjanjian sewa dapat dilihat pada Tabel 3.6.

**Tabel 3. 6 Usecase Scenario Melihat Perjanjian Sewa**

<i>Use Case ID</i>	UC-06	
<i>Use Case Name</i>	melihat dokumen Perjanjian Sewa	
<i>Brief Description</i>	Use Case ini menjelaskan proses <i>User</i> dapat melihat dokumen digital perjanjian sewa	
<i>Primary Actor</i>	Wajib retribusi	
<i>Precondition</i>	<ul style="list-style-type: none"> <li>- Pengguna sudah membuka aplikasi <i>mobile</i> Tapatupa</li> <li>- Permohonan telah disetujui</li> </ul>	
<i>Post condition</i>	Dokumen ditampilkan	
<i>Basic Flow Events</i>	<i>Actor's Action</i>	<i>System Response</i>
	1. Penyewa membuka aplikasi dan masuk ke halaman utama	
		2. Menampilkan halaman utama dengan berbagai menu
	3. Penyewa memilih menu	

	“perjanjian”	
		4. Menampilkan informasi detail perjanjian
<i>Alternative Flow Of Events</i>	Jika dokumen belum diunggah oleh admin, sistem menampilkan pesan “Dokumen belum tersedia”.	

### 3.2.3.7. Use Case Scenario Melihat Daftar Objek Retribusi

Fungsi ini digunakan oleh penyewa untuk mengakses daftar objek yang disediakan untuk membantu penyewa dalam melihat informasi daftar objek, *use case scenario* dari mengakses daftar objek dapat dilihat pada Tabel 3.7.

**Tabel 3. 7 Usecase Scenario Melihat Daftar Objek Retribusi**

<i>Use Case ID</i>	UC-07	
<i>Use Case Name</i>	Melihat daftar Objek Retribusi	
<i>Brief Description</i>	<i>Use Case</i> ini menjelaskan proses <i>User</i> melihat daftar objek aset tanah milik pemerintah yang tersedia untuk disewa melalui aplikasi Tapatupa. Objek dapat di filter berdasarkan wilayah atau kategori.	
<i>Primary Actor</i>	Wajib retribusi	
<i>Precondition</i>	<ul style="list-style-type: none"> <li>- Pengguna telah berhasil login ke aplikasi.</li> <li>- Data objek telah tersedia dan ditampilkan dari basis data.</li> </ul>	
<i>Postcondition</i>	<ul style="list-style-type: none"> <li>- Pengguna berhasil melihat daftar objek yang tersedia</li> <li>- Sistem dapat menampilkan detail objek saat pengguna memilih salah satu objek.</li> </ul>	
<i>Basic Flow Events</i>	<i>Actor's Action</i>	<i>System Response</i>
	1. Penyewa membuka aplikasi dan masuk ke halaman utama	
		2. Menampilkan halaman utama



		dengan berbagai menu
	3. Penyewa memilih menu “objek retribusi”	
		4. Menampilkan informasi daftar objek tanah yang tersedia untuk disewa.
	5. Penyewa dapat memfilter daftar berdasarkan lokasi	
	6. Penyewa memilih salah satu objek untuk melihat detailnya.	
		7. Menampilkan detail objek, termasuk luas tanah, harga sewa
<i>Alternative Flow Of Events</i>	Jika tidak ada objek yang sesuai, sistem menampilkan pesan “Data tidak ditemukan”.	

### 3.2.3.8. Use Case Scenario Melihat Tarif Objek Retribusi

Fungsi ini digunakan oleh penyewa untuk mengakses tarif objek yang disediakan untuk membantu penyewa dalam melihat informasi tarif atau harga setiap objek, *use case scenario* dari mengakses tarif objek dapat dilihat pada Tabel 3.8.

**Tabel 3. 8 Usecase Scenario Melihat Tarif Objek Retribusi**

<i>Use Case ID</i>	UC-08
<i>Use Case Name</i>	Melihat informasi tarif objek retribusi
<i>Brief Description</i>	<i>Use Case</i> ini menjelaskan proses <i>User</i> melihat tarif objek aset tanah milik pemerintah yang tersedia melalui aplikasi Tapatupa. Tarif Objek dapat dicari melalui fitur pencarian.
<i>Primary Actor</i>	Wajib retribusi

<i>Pre condition</i>	<ul style="list-style-type: none"> <li>- Pengguna telah berhasil login ke aplikasi.</li> <li>- Data tarif objek telah tersedia dan ditampilkan dari basis data.</li> </ul>	
<i>Post condition</i>	<ul style="list-style-type: none"> <li>- Pengguna berhasil melihat tarif objek yang tersedia</li> <li>- Sistem dapat menampilkan detail tarif objek saat pengguna memilih salah satu aset.</li> </ul>	
<i>Basic Flow Events</i>	<i>Actor's Action</i>	<i>System Response</i>
	1. Penyewa membuka aplikasi dan masuk ke halaman utama	
		2. Menampilkan halaman utama dengan berbagai menu
	3. Penyewa memilih menu “tarif objek”	
		4. Menampilkan informasi tarif objek tanah yang tersedia.
	5. Penyewa dapat mencari tarif objek menggunakan fitur pencarian.	
	6. Penyewa mengklik “lihat detail” untuk melihat detailnya.	
		7. Menampilkan detail tarif objek.
<i>Alternative Flow Of Events</i>	Jika tidak ada data tarif objek yang sesuai, sistem menampilkan pesan “Data tidak ditemukan”.	

### 3.2.3.9. Use Case Scenario Melihat Status Permohonan

Fungsi ini digunakan oleh penyewa untuk melihat status permohonan yang disediakan untuk membantu penyewa dalam melihat dan memantau informasi status permohonan sewa yang dilakukan oleh penyewa, *use case scenario* dari melihat status permohonan dapat dilihat pada Tabel 3.9.

**Tabel 3. 9 Usecase Scenario Melihat Status Permohonan**

<i>Use Case ID</i>	UC-09	
<i>Use Case Name</i>	Melihat status Permohonan	
<i>Brief Description</i>	<i>Use Case</i> ini menjelaskan proses penyewa untuk melihat status permohonan	
<i>Primary Actor</i>	Wajib retribusi	
<i>Pre condition</i>	<ul style="list-style-type: none"><li>- Pengguna telah berhasil login ke aplikasi.</li><li>- Permohonan telah dikirim.</li></ul>	
<i>Post condition</i>	Status ditampilkan	
<i>Basic Flow Events</i>	<i>Actor's Action</i>	<i>System Response</i>
	1. Penyewa membuka aplikasi dan masuk ke halaman utama	
		2. Menampilkan halaman utama dengan berbagai menu
	3. Penyewa memilih menu "Riwayat permohonan"	
		4. Menampilkan daftar status
<i>Alternative Flow Of Events</i>	Jika permohonan belum tersedia, sistem menampilkan pesan "Tidak ada permohonan tersedia".	

### 3.2.3.10. Use Case Scenario Melihat Tagihan Sewa

Fungsi ini digunakan oleh penyewa untuk melihat tagihan sewa yang disediakan untuk membantu penyewa dalam melihat tagihan yang dikenai kepada penyewa, *use case scenario* dari mengakses tagihan sewa dapat dilihat pada Tabel 3.10.

**Tabel 3. 10 Usecase Scenario Melihat Tagihan Sewa**

<i>Use Case ID</i>	UC-10	
<i>Use Case Name</i>	Melihat tagihan sewa	
<i>Brief Description</i>	<i>Use Case</i> ini menjelaskan proses <i>user</i> mengakses informasi tagihan retribusi sewa berdasarkan perjanjian yang telah disahkan, termasuk nominal pembayaran, tanggal jatuh tempo, dan status pembayaran untuk masing-masing periode.	
<i>Primary Actor</i>	Wajib retribusi	
<i>Pre condition</i>	<ul style="list-style-type: none"><li>- Pengguna telah melakukan permohonan sewa yang disetujui.</li><li>- Perjanjian sewa telah disahkan dan data tagihan telah dibuat.</li><li>- Pengguna sudah <i>login</i> ke aplikasi.</li></ul>	
<i>Post condition</i>	<ul style="list-style-type: none"><li>- Pengguna dapat melihat daftar tagihan sewa berdasarkan perjanjian</li></ul>	
<i>Basic Flow Events</i>	<i>Actor's Action</i>	<i>System Response</i>
	1. Penyewa membuka aplikasi dan masuk ke halaman utama	
	2. Penyewa memilih menu “tagihan”	
	3. Penyewa memilih menu “Pembayaran”	
		5. Menampilkan informasi pembayaran seperti total pembayaran, nomor <i>virtual account</i> , dan

		petunjuk pembayaran
<i>Alternative Flow Of Events</i>	-	-

### 3.2.3.11. Use Case Scenario Mengedit Profile pengguna

Fungsi ini digunakan oleh penyewa untuk mengedit profil pengguna yang memungkinkan penyewa untuk memperbarui data akun yang tersimpan di sistem aplikasi Tapatupa, *use case scenario* dari mengedit profil dapat dilihat pada Tabel 3.11.

**Tabel 3. 11 UseCase Scenario mengedit profile**

<i>Use Case ID</i>	UC-10	
<i>Use Case Name</i>	Mengedit Profil Pengguna	
<i>Brief Description</i>	<i>Use Case</i> ini menjelaskan proses <i>user</i> dapat memperbarui informasi akun mereka seperti nomor telepon, dan alamat melalui halaman profil di aplikasi Tapatupa.	
<i>Primary Actor</i>	Wajib retribusi	
<i>Pre condition</i>	<ul style="list-style-type: none"> <li>- Pengguna telah berhasil <i>login</i> ke aplikasi.</li> <li>- Data profil pengguna telah tersimpan dalam basis data.</li> </ul>	
<i>Post condition</i>	Perubahan data profil pengguna tersimpan dan diperbarui di sistem.	
<i>Basic Flow Events</i>	<i>Actor Action</i>	<i>System Response</i>
	1. Penyewa membuka aplikasi dan masuk ke halaman utama	
	2. Penyewa masuk menu “profil”	

	3. Penyewa memilih tombol “edit”	
		4. Menampilkan formulir berisi data saat ini (nomor telepon, alamat)
	5. Penyewa melakukan perubahan pada satu atau beberapa data.	
	6. Penyewa menekan tombol “simpan”	
<i>Alternative Flow Of Events</i>	-	-

#### 3.2.4. User Characteristic

Pengguna utama dari aplikasi Tapatupa berbasis *mobile* adalah masyarakat wajib retribusi, yaitu individu atau kelompok yang melakukan penyewaan tanah milik Pemerintah Daerah Kabupaten Tapanuli Utara. Karakteristik pengguna ini sangat memengaruhi desain antarmuka, alur sistem, serta kebutuhan fungsional yang dikembangkan pada aplikasi.

Secara umum, pengguna berasal dari berbagai latar belakang pendidikan dan pekerjaan, dengan tingkat pemahaman teknologi yang bervariasi yang umumnya digunakan oleh masyarakat di wilayah kabupaten.

Pengguna memiliki kebutuhan utama dalam mengajukan permohonan sewa tanah, melihat informasi tarif dan objek retribusi, serta melakukan pembayaran. Selain itu, mereka juga membutuhkan informasi yang jelas terkait status permohonan, akses ke dokumen perjanjian, dan riwayat tagihan yang akan atau sudah pernah dilakukan. Dalam rancangan aplikasi Tapatupa, karakteristik *user* dapat dilihat dari Tabel 3.12.

**Tabel 3. 12 User Characteristics**

User Group/Role	Kepentingan Akses	Hak Akses
Pengguna (masyarakat wajib retribusi)	<ol style="list-style-type: none"> <li>1. Melakukan <i>login</i></li> <li>2. Mengakses informasi objek retribusi</li> <li>3. Mengakses tarif objek retribusi</li> <li>4. Mengakses perjanjian</li> <li>5. Mengakses tagihan</li> <li>6. Melakukan permohonan sewa</li> <li>7. Melakukan unggah dokumen</li> </ol> <p>Melakukan pembayaran</p>	Melihat informasi objek retribusi, informasi tarif objek retribusi, informasi perjanjian, informasi tagihan, melakukan pembayaran, mengajukan permohonan sewa, mengunggah dokumen, dan melakukan <i>login</i> di aplikasi.

### 3.2.5. Sequence Diagram

Sequence Diagram merupakan diagram yang menunjukkan gambaran skenario atau rangkaian langkah - langkah yang dilakukan dari sebuah kejadian untuk menghasilkan keluaran gambaran skenario sebagai respons terhadap suatu kejadian yang menghasilkan suatu tertentu. Pada bagian ini akan disertakan *diagram sequence* dari masing masing fungsi yang ada pada aplikasi *mobile* Tapatupa.

#### 3.2.5.1. Sequence diagram Login

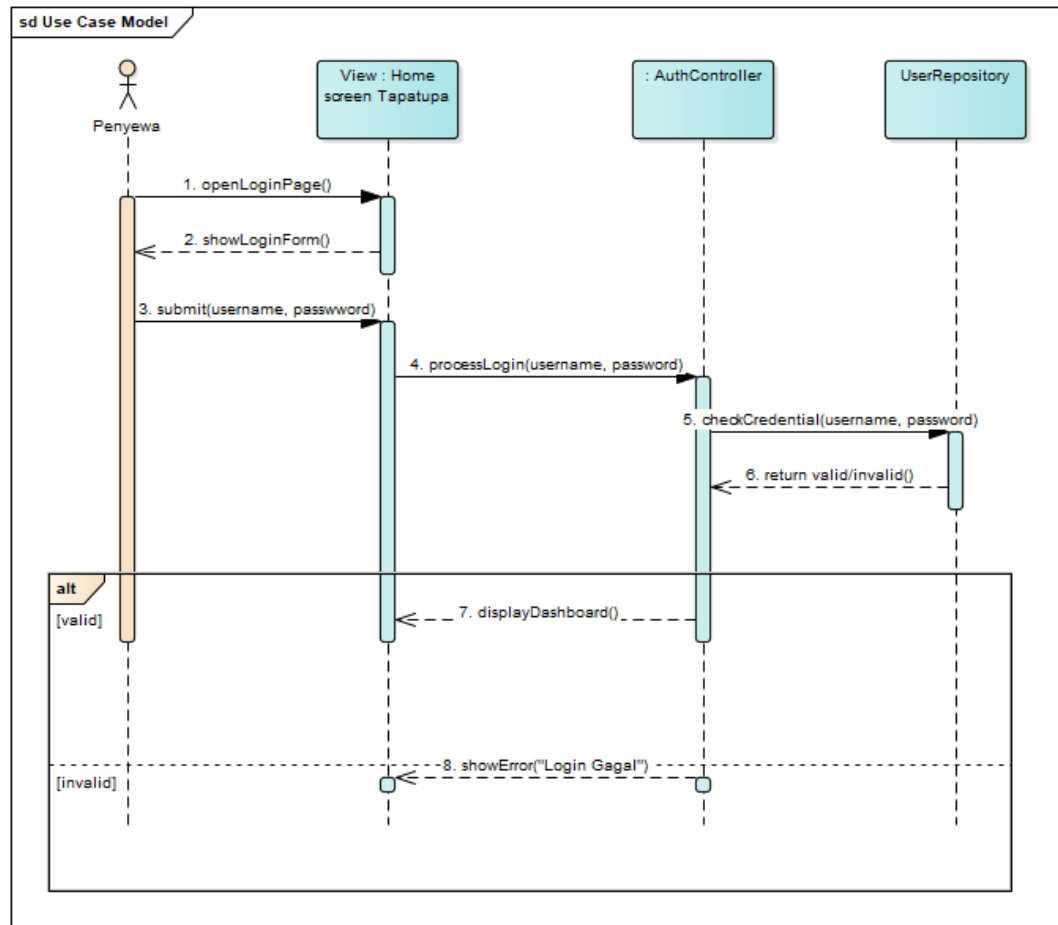
*Sequence* diagram dibawah ini menggambarkan proses autentikasi pengguna (penyewa) saat melakukan login ke dalam aplikasi Tapatupa. Diagram ini menunjukkan alur interaksi antara aktor Penyewa dengan empat entitas utama sistem, yaitu *view home screen* Tapatupa, *auth controller*, dan *user repositori*. *Sequence* ini menggambarkan seluruh proses mulai dari tampilan halaman *login*, validasi kredensial, hingga pemberian umpan balik kepada pengguna berdasarkan status autentikasi. Alur diawali dengan penyewa membuka halaman *login* melalui fungsi *open login page* yang kemudian menghasilkan tampilan *form login* dengan

metode *show login form*. Setelah *form* ditampilkan, penyewa mengisi *form* dengan *username* dan *password*, lalu mengirimkan data tersebut melalui fungsi submit (*username, password*). Data yang dikirimkan akan diproses oleh *auth controller* melalui metode *process login (username, password)*.

Selanjutnya, *auth controller* meneruskan proses verifikasi data ke *user* repositori dengan memanggil fungsi *check credential (username, password)*. Repositori akan memvalidasi kecocokan *username* dan *password* yang dimasukkan dengan data yang ada di basis data. Setelah proses verifikasi selesai, *user* repositori akan mengembalikan status hasil autentikasi berupa valid atau invalid kepada *auth controller*. Diagram ini menggunakan blok alt (alternatif) untuk mendeskripsikan dua kondisi yang dapat terjadi berdasarkan hasil autentikasi. Pada kondisi pertama, jika hasil validasi bernilai valid, sistem akan mengarahkan pengguna menuju *dashboard* utama aplikasi melalui fungsi *display dashboard*. Sebaliknya, apabila hasil validasi bernilai invalid, maka sistem akan menampilkan pesan kesalahan kepada pengguna berupa notifikasi "login gagal" melalui fungsi *show error*.

Penggunaan blok alt dalam diagram ini mencerminkan adanya percabangan logika yang sesuai dengan implementasi proses login. Diagram ini juga menunjukkan penggunaan *controller* sebagai penghubung utama antara antarmuka pengguna dan data repositori, mendukung arsitektur berbasis *model view controller* (MVC) yang meningkatkan modularitas dan keterpisahan tanggung jawab antar komponen sistem. *Sequence* diagram login ini menunjukkan bahwa aplikasi Tapatupa telah dirancang dengan alur yang sesuai. Adanya dua jalur eksekusi, yaitu valid dan invalid, menunjukkan implementasi kontrol alur yang baik dapat memberikan sistem yang menangani *input* yang valid maupun tidak valid secara aman.





Gambar 3. 11 Sequence diagram login

### 3.2.5.2. Sequence diagram Permohonan

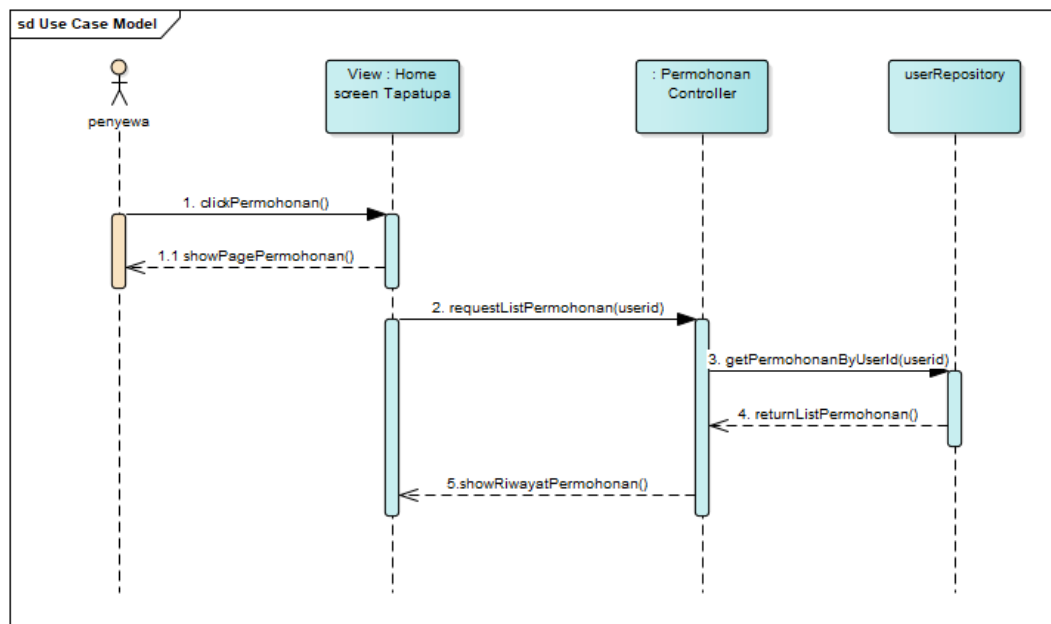
*Sequence* diagram dibagi menjadi 2 alur yaitu tampilan halaman riwayat permohonan dan tampilan halaman buat permohonan. *Sequence* diagram ini menjelaskan alur proses permohonan sewa tanah melalui aplikasi *mobile* Tapatupa.

#### 3.2.5.2.1. Sequence diagram Riwayat Permohonan

Diagram ini menjelaskan proses pengguna dalam melihat daftar atau histori permohonan yang telah dilakukan sebelumnya. Proses diawali dengan aksi click permohonan oleh penyewa, yang akan menampilkan halaman permohonan melalui metode show page permohonan. Setelah tampilan halaman muncul, sistem melakukan permintaan daftar permohonan dengan mengirim request list permohonan menggunakan user id dari view ke controller. Controller kemudian mengakses data dari repositori dengan memanggil metode get permohonan menggunakan user id. Repositori akan mengembalikan daftar permohonan milik

pengguna dalam bentuk return list permohonan. Data ditampilkan ke pengguna melalui aksi show riwayat permohonan.

Diagram ini menggambarkan proses alur riwayat permohonan oleh pengguna, dengan alur yang sederhana namun mudah dimengerti. Interaksi hanya terdiri dari pengambilan data satu arah berdasarkan user id, tanpa input atau lainnya. Proses ini menegaskan bahwa sistem menyimpan data permohonan pengguna secara terstruktur, hal ini penting sebagai bentuk transparansi layanan dan dokumentasi digital atas aktivitas penyewaan yang telah dilakukan.



Gambar 3. 12 Sequence Diagram Riwayat Permohonan

#### 3.2.5.2.2. Sequence diagram Buat permohonan

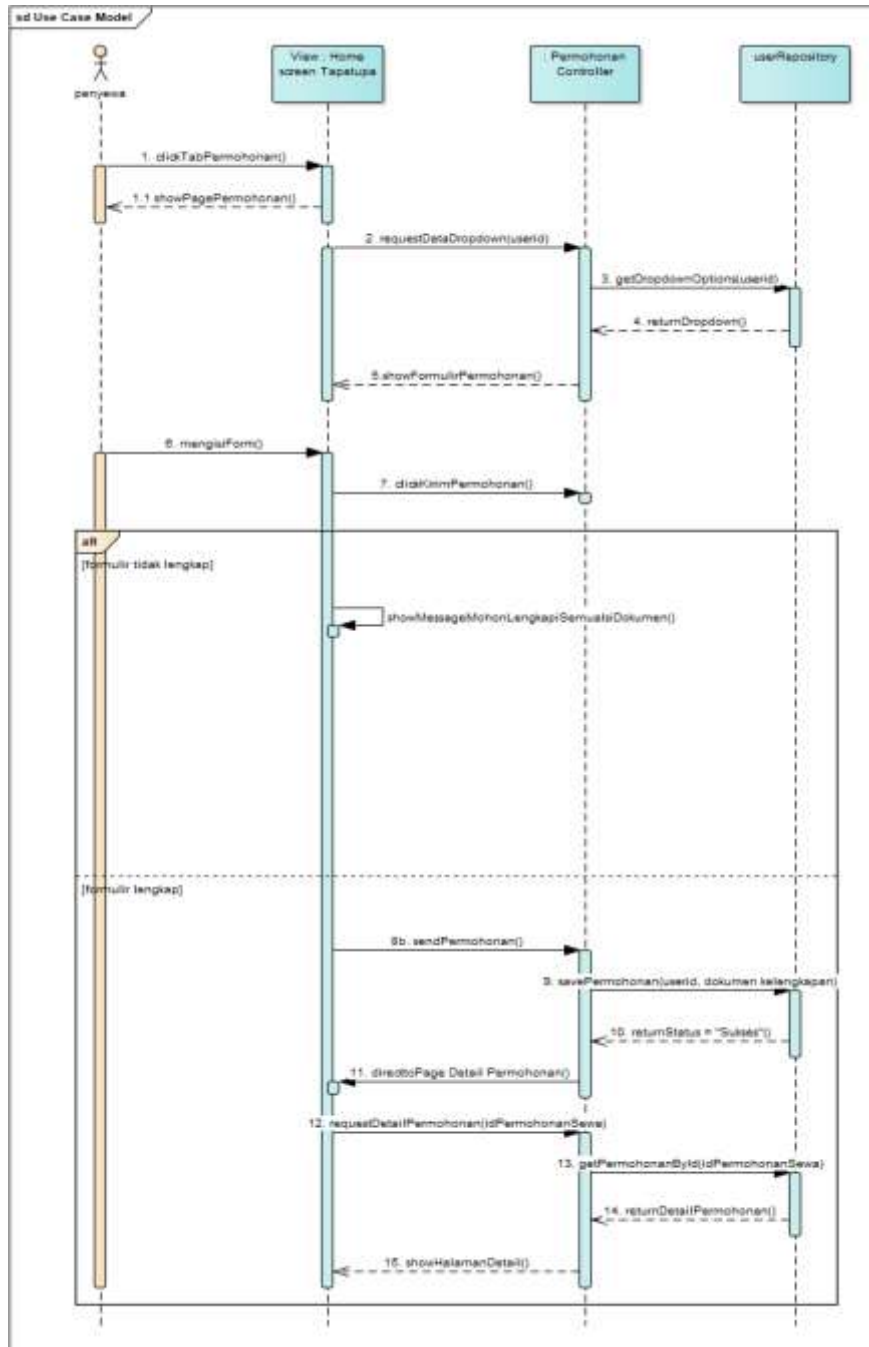
Diagram urutan dibawah ini menggambarkan proses interaktif antara pengguna (penyewa) dan sistem dalam melakukan pembuatan permohonan penyewaan melalui aplikasi Tapatupa.

Proses diawali ketika penyewa mengakses menu permohonan dengan aksi mengklik menu permohonan, yang akan menghasilkan tampilan awal halaman permohonan melalui metode show page permohonan oleh objek view. Setelah halaman ditampilkan, sistem segera mengirim permintaan data dropdown dengan parameter userid melalui request data dropdown menggunakan user id ke controller.

Controller lalu meneruskan permintaan tersebut ke objek user repositori menggunakan `get drop down options`. Data yang diminta, seperti jenis dokumen, pilihan objek, atau data lainnya, dikembalikan melalui `return dropdown`, yang kemudian ditampilkan dalam bentuk formulir permohonan pada tampilan pengguna melalui `show` formulir permohonan.

Penyewa kemudian mengisi formulir pada antarmuka melalui aksi mengisi *form*, lalu menekan tombol klik kirim permohonan untuk mengirim data permohonan ke sistem. Pada tahap ini terdapat dua kondisi alternatif yang ditangani oleh blok `alt`. Jika kondisi formulir tidak lengkap, maka sistem akan menampilkan peringatan menggunakan `show message` “mohon lengkapi semua isian dokumen” dan proses akan berhenti untuk sementara. Sebaliknya, jika formulir lengkap, maka sistem melanjutkan dengan `send permohonan`. Permohonan dikirim ke controller yang memproses dan menyimpan permohonan melalui `save permohonan` dengan menggunakan parameter `userid`, dokumen kelengkapan. Data kemudian diteruskan ke repositori, dan setelah tersimpan, sistem mengembalikan status keberhasilan dengan status “success”. Setelah permohonan berhasil dikirim, penyewa secara otomatis diarahkan menuju halaman detail permohonan melalui `direct to page detail permohonan`. Tampilan kemudian mengirim permintaan data detail dengan `request detail permohonan`, controller mengambil data dengan `get permohonan by id` ke repositori, yang kemudian dikembalikan ke view dalam bentuk `return detail permohonan`. Proses diakhiri dengan sistem menampilkan halaman detail permohonan kepada pengguna melalui `show halaman detail`.

