

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI
PENGELOLAAN PERUMAHAN KLASTER VILLA
GADING MAYANG (SIPGAMA) KOTA JAMBI BERBASIS
WEBSITE MENGGUNAKAN METODE *OBJECT
ORIENTED ANALYSIS AND DESIGN***

S K R I P S I



CHAIRUNNISA PUTRI ANDIRA

F1E119090

**PROGRAM STUDI SISTEM INFORMASI
JURUSAN TEKNIK ELEKTRO DAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS JAMBI**

2023

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi ini benar – benar karya penulis sendiri. Sepanjang pengetahuan penulis tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Tanda tangan yang tertera dalam halaman pengesahan ini adalah asli. Jika tidak asli, penulis siap menerima sanksi sesuai dengan peraturan yang berlaku.

Jambi, Desember
Yang Menyatakan

Chairunnisa Putri Andira
F1E119090

**ANALISIS DAN PERANCANGAN SISTEM INFORMASI
PENGELOLAAN PERUMAHAN KLASTER VILLA
GADING MAYANG (SIPGAMA) KOTA JAMBI BERBASIS
WEBSITE MENGGUNAKAN METODE *OBJECT
ORIENTED ANALYSIS AND DESIGN***

S K R I P S I

Diajukan sebagai salah satu syarat untuk memperoleh
Gelar Sarjana pada Program Studi Sistem Informasi



**CHAIRUNNISA PUTRI ANDIRA
F1E119090**

**PROGRAM STUDI SISTEM INFORMASI
JURUSAN TEKNIK ELEKTRO DAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS JAMBI**

2023

PENGESAHAN

Skripsi dengan Judul **ANALISIS DAN PERANCANGAN SISTEM INFORMASI PENGELOLAAN PERUMAHAN KLASTER VILLA GADING MAYANG (SIPGAMA) KOTA JAMBI BERBASIS WEBSITE MENGGUNAKAN METODE OBJECT ORIENTED ANALYSIS AND DESIGN** yang disusun oleh CHAIRUNNISA PUTRI ANDIRA, NIM : F1E119090 telah dipertahankan di depan tim penguji pada tanggal 21 Desember 2023 dan dinyatakan lulus.

Susunan Tim Penguji :

Ketua : Dedy Setiawan, S.Kom., M.IT.
Sekretaris : Daniel Arsa, S.Kom., M.S.I
Anggota : 1. Edi Saputra, S.T., M.Sc.
2. Muhammad Razi A., S.T., MMSI.
3. Benedika Ferdian Hutabarat, S.Komp., M.Kom.

Disetujui :

Pembimbing Utama

Pembimbing Pendamping

Dedy Setiawan, S.Kom., M.IT.
NIP. 198007082005111003

Daniel Arsa, S.Kom., M.S.I
NIP. 198906292019031007

Diketahui :

Dekan
Fakultas Sains dan Teknologi

Ketua Jurusan
Teknik Elektro dan Informatika

Drs. Jefri Marzal, M.Sc., D.I.T.
NIP. 196806021993031004

Nehru, S.Si., M.T.
NIP. 197602082001121002

RINGKASAN

Perumahan atau tempat tinggal merupakan salah satu dari 5 kebutuhan pokok manusia sebagai makhluk hidup berdasarkan teori Abraham Maslow “Maslow's Hierarchy of Needs” yang harus terpenuhi dengan baik sehingga perlunya pengelolaan terkait aktivitas didalamnya yang efektif dan efisien. Perumahan Villa Gading Mayang adalah sebuah lingkungan rumah masyarakat yang terdapat di Kota Jambi dengan kisaran kurang lebih 300 KK bertempat tinggal didalamnya yang menyebabkan banyak data dan aktivitas yang perlu dikelola oleh pihak administrasi perumahannya diantaranya data warga dan rumah hunian, inventaris, pembayaran iuran, permohonan surat pengantar rt, catatan laporan keuangan dan sebagainya.

Dengan banyaknya pengelolaan data perumahan, pengelola masih menggunakan metode konvensional seperti pencatatan manual yang sering menyebabkan kesalahan pendataan akibat *human error*, kerusakan kertas penagihan dan lainnya dengan kemudian dilakukan upaya penggunaan *Microsoft Excel* dalam pendataan namun masih kurang dapat memaksimalkan proses karena masih ada proses manual yang dilakukan. Selain itu, penagihan iuran dan permohonan surat yang berjalan saat ini memakan banyak waktu yang membuat warga sering merasa terlalu menghabiskan banyak waktu dan kesulitan sehingga proses yang berjalan masih dirasa kurang efektif.

Berbagai permasalahan yang terjadi maka dibutuhkan suatu solusi yang dapat memberikan kemudahan baik bagi pengelola maupun warga perumahan dengan menghadirkan suatu sistem informasi pengelolaan perumahan, dimana perlu dilakukannya analisis dan perancangan sistem sebelum sistem diimplementasikan. Metode yang digunakan dalam melakukan analisis dan perancangan adalah metode OOAD (*Object-Oriented Analysis and Design*). Pada tahap analisis didapatkan informasi terkait kebutuhan-kebutuhan sistem terperinci yang tertuang dalam *system requirement* dengan 10 kebutuhan yang teridentifikasi, dan *Use Case Diagram* dengan 1 diagram umum dan 17 sub *Use Case* beserta skenarionya yang menjadi acuan untuk lanjutan tahap perancangan. Pada Tahap Perancangan berdasarkan metode OOD dihasilkan 25 rancangan *Activity Diagram*, *Class diagram*, 14 rancangan *Sequence Diagram*, dan hasil rancangan tampilan antarmuka (UI) yang terdiri dari *Wireframe* serta *prototype* desain sistem SIPGAMA yang menggambarkan secara menyeluruh bagaimana sistem akan bekerja dalam memenuhi kebutuhan warga dan pengelola perumahan.

Setelah didapatkan hasil analisis dan rancangan yang berkesesuaian dengan kebutuhan perumahan klaster Villa Gading Mayang Kota Jambi, tahapan terakhir yaitu dilakukan pengujian atau evaluasi terhadap hasil yang diperoleh, dimana pada tahapan ini digunakan metode *requirement traceability matrix* (RTM) *functional tracing* dan didapatkan bahwa keseluruhan hasil analisis dan perancangan sistem SIPGAMA setelah dilacak oleh tim manajer proyek dan implementator telah memenuhi seluruh kebutuhan perumahan yang teridentifikasi serta dapat dilanjutkan untuk diimplementasikan.

RIWAYAT HIDUP



Chairunnisa Putri Andira, lahir di Jambi, 04 September 2001. Penulis merupakan anak pertama dan satu-satunya dari pasangan Andi Wijaya dan Lira Yanti. Penulis merupakan alumni dari : SD Negeri 66/IV Kota Jambi tahun 2013, SMP Negeri 09 Kota Jambi tahun, SMA Negeri 03 Kota Jambi. Kemudian pada tahun 2019, penulis diterima sebagai mahasiswi di Universitas Jambi, Program Strata Satu (S1) dan tercatat sebagai mahasiswa Program Studi Sistem Informasi, Jurusan Teknik Elektro dan Informatika, Fakultas Sains dan Teknologi, Universitas Jambi, melalui jalur SBMPTN. Selama menempuh Pendidikan di jenjang S1, penulis cukup aktif baik dalam bidang akademik dan non akademik seperti organisasi, dimulai dari organisasi internal himpunan mahasiswa Sistem Informasi (HIMASI) ataupun kegiatan eksternal lainnya. Dalam menambah wawasan terjun secara langsung pada dunia pekerjaan, penulis mengikuti kegiatan magang pada Kantor Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) Universitas Jambi. Dalam bimbingan dibawah Bapak Dedy Setiawan, S.Kom., M.IT. sebagai pembimbing utama dan Bapak Daniel Arsa, S.Kom., M.S.I sebagai pembimbing pendamping penulis menyelesaikan tugas akhir dengan judul "Analisis dan Perancangan Sistem Informasi Pengelolaan Perumahan Klaster Villa Gading Mayang (SIPGAMA) Kota Jambi Berbasis *Website* Menggunakan Metode *Object Oriented Analysis And Design*".

PRAKATA

Alhamdulillah rabbil'alamini, puji dan syukur saya haturkan kepada Allah SWT., yang mana atas limpahan rahmat dan hidayahnya, penulis dapat menyelesaikan skripsi dengan judul "Analisis dan Perancangan Sistem Informasi Pengelolaan Perumahan Klaster Villa Gading Mayang (SIPGAMA) Kota Jambi Berbasis *Website* Menggunakan Metode *Object Oriented Analysis and Design*" sebagai pemenuhan salah satu persyaratan kelulusan Pendidikan strata satu (S1) Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi.

Dalam menyelesaikan skripsi ini, penulis tentunya tidak hanya menyelesaikan dengan kerja keras sendiri, tetapi juga tidak luput dari dukungan, bimbingan, serta bantuan dari berbagai pihak. Dengan itu, penulis haturkan rasa hormat dan terimakasih kepada :

1. Kedua orang tua saya, Bapak Andi Wijaya dan Ibu Lira Yanti yang selalu memberikan dukungan dan motivasi, serta do'a yang tidak pernah putus dihaturkan demi keselamatan dan kelancaran bagi saya dalam menyelesaikan tugas akhir skripsi.
2. Dekan Fakultas Sains dan Teknologi Universitas Jambi.
3. Ketua Jurusan Teknik Elektro dan Informatika Fakultas Sains dan Teknologi Universitas Jambi.
4. Ketua Program Studi Sistem Informasi Fakultas Sains dan Teknologi Universitas Jambi.
5. Bapak Dedy Setiawan, S.Kom., M.IT selaku dosen pembimbing utama dan Bapak Daniel Arsa S.Kom., MSI selaku dosen pembimbing skripsi yang sangat luar biasa dalam membantu penulis menyelesaikan skripsi
6. Bapak Alm. Ir. Indra Weni, M.Kom, Bapak Edi Saputra, S.T., M.Sc., bapak Muhammad Razi A., S.T., MMSI, dan Bapak Benedika Ferdian Hutabarat, S.Komp., M.Kom. selaku Tim Penguji Skripsi yang telah memberikan saran dan masukan untuk kesempurnaan skripsi ini.
7. Seluruh Dosen dan Staff di Program Studi Sistem Informasi Universitas Jambi atas segala ilmu yang telah diberikan selama masa studi.
8. Keluarga yang selalu memberikan dukungan dan semangat selama penulis menjalankan masa studi.
9. Selki, Hani, Farah, Ozy, Anam, Iqbal, Raldi, Hendo, Nupus, Mukti, Lesi, Dini, Aldi, Sabil, Franco, Sekar, Kak Aci, Tari, dan teman – teman pada Program Studi Sistem Informasi Angkatan 2019.

10. Seluruh sahabat, junior, dan senior, yang membantu dalam penyelesaian skripsi pada Program Studi Sistem Informasi.

Akhir kata penulis mengucapkan terimakasih kepada semua pihak yang telah membantu dan memberikan dukungan. Penulis berharap skripsi ini dapat memberikan manfaat bagi banyak orang khususnya pada bidang Sistem Informasi. Penulis menyadari skripsi ini masih jauh dari kata sempurna, oleh karena itu penulis mengharapkan kritik dan saran yang membangun dari semua pihak demi kesempurnaan skripsi ini.

Jambi, Desember 2023

Chairunnisa Putri Andira
F1E119090

DAFTAR ISI

SURAT PERNYATAAN	i
PENGESAHAN	iii
RINGKASAN	iv
RIWAYAT HIDUP	v
PRAKATA	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xv
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian.....	5
II. TINJAUAN PUSTAKA	7
2.1 Perumahan Klaster Villa Gading Mayang	7
2.2 Sistem Informasi	8
2.3 Analisis dan Perancangan Sistem	8
2.4 Metode Analisis dan Perancangan.....	9
2.5 OOAD (<i>Object-Oriented Analysis and Design</i>)	12
2.6 <i>Requirement Traceability Matrix</i> (RTM)	24
2.7 Penelitian Terdahulu	25
III. METODOLOGI PENELITIAN	29
3.1 Waktu dan Tempat Penelitian	29
3.2 Alat Penelitian	29
3.3 Metode Pengumpulan Data.....	29
3.4 Langkah Kerja Penelitian	30
3.5 Jadwal Penelitian.....	35
IV. HASIL DAN PEMBAHASAN	36
4.1 Tahap Perencanaan	36
4.2 Tahap Analisis.....	38
4.3 Tahap Perancangan.....	71
4.4 Tahap Evaluasi	134
V. KESIMPULAN DAN SARAN	141
5.1 Kesimpulan	141
5.2 Saran	141

DAFTAR PUSTAKA	143
LAMPIRAN	148

DAFTAR TABEL

Tabel	Halaman
1. Perbandingan Metode SSAD, OOAD, dan Agile	10
2. Simbol <i>Use Case Diagram</i>	16
3. Simbol <i>Activity Diagram</i>	18
4. Simbol <i>Class Diagram</i>	19
5. Simbol <i>Sequence Diagram</i>	21
6. Penelitian Terdahulu	25
7. Jadwal Penelitian	35
8. Analisis Kebutuhan Fungsional Sistem	39
9. Tabel Peran Entitas Sistem	40
10. Definisi <i>Use Case</i>	50
11. Skenario <i>Use Case Login</i>	54
12. Skenario <i>Use Case Logout</i>	55
13. Skenario <i>Use Case</i> Mengelola Data Master	56
14. Skenario <i>Use Case</i> Melihat Data Master	57
15. Skenario <i>Use Case</i> Mengelola Data Inventaris	58
16. Skenario <i>Use Case</i> Melihat Data Inventaris	59
17. Skenario <i>Use Case Input</i> Permohonan Surat	60
18. Skenario <i>Use Case</i> Mengelola Data Daftar Permohonan Surat	61
19. Skenario <i>Use Case</i> Melihat Daftar Permohonan Surat	62
20. Skenario <i>Use Case Input</i> Pembayaran Iuran	63
21. Skenario <i>Use Case</i> Mengelola Data Daftar Iuran	64
22. Skenario <i>Use Case</i> Melihat Data Daftar Iuran	65
23. Skenario <i>Use Case</i> Melihat Riwayat Pembayaran Iuran	66
24. Skenario <i>Use Case</i> Validasi Permohonan Surat	66
25. Skenario <i>Use Case</i> Validasi Pembayaran Iuran	68
26. Skenario <i>Use Case</i> Mengelola Data Laporan Keuangan	69
27. Skenario <i>Use Case</i> Melihat Laporan Keuangan	70
28. <i>Class Users</i>	98
29. <i>Class</i> Rumah	99
30. <i>Class</i> Tipe Rumah	99
31. <i>Class</i> Pembayaran Iuran	99
32. <i>Class</i> Metode Pembayaran	100
33. <i>Class</i> Permohonan Surat	100

34. <i>Class Inventaris</i>	101
35. <i>Class Pengeluaran</i>	101
36. <i>Class Pemasukan</i>	102
37. <i>Link Prototype SIPGAMA</i>	134
38. Pemetaan Kode <i>Unique Identifier (ID) Activity Diagram</i>	135
39. Pemetaan Kode <i>Unique Identifier (ID) Sequence Diagram</i>	136
40. Pemetaan Kode <i>Unique Identifier (ID) User Interface</i>	136

DAFTAR GAMBAR

Gambar	Halaman
1. Kerangka Kerja Penelitian.....	30
2. Tabel Requirement Traceability Matrix (RTM)	35
3. <i>Use Case Diagram</i> Umum	42
4. <i>Use Case Diagram</i> Kelola Data Master (Admin)	43
5. <i>Use Case Diagram</i> Kelola Data Inventaris (Admin)	43
6. <i>Use Case Diagram</i> Kelola Data Daftar Permohonan Surat (Admin)	44
7. <i>Use Case Diagram</i> Kelola Laporan Keuangan (Bendahara)	44
8. <i>Use Case Diagram</i> Kelola Data Daftar Iuran (Bendahara).....	45
9. <i>Use Case Diagram</i> Melihat Laporan Keuangan	45
10. <i>Use Case Diagram</i> Melihat Data Master (Pimpinan)	46
11. <i>Use Case Diagram</i> Melihat Data Master (Bendahara)	46
12. <i>Use Case Diagram</i> Melihat Data Inventaris (Pimpinan, Bendahara)	47
13. <i>Use Case Diagram</i> Melihat Daftar Iuran (Admin, Pimpinan)	47
14. <i>Use Case Diagram</i> Melihat Riwayat Pembayaran Iuran (Warga).....	48
15. <i>Use Case Diagram</i> Melihat Daftar Permohonan Surat (Pimpinan, Bendahara)	48
16. <i>Use Case Diagram</i> Input Pembayaran Iuran (Warga).....	49
17. <i>Use Case Diagram</i> Input Permohonan Surat (Warga)	49
18. <i>Activity Diagram</i> Login	72
19. <i>Activity Diagram</i> Logout	73
20. <i>Activity Diagram</i> Tambah Data Master	74
21. <i>Activity Diagram</i> Edit Data Master.....	75
22. <i>Activity Diagram</i> Hapus Data Master	76
23. <i>Activity Diagram</i> Melihat Data Master (Pimpinan)	77
24. <i>Activity Diagram</i> Melihat Data Master (Bendahara)	78
25. <i>Activity Diagram</i> Tambah Data Inventaris.....	79
26. <i>Activity Diagram</i> Edit Data Inventaris.....	80
27. <i>Activity Diagram</i> Hapus Data Inventaris	81
28. <i>Activity Diagram</i> Melihat Data Inventaris.....	82
29. <i>Activity Diagram</i> Input Permohonan Surat.....	83
30. <i>Activity Diagram</i> Tambah Data Daftar Permohonan Surat.....	84
31. <i>Activity Diagram</i> Edit Data Daftar Permohonan Surat.....	85
32. <i>Activity Diagram</i> Melihat Daftar Permohonan Surat	86
33. <i>Activity Diagram</i> Input Pembayaran Iuran	87

34. <i>Activity Diagram</i> Tambah Data Daftar Iuran	88
35. <i>Activity Diagram</i> Edit Data Daftar Iuran	89
36. <i>Activity Diagram</i> Melihat Riwayat Pembayaran Iuran	90
37. <i>Activity Diagram</i> Validasi Permohonan Surat	91
38. <i>Activity Diagram</i> Validasi Pembayaran Iuran	92
39. <i>Activity Diagram</i> Tambah Data Laporan Keuangan	93
40. <i>Activity Diagram</i> Edit Data Laporan Keuangan	94
41. <i>Activity Diagram</i> Hapus Data Laporan Keuangan.....	95
42. <i>Activity Diagram</i> Melihat Laporan Keuangan	96
43. <i>Class Diagram</i> SIPGAMA.....	97
44. <i>Sequence Diagram</i> Login	103
45. <i>Sequence Diagram</i> Data Master (Data Warga)	104
46. <i>Sequence Diagram</i> Data Master (Data Rumah).....	105
47. <i>Sequence Diagram</i> Data Master (Data Tipe Rumah)	106
48. <i>Sequence Diagram</i> Data Master (Metode Pembayaran)	107
49. <i>Sequence Diagram</i> Inventaris	108
50. <i>Sequence Diagram</i> Input Permohonan Surat	109
51. <i>Sequence Diagram</i> Daftar Permohonan Surat (Permohonan Surat)	110
52. <i>Sequence Diagram</i> Input Pembayaran Iuran	111
53. <i>Sequence Diagram</i> Daftar Iuran	112
54. <i>Sequence Diagram</i> Validasi Permohonan Surat.....	113
55. <i>Sequence Diagram</i> Validasi Pembayaran Iuran	114
56. <i>Sequence Diagram</i> Laporan Keuangan (Pengeluaran)	115
57. <i>Sequence Diagram</i> Laporan Keuangan (Rekapitulasi Keuangan).....	116
58. Palet Warna Desain Sistem SIPGAMA	117
59. <i>Wireframe Low-Fid Landing Page</i>	118
60. <i>Wireframe Low-Fid</i> Halaman Login.....	119
61. <i>Wireframe Low-Fid</i> Dashboard	120
62. <i>Wireframe Low-Fid</i> Data Master (Data Warga).....	120
63. <i>Wireframe Low-Fid</i> Data Master (Data Rumah)	121
64. <i>Wireframe Low-Fid</i> Data Master (Data Tipe Rumah)	122
65. <i>Wireframe Low-Fid</i> Data Master (Metode Pembayaran).....	123
66. <i>Wireframe Low-Fid</i> Profil Warga	124
67. <i>Wireframe Low-Fid</i> Halaman Data Inventaris.....	124
68. <i>Wireframe Low-Fid</i> Permohonan Surat (Warga).....	125
69. <i>Wireframe Low-Fid</i> Daftar Surat.....	125

70. <i>Wireframe Low-Fid</i> Pembayaran Iuran (Warga)	126
71. <i>Wireframe Low-Fid</i> Daftar Iuran (Admin, Pimpinan)	127
72. <i>Wireframe Low-Fid</i> Daftar Iuran (Bendahara)	127
73. <i>Wireframe Low-Fid</i> Pengeluaran (Laporan Keuangan)	128
74. <i>Wireframe Low-Fid</i> Rekapitulasi Keuangan.....	129
75. <i>Wireframe Hi-Fid Landing Page</i>	130
76. <i>Wireframe Hi-Fid Login</i>	131
77. <i>Wireframe Hi-Fid Dashboard</i>	131
78. <i>Wireframe Hi-Fid Data Warga (Data Master)</i>	132
79. <i>Wireframe Hi-Fid Profil Warga</i>	132
80. <i>Wireframe Hi-Fid Pembayaran Iuran (Warga)</i>	133
81. <i>Wireframe Hi-Fid Rekapitulasi Pengeluaran (Laporan Keuangan)</i>	133

DAFTAR LAMPIRAN

Lampiran	Halaman
<i>Wireframe High Fidelity</i> SIPGAMA.....	148
Dokumentasi Penelitian	153

I. PENDAHULUAN

1.1 Latar Belakang

Manusia sebagai makhluk hidup menurut (Putra, 2021) dalam teori Abraham Maslow "*Maslow's Hierarchy of Needs*" memiliki 5 kebutuhan yang digambarkan dalam bentuk piramida dengan kebutuhan terpenting yang berada paling dasar yaitu kebutuhan fisiologi. Kebutuhan ini berisikan kebutuhan pokok yang harus terpenuhi salah satunya yaitu tempat tinggal sebagai tempat untuk menetap dan berteduh. Seiring dengan perkembangan teknologi dan informasi pada saat ini yang semakin cepat dan pesat, hampir keseluruhan kegiatan memanfaatkan teknologi sebagai penunjang data dan informasi, dimulai dari kegiatan perkantoran, perumahan, transaksi pembayaran, pembelajaran, hingga komunikasi dan hal-hal umum lainnya.

Teknologi dalam penerapannya diharapkan dapat mempermudah dan mengefisienkan tahapan aktivitas terkait pengelolaan informasi dan data melalui suatu konsep yaitu konsep sistem informasi. Sistem informasi diterapkan dengan harapan akan hadirnya solusi dan penanganan yang tepat terhadap sebuah masalah, jika suatu aktivitas dilakukan secara manual. Berdasarkan (Maydianto & Ridho, 2021) Sistem informasi merupakan kumpulan dari berbagai komponen yang saling bekerja sama dalam melakukan pengelolaan data yaitu mengumpulkan, menyimpan, memproses, hingga menjadi sebuah informasi bermakna yang dapat membantu pengguna mencapai tujuan yang diharapkan. Pengadaan sistem informasi tentunya juga dibutuhkan pada kompleks perumahan dengan tujuan agar mengefisienkan pengelolaan data penduduk terkait persuratan hingga iuran perbulan dengan terciptanya suatu sistem informasi administrasi.

Menanggapi pernyataan diatas, maka penulis mengangkat permasalahan yang terjadi pada kompleks perumahan berlokasi di daerah Mayang, Jambi yaitu Komplek perumahan klaster Villa Gading Mayang. Perumahan klaster Villa Gading Mayang merupakan sebuah kelompok lingkungan rumah masyarakat penduduk dimana banyak pengelolaan yang perlu dilakukan terkait data aktivitas didalamnya. pengelola perumahan perlu melakukan pengolahan terkait data penduduk yang menempati hunian rumah, permintaan surat pengantar, inventaris barang yang dimiliki perumahan, hingga iuran bulanan yang wajib dibayarkan oleh warga setiap sebulan sekali dengan jumlah KK (Kartu Keluarga) hunian lingkungan perumahan berkisar kurang lebih 300 keluarga serta pelaporannya dengan tepat sehingga tercapainya kenyamanan dan efisiensi.

Dengan banyaknya data yang perlu dikelola diantaranya seperti data hunian rumah yang ditempati, pembayaran iuran, data permohonan surat pengantar, data catatan pengeluaran dan pemasukan keuangan, data inventaris barang dan sebagainya namun masih dilakukan secara manual membuat banyak timbulnya berbagai masalah yang mempersulit pengelola seperti tidak adanya data rumah yang masih dihuni warga atau tidak menyebabkan kekacauan penagihan dan pencatatan keuangan iuran. Selain itu, Ketua RT sering mengalami kesulitan pencarian data terkait warga dalam permintaan surat pengantar sehingga memperlambat responsif administrasi internal keluarnya surat dalam perumahan. Kemudian tidak tersedia data terkait inventaris barang beserta status barang membuat lambannya tanggap pengelola terkait barang perumahan yang tidak layak digunakan lagi membuat terganggunya kenyamanan penduduk perumahan. Selanjutnya, pendataan pembayaran iuran bulanan yang masih dilakukan hanya dengan memanfaatkan kertas nota bukti pembayaran sering menyebabkan kebingungan, hilang serta tercecernya kertas. Keadaan iuran yang berjalan secara manual ini juga menyebabkan tidak adanya laporan transparansi dana yang seharusnya diberikan oleh bendahara sebagai pengelola keuangan kepada warga perumahan sehingga banyak warga yang enggan membayar iuran secara rutin. Pengelola telah melakukan upaya penerapan penggunaan teknologi *Microsoft Excel* namun tidak memberikan kemudahan dan solusi yang diharapkan dapat diberikan.

Oleh karena itu, diperlukannya penerapan suatu sistem informasi pengelolaan dan administrasi pada perumahan ini untuk membantu pengelola dalam pengolahan seluruh data terkait aktivitas di dalam perumahan tersebut. Sistem ini akan menghimpun dan mengolah seluruh informasi terkait seperti data hunian penduduk perumahan, data permohonan surat pengantar, informasi pemasukan dan pengeluaran keuangan, inventaris barang kompleks perumahan, hingga pembayaran iuran bulanan yang dilakukan oleh penduduk perumahan. Kehadiran sistem informasi pengelolaan hunian perumahan pada perumahan Villa Gading Mayang ini membuat pengelolaan data dan informasi dapat berjalan lebih mudah, efisien, efektif dan terotomatisasi.

Mewujudkan hal ini maka perlu dilakukan analisis dan perancangan sistem informasi pengelolaan perumahan klaster yang akan memberikan gambaran lebih jelas dan detail saat akan diimplementasikan. Analisis dan perancangan sistem berdasarkan (B. Shelly & J. Rosenblatt, 2012) merupakan proses yang dilakukan tahap per tahap sehingga didapatkan sebuah sistem berkualitas tinggi yang mampu mendukung pengambilan keputusan manajerial dan mempermudah rangkaian aktivitas bisnis yang perlu dilalui. Tahapan proses

dimulai dengan melakukan analisis proses bisnis aliran data secara sistematis, memproses kemudian menyimpan data dengan hasil keluaran berupa informasi yang dapat menjadi tolak ukur keputusan bisnis. Sistem yang dihasilkan melalui tahapan analisis dan perancangan tentunya diharapkan dapat menjadi solusi dari permasalahan yang dihadapi oleh pihak pengelola.

Terdapat beberapa metode yang dapat digunakan dalam melakukan Analisis dan perancangan sistem informasi diantaranya berdasarkan (B. Shelly & J. Rosenblatt, 2012) terdapat 3 yaitu SSAD (*Structured System Analysis and Design*), OOAD (*Object-Oriented Analysis and Design*), dan *Agile/Adaptive Methods*. Pada penelitian ini, penulis menggunakan metode OOAD (*Object-Oriented Analysis and Design*). Metode ini digunakan dengan alasan utama bahwa metode ini akan terintegrasi dengan program pengembangan sistem berorientasi objek, dimana implementasi sistem perumahan menggunakan pemrograman tersebut yaitu OOP (*Object Oriented Programming*). Metode OOAD ini mempermudah dan menghemat pengembangan sistem berkelanjutan karena kodenya bersifat modular sehingga dapat digunakan kembali (*reusable*). Kelebihan metode OOAD jika dibandingkan dengan metode analisis dan perancangan lainnya adalah model pengembangannya yang cenderung lebih interaktif dengan perencanaan, analisis, dan pembuatan rancangan berinteraksi secara berkelanjutan hingga menghasilkan *prototype* yang dapat diuji untuk selanjutnya diimplementasikan secara mudah dengan penggambaran proses bisnis sesuai dengan keadaan nyata.

Metode OOAD menurut (B. Shelly & J. Rosenblatt, 2012) merupakan suatu metode yang memandang sistem dalam bentuk objek yang menggabungkan data dan proses, dimana objeknya merepresentasikan orang, benda, transaksi, dan peristiwa yang sebenarnya terjadi. Fase tahapan pada metode ini lebih interaktif jika dibandingkan dengan metode SSAD. Kemudian menurut (S. Valacich & F. George, 2017) metode OOAD adalah metode analisis dan perancangan yang populer digunakan dengan menggabungkan data dan proses sehingga menjadi satu entitas tunggal disebut sebagai objek yang bertujuan agar elemen-elemen dalam sistem dapat digunakan kembali atau *reusable* yang mampu meningkatkan kualitas sistem yang dikembangkan. Selain itu, menurut (Pane & Sarno, 2015) metode OOAD adalah metode yang sangat ampuh untuk menganalisa dan menangkap kebutuhan pengguna dalam perancangan sebuah sistem informasi dimana pada metode ini dilakukan dengan penggambaran artififikasi visual dari siklus perancangan sistem menggunakan UML umumnya *Use Case* dalam penggambaran visualnya.

Rancangan dari hasil analisis yang telah dilakukan dengan menggunakan diagram UML tentunya perlu dilakukan pengujian sehingga dapat dipastikan terpenuhinya segala aspek kebutuhan sistem maupun pengguna sistem. Pada penelitian ini evaluasi pengujian dilakukan dengan metode RTM (*Requirement Traceability Matrix*) dimana berdasarkan (S. Pressman, 2000) dikatakan bahwa pengujian dengan metode RTM diawali dengan tahap identifikasi kebutuhan, dan jika sudah teridentifikasi selanjutnya dimasukkan pada tabel *traceability*. Evaluasi awal dalam tabel dengan menghubungkan setiap elemen sistem kepada *requirements* (persyaratan) yang sesuai dalam tabel dilanjutkan dengan menghubungkan setiap persyaratan sistem ke elemen sistem yang sesuai dengan tabel kemudian dievaluasi hubungan tersebut untuk memastikan semua persyaratan terpenuhi dan tidak ada elemen yang tidak diperlukan.

Dari penjabaran latar belakang diatas penulis mengangkat topik penelitian berjudul **“Analisis dan Perancangan Sistem Informasi Pengelolaan Perumahan Klaster Villa Gading Mayang (SIPGAMA) Berbasis Website Menggunakan Metode *Object Oriented Analysis And Design*”**. Diharapkan dengan adanya penelitian ini menghasilkan suatu rancangan sistem informasi yang dapat menjadi solusi permasalahan baik dari sisi pengelola perumahan maupun warga penduduk, serta mempermudah pengembang sistem dalam mengimplementasikan sistem pada perumahan klaster Villa Gading Mayang.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, dirumuskan permasalahan dari penelitian ini adalah sebagai berikut:

1. Bagaimana menganalisis dan merancang suatu sistem informasi pengelolaan di perumahan klaster Villa Gading Mayang berbasis website dengan menggunakan metode OOAD (*Object Oriented Analysis and Design*) yang selanjutnya dapat diimplementasikan kedalam proses tahapan pengembangan sistem.
2. Bagaimana menguji hasil analisis dan perancangan sistem yang telah dibuat sesuai dengan kebutuhan menggunakan metode evaluasi RTM (*Requirement Traceability Matrix*).

1.3 Tujuan Penelitian

Adapun tujuan yang diharapkan tercapai dalam penelitian ini adalah sebagai berikut:

1. Melakukan analisis dan membangun rancangan sistem informasi pengelolaan lingkungan perumahan di perumahan klaster Villa Gading Mayang berbasis website dengan menggunakan metode OOAD (*Object*

Oriented Analysis and Design) yang selanjutnya dapat diimplementasikan kedalam proses tahapan pengembangan sistem.

2. Melakukan pengujian terhadap hasil analisis dan perancangan sistem yang telah dibuat sesuai dengan kebutuhan menggunakan metode evaluasi RTM (*Requirement Traceability Matrix*).

1.4 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut :

- a. Output penelitian dari analisis dan perancangan sistem berupa diagram UML dan desain *prototype user interface* (UI)
- b. Desain *prototype user interface* (UI) dibuat hanya sebagai pelengkap rancangan sistem informasi pengelolaan lingkungan perumahan
- c. Sistem Informasi pengelolaan yang dianalisis dan dirancang hanya untuk mengelola data terkait lingkungan perumahan diantaranya meliputi data warga, data rumah yang ditempati, data permohonan surat pengantar, data inventaris barang, data kelola keuangan beserta pengeluaran, dan data pembayaran iuran bulanan
- d. Pengguna sistem informasi pengelolaan terdiri dari 4 yaitu ketua RT, sekretaris sebagai administrator, bendahara, dan warga perumahan klaster Villa Gading Mayang
- e. Permohonan surat yang dibuat hanya melingkupi surat pengantar yang diperlukan internal dalam ruang lingkup rt perumahan klaster Villa Gading Mayang, tidak membahas terkait surat yang terjadi diluar ruang lingkup perumahan
- f. Kelola keuangan yang dibuat hanya melingkupi keuangan yang ada di lingkungan perumahan klaster yang ada di perumahan Villa Gading Mayang diantaranya dana kas, dana keamanan, dana kebersihan, dan dana sosial, tidak membahas sistem keuangan yang terjadi di luar lingkup perumahan klaster Villa Gading Mayang.

1.5 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberi manfaat baik dari segi praktis maupun teoritis. Manfaat tersebut diantaranya adalah sebagai berikut:

- a. Memperoleh hasil analisis terkait apa saja kebutuhan yang perlu difasilitasi oleh perumahan dalam pengembangan sistem informasi pengelolaan perumahan klaster sehingga terciptanya sistem yang efisien dan bermanfaat
- b. Menghadirkan solusi yang tepat terkait permasalahan yang dihadapi dalam pengelolaan perumahan dengan bentuk sebuah rancangan sistem informasi pengelolaan klaster perumahan

- c. Menjadi acuan untuk pengembangan sistem administrasi perumahan Villa Gading Mayang berbasis web selanjutnya.
- d. Menambah ilmu sekaligus mendapatkan analisis dan perancangan sistem informasi berbasis web dengan menggunakan metode Object-oriented Analysis and Design (OOAD).

II. TINJAUAN PUSTAKA

2.1 Perumahan Klaster Villa Gading Mayang

Perumahan menurut (Sunarti, 2019) secara umum merupakan kumpulan atau gabungan dari berapa rumah baik yang dikembangkan pemerintah maupun swasta berfungsi sebagai tempat tinggal atau lingkungan hunian. Menurut (Pratama & Purwidayanta, 2018) perumahan adalah tempat hunian yang berfungsi sebagai lingkungan tempat tinggal dilengkapi dengan sarana dan prasarana dasar fisik lingkungan seperti penyediaan air bersih, pembuangan sampah, keamanan, terdapatnya listrik, telepon, serta jalan yang memungkinkan lingkungan hunian pemukiman berfungsi sebagaimana mestinya. Perumahan ini akan terus meningkat dan berlanjut seiring dengan penambahan penduduk dan perkembangan zaman sehingga dalam pengelolaannya memerlukan pemanfaatan teknologi agar berjalan lebih mudah dan efisien.

Villa Gading Mayang merupakan salah satu perumahan klaster di Provinsi Jambi yang terletak di Mayang Mangurai, Kotabaru dengan letak lokasi strategis berdekatan dengan pasar, rumah sakit, pusat perbelanjaan. Villa Gading Mayang sama seperti perumahan klaster pada umumnya, tentu menyediakan berbagai fasilitas yang dibutuhkan penduduknya dimulai dari air bersih, keamanan, lingkungan terjaga serta berbagai fasilitas lainnya. Perumahan ini dalam pengelolaannya dilakukan oleh organisasi rukun tetangga yang dikepalai oleh ketua RT. Pengelola memiliki tugas yang hampir sama seperti pengelola tempat tinggal pada umumnya.

Pengelola Villa Gading Mayang memiliki berbagai program yang dijalankan guna untuk kenyamanan dan kemudahan dalam aktivitas penduduk di dalamnya. Program pertama yang biasanya dikelola adalah terkait warga penghuni lingkungan perumahan.

Pengelola Villa Gading Mayang juga memiliki program yang berjalan secara berkala setiap bulan yaitu iuran untuk keamanan, inventaris, petugas kebersihan dan biaya tak terduga lainnya. Iuran ini didata oleh bendahara menggunakan 'kartu kuning' setiap rumah beserta klaster-nya dengan jumlah bergantung kepada jenis klaster perumahan. Setiap 6 bulan sekali catatan iuran pembayaran akan didata kemudian dimasukkan ke dalam *Microsoft Excel* agar terlihat total iuran yang sudah masuk dan berapa pengeluaran per bulan. Pembayaran iuran dapat dilakukan secara tunai maupun non-tunai dengan transfer *e-wallet* atau akun bank perumahan. Dana dari iuran akan digunakan dalam keamanan dan kebersihan lingkungan perumahan klaster Villa Gading Mayang. Terkadang penduduk perumahan juga sering memberikan dana

bantuan diluar uang iuran yang dimasukkan pencatatannya kepada kategori dana sosial. Selain dalam penjagaan keamanan, kebersihan, serta lingkungan perumahan, dana iuran ini juga akan digunakan ketika ada pengadaan acara kebersamaan seperti peringatan 17 Agustus dan lain sebagainya.

2.2 Sistem Informasi

Sistem informasi terdiri dari 2 susunan kata yang memiliki definisinya masing-masing yaitu sistem dan informasi. Sistem menurut (Akbar Fadillah et al., 2016) merupakan rangkaian prosedur yang saling terhubung bekerjasama dengan didalamnya terdapat masukan, proses, serta keluaran untuk mencapai tujuan bersama yang diharapkan. Definisi sistem (Tukino, 2018) adalah suatu jaringan kerja dari berbagai prosedur yang saling berhubungan, berkumpul bersama, untuk melakukan atau menyelesaikan suatu capaian tertentu. Kata kedua yaitu Informasi menurut (Yanuardi & Permana, 2018) merupakan data yang telah diolah dari sumber terpercaya yang mengandung pengetahuan bermanfaat dan diberi sesuai keperluan sehingga lebih berarti untuk penerimanya guna selanjutnya bertujuan untuk mengambil sebuah keputusan. Informasi didefinisikan oleh (Maydianto & Ridho, 2021) adalah sesuatu bermakna yang harus benar-benar bebas dari kesalahan yang menyesatkan sehingga mengandung nilai penuh yakni keakuratan, tepat waktu, dan relevan guna menjadi pertimbangan dan pengetahuan dalam proses penting pengambilan keputusan.

Berlandaskan penjabaran makna dari dua kata penyusun diatas, maka sistem informasi dapat didefinisikan sebagai suatu rangkaian dari berbagai prosedur yang saling berhubungan bekerjasama dalam melakukan pengolahan data mulai dari masukan, proses, kemudian menghasilkan keluaran pengetahuan fakta berupa informasi yang dapat digunakan dalam pengambilan keputusan sehingga tercapai tujuan yang diharapkan.

2.3 Analisis dan Perancangan Sistem

Analisis didefinisikan oleh (Septiani et al., 2020) sebagai kegiatan berpikir dalam menguraikan atau memecahkan suatu permasalahan dari unit menjadi unit-unit terkecil sehingga ditemukan penyelesaian yang tepat. Analisis dalam sebuah sistem adalah teknik memecahkan permasalahan yang dihadapi dengan menguraikan bagian-bagian dari komponen sebuah sistem dengan cara mempelajari seberapa bagus dan efektif bagian dari komponen tersebut bekerja dan saling berinteraksi dalam mencapai tujuannya. Menurut (Taufiq, 2013) analisis sistem merupakan kegiatan mempelajari sistem (baik sistem manual maupun sistem yang sudah komputerisasi) secara keseluruhan mulai dari

menganalisa sistem, analisis masalah, desain logik, hingga memberikan keputusan dari hasil analisis tersebut.

Perancangan didefinisikan oleh (W. Hidayat et al., 2016) adalah proses merencanakan sesuatu terlebih dahulu yang merupakan wujud visual yang dihasilkan dari perencanaan berbagai bentuk kreatif. Perancangan desain dimulai dengan hal-hal yang tidak teratur berbentuk gagasan atau ide-ide kemudian melalui proses penggarapan dan pengelolaan akan menghasilkan hal yang teratur sehingga kemudian bisa memenuhi fungsi dan kegunaan secara baik. Perancangan berisikan penggambaran, perencanaan, pembuatan sketsa dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Perancangan dalam sebuah sistem menurut (Satzinger et al., 2012) merupakan kumpulan dari berbagai aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan agar menghasilkan produk sistem yang efektif dan berkesesuaian dengan kebutuhan pengguna. Proses perancangan sistem dijalankan untuk merancang suatu sistem baru atau memperbaiki sistem yang sudah ada agar menjadi lebih baik dengan melewati tahapan *input*, *proses*, *output*.

Berlandaskan berbagai pengertian dan pemahaman diatas dapat disimpulkan bahwa analisis dan perancangan sistem merupakan kegiatan perencanaan pengembangan sebuah sistem yang diawali dengan kegiatan berpikir, menguraikan, kemudian memecahkan berbagai permasalahan yang terjadi dengan menguraikan berbagai komponen yang terdapat dalam sebuah sistem hingga bagaimana mereka saling berinteraksi untuk mencapai tujuan yang selanjutnya digambarkan ke dalam bentuk visual berisikan perancangan perencanaan sampai kepada pengelolaan secara rinci sehingga tercipta sistem informasi yang efektif dan tepat dalam memenuhi berbagai kebutuhan pengguna.

Pengembangan sistem yang telah melalui analisis dan perancangan secara rinci dan matang menurut (Muhidin et al., 2019) tentunya menghasilkan sebuah sistem dengan pondasi yang kuat dan penentu keberhasilan dari sistem informasi yang dihasilkan dengan kesesuaian terhadap kebutuhan pengguna nantinya.

2.4 Metode Analisis dan Perancangan

Analisis dan perancangan sebuah sistem informasi memiliki beberapa metode yang dapat digunakan dalam pengembangannya dimana menurut (B. Shelly & J. Rosenblatt, 2012) banyak metode yang dapat digunakan namun terdapat tiga yang paling populer diantaranya adalah SSAD (*Structured Analysis and Design*) yang merupakan metode tradisional yang masih sering digunakan, OOAD (*Object- Oriented Analysis and Design*) yang termasuk baru namun paling banyak dipilih oleh para analis pengembang sistem, serta yang ketiga yaitu metode *Agile*, dikenal juga dengan metode adaptif (*adaptive methods*) yang

termasuk trend baru dalam pengembangan sebuah sistem. Ketiga metode ini tentunya memiliki perbandingan perbedaannya masing-masing, yang dapat terlihat pada tabel dibawah ini:

Tabel 1. Perbandingan Metode SSAD, OOAD, dan *Agile*

SSAD	OOAD	<i>Agile/Adaptive methods</i>
Metode yang merepresentasikan sistem kedalam bentuk data dan segala proses yang dilakukan pada data tersebut, dimana pengembangan sistem diatur dalam fase dengan kemampuan antar proses (<i>deliverables</i>) dan setiap tahapan (<i>milestone</i>) menjadi tolak ukur kemajuan progresnya.	Metode yang memandang sebuah sistem kedalam bentuk objek yang menggabungkan data dan proses didalamnya, dimana objek ini merepresentasikan orang, benda-benda, transaksi, dan peristiwa yang sebenarnya terjadi. Metode ini menganalisa kebutuhan sistem dari sudut pandang kelas-kelas dan objek kemudian mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut sesuai dengan kombinasi antara struktur data dan perilaku dalam satu entitas.	Metode yang lebih menekankan kepada upaya berbasis tim yang intens yang tergabung ke dalam <i>Agile Community</i> . Metode ini memecah proses pengembangan kedalam bentuk iterasi yang menambah fungsionalitas, dimana iterasi ini akan dirancang, dibangun, dan diuji dalam proses berkelanjutan dengan bentuk pengembangan model spiral.
Alat pemodelan yang digunakan diantaranya terdapat DFD (<i>Data Flows Diagram</i>) dan deskripsi proses, serta pemodelan proses bisnis (<i>business process modelling</i>)	Alat pemodelan yang digunakan adalah berbagai macam diagram berorientasi objek yang menggambarkan aktor,alur sistem, metode dan pesan dengan rancangan umumnya digambarkan dalam bentuk UML (<i>Unified Model Language</i>), serta	Alat pemodelan yang lebih meningkatkan dan memfokuskan komunikasi antar tim seperti perangkat lunak kolaboratif (<i>collaborative software</i>),

	pemodelan proses bisnis (<i>business process modelling</i>)	<i>brainstorming</i> , <i>whiteboards</i> , dan sebagainya.
Metode tradisional dimana tidak ada reusabilitas sehingga perubahan dalam pengembangan membutuhkan biaya yang lebih besar apalagi jika perubahan terjadi di fase akhir.	Metode yang terintegrasi secara mudah dengan model pemrograman berbasis objek (<i>Object Oriented Programming</i>) yang populer digunakan karena kode berbentuk modular dan <i>reusable</i> yang dapat mengurangi biaya pengembangan serta mudah untuk dilakukan pemeliharaan.	Metode terbaru yang berfokus kepada kolaborasi dan fleksibilitas pengembangan terhadap perubahan namun dapat mengakibatkan revisi berulang kali karena kurangnya pandangan yang jelas tentang arah pengembangan.
Dokumentasi hasil analisa dan perancangan berfokus kepada tahapan dalam pengembangan sistem sehingga berbentuk diagram alur data dan dokumen prosedur sistem	Dokumentasi hasil analisa dan perancangan berfokus kepada pemodelan objek beserta interaksinya antar satu sama lain didalam sistem sehingga berisikan diagram-diagram UML (<i>Unified Model Language</i>)	Dokumentasi hasil analisa dan perancangan berfokus kepada kolaborasi dan hasil komunikasi tim sehingga biasa dibuat kedalam bentuk catatan-catatan hasil diskusi berakibatkan kurangnya dokumentasi yang dapat memenuhi spesifikasi dan panduan yang jelas dalam pengembangan.

(Sumber : Olahan Penulis)

2.5 OOAD (*Object-Oriented Analysis and Design*)

Object-Oriented Analysis and Design (OOAD) merupakan metode dalam analisis dan perancangan sistem informasi yang didefinisikan oleh (S. Valacich & F. George, 2017) adalah metode yang menggabungkan data dan proses sehingga menjadi satu entitas tunggal disebut sebagai objek yang bertujuan agar elemen-elemen dalam sistem dapat digunakan kembali atau *reusable* yang mampu meningkatkan kualitas sistem yang dikembangkan. Objek ini adalah yang menggabungkan antara data dan proses yang merepresentasikan orang, benda, transaksi, dan peristiwa yang sebenarnya terjadi (B. Shelly & J. Rosenblatt, 2012). OOAD ini merupakan metode dengan pendekatan yang memikirkan suatu masalah menggunakan model yang dibuat menurut konsep dunia nyata (Nasikhin et al., 2019).

Metode OOAD ini memiliki beberapa konsep dasar yang dijelaskan oleh (Purnawati et al., 2019) sebagai berikut :

1. Objek

Merupakan benda atau sesuatu berbentuk fisik dan konseptual yang ada disekitar kita seperti diantaranya manusia, dokumen, *hardware* dan sebagainya. Objek ini memiliki keadaan sesaat yaitu *state* yang menyatakan kondisi atau himpunan keadaan yang terjadi pada objek tersebut.

Objek memiliki nilai internal yaitu **Atribut** yang menggambarkan identitas, karakter, kondisi sesaat, beserta hubungan dari objek tersebut dengan yang lainnya. Perubahan keadaan atau *state* pada suatu objek didefinisikan sebagai suatu **Behaviour**. *Behaviour* ini mencerminkan perilaku atau reaksi dari objek baik itu aksi bertindak maupun aksi memberi yang ditentukan baik dari beberapa atau himpunan operasi yang dapat dilakukan objek tersebut. Perilaku dari sebuah objek dicerminkan oleh 3 hal, yaitu *Interface*, *Service*, dan *Method*. *Interface* merupakan pintu dalam mengakses segala fungsi yang dapat dilakukan oleh objek tersebut yaitu *Service*, serta mekanisme internal dari objek pencerminan *behaviour* objek yaitu *Method*.

2. Class (Kelas)

Merupakan definisi umum atau cetak biru dari himpunan objek yang sejenis, dimana akan ditetapkan spesifikasi perilaku dan atribut-atribut yang dimiliki oleh objek tersebut.

3. Black Boxes (Kotak Hitam)

Dalam dasar implementasi berorientasi objek (OO), objek atau sistem adalah kotak hitam dimana hanya pada pengembangnya yang dapat memahami detail proses-proses yang berada dalam kotak hitam tersebut. Pengguna sistem

tidak perlu mengetahui apa yang dilakukan pada sistem, yang terpenting sistem dapat digunakan untuk memproses kebutuhannya.

Pada kotak hitam ini, terdapat 2 hal yang perlu diketahui yaitu **Encapsulation (Enkapsulasi)**, dan **Message**. Enkapsulasi merupakan proses menyembunyikan seluruh detail implementasi sebuah objek sehingga salah satunya cara pengguna mengakses data objek hanya melalui *Interface* guna terhindar dari "campur tangan" pihak diluar pengembang. Pengguna dalam berkomunikasi dengan objek memerlukan *message* yang merupakan permintaan agar objek menerima untuk membawa metode dan mengembalikan hasil dari aksi perilaku kepada objek pengirim.

4. Asosiasi dan Agregasi

Asosiasi adalah hubungan bermakna antara objek yang digambarkan dengan sebuah garis penghubung antar objek tersebut, sedangkan Agregasi merupakan bentuk khusus dari sebuah asosiasi yang menggambarkan seluruh bagian pada satu objek merupakan bagian dari objek yang lain.

Metode OOAD dalam tahapan pengerjaannya terdiri dari 2 bagian diantaranya adalah analisa berorientasi objek (OOA (*Object-Oriented Analysis*)) dan perancangan berorientasi objek (OOD (*Object-Oriented Design*)) dengan pembagian tahapan sebagai berikut :

1. OOA (*Object-Oriented Analysis*)

Object-Oriented Analysis ini merupakan metode analisis yang dilakukan dengan memeriksa hal-hal terkait syarat dan kebutuhan dilihat dari sudut pandang kelas dan objek yang dijumpai dalam ruang lingkup permasalahan dimana syarat ini harus terpenuhi oleh suatu sistem yang akan dibangun sehingga tercapainya ketepatan hasil analisa, konsistensi, dan efisiensi pengembangan kedepannya (Hasanuddin, 2016).

