

**Analisis dan Perancangan Sistem Informasi Penerimaan
Pendapatan Asli Daerah (PAD) Non-PBB (Pajak Bumi Bangunan)
Kabupaten Madiun**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Putri Mutiara Tungga

NIM: 125150400111029



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

PENGESAHAN

Analisis dan Perancangan Sistem Informasi Penerimaan Pendapatan Asli Daerah
(PAD) Non-PBB (Pajak Bumi Bangunan) Kabupaten Madiun

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Putri Mutiara Tungga
NIM: 125150400111029

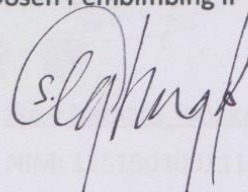
Skripsi ini telah diuji dan dinyatakan lulus pada
11 Agustus 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Ismiarta Aknuranda, S.T, M.Sc, Ph.D
NIK: 201006 740719 1 001

Dosen Pembimbing II



Satrio Agung W., 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 disitasi 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, 22 Juni 2016

Putri Mutiara Tungga

NIM: 125150400111029



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan penyusunan skripsi yang berjudul **“Analisis dan Perancangan Sistem Informasi Penerimaan Pendapatan Asli Daerah (PAD) Non-PBB (Pajak Bumi Bangunan) Kabupaten Madiun”**. Tak lupa penulis haturkan shalawat serta salam kepada Rasulullah Nabi Muhammad SAW, keluarga, sahabat-sahabat serta para pengikutnya, semoga rahmat Allah SWT selalu tercurah untuk kita semua. Aaammiiinn.

Penyusunan skripsi ini juga tak lepas dari bantuan, dukungan, doa, bimbingan, kritik, saran maupun motivasi dari berbagai pihak. Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada :

1. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D selaku dosen pembimbing I, atas segala bimbingan, waktu yang telah diluangkan, kritik, saran serta ilmu yang telah diberikan sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Satrio Agung W., S.Kom, M.Kom selaku dosen pembimbing II, atas segala bimbingan, waktu yang telah diluangkan, kritik, saran serta ilmu yang telah diberikan sehingga penulis dapat menyelesaikan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Suprpto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya.
5. Kedua orang tua, Bapak Sukamto dan Ibu Tri Wahyuningsih yang telah memberikan doa, motivasi, kasih sayang serta dukungan baik moril maupun materiil selama perkuliahan.
6. Bapak Yusi Tyroni Mursityo, S.Kom., M.S. Selaku ketua Laboratorium Sistem Informasi.
7. Aditya Rachmadi, S.ST., M.TI selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
8. Bapak dan Ibu Dosen Fakultas Ilmu Komputer/Program Teknologi Informasi dan Ilmu Komputer yang telah memberikan ilmu baik di dalam dan di luar perkuliahan.
9. Seluruh pegawai dan staf Fakultas Ilmu Komputer Universitas Brawijaya.
10. Pihak Dinas Pendapatan Kabupaten Madiun yang telah memperbolehkan dan membantu dalam penelitian sehingga penelitian dapat tercapai dan sukses.

11. Elok Mutiara Nirwana selaku adik dari penulis serta seluruh keluarga yang telah memberikan semangat doa dan dukungan penuh dalam penyusunan skripsi ini.
12. Sahabat-sahabat MENANTU IDAMAN : Krisdhamara Widyas Pritha, S.Kom; Diannisa Hakim, S.Kom; Ika Qutsiati Utami, S.Kom; Atikah Hayyu Mulia, S.Kom; Frida Ayundha Putry, S.Kom; Lia Hasanah, S.Kom; Anom Sulton Iskandar, S.Kom; Mochamad Arizal, S.Kom; Onny Iman Kandung Iriantono, S.Kom; Randi Dwi Nandra, S.Kom; Rayyan Ramadhan, S.Kom; serta Maulana Ghulam Hanifa, S.Kom yang telah banyak menemani, membantu memberikan dukungan, doa, motivasi, kritik dan saran selama perkuliahan.
13. Teman-teman seperjuangan: Dwi Retno Palupi, S.Kom; Isti Dyah Nur Safitri, S.Kom; Bella Pertiwi, S.Kom; Nurrizka Ajeng Utami, S.Kom; Voni Rizkia Ninda, S.Kom; Krisdhamara Widyas Pritha, S.Kom; Lia Hasanah, S.Kom; Sabrina Lunettavashti, S.Kom; Brilliant Chindy Maharani, S.Kom; serta Nabilah Ulfah Aulia, S.Kom.
14. Sahabatku Nurul Nur Agustina, S.Pd; Hamiddah Intan Kusumastuti, SP, MSi.; dan Septiana Dwi Rahmawati, S.Kep. Ns. yang selalu memberikan doa, dukungan, motivasi dan semangat.
15. Teman-teman mahasiswa Program Studi Sistem Informasi khususnya angkatan 2012.
16. Teman-teman dan kakak-kakak keluarga Raion Community dan Raion Studio yang telah memberikan doa, bantuan, ilmu, dukungan, kritik dan saran.
17. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Semoga segala kebaikan dan amal baik mendapatkan balasan yang setimpal dari Allah SWT. Penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna serta banyak kekurangan disebabkan keterbatasan, kemampuan dan pengalaman, dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca. Harapan penulis semoga skripsi ini dapat memberikan manfaat dari semua pihak.

Malang, 22 Juni 2016

Penulis

Putri.mutiara.t@gmail.com

ABSTRAK

Dinas Pendapatan Kabupaten Madiun merupakan sebuah badan atau organisasi yang berada di bawah pemerintahan Kabupaten Madiun yang menangani pendapatan daerah. Salah satu sumber pendapatan daerah menurut Undang-Undang Republik Indonesia Nomor 33 Tahun 2004 adalah Pendapatan Asli Daerah (PAD). Dinas Pendapatan Kabupaten Madiun telah bekerja sama dengan Bank Jatim dalam penerimaan/pembayaran Pendapatan Asli Daerahnya. Dinas Pendapatan Kabupaten Madiun saat ini belum memiliki sistem informasi yang terintegrasi secara internal antar bidang dan eksternal dengan pihak Bank Jatim dalam proses penerimaan Pendapatan Asli Daerah selain PBB. Oleh karena itu, proses rekapitulasi data-data penerimaan PAD selain PBB masih dilakukan secara manual. Dari masalah tersebut, diperlukan sistem informasi terintegrasi secara internal maupun eksternal untuk proses penerimaan PAD selain PBB.

Analisis dan perancangan merupakan tahapan penting dalam pengembangan sistem. Dari berbagai studi ditemukan 70% dari proyek gagal adalah disebabkan oleh persyaratan/*requirements* yang tidak lengkap dan akurat. Dengan dilakukannya tahap perancangan sistem pengembang dapat menilai terlebih dahulu beberapa aspek kualitas perangkat lunak sebelum dilakukan tahap implementasi.