*Sequence* diagram ini secara lengkap menunjukkan proses pembuatan permohonan sewa yang dilakukan oleh penyewa, termasuk penanganan validasi isian formulir secara kondisional (formulir lengkap/tidak lengkap) yang menunjukkan sistem dirancang untuk memastikan bahwa pengguna tidak dapat melanjutkan permohonan tanpa data yang valid.



Gambar 3. 13 Sequence Diagram Buat Permohonan

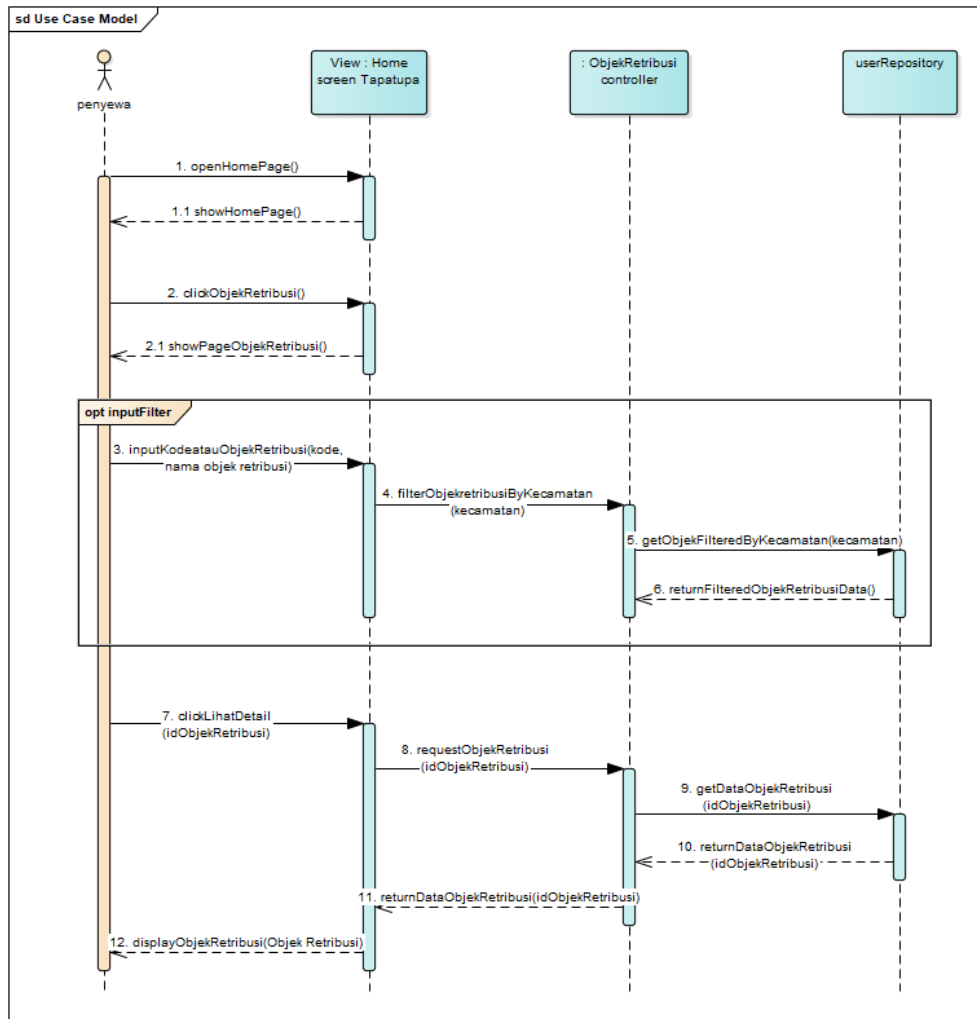
### 3.2.5.3. Sequence diagram Objek Retribusi

Diagram urutan (*sequence* diagram) yang ditampilkan menggambarkan proses ketika penyewa ingin melihat dan memfilter daftar objek retribusi (merujuk pada daftar objek sewa yang dikenakan retribusi). Proses ini diawali oleh aktor penyewa

untuk melakukan aksi open home page, yaitu membuka halaman utama aplikasi Tapatupa. Permintaan ini diterima oleh objek view yaitu home screen Tapatupa, yang kemudian mengembalikan tampilan awal melalui show home page. Selanjutnya, penyewa menekan tombol menu objek retribusi dengan menjalankan fungsi klik objek retribusi, yang kembali menampilkan halaman daftar objek retribusi menggunakan metode show page objek retribusi. Penyewa diberikan opsi untuk melakukan filter pencarian objek, misalnya berdasarkan kode objek, nama objek, atau lokasi kecamatan.

Jika penyewa mengisi input filter, maka aktivitas input kode atau objek retribusi seperti kode / nama objek dijalankan. Permintaan ini kemudian dikirim oleh tampilan ke controller objek retribusi controller melalui metode filter objek retribusi by kecamatan. Controller selanjutnya meneruskan permintaan tersebut ke repositori data melalui get objek filtered by kecamatan. Repositori akan mencari dan mengembalikan data yang sesuai dengan filter kecamatan menggunakan return filtered objek retribusi data, lalu data dikirimkan kembali oleh controller ke tampilan halaman objek. Setelah data objek ditampilkan, penyewa dapat memilih salah satu objek dengan melakukan klik lihat detail menggunakan id objek retribusi. Hal ini akan memicu permintaan ke controller melalui request objek retribusi menggunakan id objek retribusi, yang akan diteruskan ke repositori data dengan perintah get data objek retribusi. Repositori akan memproses dan mengembalikan data lengkap objek retribusi melalui return data dan dikirim kembali menjadi tampilan melalui controller, dan akhirnya ditampilkan kepada penyewa.

Secara keseluruhan, *sequence* ini merepresentasikan sistem yang terstruktur untuk mengelola akses terhadap data objek retribusi. Proses ini sangat penting dalam konteks pelayanan publik digital seperti penyewaan aset daerah, di mana transparansi data dan kemudahan akses menjadi prioritas utama bagi pengguna masyarakat.



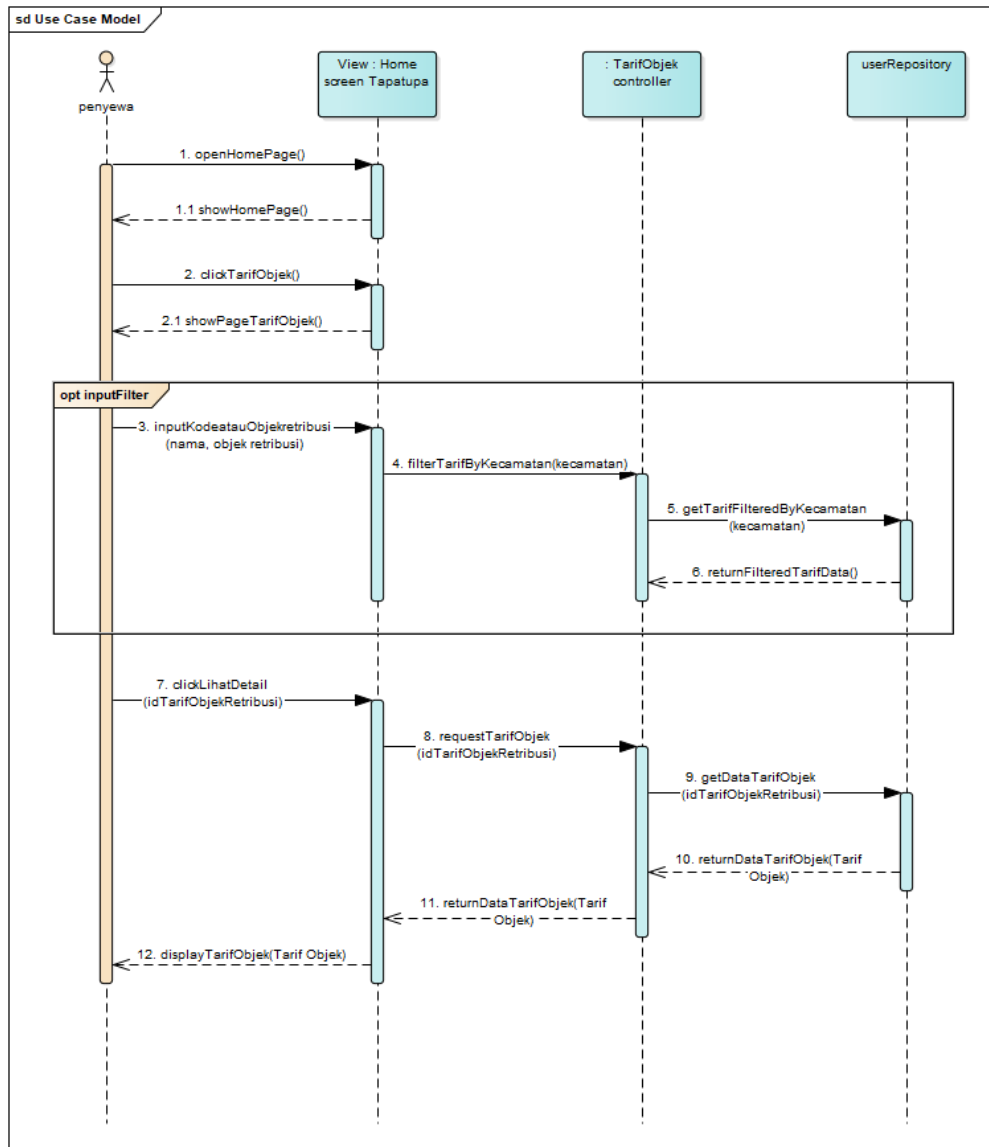
Gambar 3. 14 Sequence Diagram Objek Retribusi

#### 3.2.5.4. Sequence diagram Tarif Objek Retribusi

*Sequence* diagram ini menggambarkan alur sistem ketika penyewa ingin melihat tarif dari objek retribusi (sewa) yang tersedia dalam aplikasi Tapatupa. Proses dimulai saat penyewa membuka halaman utama aplikasi melalui aksi *open home page*, yang kemudian dijalankan oleh tampilan antarmuka *view home screen* Tapatupa melalui metode *show home page*. Setelah halaman utama ditampilkan, penyewa memilih menu “Tarif Objek” dengan menjalankan aksi klik tarif objek, yang menyebabkan sistem menampilkan halaman daftar tarif objek melalui metode *show page* tarif objek. Pada tahap ini, memanfaatkan fitur filter berdasarkan kecamatan atau nama objek retribusi guna melakukan pencarian tarif. Bagian filter ditunjukkan dalam blok *opt input filter*, yang menyatakan bahwa proses filter ini

bersifat opsional. Jika penyewa memilih untuk menggunakan filter, maka mereka akan mengisi input seperti nama objek retribusi atau kecamatan melalui input kode seperti nama, objek retribusi dan akan mengirimkan permintaan filter ke tarif objek controller menggunakan filter tarif by kecamatan. Controller kemudian meneruskan permintaan tersebut ke user repositori dengan metode get tarif filtered kecamatan. Setelah data tarif sesuai dengan filter ditemukan, repositori akan mengembalikan ke controller melalui return filtered tarif data, yang kemudian diteruskan ke tampilan antarmuka. Setelah data hasil filter (jika digunakan) atau daftar lengkap objek ditampilkan, penyewa dapat memilih salah satu objek untuk melihat rincian tarif dengan mengklik lihat detail. Tampilan kemudian mengirim permintaan detail ke controller dengan metode get tarif objek menggunakan id objek retribusi. Setelah informasi detail berhasil ditemukan, repositori mengembalikan data tersebut ke controller melalui return data tarif objek (Tarif objek), yang kemudian dikirim kembali ke tampilan antarmuka menggunakan return data.

*Sequence* diagram ini memperlihatkan bagaimana sistem Tapatupa memfasilitasi pengguna (penyewa) dalam mencari dan melihat informasi tarif dari objek retribusi sewa. Diagram ini mencakup dua alur utama, pertama, alur opsional untuk melakukan filter data tarif berdasarkan kecamatan atau nama objek, dan kedua, alur utama untuk melihat detail dari objek retribusi yang dipilih. Blok opt (opsional) digunakan untuk menandakan bahwa proses filter bukan langkah wajib, namun opsional tergantung kebutuhan pengguna. Interaksi ini melibatkan empat komponen sistem, yaitu aktor penyewa, antarmuka pengguna, Tarif objek controller, dan user repositori, yang saling bekerja sama untuk menampilkan informasi tarif secara akurat. Diagram ini sangat penting untuk menggambarkan bagaimana pengguna dapat menavigasi informasi tarif yang spesifik serta menunjukkan struktur logika dan alur data yang baik dalam mendukung transparansi biaya sewa kepada pengguna.



**Gambar 3. 15 Sequence Diagram Tarif Objek Retribusi**

### 3.2.5.5. Sequence diagram Perjanjian Sewa

*Sequence* diagram ini menjelaskan alur proses yang terjadi saat seorang penyewa mengakses dan melihat dokumen perjanjian sewa melalui aplikasi Tapatupa. Proses diawali ketika penyewa membuka halaman utama aplikasi dengan menjalankan aksi *open home page*, yang kemudian memicu tampilan antarmuka *view home screen* Tapatupa untuk halaman utama menggunakan metode *show home page*. Setelah berada di halaman utama, penyewa melanjutkan dengan menekan tombol menu “perjanjian” untuk mengakses daftar perjanjian sewa yang tersedia. Aksi ini

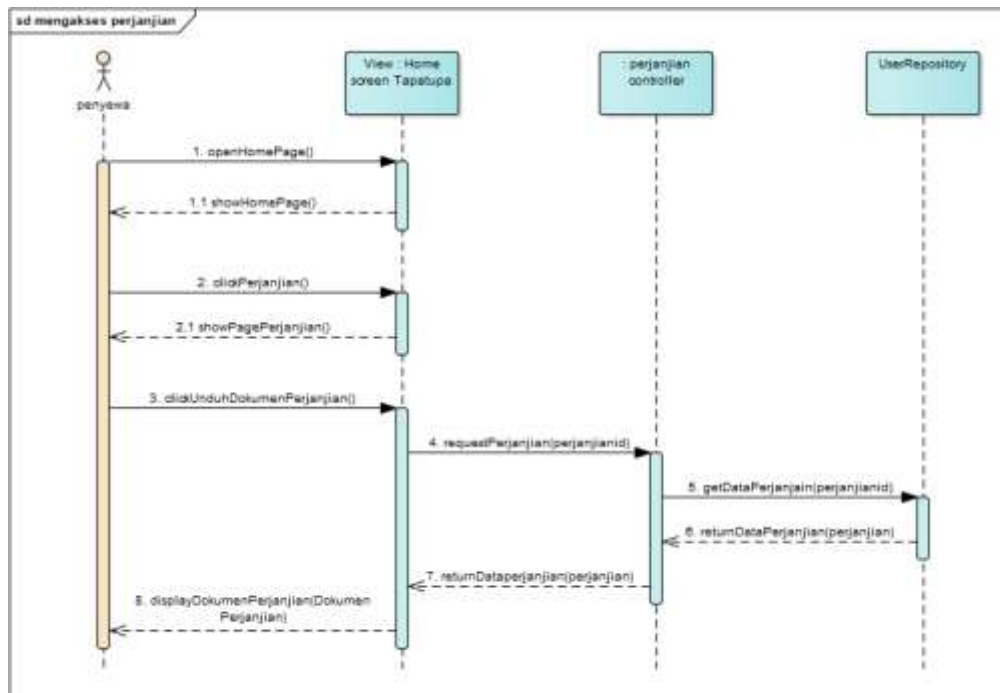


ditangani oleh fungsi klik perjanjian, yang kemudian menyebabkan antarmuka menampilkan halaman perjanjian melalui metode `show page perjanjian`.

Pada halaman perjanjian, penyewa melakukan aksi kembali berupa klik pada tombol. Unduh dokumen Perjanjian terlebih dahulu ditandatangani dengan cara mengklik unduh dokumen perjanjian. Setelah itu, akan dikirim data ke controller dengan menggunakan metode *request* unduh dokumen.

Controller kemudian meneruskan permintaan tersebut ke user repositori dengan memanggil `get data perjanjian` menggunakan `perjanjian id` untuk mengambil isi atau detail dokumen perjanjian berdasarkan id tersebut. Setelah data berhasil ditemukan, user repositori mengembalikannya ke controller melalui `return data perjanjian` dan controller meneruskan data kembali ke tampilan antarmuka dengan `return data perjanjian`. Langkah terakhir dalam sequence ini adalah sistem menampilkan isi dokumen perjanjian kepada penyewa dalam bentuk dokumen yang dapat dibaca atau diunduh, menggunakan metode `display dokumen perjanjian`. Dokumen yang ditampilkan bisa berupa PDF atau tampilan teks yang mencerminkan perjanjian resmi.

*Sequence* diagram ini menggambarkan proses penampilan dokumen perjanjian sewa secara sistematis di dalam sistem Tapatupa. Diagram ini memperlihatkan pengguna berinteraksi dengan antarmuka untuk memicu proses pengambilan data perjanjian dari repositori melalui controller. Proses ini menggambarkan hubungan tiga arsitektur utama dalam sistem, yaitu antarmuka pengguna, logika bisnis (controller), dan penyimpanan data (repositori). Diagram ini tidak memiliki kondisi percabangan eksplisit (seperti `alt`), karena semua alurnya bersifat langsung menuju pada pengambilan dan penampilan dokumen. Hal ini menunjukkan bahwa proses pengambilan dokumen perjanjian merupakan proses sederhana namun penting untuk memastikan transparansi informasi antara sistem dan penyewa. Dengan demikian, diagram ini sangat penting sebagai dokumentasi fungsional yang menunjukkan bagaimana sistem memastikan pengguna dapat mengakses kontrak atau perjanjian sewanya secara cepat.



**Gambar 3. 16 Sequence Diagram Perjanjian Sewa**

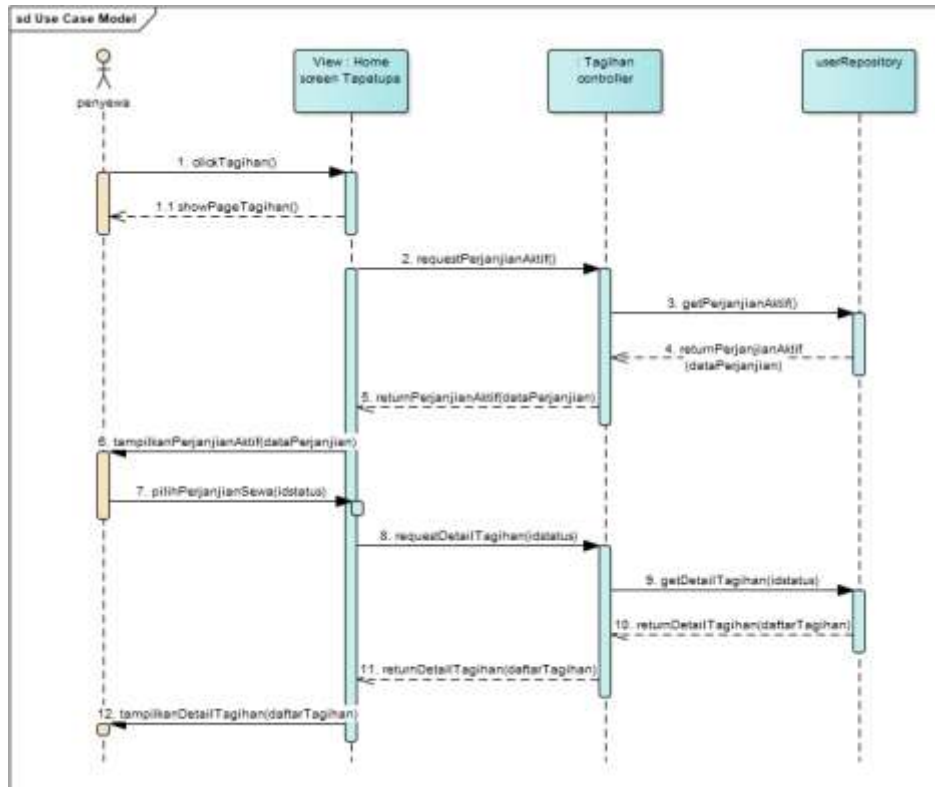
#### 3.2.5.6. Sequence diagram Tagihan sewa

Sequence diagram ini menggambarkan alur proses ketika penyewa ingin melihat tagihan sewa yang masih aktif melalui aplikasi Tapatupa. Proses dimulai ketika penyewa mengklik menu “tagihan” dari tampilan utama, yang ditangani oleh tampilan *view home screen* Tapatupa dengan menjalankan fungsi klik tagihan. Setelah itu, tampilan memuat halaman informasi tagihan dengan menjalankan metode show page tagihan. Setelah halaman tagihan ditampilkan, dilanjutkan dengan mengirim permintaan ke controller untuk mengambil data perjanjian sewa yang masih aktif dengan memanggil request perjanjian aktif. Tagihan controller kemudian melanjutkan proses dengan memanggil get perjanjian aktif ke use repositori untuk mengambil data-data perjanjian sewa yang aktif dari penyimpanan data. User repositori kemudian mengembalikan data perjanjian aktif (dalam bentuk objek data perjanjian) kepada tagihan controller yang diteruskan kembali pada tampilan melalui return perjanjian aktif.

Setelah data perjanjian aktif diterima, tampilan akan menampilkan data tersebut kepada pengguna melalui proses tampilan perjanjian aktif (*dataPerjanjian*).

Selanjutnya, penyewa dapat memilih salah satu perjanjian sewa berdasarkan status dengan menjalankan aksi pilih perjanjian sewa menggunakan id status. Setelah perjanjian dipilih, tampilan mengirimkan permintaan ke tagihan controller untuk mengambil detail tagihan berdasarkan id status dari perjanjian yang dipilih melalui pesan request detail tagihan menggunakan id status. Controller kemudian meneruskan permintaan ke user repositori dengan metode get detail tagihan untuk mendapatkan daftar tagihan yang terkait. Setelah data diambil, repositori akan mengembalikan dalam bentuk daftar tagihan ke controller dan mengirim kembali melalui return detail tagihan menggunakan daftar tagihan yang diakhiri dengan menyajikan informasi detail tagihan kepada pengguna dengan metode tampilkan detail tagihan.

Sequence diagram ini memperlihatkan bagaimana sistem menangani proses pengecekan tagihan oleh penyewa secara menyeluruh dan terstruktur. Proses ini terbagi menjadi dua tahap utama, pertama, sistem mengambil dan menampilkan daftar perjanjian sewa aktif, dan kedua, sistem mengambil serta menampilkan rincian tagihan berdasarkan perjanjian yang dipilih oleh pengguna. Diagram ini menegaskan pentingnya komunikasi antar komponen, yakni antara antarmuka pengguna, controller logika bisnis, dan repositori penyimpanan data, dalam menyediakan informasi tagihan yang akurat dan sesuai. Proses ini tidak memiliki percabangan eksplisit (seperti alt) namun menunjukkan alur yang terdiri dari dua permintaan data terpisah yang saling berurutan. Dengan demikian, diagram ini membantu alur logis untuk fitur pengecekan tagihan dalam sistem Tapatupa, yang sangat penting bagi pengguna dalam mengelola kewajiban pembayaran sewanya.



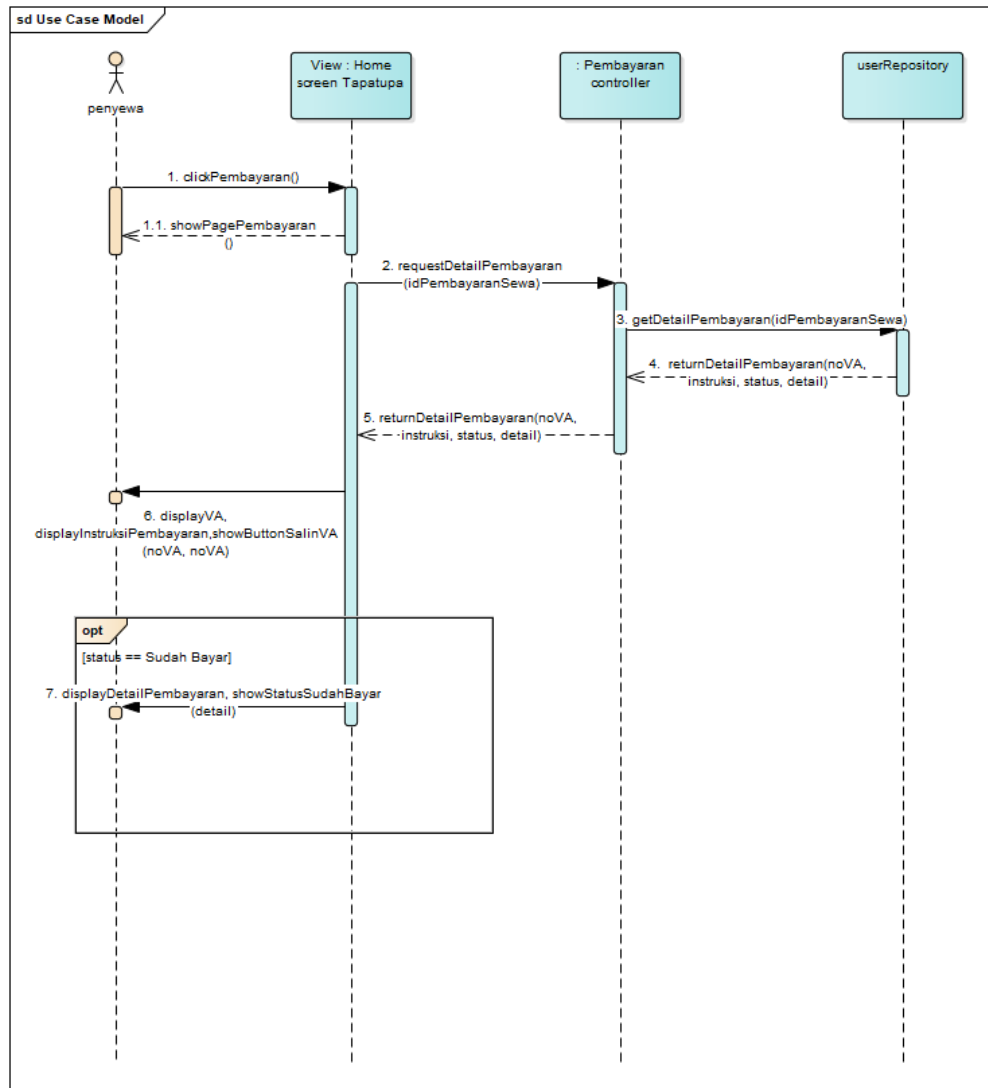
Gambar 3. 17 Sequence Diagram Tagihan Sewa

### 3.2.5.7. Sequence diagram Pembayaran

Sequence diagram ini menjelaskan alur interaksi yang terjadi ketika penyewa ingin memeriksa informasi pembayaran sewa melalui aplikasi Tapatupa. Proses dimulai ketika penyewa mengklik tombol “pembayaran” pada halaman utama aplikasi, melalui *view home screen* Tapatupa dengan pemanggilan fungsi klik pembayaran. Tampilan kemudian memuat halaman detail pembayaran menggunakan metode *show page* pembayaran. Setelah halaman pembayaran ditampilkan, tampilan mengirimkan permintaan detail pembayaran kepada pembayaran controller melalui pesan request detail pembayaran menggunakan id pembayaran sewa yang digunakan untuk mengidentifikasi transaksi pembayaran tertentu. Pembayaran controller meneruskan permintaan ke user repositori dengan menggunakan pemanggilan metode get detail pembayaran untuk mengambil data pembayaran dari penyimpanan data.

