

**ANALISIS DAN PERANCANGAN SISTEM PELAPORAN  
PEMBAYARAN PAJAK BERBASIS WEB  
STUDI KASUS : DINAS PENDAPATAN KABUPATEN MADIUN**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Krisdhamara Widyas Pritha  
NIM: 125150400111015



PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2016

## PENGESAHAN

ANALISIS DAN PERANCANGAN SISTEM PELAPORAN PEMBAYARAN PAJAK  
BERBASIS WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Nama Krisdhamara Widya Pritha

NIM: 125150400111015

Skripsi ini telah diuji dan dinyatakan lulus pada  
28 Juli 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ismiarta Aknuranda, S.T., M.Sc, Ph.D

NIK: 201006 740719 1 001

Satrio Agung Wicaksono, S.Kom, M.Kom

NIP: 19860521 201212 1 001

Mengetahui

Ketua Jurusan Sistem Informasi

Herman Tolle, Dr. Eng., S.T, M.T

NIP: 19740823 200012 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis di sitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 28 Juli 2016



Krisdhamara Widyas Pritha

NIM: 125150400111015

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Analisis dan Perancangan Sistem Pelaporan Pembayaran Pajak Berbasis Web”.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Ismiarta Aknuranda, S.T., M.Sc, Ph.D selaku dosen pembimbing I yang telah memberikan ilmu, kritik, serta saran dalam penyusunan skripsi ini.
2. Bapak Satrio Agung W., S.Kom, M.Kom, selaku dosen pembimbing II yang juga telah memberikan ilmu, kritik, serta saran dalam penyusunan skripsi ini.
3. Bapak Kasiyanto dan ibu Dra. Sri Setyani selaku orang tua dari penulis, yang selama ini telah sangat berjasa dalam membimbing dan membesarkan penulis, selalu mencurahkan kasih sayang, doa, serta berbagai dukungan moral maupun materi kepada penulis.
4. Seluruh Dosen dan seluruh *civitas* akademika Program Teknologi Informasi dan Ilmu Komputer yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Universitas Brawijaya.
5. Bapak Budi Muratno, Bapak Ari Nursurahmat, Mas Fajar dan seluruh keluarga besar Dinas Pendapatan Kabupaten Madiun yang bersedia membimbing serta menerima penulis untuk dapat melaksanakan pengerjaan skripsi di Dinas Pendapatan Madiun selama ini.
6. Krisjenta Iffah Agustasari dan Krismadha Ichwan Subekti selaku kakak-kakak dari penulis, serta seluruh keluarga besar yang telah memberikan dukungan baik secara moral maupun materi dan juga doa sehingga dapat membuat penulis lebih bersemangat dalam menyelesaikan skripsi.
7. Teman-teman Menantu Idaman yang merupakan teman seperjuangan dari semester satu hingga sekarang yang tidak pernah bosan memberikan dukungan, serta tidak segan dalam berbagi ilmu dan pengalaman yaitu Maulana Ghulam Hanifa, S.Kom; Anom Sulton Iskandar, S.Kom; Randi Dwi Nandra, S.Kom; Onny Iman Kandung Iriantono, S.Kom; Atikah Hayyu Mulia, S.Kom; Lia Hasanah, S.Kom; Putri Mutiara Tungga, S.Kom; Ika Qutsiati Utami, S.Kom; Rayyan Ramadhan, S.Kom; Diannisa Hakim, S.Kom; Frida Ayundha Putry, S.Kom; dan Mochamad Arizal, S.Kom.
8. Saudara Maulana Ghulam Hanifa, S.Kom yang telah memberikan semangat, meluangkan waktu dan selalu bersedia menemani dan membantu penulis saat merasa jenuh.
9. Teman-teman seperjuangan skripsi yang selama pengerjaan skripsi bersedia untuk membagi ilmu, *sharing*, dan memberikan semangat, yaitu : Dwi Retno Palupi, S.Kom; Isti Dyah Nur Safitri, S.Kom; Bella Pertiwi, S.Kom;

Nurriska Ajeng Utami, S.Kom; Voni Rizkia Ninda, S.Kom; Putri Mutiara Tungga, S.Kom; Lia Hasanah, S.Kom; Sabrina Lunettavashti, S.Kom; Brilliant Chindy Maharani, S.Kom; serta Nabilah Ulfah Aulia, S.Kom.

10. Seluruh Keluarga Besar Mahasiswa Sistem Informasi (KBMSI) yang telah memberikan pengalaman serta semangat dan dukungan dalam penulisan skripsi ini.
11. Seluruh keluarga besar Raion Community yang turut berpartisipasi dalam penelitian skripsi ini, serta mendukung dan memberikan banyak pengalaman yang berharga.
12. Seluruh teman-teman yang tidak dapat penulis sebutkan satu per satu yang telah memberikan semangat dan dukungan agar skripsi ini cepat terselesaikan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan.

Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 28 Juli 2016

Penulis  
Krisdhamara33@gmail.com



## ABSTRAK

Dinas Pendapatan Madiun merupakan sebuah badan atau organisasi yang berada dibawah pemerintahan Kabupaten Madiun yang menangani pemungutan pajak, retribusi, bagi hasil pajak, dana perimbangan, dan lain-lain. Jenis pajak yang ditangani pun cukup banyak, sekitar sebelas jenis pajak yang ditangani oleh Kabupaten yang disesuaikan dengan pada Undang-undang Republik Indonesia No. 28 Tahun 2009 tentang Pajak Daerah dan Retribusi Daerah BAB II Bagian Kesatu. Beberapa jenis pajak yang disebutkan diantaranya Pajak Hotel, Pajak Restoran, Pajak Hiburan, Pajak Reklame, dan lain sebagainya. Proses pengelolaan pajak terbilang masih kurang efisien karena dalam pengelolaan data masih membutuhkan waktu yang cukup lama. Selain itu, masih belum terdapat sistem yang bisa memantau penerimaan pajak dan pelaporan pajak.

Analisis dan perancangan merupakan tahapan penting dalam proses pengembangan perangkat lunak. Analisis persyaratan dan perancangan perangkat lunak sesungguhnya sangat berpengaruh terhadap kualitas perangkat lunak yang dihasilkan. Oleh karena itu, dilakukan penelitian ini untuk menghasilkan analisis dan perancangan sistem pelaporan pajak berbasis web. Penelitian ini menggunakan metode analisis dan perancangan berorientasi objek (*Object Oriented Analysis and Design*). Hasil penelitian ini menghasilkan spesifikasi persyaratan sistem pelaporan pajak berupa 11 persyaratan fungsional dan 2 persyaratan non fungsional, serta menghasilkan rancangan dari sistem pelaporan pajak diantaranya berupa diagram kelas, diagram *sequence*, *package*, dan *physical data model*. Selain itu juga dihasilkan sebuah prototipe yang dievaluasi untuk mengetahui kesesuaian dan kekurangan rancangan sistem terhadap spesifikasi persyaratan. Evaluasi ketelusuran juga dilakukan antara spesifikasi persyaratan dan rancangan sistem. Hasil evaluasi menunjukkan bahwa pengguna yang berperan sebagai partisipan evaluasi menyatakan rancangan sistem sudah sesuai dengan yang mereka harapkan. Selain itu, hasil perancangan dapat ditelusuri kembali sampai ke persyaratan sistem.

Kata kunci: Analisis dan perancangan sistem berorientasi objek, OOAD, Sistem Informasi, Pajak, Sistem Pelaporan Pembayaran Pajak.

## ABSTRACT

*Dinas Pendapatan Madiun is one of state owned institution or organization in Madiun regency which handles tax collection, retribution, tax revenues, equalization funds, and others. There are also many kind of taxes that are handled by this institution, approximately eleven kind of taxes are being handled by state government based on UU No. 28 Tahun 2009 about State Taxes and Retribution. Among those taxes are Hotel Taxes, Restaurant Taxes, Entertainment Taxes, Advertisement Taxes, and many more. The handling process of taxes as of now is not yet efficient, as this process still takes pretty long time. Besides of that, there is also no system to keep track of taxes receiving and reporting process.*

*Software analysis and design phases are important steps in a software development. Design and requirements analysis are affected to the quality of the software. According to those explanations, the research of design and analysis for web based taxes reporting system will be conducted. This research will use Object Oriented Analysis and Design method for the implementation. This research will deliver an 11 functional requirement and 2 non-functional requirements, and produce a design of taxes reporting system. Otherwise, will there a prototipe which will be evaluated to find out the compability between the prototipe with the system an alysis and design. Traceability evaluation also conducted between system design and requirement specification. The results showed that users who act as participants stated evaluation system design is in accordance with what they expect. In addition, the results of the design can be traced back to the system requirements. The evaluation results showed that users who act as participants stated evaluation system design is in accordance with what they expect. In addition, the results of the design can be traced back to the system requirements.*

*Keyword : Object Oriented Analysis and Design, OOAD, Information Systemi, Tax , Taxes reporting system.*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN .....	xviii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah.....	3
1.6 Sistematika pembahasan.....	4
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>5</b>
2.1 Tinjauan Pustaka .....	5
2.2 Dasar Teori.....	7
2.2.1 Pengertian Pajak Daerah.....	7
2.2.1.1 Wajib Pajak.....	7
2.2.1.2 Surat Ketetapan Pajak Daerah dan Surat Setoran Pajak Daerah .	7
2.2.2 DISPENDA Kabupaten Madiun .....	8
2.2.2.1 Sejarah DISPENDA Kabupaten Madiun .....	8
2.2.2.2 Visi dan Misi DISPENDA Kabupaten Madiun.....	8
2.2.2.3 Struktur Organisasi.....	9
2.2.3 Proses Bisnis.....	10
2.2.3.1 Business Process Modeling Notation (BPMN) .....	10
2.2.4 Sistem Informasi.....	16

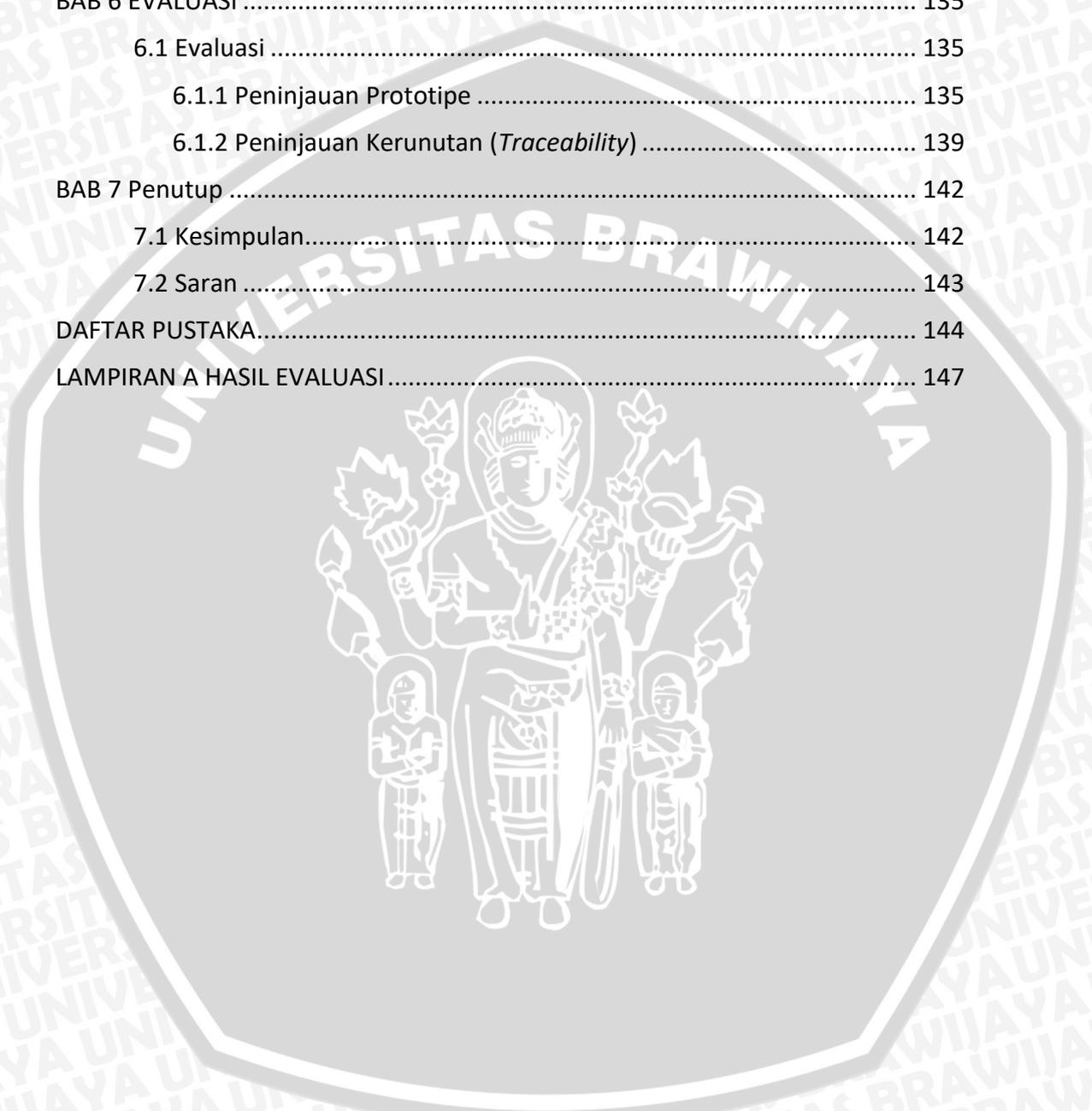
2.2.4.1 Rational Unified Process (RUP) .....	18
2.2.4.2 Analisis Sistem .....	19
2.2.4.3 Perancangan Sistem Informasi.....	23
2.2.4.4 Unified Model Language (UML) .....	26
2.2.4.5 Evaluasi.....	36
2.2.4.6 WEBUSE: Website Usability Evaluation Tool .....	38
2.2.4.7 Traceability .....	39
<b>BAB 3 METODOLOGI .....</b>	<b>41</b>
3.1 Studi Literatur .....	42
3.2 Analisis Masalah.....	42
3.3 Pengumpulan Data .....	43
3.4 Pemodelan Proses Bisnis .....	43
3.5 Analisis Persyaratan .....	44
3.6 Perancangan .....	44
3.7 Evaluasi .....	44
3.8 Kesimpulan.....	45
<b>BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN .....</b>	<b>46</b>
4.1 Pemodelan Proses Bisnis .....	46
4.1.1 Proses Bisnis Penerimaan PAD DISPENDA Kabupaten Madiun ..	46
4.1.2 Proses bisnis usulan .....	48
4.2 Analisis permasalahan .....	50
4.3 Analisis Pemangku Kepentingan dan Pengguna .....	50
4.3.1 Tipe Pemangku Kepentingan .....	50
4.3.2 Peran dan Representatif Pemangku Kepentingan .....	51
4.3.3 Tipe Pengguna.....	51
4.4 Analisis Persyaratan .....	52
4.4.1 Kebutuhan Pengguna dan Pemangku Kepentingan .....	52
4.4.2 Kedudukan Produk.....	54
4.4.3 Fitur Produk.....	54
4.4.4 Persyaratan Deklaratif .....	55
4.4.4.1 Persyaratan Fungsional .....	55
4.4.4.2 Persyaratan Non Fungsional .....	57



4.5	Pemodelan <i>Use Case</i> .....	57
4.5.1	Identifikasi Aktor .....	57
4.5.2	Diagram <i>Use Case</i> .....	58
4.6	Spesifikasi <i>Use Case</i> .....	59
4.7	Diagram Aktivitas .....	72
BAB 5 PERANCANGAN .....		84
5.1	Analisis Arsitektur .....	84
5.2	Analisis <i>Use Case</i> .....	84
5.2.1	Kelas-kelas Analisis .....	84
5.2.2	Mekanisme Analisis .....	88
5.2.3	Penggabungan Kelas Analisis .....	89
5.3	Pemodelan Interaksi .....	90
5.3.1	<i>Sequence Diagram</i> .....	90
5.4	Elemen Desain .....	101
5.4.1	<i>Package</i> .....	101
5.4.2	Diagram Kelas .....	101
5.5	Pemodelan Data .....	104
5.5.1	<i>Entity Relationship Diagram</i> .....	104
5.5.2	Perancangan Tabel .....	106
5.5.3	<i>Physical Data Model</i> .....	110
5.6	Perancangan Antarmuka Pengguna .....	111
5.6.1	Halaman <i>Login</i> .....	111
5.6.2	Notifikasi .....	112
5.6.3	Halaman Admin .....	112
5.6.4	Halaman Pembukuan dan Bendahara Umum Daerah .....	115
5.6.5	Halaman Perencanaan .....	119
5.6.6	Alur Laporan Pembayaran .....	123
5.7	Prototipe Sistem .....	126
5.7.1	Halaman <i>Login</i> .....	126
5.7.2	Notifikasi .....	127
5.7.3	Halaman Admin .....	127
5.7.4	Halaman Pembukuan dan Bendahara Umum Daerah .....	128



5.7.5 Halaman Perencanaan .....	130
5.7.6 Halaman Wajib Pajak .....	132
5.7.7 Alur Laporan Pembayaran.....	133
<b>BAB 6 EVALUASI .....</b>	<b>135</b>
6.1 Evaluasi .....	135
6.1.1 Peninjauan Prototipe .....	135
6.1.2 Peninjauan Kerunutan ( <i>Traceability</i> ) .....	139
<b>BAB 7 Penutup .....</b>	<b>142</b>
7.1 Kesimpulan.....	142
7.2 Saran .....	143
<b>DAFTAR PUSTAKA.....</b>	<b>144</b>
<b>LAMPIRAN A HASIL EVALUASI.....</b>	<b>147</b>



## DAFTAR TABEL

Tabel 2.1	Notasi Elemen BPMN .....	12
Tabel 2.2	Informasi pada Tipe Pemangku Kepentingan .....	20
Tabel 2.3	Informasi pada Tipe Pengguna .....	20
Tabel 2.4	Tabel Analisis Permasalahan .....	21
Tabel 2.5	Tabel Kedudukan Produk .....	21
Tabel 2.6	Tabel Penjelasan Fitur Produk.....	22
Tabel 2.7	Simbol-simbol Diagram Use Case.....	28
Tabel 2.8	Keterangan Simbol-Simbol Activity Diagram .....	30
Tabel 2.9	Simbol-Simbol Sequence Diagram .....	32
Tabel 2.10	Simbol-Simbol Class Diagram .....	35
Tabel 4.1	Aktivitas baru pada proses bisnis usulan .....	49
Tabel 4.2	Analisis Permasalahan.....	50
Tabel 4.3	Analisis Tipe Pemangku kepentingan.....	51
Tabel 4.4	Analisis Tipe Pengguna.....	52
Tabel 4.5	Analisis kebutuhan pengguna dan pemangku kepentingan .....	52
Tabel 4.6	Kedudukan Produk Sistem Pelaporan Pembayaran Pajak .....	54
Tabel 4.7	Fitur yang disediakan sistem .....	55
Tabel 4.8	Persyaratan Fungsional Sistem .....	56
Tabel 4.9	Persyaratan Non Fungsional .....	57
Tabel 4.10	Identifikasi Aktor .....	57
Tabel 4.11	Pemetaan Diagram Use Case .....	59
Tabel 4.12	Spesifikasi Use Case Kelola Data Pengguna .....	59
Tabel 4.13	Spesifikasi Use Case Registrasi Pengguna .....	61
Tabel 4.14	Spesifikasi Use Case Ubah Status Pembayaran.....	63
Tabel 4.15	Spesifikasi Use Case Validasi Laporan Pembayaran.....	64
Tabel 4.16	Spesifikasi Use Case Lihat data pembayaran pajak.....	65
Tabel 4.17	Spesifikasi Use case Lihat laporan target dan realisasi pajak .....	66
Tabel 4.18	Spesifikasi Use case Pantau pembayaran pajak.....	67
Tabel 4.19	Spesifikasi Use Case Laporkan pembayaran pajak.....	68
Tabel 4.20	Spesifikasi Use Case Cetak data pembayaran pajak .....	69

Tabel 4.21	Spesifikasi Use Case Cetak laporan rencana dan target pajak.....	70
Tabel 4.22	Spesifikasi Use Case Login .....	71
Tabel 5.1	Kelas Analisis .....	85
Tabel 5.2	Mekanisme Analisis.....	88
Tabel 5.3	Keterangan Rancangan Antarmuka Halaman Login .....	111
Tabel 5.4	Keterangan Rancangan Antarmuka Tampilan Notifikasi .....	112
Tabel 5.5	Keterangan Rancangan Antarmuka Halaman Daftar Pengguna.....	113
Tabel 5.6	Keterangan Rancangan Antarmuka Halaman Tambah Pengguna .....	114
Tabel 5.7	Keterangan Rancangan Antarmuka Halaman Kelola Pembayaran.....	115
Tabel 5.8	Keterangan Rancangan Antarmuka Halaman Ubah Status Pembayaran .....	117
Tabel 5.9	Keterangan Rancangan Antarmuka Halaman Laporan Pembayaran .....	117
Tabel 5.10	Keterangan Rancangan Antarmuka Halaman Laporan Pembukuan/Keuangan.....	118
Tabel 5.11	Keterangan Rancangan Antarmuka Halaman Laporan Target dan Realisasi Pajak .....	120
Tabel 5.12	Keterangan Rancangan Antarmuka Halaman Monitoring Pembayaran .....	121
Tabel 5.13	Keterangan Rancangan Antarmuka Halaman Data Pembayaran .....	122
Tabel 5.14	Keterangan Rancangan Antarmuka Halaman Alur Laporan Step 1.....	124
Tabel 5.15	Keterangan Rancangan Antarmuka Halaman Alur Laporan Step 2.....	124
Tabel 5.16	Keterangan Rancangan Antarmuka Halaman Alur Laporan Step 3 dan 4.....	125
Tabel 5.17	Keterangan Rancangan Antarmuka Halaman Alur Laporan Step 5.....	126
Tabel 6.1	Hasil Review Use case Pengguna 1 .....	135
Tabel 6.2	Hasil Review Use case Pengguna 2 .....	137
Tabel 6.3	Hasil Review Use case Pengguna 3 .....	138
Tabel 6.4	Matriks Traceability.....	140
Tabel 6.5	Software Requirement Checklist (Traceability) .....	141

## DAFTAR GAMBAR

Gambar 2.1	Matriks kerunutan Analisis dan Perancangan Sistem Antrian Bank Indonesia .....	6
Gambar 2.2	Bagan Struktur Organisasi DISPENDA Kabupaten Madiun.....	9
Gambar 2.3	Contoh Proses Kolaborasi.....	16
Gambar 2.4	Model Sistem Informasi .....	17
Gambar 2.5	Ilustrasi Fase dan Iterasi pada RUP .....	19
Gambar 2.6	Piramida Persyaratan .....	23
Gambar 2.7	Ilustrasi OOA.....	23
Gambar 2.8	IBM Best Practices .....	24
Gambar 2.9	Contoh Analisis Kelas.....	26
Gambar 2.10	Diagram Use Case Perilaku Pengguna menggunakan Telepon Seluler.....	29
Gambar 2.11	Contoh Diagram Aktivitas Proses Pemesanan pada Box Office .....	31
Gambar 2.12	Sequence Diagram Proses Transaksi dari Klien.....	33
Gambar 2.13	Diagram Kelas Aplikasi Box Office .....	35
Gambar 2.14	Representasi <i>Package</i> .....	36
Gambar 2.15	Model Pengembangan WEBUSE .....	39
Gambar 3.1	Metodologi Penelitian .....	41
Gambar 4.1	Proses bisnis penerimaan PAD DISPENDA Kab. Madiun saat ini .....	47
Gambar 4.2	Proses bisnis usulan penerimaan PAD .....	48
Gambar 4.3	Diagram Use Case Sistem Pelaporan Pembayaran Pajak.....	58
Gambar 4.4	Diagram Aktivitas Kelola data pengguna.....	73
Gambar 4.5	Diagram Aktivitas Registrasi Pengguna .....	74
Gambar 4.6	Diagram Aktivitas Ubah Status Pembayaran.....	75
Gambar 4.7	Diagram Aktivitas Validasi Laporan Pembayaran.....	76
Gambar 4.8	Diagram Aktivitas Lihat data pembayaran pajak.....	77
Gambar 4.9	Diagram Aktivitas Lihat laporan target dan realisasi pajak .....	78
Gambar 4.10	Diagram Aktivitas Pantau Pembayaran Pajak .....	79
Gambar 4.11	Diagram Aktivitas Case Laporkan Pembayaran Pajak .....	80

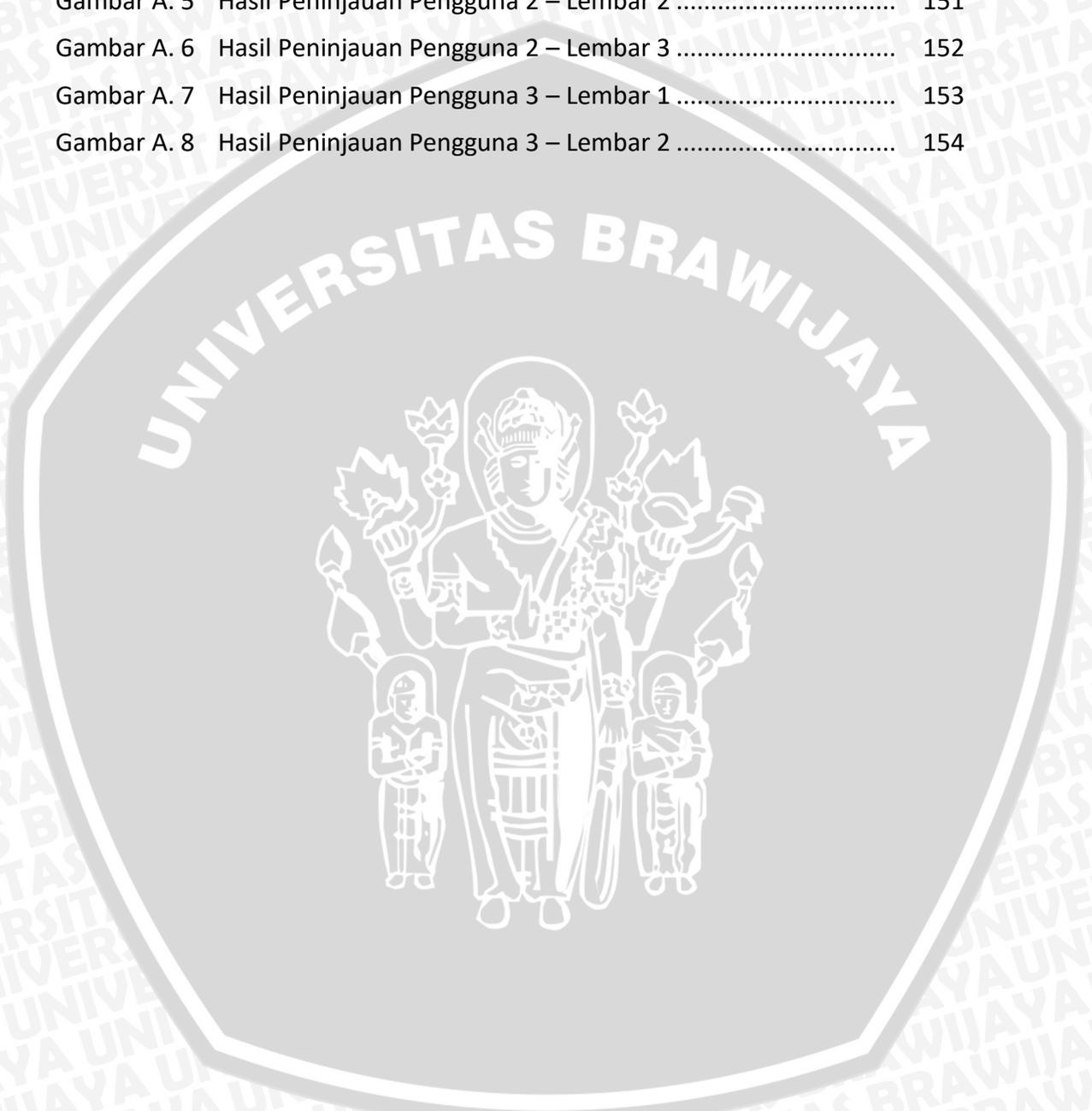
Gambar 4.12	Diagram Aktivitas Cetak data pembayaran pajak .....	81
Gambar 4.13	Diagram Aktivitas Cetak laporan dan target pajak.....	82
Gambar 4.14	Diagram Aktivitas Login .....	83
Gambar 5.1	Arsitektur Sistem .....	84
Gambar 5.2	Gabungan Kelas-kelas Analisis.....	90
Gambar 5.3	Sequence Diagram Kelola Data Pengguna .....	91
Gambar 5.4	Sequence Diagram Registrasi Pengguna .....	92
Gambar 5.5	Sequence Diagram Ubah Status Pembayaran.....	93
Gambar 5.6	Sequence Diagram Validasi Laporan Pembayaran.....	94
Gambar 5.7	Sequence Diagram Lihat Data Pembayaran Pajak .....	95
Gambar 5.8	Sequence Diagram Lihat Laporan Target dan Realisasi Pajak.....	96
Gambar 5.9	Sequence Diagram Pantau Pembayaran Pajak .....	97
Gambar 5.10	Sequence Diagram Laporkan Pembayaran Pajak.....	98
Gambar 5.11	Sequence Diagram Cetak Data Pembayaran Pajak .....	99
Gambar 5.12	Sequence Diagram Cetak Laporan Target dan Realisasi Pajak .....	99
Gambar 5.13	Sequence Diagram Login .....	100
Gambar 5.14	Package.....	101
Gambar 5.15	Diagram Kelas – Model.....	102
Gambar 5.16	Diagram Kelas- Controller.....	103
Gambar 5.17	Diagram Kelas - View .....	104
Gambar 5.18	<i>Entity Relationship Diagram</i> .....	105
Gambar 5.19	<i>Physical Data Model</i> .....	110
Gambar 5.20	Rancangan Antarmuka Halaman Login .....	111
Gambar 5.21	Rancangan Antarmuka Tampilan Notifikasi .....	112
Gambar 5.22	Rancangan Antarmuka Halaman Daftar Pengguna.....	112
Gambar 5.23	Rancangan Antarmuka Halaman Tambah Pengguna – Hotel .....	114
Gambar 5.24	Rancangan Antarmuka Halaman Tambah Pengguna - Restoran.....	114
Gambar 5.25	Rancangan Antarmuka Halaman Kelola Pembayaran.....	115
Gambar 5.26	Rancangan Antarmuka Halaman Ubah Status Pembayaran ....	116
Gambar 5.27	Rancangan Antarmuka Halaman Laporan Pembayaran.....	117



Gambar 5.28	Rancangan Antarmuka Halaman Laporan Pembukuan/Keuangan .....	118
Gambar 5.29	Rancangan Antarmuka Halaman Laporan Pembukuan/Keuangan - Semua .....	119
Gambar 5.30	Rancangan Antarmuka Halaman Laporan Target dan Realisasi - Grafik .....	119
Gambar 5.31	Rancangan Antarmuka Halaman Laporan Target dan Realisasi - Tabel .....	120
Gambar 5.32	Rancangan Antarmuka Halaman Monitoring Pembayaran .....	121
Gambar 5.33	Rancangan Antarmuka Halaman Data Pembayaran .....	122
Gambar 5.34	Rancangan Antarmuka Halaman Data Pembayaran - Semua .....	123
Gambar 5.35	Rancangan Antarmuka Halaman Alur Laporan Step 1 .....	123
Gambar 5.36	Rancangan Antarmuka Halaman Alur Laporan Step 2 .....	124
Gambar 5.37	Rancangan Antarmuka Halaman Alur Laporan Step 3 dan 4 .....	125
Gambar 5.38	Rancangan Antarmuka Halaman Alur Laporan Step 5 .....	126
Gambar 5.39	Antarmuka Halaman Login .....	126
Gambar 5.40	Antarmuka Tampilan Notifikasi .....	127
Gambar 5.41	Antarmuka Halaman Daftar Pengguna .....	127
Gambar 5.42	Antarmuka Halaman Tambah Pengguna .....	128
Gambar 5.43	Antarmuka Halaman Kelola Pembayaran .....	129
Gambar 5.44	Antarmuka Halaman Laporan Pembayaran .....	129
Gambar 5.45	Antarmuka Halaman Laporan Pembukuan/Keuangan .....	130
Gambar 5.46	Antarmuka Halaman Laporan Target dan Realisasi .....	130
Gambar 5.47	Antarmuka Halaman Monitoring Pembayaran Pajak .....	131
Gambar 5.48	Antarmuka Halaman Data Pembayaran .....	131
Gambar 5.49	Antarmuka Halaman Profil Pengguna .....	132
Gambar 5.50	Antarmuka Halaman Pelaporan Pembayaran Pajak .....	132
Gambar 5.51	Antarmuka Halaman Alur Laporan Pembayaran Step 1 .....	133
Gambar 5.52	Antarmuka Halaman Alur Laporan Pembayaran Step 2 .....	133
Gambar 5.53	Antarmuka Halaman Alur Laporan Pembayaran Step 3 dan 4 .....	134
Gambar 5.54	Antarmuka Halaman Alur Laporan Pembayaran Step 5 .....	134
Gambar A. 1	Hasil Peninjauan Pengguna 1 – Lembar 1 .....	147

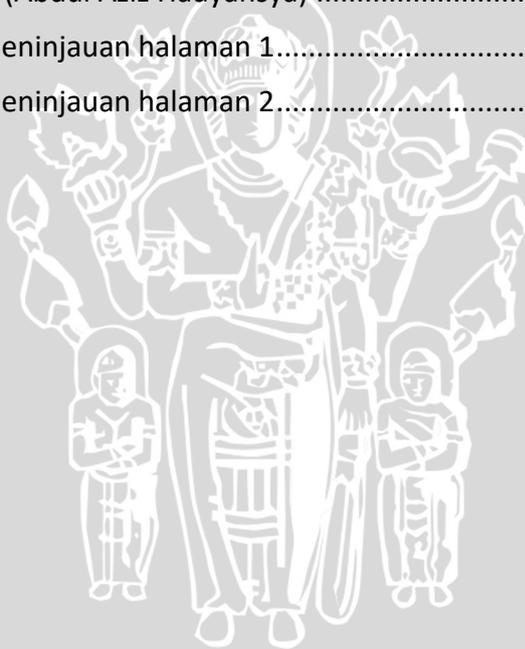


Gambar A. 2	Hasil Peninjauan Pengguna 1 – Lembar 2 .....	148
Gambar A. 3	Hasil Peninjauan Pengguna 1 – Lembar 3 .....	149
Gambar A. 4	Hasil Peninjauan Pengguna 2 – Lembar 1 .....	150
Gambar A. 5	Hasil Peninjauan Pengguna 2 – Lembar 2 .....	151
Gambar A. 6	Hasil Peninjauan Pengguna 2 – Lembar 3 .....	152
Gambar A. 7	Hasil Peninjauan Pengguna 3 – Lembar 1 .....	153
Gambar A. 8	Hasil Peninjauan Pengguna 3 – Lembar 2 .....	154



## DAFTAR LAMPIRAN

LAMPIRAN A HASIL EVALUASI .....	147
A.1 Pengguna 1 (Budi Muratno, S. Sos) .....	147
A.1.1 Hasil peninjauan halaman 1 .....	147
A.1.2 Hasil peninjauan halaman 2 .....	148
A.1.3 Hasil peninjauan halaman 3 .....	149
A.2 Pengguna 2 (Ari Nursurahmat, S. Sos) .....	150
A.2.1 Hasil peninjauan halaman 1 .....	150
A.2.2 Hasil peninjauan halaman 2 .....	151
A.2.3 Hasil peninjauan halaman 3 .....	152
A.3 Pengguna 3 (Abdul Aziz Hadyansya) .....	153
A.3.1 Hasil peninjauan halaman 1 .....	153
A.3.2 Hasil peninjauan halaman 2 .....	154



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Untuk meningkatkan kemajuan dan pembangunan pada sebuah daerah, sebuah Pemerintahan Daerah akan membutuhkan dana yang cukup besar. Dana yang dibutuhkan tersebut bisa didapatkan dari pusat maupun dari Pendapatan Asli Daerah (PAD). Seperti yang dijelaskan pada Undang-undang Republik Indonesia No. 28 Tahun 2009 tentang Pajak Daerah dan Retribusi Daerah, bahwa pajak daerah dan retribusi daerah merupakan salah satu sumber pendapatan daerah yang penting guna membiayai pelaksanaan pemerintahan daerah. Dari hal tersebut dapat dilihat bahwa pajak merupakan salah satu bentuk dari sumber pendapatan asli daerah yang sangat membantu dalam membiayai pelaksanaan pemerintahan dan pembangunan daerah dalam peningkatan layanan masyarakat.

Dinas Pendapatan Daerah (DISPENDA) terdapat dalam setiap daerah di Indonesia baik di kabupaten maupun di kota. DISPENDA Kabupaten Madiun merupakan salah satu dinas yang berada di Kabupaten Madiun yang bertugas untuk menangani dan melayani masalah pajak dan retribusi daerah Kabupaten Madiun. Jenis pajak yang ditangani pun cukup banyak, sekitar sebelas jenis pajak yang ditangani oleh Kabupaten yang disesuaikan dengan pada Undang-undang Republik Indonesia No. 28 Tahun 2009 tentang Pajak Daerah dan Retribusi Daerah BAB II Bagian Kesatu. Beberapa jenis pajak yang disebutkan diantaranya adalah Pajak Hotel, Pajak Restoran, Pajak Hiburan, Pajak Reklame, dan lain sebagainya. Pada Kabupaten Madiun sendiri terdapat sekitar 400.000 Wajib Pajak yang tersebar di 15 kecamatan (Muratno, 2015) dan dari 15 kecamatan tersebut terdapat 206 desa dan kelurahan. Jumlah Wajib Pajak tersebut merupakan jumlah total dari keseluruhan jenis pajak yang ada.

Proses pembayaran pajak oleh Wajib Pajak pada DISPENDA Kabupaten Madiun bisa dilakukan melalui dua pihak yaitu melalui bendahara penerimaan di Kantor DISPENDA atau melalui Bank Jatim. Saat ini DISPENDA Kabupaten Madiun telah bekerja sama dengan Bank Jatim untuk pelayanan pembayaran pajak daerah. Wajib Pajak harus membawa Surat Ketetapan Daerah (SKPD) yang sebelumnya telah diterima. Setiap harinya Bank Jatim bertugas untuk memasukkan uang pendapatan dari pembayaran pajak ke rekening DISPENDA dan mencetak rekening koran. Bank Jatim akan memberikan rekening koran dengan Surat Setoran Pajak Daerah (SSPD) kepada bagian pembukuan DISPENDA Kabupaten Madiun.

Bagian pembukuan kemudian akan membukukan atau mencatat setiap bukti setoran yang telah diberikan. Pada proses pencatatan atau pembukuan yang dilakukan menggunakan program *Microsoft Excel*. Bagian pembukuan melihat satu per satu rekening koran dari semua Wajib Pajak yang sudah membayar untuk kemudian dicatat. Hasil pembukuan pembayaran Wajib Pajak sangat diperlukan oleh bagian perencanaan. Bagian perencanaan menggunakan data pembayaran

Wajib Pajak untuk dilakukan evaluasi mengenai keberhasilan realisasi pendapatan dan perencanaan target pendapatan di tahun depan.

Namun, proses pengerjaan pembukuan saat ini membutuhkan waktu yang lama sehingga bisa dikatakan tidak efisien, karena pada saat bagian perencanaan menginginkan data pembayaran pajak pada bulan terkini, data tersebut baru bisa diperoleh dengan membutuhkan waktu antara 5-7 hari (Muratno, 2015). Bagian perencanaan pun merasa kesulitan dalam memantau pembayaran pajak yang dilakukan oleh wajib pajak. Selain itu, bagian perencanaan juga sering menemui adanya perbedaan yang cukup besar antara jumlah pendapatan atau realisasi PAD dengan jumlah pajak yang telah direncanakan (Muratno, 2015). Jadi untuk mempermudah pengolahan dan transparansi data pembayaran Wajib Pajak diperlukan sebuah sistem pelaporan pembayaran pajak berbasis web yang terpusat dan dapat dioperasikan secara *online*.

Pada pembuatan perangkat lunak, terdapat empat aktivitas umum yaitu analisis, perancangan, implementasi, dan pengujian. Aktivitas analisis merupakan aktivitas yang bertujuan untuk menggali, memodelkan, menganalisis, dan merumuskan persyaratan dari sistem yang akan dikembangkan. Pemodelan proses bisnis merupakan salah satu aktivitas yang biasanya mendahului analisis persyaratan untuk menggambarkan alur proses bisnis pada sebuah perusahaan baik proses bisnis yang saat ini diterapkan maupun yang diusulkan. Analisis persyaratan merupakan aktivitas yang penting, karena pada bagian inilah seluruh persyaratan akan dikumpulkan dan kesalahan persyaratan yang diperoleh akan berpengaruh pada hasil sistem yang dikembangkan.

Menurut Davis (1991) dan Leffingwell (1997) dalam Siahaan (2012) menyatakan bahwa 40% sampai dengan 60% kesalahan dalam suatu proyek pengembangan perangkat lunak yang mungkin muncul pada aktivitas berikutnya, berawal dari kesalahan yang dilakukan pada aktivitas spesifikasi persyaratan. Aktivitas selanjutnya adalah perancangan, aktivitas ini juga merupakan aktivitas yang penting dalam sebuah pengembangan sistem yang berhasil. Aktivitas perancangan merupakan aktivitas yang bertujuan untuk menerjemahkan spesifikasi atau prasyarat sistem yang akan dibangun sesuai dengan analisis yang telah dilakukan menjadi rancangan sistem agar lebih mudah diimplementasikan. Menurut Mitch Kapor (1990) dalam Pressman (2010) menjelaskan bahwa perancangan perangkat lunak sesungguhnya adalah tempat kualitas perangkat lunak ditetapkan.

Berdasarkan konsep mengenai analisis dan perancangan tersebut, dalam skripsi ini penulis melakukan penelitian untuk analisis dan perancangan dari sistem pelaporan pembayaran pajak berbasis web mengingat kedua aktivitas tersebut sangatlah penting untuk dilakukan dengan baik. Analisis dan perancangan sistem ini dilakukan dengan pendekatan berorientasi objek (*Object Oriented Analysis and Design* atau OOAD). Analisis berorientasi objek atau *Object Oriented Analysis* (OOA) adalah tahapan untuk menganalisis kebutuhan dan persyaratan sistem yang akan dibangun dengan konsep berorientasi objek. Perancangan berorientasi objek atau *Object Oriented Design* (OOD) adalah tahapan perantara untuk memetakan

spesifikasi atau kebutuhan sistem yang akan dibangun dengan konsep berorientasi objek ke desain pemodelan untuk nantinya diimplementasikan dengan pemrograman berorientasi objek. (A.S. & Shalahuddin, 2013).

## 1.2 Rumusan masalah

Berdasarkan uraian latar belakang tersebut, maka dapat dirumuskan masalah sebagai berikut :

1. Bagaimanakah proses bisnis pembayaran pajak yang saat ini telah diterapkan dan yang diusulkan sesuai dengan sistem yang akan dirancang?
2. Bagaimanakah spesifikasi persyaratan sistem pelaporan pembayaran pajak berbasis web yang diusulkan?
3. Bagaimanakah hasil perancangan sistem yang sesuai dengan persyaratan sistem yang telah dianalisis?
4. Bagaimanakah hasil evaluasi persyaratan dan rancangan sistem yang telah dibuat?

## 1.3 Tujuan

Adapun tujuan dari dilakukannya penelitian ini antara lain :

1. Memodelkan proses bisnis yang ada pada Dinas Pendapatan Kabupaten Madiun.
2. Memodelkan proses bisnis baru sesuai dengan sistem yang akan dibuat.
3. Menganalisa dan merancang sistem pelaporan pembayaran pajak dengan menggunakan pendekatan berorientasi objek.
4. Mengevaluasi persyaratan dan rancangan sistem pelaporan pembayaran pajak yang berupa prototipe.

## 1.4 Manfaat

Manfaat yang dapat diperoleh dari hasil penelitian ini bagi DISPENDA Kabupaten Madiun adalah sebagai berikut :

- Menghasilkan suatu perancangan yang nantinya bisa untuk dikembangkan dan digunakan oleh DISPENDA Kabupaten Madiun.
- Memberikan gambaran rancangan antarmuka sistem berdasarkan kebutuhan pengguna.
- Dapat menjalin kerja sama dengan pihak peneliti terkait.

## 1.5 Batasan masalah

Berdasarkan latar belakang dan rumusan masalah yang diuraikan, penelitian ini mempunyai batasan-batasan masalah sebagai berikut :

1. Penelitian ini menggunakan pendekatan *Object Oriented Analysis and Design*.
2. Perancangan sistem berupa perancangan sistem berbasis web.

3. Perancangan sistem yang dibuat melibatkan Wajib Pajak dari jenis pajak kabupaten berupa Pajak Restoran dan Pajak Hotel/Penginapan.
4. Data yang diperoleh merupakan hasil kuesioner dan wawancara langsung dengan pihak DISPENDA Kabupaten Madiun.
5. Pengguna yang dituju dari sistem ini adalah Wajib Pajak, pihak Kas Daerah, dan pihak DISPENDA Kabupaten Madiun.
6. Perancangan yang dilakukan merupakan perancangan awal yang sebaiknya dilanjutkan dengan perancangan detail di penelitian berikutnya sebelum memulai tahap implementasi.

## 1.6 Sistematika pembahasan

Sistematika penulisan dalam penyusunan skripsi ini adalah sebagai berikut :

1. BAB I : PENDAHULUAN  
Bab ini berisi latar belakang penulisan, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan skripsi.
2. BAB II : LANDASAN KEPUSTAKAAN  
Memaparkan teori dasar dan referensi yang mendukung analisis dan perancangan sistem informasi dengan metode *Object Oriented Analysis and Design*.
3. BAB III : METODOLOGI  
Membahas metode penelitian yang digunakan dalam penelitian yang terdiri dari studi literatur dan pemilihan metode *Object Oriented Analysis and Design* yang tepat dalam menganalisa kebutuhan pengguna.
4. BAB IV : PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN  
Membahas analisis kebutuhan pengguna dan perancangan sistem yang disesuaikan dengan kebutuhan dan metode yang digunakan.
5. BAB V : PERANCANGAN  
Membahas analisis kebutuhan pengguna dan perancangan sistem yang disesuaikan dengan kebutuhan dan metode yang digunakan.
6. BAB VI : EVALUASI  
Membahas tentang evaluasi yang dilakukan untuk mengetahui apakah perancangan yang dibuat sesuai dengan persyaratan sistem yang dianalisis dan harapan pengguna.
7. BAB VI : PENUTUP  
Memuat kesimpulan dari keseluruhan penelitian yang sudah dilakukan dan saran yang diberikan berdasarkan penelitian yang sudah dilakukan.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Tinjauan Pustaka

“Analisis dan Perancangan Sistem Informasi Penerapan *Preventive Maintenance* Untuk Menentukan Jadwal Perawatan Pencegahan Yang Optimum Dan Meningkatkan Keandalan Komponen Kritis Mesin Hd/Pe-120 Pada Pt. Metropoly Jaya Nusa” yang dilakukan oleh Erlina pada tahun 2007 merupakan sebuah penelitian mengenai analisis dan perancangan sistem informasi yang bertujuan untuk membuat sistem informasi yang dapat membantu bagian *maintenance* perusahaan untuk mengantisipasi kapan terjadinya kerusakan pada mesin sehingga lebih siap dan dapat mengetahui waktu yang tepat untuk melakukan perawatan terhadap mesin-mesin yang ada pada perusahaan.

Pada penelitian yang dilakukan oleh Erlina, analisis dan perancangan dari sistem yang dikembangkan menggunakan metode analisis dan perancangan berbasis objek (*Object Oriented Analysis and Design*). Pada penelitian tersebut menyebutkan bahwa OOAD memiliki keunggulan, yaitu *reusability* dan *interoperability* yang dapat menghasilkan keunggulan seperti (McLeod, 2001):

- Peningkatan kecepatan pengembangan, karena sistem dirancang seperti dunia nyata melihatnya.
- Pengurangan biaya pengembangan karena pengembangan lebih cepat.
- Kode pemrograman berkualitas tinggi, bisa memberikan keandalan lebih besar dan ketangguhan yang lebih dibandingkan yang biasa ditemukan dalam sistem berorientasi proses dan data.
- Pengurangan biaya pemeliharaan dan rekayasa ulang sistem karena kode pemrograman yang berkualitas tinggi dan kemampuan pemakaian pembeli.

Pada penelitian tersebut, dalam pelaksanaan tahapan analisis dan perancangan dari sistem yang dikembangkan menerapkan konsep dasar OOAD yang terdiri dari aktivitas *Problem Domain Analysis*, *Application Domain Analysis*, *Architectural Design*, dan *Component Design*. Selain itu, penelitian tersebut juga menggunakan *Unified Modeling Language* sebagai bahasa pemodelan untuk merancang sistem. Penelitian tersebut memiliki kesamaan dengan penelitian yang peneliti lakukan (Analisis dan Perancangan Sistem Pelaporan Pajak Berbasis Web) yaitu menggunakan metode analisis dan perancangan berorientasi objek (OOAD), serta menggunakan bahasa pemodelan UML. Namun tahapan analisis dan perancangan pada penelitian tersebut akan berbeda dengan tahapan penelitian analisis dan perancangan dari sistem pelaporan pembayaran pajak. Pada penelitian analisis dan perancangan sistem pelaporan pembayaran pajak akan menerapkan langkah-langkah yang sesuai dengan panduan yang dibuat oleh IBM yaitu *IBM Mastering OOAD with UML 2.0*.

Penelitian lain yang berjudul “Analisis dan Perancangan Sistem Antrian Terintegrasi di Bank Indonesia” merupakan penelitian yang dilakukan Renja Mantani pada tahun 2014 yang difokuskan pada layanan kas Bank Indonesia. Penelitian tersebut merupakan penelitian mengenai analisis dan perancangan sebuah sistem antrian pada Bank Indonesia yang dapat terintegrasi dengan sistem atau aplikasi yang sudah diterapkan pada instansi tersebut. Tujuan penelitian ini dilakukan adalah analisis dan perancangan sistem antrian baru guna meningkatkan efisiensi, keamanan dan produktivitas dari layanan kas di Bank Indonesia. Dalam penelitian tersebut, tahapan yang dilakukan adalah melakukan analisis dan perancangan sistem serta melakukan evaluasi dan validasi sistem.

Pada tahapan evaluasi sistem yang dikembangkan, penelitian tersebut menggunakan teknik *traceability* atau kerunutan untuk mengetahui apakah desain yang dibuat telah sesuai dengan kebutuhan fungsional dan *use case* yang dibuat. Evaluasi dilakukan dengan membuat matriks kerunutan yang menyesuaikan antara kebutuhan fungsional (KF), *use case* (UC), diagram aktivitas (AD), diagram *sequence* (SD), dan rancangan antarmuka. Matriks kerunutan yang dibuat pada penelitian tersebut dapat dilihat pada Gambar 2.1.

Kebutuhan Fungsional	Use-case	Kode Use-case	Activity Diagram	Sequence Diagram	Interface
KF-01	Menambahkan akun	UC-01.01	Gambar AD-1	Gambar SD-1	Gambar I-1
KF-01	Masuk akun	UC-01.02	Gambar AD-2	Gambar SD-2	Gambar I-2
KF-01	Keluar akun	UC-01.03	Gambar AD-3	Gambar SD-3	Gambar I-3
KF-02	Lihat daftar akun	UC-01.04	Gambar AD-4	Gambar SD-4	Gambar I-4
KF-02	Ubah akun	UC-01.05	Gambar AD-5	Gambar SD-5	Gambar I-5 Gambar I-6
KF-02	Hapus akun	UC-01.06	Gambar AD-6	Gambar SD-6	Gambar I-7 Gambar I-8
KF-02	Menonaktifkan akun	UC-01.07	Gambar AD-7	Gambar SD-7	Gambar I-9 Gambar I-10
KF-02	Mengaktifkan kembali akun	UC-01.08	Gambar AD-8	Gambar SD-8	Gambar I-11 Gambar I-12
KF-05	Hapus nasabah bank terdaftar	UC-01.09	Gambar AD-9	Gambar SD-9	Gambar I-13 Gambar I-14

**Gambar 2.1 Matriks kerunutan Analisis dan Perancangan Sistem Antrian Bank Indonesia**

Matriks kerunutan yang digunakan oleh Renja Mantani pada penelitian Analisis dan Perancangan Sistem Antrian Terintegrasi di Bank Indonesia sesuai dengan matriks yang akan diterapkan pada penelitian yang saat ini peneliti lakukan (Analisis dan Perancangan Sistem Pelaporan Pembayaran Pajak Berbasis Web). Namun, matriks kerunutan pada penelitian Analisis dan Perancangan Sistem Pelaporan Pembayaran Pajak Berbasis Web menyesuaikan antara proses bisnis, kebutuhan fungsional, *use case*, *activity diagram*, *sequence diagram*, dan rancangan antarmuka.

## 2.2 Dasar Teori

### 2.2.1 Pengertian Pajak Daerah

Pajak Daerah yang selanjutnya disebut pajak adalah iuran wajib yang dilakukan oleh pribadi atau badan kepada daerah tanpa imbalan langsung yang seimbang, dapat dipaksakan berdasarkan peraturan perundangan yang berlaku dan digunakan untuk penyelenggaraan pemerintah dan pembangunan daerah (Sugiyanto, 2007). Jenis pajak sendiri dibagi menjadi dua jenis yaitu Jenis Pajak Provinsi dan Jenis Pajak Kabupaten/Kota. Jenis pajak Kabupaten/Kota sendiri terdiri atas :

- a. Pajak Hotel;
- b. Pajak Restoran;
- c. Pajak Hiburan;
- d. Pajak Reklame;
- e. Pajak Penerangan Jalan;
- f. Pajak Mineral Bukan Logam dan Batuan;
- g. Pajak Parkir;
- h. Pajak Air Tanah;
- i. Pajak Sarang Burung Walet;
- j. Pajak Bumi dan Bangunan Pedesaan dan Perkotaan; dan
- k. Bea Perolehan dan Hak atas Tanah dan Bangunan.

#### 2.2.1.1 Wajib Pajak

Menurut Undang - Undang Republik Indonesia Nomor 28 Tahun 2009, pengertian Wajib Pajak adalah orang pribadi atau Badan, meliputi pembayar pajak, pemotong pajak, dan pemungut pajak, yang mempunyai hak dan kewajiban perpajakan sesuai dengan ketentuan peraturan perundang-undangan perpajakan daerah. Menurut Markus (2005) Wajib Pajak (*tax payer*) atau WP adalah subjek pajak yang menerima atau memperoleh penghasilan. Jika orang pribadi atau badan telah menerima atau memperoleh penghasilan (dalam teori pajak disebut telah memenuhi syarat-syarat objektif) berarti dia otomatis menjadi Wajib Pajak yakni wajib membayar pajak atas penghasilan yang telah diterima atau diperolehnya tersebut. Seorang atau badan Wajib Pajak harus membayar pajak yang telah ditetapkan oleh pemerintah daerah.

#### 2.2.1.2 Surat Ketetapan Pajak Daerah dan Surat Setoran Pajak Daerah

Menurut Undang - Undang Republik Indonesia Nomor 28 Tahun 2009, dijelaskan bahwa Surat Ketetapan Pajak Daerah, yang selanjutnya disingkat SKPD, adalah surat ketetapan pajak yang menentukan besarnya jumlah pokok pajak yang terutang. Sedangkan Surat setoran Pajak Daerah, yang selanjutnya disingkat SSPD, adalah bukti pembayaran atau penyetoran pajak yang telah dilakukan dengan

menggunakan formulir atau telah dilakukan dengan cara lain ke kas daerah melalui tempat pembayaran yang ditunjuk oleh Kepala Daerah.

SKPD ditetapkan sendiri oleh masing-masing daerah yang kemudian akan didistribusikan kepada setiap Wajib Pajak baik secara langsung maupun dikirimkan ke setiap desa atau kecamatan. SSPD didapatkan oleh Wajib Pajak saat sudah melakukan pembayaran pajak, dalam penelitian ini Wajib Pajak melakukan pembayaran melalui Bank Jatim.

## **2.2.2 DISPENDA Kabupaten Madiun**

Dinas Pendapatan Kabupaten Madiun atau biasa disebut DISPENDA Kabupaten Madiun adalah sebuah badan atau organisasi yang berada dibawah pemerintahan Kabupaten Madiun yang menangani pemungutan pajak, retribusi, bagi hasil pajak, dana perimbangan, dan lain-lain. Saat ini Kantor Dinas Pendapatan Kabupaten Madiun berlokasi di Jalan Alun-alun Utara No.4 Kota Madiun.