Berdasarkan hal-hal tersebut, dilakukan penelitian untuk menganalisis dan merancang sistem informasi penerimaan pendapatan asli daerah non-PBB di Kabupaten Madiun. Penelitian ini dilakukan dengan pendekatan berorientasi objek. Hasil dari penelitian yang dilakukan berupa model proses bisnis saat ini dan usulan, model analisis dan spesifikasi persyaratan sistem, model-model perancangan sistem serta prototipe sistem. Prototipe digunakan untuk mendemonstrasikan persyaratan fungsional sistem dan dievaluasi terhadap skenario *use case*, navigasi dan antarmuka untuk mendapatkan masukan dari pengguna serta mengetahui apakah spesifikasi persyaratan sudah sesuai dengan harapan pengguna. Evaluasi dalam penelitian ini adalah verifikasi dan validasi. Berdasarkan hasil tinjauan skenario *use case* dan evaluasi *traceability*, hasil dari verifikasi pada penelitian ini adalah alur yang dilakukan pengguna sudah sesuai dengan alur yang dispesifikasikan serta hasil dari pemodelan proses bisnis dapat ditelusuri sampai ke perancangan sistem. Berdasarkan tinjauan prototipe, hasil dari validasi pada penelitian ini adalah alur yang dispesifikasikan pada persyaratan perangkat lunak dan diimplementasikan dengan prototipe sudah sesuai dengan harapan pengguna dan pengguna merasa mudah untuk mengoperasikan prototipe sistem. Selain itu terdapat tiga masukan mengenai data dan satu masukan mengenai antarmuka.

Kata kunci: analisis dan perancangan berorientasi objek, sistem informasi penerimaan pendapatan daerah, pajak

ABSTRACT

Dinas Pendapatan Kabupaten Madiun is an agency or organization under the reign of Kabupaten Madiun that handles local revenue. One source of local revenue according to the Law of the Republic of Indonesia Number 33 of 2004 is Pendapatan Asli Daerah (PAD). Dinas Pendapatan Kabupaten Madiun has cooperated with Bank Jatim in receipt / payment a local revenue. Dinas Pendapatan Kabupaten Madiun does not currently have an integrated information system internally and externally with Bank Jatim in the process of receiving the PAD. Therefore, process recapitulation data reception of PAD is still done manually. So information system that integrate internally and externally is needed for process receiving PAD non-PBB.

Analysis and design is an important step in the development system. Previously study shown that various studies found 70% of projects fail is due to the requirements were incomplete and inaccurate. With system design phase the developers can assess first the quality of the software before implementation phase.

Based on the these things, so we conducted a research to analyze and design information systems of PAD non-PBB in Kabupaten Madiun. This research was conducted with the object oriented approach. The results of the research are current business process model and the proposed, model system requirements analysis and specification, system design models and system prototypes. The prototype was used to demonstrate the functional requirements of the system and evaluated the use case scenarios, navigation and interface to get feedback from users as well as to know whether the requirements specification are in accordance with user expectations. Evaluation in this study is the verification and validation. Based on review use case scenarios and traceability evaluation, verification of the results of this study are flow that performed by user in accordance with flow that have been specified and the results of business process modeling can be traced to the system design. Based on a review of prototypes, the results of validation in this study is the flow that specified in the software requirements and implemented in a prototype are in accordance with the expectations of users and users find it easy to operate the system prototype. In addition there are three feedback about the data and the one feedback on the interface.

Keywords : *object-oriented analysis and design, information systems local revenue receipts, tax*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori.....	6
2.2.1 Dinas Pendapatan Kabupaten Madiun	6
2.2.2 Proses Bisnis.....	10
2.2.3 Rational Unified Process (RUP)	18
2.2.4 Sistem Informasi.....	19
2.2.5 Evaluasi.....	44
BAB 3 METODOLOGI	47
3.1 Studi Literatur	47
3.2 Pengumpulan Data	48
3.3 Pemodelan Proses Bisnis	48
3.4 Analisis Persyaratan Sistem	48
3.5 Perancangan Sistem.....	49

3.6 Evaluasi	49
3.7 Kesimpulan dan Saran	49
BAB 4 pemodelan proses bisnis dan ANALISIS PERSYARATAN	50
4.1 Proses bisnis saat ini	50
4.2 Proses bisnis usulan	53
4.3 Analisis permasalahan	55
4.4 Analisis Pemangku Kepentingan dan Pengguna	56
4.4.1 Tipe Pemangku Kepentingan	56
4.4.2 Tipe Pengguna	56
4.5 Kebutuhan Pengguna dan Pemangku Kepentingan	57
4.6 Fitur	59
4.7 Persyaratan Perangkat Lunak	61
4.7.1 Persyaratan fungsional	61
4.7.2 Persyaratan non fungsional	63
4.8 Pernyataan kedudukan produk	64
4.9 Aktor	64
4.10 Diagram <i>Use Case</i>	65
4.11 Spesifikasi <i>Use Case</i>	67
4.12 <i>Activity Diagram</i>	92
BAB 5 PERANCANGAN	108
5.1 Analisis Arsitektur	108
5.2 Kelas-Kelas Analisis	109
5.3 Mekanisme Analisis	115
5.4 Unifikasi kelas-kelas analisis	117
5.5 Package diagram	118
5.6 Diagram <i>sequence</i>	119
5.7 Diagram Kelas	140
5.8 Pemodelan Data	143
5.9 Perancangan antarmuka pengguna	149
5.10 Prototipe	166
BAB 6 EVALUASI	173
6.1 Tinjauan	173

6.1.1 Tinjauan skenario <i>use case</i>	173
6.1.2 Tinjauan Prototipe	177
6.2 Evaluasi Traceability.....	178
6.2.1 Matriks Traceability.....	178
6.2.2 Tinjauan Traceability	181
BAB 7 PENUTUP	182
7.1 Kesimpulan.....	182
7.2 Saran	183
DAFTAR PUSTAKA.....	184
LAMPIRAN A SURAT IZIN PENELITIAN	187
LAMPIRAN B HASIL EVALUASI	190