OOA digunakan untuk mengidentifikasi dan mendefinisikan objek yang akan melakukan pekerjaan tertentu dan menentukan tindakan pengguna yang diperlukan dalam menyelesaikan pekerjaan tersebut dimana biasa kita sebut dengan *Use Case* (Satzinger et al., 2012). Pada tahapan OOA dilakukan analisa berdasarkan kebutuhan pengguna sehingga sistem yang dikembangkan berkualitas baik. Hasil pada OOA adalah deskripsi dari apa sistem secara fungsional diperlukan untuk melakukan, dalam bentuk sebuah model konseptual (Nazaruddin et al., 2022).

Terdapat 3 tahapan utama yang harus dilakukan dalam pengerjaan OOA berdasarkan (Pane & Sarno, 2015) yaitu :

1) *Identify the Object (actor, functionalities, system entities)*

Identifikasi objek yang ada (aktor, fungsionalitas, beserta entitas sistem)

2) *Illustrate how the object interrelate (Use Case modeling)*

Ilustrasikan bagaimana tiap objek yang ada saling terhubung (menggunakan model *Use Case*)

3) *Specify the attributes and behavior of the objects (Use Case detailed description)*

Spesifikasikan atribut dan perilaku setiap objek (deskripsi detail dari *Use Case*)

2. OOD (*Object-Oriented Design*)

Object-Oriented Design adalah tahapan dalam metode OOAD yang menerjemahkan hasil analisa pada tahap OOA ke dalam bentuk model desain visual yang berfungsi sebagai cetak biru panduan pada pembuatan sistem perangkat lunak (OOP) (S. Pressman, 2000). Pada tahapan rancangan ini harus ditemukan objek yang tepat dan relevan, kemudian dimasukkan kedalam kelas dengan perincian yang tepat, didefinisikan kelas antarmukanya dan hirarki pewarisan yang selanjutnya ditetapkan hubungan dan kunci dari hubungan antar objek yang berelasi.

Desain yang dibuat haruslah spesifik menangani masalah yang sedang dihadapi, serta cukup untuk menjadi solusi umum untuk permasalahan di masa depan yang akan dihadapi. Kegiatan utama dalam tahap OOD ini dapat dibagi menjadi 2 berdasarkan (S. Pressman, 2000) yaitu desain sistem dan desain objek. Desain sistem lebih berfokus kepada penciptaan arsitektur produk dengan mempertimbangkan 3 komponen diantaranya antarmuka pengguna (*user interface*), fungsi pengelolaan data, serta fungsi pengelolaan tugas. Sedangkan pada bagian desain objek berfokus kepada detail internal kelas individu, pendefinisian atribut, operasi dan detail *message* umumnya tergambar dengan UML (*Unified Model Language*).

Terdapat 3 tahapan utama yang harus dilakukan dalam pengerjaan OOD berdasarkan (Pane & Sarno, 2015) yaitu :

1) *Arrange diagram activities for each scenario (Activity Diagrams as an output)*

Merangkai diagram aktivitas untuk setiap skenario dengan *Activity Diagram* sebagai hasil keluarannya

2) *Build the detailed Class diagram*

Bangun *Class diagram* secara rinci

3) *Proceed to the detailed design*

Melanjutkan hingga ke detail desain sistem yang akan dikembangkan, bisa penyesuaian kebutuhan diagram beserta desain antarmuka sistem

UML (*Unified Model Language*)

Unified Model Language (UML) yang termasuk ke dalam notasi grafikal merupakan alat atau model guna menggambarkan rancangan suatu sistem perangkat lunak terkhusus sistem yang dibangun berorientasi kepada objek (OO). Model ini sudah mulai digunakan pada akhir 1980-an kemudian populer digunakan sejak tahun 1997 oleh para analis sistem dalam menerjemahkan hasil analisisnya (Fowler, 2004). UML disebut sebagai bahasa standar dokumentasi atau cetak biru acuan dalam pengembangan sebuah sistem. UML akan memvisualisasikan, menentukan, membangun, serta mendokumentasikan arsitektur sistem sehingga UML ini membantu pengembang sistem dalam membangun sistem yang tepat dan sesuai dengan kebutuhan pengguna (S. Pressman, 2010).

Dalam pemvisualan UML, banyak fitur diagram tersedia yang dapat digunakan untuk mengekspresikan semua aspek penting pada suatu sistem sehingga diagram UML berdasarkan (Booch et al., 2007) dapat diklasifikasikan kedalam 2 kelompok yaitu *Structure Diagrams* (Diagram Struktural) dan *Behavior Diagram* (Diagram Perilaku). Diagram struktural digunakan untuk menunjukkan struktur statis dari elemen yang terdapat pada sistem dengan menggambarkan elemen fisik sistem, konfigurasi *runtime*, serta elemen domain spesifik dari bisnis yang dibangun sistemnya. Diagram UML struktural diantaranya adalah *Package Diagram*, *Class Diagram*, *Component Diagram*, *Deployment Diagram*, *Object Diagram*, dan *Composite Structure Diagram*. Sedangkan diagram perilaku digunakan untuk menggambarkan semantik perilaku dinamis dari suatu masalah atau pengembangan implementasi sistem seperti objek yang dibuat kemudian dihancurkan, objek mengirim pesan satu sama lain secara teratur dan sebagainya. Diagram UML perilaku diantaranya adalah *Use Case Diagram*, *Activity Diagram*, *State Machine Diagram*, serta *Interaction Diagram* (*Sequence Diagram*, *Communication Diagram*, *Interaction Overview Diagram*, *Timing Diagram*). Diagram UML yang sering digunakan dalam dokumentasi analisis dan perancangan sistem adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* (S. Pressman, 2010).

1. Use Case Diagram

Use Case merupakan skenario yang menceritakan bagaimana *end user* memainkan salah satu dari sejumlah peran yang terdeskripsi berinteraksi dengan sistem berdasarkan keadaan tertentu dimana narasi dapat dalam berbagai bentuk dari teks naratif, garis besar tugas atau interaksi, deskripsi berbasis *template*, atau dengan representasi diagram (S. Pressman, 2010). *Use Case* dalam penggunaannya akan menganalisis dan menunjukkan perilaku dan kebutuhan

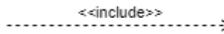
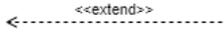
fungsional dari sebuah sistem sehingga dapat membantu pengembang memahami kebutuhan tersebut tanpa perlu khawatir bagaimana hal itu akan diimplementasikan kedalam sistem (S. Valacich & F. George, 2017).

Penggambaran *Use Case* dengan model diagram memberikan kemudahan kepada *developer* untuk memahami siapa ataupun apa yang disebut sebagai aktor berinteraksi dengan sistem. Diagram ini menunjukkan apa yang diinginkan dunia luar untuk sistem lakukan (Booch et al., 2007). *Use Case Diagram* dalam menggambarkan cara aktor berinteraksi dengan sistem harus dapat menjawab pertanyaan berikut yang disarankan Jacobson [JAC92] berdasarkan (S. Pressman, 2000) :

- 1) Apa saja tugas dan fungsi utama yang dilakukan aktor?
- 2) Informasi apa saja pada sistem yang akan didapatkan, dibuat, atau diubah?
- 3) Apakah aktor perlu memberikan informasi kepada sistem tentang perubahan lingkungan eksternal?
- 4) Informasi seperti apa yang diinginkan aktor pada sistem?
- 5) Apakah aktor ingin diberitahu tentang perubahan yang tak terduga?

Tabel 2. Simbol *Use Case Diagram*

No.	Nama Simbol	Gambar	Keterangan
1.	<i>Use Case</i>		Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan dengan aktor dan dinyatakan menggunakan kata kerja
2.	Aktor		Abstraksi dari orang atau sistem lain yang menjalankan fungsi dari target sistem. Aktor ini dapat memiliki beberapa peran bergantung pada pembagian tugas yang berkaitan pada konteks target sistem. Aktor berinteraksi dengan <i>Use</i>

			Case namun tidak memiliki kontrol terhadap <i>Use Case</i>
3.	Asosiasi (<i>Association</i>)		Penghubung antara aktor dengan <i>Use Case</i> yang saling berinteraksi secara langsung
4.	Generalisasi (<i>Generalized</i>)		Menunjukkan indikasi bahwa aktor berinteraksi secara pasif dengan sistem
5.	<i>Include</i>		Menunjukkan bahwa sebuah <i>Use Case</i> seluruhnya merupakan fungsionalitas dari <i>Use Case</i> lainnya yang berarti pemanggilan <i>Use Case</i> yang diperlukan untuk menjalankan sebuah fungsi oleh <i>Use Case</i> lainnya
6.	<i>Extend</i>		Menunjukkan perluasan dari <i>Use Case</i> bahwa <i>Use Case</i> tersebut merupakan tambahan fungsional dari <i>Use Case</i> lainnya jika suatu syarat atau kondisi terpenuhi

Sumber : (Hendini, 2016)

2. *Activity Diagram*

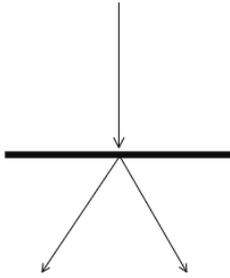
Ketika melakukan perancangan pada suatu sistem seringkali terdapat *Use Case* yang kompleks serta proses bisnis dan algoritma yang sulit untuk dipahami dan didokumentasikan dalam bentuk teks, sehingga perlu dihadirkan representasi visual dengan aliran kompleks yang kuat yaitu *Activity Diagram*. *Activity Diagram* memberikan visualisasi terkait *flow* atau aliran aktivitas baik dalam sistem, proses bisnis, alur kerja, dan proses lainnya dimana diagram ini berfokus terhadap rangkaian aktivitas yang dilakukan dan siapa atau apa yang bertanggung jawab terhadap kinerja aktivitas tersebut (Booch et al., 2007).

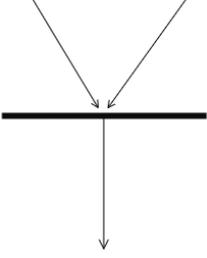
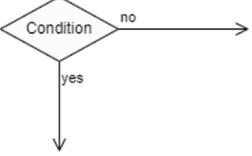
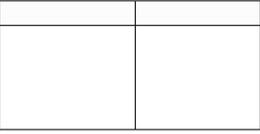
Activity Diagram menurut (S. Pressman, 2010) menggambarkan perilaku dinamis dari suatu sistem atau bagian dari sistem tersebut dengan aliran kontrol

diantara tindakan yang dilakukan oleh sistem. Penggambaran ini mirip dengan *flowchart* tetapi pada *diagram Activity* dapat menunjukkan arus alur yang bersamaan. *Activity Diagram* berdasarkan (S. Valacich & F. George, 2017) digunakan untuk memberikan solusi maupun menyelesaikan tugas seperti berikut :

- 1) Menggambarkan *flow of control* atau aliran kendali dari suatu aktivitas ke aktivitas lainnya
- 2) Memberi bantuan untuk memahami tindakan apa yang perlu dilakukan dalam analisa *Use Case*
- 3) Memberi bantuan untuk mengidentifikasi ekstensi didalam sebuah *Use Case*
- 4) Memodelkan alur kerja dan proses bisnis
- 5) Memodelkan langkah berurutan dan bersamaan (*sequential and concurrent*) dalam proses komputasi.

Tabel 3. Simbol *Activity Diagram*

No.	Nama Simbol	Gambar	Keterangan
1.	<i>Start Point</i> (Status Awal)		Menunjukkan awal aktivitas yang diletakkan di pojok kiri atas
2.	<i>End Point</i> (Status Akhir)		Menunjukkan akhir dari aktivitas
3.	<i>Activities</i> (Aktivitas)		Menggambarkan suatu proses atau kegiatan bisnis (aktivitas) yang dilakukan sistem. Aktivitas ini biasanya diawali dengan kata kerja
4.	<i>Fork</i> (Percabangan)		Menunjukkan kegiatan yang dilakukan secara paralel atau menggabungkan dua kegiatan parallel menjadi satu

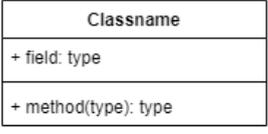
5.	Join (Penggabungan)		Menunjukkan adanya dekomposisi yaitu dua kegiatan parallel (aliran kontrol) digabungkan menjadi satu guna menyinkronkan aliran tersebut
6.	Decision (Percabangan)		Menggambarkan pilihan untuk pengambilan keputusan (<i>true / false</i>)
7.	Swimlane		Menunjukkan pembagian <i>Activity Diagram</i> yang memisahkan organisasi bisnis yang bertanggung jawab untuk menunjukkan yang bertanggung jawab terhadap aktivitas tersebut

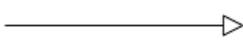
Sumber : (Hendini, 2016)

3. Class Diagram

Class diagram merupakan diagram pada UML yang digunakan untuk memodelkan kelas termasuk atribut, operasi, dan hubungan beserta asosiasinya terhadap kelas lainnya dengan menyediakan tampilan statis atau struktural dari sebuah sistem. Diagram ini tidak menunjukkan secara dinamis komunikasi antar objek kelasnya (S. Pressman, 2010). *Class diagram* menurut (S. Valacich & F. George, 2017) menunjukkan struktur statis berorientasi objek didalamnya yaitu kelas objek, struktur internalnya serta hubungan atau relasi dimana mereka saling berpartisipasi.

Tabel 4. Simbol *Class Diagram*

No.	Nama Simbol	Gambar	Keterangan
1.	<i>Class</i> (Kelas)		Menggambarkan kelas pada struktur sistem yang didalamnya terdiri dari 3 komponen yaitu nama kelas yang harus bersifat unik dan tidak berjarak dalam penulisan, kemudian atribut

			yang dapat disediakan oleh objek dari kelas sepanjang waktu, serta operasi yang menyatakan fungsi yang terdapat dalam kelas.
2.	<i>Association</i> (Asosiasi)		Menunjukkan relasi antar kelas dengan makna umum. Asosiasi ini biasanya juga diikuti dengan <i>multiplicity</i>
3.	<i>Directed Association</i> (Asosiasi berarah)		Menunjukkan relasi antar kelas dengan makna suatu kelas digunakan oleh kelas yang lainnya. Asosiasi ini biasanya juga diikuti dengan <i>multiplicity</i>
4.	<i>Generalization</i> (Generelisasi)		Menunjukkan relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
5.	Dependency (Kebergantungan)		Menunjukkan relasi antar kelas dengan makna kebergantungan antar kelas
6.	<i>Aggregation</i> (Agregasi)		Menunjukkan relasi antar kelas "has a" yang bermakna suatu kelas adalah bagian dari kelas lain namun kedua kelas tersebut dapat berdiri sendiri
7.	<i>Composition</i> (Komposisi)		Tipe spesial dari agregasi yang menunjukkan relasi antar kelas yang bermakna "part of" yaitu sebuah kelas yang merupakan bagian dari kelas lain dimana kelas yang lebih kompleks tersusun atas kelas yang lebih sederhana

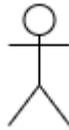
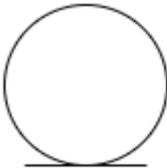
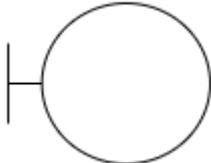
Sumber : (Sukamto & Salahuddin, 2016)

4. Sequence Diagram

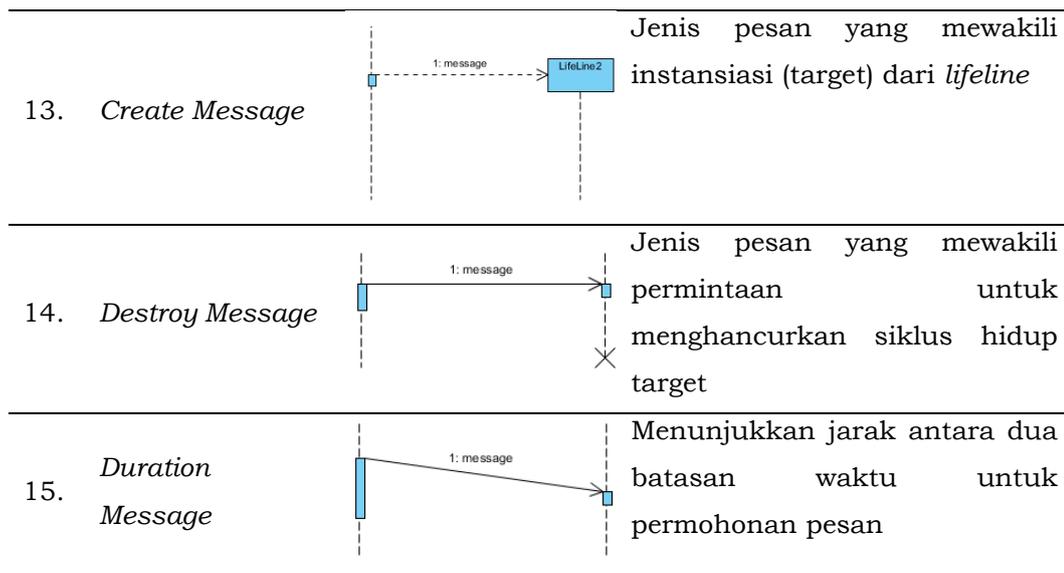
UML menyediakan diagram untuk menggambarkan interaksi antar objek untuk *Use Case* tertentu yaitu *interaction diagram* dengan jenis diagram yang sering digunakan adalah *Sequence Diagram*. *Sequence Diagram* menggambarkan interaksi objek selama periode waktu tertentu yaitu *Sequence* atau urutan pesan antara aktor eksternal dengan sistem selama skenario *Use Case*. Hal ini dikarenakan pola interaksi yang bervariasi dari satu *Use Case* ke *Use Case* lainnya, sehingga setiap *Sequence Diagram* menunjukkan interaksi terkait *Use Case* tertentu (S. Valacich & F. George, 2017).

Sequence Diagram digunakan untuk melacak pelaksanaan skenario dalam konteks yang sama dengan *communication diagram* namun memiliki keunggulan lebih baik dibandingkan *communication diagram* dalam menangkap skenario semantik pada proses awal pengembangan sebelum protokol untuk masing-masing kelas teridentifikasi sehingga lebih mudah membaca pertukaran pesan dengan urutan relatif (Booch et al., 2007).

Tabel 5. Simbol *Sequence Diagram*

No.	Nama Simbol	Gambar	Keterangan
1.	<i>Actor</i> (Aktor)		Menggambarkan abstraksi pengguna yang berada diluar sistem baik manusia atau perangkat eksternal lainnya dan sedang berinteraksi dengan subjek yaitu sistem. Aktor tidak selalu mewakili entitas fisik tertentu tetapi bisa hanya peran tertentu dari entitas tertentu.
2.	<i>Entity Class</i>		Bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas pembentuk gambaran awal sistem yang menjadi landasan untuk menyusun basis data
3.	<i>Boundary Class</i>		Berisikan kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem contohnya seperti tampilan form cetak

4. <i>Control Class</i>		Berisikan logika aplikasi yang tidak memiliki tanggung jawab terhadap entitas seperti contohnya aturan bisnis yang melibatkan berbagai objek
5. <i>Message</i>		Abstraksi pengiriman pesan antar kelas
6. <i>Activation</i>		Mewakili sebuah eksekusi dari objek dimana panjang kotak berbanding lurus dengan durasi aktivasi sebuah operasi
7. <i>Object</i>		Menggambarkan bagaimana suatu objek berperilaku pada konteks sistem tertentu
8. <i>Lifeline</i>		Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> yang memastikan keberadaan posisi mereka
9. <i>Self Message</i>		Jenis pesan yang mewakili permohonan pesan dari <i>lifeline</i> yang sama
10. <i>Call Message</i>		Jenis pesan yang mewakili permintaan operasi dari target <i>lifeline</i>
11. <i>Return Message</i>		Jenis Pesan yang mewakili pengiriman kembali informasi ke pengirim pesan atau pemanggil berdasarkan pesan sebelumnya
12. <i>Recursive Message</i>		Jenis pesan yang mewakili permohonan pesan dari <i>lifeline</i> yang sama, dimana dia menargetkan aktivasi diatas aktivasi tempat pesan tersebut berasal



Sumber : (Hendini, 2016)

User Interface (UI)

User Interface (UI) atau disebut juga antarmuka pengguna adalah sarana interaksi antara manusia dengan sistem dimana ia merupakan bagian dari sistem yang mengatur tampilan perangkat yang dilihat pengguna juga memfasilitasi interaksi yang menyenangkan antara pengguna dengan sistem (Utama, 2020). Menurut (B. Shelly & J. Rosenblatt, 2012) *User Interface* menjelaskan bagaimana pengguna berinteraksi dengan sistem komputer yang terdiri dari semua *hardware*, *software*, layar, menu, fungsi, output dan segala hal yang mempengaruhi komunikasi antara pengguna dan komputer sistem. Sebuah UI yang baik berdasarkan IBM adalah UI yang tidak disadari (*notice*) oleh pengguna sistem, dimana UI tersebut 'masuk akal' karena melakukan apa yang diharapkan pengguna.

User Interface adalah media bagi pengguna untuk berinteraksi dengan sistem dimana perancangan desain UI tentunya bertujuan untuk menghasilkan interaksi yang baik bagi pengguna sehingga meningkatkan *usability* dari antarmuka sistem (Desideria & Bandung, 2020). UI menciptakan komunikasi yang efektif antara manusia sebagai pengguna dengan komputer. Prinsip merancang desain antarmuka yaitu desain mengidentifikasi objek dan tindakan antarmuka kemudian membuat tata letak pada layar sehingga membentuk dasar untuk cetak biru (*prototype*) *user interface*. UI merupakan komponen penting karena membentuk persepsi pengguna tentang sistem (S. Pressman, 2010).

Pada buku Pressman ini dikatakan ada 3 aturan emas (*golden rules*) oleh Theo Mandel dalam merancang desain *user interface* :

- 1) Letakkan pengguna dalam kendali
- 2) Kurangi beban memori pengguna
- 3) Buatlah antarmuka yang konsisten.

2.6 Requirement Traceability Matrix (RTM)

Requirement didefinisikan IEEE sebagai suatu kemampuan yang harus dimiliki oleh sebuah sistem perangkat lunak dalam memberikan solusi pemecah masalah dengan pemenuhan kebutuhan pengembangan sistem. *Requirement* ini adalah gambaran kebutuhan *user* dari sistem yang harus terpenuhi oleh pengembang dan berperan sebagai batasan dengan sifat yang dinamis bergantung pada pendapat *stakeholder*, kebutuhan pengguna beserta proses yang terlibat dalam sistem sehingga *requirement* ini harus selalu diperbarui dalam merespon perubahan yang terjadi dalam sistem guna capaian sesuai dengan target yang diinginkan (Nur Fitriani, 2017).

Perubahan *requirement* ini dapat dilakukan pada *Requirement Management* yang merupakan rangkaian aktivitas dalam membentuk tim pengembangan mengidentifikasi, mengontrol, serta melacak persyaratan dan perubahan dari persyaratan tersebut selama proyek pengembangan berlangsung. *Requirement Management* dimulai dengan tahap identifikasi kebutuhan, dan jika sudah teridentifikasi selanjutnya masuk kepada tabel *traceability*. Tabel *traceability* dapat membantu dalam melakukan evaluasi terhadap pemenuhan kebutuhan pengguna oleh sistem dengan metode *Requirement Traceability Matrix* (RTM) (S. Pressman, 2000).

Requirement Traceability Matrix (RTM) merupakan tabel yang berisikan daftar kebutuhan, atribut yang bervariasi, serta status dari setiap kebutuhan (*Requirements*) sehingga dapat memastikan semua *requirement* ini telah terpenuhi (Fakhrillah, 2016). *Requirement Traceability Matrix* (RTM) memiliki beberapa model yang dapat digunakan, namun model yang berkesesuaian dengan penelitian ini adalah *functional tracing*. *Functional tracing* merupakan pelacakan pada setiap aspek fungsional dalam perancangan dan pengembangan sistem dengan melakukan pemetaan matriks terhadap objek yang akan di evaluasi meliputi pemetaan model analisis, proses, desain, dan organisasi dari pengembangan yang keseluruhan tercakup dalam definisi fungsional sistem (Pinheiro, 2004). Evaluasi RTM *functional tracing* dilakukan dengan pemodelan hubungan antar diagram, objek, kelas, dan lainnya yang terdefinisi dengan baik sebelumnya.

2.7 Penelitian Terdahulu

Berikut adalah beberapa penelitian terdahulu yang berkaitan dengan topik analisis dan perancangan sistem informasi administrasi. Dalam pelaksanaan penelitian, terdapat beberapa perbedaan antara penelitian yang telah ada dengan penelitian yang penulis lakukan. Penelitian yang ada memiliki perbedaan dimana terdapat penelitian dengan topik yang sama namun objek serta metode penelitian yang digunakan berbeda begitupun sebaliknya

Tabel 6. Penelitian Terdahulu

No.	Judul Penelitian	Penulis & Tahun	Tujuan	Hasil Penelitian
1.	Perancangan Sistem Informasi Kependudukan di Perumahan Citra Swarna Riverside	(Pujiyanto & Alamsyah, 2022)	Memudahkan bagian kependudukan dalam melakukan pendataan penduduk, dan yang berkaitan dengan kependudukan serta memudahkan proses administrasi kependudukan	Perancangan sistem dengan model UML dapat menyelesaikan permasalahan yang dihadapi terkait administrasi penduduk perumahan dengan menghasilkan sistem informasi kependudukan berbasis website yang efektif, efisien, serta <i>user friendly</i>
2.	Perancangan Sistem Informasi Administrasi Kependudukan Desa Berbasis Website Pada Desa Winong	(Syukron, 2019)	Membuat sistem informasi administrasi kependudukan dengan metode prototype yang dapat mempermudah pengelolaan data kependudukan	Kehadiran sistem administrasi kependudukan mempermudah penduduk dalam kepengurusan administrasi serta peningkatan efektivitas dan efisiensi dalam pelayanan masyarakat

3.	Perancangan dan Analisis Sistem Informasi Pelayanan Kependudukan	(Suruali, 2015)	Membuat sistem yang dapat melakukan fungsi pencatatan dengan metode OOAD yang diharapkan dapat memberi nilai tambah dan solusi untuk berbagai pihak	Membangun sistem informasi pelayanan kependudukan berbasis web yang mempermudah proses pencarian data penduduk, serta penekanan biaya pengeluaran dengan jalur birokrasi yang singkat
4.	Sistem Pengelolaan Data Warga Informasi Kegiatan dan Informasi Penggunaan Iuran pada Lingkungan Rukun Tetangga (RT) Berbasis Web	(Yustrinita & Aprilia, 2021)	Mengembangkan sistem pengelolaan data pada RT menggunakan Waterfall dan UML untuk menyediakan pengelolaan data warga, informasi kegiatan dan laporan pengelolaan iuran sehingga mempercepat proses pelayanan untuk warga	Aplikasi sistem informasi RT ini dapat melakukan input data warga, membuat bukti lapor tamu, input informasi kegiatan warga dan input laporan iuran yang mengefisiensikan proses dokumentasi dan pengelolaan pelayanan
5.	Sistem Informasi Pembayaran Iuran Keamanan dan Kebersihan pada Perumahan Berbasis Website	(Siregar et al., 2020)	Menghasilkan sistem pembayaran iuran di perumahan menteng indah berbasis website yang mempermudah warga dan	Sistem Informasi Pembayaran iuran berbasis website dapat menyelesaikan masalah serta meningkatkan kualitas pelayanan dalam hal proses pembayaran iuran

	Menggunakan Metode Design Thinking		<i>security</i> dalam melakukan proses pembayaran iuran	
6.	Analisis dan Perancangan Sistem Informasi LENTERA Untuk Membentuk “Smart Society” di Lingkungan Kampus Menggunakan Metode OOAD (Studi Kasus : Universitas PGRI Madiun)	(Setiawan, 2019)	Membangun Ekosistem <i>Smart Society</i> di lingkungan kampus dengan melakukan analisa dan perancangan sistem LENTERA menggunakan Metode OOAD sehingga dapat menjadi pijakan dasar dalam pengembangan sistem informasi <i>Smart Society</i> di lingkungan kampus masa mendatang	Dokumentasi rancangan sistem pembentuk ekosistem <i>social learning</i> berkualitas sehingga dapat menjadi pijakan dasar dalam pengembangan sistem informasi <i>Smart Society</i> di lingkungan kampus masa mendatang
7.	Analisis dan Perancangan Sistem Informasi Reservasi <i>Tour and Travel</i> Menggunakan Metode OOAD Pada Warok <i>Tour and Travel</i>	(Nasikhin et al., 2019)	Merancang sistem informasi pemesanan <i>tour and travel</i> untuk menyelesaikan permasalahan guna mencapai efektivitas pelayanan menggunakan metode OOAD	Rancangan sistem informasi dengan identifikasi10 kebutuhan fungsional dan 4 kebutuhan non serta diagram UML yang menggambarkan interaksi antara user dengan sistem yang diuji dengan diagram <i>traceability matrix</i> mulai dari fungsi, kebutuhan sistem,

Use Case, Activity, Sequence serta desain antarmuka dengan 10 skenario terhadap 2 pengguna terpenuhi

(Sumber : Olahan Penulis)

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu penelitian dilaksanakan kurang lebih selama 4 bulan, dimulai dari Juni 2023 dengan target penyelesaian pada bulan September 2023. Adapun tempat pelaksanaan penelitian di Perumahan klaster Villa Gading Mayang, khususnya pada lingkungan RT (Rukun Tetangga) yang beralamatkan di Kelurahan Mayang Mangurai, Kecamatan Kotabaru, Jambi.

3.2 Alat Penelitian

Alat penelitian berperan sebagai penunjang dalam pelaksanaan penelitian. Penelitian ini memerlukan alat yang terdiri atas *software* dan *hardware*. Adapun peralatan yang dibutuhkan diantaranya adalah sebagai berikut :

1. Perangkat Keras (*Hardware*)

Laptop LENOVO 81D5 AMD A4-9125 RADEON R3, 4 COMPUTE CORES 2C + 2G (2 CPUs), ~2.3GHz dengan besar RAM 8GB serta beberapa perangkat pendukung lainnya seperti Printer.

2. Perangkat Lunak (*Software*)

Sistem Operasi Windows 10, Microsoft Office sebagai pengelola dokumen, Google Chrome, Microsoft Visio 2016, diagrams.net (draw.io), serta Figma Interface Design Tool.

3.3 Metode Pengumpulan Data

Pengumpulan data penelitian dilakukan dengan menggunakan metode diantaranya adalah :

Observasi

Observasi merupakan metode pengumpulan data dengan cara mengamati dan mencatat suatu gejala atau peristiwa secara langsung pada lokasi penelitian untuk mengetahui kondisi yang terjadi guna mencapai tujuan ilmiah penelitian dengan didapatnya pengetahuan dan pemahaman terkait penelitian yang akan dilaksanakan (Syamsudin, 2015). Observasi penelitian dilakukan di Perumahan Villa Gading Mayang guna membantu penulis memahami dan mengenali lebih jauh terkait administrasi kependudukan yang diterapkan pada perumahan tersebut.

Wawancara

Wawancara atau biasa disebut *interview* merupakan salah satu wujud dari komunikasi interpersonal secara langsung antar individu yang berperan sebagai pembicara dan pendengar yang dilakukan secara bergantian untuk

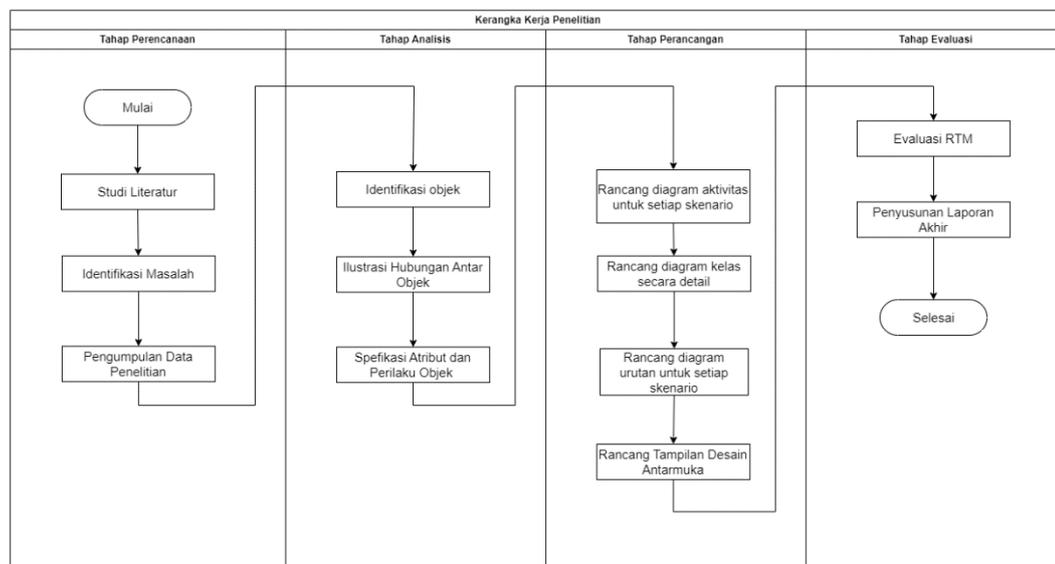
memperoleh informasi dalam bentuk pernyataan-pernyataan lisan mengenai objek penelitian (Pujaastawa, 2016). Wawancara dilakukan dengan beberapa narasumber terpercaya yang memiliki pengetahuan mendalam terkait permasalahan penelitian juga sebagai target pengguna sistem yang akan dirancang yaitu Ketua RT, bendahara, sekretaris, petugas, *security*, serta warga penduduk yang tinggal di Perumahan Villa Gading Mayang.

Studi Literatur

Studi literatur atau kepastakaan merupakan kegiatan yang diwajibkan dalam pelaksanaan penelitian dengan tujuan untuk mengembangkan aspek teoritis maupun aspek manfaat praktis yang dilakukan dengan berbagai metode diantaranya mengumpulkan data pustaka, membaca, mencatat, serta mengolah data penelitian (Kartiningrum, 2015). Studi literatur dilakukan dalam pemahaman landasan teori seperti metode analisis dan perancangan sistem, penelitian terdahulu, aplikasi berbasis *website*, sistem e-administrasi kependudukan RT dan sebagainya.

3.4 Langkah Kerja Penelitian

Pelaksanaan penelitian agar berjalan dengan terstruktur dan sesuai alur, maka dibuatlah suatu kerangka kerja penelitian guna memastikan setiap tahapan yang diperlukan untuk melakukan analisis dan perancangan sistem telah terpenuhi keseluruhannya. Adapun kerangka kerja penelitian ini digambarkan sebagai berikut :



Gambar 1. Kerangka Kerja Penelitian
(Sumber : Olahan Penulis)

Berdasarkan kerangka kerja penelitian diatas, terdapat 4 tahapan yang akan dilakukan dimulai dari tahap perencanaan, tahap analisis, tahap perancangan, hingga tahap evaluasi.

Tahap Perencanaan

Tahap perencanaan merupakan tahapan pertama yang harus dilalui sebelum melakukan penelitian dimana dilakukan persiapan segala sesuatu yang akan menjadi dasar dan kebutuhan dalam pelaksanaan penelitian. Perencanaan penelitian dimulai dengan melakukan studi literatur untuk menemukan serta mengenali teori terkait metode yang akan digunakan dalam penelitian beserta studi peneliti terdahulu yang membahas topik sistem informasi administrasi beserta iuran pada perumahan dan sebagainya guna sebagai acuan dalam penelitian sehingga didapatkan pemahaman yang mendalam mengenai hal-hal yang berkaitan dengan penelitian. Studi literatur ini dilakukan dengan mencari sumber buku, jurnal pada platform internet dan perpustakaan.

Setelah mendapatkan landasan dan pemahaman pada studi literatur, selanjutnya pada tahap perencanaan dilakukan kegiatan identifikasi masalah yaitu mendefinisikan dan memahami permasalahan terkait topik penelitian yang diangkat sehingga ditentukan perumusan masalah dan didapatkan tujuan yang tepat dari penelitian. Merumuskan permasalahan atau identifikasi masalah terkait administrasi Perumahan Villa Gading Mayang dilakukan melalui observasi dan wawancara pihak yang terlibat yaitu ketua RT, sekretaris, bendahara, pihak keamanan dan juga warga perumahan dimana didapatkan pengguna dari sistem yang akan dibuat meliputi ketua RT, staff admin, sekretaris, bendahara, dan warga perumahan Villa Gading Mayang. Observasi dilakukan dengan mengamati secara langsung kondisi di lapangan seperti apa kemudian mencatat apa saja yang perlu dijadikan pertimbangan dilanjutkan dengan wawancara keluhan dan kesulitan permasalahan yang dirasa oleh narasumber. Dengan pelaksanaan identifikasi masalah secara menyeluruh ini diharapkan penulis dapat menemukan solusi yang tepat serta efisien untuk mengatasi permasalahan yang dihadapi.

Proses selanjutnya pada perencanaan ini yaitu mengumpulkan data yang akan dibutuhkan dalam berlangsungnya penelitian. Data penelitian diperoleh dari proses observasi dan wawancara pada tahap identifikasi masalah guna menjadi dasar acuan dalam proses penelitian seperti proses pembayaran iuran perumahan, dokumen kartu iuran, data penduduk perumahan dan sebagainya dalam kegiatan administrasi perumahan Villa Gading Mayang.

Tahap Analisis

Tahapan analisis pada penelitian ini dilakukan dengan mengacu kepada penerapan metode OOA (*Object-Oriented Analysis*) yang dilakukan dengan mengidentifikasi tiap objek yang dalam ruang lingkup permasalahan yang harus terpenuhi. Objek yang telah teridentifikasi kemudian diilustrasikan berlandaskan identifikasi awal yang telah dilakukan melalui penggambaran *Use Case* dengan tahapan terakhir dilakukan penentuan atribut dan perilaku setiap objek dengan deskripsi lebih lanjut yang akan menjadi acuan tahapan perancangan.

1. Identifikasi Objek

Analisis pada tahap awal dilakukan identifikasi objek berdasarkan data identifikasi masalah yang telah dilakukan pada tahap sebelumnya dengan didapatkan siapa saja dan apa saja yang terlibat pada sistem baik itu orang atau aktor, entitas-entitas sistem serta fungsionalitas apa saja yang ada dan dapat dihadirkan maupun dilakukan sistem dalam memenuhi segala kebutuhan pengguna.

2. Ilustrasi Hubungan Antar Objek

Setelah didapatkan dan diidentifikasi apa saja objek yaitu aktor, entitas sistem beserta fungsionalitas dalam sistem, maka selanjutnya dilakukan ilustrasi bagaimana objek berkaitan atau terhubung dengan menggunakan salah satu UML yang umum digunakan dalam penggambaran OOA yaitu diagram *Use Case*. *Use Case* akan memberikan gambaran lengkap kemampuan objek atau aktor serta ruang lingkup fungsionalitas sistem.

3. Spesifikasi Atribut dan Perilaku Objek

Penggambaran diagram *Use Case* perlu dideskripsikan secara spesifik apa saja atribut yang ada serta bagaimana perilaku yang dimiliki objek dengan menggunakan deskripsi detail dari *Use Case* yang telah dirangkai. Karena *Use Case* telah memberikan gambaran lengkap sehingga terdapat penjelasan terkait perilaku atau fungsional dari masing masing aktor dalam sistem.

Tahap Perancangan

Tahapan perancangan pada penelitian ini dilakukan dengan mengacu kepada penerapan metode OOD (*Object-Oriented Design*) yang dilakukan dengan menerjemahkan hasil analisis pada tahap OOA ke dalam bentuk model desain visual yang berfungsi sebagai cetak biru panduan pada pembuatan sistem. Pada tahapan ini harus ditemukan objek yang tepat dan relevan, kemudian dimasukkan kedalam kelas dengan perincian yang tepat, didefinisikan kelas

antarmukanya dan hirarki pewarisan yang selanjutnya ditetapkan hubungan dan kunci dari hubungan antar objek yang berelasi. Model visual dari OOD ini diterjemahkan kedalam bentuk model UML yaitu *Activity Diagram*, *Class diagram*, dan *Sequence Diagram*.

1. Rancang Diagram aktivitas untuk setiap skenario

Pada tahap perancangan yang pertama dilakukan adalah dengan membuat diagram aktivitas untuk setiap skenario sistem yang ada dengan *Activity Diagram* sebagai *output*. *Activity Diagram* memberikan visualisasi terkait *flow* atau aliran aktivitas baik dalam sistem, proses bisnis, alur kerja, dan proses lainnya dimana diagram ini berfokus terhadap rangkaian aktivitas yang dilakukan dan siapa atau apa yang bertanggung jawab terhadap kinerja aktivitas tersebut.

2. Rancang Diagram Kelas Secara Detail

Setelah digambarkan rangkaian flow dengan menggunakan *Activity Diagram* kemudian menggambarkan interaksi antar objek dengan *Sequence Diagram*, selanjutnya dibangun *Class diagram* yang detail untuk memodelkan kelas termasuk atribut, operasi, dan hubungan beserta asosiasinya terhadap kelas lainnya dengan menyediakan tampilan statis atau struktural dari sebuah sistem. Dari diagram ini kita dapatkan rancangan database untuk sistem dengan menggambarkan tabel dengan notasi yang dimiliki.

3. Rancang diagram urutan untuk setiap skenario

Setiap interaksi dari objek dalam alur sistem dapat terlihat dengan jelas menggunakan diagram interaksi yaitu *Sequence Diagram* atau diagram urutan. *Sequence Diagram* menggambarkan interaksi objek selama periode waktu tertentu yaitu *Sequence* atau urutan pesan antara aktor eksternal dengan sistem selama skenario *Use Case*. *Sequence Diagram* digunakan untuk melacak pelaksanaan skenario dengan keunggulan lebih baik dibandingkan *communication diagram* dalam menangkap skenario semantik pada proses awal pengembangan sebelum protokol untuk masing-masing kelas teridentifikasi sehingga lebih mudah membaca pertukaran pesan dengan urutan relatif.

4. Rancang Tampilan Desain Antarmuka

Tahapan terakhir dari perancangan sistem ini adalah dengan melanjutkan ke detail desain yaitu perancangan tampilan desain antarmuka (*user interface*). Perancangan *User Interface* (UI) aplikasi dilakukan untuk mengatur tampilan perangkat yang dilihat pengguna sehingga interaksi pengguna dengan sistem nyaman dan bersesuaian (*user friendly*). Rancangan antarmuka (*interface*)

dimulai dengan pembuatan *Wireframe Low-Fidelity* dilanjutkan *high-fidelity* dari desain tampilan sehingga dapat menjadi acuan perancangan antarmuka yang kemudian dilanjutkan pembuatan desain antarmuka secara lengkap menjadi sebuah *prototype* sehingga dapat membantu developer sistem memahami rancangan yang telah dibuat untuk diimplementasikan.

Tahap Evaluasi

Evaluasi merupakan kegiatan yang dilakukan untuk memastikan bahwa hasil rancangan yang dibuat telah berkesesuaian dengan kebutuhan yang harus terpenuhi. Tahap Evaluasi penelitian ini dilakukan dengan menggunakan metode *Requirement Traceability Matrix* (RTM). RTM menurut (Devi et al., 2019) dapat membantu pengembang melacak serta menelusuri kebutuhan yang telah terdefinisi dari awal perancangan hingga akhir. RTM menguji dengan memeriksa keruntutan dari kebutuhan *stakeholder*, fungsional dan non-fungsional, hingga perancangan sistem yang dibuat sehingga dapat dipastikan keberadaan setiap proses analisis dan perancangan sistem telah tersusun sesuai dengan kebutuhan pemangku kepentingan yang dapat dilacak dan tidak mengandung kode kebutuhan yang sama (Nasikhin et al., 2019). RTM yang digunakan adalah RTM model *functional tracing*. RTM dibuat dengan bentuk tabel matrix yang meringkas kebutuhan (*requirements*) pemangku atau pengguna sistem sebagai persyaratan pertama yang diletakkan pada baris pertama dan kolom pertama kemudian ke elemen lainnya yaitu analisis dan desain yang telah dirancang dimana setiap elemen ini harus diberikan *unique identifier* (ID) yang digunakan sebagai referensi pengenalan matrix. Evaluasi awal dalam tabel dengan menghubungkan setiap elemen sistem kepada *requirements* (persyaratan) yang sesuai dalam tabel dilanjutkan dengan menghubungkan setiap persyaratan sistem ke elemen sistem yang sesuai dengan tabel kemudian dievaluasi hubungan tersebut untuk memastikan semua persyaratan terpenuhi dan tidak ada elemen yang tidak diperlukan (Westfall, 2006).

Requirement	Specific aspect of the system or its environment							Aii
	A01	A02	A03	A04	A05			
R01			✓		✓			
R02	✓		✓					
R03	✓			✓				✓
R04		✓			✓			
R05	✓	✓		✓				✓
Rnn	✓		✓					

Gambar 2. Tabel *Requirement Traceability Matrix* (RTM)
(Sumber : (S. Pressman, 2000))

3.5 Jadwal Penelitian

Berikut adalah jadwal yang akan dilaksanakan pada penelitian ini :

Tabel 7. Jadwal Penelitian

No.	Kegiatan	Jun				Jul				Ags				Sept			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Identifikasi Masalah	■	■														
2.	Pengumpulan Data		■	■	■												
3.	Analisis Kebutuhan				■	■	■										
4.	Rancangan Sistem					■	■	■	■	■	■	■	■	■	■		
5.	Pengujian Rancangan Sistem									■	■	■	■	■	■	■	
6.	Penyusunan Laporan Akhir														■	■	■

IV. HASIL DAN PEMBAHASAN

4.1 Tahap Perencanaan

Dalam pelaksanaan analisis dan perancangan sebuah pengembangan sistem informasi, terdapat perencanaan yang harus dilakukan dimana dipersiapkan segala dokumen kebutuhan dengan melakukan berbagai rangkaian dalam perolehan data dan informasi permasalahan yang terjadi di perumahan klaster Villa Gading Mayang dengan metode pengumpulan secara observasi, wawancara, serta studi literatur secara mendalam. Berdasarkan hal ini, dilakukan identifikasi masalah guna mendapatkan solusi yang tepat dalam penanganan setiap hambatan dan kesulitan yang dialami baik dari pengelola maupun warga perumahan Villa Gading Mayang Kota Jambi. Permasalahan yang teridentifikasi diantaranya adalah sebagai berikut :

a. Belum adanya sistem informasi khusus pada pengelolaan perumahan

Perumahan klaster Villa Gading Mayang sebagai suatu hunian tinggal warga Kota Jambi memerlukan kehadiran suatu wadah yang dapat menjadi tempat penyimpanan dan pengelolaan informasi bersama warga perumahan yang penting agar setiap informasi yang telah ada tetap terjaga apalagi dengan banyaknya kegiatan yang perlu terdokumentasi sebagai laporan ketua RT kepada penghuni perumahan sebagai pengelola. Dengan berkembangnya teknologi dalam mengelola data dan informasi maka wadah sistem informasi dapat dihadirkan untuk menjadi sebuah solusi penanganan setiap permasalahan pengelolaan yang ada, dimana sistem informasi ini diperuntukkan untuk setiap bagian pengelola dan penghuni perumahan dalam memenuhi kebutuhan bersama internal perumahan terkait keseluruhan kegiatan seperti pembayaran iuran, permohonan persuratan, pendataan keuangan dan inventaris barang, dan sebagainya.

b. Tidak adanya informasi dan pengelolaan data terkait pembayaran iuran

Warga penghuni perumahan Villa Gading Mayang memiliki program iuran rutin yang perlu dibayarkan setiap bulannya kepada bendahara perumahan guna untuk membiayai fasilitas bersama seperti keamanan, kebersihan, dan pembelian barang perumahan seperti peralatan jaga pada pos satpam, sapu, tong sampah, dan sebagainya. Iuran yang perlu dikeluarkan bagi setiap hunian rumah berbeda bergantung kepada tipe rumah yang dihuni dengan klasifikasi dimulai dari 50.000 rupiah hingga 175.000 rupiah per bulannya. Penagihan pembayaran dilakukan secara langsung maupun melalui aplikasi whatsapp grup dan pribadi sehingga sering tercecer dan hilangnya data ketika akan dihimpun

setiap 6 bulan kedalam Microsoft Excel. Apalagi sebagian besar warga perumahan sering menumpuk pembayaran selama 2 bulan, 3 bulan, hingga 6 bulan sehingga seringkali terjadi resiko salah pendataan dengan cara manual ini. Pemanfaatan sistem informasi pada internal perumahan klaster Villa Gading Mayang dapat meminimalisir risiko kekeliruan terkait pembayaran dan pencatatan iuran warga hunian perumahan.

c. Kebutuhan akan transparansi dana perumahan

Pengelola perumahan klaster Villa Gading Mayang dalam pelaksanaan kegiatan iuran dan dana lainnya tentunya perlu meyakinkan para warganya bahwa dana yang ditagihkan dan dikeluarkan oleh warga digunakan sesuai dengan kebutuhan perumahan. Pengelola saat ini belum ada memberikan rincian terkait transparansi pengeluaran dan pemasukan dana baik iuran dan lainnya yang telah dibayarkan warga perumahan sehingga sering menimbulkan keraguan dan ketidakpercayaan warga terhadap pengelola yang dapat menimbulkan masalah besar berkelanjutan jika dibiarkan. Transparansi terkait dana perumahan sangat krusial untuk diperlihatkan kepada warga dimana dengan kehadiran sistem informasi dapat mewujudkan hal ini sehingga memberikan rasa keamanan dan kenyamanan bagi warga hunian perumahan.

d. Informasi terkait rumah hunian yang ditempati

Data terkait rumah yang ditempati, kosong, ataupun ditinggal sementara tentunya perlu diketahui oleh pengelola perumahan klaster Villa Gading Mayang mengingat kegiatan internal seperti penagihan iuran, dan laporan pendataan warga perlu diberikan oleh Ketua RT sebagai pengelola utama. Tidak tersedianya data terkait hunian rumah ini seringkali menimbulkan permasalahan Ketika pendataan dimana tentunya pengelola tidak dapat mengingat diluar kepala satu persatu sehingga sering keliru saat membuat laporan penagihan iuran. Adanya sistem informasi pada perumahan Villa Gading Mayang Kota Jambi akan sangat membantu dan mempermudah perolehan dan pendataan ini secara cepat dan tepat.

e. Permohonan surat pengantar RT

Ketika warga hunian perumahan akan mengurus hal terkait administrasi ke kantor dukcapil dan sebagainya, pada umumnya memerlukan surat pengantar dari RT terlebih dahulu untuk selanjutnya diserahkan secara pribadi oleh warga kepada badan kepengurusan jenjang yang lebih tinggi dalam pemenuhan administrasi tersebut. Warga akan mengajukan permintaan surat pengantar kepada ketua RT yang kemudian surat akan dibuat oleh sekretaris yang selanjutnya akan diberikan oleh ketua RT pada warga kembali untuk pengurusan selanjutnya. Kendati begitu, seringkali permohonan surat pengantar

ini memakan waktu yang lama dikarenakan *human error* dari sisi pengelola seperti terlupanya data dan hal lainnya yang diperlukan dalam pembuatan surat membuat pengurusan administratif ini tidak efektif. Oleh karena itu, sistem informasi diperlukan guna untuk mempermudah permohonan surat pengantar RT yang dapat berjalan secara efektif dan efisien untuk pemenuhan kebutuhan administrasi internal warga perumahan.