Selanjutnya, user repositori menerima permintaan tersebut dengan mengembalikan data pembayaran, yang mencakup nomor *virtual account*, instruksi pembayaran,

status pembayaran, dan detail transaksi melalui pesan *return* detail pembayaran seperti nomor *virtual account*, instruksi, status, dan detail. Data ini diteruskan kembali oleh pembayaran controller ke tampilan, yang kemudian diproses untuk menampilkan informasi kepada penyewa. Diagram menunjukkan dua kondisi berbeda melalui blok opt, yang menggambarkan alur berdasarkan status pembayaran “sudah bayar” akan menampilkan detail pembayaran melalui metode *display detail* pembayaran menunjukkan status "sudah bayar" menggunakan metode show status sudah bayar. Detail ini memastikan bahwa pengguna mendapatkan informasi yang sesuai berdasarkan status pembayaran mereka. *Sequence* diagram ini menggambarkan alur kerja sistem secara rinci untuk memastikan bahwa penyewa mendapatkan informasi pembayaran yang sesuai dengan status transaksi mereka. Diagram ini menekankan pentingnya pengambilan data yang akurat dari repositori serta penyampaian informasi yang jelas kepada pengguna.



Gambar 3. 18 Sequence Diagram pembayaran

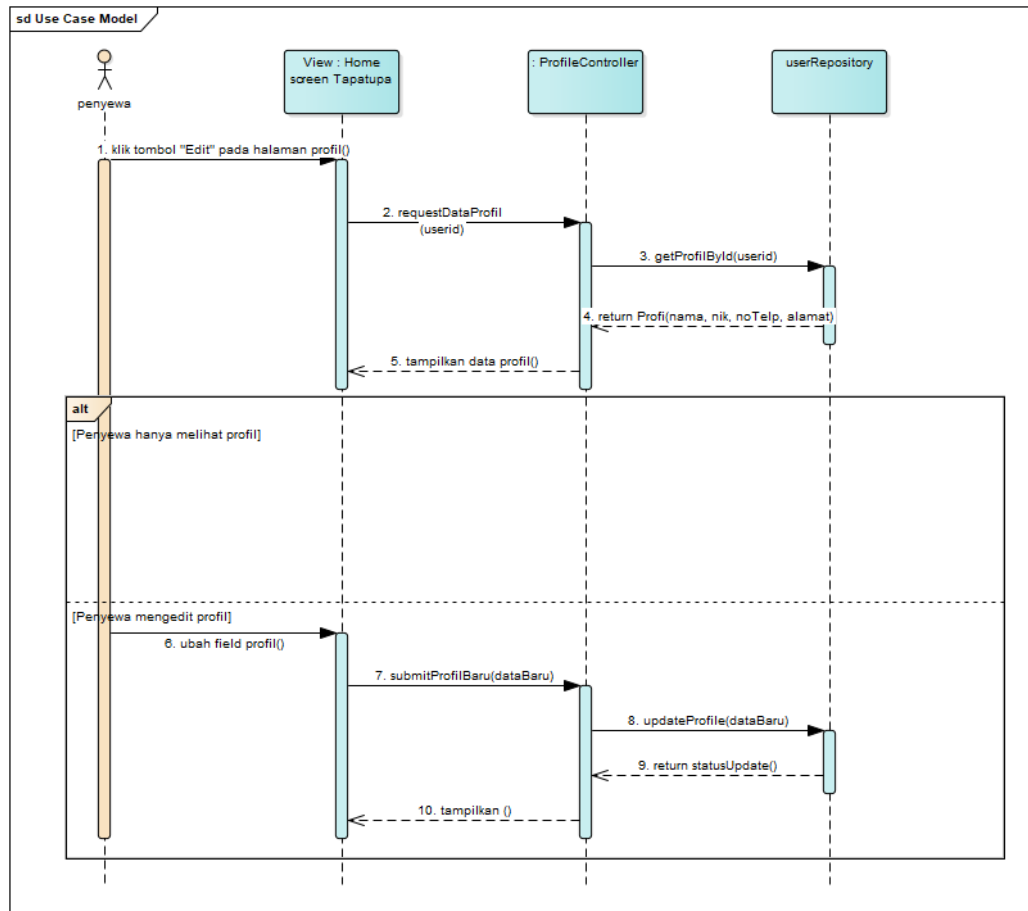
### 3.2.5.8. Sequence diagram Mengelola Profil

*Sequence* diagram di bawah ini menggambarkan alur interaksi antar objek dalam proses pengguna (penyewa) yang ingin melihat atau mengedit akun profil di aplikasi Tapatupa. Proses diawali saat aktor penyewa mengklik tombol "edit" pada halaman profil yang ditampilkan melalui tampilan antarmuka. Aksi ini menerima pengiriman permintaan data profil oleh tampilan ke *controller* melalui pesan *request* data profil menggunakan user id yang kemudian diteruskan ke repositori dalam bentuk permintaan *get* profil *by* id untuk mengambil data profil berdasarkan id pengguna. Setelah data berhasil diambil, user repositori mengembalikan dalam

bentuk atribut seperti nama, NIK, nomor telepon dan alamat. Data ini kemudian dikembalikan ke *controller*, dan selanjutnya ditampilkan kembali ke pengguna melalui tampilan antarmuka dalam bentuk informasi profil yang dapat dibaca. Sequence ini memiliki dua kemungkinan alur lanjutan yang ditampilkan dalam blok alt (alternatif) yaitu pengguna hanya melihat profil dan pengguna memilih untuk melakukan edit.

Jika pengguna hanya melihat profil, maka proses berakhir pada langkah tampilan data tanpa ada interaksi lanjutan. Namun, jika pengguna memilih untuk melakukan pengeditan, maka isi *field* profil pada tampilan berubah. Perubahan data ini dikirim kembali ke *profile controller* melalui pesan *submit* profil data baru, yang kemudian diteruskan ke *user* repositori dalam bentuk permintaan *update* data profil baru. Setelah data berhasil diperbarui, repositori akan mengembalikan *status update*, yang kemudian dikirimkan kembali ke tampilan untuk memberi tahu pengguna apakah pembaruan berhasil atau tidak, yang dibuat dalam bentuk notifikasi.

*Sequence* diagram ini menunjukkan bahwa proses edit profil penyewa merupakan kombinasi dari pengambilan data awal untuk ditampilkan dan proses pembaruan data jika terjadi perubahan oleh pengguna. Diagram ini juga menggambarkan bagaimana interaksi antara tampilan, *controller*, dan repositori berjalan dengan baik. Penggunaan blok alt menunjukkan percabangan logis yang menggambarkan dua skenario berbeda dalam satu *use case*, yaitu melihat profil saja atau melakukan pembaruan..



Gambar 3. 19 Sequence Diagram Kelola Profil



### 3.2.6. Entity Relationship Diagram (ERD)

*Entity Relationship* Diagram (ERD) berikut dirancang untuk menggambarkan struktur *database* sistem aplikasi Tapatupa versi *mobile* hasil *reengineering*. Fokus utama desain basis data adalah untuk mendukung proses bisnis penyewaan tanah milik daerah secara digital, mulai dari pengajuan permohonan hingga pelaporan pembayaran. Gambar ERD di bawah ini menggambarkan struktur basis data dari sistem yang dibangun, yang mencakup berbagai entitas utama beserta atributnya, serta relasi antar entitas yang merepresentasikan proses bisnis utama dalam sistem Tapatupa (Tapanuli Utara Penyewaan Aset) yang telah direkayasa ulang berdasarkan hasil analisis kebutuhan.

Pada dokumen, struktur rancangan ini mendukung fitur utama yang digunakan oleh penyewa wajib retribusi seperti login, permohonan, daftar objek retribusi, tarif retribusi, tagihan, pembayaran, perjanjian.

Adapun hubungan entitas dengan fitur utama seperti :

1. Login

Memiliki entitas yang terlibat dengan user, fitur ini membutuhkan autentikasi dengan menggunakan akun yang datanya tersimpan dalam entitas user, dengan atribut seperti username, password, dan role.

2. Permohonan

Memiliki entitas utama yaitu tabel permohonan. Relasi nya berhubungan dengan user dimana satu pengguna dapat membuat banyak permohonan. Dokumen pendukung juga merupakan salah satu relasi dimana satu permohonan dapat memiliki lebih dari satu dokumen. Selain itu, objek retribusi memiliki relasi satu permohonan terkait dengan satu objek, status permohonan menggambarkan status diproses, disetujui, atau ditolak. Permohonan digunakan sebagai dasar pembuatan tagihan dan perjanjian.

3. Daftar Objek Retribusi

Memiliki entitas objek retribusi, dengan menyimpan informasi objek tanah milik daerah seperti nama, lokasi, dan status ketersediaan. Data ditampilkan kepada pengguna sebelum membuat permohonan.

#### 4. Tarif Retribusi

Memiliki entitas tarif retribusi dengan relasi satu objek memiliki satu atau lebih tarif tergantung kategori. Data tarif digunakan untuk menghitung total retribusi dalam tagihan.

#### 5. Tagihan

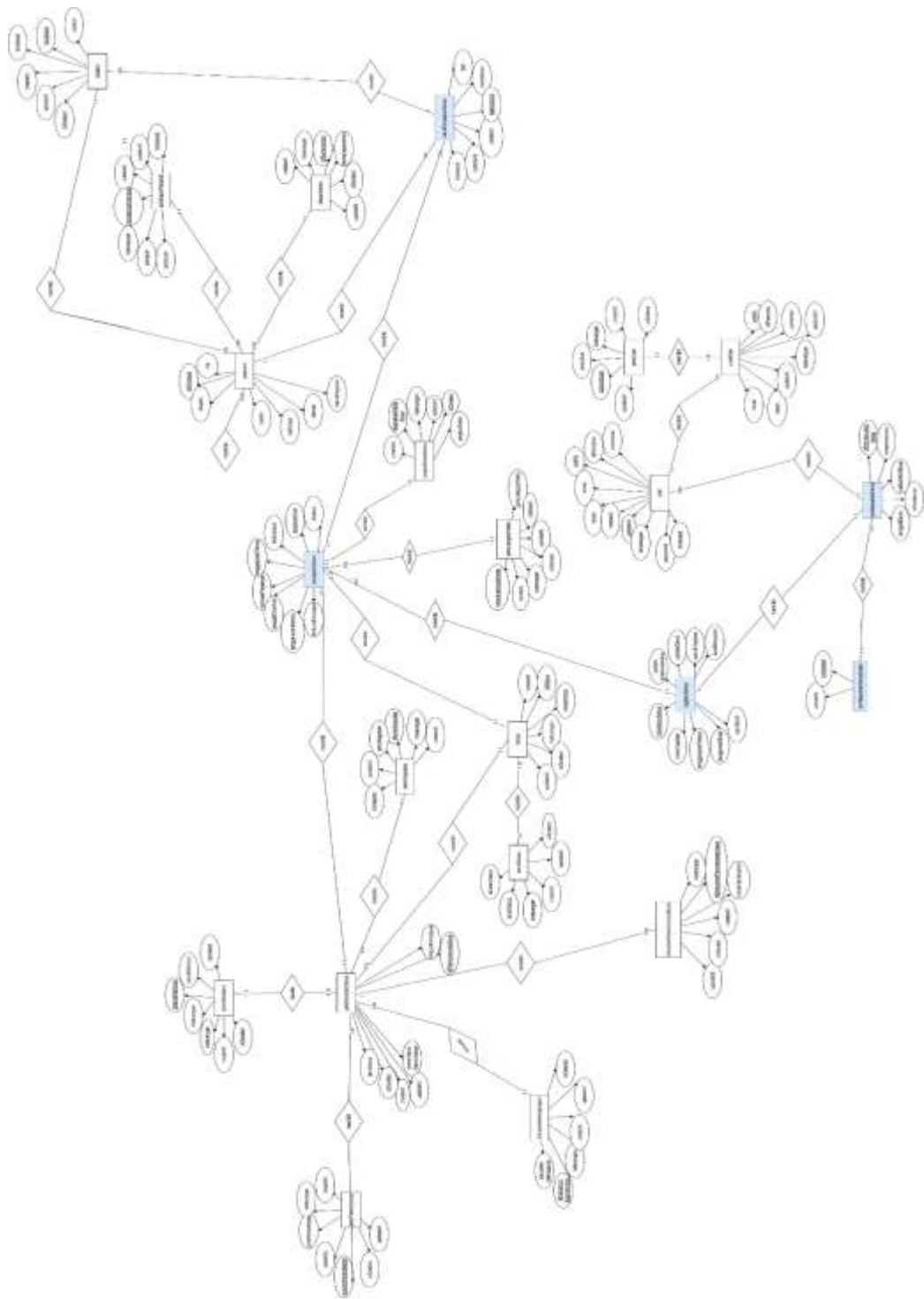
Memiliki entitas baru yaitu tagihan sewa dengan relasi satu tagihan dibuat berdasarkan satu permohonan, user sebagai pengguna akan dikenai tagihan, untuk itu *virtual account* menyimpan no VA untuk pembayaran. Relasi status tagihan menjelaskan tagihan seperti belum bayar atau sudah bayar, fitur ini memungkinkan pengguna melihat jumlah yang harus dibayar, dan jatuh tempo.

#### 6. Pembayaran

Memiliki entitas baru, seperti pembayaran sewa mencatat transaksi pembayaran, pembayaran detail sewa mencatat rincian pembayaran yang dibagi beberapa kali. Relasi pada tagihan sewa dimana, setiap pembayaran merujuk pada satu tagihan, user pengguna melakukan pembayaran, sistem dapat menampilkan riwayat pembayaran, nominal, tanggal bayar, dan nomor *virtual account*.

#### 7. Perjanjian

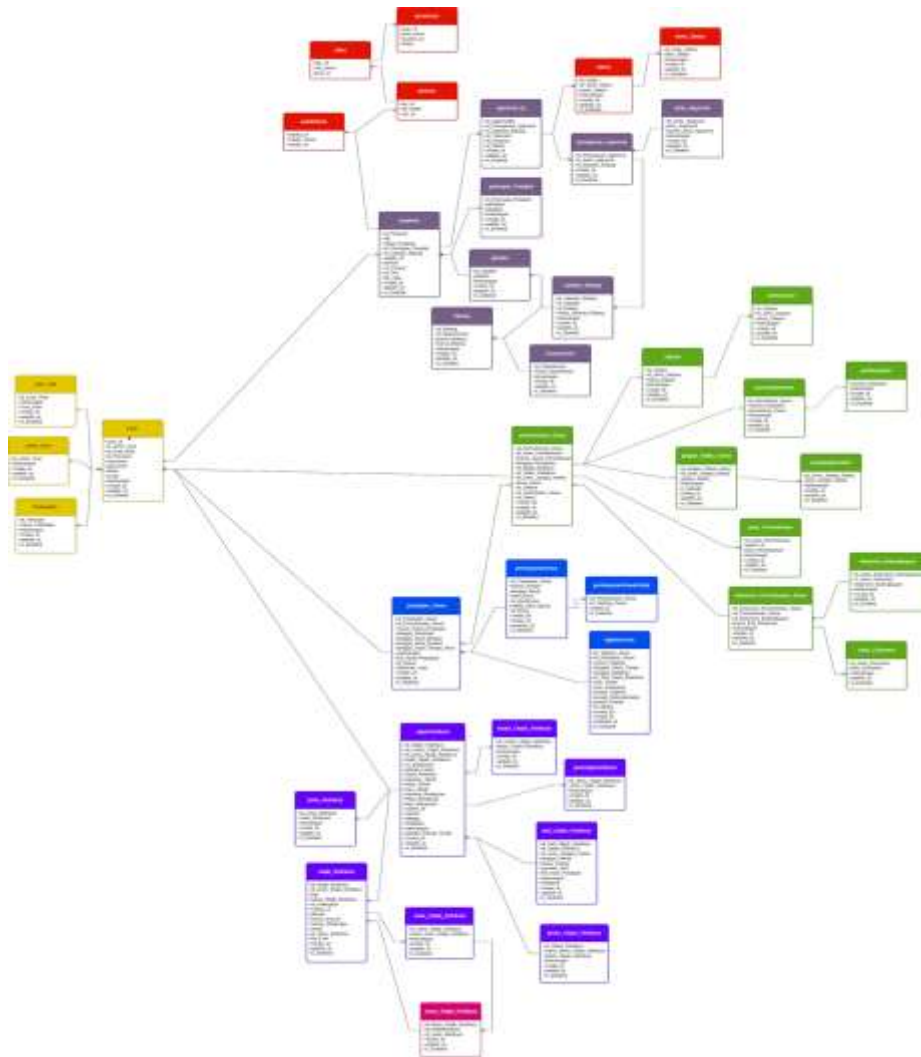
Memiliki entitas perjanjian sewa dengan relasi perjanjian dibuat berdasarkan permohonan yang disetujui, objek yang diperjanjikan, penyewa yang menandatangani perjanjian, saksi perjanjian sewa merupakan entitas baru yang mencatat pihak saksi, serta perjanjian disediakan dalam format digital dan bisa diunduh oleh pengguna.



**Gambar 3. 20 ERD Aplikasi Tapatupa**

### 3.2.7. Conceptual Data Model (CDM)

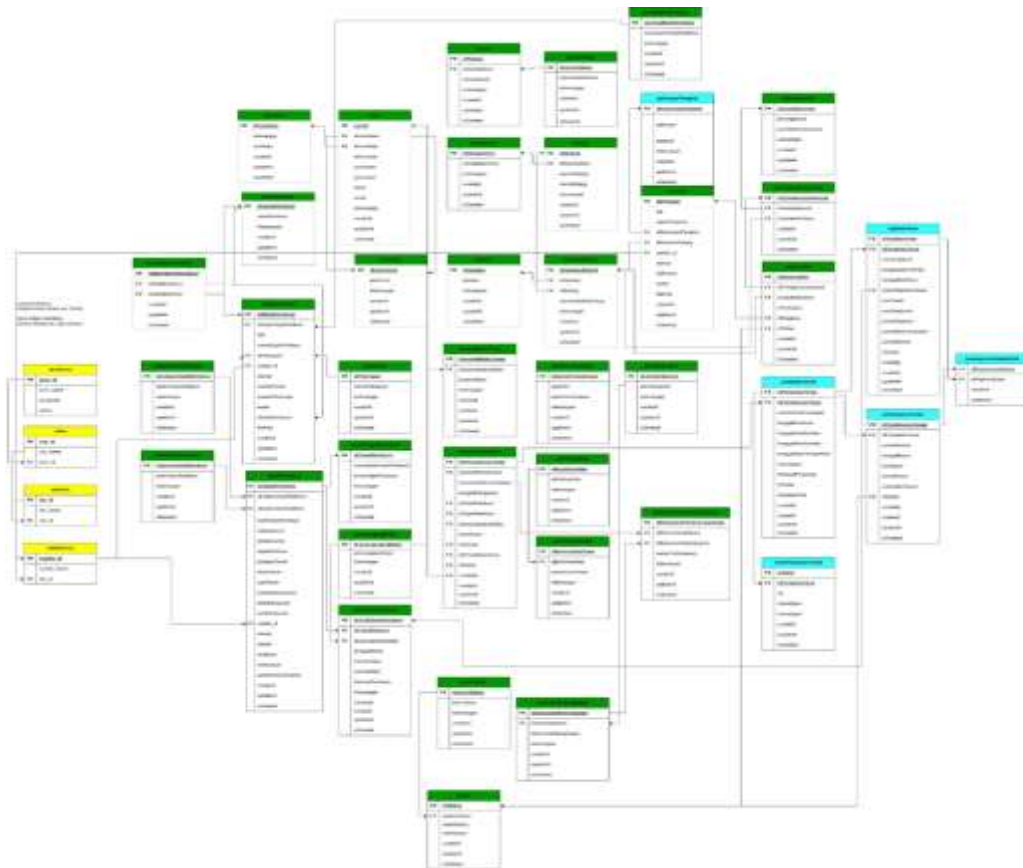
Diagram *conceptual data model* ini menggambarkan aplikasi Taptupa yang terstruktur, dimana berbagai entitas yang sesuai dengan permohonan penyewaan tanah saling berhubungan. Entitas utama meliputi permohonan sewa, perjanjian sewa dan pembayaran sewa. Setiap entitas ini memiliki atribut spesifik yang menggambarkan karakteristiknya masing-masing. Relasi antar entitas menunjukkan bagaimana setiap tabel berinteraksi satu sama lain dalam konteks aplikasi Tapatupa, misalnya setiap permohonan sewa yang sudah diajukan oleh penyewa dan sudah disetujui maka penyewa akan mendapatkan surat perjanjian yang dapat dimanfaatkan oleh pengguna. Dengan memahami relasi ini, dapat diatur dan dikelola sehingga memastikan data yang konsisten dan integrasi yang baik antara berbagai komponen dalam aplikasi Tapatupa. Conceptual Data Model yang dibuat berdasarkan ER-Diagram dapat dilihat pada Gambar 3. 21.



**Gambar 3. 21 Concept Data Model Aplikasi Tapatupa**

### 3.2.8. Physical Data Model (PDM)

Diagram *physical data model* (PDM) ini menggambarkan struktur fisik dari basis data untuk aplikasi Tapatupa, termasuk tabel - tabel yang ada, kolom - kolom yang terdapat dalam setiap tabel, serta hubungan antar tabel. *Physical Data Model* dapat dilihat pada Gambar 3.22.



**Gambar 3. 22 Physical Data Model Aplikasi Tapatupa**

### **3.2.9. User Interface Layout (Software)**

Pada sub bab ini berisi penjelasan tata letak antarmuka pengguna, berisi pedoman dan merancang UI aplikasi *mobile* Tapatupa seperti aturan dari warna sampai kepada aturan penulisan di *mobile*. *UI Style Guide* dapat dilihat pada Gambar 3.23.



Gambar 3. 23 UI Style Guide

### 3.2.9.1. Login Aplikasi

Halaman login berisi *field* yang harus diisi pengguna sebelum masuk ke halaman beranda. Terdapat 2 *field* yaitu *field username* dan *field password*. Lalu, ada *button login*, jika ditekan akan memastikan *username* dan *password* yang sudah diinput oleh pengguna terdaftar dan dapat masuk atau tidak. Apabila pengguna tidak dapat login, maka dapat mengklik tulisan “daftar disini” untuk mengarahkan pengguna ke halaman daftar akun dan melakukan registrasi terlebih dahulu. Halaman login screen user dapat dilihat pada Gambar 3.24.



**Gambar 3. 24 Tampilan awal Login**

#### **3.2.9.2. Permohonan**

Fitur Permohonan pada aplikasi Tapatupa merupakan salah satu komponen utama yang digunakan oleh penyewa untuk mengajukan permohonan sewa atas tanah milik pemerintah. Fitur ini memiliki dua bagian utama, yaitu "riwayat permohonan" yang menampilkan daftar permohonan sebelumnya, dan "buat permohonan" yang merupakan antarmuka formulir pengajuan baru. Tampilan antarmuka pengguna (*user interface*) pada bagian "buat permohonan" dirancang secara sederhana dan responsif dengan tata letak elemen yang vertikal dan mudah diakses pada perangkat *mobile*. Warna yang digunakan, seperti tombol hijau untuk mengirim permohonan dan tombol biru untuk menambahkan dokumen kelengkapan, sehingga mempermudah pengguna dalam memahami aksi yang harus dilakukan. Pada formulir permohonan, pengguna diminta untuk mengisi beberapa isian data. Isian



tersebut meliputi jenis permohonan, objek retribusi, jenis peruntukan sewa, periodisitas pembayaran, lama sewa, satuan waktu, dan catatan tambahan.

Seluruh isian disediakan dalam bentuk dropdown menu dan kotak teks untuk menjamin konsistensi data serta meminimalkan kesalahan input. Selain itu, terdapat tombol untuk menambahkan dokumen kelengkapan, seperti KTP, surat izin usaha, atau dokumen pendukung lainnya yang wajib diunggah. Input dari pengguna akan divalidasi terlebih dahulu pada sisi pengguna untuk memastikan format dan kelengkapan data telah sesuai. Setelah seluruh data diisi dengan benar, pengguna dapat mengirim permohonan dengan menekan tombol “kirim permohonan”. Sistem akan memproses data tersebut dengan menyimpannya ke dalam basis data. Dengan demikian, fitur ini mendukung transparansi dalam proses penyewaan aset tanah pemerintah secara digital.

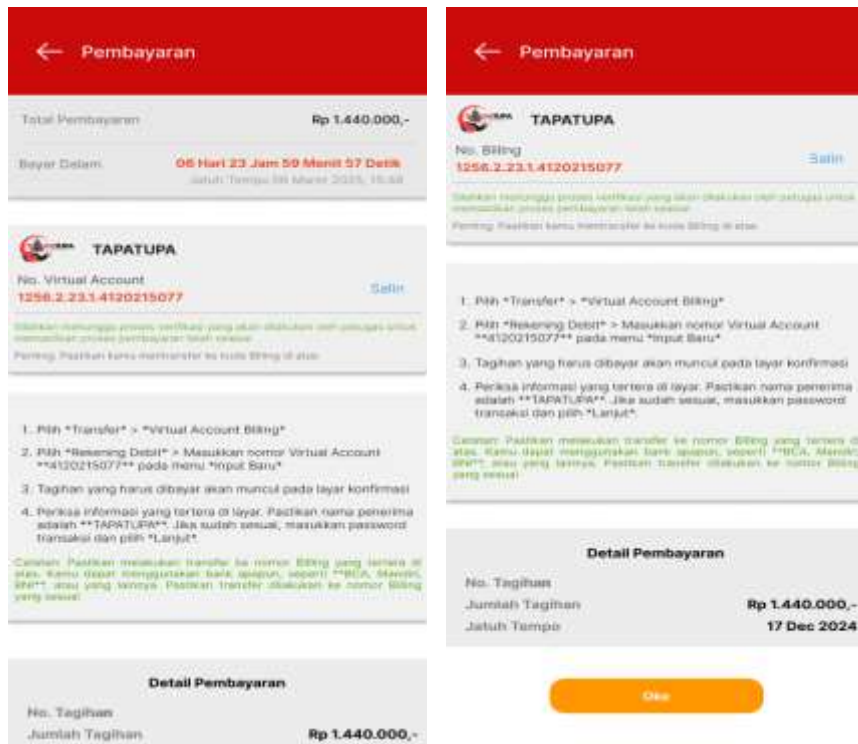
The image displays two side-by-side screenshots of a mobile application interface for submitting a rental request. Both screens feature a red header with a decorative pattern and a red house icon. The left screen is titled 'Permohonan' and shows a 'Formulir Permohonan' with the following fields: 'Jenis Permohonan', 'Objek Retribusi', 'Jenis Peruntukan Sewa', 'Perioditas', 'Lama Sewa', 'Satuan', and 'Catatan'. The right screen is titled 'Formulir Permohonan' and shows the same fields, but with a '+ Tambah Dokumen Kelengkapan' button below the 'Catatan' field and a 'Kirim Permohonan' button at the bottom. Both screens have a bottom navigation bar with icons for 'Home', 'Permohonan', 'Tagihan', 'Pembayaran', and 'Profil'.