DAFTAR TABEL

Tabel 2.1	Notasi BPMN	12
Tabel 2.2	Stereotype - ikon	23
Tabel 2.3	Simbol-Simbol Use Case Diagram	26
Tabel 2.4	Template pernyataan masaah	29
Tabel 2.5	Template pernyataan posisi produk	30
Tabel 2.6	Simbol-Simbol Activity Diagram	30
Tabel 2.7	Simbol-Simbol Sequence Diagram	33
Tabel 2.8	Simbol-Simbol Class Diagram	36
Tabel 2.9	Kuesioner WEBUSE	45
Tabel 4.1	Daftar tambahan dan perubahan aktivitas proses bisnis lanjutan	54
Tabel 4.2	Analisis Permasalahan	55
Tabel 4.3	Analisis Tipe Pemangku kepentingan	56
Tabel 4.4	Tipe pengguna	57
Tabel 4.5	Kebutuhan pengguna dan pemangku kepentingan lanjutan	58
Tabel 4.6	Fitur	60
Tabel 4.7	Persyaratan fungsional sistem	61
Tabel 4.8	Persyaratan non fungsional sistem	63
Tabel 4.9	Pernyataan kedudukan produk	64
Tabel 4.10	Aktor	64
Tabel 4.11	Pemetaan Diagram Use Case	66
Tabel 4.12	Spesifikasi Use Case Login	67
Tabel 4.13	Spesifikasi Use Case Kelola Data Pengguna	68
Tabel 4.14	Spesifikasi Use Case Kelola Pendataan Wajib Pajak	71
Tabel 4.15	Spesifikasi Use Case Kelola Pendataan Objek Pajak	75
Tabel 4.16	Spesifikasi Use Case Kelola Penetapan Pajak	78
Tabel 4.17	Spesifikasi Use Case Cetak SKPD	82
Tabel 4.18	Spesifikasi Use Case Lihat Laporan Penerimaan Pajak	83
Tabel 4.19	Spesifikasi Use Case Cetak Laporan Penerimaan Pajak	84

Tabel 4.20 Spesifikasi Use Case Lihat Buku Register	85
Tabel 4.21 Spesifikasi Use Case Cetak Buku Register	86
Tabel 4.22 Spesifikasi Use Case Lihat Buku Ekspedisi.....	87
Tabel 4.23 Spesifikasi Use Case Cetak Buku Ekspedisi	88
Tabel 4.24 Spesifikasi Use Case Validasi Penyerahan SKPD	90
Tabel 4.25 Spesifikasi Use Case Validasi Pembayaran.....	91
Tabel 5.1 Kelas-Kelas Analisis.....	109
Tabel 5.2 Mekanisme Alanisis.....	115
Tabel 6.1 Hasil pengujian prototipe skenario berhasil tambah pengguna oleh Ari Nursurahmat, S. Sos.....	173
Tabel 6.2 Hasil pengujian prototipe skenario data pengguna tidak diisi dengan lengkap oleh Ari Nursurahmat, S. Sos.....	174
Tabel 6.3 Hasil pengujian prototipe skenario berhasil melakukan edit data wajib pajak oleh Ari Nursurahmat, S. Sos.....	174
Tabel 6.4 Hasil pengujian prototipe skenario berhasil melakukan cetak SKPD oleh Ari Nursurahmat, S. Sos.....	175
Tabel 6.5 Hasil pengujian prototipe skenario berhasil melakukan lihat buku ekspedisi oleh Ari Nursurahmat, S. Sos.....	175
Tabel 6.6 Hasil pengujian prototipe skenario berhasil tambah pengguna oleh Budi Muratno, S. Sos	175
Tabel 6.7 Hasil pengujian prototipe skenario data pengguna tidak diisi dengan lengkap oleh Budi Muratno, S. Sos	176
Tabel 6.8 Hasil pengujian prototipe skenario berhasil melakukan edit data wajib pajak Budi Muratno, S. Sos	176
Tabel 6.9 Hasil pengujian prototipe skenario berhasil melakukan cetak SKPD Budi Muratno, S. Sos	177
Tabel 6.10 Hasil pengujian prototipe skenario berhasil melakukan lihat buku ekspedisi oleh Budi Muratno, S. Sos	177
Tabel 6.11 Matriks Traceability.....	179
Tabel 6.12 Tinjauan traceability.....	181

DAFTAR GAMBAR

Gambar 2.1	Bagan Strukur Organisasi Dinas Pendapatan Kabupaten Madiun	8
Gambar 2.2	Contoh BPMN	17
Gambar 2.3	Arsitektur RUP	18
Gambar 2.4	Model Sistem Informasi	20
Gambar 2.5	Notasi Packages	24
Gambar 2.6	Diagram UML	25
Gambar 2.7	Use Case Diagram Submission System	28
Gambar 2.8	Activity Diagram Issue certificate	32
Gambar 2.9	Sequence Diagram	34
Gambar 2.10	Class Diagram Submission System	37
Gambar 2.11	Antarmuka Web Service dengan back-end sistem	38
Gambar 2.12	Web service request/response permintaan yang interaktif	39
Gambar 2.13	Web service proses order pembelian barang	40
Gambar 3.1	Metodologi Penelitian	47
Gambar 4.1	Proses bisnis penerimaan PAD saat ini	52
Gambar 4.2	Proses Bisnis Penerimaan PAD usulan	53
Gambar 4.3	Diagram use case Sistem Informasi Penerimaan PAD Non-PBB	65
Gambar 4.4	Activity diagram login	93
Gambar 4.5	Activity diagram kelola data pengguna	94
Gambar 4.6	Activity diagram kelola pendataan wajib pajak	95
Gambar 4.7	Activity diagram kelola pendataan objek pajak	97
Gambar 4.8	Activity diagram kelola penetapan pajak	98
Gambar 4.9	Activity diagram cetak SKPD	99
Gambar 4.10	Activity diagram lihat laporan penerimaan	100
Gambar 4.11	Activity diagram cetak laporan penerimaan pajak	101
Gambar 4.12	Activity diagram lihat buku register	102
Gambar 4.13	Activity diagram cetak buku register	103
Gambar 4.14	Activity diagram lihat buku ekspedisi	104

Gambar 4.15	Activity diagram cetak buku ekspedisi	105
Gambar 4.16	Activity diagram validasi penyerahan SKPD	106
Gambar 4.17	Activity diagram validasi pembayaran	107
Gambar 5.1	Analisis Arsitektur	108
Gambar 5.2	Unifikasi Kelas-Kelas Analisis	117
Gambar 5.3	Package	118
Gambar 5.4	Sequence diagram login	119
Gambar 5.5	Sequence diagram kelola data pengguna	121
Gambar 5.6	Sequence diagram kelola pendataan wajib pajak.....	123
Gambar 5.7	Sequence diagram kelola pendataan objek pajak.....	125
Gambar 5.8	Sequence diagram kelola penetapan pajak	129
Gambar 5.9	Sequence diagram cetak SKPD	130
Gambar 5.10	Sequence diagram lihat laporan penerimaan pajak	131
Gambar 5.11	Sequence diagram cetak laporan penerimaan pajak.....	133
Gambar 5.12	Sequence diagram lihat buku register.....	134
Gambar 5.13	Sequence diagram cetak buku register	135
Gambar 5.14	Sequence diagram lihat buku ekspedisi	136
Gambar 5.15	Sequence diagram cetak buku ekspedisi.....	137
Gambar 5.16	Sequence Diagram Validasi Penyerahan SKPD.....	138
Gambar 5.17	Sequence diagram validasi pembayaran.....	139
Gambar 5.18	Diagram kelas model	141
Gambar 5.19	Diagram kelas controller	142
Gambar 5.20	Diagram kelas view	143
Gambar 5.21	Pemodelan data.....	144
Gambar 5.22	Perancangan antarmuka login.....	149
Gambar 5.23	Perancangan antarmuka kelola pengguna	150
Gambar 5.24	Perancangan antarmuka tambah pengguna	151
Gambar 5.25	Perancangan antarmuka edit pengguna	151
Gambar 5.26	Perancangan antarmuka konfirmasi hapus pengguna.....	152
Gambar 5.27	Perancangan antarmuka laporan penerimaan pajak.....	153
Gambar 5.28	Perancangan antarmuka kelola data wajib pajak	154
Gambar 5.29	Perancangan antarmuka tambah wajib pajak.....	155