4.2 Tahap Analisis

Berkaitan dalam perancangan sebuah sistem informasi dengan menggunakan metode OOAD (*Object Oriented Analysis and Design*), tahapan pertama yang harus dilakukan adalah OOA (*Object Oriented Analysis*) yaitu melakukan analisis dengan mengumpulkan berbagai informasi yang dapat mengidentifikasi setiap kebutuhan sistem dan pengguna secara menyeluruh, dimana kemudian hasilnya dapat mendeskripsikan setiap fungsi secara konseptual. Adapun rangkaian yang terdapat di dalam tahapan analisis ini diantaranya yaitu identifikasi objek, dilanjutkan dengan mengilustrasikan hubungan antar objek, kemudian spesifikasikan atribut dan perilaku tiap objek.

Identifikasi Objek

Objek pada metode OOAD adalah setiap hal baik data, proses, dan setiap operasi sistem yang mewakili keadaan dunia nyata diantaranya orang, benda, transaksi, dan setiap kejadian yang terjadi pada ruang lingkup operasi bisnis sistem (B. Shelly & J. Rosenblatt, 2012). Dalam analisis suatu pengembangan sistem, perlu dilakukannya identifikasi terhadap objek diantaranya analisis kebutuhan sistem, actor, beserta detail entitas sistem yaitu perilaku dan atribut dari objek sistem sehingga pada tahap ini terdiri atas 2 tahapan yaitu analisis kebutuhan sistem dan Analisis aktor beserta entitas sistem.

1. Analisis Kebutuhan Sistem

Analisis kebutuhan merupakan tahapan identifikasi objek yang perlu dilakukan dalam perancangan dan pengembangan sebuah sistem dengan tujuan memperoleh seluruh informasi terkait kemampuan, syarat ataupun kriteria yang harus terpenuhi oleh sistem informasi sehingga apa yang diharapkan pengguna ataupun pengelola perumahan dari sistem informasi dapat terwujud. Adapun analisis kebutuhan sistem (*System Requirement Analysis*) dalam prosesnya terbagi atas dua yaitu analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional. Analisis kebutuhan fungsional berisikan seluruh proses maupun layanan yang dapat dilakukan atau disediakan oleh sistem, sedangkan analisis kebutuhan non fungsional berisikan properti, karakteristik yang dimiliki

dan dibutuhkan oleh sistem untuk dapat memenuhi kinerja operasional sistem mencakup perangkat lunak (*software*) dan perangkat keras (*hardware*) sistem. Berkaitan dengan hal ini, untuk mempersingkat penulisan maka kebutuhan sistem (*System Requirement*) fungsional akan ditulis dengan kode SR dengan penomoran dimulai dari angka 1 secara berurutan (contoh: SR-1, SR-2, dan seterusnya). Kebutuhan fungsional dalam analisis dan perancangan sistem informasi pengelolaan perumahan klaster Villa Gading Mayang (SIPGAMA) Kota Jambi berbasis *website* dapat dilihat pada tabel berikut :

Tabel 8. Analisis Kebutuhan Fungsional Sistem

No.	Kode	Deskripsi Kebutuhan Fungsional
1.	SR-1	Sistem dapat memberikan akses kepada pengguna untuk <i>login</i> kedalam sistem menggunakan <i>username</i> dan <i>password</i> yang valid berkesesuaian dengan data KK Penduduk bagi warga perumahan, dan yang telah dibuat oleh Administrator bagi pengelola perumahan klaster Villa Gading Mayang Kota Jambi, serta mampu memberikan akses pengguna untuk keluar dari sistem
2.	SR-2	Sistem mampu menampilkan informasi terkait warga perumahan, rumah yang telah ditempati beserta tipe perumahan.
3.	SR-3	Sistem mampu memberikan fitur dan informasi terkait data inventaris barang meliputi nama dan jenis barang, jumlah, serta kondisi barang perumahan Villa Gading Mayang Kota Jambi
4.	SR-4	Sistem dapat memberikan informasi terkait permohonan surat pengantar RT ruang lingkup internal perumahan untuk memenuhi keperluan pengurusan administrasi internal warga perumahan klaster Villa Gading Mayang Kota Jambi
5.	SR-5	Sistem dapat menyediakan fitur permohonan surat pengantar RT untuk memenuhi keperluan pengurusan administrasi internal warga perumahan klaster Villa Gading Mayang Kota Jambi
6.	SR-6	Sistem dapat menyediakan fitur pembayaran iuran perumahan yang berkesesuaian dengan kebutuhan agar tercapainya kemudahan dan keefektifan transaksi internal perumahan klaster Villa Gading Mayang Kota Jambi

7.	SR-7	Sistem mampu memberikan informasi terkait riwayat pembayaran iuran, laporan pengeluaran dan pemasukan, serta rekapitulasi dana internal perumahan perumahan Villa Gading Mayang Kota Jambi
8.	SR-8	Sistem memungkinkan sekretaris untuk memverifikasi pemenuhan permohonan surat pengantar untuk keperluan administrasi internal berkelanjutan warga perumahan klaster Villa Gading Mayang Kota Jambi
9.	SR-9	Sistem memungkinkan bendahara untuk memverifikasi pembayaran iuran warga perumahan klaster Villa Gading Mayang Kota Jambi
10.	SR-10	Sistem dapat menampilkan informasi terkait transparansi laporan keuangan berisikan pengeluaran, pemasukan, beserta rekapitulasi keuangan perumahan

Adapun kebutuhan non fungsional dari Sistem Informasi Pengelolaan Perumahan klaster Villa Gading Mayang (SIPGAMA) Kota Jambi berbasis *website* diantaranya adalah sebagai berikut :

- a. Sistem dapat berjalan pada beberapa *web browser* seperti *Internet Explorer*, *Chrome*, *Microsoft Edge*, dan *Mozilla Firefox*.
- b. Perangkat yang digunakan dapat terhubung ke jaringan koneksi internet untuk menyediakan informasi secara *realtime* dan tersinkronisasi
- c. Sistem dapat digunakan dengan mudah oleh pengguna dikarenakan *user interface* (UI) yang dibangun sesuai dengan kebutuhan dan kenyamanan pengguna (*user friendly*).

2. Analisis Aktor dan Entitas Sistem

Setelah didapatkannya spesifikasi seluruh kebutuhan fungsional dan non fungsional sistem, selanjutnya dapat kita identifikasikan aktor dan keseluruhan entitas perannya yang akan dilakukan didalam sistem informasi pengelolaan perumahan klaster Villa Gading Mayang Kota Jambi.

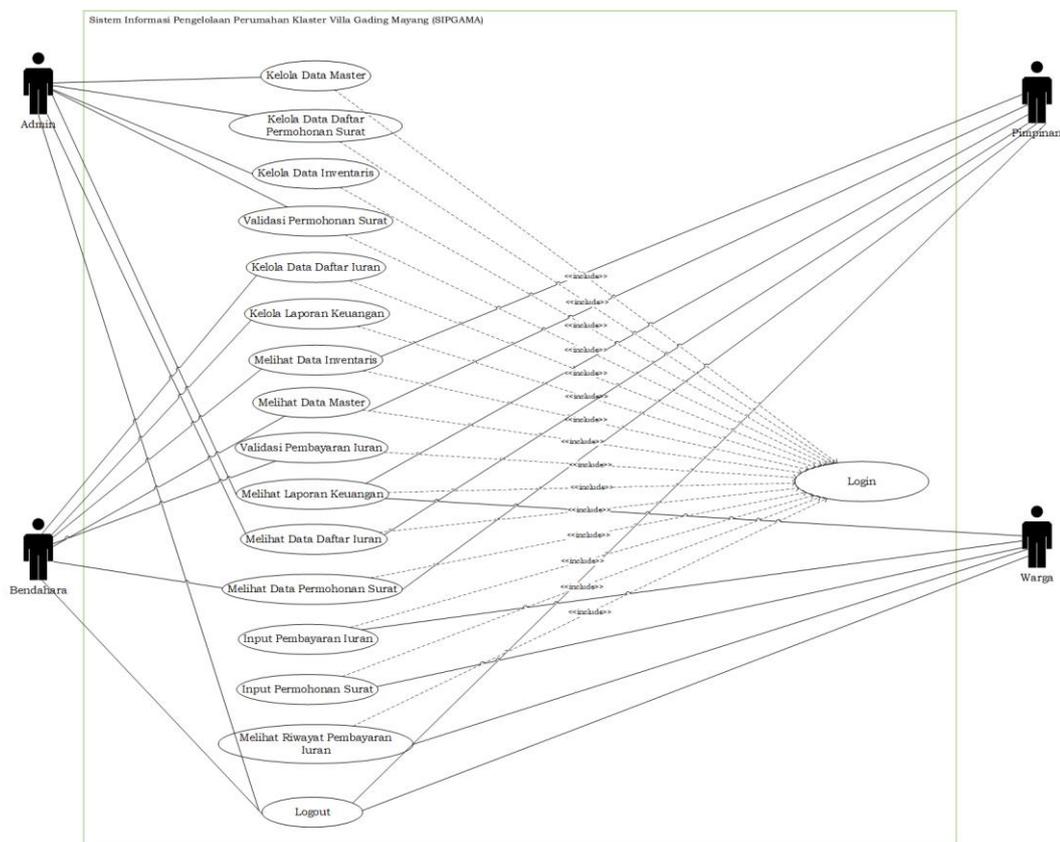
Tabel 9. Tabel Peran Entitas Sistem

No.	Aktor	Peran Entitas
1.	Admin	Admin yaitu sekretaris perumahan merupakan aktor yang menggunakan sistem informasi untuk mengelola data master berisikan warga, rumah, data tipe rumah, kehadiran pemilik rumah, data inventaris barang, metode pembayaran iuran, serta data permohonan surat pengantar. Admin juga

	menggunakan sistem informasi untuk melakukan validasi permohonan surat pengantar RT dari warga perumahan. Admin memiliki akun dan profil serta dapat melihat laporan keuangan, daftar iuran dan validasi pembayaran iuran.
2. Pimpinan	Pimpinan adalah ketua RT yang merupakan aktor yang menggunakan sistem informasi untuk melihat informasi mengenai data master berisikan data warga dan rumah hunian perumahan, inventaris barang, serta laporan keuangan perumahan. Pemimpin juga dapat melihat grafik keuangan perumahan klaster Villa Gading Mayang Kota Jambi. Pemimpin akan memiliki akun dan profil.
3. Bendahara	Bendahara merupakan aktor yang menggunakan sistem informasi dapat mengelola terkait keuangan perumahan diantaranya kelola data pembayaran iuran warga dan mengelola laporan pengeluaran, pemasukan, beserta rekapitulasi keuangan. Bendahara dapat dapat melakukan validasi pembayaran iuran dari warga perumahan. Bendahara juga dapat melihat data warga dan rumah yang ditempati beserta tipe rumah, serta data inventaris barang perumahan klaster Villa Gading Mayang Kota Jambi. Bendahara akan memiliki akun dan profil.
4. Warga	Warga merupakan aktor yang menggunakan sistem informasi untuk menginput iuran pembayaran yang telah dibayarkan kedalam sistem. Warga juga dapat menginput permohonan surat pengantar RT yang diperlukan untuk pengurusan administrasi. Warga dapat melihat data pribadi kependudukannya dan hunian rumah, riwayat pembayaran iuran, dan laporan keuangan perumahan klaster Villa Gading Mayang Kota Jambi. Warga akan memiliki akun dan profil.

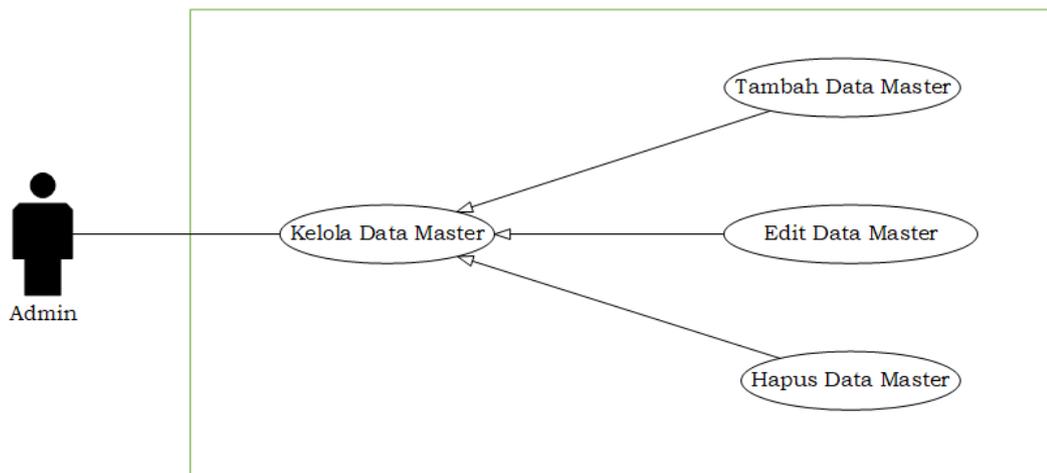
Ilustrasi Hubungan Antar Objek

Setelah teridentifikasi keseluruhan objek dari permasalahan yang diangkat, selanjutnya masuk pada tahap ilustrasi hubungan antar objek sistem dimana pada metode OOA akan digambarkan menggunakan UML *Use Case Diagram*.

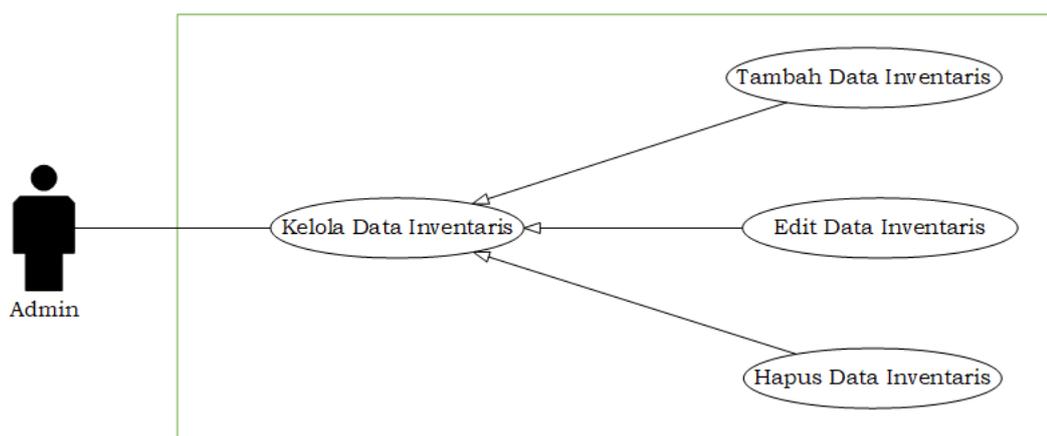


Gambar 3. Use Case Diagram Umum

Gambar 3 diatas merupakan ilustrasi *Use Case* sistem secara keseluruhan dari sistem informasi pengelolaan perumahan kluster Villa Gading Mayang dimana terlihat 4 aktor yang terdiri dari Admin sebagai pengelola utama sistem, pimpinan yaitu ketua RT, bendahara, dan warga perumahan Villa Gading Mayang Kota Jambi. Terlihat setiap aktor memiliki peran entitas dan hak akses yang berbeda sesuai dengan definisi kebutuhan dan spesifikasi yang telah diperoleh dari hasil analisis sebelumnya, Selanjutnya *Use Case* akan dijabarkan menjadi beberapa *Use Case* sesuai dengan spesifikasi fungsi entitas yang dimiliki oleh setiap aktor.

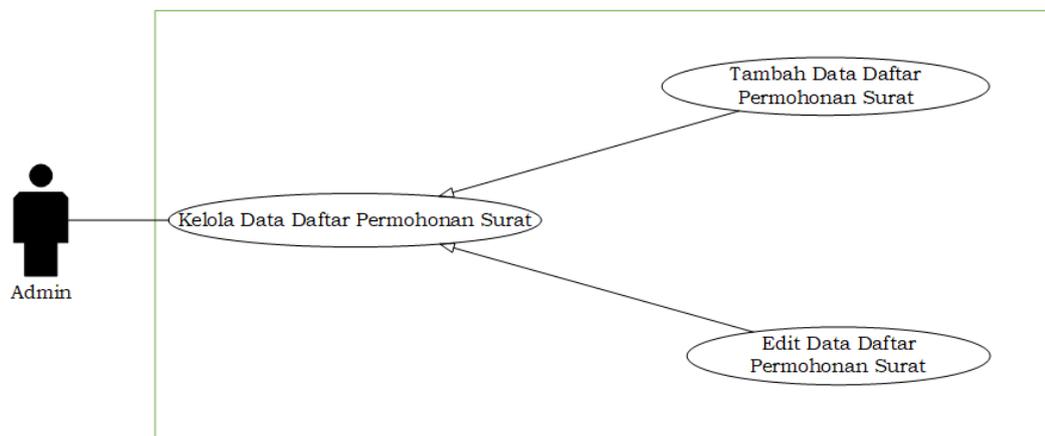
a. *Use Case Diagram* Kelola Data**Use Case Diagram Kelola Data Master****Gambar 4.** *Use Case Diagram* Kelola Data Master (Admin)

Pada Gambar 4 ditampilkan *Use Case Diagram* kelola data master yang didalamnya terdapat data warga, data rumah, data tipe rumah, dan metode pembayaran yang hanya dapat dilakukan oleh aktor pengelola utama yaitu admin. Dalam kelola data mater, admin dapat melakukan 3 aktivitas utama yaitu tambah data master, edit data master, dan hapus data master.

Use Case Diagram Kelola Data Inventaris**Gambar 5.** *Use Case Diagram* Kelola Data Inventaris (Admin)

Pada Gambar 5 ditampilkan *Use Case Diagram* kelola data inventaris yang hanya dapat dilakukan oleh aktor pengelola utama yaitu admin. Dalam kelola data inventaris, admin dapat melakukan 3 aktivitas utama yaitu tambah data inventaris, edit data inventaris, dan hapus data inventaris.

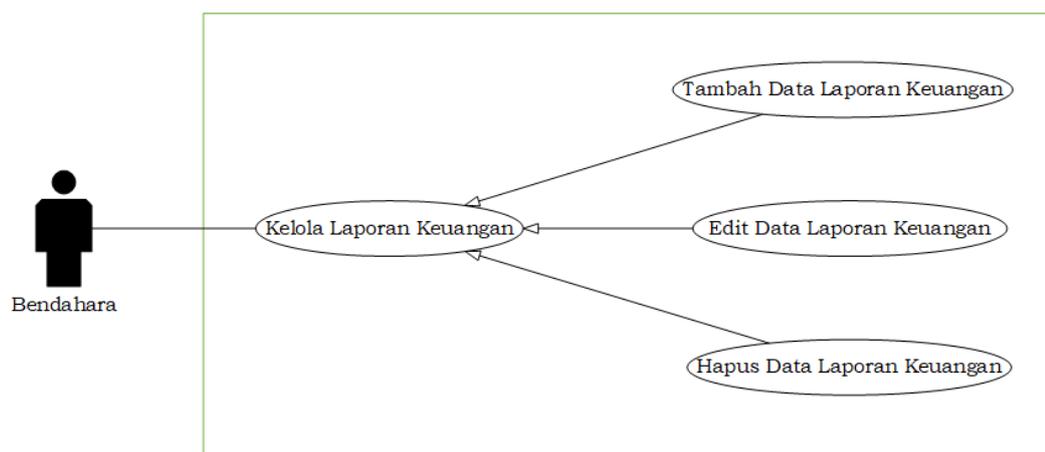
Use Case Diagram Kelola Data Surat



Gambar 6. Use Case Diagram Kelola Data Daftar Permohonan Surat (Admin)

Pada Gambar 6 ditampilkan *Use Case Diagram* kelola data daftar permohonan surat yang hanya dapat dilakukan oleh aktor pengelola utama yaitu admin. Dalam kelola data permohonan surat, admin dapat melakukan 2 aktivitas utama yaitu tambah data daftar permohonan surat dan edit data daftar permohonan surat.

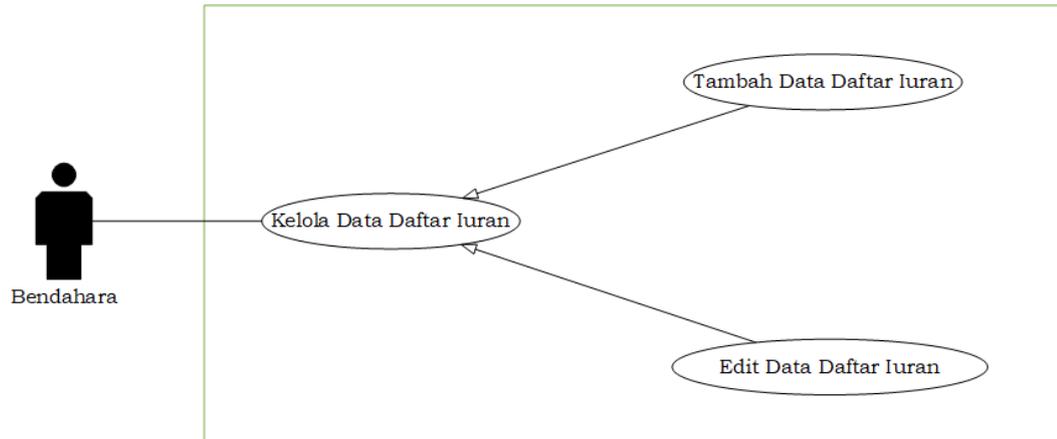
Use Case Diagram Kelola Laporan Keuangan



Gambar 7. Use Case Diagram Kelola Laporan Keuangan (Bendahara)

Pada Gambar 7 ditampilkan *Use Case Diagram* kelola laporan keuangan yang hanya dapat dilakukan oleh bendahara. Dalam kelola laporan keuangan, bendahara dapat melakukan 3 aktivitas utama yaitu tambah data laporan keuangan, edit data laporan keuangan, dan hapus data laporan keuangan.

Use Case Diagram Kelola Data Daftar Iuran

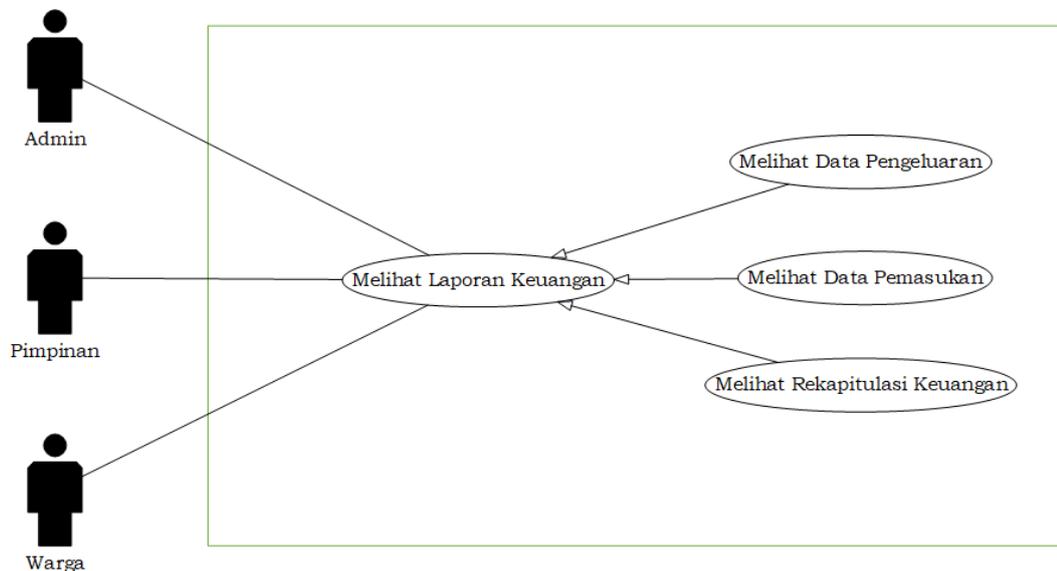


Gambar 8. Use Case Diagram Kelola Data Daftar Iuran (Bendahara)

Pada Gambar 8 ditampilkan *Use Case Diagram* kelola data pembayaran iuran yang hanya dapat dilakukan oleh aktor pengelola keuangan yaitu bendahara. Dalam kelola data daftar iuran, bendahara dapat melakukan 2 aktivitas utama yaitu tambah data daftar iuran dan edit data daftar iuran.

b. Use Case Diagram Melihat Data

Use Case Diagram Melihat Laporan Keuangan

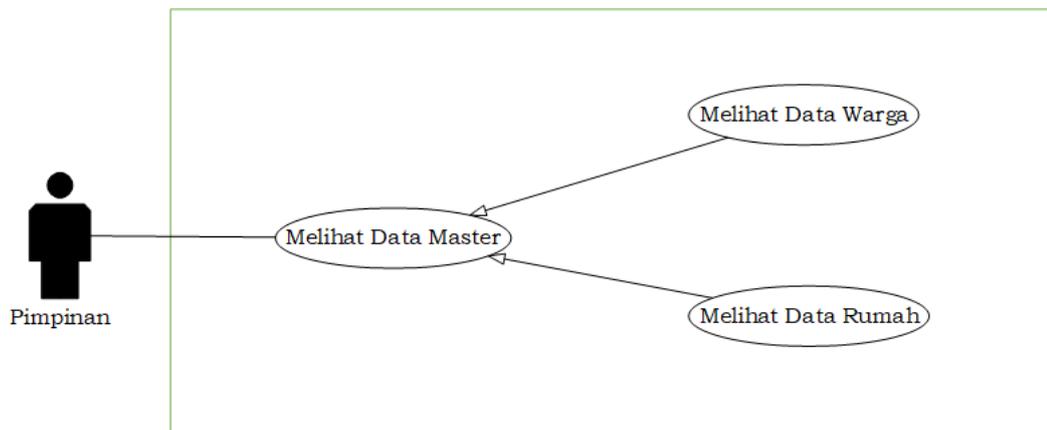


Gambar 9. Use Case Diagram Melihat Laporan Keuangan

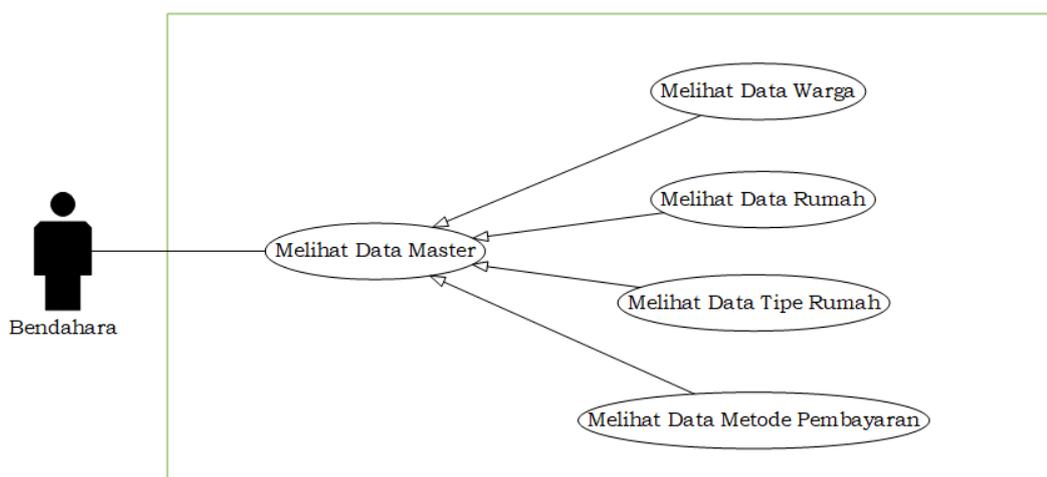
Pada Gambar 9 diatas, ditampilkan *Use Case Diagram* aktivitas melihat laporan keuangan yang dapat dilakukan dengan pemberian hak akses sebatas melihat laporan keuangan kepada 3 aktor sistem yaitu admin, pimpinan, dan warga. Pada menu laporan keuangan dapat dilihat 3 submenu diantaranya data

pengeluaran, pemasukan, dan total rekapitulasi keuangan perumahan klaster Villa Gading Mayang Kota Jambi.

Use Case Diagram Melihat Data Master



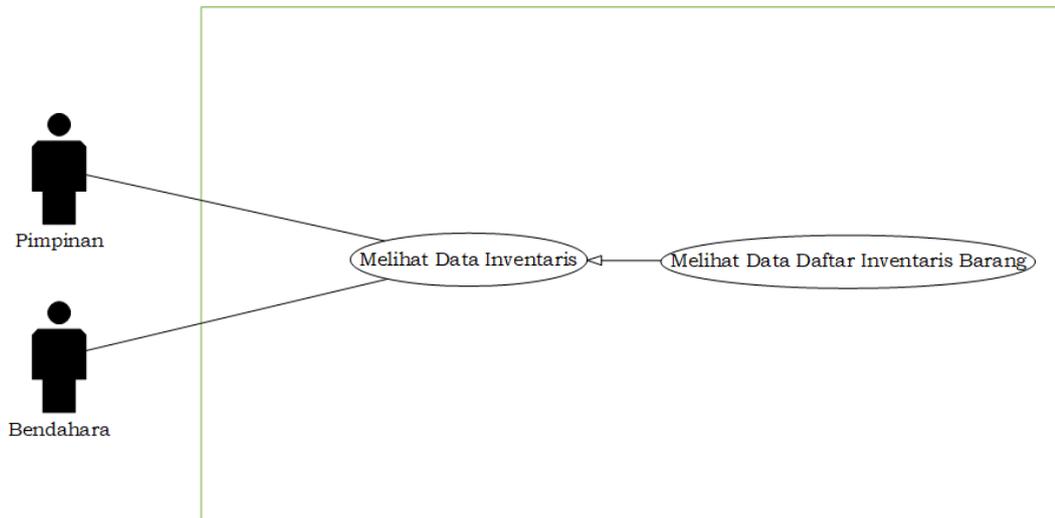
Gambar 10. Use Case Diagram Melihat Data Master (Pimpinan)



Gambar 11. Use Case Diagram Melihat Data Master (Bendahara)

Pada gambar 10 dan 11 diatas menunjukkan akses sebatas melihat data master perumahan klaster Villa Gading Mayang yang diberikan kepada pimpinan dan bendahara dimana aktor pimpinan dapat melihat data kependudukan warga perumahan seperti Nomor NIK, data diri warga, dan sebagainya, serta data rumah warga tersebut termasuk didalamnya status hunian apakah sedang ditempati. Sedangkan untuk aktor bendahara dapat melihat keseluruhan data master diantaranya meliputi data warga, rumah, tipe rumah, beserta metode pembayaran.

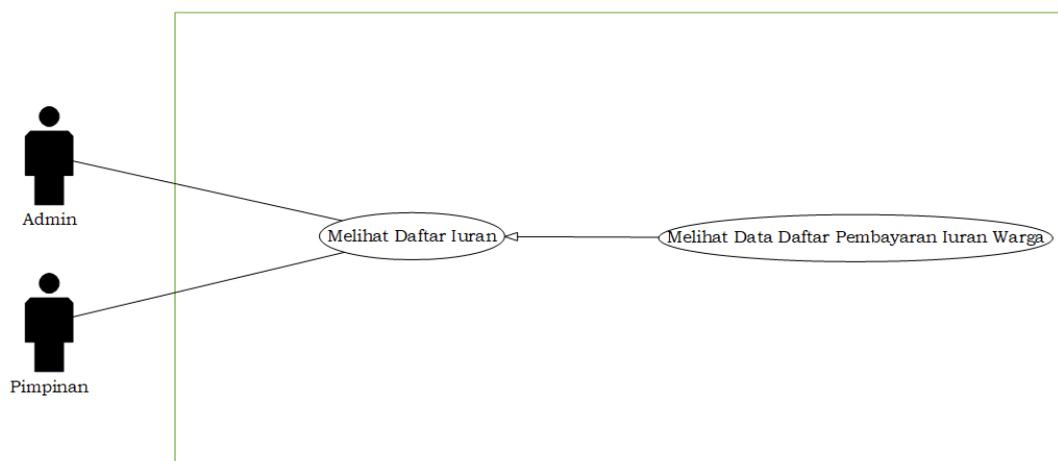
Use Case Diagram Melihat Data Inventaris



Gambar 12. Use Case Diagram Melihat Data Inventaris (Pimpinan, Bendahara)

Pada Gambar 12 diatas, ditampilkan *Use Case Diagram* aktivitas melihat data inventaris yang dapat dilakukan dengan pemberian hak akses sebatas melihat data inventaris kepada 2 aktor sistem yaitu pimpinan dan bendahara. Pada menu inventaris dapat dilihat data terkait daftar inventaris barang yang dimiliki perumahan klaster Villa Gading Mayang Kota Jambi beserta keterangan status dari kondisi barang inventaris tersebut.

Use Case Diagram Melihat Daftar Iuran

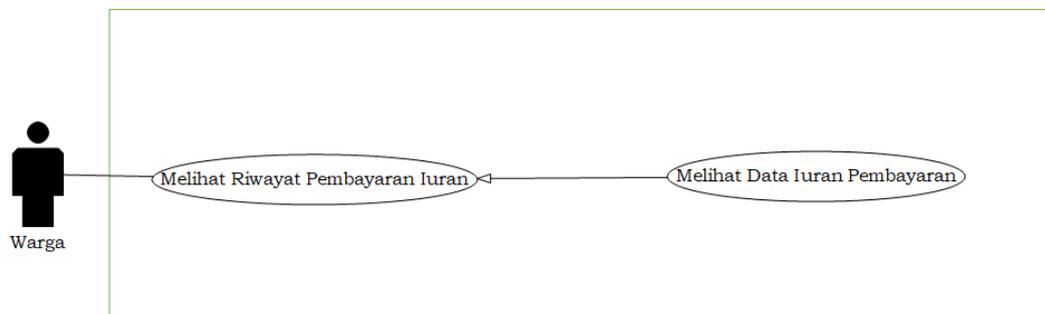


Gambar 13. Use Case Diagram Melihat Daftar Iuran (Admin, Pimpinan)

Pada Gambar 13 diatas, ditampilkan *Use Case Diagram* aktivitas melihat riwayat pembayaran iuran yang dapat dilakukan dengan pemberian hak akses sebatas melihat daftar iuran kepada aktor. Pada aktivitas ini admin dan

pimpinan dapat melihat data iuran pembayaran yang telah dilakukan oleh warga perumahan berisikan tanggal, jumlah pembayaran, metode pembayaran dan keterangan bulan yang dibayarkan.

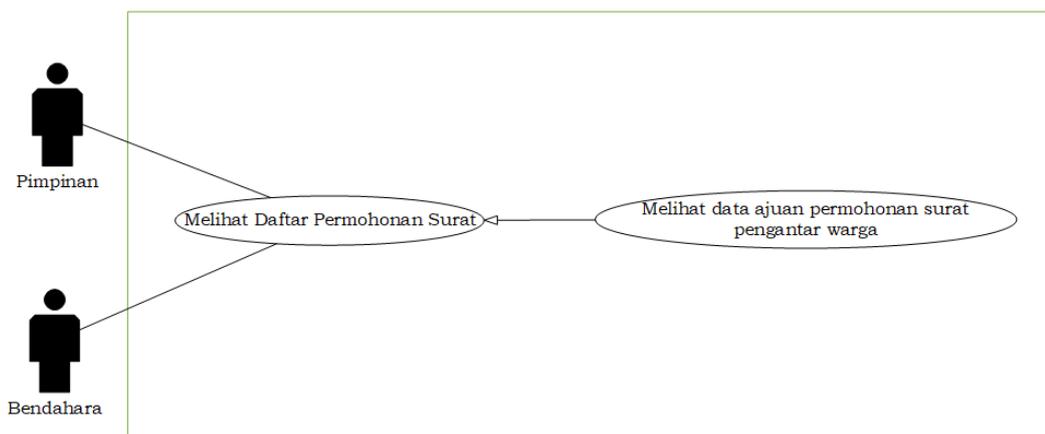
Use Case Diagram Melihat Riwayat Pembayaran Iuran



Gambar 14. Use Case Diagram Melihat Riwayat Pembayaran Iuran (Warga)

Pada Gambar 14 diatas, ditampilkan *Use Case Diagram* aktivitas melihat riwayat pembayaran iuran yang dapat dilakukan dengan pemberian hak akses sebatas melihat riwayat pembayaran kepada warga. Pada aktivitas ini warga dapat melihat data iuran pembayaran yang telah dilakukan berisikan tanggal, jumlah pembayaran, metode pembayaran dan keterangan bulan yang dibayarkan beserta status validasi pembayaran.

Use Case Diagram Melihat Daftar Permohonan Surat



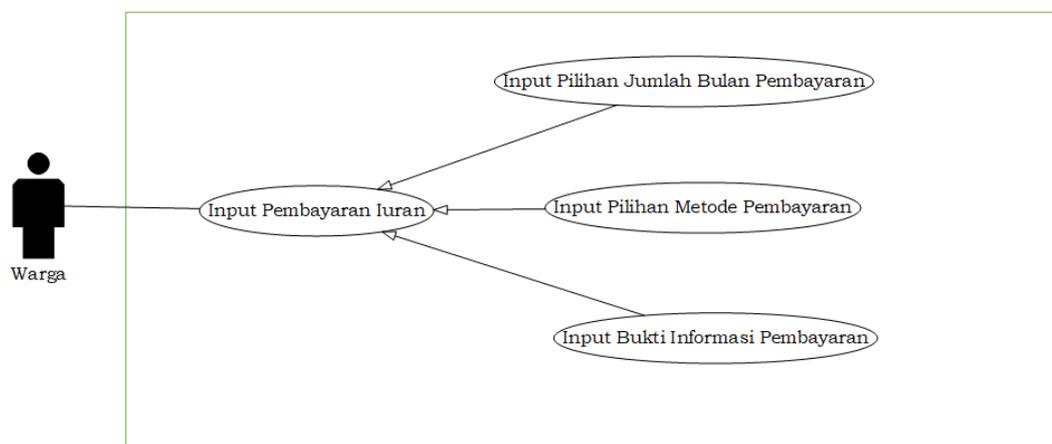
Gambar 15. Use Case Diagram Melihat Daftar Permohonan Surat (Pimpinan, Bendahara)

Pada Gambar 15 diatas, ditampilkan *Use Case Diagram* aktivitas melihat daftar permohonan surat yang dapat dilakukan dengan pemberian hak akses sebatas melihat data permintaan permohonan surat. Pada aktivitas pimpinan dan bendahara dapat melihat data surat yang diminta warga berisikan nama

warga yang meminta surat dan jenis surat keterangan yang diminta, serta keterangan terkait pengambilan surat.

c. *Use Case Diagram Input Data*

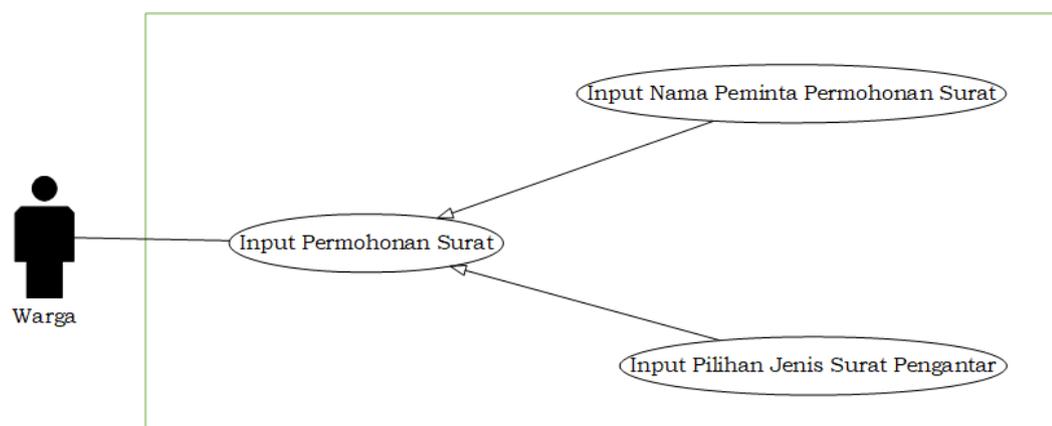
Use Case Diagram Input Pembayaran Iuran



Gambar 16. *Use Case Diagram Input Pembayaran Iuran (Warga)*

Pada gambar 16 diperlihatkan *Use Case input* pembayaran iuran yang dilakukan oleh warga untuk memasukkan data pembayaran yang telah dilakukan. Warga akan meng-*inputkan* pilihan bulan sebanyak bulan yang akan dibayarkan, apakah 1 bulan dan seterusnya, kemudian warga akan inputkan pilihan metode pembayaran yaitu pembayaran transfer rekening, *e-wallet*, atau manual. Bagi pembayaran dengan metode selain manual selanjutnya perlu meng-*inputkan* bukti pembayaran yang telah dilakukan untuk selanjutnya dapat divalidasi oleh bendahara.

Use Case Diagram Input Data Permohonan Surat



Gambar 17. *Use Case Diagram Input Permohonan Surat (Warga)*

Pada gambar 16 diperlihatkan *Use Case Diagram input* permohonan surat pengantar yang dilakukan oleh warga perumahan klaster Villa Gading Mayang Kota Jambi. Dalam permohonan surat, warga akan meng-*inputkan* nama diri yang dilanjutkan dengan meng-*inputkan* pilihan jenis surat pengantar apa yang ingin diminta apakah surat pengantar keterangan tidak mampu, surat pengantar domisili dan lain sebagainya untuk kemudian dibuat dan divalidasi oleh administrator yaitu sekretaris perumahan.

Spesifikasi Atribut dan Perilaku Objek

Tahapan terakhir dari proses analisis OOA dalam metode OOAD adalah dengan melakukan spesifikasi atau penjabaran detail terkait atribut dan perilaku objek yang saling berinteraksi di dalam sistem. Adapun spesifikasi ini dilakukan dengan mendeskripsikan keseluruhan definisi dan skenario *Use Case* dimana *Use Case* yang telah dibuat sebelumnya telah menggambarkan dan mendefinisikan seluruh kebutuhan serta atribut dan perilaku dari objek sistem.

a. Definisi *Use Case*

Berikut akan ditampilkan definisi *Use Case* dari sistem informasi pengelolaan perumahan klaster Villa Gading Mayang, dimana setiap *Use Case* akan diberikan kode UC dan nomor berurutan yaitu (UC-1, UC-2, dst...).

Tabel 10. Definisi *Use Case*

No.	Kode	<i>Use Case</i>	Deskripsi	Hak Akses
1.	UC-1	Melakukan <i>Login</i>	Fungsi yang dilakukan untuk masuk kedalam sistem informasi agar dapat mengakses setiap menu sistem sesuai dengan hak akses yang diberikan untuk masing-masing <i>role</i> aktor sistem.	Admin, Pimpinan, Bendahara, Warga.
2.	UC-2	Melakukan <i>Logout</i>	Fungsi yang dilakukan untuk aktor sistem keluar dari sistem	Admin, Pimpinan, Bendahara, Warga.
3	UC-3	Mengelola Data Master	Aktivitas yang dilakukan oleh aktor sekretaris sebagai pengelola sistem untuk melakukan kelola	Admin

			terhadap data master meliputi aksi menambah, mengubah, dan menghapus data master dalam sistem.	
4.	UC-4	Melihat Data Master	Aktivitas yang dibatasi hak akses hanya sebatas melihat data master perumahan yang terdapat didalam sistem.	Pimpinan, Bendahara
5.	UC-5	Mengelola Data Inventaris	Aktivitas yang dilakukan oleh aktor sekretaris sebagai pengelola untuk melakukan kelola terhadap data inventaris perumahan meliputi aksi menambah, mengubah, dan menghapus data inventaris barang perumahan dalam sistem.	Admin
6.	UC-6	Melihat Data Inventaris	Aktivitas yang dibatasi hak akses hanya sebatas melihat data inventaris barang perumahan yang terdapat didalam sistem.	Pimpinan, Bendahara
7.	UC-7	Input Permohonan Surat	Aktivitas yang dilakukan aktor untuk memasukkan (<i>input</i>) data terkait permohonan surat pengantar RT yang akan diajukan dengan memasukkan nama peminta permohonan surat, dan jenis surat	Warga

			pengantar apa yang akan diminta.	
8.	UC-8	Mengelola Data Daftar Permohonan Surat	Aktivitas yang dilakukan oleh aktor sekretaris sebagai pengelola sistem untuk melakukan kelola terhadap data daftar permohonan surat pengantar perumahan meliputi aksi menambah, mengubah, dan menghapus data permohonan dalam sistem.	Admin
9.	UC-9	Melihat Data Daftar Permohonan Surat	Aktivitas yang dibatasi hak akses hanya sebatas melihat data daftar permohonan surat pengantar RT perumahan yang terdapat didalam sistem.	Pimpinan, Bendahara
10.	UC-10	Input Pembayaran Iuran	Aktivitas yang dilakukan aktor untuk memasukkan (<i>input</i>) data terkait pembayaran iuran yang dilakukan dengan memasukkan pilihan bulan pembayaran, pilihan metode pembayaran, dan terakhir memasukkan bukti pembayaran yang telah dilakukan.	Warga
11.	UC-11	Mengelola Daftar Iuran	Aktivitas yang dilakukan oleh aktor bendahara sebagai pengelola keuangan untuk	Bendahara

			melakukan kelola terhadap data pembayaran iuran meliputi aksi menambah, mengubah, dan menghapus data daftar iuran perumahan dalam sistem.	
12.	UC-12	Melihat Data Daftar Iuran	Aktivitas yang dibatasi hak akses hanya sebatas melihat data daftar iuran perumahan yang terdapat didalam sistem.	Pimpinan, Bendahara
13.	UC-13	Melihat Riwayat Pembayaran Iuran	Aktivitas yang dibatasi hak akses hanya sebatas melihat data riwayat pembayaran iuran yang telah dilakukan yang terdapat didalam sistem.	Warga
14.	UC-14	Validasi Permohonan Surat	Aktivitas proses memvalidasi atau pengecekan yang dilakukan oleh aktor admin yaitu sekretaris terkait permohonan surat pengantar RT perumahan yang diajukan warga	Admin
15.	UC-15	Validasi Pembayaran Iuran	Aktivitas proses memvalidasi atau pengecekan yang dilakukan oleh aktor bendahara terkait pembayaran iuran perumahan yang di-inputkan warga perumahan.	Bendahara

16.	UC-16	Mengelola Laporan Keuangan	Aktivitas yang dilakukan oleh aktor bendahara sebagai pengelola keuangan untuk melakukan kelola terhadap laporan keuangan meliputi aksi menambah, mengubah, dan menghapus data laporan keuangan perumahan dalam sistem.	Bendahara
17.	UC-17	Melihat Laporan Keuangan	Aktivitas yang dibatasi hak akses hanya sebatas melihat data laporan keuangan perumahan yang terdapat didalam sistem.	Admin, Pimpinan, Warga

b. Skenario *Use Case*

Setelah terdefinisi setiap *Use Case* yang ada, selanjutnya dibuat skenario dari *Use Case* tersebut untuk lebih memperjelas bagaimana perilaku dan atribut setiap objek berinteraksi dengan sistem.

Tabel 11. Skenario *Use Case* Login

Nama Use Case	Login	
ID Use Case	UC-01	
Aktor	Admin, Pimpinan, Bendahara, Warga	
Deskripsi	Dilakukan oleh seluruh aktor untuk autentikasi masuk ke dalam sistem agar dapat mengakses menu sesuai hak akses yang diberikan	
Exception	Login gagal	
Precondition	Aktor mengakses laman <i>website</i>	
Aktor		Sistem
Skenario Normal		
1. Aktor mengakses laman <i>website</i>		
	2. Menampilkan halaman login	
3. Aktor memasukkan <i>username</i> dan <i>password</i> lalu menekan tombol login		

	4. Memvalidasi data yang diinputkan aktor
	5. Data valid, sistem menampilkan halaman <i>dashboard</i>
Skenario Alternatif	
1. Aktor mengakses laman <i>website</i>	
	2. Menampilkan halaman login
3. Aktor memasukkan <i>username</i> dan <i>password</i> lalu menekan tombol login	
	4. Memvalidasi data yang diinputkan aktor
	5. Data <i>username</i> dan <i>password</i> tidak valid, sistem menampilkan peringatan data salah dan menampilkan halaman login kembali
5. Aktor kembali melakukan login	
Post Condition	Aktor berhasil masuk ke dalam sistem dan ditampilkan <i>dashboard</i> sesuai dengan hak aksesnya masing-masing.