### **2.2.2.1 Sejarah DISPENDA Kabupaten Madiun**

Dinas Pendapatan merupakan unsur pelaksana otonomi dibidang pendapatan daerah, dan pengelolaan pasar yang dipimpin oleh Kepala Dinas, yang nantinya akan dipertanggung jawab kan kepada Bupati melalui Sekretaris Daerah. Dalam hal ini, Dinas Pendapatan mempunyai tugas melaksanakan urusan pemerintahan dalam bidang pendapatan pajak dan retribusi daerah berdasarkan asas otonomi dan tugas pembantuan.

Awalnya, pada tahun 2009 dalam satu struktur organisasi, Pemerintah Kabupaten Madiun tergabung dalam satu garis struktur organisasi dengan Kantor Pengelola Keuangan Aset Daerah sampai tahun 2011. Kemudian, dalam Peraturan Daerah Kabupaten Madiun Nomor 13 Tahun 2011 tentang organisasi perangkat daerah, dijelaskan bahwa pada bulan Januari 2012 Dinas Pendapatan berdiri sendiri menjadi Dinas Pendapatan Kabupaten Madiun. Maka berdasarkan Peraturan Daerah tersebut pada bulan Januari 2012 Dinas Pendapatan Kabupaten Madiun berdiri sendiri hingga sekarang.

### **2.2.2.2 Visi dan Misi DISPENDA Kabupaten Madiun**

Adapun visi dari DISPENDA Kabupaten Madiun adalah sebagai berikut :

Tercapainya pendapatan daerah yang optimal, berkesinambungan serta seimbang dengan potensi daerah untuk mewujudkan kemandirian pemerintahan kabupaten Madiun dalam rangka meningkatkan kesejahteraan masyarakat, serta menjadi inisiator peningkatan pendapatan asli daerah.

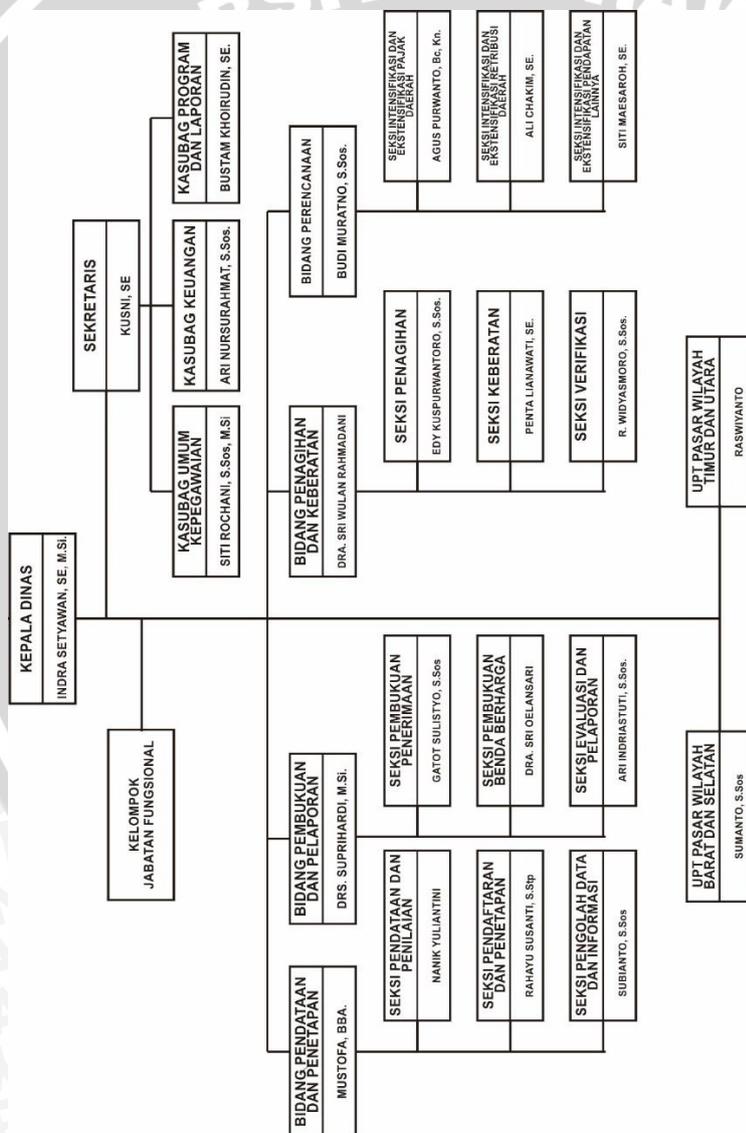
Sedangkan misi yang dimiliki oleh DISPENDA Kabupaten Madiun adalah :

1. Peningkatan kualitas pelayanan masyarakat yang berorientasi pada kepuasan serta pembudayaan taat pajak dan retribusi melalui pendayagunaan penyuluhan dan peningkatan peran serta masyarakat.

2. Pemantapan komitmen sumber daya manusia (SDM) melalui peningkatan kinerja guna menunjang kualitas manajemen penerimaan pendapatan daerah.

### 2.2.2.3 Struktur Organisasi

Saat ini DISPENDA Kabupaten Madiun dipimpin atau diketuai oleh Bapak Indra Setyawan, SE, M.Si. dan dibantu oleh sekretaris yang dijabat oleh Bapak Kusni, SE yang kemudian bagian sekretaris dibagi lagi menjadi tiga bagian yaitu bagian kepegawaian, keuangan, serta bagian program dan pelaporan. Selain itu ada bagian-bagian lain yang menangani langsung masalah perpajakan seperti yang digambarkan pada Gambar 2.1 Bagan Struktur Organisasi DISPENDA Kabupaten Madiun.



Gambar 2.2 Bagan Struktur Organisasi DISPENDA Kabupaten Madiun

### 2.2.3 Proses Bisnis

Proses bisnis didefinisikan sebagai aktivitas yang terukur dan terstruktur untuk memproduksi keluaran tertentu untuk kalangan pelanggan tertentu. Terdapat di dalamnya penekanan yang kuat pada “bagaimana” pekerjaan itu dijalankan di suatu organisasi, tidak seperti fokus dari sebuah produk yang berfokus pada aspek “apa”. Suatu proses, oleh karenanya merupakan urutan spesifik dari aktivitas kerja lintas waktu dan ruang, dengan suatu awalan dan akhiran, dan secara jelas mendefinisikan masukan dan keluaran (Davenport, 1993). Aktivitas / kegiatan dalam proses bisnis ini dapat berupa aktivitas sistem, aktivitas interaksi pengguna atau aktivitas manual (Weske, 2007).

Aktivitas manual adalah aktivitas yang tidak didukung oleh suatu sistem informasi, misalnya mengirim bingkisan kepada mitra bisnis. Aktivitas interaksi pengguna adalah kegiatan / aktivitas yang berupa melakukan pekerjaan dengan menggunakan sistem informasi. Contoh aktivitas interaksi pengguna yaitu memasukkan data asuransi dalam lingkungan *call center*. Aktivitas sistem adalah aktivitas yang dijalankan oleh sistem informasi dan tidak melibatkan pengguna. Contoh aktivitas sistem adalah mengambil informasi saham dari aplikasi broker saham atau memeriksa saldo rekening bank (Weske, 2007).

Menurut Weske (2007), proses bisnis dapat direpresentasikan atau digambarkan dengan model proses bisnis yang disajikan dalam suatu diagram proses bisnis. Setiap diagram proses bisnis tersebut terdiri dari satu set elemen pemodelan. Dalam pemodelan proses bisnis dapat menggunakan sebuah metode pemodelan yaitu *Business Process Modelling Notation* (BPMN). Menurut Owen dan Raj (2003) dalam Wulandari (2003) BPMN memiliki tujuan untuk menyediakan notasi yang mudah dipahami oleh semua pengguna bisnis dan tidak kalah pentingnya adalah untuk memastikan bahwa bahasa XML yang dirancang untuk pelaksanaan proses bisnis dapat dinyatakan secara visual dengan notasi yang umum.

#### 2.2.3.1 Business Process Modeling Notation (BPMN)

*Business Process Modeling Notation* atau BPMN adalah notasi grafis yang digunakan untuk memodelkan proses bisnis. BPMN biasa digunakan untuk menggambarkan diagram proses bisnis. Diagram proses bisnis merupakan diagram merepresentasikan aktivitas-aktivitas dan tugas-tugas dari suatu proses dan hubungan antar keduanya. Salah satu tujuan BPMN adalah untuk mengembangkan notasi yang dapat dimengerti oleh semua orang yang terlibat dalam proses bisnis. Pemodelan proses bisnis melibatkan berbagai kalangan, mulai dari pengguna bisnis, analis bisnis, dan pemilik sampai dengan arsitek teknis dan pengembang (*developer*) (Pant & Juric, 2008).

Menurut *Object Management Group* (2011), BPMN memiliki lima kategori elemen dasar yaitu:

1. *Flow Object*

*Flow Object* merupakan elemen grafis utama untuk menentukan perilaku dari proses bisnis. Terdapat tiga macam *Flow Object* yang didefinisikan yaitu :

- a. *Events*: merupakan sesuatu yang terjadi selama proses bisnis berjalan. Event yang mempengaruhi aliran proses dan biasanya memiliki pemicu atau hasil. Ada tiga jenis *event* yang terdapat dalam BPMN yaitu *start*, *intermediate*, dan *end*.
- b. Aktivitas
- c. *Gateways*.

2. Data

Data direpresentasikan oleh empat elemen utama yaitu :

- a. Data Objects
- b. Data Inputs
- c. Data Outputs
- d. Data Stores.

3. *Connecting Object*

Dalam menghubungkan antara *Flow Object* maupun informasi lainnya, dapat menggunakan empat jenis penghubung, yaitu :

- a. *Sequence Flow*: menunjukkan urutan *flow elements* dalam suatu proses atau *choreography*. *Flow Elements* terdiri dari:
  - b. *Events (Start, Intermediate, dan End)*
  - c. *Activities (Task dan Sub-Process; untuk Proses)*
  - d. *Choreography Activities (Choreography Task dan Sub-Choreography; untuk Choreographies)*
  - e. *Gateways*
  - f. *Message Flows*
  - g. *Association*
  - h. *Data Associations*.

4. *Swimlane*

*Swimlane* merupakan tempat untuk mengelompokkan elemen pemodelan primer, pengelompokannya dapat menggunakan dua cara yaitu :

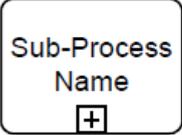
- a. Pool
- b. Lanes

5. *Artifact*

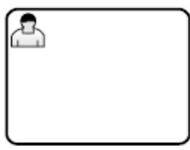
*Artifact* merupakan elemen yang digunakan untuk memberikan informasi tambahan dari proses. Terdapat dua *Artifact* standar yaitu *Group* dan *Text Annotation*.

Setiap elemen yang sudah dijelaskan memiliki notasi tersendiri yang bisa dilihat contoh notasinya pada Tabel 2.1 Notasi Elemen BPMN.

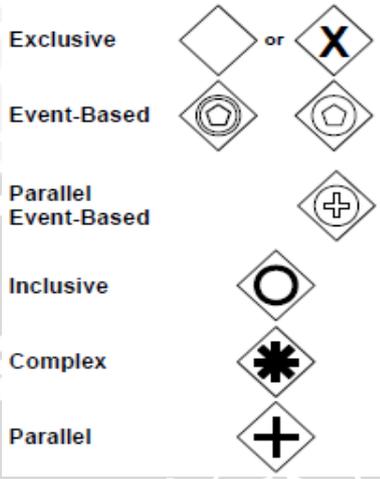
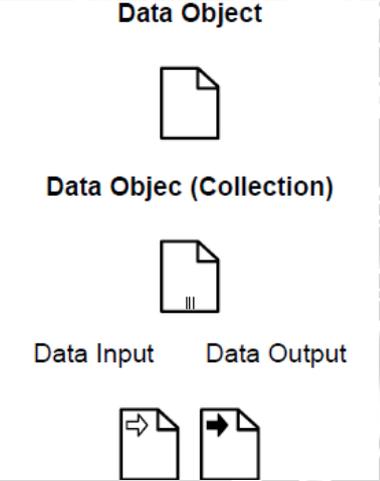
Tabel 2.1 Notasi Elemen BPMN

Nama	Notasi	Keterangan
<b>Event</b>		
Start Event		Notasi untuk memulai proses
End Events		Notasi untuk mengakhiri proses
Intermediate Events		Notasi yang menunjukkan terjadi kegiatan lain di antara awal dan akhir proses. <i>Intermediate event</i> akan mempengaruhi aliran proses tetapi tidak memulai atau mengakhiri proses
<b>Task</b>		
Activity		Kegiatan atau pekerjaan yang dilakukan oleh perusahaan dalam sebuah proses.
Collapsed Sub-Process		Merupakan aktivitas yang mana didalam aktivitas tersebut terdapat proses yang terjadi.
Service Task		Kegiatan yang menggunakan beberapa jenis layanan, dapat berupa <i>web service</i> atau aplikasi otomatis.
Send Task Object		Merupakan jenis kegiatan yang didesain untuk mengirimkan pesan kepada partisipan luar dalam sebuah proses.
Receive Task Object		Merupakan jenis kegiatan yang didesain untuk menunggu pesan dari partisipan luar dalam sebuah proses.

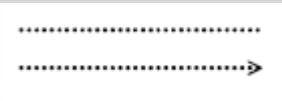
Tabel 2.1 Notasi Elemen BPMN (lanjutan)

Nama	Notasi	Keterangan
<i>Receive Task Object</i> yang menginstansiasi proses		Sama seperti <i>Receive Task Object</i> , namun jika atribut instansiasi diatur menjadi <i>true</i> maka notasinya digambarkan dengan gambar pesan didalam sebuah lingkaran.
<i>User Task Object</i>		Penggabaran sebuah kegiatan, adapun yang melakukan kegiatan tersebut adalah orang atau manusia dengan bantuan sebuah aplikasi.
<i>Manual Task Object</i>		Kegiatan yang dilakukan tanpa menggunakan alat proses bisnis atau aplikasi apa pun.
<i>Business Rule Task Object</i>		Menyediakan mekanisme untuk sebuah proses untuk memberikan masukan ke <i>Business Rule Engine</i> dan untuk mendapatkan keluaran yang mungkin diberikan oleh <i>Business Rule Engine</i> .
<i>Script Task Object</i>		Menunjukkan kegiatan sebuah mesin proses bisnis yang menjalankan suatu skrip.
<b>Gateway</b>		
<i>Gateway</i>		Menunjukkan bagaimana interaksi <i>sequence flow</i> dalam sebuah proses.

Tabel 2.1 Notasi Elemen BPMN (lanjutan)

Nama	Notasi	Keterangan
<p>Gateway Control Types</p>		<ul style="list-style-type: none"> <li>• <b>Exclusive (Keputusan):</b> digunakan untuk membuat jalur alternatif dalam sebuah aliran proses.</li> <li>• <b>Event-based:</b></li> <li>• <b>Inclusive:</b> digunakan untuk membuat jalur alternatif maupun paralel dalam sebuah aliran proses, semua ekspresi kondisi dievaluasi.</li> <li>• <b>Complex:</b> digunakan untuk model perilaku sinkronisasi kompleks.</li> <li>• <b>Parallel:</b> digunakan untuk menggabungkan aliran paralel dan untuk membuat aliran paralel.</li> </ul>
<b>Data</b>		
<p>Data</p>		<ul style="list-style-type: none"> <li>• <b>Data Object:</b> menunjukkan aliran informasi yang ada pada sebuah proses seperti dokumen bisnis, e-mail, atau surat.</li> <li>• <b>Collection:</b> menunjukkan sekumpulan informasi seperti daftar permintaan item.</li> <li>• <b>Data Input-Output:</b> Data masukan adalah masukan luaran untuk seluruh proses, data keluaran adalah data hasil dari seluruh proses.</li> </ul>
<b>Connecting Object</b>		
<p>Sequence Flow</p>		<p>Menunjukkan urutan <i>Flow Elements</i> dalam sebuah proses atau <i>Choreography</i>. Setiap <i>Sequence Flow</i> hanya memiliki satu sumber dan satu target.</p>
<p>Conditional Flow</p>		<p><i>Sequence flow</i> yang mendefinisikan ekspresi kondisi, aliran akan berjalan hanya jika kondisi bernilai <i>true</i>.</p>

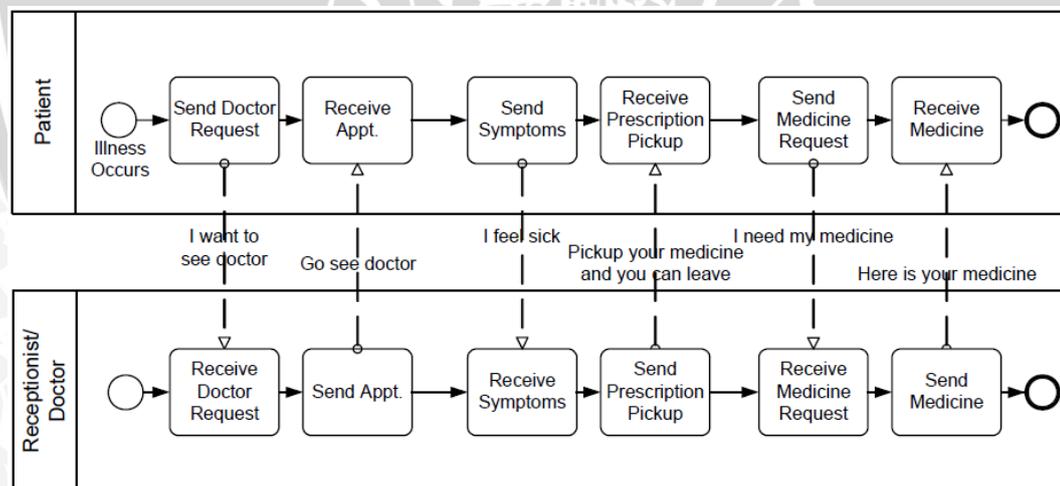
Tabel 2.1 Notasi Elemen BPMN (lanjutan)

Nama	Notasi	Keterangan
Default Flow		Digunakan jika semua kondisi lainnya dinilai salah ( <i>false</i> )
Message Flow		Digunakan untuk menunjukkan aliran pesan antara dua partisipan yang disediakan untuk mengirim dan menerima pesan-pesan tersebut.
Association / Asosiasi		Digunakan untuk menghubungkan informasi dengan sebuah <i>Flow Objects</i> .
<b>Swimlane</b>		
Pool		Representasi grafis dari sebuah partisipan dalam tipe proses bisnis <i>Collaboration</i> , juga bisa dijadikan sebagai "swimlane" dan wadah grafis untuk membagi serangkaian kegiatan dari <i>pool</i> lain
Lane		Sebuah <i>lane</i> adalah sub bagian dalam sebuah proses, terkadang dalam satu <i>pool</i> , dan akan muncul selama proses tersebut berjalan.
Message		Pesan digunakan untuk menggambarkan isi dari sebuah komunikasi antara dua partisipan.
Group		Grup digunakan untuk mengelompokkan elemen grafis yang ada pada kategori yang sama.
Text Annotation		Memberikan informasi teks tambahan untuk pembaca diagram BPMN

Pemodelan proses bisnis digunakan untuk berkomunikasi berbagai informasi untuk berbagai khalayak. BPMN dirancang untuk menutupi berbagai jenis pemodelan dan memungkinkan penciptaan proses bisnis *end-to-end*. Unsur-unsur struktural BPMN memungkinkan pengunjung untuk dapat dengan mudah membedakan antara bagian dari Diagram BPMN.

Ada tiga jenis dasar sub-model dalam model *end-to-end* BPMN :

- Processes (Orchestration)*: dalam tipe ini dibagi lagi menjadi tiga jenis berdasarkan proses bisnisnya yaitu *Private non-executable (internal)*, *private executable (internal)*, dan *public*. Proses bisnis yang bersifat *private* atau pribadi merupakan proses bisnis yang terjadi dalam internal perusahaan itu sendiri. Sedangkan yang bersifat publik merupakan interaksi antara proses bisnis internal dengan proses atau partisipan lain. Hanya kegiatan mereka yang digunakan untuk berkomunikasi dengan partisipan lain yang termasuk dalam proses (Object Management Group (OMG), 2011).
- Choreographies (Koreografi)*: definisi dari perilaku yang diharapkan, pada dasarnya kontrak prosedural, antara partisipan yang berinteraksi. Sementara proses yang normal ada dalam suatu *pool*, koreografi ada di antara *pool* (atau partisipan) (Object Management Group (OMG), 2011).
- Collaboration (Kolaborasi)*: Sebuah kolaborasi menggambarkan interaksi antara dua atau lebih entitas bisnis. Sebuah kolaborasi biasanya berisi dua atau lebih *pools*, yang menggambarkan partisipan pada proses bertipe kolaborasi. Pertukaran pesan antara partisipan ditunjukkan oleh aliran pesan (*message flow*) yang menghubungkan dua *pools* (atau objek dalam *pool*). Pesan yang terkait dengan aliran pesan juga dapat ditampilkan. Kolaborasi dapat ditunjukkan sebagai dua atau lebih proses publik yang berkomunikasi satu sama lain (Object Management Group (OMG), 2011). Contoh dari proses kolaboratif dapat dilihat pada Gambar 2.3.



**Gambar 2.3 Contoh Proses Kolaborasi**

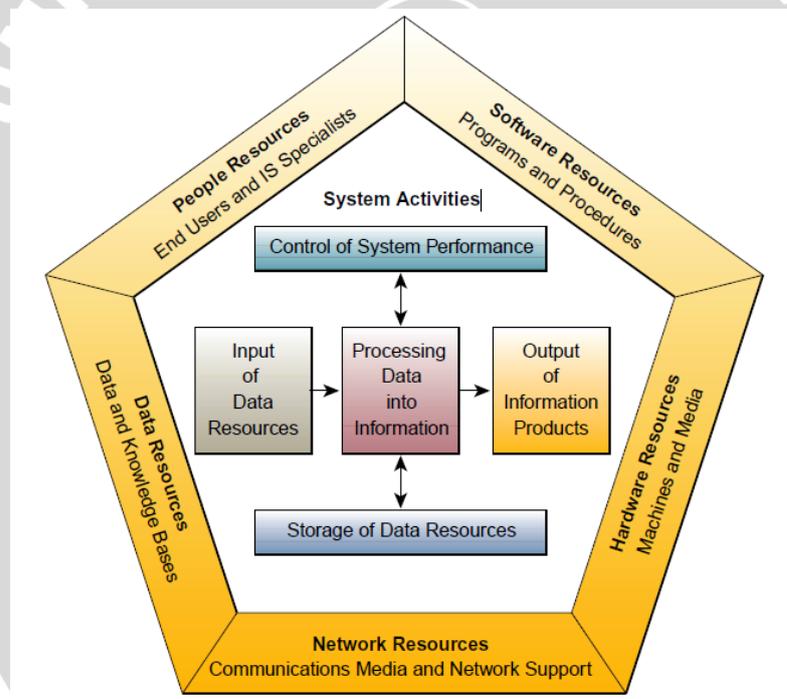
Proses bisnis tipe ini yang nantinya digunakan untuk memodelkan proses bisnis pada DISPENDA Kabupaten Madiun, baik proses bisnis yang saat ini berjalan maupun yang diusulkan.

#### 2.2.4 Sistem Informasi

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-

kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi (Jogiyanto, 2005). Jogiyanto (2005) mengemukakan sistem mempunyai karakteristik atau sifat-sifat tertentu, yakni komponen, batasan sistem, lingkungan luar sistem, dan penghubung sistem. Informasi dapat dipahami sebagai pemrosesan masking yang terorganisir, memiliki arti, dan berguna bagi orang yang menerimanya. Informasi dapat diandalkan jika bebas dari kesalahan atau bias, secara akurat merepresentasikan keadaan atau aktivitas organisasi (Tantra, 2012).

Sebuah sistem informasi merupakan kombinasi yang terorganisir antara orang, *hardware*, *software*, jaringan komunikasi, sumber data dan kebijakan, serta prosedur yang menyimpan, mengambil, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Sistem informasi digunakan oleh orang atau pengguna untuk berkomunikasi satu sama lain menggunakan *hardware* sebagai perangkat fisik, *software* untuk pemrosesan informasi dan prosedur, jaringan sebagai saluran komunikasi, dan data yang tersimpan sebagai sumber data (O'Brien & Marakas, 2010).



**Gambar 2.4 Model Sistem Informasi**

Sumber : O'Brien & Marakas (2010)

Gambar 2.4 merupakan model sistem informasi yang juga mengungkapkan kerangka konseptual mendasar untuk komponen dan aktivitas utama dalam sistem informasi. Semua sistem informasi menggunakan orang, *hardware*, *software*, data, dan jaringan untuk melakukan masukan, proses, keluaran, penyimpanan dan kontrol yang mengubah sumber data menjadi sebuah informasi (O'Brien & Marakas, 2010). Sistem informasi adalah suatu sistem yang menerima

sumber data sebagai input dan memproses data tersebut menjadi hingga menghasilkan output berupa informasi. Untuk mencapai tujuannya sistem informasi melibatkan berbagai komponen dan aktivitas.

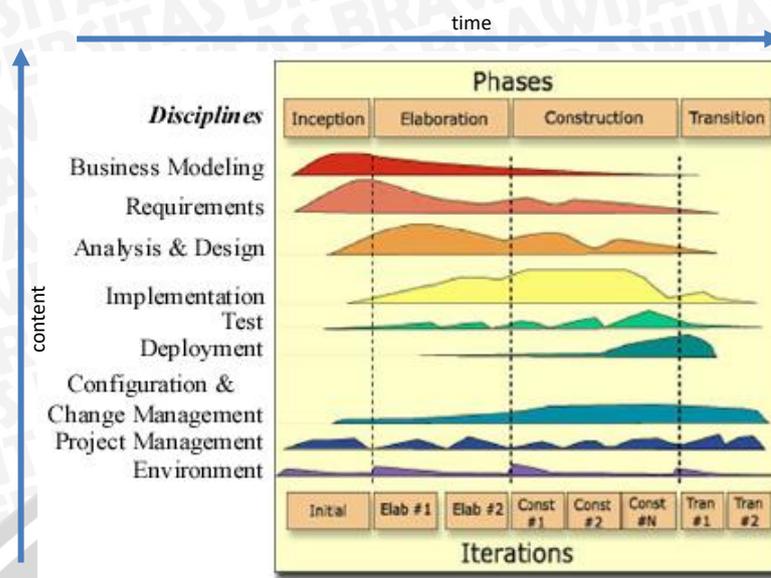
#### 2.2.4.1 Rational Unified Process (RUP)

*Rational Unified Process* (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*). Merupakan proses umum untuk pengembangan perangkat lunak yang mendeskripsikan struktur umum dan proses arsitektur yang umum dari proses rekayasa perangkat lunak yang terkait. Tujuan utamanya adalah untuk memastikan hasil dari perangkat lunak kualitas terbaik memenuhi kebutuhan dari pengguna akhir yang telah disesuaikan dengan jadwal dan biaya yang ditetapkan. RUP menerapkan *best practice* pada pengembangan perangkat lunak modern dalam bentuk yang dapat disesuaikan untuk berbagai proyek dan organisasi (IBM Software Group, 2003).

RUP mendeskripsikan beberapa disiplin pengembangan sistem yaitu pemodelan bisnis, persyaratan, analisis dan perancangan, implementasi, pengujian, penerapan, manajemen perubahan dan konfigurasi, manajemen proyek, dan lingkungan. Selain itu terdapat beberapa fase atau tahapan dalam pelaksanaannya yaitu:

1. *Inception*: fase untuk mendefinisikan cakupan dari proyek, apa yang termasuk didalamnya dan apa yang tidak. Tahapan ini dilakukan dengan mengidentifikasi seluruh aktor dan *use case*.
2. *Elaboration*: Tahapan untuk merencanakan proyek. Pada tahapan ini berfokus pada dua hal : mendapatkan pemahaman yang baik dari persyaratan dan membentuk dasar arsitektur yang bertujuan untuk menghilangkan banyak risiko dan dapat menentukan berapa jumlah pekerjaan yang harus dilakukan. Estimasi biaya dan sumber daya dapat dilakukan pada akhir tahapan ini.
3. *Construction*: Tahapan untuk membangun produk yang ditetapkan, pembangunan produk ini dilakukan pada beberapa iterasi.
4. *Transition*: Tahapan transisi untuk menyerahkan produk ke pengguna akhir. Pada tahapan ini berfokus pada percobaan, instalasi, dan dukungan. (IBM Software Group, 2003)

RUP terorganisir berdasarkan dua hal, yaitu oleh waktu (dengan fase dan iterasi) dan oleh isi (disiplin). Struktur waktu mengacu pada aspek siklus hidup dari proses, bagaimana prosesnya dimulai selama waktu pengembangan. Pada Gambar 2.5 menggambarkan bagaimana fase dan iterasi (dimensi waktu) berhubungan dengan kegiatan pembangunan (dimensi konten). Ukuran relatif dari area warna menunjukkan berapa banyak aktivitas yang berjalan pada setiap fase maupun iterasi. Jumlah relatif dari pekerjaan berkaitan dengan perubahan disiplin antara iterasi.



**Gambar 2.5 Ilustrasi Fase dan Iterasi pada RUP**

Pada penelitian ini, menerapkan fase pada RUP yaitu *Inception* dan *Elaboration* adapun kedua fase tersebut yang sesuai dengan penelitian ini yang menekankan pada analisis dan perancangan sistem yang sesuai dengan apa yang dilakukan pada dua fase RUP tersebut. Namun, pada fase *Elaboration* tidak dilakukan sampai analisis estimasi biaya dan sumber daya. Sedangkan disiplin RUP yang digunakan pada penelitian ini adalah *business modeling*, *requirement*, dan analisis serta perancangan. Selain itu, penelitian ini juga menerapkan *style* pengerjaan yang sama dengan RUP, seperti pada pengerjaan analisis persyaratan, pemodelan *use case*, dan spesifikasi *use case*.

#### 2.2.4.2 Analisis Sistem

Analisis sistem atau *analysis system* merupakan penguraian dari suatu sistem informasi yang utuh ke dalam komponen-komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi berbagai permasalahan, kesempatan, hambatan yang terjadi dan segala kebutuhan yang diinginkan sehingga dapat diusulkan perbaikan-perbaikannya. Tahap analisis sistem dilakukan setelah perencanaan sistem dan sebelum tahapan desain sistem. Tahap analisis sistem ini merupakan tahapan yang penting karena kesalahan dalam tahapan ini dapat menyebabkan juga kesalahan di tahapan selanjutnya (Jogiyanto, 1995).

Menurut Bittner & Spence (2002), banyak tim proyek yang mendalami detail dari model *use case* sebelum menetapkan pemangku kepentingan, tujuan bersama, kebutuhan nyata produk, atau batasan pengembangan sistem. Hal tersebut dapat menyebabkan sebuah masalah, proyek akan selesai sebelum tim menyadari sistem yang dihasilkan tidak sesuai dengan keinginan pemangku kepentingan. Untuk mencegah masalah seperti tersebut, maka sebaiknya dilakukan :

1. Membangun pemahaman yang baik tentang pemangku kepentingan.
2. Memberikan pemahaman tentang masalah yang akan diselesaikan.

3. Menangkap keinginan sebenarnya dari pemangku kepentingan dan fitur sistem yang dibutuhkan untuk memenuhinya.
4. Memastikan bahwa proyek yang dilakukan telah mewakili kemauan dari pemangku kepentingan.

Pada Gambar 2.6 Piramida Persyaratan merupakan sebuah konsep piramida persyaratan yang bertujuan untuk memastikan peran, tujuan, dan konteks dari *use case*. Berdasarkan konsep tersebut, maka tahapan yang dapat dilakukan sebelum membuat pemodelan *use case* adalah :

1. Identifikasi pemangku kepentingan dan pengguna.

Mencakupi tipe dan peran dari pemangku kepentingan dan pengguna. Saat mendefinisikan tipe pemangku kepentingan pastikan untuk memenuhi informasi seperti yang ditunjukkan pada Tabel 2.2.

**Tabel 2.2 Informasi pada Tipe Pemangku Kepentingan**

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
[Tipe dari pemangku kepentingan.]	[Deskripsi singkat dari apa hubungan tipe pengguna tersebut dengan sistem yang dikembangkan.]	[Menyebutkan siapa saja pemangku kepentingan yang dimaksudkan sesuai dengan tipe pemangku kepentingan.]

Sumber : Kurt Bittner & Ian Spence (2002)

Sedangkan untuk tipe pengguna, informasi yang harus dicakup dapat dilihat pada Tabel 2.3.

**Tabel 2.3 Informasi pada Tipe Pengguna**

Tipe Pengguna	Karakteristik	Pengguna
[Tipe dari pengguna.]	[Deskripsi singkat mengenai karakteristik dari tipe pengguna yang berkaitan.]	[Menyebutkan siapa pengguna yang sesuai dengan tipe pengguna yang didefinisikan dan berkaitan dengan sistem yang dikembangkan.]

Sumber : Kurt Bittner & Ian Spence (2002)

Membuat tujuan bersama, pada bagian ini akan melakukan analisis masalah, kebutuhan pemangku kepentingan dan pengguna, mendefinisikan kedudukan produk dan fitur produk. Analisis permasalahan akan disajikan dalam bentuk tabel seperti pada Tabel 2.4.

**Tabel 2.4 Tabel Analisis Permasalahan**

Masalah	[Mendeskripsikan masalah yang dialami.]
Mempengaruhi	[Menyebutkan pemangku kepentingan yang terkena dampak dari masalah yang terjadi.]
Dampak	[Dampak dari masalah yang terjadi.]
Solusi	[Solusi yang diberikan beserta manfaatnya.]

Sumber : Kurt Bittner & Ian Spence (2002)

Pada analisis kebutuhan pemangku kepentingan dan pengguna akan menjelaskan hal apa yang diharapkan untuk ada dalam sistem, namun persyaratan sistem tidak dijelaskan secara spesifik, lebih menjelaskan mengapa persyaratan sistem tersebut dibutuhkan. Pada bagian kedudukan produk, dijelaskan dengan tabel seperti pada Tabel 2.5

**Tabel 2.5 Tabel Kedudukan Produk**

Untuk	[Target pelanggan.]
Siapa	[Pernyataan dari kebutuhan atau kesempatan oleh pelanggan.]
Sebuah	[Menyebutkan nama produk beserta kategori atau penjelasan singkat.]
Yang	[Menyatakan keuntungan atau kelebihan dari produk untuk diterapkan.]
Tidak seperti	[Alternatif kompetitif.]
Produk kita	[Menyatakan perbedaan dengan alternatif lain.]

Sumber: Kurt Bittner & Ian Spence (2002)

Sedangkan pada penjelasan fitur produk, dilengkapi dengan prioritas fitur yang menerapkan aturan *MoSCoW*. *MoSCoW* merupakan singkatan dari : *Must have (Mo)*, *Should have (S)*, *Could have (Co)*, *Won't have (W)*. Prioritas dapat diterapkan dalam persyaratan, tugas, produk, *use case*, pengalaman pengguna, kriteria dan uji penerimaan (*acceptance*). *MoSCoW* merupakan salah satu teknik untuk membantu memahami prioritas. Aturan dari *MoSCoW* adalah (DSDM Consortium, 2008):

- *Must Have*: menyatakan sebuah kebutuhan atau persyaratan yang harus ada agar proyek bisa berjalan. Untuk dapat mengetahui apakah persyaratan tersebut termasuk dalam prioritas ini, bisa menggunakan pertanyaan “Apa yang terjadi jika persyaratan ini tidak terpenuhi?”, jika jawabannya adalah “tidak ada gunanya

menerapkan solusi yang tidak memenuhi persyaratan ini” maka persyaratan tersebut termasuk dalam prioritas *Must Have*.

- *Should Have*: menunjukkan prioritas dari persyaratan yang cukup penting namun tidak vital terhadap proyek yang dikerjakan. Prioritas *Should Have* dapat dibedakan dari *Could Have* dengan meninjau tingkat kegagalan yang disebabkan oleh persyaratan yang dipenuhi, dalam hal nilai bisnis atau jumlah orang yang terdampak.
- *Could Have*: menunjukkan prioritas persyaratan sistem yang ingin atau diinginkan namun tidak terlalu penting. Dampak yang dihasilkan jika persyaratan tidak terpenuhi lebih sedikit dibandingkan dengan prioritas *Should Have*.
- *Won't Have*: menyatakan persyaratan yang telah disepakati oleh tim pengembang untuk tidak dipenuhi. Dituliskan dalam daftar persyaratan untuk membantu memperjelas lingkup proyek. (DSDM Consortium, 2008)

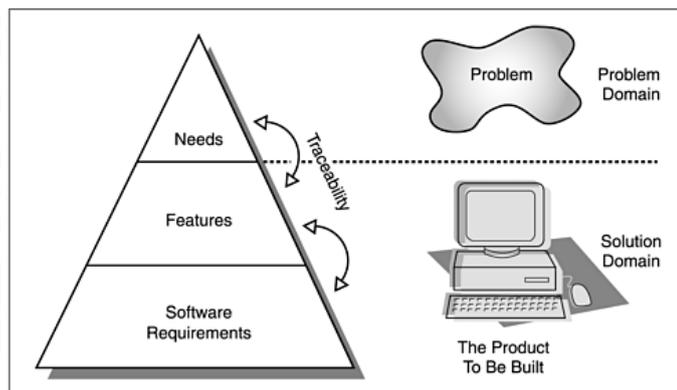
Pada penjelasan fitur produk disarankan untuk memberikan *identifier* atau semacam kode untuk masing-masing fitur yang dideskripsikan. Penjelasan fitur produk dapat disajikan dalam bentuk tabel seperti pada Tabel 2.6.

**Tabel 2.6 Tabel Penjelasan Fitur Produk**

Kode Fitur	Deskripsi	Prioritas
[Kode fitur, fitur yang berbeda memiliki kode yang berbeda.]	[Deskripsi dari fitur produk.]	[Prioritas dari fitur yang disebutkan.]

Sumber : Kurt Bittner & Ian Spence (2002)

2. Identifikasi persyaratan produk. Identifikasi persyaratan produk dimulai dengan analisis persyaratan fungsional dan non fungsional. Setelah persyaratan di analisis maka akan digunakan untuk mengidentifikasi aktor serta *use case* untuk dibuat diagram *use case*. (Bittner & Spence, 2002)



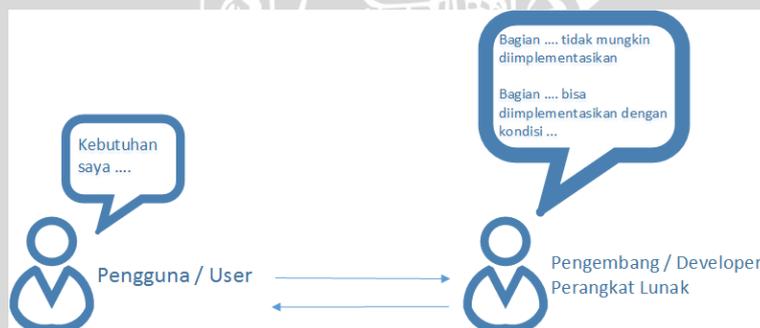
**Gambar 2.6 Piramida Persyaratan**

Sumber : Kurt Bittner & Ian Spence (2002)

### 2.2.4.3 Perancangan Sistem Informasi

#### A. Object Oriented Analysis and Design

Analisis berorientasi objek atau *Object Oriented Analysis* (OOA) adalah tahapan untuk menganalisis spesifikasi atau kebutuhan akan sistem yang akan dibangun dengan konsep berorientasi objek, apakah benar kebutuhan yang ada dapat diimplementasikan menjadi sebuah sistem yang berorientasi objek (A.S. & Shalahuddin, 2013). Dalam melakukan analisis kebutuhan yang berorientasi objek, seharusnya juga memahami tentang implementasi sistem yang berorientasi objek agar sistem yang dihasilkan bisa realistis untuk diimplementasikan dengan berorientasi objek. Ilustrasi dalam tahapan analisis berorientasi objek dapat dilihat pada Gambar 2.7 Ilustrasi OOA.



**Gambar 2.7 Ilustrasi OOA**

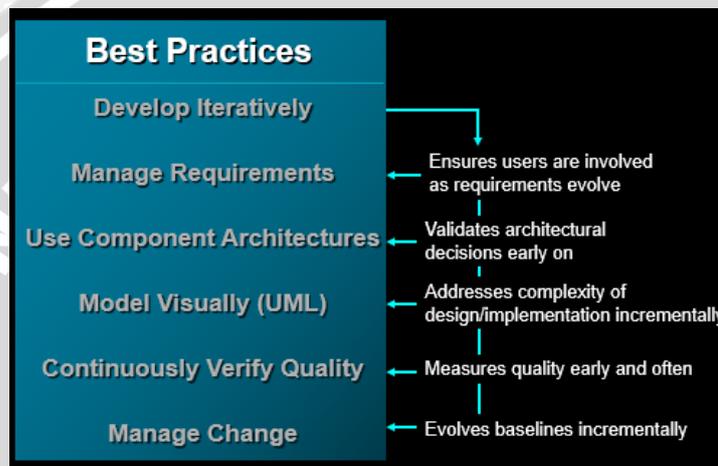
Sumber : Diadaptasi dari A.S. & Shalahuddin (2013)

Perancangan berorientasi objek atau *Object Oriented Design* (OOD) adalah tahapan perantara untuk memetakan spesifikasi atau kebutuhan sistem yang akan dibangun dengan konsep berorientasi objek ke desain pemodelan agar lebih mudah diimplementasikan dengan pemrograman berorientasi objek. Pemodelan berorientasi objek biasanya dituangkan dalam dokumentasi perangkat lunak dengan menggunakan perangkat pemodelan berorientasi objek seperti UML (*Unified Model Language*). Kendala dan permasalahan pembangunan sistem berorientasi objek dapat dikenali dalam

tahap ini. OOA dan OOD dalam proses yang berulang-ulang seringkali memiliki batasan yang samar, sehingga kedua tahapan ini sering juga disebut OOAD (*Object Oriented Analysis and Design*) (A.S. & Shalahuddin, 2013).

## B. IBM Mastering OOAD

*International Business Machine (IBM) Software Group* memiliki panduan khusus untuk melakukan pengembangan sistem dengan menggunakan metode *Object Oriented Analysis and Design (OOAD)*. IBM menggunakan UML 2.0 untuk memvisualisasikan model analisis dan perancangan sistem. Dalam penerapannya, IBM memiliki pendekatan *best practice* dalam pengembangan perangkat lunak yang dapat dilihat pada Gambar 2.8.



Gambar 2.8 IBM Best Practices

Masing-masing bagian dari *best practice* tersebut saling memperkuat dan pada kasus tertentu salah satu bagian dapat mengaktifkan bagian lainnya. Berdasarkan *best practice* tersebut, dalam pengembangan perangkat lunak menggunakan metode OOAD, IBM mengimplementasikan pendekatan *Rational Unified Process* dalam pengembangannya, karena RUP juga menerapkan *best practice* yang sama dan menggunakan metode pengembangan perangkat lunak secara iterasi, selain itu pada disiplin manajemen proyek RUP menjelaskan bagaimana cara untuk mengatur dan mengimplementasikan proyek menggunakan fase dan iterasi, fase tersebut sesuai dengan langkah dalam *best practice* yang digunakan IBM (IBM Software Group, 2004).

Berdasarkan tahapan analisis dan perancangan yang ditunjukkan pada IBM *Software Group* (2004), tahapan yang diterapkan pada penelitian ini adalah :

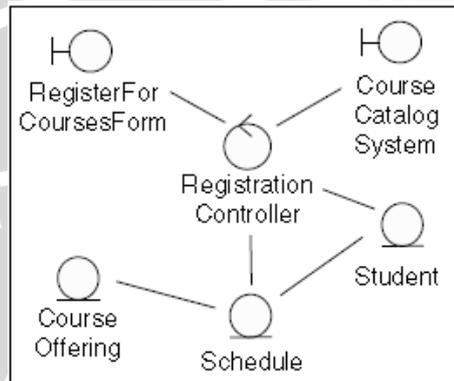
1. Pemodelan *Use Case* dan spesifikasi persyaratan.
  - a. Mendefinisikan persyaratan yang dibutuhkan oleh sistem yang dikembangkan.

- b. Memodelkan *use case* untuk sistem yang merupakan dasar utama untuk keseluruhan proses pengembangan.
2. Analisis Arsitektur: merupakan salah satu tahapan untuk mempertimbangkan pola arsitektur apa yang digunakan dalam pengembangan. Beberapa contoh dari pola arsitektur adalah *Layers*, *Model View Controller (MVC)*, *Pipes and Filters*, dan *Blackboard*. MVC merupakan sebuah pola dimana dalam sebuah aplikasi dibagi menjadi tiga partisi. *Model* merupakan bagian yang menangani aturan bisnis dan data, *View* adalah bagian yang menangani bagaimana informasi ditampilkan pada pengguna, dan *Controller* merupakan bagian yang menangani masukan dari pengguna. Pada penelitian ini menggunakan pola MVC dalam perancangan sistem yang dikembangkan.
3. Mekanisme Arsitektur, pada tahapan ini mencakup:
  - a. Mekanisme Analisis: Menjaring atau menentukan aspek-aspek utama dari sebuah solusi melalui cara implementasi independen. Menyediakan perilaku khusus untuk domain yang berkaitan dengan kelas atau komponen, atau sesuai dengan implementasi dari hubungan antara kelas dan/atau komponen. Beberapa contoh dari mekanisme analisis adalah *persistency*, komunikasi (IPC dan RPC), *message routing*, distribusi, dan *security*, serta banyak contoh lainnya. Masing-masing contoh tersebut memiliki karakteristik tersendiri. Mekanisme analisis digunakan untuk mengurangi kompleksitas dari analisis dan meningkatkan konsistensi. Pada penelitian ini menggunakan mekanisme analisis *persistency* dan *security*.
  - b. Mekanisme Perancangan: asumsi beberapa detail dari lingkungan implementasi, tapi tidak terikat pada implementasi yang lebih spesifik.
  - c. Mekanisme Implementasi: menentukan implementasi yang tepat dari mekanisme, terikat pada teknologi tertentu, bahasa implementasi, vendor atau faktor lain. Pada mekanisme perancangan beberapa teknologi tertentu dipilih seperti RDMS vs ODBMS, sedangkan pada mekanisme implementasi teknologi yang sangat spesifiklah yang dipilih (seperti Oracle vs SYBASE).

Pada penelitian ini, mekanisme arsitektur yang diterapkan adalah mekanisme analisis karena disesuaikan dengan batasan penelitian yang berupa analisis dan perancangan.

4. Analisis *Use Case*, pada tahapan ini yang dilakukan adalah:
  - a. Spesifikasi *Use Case*, merupakan deskripsi dari setiap *use case* yang telah didefinisikan. Berisi tentang alur proses *use case*.
  - b. *Use Case Realization*:
    - Kelas Analisis: realisasi dari *use case* yang menggambarkan model konseptual awal dari sistem. Tingkah laku yang lengkap dari *use case* didistribusikan ke kelas analisis. Ada tiga bagian utama pada kelas analisis yaitu kelas *boundary*, *entity*, dan

*control*. Kelas *boundary* merupakan kelas penengah antara antarmuka sistem dan sesuatu yang ada diluar sistem. Kelas *entity* menunjukkan konsep utama dari sistem yang menunjukkan struktur data logis. Bisa didapatkan dari *glossary* yang ada pada analisis persyaratan. Kelas *control* merupakan kelas yang digunakan untuk memodelkan perilaku kontrol khusus untuk satu atau lebih *use case*. Contoh kelas analisis dapat dilihat pada Gambar 2.9.



**Gambar 2.9 Contoh Analisis Kelas**

Sumber : IBM Software Group (2004)

- Memodelkan interaksi kelas analisis, pemodelan dapat berupa diagram interaksi. Dapat menggunakan diagram *sequence* atau diagram komunikasi.
5. Identifikasi Desain Elemen
    - a. Identifikasi Kelas, ditunjukkan berupa pemodelan diagram kelas.
    - b. Identifikasi *package*, ditunjukkan berupa pemodelan *package*.
  6. Pemodelan Data
    - a. Pemodelan *Relational Model*, memodelkan relasi dari entitas yang telah di analisis. Pada penelitian ini menggunakan pemodelan *Entity Relationship Diagram*.
    - b. Perancangan Tabel, menggambarkan struktur tabel berdasarkan atribut pada kelas.
    - c. Perancangan *Database*, pada penelitian ini dimodelkan dengan membuat *Physical Data Model* yang disesuaikan dengan ERD dan tabel yang telah dirancang.

#### 2.2.4.4 Unified Model Language (UML)

UML adalah bahasa standar yang digunakan dan memvisualisasikan artefak dari proses analisis dan desain berorientasi objek. UML hanya merupakan bahasa pemodelan dan tidak menggambarkan bagaimana melakukan analisis dan desain berorientasi objek. Dalam pemodelan suatu sistem UML menyediakan standar pada notasi dan diagram. Pemodelan UML tidak didominasi oleh narasi, pemodelan dapat dilakukan secara visual. Pemodelan secara visual dapat membantu menangkap struktur dan perilaku dari objek, mempermudah

penggambaran interaksi antar elemen dalam sistem, dan mempertahankan konsistensi antara desain dan implementasi dalam pemrograman (Hermawan, 2015). Menurut Yasin (2013), ada dua tujuan penggunaan UML yaitu:

1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

Secara konseptual, terdapat tiga unsur utama pada UML yaitu :

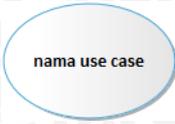
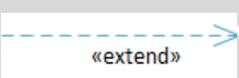
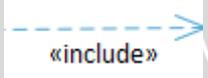
1. *Things*, pada UML terdapat empat jenis hal yaitu *Structural*, *Behavioral*, *Grouping*, dan *Annotation*.
2. *Relationship*, merupakan hubungan antar elemen pada sebuah pemodelan UML. Terdapat empat jenis relasi pada UML yaitu dependensi, asosiasi, generalisasi, dan realisasi.
3. Diagram, merupakan presentasi grafis dari serangkaian elemen yang sering diterjemahkan sebagai graf terhubung simpul (*things*) dan jalur (*relationship*). Terdapat 13 jenis diagram yang termasuk pada UML, diantaranya adalah *Class Diagram*, *Object Diagram*, *Component Diagram*, *Composite Structure Diagram*, *Use Case Diagram*, *Sequence Diagram*, *Communication Diagram*, *State Diagram*, *Activity Diagram*, *Deployment Diagram*, *Package Diagram*, *Timing Diagram*, dan *Interaction Overview Diagram*.

#### A. Use Case Diagram

*Use case* menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu dalam *Use Case Specification*. Namun, *use case* hanya menjelaskan tentang apa saja yang dilakukan oleh aktor dan sistem dan tidak menjelaskan bagaimana aktor dan sistem melakukan kegiatan tersebut. *Use case specification* terdiri dari nama *use case*, deskripsi singkat (*Brief Description*), Aliran Normal (Basic Flow), Aliran Alternatif (Alternate Flow), *Special Requirement*, *Pre-Condition* dan *Post-Condition* (Hermawan, 2015).

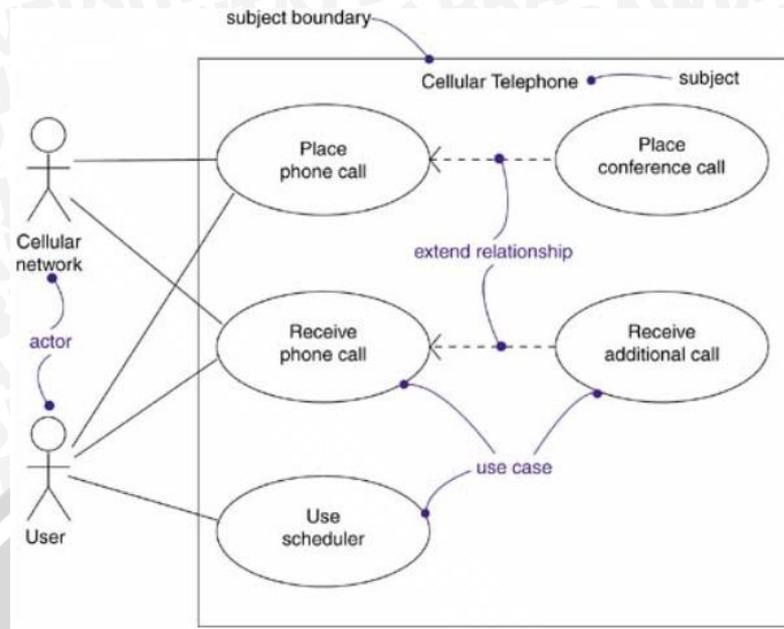
*Use case* adalah unit bagian dari fungsionalitas sistem yang menyediakan *classifier* dan urutan pertukaran pesan oleh subjek dan satu atau lebih aktor dalam unit sistem. Sebuah *use case* melibatkan interaksi aktor dan sistem atau subjek lainnya. (Booch, et al., 2005). Simbol-simbol yang digunakan pada diagram *use case* dapat dilihat pada Tabel 2.7.

Tabel 2.7 Simbol-simbol Diagram *Use Case*

Nama	Simbol	Deskripsi
<i>Use Case</i>		Spesifikasi tingkah laku sebuah entitas terhadap interaksi yang terjadi di luar.
Aktor / <i>actor</i>		Penggambaran pengguna di luar sistem, sistem atau sub sistem yang memiliki peran terhadap sistem. Setiap aktor berpartisipasi terhadap satu atau lebih <i>use case</i> di dalam sistem. Seorang aktor bisa berupa manusia, sistem komputer atau proses eksekusi yang lainnya.
Asosiasi / <i>association</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi / <i>extend</i>		Tingkah laku yang hanya berjalan jika ada kondisi tertentu .
Menggunakan / <i>include</i>		Tingkah laku yang harus dipenuhi supaya <i>event</i> dapat terjadi dan <i>use case</i> tersebut merupakan bagian dari <i>use case</i> lainnya
Generalisasi / <i>generalization</i>		Hubungan antar <i>use case</i> yang umum dan lebih spesifik.

Sumber : Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

Dengan menggunakan UML, diagram *use case* dapat diterapkan untuk menggambarkan perilaku sistem, sub sistem, atau kelas sehingga pengguna dapat memahami bagaimana dapat memahami bagaimana menggunakan elemen tersebut dan juga pengembang dapat mengimplementasikan elemen tersebut. Seperti yang ditunjukkan pada Gambar 2.10, merupakan sebuah contoh penggunaan diagram *use case* untuk memodelkan perilaku kebanyakan orang saat menggunakan telepon seluler.



**Gambar 2.10 Diagram Use Case Perilaku Pengguna menggunakan Telepon Seluler**

Sumber : Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

### B. Diagram Aktivitas

Diagram aktivitas atau *activity diagram* merupakan sebuah diagram yang menggambarkan sebuah aliran aktivitas yang terjadi dalam sistem yang sedang dirancang. Diagram aktivitas merupakan *diagram state* khusus yang menunjukkan sebagian besar *state* adalah aksi dan sebagian besar transisi dipacu oleh selesainya *state* sebelumnya. Oleh karenanya, diagram aktivitas tidak menggambarkan perilaku internal sebuah sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum (Yasin, 2012).

Diagram aktivitas adalah salah satu dari limas diagram di UML untuk memodelkan aspek dinamis dari sistem. Diagram aktivitas pada umumnya merupakan sebuah *flowchart* yang menunjukkan aliran kontrol dari aktivitas ke aktivitas lainnya. Diagram aktivitas menunjukkan *concurrency* serta percabangan (Booch, et al., 2005). Tabel merupakan keterangan dari simbol-simbol yang digunakan pada diagram aktivitas.