Gambar 5.30	Perancangan antarmuka edit data wajib pajak.....	156
Gambar 5.31	Perancangan antarmuka konfirmasi hapus data wajib pajak	156
Gambar 5.32	Perancangan antarmuka kelola objek pajak	157
Gambar 5.33	Perancangan antarmuka tambah objek pajak	158
Gambar 5.34	Perancangan antarmuka edit data objek pajak	159
Gambar 5.35	Perancangan antarmuka konfirmasi hapus objek pajak	159
Gambar 5.36	Perancangan antarmuka kelola penetapan pajak – data pajak	160
Gambar 5.37	Perancangan antarmuka cetak SKPD	161
Gambar 5.38	Perancangan antarmuka penetapan pajak air tanah	161
Gambar 5.39	Perancangan antarmuka penetapan pajak parkir.....	163
Gambar 5.40	Perancangan antarmuka buku ekspedisi.....	164
Gambar 5.41	Perancangan antarmuka buku register	165
Gambar 5.42	Halaman login	166
Gambar 5.43	Halaman Kelola Pengguna	166
Gambar 5.44	Halaman Laporan Penerimaan Pajak	167
Gambar 5.45	Halaman Kelola Data Wajib Pajak	167
Gambar 5.46	Halaman Tambah Data Wajib Pajak	168
Gambar 5.47	Halaman Edit Data Wajib Pajak.....	168
Gambar 5.48	Halaman Konfirmasi Hapus Wajib Pajak	169
Gambar 5.49	Halaman Kelola Objek Pajak.....	169
Gambar 5.50	Halaman Kelola Penetapan Pajak-Data SKPD.....	170
Gambar 5.51	Halaman Formulir Cetak SKPD	170
Gambar 5.52	Halaman Kelola Penetapan Pajak – Formulir Penetapan Pajak Air Tanah.....	171
Gambar 5.53	Halaman Buku Ekspedisi.....	171
Gambar 5.54	Halaman Buku Register	172
Gambar A.1	Surat izin peneltian lembar pertama.....	187
Gambar A.2	Surat izin penelitian lembar kedua.....	188
Gambar A.3	Surat Izin Penelitian lembar ketiga.....	189

Gambar B.1	Hasil tinjauan prototipe oleh Ari Nursurahmat, S. Sos lembar pertama.....	190
Gambar B.2	Hasil tinjauan prototipe oleh Ari Nursurahmat, S. Sos lembar kedua.....	191
Gambar B.3	Hasil tinjauan navigasi prototipe oleh Ari Nursurahmat, S. Sos	192
Gambar B.4	Hasil tinjauan antarmuka prototipe oleh Ari Nursurahmat, S. Sos	193
Gambar B.5	Hasil tinjauan prototipe oleh Budi Muratno, S. Sos lembar pertama.....	194
Gambar B.6	Hasil tinjauan prototipe oleh Budi Muratno, S. Sos lembar kedua.....	195
Gambar B.7	Hasil tinjauan navigasi prototipe oleh Budi Muratno, S. Sos.....	196
Gambar B.8	Hasil tinjauan antarmuka prototipe oleh Budi Muratno, S. Sos.....	197



DAFTAR LAMPIRAN

LAMPIRAN A SURAT IZIN PENELITIAN	187
A.1 Surat Izin Penelitian Lembar Pertama	187
A.2 Surat Izin Penelitian Lembar kedua	188
A.3 Surat Izin Penelitian Lembar ketiga	189
LAMPIRAN B HASIL EVALUASI	190
B.1 Hasil tinjauan prototipe oleh Ari Nursurahmat, S. Sos lembar pertama	190
B.2 Hasil tinjauan prototipe oleh Ari Nursurahmat, S. Sos lembar kedua	191
B.3 Hasil tinjauan navigasi prototipe oleh Ari Nursurahmat, S. Sos	192
B.4 Hasil tinjauan antarmuka prototipe oleh Ari Nursurahmat, S. Sos	193
B.5 Hasil tinjauan prototipe oleh Budi Muratno, S. Sos lembar pertama.....	194
B.6 Hasil tinjauan prototipe oleh Budi Muratno, S. Sos lembar kedua.....	195
B.7 Hasil tinjauan navigasi prototipe oleh Budi Muratno, S. Sos	196
B.8 Hasil tinjauan antarmuka prototipe oleh Budi Muratno, S. Sos.....	197

BAB 1 PENDAHULUAN

1.1 Latar belakang

Salah satu sumber Pendapatan suatu Daerah menurut Undang-Undang Republik Indonesia Nomor 33 Tahun 2004 adalah Pendapatan Asli Daerah (PAD). Pendapatan Asli Daerah bisa diperoleh dari Pajak Daerah ataupun Restribusi Daerah. Pendapatan Asli Daerah ini dipungut berdasarkan Peraturan Daerah sesuai dengan peraturan perundang-undangan. (Undang-Undang Republik Indonesia Nomor 33 Tahun 2004).

Pemungutan PAD di Kabupaten Madiun dilakukan oleh Dinas Pendapatan (DISPENDA) Kabupaten Madiun. DISPENDA Kabupaten Madiun merupakan sebuah badan atau organisasi yang berada dibawah pemerintahan Kabupaten Madiun. Pada tahun 2013, Kabupaten Madiun memiliki 15 kecamatan, 8 kelurahan, 198 desa (Badan Pusat Statistik, 2013). Banyaknya desa yang dimiliki oleh Kabupaten Madiun tentu akan mempengaruhi banyaknya sumber-sumber penerimaan Pendapatan Asli Daerah.