Tabel 12. Skenario Use Case Logout

Nama Use Case	Logout	
ID Use Case	UC-2	
Aktor	Admin, Pimpinan, Bendahara, Warga	
Deskripsi	Aktivitas fungsi yang dilakukan untuk aktor dapat keluar dari sistem	
Precondition	Aktor telah login kedalam sistem	
Aktor		Sistem
Skenario Normal		
1. Aktor menekan tombol logout		
		2. Mengeluarkan aktor dari sistem
Post Condition	Aktor keluar dari sistem	

Tabel 13. Skenario *Use Case* Mengelola Data Master

Nama Use Case	Mengelola Data Master
ID Use Case	UC-3
Aktor	Admin
Deskripsi	Aktivitas yang dilakukan oleh aktor pengelola sistem untuk melakukan kelola terhadap data master perumahan meliputi aksi menambah, mengubah, dan menghapus data master dalam sistem.
Exception	Perubahan data yang dikelola tidak tersimpan ke dalam <i>database</i>
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses menu data master	
	2. Sistem menampilkan menu data master
3. Aktor melakukan kelola terhadap data master berupa menambah, mengedit, atau menghapus data	
	4. Sistem menyimpan penambahan dan perubahan data master ke dalam <i>database</i> sistem
	5. Menampilkan pesan perubahan kelola data master telah berhasil dan menampilkan kembali halaman data master
Skenario Alternatif	
1. Aktor mengakses menu data master	
	2. Sistem menampilkan menu data master
3. Aktor melakukan kelola terhadap data master berupa menambah, mengedit, atau menghapus data	

	4. Sistem gagal menyimpan penambahan dan perubahan data master ke dalam <i>database</i> sistem
	5. Sistem menampilkan kembali halaman data master
5. Aktor kembali mengakses menu data master	
Post Condition	Kelola data master yang dilakukan oleh aktor tersimpan kedalam database dan dapat diakses

Tabel 14. Skenario *Use Case* Melihat Data Master

Nama Use Case	Melihat Data Master
ID Use Case	UC-4
Aktor	Pimpinan, Bendahara
Deskripsi	Aktivitas aktor yang dibatasi hak akses hanya sebatas melihat data master perumahan yang terdapat didalam sistem.
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Pimpinan	
1. Aktor mengakses menu data master	
2. Aktor memilih sub menu yang ingin diakses diantaranya data warga, data rumah, serta data tipe rumah	
	3. Sistem menampilkan sub menu yang dipilih
4. Aktor melihat data yang ditampilkan	
Skenario Bendahara	
1. Aktor mengakses menu data master	
2. Aktor memilih sub menu yang ingin diakses diantaranya data warga, data rumah, data tipe rumah, serta data metode pembayaran	
	3. Sistem menampilkan sub menu yang dipilih

4. Aktor melihat data yang ditampilkan	
Post Condition	Aktor dapat melihat dan memeriksa data terkait data master yang ingin dilihat

Tabel 15. Skenario *Use Case* Mengelola Data Inventaris

Nama Use Case	Mengelola Data Inventaris
ID Use Case	UC-5
Aktor	Admin
Deskripsi	Aktivitas yang dilakukan oleh aktor pengelola sistem untuk melakukan 58elola terhadap data inventaris meliputi aksi menambah, mengubah, dan menghapus data inventaris dalam sistem.
Exception	Perubahan data yang dikelola tidak tersimpan ke dalam <i>database</i>
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses menu inventaris	
	2. Sistem menampilkan menu inventaris
3. Aktor melakukan kelola terhadap data inventaris berupa menambah, mengedit, atau menghapus data	
	4. Sistem menyimpan penambahan dan perubahan data inventaris ke dalam <i>database</i> sistem
	5. Menampilkan pesan perubahan kelola data inventaris telah berhasil dan menampilkan kembali halaman inventaris
Skenario Alternatif	
1. Aktor mengakses menu inventaris	
	2. Sistem menampilkan menu inventaris

3. Aktor melakukan kelola terhadap data inventaris berupa menambah, mengedit, atau menghapus data	
	4. Sistem gagal menyimpan penambahan dan perubahan data inventaris ke dalam <i>database</i> sistem
	5. Sistem menampilkan kembali halaman data inventaris
5. Aktor kembali mengakses menu inventaris	
Post Condition	Kelola data inventaris yang dilakukan oleh aktor tersimpan kedalam database dan dapat diakses

Tabel 16. Skenario *Use Case* Melihat Data Inventaris

Nama Use Case	Melihat Data Inventaris
ID Use Case	UC-6
Aktor	Pimpinan, Bendahara
Deskripsi	Aktivitas aktor yang dibatasi hak akses hanya sebatas melihat data inventaris barang perumahan yang terdapat didalam sistem.
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses menu inventaris	
	2. Sistem menampilkan menu inventaris
3. Aktor melihat data inventaris barang dan keterangan yang ditampilkan	
Post Condition	Aktor dapat melihat dan memeriksa data terkait inventaris yang ingin dilihat

Tabel 17. Skenario *Use Case* Input Permohonan Surat

Nama Use Case	Input Permohonan Surat	
ID Use Case	UC-7	
Aktor	Warga	
Deskripsi	Aktivitas yang dilakukan aktor untuk <i>input</i> data terkait permohonan surat yang ingin diajukan	
Precondition	Aktor telah login kedalam sistem	
Aktor		Sistem
Skenario Normal		
1. Aktor mengakses menu permohonan surat		
		2. Sistem menampilkan menu permohonan surat
		4. Sistem menampilkan form input permohonan surat
5. Aktor mengisi data form permohonan surat		
6. Aktor menekan tombol submit		
		7. Sistem menyimpan data dan menampilkan pesan berhasil
		8. Sistem menampilkan halaman permohonan surat kembali
Skenario Alternatif		
1. Aktor mengakses menu permohonan surat		
		2. Sistem menampilkan menu permohonan surat
		4. Sistem menampilkan form input permohonan surat
5. Aktor mengisi data form permohonan surat		
6. Aktor menekan tombol submit		
		7. Sistem menampilkan pesan keseluruhan data belum terisi
8. Aktor mengisi ulang kembali data permohonan surat		

	9. Sistem menyimpan data dan menampilkan pesan data berhasil
	8. Sistem menampilkan halaman permohonan surat kembali
Post Condition	Aktor melihat status pengajuan permohonan surat

Tabel 18. Skenario *Use Case* Mengelola Data Daftar Permohonan Surat

Nama Use Case	Mengelola Data Daftar Permohonan Surat
ID Use Case	UC-8
Aktor	Admin
Deskripsi	Aktivitas yang dilakukan oleh aktor pengelola sistem untuk melakukan pengelolaan terhadap data daftar permohonan surat meliputi aksi menambah, mengubah, dan menghapus data ajuan permohonan surat warga dalam sistem.
Exception	Perubahan data yang dikelola tidak tersimpan ke dalam <i>database</i>
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses menu daftar permohonan surat	
	2. Sistem menampilkan menu daftar permohonan surat
3. Aktor melakukan kelola terhadap data daftar permohonan surat berupa menambah dan mengedit data	
	4. Sistem menyimpan penambahan dan perubahan data daftar permohonan surat ke dalam <i>database</i> sistem
	5. Menampilkan pesan perubahan kelola data daftar permohonan surat telah berhasil dan

	menampilkan halaman daftar permohonan surat kembali
Skenario Alternatif	
1. Aktor mengakses menu daftar permohonan surat	
	2. Sistem menampilkan menu daftar permohonan surat
3. Aktor melakukan kelola terhadap data daftar permohonan surat berupa menambah, mengedit, atau menghapus data	
	4. Sistem gagal menyimpan penambahan dan perubahan data daftar permohonan surat ke dalam <i>database</i> sistem dan menampilkan halaman daftar permohonan surat kembali
6. Aktor kembali mengakses menu daftar permohonan surat	
Post Condition	Kelola data daftar permohonan surat yang dilakukan oleh aktor tersimpan kedalam database dan dapat diakses

Tabel 19. Skenario *Use Case* Melihat Daftar Permohonan Surat

Nama Use Case	Melihat Data Daftar Permohonan Surat
ID Use Case	UC-9
Aktor	Pimpinan, Bendahara
Deskripsi	Aktivitas aktor yang dibatasi hak akses hanya sebatas melihat data daftar permohonan surat pengantar RT perumahan yang terdapat didalam sistem.
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses daftar permohonan surat	

	2. Sistem menampilkan menu daftar permohonan surat
3. Aktor melihat data permohonan surat yang telah diajukan warga	
Post Condition	Aktor melihat data pada menu permohonan surat

Tabel 20. Skenario *Use Case* Input Pembayaran Iuran

Nama Use Case	Input Pembayaran Iuran	
ID Use Case	UC-10	
Aktor	Warga	
Deskripsi	Aktivitas yang dilakukan aktor untuk <i>input</i> data terkait pembayaran iuran yang telah dilakukan	
Precondition	Aktor telah login kedalam sistem	
Aktor	Sistem	
Skenario Normal		
1. Aktor mengakses menu iuran dan memilih submenu pembayaran iuran		
		2. Sistem menampilkan submenu pembayaran iuran
3. Aktor mengisi data form pembayaran iuran		
4. Aktor menekan tombol submit		
		5. Sistem menyimpan data dan menampilkan data berhasil tersimpan
		6. Sistem menampilkan halaman pembayaran iuran kembali
Skenario Alternatif		
1. Aktor mengakses menu iuran dan memilih submenu pembayaran iuran		
		2. Sistem menampilkan submenu pembayaran iuran
3. Aktor mengisi data form pembayaran iuran		
4. Aktor menekan tombol submit		

	5. Sistem menampilkan pesan keseluruhan data belum terisi
6. Aktor mengisi ulang kembali data pembayaran	
	7. Sistem menyimpan data dan menampilkan pesan data berhasil tersimpan
	8. Sistem menampilkan halaman pembayaran iuran kembali
Post Condition	Aktor melihat data pembayaran iuran yang telah diinputkan

Tabel 21. Skenario *Use Case* Mengelola Data Daftar Iuran

Nama Use Case	Mengelola Daftar Iuran
ID Use Case	UC-11
Aktor	Bendahara
Deskripsi	Aktivitas yang dilakukan oleh aktor pengelola sistem untuk melakukan kelola terhadap daftar iuran meliputi aksi menambah, mengubah, dan menghapus data daftar pembayaran iuran dalam sistem.
Exception	Perubahan data yang dikelola tidak tersimpan ke dalam <i>database</i>
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses menu daftar iuran	
	2. Sistem menampilkan menu daftar iuran
3. Aktor melakukan kelola terhadap data daftar iuran berupa menambah, mengedit, atau menghapus data	
	4. Sistem menyimpan penambahan dan perubahan data daftar iuran ke dalam <i>database</i> sistem

	5. Menampilkan pesan perubahan kelola data daftar iuran telah berhasil dan menampilkan kembali halaman daftar iuran
Skenario Alternatif	
1. Aktor mengakses menu daftar iuran	
	2. Sistem menampilkan menu daftar iuran
3. Aktor melakukan kelola terhadap data daftar iuran berupa menambah, mengedit, atau menghapus data	
	4. Sistem gagal menyimpan penambahan dan perubahan data daftar iuran ke dalam <i>database</i> sistem
	5. Sistem menampilkan kembali halaman daftar iuran
6. Aktor kembali mengakses menu daftar iuran	
Post Condition	Kelola data daftar iuran yang dilakukan oleh aktor tersimpan kedalam database dan dapat diakses

Tabel 22. Skenario *Use Case* Melihat Data Daftar Iuran

Nama Use Case	Melihat Data Permohonan Surat
ID Use Case	UC-12
Aktor	Pimpinan, Bendahara
Deskripsi	Aktivitas aktor yang dibatasi hak akses hanya sebatas melihat data daftar iuran perumahan yang terdapat didalam sistem.
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses menu daftar iuran	
	2. Sistem menampilkan menu daftar iuran

3. Aktor melihat data daftar iuran yang telah dibayar oleh warga perumahan	
Post Condition	Aktor melihat data pada menu daftar iuran

Tabel 23. Skenario *Use Case* Melihat Riwayat Pembayaran Iuran

Nama Use Case	Melihat Riwayat Pembayaran Iuran
ID Use Case	UC-13
Aktor	Warga
Deskripsi	Aktivitas aktor yang dibatasi hak akses hanya sebatas melihat data laporan keuangan perumahan yang terdapat didalam sistem
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses menu iuran dan memilih submenu riwayat pembayaran iuran	
	2. Sistem menampilkan submenu riwayat pembayaran iuran pribadi
3. Aktor melihat riwayat pembayaran iuran yang telah dibayarkan	
Post Condition	Aktor melihat riwayat pembayaran iuran pada menu iuran

Tabel 24. Skenario *Use Case* Validasi Permohonan Surat

Nama Use Case	Validasi Permohonan Surat
ID Use Case	UC-14
Aktor	Admin
Deskripsi	Aktivitas proses validasi pengecekan yang dilakukan oleh aktor terkait permohonan surat pengantar RT perumahan yang diajukan warga
Exception	Data tidak valid
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	

1. Aktor mengakses menu permohonan surat	
	2. Sistem menampilkan menu permohonan surat
3. Aktor memeriksa data permohonan surat yang diajukan	
4. Aktor menekan tombol valid	
	5. Sistem menampilkan pesan validasi berhasil dan merubah status validasi menjadi proses
	6. Sistem menampilkan halaman permohonan surat
7. Aktor menekan tombol ceklis validasi tahap selanjutnya	
	8. Sistem menampilkan pesan masukkan nomor surat
9. Aktor input nomor surat	
	10. Sistem menampilkan halaman kembali permohonan surat dan merubah status menjadi selesai
Skenario Alternatif	
1. Aktor mengakses menu permohonan surat	
	2. Sistem menampilkan menu permohonan surat
3. Aktor memeriksa data permohonan surat yang diajukan	
4. Aktor menekan tombol invalid	
	5. Sistem menampilkan pesan berhasil
	6. Sistem menampilkan halaman daftar permohonan surat kembali
Post Condition	Aktor ditampilkan data halaman permohonan surat yang telah tervalidasi

Tabel 25. Skenario *Use Case* Validasi Pembayaran Iuran

Nama Use Case	Validasi Pembayaran Iuran	
ID Use Case	UC-15	
Aktor	Bendahara	
Deskripsi	Aktivitas proses memvalidasi atau pengecekan yang dilakukan oleh aktor bendahara terkait pembayaran iuran perumahan yang di-inputkan warga perumahan.	
Exception	Data Tidak Valid	
Precondition	Aktor telah login kedalam sistem	
Aktor		Sistem
Skenario Normal		
1. Aktor mengakses menu daftar iuran		
		2. Sistem menampilkan menu daftar iuran
3. Aktor memilih informasi bukti pembayaran iuran yang akan dilakukan pengecekan		
		4. Sistem menampilkan gambar bukti data pembayaran
5. Aktor mengecek data pembayaran dan menekan tombol valid		
		6. Sistem menampilkan pesan validasi berhasil dan merubah status validasi
		7. Sistem menampilkan halaman pembayaran iuran kembali
Skenario Alternatif		
1. Aktor mengakses menu daftar iuran		
		2. Sistem menampilkan menu daftar iuran
3. Aktor memilih informasi bukti pembayaran iuran yang akan dilakukan pengecekan		
		4. Sistem menampilkan gambar bukti data pembayaran

5. Aktor mengecek data pembayaran dan menekan tombol invalid	
	6. Sistem menyimpan data form dan menampilkan pesan berhasil
	7. Sistem menampilkan halaman pembayaran iuran
Post Condition	Aktor ditampilkan halaman menu pembayaran iuran

Tabel 26. Skenario *Use Case* Mengelola Data Laporan Keuangan

Nama Use Case	Mengelola Laporan Keuangan
ID Use Case	UC-16
Aktor	Bendahara
Deskripsi	Aktivitas yang dilakukan oleh aktor pengelola sistem untuk melakukan kelola terhadap laporan keuangan meliputi aksi menambah, mengubah, dan menghapus data laporan keuangan dalam sistem.
Exception	Perubahan data yang dikelola tidak tersimpan ke dalam <i>database</i>
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor mengakses menu laporan keuangan	
	2. Sistem menampilkan menu laporan keuangan
3. Aktor melakukan kelola terhadap data laporan keuangan berupa menambah, mengedit, atau menghapus data	
	4. Sistem menyimpan penambahan dan perubahan data laporan keuangan ke dalam <i>database</i> sistem

	5. Menampilkan pesan perubahan kelola data laporan keuangan telah berhasil dan menampilkan kembali halaman laporan keuangan
Skenario Alternatif	
1. Aktor mengakses menu laporan keuangan	
	2. Sistem menampilkan menu laporan keuangan
3. Aktor melakukan kelola terhadap data laporan keuangan berupa menambah, mengedit, atau menghapus data	
	4. Sistem gagal menyimpan penambahan dan perubahan data laporan keuangan ke dalam <i>database</i> sistem
	5. Menampilkan halaman laporan keuangan kembali
6. Aktor kembali mengakses menu laporan keuangan	
Post Condition	Kelola data laporan keuangan yang dilakukan oleh aktor tersimpan kedalam database dan dapat diakses

Tabel 27. Skenario *Use Case* Melihat Laporan Keuangan

Nama Use Case	Melihat Laporan Keuangan
ID Use Case	UC-17
Aktor	Admin, Pimpinan, Warga
Deskripsi	Aktivitas aktor yang dibatasi hak akses hanya sebatas melihat data laporan keuangan perumahan yang terdapat didalam sistem
Precondition	Aktor telah login kedalam sistem
Aktor	Sistem
Skenario Normal	

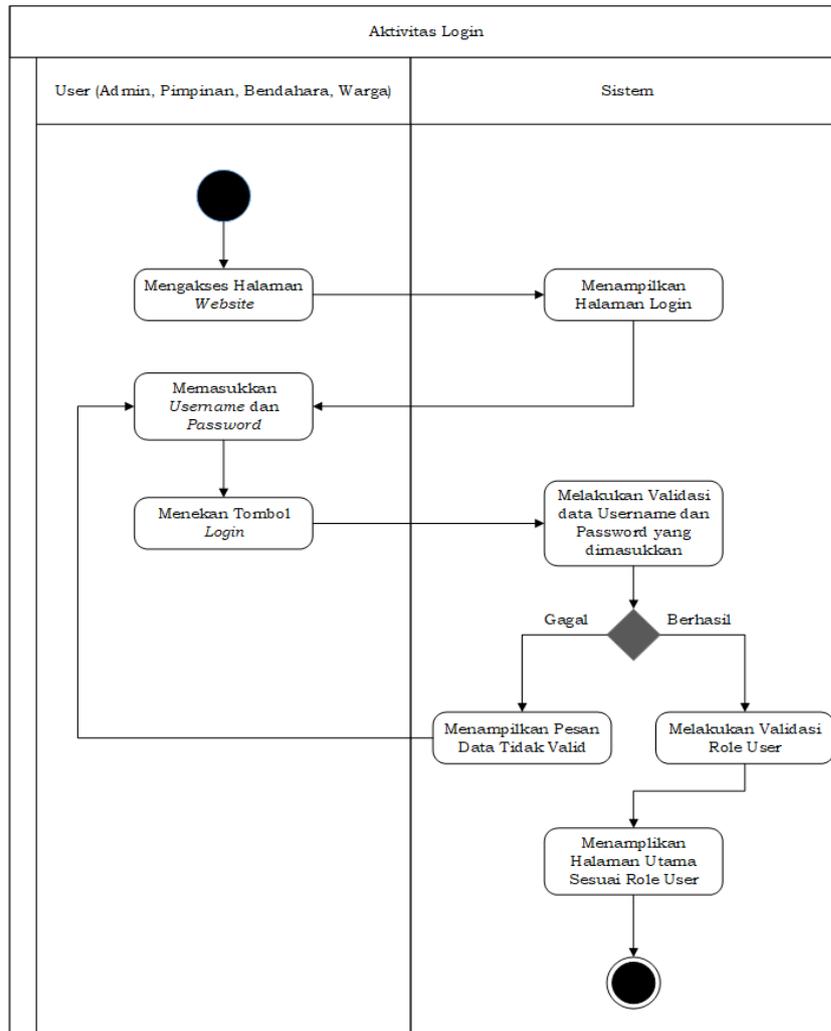
1. Aktor mengakses menu laporan keuangan	
	2. Sistem menampilkan menu laporan keuangan
3. Aktor melihat data laporan keuangan diantaranya data pengeluaran, pemasukan, dan rekapitulasi keuangan	
Post Condition	Aktor melihat data pada menu laporan keuangan

4.3 Tahap Perancangan

Tahap perancangan sistem merupakan tahap selanjutnya dalam sebuah analisis dan perancangan sistem yang berdasarkan metode OOAD termasuk pada bagian OOD (*Object Oriented Design*). *Object-Oriented Design* merupakan tahap kedua dalam metode OOAD yang akan lebih memetakan spesifikasi kelas dan kebutuhan sistem dengan berorientasi terhadap objek yang telah didefinisikan pada tahapan OOA sebelumnya menggunakan UML serta membuat tampilan *prototype* dari antarmuka sistem (UI) (A.S. & Shalahuddin, 2015). Dalam perancangan, terdapat 4 tahapan yang dilakukan yaitu merancang diagram aktivitas untuk setiap skenario, rancang diagram kelas secara detail, rancang diagram urutan untuk setiap scenario, dan rancang tampilan desain antarmuka (UI).

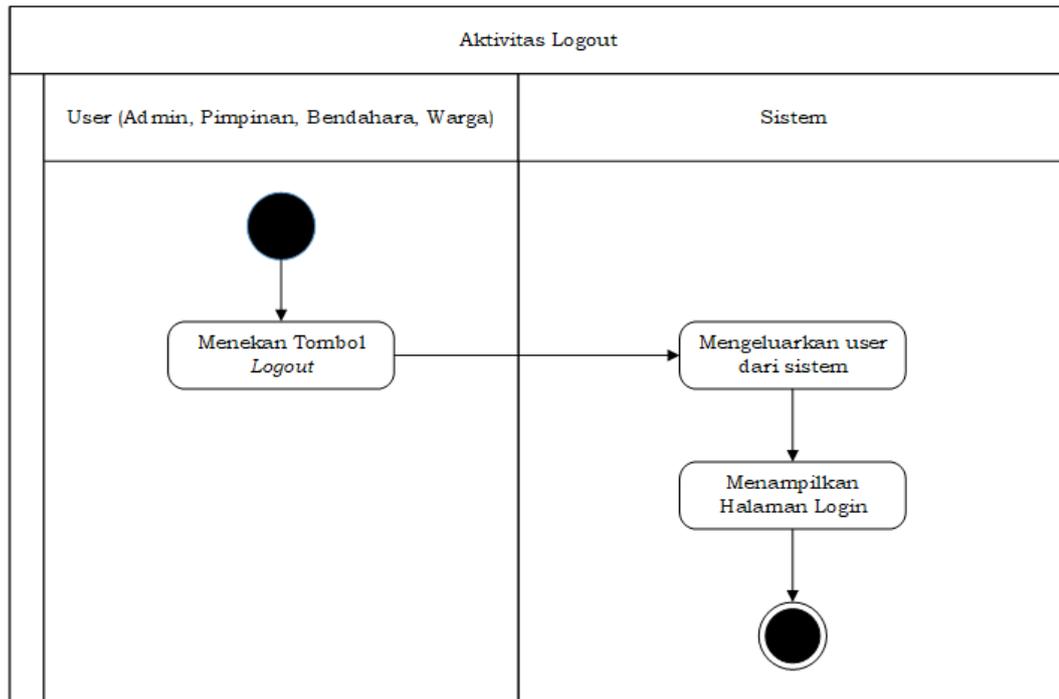
Rancang Diagram Aktivitas untuk Setiap Skenario

Diagram aktivitas atau *Activity Diagram* menggambarkan perilaku dinamis dari suatu sistem atau bagian dari sistem tersebut dengan aliran kontrol atau aliran aktivitas diantara tindakan yang dilakukan oleh sistem. Diagram aktivitas akan memberikan urutan aktivitas rancangan proses bisnis sistem dan runutan tampilan dari antarmuka sistem untuk setiap aktivitas menggambarkan rancangan menu yang akan ditampilkan pada *interface* perangkat lunak. Diagram aktivitas (*Activity Diagram*) dari sistem SIPGAMA diantaranya adalah sebagai berikut :

a. *Activity Diagram Login*

Gambar 18. *Activity Diagram Login*

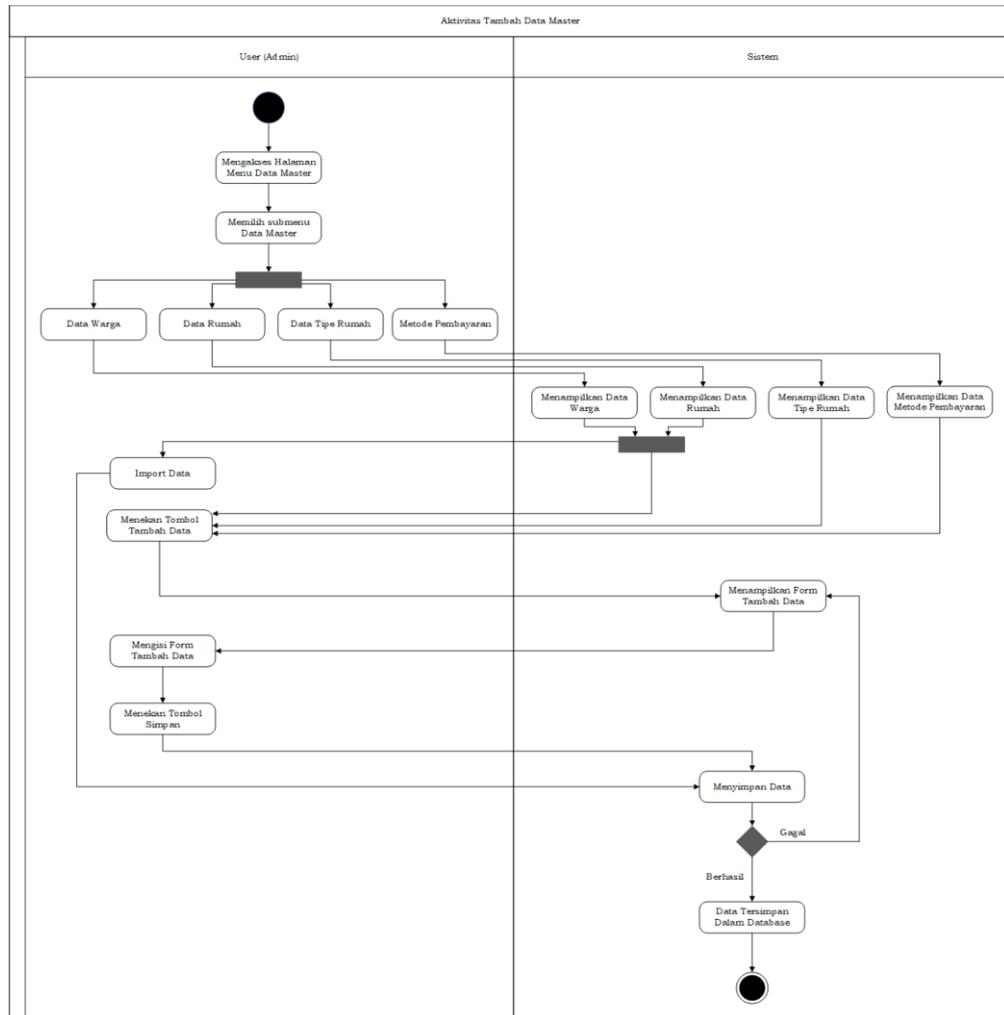
Pada gambar 18 diatas, ditampilkan aliran aktivitas *login* yang harus dilakukan oleh pengguna sistem untuk masuk kedalam sistem sehingga dapat mengakses setiap fungsi sistem. Dalam proses *login*, aktor terlebih dahulu mengakses laman website yang kemudian oleh sistem akan ditampilkan halaman *login*. Aktor memasukkan *username* dan *password* yang telah terdaftar untuk selanjutnya divalidasi oleh sistem apakah data yang dimasukkan telah sesuai. Apabila data telah sesuai, maka akan ditampilkan halaman utama. Jika data *login* salah, maka sistem akan menampilkan pesan bahwa data tidak valid dan kemudian aktor perlu memasukkan ulang *username* dan *password* yang benar.

b. *Activity Diagram Logout*

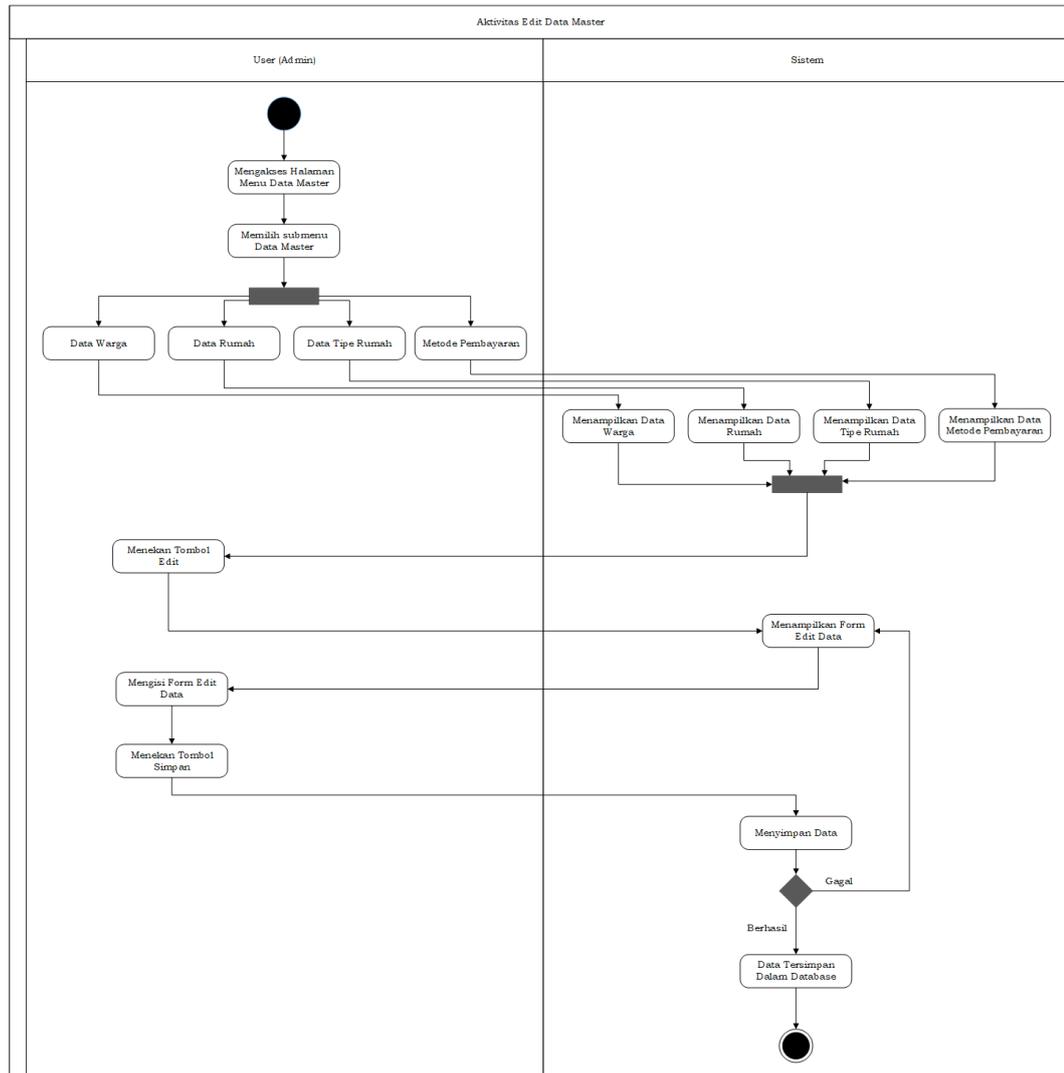
Gambar 19. *Activity Diagram Logout*

Pada Gambar 19 diatas, diperlihatkan aliran aktivitas *logout* yang dilakukan oleh pengguna Ketika ingin keluar dari sistem. Aktor perlu menekan tombol *logout* untuk selanjutnya dikeluarkan dari sistem dan sistem akan menampilkan halaman login kembali.

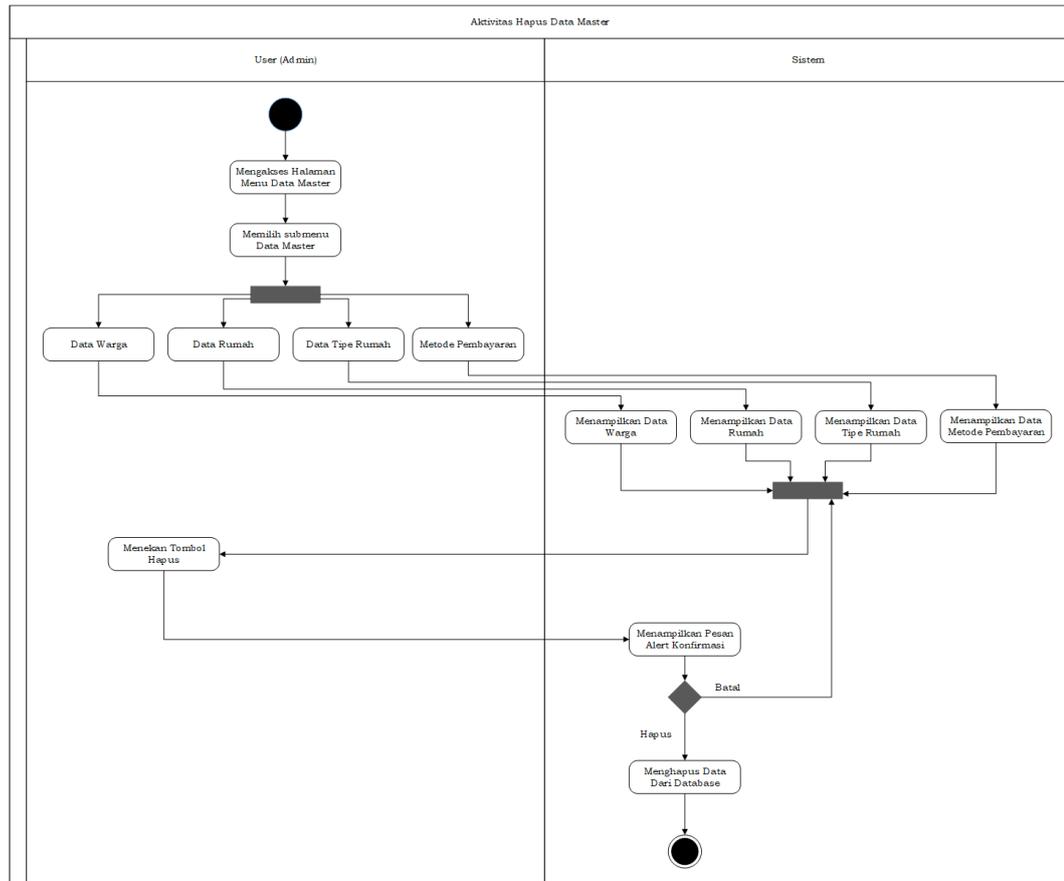
c. *Activity Diagram* Kelola Data Master



Gambar 20. *Activity Diagram* Tambah Data Master

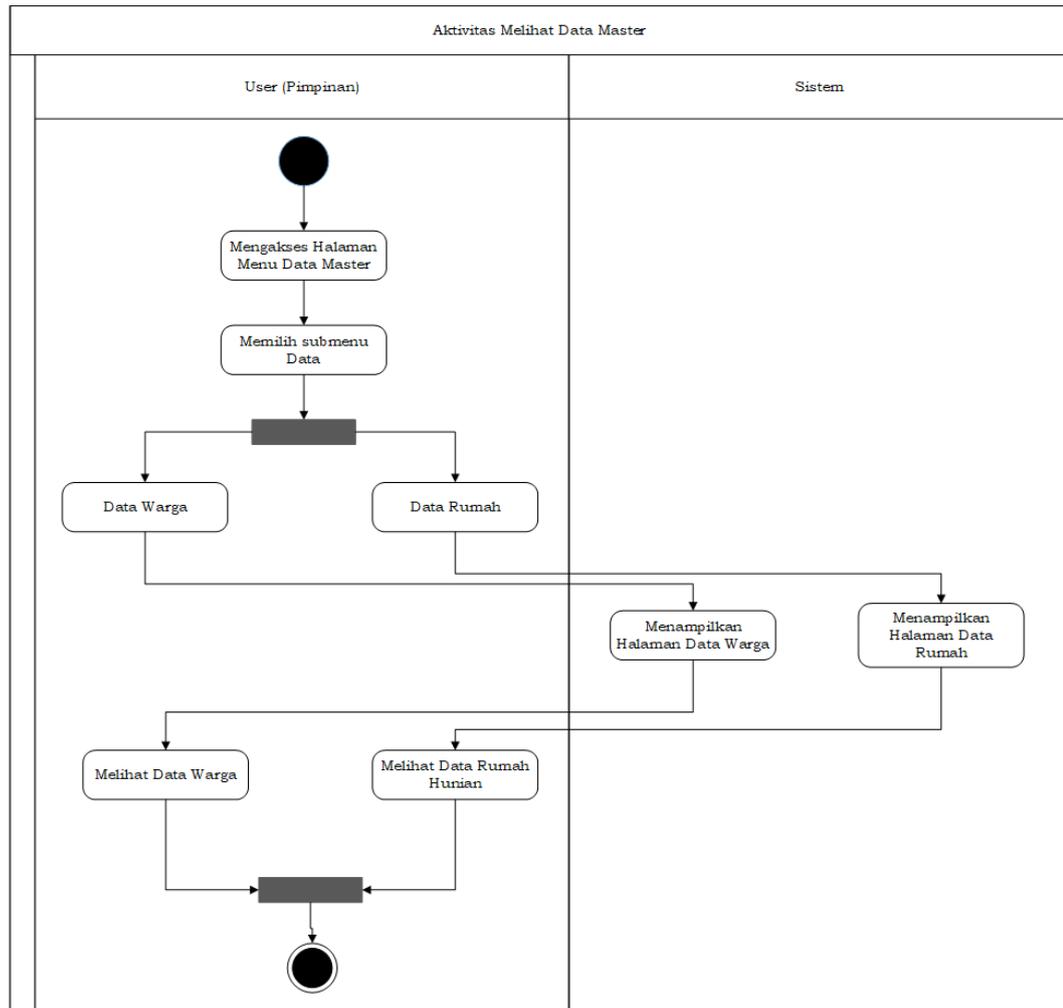


Gambar 21. Activity Diagram Edit Data Master

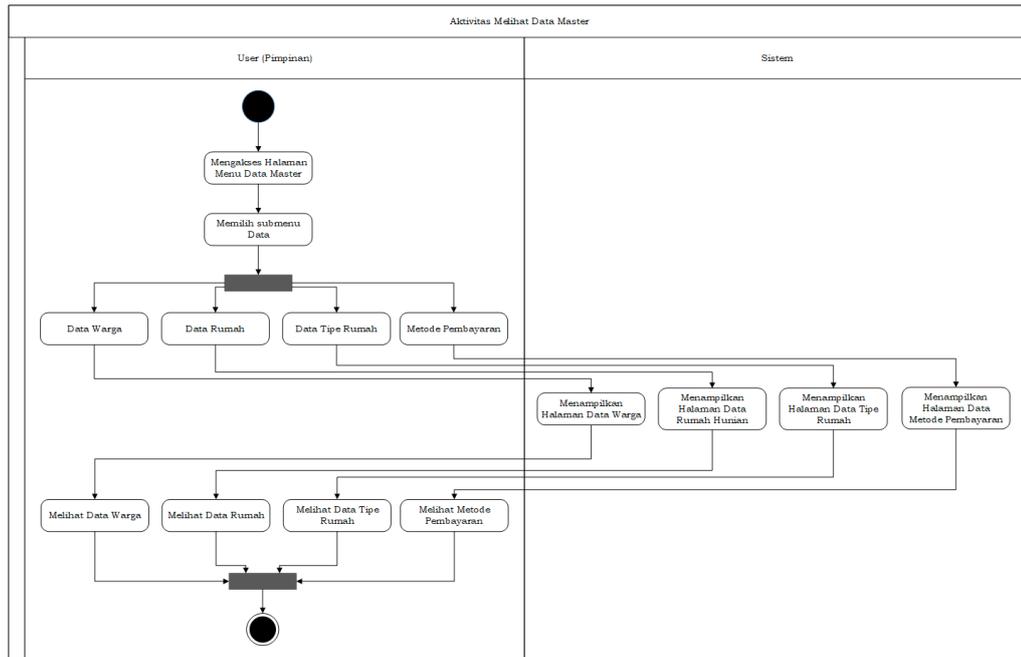


Gambar 22. Activity Diagram Hapus Data Master

Pada Gambar 20-22 diatas, terlihat aktivitas pengelolaan data master meliputi tambah, edit, dan hapus data dengan masing-masing aliran dimulai dari mengakses halaman menu dan diakhiri dengan penyimpanan data kedalam *database* sistem.

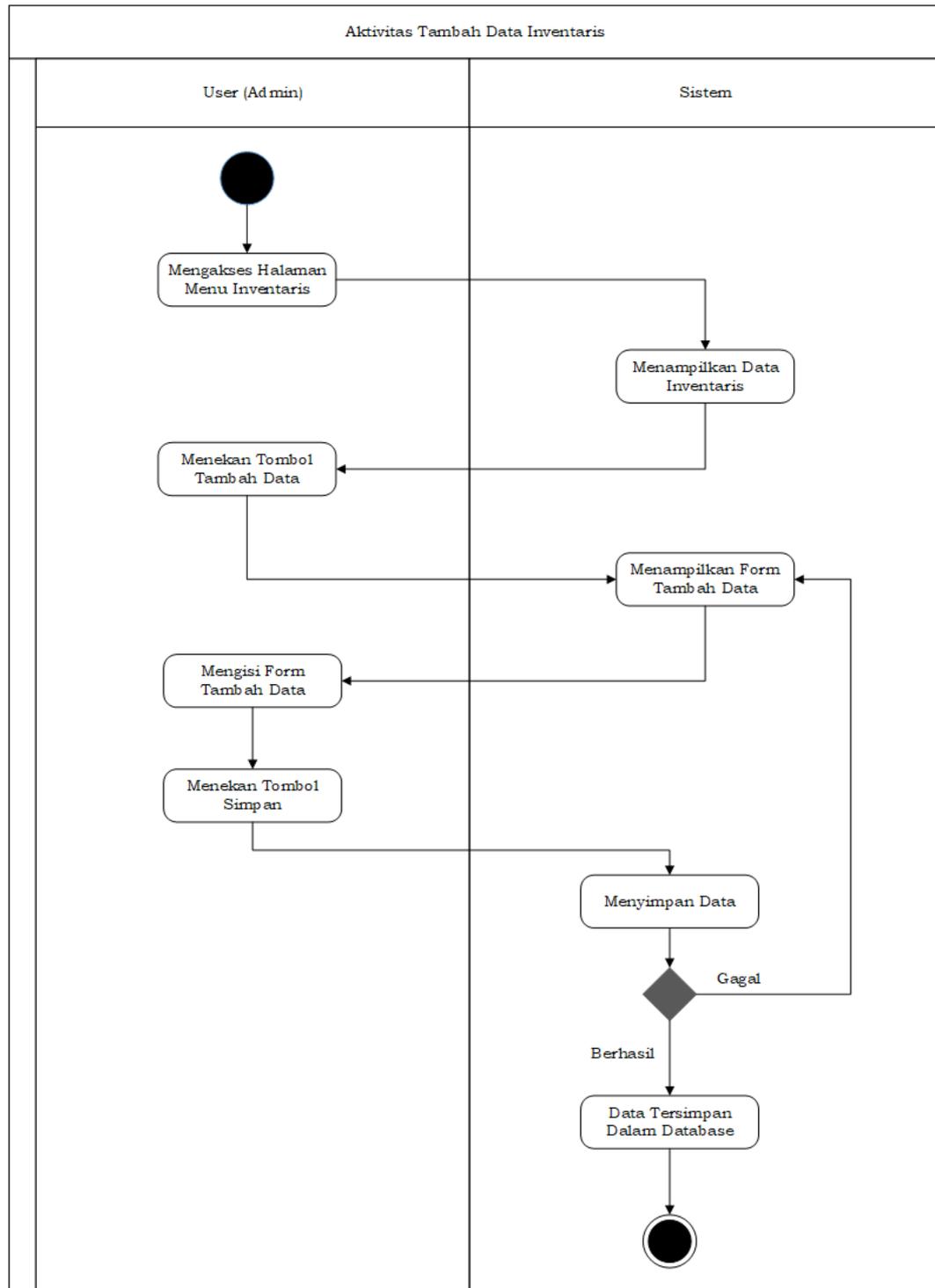
d. *Activity Diagram* Melihat Data Master

Gambar 23. *Activity Diagram* Melihat Data Master (Pimpinan)

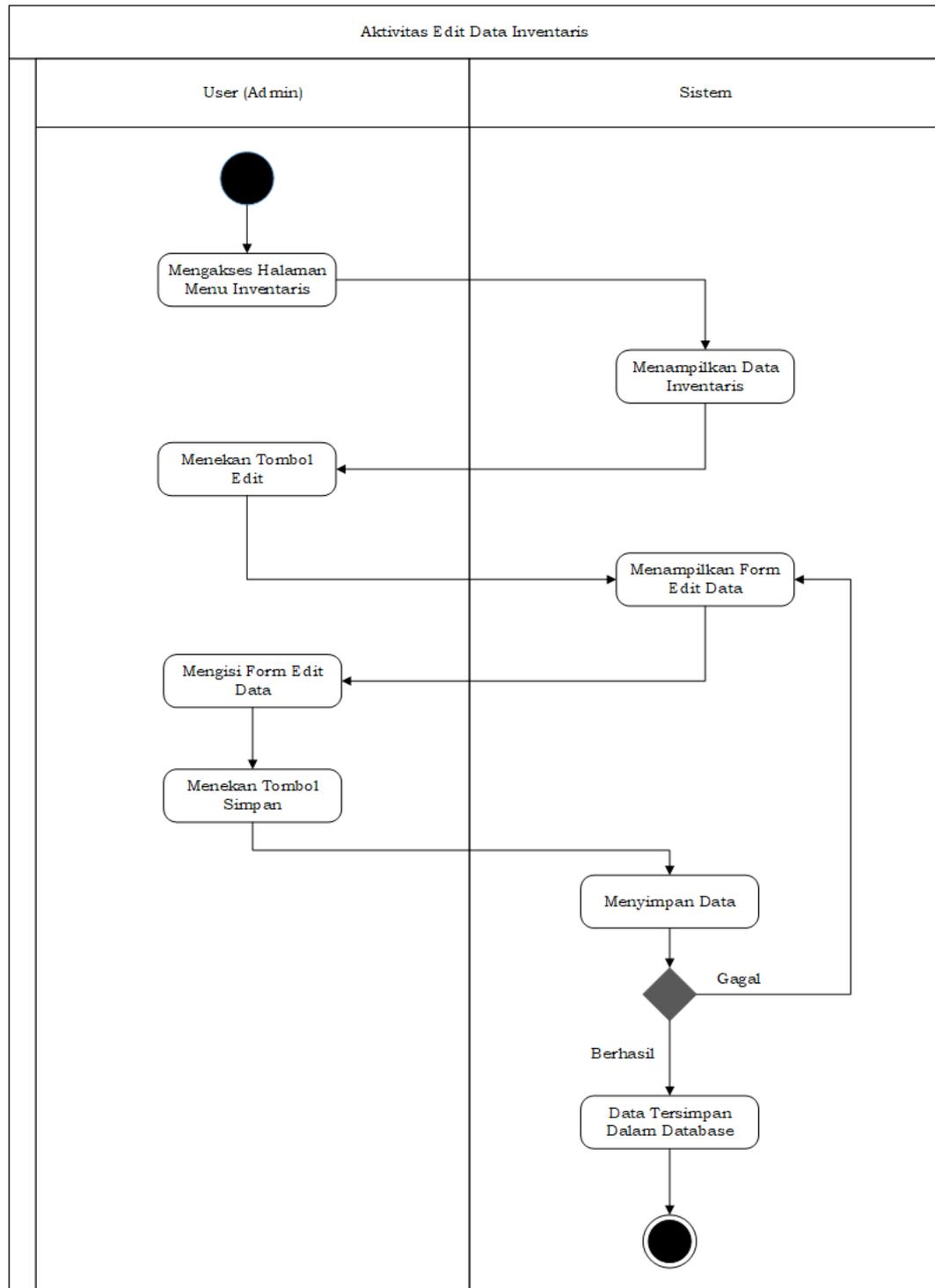


Gambar 24. Activity Diagram Melihat Data Master (Bendahara)

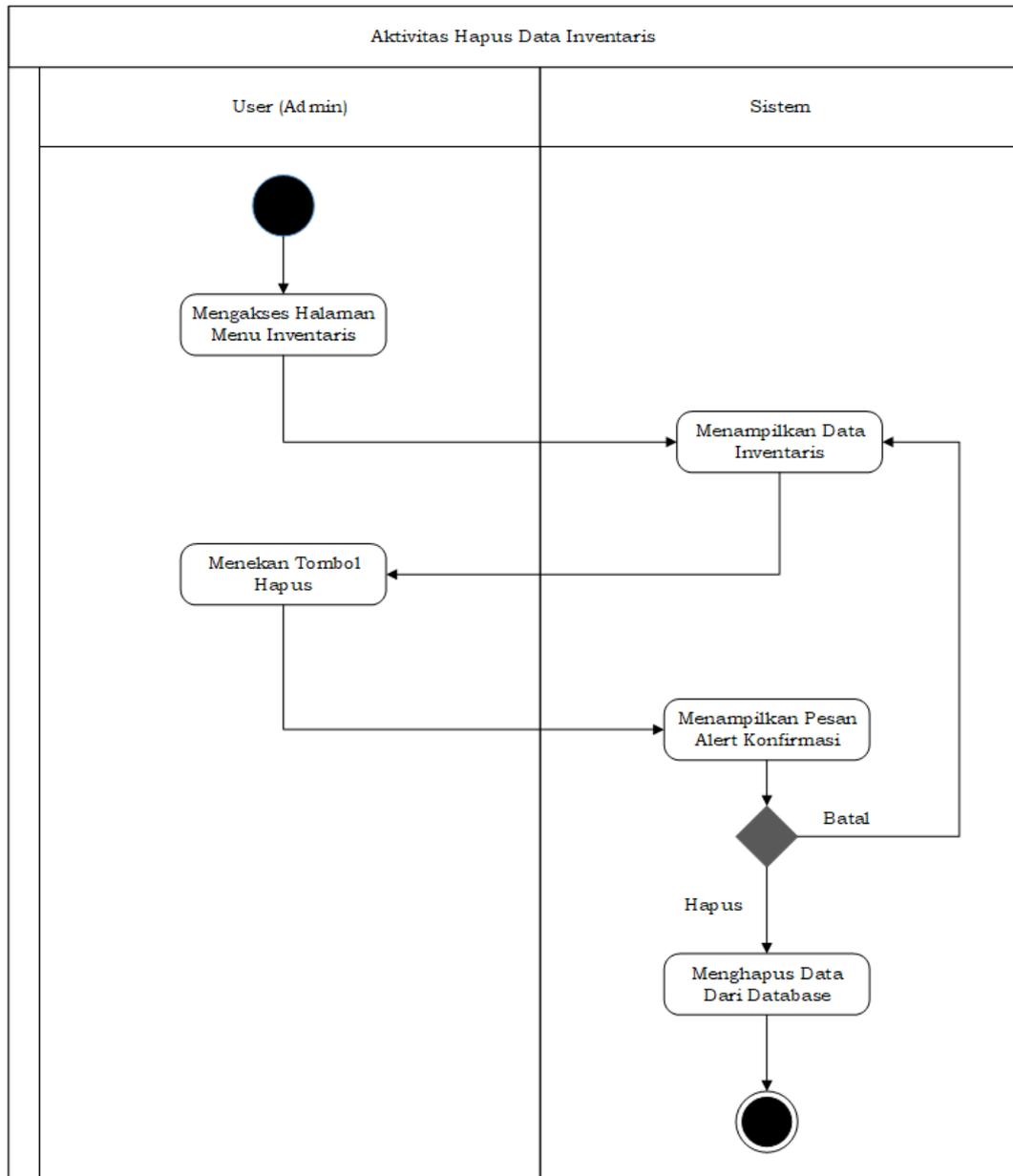
Pada gambar 23 dan 24 diatas, dapat dilihat aliran aktivitas melihat data master yang dapat dilakukan sebatas akses melihat data oleh aktor pimpinan dan bendahara dimana data dapat dilihat dengan mengakses menu data master terlebih dahulu yang akan terdapat 2 submenu didalamnya bagi pimpinan yaitu data warga dan data rumah, serta 4 submenu bagi bendahara yaitu data warga, data rumah, data tipe rumah, dan metode pembayaran. Setelah dipilih submenu yang akan dilihat kemudian sistem akan menampilkan halaman berisikan data terkait yang dapat dilihat oleh aktor pengguna sistem.