Tabel 2.8 Keterangan Simbol-Simbol *Activity Diagram*

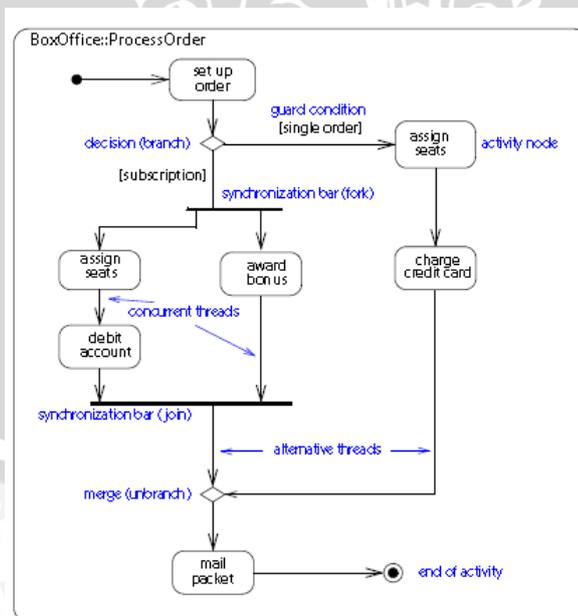
Nama	Simbol	Deskripsi
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Branch/cabang		Sebuah cabang mungkin memiliki satu arus masuk dan dua atau lebih arus keluar dan digambarkan dalam simbol <i>diamond</i> . Pada setiap aliran keluar, ditempatkan ekspresi <i>Boolean</i> , yang dievaluasi saat memasuki cabang. Pada kondisi tersebut cabang berperan sebagai <i>decision</i> . Ketika dua aliran kontrol kembali bergabung bersama-sama, simbol <i>diamond</i> dapat digunakan dengan dua arus masuk dan satu arus keluar. Pada kondisi ini cabang berperan sebagai <i>merge</i> .
Fork-join		<i>Fork</i> merupakan pemisahan alur kontrol tunggal menjadi dua atau lebih alur kontrol bersamaan. Pada <i>fork</i> , mungkin memiliki satu transisi masuk dan dua atau lebih transisi keluar, masing-masing merepresentasikan aliran kontrol yang independen. <i>Join</i> merupakan sinkronisasi dua atau lebih alur kontrol yang bersamaan. Pada <i>join</i> mungkin memiliki dua atau lebih transisi masuk dan satu transisi keluar.

Tabel 2.8 Keterangan Simbol-Symbol *Activity Diagram* (lanjutan)

Nama	Simbol	Deskripsi
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane		Bertujuan untuk partisi aktivitas pada diagram aktivitas dalam kelompok-kelompok, setiap kelompok mewakili organisasi bisnis yang bertanggung jawab.

Sumber : Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

Pada Gambar 2.11 menunjukkan contoh dari diagram aktivitas untuk proses pemesanan pada *Box Office*.



Gambar 2.11 Contoh Diagram Aktivitas Proses Pemesanan pada *Box Office*

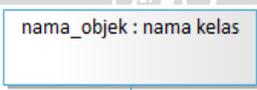
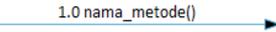
Sumber : James Rumbaugh, Grady Booch, Ivar Jacobson (2005)

### C. Sequence Diagram

*Sequence diagram* atau diagram sekuen menggambarkan interaksi antar objek didalam dan di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) berupa pesan atau *message* yang digambarkan terhadap waktu. Diagram sekuen biasa menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari *event* untuk menghasilkan keluaran tertentu. *Message* yang digambarkan pada diagram sekuen nantinya akan dipetakan menjadi fungsi/metoda dari *class* pada tahapan desain selanjutnya. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan simbol khusus untuk objek *boundary*, *controller* dan *persistent entity* (Yasin, 2012).

Diagram sekuen merupakan diagram interaksi yang menekankan pada saat permintaan pesan yang menunjukkan serangkaian peranan dan pesan yang dikirim dan diterima oleh bagian yang ada pada peranan tersebut. Diagram sekuen digunakan untuk menggambarkan *dynamic view* dari sistem (Booch, et al., 2005). Simbol-simbol yang digunakan pada diagram *sequence* ditunjukkan pada Tabel 2.9.

Tabel 2.9 Simbol-Simbol *Sequence Diagram*

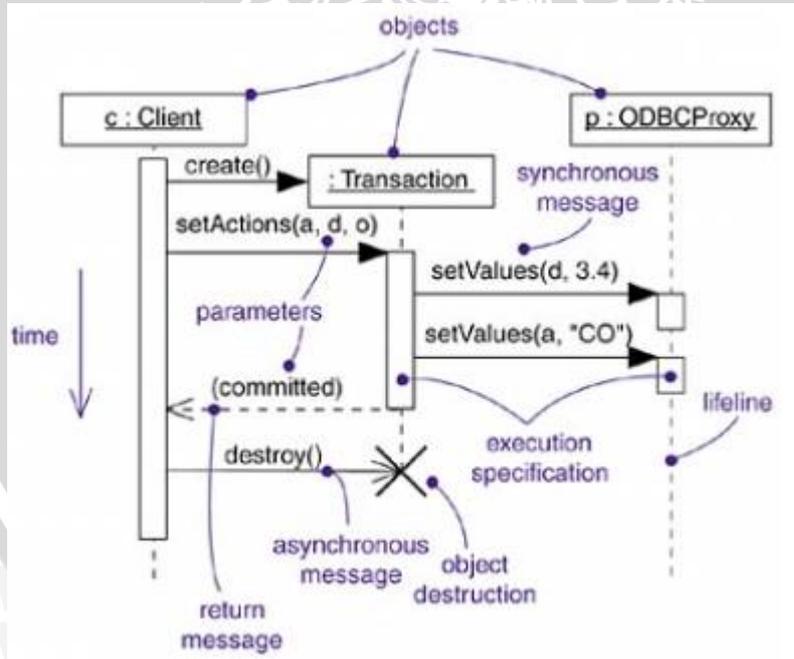
Nama	Simbol	Deskripsi
<i>Lifeline</i>		Menunjukkan keberadaan objek selama periode waktu.
Objek		Menyatakan objek yang berinteraksi menggunakan pesan atau <i>message</i> .
<i>Activation Bar</i>		Menunjukkan periode waktu selama sebuah objek melakukan kegiatan, baik secara langsung maupun melalui prosedur perintah.
Pesan tipe <i>call</i> ( <i>synchronous</i> )		Menyatakan suatu objek memanggil fungsi yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki fungsi, karena ini memanggil sebuah fungsi maka fungsi yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.

Tabel 2.9 Simbol-Simbol *Sequence Diagram* (lanjutan)

Nama	Simbol	Deskripsi
Pesan tipe <i>send</i> ( <i>asynchronous</i> )	1.2 pemberitahuan()	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya;
Pesan tipe <i>return</i>	1.3 :keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu fungsi menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i>	«destroy»	Objek dapat diakhiri selama interaksi yang berlangsung. <i>Lifeline</i> objek berakhir apabila menerima pesan <i>destroy</i> . Bila ada pesan tipe <i>create</i> maka sebaiknya ada pesan tipe <i>destroy</i> .

Sumber : Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

Pada Gambar 2.12 merupakan contoh *sequence diagram* dari proses transaksi dari klien.



Gambar 2.12 *Sequence Diagram* Proses Transaksi dari Klien

Sumber : Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

#### D. Class Diagram

*Class Diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* atau kelas menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (fungsi). Diagram kelas menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Yasin, 2012).

Suatu atribut dan fungsi suatu kelas dapat memiliki salah satu dari sifat berikut ini :

- a. *Private*, tidak dapat dipanggil dari luar kelas yang bersangkutan.
- b. *Protected*, hanya dapat dipanggil oleh kelas yang bersangkutan dan anak-anak yang mewarisinya.
- c. *Public*, dapat dipanggil oleh siapa saja (Yasin, 2012).

Diagram kelas digunakan untuk memodelkan tampilan statis perancangan dari sebuah sistem. Diagram kelas tidak hanya penting untuk memvisualisasikan, menentukan, dan mendokumentasikan model struktural, tapi juga penting untuk menyusun sistem yang dapat dijalankan melalui teknik *forward* maupun *reverse*. Diagram kelas secara umum memuat tiga hal, yaitu kelas, *interfaces*, dan hubungan dependensi, generalisasi, serta asosiasi (Booch, et al., 2005).

- a. Kelas, adalah deskripsi dari sekumpulan objek yang memiliki atribut, operasi, hubungan, dan semantik yang sama. Sebuah kelas mengimplementasikan satu atau lebih *interfaces*.
- b. *Interfaces*, dalam UML *interfaces* digunakan untuk memodelkan lapisan dalam suatu sistem yang merupakan kumpulan dari operasi yang digunakan untuk menentukan sebuah layanan dari kelas atau komponen. Dengan mendeklarasikan antarmuka maka dapat menyatakan perilaku dari independen abstraksi dari sebuah implementasi abstraksi tersebut.
- c. Dependensi, merupakan hubungan yang menyatakan bahwa sebuah hal (kelas) menggunakan informasi dan layanan dari hal lain (kelas lain), tetapi belum tentu sebaliknya. Biasanya menggunakan dependensi antar kelas untuk menunjukkan bahwa satu kelas menggunakan operasi dari kelas lain atau menggunakan variabel yang ada pada kelas lain.
- d. Generalisasi, adalah hubungan antara sebuah hal yang umum (disebut *superclass* atau *parent*) dan jenis hal yang lebih spesifik (disebut *subclass* atau *child*). Objek pada kelas *child* mungkin digunakan untuk variabel atau tipe parameter oleh kelas *parent*, tapi tidak sebaliknya. Hal ini berarti kelas *child* disubstitusikan untuk deklarasi kelas *parent*.
- e. Asosiasi, merupakan hubungan struktural yang menentukan bahwa objek dari satu kelas yang terhubung ke objek kelas lain. (Booch, et al., 2005)

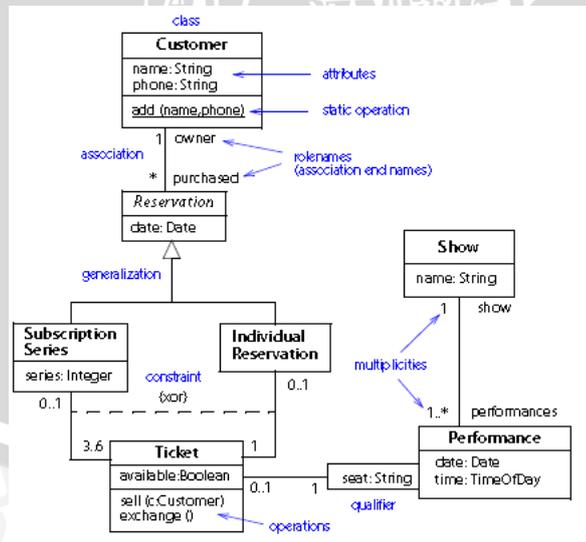
Pada Tabel 2.10 menjelaskan tentang simbol-simbol yang digunakan dalam pembuatan *class diagram* pada umumnya.

**Tabel 2.10 Simbol-Simbol *Class Diagram***

Nama	Simbol	Deskripsi
Kelas		Kelas pada suatu struktur sistem.
Asosiasi / <i>association</i>		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Ketergantungan / <i>dependency</i>		Ketergantungan antar kelas.
Agregasi / <i>aggregation</i>		Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> ).
Asosiasi berarah / <i>directed association</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
Generalisasi		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

Sumber : Grady Booch, James Rumbaugh, Ivar Jacobson (2005)

Contoh dari diagram kelas dapat dilihat pada Gambar 2.13 yang merupakan contoh diagram kelas sebuah aplikasi *Box Office*.



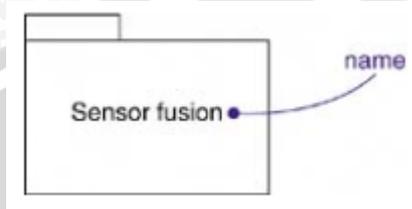
**Gambar 2.13 Diagram Kelas Aplikasi *Box Office***

Sumber : James Rumbaugh, Grady Booch, Ivar Jacobson (2005)



### E. Package

Dalam UML, potongan- potongan yang mengatur suatu model disebut *package* atau paket. Sebuah paket adalah tujuan umum mekanisme untuk mengorganisir elemen dalam kelompok-kelompok. Paket membantu untuk mengatur elemen dalam model sehingga lebih mudah untuk dipahami (Booch, et al., 2005). UML menyediakan representasi grafis dari paket, seperti yang terlihat pada Gambar 2.14.



**Gambar 2.14 Representasi Package**

Sumber : James Rumbaugh, Grady Booch, Ivar Jacobson (2005)

Setiap paket harus memiliki nama yang dapat membedakannya dengan paket lain yang berupa teks. Sebuah paket mungkin memiliki unsur-unsur elemen tersendiri, termasuk kelas, *interfaces*, komponen, *node*, kolaborasi, *use case*, diagram, dan bahkan paket lain

#### 2.2.4.5 Evaluasi

Evaluasi tidak dianggap sebagai tahapan tunggal dalam proses perancangan, tetapi idealnya evaluasi harus terjadi di sepanjang siklus perancangan dengan hasil evaluasi akan digunakan kembali untuk memodifikasi rancangan. Ada hubungan erat antara evaluasi dengan prinsip *prototyping*, yaitu membantu untuk memastikan bahwa perancangan dinilai secara berkelanjutan. Evaluasi memiliki tiga tujuan utama yaitu (Dix, et al., 2004): untuk menilai aksesibilitas dan sejauh mana fungsi sistem berhasil dijalankan, untuk menilai pengalaman interaksi pengguna, dan untuk mengidentifikasi masalah tertentu pada sistem. Fungsionalitas pada sistem itu penting oleh karena itu harus sesuai dengan kebutuhan pengguna. Dengan kata lain, perancangan sistem harus memungkinkan pengguna untuk melakukan tugas-tugas mereka dengan lebih mudah. Hal ini termasuk untuk tidak sekedar membuat fungsi yang tepat pada sistem, tetapi juga membuatnya lebih jelas untuk dijangkau oleh pengguna. Hal ini juga melibatkan kecocokan antara kegunaan sistem dan harapan pengguna (Dix, et al., 2004).

#### A. Peninjauan Persyaratan (*Requirement review*)

Peninjauan (*review*) merupakan sebuah proses yang dilakukan untuk mendeteksi kesalahan, yang dibandingkan dengan proses pengujian komponen yang lebih privat. *Requirement review* atau peninjauan persyaratan merupakan proses manual yang melibatkan orang-orang dari pelanggan dan pihak pengembang. Peninjauan persyaratan bisa dilakukan secara formal maupun informal. Peninjauan informal hanya melibatkan

pengembang yang mendiskusikan persyaratan dengan sebanyak mungkin pemangku kepentingan. Dalam peninjauan secara formal, pengembang harus “mengantar” para pelanggan melalui persyaratan sistem menjelaskan dampak dari tiap persyaratan. Tim peninjau harus memeriksa setiap persyaratan untuk konsistensi (*consistency*) dan memeriksa keseluruhan persyaratan untuk kelengkapan (*completeness*) (Sommerville, 2009).

## B. Verifikasi dan Validasi

Dalam penyusunan seluruh kebutuhan sebuah sistem atau perangkat lunak, diperlukan sebuah tahapan untuk memastikan bahwa kebutuhan perangkat lunak yang disusun telah memenuhi spesifikasi perangkat lunak yang akan dikembangkan. Siahaan (2012) menjelaskan bahwa, banyak pengembang perangkat lunak yang mengalami frustrasi dikarenakan kesulitan untuk mengimplementasikan kebutuhan pengguna yang rancu dan tidak menyeluruh. Bahkan, para pengembang perangkat lunak akan membuat interpretasi sendiri jika mereka tidak mendapatkan informasi yang dibutuhkan mengenai kejelasan perangkat lunak yang dikembangkan. Oleh karenanya, diperlukan sebuah tahapan verifikasi dan validasi terhadap kebutuhan dalam penentuan spesifikasi perangkat lunak.

Validasi perangkat lunak atau secara umum, verifikasi dan validasi (V&V) bertujuan untuk menunjukkan bahwa sistem sesuai dengan spesifikasi sistem dan sistem tersebut sesuai dengan harapan pengguna sistem. Verifikasi dan validasi mungkin juga melibatkan proses pengecekan, seperti inspeksi atau *review*, pada setiap tahapan dari proses pengembangan perangkat lunak, mulai dari definisi persyaratan sampai pengembangan sistem (Sommerville, 2009).

Menurut Sommerville (2009), tujuan dari validasi adalah untuk memastikan bahwa perangkat lunak memenuhi harapan pelanggan. Hal ini lebih dari hanya memeriksa kesesuaian dengan spesifikasi untuk menunjukkan bahwa perangkat lunak telah melakukan apa yang pelanggan harapkan bisa dilakukan oleh perangkat lunak tersebut. Proses pada validasi merupakan proses yang lebih umum daripada verifikasi. Tujuan dari verifikasi adalah untuk memeriksa bahwa perangkat lunak memenuhi persyaratan yang telah ditetapkan. Salah satu teknik dalam validasi adalah *prototyping*, pendekatan validasi ini merupakan model eksekusi dari sistem yang bersangkutan yang ditunjukkan pada pengguna akhir dan pelanggan. Mereka dapat melakukan percobaan dengan model ini untuk melihat apakah sistem memenuhi kebutuhan atau harapan mereka.

Verifikasi merupakan pendekatan untuk mengevaluasi apakah sebuah sistem telah memenuhi persyaratan atau kondisi yang ditentukan pada sistem sesuai dengan standar atau prosesnya. Verifikasi disebut juga “teknik statis” karena tidak melibatkan eksekusi dari baris kode atau hasil sistem. Kelebihan dari verifikasi diantaranya adalah (Limaye, 2009):

1. Verifikasi dapat mengkonfirmasi apakah cara kerja sebuah produk sudah mengikuti proses dengan benar seperti yang telah didefinisikan oleh pelanggan.
2. Dapat menemukan kerusakan atau kesalahan (dari standar yang telah ditetapkan) dengan mudah beserta lokasi kesalahan.
3. Mengurangi biaya dari pencarian dan perbaikan kesalahan sebagaimana hasil produk telah ditinjau dan dikoreksi dengan cepat.
4. Dapat digunakan dengan efektif untuk pelatihan orang tentang proses dan standar. Secara khusus *peer review*, *superior review*, dan *walkthrough* bisa dijadikan alat yang baik untuk pelatihan.

Ada beberapa metode yang tersedia untuk verifikasi sistem salah satunya adalah metode *walkthrough*. *Walkthrough* merupakan review yang bertipe semi formal karena melibatkan tim dengan reviewer untuk meninjau sistem. Dikatakan semi formal karena hanya orang yang berhubungan dengan produk yang dihasilkan yang akan terlibat dalam kegiatan *walkthrough*. Kelebihan dari metode *walkthrough* diantaranya adalah (Limaye, 2009):

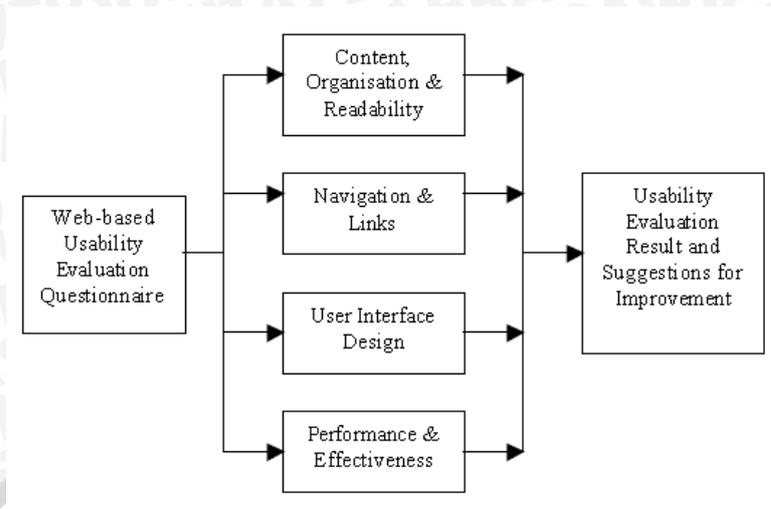
1. Merupakan alat yang baik untuk kolaborasi dimana keputusan bersama dapat dibuat oleh tim.
2. Berguna untuk pelatihan seluruh anggota tim pada satu waktu. Uji skenario dan perancangan biasanya digunakan dalam metode ini.
3. Orang akan mengerti apa yang mereka harapkan dari dokumen dibawah pertimbangannya.
4. Masalah dapat terekam dan saran dapat diterima dari tim untuk meningkatkan hasil produk kedepannya. (Limaye, 2009)

#### 2.2.4.6 WEBUSE: *Website Usability Evaluation Tool*

Melakukan evaluasi terhadap sebuah situs web itu penting. Namun, masalah dalam mendapatkan hasil *usability* digunakan lebih dalam pada pengembangan pada dasarnya, karena kurangnya kegunaan pada metode dan hasil evaluasi *usability*. Ketepatan metode evaluasi *usability* sendiri akan menentukan keakuratan hasil evaluasi (Chiew & Salim, 2003). Berdasarkan penelitian yang dilakukan oleh Chiew dan Salim (2003), setelah melakukan analisis terhadap beberapa *tools* untuk evaluasi *usability* yaitu WAMMI, *WebSAT*, *Bobby*, dan *Protocol Analysis*, didapatkan empat kategori dalam menentukan *usability* yaitu :

- a. Konten, organisasi, dan keterbacaan (*readability*),
- b. Navigasi dan *link*,
- c. Desain antarmuka pengguna,
- d. Performa dan keefektifan.

Model pengembangan evaluasi pada WEBUSE dapat dilihat pada Gambar 2.15.



**Gambar 2.15 Model Pengembangan WEBUSE**

Contoh pertanyaan kuesioner pada WEBUSE pada kriteria navigasi dan *link*, adalah :

- a. Saya dapat mengetahui dengan mudah dimana Saya pada situs web ini.
- b. Situs web ini memberikan pertanda dan *link* yang bermanfaat untuk Saya mendapatkan informasi yang saya inginkan.
- c. Mudah untuk menelusuri situs web ini dengan menggunakan *link* atau tombol kembali pada peramban.
- d. *Link* pada situs web ini dirawat dan di perbaharui dengan baik.
- e. Situs web ini tidak membuka banyak jendela peramban ketika Saya menggunakannya.
- f. Penempatan *link* atau menu standar atau sama di seluruh situs web dan Saya dapat dengan mudah mengenalinya. (Chiew & Salim, 2003)

Contoh pertanyaan kuesioner pada kriteria desain antarmuka pengguna adalah :

- a. Situs web memiliki antarmuka yang konsisten.
- b. Rancangan situs web mudah untuk digunakan.

Pada penelitian analisis dan perancangan sistem pelaporan pembayaran pajak ini, akan menggunakan dua kriteria WEBUSE untuk mengevaluasi perancangan sistem yang dikembangkan. Karena pada penelitian ini terbatas pada analisis dan perancangan dan kriteria yang sesuai untuk diterapkan adalah kriteria desain antarmuka pengguna dan kriteria navigasi dan *link*.

#### **2.2.4.7 Traceability**

*Traceability* adalah sejauh mana hubungan dapat dibentuk antara dua atau lebih produk dari proses pengembangan, khususnya hubungan satu sama lain antara produk yang memiliki pendahulu-penerus atau master-subordinat, dengan

kata lain, sejauh mana persyaratan dan perancangan yang ada pada perangkat lunak memiliki kecocokan (Institute of Electrical and Electronics Engineers (IEEE), 2006). Gotel dan Finkelstein dalam Institute of Electrical and Electronics Engineers (2008) menyebutkan bahwa sebuah verifikasi dan kontrol utama dari proses pengembangan sistem adalah *traceability* dari persyaratan. *Tracing* atau penelusuran ini terjadi dalam dua direksi yaitu *backwards* untuk kebutuhan pengguna didefinisikan pada dokumen konsep operasi dan *forward* untuk spesifikasi, perancangan, implementasi dan pengujian. *Traceability* dari persyaratan merujuk pada kemampuan untuk mengikuti kehidupan persyaratan dari asal-usulnya melalui pengembangan dan spesifikasi untuk penggunaan dan penyebaran selanjutnya. *Traceability* juga merupakan alat utama untuk memastikan bahwa sistem yang dikembangkan sepenuhnya telah memenuhi persyaratan dan kebutuhan yang didefinisikan oleh pengguna (Institute of Electrical and Electronics Engineers (IEEE), 2008).

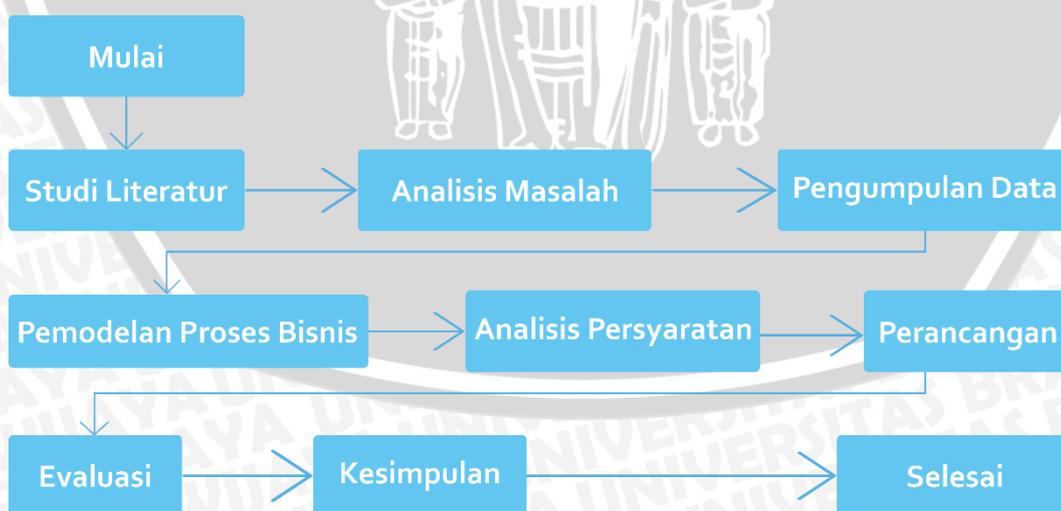


## BAB 3 METODOLOGI

Penelitian ini merupakan penelitian yang bertipe implementatif perancangan. Pada bab ini peneliti akan mencoba menguraikan mengenai langkah-langkah yang akan dilakukan untuk memecahkan masalah yang dibahas. Tahap pertama yang dilakukan oleh peneliti adalah mencari studi literatur, dengan mencari penelitian sebelumnya terkait analisis dan perancangan sistem serta evaluasi. Tahapan kedua adalah tahap analisis masalah yang bertujuan untuk menganalisa hasil wawancara untuk menentukan masalah apa yang bisa untuk dijadikan topik penelitian dan studi pustaka berdasarkan topik penelitian yang telah didapatkan.

Langkah selanjutnya adalah pengumpulan data, peneliti mengumpulkan data yang diperoleh dari wawancara dari calon pengguna aplikasi yaitu pihak DISPENDA dan pengguna Wajib Pajak. Langkah selanjutnya adalah melakukan analisis persyaratan dan perancangan. Persyaratan diperoleh dari data yang sudah didapatkan dan perancangan disesuaikan dengan kebutuhan pengguna. Setelah dihasilkan rancangan sistem yang sesuai dengan kebutuhan, kemudian dilakukan evaluasi. Setelah evaluasi selesai dilakukan, kemudian dapat dilakukan penarikan kesimpulan.

Pada penelitian ini mengacu kepada fungsionalitas dan prototipe sistem yang merupakan desain tampilan antarmuka sistem dan kebenaran dari sistem yang dibuat. Penggalan kebutuhan pengguna dilakukan dengan wawancara secara langsung. Pada bagian perancangan, akan dibuat sebuah desain antarmuka yang nantinya akan dievaluasi untuk mengidentifikasi apakah sistem yang dibuat sudah sesuai dengan kebutuhan pengguna. Dari penjelasan tersebut, Gambar 3.1 adalah rancangan *flowchart* yang menjelaskan tentang langkah-langkah penelitian yang akan dilakukan.



Gambar 3.1 Metodologi Penelitian

### 3.1 Studi Literatur

Tahapan ini merupakan tahap mencari, memilih, mempelajari, dan menyusun teori-teori dasar dan referensi yang mendukung dan sesuai dengan penelitian yang dilakukan. Pada tahap studi literatur ini akan dilakukan beberapa aktivitas diantaranya melakukan mengumpulkan informasi di Dinas Pendapatan Kabupaten Madiun dan melakukan pencarian dan pengumpulan literatur yang menunjang pengerjaan skripsi baik secara *online* maupun *offline*. Literatur yang dicari dapat berasal dari buku-buku, jurnal atau *paper*, atau penelitian yang sebelumnya telah dilakukan oleh peneliti lain (kajian pustaka).

Kajian pustaka yang dikaji pada penelitian ini merupakan penelitian sebelumnya yang berkaitan dengan penelitian yang dilakukan, yaitu penelitian tentang analisis dan perancangan berorientasi objek dan evaluasi perancangan sistem. Sedangkan teori-teori yang dijelaskan pada penelitian ini diantaranya adalah mengenai Dinas Pendapatan Kabupaten Madiun beserta pajak, proses bisnis dan pemodelan proses bisnis menggunakan BPMN, sistem informasi beserta analisis dan perancangan sistem berorientasi objek, bahasa pemodelan UML (seperti *use case*, *activity diagram*, *sequence diagram* dan *class diagram*), evaluasi analisis dan perancangan sistem.

### 3.2 Analisis Masalah

Setelah menemukan studi literatur yang sesuai, kemudian peneliti melakukan analisis masalah yang ada pada DISPENDA Kabupaten Madiun. Tahapan ini dilakukan dengan melakukan wawancara dengan pihak DISPENDA Kabupaten Madiun, yaitu dengan divisi perencanaan, pembukuan, dan penagihan. Dari hasil wawancara tersebut, peneliti mengumpulkan semua informasi dan masalah yang didapatkan. Kemudian peneliti menganalisa manakah masalah yang bisa dijadikan sebagai topik penelitian dan bisa diselesaikan dengan melakukan penelitian tersebut dan sesuai dengan metode serta studi literatur yang digunakan. Sedangkan studi pustaka dilakukan untuk memahami lebih lanjut mengenai analisis dan perancangan sistem menggunakan metode *Object Oriented Analysis and Design*. Teori-teori pendukung tersebut diperoleh melalui buku, jurnal, buku elektronik. Adapun teori-teori yang berkaitan dengan penelitian ini adalah :

1. Pengertian Pajak Daerah
  - Wajib Pajak
  - SKPD dan SSPD
2. DISPENDA Kabupaten Madiun
  - Visi dan Misi
  - Struktur Organisasi
  - Sejarah
3. Proses Bisnis

4. Sistem Informasi
  - Definisi Sistem Informasi
  - Analisis Sistem Informasi
5. Perancangan Sistem Informasi
  - *Unified Modelling Language*
  - *Object Oriented Analysis and Design*
6. Evaluasi

### 3.3 Pengumpulan Data

Teknik pengumpulan data dilakukan melalui tiga teknik, yaitu :

#### 1. Wawancara

Wawancara pada tahap ini sama seperti tahap wawancara diawal, yaitu dengan wawancara langsung dengan pihak berkaitan. Pihak yang berkaitan yaitu DISPENDA Kabupaten Madiun dan Wajib Pajak. Wawancara merupakan langkah untuk mendapatkan informasi yang nantinya akan dijadikan sebagai kebutuhan sistem. Wawancara ini juga bertujuan untuk memperjelas proses bisnis perusahaan dan sistem Portal Pelaporan Pembayaran Pajak. Pengumpulan data dengan wawancara dilakukan untuk meyakinkan bahwa data yang diperoleh adalah data yang akurat. Subjek yang diwawancarai adalah divisi perencanaan dan penagihan.

#### 2. Observasi

Teknik ini merupakan teknik yang digunakan oleh peneliti untuk melakukan pengamatan di lingkungan DISPENDA Kabupaten Madiun, pada lingkungan tersebut terdapat informasi-informasi yang dibutuhkan untuk melakukan penelitian ini. Teknik ini bisa jadi tidak melibatkan pengguna secara langsung.

#### 3. Studi Dokumen

Studi dokumen merupakan teknik yang dipakai peneliti untuk mempelajari dokumen-dokumen yang berkaitan atau digunakan dalam penelitian ini. Dokumen akan didapatkan setelah mendapatkan izin dari pihak DISPENDA Kabupaten Madiun terlebih dahulu.

### 3.4 Pemodelan Proses Bisnis

Berdasarkan hasil wawancara pada tahapan sebelumnya, hal yang kemudian dilakukan adalah melakukan pemodelan proses bisnis. Pemodelan proses bisnis yang dilakukan menggunakan standar notasi *Business Process Model and Notation* (BPMN). Proses bisnis yang dimodelkan pertama kali adalah proses bisnis yang saat ini telah terjadi pada DISPENDA Kabupaten Madiun mengenai proses pembayaran pajak. Setelah memodelkan proses bisnis yang digunakan saat ini, kemudian yang dilakukan adalah memodelkan proses bisnis usulan yang

disesuaikan dengan *prototype* dari perancangan sistem yang akan dibuat. Proses bisnis usulan ini disesuaikan dengan masalah dan solusi yang didapatkan yang kemudian disesuaikan dengan *prototype* yang akan dibuat.

### 3.5 Analisis Persyaratan

Tahapan selanjutnya yang dilakukan adalah melakukan analisis persyaratan, dalam melakukan analisis persyaratan ini menggunakan model RUP yang telah dibukukan oleh Kurt Bittner dan Ian Spence (Bittner & Spence, 2002). Langkah-langkah yang dikerjakan adalah :

1. Melakukan analisis permasalahan berdasarkan dari masalah-masalah yang terdapat di DISPENDA Kabupaten Madiun dan menyertakan solusi yang akan dikembangkan.
2. Melakukan analisis pemangku kepentingan dan pengguna yang mencakup analisis tipe dan peran pemangku kepentingan serta peran pengguna.
3. Menganalisa kebutuhan pengguna dan pemangku kepentingan yang bertujuan untuk menjelaskan persyaratan sistem apa yang dibutuhkan oleh pengguna maupun pemangku kepentingan.
4. Menjelaskan kedudukan dari produk (sistem) yang akan dirancang.
5. Menganalisa apa saja fitur-fitur yang tersedia dalam sistem yang akan dirancang. Analisis fitur ini dilengkapi dengan prioritas pada masing-masing fitur menggunakan aturan MoSCoW. Pada analisis fitur sekaligus menyebutkan kebutuhan fungsional dan non-fungsional dari sistem.
6. Langkah selanjutnya adalah pemodelan *use case*, metode pemodelan mengikuti model RUP dan pemodelan ini meliputi analisis aktor, diagram *use case*, dan spesifikasi *use case*.

### 3.6 Perancangan

Setelah semua persyaratan sistem sudah di analisis dalam tahapan analisis persyaratan, maka selanjutnya melakukan tahapan desain atau perancangan. Tahap desain berorientasi objek menitikberatkan pada mendefinisikan objek perangkat lunak dan bagaimana mereka bekerja sama untuk memenuhi kebutuhan. Solusi perancangan yang dihasilkan berupa *prototype* sistem yang disesuaikan dengan persyaratannya, adapun dalam pembuatannya menggunakan *framework Code Igniter* (CI). Langkah yang dilakukan pada tahapan ini sesuai pada langkah-langkah yang telah ada pada landasan teori.

### 3.7 Evaluasi

Dalam tahapan ini dilakukan sebuah evaluasi terhadap hasil-hasil penelitian yang dilakukan. Adapun evaluasi yang dilakukan adalah :

1. Peninjauan *prototype* terhadap *use case* dan perancangan dasar antarmuka pengguna. Evaluasi ini terbagi mencakup dua aktivitas berikut:

- a. Peninjauan *use case* (*use case review*) bertujuan untuk mengetahui kesesuaian dari alur yang dilalui oleh pengguna dengan alur yang terspesifikasi pada *use case* (alur dasar). Pengguna akan diberikan beberapa *use case* yang dijadikan kasus uji untuk dilakukan pengguna menggunakan prototipe yang sudah ada. Kemudian akan diamati dan dicatat langkah yang dilakukan oleh pengguna tersebut. Setelah itu, pengguna akan diberikan pertanyaan-pertanyaan mengenai alur *use case* yang diujikan dan meminta saran dari pengguna terhadap alur tersebut.
  - b. Pengumpulan umpan balik terhadap desain dasar antarmuka pengguna yang meliputi nilai *usability* parsial dan saran dari pengguna. Aktivitas ini menggunakan instrumen WEBUSE untuk mengevaluasi kriteria navigasi dan antarmuka dari prototipe yang telah dibuat.
2. Peninjauan keruntutan (*traceability*) bertujuan untuk mengetahui keruntutan dari perancangan dasar yang dibuat dengan persyaratan yang telah dianalisis. Peninjauan ini terbagi menjadi dua aktivitas yaitu :
- a. Analisis matriks keruntutan yang menghubungkan antara proses bisnis, kebutuhan fungsional, *use case*, *activity diagram*, *sequence diagram*, dan rancangan antarmuka. Aktivitas ini dilakukan oleh peneliti dengan tujuan untuk mengetahui apakah perancangan awal yang dibuat dapat ditelusuri sampai ke persyaratan dan kebutuhan yang telah disusun.
  - b. Pengecekan persyaratan dalam aspek keruntutan melalui daftar pengecekan persyaratan (*Requirement Checklist*). Pengecekan ini dilakukan oleh peneliti dengan menjawab pertanyaan mengenai persyaratan yang sudah dianalisis. Pengecekan persyaratan yang dilakukan menggunakan aspek *Traceability* Daftar pengecekan persyaratan yang dibuat mengikuti daftar yang telah dibuat oleh Karl E. Wiegers (2001).

### 3.8 Kesimpulan

Kesimpulan dilakukan setelah semua tahapan sudah dilakukan. Pada akhir penulisan terdapat saran yang bertujuan untuk memperbaiki kesalahan dan penyempurnaan terhadap penelitian ini.

## BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN

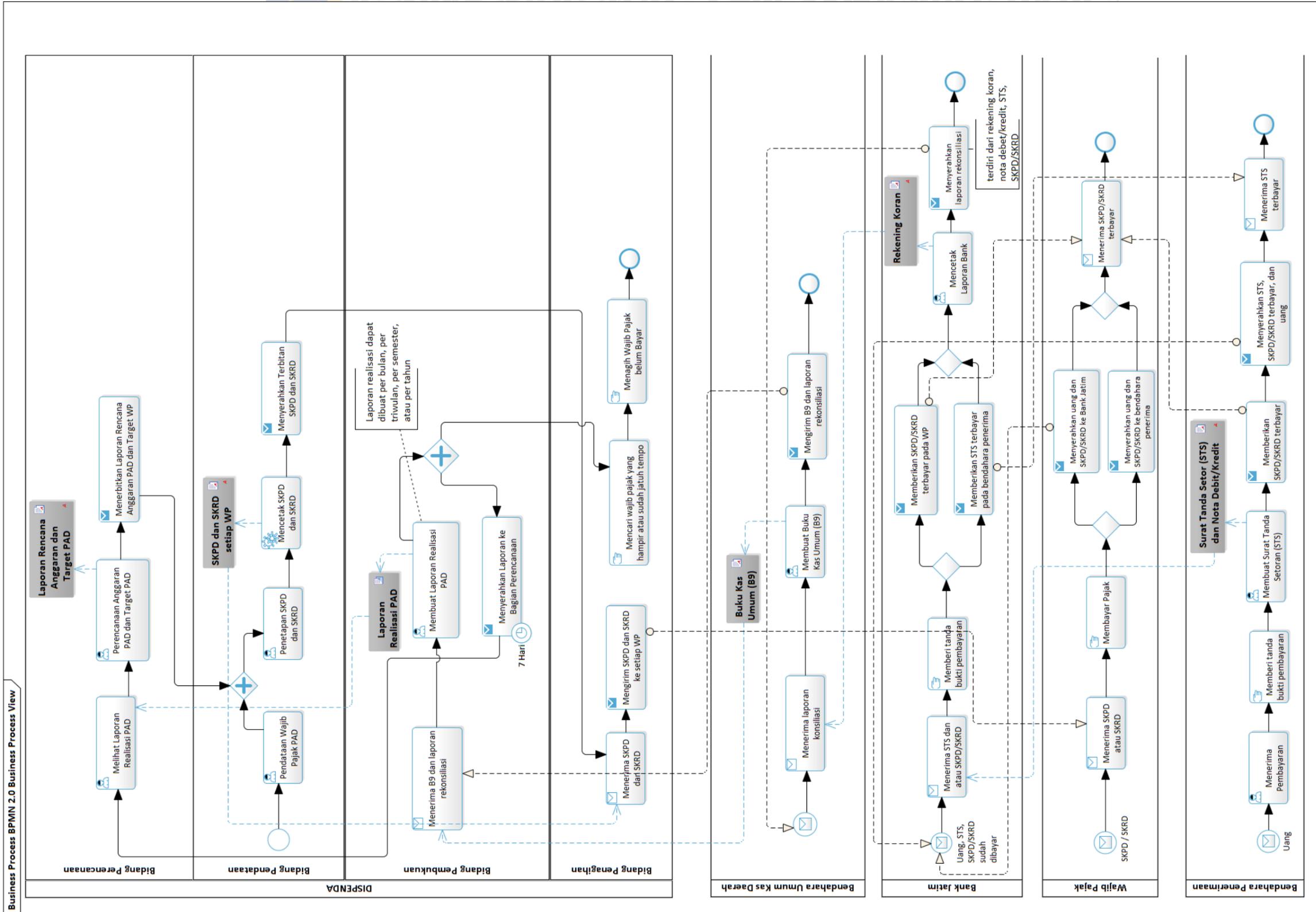
### 4.1 Pemodelan Proses Bisnis

#### 4.1.1 Proses Bisnis Penerimaan PAD DISPENDA Kabupaten Madiun

Proses bisnis saat ini yang terjadi dalam kegiatan penerimaan PAD yang didapatkan dari pembayaran oleh wajib pajak pada DISPENDA Kabupaten Madiun adalah dimulai dari penentuan jumlah pembayaran pajak oleh bagian penetapan, yang kemudian dilanjutkan dengan mencetak Surat Ketetapan Pajak Daerah (SKPD) masing-masing wajib pajak. Setelah itu SKPD tersebut didistribusikan pada setiap wajib pajak yang ada, yang bertugas dalam pendistribusian SKPD tersebut adalah bidang penagihan. Setelah menerima SKPD, wajib pajak bertanggung jawab untuk segera membayarkan pajak sesuai nominal yang sudah ditetapkan. Untuk melakukan pembayaran, wajib pajak harus membawa SKPD dan pembayaran bisa dilakukan melalui bendahara penerima atau melalui Bank Jatim. Saat ini DISPENDA Kabupaten Madiun telah bekerja sama dengan Bank Jatim untuk pelayanan pembayaran pajak daerah. Jika pembayaran dilakukan melalui bendahara penerima, maka bendahara penerima kemudian harus menyerahkan uang pembayaran pajak ke Bank Jatim untuk kemudian dimasukkan ke rekening DISPENDA Kabupaten Madiun.

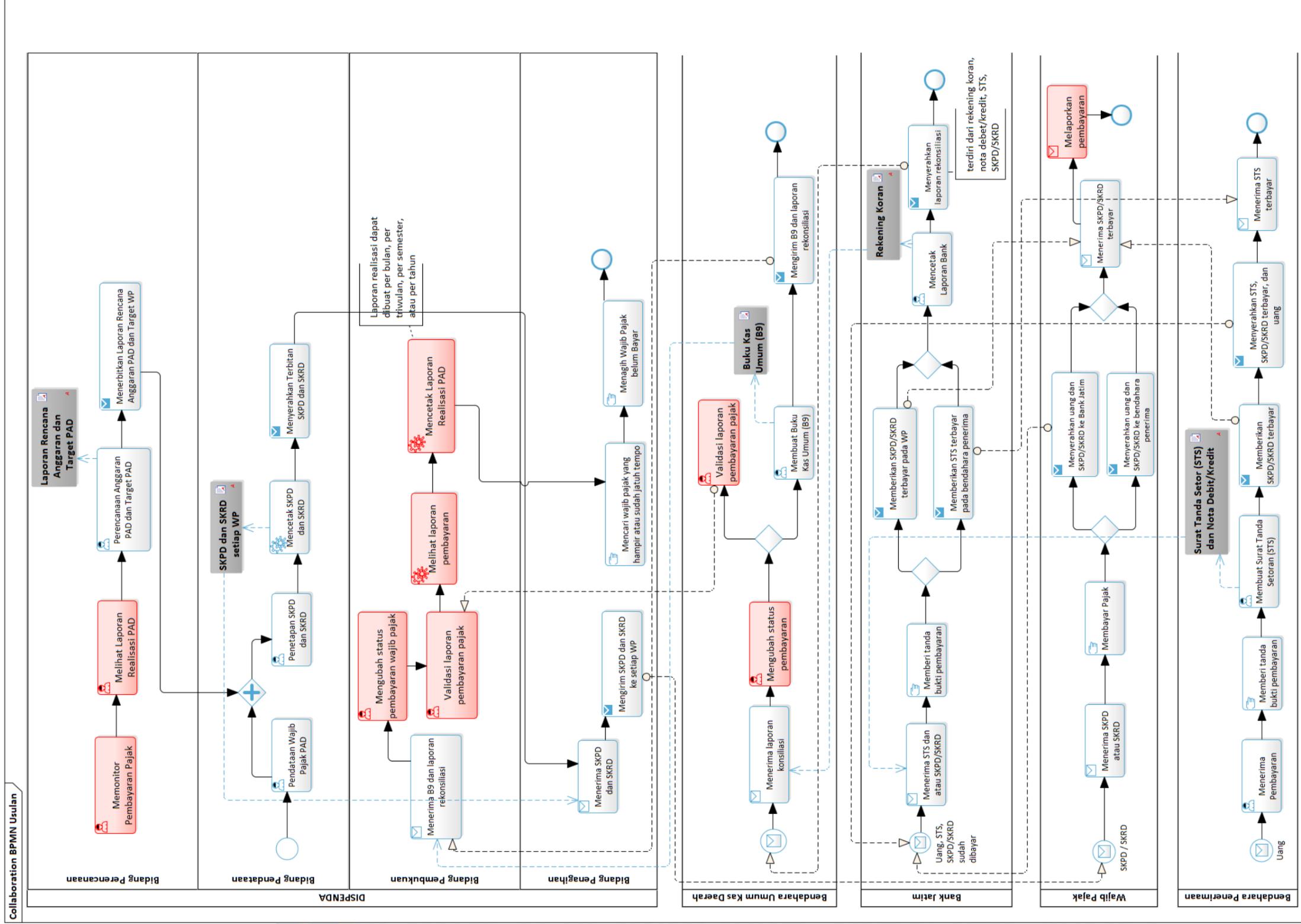
Jika pembayaran langsung ke Bank Jatim, maka uang pembayaran pajak akan langsung dimasukkan ke rekening DISPENDA Kabupaten Madiun. Setelah memasukkan uang ke rekening, Bank Jatim akan mencetak rekening koran dengan Surat Setoran Pajak Daerah (SSPD) yang merupakan rekap pembayaran yang terjadi kepada Bendahara Umum Daerah (BUD)/ Kas Daerah. Setiap hari Bank Jatim akan memberikan laporan tersebut ke BUD, yang kemudian setiap hari juga BUD akan menyerahkan laporan tersebut ke bagian pembukuan DISPENDA Kabupaten Madiun.

Bagian pembukuan kemudian akan membukukan atau mencatat setiap bukti setoran yang telah diberikan. Pada proses pencatatan atau pembukuan yang dilakukan menggunakan program *Microsoft Excel*. Bagian pembukuan melihat satu per satu rekening koran dari semua Wajib Pajak yang sudah membayar untuk kemudian dicatat. Hasil pembukuan pembayaran Wajib Pajak sangat diperlukan oleh bagian perencanaan. Bagian perencanaan menggunakan data pembayaran wajib pajak untuk dilakukan evaluasi mengenai keberhasilan realisasi pendapatan dan perencanaan target pendapatan di tahun depan. Pada Gambar 4.1 merupakan gambaran proses bisnis penerimaan PAD pada DISPENDA Kabupaten Madiun yang telah dimodelkan sesuai dengan notasi BPMN.



Gambar 4.1 Proses bisnis penerimaan PAD DISPENDA Kab. Madiun saat ini

4.1.2 Proses bisnis usulan



Gambar 4.2 Proses bisnis usulan penerimaan PAD

Proses bisnis usulan pada Gambar 4.2 merupakan proses bisnis yang akan berjalan setelah sistem yang akan dirancang diterapkan. Terdapat beberapa aktivitas atau *task* yang ditambahkan, begitu juga dengan proses yang terdapat didalamnya juga mengalami perubahan. Aktivitas yang baru tersebut dapat dilihat penjelasannya pada Tabel 4.1.

**Tabel 4.1 Aktivitas baru pada proses bisnis usulan**

Kode Aktivitas	Aktivitas	Keterangan
P-PB-001	Melaporkan pembayaran	Dilakukan oleh wajib pajak saat wajib pajak telah melakukan pembayaran pajak.
P-PB-002	Mengubah status pembayaran	Dilakukan oleh bidang pembukuan dan bendahara umum daerah saat menerima laporan rekonsiliasi pembayaran pajak.
P-PB-003	Validasi pembayaran	Dilakukan oleh bidang pembukuan dan bendahara umum daerah setelah menerima laporan pembayaran dari wajib pajak.
P-PB-004	Melihat laporan realisasi PAD	Dilakukan oleh bidang pembukuan dan bidang perencanaan untuk melihat daftar realisasi pajak.
P-PB-005	Memonitor pembayaran pajak	Dilakukan oleh bidang perencanaan untuk memantau pembayaran pajak oleh wajib pajak
P-PB-006	Mencetak laporan realisasi PAD	Dilakukan oleh bidang pembukuan untuk mencetak data realisasi PAD.

Pada penelitian ini, ada beberapa aktivitas/proses bisnis usulan yang menjadi pendukung dalam penelitian ini namun tidak dicantumkan dalam penelitian ini dan tidak menjadi fokus bahasan. Aktivitas/proses bisnis usulan tersebut akan memiliki kode tersendiri, yaitu P-PB-XX guna kepentingan *traceability* atau keruntutan.

## 4.2 Analisis permasalahan

Bagian ini menjelaskan mengenai analisis permasalahan yang dihasilkan berdasarkan masalah yang ada pada DISPENDA Kabupaten Madiun dan proses bisnis yang telah dibuat. Hasil analisis permasalahan tersebut dijelaskan dalam bentuk tabel seperti yang ditunjukkan pada Tabel 4.2.

**Tabel 4.2 Analisis Permasalahan**

<b>Masalah</b>	<ol style="list-style-type: none"><li>1. Membutuhkan waktu yang lama untuk mendapatkan laporan data wajib pajak yang sudah membayar.</li><li>2. Sistem yang ada hanya menyediakan informasi penetapan dan pendataan pajak dan retribusi, belum ada sistem yang menyediakan data mengenai wajib pajak yang belum membayar maupun yang sudah membayar.</li><li>3. Belum ada sistem yang bisa digunakan untuk memantau pembayaran yang dilakukan oleh wajib pajak.</li><li>4. Sering terjadi perbedaan nominal antara rencana biaya pajak dan realisasi pajak yang dibayarkan.</li><li>5. Belum terdapat analisis dan perancangan sistem yang jelas yang bisa digunakan untuk mengimplementasikan sistem sebagai penyelesaian masalah pelaporan pajak.</li></ol>
<b>Mempengaruhi</b>	Dinas pendapatan, bidang perencanaan, bidang pembukuan dan pelaporan, bendahara penerimaan serta kas daerah.
<b>Dampak</b>	<ol style="list-style-type: none"><li>1. Perlu dilakukan pencarian manual mengenai wajib pajak yang belum membayar pajak ataupun yang sudah membayar.</li><li>2. Waktu pembuatan realisasi pajak tertunda.</li><li>3. Target pendapatan tidak tercapai sesuai perencanaan.</li><li>4. Sistem yang akan dikembangkan belum bisa terealisasi.</li></ol>
<b>Solusi</b>	Sesuai latar belakang yang terdapat dalam penelitian ini, telah dijelaskan bahwa dalam pengembangan sistem terdapat dua tahapan yang penting yaitu analisis dan perancangan sistem, maka solusi yang diberikan adalah analisis dan perancangan sistem pelaporan pajak untuk Dinas Pendapatan Kabupaten Madiun.

## 4.3 Analisis Pemangku Kepentingan dan Pengguna

### 4.3.1 Tipe Pemangku Kepentingan

Pemangku kepentingan yang terkait dalam penelitian ini terbagi menjadi tiga tipe pemangku kepentingan, yaitu pengguna, pelanggan, dan pengembang. Penjelasan mengenai tipe pemangku kepentingan dan pemangku kepentingan yang terlibat dapat dilihat pada Tabel 4.3.

Tabel 4.3 Analisis Tipe Pemangku kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
<b>Pengguna</b>	Pihak yang menggunakan sistem.	Kepala bidang perencanaan, petugas bidang pembukuan dan pelaporan, petugas bidang pendataan, bendahara umum daerah, wajib pajak
<b>Pelanggan</b>	Perusahaan yang akan menjadi pemilik dari sistem.	Dinas Pendapatan Kabupaten Madiun
<b>Pengembang</b>	Orang yang akan melakukan pengembangan/pembuatan sistem dengan tahap-tahap tertentu dalam pengembangan.	Penulis

#### 4.3.2 Peran dan Representatif Pemangku Kepentingan

Sesuai dengan analisis tipe pemangku kepentingan, terdapat tiga tipe pemangku kepentingan yang terlibat dalam penelitian ini. Pemangku kepentingan pertama adalah pengguna, dimana pengguna ini berperan sebagai pihak yang nantinya akan menggunakan sistem. Dalam penelitian ini, pengguna nantinya akan berperan pada tahapan evaluasi untuk mengetahui apakah sistem yang dirancang sudah sesuai dengan keinginan pengguna atau belum. Pemangku kepentingan kedua adalah pelanggan. Pelanggan berperan penting dalam penelitian ini, karena dari pelangganlah semua persyaratan sistem dapat diketahui.

Pelanggan terlibat dalam penelitian sebagai narasumber utama yang diwawancarai oleh pengembang. Pelanggan bertanggung jawab untuk menyampaikan permasalahan, menyediakan informasi yang diperlukan untuk analisis dan perancangan sistem, dan ikut serta dalam kegiatan analisis persyaratan dan perancangan sistem. Pengembang merupakan pemangku kepentingan yang terakhir yang berperan sebagai analis sistem, dimana tanggung jawabnya adalah untuk menganalisis persyaratan sistem, merancang sistem, dan membuat dokumen hasil analisis dan perancangan sistem.

#### 4.3.3 Tipe Pengguna

Dari analisis yang dilakukan, ada dua tipe pengguna yang terlibat dalam sistem yang akan dirancang. Tipe pertama adalah adapter teknologi, tipe ini merupakan golongan pengguna yang menggunakan teknologi dan mau mempelajari teknologi lebih lanjut. Tipe kedua adalah pengguna standar, yaitu merupakan tipe pengguna dimana pengguna tersebut hanya sebatas mau menggunakan teknologi tapi enggan untuk mempelajari teknologi lebih lanjut. Tabel 4.4 menjelaskan mengenai

pengguna yang telah dianalisis beserta karakteristik masing-masing tipe pengguna.

**Tabel 4.4 Analisis Tipe Pengguna**

Tipe Pengguna	Karakteristik	Pengguna
Adapter Teknologi	<ul style="list-style-type: none"> <li>- Usia antara 20 – 50 tahun, pria atau wanita.</li> <li>- Bekerja dalam lingkungan pemerintahan.</li> </ul>	<ul style="list-style-type: none"> <li>- Kepala bidang perencanaan</li> <li>- Petugas bidang pembukuan dan pelaporan</li> <li>- Petugas Bendahara Umum Daerah</li> </ul>
Pengguna Standar	<ul style="list-style-type: none"> <li>- Usia antara 20 – 50 tahun, pria atau wanita.</li> <li>- Memiliki bidang usaha.</li> </ul>	Wajib pajak

## 4.4 Analisis Persyaratan

### 4.4.1 Kebutuhan Pengguna dan Pemangku Kepentingan

Bagian ini menjelaskan tentang kebutuhan pengguna dan pemangku kepentingan yang diharapkan ada pada sistem yang akan dirancang. Pada bagian ini persyaratan sistem tidak dijelaskan secara spesifik, namun lebih menjelaskan mengapa persyaratan sistem tersebut dibutuhkan. Kebutuhan pengguna dan pemangku kepentingan tersebut dijelaskan pada Tabel 4.5.