Dinas Pendapatan Kabupaten Madiun sudah bekerja sama dengan Bank Jatim dalam pembayaran / penerimaan Pendapatan Asli Daerahnya. Jadi, semua pembayaran PAD Kabupaten Madiun dapat dilakukan melalui Bank Jatim. Alur penerimaan PAD di Kabupaten Madiun ini dimulai dari pendataan dan penetapan siapa yang wajib menyetor, berapa jumlah penyetoran, dan jenis PAD apa yang harus disetorkan oleh bidang pendataan dan penetapan. Kemudian dicetak Surat Ketetapan Pajak Daerah (SKPD) ataupun Surat Ketetapan Restribusi Daerah (SKRD) untuk diserahkan kepada penyetor oleh bidang penagihan dan keberatan. Setelah pihak penyetor menerima SKPD dan/atau SKRD, penyetor wajib membayar jumlah setoran di Bank Jatim dengan membawa SKPD dan/atau SKRD. Setelah itu uang pembayaran, struk pembayaran dan rekening koran akan diserahkan ke Kas Daerah oleh pihak Bank Jatim. Setelah kas daerah selesai melakukan perekapan, rekening koran dan struk pembayaran akan diserahkan ke pihak DISPENDA oleh Kas Daerah. DISPENDA bidang pembukuan akan melakukan pembukuan dan perekapan dari struk dan rekening koran tersebut. Untuk mengetahui siapa saja yang harus ditagih bidang penagihan harus melihat rekap dari bidang pembukuan serta SKPD dan/atau SKRD.

Penerimaan PAD selain PBB di DISPENDA belum memiliki sistem yang terintegrasi secara internal antar bidang di DISPENDA dan eksternal dengan pihak Bank Jatim. Seperti yang sudah dijelaskan bahwa untuk mengetahui siapa pihak penyetor yang sudah membayar ataupun belum membayar bidang penagihan harus melihat dari rekap bidang pembukuan ataupun struk pembayaran dari Bank Jatim kemudian dibandingkan dengan SKPD dan/atau SKRD. Hal seperti ini tentu akan memperlambat penerimaan PAD Kabupaten Madiun. Untuk menangani masalah tersebut dibutuhkan adanya Sistem Informasi Penerimaan PAD yang terintegrasi secara internal antar bidang di DISPENDA maupun secara eksternal dengan pihak Bank Jatim.

Sistem informasi terintegrasi dalam penerimaan Pendapatan Asli Daerah non-PBB menjadi hal yang sangat penting untuk membantu proses pengelolaan penerimaan Pendapatan Asli Daerah Kabupaten Madiun agar proses pendataan, pelaporan dan penagihan dapat bekerja dengan cepat tanpa harus merekap ulang Rekening Koran dari Bank Jatim. Dengan sistem informasi yang terintegrasi hubungan antar bagian dan kepada pihak Bank Jatim akan menjadi lebih mudah serta setiap bagian yang terlibat proses penerimaan PAD akan lebih mudah menjalankan tugasnya. Selain itu kemungkinan terjadi kesalahan manusia (*human error*) akan lebih kecil karena informasi pembayaran yang divalidasi oleh pihak Bank Jatim melalui sistem serta informasi tersebut akan bisa dilihat oleh pihak Dinas Pendapatan.

Dalam pengembangan suatu sistem informasi, sebelum dilakukan implementasi perlu dilakukan tahap analisis persyaratan sistem dan perancangan sistem. Dari berbagai studi ditemukan 70% dari proyek gagal adalah disebabkan oleh persyaratan/*requirements* yang tidak lengkap dan akurat. Oleh karena itu, tahap analisis persyaratan merupakan tahap yang penting untuk mencegah kegagalan proyek (Tantra, 2012). Tahap perancangan sistem juga merupakan tahap yang penting dalam pengembangan suatu sistem, karena dengan perancangan sistem, pengembang (*developer*) sudah dapat menilai terlebih dahulu beberapa aspek kualitas perangkat lunak atau sistem sebelum dilaksanakan tahap implementasi sistem. Selain itu perancangan merupakan tahap membuat representasi atau model perangkat lunak berupa rincian arsitektur perangkat lunak, struktur data, antarmuka dan komponen lain yang diperlukan untuk melakukan tahap implementasi sistem (Pressman, 2010).

Berdasarkan informasi yang telah disampaikan, penulis mengambil judul skripsi “Analisis dan Perancangan Sistem Informasi Penerimaan Pendapatan Asli Daerah (PAD) Non-PBB (Pajak Bumi Bangunan) Kabupaten Madiun”. Analisis dan perancangan Sistem Informasi Pengelolaan Penerimaan PAD ini diharapkan dapat mempermudah dan menjadi pedoman dalam tahap implementasi sistem serta akan menghasilkan perangkat lunak yang berkualitas. Untuk mendukung analisis dan perancangan sistem, dilakukan juga pemodelan proses bisnis yang menjadi konteks dari sistem informasi tersebut.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah disampaikan di atas, maka dapat dirumuskan permasalahan pada skripsi ini adalah sebagai berikut :

1. Bagaimanakah proses bisnis penerimaan pendapatan asli daerah yang saat ini telah diterapkan dan yang diusulkan sesuai dengan sistem informasi yang akan dirancang?
2. Bagaimanakah persyaratan Sistem Informasi Penerimaan Pendapatan Asli Daerah (PAD) Non-PBB (Pajak Bumi Bangunan) Kabupaten Madiun yang diusulkan?
3. Bagaimanakah rancangan sistem informasi tersebut?

4. Bagaimanakah hasil evaluasi terhadap persyaratan dan rancangan sistem informasi tersebut?

1.3 Tujuan

Tujuan yang ingin dicapai dalam skripsi ini adalah :

1. Menganalisis dan memodelkan proses bisnis penerimaan pajak di Dinas Pendapatan Kabupaten Madiun saat ini.
2. Menganalisis dan mengusulkan proses bisnis baru sesuai dengan sistem informasi penerimaan pajak yang akan dibuat.
3. Menganalisis persyaratan dan merancang Sistem Informasi Penerimaan Pendapatan Asli Daerah (PAD) Non-PBB (Pajak Bumi Bangunan) Kabupaten Madiun.
4. Mengevaluasi hasil analisis persyaratan dan perancangan sistem informasi tersebut.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat bagi penulis, pembaca dan instansi. Adapun manfaat yang diharapkan adalah sebagai berikut :

1. Bagi Dinas Pendapatan Kabupaten Madiun
 - a. Mendapatkan informasi mengenai proses bisnis penerimaan PAD yang berjalan saat ini.
 - b. Mendapatkan analisis dan perancangan Sistem Informasi Penerimaan PAD.
 - c. Mempunyai pedoman ketika akan mengimplementasikan Sistem Informasi Penerimaan PAD.
2. Bagi Penulis
 - a. Menambah pengetahuan mengenai analisis dan pemodelan proses bisnis.
 - b. Menambah pengetahuan mengenai analisis persyaratan dan perancangan untuk pengembangan suatu sistem yang berkualitas.
 - c. Sarana pengembangan pengetahuan dan wawasan terkait perancangan sistem berdasarkan kebutuhan sistem.
 - d. Menambah pengetahuan mengenai evaluasi dari analisis dan perancangan yang telah dibuat.
3. Bagi Pembaca
 - a. Memberikan informasi seputar Pendapatan Asli Daerah
 - b. Memberikan wawasan mengenai analisis dan pemodelan proses bisnis.
 - c. Memberikan wawasan mengenai analisis dan perancangan suatu sistem.