e. *Activity Diagram* Kelola Data Inventaris

Gambar 25. *Activity Diagram* Tambah Data Inventaris

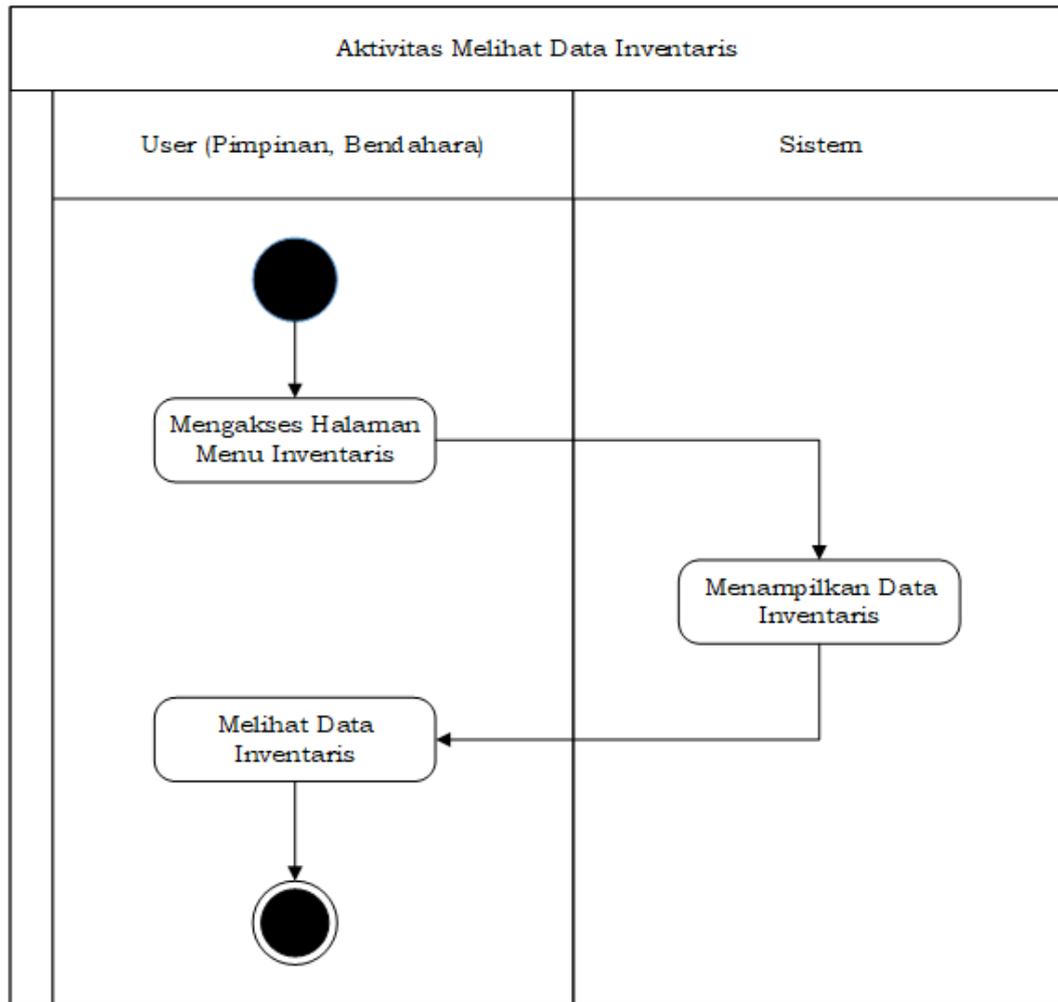


Gambar 26. Activity Diagram Edit Data Inventaris



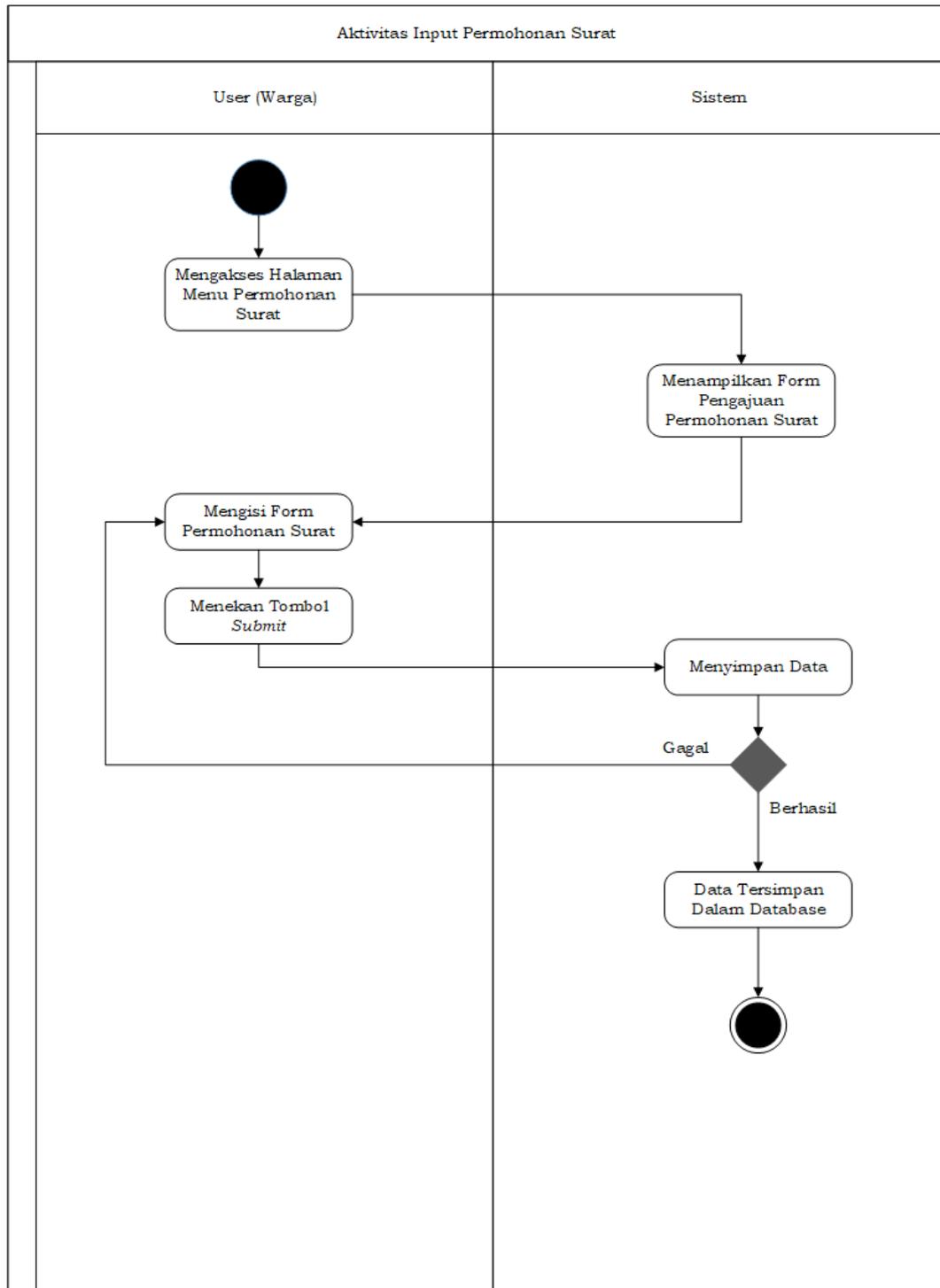
Gambar 27. Activity Diagram Hapus Data Inventaris

Pada Gambar 25-27 diatas, terlihat aktivitas pengelolaan data inventaris meliputi tambah, edit, dan hapus data dengan masing-masing aliran dimulai dari mengakses halaman menu dan diakhiri dengan penyimpanan data kedalam database sistem.

f. *Activity Diagram* Melihat Data Inventaris

Gambar 28. *Activity Diagram* Melihat Data Inventaris

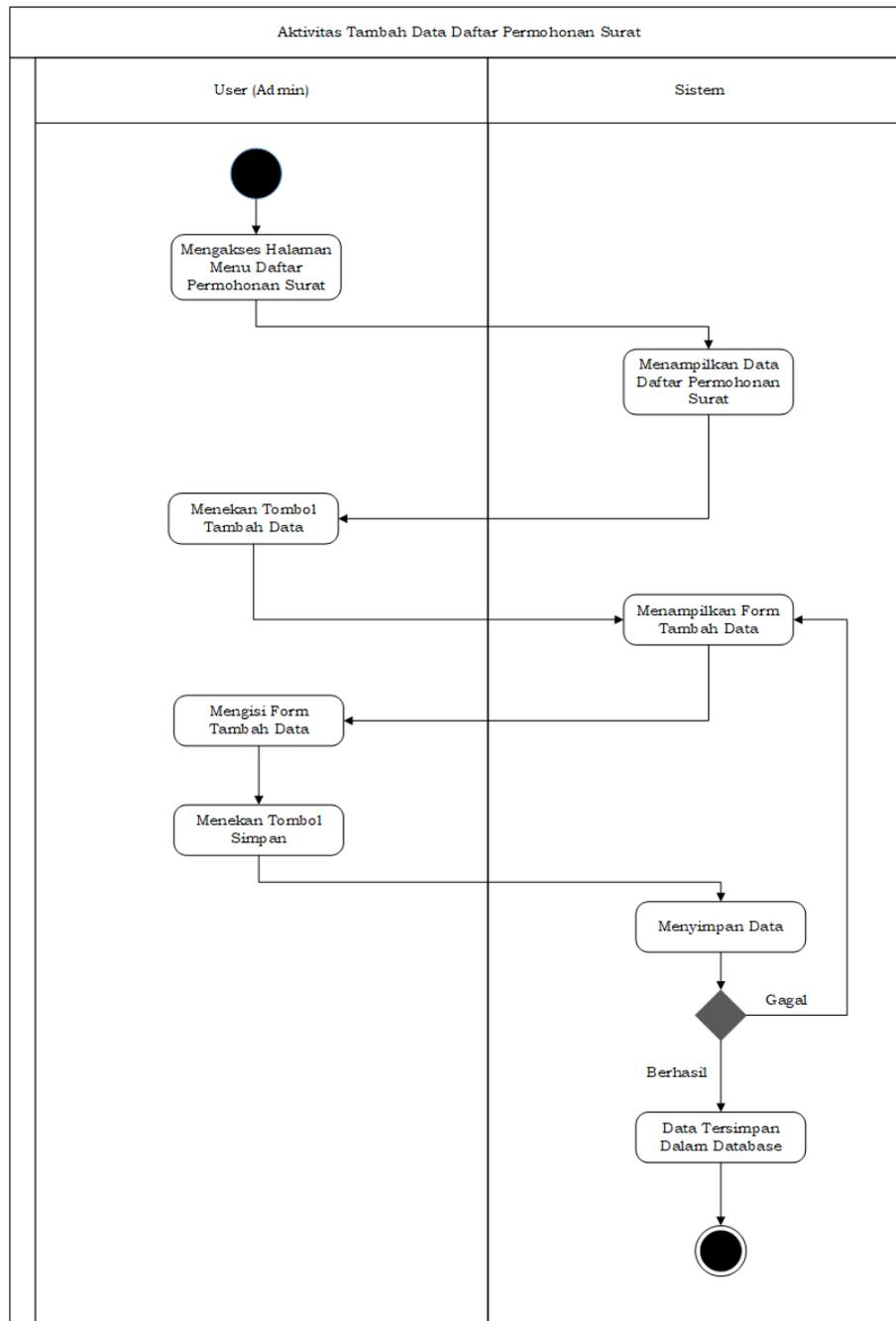
Pada gambar 28 diatas, dapat dilihat aliran aktivitas melihat data inventaris yang dapat dilakukan sebatas akses melihat data oleh aktor pimpinan dan bendahara dimana data dapat dilihat dengan mengakses menu inventaris terlebih dahulu, kemudian sistem akan menampilkan halaman berisikan data inventaris yang dapat dilihat oleh aktor pengguna sistem.

g. *Activity Diagram Input Permohonan Surat***Gambar 29.** *Activity Diagram Input Permohonan Surat*

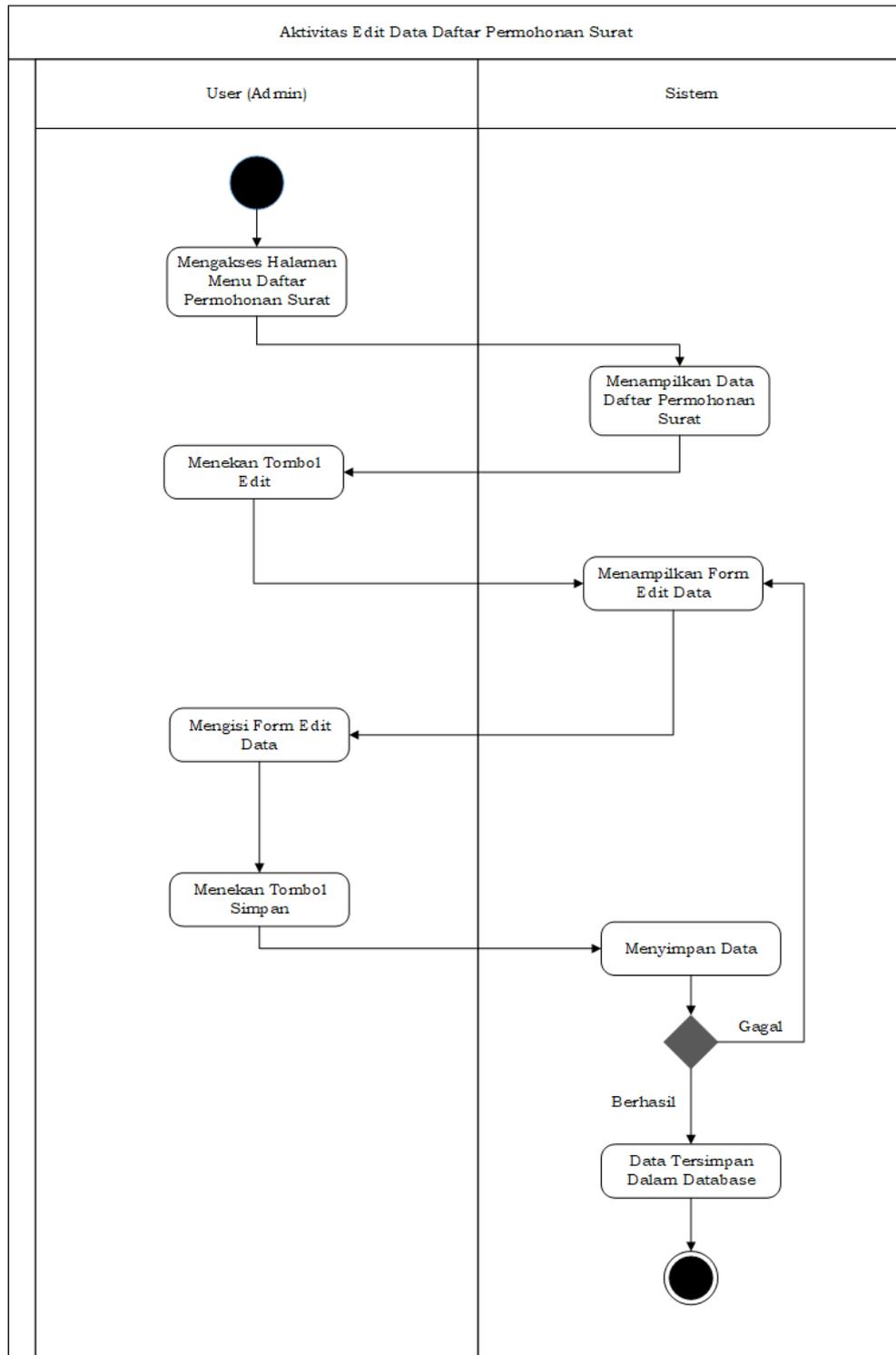
Pada gambar 29 diatas, dapat dilihat aliran aktivitas input permohonan surat yang dilakukan oleh aktor warga untuk memasukkan data permohonan surat yang ingin diminta sebagai pengantar, dimana aktor harus mengakses menu permohonan surat terlebih dahulu untuk kemudian sistem akan menampilkan

form pengajuan permohonan surat. Selanjutnya aktor dapat mengisi data terkait permohonan surat yaitu nama pemohon beserta pilihan jenis surat pengantar yang ingin diminta. Setelah yakin data yang *diinputkan* sesuai maka aktor dapat menekan tombol *submit*, jika berhasil data akan tersimpan kedalam *database* sistem dan aktor dapat melihat status pengajuannya, namun jika gagal maka aktor dapat mengisi ulang kembali form *input* permohonan surat.

h. *Activity Diagram* Kelola Data Daftar Permohonan Surat

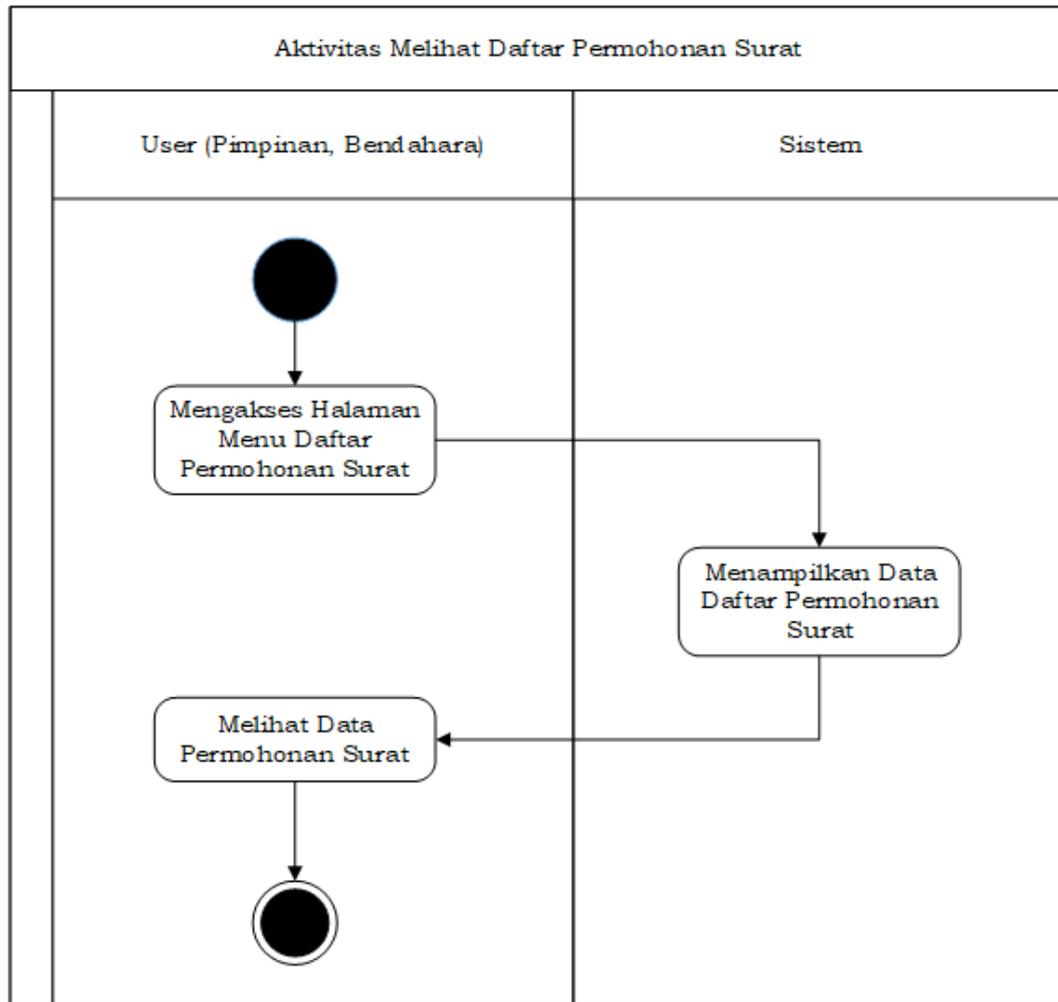


Gambar 30. *Activity Diagram* Tambah Data Daftar Permohonan Surat



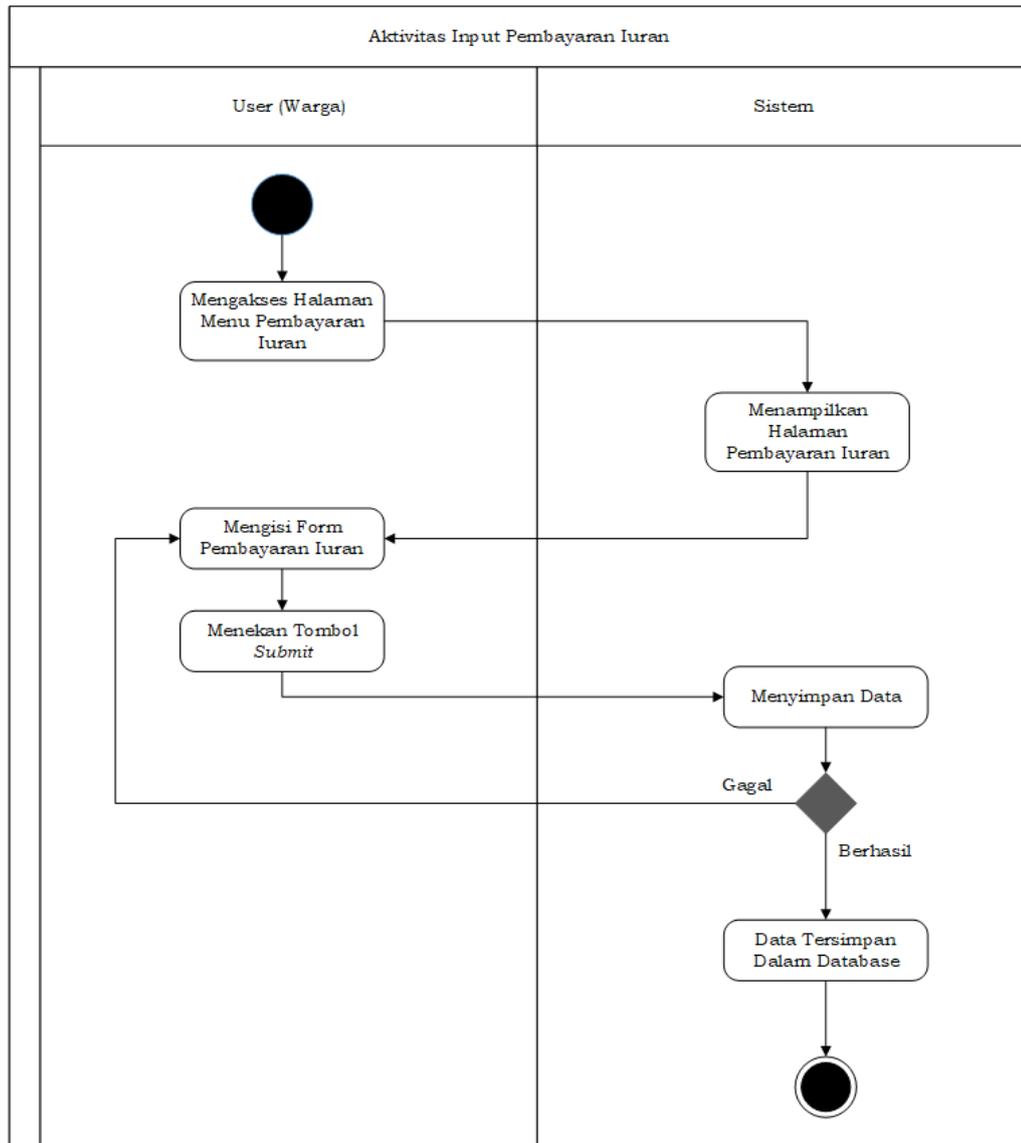
Gambar 31. Activity Diagram Edit Data Daftar Permohonan Surat

Pada Gambar 30-31 diatas, terlihat aktivitas pengelolaan data daftar permohonan surat meliputi tambah dan edit data dengan masing-masing aliran dimulai dari mengakses halaman menu dan diakhiri dengan penyimpanan data kedalam database sistem.

i. *Activity Diagram* Melihat Data Permohonan Surat

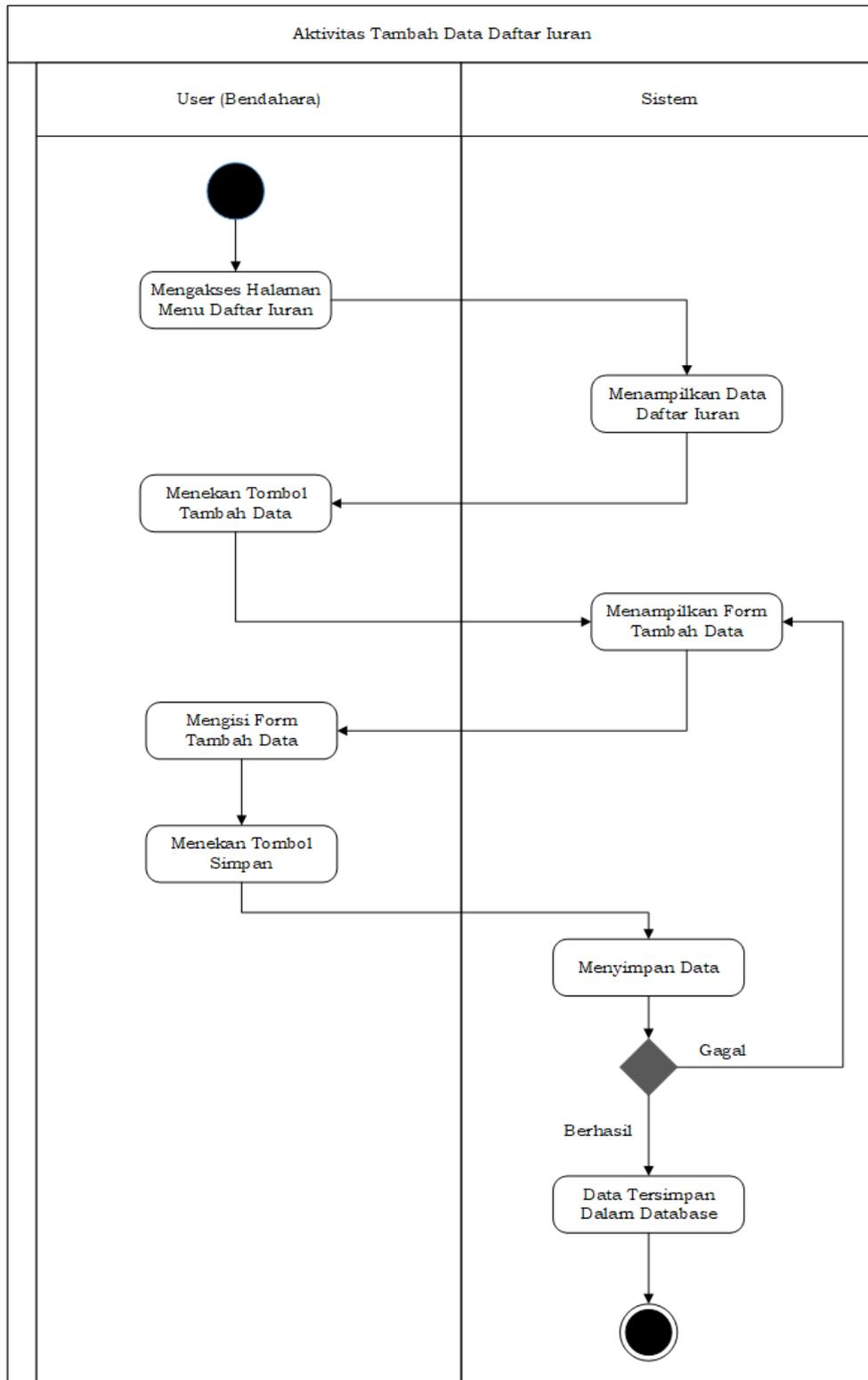
Gambar 32. *Activity Diagram* Melihat Daftar Permohonan Surat

Pada gambar 32 diatas, dapat dilihat aliran aktivitas melihat daftar permohonan surat yang dapat dilakukan sebatas akses melihat data oleh aktor pimpinan dan bendahara dimana data dapat dilihat dengan mengakses menu daftar permohonan surat terlebih dahulu, kemudian sistem akan menampilkan halaman berisikan data permohonan surat yang dapat dilihat oleh aktor pengguna sistem.

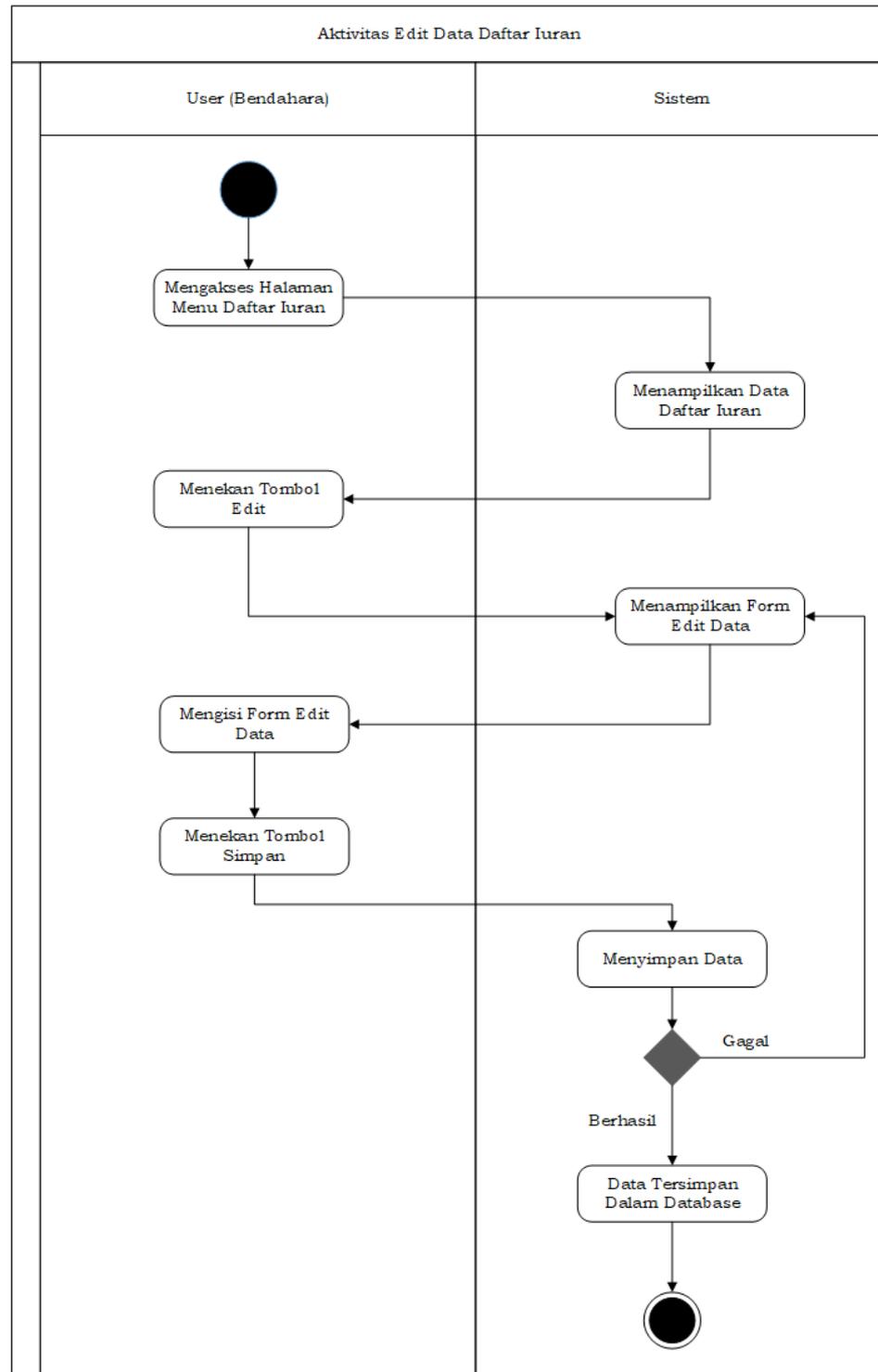
j. *Activity Diagram Input Pembayaran Iuran*

Gambar 33. *Activity Diagram Input Pembayaran Iuran*

Pada gambar 33 diatas, dapat dilihat aliran aktivitas input pembayaran iuran yang dilakukan oleh aktor warga untuk memasukkan data pembayaran iuran yang telah dilakukan, dimana aktor harus mengakses menu pembayaran iuran terlebih dahulu untuk kemudian sistem akan menampilkan halaman pembayaran iuran berisikan form *input* pembayaran. Selanjutnya aktor dapat mengisi data terkait pembayaran iuran yang diminta mulai dari berapa jumlah bulan yang dibayarkan, metode pembayaran, dan informasi bukti pembayaran yang telah dilakukan. Setelah yakin data yang diinputkan sesuai maka aktor dapat menekan tombol *submit*, jika berhasil data akan tersimpan dalam *database* sistem, namun jika gagal maka aktor dapat mengisi ulang kembali form *input* pembayaran iuran.

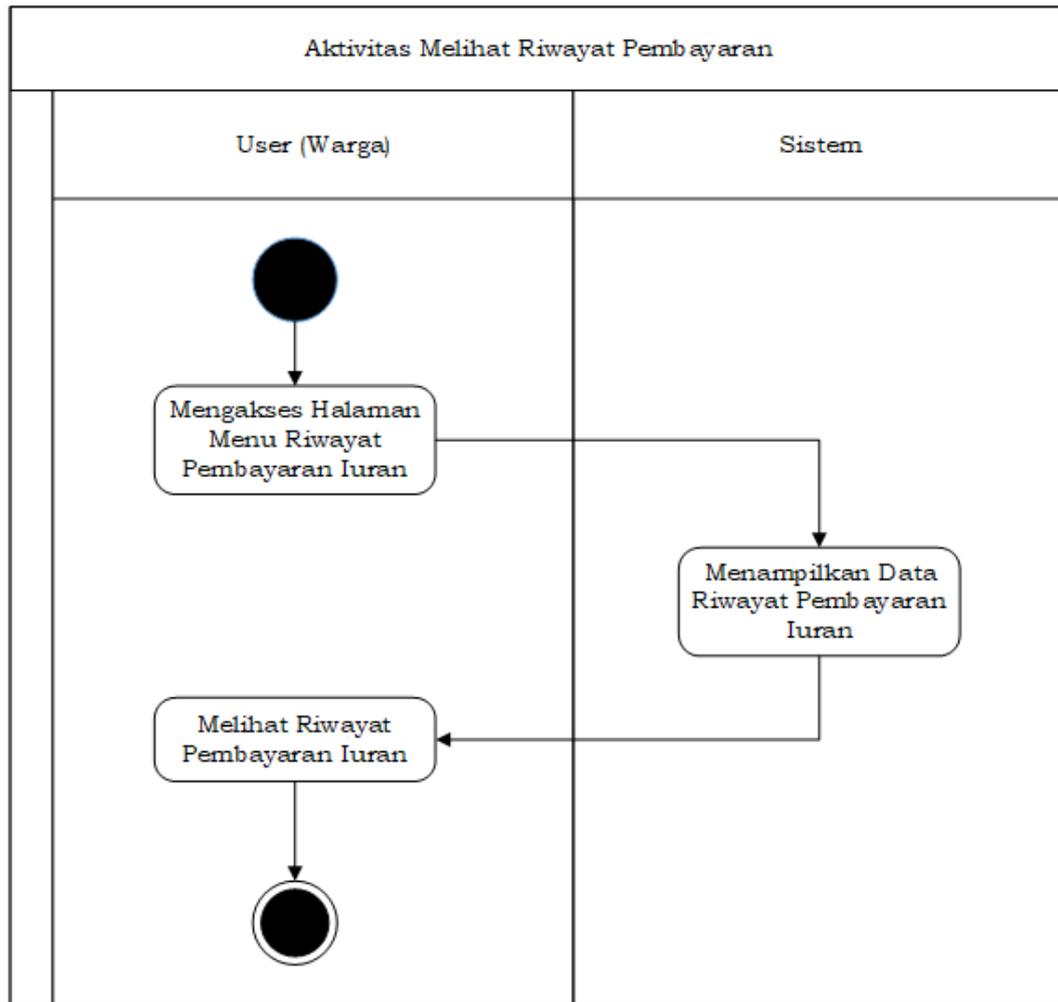
k. *Activity Diagram* Kelola Data Daftar Iuran

Gambar 34. *Activity Diagram* Tambah Data Daftar Iuran



Gambar 35. Activity Diagram Edit Data Daftar Iuran

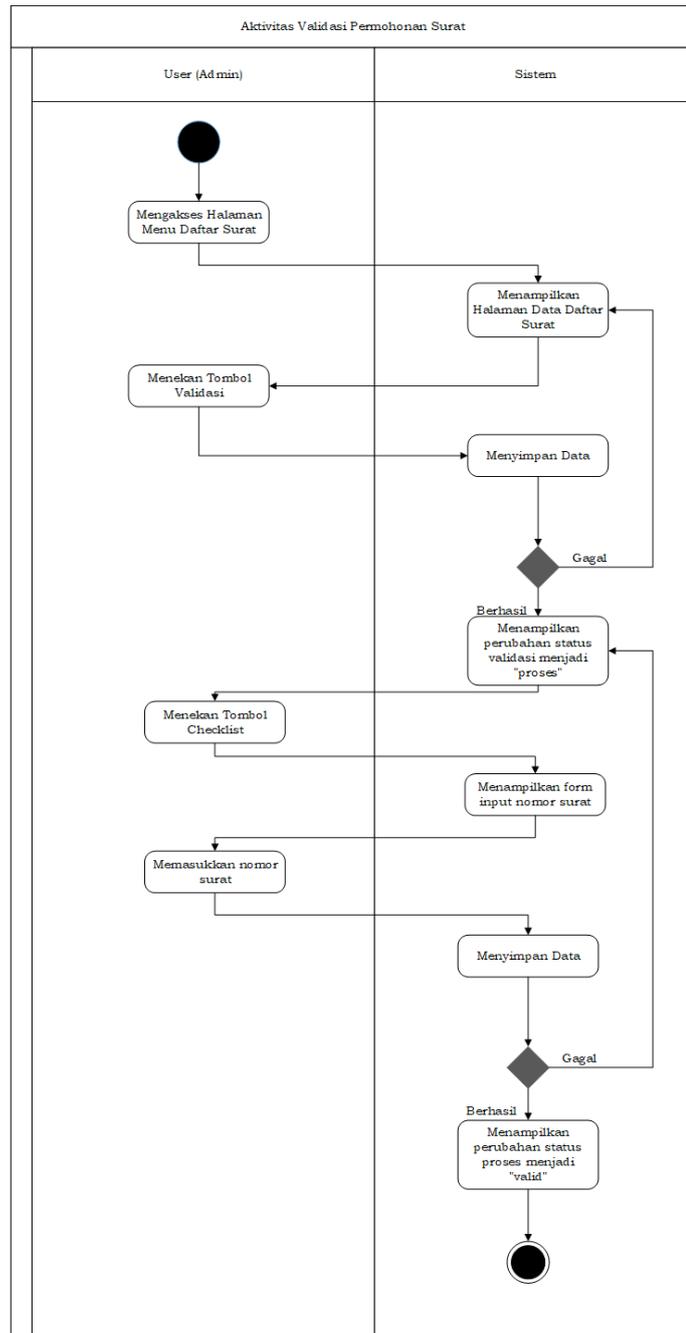
Pada Gambar 34 dan 35 diatas, terlihat aktivitas pengelolaan data daftar iuran meliputi tambah dan edit data dengan masing-masing aliran dimulai dari mengakses halaman menu dan diakhiri dengan penyimpanan data kedalam database sistem.

1. *Activity Diagram* Melihat Riwayat Pembayaran Iuran

Gambar 36. *Activity Diagram* Melihat Riwayat Pembayaran Iuran

Pada gambar 36 diatas, dapat dilihat aliran aktivitas melihat data riwayat pembayaran iuran yang dapat dilakukan sebatas akses melihat data oleh aktor warga dimana data dapat dilihat dengan mengakses menu riwayat pembayaran iuran terlebih dahulu, kemudian sistem akan menampilkan halaman berisikan data pembayaran iuran yang telah dilakukan untuk dapat dilihat oleh aktor pengguna sistem.

m. *Activity Diagram* Validasi Permohonan Surat

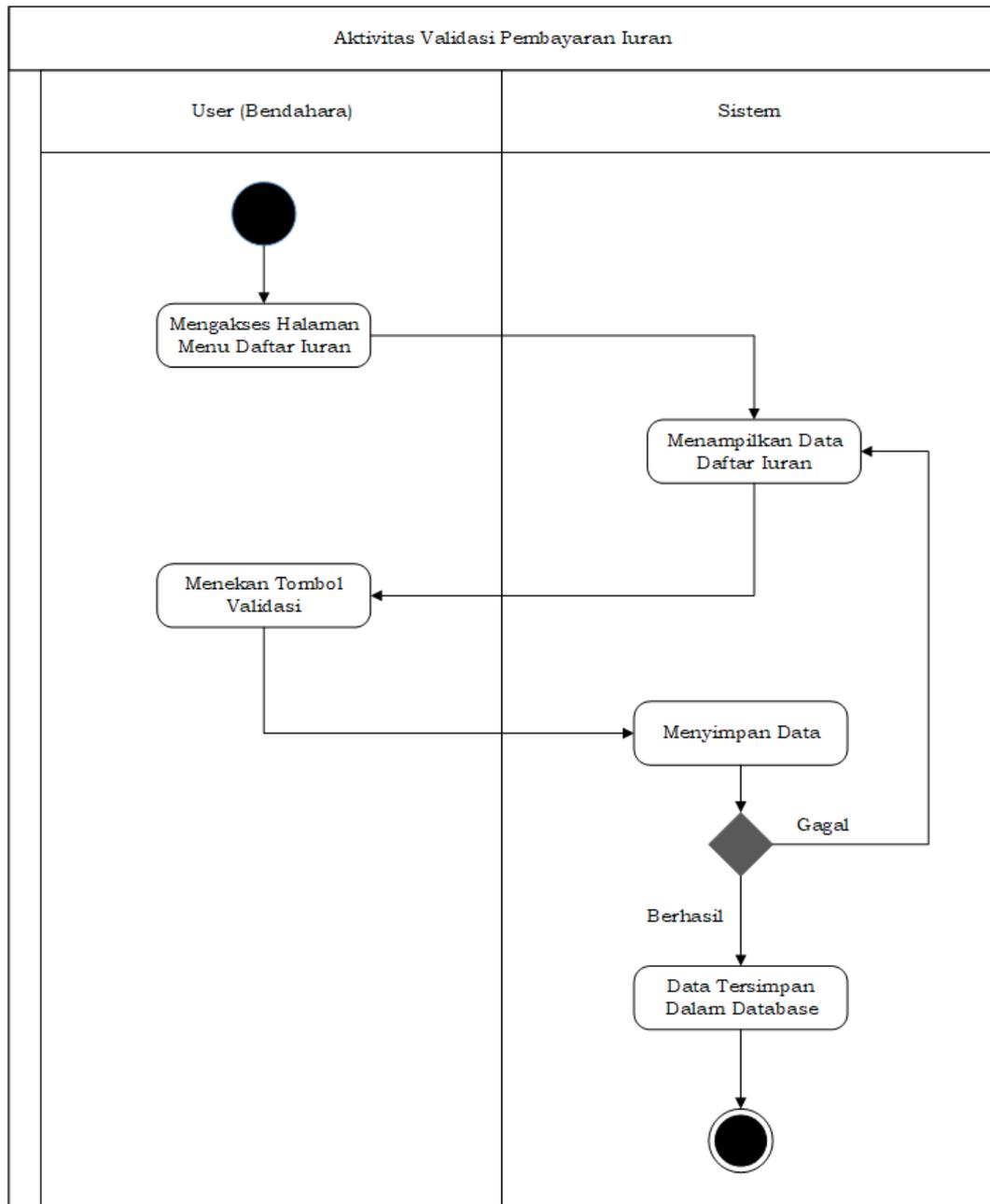


Gambar 37. *Activity Diagram* Validasi Permohonan Surat

Pada gambar 37 diatas, dapat dilihat aliran aktivitas validasi permohonan surat yang dilakukan oleh aktor admin dimulai dengan mengakses halaman menu permohonan surat kemudian ditampilkan data permohonan surat oleh sistem untuk selanjutnya akan di validasi jika data telah sesuai yang dapat ditekan tombol validasi kemudian sistem akan merubah statusnya menjadi proses. Setelah itu, admin dapat menekan tombol checklist jika surat sudah siap untuk diberikan pada warga perumahan yang kemudian admin perlu

menginputkan nomor surat. Jika nomor surat sudah diberikan, selanjutnya sistem akan merubah status menjadi valid dan admin dapat memasukkan dokumen surat kedalam sitem.

n. *Activity Diagram* Validasi Pembayaran Iuran

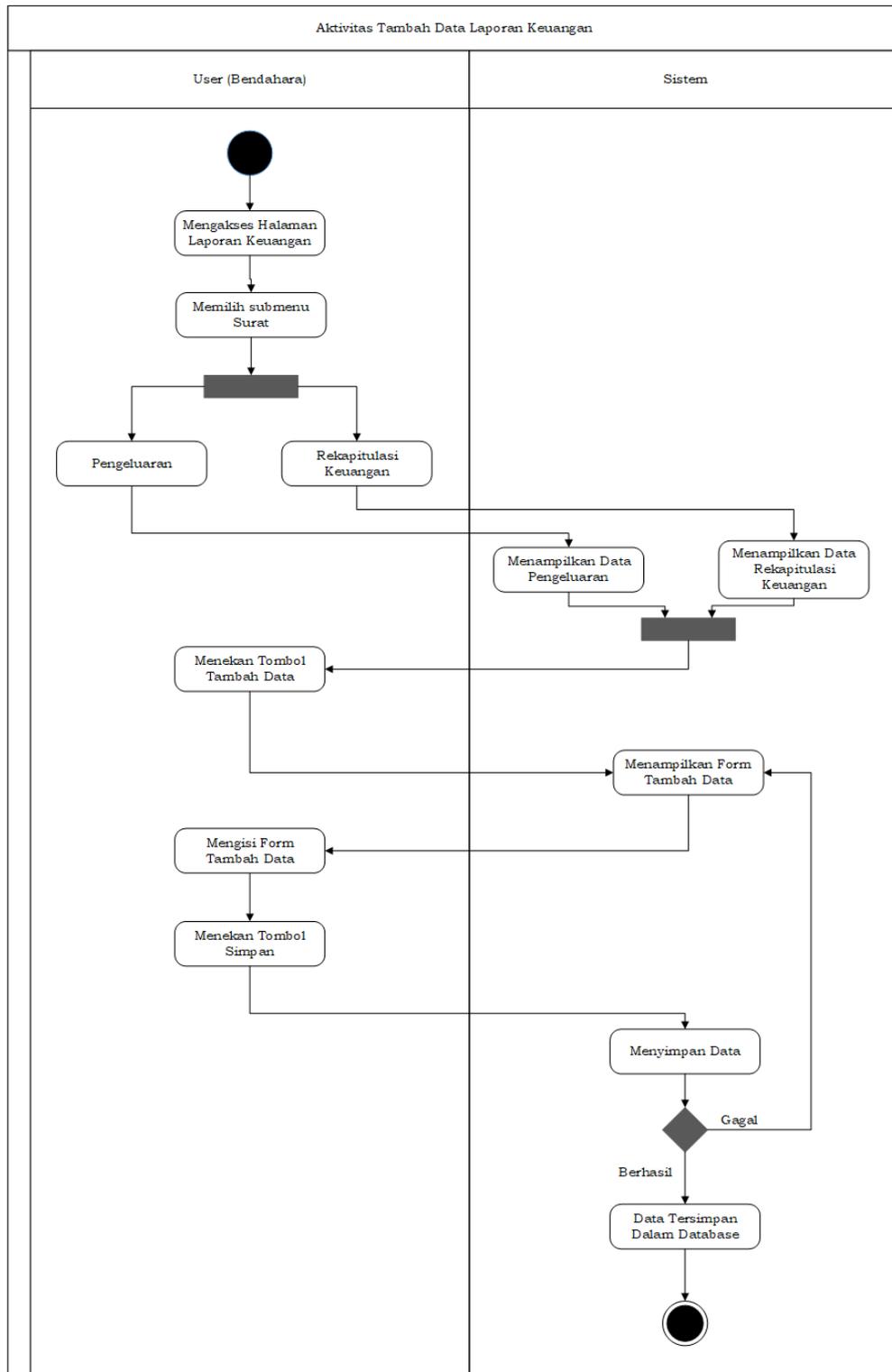


Gambar 38. *Activity Diagram* Validasi Pembayaran Iuran

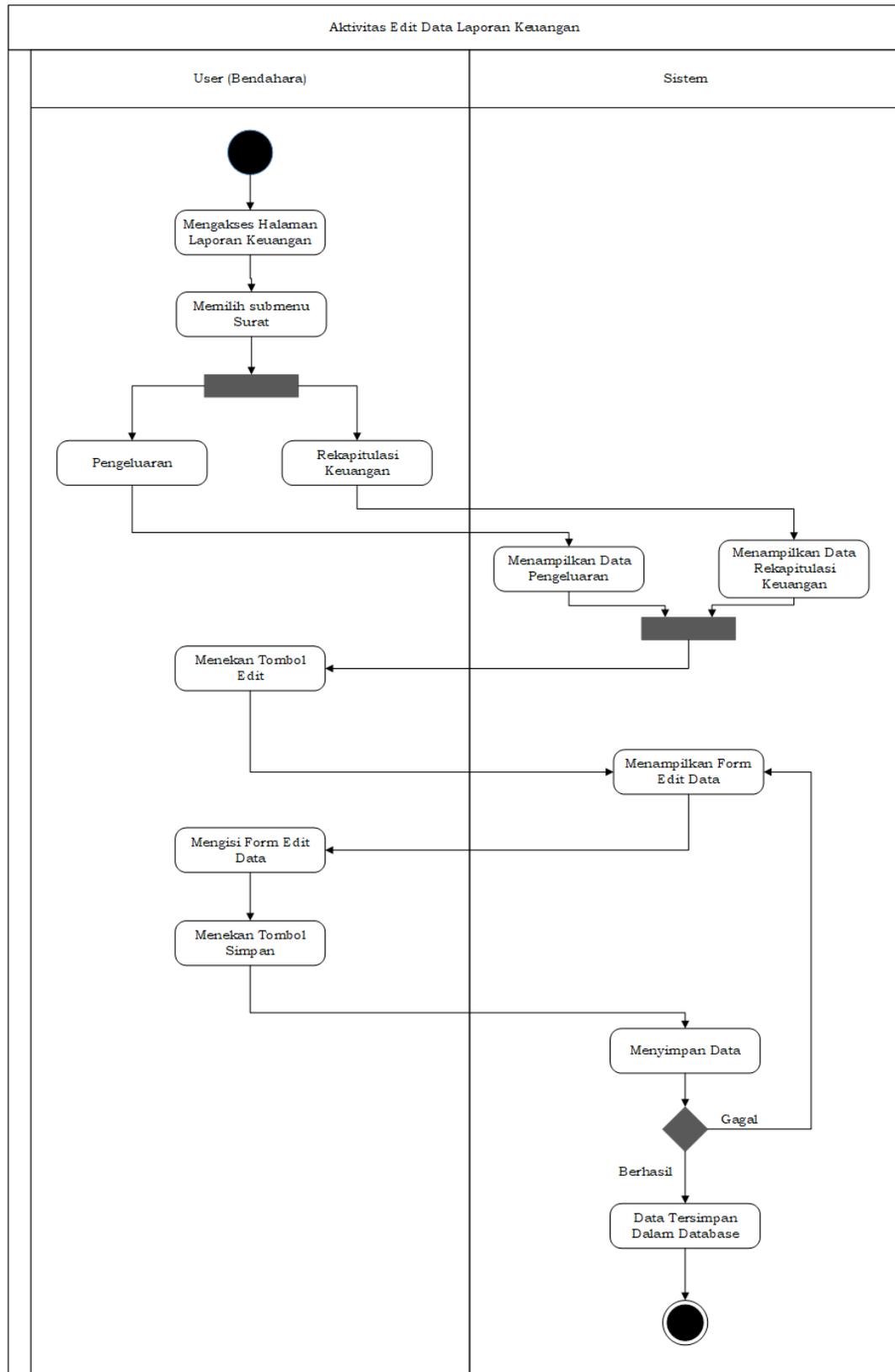
Pada gambar 38 diatas, dapat dilihat aliran aktivitas validasi pembayaran iuran yang dilakukan oleh aktor bendahara dimulai dengan mengakses halaman menu daftar iuran kemudian ditampilkan data daftar iuran yang telah dilakukan warga perumahan oleh sistem. Selanjutnya, aktor dapat memilih data

yang ingin di cek untuk selanjutnya dapat divalidasi dengan menekan tombol validasi. Setelah itu, sistem akan menyimpan data, jika gagal maka akan ditampilkan kembali oleh sistem halaman menu data pembayaran iuran.