**Tabel 4.5 Analisis kebutuhan pengguna dan pemangku kepentingan**

Kebutuhan	Pemangku kepentingan	Prioritas	Situasi saat ini	Solusi yang ditawarkan
Kemudahan melihat data realisasi PAD (pembayaran pajak)	Bidang Perencanaan	Mo	Bidang perencanaan harus meminta data realisasi PAD ke bidang pembukuan terlebih dahulu.	Bidang perencanaan dapat melihat data realisasi PAD secara <i>online</i> secara langsung melalui sistem
Kemudahan melihat status dan realisasi pembayaran wajib pajak	Bidang pembukuan, bidang perencanaan	Mo	Untuk melihat pembayaran pajak harus melihat ulang pada surat setoran yang masuk atau pada <i>Microsoft Excel</i> .	Untuk melihat status pembayaran pajak cukup melihat secara langsung pada sistem.

Tabel 4.5 Analisis kebutuhan pengguna dan pemangku kepentingan (lanjutan)

Kebutuhan	Pemangku kepentingan	Prioritas	Situasi saat ini	Solusi yang ditawarkan
Transparansi penerimaan pajak	Wajib pajak, bidang perencanaan	S	Wajib pajak dan bidang perencanaan tidak tahu pasti apakah jumlah uang yang dibayarkan sesuai dengan yang dituliskan di laporan pembayaran pajak.	Dengan adanya pelaporan pajak, wajib pajak bisa memonitor proses pembayaran pajaknya dan bidang perencanaan bisa memastikan pendapatan pajak sesuai dengan yang dibayarkan.
Terdapat pemberitahuan apabila ada wajib pajak yang melaporkan pembayarannya	Bidang pembukuan, bendahara umum daerah	S	-	Sistem akan mengirimkan pemberitahuan otomatis saat ada pelaporan pajak yang baru.
Kemudahan akses bagi pengguna sistem	Bidang perencanaan, pembukuan, Bendahara Umum daerah, Wajib Pajak	Mo	-	Sistem dapat diakses di setiap waktu dan tempat selama terdapat koneksi internet
Integrasi dan keamanan data	Bidang perencanaan, pembukuan, Bendahara Umum daerah, Wajib Pajak	Mo	-	Sistem menyediakan proses autentifikasi pengguna.
Kecepatan akses dan waktu respons dari sistem.	Bidang perencanaan, pembukuan, Bendahara Umum daerah, Wajib Pajak	S	-	Sistem bisa diakses maksimal 5 detik.
Mengikuti kebijakan perusahaan terkait dengan proses izin lokasi	Bidang perencanaan, pembukuan, Bendahara Umum daerah, Wajib Pajak	Mo	-	Sistem dikembangkan sesuai dengan peraturan dan SOP dari DISPENDA Kabupaten Madiun.

#### 4.4.2 Kedudukan Produk

Bagian ini menjelaskan secara umum mengenai manfaat yang diberikan oleh sistem dan kemampuan sistem. Bagaimana kedudukan sistem pelaporan pembayaran pajak terhadap instansi yang terkait dan bagaimana sistem ini bekerja. Pada Tabel 4.6 merupakan penjelasan kedudukan produk dari sistem pelaporan pembayaran pajak.

**Tabel 4.6 Kedudukan Produk Sistem Pelaporan Pembayaran Pajak**

<b>Untuk</b>	Pegawai DISPENDA Kabupaten Madiun dan wajib pajak Kabupaten Madiun.
<b>Yang</b>	Melaporkan pembayaran pajak (dalam hal ini adalah wajib pajak) dan mengelola laporan pembayaran dan perencanaan pajak (dalam hal ini adalah pegawai DISPENDA Madiun).
<b>Sebuah</b>	Sistem pelaporan pembayaran pajak merupakan sistem berbasis web yang bertujuan untuk memfasilitasi wajib pajak melaporkan pembayaran pajak serta memfasilitasi pegawai instansi untuk mengelola pembayaran dan data pajak.
<b>Yang</b>	Bermanfaat untuk mempermudah memonitor pemasukan pembayaran pajak, melaporkan pembayaran pajak, pembuatan laporan pembayaran pajak atau realisasi PAD, membuat laporan target dan rencana pajak dan kemudahan melihat dan mencetak segala laporan.
<b>Tidak Seperti</b>	Pendataan pembayaran pajak membutuhkan waktu yang lama karena mendata satu persatu dari rekening koran yang masuk setiap hari. Sebelumnya transparansi uang pajak yang masuk tidak jelas karena tidak adanya pelaporan pembayaran pajak oleh wajib pajak.
<b>Produk</b>	Sistem pelaporan pembayaran pajak dapat diakses selama 24 jam selama terkoneksi dengan internet, transparansi uang pajak terlihat, pembuatan laporan pajak dimudahkan dengan melalui sistem.

#### 4.4.3 Fitur Produk

Analisis fitur dilakukan untuk mengetahui fitur apa saja yang tersedia di produk yang dikembangkan berdasarkan hasil analisis kebutuhan pemangku kepentingan dan pengguna. Daftar fitur yang dianalisis dijelaskan pada Tabel 4.7.

Tabel 4.7 Fitur yang disediakan sistem

Kode Fitur	Nama Fitur	Deskripsi	Prioritas
FITUR01	<i>Login</i>	Sistem dapat diakses oleh kepala bidang perencanaan, petugas bidang pembukuan dan pelaporan, petugas bendahara umum daerah dan wajib pajak.	MO
FITUR02	Kelola data pengguna	Sistem dapat memberikan informasi mengenai data wajib pajak dan pengguna.	MO
FITUR03	Ubah status pembayaran	Sistem menyediakan fitur untuk petugas bendahara umum daerah dan petugas bidang pembukuan untuk mengubah status pembayaran.	MO
FITUR04	Validasi laporan pembayaran	Sistem menyediakan fitur untuk petugas bendahara umum daerah dan petugas bidang pembukuan untuk mengubah status validasi pembayaran saat wajib pajak melaporkan pembayaran pajak.	MO
FITUR05	Lihat data pembayaran pajak	Sistem dapat memberikan informasi data pembayaran pajak.	MO
FITUR06	Pantau pembayaran pajak	Sistem dapat memberikan fitur untuk memantau pembayaran pajak oleh setiap wajib pajak.	MO
FITUR07	Pelaporan Pembayaran Pajak	Sistem menyediakan fitur pelaporan pembayaran pajak untuk pengguna wajib pajak.	MO
FITUR08	Registrasi pengguna	Sistem dapat memberikan fitur untuk mendaftarkan pengguna baru.	MO
FITUR09	Cetak data pembayaran pajak	Sistem menyediakan fitur cetak pada data pembayaran pajak.	S
FITUR10	Cetak laporan rencana dan realisasi pajak	Sistem menyediakan fitur cetak pada laporan rencana dan realisasi pajak.	S
FITUR11	-	Sistem memiliki waktu respons maksimal 5 detik	S
FITUR12	-	Sistem memiliki kemudahan akses di setiap tempat selama terkoneksi dengan internet	S

#### 4.4.4 Persyaratan Deklaratif

Dalam pengembangan sistem, persyaratan dibedakan menjadi dua yaitu kebutuhan fungsional dan kebutuhan fungsional. Kebutuhan-kebutuhan tersebut merupakan penjelasan secara spesifik tentang kebutuhan dari pengguna dan apa yang dapat dilakukan oleh sistem.

##### 4.4.4.1 Persyaratan Fungsional

Persyaratan fungsional merupakan pengelompokan persyaratan-persyaratan yang sudah teridentifikasi sebelumnya sesuai dengan fungsi dari sistem pelaporan

pembayaran pajak, adapun kebutuhan fungsional tersebut dijelaskan pada Tabel 4.8.

**Tabel 4.8 Persyaratan Fungsional Sistem**

Kode Fitur	Kode Dasar Fungsi	Kode Lengkap Fungsi	Deskripsi
FITUR01	P-KF-001	P-KF-001.1	Sistem harus menyediakan fungsi bagi pengguna untuk melakukan masuk ke akun pengguna.
FITUR02	P-KF-002	P-KF-002.1	Sistem harus menyediakan fungsi bagi admin untuk mengedit data pengguna dari sistem.
		P-KF-002.2	Sistem harus menyediakan fungsi bagi admin untuk menghapus pengguna dari sistem.
FITUR08	P-KF-003	P-KF-003.1	Sistem harus menyediakan fungsi bagi admin untuk mendaftarkan pengguna baru di sistem.
FITUR03	P-KF-004	P-KF-004.1	Sistem harus menyediakan fungsi bagi bidang pembukuan dan bendahara umum daerah untuk mengubah status pembayaran dari wajib pajak.
FITUR04	P-KF-005	P-KF-005.1	Sistem harus menyediakan fungsi bagi bidang pembukuan dan bendahara umum daerah untuk melakukan validasi atas laporan pembayaran pajak oleh wajib pajak.
FITUR05	P-KF-006	P-KF-006.1	Sistem harus menyediakan fungsi untuk melihat data pembayaran pajak.
FITUR09	P-KF-007	P-KF-007.1	Sistem menyediakan fungsi untuk mencetak data pembayaran pajak.
FITUR06	P-KF-008	P-KF-008.1	Sistem harus menyediakan fungsi untuk memantau pembayaran pajak oleh setiap wajib pajak dan memberikan catatan pembayaran pajak.
FITUR05	P-KF-009	P-KF-009.1	Sistem harus menyediakan fungsi bagi bidang perencanaan untuk melihat laporan rencana target dan realisasi pajak baik berupa tabel maupun grafik.

**Tabel 4.8 Persyaratan Fungsional Sistem (lanjutan)**

Kode Fitur	Kode Dasar Fungsi	Kode Lengkap Fungsi	Deskripsi
FITUR10	P-KF-010	P-KF-010.1	Sistem menyediakan fungsi untuk mencetak laporan rencana dan target pajak.
FITUR07	P-KF-011	P-KF-011.1	Sistem harus menyediakan fungsi bagi wajib pajak untuk melaporkan pembayaran pajak

#### 4.4.4.2 Persyaratan Non Fungsional

Bagian ini menjelaskan mengenai kebutuhan non fungsional yang terkait dengan kemudahan pengguna dalam menggunakan sistem. Kebutuhan fungsional yang dibutuhkan sistem dapat dilihat pada Tabel 4.9.

**Tabel 4.9 Persyaratan Non Fungsional**

Kode Fungsi	Nama Fungsi	Deskripsi
P-KNF-001	<i>Performance</i>	Sistem memiliki waktu respons maksimal 5 detik
P-KNF-002	<i>Usability</i>	Sistem memiliki kemudahan akses di setiap tempat selama terkoneksi dengan internet

### 4.5 Pemodelan Use Case

#### 4.5.1 Identifikasi Aktor

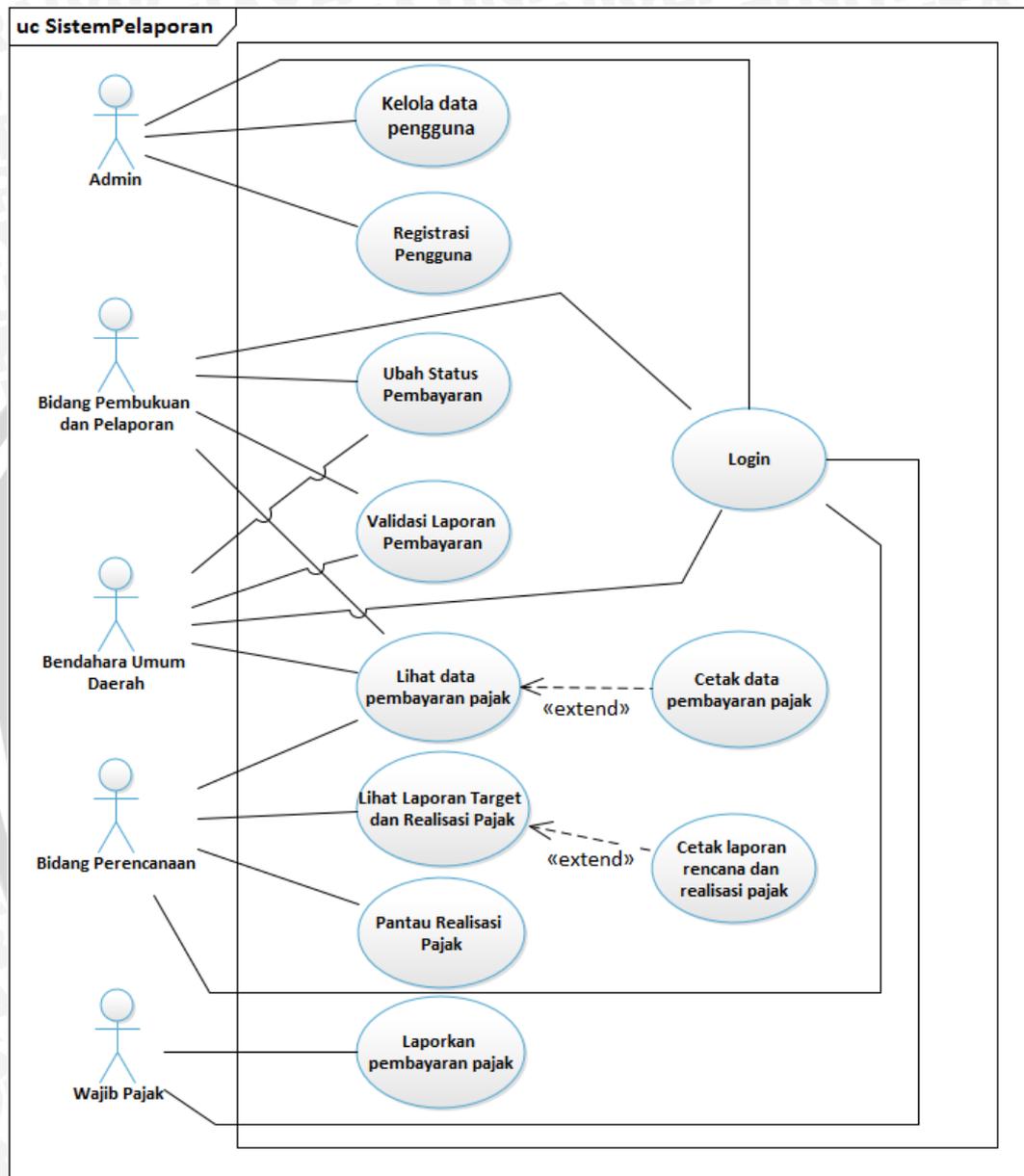
Pada bagian ini menjelaskan mengenai siapa saja aktor yang terlibat pada pemodelan *use case*. Pada Tabel 4.10 merupakan penjelasan mengenai aktor-aktor yang terlibat.

**Tabel 4.10 Identifikasi Aktor**

Aktor	Tipe Pengguna	Contoh
Admin	Adapter Teknologi	Petugas bidang pembukuan dan pelaporan
Bidang Pembukuan dan Pelaporan	Adapter Teknologi	Petugas bidang pembukuan dan pelaporan
Bendahara Umum Daerah	Adapter Teknologi	Petugas Bendahara Umum Daerah
Bidang Perencanaan	Adapter Teknologi	Kepala bidang perencanaan
Wajib Pajak	Pengguna Standar	Wajib pajak

#### 4.5.2 Diagram Use Case

Use case yang digambarkan pada Gambar 4.3 merupakan hasil dari penggalan kebutuhan yang sebelumnya telah dilakukan.



Gambar 4.3 Diagram Use Case Sistem Pelaporan Pembayaran Pajak

Pada Tabel 4.11 menjelaskan pemetaan hubungan antara persyaratan fungsional dengan use case yang telah digambarkan pada Gambar 4.3.

Tabel 4.11 Pemetaan Diagram Use Case

Kode Fungsi	Kode Use case	Use case	Aktor
P-KF-002	P-UC-001	Kelola data pengguna	Admin
P-KF-003	P-UC-002	Registrasi pengguna	Admin
P-KF-004	P-UC-003	Ubah status pembayaran	Bidang pembukuan, bendahara umum daerah
P-KF-005	P-UC-004	Validasi laporan pembayaran	Bidang pembukuan, bendahara umum daerah
P-KF-006	P-UC-005	Lihat data pembayaran pajak	Bidang pembukuan, bendahara umum daerah, bidang perencanaan
P-KF-009	P-UC-006	Lihat laporan target dan realisasi pajak	Bidang perencanaan
P-KF-008	P-UC-007	Pantau pembayaran pajak	Bidang perencanaan
P-KF-011	P-UC-008	Laporkan pembayaran pajak	Wajib pajak
P-KF-007	P-UC-009	Cetak data pembayaran pajak	Bidang pembukuan, bendahara umum daerah, bidang perencanaan
P-KF-010	P-UC-010	Cetak laporan target dan realisasi pajak	Bidang perencanaan
P-KF-001	P-UC-011	Login	Bidang pembukuan, bendahara umum daerah, bidang perencanaan, wajib pajak

#### 4.6 Spesifikasi Use Case

Pada bagian ini, setiap *use case* yang telah digambarkan akan dijabarkan dengan cara deskripsi. Deskripsi tersebut berisi tentang alur masing-masing *use case* yang telah dibuat.

##### A. Spesifikasi Use Case – Kelola Data Pengguna

Tabel 4.12 Spesifikasi Use Case Kelola Data Pengguna

<b>Name</b>	Kelola Data Pengguna
<b>Actor</b>	Admin
<b>Brief Description</b>	Use case ini mendeskripsikan tentang bagaimana aktor (admin) mengelola data dari pengguna yang sudah tersimpan.

<p><b>Pre-Condition</b></p>	<ol style="list-style-type: none"> <li>Admin menggunakan perangkat yang terhubung dengan jaringan internet.</li> <li>Admin telah berhasil masuk ke sistem.</li> </ol>
<p><b>Basic Flow / flow of events</b></p>	<p><b>{Menampilkan daftar pengguna}</b></p> <ol style="list-style-type: none"> <li>Dimulai saat aktor memilih menu daftar pengguna.</li> <li>Sistem menampilkan daftar pengguna.</li> </ol> <p><b>{Pilih detail pengguna}</b></p> <ol style="list-style-type: none"> <li>Memilih salah satu pengguna.</li> <li>Admin memilih menu detail dari salah satu pengguna yang diinginkan.</li> </ol> <p><b>{Menampilkan detail data pengguna}</b></p> <ol style="list-style-type: none"> <li>Sistem akan menampilkan modal formulir detail pengguna dengan data terisi sesuai data yang sudah dimasukkan sebelumnya.</li> <li>Aktor menutup modal detail pengguna.</li> </ol> <p><b>{Use case selesai}</b></p> <ol style="list-style-type: none"> <li><i>Use case</i> selesai.</li> </ol>
<p><b>Alternative Flows</b></p>	<p>A1. Alternatif Edit Data Pengguna</p> <ol style="list-style-type: none"> <li>Jika aktor mengubah data dari detail pengguna. Saat pada <b>{Menampilkan detail data pengguna}</b> aktor mengubah salah satu atau beberapa kolom data pada formulir detail pengguna.                     <p><b>{Mengedit data pengguna}</b></p> <ol style="list-style-type: none"> <li>Aktor mengedit data pengguna dengan data baru.</li> </ol> <p><b>{Pilih simpan data}</b></p> <ol style="list-style-type: none"> <li>Aktor memilih untuk menyimpan perubahan.</li> </ol> <p><b>{Menyimpan data}</b></p> <ol style="list-style-type: none"> <li>Sistem menyimpan data pengguna yang telah diubah dan telah dikonfirmasi.</li> </ol> </li> <li>Melanjutkan <i>use case</i> dari (<b><i>Use case</i> selesai</b>).</li> </ol> <p>A2. Alternatif Hapus Pengguna</p> <ol style="list-style-type: none"> <li>Jika aktor memilih hapus pengguna. Saat <b>{Menampilkan daftar pengguna}</b> aktor memilih menghapus pengguna.                     <p><b>{Pilih hapus pengguna}</b></p> <ol style="list-style-type: none"> <li>Aktor memilih untuk menghapus dari salah satu pengguna yang diinginkan.</li> </ol> <p><b>{Konfirmasi hapus}</b></p> <ol style="list-style-type: none"> <li>Sistem menampilkan jendela konfirmasi.</li> <li>Aktor melakukan konfirmasi hapus pengguna.</li> </ol> <p><b>{Menghapus data}</b></p> <ol style="list-style-type: none"> <li>Sistem menghapus pengguna yang telah dipilih dan dikonfirmasi.</li> </ol> </li> <li>Melanjutkan aliran <i>use case</i> dari <b>{Use case selesai}</b>.</li> </ol> <p>A3. Penanganan kesalahan pengisian formulir</p>

	<ol style="list-style-type: none"> <li>1. Jika terdapat kolom formulir yang belum diisi Pada alternatif A1 saat aktor melakukan <b>{Pilih simpan data}</b> atau formulir tidak diisi dengan benar.             <ol style="list-style-type: none"> <li>a. Sistem akan menampilkan pesan meminta melengkapi formulir registrasi.</li> </ol> </li> <li>2. Melanjutkan aliran <i>use case</i> dari <b>{Mengedit data pengguna}</b></li> </ol> <p>A4. Penanganan pelanggaran</p> <ol style="list-style-type: none"> <li>1. Jika aktor membatalkan edit data Saat melakukan alternatif A1, aktor membatalkan pengeditan data.</li> <li>2. Jika aktor tidak mengkonfirmasi hapus data Saat aktor memilih untuk tidak mengonfirmasi pada saat <b>{Konfirmasi hapus}</b> pada alternatif A2 atau memilih tombol tutup.</li> <li>3. Melanjutkan aliran <i>use case</i> dari <b>{Menampilkan daftar pengguna}</b></li> </ol>
<b>Sub-flow</b>	-
<b>Post-Condition</b>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan daftar pengguna dengan data yang telah terganti.</li> <li>2. Sistem menampilkan daftar pengguna dengan tidak ada data yang terganti.</li> </ol>

<b>Glossary</b>	
Data pengguna yang dimaksud adalah data dari wajib pajak hotel atau restoran, yaitu : nama, NPWP, alamat, lokasi usaha, nama usaha, ketetapan pajak.	

**B. Spesifikasi Use Case – Registrasi Pengguna**

**Tabel 4.13 Spesifikasi Use Case Registrasi Pengguna**

<b>Name</b>	Pendaftaran Pengguna
<b>Actor</b>	Admin
<b>Brief Description</b>	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor (admin) menambahkan pengguna dari sistem ini.
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Admin menggunakan perangkat yang terhubung dengan jaringan internet.</li> <li>2. Admin telah berhasil masuk ke sistem.</li> </ol>
<b>Basic Flow / flow of events</b>	<p><b>{Pilih registrasi pengguna}</b></p> <ol style="list-style-type: none"> <li>1. Admin masuk ke halaman utama admin.</li> <li>2. Admin memilih menu registrasi pengguna.</li> </ol> <p><b>{Menampilkan formulir registrasi}</b></p> <ol style="list-style-type: none"> <li>3. Sistem akan menampilkan formulir registrasi pengguna baru.</li> </ol> <p><b>{Mengisi formulir registrasi}</b></p>

	<p>4. Admin memilih terlebih dahulu jenis pajak yang ditanggung oleh wajib pajak.</p> <p>5. Admin mengisi formulir registrasi pengguna baru.</p> <p><b>{Pilih register}</b></p> <p>6. Admin memilih pilihan <i>register</i>.</p> <p><b>{Menyimpan data}</b></p> <p>7. Sistem menyimpan data pengguna baru yang telah ditambahkan dan dikonfirmasi.</p> <p><b>{Use case selesai}</b></p> <p>8. <i>Use case</i> selesai</p>
<b>Alternative Flows</b>	<p>A1. Penanganan kesalahan pengisian formulir</p> <p>1. Jika <i>username</i> pengguna baru sudah ada atau sudah digunakan.                  Pada saat <b>{Menyimpan data}</b> <i>username</i> pengguna baru yang dimasukkan oleh aktor, sistem akan mengecek basis data dan <i>username</i> sudah dipakai oleh pengguna lain atau sudah ada di basis data.</p> <p>a. Sistem akan menampilkan pesan bahwa <i>username</i> telah digunakan dan meminta mengganti <i>username</i> pengguna yang di daftarkan.</p> <p>2. Jika terdapat kolom formulir yang belum diisi                  Saat aktor melakukan <b>{Pilih register}</b> dan formulir tidak diisi dengan lengkap.</p> <p>a. Sistem akan meminta melengkapi formulir registrasi.</p> <p>3. Melanjutkan aliran <i>use case</i> dari <b>{Mengisi formulir registrasi}</b></p> <p>A2. Penanganan pelanggaran</p> <p>1. Pembatalan registrasi                  Saat sedang mengisi formulir, aktor membatalkan registrasi.</p> <p>a. Sistem akan menampilkan notifikasi apakah aktor benar-benar ingin membatalkan registrasi pengguna.</p> <p>2. Melanjutkan aliran <i>use case</i> dari <b>{Use case selesai}</b></p>
<b>Sub-flow</b>	-
<b>Post-Condition</b>	<p>1. Sistem menampilkan daftar pengguna dengan terdapat pengguna yang baru didaftarkan.</p> <p>2. Sistem menampilkan daftar pengguna dengan tidak terdapat pengguna baru.</p>

**Glossary**

Data pengguna yang dimaksud adalah data dari wajib pajak hotel atau restoran, yaitu : nama, NPWP, alamat, lokasi usaha, nama usaha, ketetapan pajak.



C. Spesifikasi Use Case – Ubah Status Pembayaran

Tabel 4.14 Spesifikasi Use Case Ubah Status Pembayaran

<b>Name</b>	Ubah Status Pembayaran
<b>Actor</b>	Bendahara Umum Daerah dan bidang pembukuan
<b>Brief Description</b>	Use case ini mendeskripsikan tentang bagaimana aktor (Bendahara Umum Daerah dan Bidang Pembukuan) mengubah status pembayaran dari wajib pajak.
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Bendahara umum daerah dan bidang pembukuan menggunakan perangkat yang terhubung dengan jaringan internet.</li> <li>2. Bendahara umum daerah atau bidang pembukuan telah berhasil masuk ke sistem.</li> </ol>
<b>Basic Flow / flow of events</b>	<p><b>{Pilih edit status pembayaran}</b></p> <ol style="list-style-type: none"> <li>1. Dimulai saat bendahara umum daerah dan bidang pembukuan memilih menu manajemen pembayaran.</li> <li>2. Sistem menampilkan daftar data pembayaran.</li> <li>3. Bendahara umum daerah dan bidang pembukuan memilih salah satu nama pembayar pajak dengan status pembayaran “belum” dan memilih pilihan ubah pembayaran.</li> </ol> <p><b>{Menampilkan detail wajib pajak}</b></p> <ol style="list-style-type: none"> <li>4. Sistem akan menampilkan formulir data wajib pajak.</li> <li>5. Bendahara umum daerah dan bidang pembukuan mencocokkan data.</li> </ol> <p><b>{Mengubah status pembayaran}</b></p> <ol style="list-style-type: none"> <li>6. Bendahara umum daerah dan bidang pembukuan memilih mengubah status pembayaran.</li> </ol> <p><b>{Konfirmasi ubah status pembayaran}</b></p> <ol style="list-style-type: none"> <li>7. Sistem menampilkan jendela konfirmasi.</li> <li>8. Bendahara umum daerah dan bidang pembukuan pilihan ubah.</li> </ol> <p><b>{Menyimpan data}</b></p> <ol style="list-style-type: none"> <li>9. Sistem menyimpan status pembayaran yang sudah diubah.</li> <li>10. Status pembayaran berubah.</li> </ol> <p><b>{Use case selesai}</b></p> <ol style="list-style-type: none"> <li>11. Use case selesai.</li> </ol>
<b>Alternative Flows</b>	<p>A1. Penanganan pelanggaran</p> <ol style="list-style-type: none"> <li>1. Jika aktor tidak mengkonfirmasi ubah status pembayaran. Saat aktor memilih batal pada saat <b>{Konfirmasi ubah status pembayaran}</b> atau memilih tombol tutup.</li> </ol>

	2. Melanjutkan aliran <i>use case</i> dari <b>{Use case selesai}</b>
<b>Sub-flow</b>	-
<b>Post-Condition</b>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan daftar pengguna dengan status pembayaran dari pengguna yang dipilih berubah menjadi "Lunas".</li> <li>2. Sistem menampilkan daftar pengguna dengan status pembayaran dari pengguna yang dipilih tidak berubah menjadi "Lunas".</li> </ol>

<b>Glossary</b>
Proses pengubahan status pembayaran meliputi data tanggal pembayaran, jumlah pembayaran, ketetapan pajak, dan status pembayaran.

#### D. Spesifikasi Use Case – Validasi Laporan Pembayaran

Tabel 4.15 Spesifikasi Use Case Validasi Laporan Pembayaran

<b>Name</b>	Validasi Laporan Pembayaran
<b>Actor</b>	Bendahara Umum Daerah dan Bidang Pembukuan
<b>Brief Description</b>	<i>Use case</i> ini mendeskripsikan tentang bagaimana bendahara umum daerah dan bidang pembukuan melakukan validasi laporan pembayaran dari wajib pajak.
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Bendahara umum daerah dan bidang pembukuan menggunakan perangkat yang terhubung dengan jaringan internet.</li> <li>2. Bendahara umum daerah atau bidang pembukuan telah berhasil masuk ke sistem.</li> </ol>
<b>Basic Flow / flow of events</b>	<p><b>{Pilih detail laporan pembayaran}</b></p> <ol style="list-style-type: none"> <li>1. Bendahara umum daerah dan bidang pembukuan memilih pilihan laporan pembayaran.</li> <li>2. Sistem menampilkan daftar laporan pembayaran pajak.</li> <li>3. Bendahara umum daerah dan bidang pembukuan memilih detail laporan pembayaran dari salah satu nama wajib pajak.</li> </ol> <p><b>{Menampilkan detail pembayaran pajak}</b></p> <ol style="list-style-type: none"> <li>4. Sistem akan menampilkan detail pembayaran pajak dari pengguna yang dipilih.</li> <li>5. Bendahara umum daerah dan bidang pembukuan memeriksa data pembayaran pajak dari pengguna tersebut.</li> </ol> <p><b>{Validasi pembayaran}</b></p> <ol style="list-style-type: none"> <li>6. Bendahara umum daerah dan bidang pembukuan memilih validasi laporan pembayaran.</li> </ol>

	<p>7. Bendahara umum daerah dan bidang pembukuan memilih konfirmasi validasi laporan pembayaran.  <b>{Menyimpan data}</b></p> <p>8. Sistem menyimpan status validasi pembayaran yang sudah diubah.</p> <p>9. Status validasi berubah.  <b>{Use case selesai}</b></p> <p>10. <i>Use case</i> selesai.</p>
<b>Alternative Flows</b>	<p>A1. Penanganan pelanggaran</p> <p>1. Jika bendahara umum daerah dan bidang pembukuan tidak mengkonfirmasi validasi Saat setelah melakukan validasi bendahara umum daerah dan bidang pembukuan tidak melakukan konfirmasi atau memilih kembali ke menu selanjutnya.</p> <p>2. Melanjutkan aliran <i>use case</i> dari <b>{Use case selesai}</b></p>
<b>Sub-flow</b>	=
<b>Post-Condition</b>	<p>1. Sistem melanjutkan alur pelaporan ke tahap selanjutnya.</p> <p>2. Sistem tidak melanjutkan alur pelaporan ke tahap selanjutnya.</p>

<b>Glossary</b>
Proses validasi laporan pembayaran meliputi data tanggal pembayaran, tanggal pelaporan, jumlah pembayaran, ketentuan pajak, status pembayaran, dan status validasi.

**E. Spesifikasi Use Case – Lihat data pembayaran pajak**

**Tabel 4.16 Spesifikasi Use Case Lihat data pembayaran pajak**

<b>Name</b>	Lihat data pembayaran pajak
<b>Actor</b>	Bendahara Umum Daerah, bidang pembukuan, bidang perencanaan
<b>Brief Description</b>	<i>Use case</i> ini mendeskripsikan tentang bagaimana bendahara umum daerah atau bidang pembukuan atau bidang perencanaan melihat data pembayaran dari wajib pajak.
<b>Pre-Condition</b>	<p>1. Bendahara Umum Daerah, bidang pembukuan, dan bidang perencanaan menggunakan perangkat yang terhubung dengan jaringan internet.</p> <p>2. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan telah berhasil masuk ke sistem.</p>
<b>Basic Flow / flow of events</b>	<b>{Lihat data pembayaran pajak}</b>

	<ol style="list-style-type: none"> <li>1. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan memilih menu data pembayaran pajak.</li> <li>2. Sistem menampilkan menu data pembayaran pajak.</li> <li>3. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan mengisi data filter.</li> <li>4. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan memilih pilihan cari.</li> <li>5. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan melihat daftar pembayaran pajak.</li> </ol> <p><b>{Use case selesai}</b></p> <ol style="list-style-type: none"> <li>6. Use case selesai.</li> </ol>
<b>Alternative Flows</b>	-
<b>Sub-flow</b>	-
<b>Post-Condition</b>	<ol style="list-style-type: none"> <li>1. Sistem melanjutkan alur pelaporan ke tahap selanjutnya.</li> <li>2. Sistem tidak melanjutkan alur pelaporan ke tahap selanjutnya.</li> </ol>

<b>Glossary</b>
Data pembayaran pajak meliputi tanggal pembayaran, jumlah pembayaran, ketetapan pajak, dan status pembayaran.

**F. Spesifikasi Use Case – Lihat laporan target dan realisasi pajak**

**Tabel 4.17 Spesifikasi Use case Lihat laporan target dan realisasi pajak**

<b>Name</b>	Lihat laporan target dan realisasi pajak
<b>Actor</b>	Bidang perencanaan
<b>Brief Description</b>	Use case ini mendeskripsikan tentang bagaimana bidang perencanaan melihat laporan target dan realisasi wajib pajak.
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Bidang perencanaan menggunakan perangkat yang terhubung dengan jaringan internet.</li> <li>2. Bidang perencanaan telah berhasil masuk ke sistem.</li> </ol>
<b>Basic Flow / flow of events</b>	<p><b>{Mengisi data filter}</b></p> <ol style="list-style-type: none"> <li>1. Bidang perencanaan mengisi data filter untuk menampilkan data laporan target dan realisasi.</li> </ol> <p><b>{Lihat laporan perencanaan}</b></p> <ol style="list-style-type: none"> <li>2. Sistem menampilkan menu laporan perencanaan sesuai filter yang diisi oleh bidang perencanaan.</li> <li>3. Bidang perencanaan melihat laporan target dan realisasi pajak.</li> </ol>

	<b>{Use case selesai}</b> 4. <i>Use case</i> selesai.
<b>Alternative Flows</b>	-
<b>Sub-flow</b>	-
<b>Post-Condition</b>	1. Sistem menampilkan laporan target dan realisasi pajak. 2. Sistem tidak menampilkan laporan target dan realisasi pajak yang telah dibuat.

<b>Glossary</b>
Data laporan target dan realisasi pajak terdiri dari tahun, rencana pendapatan, target pendapatan, dan pendapatan pajak tahun sebelumnya.

### G. Spesifikasi Use Case – Pantau pembayaran pajak

Tabel 4.18 Spesifikasi Use case Pantau pembayaran pajak

<b>Name</b>	Pantau pembayaran pajak
<b>Actor</b>	Bidang perencanaan
<b>Brief Description</b>	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor (bidang perencanaan) memantau pembayaran pajak dari wajib pajak.
<b>Pre-Condition</b>	1. Bidang perencanaan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Bidang perencanaan telah berhasil masuk ke sistem.
<b>Basic Flow / flow of events</b>	<p><b>{Mengisi formulir data wajib pajak}</b></p> <ol style="list-style-type: none"> <li>1. Bidang perencanaan memilih menu <i>monitoring</i> pembayaran.</li> <li>2. Sistem menampilkan menu <i>monitoring</i> pembayaran.</li> <li>3. Bidang perencanaan mengisi data wajib pajak yang ingin ditampilkan pembayarannya.</li> </ol> <p><b>{Menampilkan data realisasi pembayaran}</b></p> <ol style="list-style-type: none"> <li>4. Sistem akan menampilkan rencana target dan realisasi pembayaran pajak oleh wajib pajak yang diinginkan.</li> </ol> <p><b>{Mengisi catatan pembayaran}</b></p> <ol style="list-style-type: none"> <li>5. Bidang perencanaan mengisi catatan mengenai realisasi pembayaran wajib pajak.</li> </ol> <p><b>(Pilih simpan catatan)</b></p> <ol style="list-style-type: none"> <li>6. Bidang perencanaan memilih menyimpan catatan.</li> </ol> <p><b>{Menyimpan data}</b></p> <ol style="list-style-type: none"> <li>7. Sistem menyimpan data catatan realisasi pembayaran pajak oleh bidang perencanaan.</li> </ol> <p><b>{Use case selesai}</b></p> <ol style="list-style-type: none"> <li>8. <i>Use case</i> selesai</li> </ol>

<b>Alternative Flows</b>	<p>A1. Penanganan pelanggaran</p> <ol style="list-style-type: none"> <li>1. Pembatalan pembuatan catatan Saat sedang mengisi kolom catatan pembayaran, bidang perencanaan tidak melakukan penyimpanan atau keluar dari menu <i>monitoring</i> pembayaran.             <ol style="list-style-type: none"> <li>a. Sistem akan menampilkan notifikasi apakah aktor benar-benar ingin membatalkan pembuatan catatan pembayaran.</li> <li>b. Bidang perencanaan memilih membatalkan pada jendela notifikasi.</li> </ol> </li> <li>2. Melanjutkan <i>use case</i> <b>{Use case selesai}</b></li> </ol>
<b>Sub-flow</b>	-
<b>Post-Condition</b>	1. Catatan pembayaran wajib pajak tersimpan.

<b>Glossary</b>
Data wajib pajak yang diisi sebelum menampilkan data pembayarannya adalah NPWP, jenis pajak, dan tahun.

#### H. Spesifikasi Use Case – Laporkan pembayaran pajak

Tabel 4.19 Spesifikasi Use Case Laporkan pembayaran pajak

<b>Name</b>	Laporkan pembayaran pajak
<b>Actor</b>	Wajib Pajak
<b>Brief Description</b>	Use case ini mendeskripsikan tentang bagaimana aktor (wajib pajak) melaporkan pembayaran pajak.
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Wajib pajak menggunakan perangkat yang terhubung dengan jaringan internet.</li> <li>2. Wajib pajak telah berhasil masuk ke sistem.</li> </ol>
<b>Basic Flow / flow of events</b>	<p><b>{Pilih tambah laporan pembayaran}</b></p> <ol style="list-style-type: none"> <li>1. Dimulai saat wajib pajak memilih menu pembayaran.</li> <li>2. Sistem menampilkan daftar pembayaran.</li> <li>3. Wajib pajak memilih tambah laporan pembayaran.</li> </ol> <p><b>{Menampilkan formulir laporan}</b></p> <ol style="list-style-type: none"> <li>4. Sistem menampilkan alur pelaporan pembayaran pajak</li> <li>5. Sistem akan menampilkan formulir pembayaran pajak.</li> </ol> <p><b>{Mengisi formulir pembayaran}</b></p> <ol style="list-style-type: none"> <li>6. Wajib pajak mengisi formulir pelaporan pembayaran pajak.</li> </ol> <p><b>{Memilih selanjutnya}</b></p> <ol style="list-style-type: none"> <li>7. Wajib pajak memilih melakukan tahapan selanjutnya.</li> </ol> <p><b>{Unggah bukti pembayaran}</b></p> <ol style="list-style-type: none"> <li>8. Sistem menampilkan formulir unggah foto bukti pembayaran pajak.</li> </ol>



	<p>9. Wajib pajak mengunggah foto bukti pembayaran pajak.  <b>{Memilih selanjutnya}</b></p> <p>10. Wajib pajak memilih melakukan tahapan selanjutnya.  <b>{Menyimpan data}</b></p> <p>11. Sistem menyimpan/menambahkan foto dan laporan pembayaran pajak oleh aktor.  <b>{Use case selesai}</b></p> <p>12. <i>Use case</i> selesai</p>
<b>Alternative Flows</b>	<p>A1. Penanganan kesalahan pengisian formulir</p> <ol style="list-style-type: none"> <li>1. Jika terdapat kolom formulir yang belum diisi Saat wajib pajak melakukan <b>{memilih selanjutnya}</b> dan formulir tidak diisi dengan lengkap. <ol style="list-style-type: none"> <li>a. Sistem akan meminta melengkapi formulir registrasi.</li> </ol> </li> <li>2. Melanjutkan <i>use case</i> dari <b>{Mengisi formulir pembayaran}</b></li> </ol>
<b>Sub-flow</b>	-
<b>Post-Condition</b>	<ol style="list-style-type: none"> <li>1. Sistem melanjutkan alur pelaporan ke tahap selanjutnya.</li> <li>2. Sistem tidak melanjutkan alur pelaporan ke tahap selanjutnya.</li> </ol>

**Glossary**

Data pelaporan pembayaran pajak yang dimaksud adalah nama wajib pajak, NPWP, nama usaha, tanggal pembayaran, ketentuan pajak, nominal yang dibayarkan, dan foto bukti pembayaran.

**I. Spesifikasi Use Case - Cetak data pembayaran pajak**

**Tabel 4.20 Spesifikasi Use Case Cetak data pembayaran pajak**

<b>Name</b>	Cetak data pembayaran pajak
<b>Brief Description</b>	<i>Use case</i> ini mendeskripsikan tentang bagaimana mencetak laporan data pembayaran pajak
<b>Actor</b>	Bidang perencanaan, bidang pembukuan, Bendahara Umum Daerah
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Bendahara umum daerah dan bidang pembukuan menggunakan perangkat yang terhubung dengan jaringan internet.</li> <li>2. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan telah berhasil masuk ke sistem.</li> </ol>
<b>Basic Flow / flow of events</b>	<b>{Memilih cetak laporan pembayaran}</b>

	<ol style="list-style-type: none"> <li>1. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan memilih menu data pembayaran pajak.</li> <li>2. Sistem menampilkan data pembayaran pajak.</li> <li>3. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan memilih cetak data pembayaran pajak.</li> </ol> <p><b>{Tampilan <i>preview print</i>}</b></p> <ol style="list-style-type: none"> <li>4. Muncul <i>preview</i> data pembayaran pajak.</li> </ol> <p><b>{Memilih <i>print</i>}</b></p> <ol style="list-style-type: none"> <li>5. Bendahara umum daerah atau bidang pembukuan atau bidang perencanaan memilih <i>print</i></li> </ol> <p><b>{Use case selesai}</b></p> <ol style="list-style-type: none"> <li>6. <i>Use case</i> selesai</li> </ol>
<b>Alternative Flow</b>	-
<b>Sub-flow</b>	-
<b>Post-Condition</b>	1. Laporan penerimaan pajak tercetak

**Glossary**

Data pembayaran pajak meliputi tanggal pembayaran, jumlah pembayaran, ketetapan pajak, dan status pembayaran.

**J. Spesifikasi Use Case - Cetak laporan rencana dan target pajak**

**Tabel 4.21 Spesifikasi Use Case Cetak laporan rencana dan target pajak**

<b>Name</b>	Cetak laporan penerimaan pajak
<b>Brief Description</b>	<i>Use case</i> ini mendeskripsikan tentang bagaimana mencetak laporan rencana dan target pajak
<b>Actor</b>	Bidang perencanaan
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Bidang perencanaan menggunakan perangkat yang terhubung dengan jaringan internet.</li> <li>2. Bidang perencanaan telah berhasil masuk ke sistem.</li> </ol>
<b>Basic Flow / flow of events</b>	<p><b>{Memilih cetak laporan pembayaran}</b></p> <ol style="list-style-type: none"> <li>1. Bidang perencanaan memilih menu laporan perencanaan.</li> <li>2. Sistem menampilkan data laporan perencanaan pajak.</li> <li>3. Aktor memilih cetak laporan perencanaan pajak</li> </ol> <p><b>{Tampilan <i>preview print</i>}</b></p> <ol style="list-style-type: none"> <li>4. Muncul <i>preview</i> laporan perencanaan pajak</li> </ol> <p><b>{Memilih <i>print</i>}</b></p> <ol style="list-style-type: none"> <li>5. Aktor memilih <i>print</i></li> </ol> <p><b>{Use case selesai}</b></p> <ol style="list-style-type: none"> <li>6. <i>Use case</i> selesai</li> </ol>
<b>Alternative Flow</b>	-



<b>Sub-flow</b>	-
<b>Post-Condition</b>	1. Laporan perencanaan dan target pajak tercetak.

<b>Glossary</b>
Data laporan rencana dan target pajak terdiri dari tahun, rencana pendapatan, target pendapatan, dan pendapatan pajak tahun sebelumnya.

### K. Spesifikasi Use Case – Login

Tabel 4.22 Spesifikasi Use Case Login

<b>Name</b>	<i>Login</i>
<b>Brief Description</b>	Use case ini mendeskripsikan tentang bagaimana pengguna sistem masuk ke dalam sistem dengan jenis pengguna yang berbeda-beda sehingga dapat menggunakan fungsi/fitur yang disediakan oleh sistem.
<b>Actor</b>	Bidang perencanaan, wajib pajak, bidang pembukuan, Bendahara Umum Daerah
<b>Pre-Condition</b>	1. Bidang perencanaan atau wajib pajak atau bidang pembukuan atau Bendahara Umum Daerah menggunakan perangkat yang terhubung dengan jaringan internet.
<b>Basic Flow / flow of events</b>	<p><b>{Meminta data login}</b></p> <ol style="list-style-type: none"> <li>1. Use case dimulai saat bidang perencanaan atau wajib pajak atau bidang pembukuan atau Bendahara Umum Daerah membuka sistem.</li> <li>2. Sistem meminta data <i>login</i>.</li> </ol> <p><b>{Mengisi formulir login}</b></p> <ol style="list-style-type: none"> <li>3. Bidang perencanaan atau wajib pajak atau bidang pembukuan atau Bendahara Umum Daerah mengisi formulir log in sistem</li> </ol> <p><b>{Melakukan login}</b></p> <ol style="list-style-type: none"> <li>4. Bidang perencanaan atau wajib pajak atau bidang pembukuan atau Bendahara Umum Daerah melakukan log in ke sistem</li> </ol> <p><b>{Melakukan validasi akun}</b></p> <ol style="list-style-type: none"> <li>5. Sistem melakukan validasi akun dengan memeriksa <i>username</i> dan <i>password</i> yang dimasukkan oleh bidang perencanaan atau wajib pajak atau bidang pembukuan atau Bendahara Umum Daerah.</li> <li>6. Sistem mengkonfirmasi <i>username</i> dan <i>password</i> yang diisikan valid.</li> </ol> <p><b>{Menampilkan notifikasi}</b></p> <ol style="list-style-type: none"> <li>7. Sistem menampilkan pesan berhasil masuk ke sistem</li> </ol> <p><b>{Use case selesai}</b></p>

	8. Dapat melakukan <i>use case</i> sesuai jenis pengguna
<b>Alternative Flow</b>	<p>A1. Penanganan kesalahan pengisian formulir</p> <ol style="list-style-type: none"> <li>1. <i>Username</i> dan <i>password</i> pengguna tidak valid Pada saat {melakukan validasi akun} <i>username</i> dan <i>password</i> tidak sesuai dengan yang terdaftar di sistem.             <ol style="list-style-type: none"> <li>a. Sistem akan meminta untuk memasukkan <i>username</i> dan <i>password</i> lagi.</li> </ol> </li> <li>2. Melanjutkan <i>use case</i> dari {<b>Mengisi formulir login</b>}</li> </ol> <p>A2. Penanganan pelanggaran</p> <ol style="list-style-type: none"> <li>1. Pembatalan <i>login</i> Saat sedang {mengisi formulir <i>login</i>}, aktor membatalkan <i>login</i> atau menutup program.</li> <li>2. Melanjutkan <i>use case</i> {<b>Use case selesai</b>}</li> </ol>
<b>Sub-flow</b>	-
<b>Post-Condition</b>	<ol style="list-style-type: none"> <li>1. Aktor dapat masuk ke dalam sistem.</li> <li>2. Aktor dapat melakukan <i>use case</i> lain sesuai autentifikasi pengguna.</li> </ol>

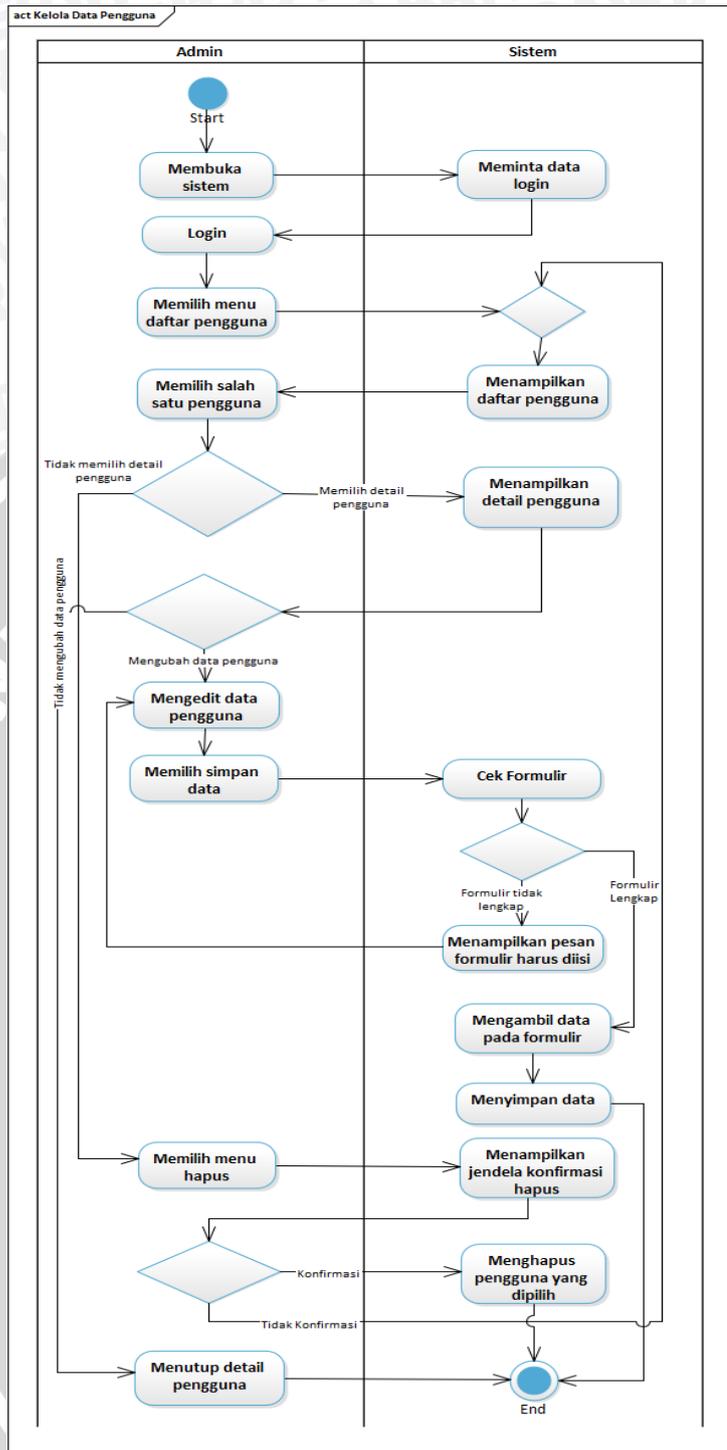
<b>Glossary</b>
Proses <i>login</i> meminta pengguna untuk memasukkan data berupa <i>username</i> dan <i>password</i>

#### 4.7 Diagram Aktivitas

Diagram aktivitas merupakan diagram yang menggambarkan alur dari aktivitas yang berhubungan dengan sistem sesuai dengan *use case* yang telah dibuat.

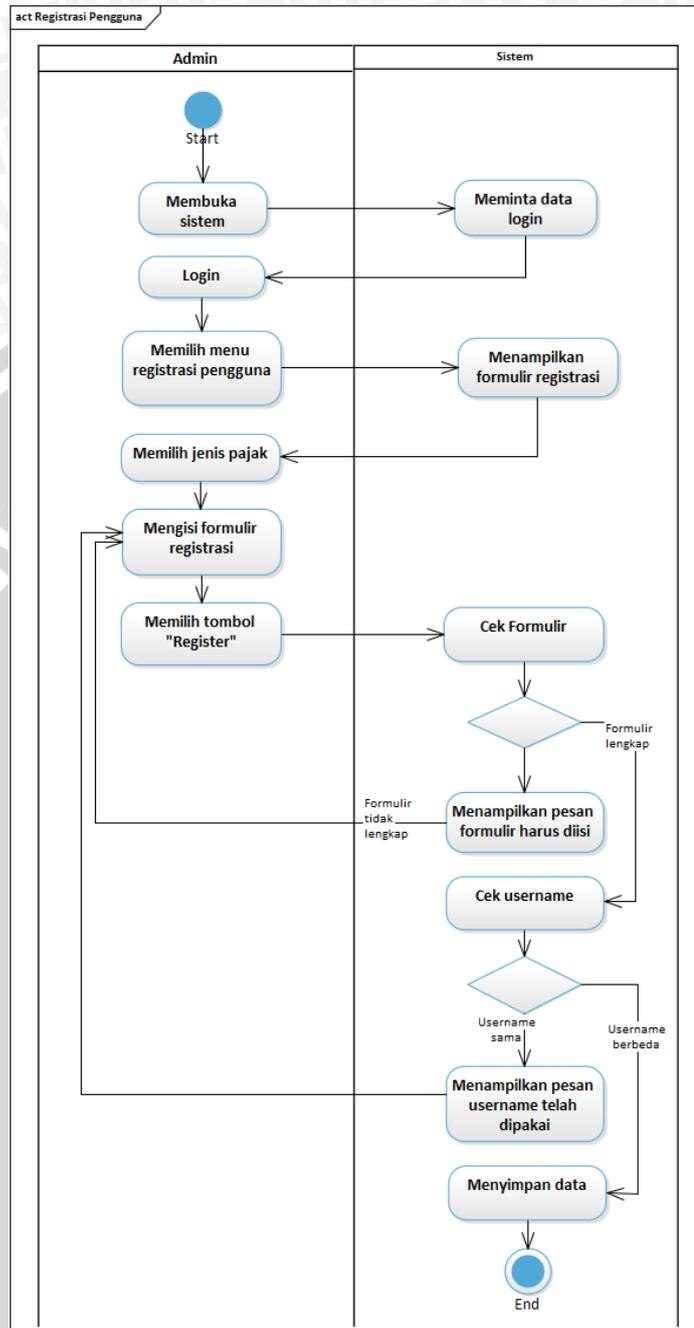
##### A. Diagram aktivitas – *Kelola Data Pengguna*

Aktivitas yang berjalan pada *use case* *Kelola Data Pengguna* dapat dilihat pada Gambar 4.4. Aktivitas tersebut menggambarkan proses pengelolaan data pengguna yang diawali oleh admin dengan melakukan *login* dan memilih menu daftar pengguna. Proses yang digambarkan mengikuti proses *basic flow* dan *alternative flow* pada deskripsi *use case*.