- d. Memberikan wawasan mengenai evaluasi hasil analisis dan perancangan yang telah dibuat.

1.5 Batasan masalah

Ruang lingkup masalah untuk menyelesaikan masalah yang sudah dirumuskan dalam penelitian ini adalah sebagai berikut :

1. Sistem yang akan dibuat dapat digunakan untuk mengelola penerimaan PAD yang bersumber dari Pajak Parkir dan Pajak Air Tanah.
2. Sistem dirancang untuk dapat diimplementasikan dan nantinya digunakan oleh pihak Dinas Pendapatan Kabupaten Madiun bidang Pendataan dan Penetapan, bidang Pembukuan dan Pelaporan, bidang Penagihan dan Keberatan serta Bank Jatim.
3. 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 laporan skripsi ini adalah :

BAB I

PENDAHULUAN

Membahas latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan penelitian.

BAB II

LANDASAN KEPUSTAKAAN

Menguraikan dasar-dasar teori dan referensi yang berkaitan dengan penelitian ini.

BAB III

METODOLOGI

Membahas mengenai metode, prosedur dan strategi yang digunakan dalam melakukan penelitian.

BAB IV

PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN

Membahas pemodelan proses bisnis yang terjadi saat ini dan proses bisnis usulan, analisis permasalahan, identifikasi pemangku kepentingan dan aktor, serta analisis persyaratan sistem.

BAB V

PERANCANGAN

Membahas tentang perancangan berdasarkan analisis yang telah dilakukan pada tahap sebelumnya

BAB VI

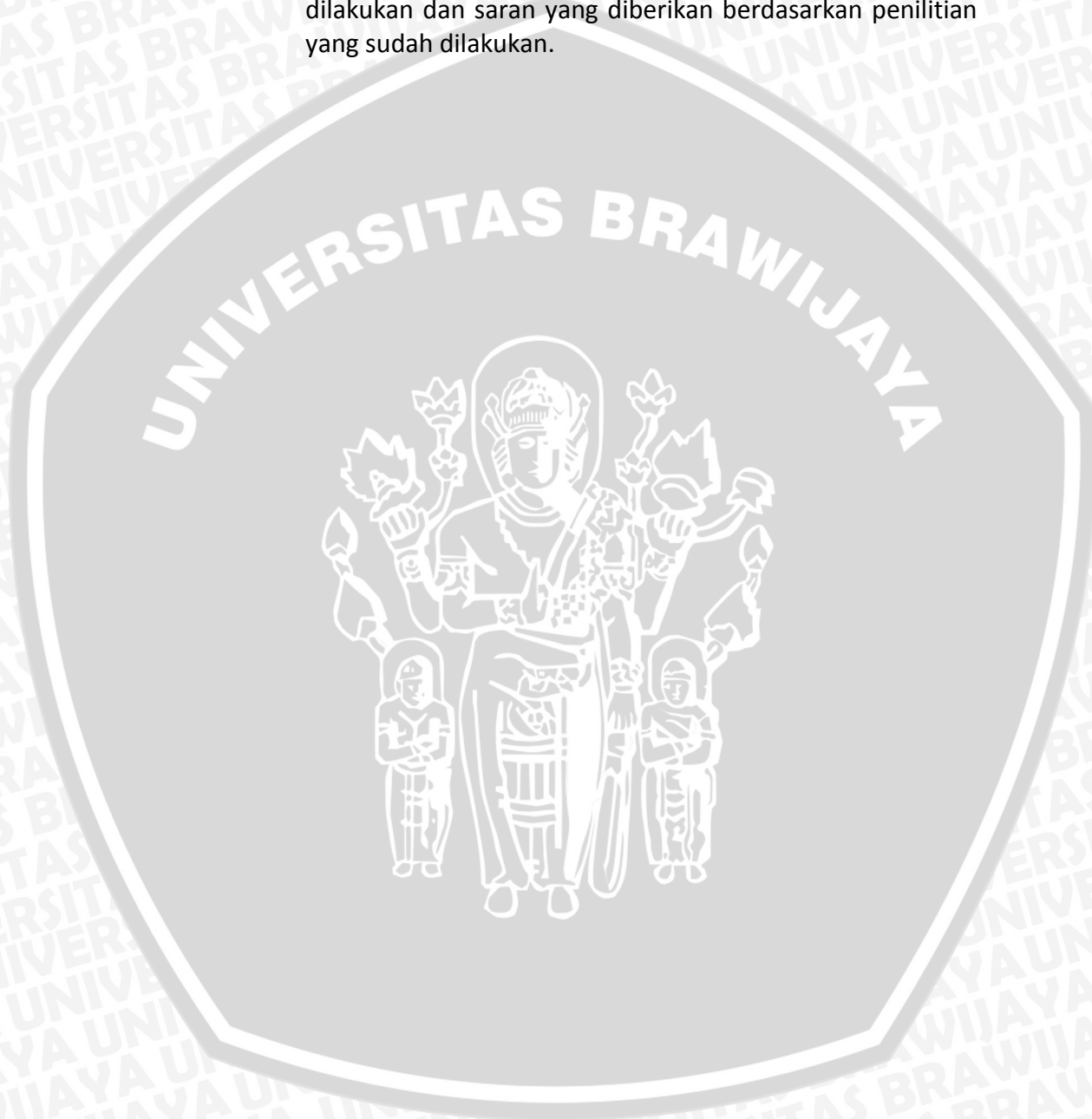
EVALUASI

Membahas tentang evaluasi terhadap prototipe serta hasil dari analisis dan perancangan sistem yang telah dibuat.

BAB VII

PENUTUP

Memuat kesimpulan dari keseluruhan penelitian yang sudah dilakukan dan saran yang diberikan berdasarkan penilitian yang sudah dilakukan.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Penelitian sebelumnya yang berjudul “Sistem Pelaporan dan Pengolahan Data Pajak Bumi dan Bangunan (PBB) Kabupaten Jombang Berbasis Web Menggunakan Framework Codeigniter” oleh Vinishya Amin pada tahun 2014 merupakan penelitian implementatif yang bertujuan untuk membuat merancang sistem pelaporan dan pengolahan data untuk membantu memusatkan pengolahan data pembayaran PBB. Alasan mengapa dirancang sebuah sistem pelaporan dan pengolahan data pada skripsi ini adalah pada Dinas Pendapatan Daerah Kabupaten Jombang data pembayaran PBB terpencar-pencar sesuai tempat pembayaran yang dilakukan oleh wajib pajak. Dalam hal ini pembayaran PBB di Kabupaten Jombang dapat dilakukan melalui instansi yang ditunjuk atau Bank. Selain itu setiap instansi yang ditunjuk perlu melaporkan data pembayaran secara manual kepada Dinas Pendapatan Daerah Kabupaten Jombang untuk dilakukan penggabungan dan pengolahan data lebih lanjut (Amin, 2014).