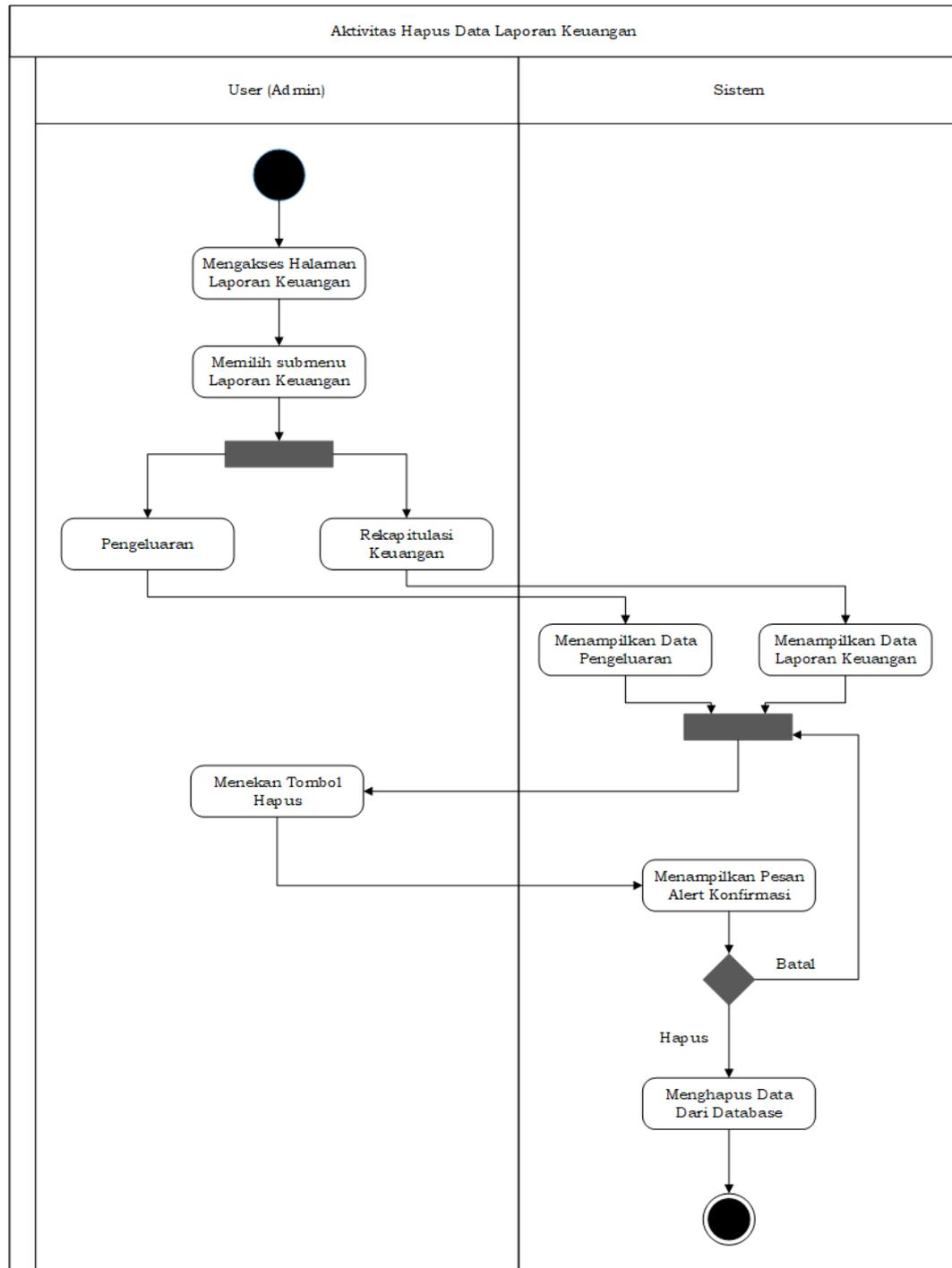
o. *Activity Diagram* Kelola Laporan Keuangan



Gambar 39. *Activity Diagram* Tambah Data Laporan Keuangan



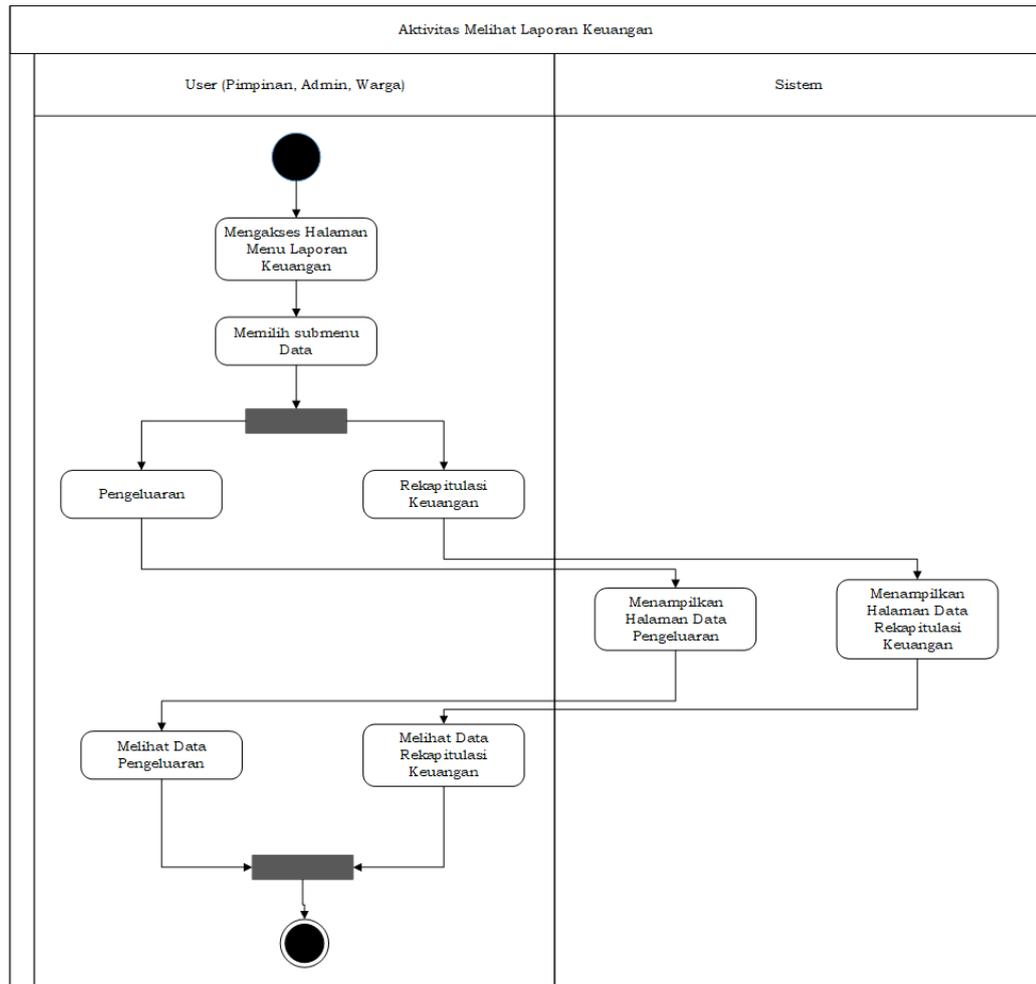
Gambar 40. Activity Diagram Edit Data Laporan Keuangan



Gambar 41. Activity Diagram Hapus Data Laporan Keuangan

Pada Gambar 39-41 diatas, terlihat aktivitas pengelolaan data laporan keuangan meliputi tambah, edit, dan hapus data dengan masing-masing aliran dimulai dari mengakses halaman menu dan diakhiri dengan penyimpanan data kedalam database sistem.

p. *Activity Diagram* Melihat Laporan Keuangan

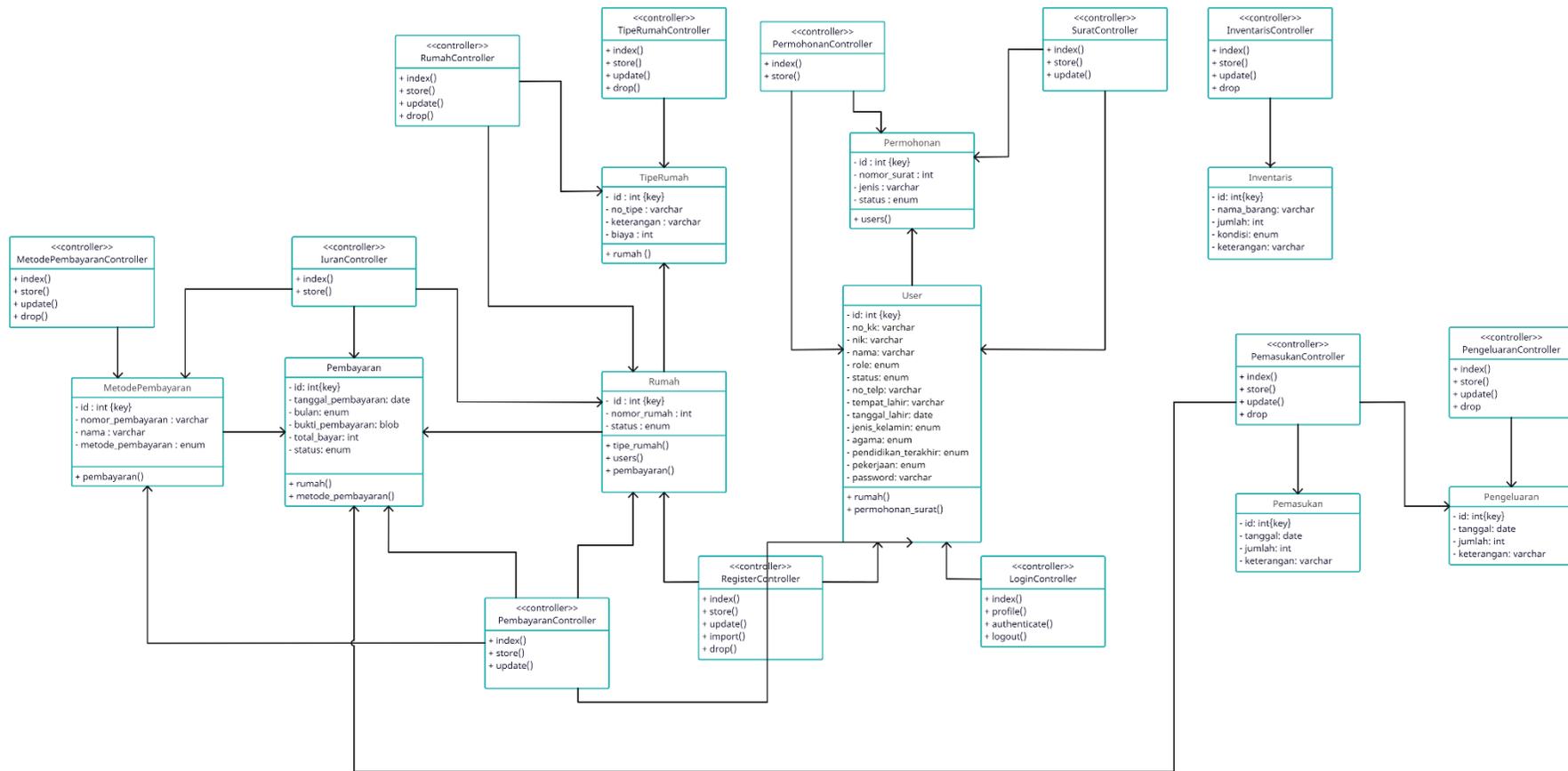


Gambar 42. *Activity Diagram* Melihat Laporan Keuangan

Pada gambar 42 diatas, dapat dilihat aliran aktivitas melihat laporan keuangan yang dapat dilakukan sebatas akses melihat data oleh aktor pimpinan, admin, dan warga dimana data dapat dilihat dengan mengakses menu laporan keuangan terlebih dahulu lalu submenu yang ingin diakses, kemudian sistem akan menampilkan halaman berisikan data laporan keuangan yang dapat dilihat oleh aktor pengguna sistem.

Rancang Diagram Kelas Secara Detail

Diagram kelas atau yang umumnya dikenal dengan *Class diagram* merupakan diagram yang memberikan gambaran terkait struktur pemodelan kelas termasuk atribut, operasi, dan hubungan beserta asosiasinya terhadap kelas lainnya dengan menyediakan tampilan statis atau struktural dari sebuah sistem. *Class diagram* menjadi *blueprint* alur *database* yang akan dibuat pada sistem (Sandfreni et al., 2021). Adapun rancangan *Class diagram* yang dibuat untuk sistem SIPGAMA adalah sebagai berikut :



Gambar 43. Class Diagram SIPGAMA

Terlihat pada gambar 43 diatas, terdapat beberapa tabel atau *Class* penyusun sistem SIPGAMA beserta 12 *controller* dengan *Class* yang diantaranya terdiri dari:

a. *Class Users*

Class users merupakan kelas yang menyimpan data terkait pengguna sistem termasuk didalamnya *username* dan *password* yang menjadi kunci untuk dapat masuk kedalam sistem, serta *role* yang berfungsi untuk mengendalikan hak akses setiap pengguna sistem SIPGAMA yang terdiri dari 4 *role* yaitu Admin, Pimpinan, Bendahara, dan Warga.

Tabel 28. *Class Users*

No.	Nama Field	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	no_kk	Varchar	16	
3.	nik	Varchar	16	
4.	nama	Varchar	50	
5.	role	Enum		
6.	tempat_lahir	Varchar	35	
7.	tanggal_lahir	Date		
8.	jenis_kelamin	Enum		
9.	agama	Enum		
10.	pendidikan_terakhir	Enum		
11.	pekerjaan	Enum		
12.	no_telp	Varchar	13	
13.	Password	Varchar	16	
14.	Id_rumah	Int	11	FK

b. *Class Rumah*

Class rumah merupakan kelas yang menyimpan data terkait detail rumah seperti nomor rumah beserta ukuran pada perumahan klaster Villa Gading Mayang Kota Jambi yang dapat menjadi index penentu besarnya jumlah pembayaran iuran warga. *Class* ini juga memberikan informasi terkait status rumah tersebut apakah dihuni atau kosong.

Tabel 29. *Class Rumah*

No.	Nama Field	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	nomor_rumah	Int	11	
3.	Status	Enum		
4.	Id_tipe_rumah	Int	11	FK

c. *Class Tipe Rumah*

Class tipe rumah merupakan kelas yang menyimpan data terkait informasi tipe rumah dalam perumahan klaster Villa Gading Mayang Kota Jambi meliputi *field* keterangan yang menyimpan tipe rumah seperti kios, rumah, atau ruko. Kemudian terdapat *field* biaya yang menyimpan data terkait besarnya jumlah biaya iuran setiap tipe rumah.

Tabel 30. *Class Tipe Rumah*

No.	Nama Field	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	nomor_tipe	Varchar	15	
3.	keterangan	Varchar	50	
4.	biaya	Int	11	

d. *Class Pembayaran Iuran*

Class pembayaran iuran merupakan kelas yang menyimpan data terkait pembayaran iuran yang dilakukan oleh warga perumahan klaster Villa Gading Mayang Kota Jambi diantaranya tanggal pembayaran, bulan yang dibayarkan, total pembayaran beserta bukti bayar yang harus diberikan. *Class* ini juga menyimpan data status validasi pembayaran iuran yang akan divalidasi oleh aktor Bendahara.

Tabel 31. *Class Pembayaran Iuran*

No.	Nama Field	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	tanggal_pembayaran	Date		
3.	Bulan	Enum		
4.	bukti_pembayaran	Blob		
5.	total_bayar	Int	11	

6.	Status	Enum		
7.	id_rumah	Int	11	FK
8.	id_metode_pembayaran	Int	11	FK

e. *Class Metode Pembayaran*

Class metode pembayaran merupakan kelas yang menyimpan data berbagai metode pembayaran beserta nama akun beserta nomor pembayaran yang dapat digunakan oleh warga untuk melakukan pembayaran iuran.

Tabel 32. *Class Metode Pembayaran*

No.	Nama Field	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	nomor_pembayaran	Int	11	
3.	nama	Varchar	50	
4.	metode_pembayaran	enum		

f. *Class Permohonan Surat*

Class permohonan surat merupakan kelas yang menyimpan data pengajuan permohonan surat pengantar rt internal perumahan klaster Villa Gading Mayang yang dilakukan oleh warga berisikan nomor surat dan jenis surat pengantar seperti domisili, keterangan tidak mampu dan sebagainya. Kemudian terdapat *field* status yang menyimpan perkembangan status validasi ajuan permohonan surat dilakukan oleh aktor Admin yaitu sekretaris perumahan.

Tabel 33. *Class Permohonan Surat*

No.	Nama Field	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	nomor_surat	Int	15	
3.	jenis	Varchar	50	
4.	status	Enum		
5.	id_user	Int	11	FK

g. *Class Inventaris*

Class inventaris merupakan kelas yang menyimpan data daftar barang inventaris yang dimiliki oleh perumahan klaster Villa Gading Mayang Kota Jambi berisikan nama, jumlah, serta keterangan barang tersebut. Kemudian terdapat *field* kondisi yang menyimpan keadaan barang apakah rusak, hilang, atau telah diganti dengan barang terbaru.

Tabel 34. *Class Inventaris*

No.	Nama <i>Field</i>	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	nama_barang	Varchar	15	
3.	jumlah	Int	11	
4.	kondisi	enum		
5.	keterangan	Varchar	50	

h. *Class Pengeluaran*

Class pengeluaran merupakan kelas yang menyimpan data pengeluaran keuangan perumahan klaster Villa Gading Mayang Kota Jambi berisikan tanggal, jumlah, serta keterangan transaksi uang keluar dari dana kas perumahan. *Class* ini berfungsi memberikan transparansi laporan keuangan beserta index penentu hasil akhir rekapitulasi keuangan yang dimiliki perumahan.

Tabel 35. *Class Pengeluaran*

No.	Nama <i>Field</i>	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	tanggal	Date		
3.	jumlah	Int	11	
4.	keterangan	Varchar	50	

i. *Class Pemasukan*

Class pemasukan merupakan kelas yang menyimpan data pemasukan keuangan perumahan klaster Villa Gading Mayang Kota Jambi berisikan tanggal, jumlah, serta keterangan transaksi uang masuk yang berasal dari pembayaran iuran perumahan serta sumber dana lainnya seperti iuran, donasi, dan sebagainya. *Class* ini berfungsi memberikan transparansi laporan

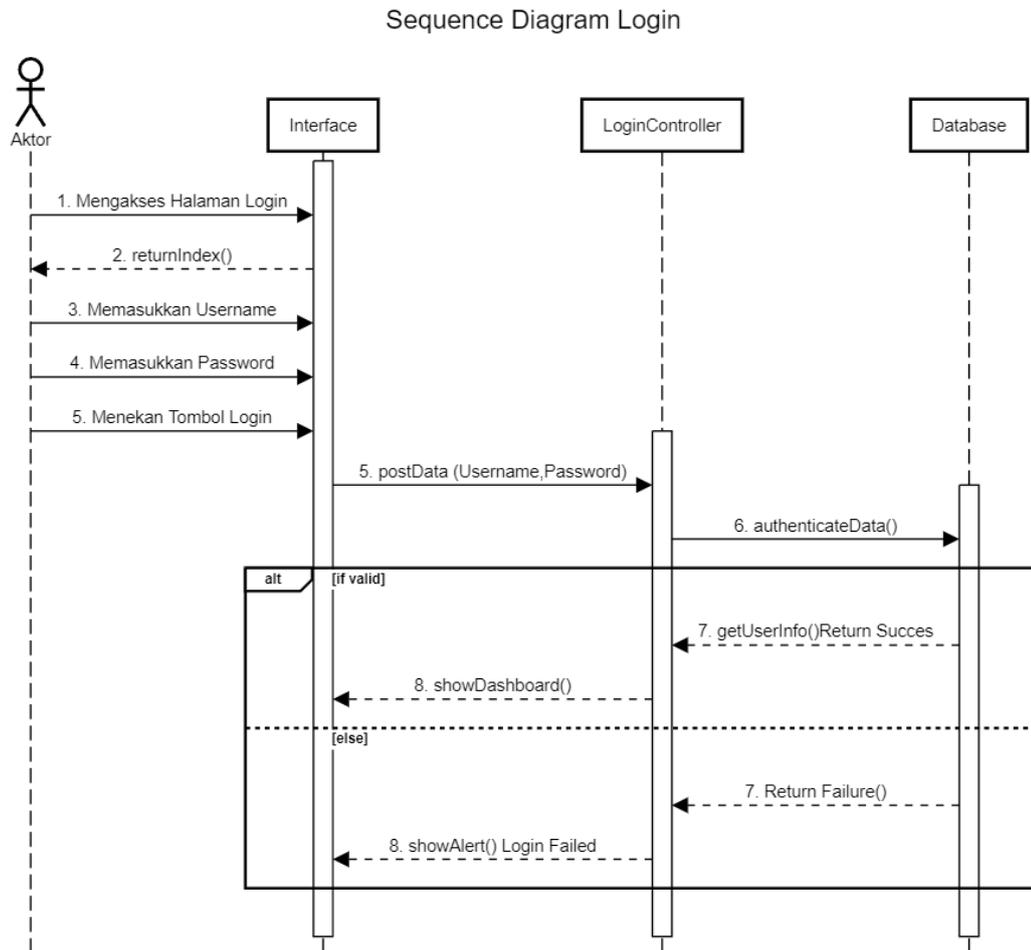
keuangan beserta index penentu hasil akhir rekapitulasi keuangan yang dimiliki perumahan.

Tabel 36. *Class Pemasukan*

No.	Nama Field	Tipe	Panjang	Indeks
1.	Id	Int	11	PK
2.	tanggal	Date		
3.	jumlah	Int	11	
4.	keterangan	Varchar	50	

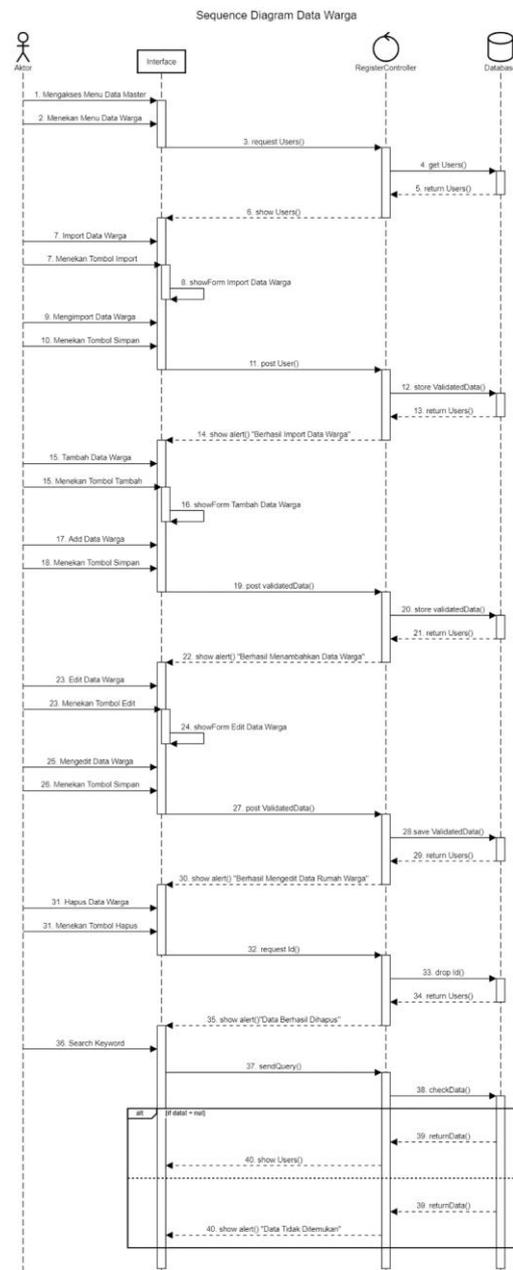
Rancang Diagram Urutan Untuk Setiap Skenario

Diagram urutan atau yang lebih umum dikenal dengan *Sequence Diagram* merupakan salah satu diagram UML *interaction* yang menggambarkan interaksi objek selama periode waktu tertentu yaitu *Sequence* atau urutan pesan antara aktor eksternal dengan sistem selama skenario *Use Case*. *Sequence Diagram* digunakan untuk melacak pelaksanaan skenario dengan keunggulan lebih baik dibandingkan *communication diagram* dalam menangkap skenario semantik pada proses awal pengembangan sebelum protokol untuk masing-masing kelas teridentifikasi sehingga lebih mudah membaca pertukaran pesan dengan urutan relatif. Berikut merupakan rancangan *Sequence Diagram* dari sistem SIPGAMA yang dapat dilihat dibawah ini :

a. *Sequence Diagram Login*

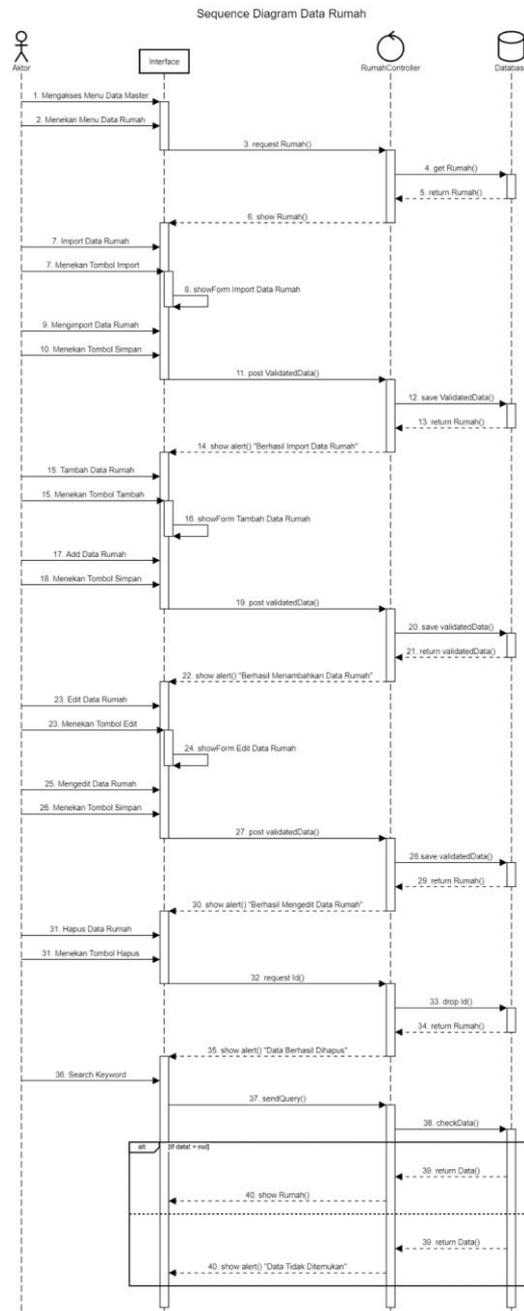
Gambar 44. *Sequence Diagram Login*

Pada gambar 44 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* saat melakukan login, dimana alur diawali dengan aktor mengakses halaman yang kemudian ditampilkan oleh interface, selanjutnya aktor dapat memasukkan *username* berupa NIK dan *password* dan menekan tombol login untuk selanjutnya diverifikasi sistem sesuai kecocokan data dalam database dengan perantara penghubung pesan yaitu *controller* LoginController. Jika data yang dimasukkan sesuai, maka interface akan menampilkan Dashboard, namun jika salah akan ditampilkan alert Login gagal.

b. *Sequence Diagram Data Master*

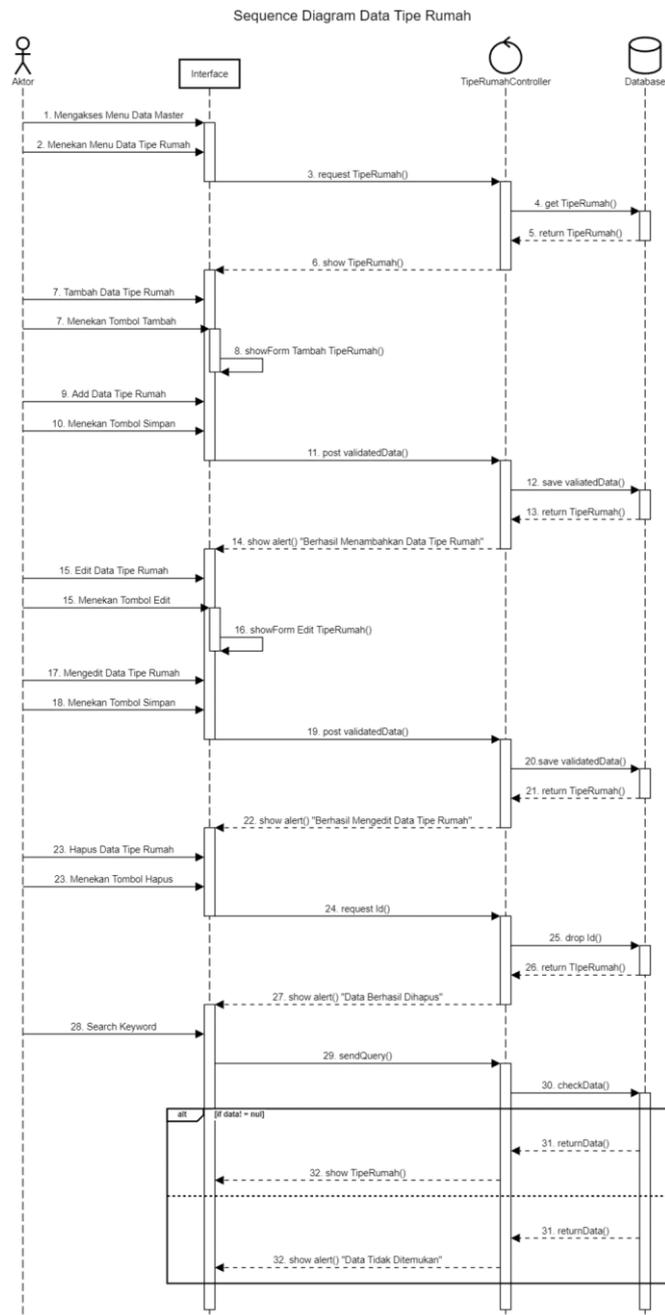
Gambar 45. *Sequence Diagram Data Master (Data Warga)*

Pada gambar 45 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada data master yaitu menu data warga dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data, import data, edit data, serta hapus data, dan pencarian data pada *search bar* dengan menggunakan *keyword* yang diinginkan dimana jika data terdapat dalam *database*, maka selanjutnya *interface* aka menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan.



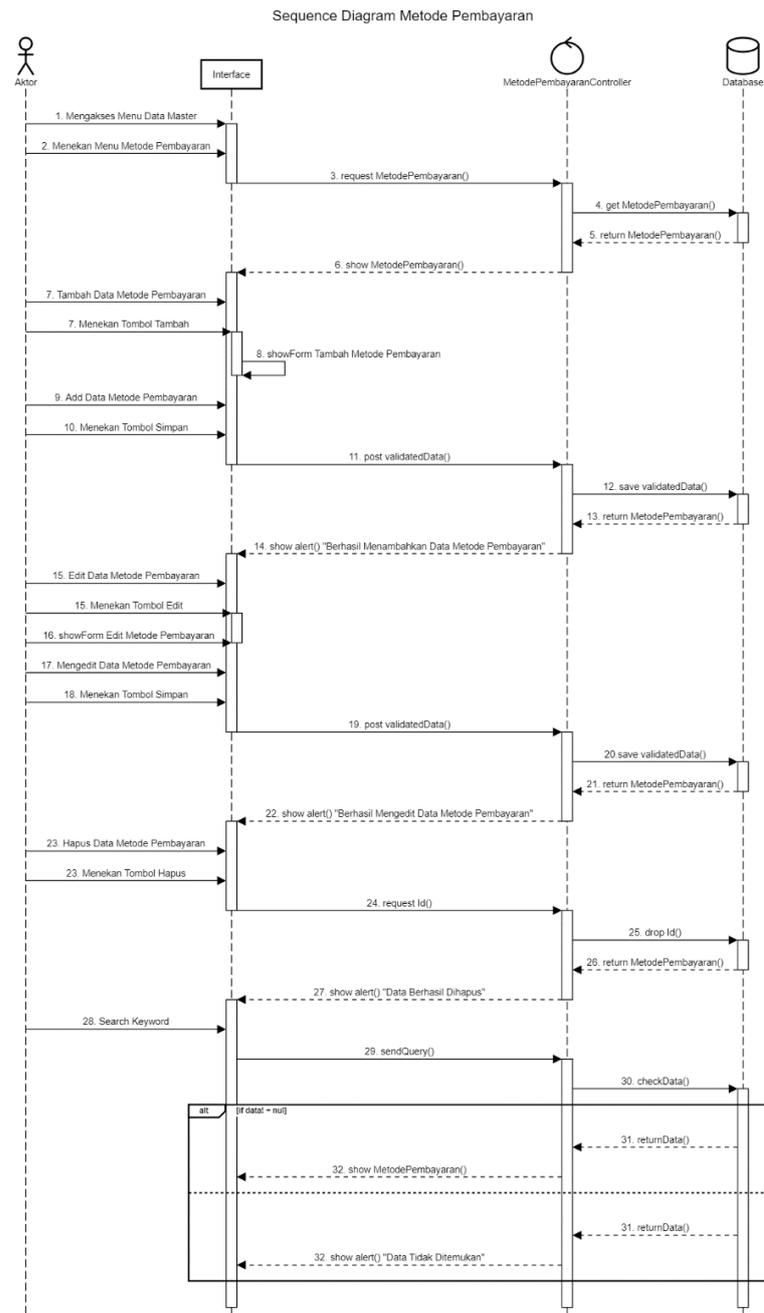
Gambar 46. Sequence Diagram Data Master (Data Rumah)

Pada gambar 46 diatas, dapat terlihat urutan alur interaksi antara aktor, interface, controller, beserta database pada data master yaitu menu data rumah dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data, import data, edit data, serta hapus data, dan pencarian data pada *search bar* dengan menggunakan keyword yang diinginkan dimana jika data terdapat dalam database, maka selanjutnya interface akan menampilkan data namun jika tidak ada yang cocok maka interface akan menampilkan pesan alert bahwa data tidak ditemukan.



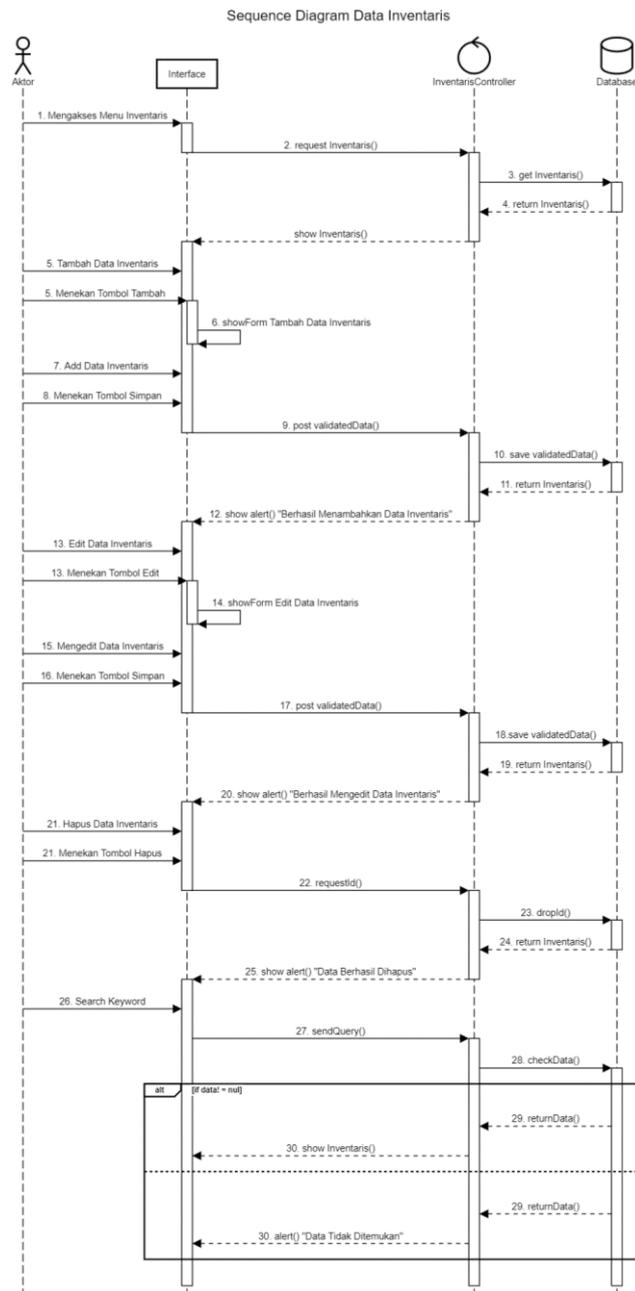
Gambar 47. Sequence Diagram Data Master (Data Tipe Rumah)

Pada gambar 47 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada data master yaitu menu data tipe rumah dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data, edit data, serta hapus data, dan pencarian data pada *search bar* dengan menggunakan *keyword* yang diinginkan dimana jika data terdapat dalam *database*, maka selanjutnya *interface* aka menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan.



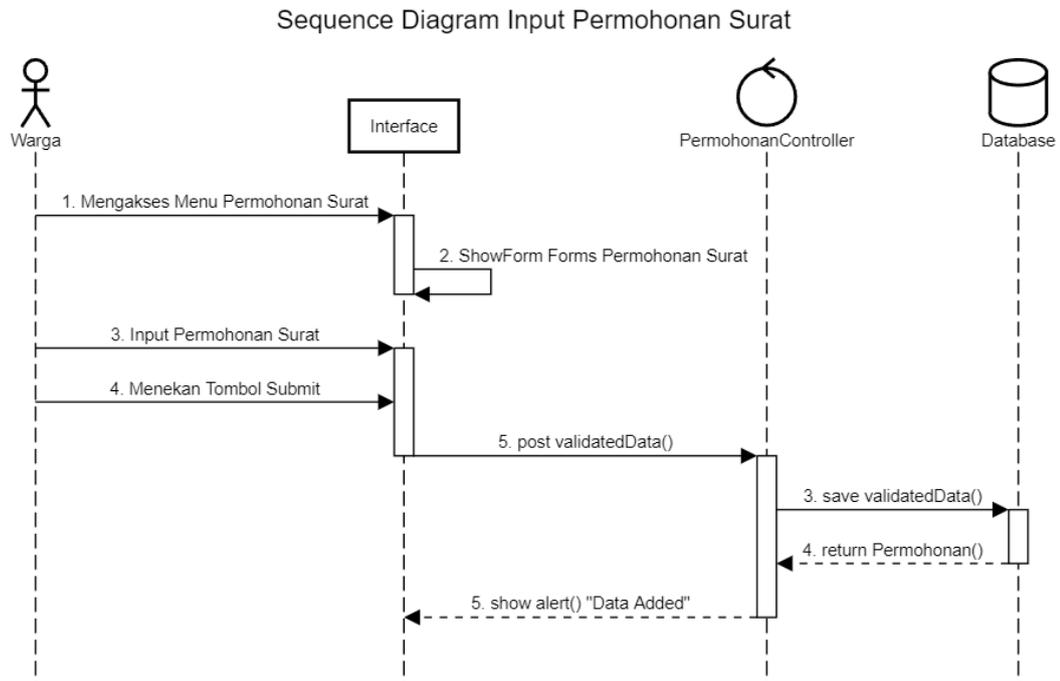
Gambar 48. Sequence Diagram Data Master (Metode Pembayaran)

Pada gambar 48 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada data master yaitu menu metode pembayaran dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data, edit data, serta hapus data, dan pencarian data pada *search bar* dengan menggunakan *keyword* yang diinginkan dimana jika data terdapat dalam *database*, maka selanjutnya *interface* aka menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan.

c. *Sequence Diagram Inventaris*

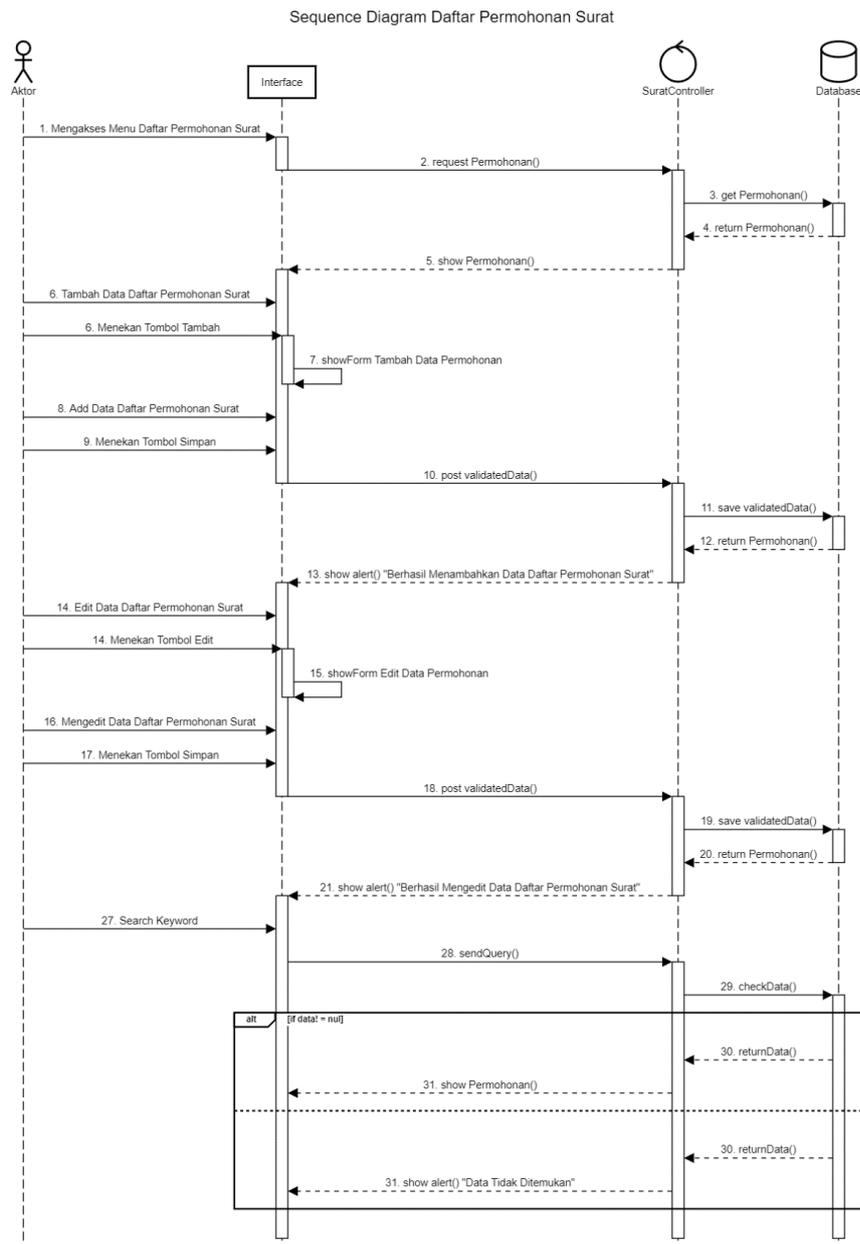
Gambar 49. *Sequence Diagram Inventaris*

Pada gambar 49 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* menu inventaris dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data, edit data, serta hapus data, dan pencarian data pada *search bar* dengan menggunakan *keyword* yang diinginkan dimana jika data terdapat dalam *database*, maka selanjutnya *interface* aka menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan.

d. *Sequence Diagram* Input Permohonan Surat

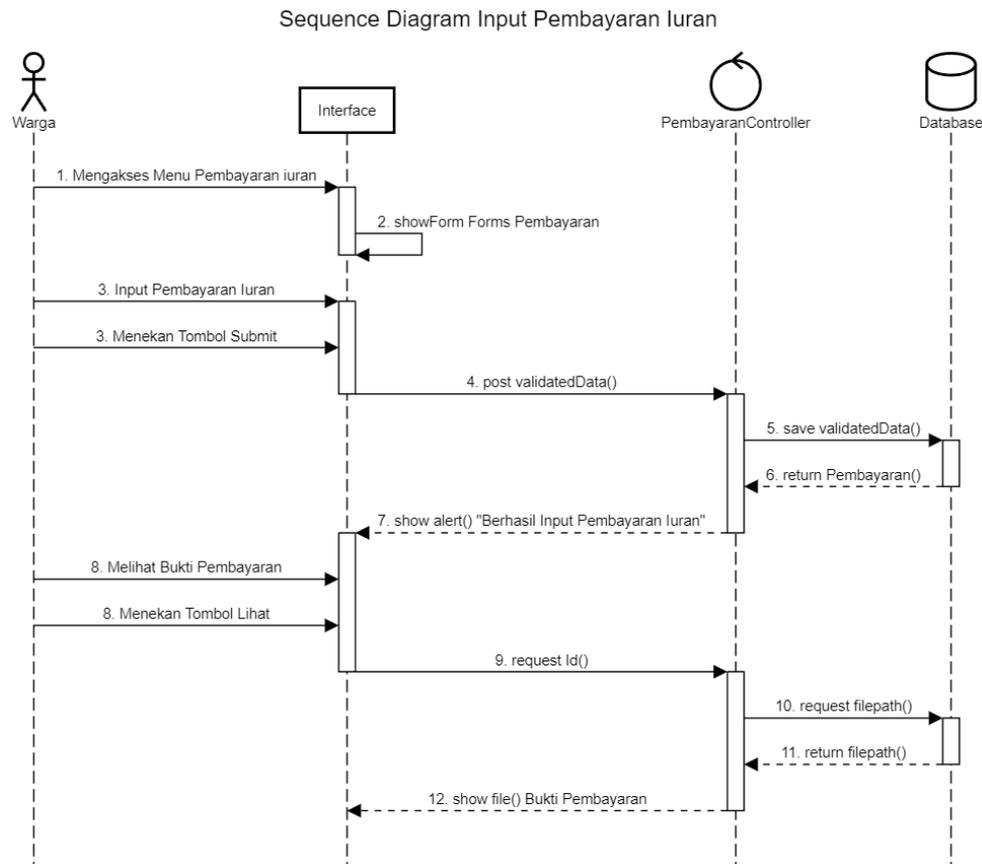
Gambar 50. *Sequence Diagram* Input Permohonan Surat

Pada gambar 50 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada proses *input* permohonan surat yang dilakukan oleh warga perumahan klaster Villa Gading Mayang Kota Jambi. Warga akan mengakses menu permohonan surat kemudian *interface* sistem akan menampilkan form *input* data yang perlu diisi, setelah selesai warga menekan tombol *submit* untuk *interface* meneruskan data kepada *database* melalui perantara *controller*. Selanjutnya *database* akan mengembalikan data permohonan sehingga *interface* menampilkan pesan bahwa data telah berhasil *diinputkan*.

e. *Sequence Diagram* Daftar Permohonan Surat

Gambar 51. *Sequence Diagram* Daftar Permohonan Surat (Permohonan Surat)

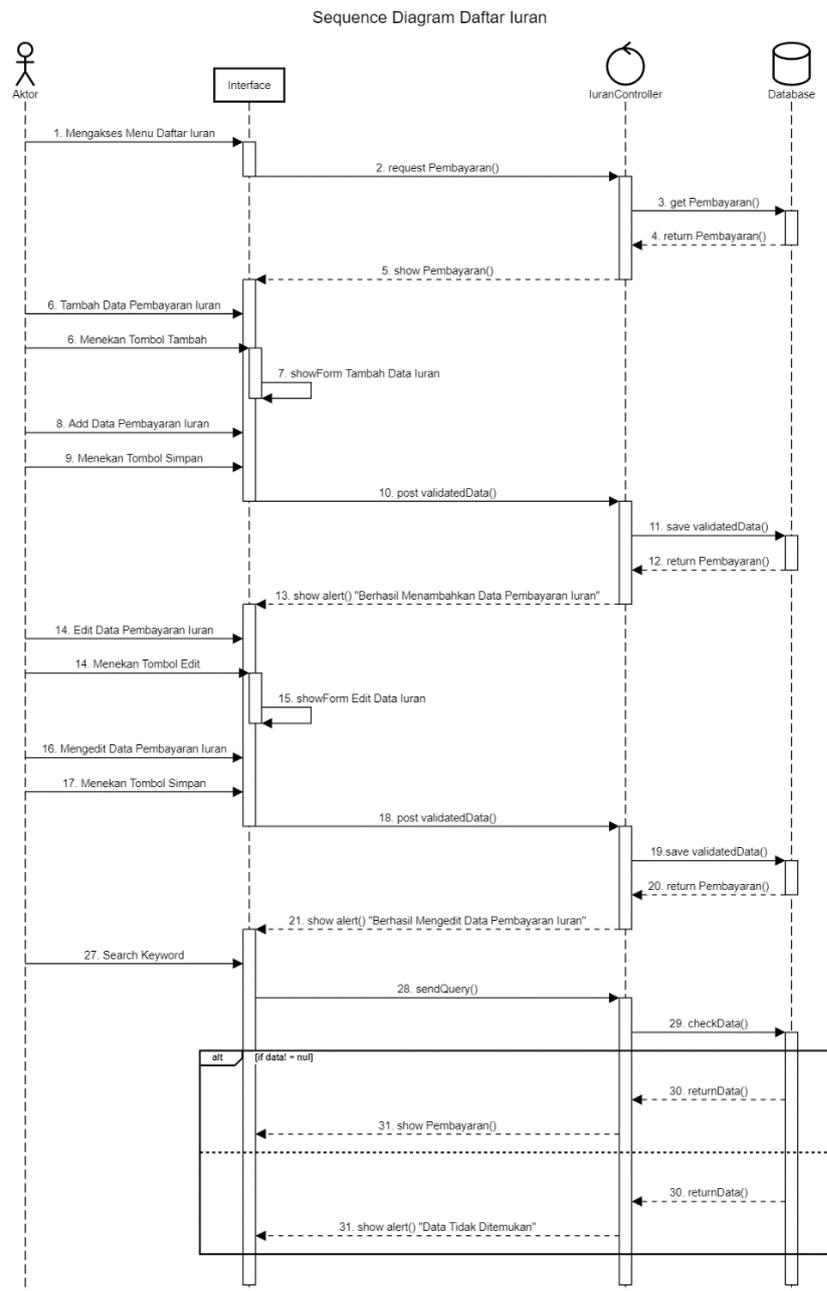
Pada gambar 51 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada menu daftar permohonan surat dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data dan edit data, serta pencarian data pada *search bar* dengan menggunakan *keyword* yang diinginkan dimana jika data terdapat dalam *database*, maka selanjutnya *interface* aka menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan.