Gambar 4.4 Diagram Aktivitas Kelola data pengguna

B. Diagram aktivitas – Registrasi Pengguna



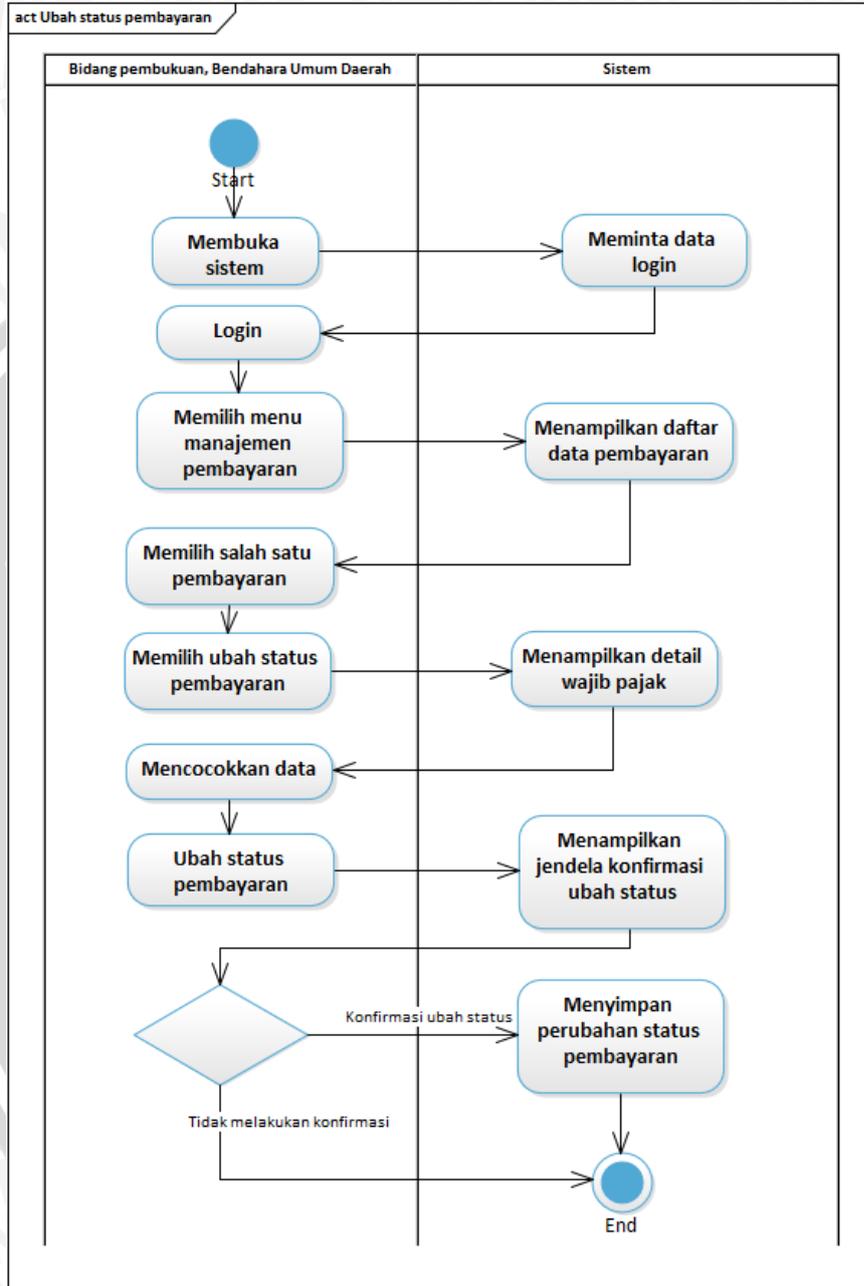
Gambar 4.5 Diagram Aktivitas Registrasi Pengguna

Pada Gambar 4.5 merupakan diagram aktivitas dari *use case* Registrasi Pengguna. Diagram tersebut menggambarkan alur yang dilakukan saat admin melakukan proses registrasi pengguna sesuai dengan spesifikasi *use case* yang telah dibuat.



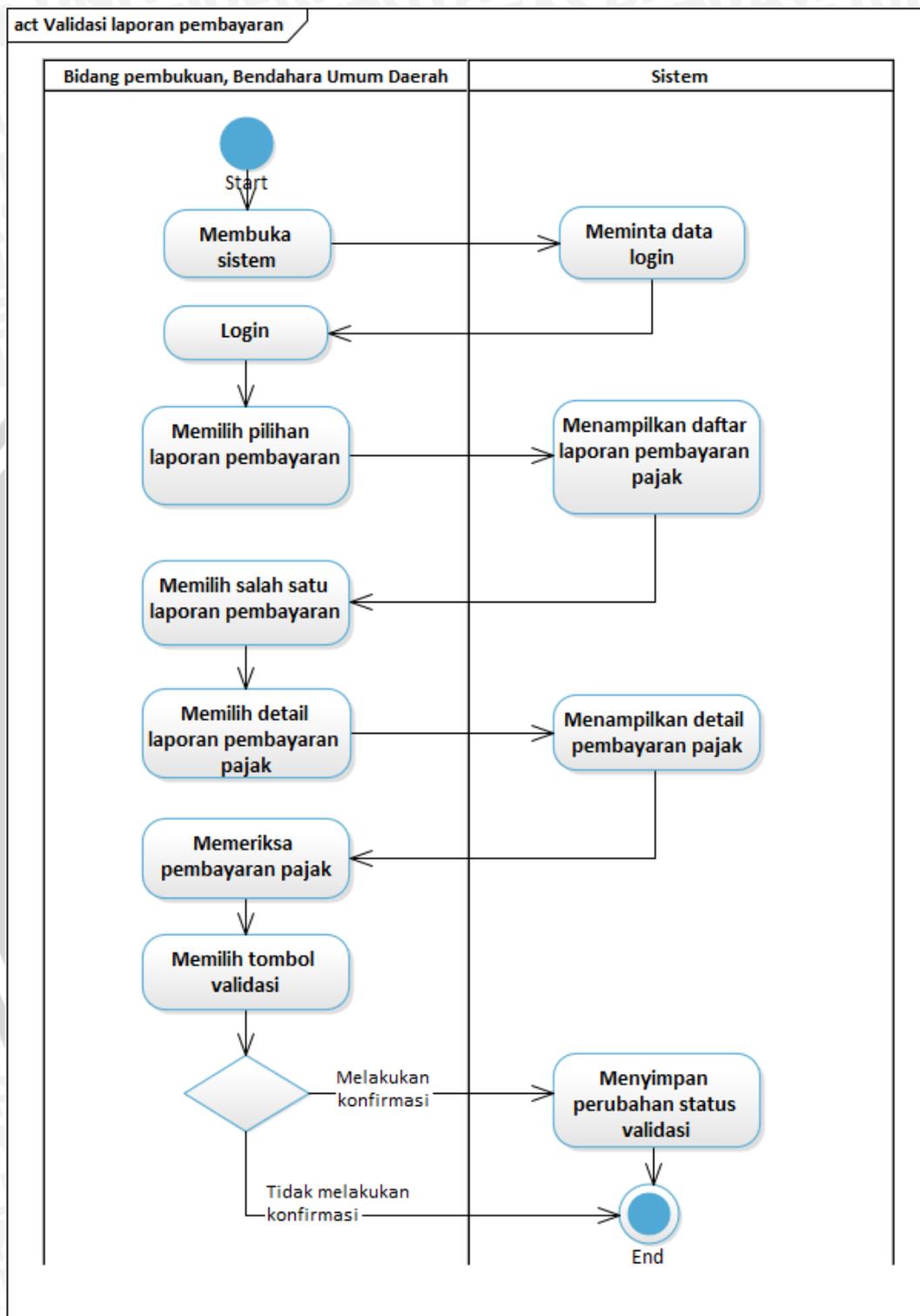
C. Diagram aktivitas – *Ubah Status Pembayaran*

Aktivitas yang berjalan pada *use case* Ubah Status Pembayaran dapat dilihat pada Gambar 4.6. Aktivitas tersebut menggambarkan proses perubahan status pembayaran pajak.



Gambar 4.6 Diagram Aktivitas Ubah Status Pembayaran

D. Diagram aktivitas – *Validasi Laporan Pembayaran*



**Gambar 4.7 Diagram Aktivitas Validasi Laporan Pembayaran**

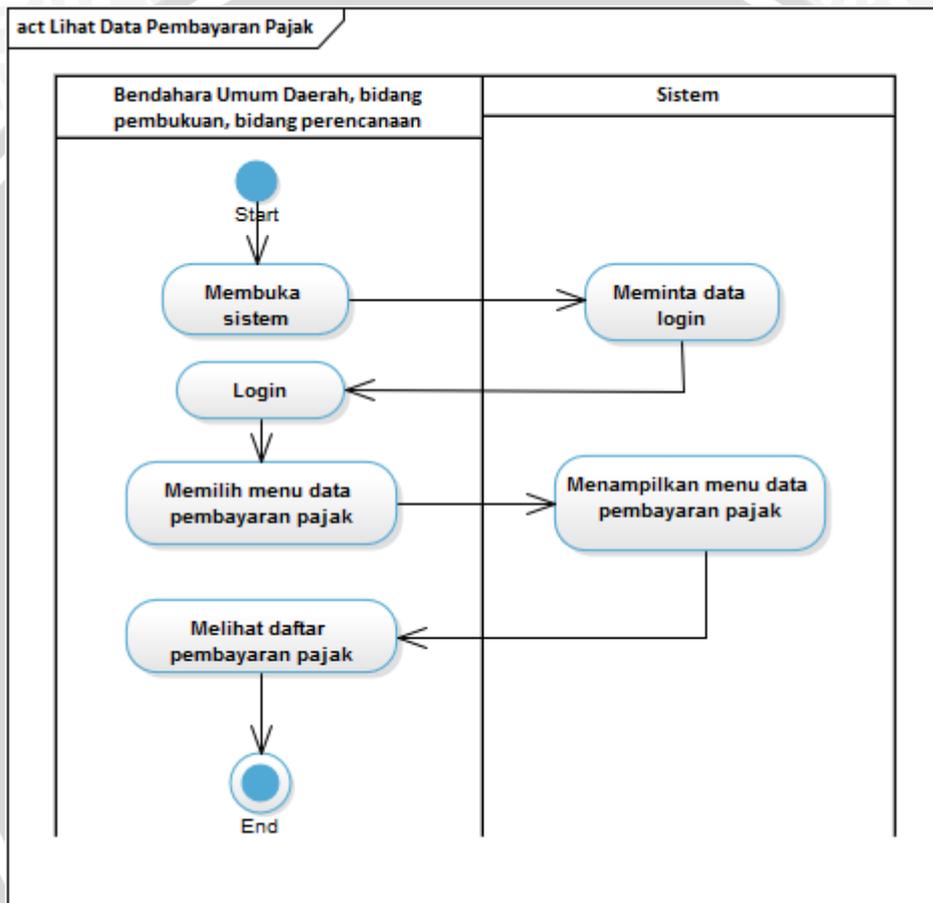
Pada Gambar 4.7 merupakan diagram aktivitas dari *use case* Registrasi Validasi Laporan Pembayaran. Diagram tersebut menggambarkan alur yang dilakukan saat



bidang pembukuan atau Bendahara Umum Daerah melakukan proses validasi laporan pembayaran wajib pajak sesuai dengan spesifikasi *use case* yang telah dibuat.

**E. Diagram aktivitas – *Lihat data pembayaran pajak***

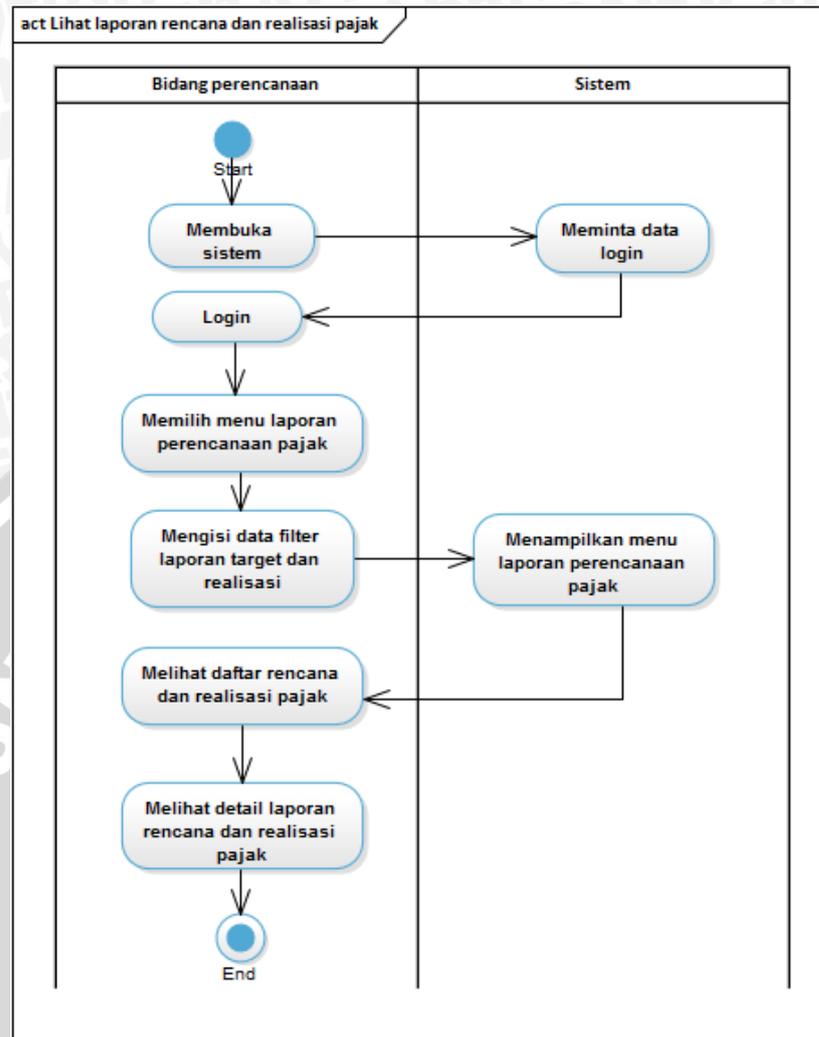
Aktivitas yang berjalan pada *use case* Lihat Data Pembayaran Pajak dapat dilihat pada Gambar 4.8. Aktivitas tersebut menggambarkan proses saat Bendahara Umum Daerah, bidang pembukuan, dan bidang perencanaan melihat data pembayaran pajak yang ada sesuai dengan spesifikasi *use case* yang telah dibuat.



**Gambar 4.8 Diagram Aktivitas Lihat data pembayaran pajak**

**F. Diagram aktivitas – *Lihat laporan target dan realisasi pajak***

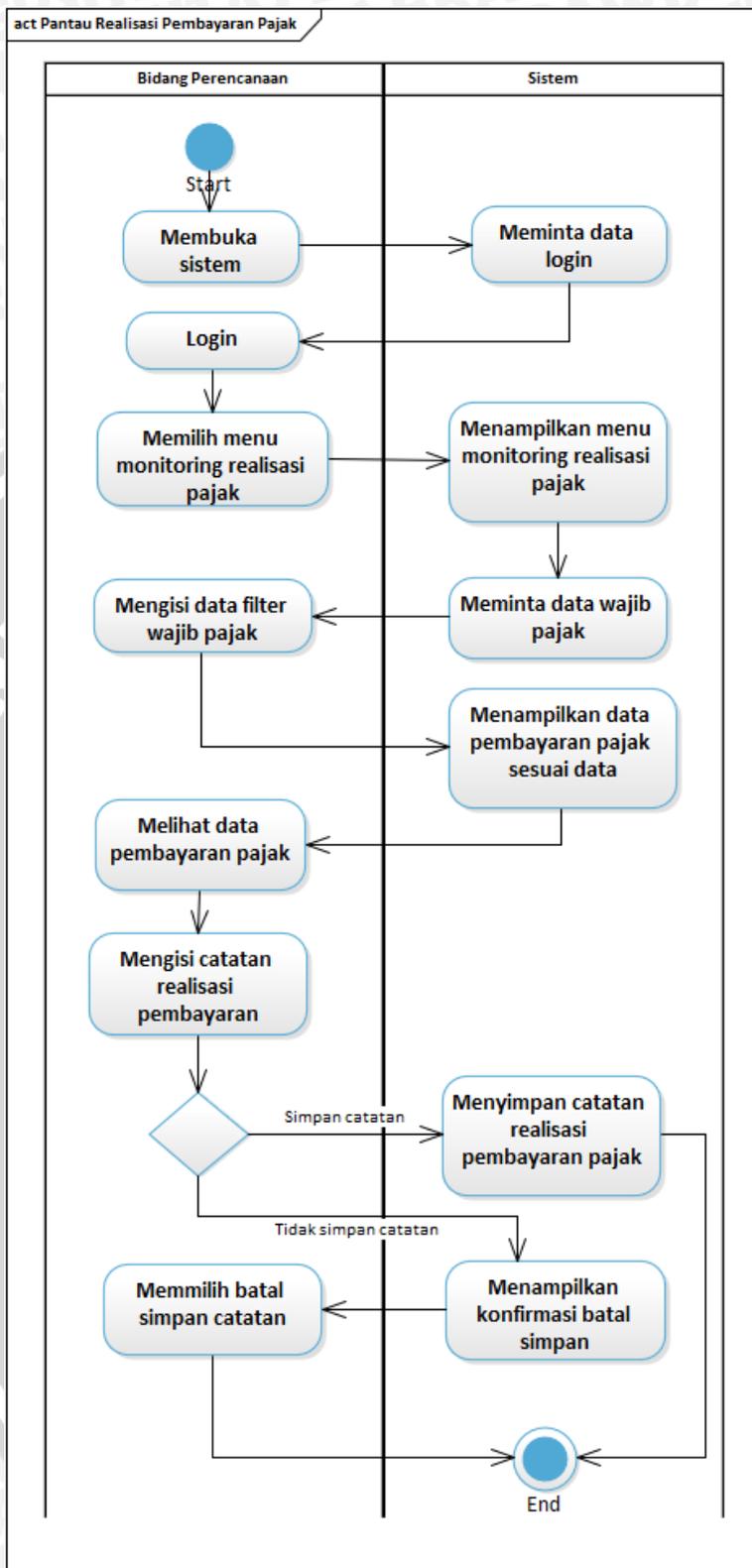
Aktivitas yang berjalan pada *use case* Lihat Laporan Target dan Realisasi Pajak dapat dilihat pada Gambar 4.9. Aktivitas tersebut menggambarkan proses saat bidang perencanaan melihat laporan dari target dan realisasi pajak yang ada sesuai dengan spesifikasi *use case* yang telah dibuat.



Gambar 4.9 Diagram Aktivitas Lihat laporan target dan realisasi pajak

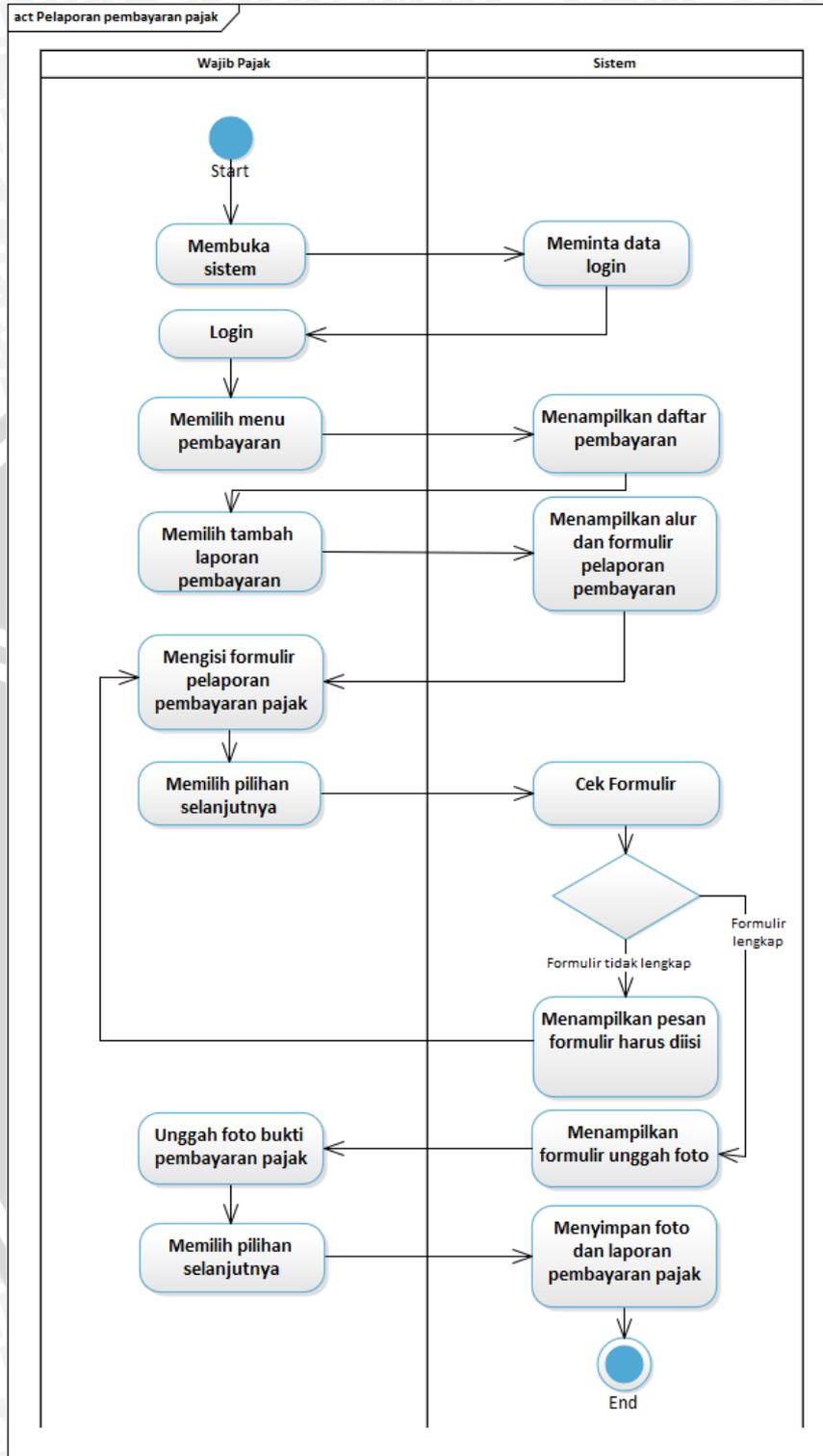
G. Diagram aktivitas – Pantau pembayaran pajak

Pada Gambar 4.10 merupakan diagram aktivitas dari use case Pantau Pembayaran Pajak. Diagram tersebut menggambarkan alur yang dilakukan saat bidang perencanaan akan memantu pembayaran pajak yang dilakukan oleh wajib pajak sesuai dengan spesifikasi use case yang telah dibuat sebelumnya.



Gambar 4.10 Diagram Aktivitas Pantau Pembayaran Pajak

H. Diagram aktivitas – Laporkan pembayaran pajak



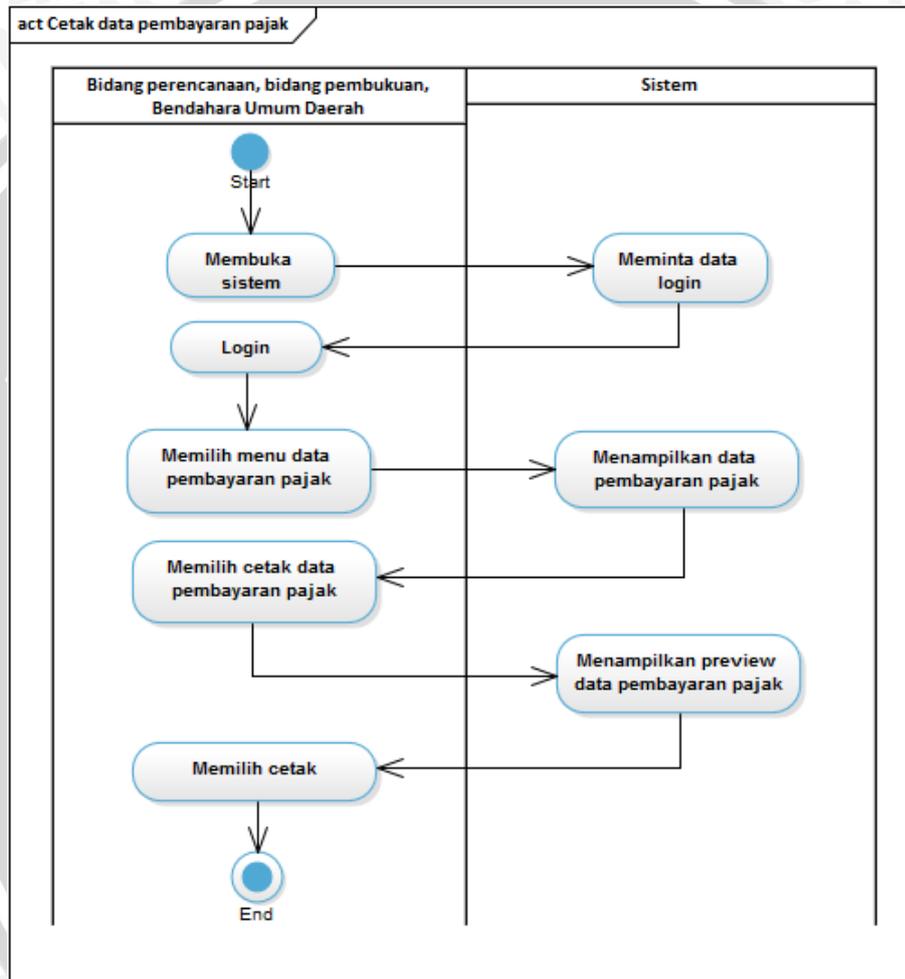
Gambar 4.11 Diagram Aktivitas Case Laporkan Pembayaran Pajak

Pada Gambar 4.11 merupakan diagram aktivitas dari use case Pelaporan Pembayaran Pajak yang dilakukan oleh aktor wajib pajak. Diagram tersebut

menggambarkan proses pembuatan laporan pembayaran pajak sesuai dengan spesifikasi *use case* yang telah dibuat.

**I. Diagram aktivitas – Cetak data pembayaran pajak**

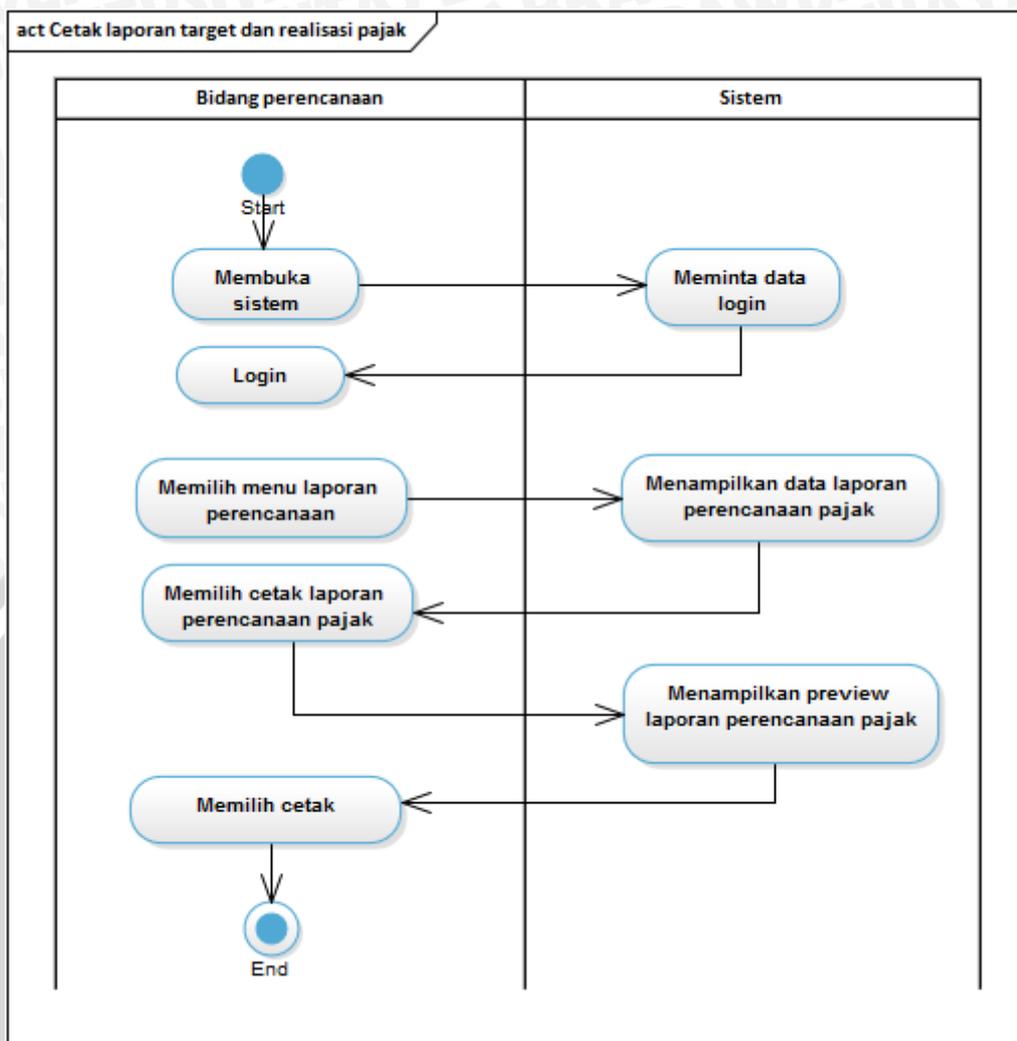
Diagram aktivitas Cetak Data Pembayaran Pajak merupakan diagram aktivitas yang menggambarkan alur proses mencetak data pembayaran pajak yang dapat dilakukan oleh aktor bidang perencanaan, bidang pembukuan, dan Bendahara Umum Daerah. Diagram aktivitas tersebut dapat dilihat pada Gambar 4.12.



**Gambar 4.12 Diagram Aktivitas Cetak data pembayaran pajak**

**J. Diagram aktivitas – Cetak laporan target dan realisasi pajak**

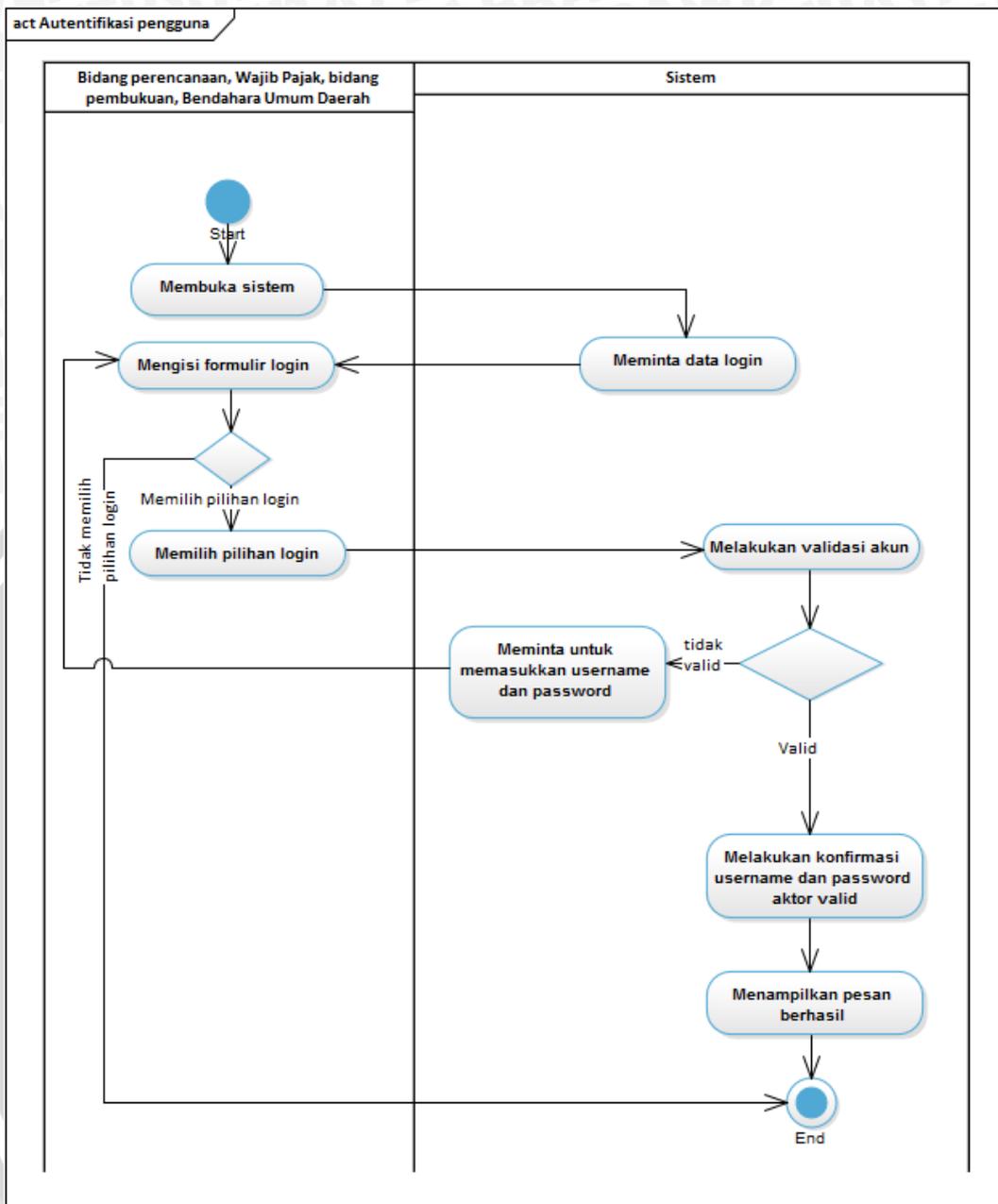
Aktivitas yang berjalan pada *use case* Cetak Laporan Target dan Realisasi Pajak dapat dilihat pada Gambar 4.13. Diagram tersebut menggambarkan aktivitas saat bidang perencanaan akan mencetak laporan dari target dan realisasi pajak yang ada sesuai dengan spesifikasi *use case* yang telah dibuat.



Gambar 4.13 Diagram Aktivitas Cetak laporan dan target pajak

#### K. Diagram aktivitas – Login

Aktivitas yang berjalan pada *use case Login* dapat dilihat pada Gambar 4.14. Diagram tersebut menggambarkan aktivitas yang dilakukan saat semua aktor akan melakukan proses *login* sesuai dengan spesifikasi *use case* yang telah dibuat.

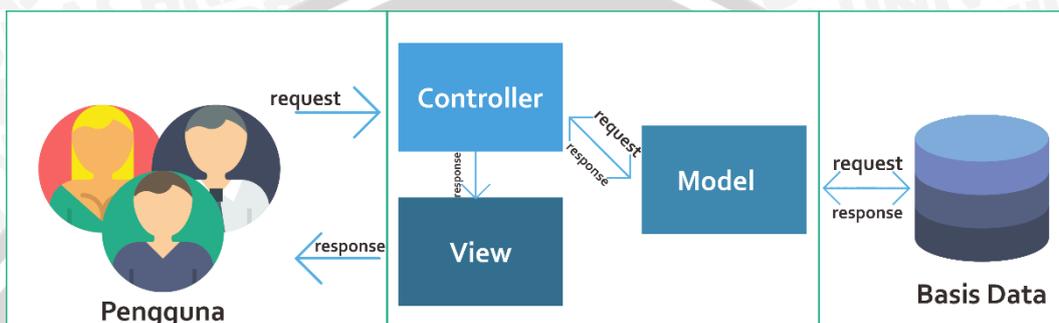


Gambar 4.14 Diagram Aktivitas Login

## BAB 5 PERANCANGAN

### 5.1 Analisis Arsitektur

Bagian ini menggambarkan arsitektur dari sistem pelaporan pembayaran pajak yang disesuaikan dengan prinsip desain pola arsitektur *Model View Controller* (MVC). Sistem pelaporan pembayaran pajak dirancang berbasis web.



**Gambar 5.1 Arsitektur Sistem**

Pada Gambar 5.1, menjelaskan arsitektur dari sistem pelaporan pembayaran pajak. Saat pengguna melakukan sebuah aksi pada sistem untuk meminta suatu data, maka *controller* akan memproses permintaan data dan fungsi pada *controller* akan dijalankan sesuai permintaan data. *Controller* akan memanggil fungsi yang ada pada *model* untuk memproses permintaan khusus dan mengambil data pada basis data. Jika data sudah diterima maka akan dikirimkan ke *view* untuk kemudian dikirimkan ke *web browser* untuk dilihat.

### 5.2 Analisis Use Case

#### 5.2.1 Kelas-kelas Analisis

Kelas-kelas analisis (*Analysis Classes*) menggambarkan model konseptual awal untuk “hal pada sistem yang memiliki tingkah laku dan tanggung jawab”. Pada tahapan ini dilakukan analisis berdasarkan *use case* dan spesifikasinya yang telah dibuat. Kelas-kelas analisis yang dibuat bisa dilihat pada Tabel 5.1.

Tabel 5.1 Kelas Analisis

Use Case	Kelas Analisis
<p>Kelola data pengguna</p>	
<p>Registrasi pengguna</p>	
<p>Ubah status pembayaran</p>	



Tabel 5.1 Kelas Analisis (lanjutan)

Use Case	Kelas Analisis
Validasi laporan pembayaran	<pre> classDiagram     class DaftarLaporan     class FormValidasi     class pembayaranController     class pembayaran     class ObjekPajak     DaftarLaporan --&gt; pembayaranController     FormValidasi --&gt; pembayaranController     pembayaranController --&gt; pembayaran     pembayaranController --&gt; ObjekPajak         </pre>
Lihat data pembayaran pajak	<pre> classDiagram     class FormDataKeuangan     class DataPembayaran     class pembayaranController     class WajibPajak     class ObjekPajak     FormDataKeuangan --&gt; pembayaranController     DataPembayaran --&gt; pembayaranController     pembayaranController --&gt; WajibPajak     pembayaranController --&gt; ObjekPajak         </pre>
Lihat laporan target dan realisasi pajak	<pre> classDiagram     class FormFilterLaporan     class LaporanPerencanaan;     class PerencanaanController     class ObjekPajak     class pembayaran     FormFilterLaporan --&gt; PerencanaanController     LaporanPerencanaan; --&gt; PerencanaanController     PerencanaanController --&gt; ObjekPajak     PerencanaanController --&gt; pembayaran         </pre>

Tabel 5.1 Kelas Analisis (lanjutan)

Use Case	Kelas Analisis
Pantau pembayaran pajak	<pre> classDiagram     class formDataWP     class Monitoring     class perencanaanController     class pembayaran     class ObjekPajak     perencanaanController --&gt; formDataWP     perencanaanController --&gt; Monitoring     perencanaanController --&gt; pembayaran     perencanaanController --&gt; ObjekPajak     </pre>
Laporkan pembayaran pajak	<pre> classDiagram     class AlurLaporan     class RiwayatLaporan     class FormPelaporan     class pembayaranController     class formUnggahFoto     class pengguna     class ObjekPajak     class WajibPajak     pembayaranController --&gt; AlurLaporan     pembayaranController --&gt; RiwayatLaporan     pembayaranController --&gt; FormPelaporan     pembayaranController --&gt; formUnggahFoto     pembayaranController --&gt; pengguna     pembayaranController --&gt; ObjekPajak     pembayaranController --&gt; WajibPajak     </pre>
Cetak data pembayaran pajak	<pre> classDiagram     class DataPembayaran     class FormDataKeuangan     class cetakDataPembayaran     class pembayaranController     class ObjekPajak     class WajibPajak     pembayaranController --&gt; DataPembayaran     pembayaranController --&gt; FormDataKeuangan     pembayaranController --&gt; cetakDataPembayaran     pembayaranController --&gt; ObjekPajak     pembayaranController --&gt; WajibPajak     </pre>

Tabel 5.1 Kelas Analisis (lanjutan)

Use Case	Kelas Analisis
Cetak laporan target dan realisasi pajak	<pre> classDiagram     class FormFilterLaporan     class LaporanPembayaran     class cetakLaporan     class PerencanaanController     class ObjekPajak     class pembayaran      FormFilterLaporan --&gt; PerencanaanController     LaporanPembayaran --&gt; PerencanaanController     cetakLaporan --&gt; PerencanaanController     ObjekPajak --&gt; PerencanaanController     pembayaran --&gt; PerencanaanController     </pre>
Login	<pre> classDiagram     class FormLogin     class mainControoler     class pengguna      FormLogin --&gt; mainControoler     pengguna --&gt; mainControoler     </pre>

5.2.2 Mekanisme Analisis

Bagian ini menjelaskan mekanisme dari kelas yang sudah dianalisis. Mekanisme yang ada pada penelitian ini adalah *persistency* dan *security*. *Persistency* menunjukkan mekanisme dari kelas yang berhubungan dengan data yang akan disimpan dan bisa diakses lagi, serta kelas tersebut memiliki kemampuan untuk menyediakan data selama sistem masih ada. Pemetaan mekanisme analisis dari setiap kelas ditunjukkan pada Tabel 5.2.

Tabel 5.2 Mekanisme Analisis

Kelas Analisis	Mekanisme Analisis
Pembayaran	<i>Persistency, security</i>
ObjekPajak	<i>Persistency, security</i>
WajibPajak	<i>Persistency, security</i>
Pengguna	<i>Persistency, security</i>
DaftarPengguna	None

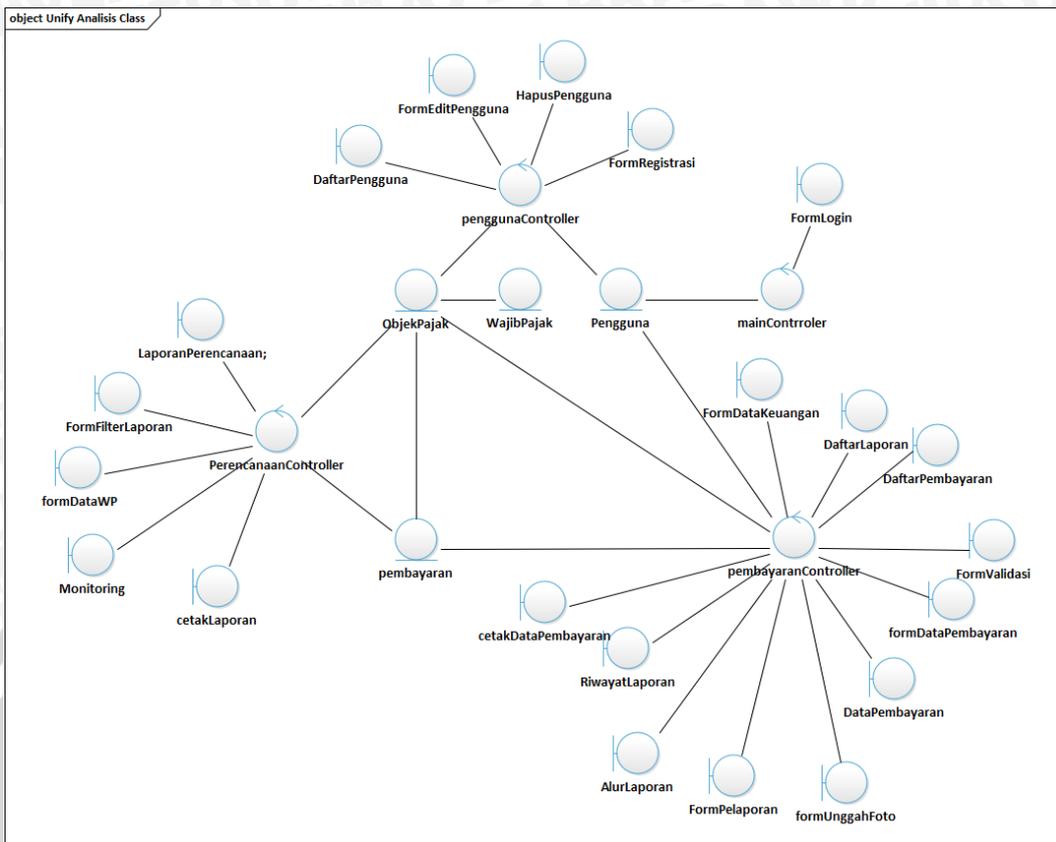


Tabel 5.2 Mekanisme Analisis (lanjutan)

Kelas Analisis	Mekanisme Analisis
FormEditPengguna	None
HapusPengguna	None
FormRegistrasi	None
formLogin	None
LaporanPerencanaan	None
FormFilterLaporan	None
formDataWP	None
Monitoring	None
cetakLaporan	None
formDataKeuangan	None
DaftarLaporan	None
DaftarPembayaran	None
formValidasi	None
formDataPembayaran	None
DataPembayaran	None
formUnggahFoto	None
formPelaporan	None
AlurLaporan	None
RiwayatLaporan	None
cetakDataPembayaran	None
PerencanaanController	None
penggunaController	None
pembayaranController	None
mainController	None

### 5.2.3 Penggabungan Kelas Analisis

Pada Gambar 5.2 menunjukkan kesatuan dari seluruh kelas yang telah dianalisis sebelumnya. Gambar tersebut menggambarkan hubungan dari setiap kelas yang telah dianalisis pada bagian kelas analisis.



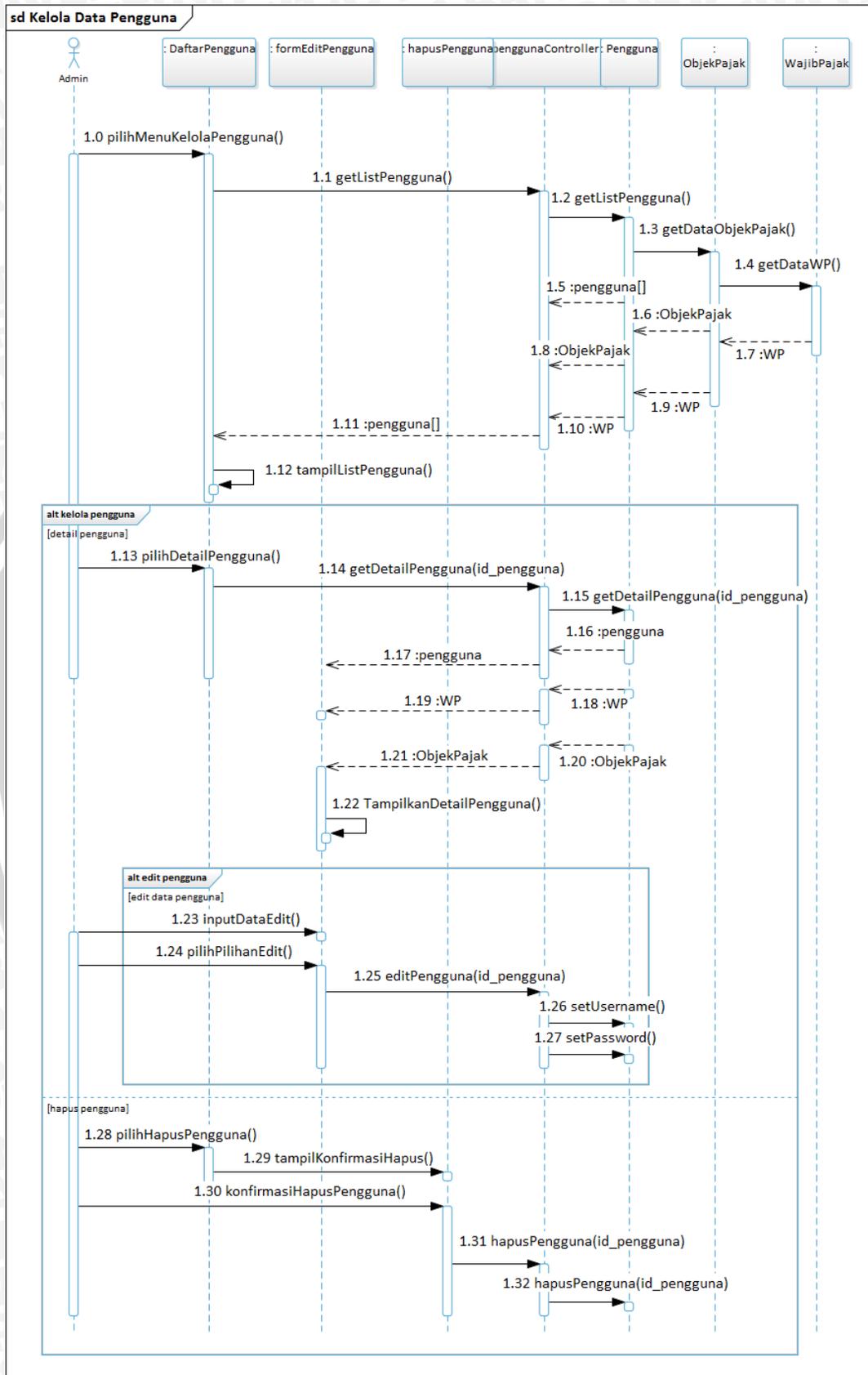
Gambar 5.2 Gabungan Kelas-kelas Analisis

### 5.3 Pemodelan Interaksi

#### 5.3.1 Sequence Diagram

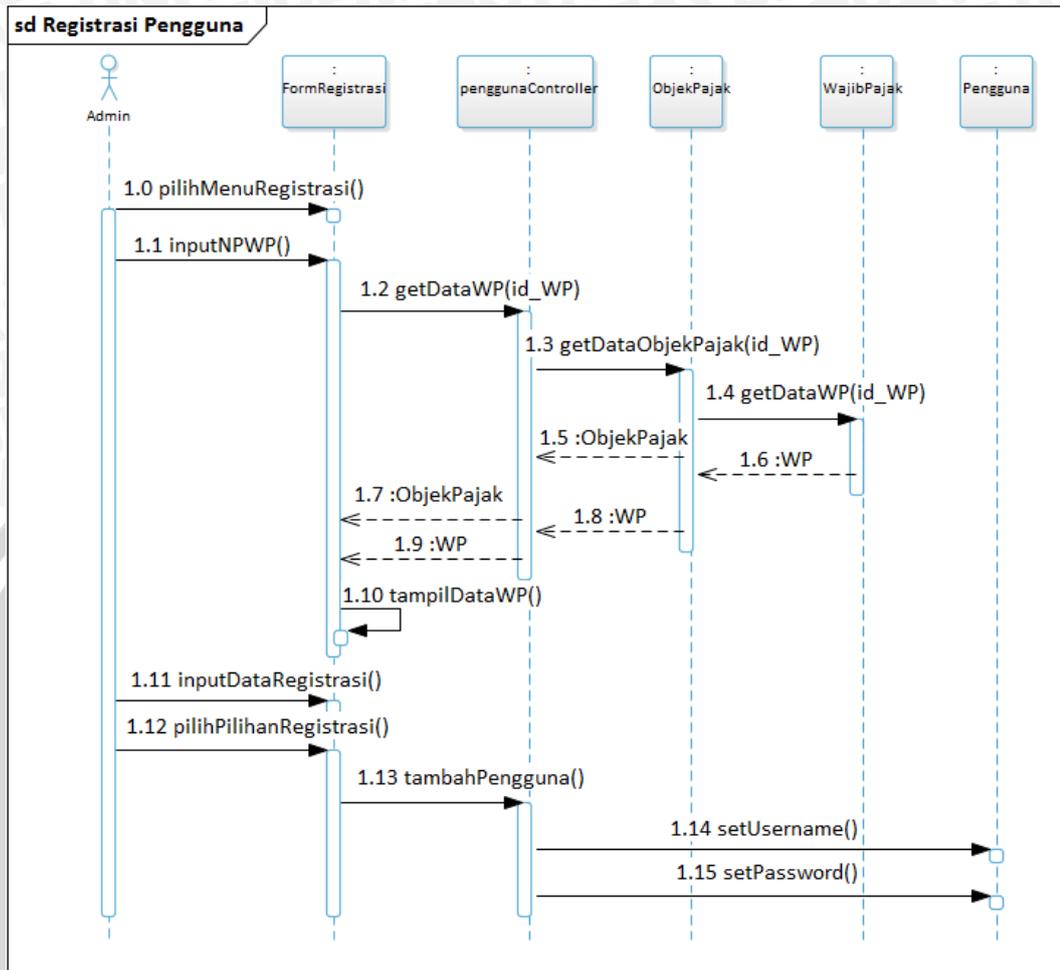
##### A. Sequence Diagram – Kelola Data Pengguna

Pada Gambar 5.3 menggambarkan interaksi pada proses kelola pengguna. Interaksi tersebut telah disesuaikan dengan spesifikasi *use case* Kelola Data Pengguna. Pada interaksi ini diawali oleh aktor admin membuka menu Kelola Pengguna yang akan menampilkan daftar pengguna yang didapatkan melalui fungsi *getListPengguna()* milik objek `:penggunaController`. Selanjutnya `:penggunaController` memanggil fungsi *getListPengguna* pada `Pengguna` yang dilanjutkan memanggil fungsi *getDataObjekPajak* dan *getDataWP* milik `:ObjekPajak` dan `:WajibPajak`. Kemudian semua data akan dikembalikan berupa daftar pengguna dalam bentuk *array* yaitu `pengguna[]` dan ditampilkan oleh `:DaftarPengguna` menggunakan fungsi *tampilListPengguna()*. Pada interaksi ini juga terdapat tiga alternatif interaksi yaitu detail pengguna, edit data pengguna, dan hapus pengguna.



Gambar 5.3 Sequence Diagram Kelola Data Pengguna

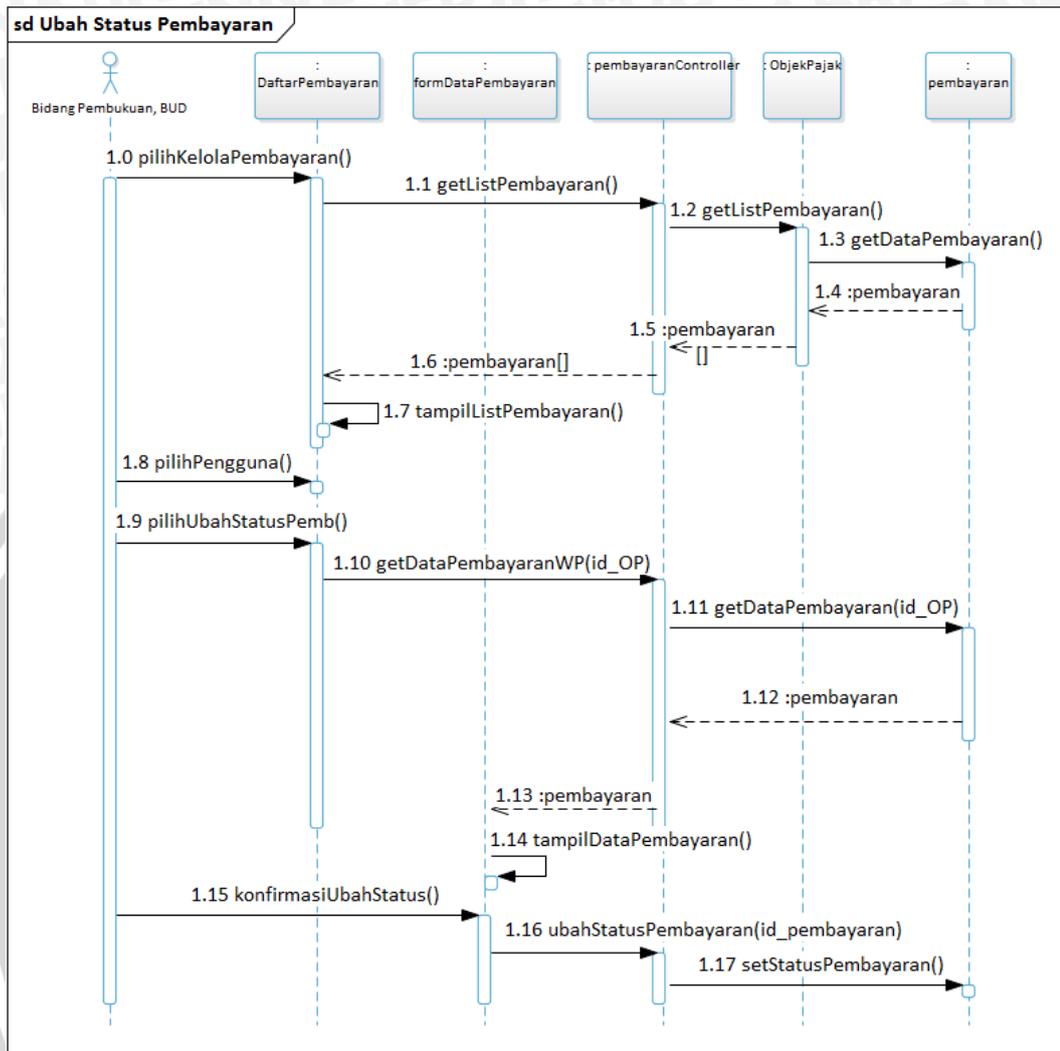
B. Sequence Diagram – Registrasi Pengguna



Gambar 5.4 Sequence Diagram Registrasi Pengguna

Interaksi pada proses Registrasi Pengguna digambarkan pada Gambar 5.4. Interaksi tersebut telah disesuaikan oleh spesifikasi *use case* Registrasi Pengguna dan diagram aktivitas Registrasi Pengguna. Pertama aktor (admin) memilih menu Registrasi Pengguna yang telah tersedia formulir data Wajib Pajak. Aktor mengisi NPWP yang merupakan *id\_WP*. Kemudian memanggil fungsi *getDataWP(id\_WP)* yang menggunakan parameter *id\_WP* pada objek *:penggunaController*. Kemudian dilanjutkan memanggil fungsi *getDataObjekPajak(id\_WP)* dan *getDataWP(id\_WP)* pada objek *:ObjekPajak* dan *:WajibPajak*. Data *ObjekPajak* dan *WP* kemudian akan dikembalikan untuk ditampilkan pada *:FormRegistrasi*. Kemudian aktor melanjutkan mengisi data registrasi berupa *username* dan *password*. Setelah aktor memilih pilihan Registrasi, maka fungsi *tambahPengguna* akan dijalankan dan fungsi *setUsername()* dan *setPassword()* pada objek *:Pengguna*.