Gambar 3. 25 Tampilan permohonan

### 3.2.9.3. Pembayaran

Fitur pembayaran pada aplikasi Tapatupa dirancang untuk memfasilitasi pengguna dalam melakukan pelunasan tagihan sewa atas tanah milik pemerintah secara digital dan mandiri. Tampilan antarmuka (*user interface*) pada fitur ini menampilkan informasi yang bersifat informatif dan *real-time*, termasuk total tagihan, waktu jatuh tempo, serta nomor *Virtual account* (VA) yang akan digunakan untuk transaksi. Pada bagian atas layar, pengguna dapat melihat batas waktu pembayaran yang ditampilkan beserta tanggal dan jam jatuh tempo secara spesifik. Desain visualnya menggunakan warna merah sebagai penanda perhatian untuk informasi penting, dan warna abu-abu sebagai latar untuk detail instruksi, yang membuat tampilan tetap jelas dan tidak membingungkan. Pengguna tidak perlu menginput data secara manual dalam fitur ini. Seluruh informasi seperti nomor *virtual account*, jumlah tagihan, dan instruksi pembayaran telah disediakan secara otomatis oleh sistem. Pengguna hanya diminta untuk menyalin nomor *virtual account* yang tertera dan melakukan pembayaran melalui aplikasi *mobile*. Proses pada fitur ini dimulai ketika pengguna masuk ke menu pembayaran, di mana sistem akan mengambil data tagihan yang masih aktif. Setelah pengguna melakukan transfer melalui bank daerah, sistem *backend* akan melakukan verifikasi otomatis berdasarkan data transaksi yang masuk. Verifikasi ini mencocokkan jumlah pembayaran, nomor VA, dan waktu transaksi. Jika pembayaran berhasil dan sesuai, maka status tagihan akan berubah menjadi “sudah bayar” dan sistem akan menghapus tagihan tersebut.

Output dari fitur ini berupa tampilan informasi status pembayaran yang dapat dicek langsung oleh pengguna, serta pencatatan transaksi ke dalam sistem sebagai bukti pembayaran. Selain itu, pengguna juga akan melihat rincian pembayaran seperti nomor tagihan, jumlah tagihan dalam format mata uang rupiah, serta tanggal jatuh tempo. Seluruh proses pada fitur ini bertujuan untuk memastikan bahwa pembayaran dilakukan secara tepat, aman, dan terdokumentasi dengan baik dalam sistem penyewaan digital Tapatupa.



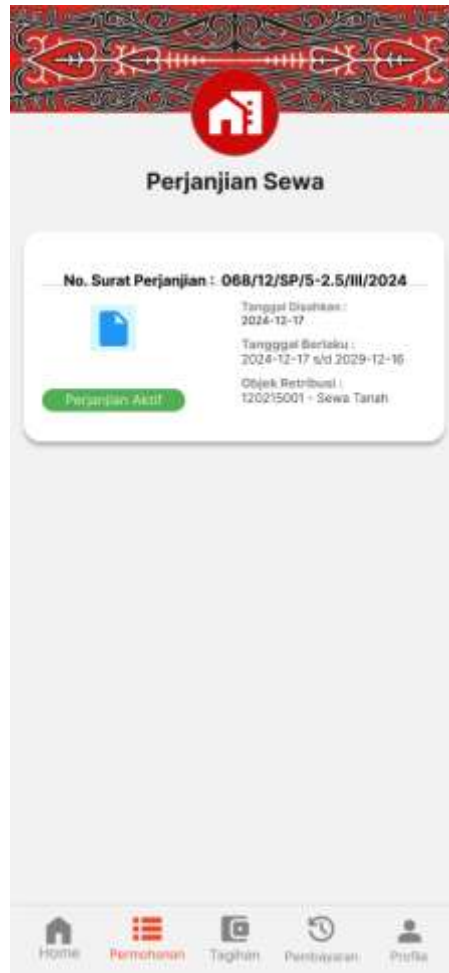
Gambar 3. 26 Tampilan pembayaran

#### 3.2.9.4. Perjanjian Sewa

Fitur Detail Perjanjian Sewa merupakan bagian dari aplikasi *mobile* yang dirancang untuk menampilkan informasi lengkap terkait perjanjian sewa milik pengguna. Tampilan antarmuka pengguna pada fitur ini didesain secara sederhana namun informatif, menampilkan *header* bertuliskan "detail perjanjian sewa". Fitur ini tidak memerlukan input secara langsung dari pengguna karena bersifat tampilan informasi. Seluruh data yang ditampilkan pada halaman ini diambil dari hasil input yang telah dilakukan sebelumnya pada saat permohonan atau pembuatan perjanjian. Data yang ditampilkan mencakup nomor surat perjanjian, tanggal disahkannya perjanjian dengan format standar YYYY-MM-DD, serta kode objek retribusi berupa angka, jenis objek retribusi seperti "sewa tanah", jenis jangka waktu, satuan waktu (tahun), serta bentuk panjang dan lebar tanah dalam format numerik desimal (misalnya 00.00).

Hasil *output* dari fitur ini adalah tampilan visual yang terstruktur berisi dokumen informasi lengkap mengenai perjanjian sewa, yang dapat digunakan oleh pengguna

sebagai bukti valid atas perjanjian yang telah disahkan. Format data yang ditampilkan disusun dengan pendekatan *user-friendly* agar mudah dibaca dan dipahami oleh pengguna umum tanpa memerlukan pengetahuan teknis mendalam.



**Gambar 3. 27 Tampilan Perjanjian Sewa**

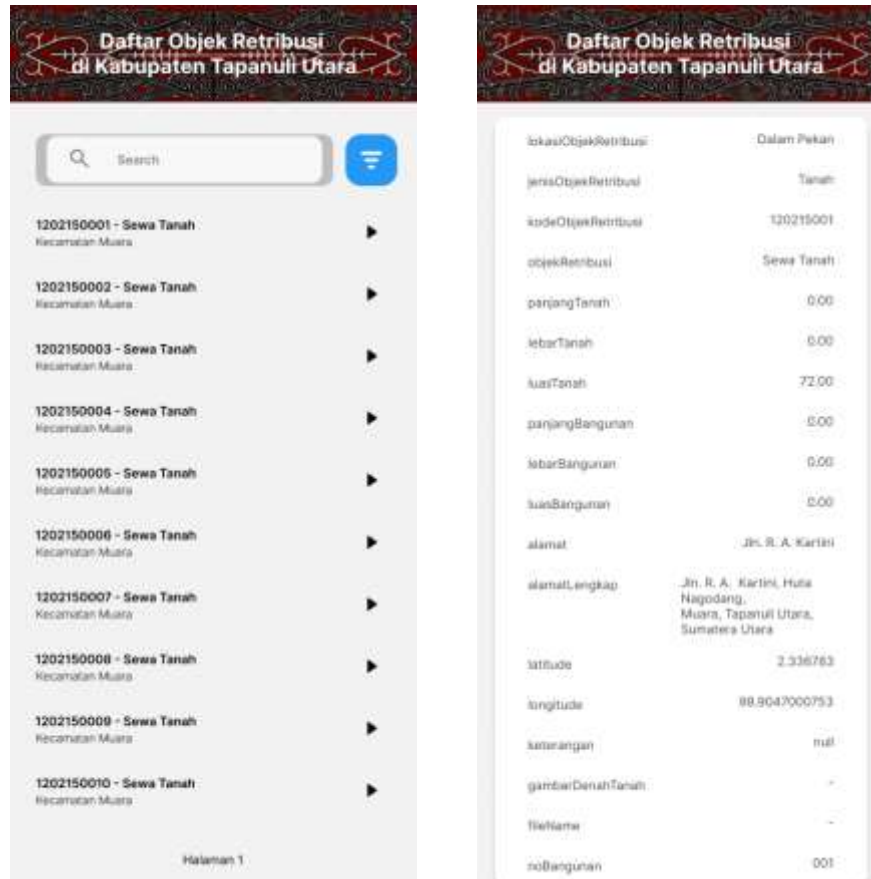
#### **3.2.9.5. Daftar Objek Retribusi**

Fitur Daftar Objek Retribusi di Kabupaten Tapanuli Utara merupakan bagian dari aplikasi yang berfungsi untuk menampilkan data seluruh objek retribusi yang tersedia di wilayah tersebut. Antarmuka pengguna pada fitur ini dirancang secara responsif dan sederhana, menampilkan daftar objek retribusi dalam bentuk list vertikal. Setiap daftar menampilkan kode objek retribusi, jenis objek (contoh: “sewa tanah”), serta lokasi (contoh : “Kecamatan Muara”). Pengguna juga dapat menggunakan kolom pencarian untuk mencari objek berdasarkan kata kunci tertentu dan ikon filter untuk mempersempit hasil pencarian sesuai dengan kategori

atau wilayah tertentu. Dari sisi input, fitur ini menerima masukan dari sistem yang sudah tersimpan di dalam basis data.

Secara eksplisit, pengguna dapat memberikan input berupa teks pada kolom pencarian untuk mencari objek tertentu, atau memilih filter dengan kategori yang telah ditentukan sebelumnya. Input yang diterima sistem dapat berupa kata kunci pencarian (kecamatan), sistem kemudian akan menyesuaikan hasil tampilan berdasarkan input pengguna tersebut. Secara proses, ketika pengguna membuka fitur ini, sistem akan secara otomatis mengambil seluruh data objek retribusi yang tersedia dari *database*, lalu menampilkannya dalam bentuk daftar. Jika pengguna menggunakan fitur pencarian atau filter, sistem akan memproses permintaan tersebut untuk menampilkan hasil yang sesuai. Ketika pengguna memilih salah satu objek dalam daftar, sistem akan menampilkan halaman detail yang memuat informasi lengkap dari objek retribusi tersebut.

Output dari fitur ini berupa dua tampilan utama, yaitu daftar objek retribusi dan detail objek retribusi. Pada tampilan daftar, data yang ditampilkan mencakup kode objek retribusi, nama objek, dan wilayah lokasi. Sementara pada tampilan detail objek, ditampilkan informasi lengkap seperti lokasi objek retribusi, jenis objek, kode objek, jenis retribusi, ukuran tanah, luas tanah, jumlah bangunan, alamat, koordinat latitude dan longitude, keterangan tambahan, file gambar denah (jika tersedia), nama file, serta nomor bangunan dalam bentuk yang mudah dibaca oleh pengguna.



Gambar 3. 28 Tampilan daftar objek retribusi

### 3.2.9.6. Tarif Objek Retribusi

Fitur Daftar Tarif Objek Retribusi pada aplikasi Tapatupa dirancang untuk menampilkan informasi tarif sewa tanah milik pemerintah Kabupaten Tapanuli Utara secara terstruktur dan informatif kepada pengguna umum. Antarmuka pengguna (user interface) fitur ini disusun dalam bentuk kartu informasi yang menyajikan data setiap objek retribusi secara individual. Menampilkan poin penting seperti kode objek retribusi, jenis objek, luas tanah, alamat lengkap, jangka waktu sewa, hingga nominal tarif yang dikenakan. Tampilan ini mengedepankan keterbacaan dan kemudahan akses informasi. Data yang ditampilkan pada fitur ini berasal dari input yang dimasukkan oleh petugas pengelola retribusi.

Format input mengikuti standar tertentu, misalnya kode objek retribusi dalam bentuk numerik maksimal 12 digit, luas tanah dan nominal tarif dalam format numerik desimal, serta tanggal yang mengikuti format YYYY-MM-DD. Data seperti alamat lengkap dan lokasi objek retribusi berupa teks deskriptif. Output dari

fitur ini berupa tampilan visual yang menyajikan informasi lengkap tentang tarif objek retribusi. Pengguna dapat melihat rincian seperti tarif objek, jenis retribusi, lokasi tanah, dan koordinat geografis (*latitude* dan *longitude*). Format output disusun dalam bentuk teks informatif yang terstruktur dan ditampilkan untuk memastikan transparansi dan mencegah perubahan data oleh pihak yang tidak berwenang. Dengan demikian, fitur ini menjadi sarana utama dalam memberikan informasi kepada masyarakat mengenai tarif dan detail teknis sewa tanah milik pemerintah, serta mendukung prinsip keterbukaan dan pelayanan publik yang responsif.

Daftar Tarif Objek Retribusi di Kabupaten Tapanuli Utara		Daftar Tarif Objek Retribusi di Kabupaten Tapanuli Utara	
<div> <div> <div></div> <div>Search</div> </div> </div>			
120201001 - Sewa Tanah	Rp 40.000,- <a href="#">Lihat Detail</a>	kodeObjekRetribusi	120201001
120215031 - Sewa Tanah	Rp 5.850,- <a href="#">Lihat Detail</a>	objekRetribusi	Sewa Tanah
120215006 - Sewa Tanah	Rp 30.000,- <a href="#">Lihat Detail</a>	noBangunan	1
120215001 - Sewa Tanah	Rp 20.000,- <a href="#">Lihat Detail</a>	jenisObjekRetribusi	Tanah
120201007 - Sewa Tanah	Rp 45.123,8 <a href="#">Lihat Detail</a>	lokasiObjekRetribusi	Luar Pekan
		panjangTanah	0.00
		lebarTanah	0.00
		luasTanah	95.00
		panjangBangunan	0.00
		lebarBangunan	0.00
		luasBangunan	0.00
		alamatLengkap	Jln. SM. Raja, Tarutung Hutatoruan X, Hutatoruan X, Tarutung, Tapanuli Utara, Sumatera Utara
		jumlahLantai	1
		kapasitas	10
		idJenisJangkaWaktu	1
		jenisJangkawaktu	Per Tahun
		tanggalDinilai	1970-01-01
		namaPenilai	null
		nominalTarif	40000
		hargaTanah	400000000

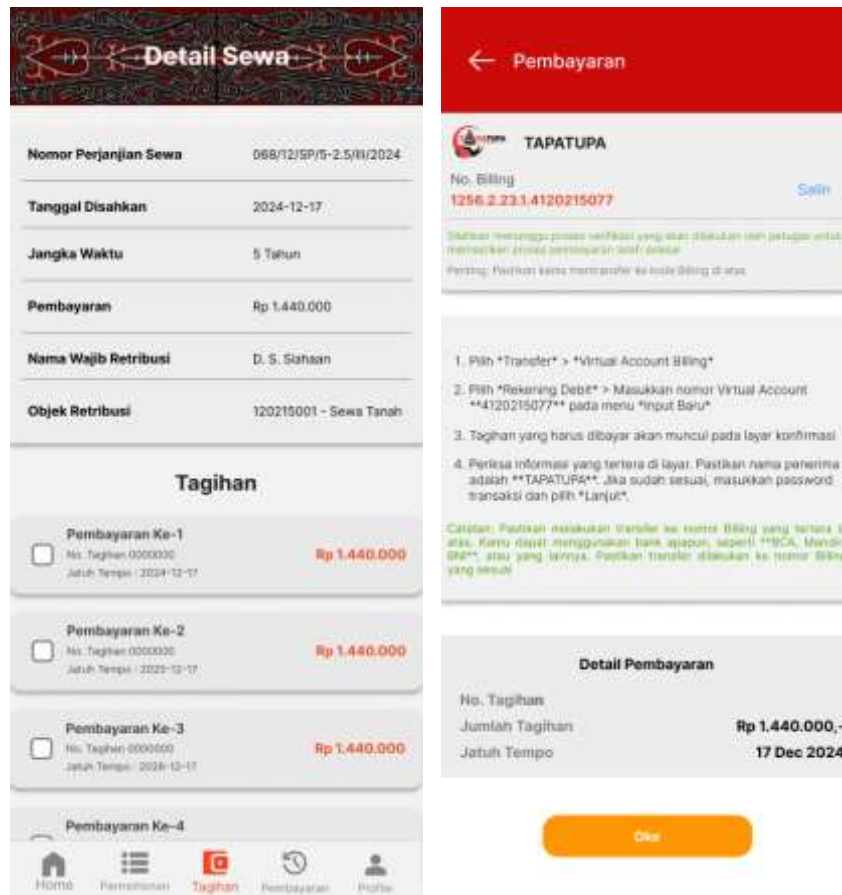
Gambar 3. 29 Tampilan Daftar Tarif Objek

### 3.2.9.7. Tagihan Sewa

Fitur detail sewa pada aplikasi Tapatupa berfungsi untuk menyajikan informasi lengkap mengenai perjanjian sewa tanah yang telah disahkan antara pengguna (penyewa) dengan pihak pengelola aset pemerintah. Tampilan antarmuka pengguna (*user interface*) dari fitur ini dirancang secara sederhana dan informatif, dengan menyusun data dalam format vertikal yang mudah dibaca. Informasi utama yang ditampilkan mencakup nomor perjanjian sewa, tanggal disahkan, jangka waktu, pembayaran per tahun, nama wajib retribusi, serta objek retribusi. Informasi-informasi ini ditampilkan dalam bentuk pasangan label dan nilai, yang membantu pengguna untuk memahami isi kontrak dan kewajiban sewa yang telah disepakati. Masukan (input) pada fitur ini berasal dari sistem administrasi atau petugas pengelola yang sebelumnya memasukkan data perjanjian sewa melalui antarmuka manajemen.

*Output* dari fitur Detail Sewa ini berupa informasi statis yang bersifat informatif. Format output disusun dalam tampilan antarmuka yang terstruktur, sehingga memudahkan pengguna dalam memahami isi perjanjian, termasuk besar pembayaran yang harus dilakukan setiap tahun serta masa berlaku sewa. Fitur ini bertujuan untuk meningkatkan transparansi, kejelasan, dan akuntabilitas dalam proses penyewaan aset milik pemerintah.



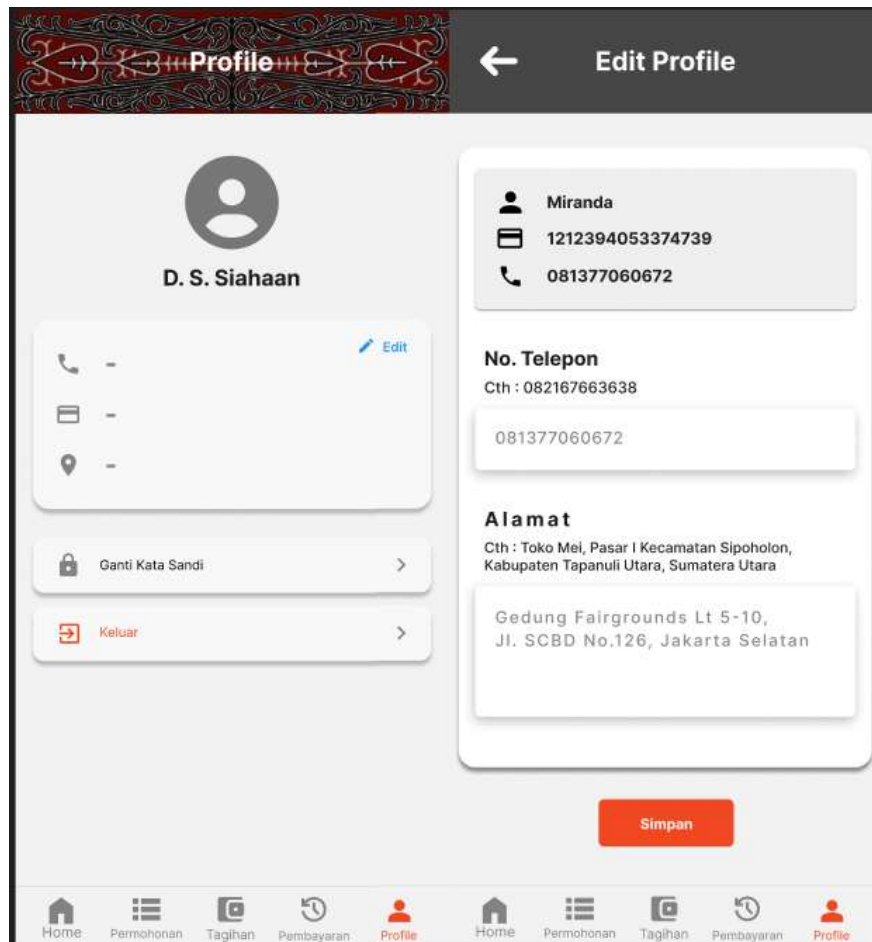


Gambar 3. 30 Tampilan Tagihan Sewa

### 3.2.9.8. Akun Profile

Fitur Pengelolaan Profil pada aplikasi Tapatupa memungkinkan pengguna untuk melihat dan memperbarui informasi akun pribadi. Tampilan antarmuka pengguna (*user interface*) pada halaman ini dirancang dengan sederhana, menampilkan nama pengguna di bagian atas. Di bawahnya, tersedia informasi dasar seperti nomor telepon, alamat email, dan alamat domisili pengguna. Setiap informasi tersebut ditampilkan dalam format ikon dan teks, serta dilengkapi dengan tombol edit yang memungkinkan pengguna melakukan perubahan data secara langsung. Selain itu, terdapat dua tombol aksi di bagian bawah, yaitu ganti kata sandi untuk keamanan akun, dan Keluar untuk keluar dari sesi aplikasi. Masukan (*input*) pada fitur ini mencakup data pribadi pengguna, seperti, nomor telepon, alamat email, alamat, dan kata sandi Baru.

Output dari fitur ini berupa tampilan informasi pengguna yang telah diperbarui, yang ditampilkan secara langsung setelah proses penyimpanan berhasil. Format output ditampilkan dalam bentuk teks statis yang hanya bisa diubah melalui aksi edit. Dengan adanya fitur ini, aplikasi Tapatupa memastikan bahwa pengguna dapat menjaga akurasi dan keamanan data pribadinya, serta memberikan kendali penuh terhadap informasi akun secara mandiri.



Gambar 3. 31 Tampilan profile

### 3.2.10. Arsitektur Sistem (*Hardware*)

Pada sistem ini, pengguna (wajib retribusi) hanya dapat mengakses layanan melalui aplikasi *mobile* yang dikembangkan menggunakan *Flutter*. Perangkat keras yang digunakan berupa *smartphone* atau tablet berbasis Android, dengan dukungan koneksi internet stabil melalui jaringan Wi-Fi atau data seluler.

Pengguna melakukan berbagai aktivitas utama seperti registrasi dan *login* (otentikasi), menelusuri daftar aset tanah yang tersedia, melihat informasi tarif sewa, mengajukan permohonan sewa, melakukan pembayaran tagihan, serta mengunduh dokumen perjanjian sewa. Semua permintaan dari aplikasi *mobile* dikirim dan diterima ke server *backend* menggunakan REST API dalam format JSON.

Pada sisi server, infrastruktur *backend* menggunakan *framework* Laravel yang bertugas melayani seluruh permintaan dari aplikasi *mobile*. *Backend* ini terhubung dengan server *database* MySQL yang berfungsi menyimpan data akun pengguna, data aset tanah seperti lokasi, tarif, harga, serta data transaksi penyewaan termasuk permohonan dan pembayaran. Alur komunikasi berjalan dari aplikasi *Flutter* ke server *backend*, dilanjutkan dengan query ke *database*, kemudian hasilnya dikembalikan lagi melalui *backend* ke aplikasi *mobile*, semuanya melalui koneksi yang aman. Arsitektur ini dirancang untuk memastikan pengalaman pengguna yang berfokus pada seluruh operasi utama berjalan lancar di perangkat *mobile* tanpa bergantung pada akses *web*.

## **BAB IV**

### **PRODUCT IMPLEMENTATION (PI)**

#### **(IMPLEMENTASI PENGEMBANGAN PRODUK)**

##### **4.1. PENDAHULUAN**

Bab ini merujuk pada desain pengembangan produk yang telah dibahas pada bab sebelumnya. Implementasi pengembangan aplikasi Tapatupa dilakukan berdasarkan rancangan sistem yang telah ditetapkan, meliputi struktur data, alur proses fungsional, serta desain antarmuka pengguna. Proses implementasi bertujuan untuk menerjemahkan rancangan tersebut ke dalam bentuk nyata berupa aplikasi *mobile* yang dapat digunakan oleh masyarakat sebagai penyewa serta petugas pengelola aset daerah di Kabupaten Tapanuli Utara.

Sistem hasil *reengineering* ini dikembangkan untuk mendukung digitalisasi proses administrasi penyewaan tanah milik daerah, dengan tetap mengacu pada ketentuan Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024. Proses implementasi mengacu pada pendekatan software re-engineering dan dilaksanakan melalui beberapa tahapan utama yang saling terintegrasi.

Adapun tahapan implementasi dalam tugas akhir ini meliputi:

1. *Inventory Analysis*, mengidentifikasi fitur, struktur data, dan alur proses dari aplikasi sebelumnya untuk mengevaluasi bagian yang dapat dipertahankan atau perlu ditingkatkan.
2. *Document Restructuring*, menyusun ulang dokumentasi teknis, termasuk diagram alur sistem dan kebutuhan fungsional, agar sesuai dengan standar pengembangan perangkat lunak.
3. *Code Refactoring*, melakukan perbaikan dan penyusunan ulang kode program yang masih sesuai agar lebih bersih, modular, dan siap digunakan kembali pada pengembangan sistem baru.
4. *Data Restructuring*, melakukan desain ulang struktur basis data untuk meningkatkan keakuratan, keterhubungan antar entitas, dan kemudahan integrasi dengan sistem pembayaran.
5. *Forward Engineering*, menerjemahkan hasil desain ke dalam bentuk aplikasi *mobile* melalui proses pengkodean dan pengujian fungsional.

Pada tahap *forward engineering*, proses pengembangan dilakukan dengan pendekatan Agile Scrum yang memungkinkan sistem dibangun secara bertahap dan adaptif terhadap kebutuhan yang telah diidentifikasi pada tahap analisis. Setiap iterasi atau *sprint* menghasilkan peningkatan fitur berdasarkan hasil evaluasi sistem sebelumnya dan rencana pengembangan yang telah ditetapkan dalam dokumen desain.

Implementasi ini diharapkan menghasilkan sistem aplikasi Tapatupa hasil *reengineering* yang selaras dengan kebutuhan pemangku kepentingan, meningkatkan ketepatan pengelolaan administrasi penyewaan aset daerah, serta mempercepat layanan kepada masyarakat melalui sistem *mobile* yang modern, terstruktur, dan mudah diakses.

#### **4.2. DESKRIPSI**

Bagian ini menjelaskan proses dan prinsip implementasi produk aplikasi Tapatupa berbasis *mobile* secara lebih mendalam. Implementasi dilakukan berdasarkan dokumen desain produk yang telah disusun sebelumnya pada tahap *Product Design*, serta mengikuti tahapan dan prinsip *software reengineering* untuk memastikan kualitas dan keberlanjutan sistem.

Proses implementasi bertujuan untuk membangun kembali sistem yang sebelumnya terbatas pada penggunaan internal berbasis web, menjadi aplikasi *mobile* yang dapat diakses langsung oleh masyarakat wajib retribusi. Seluruh aktivitas implementasi mengacu pada prinsip pengembangan yang adaptif dan terstruktur, yaitu dengan menggunakan metodologi Agile Scrum dan pendekatan *User-Centered Design*(UCD).

Pengembangan dilakukan dalam beberapa iterasi dengan tujuan mengembangkan fitur utama secara bertahap, dimulai dari pengembangan *Minimum Viable Product* (MVP) hingga ke tahap penyempurnaan antarmuka dan fungsionalitas sistem. Selain itu, implementasi juga mengikuti lima tahapan *software reengineering*, yaitu *inventory analysis*, *document restructuring*, *code refactoring*, *data restructuring*, dan *forward engineering*.