f. *Sequence Diagram* Input Pembayaran Iuran

Gambar 52. *Sequence Diagram* Input Pembayaran Iuran

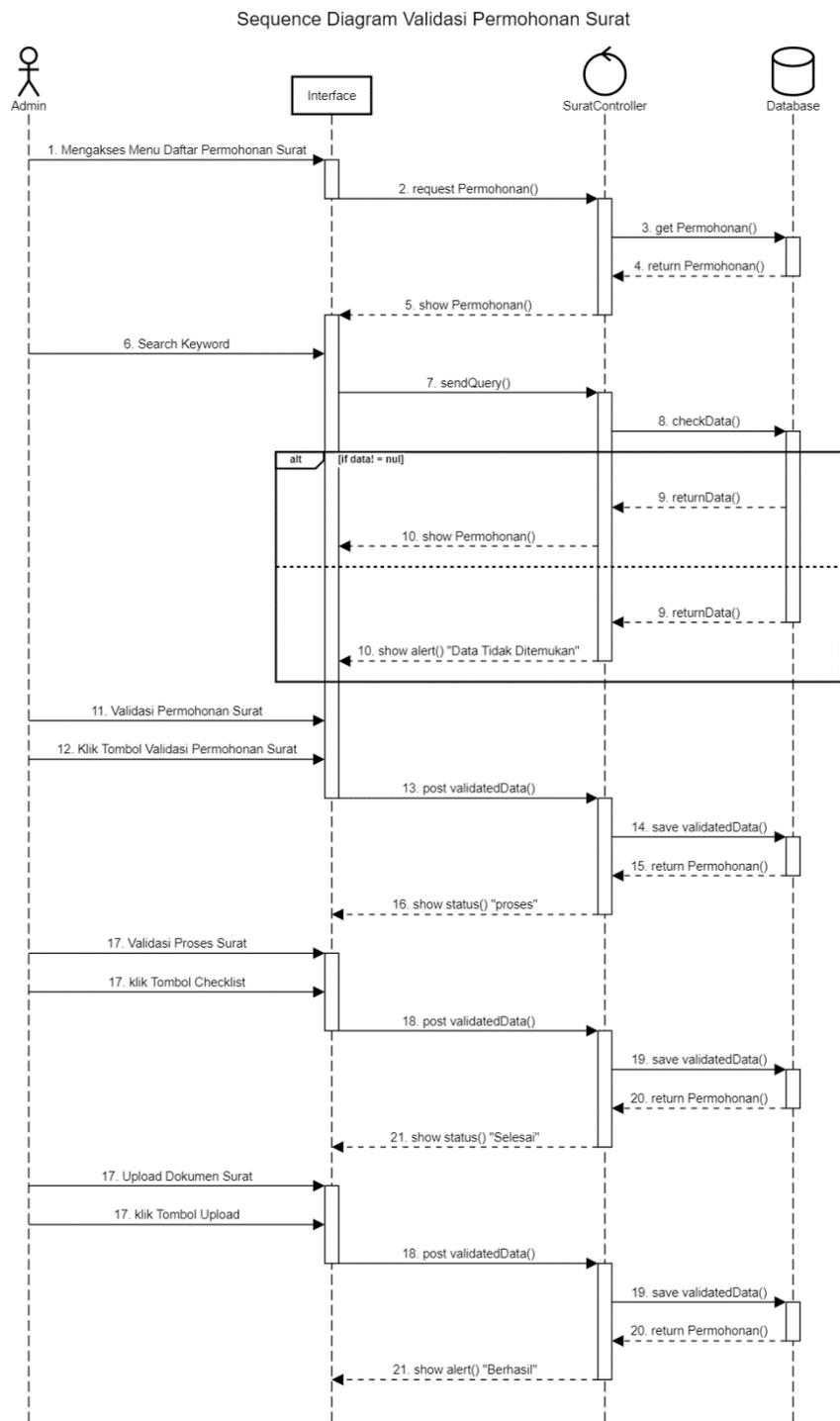
Pada gambar 52 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada proses *input* pembayaran iuran yang dilakukan oleh warga perumahan klaster Villa Gading Mayang Kota Jambi. Warga akan mengakses menu permohonan surat kemudian *interface* sistem akan menampilkan form *input* data yang perlu diisi termasuk bukti pembayaran iuran, setelah selesai warga menekan tombol *submit* untuk *interface* meneruskan data kepada *database* melalui perantara *controller*. Selanjutnya *database* akan mengembalikan data permohonan sehingga *interface* menampilkan pesan bahwa data telah berhasil *diinputkan*. Warga juga dapat melihat bukti pembayaran yang sudah *diinputkan* dengan menekan tombol lihat yang selanjutnya diteruskan oleh *interface* kepada *controller* lalu *controller* meneruskan kepada *database*, dan *database* akan mengembalikan file pembayaran untuk ditampilkan oleh *interface*.

g. Sequence Diagram Daftar Iuran



Gambar 53. Sequence Diagram Daftar Iuran

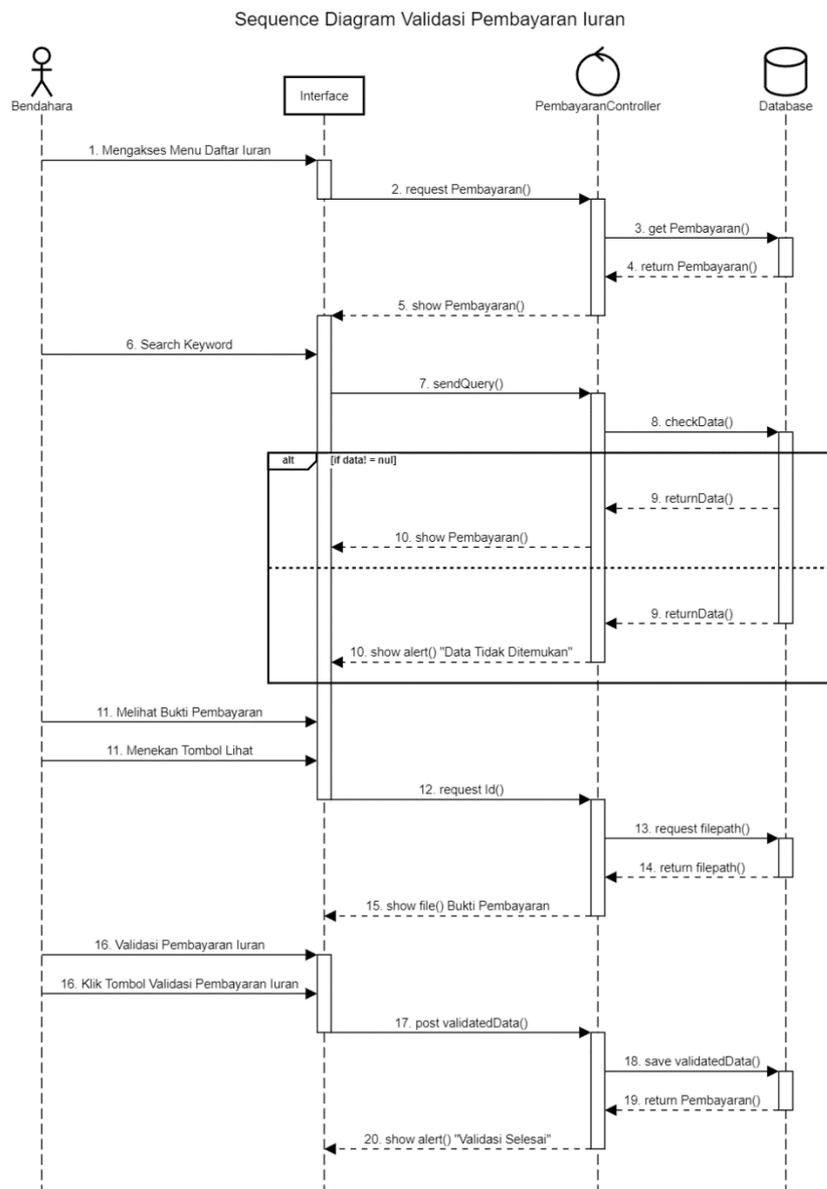
Pada gambar 53 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada menu data daftar iuran dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data dan edit data, serta pencarian data pada *search bar* dengan menggunakan *keyword* yang diinginkan dimana jika data terdapat dalam *database*, maka selanjutnya *interface* aka menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan.

h. *Sequence Diagram Validasi Data*

Gambar 54. *Sequence Diagram Validasi Permohonan Surat*

Pada gambar 54 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada validasi pengajuan permohonan surat yang *diinputkan* oleh warga perumahan klaster Villa Gading Mayang Kota Jambi. Admin sebagai validator dapat mencari data yang ingin ditemukan dengan *search bar* dimana jika data terdapat dalam *database*, maka selanjutnya

interface akan menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan. Admin melakukan validasi awal dengan menekan tombol valid yang kemudian interface akan menampilkan perubahan status validasi lanjutna menjadi proses. Setelah admin memastikan nomor surat beserta keseluruhan surat telah siap maka admin dapat menekan tombol *checklist* untuk menyelesaikan proses validasi. Admin dapat memasukkan *file* pdf surat yang sudah dibuat untuk dapat diunduh oleh warga yang menjadi arsip pada sistem.

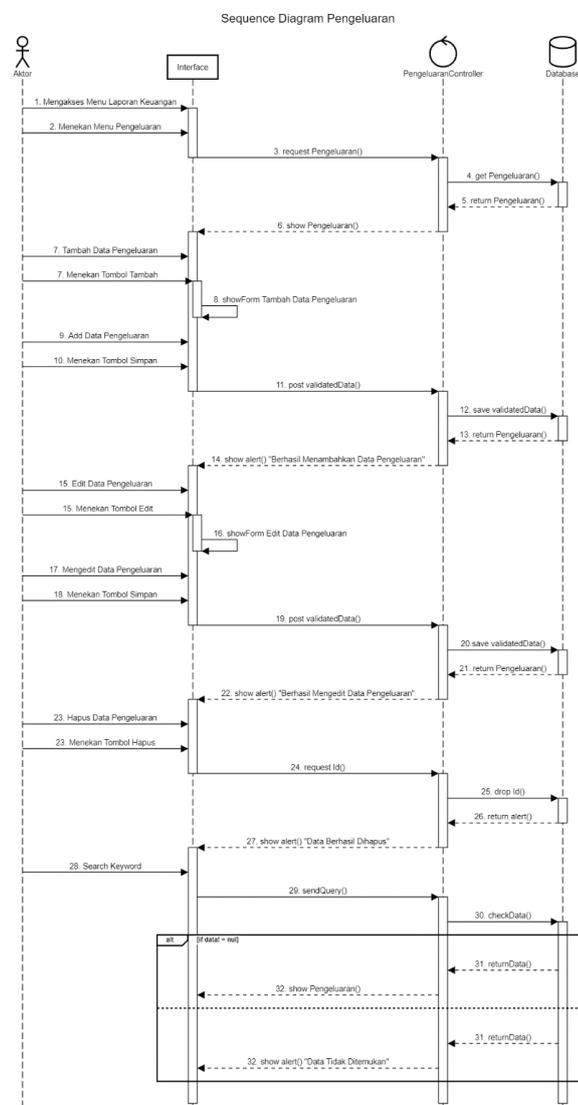


Gambar 55. Sequence Diagram Validasi Pembayaran Iuran

Pada gambar 55 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada validasi pembayaran iuran yang

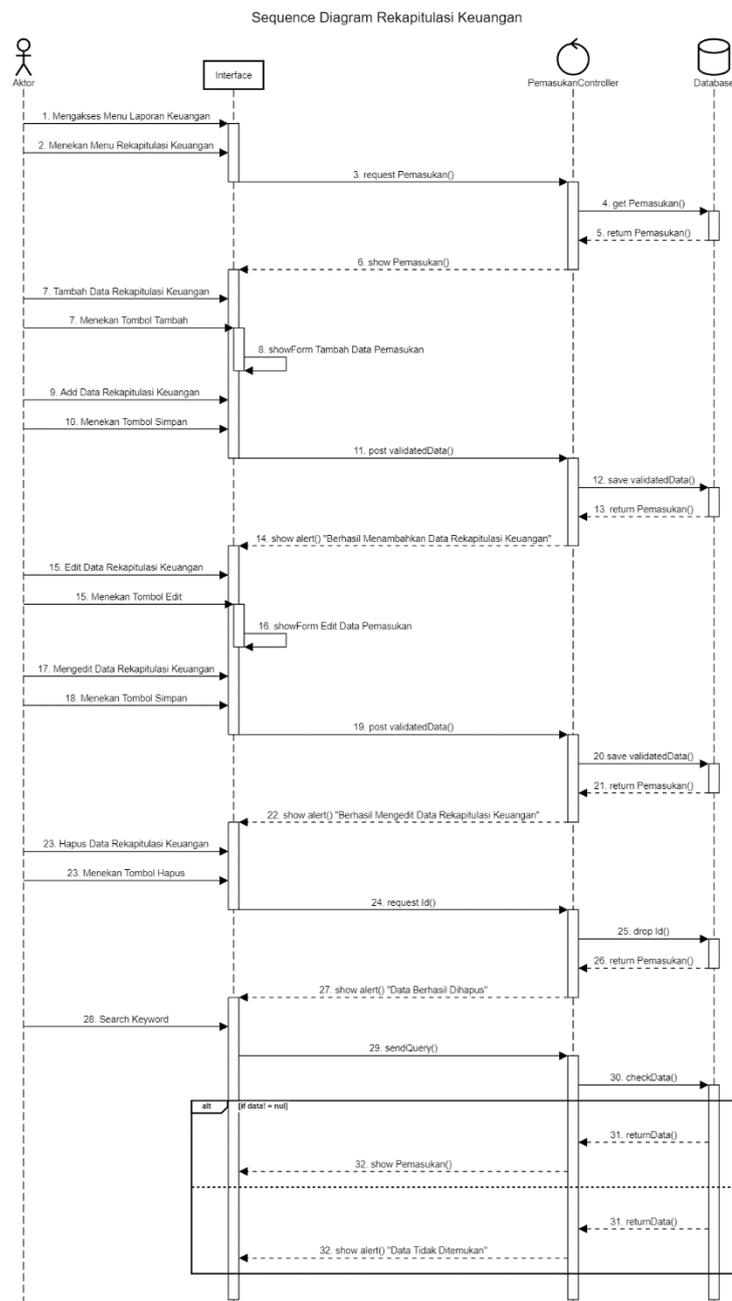
dilakukan oleh warga perumahan klaster Villa Gading Mayang Kota Jambi. Bendahara sebagai validator dapat mencari data yang ingin ditemukan dengan *search bar* dimana jika data terdapat dalam *database*, maka selanjutnya *interface* akan menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan. Admin melakukan validasi dengan mengecek data yang diinputkan warga dan foto bukti pembayaran yang telah sesuai dengan selanjutnya menekan tombol validasi dan *interface* akan mengirimkan data pada *controller* untuk diteruskan pada *database* yang akan mengembalikan pesan kepada *interface* bahwa validasi selesai.

i. *Sequence Diagram* Laporan Keuangan



Gambar 56. *Sequence Diagram* Laporan Keuangan (Pengeluaran)

Pada gambar 56 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada laporan keuangan yaitu menu pengeluaran dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data, edit data, serta hapus data, dan pencarian data pada *search bar* dengan menggunakan *keyword* yang diinginkan dimana jika data terdapat dalam *database*, maka selanjutnya *interface* aka menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan.



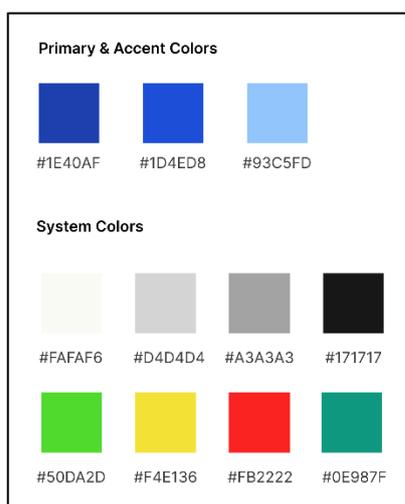
Gambar 57. Sequence Diagram Laporan Keuangan (Rekapitulasi Keuangan)

Pada gambar 57 diatas, dapat terlihat urutan alur interaksi antara aktor, *interface*, *controller*, beserta *database* pada laporan keuangan yaitu menu rekapitulasi keuangan dengan beberapa aktivitas yang dapat dilakukan yaitu melihat data, kelola data meliputi tambah data, edit data, serta hapus data, dan pencarian data pada *search bar* dengan menggunakan *keyword* yang diinginkan dimana jika data terdapat dalam *database*, maka selanjutnya *interface* aka menampilkan data namun jika tidak ada yang cocok maka *interface* akan menampilkan pesan *alert* bahwa data tidak ditemukan.

Rancang Tampilan Desain Antarmuka

Desain Antarmuka atau yang biasa dikenal dengan UI (*User Interface*) merupakan media komunikasi antara pengguna dengan sistem yang termasuk kedalam bagian perangkat lunak sistem dimana cakupannya meliputi tampilan fisik, penggunaan warna, tampilan animasi, hingga pola komunikasi dengan pengguna. *User Interface* dirancang berfokus kepada tingkat kebutuhan dasar pengguna pada sistem sehingga didapatkan sistem yang mudah digunakan (*user friendly*) (Himawan & Yanu F, 2020).

Perancangan desain *interface* sistem SIPGAMA diawali dengan penentuan warna yang berkesesuaian dengan kebutuhan dan keinginan pengguna sistem. Terdapat beberapa kelompok warna yang digunakan dalam perancangan desain SIPGAMA pada penelitian ini yaitu dengan menggunakan rangkaian yang dapat dilihat sebagai berikut :



Gambar 58. Palet Warna Desain Sistem SIPGAMA

Pemilihan warna utama yang akan diterapkan dalam sistem SIPGAMA diatas berkesesuaian dengan permintaan pengguna dan berlandaskan psikologi warna yang baik untuk penggunaan sistem dengan warna dominan biru. Setelah didapatkan warna yang akan digunakan dalam tampilan sistem, selanjutnya

dapat dilanjutkan tahapan utama dari rancangan *user interface* sistem yaitu pembuatan *Wireframe* dan *prototype*.

a. *Wireframe*

Dalam perancangan desain suatu *interface* sistem, perlu dibuat sebuah *Wireframe* yang berdasarkan (A. Hidayat & Meilina Fauziyyah, 2022) merupakan kerangka dasar dalam penyusunan tata komponen-komponen suatu *website* atau aplikasi untuk memberikan pandangan awal bentuk *interface* dari sistem yang akan dibangun sehingga mempermudah *developer* dalam mengimplementasikan hasil rancangan antarmuka. Beberapa komponen yang disusun diantaranya berupa *banner*, *header*, *content*, *footer*, *form*, dan lainnya. Perancangan *Wireframe* dalam penelitian ini terbagi menjadi 2 diantaranya *Wireframe Low-Fidelity* dan *Wireframe high-fidelity*. *Wireframe* yang dirancang akan mencakup keseluruhan desain berdasarkan pengguna SIPGAMA yaitu admin, pimpinan, bendahara, dan warga.

1. *Wireframe Low-Fidelity*

Sebelum menjadi sebuah desain yang utuh, rancangan interface dibuat dengan *Wireframe* presisi rendah atau disebut dengan *Low-Fidelity*. *Wireframe Low-Fidelity* merupakan tahapan perancangan desain yang sederhana dengan berfokus kepada tata letak, hierarki informasi, dan struktur elemen detail *interface* sistem seperti garis-garis kasar, bentuk-bentuk sederhana, dan teks sederhana. *Wireframe* ini dirancang agar dapat memberikan gambaran terkait elemen-elemen akan diatur sehingga mempermudah dan meng-efisienkan pengerjaan detail desain *interface* sistem, juga menguji konsep dan memastikan bahwa struktur dan hirarki informasi sudah benar sebelum memasuki tahap desain yang lebih rinci.

- Halaman *Landing Page*



Gambar 59. *Wireframe Low-Fid Landing Page*

Gambar 59 diatas merupakan *Wireframe low fidelity* dari *landing page* yang akan muncul saat pengguna pertama kali mengakses laman website. Halaman ini menampilkan informasi terkait perumahan klaster Villa Gading Mayang Kota Jambi seperti jumlah warga dan perumahan, pengenalan singkat perumahan, dan fitur pengelolaan yang tersedia pada *website* SIPGAMA. Di pojok kanan atas terdapat tombol Login untuk dapat masuk dan mengakses sistem SIPGAMA.

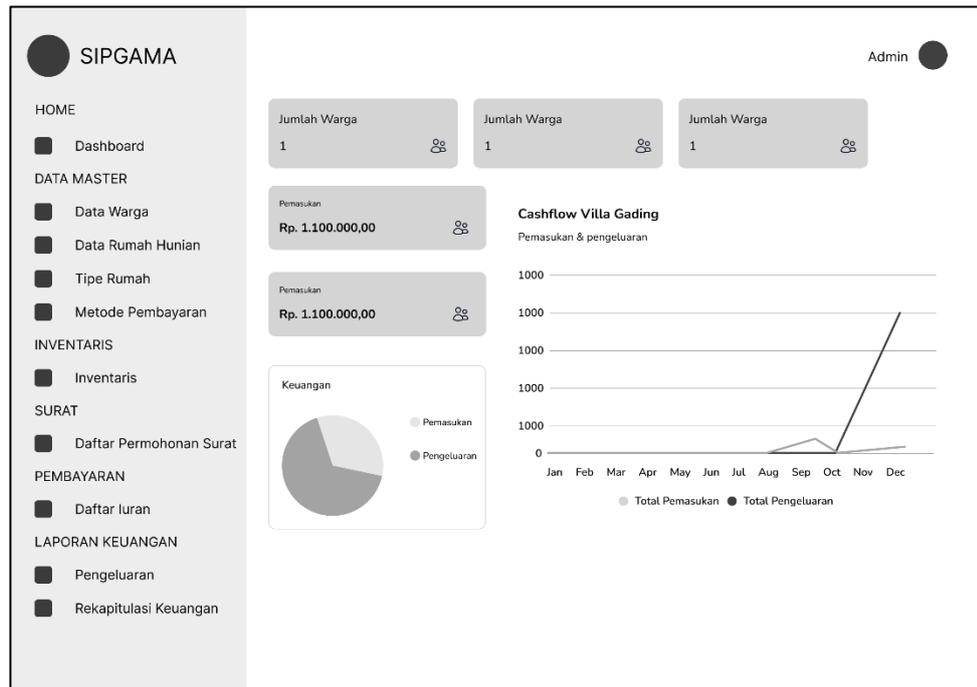
- Halaman *Login*

A wireframe illustration of a login page. At the top center, there is a dark circle icon followed by the text 'SIPGAMA'. Below this, there are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. At the bottom of the form, there is a dark rectangular button with the text 'Sign In' in white.

Gambar 60. *Wireframe Low-Fid* Halaman Login

Gambar 60 diatas merupakan *Wireframe low fidelity* dari halaman Login yang akan tampil setelah kita menekan tombol Login di *landing page*. Halaman login akan menampilkan form *input username* dan *password* yang menjadi autentikator ketika pengguna ingin masuk kedalam sistem dan terdapat tombol Login di bawahnya.

- Halaman Dashboard



Gambar 61. Wireframe Low-Fid Dashboard

Gambar 61 diatas merupakan *Wireframe low fidelity* dari halaman dashboard yang akan muncul setelah pengguna berhasil masuk kedalam sistem. Dashboard ini menampilkan informasi terkait jumlah warga, rekapitulasi keuangan perumahan, beserta grafik yang menunjukkan keuangan perumahan setiap bulannya.

- Halaman Data Master

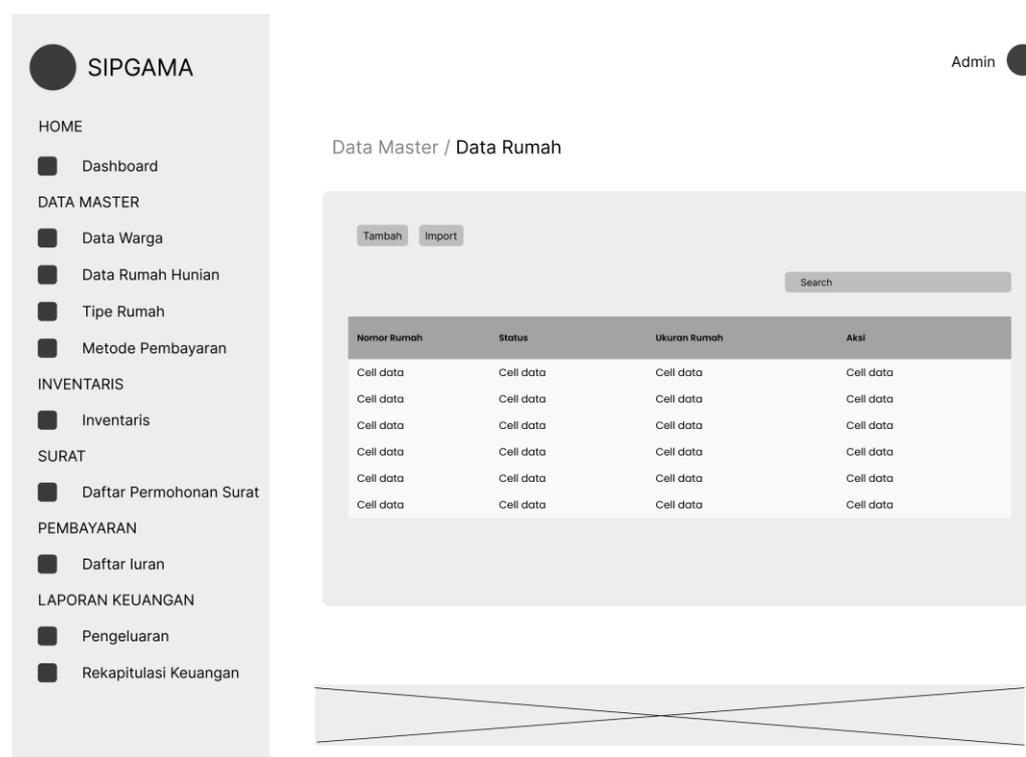
The Data Master wireframe for SIPGAMA includes a sidebar with the following menu items: HOME (Dashboard), DATA MASTER (Data Warga, Data Rumah Hunian, Data Tipe Rumah, Metode Pembayaran), INVENTARIS (Inventaris), SURAT (Daftar Permohonan Surat), PEMBAYARAN (Daftar Iuran), and LAPORAN KEUANGAN (Pengeluaran, Rekapitulasi Keuangan). The main content area features a 'Data Master / Data Warga' section with 'Tambah' and 'Import' buttons, a search bar, and a table with the following columns: NIK, Nomor KK, Nama, Role, No. Telepon, Tempat Lahir, Tanggal Lahir, Jenis Kelamin, Agama, Pendidikan Terakhir, Pekerjaan, No. Rumah, and Akas.

NIK	Nomor KK	Nama	Role	No. Telepon	Tempat Lahir	Tanggal Lahir	Jenis Kelamin	Agama	Pendidikan Terakhir	Pekerjaan	No. Rumah	Akas
Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data				
Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data				
Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data				
Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data				
Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data				
Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data	Cell data				

Gambar 62. Wireframe Low-Fid Data Master (Data Warga)

Gambar 62 diatas merupakan *Wireframe low fidelity* dari menu data master yaitu data warga yang menampilkan keseluruhan data warga perumahan Villa Gading Mayang Kota Jambi meliputi data NIK, KK, nama, *role* pengguna dalam sistem, tempat dan tanggal lahir, dan sebagainya. Pada pojok kanan atas tabel data warga terdapat kolom pencarian yang bisa digunakan untuk mencari data sesuai keyword yang diinginkan.

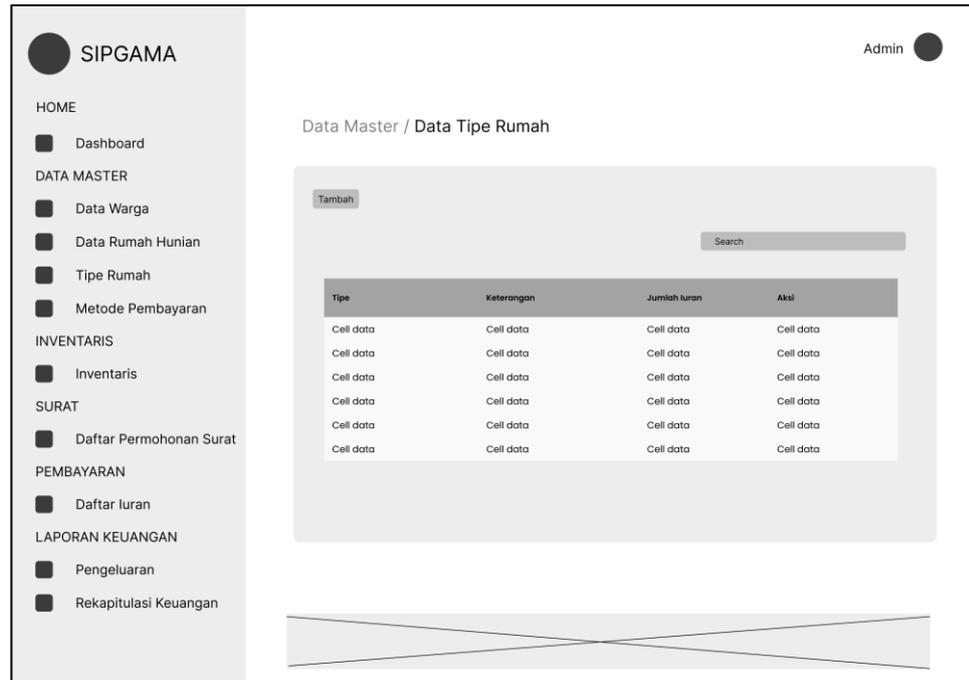
Pada halaman data warga khusus *role* admin akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah dan import data, juga tombol edit dan hapus di kolom pojok kanan tabel data.



Gambar 63. *Wireframe Low-Fid* Data Master (Data Rumah)

Gambar 63 diatas merupakan *Wireframe low fidelity* dari menu data master yaitu data rumah yang menampilkan keseluruhan data rumah pada perumahan Villa Gading Mayang Kota Jambi meliputi data nomor rumah, status rumah, serta tipe rumah tersebut. Pada pojok kanan atas tabel data terdapat kolom pencarian yang bisa digunakan untuk mencari data sesuai keyword yang diinginkan.

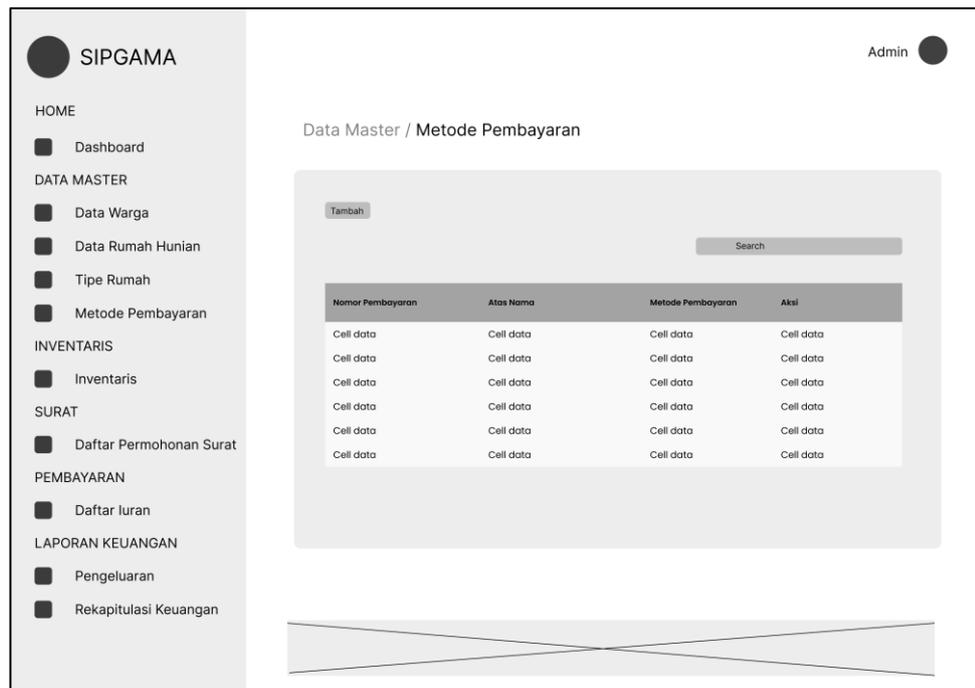
Pada halaman data rumah khusus *role* admin akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah dan import data, juga tombol edit dan hapus di kolom pojok kanan tabel data.



Gambar 64. Wireframe Low-Fid Data Master (Data Tipe Rumah)

Gambar 64 diatas merupakan *Wireframe low fidelity* dari menu data master yaitu data tipe rumah yang menampilkan keseluruhan data terkait tipe rumah pada perumahan Villa Gading Mayang Kota Jambi meliputi data tipe, keterangan, serta jumlah iuran sesuai tipe rumah tersebut. Pada pojok kanan atas tabel terdapat kolom pencarian yang bisa digunakan untuk mencari data sesuai keyword yang diinginkan.

Pada halaman data tipe rumah khusus *role* admin akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah, edit, dan hapus di kolom pojok kanan tabel data

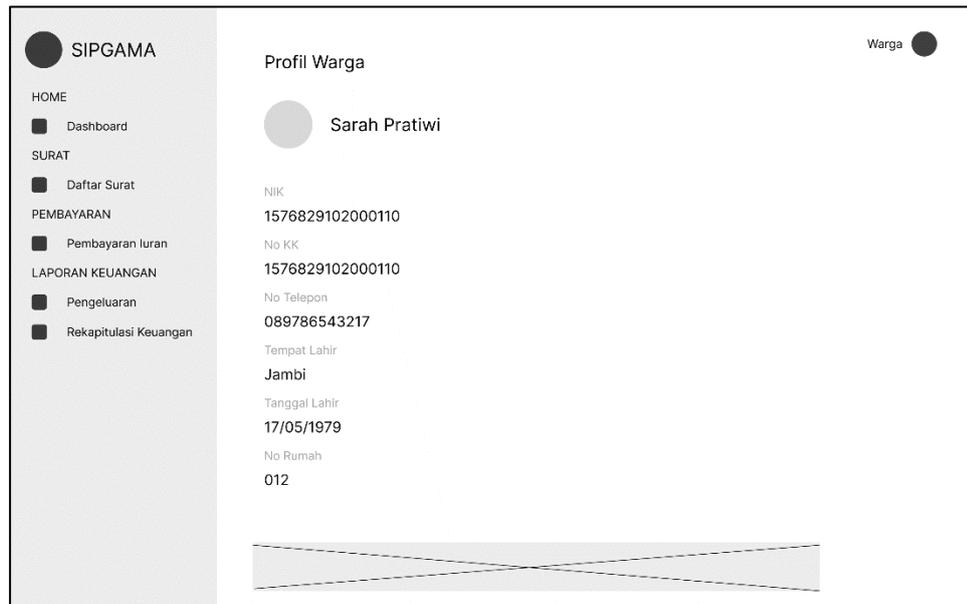


Gambar 65. Wireframe Low-Fid Data Master (Metode Pembayaran)

Gambar 65 diatas merupakan *Wireframe low fidelity* dari menu data master yaitu metode pembayaran yang menampilkan keseluruhan metode pembayaran yang dimiliki oleh perumahan Villa Gading Mayang Kota Jambi untuk tujuan pembayaran iuran warga meliputi data nomor pembayaran, atas nama dari nomor pembayaran tersebut, serta jenis metode pembayarannya. Pada pojok kanan atas tabel data terdapat kolom pencarian yang bisa digunakan untuk mencari data sesuai keyword yang diinginkan.

Pada halaman metode pembayaran khusus *role* admin akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah, edit, dan hapus di kolom pojok kanan tabel data

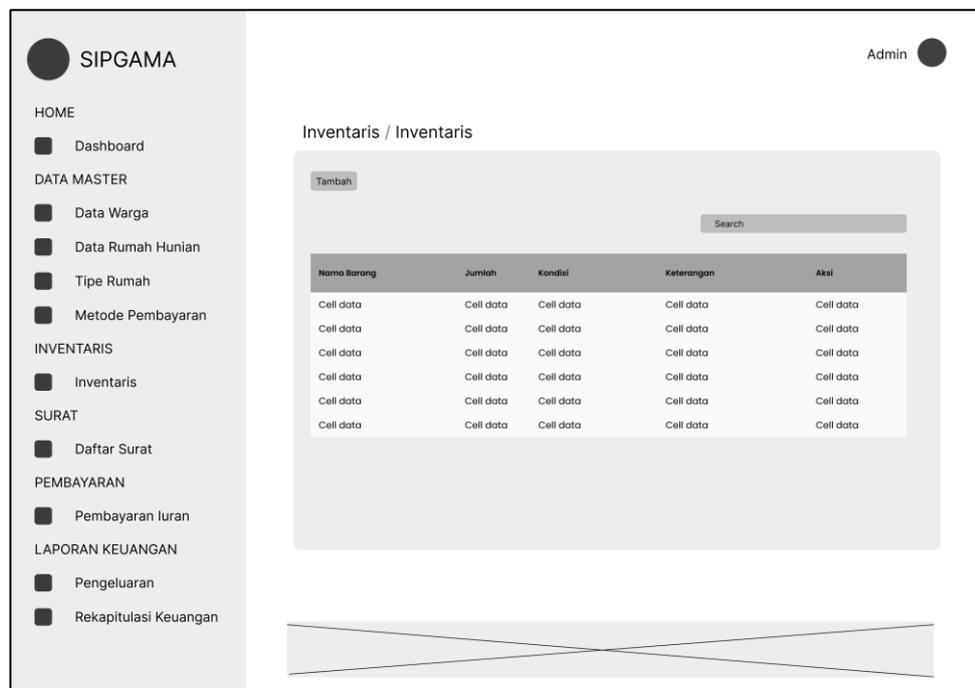
- Halaman Profil Warga



Gambar 66. Wireframe Low-Fid Profil Warga

Gambar 66 di atas merupakan *Wireframe low fidelity* dari menu profil warga yang menampilkan data pribadi terkait warga perumahan Villa Gading Mayang Kota Jambi tersebut meliputi nama, NIK, nomor KK, dan lain sebagainya.

- Halaman Inventaris

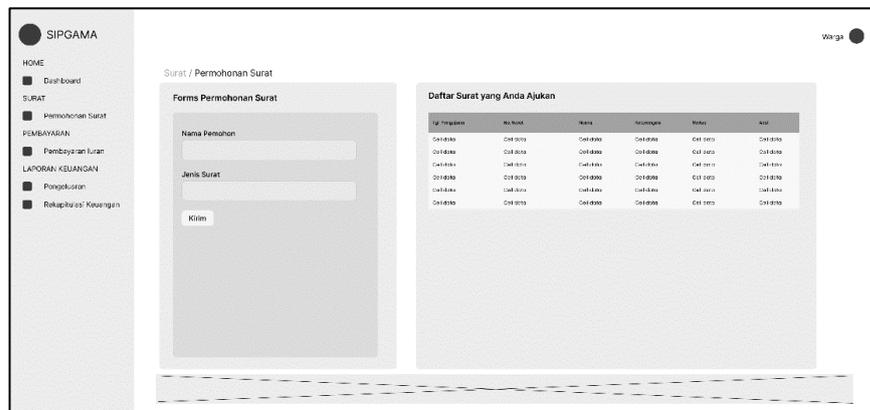


Gambar 67. Wireframe Low-Fid Halaman Data Inventaris

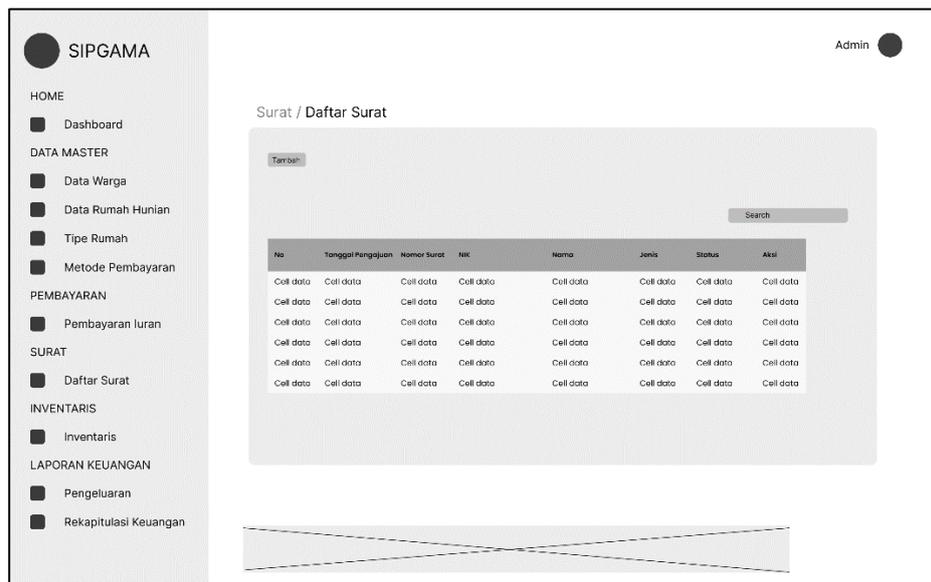
Gambar 67 diatas merupakan *Wireframe low fidelity* dari menu inventaris yang menampilkan keseluruhan data terkait barang inventaris yang dimiliki oleh perumahan klaster Villa Gading Mayang Kota Jambi diantaranya data nama barang, jumlah, kondisi barang, serta keterangan barang inventaris. Pada pojok kanan atas tabel terdapat kolom pencarian yang bisa digunakan untuk mencari data sesuai keyword yang diinginkan.

Pada halaman inventaris khusus *role* admin akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah, edit, dan hapus di kolom pojok kanan tabel data

- Halaman Surat



Gambar 68. *Wireframe Low-Fid* Permohonan Surat (Warga)



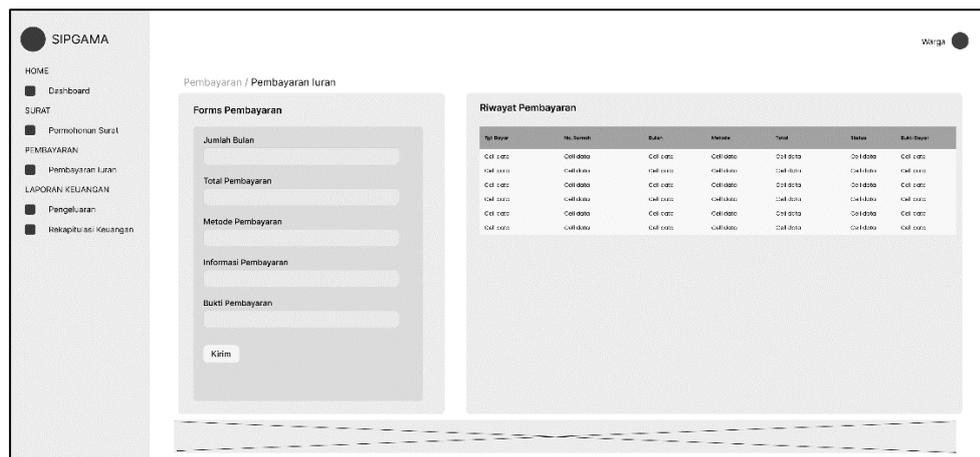
Gambar 69. *Wireframe Low-Fid* Daftar Surat

Gambar 68 dan 69 diatas merupakan *Wireframe low fidelity* dari menu surat yaitu permohonan surat untuk warga serta daftar surat untuk admin,

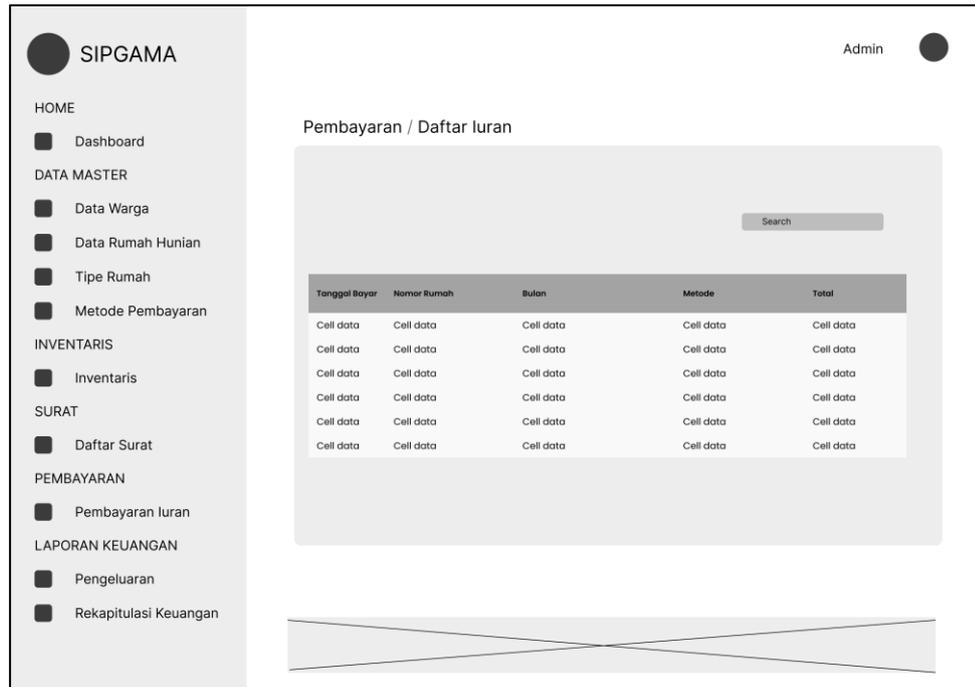
pimpinan, dan bendahara. Halaman permohonan surat untuk warga akan menampilkan form permohonan surat yang diisi untuk pengajuan permohonan surat pengantar pada administrasi perumahan klaster Villa Gading Mayang, pada sisi kanan terdapat tabel yang membuat daftar permohonan yang ada diajukan oleh warga berisikan data pengajuan dan status dari permohonan.

Sedangkan halaman daftar surat untuk pimpinan dan bendahara berisikan keseluruhan data pengajuan permohonan surat perumahan meliputi tanggal pengajuan, nomor surat, NIK, nama, serta jenis surat. Pada halaman daftar surat khusus *role* admin akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah, juga tombol validasi, download, lihat dokumen, dan edit di kolom pojok kanan tabel data.

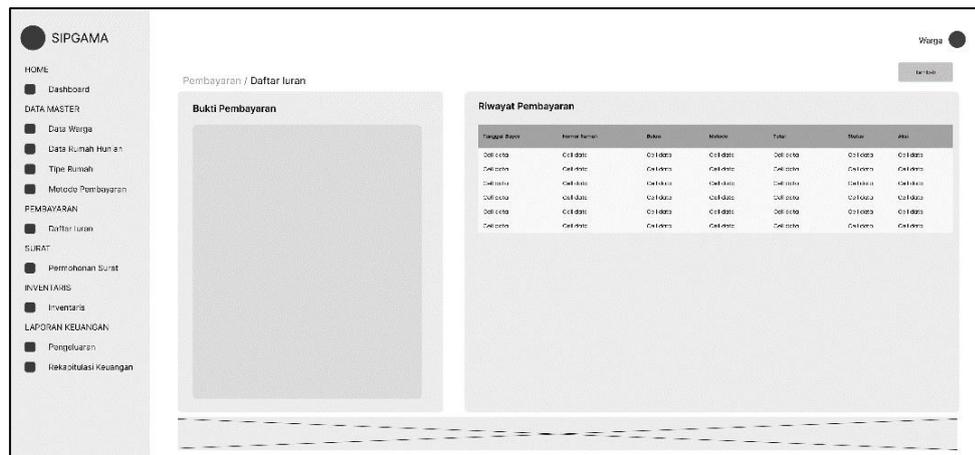
- Halaman Pembayaran



Gambar 70. Wireframe Low-Fid Pembayaran Iuran (Warga)



Gambar 71. *Wireframe Low-Fid* Daftar Iuran (Admin, Pimpinan)



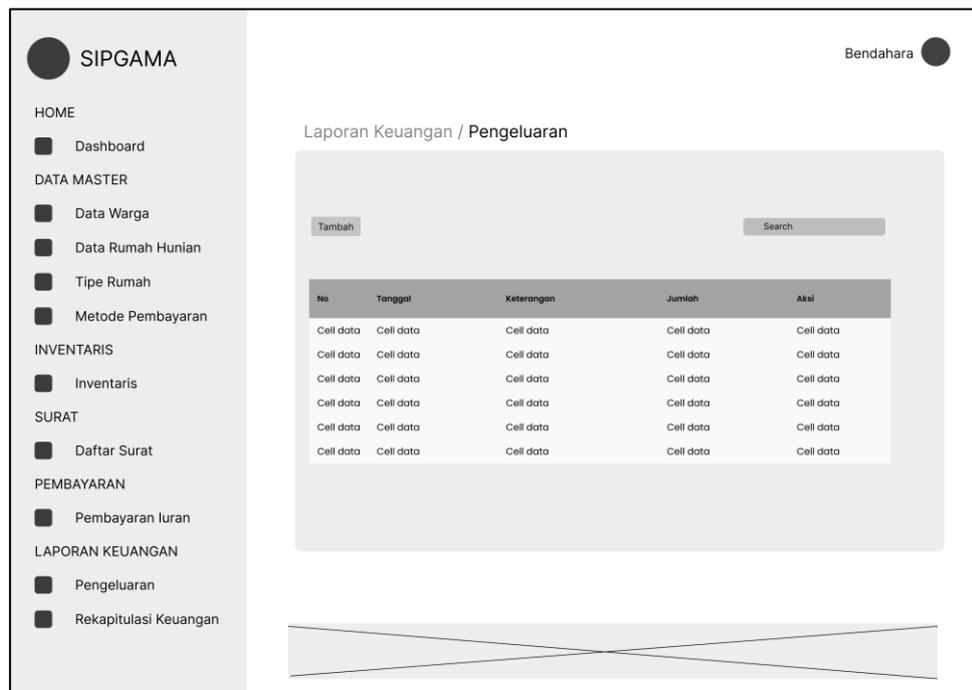
Gambar 72. *Wireframe Low-Fid* Daftar Iuran (Bendahara)

Gambar 70, 71, dan 72 di atas merupakan *Wireframe low fidelity* dari menu pembayaran yaitu pembayaran iuran untuk warga serta daftar iuran pertama untuk admin dan pimpinan, dan halaman daftar iuran khusus pengelolaan bendahara. Halaman pembayaran iuran untuk warga akan menampilkan form pembayaran iuran perumahan klaster Villa Gading Mayang Kota Jambi yang, pada sisi kanan terdapat tabel yang memuat riwayat pembayaran iuran yang telah dilakukan oleh warga tersebut.

Sedangkan halaman daftar surat untuk admin dan pimpinan berisikan tabel data berisikan keseluruhan pembayaran iuran yang telah dilakukan

warga perumahan. Pada halaman daftar iuran khusus *role* bendahara akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah, juga tombol validasi, lihat bukti pembayaran, dan edit di kolom pojok kanan tabel data.