Keterkaitan penelitian tersebut dengan yang dilakukan oleh penulis adalah sama-sama menangani masalah pembayaran pajak daerah pada suatu Dinas Pendapatan Daerah. Pembayaran pajak daerah pada Dinas Pendapatan Kabupaten Jombang dapat dilakukan di instansi dan juga di Bank yang ditunjuk begitu juga di Dinas Pendapatan Kabupaten Madiun, hanya saja di Kabupaten Jombang pembayaran tidak terpusat pada bank. Pembayaran pajak di Dinas Pendapatan Kabupaten Madiun sudah terpusat pada Bank. Jadi, meskipun wajib pajak membayar di instansi nantinya uang pajak tersebut akan diserahkan di Bank Jatim untuk dimasukkan ke rekening Dispenda Kabupaten Madiun. Tetapi dalam hal pengolahan data dari Bank yang ditunjuk, Dinas Pendapatan Kabupaten Jombang dan Dinas Pendapatan Kabupaten Madiun juga sama-sama masih dilakukan secara manual dan tidak langsung terintegrasi dengan pihak Bank.

2.2 Dasar Teori

2.2.1 Dinas Pendapatan 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. Dinas Pendapatan Kabupaten Madiun ini berlokasi di Jalan Alun-alun Utara No. 4 Madiun.

2.2.1.1 Sejarah Dinas Pendapatan Kabupaten Madiun

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

Pada awalnya, tahun 2009 dalam struktur organisasi Pemerintah Kabupaten Madiun bergabung dalam satu garis struktur organisasi dengan Kantor Pengelola Keuangan Aset Daerah sampai tahun 2011. 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.

2.2.1.2 Visi dan Misi Dinas Pendapatan Kabupaten Madiun

Visi :

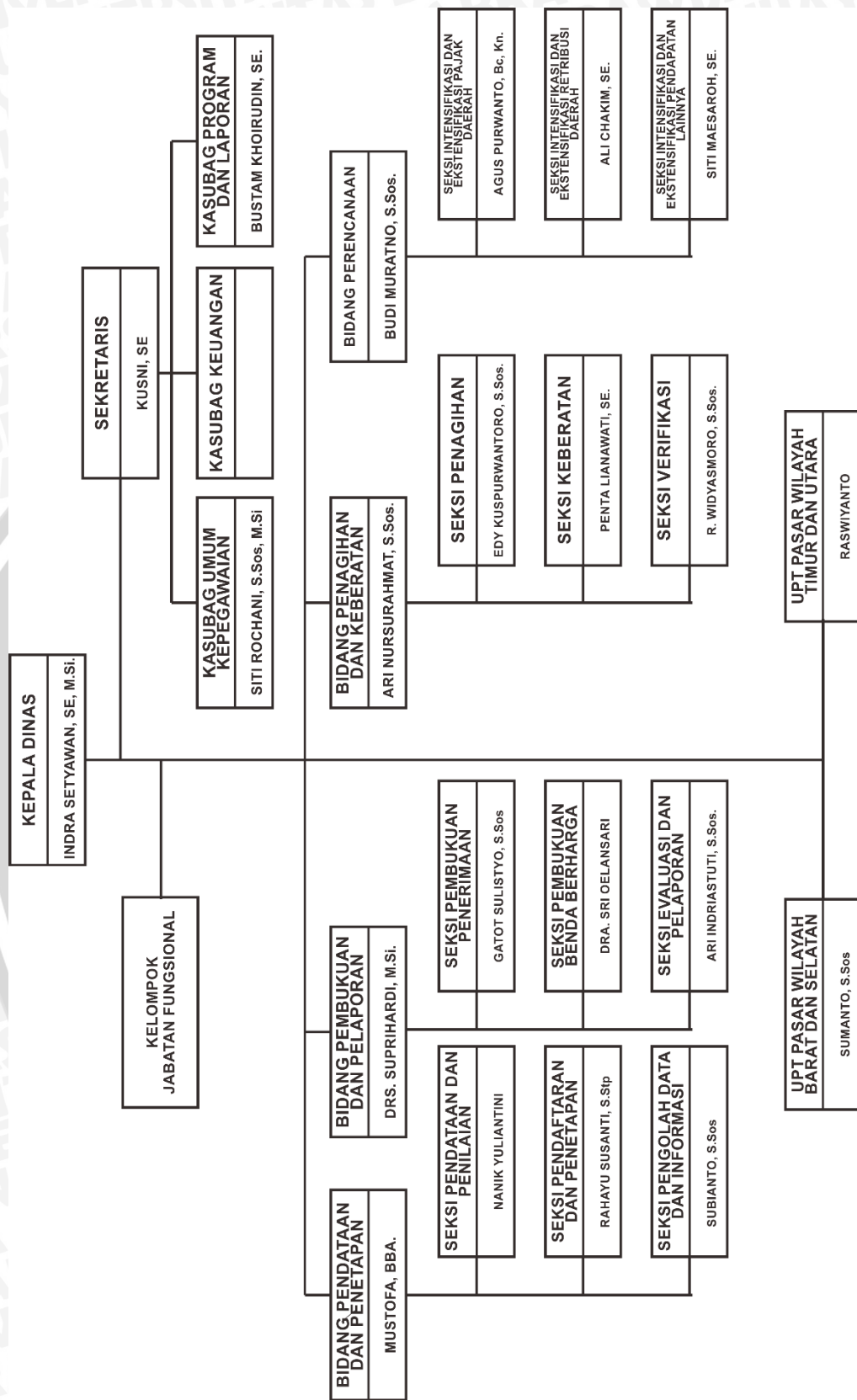
Tercapainya pendapatan daerah yang optimal, berkesinambungan serta seimbang dengan potensi daerah untuk mewujudkan kemandirian pemerintahan kabupaten Madiun dalam rangka meningkatkan kesejahteraan masyarakat, serta menjadi inisiator peningkatan pendapatan asli daerah.

Misi :

1. Peningkatan kualitas pelayanan masyarakat yang berorientasi pada kepuasan serta pembudayaan taat pajak dan restribusi 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.