Masing-masing tahapan dijalankan secara berurutan untuk memecahkan masalah dari sistem lama, menyusun ulang struktur dokumentasi dan data, menyempurnakan kode program, serta membangun kembali sistem dengan fitur yang sesuai kebutuhan pengguna. Proses ini memastikan bahwa aplikasi dapat dikembangkan dengan kebutuhan operasional serta regulasi daerah yang berlaku.

Dengan pendekatan tersebut, implementasi produk Tapatupa tidak hanya menghasilkan sistem yang dapat berjalan secara fungsional, tetapi juga siap untuk digunakan oleh masyarakat dan mendukung transparansi dalam pengelolaan penyewaan aset daerah.

#### **4.2.1. Prinsip Implementasi**

Pada sub bab ini, akan dijelaskan prinsip - prinsip yang digunakan selama proses implementasi pengembangan aplikasi Tapatupa untuk memastikan bahwa produk yang dihasilkan sesuai dengan prinsip-prinsip yang digunakan selama proses implementasi pengembangan aplikasi Tapatupa. Prinsip-prinsip ini disusun berdasarkan hasil analisis terhadap aplikasi versi sebelumnya, aplikasi sejenis sebagai pembandingan, serta regulasi yang berlaku menurut Peraturan Bupati no 12 tahun 2024. Prinsip - prinsip ini diterapkan pada setiap tahap pengembangan produk dan berperan penting dalam memastikan kualitas dan fungsionalitas produk akhir. Kemudian, pada proses pengembangan ulang aplikasi Tapatupa, digunakan pendekatan metodologi gabungan antara Agile Scrum sebagai metode pengembangan perangkat lunak dan *User-Centered Design*(UCD) sebagai pendekatan penelitian yang berfokus pada kebutuhan pengguna. Integrasi kedua metodologi ini bertujuan untuk menghasilkan sistem yang iteratif, fleksibel, dan responsif terhadap umpan balik pengguna di setiap tahapan pengembangan

Tahapan implementasi dalam integrasi Agile Scrum dengan UCD terdiri atas:

1. *Empathize dan Define* (UCD):

Sebagai bagian dari tahapan *Empathize dan Define* dalam pendekatan *User-Centered Design*(UCD), proses pengumpulan kebutuhan pengguna dilakukan melalui penyebaran instrumen kuesioner kepada responden yang merupakan calon pengguna sistem. Untuk menjamin bahwa data yang diperoleh dari

kuesioner memiliki tingkat keandalan dan ketepatan yang memadai, dilakukan uji validitas dan reliabilitas terhadap setiap butir pertanyaan yang disusun. Uji ini bertujuan untuk memastikan bahwa kuesioner benar-benar mengukur apa yang seharusnya diukur dan menghasilkan data yang konsisten. Proses penyebaran kuesioner dilakukan sebanyak tiga kali dalam tiga tahapan iterasi berbeda, seiring dengan proses pengembangan dan penyempurnaan sistem. Pada subbab ini, disajikan hasil pengujian validitas dan reliabilitas terhadap instrumen kuesioner yang digunakan, sebagai dasar pengambilan keputusan dalam perancangan antarmuka dan fitur sistem yang berorientasi pada kebutuhan pengguna.

2. *Ideate* dan *Plan* (UCD + Scrum Planning):

Menentukan solusi awal terhadap kebutuhan pengguna dan menyusun *product backlog*. Tim menyusun sprint planning untuk iterasi pertama berdasarkan fitur prioritas.

3. *Develop* dan *Prototype* (Scrum Sprint):

Proses pengembangan fitur dilakukan dalam tiga *sprint* (iterasi) utama. Pada akhir setiap sprint, tim menghasilkan increment fungsional dari sistem.

4. *Test* dan *Evaluate* (UCD Testing + Sprint Review):

Setiap akhir sprint dilakukan evaluasi dengan melibatkan pengguna (melalui kuesioner, usability test, dan feedback terbuka). Hasil evaluasi menjadi dasar revisi backlog pada sprint berikutnya.

5. *Refine* dan *Deliver* (Scrum Retrospective):

Sprint terakhir ditujukan untuk perbaikan akhir berdasarkan umpan balik dari iterasi sebelumnya dan penyempurnaan sistem sebelum implementasi akhir.

Pengembangan aplikasi dilakukan dalam tiga sprint, yang masing-masing mencakup fitur-fitur berikut:

Sprint 1: Permohonan, tagihan, dan pembayaran

Sprint 2: Objek retribusi, tarif, perjanjian, profil pengguna, dan (fokus UAT pembayaran)

Sprint 3: Penyempurnaan keseluruhan sistem berdasarkan *feedback usability* dan hasil kuesioner akhir

Penggabungan metode Agile Scrum dengan pendekatan *User-Centered Design*(UCD) dalam pengembangan aplikasi ini bertujuan untuk memastikan bahwa proses tidak hanya berfokus pada sisi teknis, tetapi juga memperhatikan kebutuhan pengguna. Dengan pendekatan ini, pengembangan dilakukan secara bertahap dan melibatkan pengguna dalam setiap iterasi, sehingga masukan dari pengguna dapat langsung diterima dan dijadikan dasar perbaikan sistem. Melalui cara ini, aplikasi Tapatupa diharapkan dapat menjadi solusi digital yang tidak hanya fungsional, tetapi juga lebih mudah digunakan, sesuai dengan kondisi lapangan, serta mampu meningkatkan kepuasan pengguna secara keseluruhan.

#### **4.2.2. Tahapan dan Hasil Implementasi**

Tahapan ini merupakan bagian dari proses *software reengineering* yang berfokus pada pembangunan ulang sistem berdasarkan hasil perancangan yang telah disusun sebelumnya. Implementasi dilakukan secara bertahap menggunakan pendekatan iteratif dalam tiga siklus pengembangan, dengan mengadopsi prinsip Scrum untuk memastikan pelaksanaan backlog berjalan sistematis dan terkontrol. Masing-masing iterasi memuat tugas-tugas implementasi yang dirancang berdasarkan prioritas kebutuhan sistem, dimulai dari pengembangan fungsi dasar, integrasi antar modul, hingga penyempurnaan antarmuka dan pengujian akhir.

##### **4.2.2.1. Inventory Analysis**

*Inventory analysis* diawali dengan pengumpulan dokumen terkait, yaitu dokumen aplikasi Tapatupa versi awal, dokumen regulasi berupa Peraturan Bupati Tapanuli Utara Nomor 12 Tahun 2024, dan referensi dari tiga aplikasi sejenis, Siap Sewa, Amanah, dan SIWA. Proses ini dilakukan oleh tim peneliti internal dan mencakup tiga aktivitas utama. Pertama, analisis aplikasi pembandingan dilakukan untuk mengidentifikasi fitur unggulan dan kekurangan dari sistem lain, terutama terkait proses sewa aset daerah. Kedua, evaluasi sistem Tapatupa versi lama dilakukan terhadap enam fitur utama, seperti permohonan, pembayaran, dan tagihan. Evaluasi menghasilkan temuan spesifik, seperti ketiadaan fitur unggah dokumen dan



pembayaran manual tanpa otomatisasi. Ketiga, dilakukan analisis terhadap seluruh pasal dan ketentuan dalam Peraturan Bupati yang mencakup minimal enam aspek hukum, termasuk durasi sewa, penetapan tarif, dan perpanjangan. Output dari tahap ini adalah dokumen analisis kebutuhan, daftar 7 legacy fitur yang diidentifikasi (seperti halaman permohonan, objek retribusi, tarif, perjanjian, tagihan, pembayaran, dan beranda), serta daftar kebutuhan sistem baru berdasarkan regulasi yang berlaku.

**Tabel 4.1 Hasil Inventory Analysis**

Sumber Analisis	Langkah yang Dilakukan	Temuan Utama	Tindak Lanjut
Aplikasi Sebelumnya	Review seluruh fitur dan tampilan antarmuka pengguna serta alur aplikasi.	7 fitur utama teridentifikasi : <ol style="list-style-type: none"> <li>1. Permohonan</li> <li>2. objek Retribusi</li> <li>3. Tarif Objek Retribusi</li> <li>4. Perjanjian</li> <li>5. Tagihan</li> <li>6. Pembayaran</li> <li>7. <i>Profile</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Pertahankan seluruh fitur utama.</li> <li>2. Memperbaiki alur fitur permohonan, tagihan sewa, dan pembayaran.</li> </ol>
Aplikasi Sejenis (berdasarkan Tabel Objek Perbandingan pada <b>Tabel 1.7</b> )	Melakukan benchmarking terhadap tiga aplikasi penyewaan aset publik, yaitu SiapSEWA, Amanah Bandung, dan SIWA App. Analisis dilakukan terhadap fitur utama, kelengkapan informasi, sistem pembayaran, dan pelaporan.	<ol style="list-style-type: none"> <li>1. Aplikasi ideal memiliki informasi aset yang detail dan menarik (gambar, video, deskripsi).</li> <li>2. Sistem pembayaran dan pelaporan yang terintegrasi penting untuk efisiensi pengelolaan aset.</li> <li>3. Pengguna membutuhkan pelaporan aktivitas dan status aset secara real-time.</li> <li>4. Fitur pencarian dan navigasi yang intuitif</li> </ol>	<ol style="list-style-type: none"> <li>1. Tambahkan informasi aset secara lengkap (gambar, dan deskripsi).</li> <li>2. Kembangkan fitur pembayaran dan pelaporan yang terintegrasi.</li> <li>3. Tambahkan kolom pencarian dan penyaringan aset untuk kemudahan akses.</li> </ol>

Sumber Analisis	Langkah yang Dilakukan	Temuan Utama	Tindak Lanjut
		memudahkan pengguna.	
Dokumen Peraturan Bupati No. 12 Tahun 2024	Menganalisis isi peraturan secara sistematis, khususnya pada bab yang mengatur mekanisme sewa tanah dan bangunan milik daerah, tarif, durasi sewa, proses perjanjian, dan metode pembayaran.	1. Prosedur sewa harus mengikuti tahapan permohonan, penetapan tarif, pembuatan perjanjian, dan pelunasan. 2. Terdapat ketentuan jangka waktu sewa, tarif berdasarkan jenis dan lokasi objek. 3. Dokumen sah harus disertakan dalam permohonan sewa. 4. Pembayaran sewa harus dilakukan sebelum jatuh tempo dan dicatat dalam sistem.	1. Menambahkan form unggah dokumen resmi pada proses permohonan. 2. Menyediakan informasi tarif berdasarkan kategori objek dan lokasi. 3. Menyusun alur sistem yang sesuai dengan tahapan sewa sesuai regulasi. 4. Mengintegrasikan sistem notifikasi pembayaran untuk mengingatkan tenggat waktu dan memastikan akuntabilitas.

#### 4.2.2.2. Document Restructuring

Input utama pada tahap ini adalah dokumen *Software Technical Document* (SWTD) versi awal yang disusun oleh lima peneliti sebelumnya. Dokumen tersebut terdiri dari enam modul utama, yaitu permohonan, objek retribusi, tarif objek retribusi, perjanjian, tagihan, dan pembayaran. Proses restrukturisasi dilakukan oleh tiga peneliti lanjutan yang bertanggung jawab atas penyusunan ulang struktur dokumen, pembaruan isi, serta pengembangan modul tambahan berdasarkan kebutuhan sistem target. Langkah-langkah teknis yang dilakukan meliputi, reorganisasi bab dan subbab untuk menyusun kembali alur pembahasan menjadi lebih sistematis, penghapusan konten yang tidak lagi digunakan seperti prosedur lama dalam fitur permohonan, penambahan enam modul baru yang meliputi login pengguna, autentikasi, unggah dokumen, integrasi *virtual account* untuk pembayaran, riwayat transaksi, dan notifikasi otomatis berbasis status permohonan.

Selain itu, beberapa bagian dalam dokumen yang sebelumnya terlalu panjang dipecah menjadi bagian-bagian yang lebih fokus, sedangkan beberapa bagian dengan isi yang saling terkait digabungkan untuk menghindari pengulangan informasi. Elemen visual seperti diagram BPMN dan user flow diagram diperbarui berdasarkan analisis use case aktual yang mencakup seluruh fitur utama. BPMN digunakan untuk menggambarkan proses permohonan hingga validasi pembayaran, sementara user flow diagram menunjukkan interaksi pengguna dengan sistem dari awal login hingga menerima notifikasi. Daftar isi juga disusun ulang agar mencerminkan struktur terbaru secara akurat. Dokumen yang telah direstrukturisasi kemudian diberi versi terbaru yang terdokumentasi dengan jelas untuk membedakannya dari versi sebelumnya. Output dari tahap ini adalah Dokumen Pengembangan Produk (DPP) final yang memiliki struktur rapi, diagram yang relevan, dan isi yang konsisten dengan kebutuhan pengembangan sistem *reengineering* Tapatupa.

**Tabel 4. 2 Hasil Document Restructuring**

Aspek dokumen restrukturisasi	Deskripsi Singkat
Reorganisasi Bab/ Subbab	Memindahkan bab atau subbab untuk mengikuti alur yang lebih baik.
Penambahan atau Penghapusan Konten	Menghapus modul usang dan menambahkan fitur baru seperti login, autentikasi, unggah dokumen, integrasi VA, riwayat, dan notifikasi otomatis.
Pemecahan dan Penggabungan Konten	Membagi bagian yang terlalu panjang dan menggabungkan bagian sejenis untuk meningkatkan kejelasan untuk mengurangi redundansi.
Penambahan Elemen Visual	Memperbarui BPMN dan <i>user flow</i> diagram agar sesuai dengan skenario sistem terkini dan mudah dipahami oleh tim pengembang maupun <i>stakeholder</i> .
Penyusunan Ulang Daftar Isi	Menyesuaikan daftar isi dengan perubahan struktur dan penambahan modul baru.

Aspek dokumen restrukturisasi	Deskripsi Singkat
Perubahan versi dokumentasi	Memperbarui versi dari dokumentasi sesuai dengan sejarah versi yang sudah ada.
Penyusunan Ulang Dokumentasi <i>Business Process Model</i> .	Menyusun ulang proses bisnis menggunakan notasi BPMN untuk meningkatkan konsistensi dan pemahaman alur kerja sistem.
Penggambaran Ulang <i>User Flow Diagram</i>	Alur interaksi pengguna ( <i>user flow</i> ) diperbaharui dengan membuat skenario berbasis <i>use case</i> untuk semua fitur utama. Merancang ulang diagram interaksi pengguna berbasis <i>use case</i> untuk semua fitur utama, memastikan alur interaksi terekam secara sistematis.

#### 4.2.2.3. Code Refactoring

Pada tahap ini dilakukan upaya penyempurnaan terhadap struktur kode sumber aplikasi Tapatupa melalui proses *refactoring*. *Refactoring* dipilih sebagai pendekatan karena bertujuan memperbaiki kualitas internal perangkat lunak tanpa mengubah fungsionalitas yang telah berjalan. Fokus utama dalam tahap ini adalah menyederhanakan logika, memisahkan tanggung jawab modul, serta meningkatkan keterbacaan dan kesiapan sistem untuk dikembangkan di masa mendatang.

Pada tahap ini, dilakukan *refactoring* kode terhadap tiga fitur utama dalam aplikasi Tapatupa, yaitu Permohonan, Tagihan, dan Perjanjian. Evaluasi dilakukan menggunakan alat bantu *SonarQube*, yang menghasilkan berbagai metrik kuantitatif seperti *lines of code*, *maintainability*, *reliability issues*, *cyclomatic complexity*, dan *duplication density*. Namun, dalam praktiknya, hasil *refactoring* tidak sepenuhnya selaras dengan harapan sebelumnya. Beberapa metrik justru menunjukkan peningkatan angka yang secara sekilas tampak negatif. Hal ini menjadi bahan pertimbangan yang penting dalam menilai ulang dari proses *refactoring* itu sendiri bahwa perbaikan kualitas sistem tidak selalu tercermin dalam perbaikan angka semata. Namun demikian, hasil analisis menunjukkan bahwa tidak semua indikator metrik mengalami perbaikan kuantitatif secara langsung. Hal ini



menandakan bahwa *refactoring* memiliki nilai strategis jangka panjang, yang tidak sepenuhnya tercermin dari metrik statik seperti jumlah baris kode atau persentase duplikasi.

#### 4.2.2.3.1. Fitur Permohonan

*Refactoring* pada fitur Permohonan dilakukan dengan memecah logika dari dua *file* utama (*permohonan.dart* dan *buat-permohonan.dart*) menjadi tiga *file* yang memiliki tanggung jawab lebih spesifik: *permohonan.dart* untuk daftar, *buat-permohonan-baru.dart* untuk form permohonan, dan *detail-permohonan.dart* untuk detail pengajuan.



*Refactoring* pada fitur ini melibatkan pemisahan modul menjadi tiga *file* yang berbeda, sesuai prinsip *Single Responsibility*. Penambahan duplikasi dan penurunan *maintainability index* terjadi karena kode yang sebelumnya tersentralisasi harus dituliskan ulang secara modular di masing-masing *file*. Meskipun secara metrik terjadi peningkatan duplikasi, dari sisi struktur dan pemeliharaan, kode menjadi lebih terbaca dan dapat diuji secara terpisah.

1. Kode sebelumnya (*permohonan.dart*, *buat-permohonan.dart*)

	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
 <i>permohonan.dart</i>	201	0	1	137	0	—	0.0%
 <i>buat-permohonan.dart</i>	457	0	0	0	0	—	0.0%

**Gambar 4. 1 Implementasi SonarQube pada fitur permohonan sebelum refactoring**

2. Kode Setelahnnya (*permohonan.dart*, *buat-permohonan-baru.dart*, *detail permohonan.dart*)

	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
 <i>permohonan.dart</i>	203	0	6	119	0	—	25.5%
 <i>buat-permohonan-baru.dart</i>	469	0	0	0	0	—	0.0%
 <i>detail_permohonan.dart</i>	144	0	2	80	0	—	0.0%

**Gambar 4. 2 Implementasi SonarQube pada fitur permohonan sesudah refactoring**

#### 4.2.2.3.2. Fitur Tagihan

Untuk fitur Tagihan, *refactoring* dilakukan tanpa perubahan jumlah *file*, namun fokus pada penataan ulang struktur kode di dalam *file* tagihan-baru.dart, detail-sewa.dart, dan pembayaran.dart. Beberapa fungsi validasi dan alur pembayaran yang sebelumnya bercampur dengan tampilan, kini dipisahkan menjadi fungsi terstruktur.

*Refactoring* pada fitur Tagihan difokuskan pada pemisahan logika visual, validasi pembayaran, serta penyusunan ulang struktur kontrol. Peningkatan duplikasi yang signifikan terjadi karena pola logika pembayaran yang mirip perlu digunakan di beberapa tempat (misalnya verifikasi status sewa). Namun, secara struktural, pemisahan ini membuat fungsi-fungsi menjadi lebih dapat diuji dan dikelola secara terpisah.

1. Kode sebelumnya (tagihan-baru.dart, detail-sewa.dart, pembayaran.dart)



	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
tagihan-baru.dart	195	0	0	0	0	—	41.9%
detail-sewa.dart	305	0	0	0	0	—	11.9%
pembayaran.dart	207	0	0	178	0	—	0.0%

Gambar 4. 3 Implementasi SonarQube pada fitur tagihan sebelum refactoring

2. Kode setelahnya (tagihan-baru.dart, detail-sewa.dart, pembayaran.dart)



	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
tagihan-baru.dart	225	0	0	0	0	—	89.5%
detail-sewa.dart	478	0	0	0	0	—	98.5%
pembayaran.dart	353	0	4	182	0	—	0.0%

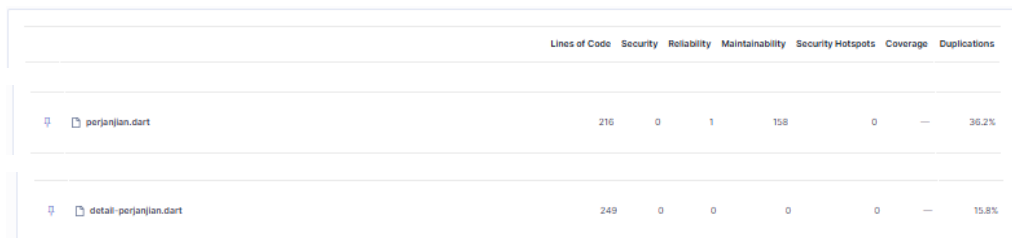
Gambar 4. 4 Implementasi SonarQube pada fitur tagihan sesudah refactoring

#### 4.2.2.3.3. Fitur perjanjian

*Refactoring* dilakukan pada dua *file* yaitu *perjanjian.dart* dan *detail-perjanjian.dart*. Beberapa fungsi digabung atau dipecah ulang untuk mengurangi kedalaman logika dan meningkatkan keterbacaan.

Hasil ini menunjukkan perbedaan yang kontras. Pada *perjanjian.dart*, duplikasi meningkat karena logika dipecah menjadi beberapa blok eksplisit. Sebaliknya, pada *detail-perjanjian.dart*, *refactoring* berhasil menghilangkan seluruh duplikasi yang ada. Penurunan *maintainability* pada *file* utama tidak sepenuhnya negatif, karena justru membuat fungsionalitas lebih terfokus

1. Kode sebelumnya (*perjanjian.dart*, *detail-perjanjian.dart*)

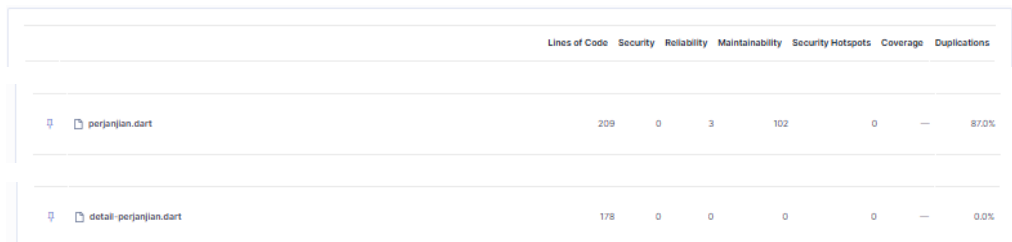


This screenshot shows the SonarQube interface for two Dart files before refactoring. The table lists metrics for 'perjanjian.dart' and 'detail-perjanjian.dart'.

	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
perjanjian.dart	216	0	1	158	0	—	36.2%
detail-perjanjian.dart	249	0	0	0	0	—	15.8%

**Gambar 4. 5 Implementasi SonarQube pada fitur perjanjian sebelum refactoring**

2. Kode setelahnya (*perjanjian.dart*, *detail-perjanjian.dart*)



This screenshot shows the SonarQube interface for the same two Dart files after refactoring. The table lists metrics for 'perjanjian.dart' and 'detail-perjanjian.dart'.

	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
perjanjian.dart	209	0	3	102	0	—	87.0%
detail-perjanjian.dart	178	0	0	0	0	—	0.0%

**Gambar 4. 6 Implementasi SonarQube pada fitur perjanjian sesudah refactoring**

Untuk melihat gambaran secara umum, berikut adalah gambar rekapitulasi perubahan metrik utama dari seluruh sistem:

Reliability ?	
Overview	
Overall Code	
Issues	11
Rating	D
Remediation Effort	55min

**Gambar 4. 7 Reliability sebelum refactoring**

Reliability ?	
Overview	
Overall Code	
Issues	73
Rating	D
Remediation Effort	6h 5min

**Gambar 4. 8 Reliability setelah refactoring**

Maintainability ?	
Overview	
Overall Code	
Issues	1,746
Debt	5d 7h
Debt Ratio	2.4%
Rating	A
Effort to Reach A	0

**Gambar 4. 9 Maintainability sebelum refactoring**

Maintainability ?	
Overview	
Overall Code	
Issues	3,510
Debt	12d
Debt Ratio	2.1%
Rating	A
Effort to Reach A	0

**Gambar 4. 10 Maintainability sesudah refactoring**



Duplications	
Overview	
Overall Code	
Density	18.2%
Duplicated Lines	1,154
Duplicated Blocks	27
Duplicated Files	9

**Gambar 4. 11 Duplications sebelum refactoring**

Duplications	
Overview	
Overall Code	
Density	51.9%
Duplicated Lines	7,948
Duplicated Blocks	22
Duplicated Files	20

**Gambar 4. 12 Duplications sesudah refactoring**

Size	
Lines of Code	
Lines	6,358
Files	24
Comment Lines	482
Comments (%)	10.9%

**Gambar 4. 13 Size sebelum refactoring**

Size	
Lines of Code	
Lines	15,304
Files	40
Comment Lines	1,271
Comments (%)	11.6%

**Gambar 4. 14 Size sesudah refactoring**

Complexity ?	▼
Cyclomatic Complexity	120
Issues	▼
Overall Code	
Open Issues	1,757
Confirmed Issues	0
Accepted Issues	0
False Positive Issues	0

**Gambar 4. 15 Complexity sebelum refactoring**

Complexity ?	▼
Cyclomatic Complexity	243
Issues	▼
Overall Code	
Open Issues	3,583
Confirmed Issues	0
Accepted Issues	0
False Positive Issues	0

**Gambar 4. 16 Complexity sesudah refactoring**

Melalui hasil ini, menjadi jelas bahwa *refactoring* tidak selalu identik dengan perbaikan angka. Pada kondisi tertentu, refactoring justru menambah duplikasi atau kompleksitas secara metrik, namun mengurangi beban kognitif pengembang karena struktur kode yang lebih jelas dan terpisah. Oleh sebab itu, keberhasilan *refactoring* dalam penelitian ini tidak hanya dinilai dari hasil analisis otomatis seperti *SonarQube*, melainkan juga dari perubahan positif dalam struktur, keterbacaan, modularitas, dan kesiapan untuk pengembangan lebih lanjut. *Refactoring* dianggap berhasil apabila kode menjadi lebih mudah dipahami, diuji, dan dimodifikasi, meskipun konsekuensinya adalah meningkatnya baris kode atau kemunculan pola yang mirip antar *file*.

#### **4.2.2.4. Data Restructuring**

Tahapan data *restructuring* dilakukan setelah proses analisis sistem lama untuk memastikan bahwa basis data yang digunakan mampu mendukung sistem yang telah diperbarui secara fungsional dan visual. Input utama pada tahap ini adalah struktur *database* dari aplikasi Tapatupa versi awal yang terdiri dari sejumlah tabel dengan penamaan dan relasi yang tidak terdokumentasi dengan baik, serta masih

terdapat ketidakkonsistenan dalam penamaan atribut dan relasi yang mengakibatkan redundansi pada data data.

Proses restrukturisasi data dilaksanakan melalui dua pendekatan utama. Pertama, dilakukan pembuatan ulang diagram basis data yang mencakup *Entity Relationship Diagram* (ERD) sebagai dasar, kemudian dikembangkan menjadi *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM). Diagram ini dibuat menggunakan notasi standar untuk memudahkan pengembangan selanjutnya. Kedua, dilakukan penyesuaian struktur fisik basis data, meliputi penyamaan penamaan kolom dan tabel menggunakan konvensi *snake\_case*, penggunaan huruf kecil secara konsisten, dan penambahan relasi yang sebelumnya belum ada antar entitas untuk mendukung alur bisnis yang telah didesain ulang.