C. Sequence Diagram – Ubah Status Pembayaran



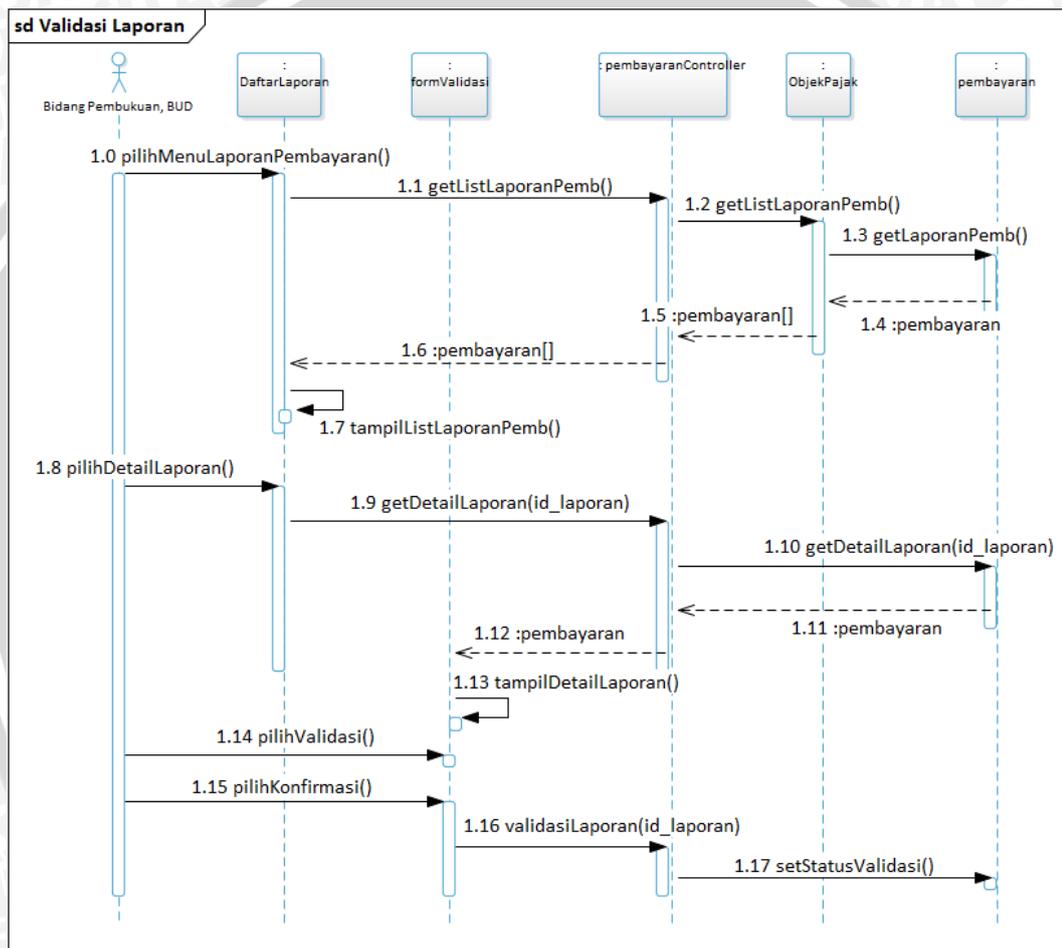
Gambar 5.5 Sequence Diagram Ubah Status Pembayaran

Pada Gambar 5.5 menggambarkan interaksi pada proses Ubah Status Pembayaran yang dilakukan oleh aktor Bidang Pembukuan dan BUD. Interaksi yang digambarkan telah disesuaikan dengan spesifikasi *use case* dan diagram aktivitas Ubah Status Pembayaran. Interaksi diawali saat aktor memilih menu Kelola Pembayaran yang kemudian memanggil fungsi *getListPembayaran()* pada objek *:pembayaranController*. Lalu dilanjutkan dengan memanggil fungsi *getListPembayaran()* pada *:ObjekPajak* dan diteruskan memanggil fungsi *getDataPembayaran()*. Kembalian data *pembayaran* dan daftar pembayaran *pembayaran[]* akan dikirimkan dan ditampilkan pada objek *:DaftarPembayaran*.

Setelah data pembayaran ditampilkan maka aktor akan memilih untuk mengubah status pembayaran dan memanggil fungsi *getDataPembayaranWP(id\_OP)* pada *:pembayaranController* dengan parameter *id\_OP* sebagai id dari wajib pajak. Kemudian dilanjutkan dengan fungsi yang sama

pada `:pembayaran`. Kemudian data pembayaran akan dikembalikan dan menampilkan data pembayaran dari wajib pajak yang dipilih. Kemudian memilih untuk konfirmasi ubah status pembayaran menggunakan fungsi `UbahStatusPembayaran(id_pembayaran)` pada `:pembayaranController` yang kemudian pada `:pembayaran` akan mengubah status pembayaran dengan fungsi `setStatusPembayaran()`.

#### D. Sequence Diagram – Validasi Laporan Pembayaran



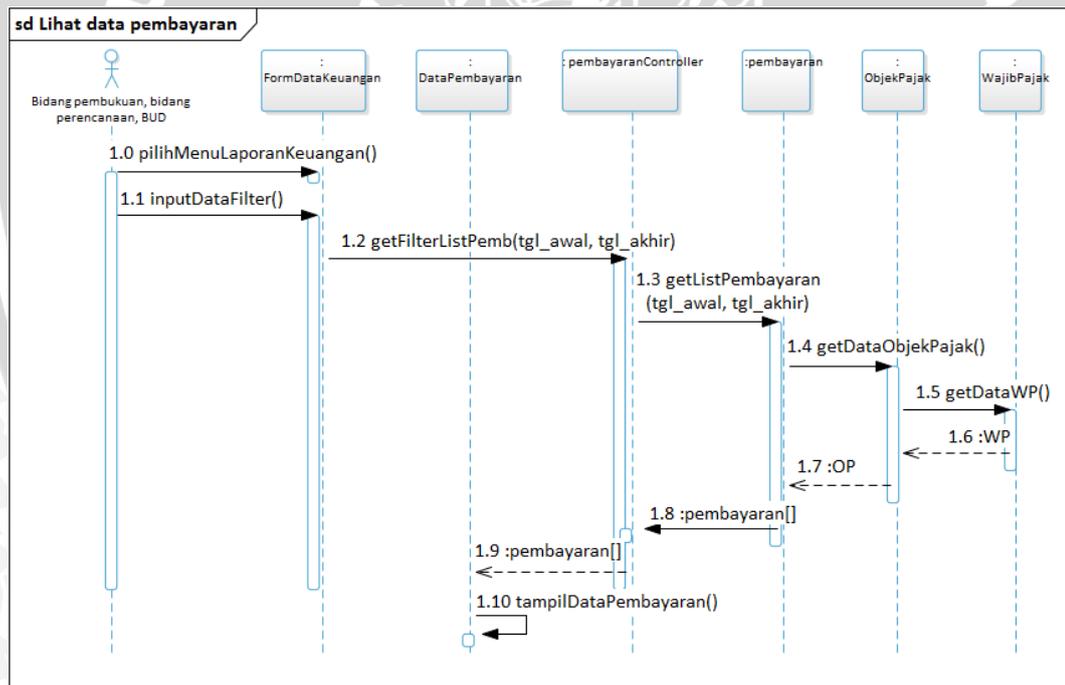
**Gambar 5.6 Sequence Diagram Validasi Laporan Pembayaran**

Interaksi pada proses Validasi Laporan Pembayaran digambarkan seperti pada Gambar 5.6. Aktor yang terlibat adalah bidang pembukuan dan BUD. Interaksi dimulai saat aktor melakukan aksi memilih menu laporan pembayaran, yang kemudian fungsi `getListLaporanPemb()` pada `:pembayaranController` dipanggil dan diteruskan memanggil fungsi `getListLaporanPemb()` pada `:ObjekPajak` dan fungsi `getLaporanPemb()`. Kemudian kembalian data pembayaran akan dikirimkan untuk menampilkan daftar laporan pembayaran.

Aktor kemudian melakukan aksi memilih detail laporan dari salah satu pengguna dan memanggil fungsi *getDetailLaporan(id\_laporan)* dengan parameter *id\_laporan* lalu dilanjutkan memanggil fungsi *getDetailLaporan(id\_laporan)* pada *:pembayaran*. Setelah nilai kembalian dikirim dan ditampilkan pada sistem, aktor memilih aksi untuk memvalidasi laporan dan menjalankan fungsi *validasiLaporan(id\_laporan)* dan status validasi akan diubah.

**E. Sequence Diagram – Lihat data pembayaran pajak**

Pada Gambar 5.7 menggambarkan interaksi pada proses Lihat Data Pembayaran Pajak yang dilakukan oleh aktor bidang perencanaan, bidang pembukuan, dan BUD. Interaksi yang digambarkan telah disesuaikan dengan spesifikasi *use case* dan diagram aktivitas Lihat data pembayaran. Interaksi diawali saat aktor memilih menu laporan keuangan dan mengisikan data filter berupa tanggal awal dan akhir. Kemudian fungsi *getFilterListPemb(tgl\_awal, tgl\_akhir)* pada *:pembayaranController* dipanggil dan dilanjutkan memanggil fungsi *getListPembayaran()* dan *getDataPembayaran* pada *:ObjekPajak* dan *:WajibPajak*. Kemudian kembalian berupa data pembayaran.



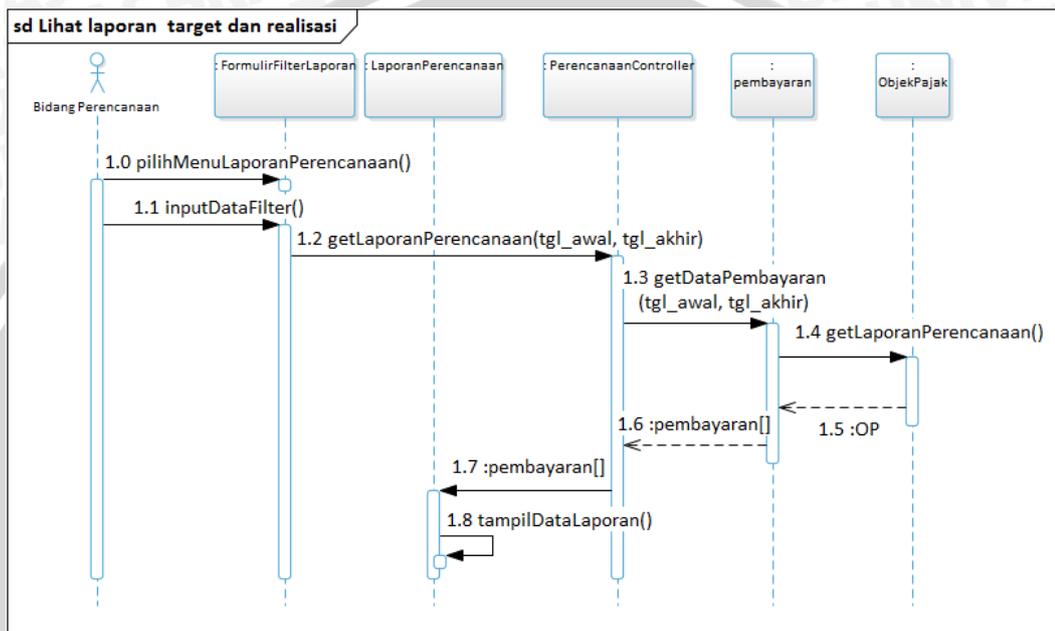
**Gambar 5.7 Sequence Diagram Lihat Data Pembayaran Pajak**

**F. Sequence Diagram – Lihat Laporan Target dan Realisasi Pajak**

Pada Gambar 5.8 Gambar 5.7 menggambarkan interaksi pada proses Lihat Laporan Target dan Realisasi Pajak yang dilakukan oleh aktor bidang perencanaan. Interaksi yang digambarkan telah disesuaikan dengan spesifikasi *use case* dan



diagram aktivitas Lihat Laporan Target dan Realisasi Pajak. Interaksi diawali saat aktor memilih menu laporan perencanaan dan mengisi data filter berupa tanggal awal, tanggal akhir, dan tampilan data. Kemudian fungsi *getLaporanPerencanaan(tgl\_awal, tgl\_akhir)* pada *:perencanaanController* dipanggil dan dilanjutkan memanggil fungsi *getLaporanPerencanaan()* dan *getDataPembayaran* pada *:ObjekPajak* dan *:WajibPajak*. Kemudian kembalian berupa data laporan perencanaan.

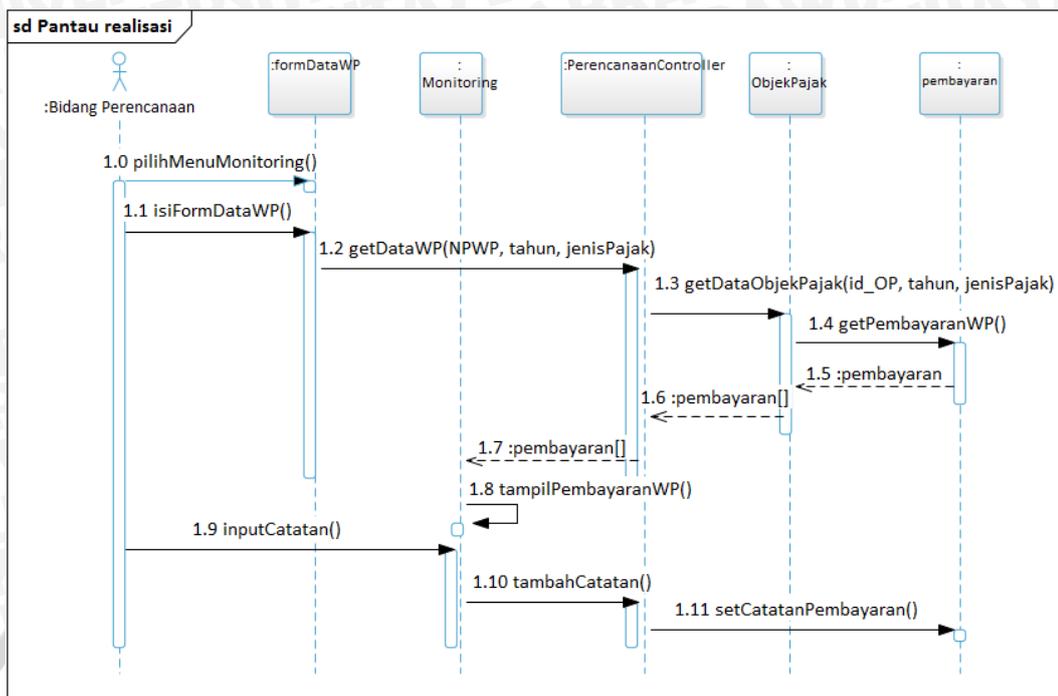


Gambar 5.8 Sequence Diagram Lihat Laporan Target dan Realisasi Pajak

### G. Sequence Diagram – Pantau Pembayaran Pajak

Interaksi pada proses Pantau Pembayaran Pajak dilakukan oleh aktor bidang perencanaan digambarkan pada Gambar 5.9 dan telah disesuaikan dengan spesifikasi *use case* dan diagram aktivitas Pantau Pembayaran Pajak. Interaksi tersebut diawali saat aktor memilih menu *monitoring* dan mengisi data filter berupa NPWP, tahun, dan jenis pajak. Kemudian fungsi *getDataWP(NPWP, tahun, jenisPajak)* pada *:perencanaanController* dipanggil dan dilanjutkan memanggil fungsi *getDataObjekPajak(id\_OP, tahun, jenisPajak)* dan *getPembayaranWP()* pada *:ObjekPajak* dan *:pembayaran*. Kemudian kembalian berupa data pembayaran. Kemudian aktor melakukan aksi menambahkan catatan pembayaran dan diproses dengan memanggil fungsi *tambahCatatan()* pada *:PerencanaanController* dilanjutkan memanggil fungsi *setCatatanPembayaran()* untuk menyimpan data ke sistem.



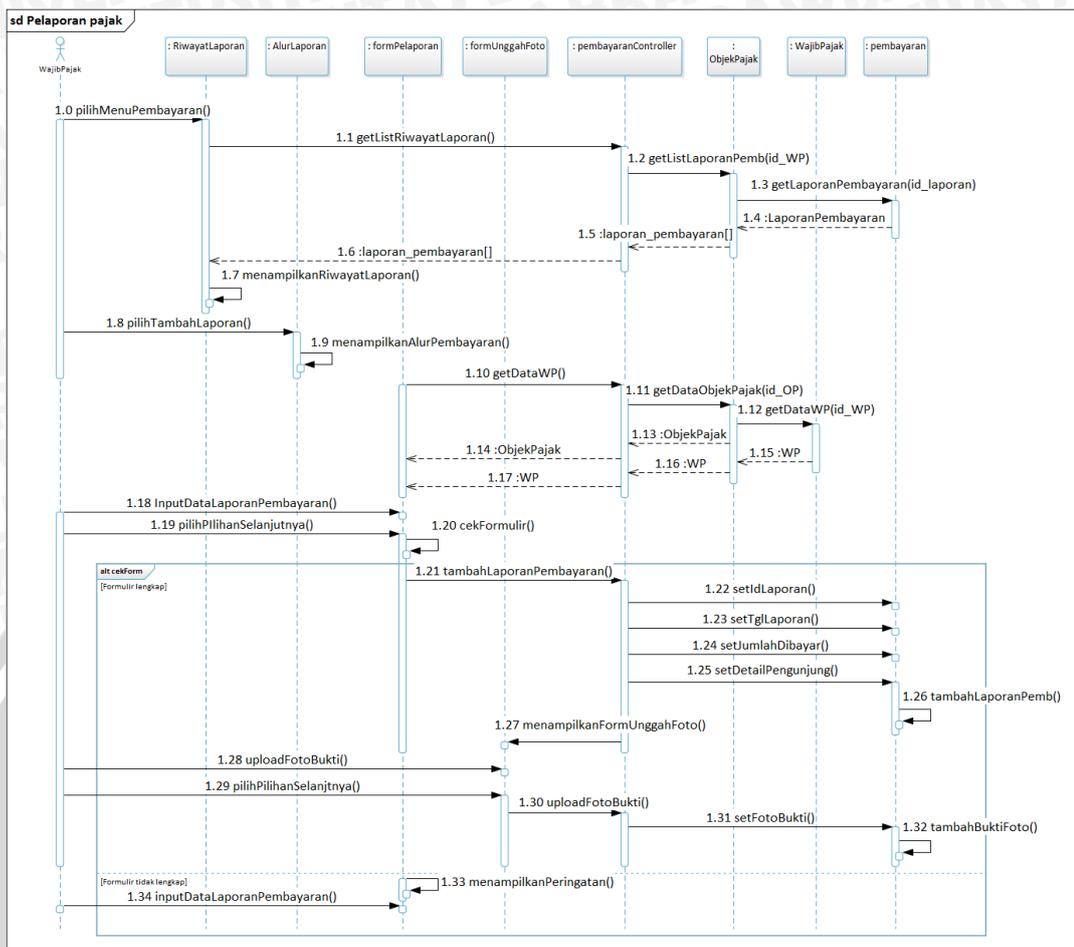


Gambar 5.9 Sequence Diagram Pantau Pembayaran Pajak

#### H. Sequence Diagram – Laporkan Pembayaran Pajak

Interaksi ini menjelaskan proses Laporkan Pembayaran Pajak yang dilakukan oleh aktor wajib pajak. Interaksi tersebut digambarkan pada Gambar 5.9 dan telah disesuaikan dengan spesifikasi *use case* dan diagram aktivitas Laporkan Pembayaran Pajak. Interaksi tersebut diawali saat aktor memilih menu laporan pembayaran dan fungsi *getListRiwayatLaporan(id\_pengguna)* pada *:pembayaranController*. Kemudian data akan diambil dengan memanggil fungsi *getListLaporanPemb(id\_WP)* pada objek *:ObjekPajak* dan diteruskan memanggil fungsi *getLaporanPembayaran(id\_laporan)*. Kembalian data berupa *array* daftar riwayat laporan akan dikirimkan dan ditampilkan di sistem.

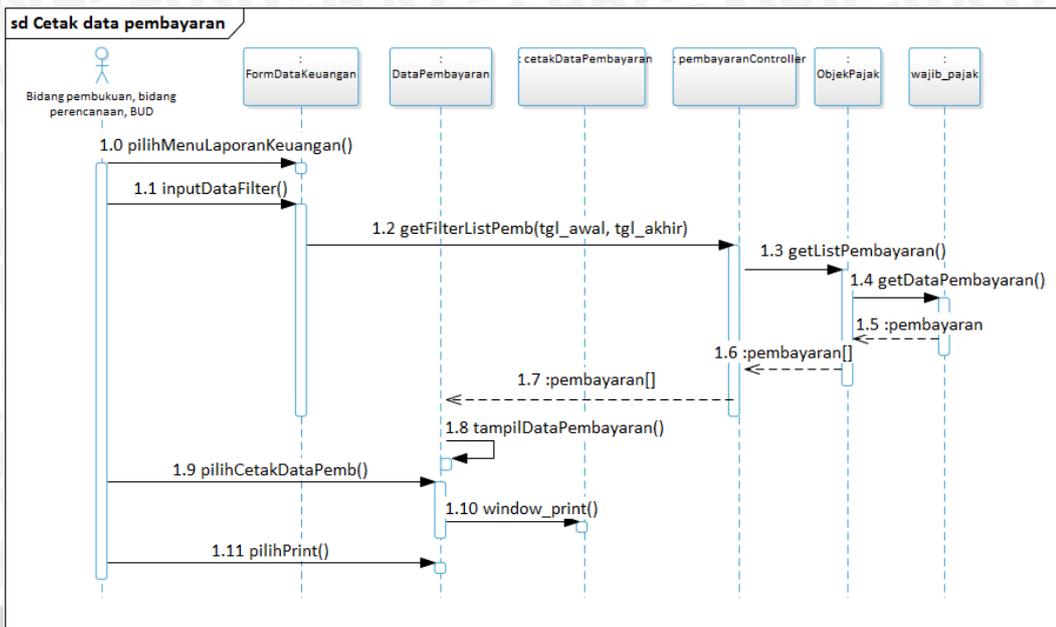
Aktor kemudian memilih pilihan untuk menambahkan laporan dan sistem akan menampilkan alur laporan pembayaran. Aktor kemudian mengisikan data-data untuk melaporkan pembayaran pajak dan juga mengunggah foto bukti pembayaran. Untuk menambah laporan akan memanggil fungsi *tambahLaporanPembayaran()* pada *:pembayaranController* dan menjalankan fungsi-fungsi lain yaitu *setIdLaporan()*, *setTglLaporan()*, *setJumlahLaporan()*, *setDetailPengunjung()*, dan *tambahLaporanPemb()* pada *:pembayaran*. Kemudian untuk mengunggah foto dipanggil fungsi *uploadFotoBukti()* pada objek *:pembayaranController* dan fungsi yang dipanggil pada objek *:pembayaran* untuk memasukkan data ke basis data adalah *setFotoBukti()* dan *tambahBuktiFoto()*.



Gambar 5.10 Sequence Diagram Laporkan Pembayaran Pajak

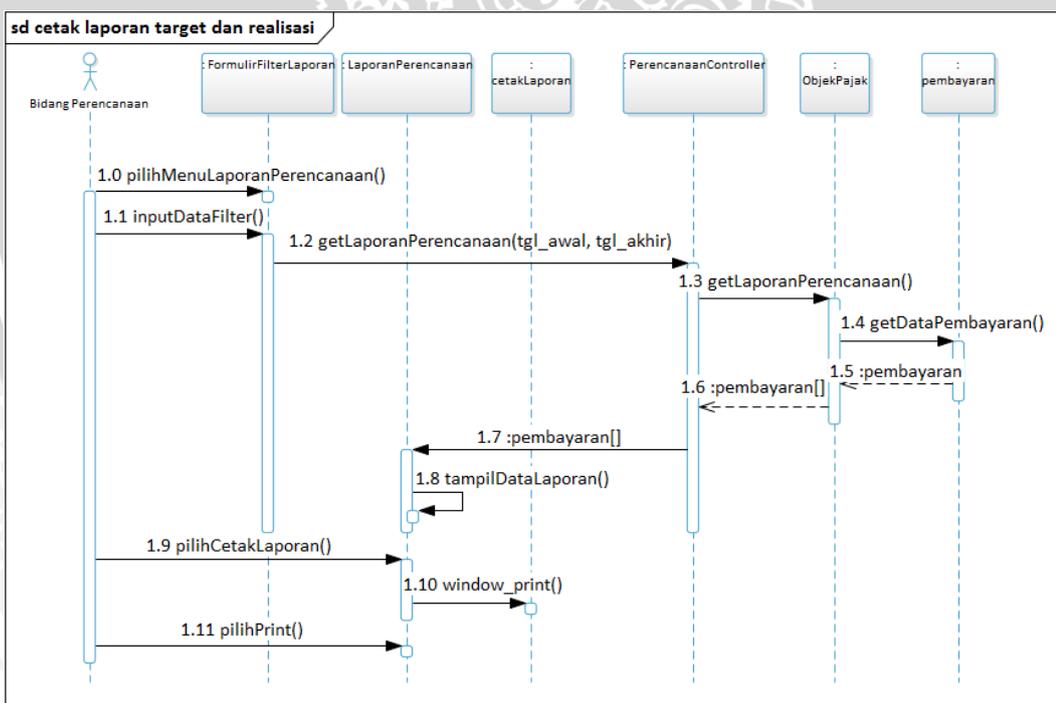
### I. Sequence Diagram – Cetak data pembayaran pajak

Pada Gambar 5.11 menggambarkan interaksi pada proses Cetak Data Pembayaran Pajak yang dilakukan oleh aktor bidang perencanaan, bidang pembukuan, dan BUD. Interaksi yang digambarkan telah disesuaikan dengan spesifikasi *use case* dan diagram aktivitas cetak data pembayaran. Interaksi diawali dengan melakukan proses yang sama pada proses Lihat Data Pembayaran Pajak (Gambar 5.7) dan untuk proses cetak laporan aktor melakukan aksi memilih pilihan cetak dan fungsi *window\_print()* akan berjalan untuk menampilkan jendela pratinjau cetak.



Gambar 5.11 Sequence Diagram Cetak Data Pembayaran Pajak

J. Sequence Diagram – Cetak laporan target dan realisasi pajak



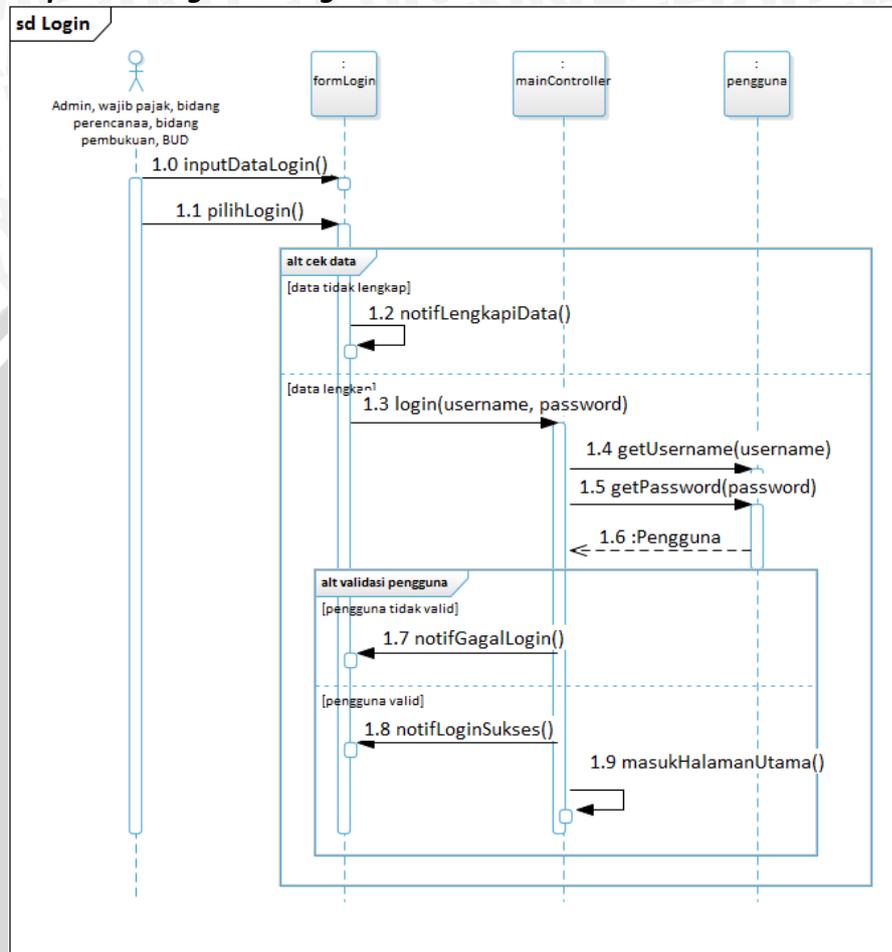
Gambar 5.12 Sequence Diagram Cetak Laporan Target dan Realisasi Pajak

Pada Gambar 5.12 menggambarkan interaksi pada proses Cetak Laporan Target dan Realisasi Pajak yang dilakukan oleh aktor bidang perencanaan. Interaksi yang digambarkan telah disesuaikan dengan spesifikasi use case dan diagram aktivitas cetak laporan target dan realisasi pajak. Interaksi diawali dengan



melakukan proses yang sama pada interaksi Lihat Laporan Target dan Realisasi Pajak (Gambar 5.8) dan untuk proses cetak laporan aktor melakukan aksi memilih pilihan cetak dan fungsi *window\_print()* akan berjalan untuk menampilkan jendela pratinjau cetak.

### K. Sequence Diagram – Login



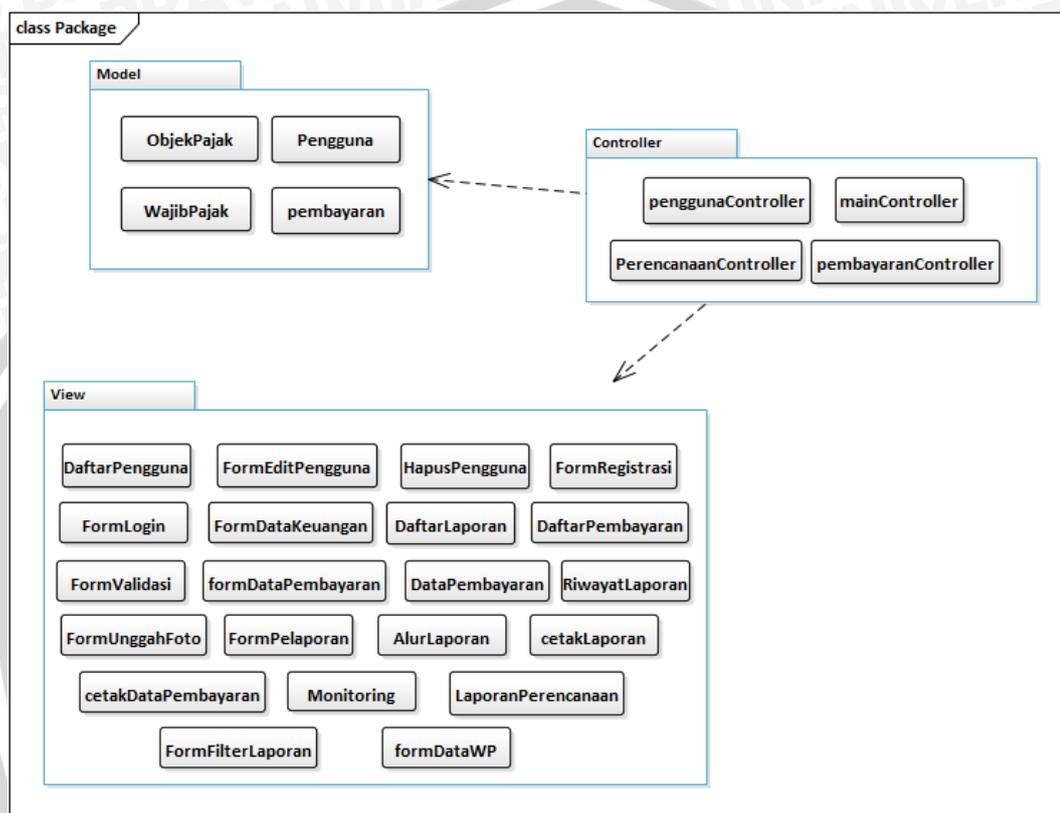
Gambar 5.13 Sequence Diagram Login

Interaksi yang ditunjukkan pada Gambar 5.13 menggambarkan alur dari proses *login*. Aktor yang berperan pada interaksi adalah admin, BUD, bidang perencanaan, bidang pembukuan, dan wajib pajak. Proses diawali saat aktor melakukan aksi mengisi data *login* dan memilih untuk *login*. Sistem melakukan pengecekan apakah data lengkap, jika data lengkap maka fungsi *login(username, password)* akan dipanggil dan dicek untuk mengetahui apakah *username* dan *password* valid. Jika valid maka aktor bisa mengakses sistem dan menjalankan pilihan yang tersedia.

## 5.4 Elemen Desain

### 5.4.1 Package

Bagian ini memodelkan *package* dalam bentuk diagram untuk mengumpulkan elemen-elemen menjadi kelompok-kelompok. Pada Gambar 5.14 menunjukkan pengelompokan kelas-kelas dari kelas ke dalam beberapa *package*, yaitu *package model*, *package controller*, dan *package view*.



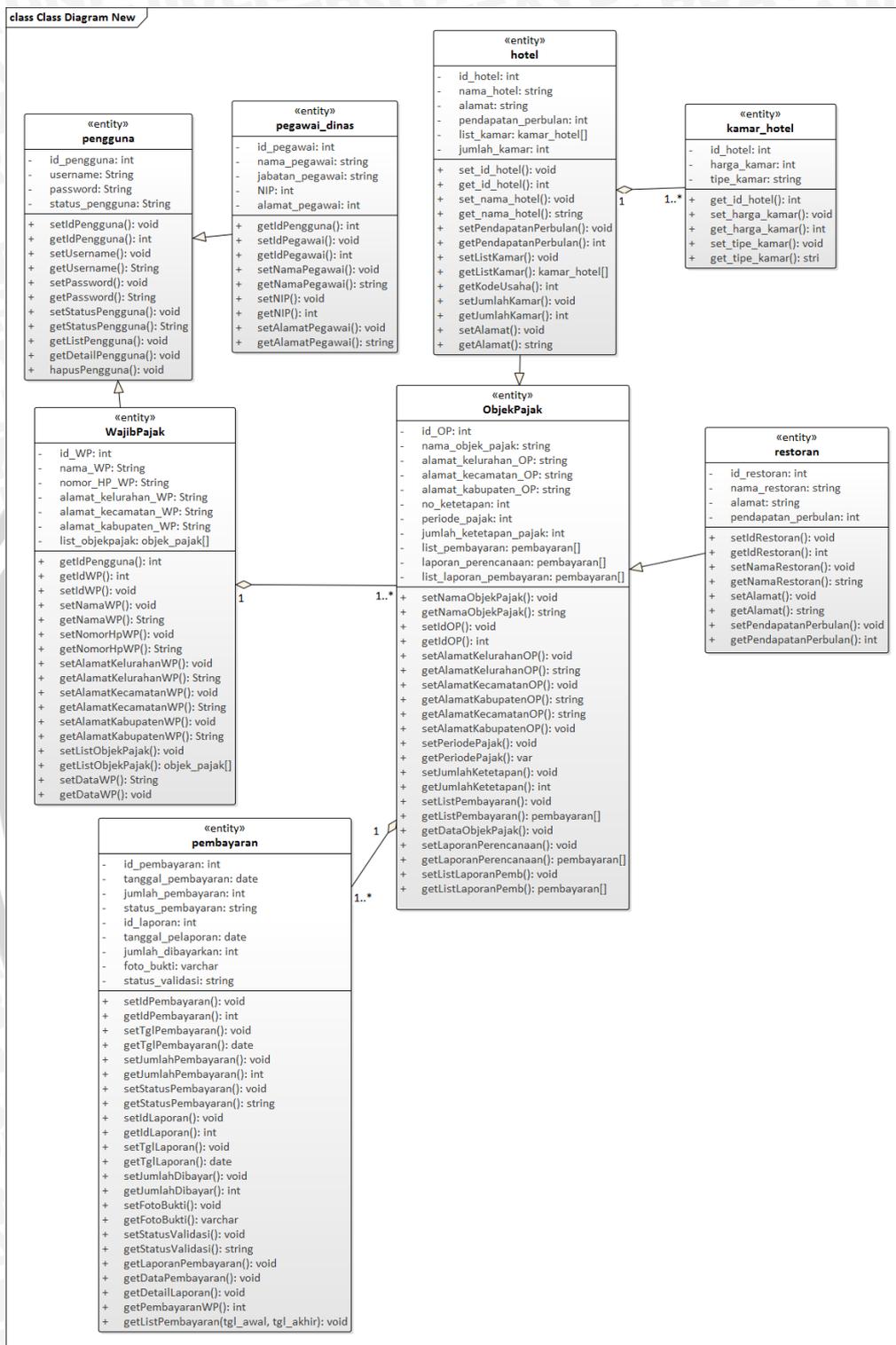
Gambar 5.14 Package

### 5.4.2 Diagram Kelas

Bagian ini menunjukkan perancangan dari diagram kelas yang bertujuan untuk menggambarkan atribut dari suatu sistem dan layanan untuk memanipulasi atribut tersebut dalam bentuk fungsi. Diagram kelas yang dihasilkan, didapatkan dari penyempurnaan kelas analisis yang telah di analisis sebelumnya. Perancangan diagram kelas yang dilakukan menghasilkan diagram kelas *model*, *controller*, dan *boundary* atau *view*.

#### A. Diagram Kelas – Model

Pada bagian ini menunjukkan hasil rancangan dari diagram kelas *model* beserta hubungan atau *relationship* antar tiap *model*. Kelas *model* yang dihasilkan adalah *pengguna*, *pegawai\_dinas*, *WajibPajak*, *ObjekPajak*, *restoran*, *hotel* dan *kamar\_hotel*.



Gambar 5.15 Diagram Kelas – Model

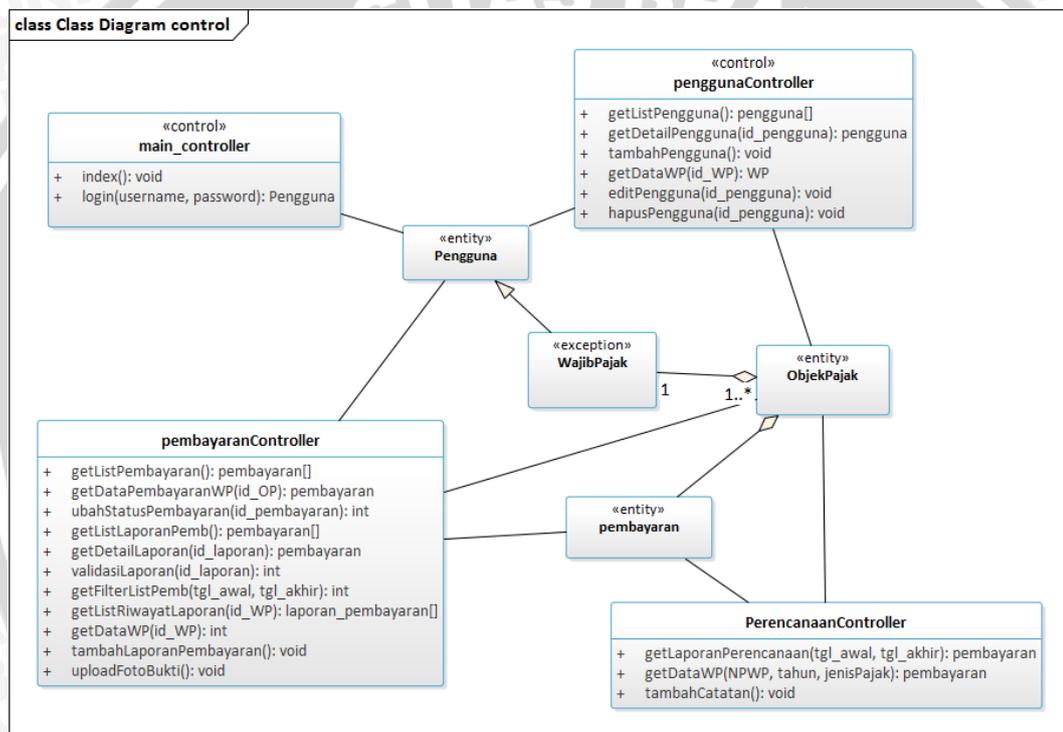
Pada Gambar 5.15 menunjukkan apa saja atribut beserta fungsi yang ada pada masing-masing kelas. Atribut yang ada mengacu dari *glossary* spesifikasi *use case*



yang telah dibuat, sedangkan fungsi yang ada didapatkan dari fungsi-fungsi yang ada pada interaksi antar kelas yang telah dianalisis pada pemodelan interaksi (diagram *sequence*). Kemudian, hubungan antar kelas yang digambarkan adalah :

- *WajibPajak* dan *pegawai\_dinas* memiliki hubungan generalisasi dengan *pengguna*.
- *WajibPajak* dan *ObjekPajak* memiliki hubungan agregasi, karena *ObjekPajak* keberadaannya melekat terhadap *WajibPajak*. Begitu juga dengan kelas *pembayaran* dengan kelas *ObjekPajak*.
- Kelas *restoran* dan *hotel* memiliki hubungan generalisasi dengan kelas *ObjekPajak*.

## B. Diagram Kelas – Controller

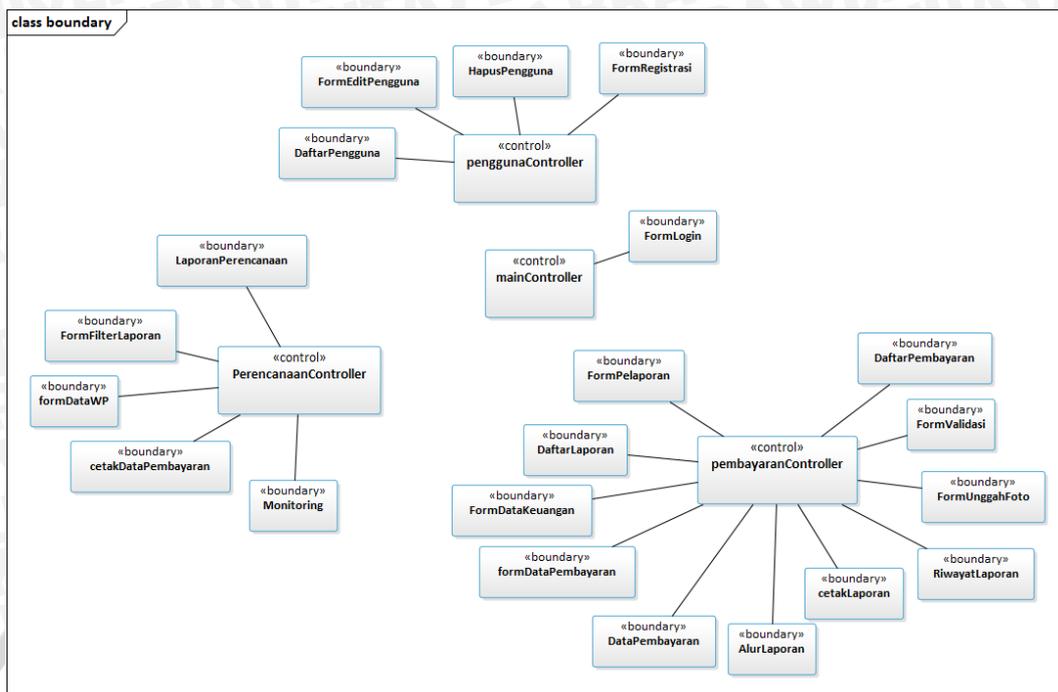


**Gambar 5.16 Diagram Kelas- Controller**

Pada Gambar 5.16 menggambarkan pemodelan diagram kelas dari *controller* beserta hubungannya dengan kelas-kelas *model*. Terdapat empat kelas *controller* yang dimodelkan yaitu *main\_controller*, *penggunaController*, *pembayaranController*, dan *PerencanaanController*.

## C. Diagram Kelas – View

Pada Gambar 5.17 menggambarkan *boundary* atau *view* apa saja yang digunakan dalam rancangan sistem yang telah disesuaikan dengan kelas analisis yang telah didefinisikan sebelumnya. Pada gambar tersebut juga menunjukkan hubungan antara *view* dengan *controller* yang ada.

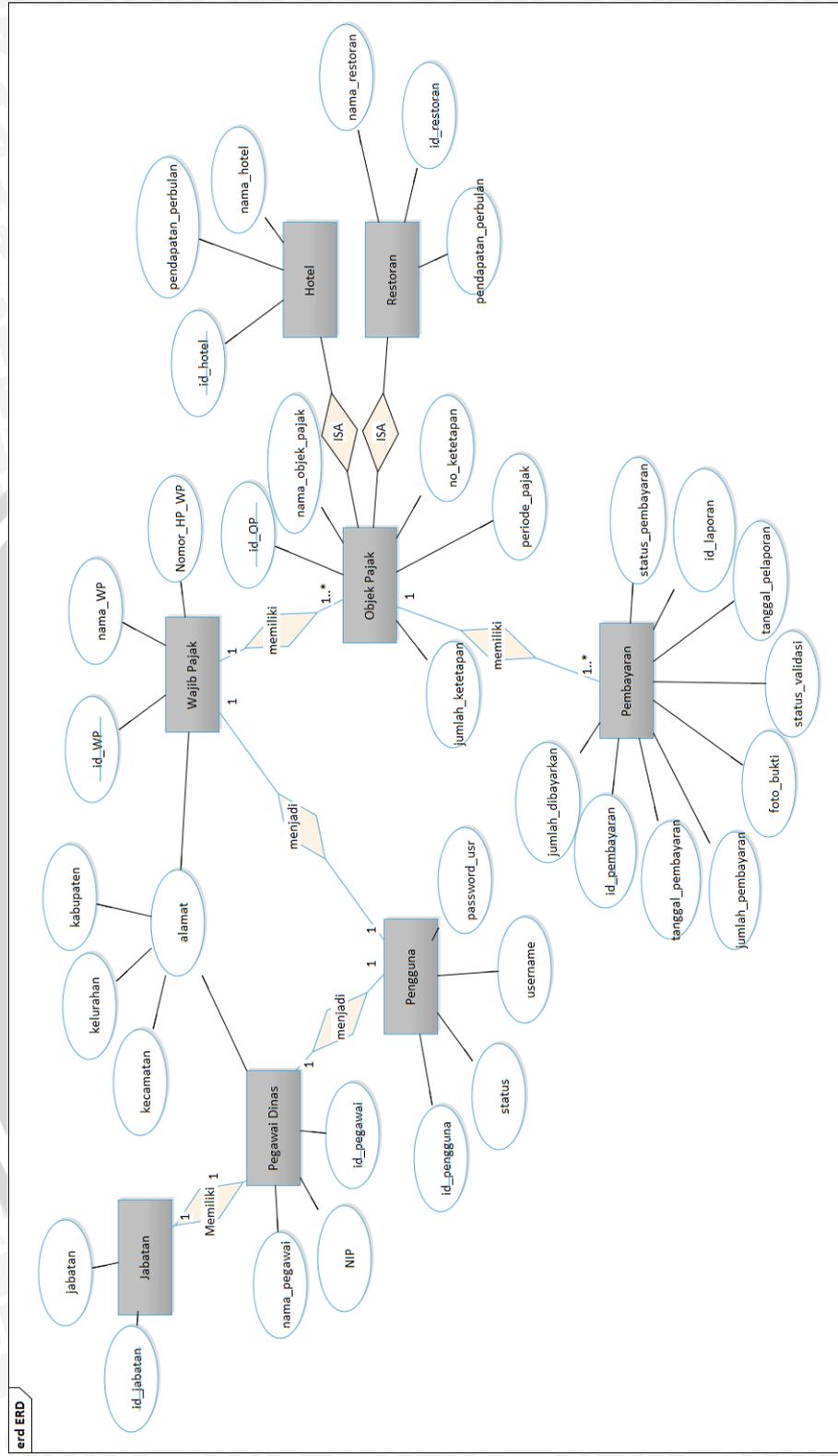


Gambar 5.17 Diagram Kelas - View

## 5.5 Pemodelan Data

### 5.5.1 Entity Relationship Diagram

Bagian ini melakukan analisis terhadap apa saja entitas yang dibutuhkan oleh sistem berdasarkan analisis kelas *model*, atribut apa saja yang dimiliki oleh entitas tersebut, dan bagaimana entitas tersebut saling berhubungan. Pada Gambar 5.18 menunjukkan hasil dari analisis ERD yang menunjukkan terdapat delapan entitas yang didefinisikan dan masing-masing entitas sudah terdapat penjelasan hubungan dengan entitas lain yang berhubungan. ERD ini nantinya bisa digunakan untuk acuan perancangan tabel yang akan dibuat.



Gambar 5.18 Entity Relationship Diagram

### 5.5.2 Perancangan Tabel

#### A. Tabel Jabatan

Nama tabel : jabatan

Jumlah kolom : 2

Fungsi : menyimpan tipe jabatan yang ada.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_jabatan	Integer	-	Id dari tipe jabatan.
2.	jabatan	Varchar	30	Nama dari tipe jabatan atau jenis jabatan.

#### B. Tabel Pegawai Dinas

Nama tabel : pegawai\_dinas

Jumlah kolom : 3

Fungsi : menyimpan data pegawai dinas.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_pegawai	Integer	-	Id dari pegawai dinas.
2.	nama_pegawai	Varchar	25	Nama dari pegawai dinas.
3.	NIP	Integer	-	Nomor induk pegawai

#### C. Tabel Status Pengguna

Nama tabel : status\_pengguna

Jumlah kolom : 2

Fungsi : menyimpan tipe status pengguna.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_status_pengguna	Integer	-	Id status pengguna
2.	status_pengguna	Varchar	25	Status pengguna

#### D. Tabel pengguna

Nama tabel : pengguna

Jumlah kolom : 3

Fungsi : menyimpan data pengguna.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_pengguna	Integer	-	Id dari pengguna.
2.	username	Varchar	60	Nama pengguna yang digunakan untuk <i>login</i> .
3.	password_usr	Varchar	25	Kata sandi yang digunakan untuk <i>login</i> .

#### E. Tabel Wajib Pajak

Nama tabel : wajib\_pajak

Jumlah kolom : 3

Fungsi : menyimpan data wajib pajak.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_WP	Integer	-	Id dari wajib pajak yang berupa NPWP.
2.	nama_WP	Varchar	25	Nama dari wajib pajak.
3.	Nomor_HP_WP	Integer	-	Nomor HP dari wajib pajak.

#### F. Tabel Kabupaten

Nama tabel : Kabupaten

Jumlah kolom : 2

Fungsi : menyimpan data kabupaten.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_kabupaten	Integer	-	Id dari kabupaten.
2.	nama_kabupaten	Varchar	60	Nama kabupaten.

#### G. Tabel Kecamatan

Nama tabel : Kecamatan

Jumlah kolom : 2

Fungsi : menyimpan data kecamatan.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_kecamatan	Integer	-	Id dari kecamatan.
2.	Nama_kecamatan	Varchar	60	Nama kecamatan.

H. Tabel Kelurahan

Nama tabel : Kelurahan

Jumlah kolom : 2

Fungsi : menyimpan data kelurahan.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_kelurahan	Integer	-	Id dari kelurahan.
2.	Nama_kelurahan	Varchar	60	Nama kelurahan.

I. Tabel Objek Pajak

Nama tabel : ObjekPajak

Jumlah kolom : 5

Fungsi : menyimpan data objek pajak.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_OP	Integer	-	Id dari objek pajak.
2.	nama_objek_pajak	Varchar	60	Nama dari objek pajak.
3.	no_ketetapan	Integer	-	Nomor ketetapan pajak.
4.	periode_pajak	Integer	-	Tahun periode pajak.
5.	jumlah_ketetapan	Integer	-	Jumlah ketetapan pajak

J. Tabel Hotel

Nama tabel : hotel

Jumlah kolom : 3

Fungsi : menyimpan data hotel yang merupakan salah satu jenis objek pajak.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_hotel	Integer	-	Id dari hotel.
2.	nama_hotel	Varchar	60	Nama dari hotel.
3.	Pendapatan_perbulan	Integer	-	Pendapatan dari hotel tiap bulan.

## K. Tabel Kamar Hotel

Nama tabel : kamar\_hotel

Jumlah kolom : 3

Fungsi : menyimpan data jenis kamar hotel.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_kamar_hotel	Integer	-	Id dari jenis kamar hotel.
2.	harga_kamar	Integer	-	Harga kamar sesuai tipe kamar.
3.	tipe_kamar	Varchar	5	Tipe kamar hotel.

## L. Tabel Restoran

Nama tabel : restoran

Jumlah kolom : 3

Fungsi : menyimpan data restoran yang merupakan jenis objek pajak.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_restoran	Integer	-	Id dari restoran.
2.	nama_restoran	Varchar	60	Nama dari restoran.
3.	pendapatan_perbulan	Integer	-	Pendapatan dari restoran per bulan.

## M. Tabel Pembayaran

Nama tabel : pembayaran

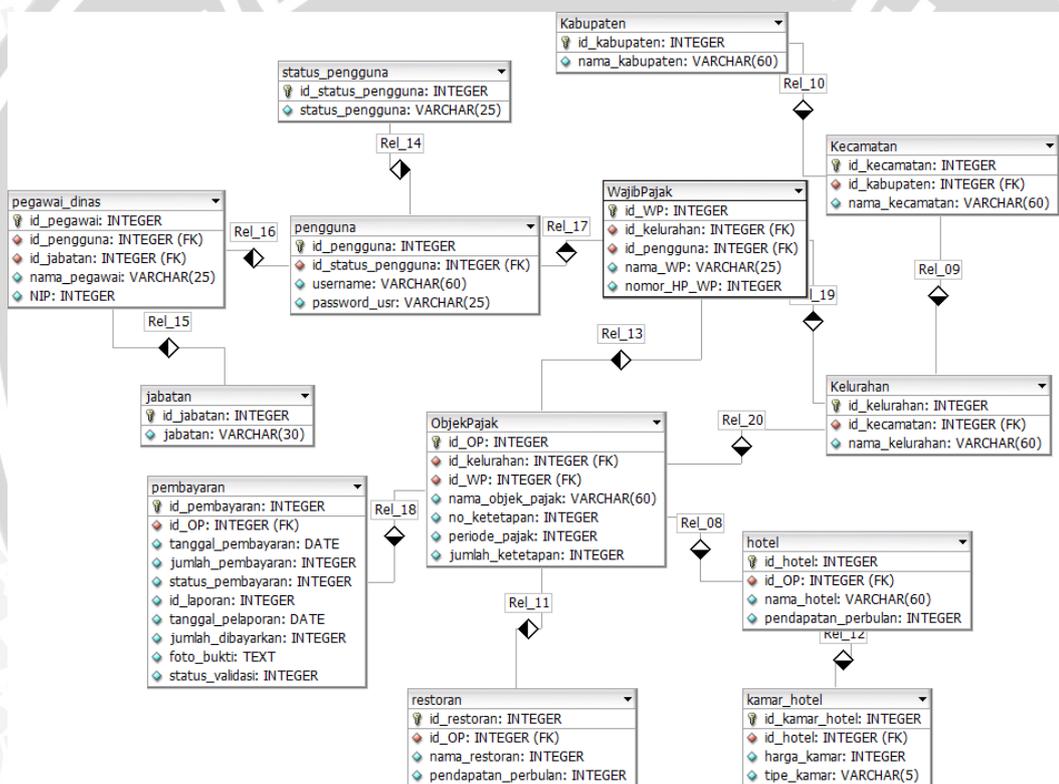
Jumlah kolom : 9

Fungsi : menyimpan data pembayaran.

No.	Nama Kolom	Tipe Data	Panjang Data	Keterangan
1.	id_pembayaran	Integer	-	Id dari pembayaran.
2.	Tanggal_pembayaran	Date	-	Tanggal pembayaran pajak.
3.	jumlah_pembayaran	Integer	-	Nominal uang pembayaran pajak.
4.	status_pembayaran	Integer	-	Status pembayaran pajak.

5.	id_laporan	Integer	-	Id laporan pembayaran pajak
6.	tanggal_pelaporan	Date	-	Tanggal pelaporan pembayaran pajak.
7.	jumlah_dibayarkan	Integer	-	Jumlah pajak yang dibayarkan.
8.	foto_bukti	Text	-	Foto bukti pembayaran pajak.
9.	status_validasi	Integer	-	Status validasi pelaporan pembayaran pajak.