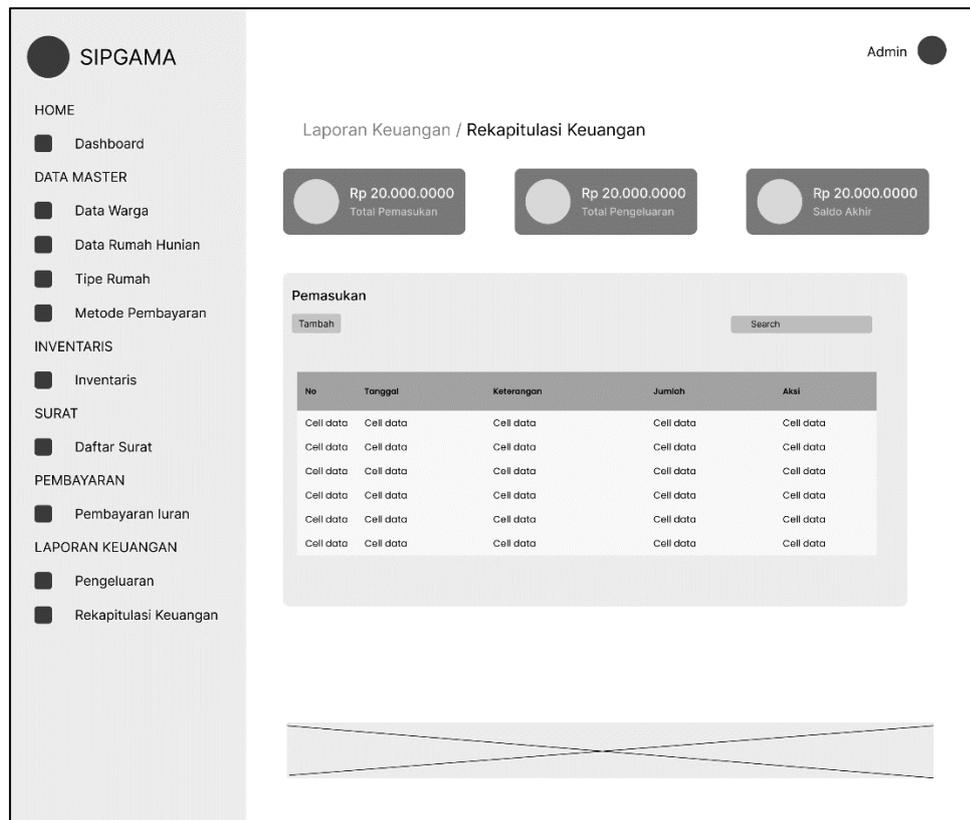
- Halaman Laporan Keuangan



Gambar 73. Wireframe Low-Fid Pengeluaran (Laporan Keuangan)

Gambar 73 diatas merupakan *Wireframe low fidelity* dari menu laporan keuangan yaitu pengeluaran yang menampilkan keseluruhan data pengeluaran perumahan Villa Gading Mayang Kota Jambi meliputi tanggal pengeluaran, keterangan, serta jumlah pengeluaran. Pada pojok kanan atas tabel data pengeluaran terdapat kolom pencarian yang bisa digunakan untuk mencari data sesuai keyword yang diinginkan.

Pada halaman pengeluaran khusus *role* bendahara akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah data, edit, dan hapus di kolom pojok kanan tabel data.



Gambar 74. Wireframe Low-Fid Rekapitulasi Keuangan
(Laporan Keuangan)

Gambar 74 diatas merupakan *Wireframe low fidelity* dari menu laporan keuangan yaitu rekapitulasi keuangan yang pada bagian atasnya menampilkan rekap pemasukan, pengeluaran, serta total keseluruhan kas keuangan yang dimiliki perumahan klaster Villa Gading Mayang Kota Jambi. Selanjutnya, terdapat juga tabel data pemasukan keuangan yang berasal dari iuran dan dana lainnya meliputi tanggal pengeluaran, keterangan, serta jumlah pemasukan. Pada pojok kanan atas tabel data pengeluaran terdapat kolom pencarian yang bisa digunakan untuk mencari data sesuai keyword yang diinginkan.

Pada halaman pengeluaran khusus *role* bendahara akan terdapat tombol kelola data diantaranya tombol yang terletak diatas tabel data yaitu tombol tambah data, edit, dan hapus di kolom pojok kanan tabel data.

2. Wireframe high-fidelity

Rampungnya *Wireframe Low-Fidelity* selanjutnya dirancang desain lanjutan *Wireframe* dengan presisi yang tinggi atau *high-fidelity*. *Wireframe high-fidelity* merupakan desain rancangan *Wireframe* interface yang lebih detail mendekati produk akhir *interface* sistem mencakup elemen-elemen visual seperti warna, tipografi, dan gambar dengan lebih mendetail. *Wireframe* ini dirancang agar dapat terlihat bagaimana setiap elemen desain saling berinteraksi serta memberikan penggambaran lebih akurat terkait pengalaman pengguna sistem.

- Halaman *Landing Page*



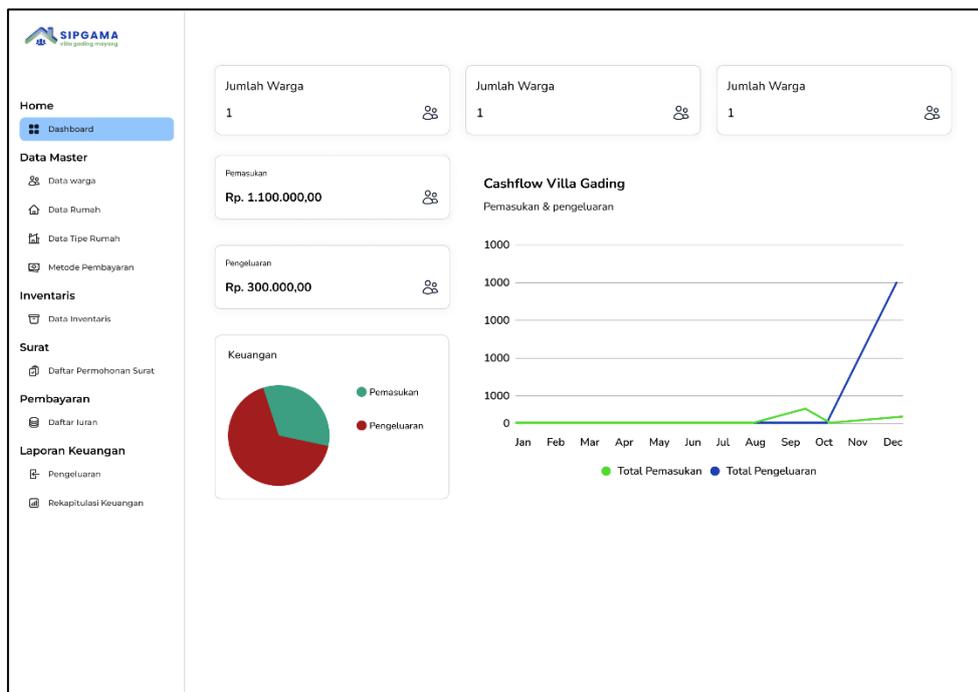
Gambar 75. Wireframe Hi-Fid Landing Page

- Halaman *Login*

The login page displays the SIPGAMA logo with the tagline 'villa gading mayang'. Below the logo, there are two text input fields, both labeled 'Nama', and a blue button labeled 'Masuk'.

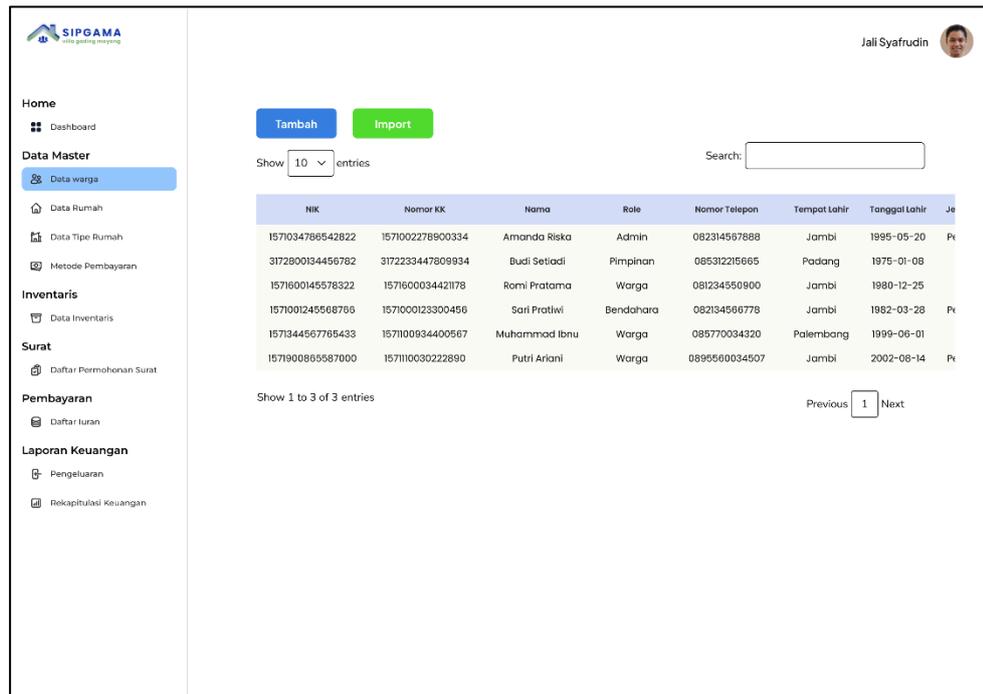
Gambar 76. Wireframe Hi-Fid Login

- Halaman *Dashboard*



Gambar 77. Wireframe Hi-Fid Dashboard

- Halaman Data Master

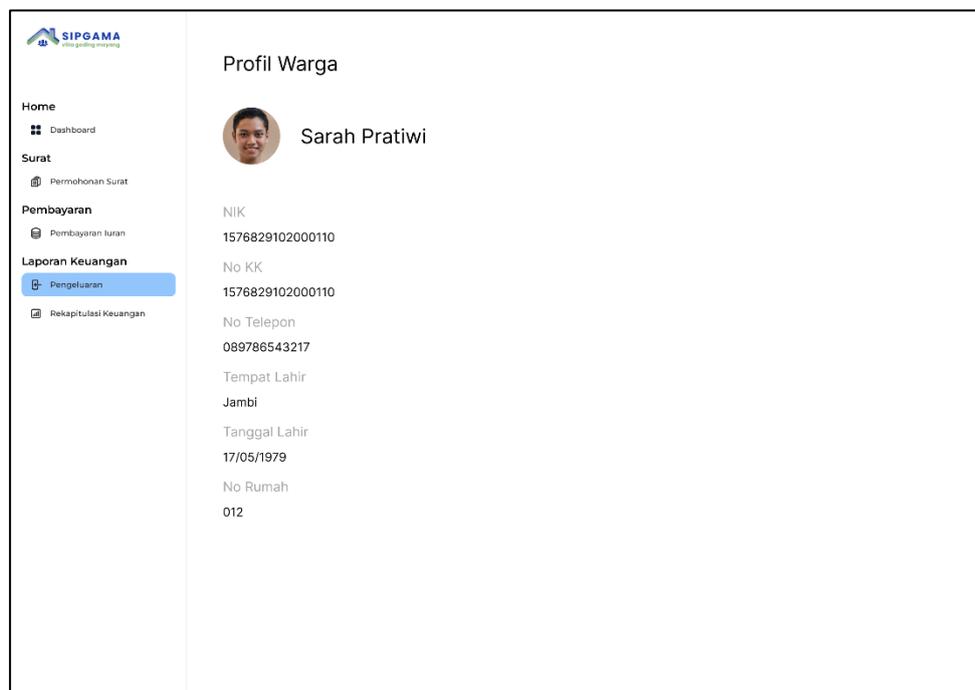


The wireframe shows a dashboard for 'SIPGAMA' with a sidebar menu on the left. The main content area displays a 'Data warga' section with 'Tambah' and 'Import' buttons, a search bar, and a table of 3 entries. The table has columns for NIK, Nomor KK, Nama, Role, Nomor Telepon, Tempat Lahir, and Tanggal Lahir. Below the table are pagination controls showing '1' of 3 entries.

NIK	Nomor KK	Nama	Role	Nomor Telepon	Tempat Lahir	Tanggal Lahir	Je
1571034786542822	1571002278900334	Amanda Riska	Admin	082314567888	Jambi	1995-05-20	Pe
3172800134456762	3172233447809934	Budi Setiadi	Pimpinan	085312216665	Padang	1975-01-08	Pe
1571600145578322	157160003442178	Romi Pratama	Warga	081234550900	Jambi	1980-12-25	Pe
1571001245568766	1571000123300456	Sari Pratiwi	Bendahara	082134566778	Jambi	1982-03-28	Pe
1571344567765433	1571000934400567	Muhammad Ibnu	Warga	085770034320	Palembang	1999-06-01	Pe
1571900865587000	157110030222890	Putri Ariani	Warga	0895560034507	Jambi	2002-08-14	Pe

Gambar 78. Wireframe Hi-Fid Data Warga (Data Master)

- Halaman Profil Warga

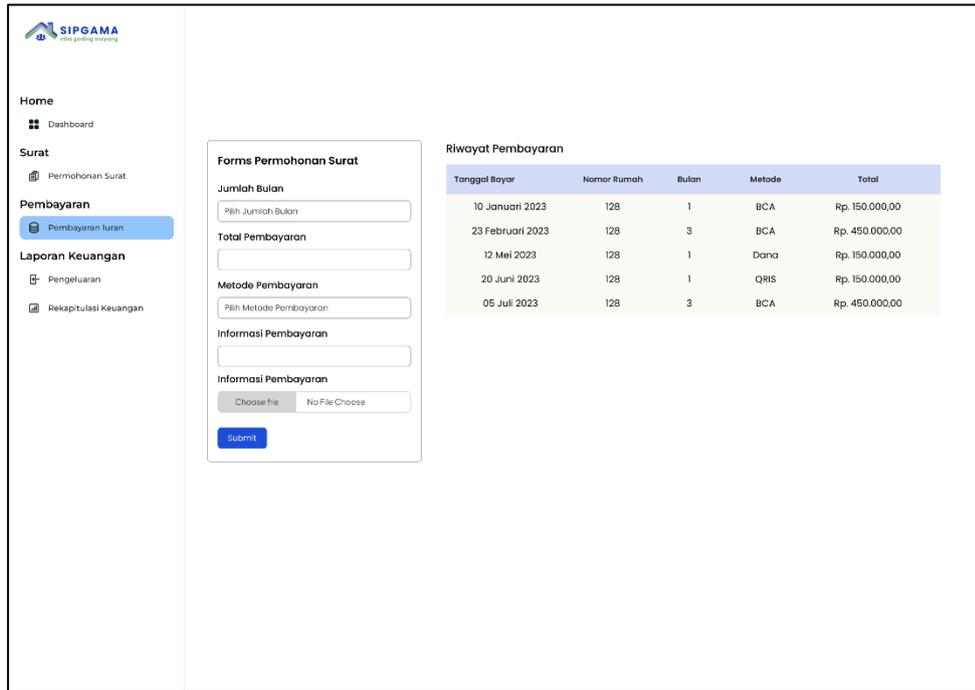


The wireframe shows a profile page for 'Sarah Pratiwi'. The sidebar menu is on the left. The profile card includes a profile picture, name, and several fields: NIK, No KK, No Telepon, Tempat Lahir, Tanggal Lahir, and No Rumah.

NIK	1576829102000110
No KK	1576829102000110
No Telepon	089786543217
Tempat Lahir	Jambi
Tanggal Lahir	17/05/1979
No Rumah	012

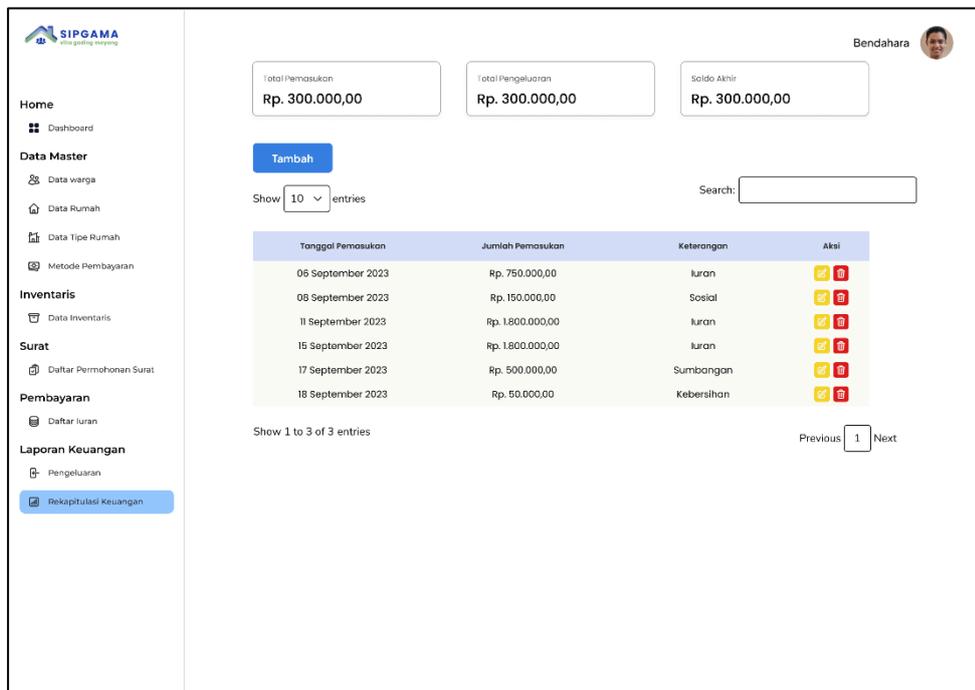
Gambar 79. Wireframe Hi-Fid Profil Warga

• Halaman Pembayaran



Gambar 80. Wireframe Hi-Fid Pembayaran Iuran (Warga)

• Halaman Laporan Keuangan



Gambar 81. Wireframe Hi-Fid Rekapitulasi Pengeluaran (Laporan Keuangan)

b. *Prototype*

Perancangan desain tampilan antarmuka sistem (UI) mencapai tahapan terakhir dengan rancangan *prototype* yang memberikan gambaran alur proses kerja sistem guna diketahui apakah fungsi dan fitur yang dihadirkan telah memenuhi kebutuhan pengguna. Pada penelitian ini, desain yang dirancang hanya sebagai pelengkap rancangan sistem sehingga mempermudah pengembangan tahap implementasi sistem dengan memberikan gambaran jelas bagaimana sistem SIPGAMA akan beroperasi untuk memenuhi kebutuhan warga perumahan klaster Villa Gading Mayang Kota Jambi. Berikut merupakan hasil rancangan prototipe dari sistem pengelolaan perumahan klaster Villa Gading Mayang Kota Jambi (SIPGAMA) :

Tabel 37. *Link Prototype* SIPGAMA

<i>User</i>	Link Prototipe
Admin	https://figmashort.link/Bhs6QH
Pimpinan	https://figmashort.link/tjrQDG
Bendahara	https://figmashort.link/G2AGtF
Warga	https://figmashort.link/jF4yz2

4.4 Tahap Evaluasi

Setelah dilakukan analisis dan perancangan terhadap kebutuhan serta alur proses bisnis sistem SIPGAMA, maka tahapan terakhir yaitu memastikan bahwa setiap *output* perancangan sistem telah memenuhi kebutuhan pengguna maupun *stakeholder* dengan dilakukannya pengujian atau evaluasi. Evaluasi pada penelitian ini menggunakan teknik *Requirement Traceability Matrix* (RTM) yaitu *functional tracing* dimana evaluasi pelacakan kebutuhan melalui deskripsi fungsional yang telah terdefinisi sesuai objek yang akan terlibat dalam pengembangan sistem (Pinheiro, 2004). Evaluasi RTM *functional tracing* dilakukan dengan pemodelan hubungan antar diagram, objek, kelas, dan lainnya yang terdefinisi dengan baik sebelumnya. Pada penelitian ini setiap fungsional yang dibutuhkan telah didefinisikan sebelumnya pada tahap analisis yaitu *system requirement*.

Requirement traceability matrix dibuat dengan menyusun tabel *tracing* dimana pada bagian baris pertama kolom pertama yang berisikan *system requirements* yang telah terdefinisi sebelumnya dan kemudian dihubungkan ke elemen lainnya yaitu analisis dan desain yang telah dirancang dimana setiap elemen ini diberikan *unique identifier* (ID) yang digunakan sebagai referensi

pengenal *matrix*. Oleh karena itu, maka dibuat terlebih dahulu kode ID untuk setiap rancangan yang telah dibuat yang dapat dilihat pada tabel berikut :

Tabel 38. Pemetaan Kode *Unique Identifier* (ID) *Activity Diagram*

No.	Kode (ID)	<i>Activity Diagram</i>
1.	AD-1	<i>Activity Diagram</i> Login
2.	AD-2	<i>Activity Diagram</i> Logout
3.	AD-3	<i>Activity Diagram</i> Tambah Data Master
4.	AD-4	<i>Activity Diagram</i> Edit Data Master
5.	AD-5	<i>Activity Diagram</i> Hapus Data Master
6.	AD-6	<i>Activity Diagram</i> Melihat Data Master (Pimpinan
7.	AD-7	<i>Activity Diagram</i> Melihat Data Master (Bendahara)
8.	AD-8	<i>Activity Diagram</i> Tambah Data Inventaris
9.	AD-9	<i>Activity Diagram</i> Edit Data Inventaris
10.	AD-10	<i>Activity Diagram</i> Hapus Data Inventaris
11.	AD-11	<i>Activity Diagram</i> Melihat Data Inventaris
12.	AD-12	<i>Activity Diagram</i> Input Permohonan Surat
13.	AD-13	<i>Activity Diagram</i> Tambah Data Daftar Permohonan Surat
14.	AD-14	<i>Activity Diagram</i> Edit Data Daftar Permohonan Surat
15.	AD-15	<i>Activity Diagram</i> Melihat Daftar Permohonan Surat
16.	AD-16	<i>Activity Diagram</i> Input Pembayaran Iuran
17.	AD-17	<i>Activity Diagram</i> Tambah Data Daftar Iuran
18.	AD-18	<i>Activity Diagram</i> Edit Data Daftar Iuran
19.	AD-19	<i>Activity Diagram</i> Melihat Daftar Iuran
20.	AD-20	<i>Activity Diagram</i> Melihat Riwayat Pembayaran Iuran
20.	AD-21	<i>Activity Diagram</i> Validasi Permohonan Surat
21.	AD-22	<i>Activity Diagram</i> Validasi Pembayaran Iuran
22.	AD-23	<i>Activity Diagram</i> Melihat Laporan Keuangan
23.	AD-24	<i>Activity Diagram</i> Tambah Data Laporan Keuangan
24.	AD-25	<i>Activity Diagram</i> Edit Data Laporan Keuangan
25.	AD-26	<i>Activity Diagram</i> Hapus Data Laporan Keuangan

Tabel 39. Pemetaan Kode *Unique Identifier* (ID) *Sequence Diagram*

No.	Kode (ID)	<i>Sequence Diagram</i>
1.	SD-1	<i>Sequence Diagram</i> Login
2.	SD-2	<i>Sequence Diagram</i> Data Master (Data Warga)
3.	SD-3	<i>Sequence Diagram</i> Data Master (Data Rumah)
4.	SD-4	<i>Sequence Diagram</i> Data Master (Data Tipe Rumah)
5.	SD-5	<i>Sequence Diagram</i> Data Master (Metode Pembayaran)
6.	SD-6	<i>Sequence Diagram</i> Inventaris
7.	SD-7	<i>Sequence Diagram</i> Input Permohonan Surat
8.	SD-8	<i>Sequence Diagram</i> Daftar Permohonan Surat
9.	SD-9	<i>Sequence Diagram</i> Input Pembayaran Iuran
10.	SD-10	<i>Sequence Diagram</i> Daftar Iuran
11.	SD-11	<i>Sequence Diagram</i> Validasi Permohonan Surat
12.	SD-12	<i>Sequence Diagram</i> Validasi Pembayaran Iuran
13.	SD-13	<i>Sequence Diagram</i> Laporan Keuangan (Pengeluaran)
14.	SD-14	<i>Sequence Diagram</i> Laporan Keuangan (Rekapitulasi Keuangan)

Tabel 40. Pemetaan Kode *Unique Identifier* (ID) *User Interface*

No.	Kode (ID)	<i>User Interface</i>
1.	UI-1	Halaman Login
2.	UI-2	Halaman Data Master (Data Warga)
3.	UI-3	Halaman Data Master (Data Rumah)
4.	UI-4	Halaman Data Master (Data Tipe Rumah)
5.	UI-5	Halaman Data Master (Metode Pembayaran)
6.	UI-6	Halaman Profil Warga
7.	UI-7	Halaman Data Inventaris
8.	UI-8	Halaman Permohonan Surat
9.	UI-9	Halaman Daftar Permohonan Surat
10.	UI-10	Halaman Pembayaran Iuran
11.	UI-11	Halaman Daftar Iuran
12.	UI-12	Halaman Daftar Surat (Admin)
13.	UI-13	Halaman Daftar Iuran (Bendahara)
14.	UI-14	Halaman Laporan Keuangan (Pengeluaran)
15.	UI-15	Halaman Laporan Keuangan (Rekapitulasi Keuangan)

Setelah terdefinisi keseluruhan Kode ID yang dibutuhkan, maka selanjutnya dapat dirancang tabel RTM *functional tracing* sistem SIPGAMA sebagai berikut :

No.	Req ID	<i>Specific Aspect of the system or its environment</i>					
		Definisi Kebutuhan (SR)	<i>Use Case</i>	<i>Activity Diagram</i>	<i>Sequence Diagram</i>	<i>User Interface</i>	<i>Verified</i>
1.	R-1	Sistem dapat memberikan akses kepada pengguna untuk <i>login</i> kedalam sistem dan keluar dari sistem	UC-1, UC-2	AD-1, AD-2	SD-1	UI-1	√
2.	R-2	Sistem dapat menampilkan informasi dan menyediakan fitur pengelolaan terkait warga perumahan, rumah yang telah ditempati beserta tipe perumahan.	UC-3, UC-4	AD-3, AD-4, AD-5, AD-6, AD-7	SD-2, SD-3, SD-4, SD-5	UI-2, UI-3, UI-4, UI-5, UI-6	√
3.	R-3	Sistem dapat menampilkan dan menyediakan fitur data inventaris barang perumahan	UC-5, UC-6	AD-8, AD-9, AD-10, AD-11	SD-6	UI-7	√
4.	R-4	Sistem dapat menyediakan fitur permohonan surat pengantar RT	UC-7	AD-12	SD-7	UI-8	√
5.	R-5	Sistem dapat memberikan informasi terkait permohonan surat pengantar RT ruang lingkup internal perumahan beserta pengelolaan datanya	UC-8, UC-9	AD-13, AD-14, AD-15	SD-8	UI-9	√
6.	R-6	Sistem dapat menyediakan fitur pembayaran iuran perumahan	UC-10,	AD-16	SD-9	UI-10	√
7.	R-7	Sistem mampu memberikan informasi terkait riwayat pembayaran iuran yang telah dilakukan setiap warga beserta pengelolaan datanya	UC-11, UC-12, UC-13	AD-17, AD-18, AD-19, AD-20	SD-10	UI-11	√
8.	R-8	Sistem memungkinkan sekretaris untuk memverifikasi pemenuhan permohonan surat pengantar	UC-14	AD-21	SD-11	UI-12	√

9.	R-9	Sistem memungkinkan bendahara untuk memverifikasi pembayaran iuran warga	UC-15	AD-22	SD-12	UI-13	√
10.	R-10	Sistem dapat menampilkan informasi dan menyediakan fitur pengelolaan terkait transparansi laporan keuangan yang berisikan pengeluaran, pemasukan, beserta rekapitulasi keuangan perumahan	UC-16, UC-17	AD-23, AD-24, AD-25, AD-26	SD-13, SD-14	UI-14, UI-15	√

Evaluasi hasil analisis dan perancangan diatas dilakukan oleh manajer pengembangan proyek yaitu bapak Dedy Setiawan, S.Kom., M.I.T serta tim pengembangan implementasi dengan pengujian berkala dimulai dari penetapan kebutuhan terutama kebutuhan fungsional sistem berdasarkan hasil observasi dan wawancara, berkelanjutan hingga terdefinisi yang dituangkan dalam tabel pada baris pertama kolom pertama yaitu *Requirements*. Selanjutnya dilakukan *tracing* atau pelacakan terhadap hasil analisis dan perancangan dengan setiap kebutuhan yang telah didefinisikan.

Berdasarkan hasil evaluasi didapatkan bahwa setiap hasil perancangan dimulai dari *Use Case*, *Activity Diagram*, *Sequence Diagram*, serta *user interface* (UI) telah sesuai dengan seluruh kebutuhan warga perumahan Villa Gading Mayang dan mendefinisikan secara rinci setiap alur objek dan fitur yang harus dihadirkan dalam sistem. Setiap rancangan UML dan interface juga berdasarkan hasil pengujian tim bisa diteruskan untuk pengembangan implementasi secara menyeluruh menjadi sistem SIPGAMA (Sistem Informasi Pengelolaan Perumahan Klaster Villa Gading Mayang).

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan analisis dan perancangan sistem informasi pengelolaan menggunakan metode OOAD yang telah dilaksanakan, diperoleh rangkaian hasil penelitian yang dapat menunjang pengembangan implementasi sistem SIPGAMA, dimana pada tahap analisis penelitian dengan tahapan metode OOA menghasilkan informasi terkait kebutuhan-kebutuhan sistem terperinci yang tertuang dalam *system requirement* dengan 10 kebutuhan yang teridentifikasi, dan *Use Case Diagram* dengan 1 diagram umum dan 17 sub *Use Case* beserta skenarionya yang menjadi acuan untuk lanjutan tahap perancangan

Pada Tahap Perancangan berdasarkan metode OOD dihasilkan 25 rancangan *Activity Diagram*, *Class diagram*, 14 rancangan *Sequence Diagram*, dan hasil rancangan tampilan antarmuka (UI) yang terdiri dari *Wireframe* serta *prototype* desain sistem SIPGAMA yang menggambarkan secara menyeluruh bagaimana sistem akan bekerja dalam memenuhi kebutuhan warga dan pengelola perumahan.

Setelah didapatkan hasil analisis dan rancangan yang berkesesuaian dengan kebutuhan perumahan klaster Villa Gading Mayang Kota Jambi, tahapan terakhir yaitu dilakukan pengujian atau evaluasi terhadap hasil yang diperoleh, dimana pada tahapan ini digunakan metode *requirement traceability matrix* (RTM) *functional tracing* dan didapatkan bahwa keseluruhan hasil analisis dan perancangan sistem SIPGAMA setelah dilacak oleh tim manajer proyek dan implementator telah memenuhi seluruh kebutuhan perumahan yang teridentifikasi serta dapat dilanjutkan untuk diimplementasikan.

5.2 Saran

Analisis dan perancangan sistem informasi pengelolaan perumahan klaster Villa Gading Mayang Kota Jambi ini masih jauh dari kata sempurna dikarenakan adanya keterbatasan ataupun kekurangan yang dapat disebabkan oleh berbagai faktor. Terdapat beberapa saran dalam pengembangan sistem yakni diantaranya:

1. Penelitian yang dilaksanakan menggunakan metode OOAD dalam melakukan analisis dan perancangan sistem sehingga disarankan pada penelitian selanjutnya agar dapat menggunakan metode perancangan

selain OOAD dalam melakukan analisis dan perancangan untuk pengembangan sistem pengelolaan perumahan klaster.

2. Analisis dan perancangan yang telah dilakukan masih terbatas pada perancangan terkhusus alur dan proses kerja sistem saja tidak secara menyeluruh melakukan penelitian pada rancangan UI/UX sistem sehingga disarankan pada penelitian selanjutnya diharapkan dapat melakukan analisis dan perancangan UI/UX dari sistem pengelolaan klaster perumahan.
3. Pengeluaran yang didata perlu memuat bukti terlampir *invoice* pembelian yang dilakukan benar-benar ada dengan diketahui seluruh pihak pengelola sehingga disarankan untuk menambah fitur *upload* bukti pengeluaran pada menu pengeluaran laporan keuangan.

DAFTAR PUSTAKA

- A.S., R., & Shalahuddin, M. (2015). Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. In *Informatika Bandung*.
- Akbar Fadillah, M., Zakky Muttaqin, F., & Milly Van Barry Marbun, A. (2016). *Aplikasi Sistem Informasi Simpan Pinjam Pada Koperasi Usaha Bersama Syari'ah At-Tahwil Kota Tangerang*. 7–12.
- B. Shelly, G., & J. Rosenblatt, H. (2012). *System Analysis and Design: Ninth Edition*.
- Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Connallen, J., & Houston, K. A. (2007). Object-Oriented Analysis and Design With Applications: Third Edition. In *ACM SIGSOFT Software Engineering Notes* (Vol. 33, Issue 5). <https://doi.org/10.1145/1402521.1413138>
- Desideria, G., & Bandung, Y. (2020). *User Efficiency Model in Usability Engineering for User Interface Design Refinement of Mobile Application*. 14(1), 16–33. <https://doi.org/10.5614/itbj.ict.res.appl.2020.14.1.2>
- Devi, A. S., Wicaksono, S. A., & ... (2019). ... Sistem Informasi Manajemen Pendidikan Praktik Pengalaman Lapangan (Studi Kasus: Jurusan Sistem Informasi Program Studi Pendidikan Teknologi Informasi). ... *Teknologi Informasi Dan ...*, 3(9), 8787–8796. <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6266%0Ahttp://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/6266/3013>
- Fakhrillah, R. M. (2016). *Memodelkan traceability matrix untuk melakukan proses reverse engineering*. <http://repository.unj.ac.id/2446/>
- Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language. In *Pearson Paravia Bruno Mondad*.
- Hasanuddin. (2016). Sistem Informasi Keuangan Dengan Metode Object Oriented Analisis Design. *Jurnal Ilmiah Fakultas Teknik (Technologia)*, 7(2), 89–95.
- Hendini, A. (2016). Pemodelan UML Sistem Informasi Penjualan dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak). *Jurnal Khatulistiwa Informatika*, 4(2), 107–116. <https://doi.org/10.1145/358315.358387>
- Hidayat, A., & Meilina Fauziyyah, H. (2022). Perancangan Desain Antarmuka Aplikasi Pembelajaran Online Berbasis Mobile Menggunakan Metode Design Thinking. *Jurnal Teknik Informatika*, 10(1), 1–10.

- Hidayat, W., B. Wandayana, A., & Fadriansyah, R. (2016). PERANCANGAN VIDEO PROFILE SEBAGAI MEDIA PROMOSI DAN INFORMASI DI SMK AVICENA RAJEG TANGERANG. *CERITA*, 2 No. 1, 56–69.
- Himawan, H., & Yanu F, M. (2020). *Buku Ajar : Interface dan User Experience* (Issue 1).
- Kartiningrum, E. D. (2015). Panduan Penyusunan Studi Literatur. *Lembaga Penelitian Dan Pengabdian Masyarakat Politeknik Kesehatan Majapahit, Mojokerto*, 1–9.
- Maydianto, & Ridho, M. R. (2021). Rancang Bangun Sistem Informasi Point of Sale Dengan Framework Codeigniter Pada Cv Powershop. *Jurnal Comasie*, 02, 50–59.
- Muhidin, R., Kharie, N. F., & Kubais, M. (2019). Analisis Dan Perancangan Sistem Informasi Pada Sma Negeri 18 Halmahera Selatan Sebagai Media Promosi Berbasis Web Analysis and Information System Design in Sma Negeri 18 South Halmahera As Media Promotion of Web-Based. *IJIS-Indonesia Journal on Information System*, 4(April), 69–76. <https://media.neliti.com/media/publications/260171-sistem-informasi-pengolahan-data-pembeli-e5ea5a2b.pdf>
- Nasikhin, T. K., Hayuhardhika, W., Putra, N., & Pramono, D. (2019). Analisis dan Perancangan Sistem Informasi Reservasi Tour and Travel Menggunakan Metode OOAD Pada Warok Tour and Travel. *Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(11), 10659–10666.
- Nazaruddin, A., Krisnanik, E., Muliawati, A., & Syamsiah, N. (2022). Analisa Perancangan Sistem Informasi Berorientasi Objek. In *Angewandte Chemie International Edition*, 6(11), 951–952. www.penerbitwidina.com
- Nur Fitriani, S. (2017). *IMPLEMENTASI REQUIREMENT TRACEABILITY MATRIX PADA SISTEM INFORMASI AKADEMIK UNIVERSITAS NEGERI JAKARTA Skripsi*.
- Pane, E. S., & Sarno, R. (2015). Capability Maturity Model Integration (CMMI) for Optimizing Object-Oriented Analysis and Design (OOAD). *Procedia Computer Science*, 72(Cmmi), 40–48. <https://doi.org/10.1016/j.procs.2015.12.103>
- Pinheiro, F. A. C. (2004). Requirements Traceability: Chapter 5 Perspective on Software Requirements. In *Perspectives on Software Requirements* (pp. 91–113). https://doi.org/10.1007/978-1-4615-0465-8_5

- Pratama, I. A., & Purwidayanta, S. (2018). Sistem Informasi Geografis Lokasi Perumahan Di Kabupaten Tasikmalaya Berbasis Web. *Jurnal Manajemen Dan Teknik Informatika*, 02(01), 51–60. <http://jurnal.stmik-dci.ac.id/index.php/jumantaka/article/view/350/420>
- Pujaastawa, I. B. G. (2016). Teknik wawancara dan observasi untuk pengumpulan bahan informasi. *Universitas Udayana*, 4. https://simdos.unud.ac.id/uploads/file_penelitian_1_dir/8fe233c13f4addf4cee15c68d038aeb7.pdf
- Pujiyanto, A., & Alamsyah, N. (2022). Perancangan Sistem Informasi Kependudukan Di Perumahan Citra Swarna Riverside. *Semnas Ristek (Seminar Nasional Riset Dan Inovasi Teknologi)*, 6(1), 1189–1195. <https://doi.org/10.30998/semnasristek.v6i1.5873>
- Purnawati, E., Akademik, D., & Mada, U. G. (2019). *Sistem informasi dan OOAD*. 1–19.
- Putra, M. D. (2021). *Kebutuhan Manusia Dalam Pemikiran Abrahamam Maslow*. 91–94.
- S. Pressman, R. (2000). Software Engineering: a Practitioner's Approach. In *Software Engineering Journal* (Vol. 10, Issue 5). <https://doi.org/10.1049/sej.1995.0031>
- S. Pressman, R. (2010). *Software Engineering : A Practitioner's Approach*.
- S. Valacich, J., & F. George, J. (2017). *Modern Systems Analysis and Design : Eighth Edition*.
- Sandfreni, S., Ulum, M. B., & Azizah, A. H. (2021). Analisis Perancangan Sistem Informasi Pusat Studi Pada Fakultas Ilmu Komputer Universitas Esa Unggul. *Sebatik*, 25(2), 345–356. <https://doi.org/10.46984/sebatik.v25i2.1587>
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2012). *Systems Analysis And Design In A Changing World : Sixth Edition*.
- Septiani, Y., Aribbe, E., & Diansyah, R. (2020). ANALISIS KUALITAS LAYANAN SISTEM INFORMASI AKADEMIK UNIVERSITAS ABDURRAB TERHADAP KEPUASAN PENGGUNA MENGGUNAKAN METODE SEVQUAL (Studi Kasus : Mahasiswa Universitas Abdurrab Pekanbaru). *Jurnal Teknologi Dan Open Source*, 3(1), 131–143. <https://doi.org/10.36378/jtos.v3i1.560>
- Setiawan, D. (2019). Analisis Dan Perancangan Sistem Informasi LENTERA Untuk

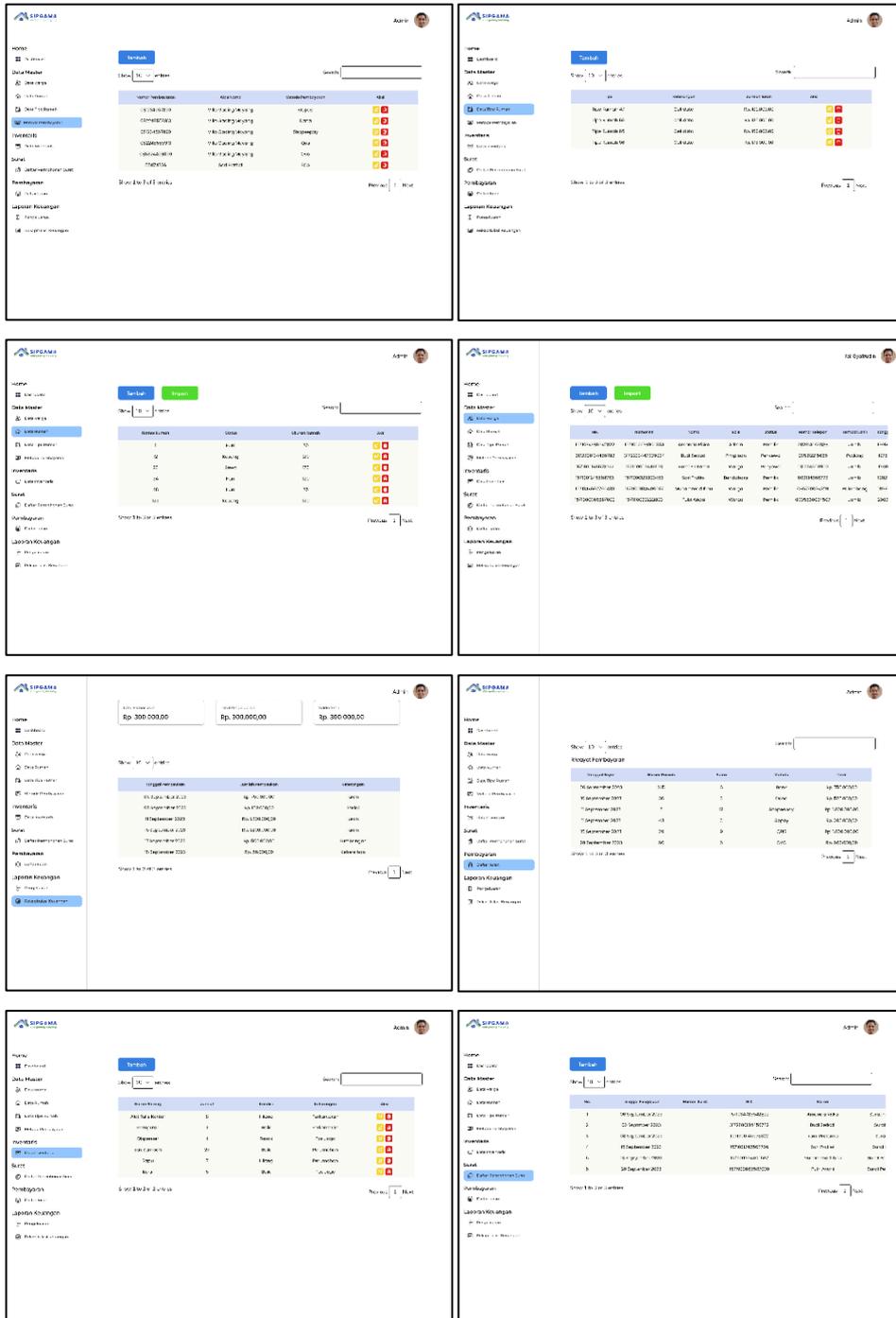
- Membentuk" Smart Society "Di Lingkungan Kampus Menggunakan Metode OOAD (Studi Kasus *Prosiding Seminar Nasional Teknologi ...*, 155–159. <http://prosiding.unipma.ac.id/index.php/SENATIK/article/view/1133>
- Siregar, W., Irvan, I., & Rahayu, E. (2020). Sistem Informasi Pembayaran Iuran Keamanan Dan Kebersihan Pada Perumahan Berbasis Website Menggunakan Metode Design Thinking. *JITEKH*, 8(2), 50–58. <https://doi.org/10.35447/jitek.v8i2.204>
- Sukamto, A. R., & Salahuddin, M. (2016). *Rekayasa Perangkat Lunak*. Bandung : Informatika.
- Sunarti. (2019). *Buku Ajar Perumahan Dan Permukiman Undip Press Semarang*.
- Suruali, N. (2015). *Perancangan dan Analisis Sistem Informasi Kependudukan*. 09(2).
- Syamsudin, A. (2015). Pengembangan Instrumen Evaluasi Non Tes (Informal) untuk Menjaring Data Kualitatif Perkembangan Anak Usia Dini. In *Jurnal Pendidikan Anak* (Vol. 3, Issue 1). <https://doi.org/10.21831/jpa.v3i1.2882>
- Syukron, A. (2019). Perancangan Sistem Informasi Administrasi Kependudukan Desa Berbasis Website Pada Desa Winong. *Bianglala Informatika*, 7(1), 16–21. <https://doi.org/10.31294/bi.v7i1.5790>
- Taufiq, R. (2013). Sistem Informasi Manajemen : Konsep Dasar , Analisis, dan Metode Pengembangan. In *Graha Ilmu*.
- Tukino, T. (2018). Perancangan Sistem Informasi Pelaporan Gangguan Dan Restitusi Pelanggan Internet Corporate Berbasis Web (Studi Kasus Di PT. Indosat Mega Media West Regional). *Jurnal Ilmiah Informatika*, 6(01), 1–10. <https://doi.org/10.33884/jif.v6i01.324>
- Utama, B. S. (2020). *Perancangan Ulang User Interface Dan User Experience Pada Website Cosmic Clothes*. 8–45.
- Westfall, L. (2006). Bidirectional Requirements Traceability. *White Paper, The Westfall Team, Dallas*. [http://www.westfallteam.com/Papers/Bi-Directional Requirements Traceability updated 2009 03 18.pdf](http://www.westfallteam.com/Papers/Bi-Directional%20Requirements%20Traceability%20updated%202009%2003%2018.pdf)
- Yanuardi, Y., & Permana, A. A. (2018). Rancang Bangun Sistem Informasi Keuangan Pada Pt. Secret Discoveries Travel and Leisure Berbasis Web. *JIKA (Jurnal Informatika)*, 2(2), 1–7. <https://doi.org/10.31000/.v2i2.1513>

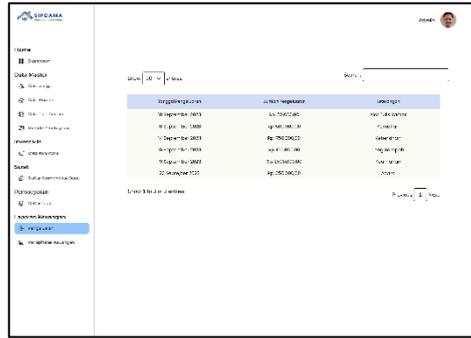
Yustrinita, M. Y., & Aprilia, S. O. (2021). Sistem Pengelolaan Data Warga, Informasi Kegiatan dan Informasi Penggunaan Iuran pada Lingkungan Rukun Tetangga (RT) Berbasis Web. *Jurnal Informatika Universitas Pamulang*, 6(1), 77. <https://doi.org/10.32493/informatika.v6i1.8283>

LAMPIRAN

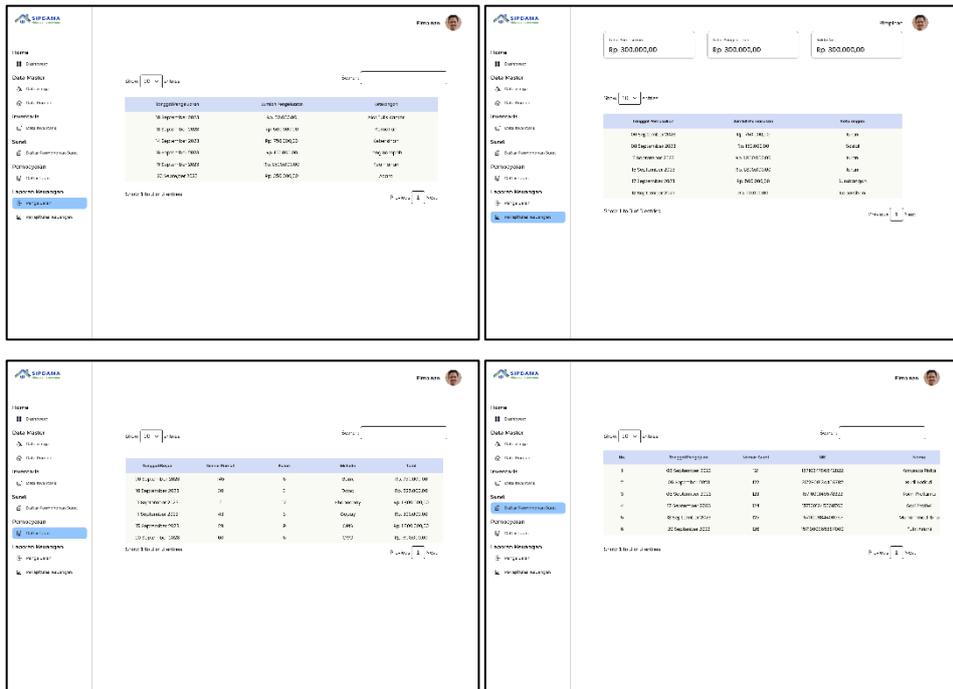
Lampiran 1. Wireframe High Fidelity SIPGAMA

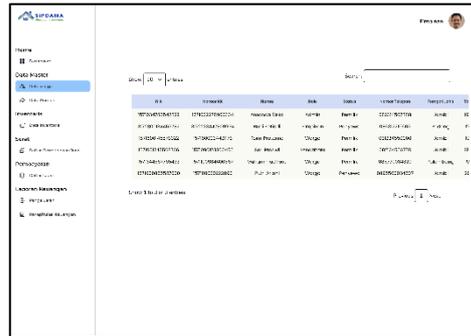
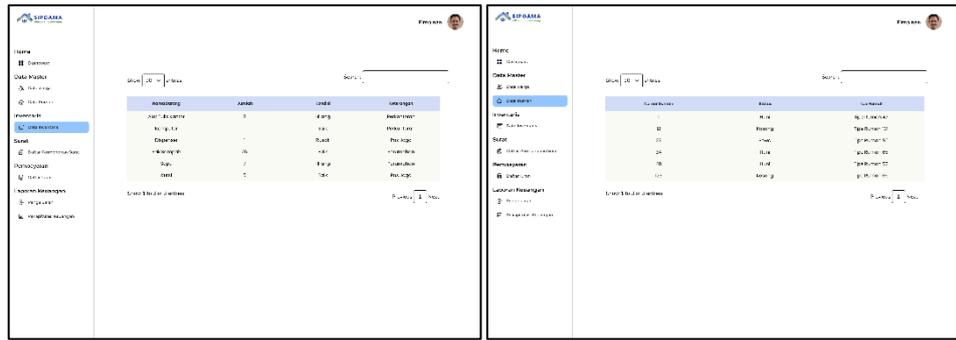
1. Wireframe Hi-Fid Admin



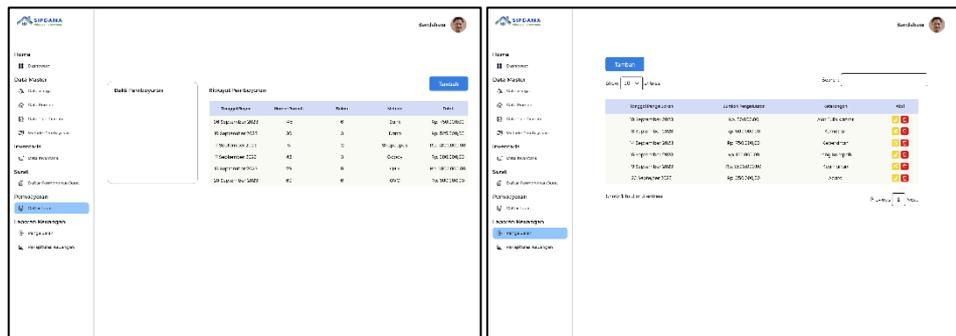


2. Wireframe Hi-Fid Pimpinan





3. Wireframe Hi-Fid Bendahara



Lampiran 2. Dokumentasi Penelitian