Restrukturisasi mencakup beberapa aspek penting. Pada sisi penamaan, tabel dan atribut yang sebelumnya sebagian besar dalam Bahasa Indonesia diubah menggunakan kombinasi Bahasa Inggris untuk istilah teknis dan administratif guna meningkatkan interoperabilitas dan memperjelas dokumentasi. Dalam hal struktur wilayah, tabel-tabel seperti *provinces*, *cities*, *districts*, dan *subdistricts* tetap dipertahankan karena telah sesuai dengan hierarki wilayah nasional. Pada bagian wajib retribusi, dilakukan normalisasi dengan memisahkan data pekerjaan, jenis retribusi, dan atribut personal ke dalam tabel tersendiri seperti jenis wajib retribusi dan pekerjaan guna menghindari redundansi. Untuk objek retribusi, struktur data diperluas dengan penambahan atribut seperti foto objek, denah tanah, dan koordinat GPS (*latitude*, *longitude*). Selanjutnya, pada proses bisnis utama seperti permohonan sewa, perjanjian, tagihan, dan pembayaran, dilakukan perancangan ulang skema tabel dengan menambahkan atribut pelacakan seperti *createdAt*, *updatedAt*, dan *isDeleted* untuk mendukung mekanisme *soft delete*. *Foreign key* diperjelas untuk memastikan keterhubungan antar entitas dapat ditelusuri dengan akurat.

Struktur organisasi internal turut disesuaikan dengan cara menggabungkan tabel jabatan dan bidang menjadi jabatan bidang, sehingga alur penugasan dan persetujuan lebih mudah dikelola melalui hubungan yang eksplisit ke entitas pegawai. Selain itu, beberapa tabel seperti *approval by*, jenis *approval*, dan

penugasan *approval* diperinci dengan atribut tambahan yang relevan dengan proses bisnis persetujuan. Untuk aspek transaksi, dilakukan pemisahan dan penguatan struktur pada tabel tagihan sewa, pembayaran, dan pembayaran sewa detail, serta ditambahkan *field* untuk mencatat waktu pembayaran dan status keterlambatan. Entitas lain seperti jenis dokumen, kelengkapan dokumen, dan jenis jangka waktu juga disempurnakan melalui normalisasi dan penambahan atribut seperti *default* dan relasi *foreign key* ke entitas utama. Output dari tahap ini adalah sebuah struktur basis data baru yang terdokumentasi lengkap, menggunakan notasi standar, serta siap digunakan dalam pengembangan sistem digital pengelolaan retribusi tanah milik pemerintah. Struktur ini mendukung alur kerja dari permohonan hingga pelunasan pembayaran, memastikan sistem dapat beroperasi dengan mudah.

**Tabel 4. 3 Hasil Data Restructuring**

No	Aspek	Versi Lama	Versi Baru	Bentuk Restrukturisasi
1	Penamaan Tabel dan Atribut	Menggunakan Bahasa Indonesia penuh, huruf campuran, dan format tidak seragam	Menggunakan Bahasa Inggris untuk istilah teknis, huruf kecil semua, format snake_case pada seluruh nama tabel dan kolom	Dilakukan perubahan nama pada seluruh tabel dan atribut dengan pola penamaan konsisten untuk seluruh basis data
2	Struktur Wilayah	Terdiri atas tabel, <i>provinces, cities, districts, subdistricts</i> tanpa perubahan	Tetap digunakan tanpa revisi struktur ini	Dipertahankan, karena sudah sesuai dengan struktur hierarki wilayah
3	Informasi Wajib Retribusi	Tabel wajib retribusi mencakup jenis wajib retribusi, pekerjaan, alamat, subdistrict, mencampurkan data pekerjaan, jenis	Relasi dipindahkan ke dalam tabel jenis wajib retribusi, pekerjaan, many wajib retribusi	Normalisasi data dengan membuat 3 tabel baru dan menyambungkannya dengan <i>Foreign Key</i>

No	Aspek	Versi Lama	Versi Baru	Bentuk Restrukturisasi
		retribusi, dan alamat dalam satu tabel		
4	Objek Retribusi	Tabel menyimpan nama, ukuran tanah, dan lokasi secara deskriptif	Ditambahkan atribut file foto, gambar denah tanah, <i>latitude</i> , <i>longitude</i>	Penambahan 4 atribut baru untuk mendukung visualisasi dan geolokasi aset
5	Tarif Retribusi	Menyimpan nilai tarif dan tanggal penilaian	Ditambahkan atribut file hasil penilaian, nama penilai	Penambahan 3 kolom baru untuk menyimpan dokumen penilaian, nama penilai, dan status default tarif
6	Permohonan dan Dokumen Sewa	Tidak memiliki relasi eksplisit ke status dan entitas pendukung	Struktur diperjelas dengan FK ke status, peruntukan, satuan, dan jangka waktu	Penambahan 4 <i>foreign key</i> untuk memperjelas hubungan entitas permohonan dan entitas pendukung
7	Struktur Organisasi	Terdiri atas tabel jabatan, bidang, departemen, terpisah	Dibentuk tabel jabatan bidang yang menggabungkan jabatan dan bidang	Penggabungan dua tabel menjadi satu tabel baru untuk menyederhanakan referensi dalam entitas pegawai dan approval
8	<i>Approval</i> dan Penugasan	Terdapat tabel <i>approval by</i> , penugasan approval, jenis Approval	Ditambahkan kolom untuk referensi jabatan dan status <i>approval</i>	Penambahan 2 kolom baru pada setiap tabel untuk memperjelas aturan persetujuan

No	Aspek	Versi Lama	Versi Baru	Bentuk Restrukturisasi
9	Status dan Progres	Status hanya diterapkan pada 2 entitas	Status ditambahkan ke 4 entitas, permohonan sewa, perjanjian sewa, tagihan sewa, dan pembayaran	Penambahan kolom status id dan penerapan relasi FK pada 4 entitas
10	Saksi Perjanjian Sewa	Menyimpan nama dan NIK saksi dalam 2 kolom	Struktur tetap, tidak dilakukan penambahan atribut	Tidak dilakukan restrukturisasi karena telah memenuhi kebutuhan data
11	Dokumen dan Jenis Dokumen	Terdapat tabel berisi dokumen kelengkapan, jenis dokumen	Tetap digunakan dengan FK ke permohonan sewa dan atribut file	Penambahan 3 kolom baru dan penerapan FK untuk memperjelas keterkaitan dokumen
12	Satuan dan Jenis Satuan	satuan, jenis satuan dipisah namun kadang duplikatif	Disempurnakan dengan PK	Duplikasi dihilangkan dengan validasi PK dan <i>constraint</i>
13	Transaksi dan Pembayaran	Pembayaran tidak mencatat waktu atau status keterlambatan	Tagihan sewa, pembayaran, pembayaran sewa detail disempurnakan struktur FK dan waktu pembayaran	Penambahan histori tagihan dan keterlambatan
14	<i>Soft Delete</i> dan <i>Audit Trail</i>	Tidak semua tabel memiliki pengaturan historis	Semua tabel penting memiliki <i>createAt</i> ,	Implementasi mekanisme soft delete dan audit trail

No	Aspek	Versi Lama	Versi Baru	Bentuk Restrukturisasi
			<i>updateAt, isDeleted</i>	
15	Jangka waktu dan Durasi	Jenis jangkaWaktu dan jangkawaktu Sewa sering membingungkan	Disusun ulang dan ditambahkan field isDefault, relasi FK ditata ulang	Penyederhanaan dan pelurusan jangka waktu sewa

#### 4.2.2.5. Forward Engineering

Tahap *forward engineering* merupakan tahapan implementasi sistem berdasarkan hasil rekayasa ulang yang telah dilakukan sebelumnya. Proses ini dilakukan melalui tiga iterasi menggunakan metodologi Scrum dan prinsip *User-Centered Design* (UCD). Setiap iterasi memiliki daftar *backlog* yang jelas dan dapat diukur, dengan hasil yang terdokumentasi secara sistematis. Input pada tahap ini mencakup rancangan antarmuka hasil desain akhir, diagram *user flow*, struktur basis data baru hasil data *restructuring*, serta dokumentasi fungsional yang telah direvisi.

Output dari tahap ini adalah aplikasi Tapatupa versi terbaru yang telah melewati proses implementasi tiga iterasi dengan backlog yang terselesaikan secara lengkap, serta didukung oleh dokumentasi teknis, hasil uji fungsional, dan evaluasi kepuasan pengguna yang terukur. Sistem ini telah siap untuk digunakan dalam lingkungan operasional riil oleh instansi pemerintah terkait pengelolaan retribusi tanah daerah.

**Tabel 4.4 IDE dan Tools**

Iterasi	Sprint Backlog
Iterasi 1	6. Implementasi form permohonan 7. Fitur unggah dokumen 8. Modul tagihan sewa 9. Integrasi nomor <i>virtual account</i> (VA) 10. Pengujian fungsi <i>black - box</i>
Iterasi 2	7. Modul Daftar Objek Retribusi 8. Modul Tarif Retribusi 9. Modul Perjanjian Sewa

Iterasi	Sprint Backlog
	10. Modul Autentikasi pengguna 11. Pengujian fungsi <i>black - box</i> 12. Observasi UAT yang dilakukan oleh BKAD dan bank daerah
Iterasi 3	6. Penyempurnaan UI / UX 7. Validasi dan perbaikan fungsi 8. Penyempurnaan setiap modul secara menyeluruh 9. Penyebaran kuesioner sebagai evaluasi uji kepuasan pengguna terhadap hasil <i>reengineering</i>

#### 4.2.3. Prinsip Keamanan

Keamanan adalah prioritas utama dalam pengembangan aplikasi Tapatupa. Setiap tahap pengembangan, mulai dari desain hingga implementasi, memperhatikan aspek keamanan data dan keamanan aplikasi. Penggunaan autentikasi yang kuat, enkripsi data sensitif, dan penerapan standar keamanan dalam komunikasi API menjadi bagian integral dari implementasi ini. Seluruh data yang diproses oleh aplikasi Tapatupa harus terlindungi dengan baik dari potensi ancaman atau kebocoran data.

#### 4.2.4. Lingkungan Pengembangan Terintegrasi (IDE)

Lingkungan pengembangan yang digunakan dalam implementasi aplikasi Tapatupa terdiri dari beberapa IDE dan *tools* pendukung, dengan spesifikasi sebagai berikut:

**Tabel 4. 5 IDE dan Tools**

No	Nama / IDE Tools	Fitur Utama	Alasan Memilih
1	Visual Studio Code	IntelliSense, Integrated Terminal, Debugger, Git Integration	Ringan, fleksibel, banyak ekstensi mendukung Dart dan PHP.
2	Android Studio	Emulator Android, Flutter Inspector, Profiling Tools, Advanced Debugging	Dukungan penuh Flutter, pengujian <i>mobile</i> di emulator Android.
3	Postman	API Request	Terstruktur dan mudah digunakan.



## **BAB V**

### **PRODUCT TESTING (PT)**

#### **(PENGUJIAN PRODUK)**

##### **5.1. PENDAHULUAN**

Pengujian produk merupakan tahap krusial dalam proses pengembangan untuk memastikan bahwa produk yang dihasilkan memenuhi standar kualitas, fungsionalitas, keandalan, dan kinerja yang telah ditetapkan. Proses pengujian ini dilakukan dengan mengacu pada dokumen Desain Produk (*Product Design*) dan Implementasi Produk (*Product Implementation*), yang menjadi dasar validasi terhadap spesifikasi yang telah direncanakan. Lingkup pengujian mencakup aspek perangkat keras (*hardware*) dan perangkat lunak (*software*), apabila keduanya sesuai dengan produk yang dikembangkan. Melalui pengujian ini, potensi kekurangan dapat diidentifikasi lebih awal, sehingga produk akhir mampu memberikan performa optimal sesuai harapan pengguna dan kebutuhan pasar.

##### **5.2. DESKRIPSI PENGUJIAN**

Bagian *Product Testing* menguraikan proses pengujian produk yang dilakukan untuk memastikan kualitas, fungsionalitas, dan keandalan sistem yang dikembangkan. Pengujian dilakukan berdasarkan sejumlah skenario yang merepresentasikan penggunaan aplikasi oleh pengguna akhir. Dengan melakukan pengujian ini, tim pengembang dapat mengidentifikasi dan memperbaiki potensi kesalahan atau ketidaksesuaian, sehingga produk dapat berfungsi sebagaimana mestinya dan siap digunakan sesuai tujuan pengembangan. Pengujian ini mencakup seluruh *main function* yang ada pada aplikasi Tapatupa.

###### **5.2.1. BUTIR UJI**

Pada bagian ini diuraikan sejumlah butir uji, yang dibagi kedalam beberapa pengujian terhadap iterasi yang telah dilakukan, yakni :

#### 5.2.1.1. BUTIR UJI ITERASI I

Iterasi pertama difokuskan pada pengujian fitur utama (*Minimum Viable Product*/MVP) dalam sistem Tapatupa yang mencakup tiga proses inti, yaitu permohonan sewa aset, penanganan tagihan, dan pembayaran retribusi. Ketiga fitur ini dianggap sebagai alur utama yang paling krusial dalam menunjang layanan digital penyewaan aset tanah milik pemerintah daerah. Pengujian dilakukan secara menyeluruh terhadap seluruh skenario yang berkaitan dengan proses ini, mulai dari input data oleh pengguna, validasi sistem, hingga keluaran (*output*) yang dihasilkan. Pemilihan fitur MVP sebagai prioritas dalam iterasi pertama bertujuan untuk memastikan bahwa proses utama dapat berjalan dengan lancar dan memenuhi kebutuhan pengguna. Oleh karena itu, jumlah butir uji pada iterasi ini tergolong lebih banyak dibandingkan iterasi selanjutnya. Pengujian dilakukan dengan metode *black-box testing* dan mencakup beberapa skenario seperti pembuatan permohonan baru, pengecekan status permohonan, sistem penjadwalan tagihan, hingga simulasi pembayaran melalui sistem *virtual account*.

##### 5.2.1.1.1. Test Script Butir-Uji-1

Tabel 5. 1 butir Uji 1

<b>Identifikasi</b>	BU-01
<b>No. Fungsi</b>	F-01
<b>Nama Butir Uji</b>	Pengujian mengakses menu riwayat permohonan
<b>Tujuan</b>	Memastikan pengguna dapat mengakses halaman Riwayat Permohonan
<b>Deskripsi</b>	Pengguna mengklik menu "Riwayat Permohonan" untuk melihat daftar permohonan sebelumnya
<b>Kondisi Awal</b>	1. Wajib retribusi sudah <i>install</i> aplikasi di <i>smartPhone</i> . 2. Wajib retribusi berhasil <i>login</i> ke dalam aplikasi.

Tanggal Pengujian	10 Maret 2025		
Penguji	Gabriel Situmeang  Anita Munthe  Yudi Saragih		
Skenario Pengujian			
1. Wajib retribusi membuka aplikasi yang sudah terinstall di smartPhonanya.  2. Login ke aplikasi  3. Klik menu permohonan.  4. Akses menu “Riwayat Permohonan” melalui <i>dashboard</i>			
Kriteria Evaluasi Hasil			
1. Menu dapat diakses tanpa error  2. Halaman menampilkan data riwayat permohonan			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Akun pengguna dengan minimal satu riwayat permohonan	Halaman "Riwayat Permohonan" muncul dan menampilkan daftar permohonan yang sesuai	1. Halaman berhasil diakses dan menampilkan permohonan sesuai data akun  2. Tampilan sesuai desain	[ X ] diterima  [   ] ditolak
Catatan			

#### 5.2.1.1.2. Test Script Butir-Uji-2

Tabel 5. 2 butir Uji 2

Identifikasi	BU-02
No. Fungsi	F-02
Nama Butir Uji	Pengujian Validasi Tampilan Detail Permohonan
Tujuan	Memastikan data detail permohonan ditampilkan lengkap dan benar
Deskripsi	Pengguna memilih salah satu permohonan dari daftar untuk melihat detailnya
Kondisi Awal	Pengguna berada di halaman Riwayat Permohonan dan telah login
Tanggal Pengujian	10 Maret 2025
Penguji	Gabriel Situmeang Anita Munthe Yudi Saragih
<b>Skenario Pengujian</b>  1. Pengguna <i>login</i> ke aplikasi. 2. Pengguna mengakses menu “Riwayat Permohonan”. 3. Pengguna memilih salah satu permohonan dari daftar riwayat. 4. Sistem menampilkan detail permohonan sesuai pilihan.	
<b>Kriteria Evaluasi Hasil</b>	

<p>Aplikasi menampilkan seluruh informasi detail permohonan yang dipilih, seperti:</p> <ul style="list-style-type: none"> <li>• Nomor permohonan</li> <li>• Tanggal permohonan</li> <li>• Status permohonan</li> </ul>			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<ol style="list-style-type: none"> <li>1. Klik pada salah satu data permohonan dalam daftar riwayat.</li> <li>2. Data detail tampil lengkap dan sesuai</li> </ol>	<ol style="list-style-type: none"> <li>1. Sistem berhasil menampilkan detail lengkap permohonan</li> </ol>	<ol style="list-style-type: none"> <li>1. Halaman detail muncul tanpa <i>delay</i> berlebih.</li> <li>2. Seluruh informasi tampil lengkap dan akurat.</li> <li>3. Dokumen (jika ada) dapat diklik dan terbuka.</li> </ol>	<p>[ X ] diterima</p> <p>[   ] ditolak</p>
<b>Catatan</b>			

#### 5.2.1.1.3. Test Script Butir-Uji-3

Tabel 5. 3 butir Uji 3

<b>Identifikasi</b>	BU-03
<b>No. Fungsi</b>	F-03

Nama Butir Uji	Pengisian dan Pengajuan <i>Form</i> Permohonan		
Tujuan	Memastikan bahwa pengguna dapat mengisi seluruh <i>field form</i> permohonan dengan data valid dan sistem menyimpan permohonan dengan benar.		
Deskripsi	Pengujian dilakukan untuk memverifikasi validasi form permohonan, proses unggah dokumen, dan submit permohonan hingga muncul notifikasi keberhasilan.		
Kondisi Awal	1. Pengguna telah berhasil login 2. Klik menu "Buat Permohonan" tersedia dan dapat diakses		
Tanggal Pengujian	11 Maret 2025		
Penguji	Gabriel Situmeang  Anita Munthe  Yudi Saragih		
Skenario Pengujian			
1. Pengguna login ke aplikasi 2. Akses menu "Buat Permohonan" 3. Mengisi seluruh <i>field</i> (jenis permohonan, objek retribusi, jenis peruntukan sewa, perioditas, lama sewa, satuan, catatan, tambah dokumen kelengkapan) 4. Klik tombol "Kirim Permohonan"			
Kriteria Evaluasi Hasil			
1. Sistem menyimpan data permohonan ke database 2. Menampilkan pesan notifikasi “Permohonan berhasil diajukan” 3. Data tampil pada halaman Riwayat Permohonan			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

1. Jenis Permohonan : permohonan sewa Baru 2. Objek Retribusi : Sewa tanah - 120215001 3. Perioditas : Per Tahun 4. Lama Sewa : 2 5. Satuan : tahun, Bulan, Hari 6. Catatan : - 7. Tambah dokumen kelengkapan : KTP.jpg	1. Data tampil di riwayat permohonan	1. Semua field menerima input valid 2. Dokumen berhasil diunggah dan tersimpan	[ X ] diterima  [   ] ditolak
<b>Catatan</b>			

#### 5.2.1.1.4. Test Script Butir-Uji-4

Tabel 5. 4 butir Uji 4

<b>Identifikasi</b>	BU-04
<b>No. Fungsi</b>	F-04
<b>Nama Butir Uji</b>	Pengujian <i>form</i> permohonan jika field Jenis Permohonan tidak diisi.
<b>Tujuan</b>	Memastikan sistem menolak <i>form</i> permohonan jika <i>field</i> Jenis Permohonan tidak diisi.

Deskripsi	Pengguna mencoba mengajukan permohonan tanpa mengisi Jenis Permohonan.		
Kondisi Awal	1. Pengguna sudah login 2. <i>Form</i> permohonan dapat diakses		
Tanggal Pengujian	11 Maret 2025		
Penguji	Gabriel Situmeang  Anita Munthe  Yudi Saragih		
Skenario Pengujian			
1. Pengguna login ke aplikasi 2. Akses menu "Buat Permohonan" 3. Mengisi <i>field</i> ( objek retribusi, jenis peruntukan sewa, perioditas, lama sewa, satuan, catatan, tambah dokumen kelengkapan) 4. Mengosongkan field jenis permohonan 5. Klik tombol "Kirim Permohonan"			
Kriteria Evaluasi Hasil			
Aplikasi menampilkan pesan error “mohon tambahkan satu dokumen”			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Jenis Permohonan : (kosong) 2. Objek Retribusi : Sewa tanah - 120215001	1. Permohonan tidak diajukan, muncul validasi <i>error</i> pada <i>field</i> Jenis Permohonan.	1. Validasi error muncul 2. Tombol "Kirim Permohonan" tidak memproses data	[ X ] diterima  [ ] ditolak



3. Perioditas : Per Tahun 4. Lama Sewa : 2 5. Satuan : tahun, Bulan, Hari 6. Catatan : - 7. Tambah dokumen kelengkapan : KTP.jpg		3. Tidak ada data baru masuk ke riwayat	
<b>Catatan</b>			

#### 5.2.1.1.5. Test Script Butir-Uji-5

**Tabel 5. 5 Butir Uji 5**

<b>Identifikasi</b>	BU-05
<b>No. Fungsi</b>	F-05
<b>Nama Butir Uji</b>	Pengujian <i>form</i> permohonan jika dokumen tidak diunggah.
<b>Tujuan</b>	Memastikan bahwa form permohonan gagal dikirim jika dokumen pendukung tidak dilampirkan.
<b>Deskripsi</b>	Pengguna mengisi seluruh field namun tidak mengunggah dokumen.
<b>Kondisi Awal</b>	Pengguna mengisi seluruh field namun tidak mengunggah dokumen.

Tanggal Pengujian	13 Maret 2025		
Penguji	Gabriel Situmeang  Anita Munthe  Yudi Saragih		
Skenario Pengujian			
1. Pengguna login ke aplikasi 2. Akses menu "Buat Permohonan" 3. Mengisi field ( jenis permohonan, objek retribusi, jenis peruntukan sewa, perioditas, lama sewa, satuan, catatan) 4. Mengosongkan unggah dokumen 5. Klik tombol "Kirim Permohonan"			
Kriteria Evaluasi Hasil			
Sistem menolak pengajuan dan menampilkan pesan error “mohon tambahkan satu dokumen”			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Jenis Permohonan : permohonan sewa Baru 2. Objek Retribusi : Sewa tanah - 120215001 3. Perioditas : Per Tahun 4. Lama Sewa : 2 5. Satuan : tahun, Bulan, Hari	1. Permohonan tidak berhasil diajukan tanpa dokumen.	1. Validasi error muncul 2. Tombol "Kirim Permohonan" tidak memproses data 3. Tidak ada data baru masuk ke riwayat	[ X ] diterima  [   ] ditolak

6. Catatan : - 7. Tambah dokumen kelengkapan : (kosong)			
<b>Catatan</b>			

#### 5.2.1.1.6. Test Script Butir-Uji-6

**Tabel 5. 6 Butir Uji 6**

<b>Identifikasi</b>	BU-06
<b>No. Fungsi</b>	F-06
<b>Nama Butir Uji</b>	Pengujian untuk akses halaman tagihan sewa
<b>Tujuan</b>	Memastikan bahwa pengguna yang telah login dapat mengakses halaman tagihan sewa dan melihat daftar tagihan yang tersedia.
<b>Deskripsi</b>	Pengguna akan mengakses menu “Tagihan” dari dashboard utama untuk melihat semua daftar tagihan yang aktif.
<b>Kondisi Awal</b>	<ol style="list-style-type: none"> <li>1. Pengguna telah <i>login</i> ke sistem</li> <li>2. Terdapat minimal satu data tagihan aktif untuk pengguna tersebut</li> </ol>
<b>Tanggal Pengujian</b>	13 Maret 2025

Penguji	Gabriel Situmeang		
	Anita Munthe		
	Yudi Saragih		
Skenario Pengujian			
<div>1. Login sebagai pengguna terdaftar</div> <div>2. Klik menu “Tagihan Sewa” pada <i>dashboard</i> utama</div>			
Kriteria Evaluasi Hasil			
Halaman tagihan berhasil ditampilkan dan menampilkan daftar tagihan yang sesuai milik pengguna.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Klik menu “ Tagihan Sewa”	1. Sistem berhasil menampilkan daftar tagihan milik pengguna dengan informasi ringkas ( nomor tagihan, jatuh tempo, dan nominal)	1. Tidak ada <i>error</i> atau <i>redirect</i> 2. Daftar tagihan muncul 3. Informasi ringkasan tagihan sesuai dan benar	[ X ] diterima  [   ] ditolak
Catatan			

#### 5.2.1.1.7. Test Script Butir-Uji-7

**Tabel 5. 7 Butir Uji 7**

<b>Identifikasi</b>	BU-07
<b>No. Fungsi</b>	F-07
<b>Nama Butir Uji</b>	Pengujian klik detail tagihan sewa
<b>Tujuan</b>	Memastikan aplikasi dapat menampilkan detail tagihan dengan lengkap saat pengguna memilih salah satu tagihan dari daftar.
<b>Deskripsi</b>	Menguji apakah sistem menampilkan informasi lengkap tagihan ketika pengguna klik salah satu tagihan di halaman daftar tagihan.
<b>Kondisi Awal</b>	<ol style="list-style-type: none"><li>1. Wajib retribusi berhasil login ke dalam aplikasi.</li><li>2. Wajib retribusi sudah berada di halaman daftar tagihan</li><li>3. Data tagihan tersedia</li></ol>
<b>Tanggal Pengujian</b>	13 Maret 2025
<b>Penguji</b>	Gabriel Situmeang Anita Munthe Yudi Saragih
<b>Skenario Pengujian</b>	

1. Wajib retribusi membuka aplikasi yang sudah terinstall di smartPhonanya.  2. Wajib retribusi akses menu “Tagihan Sewa”.  3. Sistem menampilkan detail tagihan.			
<b>Kriteria Evaluasi Hasil</b>			
Informasi tagihan tampil lengkap seperti : <ul style="list-style-type: none"> <li>• Nomor Perjanjian Sewa</li> <li>• Tanggal disahkan</li> <li>• Jangka Waktu</li> <li>• Pembayaran</li> <li>• Nama Wajib Retribusi</li> <li>• Objek Retribusi</li> </ul>			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
1. Klik pada salah satu data tagihan dari daftar	1. Sistem berhasil menampilkan halaman detail tagihan dengan semua informasi sesuai	1. Informasi muncul tanpa error  2. VA dapat disalin  3. Status ditampilkan (Belum Bayar, Sudah Bayar)	[ X ] diterima  [   ] ditolak
<b>Catatan</b>			

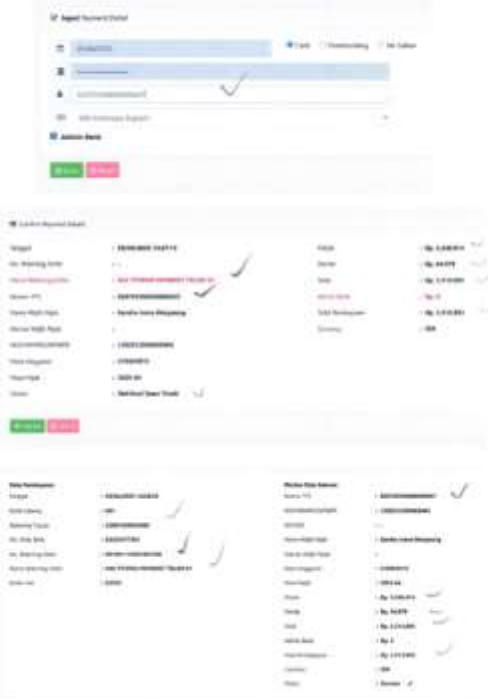
#### 5.2.1.1.8. Test Script Butir-Uji-8

Tabel 5. 8 Butir Uji 8

<b>Identifikasi</b>	BU-08
<b>No. Fungsi</b>	F-08
<b>Nama Butir Uji</b>	Pengujian melakukan pembayaran tagihan dengan VA aktif dengan <i>inquiry</i> dan Pembayaran Melalui Teller
<b>Tujuan</b>	Memastikan sistem dapat menerima dan mencatat pembayaran tagihan melalui teller bank dengan VA dan nominal valid.
<b>Deskripsi</b>	Pengguna mendatangi teller bank, menyebutkan nomor VA dan nominal, lalu teller melakukan transaksi pembayaran.
<b>Kondisi Awal</b>	<ol style="list-style-type: none"> <li>1. Status tagihan “Belum Bayar”</li> <li>2. VA valid dan aktif</li> <li>3. Nominal tagihan sudah diketahui</li> </ol>
<b>Tanggal Pengujian</b>	29 April 2025
<b>Penguji</b>	BKAD Kabupaten Tapanuli Utara, Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.
<b>Skenario Pengujian</b>	
<ol style="list-style-type: none"> <li>1. Pengguna mengakses detail tagihan</li> <li>2. Mencatat VA &amp; nominal</li> <li>3. Memberikan data ke teller</li> <li>4. Teller memproses pembayaran</li> <li>5. Sistem menerima notifikasi pembayaran</li> </ol>	

<b>Kriteria Evaluasi Hasil</b>			
1. Sistem menerima transaksi pembayaran dari bank 2. Sistem memperbarui status tagihan menjadi “Sudah Bayar” 3. Informasi transaksi tercatat dengan benar			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
1. VA : 9267 05 3 00 00 0 00 0 A 47  2. Nominal : Rp. 3.313.893	1. Notifikasi transaksi sukses  2. Status berubah otomatis  3. Histori pembayaran tercatat	1. VA dikenali di sistem bank  2. Pembayaran diterima dan dicatat	[ X ] diterima  [    ] ditolak
<b>Catatan</b>  Pada tanggal 29 April 2025 bertempat di Kantor Badan Pengelolaan Pendapatan Daerah Kabupaten Toba dengan pelaksanaannya kegiatan UAT Host to Host Integrasi Pembayaran <i>Virtual account</i> Retribusi Sewa Tanah dengan Aplikasi Tapatupa yang dilakukan bersama dengan BKAD Kabupaten Tapanuli Utaraq Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.			