### 5.5.3 Physical Data Model

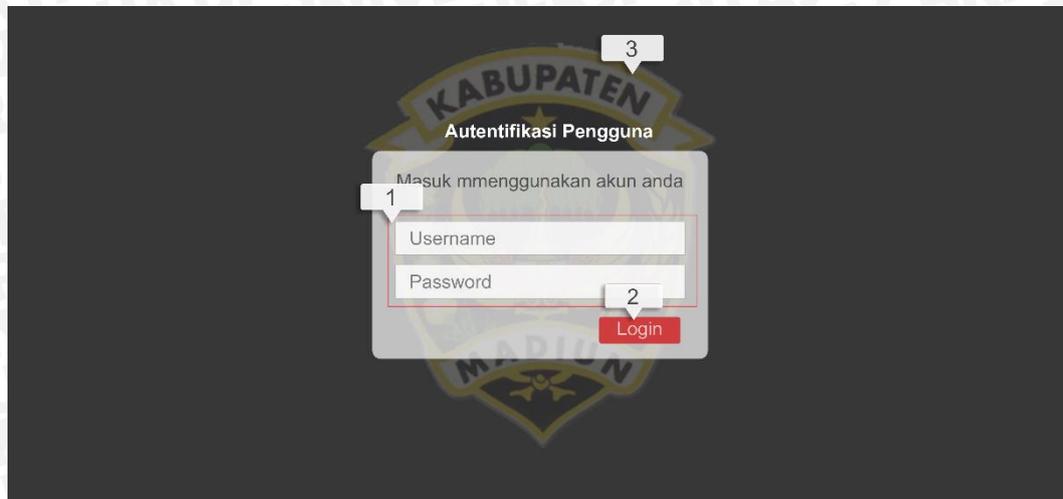


Gambar 5.19 Physical Data Model

Pada Gambar 5.19 menunjukkan model data fisik yang dibutuhkan sistem dan nantinya akan diimplementasikan ke dalam basis data sebagai tabel-tabel basis data sistem. PDM tersebut telah disesuaikan dengan perancangan tabel yang dibuat sebelumnya.

## 5.6 Perancangan Antarmuka Pengguna

### 5.6.1 Halaman *Login*



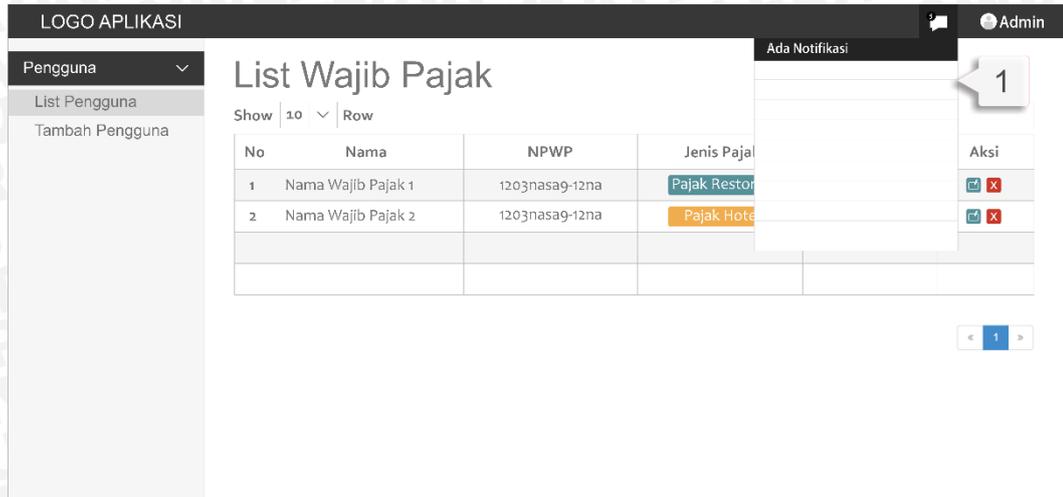
**Gambar 5.20 Rancangan Antarmuka Halaman *Login***

Pada Gambar 5.20 menunjukkan rancangan dari antarmuka sistem yang dirancang dengan keterangan isi pada rancangan dijelaskan pada Tabel 5.3.

**Tabel 5.3 Keterangan Rancangan Antarmuka Halaman *Login***

No.	Nama objek	Tipe	Keterangan
1.	Formulir <i>Login</i>	Formulir	Formulir yang digunakan untuk memasukkan data <i>username</i> dan <i>password</i> untuk melakukan <i>login</i> .
2.	<i>Login</i>	Tombol	Tombol untuk melakukan <i>submit</i> data <i>login</i> .
3.	Logo DISPENDA	Gambar	Gambar Logo Dinas Kabupaten Madiun yang digunakan menjadi latar belakang halaman.

## 5.6.2 Notifikasi



**Gambar 5.21 Rancangan Antarmuka Tampilan Notifikasi**

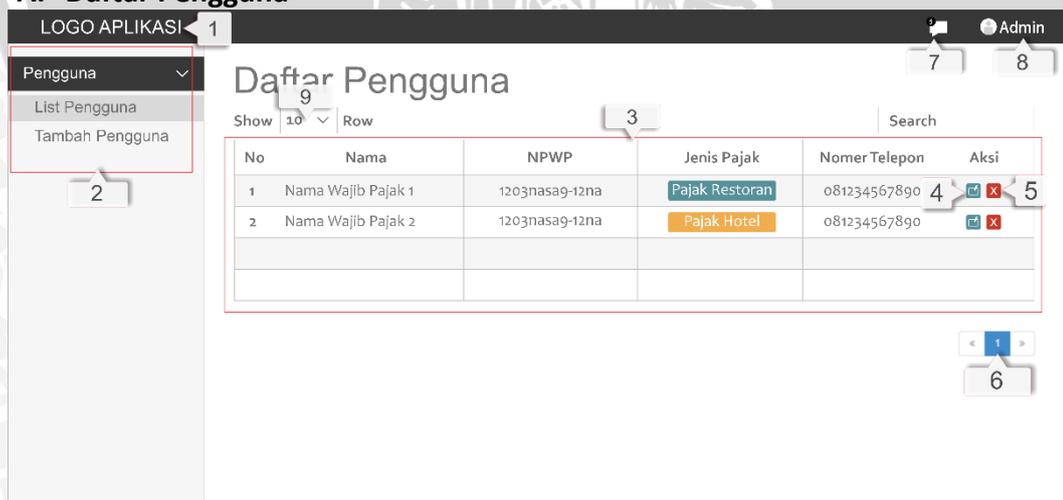
Pada Tabel 5.4 menjelaskan mengenai isi dari rancangan antarmuka tampilan notifikasi seperti yang digambarkan pada Gambar 5.21.

**Tabel 5.4 Keterangan Rancangan Antarmuka Tampilan Notifikasi**

No.	Nama objek	Tipe	Keterangan
1.	Notifikasi	Panel	Panel yang menampilkan pemberitahuan yang masuk.

## 5.6.3 Halaman Admin

### A. Daftar Pengguna



**Gambar 5.22 Rancangan Antarmuka Halaman Daftar Pengguna**

Pada Gambar 5.22 merupakan rancangan antarmuka pada halaman daftar pengguna dan pada Tabel 5.5 merupakan penjelasan dari gambar tersebut.

Tabel 5.5 Keterangan Rancangan Antarmuka Halaman Daftar Pengguna

No.	Nama objek	Tipe	Keterangan
1.	Logo Aplikasi	Gambar	Gambar yang merupakan logo dari sistem yang dibuat. Logo ini akan selalu ada pada setiap halaman dan terletak pada menu bar atas.
2.	Menu Bar	Tombol	Merupakan menu dari sistem yang dapat diakses oleh akun admin, terdiri dari : <ul style="list-style-type: none"> <li>- List Pengguna: menampilkan halaman daftar pengguna.</li> <li>- Tambah Pengguna : menampilkan halaman registrasi pengguna.</li> </ul>
3.	Tabel Daftar Pengguna	Tabel	Merupakan tabel yang menampilkan daftar pengguna dari sistem. Kolom terdiri dari : <ul style="list-style-type: none"> <li>- Nomor</li> <li>- Nama</li> <li>- NPWP</li> <li>- Jenis Pajak</li> <li>- Nomor Telepon</li> <li>- Aksi</li> </ul>
4.	Detail	Tombol	Tombol yang digunakan untuk menampilkan detail pengguna.
5.	Hapus	Tombol	Tombol yang digunakan untuk menghapus pengguna.
6.	Navigasi Halaman	Tombol	Merupakan navigasi yang menunjukkan posisi halaman dan digunakan untuk berpindah ke halaman tabel lain.
7.	Icon Notifikasi	Tombol	Icon dari fitur notifikasi yang menampilkan jumlah pemberitahuan dan dapat menampilkan panel notifikasi.
8.	Profil	Tombol	Merupakan tombol yang menunjukkan status pengguna (nama pengguna) dan dapat menampilkan pilihan <i>logout</i> .
9.	Show Row	Tombol	Digunakan untuk memberikan pilihan jumlah baris dalam tabel yang ingin ditampilkan.

## B. Tambah Pengguna

LOGO APLIKASI Admin

Pengguna

List Pengguna

Tambah Pengguna

### Registrasi Pengguna

Hotel Restoran 1 2

Nama Nama

NPWP NPWP

Jenis Pajak Jenis Pajak

Lokasi Lokasi

Jenis Kamar dan Harga

Tipe A Jumlah Tipe A Harga Tipe A

Tipe B Jumlah Tipe B Harga Tipe B

Tipe C Jumlah Tipe C Harga Tipe C

Biaya Penetapan Pajak Biaya

Username

Password

BATAL REGISTER

3 4

**Gambar 5.23 Rancangan Antarmuka Halaman Tambah Pengguna - Hotel**

LOGO APLIKASI Admin

Pengguna

List Pengguna

Tambah Pengguna

### Registrasi Pengguna

Hotel Restoran 5

Nama Nama

NPWP NPWP

Jenis Pajak Jenis Pajak

Lokasi Lokasi

Biaya Penetapan Pajak Biaya

Username

Password

BATAL REGISTER

**Gambar 5.24 Rancangan Antarmuka Halaman Tambah Pengguna - Restoran**

Pada Gambar 5.23 dan Gambar 5.24 menunjukkan rancangan antarmuka dari halaman tambah pengguna dan konten yang terdapat pada halaman tersebut dijelaskan pada Tabel 5.6.

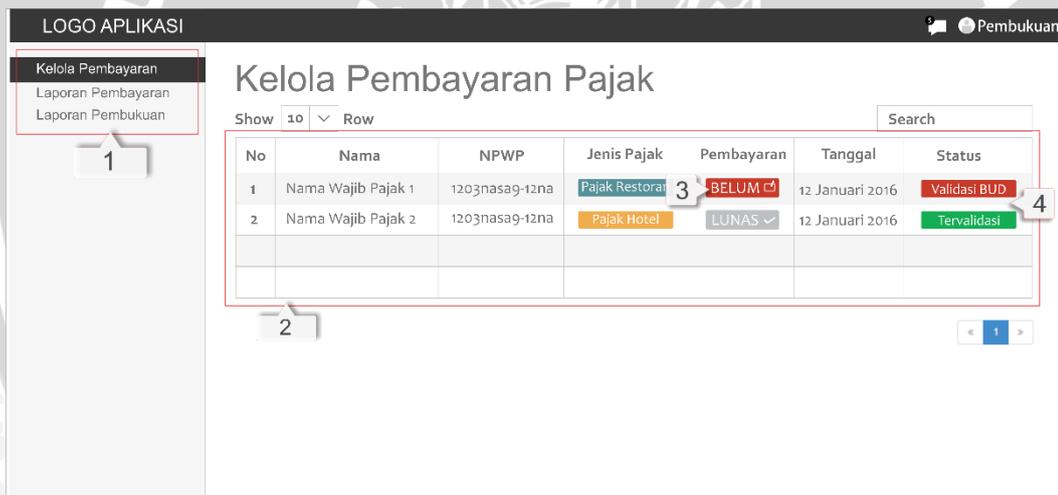
**Tabel 5.6 Keterangan Rancangan Antarmuka Halaman Tambah Pengguna**

No.	Nama objek	Tipe	Keterangan
1.	Tab Menu	Tombol / Tab	Merupakan tabulasi untuk dapat berpindah ke menu lain dalam satu halaman.
2.	Formulir Tambah Pengguna Hotel	Formulir	Merupakan formulir yang digunakan untuk menambahkan Wajib Pajak Hotel yang berisi

			kolom : Nama, NPWP, Jenis Pajak, Lokasi, Jenis Kamar dan Harga, Biaya Penetapan Pajak, <i>username</i> , dan <i>password</i> .
3.	Batal	Tombol	Merupakan tombol yang digunakan untuk membatalkan aksi pendaftaran pengguna.
4.	Register	Tombol	Tombol yang digunakan untuk menambahkan atau mendaftarkan pengguna.
5.	Formulir Tambah Pengguna Restoran	Formulir	Merupakan formulir yang digunakan untuk menambahkan Wajib Pajak Hotel yang berisi kolom : Nama, NPWP, Jenis Pajak, Lokasi, Biaya Penetapan Pajak, <i>username</i> , dan <i>password</i> .

### 5.6.4 Halaman Pembukuan dan Bendahara Umum Daerah

#### A. Kelola Pembayaran



Gambar 5.25 Rancangan Antarmuka Halaman Kelola Pembayaran

Pada Tabel 5.7 menjelaskan mengenai isi dari rancangan antarmuka halaman kelola pembayaran seperti yang digambarkan pada Gambar 5.25.

Tabel 5.7 Keterangan Rancangan Antarmuka Halaman Kelola Pembayaran

No.	Nama objek	Tipe	Keterangan
1.	Menu Bar	Tombol	Merupakan menu dari sistem yang dapat diakses oleh akun pembukuan dan BUD, terdiri dari : - Kelola Pembayaran : menampilkan halaman kelola pembayaran.

			<ul style="list-style-type: none"> <li>- Laporan Pembayaran : menampilkan halaman laporan pembayaran.</li> <li>- Laporan pembukuan : menampilkan halaman laporan pembukuan.</li> </ul>
2.	Tabel Data Pembayaran	Tabel	<p>Merupakan tabel yang menampilkan daftar pembayaran dari wajib pajak, terdiri dari kolom :</p> <ul style="list-style-type: none"> <li>- Nomor</li> <li>- Nama</li> <li>- NPWP</li> <li>- Jenis Pajak</li> <li>- Pembayaran</li> <li>- Tanggal</li> <li>- Status</li> </ul>
3.	Keterangan Pembayaran	Tombol	<p>Merupakan keterangan pembayaran dari wajib pajak. Jika keterangan berupa kata 'Belum' maka tombol akan aktif dapat dilakukan perintah pengubahan status pembayaran. Namun jika statusnya sudah 'Lunas' maka tombol akan <i>disable</i>.</p>
4.	Status Laporan Pembayaran	Teks	<p>Merupakan keterangan bagaimanakah status dari laporan pembayaran sistem.</p>

### B. Ubah Status Pembayaran



Gambar 5.26 Rancangan Antarmuka Halaman Ubah Status Pembayaran

Pada Gambar 5.26 merupakan rancangan antarmuka pada halaman ubah status pembayaran dan pada Tabel 5.8 merupakan penjelasan dari gambar tersebut.

**Tabel 5.8 Keterangan Rancangan Antarmuka Halaman Ubah Status Pembayaran**

No.	Nama objek	Tipe	Keterangan
1.	Formulir Keterangan Pembayaran	Formulir	Merupakan formulir yang berisi Nama, NPWP, Jenis Pajak, dan Biaya Penetapan yang diambil dari basis data sistem.
2.	Tombol Lunas	Tombol	Merupakan tombol yang digunakan untuk mengubah status pembayaran menjadi Lunas.
3.	Tombol Batal	Tombol	Merupakan tombol yang digunakan untuk membatalkan aksi.

### C. Laporan Pembayaran



**Gambar 5.27 Rancangan Antarmuka Halaman Laporan Pembayaran**

Pada Gambar 5.27 menunjukkan rancangan dari halaman laporan pembayaran dengan keterangan konten pada rancangan dijelaskan pada Tabel 5.9.

**Tabel 5.9 Keterangan Rancangan Antarmuka Halaman Laporan Pembayaran**

No.	Nama objek	Tipe	Keterangan
1.	Tabel Laporan Pembayaran Pajak	Tabel	Merupakan tabel yang menampilkan daftar pembayaran dari wajib pajak, terdiri dari kolom : <ul style="list-style-type: none"> <li>- Nomor</li> <li>- Nama</li> <li>- NPWP</li> </ul>

			<ul style="list-style-type: none"> <li>- Tanggal</li> <li>- Jenis Pajak</li> <li>- Status</li> <li>- Detail</li> </ul>
2.	Tombol Detail	Tombol	Merupakan tombol yang digunakan untuk masuk ke halaman detail laporan pembayaran pajak.

#### D. Laporan Pembukuan/Kuangan



**Gambar 5.28 Rancangan Antarmuka Halaman Laporan Pembukuan/Kuangan**

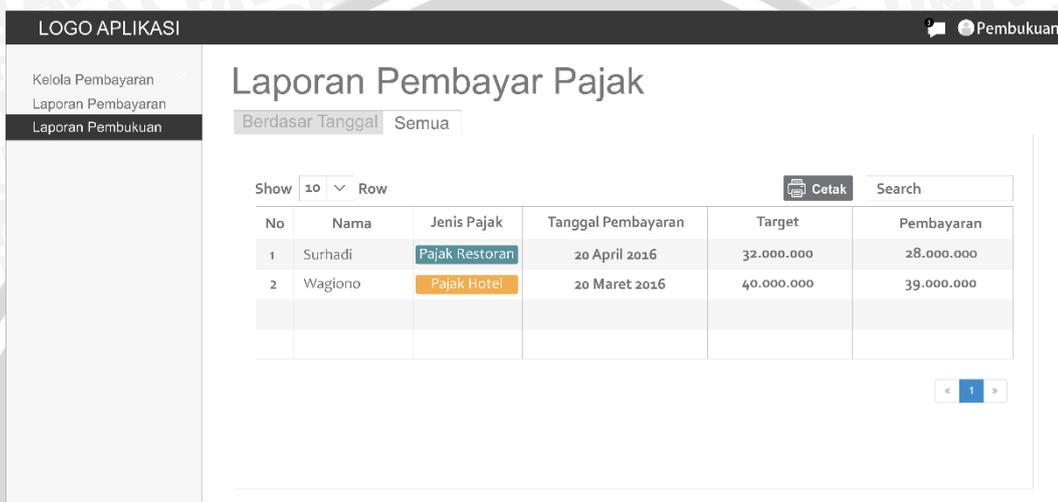
Pada Gambar 5.28 dan Gambar 5.29 menunjukkan rancangan antarmuka dari halaman laporan pembukuan atau keuangan dan konten yang terdapat pada halaman tersebut dijelaskan pada Tabel 5.10.

**Tabel 5.10 Keterangan Rancangan Antarmuka Halaman Laporan Pembukuan/Kuangan**

No.	Nama objek	Tippe	Keterangan
1.	Tab Menu	Tab	Merupakan tabulasi untuk dapat berpindah ke menu lain dalam satu halaman.
2.	Formulir Filter	Formulir	Merupakan formulir yang diisi untuk melakukan filter data berupa jenis pajak dan tanggal.
3.	Tombol Cari	Tombol	Tombol yang digunakan untuk mencari data sesuai filter yang diisi.
4.	Tabel Data Laporan Pembukuan	Tabel	Merupakan tabel yang menampilkan data pembukuan berupa Nomor, Nama, Jenis Pajak,

			Tanggal Pembayaran, Target, dan Pembayaran.
5.	Tombol Cetak	Tombol	Tombol yang digunakan untuk mencetak data pembayaran pajak.

Tampilan laporan pembukuan atau keuangan pada tab pilihan “Semua” dapat dilihat pada Gambar 5.29.



Gambar 5.29 Rancangan Antarmuka Halaman Laporan Pembukuan/Keuangan - Semua

### 5.6.5 Halaman Perencanaan

#### A. Laporan target dan realisasi

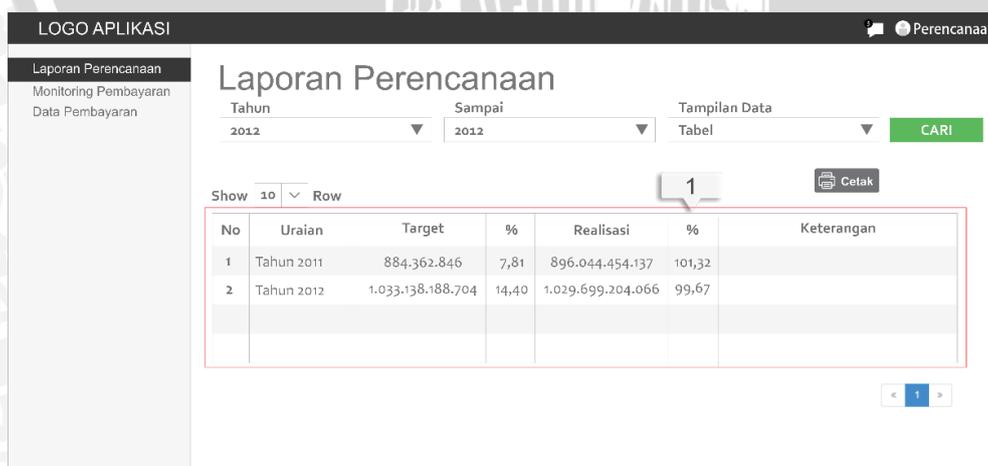


Gambar 5.30 Rancangan Antarmuka Halaman Laporan Target dan Realisasi - Grafik

Gambar 5.30 merupakan rancangan antarmuka dari halaman laporan target dan realisasi pajak yang ditampilkan dalam bentuk grafik. Penjelasan konten dari halaman tersebut dijelaskan pada Tabel 5.11.

**Tabel 5.11 Keterangan Rancangan Antarmuka Halaman Laporan Target dan Realisasi Pajak**

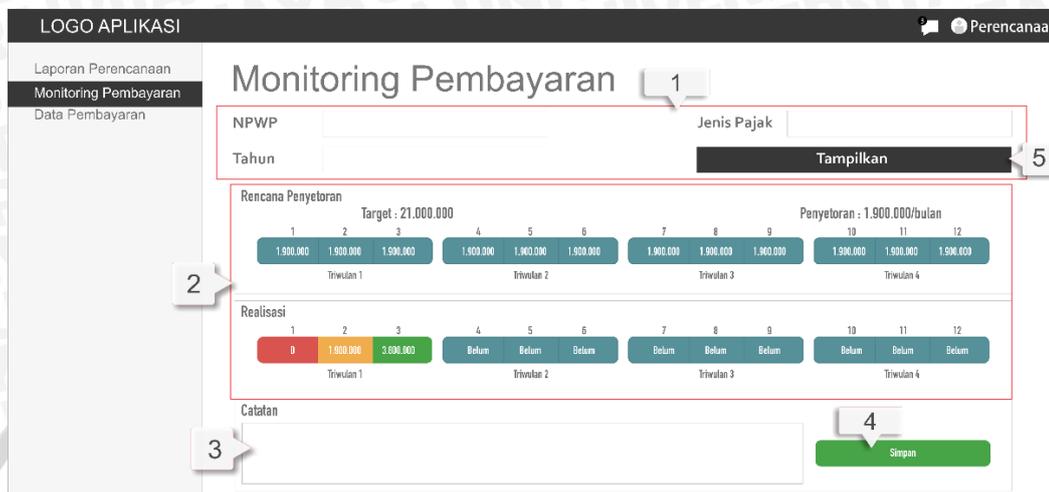
No.	Nama objek	Tipe	Keterangan
1.	Menu Bar	Tombol	Merupakan menu dari sistem yang dapat diakses oleh akun perencanaan, terdiri dari : - Laporan Perencanaan : Menampilkan halaman laporan perencanaan. - <i>Monitoring</i> Pembayaran: menampilkan halaman <i>monitoring</i> pembayaran. - Data Pembayaran : menampilkan menu data pembayaran.
2.	Formulir filter	Formulir	Merupakan formulir yang diisi untuk melakukan filter data, berupa Tahun dan Tampilan Data.
3.	Tombol Cari	Tombol	Tombol yang digunakan untuk mencari data sesuai dengan filter yang diisi.
4.	Grafik	Grafik	Tampilan data yang berupa grafik. Terdiri dua bar yaitu data berupa target dan realisasi pajak.
5.	Tombol Cetak	Tombol	Tombol yang digunakan untuk mencetak data pembayaran pajak.



**Gambar 5.31 Rancangan Antarmuka Halaman Laporan Target dan Realisasi - Tabel**

**Gambar 5.31** merupakan tampilan laporan perencanaan dalam bentuk tabel, yang ditunjukkan pada nomor 1 pada gambar merupakan data yang ditampilkan.

### B. Monitoring Pembayaran Pajak



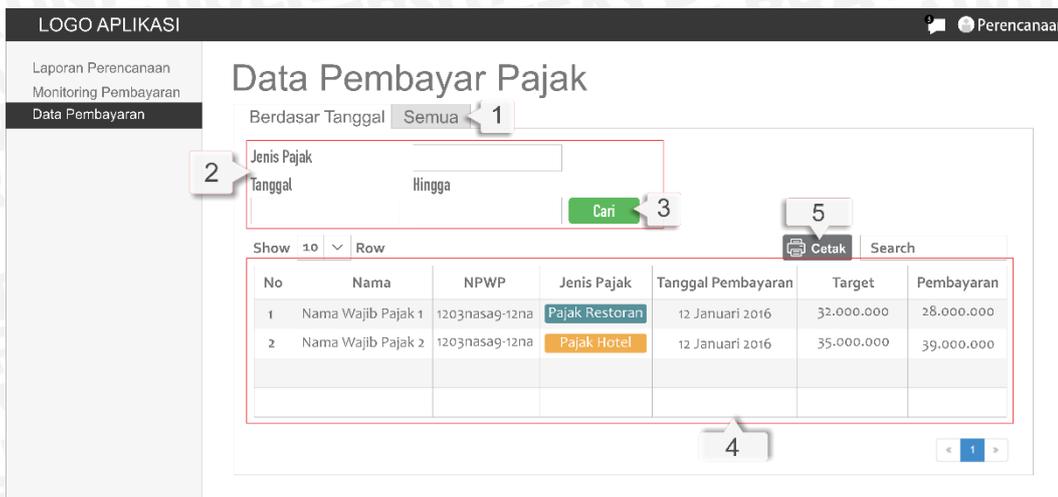
**Gambar 5.32** Rancangan Antarmuka Halaman *Monitoring Pembayaran*

Tabel 5.12 merupakan penjelasan konten dari rancangan antarmuka halaman *monitoring* pembayaran yang digambarkan pada Gambar 5.32.

**Tabel 5.12** Keterangan Rancangan Antarmuka Halaman *Monitoring Pembayaran*

No.	Nama objek	Tipe	Keterangan
1.	Formulir filter	Formulir	Merupakan formulir yang diisi untuk melakukan filter data, berupa NPWP, Tahun, dan Jenis Pajak.
2.	Data Pembayaran Pajak	Tabel	Merupakan data yang berupa tabel berisi rencana pembayaran yang telah ditetapkan dan realisasi pembayaran.
3.	Kolom Catatan	Teks area	Merupakan isian yang digunakan oleh bidang perencanaan untuk memberikan catatan mengenai pembayaran oleh wajib pajak.
4.	Tombol Simpan	Tombol	Tombol yang digunakan untuk menyimpan catatan perencanaan.
5.	Tombol Tampilkan	Tombol	Tombol yang digunakan untuk menampilkan data sesuai dengan filter yang diisi.

### C. Data Pembayaran



**Gambar 5.33 Rancangan Antarmuka Halaman Data Pembayaran**

Pada Gambar 5.33 dan Gambar 5.34 menunjukkan rancangan antarmuka dari halaman data pembayaran dan konten yang terdapat pada halaman tersebut dijelaskan pada Tabel 5.13.

**Tabel 5.13 Keterangan Rancangan Antarmuka Halaman Data Pembayaran**

No.	Nama objek	Tipe	Keterangan
1.	Tab Menu	Tab	Merupakan tabulasi untuk dapat berpindah ke menu lain dalam satu halaman.
2.	Formulir Filter	Formulir	Merupakan formulir yang diisi untuk melakukan filter data, berupa Jenis Pajak dan Tanggal.
3.	Tombol Cari	Tombol	Tombol yang digunakan untuk menampilkan data sesuai dengan filter yang diisi.
4.	Tabel Data Laporan Pembayaran	Tabel	Merupakan tabel yang menampilkan data pembayaran wajib pajak berupa Nomor, Nama, NPWP, Jenis Pajak, Tanggal Pembayaran, Target, dan Pembayaran.
5.	Tombol Cetak	Tombol	Tombol yang digunakan untuk mencetak data pembayaran pajak.

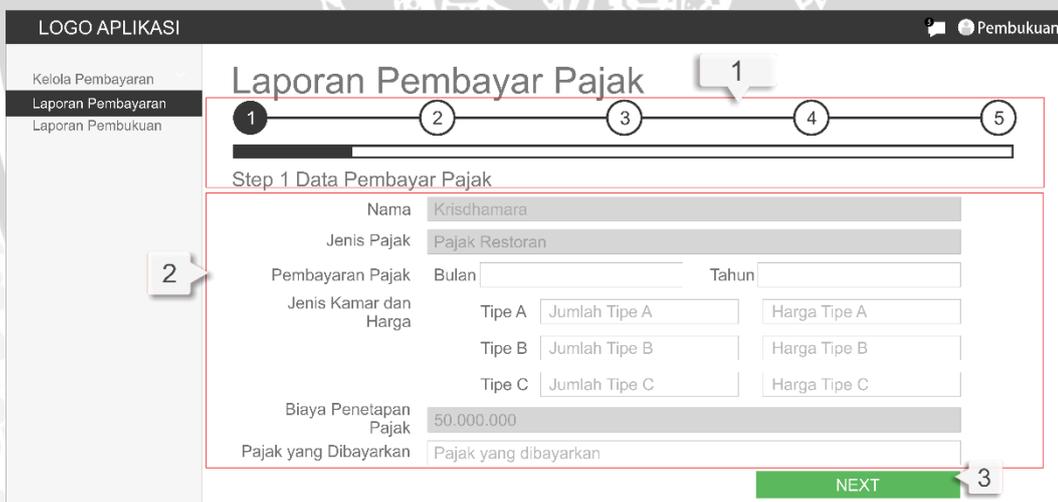


**Gambar 5.34 Rancangan Antarmuka Halaman Data Pembayaran - Semua**

Pada Gambar 5.34 merupakan tampilan data pembayaran pada tab pilihan semua. Menampilkan semua data pembayaran tanpa filter.

### 5.6.6 Alur Laporan Pembayaran

#### A. Step 1



**Gambar 5.35 Rancangan Antarmuka Halaman Alur Laporan Step 1**

Rancangan antarmuka dari halaman alur laporan step 1 ditunjukkan pada Gambar 5.35 dengan penjelasan konten dari halaman tersebut dijelaskan pada Tabel 5.14.

**Tabel 5.14 Keterangan Rancangan Antarmuka Halaman Alur Laporan Step 1**

No.	Nama objek	Tipe	Keterangan
1.	Tahap Laporan Pembayaran	Alur	Merupakan tanda yang menunjukkan pada tahapan berapa proses pelaporan pembayaran dilakukan.
2.	Formulir Laporan Pembayaran	Formulir	Merupakan formulir yang diisi oleh wajib pajak saat akan melaporkan pembayaran pajak, data yang harus diisi adalah jumlah pajak yang dibayarkan.
3.	Tombol Selanjutnya	Tombol	Tombol yang digunakan untuk masuk ke langkah atau tahapan selanjutnya.

**B. Step 2**



**Gambar 5.36 Rancangan Antarmuka Halaman Alur Laporan Step 2**

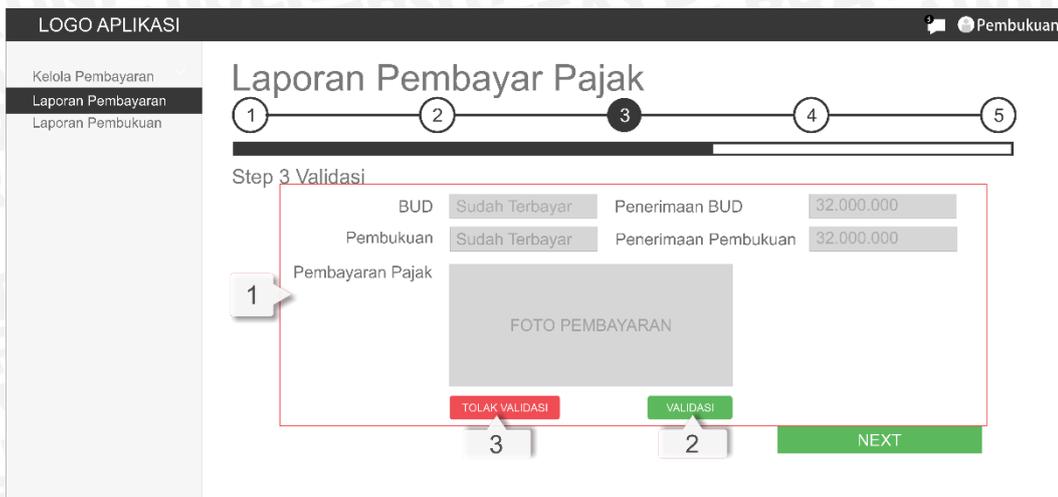
Tabel 5.15 merupakan penjelasan konten dari rancangan antarmuka halaman alur laporan step 2 yang ditunjukkan seperti pada Gambar 5.36.

**Tabel 5.15 Keterangan Rancangan Antarmuka Halaman Alur Laporan Step 2**

No.	Nama objek	Tipe	Keterangan
1.	Formulir <i>Upload</i> Bukti Pembayaran	Formulir	Merupakan formulir yang digunakan untuk mengunggah foto bukti pembayaran pajak.
2.	Tombol Pilih Gambar	Tombol	Merupakan tombol yang digunakan untuk memilih berkas foto bukti pembayaran pajak.



C. Step 3 dan 4



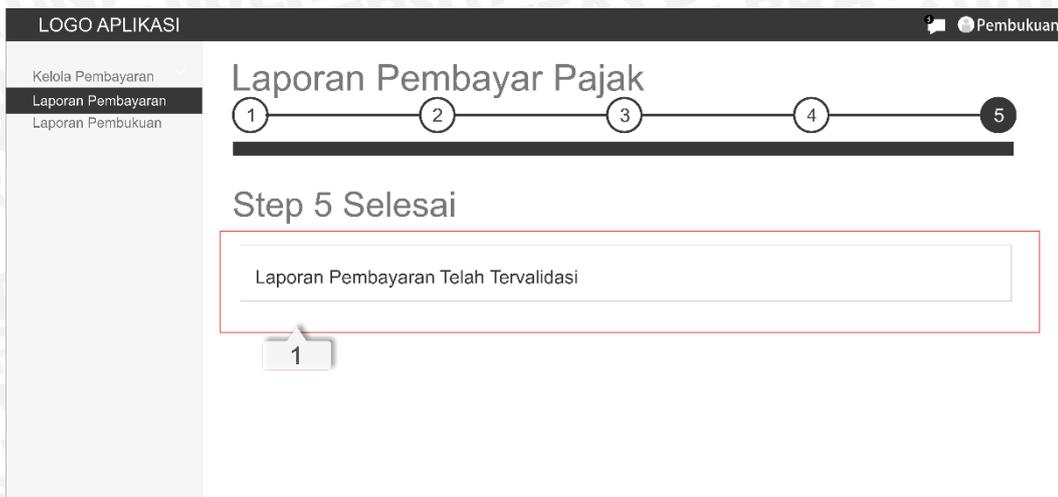
Gambar 5.37 Rancangan Antarmuka Halaman Alur Laporan Step 3 dan 4

Rancangan antarmuka halaman alur laporan step 3 dan 4 ditunjukkan pada Gambar 5.37 dengan penjelasan konten pada Tabel 5.16.

Tabel 5.16 Keterangan Rancangan Antarmuka Halaman Alur Laporan Step 3 dan 4

No.	Nama objek	Tipe	Keterangan
1.	Formulir Keterangan Pembayaran	Formulir	Merupakan formulir yang menampilkan keterangan pembayaran oleh wajib pajak.
2.	Tombol Validasi	Tombol	Tombol yang digunakan oleh BUD atau bidang pembukuan untuk melakukan validasi terhadap laporan dari wajib pajak.
3.	Tombol Tolak Validasi	Tombol	Tombol yang digunakan oleh BUD atau pembukuan untuk menolak validasi pelaporan pembayaran pajak.

D. Step 5



Gambar 5.38 Rancangan Antarmuka Halaman Alur Laporan Step 5

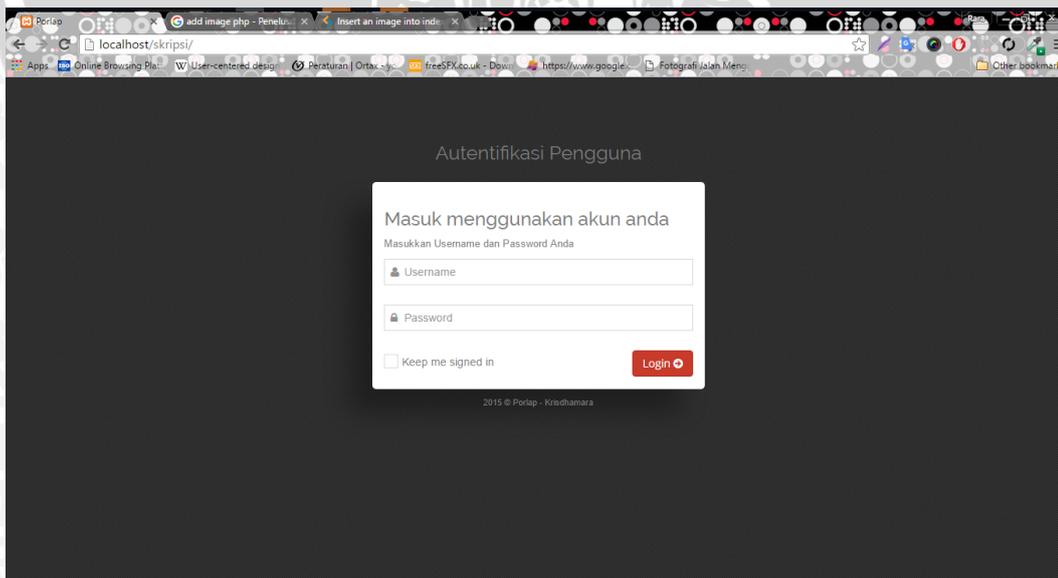
Pada Gambar 5.38 menunjukkan rancangan antarmuka dari halaman alur laporan step 5 dengan penjelasan konten pada Tabel 5.17.

Tabel 5.17 Keterangan Rancangan Antarmuka Halaman Alur Laporan Step 5

No.	Nama objek	Tipe	Keterangan
1.	Keterangan status laporan	Teks	Keterangan bahwa laporan pembayaran telah tervalidasi.

5.7 Prototipe Sistem

5.7.1 Halaman Login

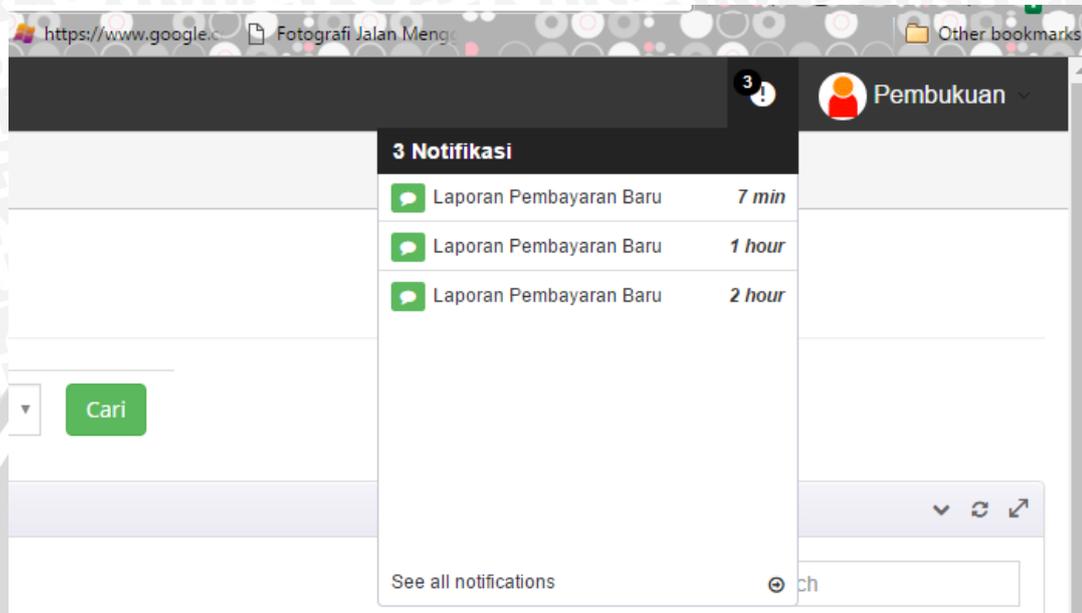


Gambar 5.39 Antarmuka Halaman Login



Pada Gambar 5.39 merupakan prototipe sistem halaman *login*. Prototipe tersebut telah disesuaikan dengan rancangan antarmuka yang telah dibuat sebelumnya yaitu pada Gambar 5.20. Untuk melakukan *login* terdapat 2 *file input* yang harus diisi yaitu *username* dan *password* dan terdapat tombol untuk *login*.

### 5.7.2 Notifikasi

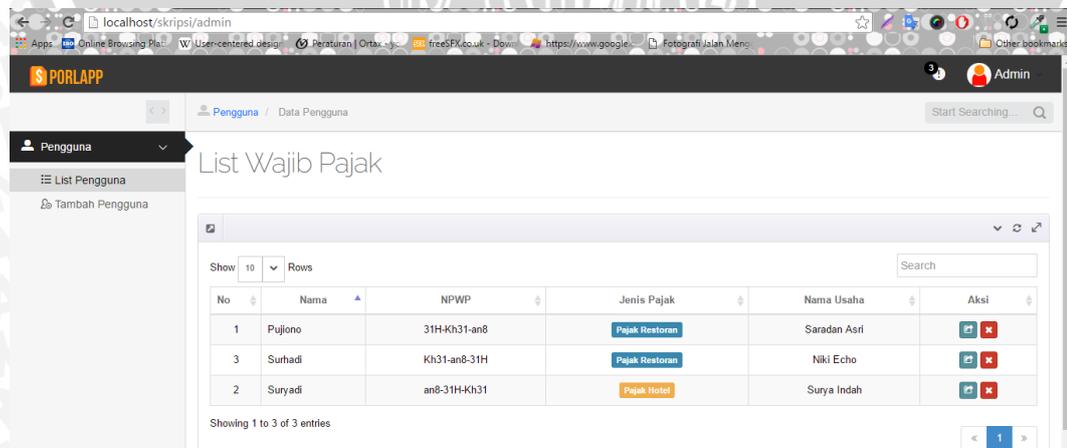


Gambar 5.40 Antarmuka Tampilan Notifikasi

Pada Gambar 5.40 merupakan prototipe tampilan notifikasi. Prototipe ini telah disesuaikan dengan perancangan antarmuka pada Gambar 5.21. Notifikasi dapat terlihat jika pengguna menekan tombol notifikasi yang berada di samping tombol akun pengguna.

### 5.7.3 Halaman Admin

#### A. Daftar Pengguna



Gambar 5.41 Antarmuka Halaman Daftar Pengguna



Pada Gambar 5.41 merupakan prototipe halaman daftar pengguna yang berada pada *dashboard* admin. Halaman ini telah disesuaikan dengan rancangan antarmuka yang ada pada Gambar 5.22.

## B. Tambah Pengguna

The screenshot shows a web browser window displaying the 'Tambah Pengguna' form. The form is titled 'PORLAPP' and is accessed by an 'Admin' user. The form contains the following fields and values:

Field	Value
Nama Usaha *	Surya Indah
Lokasi *	Geger
Jumlah Kamar dan Harga	Tipe A: 30 (150.000) Tipe B: 20 (225.000) Tipe C: 20 (400.000)
Biaya Penetapan Pajak *	18.250.000
Username *	Username
Password *	Password

At the bottom of the form, there are two buttons: 'Batal' (Cancel) and 'Register'.

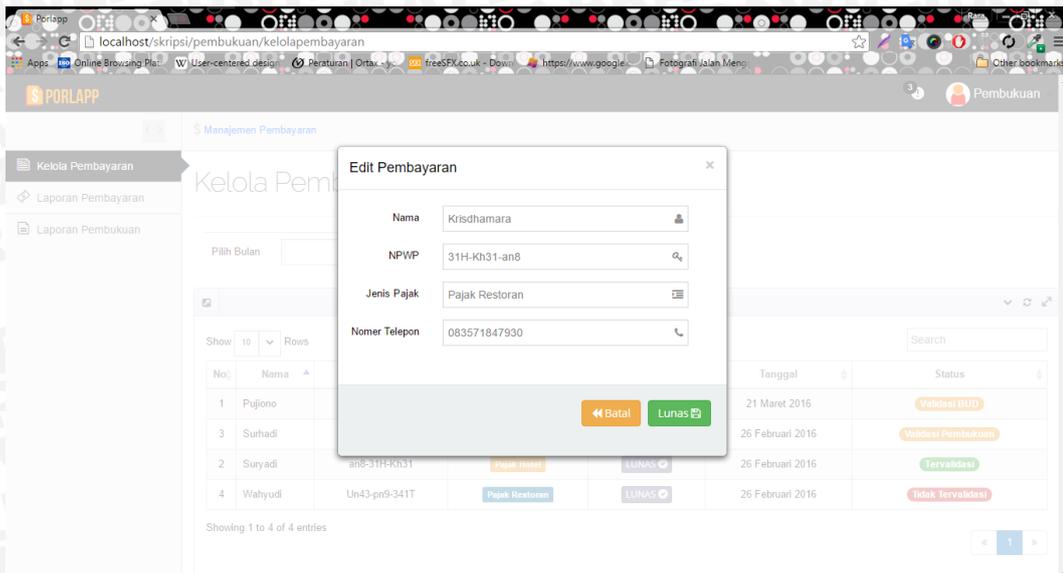
**Gambar 5.42 Antarmuka Halaman Tambah Pengguna**

Pada Gambar 5.42 merupakan prototipe halaman tambah pengguna yang dapat diakses oleh *admin* pada *dashboard* admin. Halaman ini berguna untuk menambahkan pengguna baru. Halaman ini juga telah disesuaikan dengan rancangan antarmuka pada Gambar 5.23.

## 5.7.4 Halaman Pembukuan dan Bendahara Umum Daerah

### A. Kelola Pembayaran - Ubah Status Pembayaran

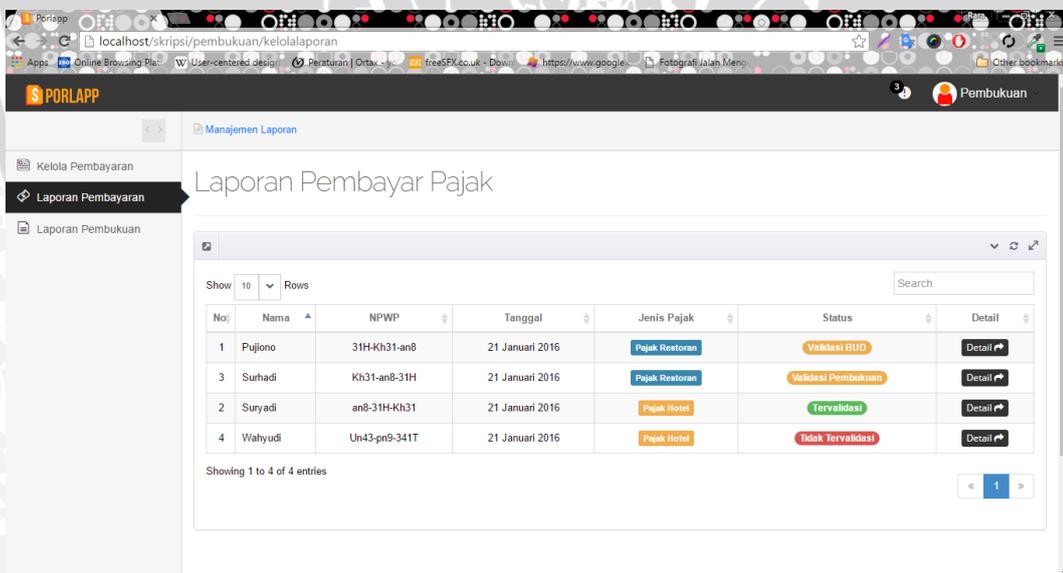
Pada Gambar 5.43 merupakan prototipe antarmuka kelola pembayaran. Halaman ini dapat diakses oleh aktor bagian pembukuan. Halaman ini berguna untuk mengedit data pembayaran. Halaman ini juga telah disesuaikan dengan rancangan antarmuka pengguna pada Gambar 5.26.



Gambar 5.43 Antarmuka Halaman Kelola Pembayaran

### B. Laporan Pembayaran

Pada Gambar 5.44 merupakan prototipe halaman laporan pembayaran yang ada pada *dashboard* bagian pembukuan. Halaman ini berisi daftar nama-nama pembayaran pajak dan data pembayarannya. Halaman ini telah disesuaikan dengan rancangan antarmuka pengguna yang ada pada Gambar 5.27.

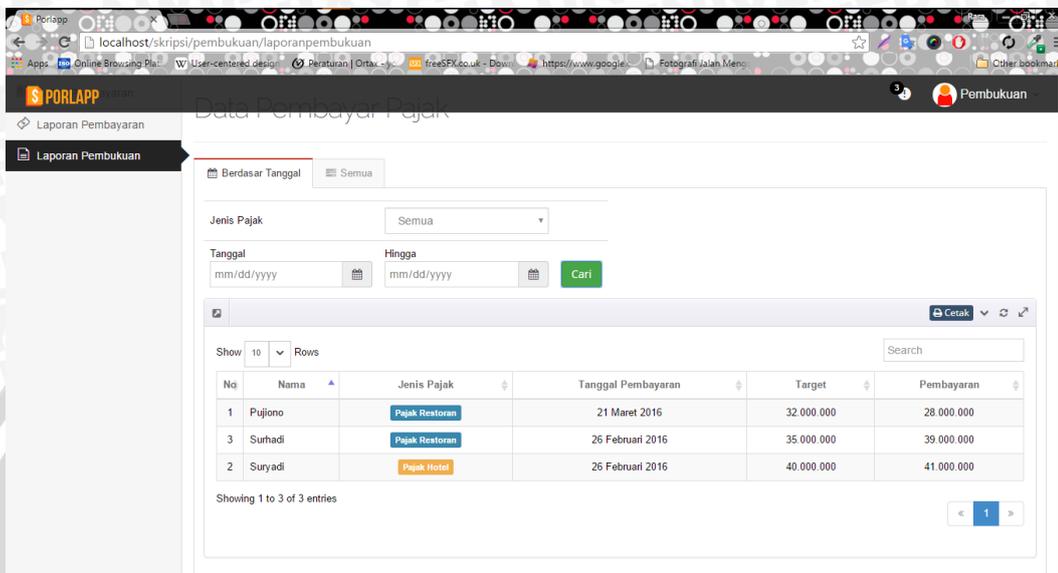


Gambar 5.44 Antarmuka Halaman Laporan Pembayaran



### C. Laporan Pembukuan/Keuangan

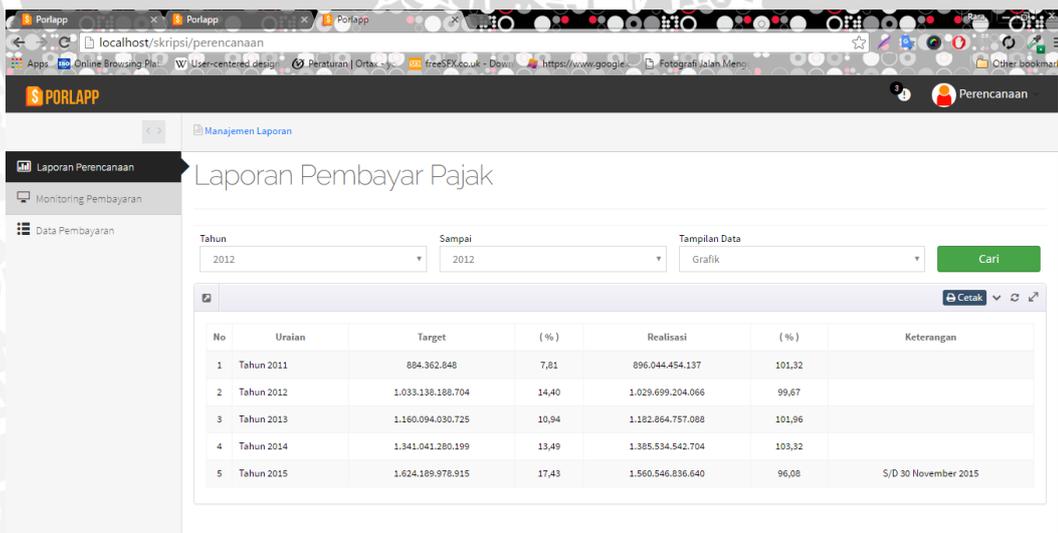
Pada Gambar 5.45 merupakan hasil prototipe dari halaman laporan pembukuan atau keuangan yang ada pada *dashboard* bagian pembukuan. Halaman ini telah disesuaikan dengan rancangan antarmuka pada Gambar 5.28.



Gambar 5.45 Antarmuka Halaman Laporan Pembukuan/Keuangan

## 5.7.5 Halaman Perencanaan

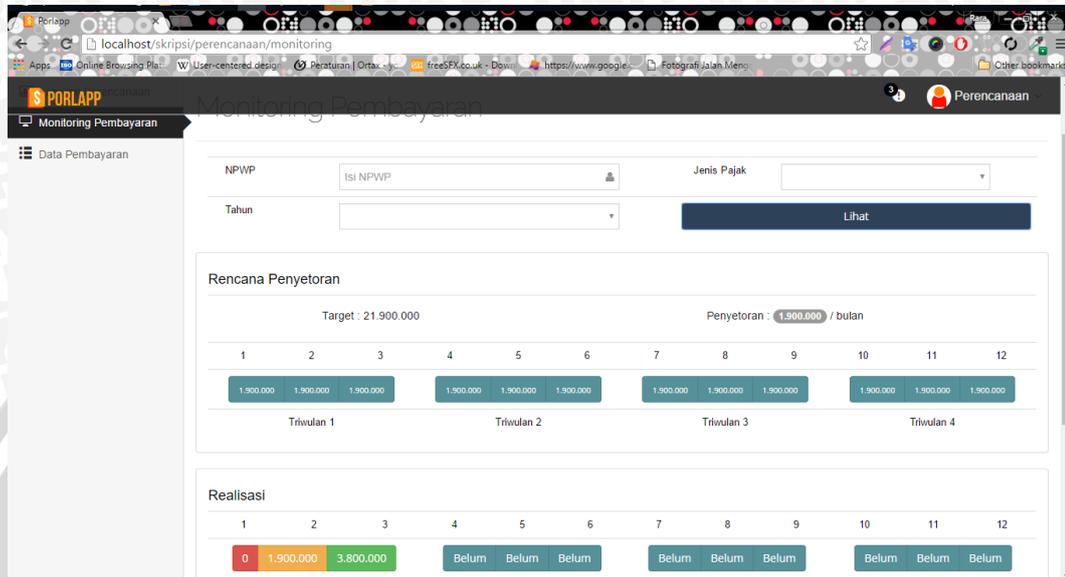
### A. Laporan target dan realisasi



Gambar 5.46 Antarmuka Halaman Laporan Target dan Realisasi

Pada Gambar 5.46 merupakan prototipe halaman laporan target dan realisasi yang dapat diakses oleh pengguna perencanaan. Halaman ini disesuaikan dengan hasil rancangan antarmuka yang telah dibuat pada Gambar 5.31.