Nama Halaman / Fungsi	Inquiry dan Pembayaran dengan berbagai channel
Deskripsi	Proses inquiry dan pembayaran dengan beberapa channel pembayaran
Hasil	<p><b>A. Payment Teller</b></p> <p>Pembayaran dengan kode bayar 926705300000000047 ✓</p>  <p>The screenshot displays a web application for payment processing. At the top, there's a header 'A. Payment Teller' and a note about a payment code. Below this is a form with several input fields: 'No. Rekening', 'Jumlah', 'Saldo', and 'Saldo Rekening'. There are also buttons for 'Pembayaran' and 'Pembayaran Kembali'. Below the form is a table with columns for 'No. Rekening', 'Jumlah', 'Saldo', and 'Saldo Rekening'. The table contains several rows of data, including account numbers, amounts, and balances. At the bottom, there's another table with columns for 'No. Rekening', 'Jumlah', 'Saldo', and 'Saldo Rekening', also containing data rows.</p>

#### 5.2.1.1.9. Test Script Butir-Uji-9

Tabel 5. 9 Butir Uji 9

Identifikasi	BU-09
No. Fungsi	F-09
Nama Butir Uji	Pengujian melakukan pembayaran tagihan dengan VA aktif dengan <i>inquiry</i> dan Pembayaran Melalui ATM

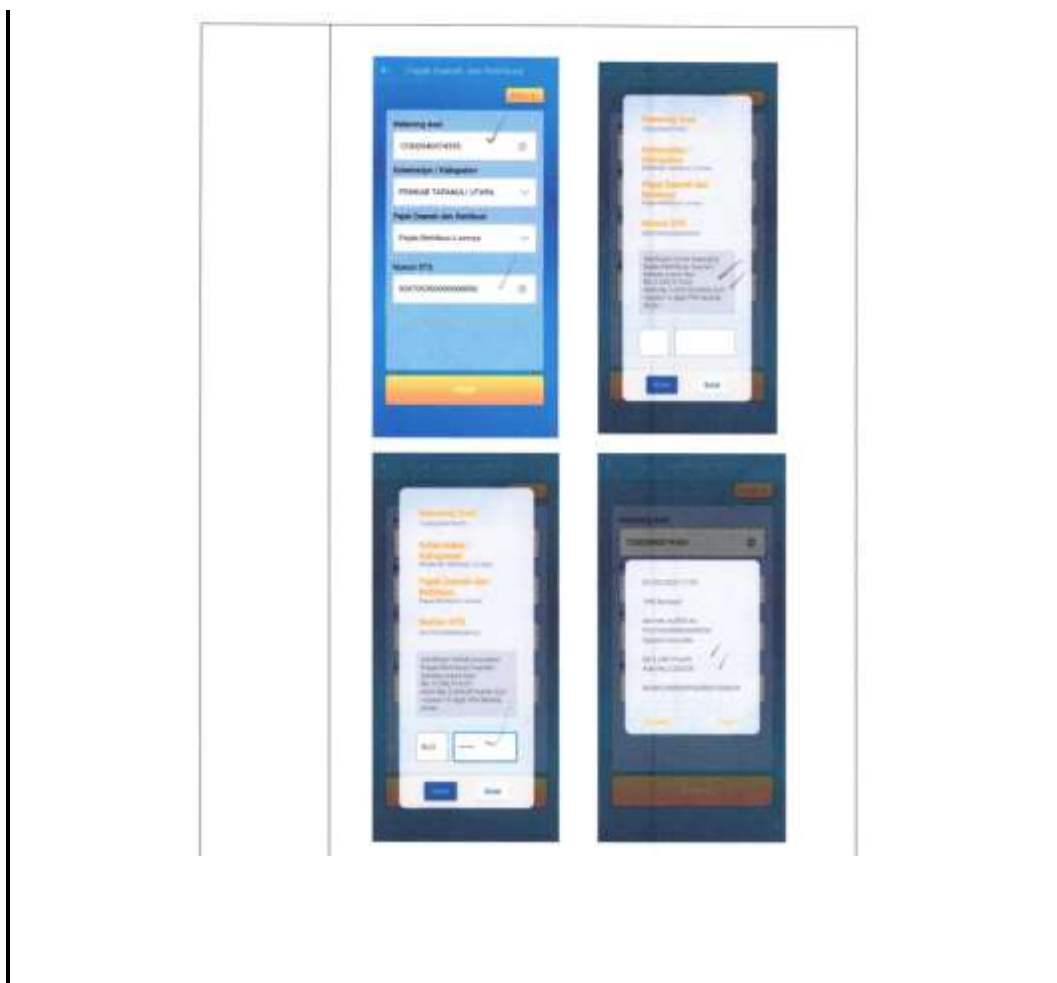
<b>Tujuan</b>	Memastikan sistem dapat mengenali pembayaran dari kanal ATM menggunakan input VA dan nominal yang sesuai.
<b>Deskripsi</b>	Pengujian ini dilakukan untuk memverifikasi apakah halaman pembayaran pada aplikasi Tapatupa dapat menampilkan informasi <i>Virtual account</i> dengan benar dan apakah Pengguna melakukan transfer tagihan melalui menu pembayaran VA di mesin ATM berhasil.
<b>Kondisi Awal</b>	<ol style="list-style-type: none"> <li>1. Pengguna memiliki kartu ATM dan saldo cukup</li> <li>2. VA aktif</li> <li>3. Status tagihan “Belum Bayar”</li> </ol>
<b>Tanggal Pengujian</b>	29 April 2025
<b>Penguji</b>	BKAD Kabupaten Tapanuli Utara, Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.
<b>Skenario Pengujian</b>	
<ol style="list-style-type: none"> <li>1. Akses ATM</li> <li>2. Pilih menu “Pembayaran” &gt; “<i>Virtual account</i>”</li> <li>3. Masukkan VA &amp; nominal</li> <li>4. Konfirmasi dan bayar</li> </ol>	
<b>Kriteria Evaluasi Hasil</b>	
<ol style="list-style-type: none"> <li>1. Pembayaran diterima sistem</li> <li>2. Status berubah menjadi “Sudah Bayar”</li> </ol>	
<b>Kasus dan Hasil Pengujian</b>	

Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. VA : 9267 05300000000048  2. Nominal : Rp. 3.250.914,00	1. Pembayaran berhasil  2. Tagihan diperbarui otomatis di sistem	1. Informasi VA tampil di ATM  2. Sistem menerima transaksi ATM  3. Status sesuai dengan aplikasi	[ X ] diterima  [   ] ditolak
<p><b>Catatan</b></p> <p>Pada tanggal 29 April 2025 bertempat di Kantor Badan Pengelolaan Pendapatan Daerah Kabupaten Toba dengan pelaksanaan kegiatan UAT Host to Host Integrasi Pembayaran <i>Virtual account</i> Retribusi Sewa Tanah dengan Aplikasi Tapatupa yang dilakukan bersama dengan BKAD Kabupaten Tapanuli Utara, Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.</p>			



<b>Nama Butir Uji</b>	Pengujian melakukan pembayaran tagihan dengan VA aktif dengan <i>inquiry</i> dan Pembayaran Melalui <i>Mobile Banking (M - Banking)</i>
<b>Tujuan</b>	Memastikan bahwa proses pembayaran tagihan melalui <i>Virtual account (VA)</i> dapat dilakukan dengan benar dan bahwa sistem dapat melakukan verifikasi pembayaran melalui <i>m-banking</i> dengan input VA & nominal berhasil dan tercatat otomatis.
<b>Deskripsi</b>	Pengguna membuka aplikasi <i>m-banking</i> dan melakukan transfer ke nomor VA.
<b>Kondisi Awal</b>	<ol style="list-style-type: none"> <li>1. Pengguna login ke m - banking</li> <li>2. Pengguna telah memiliki tagihan aktif dengan nomor <i>Virtual account</i> yang valid.</li> <li>3. Tagihan status “Belum Bayar”</li> </ol>
<b>Tanggal Pengujian</b>	29 April 2025
<b>Penguji</b>	BKAD Kabupaten Tapanuli Utara, Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.
<b>Skenario Pengujian</b>	
<ol style="list-style-type: none"> <li>1. Pengguna membuka aplikasi <i>mobile banking</i>.</li> <li>2. Pengguna memilih menu Transfer ke <i>Virtual account</i>.</li> <li>3. Pengguna memasukkan nomor <i>Virtual account</i> yang didapat dari aplikasi Tapatupa.</li> <li>4. Pengguna memasukkan nominal sesuai dengan tagihan yang tercantum.</li> </ol>	

5. Pengguna melakukan konfirmasi dan menyelesaikan transaksi.			
<b>Kriteria Evaluasi Hasil</b>			
1. Transaksi dikirim dan diterima oleh sistem tagihan 2. Status berubah menjadi “Sudah Bayar”			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
1. VA : 9267 053000000 00050  2. Nominal : Rp. 3.248.914, 00	1. Pembayaran sukses  2. Status update otomatis tanpa perlu refresh manual	1. Informasi VA terbaca di <i>m-banking</i>  2. Respon aplikasi cepat  3. Sinkronisasi status instan di sistem	[ X ] diterima  [    ] ditolak
<b>Catatan</b>  Pada tanggal 29 April 2025 bertempat di Kantor Badan Pengelolaan Pendapatan Daerah Kabupaten Toba dengan pelaksaffurn kegiatan UAT Host to Host Integrasi Pembayaran <i>Virtual account</i> Retribusi Sewa Tanah dengan Aplikasi Tapatupa yang dilakukan bersama dengan BKAD Kabupaten Tapanuli Utarq Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.			



#### 5.2.1.1.11. Test Script Butir-Uji-11

Tabel 5. 11 Butir Uji 11

<b>Identifikasi</b>	BU-11
<b>No. Fungsi</b>	F-11
<b>Nama Butir Uji</b>	Pengujian Kode Bayar tagihan tidak ditemukan / sudah dibayarkan
<b>Tujuan</b>	

	Memastikan sistem menolak transaksi pembayaran jika kode bayar (VA) tidak valid atau sudah pernah digunakan untuk transaksi sebelumnya.		
Deskripsi	Pengguna mencoba melakukan <i>inquiry</i> atau pembayaran melalui <i>mobile banking</i> menggunakan kode bayar yang tidak terdaftar atau sudah lunas.		
Kondisi Awal	<div>1. Kode bayar tidak aktif atau tagihan sudah dibayar sebelumnya.</div> <div>2. Tidak ada status tagihan “Belum Bayar” untuk kode tersebut.</div>		
Tanggal Pengujian	29 April 2025		
Penguji	BKAD Kabupaten Tapanuli Utara, Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.		
Skenario Pengujian			
<div>1. Login ke aplikasi <i>mobile banking</i>.</div> <div>2. Masukkan nomor VA tagihan yang tidak aktif/sudah dibayar.</div> <div>3. Lakukan <i>inquiry</i> / pembayaran.</div>			
Kriteria Evaluasi Hasil			
<div>1. Sistem menolak <i>inquiry</i></div> <div>2. Menampilkan pesan: “Kode bayar tidak ditemukan” atau “Tagihan sudah dibayarkan”</div>			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
<div>1. VA : 9267 053000000000 50</div>	<div>1. Sistem menolak transaksi</div>	<div>1. <i>Mobile banking</i> gagal</div>	<div>[ X ] diterima</div> <div>[   ] ditolak</div>





#### 5.2.1.1.12. Test Script Butir-Uji-12

Tabel 5. 12 Butir Uji 12

<b>Identifikasi</b>	BU-12
<b>No. Fungsi</b>	F-12
<b>Nama Butir Uji</b>	Pengujian Saldo Tidak Cukup Saat Pembayaran melalui <i>Teller</i>
<b>Tujuan</b>	Memastikan sistem menolak proses pembayaran jika rekening sumber  ( <i>cash overbooking</i> ) tidak memiliki saldo mencukupi.
<b>Deskripsi</b>	Teller melakukan pembayaran tagihan menggunakan rekening pengguna, tetapi dana tidak mencukupi nominal tagihan.
<b>Kondisi Awal</b>	<ol style="list-style-type: none"> <li>1. Pengguna memiliki tagihan “Belum Bayar”</li> <li>2. Rekening sumber pembayaran ( <i>overbooking</i>) saldonya kurang dari total tagihan</li> </ol>
<b>Tanggal Pengujian</b>	29 April 2025
<b>Penguji</b>	BKAD Kabupaten Tapanuli Utara, Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.
<b>Skenario Pengujian</b>	
<ol style="list-style-type: none"> <li>1. Pengguna datang ke bank</li> </ol>	

2. Teller input VA dan nominal sesuai 3. Teller memilih sumber dana dari rekening pengguna 4. Saldo tidak mencukupi			
<b>Kriteria Evaluasi Hasil</b>			
1. Sistem menolak transaksi di teller 2. Muncul notifikasi “ Transaksi gagal, saldo tidak mencukupi”			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
1. VA : 9267 053000000 00050 2. Nominal : Rp. 3.248.914, 00	1. Transaksi tidak diproses 2. Tidak ada perubahan status tagihan 3. Teller diberi notifikasi error oleh sistem bank	1. Tidak ada transaksi masuk ke sistem tagihan 2. Status tetap “belum Bayar” 3. Teller tidak bisa lanjut proses	[ X ] diterima [   ] ditolak
<b>Catatan</b> <p>Pada tanggal 29 April 2025 bertempat di Kantor Badan Pengelolaan Pendapatan Daerah Kabupaten Toba dengan pelaksanaannya kegiatan UAT Host to Host Integrasi Pembayaran <i>Virtual account</i> Retribusi Sewa Tanah dengan Aplikasi Tapatupa yang dilakukan bersama dengan BKAD Kabupaten Tapanuli Utaraq Institut Teknologi Del, Divisi Dana dan Jasa, Divisi Strategi dan Transformasi, dan Unit Kantor Cabang Tarutung.</p>			



#### 5.2.1.2.1. Test Script Butir-Uji-13

Tabel 5. 13 Butir Uji 13

Identifikasi	BU-013
No. Fungsi	F-13
Nama Butir Uji	Pengujian akses daftar objek retribusi untuk Wajib Retribusi.
Tujuan	Menguji halaman objek retribusi untuk wajib retribusi apakah objek retribusi menampilkan daftar objek retribusi di Tapanuli Utara.
Deskripsi	Halaman Objek Retribusi untuk wajib retribusi menampilkan objek retribusi yang ada di Tapanuli Utara.
Kondisi Awal	<ol style="list-style-type: none"><li>1. Wajib retribusi sudah install aplikasi di smartPhone.</li><li>2. Wajib retribusi berhasil <i>login</i> ke dalam aplikasi.</li></ol>
Tanggal Pengujian	25 April 2025
Penguji	Anita Munthe
Skenario Pengujian	
<ol style="list-style-type: none"><li>1. Wajib retribusi mengakses aplikasi yang sudah <i>terinstall</i> di <i>smartphon</i>enya.</li><li>2. Wajib retribusi memilih menu Objek retribusi.</li><li>3. Halaman Objek retribusi menampilkan objek retribusi yang terdaftar.</li><li>4. Wajib retribusi memilih salah satu objek retribusi untuk melihat detailnya</li></ol>	

<b>Kriteria Evaluasi Hasil</b>			
Halaman Objek retribusi untuk wajib retribusi berhasil menampilkan detail objek retribusi yang terdaftar.			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
1. (Klik) Sub Menu Objek Retribusi  2. (Klik) objek retribusi	1. User dapat melihat list daftar tagihan sewa untuk perjanjian yang aktif  2. User dapat melihat detail informasi objek retribusi yang tersedia	1. User dapat masuk ke halaman objek retribusi  2. User dapat melihat informasi objek retribusi yang tersedia.	[ X ] diterima  [   ] ditolak
<b>Catatan</b>			

#### 5.2.1.2.2. Test Script Butir-Uji-14

**Tabel 5. 14 Butir Uji 14**

<b>Identifikasi</b>	BU-14
<b>No. Fungsi</b>	F-14
<b>Nama Butir Uji</b>	Pengujian Filter Objek Retribusi dengan Kata Kunci Tidak Sesuai

Tujuan	Memastikan sistem dapat menangani input filter yang tidak cocok dengan data apapun, dan menampilkan umpan balik yang informatif.		
Deskripsi	Pengguna mengetik kata kunci “semabarangan” pada kolom pencarian / filter objek retribusi, yang tidak cocok dengan nama objek manapun dalam sistem.		
Kondisi Awal	<div> <div>1.</div> <div>Pengguna telah <i>login</i> ke dalam aplikasi.</div> </div> <div> <div>2.</div> <div>Terdapat data daftar objek retribusi dalam aplikasi</div> </div>		
Tanggal Pengujian	25 April 2025		
Penguji	Anita Munthe		
Skenario Pengujian			
<div> <div>1.</div> <div>Pengguna <i>login</i> ke aplikasi Tapatupa.</div> </div> <div> <div>2.</div> <div>Pengguna akses halaman daftar objek retribusi</div> </div> <div> <div>3.</div> <div>Ketik kata kunci “semabarangan” pada kolom filter / pencarian objek</div> </div> <div> <div>4.</div> <div>Klik pencarian</div> </div>			
Kriteria Evaluasi Hasil			
<div>Sistem tidak menampilkan data objek retribusi apapun.</div> <div>Menampilkan pesan seperti “Tidak ada data yang ditemukan”</div>			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

1. Kata kunci : (sembarang)	1. Tampilan daftar objek menjadi kosong  2. Sistem memberikan tampilan yang tidak error	1. UI menerima pencarian dengan cepat  2. Pesan “Tidak ada data yang ditemukan” muncul dengan jelas	[ X ] diterima  [   ] ditolak
<b>Catatan</b>			

#### 5.2.1.2.3. Test Script Butir-Uji-15

**Tabel 5. 15 Butir Uji 15**

<b>Identifikasi</b>	BU-15
<b>No. Fungsi</b>	F-15
<b>Nama Butir Uji</b>	Pengujian Pencarian Objek dengan Karakter tidak bermakna
<b>Tujuan</b>	Memastikan sistem dapat menangani input pencarian kosong atau  karakter tidak bermakna (karakter aneh), tanpa menyebabkan error atau menampilkan data yang salah.
<b>Deskripsi</b>	Pengguna melakukan pencarian objek sewa dengan mengetik input kosong, hanya spasi, atau karakter tidak dikenal seperti #@!%&^* di kolom pencarian.



Kondisi Awal	1. Pengguna telah berhasil <i>login</i> ke aplikasi Tapatupa.  2. Terdapat data Daftar Objek Retribusi		
Tanggal Pengujian	25 April 2025		
Penguji	Anita Munthe		
Skenario Pengujian			
1. Pengguna telah berhasil <i>login</i> ke aplikasi Tapatupa. 2. Akses halaman Daftar Objek Retribusi 3. Ketik input berikut satu per satu : Simbol (@#\$%^&*()_+) 4. Tekan enter / cari			
Kriteria Evaluasi Hasil			
1. Sistem tidak menampilkan data yang tidak sesuai 2. Sistem tidak mengalami error 3. Tampilkan pesan seperti “Tidak ada data yang ditemukan”			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Kolom pencarian : "@#\$%^"  2. Tekan cari	1. Sistem tidak mengembalikan hasil pencarian.  2. Tidak terjadi error  3. Menampilkan pesan seperti “Tidak ada data yang ditemukan”	1. UI tetap stabil  2. Pencarian tidak memaksa <i>reload</i>	[ X ] diterima  [   ] ditolak

Catatan
---------

#### 5.2.1.2.4. Test Script Butir-Uji-16

Tabel 5. 16 Butir Uji 16

<b>Identifikasi</b>	BU-16
<b>No. Fungsi</b>	F-16
<b>Nama Butir Uji</b>	Pengujian akses tarif objek retribusi
<b>Tujuan</b>	Memastikan bahwa pengguna dapat menggunakan fitur pencarian untuk menemukan tarif objek retribusi tertentu secara cepat dan akurat dari tarif yang tersedia.
<b>Deskripsi</b>	Pengujian ini bertujuan menampilkan fitur pencarian dalam tarif objek retribusi. Fitur ini mempermudah pengguna untuk mencari tarif objek tertentu tanpa harus menelusuri daftar tarif secara manual.
<b>Kondisi Awal</b>	<ol style="list-style-type: none"><li>1. Pengguna telah berhasil <i>login</i> ke aplikasi Tapatupa.</li><li>2. Terdapat data tarif objek retribusi.</li><li>3. Fitur pencarian tersedia dan dapat diakses dari halaman tarif objek retribusi.</li></ol>
<b>Tanggal Pengujian</b>	26 April 2025
<b>Penguji</b>	Anita Munthe

<b>Skenario Pengujian</b>			
<ol style="list-style-type: none"> <li>1. Pengguna <i>login</i> ke aplikasi.</li> <li>2. Pengguna masuk ke halaman tarif objek retribusi.</li> <li>3. Pengguna mengetikkan kata kunci di kolom pencarian.</li> <li>4. Sistem menyaring tarif objek berdasarkan kata kunci yang dimasukkan.</li> <li>5. Sistem menampilkan hasil pencarian yang sesuai.</li> <li>6. Pengguna dapat mengklik hasil pencarian untuk melihat tarif objek.</li> </ol>			
<b>Kriteria Evaluasi Hasil</b>			
Pengguna dapat mengakses kolom pencarian yang dapat diakses			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<ol style="list-style-type: none"> <li>1. Klik Sub Menu Tarif Objek</li> <li>2. Input Kode Tarif Objek Retribusi {120201001}</li> <li>3. Klik Button “Lihat Detail”</li> </ol>	<ol style="list-style-type: none"> <li>1. User melihat daftar tarif dari objek retribusi yang tersedia</li> <li>2. User dapat melihat tarif dari masing - masing objek retribusi bersama dengan informasi detail dari objek retribusi tersebut</li> </ol>	<ol style="list-style-type: none"> <li>1. User dapat masuk kedalam halaman Daftar Tarif Objek Retribusi</li> <li>2. User dapat melakukan pencarian Tarif dari Objek Retribusi dengan Kode atau Nama Objek Retribusi yang diinginkan</li> <li>3. User dapat melihat tarif objek retribusi tersebut bersama</li> </ol>	<p>[ X ] diterima</p> <p>[   ] ditolak</p>

		dengan melihat informasi detail dari objek retribusi yang diinginkan	
<b>Catatan</b>			

#### 5.2.1.2.5. Test Script Butir-Uji-17

**Tabel 5. 17 Butir Uji 17**

<b>Identifikasi</b>	BU-17
<b>No. Fungsi</b>	F-17
<b>Nama Butir Uji</b>	Pengujian akses perjanjian sewa untuk wajib retribusi
<b>Tujuan</b>	Memastikan bahwa pengguna Wajib Retribusi dapat mengakses detail Perjanjian Sewa setelah permohonannya disetujui.
<b>Deskripsi</b>	Pengguna yang sudah mendapatkan status “disetujui” pada permohonan dapat mengakses menu perjanjian dan melihat detail perjanjian sewa.
<b>Kondisi Awal</b>	<ol style="list-style-type: none"> <li>1. Pengguna telah berhasil <i>login</i> ke aplikasi Tapatupa.</li> <li>2. Permohonan sewa sebelumnya telah disetujui</li> <li>3. Perjanjian aktif sudah dibuat oleh petugas / admin</li> </ol>
<b>Tanggal Pengujian</b>	27 April 2025

Penguji	Anita Munthe		
Skenario Pengujian			
<div>1. Login ke aplikasi</div> <div>2. Akses menu “Perjanjian Sewa”</div> <div>3. Aplikasi menampilkan daftar perjanjian aktif</div> <div>4. Klik salah satu perjanjian untuk melihat detail</div>			
Kriteria Evaluasi Hasil			
<div>1. Sistem menampilkan daftar perjanjian</div> <div>2. Informasi detail perjanjian seperti nomor surat perjanjian, objek retribusi, dan status muncul.</div> <div>3. Pengguna memilih salah satu perjanjian sewa yang telah disetujui.</div>			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
<div>1. Klik perjanjian Sewa</div> <div>2. Perjanjian sewa terkait telah dibuat dalam database</div>	<div>1. Perjanjian dapat diakses tanpa error</div> <div>2. Informasi lengkap ditampilkan</div> <div>3. Tidak ada akses ke perjanjian milik pengguna lain</div>	<div>1. Perjanjian yang ditampilkan benar milik pengguna yang login</div> <div>2. Informasi lengkap ditampilkan</div> <div>3. Halaman tidak error</div>	<div>[ X ] diterima</div> <div>[   ] ditolak</div>

<b>Catatan</b>			

#### 5.2.1.2.6. Test Script Butir-Uji-18

**Tabel 5. 18 Butir Uji 18**

<b>Identifikasi</b>	BU-18
<b>No. Fungsi</b>	F-18
<b>Nama Butir Uji</b>	Pengujian Validasi Tampilan Detail Perjanjian
<b>Tujuan</b>	Memastikan informasi detail perjanjian ditampilkan ketika pengguna mengklik salah satu entri perjanjian
<b>Deskripsi</b>	Pengguna mengakses daftar perjanjian, lalu memilih salah satu entri untuk melihat informasi lengkap perjanjian
<b>Kondisi Awal</b>	Pengguna telah <i>login</i> dan berada di halaman daftar perjanjian
<b>Tanggal Pengujian</b>	27 April 2025