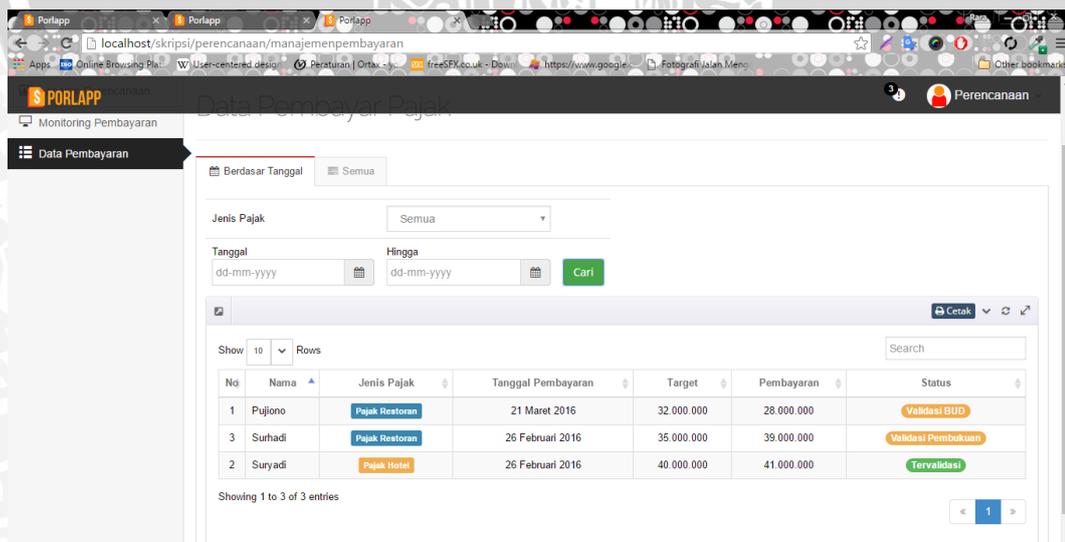
### B. Monitoring Pembayaran Pajak



Gambar 5.47 Antarmuka Halaman *Monitoring* Pembayaran Pajak

Pada Gambar 5.47 menunjukkan prototipe antarmuka halaman *monitoring* pembayaran pajak. Halaman ini diakses oleh pengguna perencanaan. Halaman ini juga telah disesuaikan dengan hasil rancangan yang ada pada Gambar 5.32.

### C. Data Pembayaran

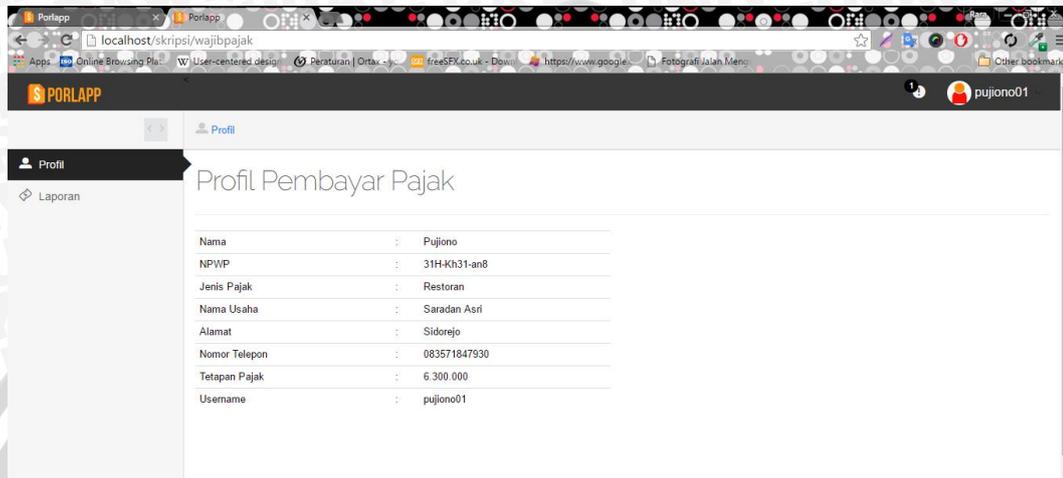


Gambar 5.48 Antarmuka Halaman Data Pembayaran

Pada Gambar 5.48 merupakan prototipe halaman data pembayaran, halaman ini dapat diakses oleh pengguna perencanaan. Halaman ini juga telah disesuaikan dengan hasil rancangan yang ada pada Gambar 5.33.

## 5.7.6 Halaman Wajib Pajak

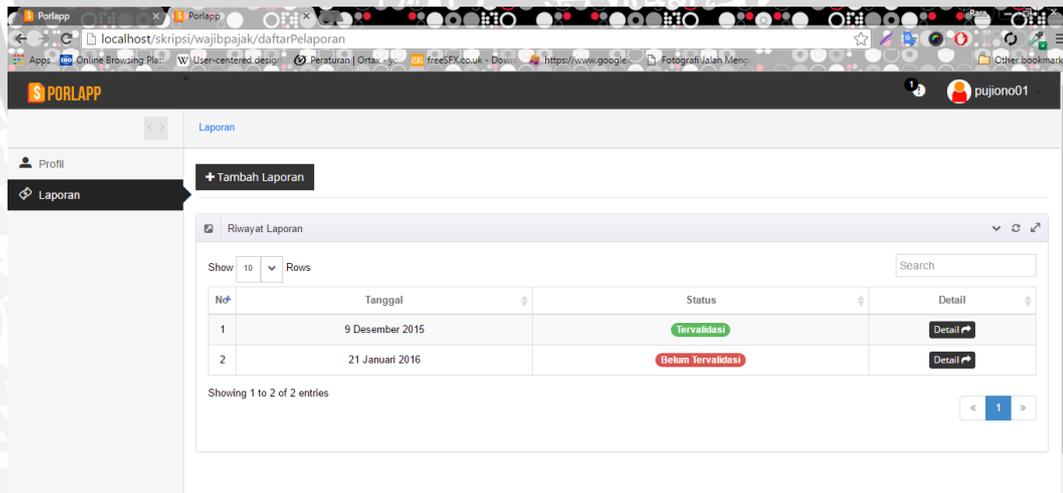
### A. Halaman Profil



**Gambar 5.49 Antarmuka Halaman Profil Pengguna**

Pada Gambar 5.49 merupakan prototipe halaman profil pengguna wajib pajak. Halaman ini berisi data-data pembuat pajak, halaman ini dapat diakses oleh pengguna wajib pajak.

### B. Halaman Pelaporan Pembayaran Pajak

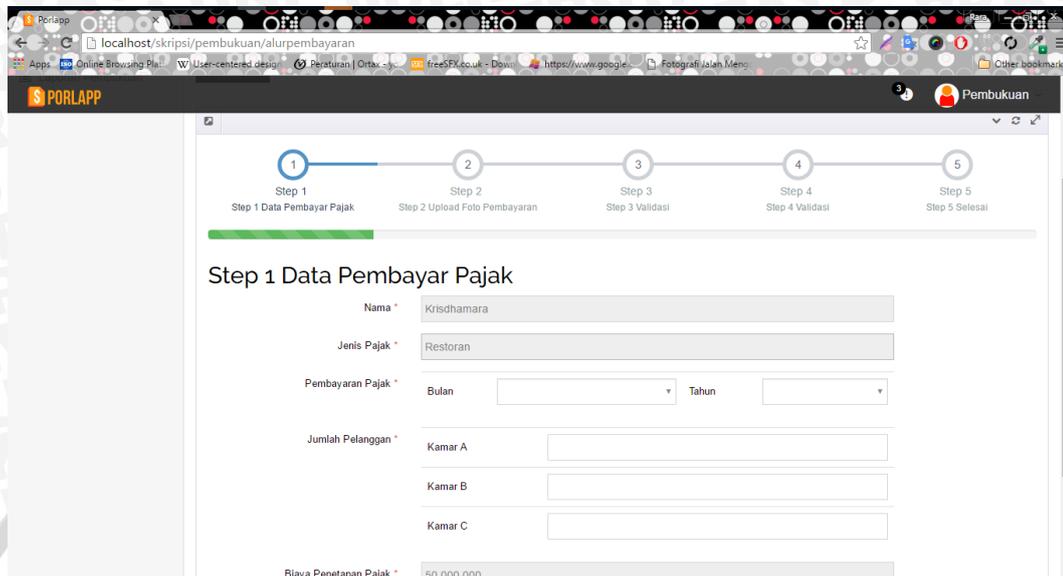


**Gambar 5.50 Antarmuka Halaman Pelaporan Pembayaran Pajak**

Pada Gambar 5.50 merupakan antarmuka prototipe halaman pelaporan pembayaran pajak. Halaman ini dapat diakses oleh para wajib pajak. Halaman ini berisi status pembayar pajak dan detail dari laporan pembayaran pajak.

## 5.7.7 Alur Laporan Pembayaran

### A. Step 1



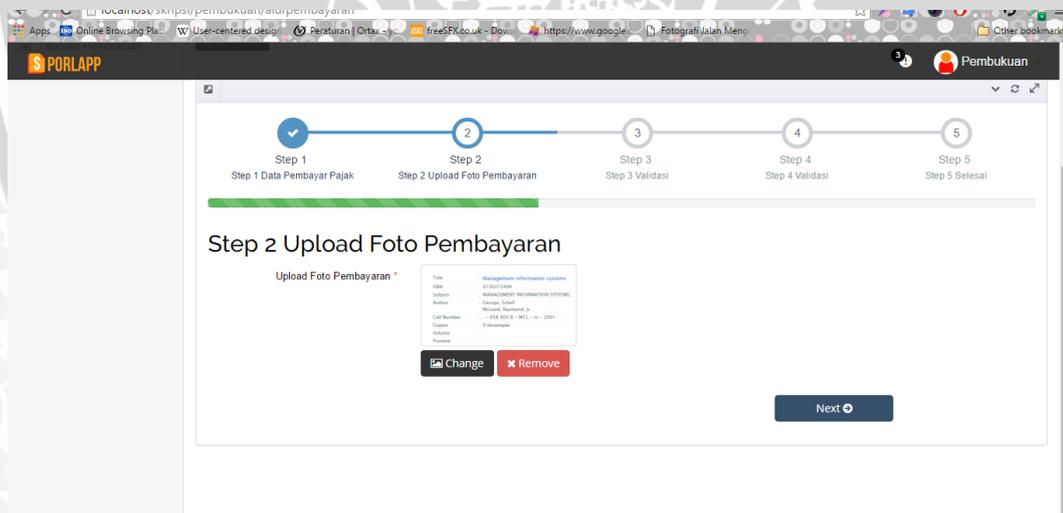
The screenshot shows a web browser displaying the 'Step 1 Data Pembayar Pajak' form. At the top, a progress bar indicates five steps: Step 1 (active), Step 2, Step 3, Step 4, and Step 5. The form fields are as follows:

- Nama \***: Krisdhamara
- Jenis Pajak \***: Restoran
- Pembayaran Pajak \***: Bulan (dropdown), Tahun (dropdown)
- Jumlah Pelanggan \***: Kamar A, Kamar B, Kamar C (text input fields)
- Biaya Penetapan Pajak \***: 50.000.000

**Gambar 5.51 Antarmuka Halaman Alur Laporan Pembayaran Step 1**

Pada Gambar 5.51 merupakan prototipe halaman alur laporan pembayaran step 1. Halaman ini telah disesuaikan dengan rancangan antarmuka yang ada pada Gambar 5.35. pada halaman ini berisi tentang identitas para pembayar pajak.

### B. Step 2



The screenshot shows the 'Step 2 Upload Foto Pembayaran' form. The progress bar at the top shows Step 1 as completed and Step 2 as active. The form includes:

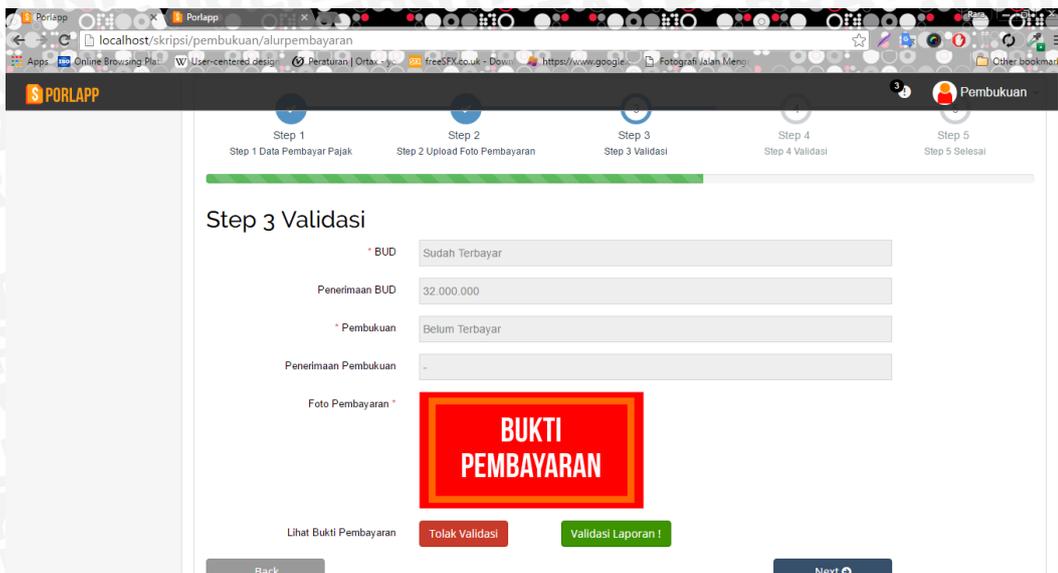
- Upload Foto Pembayaran \***: A file upload area with a 'Change' button and a 'Remove' button.
- Next**: A blue button to proceed to the next step.
- Information System Details**: A table with the following data:

Management Information System	
URL	01.00115404
Webster	00000000000000000000000000000000
Author	Genes, Email
	Widada, Rameza, D
Call Number	000 000 000 000 000 000 000 000 000 000
Color	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Volume	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Pages	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

**Gambar 5.52 Antarmuka Halaman Alur Laporan Pembayaran Step 2**

Pada Gambar 5.52 merupakan prototipe alur laporan pembayaran step 2. Pada halaman ini berisi tentang *upload file* foto pembayaran. Halaman ini telah disesuaikan dengan rancangan yang telah dilakukan pada Gambar 5.36.

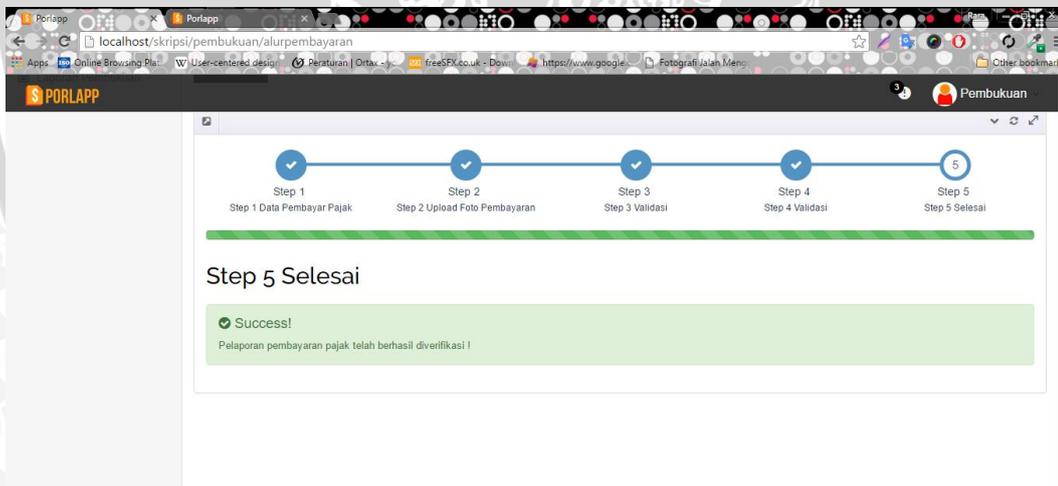
**C. Step 3 dan 4**



**Gambar 5.53 Antarmuka Halaman Alur Laporan Pembayaran Step 3 dan 4**

Pada Gambar 5.53 merupakan halaman alur laporan pembayaran step 3 dan 4. Halaman ini merupakan halaman validasi dari pembayaran. Validasi dilakukan dengan cara melihat foto bukti pembayaran yang telah dikirim oleh wajib pajak. Halaman ini telah disesuaikan dengan rancangan antarmuka pada Gambar 5.37.

**D. Step 5**



**Gambar 5.54 Antarmuka Halaman Alur Laporan Pembayaran Step 5**

Pada Gambar 5.54 merupakan halaman alur laporan pembayaran step 5 yang merupakan notifikasi bahwa laporan pembayaran telah selesai dilakukan.



## BAB 6 EVALUASI

### 6.1 Evaluasi

Ada dua macam evaluasi yang dilakukan pada penelitian ini yaitu (1) peninjauan prototipe terhadap *use case* dan desain dasar antarmuka pengguna; dan (2) peninjauan kerunutan. Evaluasi peninjauan prototipe menggunakan metode *walkthrough* yang bertujuan untuk mengetahui apakah prototipe yang dirancang sudah sesuai dengan persyaratan yang telah didefinisikan. Selain itu, evaluasi ini juga bertujuan untuk menggali saran dan kekurangan yang ada pada sistem dari sudut pandang pengguna. Evaluasi yang dilakukan melibatkan pengguna dari sistem yang telah didefinisikan dan mengambil beberapa sampel pengguna. Sampel yang ditentukan untuk melakukan evaluasi ini adalah tiga orang.

Evaluasi peninjauan kerunutan dilakukan oleh peneliti sendiri dengan membuat matriks kerunutan dan melakukan pengecekan kerunutan dengan menggunakan daftar pengecekan persyaratan berdasarkan aspek kerunutan. Penjelasan lebih lanjut tentang hasil kedua macam evaluasi tersebut dapat dilihat pada beberapa sub bab berikut.

#### 6.1.1 Peninjauan Prototipe

##### A. Peninjauan *Use Case*

Evaluasi pada bagian ini merupakan evaluasi yang melibatkan pengguna dari sistem dimana dari sampel pengguna yang telah ditentukan diminta untuk mencoba prototipe yang telah dirancang dan diberikan kasus uji dari beberapa *use case* yang telah dibuat. Dari masing-masing kasus uji tersebut dapat diketahui bagaimana pengguna menggunakan prototipe sistem sesuai dengan *use case* yang telah ditentukan, apakah langkah-langkah yang dijalankan pengguna telah sesuai dengan alur pada spesifikasi *use case*.

Dari tiga responden yang melakukan kasus uji, diketahui bahwa setiap pengguna telah melakukan kasus uji dari beberapa *use case* dengan langkah-langkah yang sesuai dengan alur dasar pada spesifikasi *use case*. Adapun hasil tinjauan dari ketiga sampel tersebut dapat dilihat pada Tabel 6.1, Tabel 6.2, dan Tabel 6.3.

Pengguna 1 : Budi Muratno, S. Sos (Kepala bidang Perencanaan, DISPENDA Kab. Madiun)

**Tabel 6.1 Hasil Review *Use case* Pengguna 1**

<i>Use case</i>	Registrasi Pengguna		
Test Case	Berhasil melakukan registrasi : Basic Flow		
Langkah-langkah Pengguna	Alur Seharusnya	Cek	Keterangan
1. Memilih menu kelola pengguna.	1. Memilih menu registrasi pengguna.	✓	Sudah sesuai, tapi saat awal salah

2. Memilih menu registrasi pengguna. 3. Memilih jenis pengguna. 4. Mengisi formulir. 5. Menyimpan data.	2. Memilih jenis pengguna. 3. Mengisi formulir. 4. Menyimpan data.		memilih menu kelola pengguna.
<b>Use case</b>		<b>Registrasi Pengguna</b>	
<b>Test Case</b>		<b>Data yang diisi tidak lengkap : Basic Flow, Alternative Flow</b>	
<b>Langkah-langkah Pengguna</b>	<b>Alur Seharusnya</b>	<b>Cek</b>	<b>Keterangan</b>
1. Memilih menu registrasi pengguna. 2. Memilih jenis pengguna. 3. Mengisi formulir dan mengosongkan kolom <i>username</i> dan <i>password</i> . 4. Memilih tombol simpan. 5. Menampilkan pemberitahuan kolom harus diisi pada kolom <i>username</i> dan <i>password</i> .	1. Memilih menu registrasi pengguna. 2. Memilih jenis pengguna. 3. Mengisi formulir dan mengosongkan salah satu atau beberapa kolom. 4. Memilih tombol simpan. 5. Menampilkan pemberitahuan kolom harus diisi.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .
<b>Use Case</b>		<b>Kelola Data Pengguna</b>	
<b>Test Case</b>		<b>Edit data pengguna : Basic Flow, Alternative Flow</b>	
<b>Langkah-langkah Pengguna</b>	<b>Alur Seharusnya</b>	<b>Cek</b>	<b>Keterangan</b>
1. Memilih menu kelola pengguna. 2. Memilih detail pengguna. 3. Mengubah <i>username</i> pengguna. 4. Memilih tombol simpan.	1. Memilih menu kelola pengguna. 2. Memilih detail pengguna. 3. Mengedit data pengguna. 4. Memilih tombol simpan.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .
<b>Use case</b>		<b>Monitoring Pembayaran Pajak</b>	
<b>Test Case</b>		<b>Melakukan monitoring pembayaran pajak</b>	
<b>Langkah-langkah Pengguna</b>	<b>Alur Seharusnya</b>	<b>Cek</b>	<b>Keterangan</b>
1. Memilih menu monitoring pembayaran. 2. Mengisi NPWP, tahun dan jenis pajak. 3. Memilih tombol Cari.	1. Memilih menu monitoring pembayaran. 2. Mengisi NPWP, tahun dan jenis pajak. 3. Memilih tombol Cari. 4. Mengisi catatan pembayaran.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .

4. Mengisi catatan pembayaran.	5. Memilih tombol simpan.		
5. Memilih tombol simpan.			

Pengguna 2 : Ari Nursurahmat (Kepala bidang Penagihan & Keberatan, DISPENDA Kab. Madiun)

**Tabel 6.2 Hasil Review Use case Pengguna 2**

Use case		Registrasi Pengguna		
Test Case		Berhasil melakukan registrasi : Basic Flow		
Langkah-langkah Pengguna	Alur Seharusnya	Cek	Keterangan	
1. Memilih menu registrasi pengguna.	1. Memilih menu registrasi pengguna.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .	
2. Memilih jenis pengguna.	2. Memilih jenis pengguna.			
3. Mengisi formulir.	3. Mengisi formulir.			
4. Menyimpan data.	4. Menyimpan data.			
Use case		Registrasi Pengguna		
Test Case		Data yang diisi tidak lengkap : Basic Flow, Alternative Flow		
Langkah-langkah Pengguna	Alur Seharusnya	Cek	Keterangan	
1. Memilih menu registrasi pengguna.	1. Memilih menu registrasi pengguna.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .	
2. Memilih jenis pengguna.	2. Memilih jenis pengguna.			
3. Mengisi formulir dan mengosongkan kolom <i>username</i> .	3. Mengisi formulir dan mengosongkan salah satu atau beberapa kolom.			
4. Memilih tombol simpan.	4. Memilih tombol simpan.			
5. Menampilkan pemberitahuan kolom harus diisi pada kolom <i>username</i> .	5. Menampilkan pemberitahuan kolom harus diisi.			
Use case		Kelola Data Pengguna		
Test Case		Edit data pengguna : Basic Flow, Alternative Flow		
Langkah-langkah Pengguna	Alur Seharusnya	Cek	Keterangan	
1. Memilih menu kelola pengguna.	1. Memilih menu kelola pengguna.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .	
2. Memilih detail pengguna.	2. Memilih detail pengguna.			

3. Mengubah <i>username</i> pengguna.	3. Mengedit data pengguna.		
4. Memilih tombol simpan.	4. Memilih tombol simpan.		
<b>Use case</b>	<b>Monitoring Pembayaran Pajak</b>		
<b>Test Case</b>	<b>Melakukan monitoring pembayaran pajak</b>		
<b>Langkah-langkah Pengguna</b>	<b>Alur Seharusnya</b>	<b>Cek</b>	<b>Keterangan</b>
1. Memilih menu monitoring pembayaran. 2. Mengisi NPWP, tahun dan jenis pajak. 3. Memilih tombol Cari. 4. Mengisi catatan pembayaran. 5. Memilih tombol simpan.	1. Memilih menu monitoring pembayaran. 2. Mengisi NPWP, tahun dan jenis pajak. 3. Memilih tombol Cari. 4. Mengisi catatan pembayaran. 5. Memilih tombol simpan.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .

Pengguna 3 : Abdul Aziz Hadyansya (Mahasiswa)

**Tabel 6.3 Hasil Review Use case Pengguna 3**

<b>Use case</b>	<b>Tambah Laporan Pajak</b>		
<b>Test Case</b>	<b>Berhasil menambahkan laporan : Basic Flow</b>		
<b>Langkah-langkah Pengguna</b>	<b>Alur Seharusnya</b>	<b>Cek</b>	<b>Keterangan</b>
1. Memilih menu Laporan Pajak. 2. Memilih tambah laporan. 3. Mengisi formulir data pembayaran pajak. 4. Memilih pilihan selanjutnya. 5. Memilih foto bukti pembayaran pajak. 6. Memilih pilihan selesai.	1. Memilih menu Laporan Pajak. 2. Memilih tambah laporan. 3. Mengisi formulir data pembayaran pajak. 4. Memilih pilihan selanjutnya. 5. Memilih foto bukti pembayaran pajak. 6. Memilih pilihan selesai.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .
<b>Use case</b>	<b>Tambah Laporan Pajak</b>		
<b>Test Case</b>	<b>Data yang diisi tidak lengkap : Basic Flow, Alternative Flow</b>		
<b>Langkah-langkah Pengguna</b>	<b>Alur Seharusnya</b>	<b>Cek</b>	<b>Keterangan</b>
1. Memilih menu Laporan Pajak. 2. Memilih tambah laporan. 3. Mengisi formulir data pembayaran pajak. 4. Kolom pajak yang dibayarkan tidak diisi. 5. Memilih pilihan selanjutnya.	1. Memilih menu Laporan Pajak. 2. Memilih tambah laporan. 3. Mengisi formulir data pembayaran pajak. 4. Satu atau beberapa kolom formulir tidak diisi 5. Memilih pilihan selanjutnya.	✓	Sudah sesuai dengan alur seharusnya pada skenario <i>use case</i> .



6. Menampilkan pemberitahuan kolom harus diisi.	6. Menampilkan pemberitahuan kolom harus diisi.		
---	---	--	--

Setelah ketiga pengguna selesai melakukan percobaan prototipe dengan beberapa kasus uji, mereka diberi beberapa pertanyaan tinjauan yang berhubungan dengan alur dasar (*basic flow*) pada *use case* untuk mengetahui pendapat pengguna mengenai alur dari *use case* yang dijadikan kasus uji dan menggali saran pengguna. Dari pertanyaan yang diajukan dan dijawab oleh ketiga pengguna tersebut didapatkan hasil bahwa alur *use case* yang diujikan sudah sesuai dengan yang diharapkan. Jenis data yang dibutuhkan dan penanganan kesalahan pada setiap *use case* yang diujikan pun telah sesuai dengan harapan pengguna. Hasil pertanyaan dan jawaban dari pengguna telah dilampirkan pada Lampiran A Gambar A. 1, Gambar A. 4, dan Gambar A. 7.

### B. Peninjauan Rancangan Dasar Antarmuka Pengguna

Evaluasi pada bagian ini bertujuan untuk mengetahui bagaimanakah pendapat pengguna tentang rancangan prototipe yang telah dibuat setelah mencoba menjalankan prototipe terhadap kriteria navigasi dan antarmuka dari prototipe yang dibuat dan menggali saran dari ketiga sampel pengguna mengenai rancangan antarmuka yang telah dibuat. Dari hasil tinjauan yang dilakukan oleh ketiga pengguna tersebut, didapatkan hasil bahwa dua pengguna memilih setuju dengan semua butir-butir pertanyaan yang diajukan. Namun, ada satu pengguna yang menyatakan tidak setuju pada salah satu butir pertanyaan navigasi yaitu terdapat *link* yang tidak diketahui dengan jelas fungsinya. Dari peninjauan yang dilakukan ada beberapa saran yang diberikan yaitu :

1. Pada menu laporan perencanaan, menambahkan detail filter tanggal atau bulan agar data yang ditampilkan lebih detail.
2. Menambahkan logo transparan DISPENDA Kabupaten Madiun pada latar belakang tampilan prototipe.
3. Menambahkan tampilan menu selamat datang dan tampilan menu utama pada layar awal setelah *login*.

Hasil dari peninjauan antarmuka pengguna telah dilampirkan pada Lampiran A Gambar A. 2, Gambar A. 3, Gambar A. 5, Gambar A. 6, dan Gambar A. 8.

#### 6.1.2 Peninjauan Kerunutan (*Traceability*)

Pada peninjauan ini dibuat matriks kerunutan yang digunakan untuk mengetahui apakah perancangan dasar yang dibuat telah sesuai dengan persyaratan fungsional dan *use case* yang dibuat. Pada matriks kerunutan ini akan menyesuaikan antara aktivitas baru pada proses bisnis, persyaratan fungsional, *use case*, diagram aktivitas, *sequence diagram* dan antarmuka pengguna. Kemudian akan dilakukan pengecekan persyaratan berdasarkan kerunutan yang telah dibuat. Pada Tabel 6.4 merupakan matriks kerunutan yang dibuat dan didapatkan hasil bahwa dari analisis persyaratan yang dibuat, mulai dari aktivitas proses bisnis usulan, dapat dirunutkan sampai perancangan sistem yang telah dibuat.

Tabel 6.4 Matriks *Traceability*

Kode Aktivitas Proses Bisnis	Persyaratan Fungsional	Nama Use case	Kode Use case	Diagram Aktivitas	Sequence Diagram	Antarmuka Pengguna
P-PB-XX	P-KF-002	Kelola data pengguna	P-UC-001	Gambar 4.4	Gambar 5.3	Gambar 5.22
P-PB-XX	P-KF-003	Registrasi pengguna	P-UC-002	Gambar 4.5	Gambar 5.4	Gambar 5.23 Gambar 5.24
P-PB-002	P-KF-004	Ubah status pembayaran	P-UC-003	Gambar 4.6	Gambar 5.5	Gambar 5.25 Gambar 5.26
P-PB-003	P-KF-005	Validasi laporan pembayaran	P-UC-004	Gambar 4.7	Gambar 5.6	Gambar 5.37
P-PB-004	P-KF-006	Lihat data pembayaran pajak	P-UC-005	Gambar 4.8	Gambar 5.7	Gambar 5.28 Gambar 5.33
P-PB-004	P-KF-009	Lihat laporan rencana target dan realisasi pajak	P-UC-006	Gambar 4.9	Gambar 5.8	Gambar 5.30 Gambar 5.31
P-PB-005	P-KF-008	Pantau pembayaran pajak	P-UC-007	Gambar 4.10	Gambar 5.9	Gambar 5.32
P-PB-001	P-KF-011	Laporkan pembayaran pajak	P-UC-008	Gambar 4.11	Gambar 5.10	Gambar 5.35 Gambar 5.36
P-PB-006	P-KF-007	Cetak data pembayaran pajak	P-UC-009	Gambar 4.12	Gambar 5.11	Gambar 5.28 Gambar 5.33
P-PB-004	P-KF-010	Cetak laporan rencana dan target pajak	P-UC-010	Gambar 4.13	Gambar 5.12	Gambar 5.30
P-PB-XX	P-KF-001	<i>Login</i>	P-UC-011	Gambar 4.14	Gambar 5.13	Gambar 5.20

Setelah matriks kerunutan dibuat maka dilakukan pengecekan persyaratan berdasarkan kerunutan yang telah dilakukan. Evaluasi dilakukan oleh peneliti untuk mengetahui apakah persyaratan yang didefinisikan telah sesuai dengan yang diharapkan dilihat dari parameter kerunutan atau *traceability*. Daftar pertanyaan yang dibuat dapat dilihat pada Tabel 6.5.

**Tabel 6.5 Software Requirement Checklist (Traceability)**

<b>Software Requirement Checklist (Traceability)</b>				
<b>No.</b>	<b>Pertanyaan</b>	<b>Ya</b>	<b>Tidak</b>	<b>Keterangan</b>
1	Apakah setiap persyaratan didefinisikan dengan benar dan unik?	✓		Setiap persyaratan telah diberikan kode atau penomoran yang unik.
2	Apakah setiap persyaratan fungsional dari perangkat lunak dapat dilacak sampai persyaratan pada tingkat yang lebih tinggi?	✓		Sudah ada pemetaan kerunutan yang terhubung pada aktivitas proses bisnis, fitur, persyaratan fungsional, dan <i>use case</i> .
3	Dapatkah semua bagian desain tingkat tinggi dapat dilacak kembali ( <i>high-level design</i> ) sesuai persyaratan?	✓		Sudah terdapat kode unik pada desain tingkat tinggi seperti pada <i>use case</i> .

## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan penelitian yang telah dijalankan, didapatkan beberapa kesimpulan, yaitu :

1. Proses bisnis penerimaan PAD yang saat ini berjalan menunjukkan adanya aktivitas-aktivitas yang masih dilakukan secara manual, sehingga menyebabkan adanya aktivitas yang memerlukan waktu yang cukup lama untuk dapat dilaksanakan. Dari analisis proses bisnis tersebut dapat ditemukan masalah dan solusi yang ditawarkan, sehingga dibuat proses bisnis usulan penerimaan PAD. Pada analisis proses bisnis usulan terdapat enam aktivitas baru yang diterapkan sesuai dengan solusi dari masalah yang ditemukan. Pada proses bisnis usulan tersebut terdapat perubahan berupa penambahan aktivitas baru dan perubahan *task* dari aktivitas. Dari aktivitas baru tersebut kemudian dapat digunakan sebagai acuan untuk menentukan persyaratan sistem yang akan diusulkan.
2. Persyaratan sistem didapatkan dengan melakukan tahapan : analisis pemangku kepentingan, analisis pengguna, analisis kebutuhan pengguna dan pemangku kepentingan, serta analisis fitur produk. Dari analisis hal-hal tersebut kemudian dapat ditentukan persyaratan sistem yang terdiri dari persyaratan fungsional dan non fungsional sistem. Analisis persyaratan sistem yang dilakukan menghasilkan 11 persyaratan fungsional dan 2 persyaratan non fungsional. Selain itu dihasilkan juga model *use case* yang digambarkan secara visual maupun tekstual. Persyaratan sistem yang didapatkan berdasarkan aktivitas baru dari analisis proses bisnis usulan, yang dijadikan acuan untuk menentukan apa saja persyaratan yang harus ada pada sistem.
3. Pada tahapan perancangan dilakukan beberapa tahapan agar hasil rancangan sesuai dengan analisis persyaratan yang telah dilakukan, tahapan tersebut beserta hasilnya adalah :
  - a. Analisis *use case* : merupakan tahapan untuk menganalisis *use case* yang telah dibuat untuk dapat menghasilkan kelas yang digunakan dalam perancangan. Adapun hasilnya adalah kelas analisis, mekanisme analisis, dan *unify class analysis*.
  - b. Pemodelan interaksi yang menghasilkan diagram interaksi (*sequence diagram*) yang disesuaikan dengan kelas analisis dan skenario *use case*.
  - c. Elemen desain yang menghasilkan *package* dan diagram kelas.
  - d. Pemodelan data yang menghasilkan *Entity Relationship Diagram*, *Physical Data Model*, dan perancangan tabel.
  - e. Rancangan antarmuka pengguna dan prototipe. Setelah tahapan perancangan selesai dilakukan, maka dapat dilakukan tahapan evaluasi. Hasil analisis persyaratan sistem yang berupa diagram *use case*, spesifikasi *use case*, dan diagram aktivitas digunakan dalam tahapan perancangan

untuk dijadikan acuan membuat rancangan sistem agar sesuai dengan kebutuhan dan persyaratan sistem.

4. Dari evaluasi yang dilakukan, didapatkan hasil sebagai berikut:

a. Peninjauan prototipe terhadap *use case* dan perancangan dasar antarmuka pengguna:

Dari tiga sampel pengguna yang melakukan kasus uji *use case*, diketahui bahwa setiap pengguna telah melakukan kasus uji dari masing-masing *use case* dengan langkah-langkah yang sesuai dengan alur dasar *use case*. Dan dari peninjauan yang dilakukan oleh pengguna dengan menjawab pertanyaan, setiap pengguna merasa puas terhadap alur *use case* yang diujikan dan menyatakan bahwa alur dari *use case* yang diujikan sudah sesuai dengan yang diharapkan. Jenis data yang dibutuhkan dan penanganan kesalahan pada setiap *use case* yang diujikan pun telah sesuai. Kemudian untuk umpan balik pengguna terhadap perancangan dasar antarmuka pengguna dengan parameter navigasi dan antarmuka pengguna, dua pengguna memilih setuju dengan butir-butir pertanyaan yang diajukan dan satu pengguna menyatakan tidak setuju dengan salah satu pertanyaan. Ketiga pengguna juga memberikan saran berupa menambahkan detail filter pada menu laporan perencanaan, menambahkan logo transparan DISPENDA Kabupaten Madiun pada latar belakang antarmuka sistem, menambahkan tampilan menu selamat datang setelah melakukan *login*.

b. Peninjauan kerunutan (*traceability*) dari matriks kerunutan yang dibuat:

Semua aktivitas baru pada proses bisnis usulan dapat dirunut sampai ke spesifikasi persyaratan dan rancangan sistem. Selain itu, setiap persyaratan telah tergambar ke dalam *use case*, diagram aktivitas, *sequence diagram*, dan rancangan antarmuka pengguna. Dan berdasarkan daftar pengecekan persyaratan yang menggunakan parameter kerunutan, dihasilkan bahwa setiap persyaratan bisa dirunut sampai persyaratan pada tingkat tinggi dan perancangan tingkat tinggi dapat dilacak sesuai persyaratan.

## 7.2 Saran

1. Hasil analisis persyaratan dan perancangan sistem dapat dijadikan dasar untuk melakukan tahapan selanjutnya yaitu tahap perancangan detail dan implementasi.
2. Penelitian selanjutnya dapat mengembangkan perancangan dengan menambahkan detail pada analisis persyaratan non fungsional dan mekanisme analisis.
3. Penelitian selanjutnya dapat mengembangkan evaluasi menjadi tahap pengujian dengan metode pengujian persyaratan yang sesuai.
4. Penelitian selanjutnya dapat memperbaiki antarmuka pengguna dari sistem sesuai dengan saran yang didapatkan dari hasil evaluasi.

## DAFTAR PUSTAKA

- A.S., R. & Shalahuddin, M., 2013. *Rekayasa Perangkat Lunak : Terstruktur dan Berorientasi Objek*. Bandung: Penerbit INFORMATIKA.
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. Canada: Addison Wesley.
- Booch, G., Rumbaugh, J. & Jacobson, I., 2005. *The Unified Modeling Language User Guide Second Edition*. United States: Addison Wesley Professional.
- Chiew, T. K. & Salim, S. S., 2003. WEBUSE: Website Usability Evaluation Tool. *Malaysian Journal of Computer Science*, 16(1), pp. 47-57.
- Davenport, T. H., 1993. *Process Inovation : Reengineering Work Through Information Technology*. Boston, MA: Havard Business Press.
- Dix, A., Finlay, J., Abowd, G. D. & Beale, R., 2004. *Human Computer Interaction*. 3rd penyunt. London: Pearson Education Limited.
- DSDM Consortium, 2008. *DSDM Atern Handbook*. [Online]  
Available at: <https://www.dsdm.org/>  
[Diakses 18 July 2016].
- Erlina, 2007. *Analisis dan Perancangan Sistem Informasi Penerapan Preventive Maintenance Untuk Menentukan Jadwal Perawatan Pencegahan Yang Optimum Dan Meningkatkan Keandalan Komponen Kritis Mesin Hd/Pe-120 Pada Pt. Metropoly Jaya Nusa*, Jakarta: Binus University.
- Gotel, O., 2012. *Software and System Traceability*. New York: Springer.
- Hermawan, J., 2015. *Analisa & Desain Pemrograman Berorientasi Obyek dengan UML dan VISUAL BASIC. NET*. [Online]  
Available at:  
[https://books.google.co.id/books?id=INoYnOClrBkC&pg=PA118&dq=analisa+dan+desain+pemrograman&hl=id&sa=X&ved=0CBkQ6AEwAGoVChMI4\\_SYmqCoyAIVDsK0Ch2-  
jwBf#v=onepage&q=analisa%20dan%20desain%20pemrograman&f=false](https://books.google.co.id/books?id=INoYnOClrBkC&pg=PA118&dq=analisa+dan+desain+pemrograman&hl=id&sa=X&ved=0CBkQ6AEwAGoVChMI4_SYmqCoyAIVDsK0Ch2-<br/>jwBf#v=onepage&q=analisa%20dan%20desain%20pemrograman&f=false)  
[Diakses 4 Oktober 2015].
- IBM Software Group, 2003. *Essentials of Rational Unified Process*. United States: IBM Corporation.
- IBM Software Group, 2004. *Mastering Object-Oriented Analysis and Design with UML 2.0*. United States: IBM Corporation.
- Institute of Electrical and Electronics Engineers (IEEE), 2006. *IEEE 610 Standard Glossary of Software Engineering Terminology*. United States of America: The Institute of Electrical and Electronics Engineer.
- Institute of Electrical and Electronics Engineers (IEEE), 2008. *IEEE 1512 Implementation Guide*. United States of America: The Institute of Electrical and Electronics Engineers, Inc..

- Jogiyanto, H. M., 1995. *Analisis & Disain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset.
- Jogiyanto, H. M., 2005. *Sistem Teknologi Informasi*. Yogyakarta: ANDI.
- Limaye, M. G., 2009. *Software Testing Principles, Techniques and Tools*. New Delhi, Tata McGraw Hill Education Private Limited.
- Mantani, R., 2014. *Analisis dan Perancangan Sistem Antrian Terintegrasi di Bank Indonesia*, Surabaya: Institut Sepuluh November.
- Markus, D. M., 2005. *Perpajakan Indonesia*. Jakarta: Gramedia Pustaka Utama.
- McLeod, R. J., 2001. *Management Information Systems*. New Jersey: Prentice-Hall.
- Muratno, B., 2015. *Proses Bisnis Pembayaran Pajak [Wawancara]* (29 September 2015).
- Object Management Group (OMG), 2011. *OMG Object Mangement Group*. [Online]  
Available at: <http://www.omg.org/spec/BPMN/2.0/PDF/>  
[Diakses 15 December 2015].
- O'Brien, J. A. & Marakas, G. M., 2010. *Introduction to Information System*. New York: The McGraw-Hill Companies.
- Owen, M. & Raj, J., 2003. *BPMN and Business Process Management-Introduction to the New Business Process Modelling Standard*. New Jersey: Popkin Software & System Inc.
- Pant, K. & Juric, M. B., 2008. *Business Process Driven SOA Using BPMN and BPEL: From Business Process Modeling to Orchestration and Service Oriented Architecture*. Brimingham: Packt Publishing Ltd..
- Pressman, R. S., 2010. *Software Engineering : A Practitioner's Approach, Seventh Edition*. New York: McGraw-Hill.
- Siahaan, D., 2012. *Analisa Kebutuhan dalam Rekayasa Perangkat Lunak*. Yogyakarta: Penerbit Andi.
- Sommerville, I., 2009. *Software Engineering*. 9th penyunt. United States: Addison Wesley.
- Sugiyanto, S. M., 2007. *Pajak & Retribusi Daerah*. [Online]  
Available at: [https://books.google.co.id/books?id=7\\_ZIOGi2jagC&dq](https://books.google.co.id/books?id=7_ZIOGi2jagC&dq)  
[Diakses 27 Oktober 2015].
- Tantra, R., 2012. *Manajemen Proyek Sistem Informasi : Bagaimana Mengelola Proyek Sistem Informasi Secara Efektif dan Efisien*. Yogyakarta: Penerbit Andi.
- Weske, M., 2007. *Business Process Management*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg New York.
- Wieggers, K. E., 2001. *Process Assets for Software Peer Reviews and Inspections*. [Online]

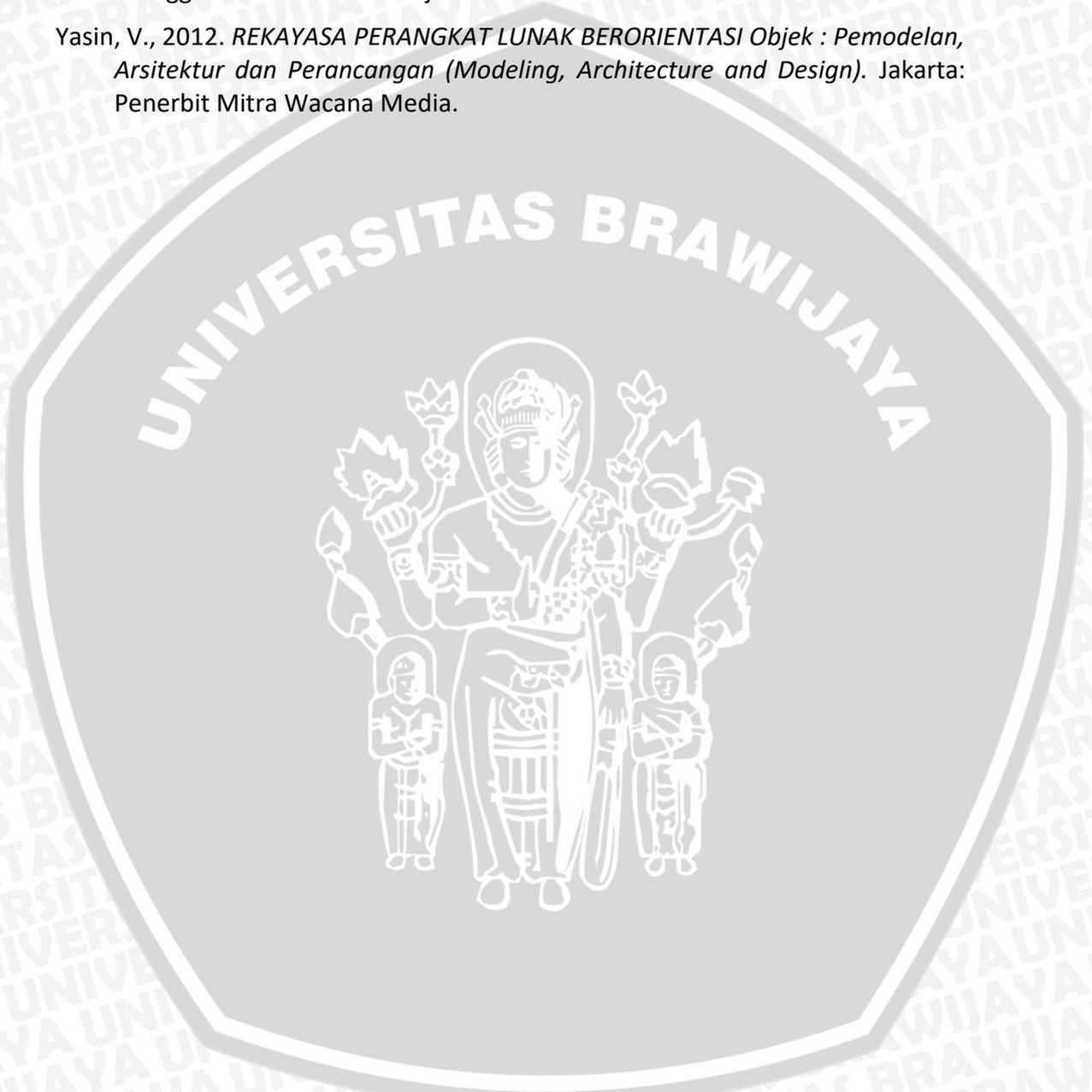
Available at:

<http://www.cs.toronto.edu/~sme/CSC340F/2005/assignments/inspections/index.html>

[Diakses 28 Maret 2016].

Wulandari, M. A., 2013. Analisis dan Perancangan Proses Bisnis PT. Medix Soft Menggunakan Sistem Manajemen Mutu ISO 9001:2008.

Yasin, V., 2012. *REKAYASA PERANGKAT LUNAK BERORIENTASI Objek : Pemodelan, Arsitektur dan Perancangan (Modeling, Architecture and Design)*. Jakarta: Penerbit Mitra Wacana Media.



## LAMPIRAN A HASIL EVALUASI

### A.1 Pengguna 1 (Budi Muratno, S. Sos)

#### A.1.1 Hasil peninjauan halaman 1

REVIEW PROTOTYPE	
A. HARAPAN ALUR PENGGUNA	
Use case : Registrasi Pengguna Skenario : Berhasil Registrasi Pengguna	
Pertanyaan	Jawaban
Apakah alur registrasi sesuai dengan yang diharapkan?	Sangat Sesuai yg kita butuhkan.
Apakah jenis-jenis data pada formulir registrasi pengguna sudah benar dan lengkap?	Berman & Lengkap.
Apakah proses registrasi mudah dioperasikan?	Sangat mudah dioperasikan.
Saran : Bisa ditambah & dipromosikan oleh pengguna.	
Use case : Registrasi Pengguna Skenario : Data registrasi tidak diisi lengkap	
Pertanyaan	Jawaban
Apakah penanganan kesalahan pengisian data sudah benar?	Sudah sesuai konsep.
Saran : Registrasi hrs diisi lengkap, & apabila tdk lengkap sistem tdk bisa melanjutkan Registrasi Pengguna.	
Use case : Kelola Data Pengguna Skenario : Edit Data Pengguna	
Pertanyaan	Jawaban
Apakah alur edit data pengguna sesuai dengan yang diharapkan?	Sudah jauh lebih baik Edit.
Apakah jenis-jenis data pada formulir edit data pengguna sudah benar dan lengkap?	Sudah, apabila sdh sdh kelengkapan Edit tdk dpt. maka Edit tdk berhasil.
Apakah proses edit data pengguna mudah dioperasikan?	Sangat Mudah.
Saran : Mau Edit Data, Sangat Mudah dioperasikan.	

Gambar A. 1 Hasil Peninjauan Pengguna 1 – Lembar 1

### A.1.2 Hasil peninjauan halaman 2

Use case : Monitoring Pembayaran Pajak	
Skenario : Melakukan monitoring pembayaran pajak	
Pertanyaan	Jawaban
Apakah alur <i>monitoring</i> pembayaran pajak sudah sesuai dengan yang diharapkan?	Alur <i>Monitoring</i> sudah sesuai
Apakah jenis-jenis data pada kolom masukan sudah benar dan lengkap?	Data 2 kolom (sukses & gagal) tertera
Apakah jenis-jenis data yang ditampilkan (keluaran) pada tabel <i>monitoring</i> sudah benar dan lengkap?	Sangat jelas dan sesuai tabel <i>monitoring</i>
Saran :	
Saran/masukan untuk registrasi pengguna, edit pengguna dan <i>monitoring</i> tidak perlu label judul lagi (new status). atau ditambahkan pd. label warna tertentu.	

### B. NAVIGASI DAN ANTARMUKA

Navigasi				
Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui peran saya pada sistem.	✓	—	—	Baik
Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi.	✓	—	—	Baik
Sistem mudah digunakan dengan menggunakan tombol kembali atau <i>link</i> .	✓	—	—	Baik
Sistem tidak membuka banyak jendela browser ketika digunakan.	✓	—	—	Baik.
Peletakan menu atau <i>link</i> dapat dengan mudah dikenali	✓	—	—	Baik
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas	✓	—	—	Baik
Masukan dan Saran	—			

Gambar A. 2 Hasil Peninjauan Pengguna 1 – Lembar 2

### A.1.3 Hasil peninjauan halaman 3

Antarmuka Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antarmuka sistem mudah untuk dipelajari	✓	—	—	Baik
Sistem memiliki antarmuka yang konsisten	✓	—	—	Bah.
<b>Masukan dan Saran</b>		—		

Diisi oleh (reviewer) : *BUDI MURATIWO, S. Sos.*  
Pekerjaan/Jabatan : *DIREKTOR KEB. MADURAY / KOTAK KAWASAN*

Madiun, 9 Mei 2016

Reviewer

*(BUDI MURATIWO, S. Sos.)*

Gambar A. 3 Hasil Peninjauan Pengguna 1 – Lembar 3

## A.2 Pengguna 2 (Ari Nursurahmat, S. Sos)

### A.2.1 Hasil peninjauan halaman 1

REVIEW PROTOTYPE	
<b>A. HARAPAN ALUR PENGGUNA</b>	
Use case : Registrasi Pengguna Skenario : Berhasil Registrasi Pengguna	
Pertanyaan	Jawaban
Apakah alur registrasi sesuai dengan yang diharapkan?	Sudah sesuai
Apakah jenis-jenis data pada formulir registrasi pengguna sudah benar dan lengkap?	Sudah
Apakah proses registrasi mudah dioperasikan?	mudah
Saran :	—
Use case : Registrasi Pengguna Skenario : Data registrasi tidak diisi lengkap	
Pertanyaan	Jawaban
Apakah penanganan kesalahan pengisian data sudah benar?	Sudah
Saran :	—
Use case : Kelola Data Pengguna Skenario : Edit Data Pengguna	
Pertanyaan	Jawaban
Apakah alur edit data pengguna sesuai dengan yang diharapkan?	Sesuai
Apakah jenis-jenis data pada formulir edit data pengguna sudah benar dan lengkap?	Sudah
Apakah proses edit data pengguna mudah dioperasikan?	mudah
Saran :	—

Gambar A. 4 Hasil Peninjauan Pengguna 2 – Lembar 1

## A.2.2 Hasil peninjauan halaman 2

Use case : Monitoring Pembayaran Pajak	
Skenario : Melakukan monitoring pembayaran pajak	
Pertanyaan	Jawaban
Apakah alur <i>monitoring</i> pembayaran pajak sudah sesuai dengan yang diharapkan?	Sudah
Apakah jenis-jenis data pada kolom masukan sudah benar dan lengkap?	Sudah
Apakah jenis-jenis data yang ditampilkan (keluaran) pada tabel <i>monitoring</i> sudah benar dan lengkap?	Sudah
Saran : -	

### B. NAVIGASI DAN ANTARMUKA

Navigasi Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui peran saya pada sistem.	✓			
Sistem menyediakan navigasi menu dan <i>link</i> yang berguna untuk mendapatkan informasi.	✓			
Sistem mudah digunakan dengan menggunakan tombol kembali atau <i>link</i> .	✓			
Sistem tidak membuka banyak jendela browser ketika digunakan.	✓			
Peletakan menu atau <i>link</i> dapat dengan mudah dikenali	✓			
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas	✓			
<b>Masukan dan Saran</b>	Laporan perencanaan pada grafik/tabel lebih detail (tgl/bulan)			

Gambar A. 5 Hasil Peninjauan Pengguna 2 – Lembar 2

### A.2.3 Hasil peninjauan halaman 3

Antarmuka Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antarmuka sistem mudah untuk dipelajari	✓			
Sistem memiliki antarmuka yang konsisten	✓			
<b>Masukan dan Saran</b>	Kasih logo transparan pencala modulan di background.			

Diisi oleh (reviewer) : ARI KURSURAHMAT  
 Pekerjaan/Jabatan : KABID PENGGIHMAN & KEBERATAN

Madiun, Mei 2016

Reviewer

  
 (ARI KURSURAHMAT)

Gambar A. 6 Hasil Peninjauan Pengguna 2 – Lembar 3

### A.3 Pengguna 3 (Abdul Aziz Hadyansya)

#### A.3.1 Hasil peninjauan halaman 1

#### REVIEW PROTOTYPE

**A. HARAPAN ALUR PENGGUNA**

Use case : Laporkan Pembayaran Pajak Skenario : Berhasil melakukan pelaporan pajak	
Pertanyaan	Jawaban
Apakah alur pelaporan pajak sesuai dengan yang diharapkan?	Ya.
Apakah jenis-jenis data pada formulir laporan pajak pengguna sudah benar dan lengkap?	Ya.
Apakah proses laporan pajak mudah dioperasikan?	Mudah.
Saran :	
Masih ada bug . semangat :.)	

**B. NAVIGASI DAN ANTARMUKA**

Navigasi Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui peran saya pada sistem.	✓			
Sistem menyediakan navigasi menu dan link yang berguna untuk mendapatkan informasi.	✓			
Sistem mudah digunakan dengan menggunakan tombol kembali atau link.	✓			
Sistem tidak membuka banyak jendela browser ketika digunakan.	✓			
Peletakan menu atau link dapat dengan mudah dikenali			✓	Tetes berupa link yg tidak diketahui fungsi
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas	✓			
<b>Masukan dan Saran</b>				

Gambar A. 7 Hasil Peninjauan Pengguna 3 – Lembar 1

### A.3.2 Hasil peninjauan halaman 2

Antarmuka Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antarmuka sistem mudah untuk dipelajari	✓			
Sistem memiliki antarmuka yang konsisten	✓			
<b>Masukan dan Saran</b>				

Diisi oleh (reviewer) :

Pekerjaan/Jabatan :

Madiun, Mei 2016

Reviewer



Gambar A. 8 Hasil Peninjauan Pengguna 3 – Lembar 2