Penguji	Anita Munthe		
Skenario Pengujian			
1. Login sebagai penyewa 2. Buka halaman daftar perjanjian 3. Klik salah satu entri perjanjian 4. Sistem menampilkan halaman detail perjanjian			
Kriteria Evaluasi Hasil			
1. Halaman detail berhasil dimuat tanpa error 2. Informasi detail perjanjian seperti nomor surat perjanjian, objek retribusi, dan status) muncul dan ditampilkan. 3. Format data sesuai 4. Tidak ada informasi kosong atau salah.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Klik perjanjian Sewa  2. Perjanjian sewa terkait telah dibuat dalam database	Halaman detail perjanjian menampilkan semua informasi yang sesuai dan benar sesuai basis data	1. Detail perjanjian ditampilkan lengkap  2. Tidak ditemukan error tampilan maupun kesalahan data	[ X ] diterima  [   ] ditolak
Catatan			

#### 5.2.1.2.7. Test Script Butir-Uji-19

Tabel 5. 19 Butir Uji 19

<b>Identifikasi</b>	BU-19
<b>No. Fungsi</b>	F-19
<b>Nama Butir Uji</b>	Pengujian akses <i>profile</i>
<b>Tujuan</b>	Memastikan pengguna dapat melihat data profil pribadi secara lengkap, seperti NIK, nomor telepon, dan alamat.
<b>Deskripsi</b>	Pengguna membuka halaman profil dan sistem menampilkan data diri seperti NIK, nomor telepon, dan alamat.
<b>Kondisi Awal</b>	Pengguna telah login dan memiliki data profil yang lengkap di sistem
<b>Tanggal Pengujian</b>	28 April 2025
<b>Penguji</b>	Anita Munthe
<b>Skenario Pengujian</b>	
<ol style="list-style-type: none"><li>1. <i>Login</i> ke aplikasi</li><li>2. Akses menu “<i>Profile</i>”</li><li>3. Aplikasi menampilkan data profil pengguna</li></ol>	
<b>Kriteria Evaluasi Hasil</b>	



1. Halaman profil tampil tanpa error 2. Data NIK, nomor telepon, dan alamat muncul sesuai			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
1. Akun pengguna : Sarba_Hutauruk 2. NIK : 12345678910003 3. No. HP : 082199865437 4. Alamat : Jl. Sisingama ngaraja No.258 Siborongb orong	Aplikasi menampilkan data profil secara legkap dan benar, mencakup NIK, nomor telepon, dan alamat	1. Halaman profile ditampilkan dengan baik 2. NIK, nomor HP, dan alamat ditampilkan dengan benar dan 3. Tidak ada kesalahan format atau tampilan	[ X ] diterima [ ] ditolak
<b>Catatan</b>			

#### 5.2.1.2.8. Test Script Butir-Uji-20

**Tabel 5. 20 Butir Uji 20**

<b>Identifikasi</b>	BU-20
<b>No. Fungsi</b>	F-20

Nama Butir Uji	Pengujian <i>login</i> Berhasil ( <i>Username</i> dan <i>Password</i> Valid)		
Tujuan	Memastikan pengguna dapat login dengan <i>username</i> dan <i>password</i> yang valid		
Deskripsi	Pengguna memasukkan kombinasi <i>username</i> dan <i>password</i> yang benar untuk mengakses aplikasi		
Kondisi Awal	Akun pengguna telah terdaftar dalam sistem dengan kredensial yang valid		
Tanggal Pengujian	28 April 2025		
Penguji	Anita Munthe		
Skenario Pengujian			
1. Akses halaman login 2. Masukkan username dan password yang benar 3. Klik tombol “ <i>Login</i> ” 4. Aplikasi mengarahkan ke <i>dashboard</i>			
Kriteria Evaluasi Hasil			
1. Tidak ada error saat <i>login</i> 2. Aplikasi menampilkan halaman <i>dashboard</i> setelah <i>login</i>			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
1. Username : user 2. Password : user1234	Pengguna berhasil masuk ke sistem dan diarahkan ke halaman utama/dashboard	1. <i>Login</i> berhasil 2. Dashboard ditampilkan	[ X ] diterima [   ] ditolak

		3. Tidak ada pesan <i>error</i> muncul	
Catatan			

#### 5.2.1.2.9. Test Script Butir-Uji-21

Tabel 5. 21 Butir Uji 21

Identifikasi	BU-21
No. Fungsi	F-21
Nama Butir Uji	Pengujian <i>login</i> Gagal ( <i>Password</i> salah )
Tujuan	Memastikan sistem menolak <i>login</i> jika <i>password</i> yang dimasukkan salah
Deskripsi	Pengguna mencoba <i>login</i> menggunakan <i>username</i> yang benar tetapi <i>password</i> salah
Kondisi Awal	Akun pengguna terdaftar dengan password yang berbeda dari yang dimasukkan
Tanggal Pengujian	28 April 2025
Penguji	Anita Munthe
Skenario Pengujian	

1. Akses halaman login 2. Masukkan <i>username</i> yang benar dan <i>password</i> yang salah 3. Klik tombol “Login”			
<b>Kriteria Evaluasi Hasil</b>			
1. Sistem menolak <i>login</i> 2. Pesan error muncul “ <i>Login failed</i> ”			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
1. <i>Username</i> : user 2. <i>Password</i> : usersalah1 23456	<i>Login</i> gagal dan muncul pesan <i>error</i> yang jelas tanpa menampilkan data sensitif	1. <i>Login</i> ditolak 2. Pesan <i>error</i> “ <i>Login failed</i> ” 3. Tetap berada di halaman <i>login</i>	[ X ] diterima [   ] ditolak
<b>Catatan</b>			

### 5.2.2. TOOLS PENGUJIAN

Dalam proses pengujian aplikasi *mobile* Tapatupa, metode yang diterapkan adalah *black box testing* secara manual. Metode ini tidak memerlukan penggunaan alat bantu otomatisasi maupun perangkat lunak khusus untuk pengujian seperti Postman, JMeter, atau emulator tertentu, karena fokus utama pengujian terletak pada validasi fungsi-fungsi aplikasi dari perspektif pengguna akhir.

*Black box testing* merupakan pendekatan pengujian perangkat lunak yang menitikberatkan pada keluaran sistem berdasarkan masukan yang diberikan, tanpa memperhatikan struktur internal perangkat lunak maupun logika pemrograman yang digunakan. Oleh karena itu, pengujian dilakukan dengan merancang sejumlah skenario uji yang mencerminkan kondisi-kondisi penggunaan aktual, kemudian mengamati apakah hasil keluaran aplikasi telah sesuai dengan spesifikasi yang telah ditentukan.

Secara teknis, pengujian dilaksanakan menggunakan perangkat *smartphone* berbasis sistem operasi Android. Perangkat ini digunakan untuk:

1. Mengakses dan menjalankan aplikasi Tapatupa secara langsung.
2. Melakukan eksplorasi dan interaksi terhadap fitur-fitur utama aplikasi, seperti login, permohonan sewa, daftar objek retribusi, pembayaran, dan perjanjian sewa.
3. Mengamati dan mencatat hasil dari setiap tindakan pengujian ke dalam butir uji sebagai dokumentasi evaluasi.

Alasan pemilihan metode manual ini adalah karena aplikasi masih berada pada tahap pengembangan berbasis prototipe, sehingga pengujian fungsional secara langsung oleh pengguna lebih efektif untuk memperoleh umpan balik awal. Selain itu, metode ini memungkinkan pengujian dilakukan dari sudut pandang pengguna akhir (*end user*), sehingga dapat memberikan gambaran yang lebih realistis terhadap keandalan dan ketergunaan aplikasi.

### 5.3. METODE PENGUJIAN

Metode pengujian yang digunakan dalam pengujian aplikasi *mobile* Tapatupa adalah pengujian fungsional dengan pendekatan *black box testing*. Pengujian ini dilakukan secara manual dan bertujuan untuk memastikan bahwa setiap fitur dan fungsi dalam aplikasi berjalan sesuai dengan spesifikasi dan kebutuhan pengguna yang telah ditentukan.

#### 5.3.1. Pengujian Fungsional

Pengujian fungsional mencakup pengujian terhadap setiap fitur inti aplikasi. Adapun cakupan fungsi yang diuji antara lain:

Tabel 5. 22 Pengujian Fungsional

Kode	Nama Fungsi	Deskripsi
SW-F-01	Akses Riwayat Permohonan	Menguji apakah penyewa dapat melihat riwayat permohonan yang pernah diajukan.

Kode	Nama Fungsi	Deskripsi
SW-F-02	Detail Permohonan	Menguji apakah tampilan detail permohonan menampilkan informasi lengkap dan sesuai.
SW-F-03	Form Permohonan	Menguji pengisian dan pengajuan form permohonan secara lengkap.
SW-F-04	Validasi Field Jenis Permohonan	Menguji sistem saat field "Jenis Permohonan" tidak diisi (wajib diisi).
SW-F-05	Validasi Dokumen Permohonan	Menguji sistem ketika dokumen tidak diunggah (wajib unggah).
SW-F-06	Akses Halaman Tagihan	Menguji apakah penyewa dapat membuka halaman informasi tagihan sewa.
SW-F-07	Detail Tagihan Sewa	Menguji tampilan detail tagihan dan ketersediaan informasi seperti VA, status, instruksi.
SW-F-08	Pembayaran melalui Teller	Menguji keberhasilan proses pembayaran dengan VA melalui Teller bank.
SW-F-09	Pembayaran melalui ATM	Menguji keberhasilan proses pembayaran dengan VA melalui mesin ATM.
SW-F-10	Pembayaran melalui <i>Mobile Banking</i>	Menguji keberhasilan proses pembayaran menggunakan VA melalui aplikasi <i>Mobile Banking</i> .
SW-F-11	Validasi Kode Bayar Tidak Ditemukan/Sudah Bayar	Menguji sistem saat pengguna memasukkan kode VA yang tidak ditemukan atau sudah dibayar.
SW-F-12	Validasi Saldo Tidak Cukup	Menguji apakah sistem dapat mendeteksi kegagalan transaksi akibat saldo rekening tidak cukup.
SW-F-13	Akses Objek Retribusi	Menguji tampilan dan

Kode	Nama Fungsi	Deskripsi
		fungsionalitas daftar objek retribusi yang dapat disewa.
SW-F-14	Filter Objek Retribusi	Menguji fungsi filter objek berdasarkan kata kunci yang tidak sesuai.
SW-F-15	Pencarian Objek dengan Karakter Tidak Bermakna	Menguji sistem ketika dilakukan pencarian dengan input karakter acak atau tidak bermakna.
SW-F-16	Akses Tarif Objek Retribusi	Menguji apakah tarif untuk setiap objek retribusi ditampilkan secara benar.
SW-F-17	Akses Perjanjian Sewa	Menguji apakah penyewa dapat mengakses perjanjian sewa yang berlaku untuk objek yang dipilih.
SW-F-18	Validasi Tampilan Perjanjian	Menguji kelengkapan dan keterbacaan informasi dalam halaman perjanjian sewa.
SW-F-19	Akses Profil Pengguna	Menguji apakah pengguna dapat mengakses dan melihat data dirinya melalui menu profil.
SW-F-20	Login Berhasil	Menguji proses login dengan kredensial yang benar.
SW-F-21	Login Gagal	Menguji penanganan login dengan password yang salah.

### 5.3.2. Pengujian Non Fungsional

Kebutuhan non-fungsional merupakan uraian kebutuhan yang mendukung agar proses proses yang terdapat pada sistem dapat berjalan dengan baik. Berikut dibawah ini akan menunjukkan kebutuhan non-fungsional sistem dapat dilihat pada Tabel 5.22.

**Tabel 5. 23 Non Functional Requirement**

<b>Kode</b>	<b>Parameter</b>	<b>Requirement</b>
SW-NF-01	<i>Availability</i>	Aplikasi yang dikembangkan tersedia dan dapat digunakan selama terhubung dengan jaringan internet.
SW-NF-02	<i>Reliability</i>	Aplikasi dapat digunakan oleh pengguna untuk mengetahui kegagalan apa yang dapat ditolerir sehingga tidak berakibat fatal.
SW-NF-03	<i>Usability</i>	Aplikasi Tapatupa memiliki tampilan antarmuka yang mudah untuk digunakan <i>user</i> .
SW-NF-04	<i>Ergonomy</i>	Aplikasi Tapatupa menggunakan bahasa yang mudah untuk dipahami.

### **5.3.3. Pengujian Integrasi Software dan Hardware**

Pengujian integrasi *software* dan *hardware* dilakukan untuk memastikan bahwa perangkat lunak aplikasi Tapatupa versi *mobile* dapat berfungsi dengan baik pada perangkat keras pengguna (*smartphone*) serta dapat memanfaatkan fitur-fitur perangkat dengan optimal. Pengujian ini penting untuk memastikan bahwa tidak terjadi kendala komunikasi antara aplikasi dan sistem operasi atau fitur perangkat keras seperti konektivitas jaringan dan penyimpanan.

Pengujian dilakukan dengan menjalankan aplikasi pada beberapa perangkat Android dengan spesifikasi berbeda untuk mengamati kompatibilitas tampilan, performa aplikasi, serta respon terhadap perintah pengguna. Selain itu, komunikasi data antara aplikasi dan server diuji melalui proses pengambilan informasi riwayat



permohonan, informasi tagihan, informasi objek retribusi, informasi tarif objek, informasi profil dan data perjanjian yang membutuhkan koneksi internet.

#### **5.3.4. Evaluasi Kepuasan Pengguna**

Evaluasi kepuasan pengguna bertujuan untuk mengukur tingkat kepuasan terhadap antarmuka pengguna dan fungsi fitur setelah dilakukan proses rekayasa ulang sistem. Populasi sasaran terdiri dari 526 pengguna aktif yang terdaftar dalam sistem pengelolaan retribusi daerah. Jumlah sampel ditentukan menggunakan rumus Slovin dengan tingkat kesalahan 10%, menghasilkan 84 responden. Jumlah ini digunakan secara konsisten untuk pengisian kuesioner, serta pengujian validitas dan reliabilitas instrumen. Evaluasi dilakukan dengan menggunakan dua indikator utama, yaitu penilaian antarmuka pengguna dan penilaian terhadap fitur sistem, yang masing-masing terdiri atas 10 pernyataan tertutup berbasis skala Likert. Data dikumpulkan melalui kuesioner secara *online* yang disebarkan kepada 84 responden. Instrumen terdiri dari 20 pernyataan tertutup, terbagi menjadi dua indikator: 10 butir untuk antarmuka pengguna dan 10 butir untuk fitur sistem. Skala penilaian menggunakan Likert 5 poin, yaitu:

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Netral

4 = Setuju

5 = Sangat Setuju

Periode pengumpulan setelah aplikasi selesai di *reengineering* dan sudah dibangun, dengan hasil seluruh data dinyatakan valid untuk dilakukan analisis dan pengujian instrumen.

##### **5.3.4.1. Uji Validitas dan Reliabilitas Instrumen**

Uji validitas dan reliabilitas dilakukan terhadap dua indikator utama dalam kuesioner, yaitu penilaian antarmuka pengguna dan penilaian fitur sistem. Seluruh pengujian menggunakan 84 responden, yang ditentukan berdasarkan rumus Slovin

dari populasi pengguna sebanyak 526 orang. Pada tingkat signifikansi  $\alpha = 0,05$ , nilai r tabel minimum adalah 0,2120.

### 1. Penilaian Antarmuka Pengguna

Indikator ini terdiri dari 10 butir pernyataan. Uji validitas menggunakan korelasi Pearson Product Moment menunjukkan bahwa seluruh item memiliki r hitung lebih besar dari 0,2120, sehingga semua item pada indikator ini dinyatakan valid. Uji reliabilitas menggunakan Cronbach's Alpha menghasilkan nilai 0,704, yang melebihi ambang batas 0,70, sehingga instrumen dalam indikator ini dinyatakan reliabel dan konsisten.

		Correlations										
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total
P1	Pearson Correlation	1	.269*	.253*	.113	.042	.220*	.289**	.176	.228*	.170	.528**
	Sig. (2-tailed)		.013	.020	.305	.703	.044	.008	.110	.037	.121	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P2	Pearson Correlation	.269*	1	.108	.181	.104	.219*	.276*	.306**	.176	.086	.522**
	Sig. (2-tailed)	.013		.330	.100	.347	.045	.011	.005	.110	.439	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P3	Pearson Correlation	.253*	.108	1	.422**	.284**	.268*	.263*	.359**	.337**	-.021	.630**
	Sig. (2-tailed)	.020	.330		.000	.009	.014	.016	.001	.002	.853	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P4	Pearson Correlation	.113	.181	.422**	1	.131	.297**	.172	.199	.263*	.102	.560**
	Sig. (2-tailed)	.305	.100	.000		.234	.006	.117	.070	.016	.354	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P5	Pearson Correlation	.042	.104	.284**	.131	1	.011	.220*	.225*	.118	-.063	.372**
	Sig. (2-tailed)	.703	.347	.009	.234		.921	.045	.040	.285	.568	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P6	Pearson Correlation	.220*	.219*	.268*	.297**	.011	1	.343**	.172	.112	.174	.550**
	Sig. (2-tailed)	.044	.045	.014	.006	.921		.001	.117	.310	.112	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P7	Pearson Correlation	.289**	.276*	.263*	.172	.220*	.343**	1	.096	.197	.095	.565**
	Sig. (2-tailed)	.008	.011	.016	.117	.045	.001		.385	.072	.392	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P8	Pearson Correlation	.176	.306**	.359**	.199	.225*	.172	.096	1	.291**	.120	.561**
	Sig. (2-tailed)	.110	.005	.001	.070	.040	.117	.385		.007	.278	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P9	Pearson Correlation	.228*	.176	.337**	.263*	.118	.112	.197	.291**	1	.164	.561**
	Sig. (2-tailed)	.037	.110	.002	.016	.285	.310	.072	.007		.137	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P10	Pearson Correlation	.170	.086	-.021	.102	-.063	.174	.095	.120	.164	1	.356**
	Sig. (2-tailed)	.121	.439	.853	.354	.568	.112	.392	.278	.137		.001
	N	84	84	84	84	84	84	84	84	84	84	84
Total	Pearson Correlation	.528**	.522**	.630**	.560**	.372**	.550**	.565**	.561**	.561**	.356**	1
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000	.001	
	N	84	84	84	84	84	84	84	84	84	84	84

\*. Correlation is significant at the 0.05 level (2-tailed).

\*\* Correlation is significant at the 0.01 level (2-tailed).

**Gambar 5. 1 Hasil Uji Validitas**

Reliability Statistics	
Cronbach's Alpha	N of Items
.704	10

**Gambar 5. 2 Hasil Uji Reliabel UI**

## 2. Penilaian Fitur Sistem

Indikator ini juga terdiri dari 10 butir pernyataan. Hasil uji validitas menunjukkan bahwa seluruh item memiliki nilai  $r$  hitung di atas 0,2120, sehingga semua pernyataan pada indikator ini juga valid. Uji reliabilitas dengan metode Cronbach's Alpha memberikan hasil sebesar 0,704, yang juga menunjukkan bahwa instrumen memiliki konsistensi internal yang baik dan dapat diandalkan.

Dengan demikian, seluruh item dalam kedua indikator kuesioner terbukti valid dan reliabel, sehingga dapat digunakan secara sah dalam evaluasi sistem hasil *reengineering* aplikasi.

		Correlations										
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total
P1	Pearson Correlation	1	.180	.150	.299**	.127	.267*	.349**	.250*	.256*	.176	.566**
	Sig. (2-tailed)		.102	.174	.006	.251	.014	.001	.022	.019	.110	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P2	Pearson Correlation	.180	1	.279*	.158	.323**	.217*	.403**	.294**	.204	.083	.595**
	Sig. (2-tailed)	.102		.010	.152	.003	.048	.000	.007	.062	.451	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P3	Pearson Correlation	.150	.279*	1	.068	.143	-.041	.016	.402**	.232*	.158	.462**
	Sig. (2-tailed)	.174	.010		.537	.194	.714	.884	.000	.034	.152	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P4	Pearson Correlation	.299**	.158	.068	1	-.041	.381**	.194	.251*	.071	-.033	.443**
	Sig. (2-tailed)	.006	.152	.537		.710	.000	.077	.022	.521	.763	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P5	Pearson Correlation	.127	.323**	.143	-.041	1	.173	.152	.116	.254*	.282**	.491**
	Sig. (2-tailed)	.251	.003	.194	.710		.115	.167	.292	.020	.009	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P6	Pearson Correlation	.267*	.217*	-.041	.381**	.173	1	.254*	.220*	.337**	.165	.564**
	Sig. (2-tailed)	.014	.048	.714	.000	.115		.020	.044	.002	.133	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P7	Pearson Correlation	.349**	.403**	.016	.194	.152	.254*	1	.055	.420**	.058	.553**
	Sig. (2-tailed)	.001	.000	.884	.077	.167	.020		.618	.000	.602	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P8	Pearson Correlation	.250*	.294**	.402**	.251*	.116	.220*	.055	1	.235*	.020	.534**
	Sig. (2-tailed)	.022	.007	.000	.022	.292	.044	.618		.031	.855	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P9	Pearson Correlation	.256*	.204	.232*	.071	.254*	.337**	.420**	.235*	1	.217*	.622**
	Sig. (2-tailed)	.019	.062	.034	.521	.020	.002	.000	.031		.047	.000
	N	84	84	84	84	84	84	84	84	84	84	84
P10	Pearson Correlation	.176	.083	.158	-.033	.282**	.165	.058	.020	.217*	1	.416**
	Sig. (2-tailed)	.110	.451	.152	.763	.009	.133	.602	.855	.047		.000
	N	84	84	84	84	84	84	84	84	84	84	84
Total	Pearson Correlation	.566**	.595**	.462**	.443**	.491**	.564**	.553**	.534**	.622**	.416**	1
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	
	N	84	84	84	84	84	84	84	84	84	84	84

\*\* . Correlation is significant at the 0.01 level (2-tailed).

\* . Correlation is significant at the 0.05 level (2-tailed).

Gambar 5. 3 Hasil Uji Validitas Fitur

Reliability Statistics	
Cronbach's Alpha	N of Items
.704	10

Gambar 5. 4 Hasil Uji Reliabel Fitur

### 5.3.4.2. Hasil Evaluasi dan Interpretasi Tingkat Kepuasan Pengguna

Evaluasi kepuasan pengguna dilakukan melalui kuesioner yang terdiri atas 20 item pernyataan, terbagi secara merata ke dalam dua indikator utama, yaitu antarmuka pengguna dan fitur sistem. Setiap indikator mencakup 10 butir pernyataan, dinilai oleh 84 responden menggunakan skala Likert 1-5. Dengan demikian, skor maksimal untuk masing-masing indikator adalah sebesar 4.200, dan total skor maksimal keseluruhan adalah 8.400. Hasil pengolahan data menunjukkan bahwa

indikator antarmuka pengguna memperoleh skor aktual sebesar 3.196 (76,1%), sedangkan indikator fitur sistem memperoleh skor sebesar 3.449 (82,1%). Jika digabungkan, total skor keseluruhan adalah 6.645, yang menunjukkan tingkat kepuasan pengguna sebesar 79,17%. Berdasarkan skala interpretasi yang digunakan, nilai tersebut berada dalam kategori layak, yang mengindikasikan bahwa sistem hasil rekayasa ulang telah mampu memenuhi ekspektasi pengguna baik dari sisi tampilan maupun fungsionalitas.

## **BAB VI**

### **PRODUCT RELEASE (PR)**

### **(PELUNCURAN PRODUK)**

#### **6.1. PENDAHULUAN**

Tahap peluncuran produk merupakan fase akhir dari proses rekayasa ulang (*reengineering*) aplikasi Tapatupa, yaitu sistem penyewaan aset milik Pemerintah Kabupaten Tapanuli Utara. Setelah melalui tahapan analisis sistem lama, perancangan ulang, dan implementasi, peluncuran ini bertujuan untuk mengenalkan versi terbaru aplikasi kepada pengguna awal serta menguji kesiapan sistem di lapangan. Tahapan ini merupakan tindak lanjut dari dokumen *Product Design* dan *Product Implementation* yang sebelumnya telah disusun untuk memastikan bahwa produk akhir memenuhi standar fungsional dan teknis yang diharapkan. Peluncuran juga menandai bahwa aplikasi telah melewati tahap pengujian internal, validasi fitur utama, serta evaluasi awal oleh *stakeholder*, dan dinyatakan siap untuk digunakan oleh masyarakat luas.

#### **6.2. DESKRIPSI**

Aplikasi Tapatupa dirancang agar dapat diakses melalui perangkat *mobile* berbasis Android. Untuk tahap awal, aplikasi ini direncanakan akan dirilis versi aplikasi Android (.apk) yang dapat didistribusikan langsung kepada pengguna guna mempermudah akses dan penggunaan bagi masyarakat di kecamatan Muara, Tarutung, dan Siborongborong.

#### **6.3. DAYA GUNA PRODUK**

Evaluasi daya guna (*usability*) pada aplikasi Tapatupa dilakukan untuk mengukur sejauh mana sistem hasil *reengineering* dapat digunakan oleh pengguna sesuai dengan kebutuhannya. Evaluasi ini dilakukan melalui dua pendekatan utama, yaitu pengujian fungsional dan penyebaran kuesioner kepada pengguna akhir setelah iterasi pengembangan selesai.

Pengujian fungsional dilakukan terhadap seluruh fitur utama aplikasi, yang mencakup proses pengajuan permohonan penyewaan aset, akses informasi tagihan, pembayaran melalui *virtual account*, informasi daftar objek retribusi, informasi

tarif objek retribusi, serta pembacaan dan persetujuan perjanjian digital. Pengujian ini dilakukan pada berbagai perangkat android dengan versi sistem operasi yang berbeda untuk memastikan kesesuaian sistem. Hasil dari pengujian menunjukkan bahwa seluruh fungsi utama dapat dijalankan dengan baik, hal ini menunjukkan bahwa aplikasi telah memenuhi aspek fungsionalitas secara menyeluruh berdasarkan skenario uji yang telah dirancang sebelumnya.

Selain pengujian fungsional, evaluasi daya guna juga dilakukan melalui penyebaran kuesioner kepada pengguna akhir sebanyak 84 responden. Kuesioner ini dirancang dengan menunjukkan bahwa tingkat kepuasan pengguna terhadap aplikasi mencapai skor rata-rata sebesar 79,17%. Persentase ini berada dalam kategori interpretasi “layak digunakan”, yang berarti bahwa mayoritas responden merasa aplikasi telah mampu memenuhi ekspektasi mereka dari sisi kenyamanan penggunaan dan pemenuhan kebutuhan informasi serta layanan.

Peningkatan daya guna yang dicapai tidak terlepas dari penerapan pendekatan *User-Centered Design* (UCD) yang digunakan selama proses *reengineering*. Dalam pendekatan ini, pengguna dilibatkan secara aktif melalui proses iteratif yang terdiri dari tiga siklus evaluasi dan perbaikan. Setiap iterasi dilakukan berdasarkan umpan balik dari pengguna yang diperoleh melalui pengamatan langsung maupun kuesioner. Dengan pendekatan tersebut, aplikasi mengalami perbaikan berkelanjutan dalam hal struktur antarmuka, alur proses, dan kejelasan informasi yang disajikan.

Berdasarkan hasil pengujian fungsional dan penilaian pengguna yang bersifat kuantitatif, dapat disimpulkan bahwa aplikasi Tapatupa hasil *reengineering* memiliki tingkat daya guna yang baik. Aplikasi ini tidak hanya berfungsi dengan stabil secara teknis, tetapi juga dinilai layak digunakan oleh pengguna akhir dalam pengelolaan permohonan sewa aset tanah milik pemerintah daerah.