2.2.1.3 Struktur Organisasi Dinas Pendapatan Kabupaten Madiun

Gambar 2.1 menunjukkan struktur organisasi Dinas Pendapatan Kabupaten Madiun. Dinas Pendapatan Kabupaten Madiun dipimpin oleh Kepala Dinas.



Gambar 2.1 Bagan Strukur Organisasi Dinas Pendapatan Kabupaten Madiun

2.2.1.4 Pendapatan Asli Daerah

Pendapatan Daerah menurut Undang-Undang Republik Indonesia Nomor 33 Tahun 2004 adalah Pendapatan Asli Daerah (PAD). Pendapatan Asli Daerah merupakan pendapatan yang diperoleh Daerah yang dipungut berdasarkan Peraturan Daerah sesuai dengan peraturan perundang-undangan. Pendapatan Asli Daerah ini bersumber dari Pajak Daerah, Restribusi Daerah, hasil pengelolaan kekayaan Daerah yang dipisahkan, dan lain-lain PAD yang sah. Lain-lain PAD yang sah disini meliputi hasil penjualan kekayaan Daerah yang tidak dipisahkan; jasa giro; pendapatan bunga; keuntungan selisih nilai tukar rupiah terhadap mata uang asing; dan komisi, potongan, ataupun bentuk lain sebagai akibat dari penjualan dan/atau pengadaan barang dan/atau jasa oleh Daerah (Undang-Undang Republik Indonesia Nomor 33 Tahun 2004).

2.2.1.5 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.

2.2.1.6 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 : 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.

2.2.1.7 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 Proses Bisnis

Proses bisnis adalah sekumpulan aktivitas terkoordinasi yang dilakukan oleh manusia atau alat dengan tujuan untuk mewujudkan hasil tertentu. Proses bisnis terdiri dari serangkaian kegiatan terkoordinasi yang mencapai tujuan bisnis tertentu (Pant & Juric, 2008). 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 sistem informasi. Contoh aktivitas manual yaitu 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).

Proses bisnis direpresentasikan / digambarkan dengan model proses bisnis. Model proses bisnis disajikan dalam suatu diagram proses bisnis. Setiap diagram proses bisnis terdiri dari satu set elemen pemodelan (Weske, 2007). Sebuah pemahaman yang baik mengenai proses bisnis sangat penting untuk mengembangkan IT *support*. Aplikasi yang menyediakan dukungan *end-to-end* untuk proses bisnis, dapat dikembangkan secara efisien hanya jika kita memahami proses bisnis secara rinci (Pant & Juric, 2008).

Pemodelan proses bisnis memerlukan notasi dalam memodelkan prosesnya. Beberapa contoh notasi yang dapat digunakan untuk memodelkan proses bisnis diantaranya adalah EPC (*Event Proses Chain*), EEPK (*Extended Event Process Chain*), UML *Activity Diagram*, dan BPMN (*Business Process Model and Notation*) (Pant & Juric, 2008).

2.2.2.1 BPMN (*Business Process Model and Notation*)

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).

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

1. *Flow Object*
2. *Data*
3. *Connecting Object*
4. *Swimlanes*
5. *Artifact*.

Tabel 2.1 merupakan tabel notasi setiap elemen dari BPMN.

Flow Object adalah elemen grafis utama untuk menentukan perilaku dari proses bisnis. Terdapat tiga Flow Object (Object Management Group (OMG), 2011):

1. *Events*
2. *Aktivitas*
3. *Gateways*.

Data direpresentasikan oleh empat elemen (Object Management Group (OMG), 2011) :

1. *Data Objects*
2. *Data Inputs*
3. *Data Outputs*
4. *Data Stores*.

Terdapat empat cara untuk menghubungkan antar *Flow Object* atau informasi lainnya (Object Management Group (OMG), 2011) :

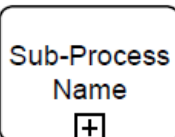
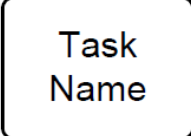
1. *Sequence Flows*
2. *Message Flows*
3. *Association*
4. *Data Associations*.

Terdapat dua acara pengelompokan elemen pemodelan primer melalui “Swimlanes” (Object Management Group (OMG), 2011):

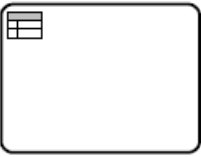
1. *Poll*
2. *Lanes*

Artifact merupakan elemen yang digunakan untuk memberikan informasi tambahan dari proses. Terdapat dua *Artifact* standar yaitu *Group* dan *Text Annotation* (Object Management Group (OMG), 2011).

Tabel 2.1 Notasi BPMN

Nama	Notasi	Deskripsi
<i>Start Event</i>		Notasi untuk memulai proses
<i>End Events</i>		Notasi untuk mengakhiri proses
<i>Intermediate Events</i>		Notasi yang menunjukkan di mana terjadi sesuatu (Event) di antara awal dan akhir proses. <i>Intermediate event</i> akan mempengaruhi aliran proses tetapi tidak memulai atau mengakhiri proses
<i>Activity</i>		<i>Activity</i> merupakan kegiatan atau pekerjaan yang dilakukan oleh perusahaan dalam sebuah proses.
<i>Collapsed Sub-Process</i>		Merupakan aktivitas di mana di dalam aktivitas tersebut terdapat proses yang terjadi.
<i>Task</i>		<i>Task</i> adalah bentuk <i>atomic</i> dari <i>Activity</i> yang disertakan dalam aliran proses. <i>Task</i> digunakan ketika pekerjaan dalam proses tidak dapat dipecah ke tingkat yang lebih detail.
<i>Service Task</i>		Task yang menggunakan beberapa jenis layanan, dapat berupa <i>web service</i> atau aplikasi otomatis.

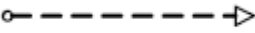
Tabel 2.1 Notasi BPMN (lanjutan)

Nama	Notasi	Deskripsi
Send Task		<i>Simple task</i> yang dirancang untuk mengirim pesan kepada partisipan eksternal (relatif terhadap proses). <i>Task</i> selesai ketika pesan selesai dikirim.
Receive Task		<i>Simple task</i> yang dirancang untuk menunggu pesan tiba dari partisipan eksternal (relatif terhadap proses). <i>Task</i> selesai ketika pesan telah diterima.
User Task		<i>Task</i> di mana manusia melakukan tugas dengan bantuan perangkat lunak aplikasi.
Manual Task		<i>Task</i> yang dilakukan tanpa bantuan mesin eksekusi proses bisnis atau aplikasi lain.
Business Rule Task		<i>Task</i> yang menyediakan mekanisme proses untuk memberikan masukan ke mesin aturan bisnis dan untuk mendapatkan output dari perhitungan yang disediakan oleh mesin aturan bisnis.
Script Task		<i>Task</i> yang dijalankan oleh <i>business process engine</i> . Pemodel atau pelaksana mendefinisikan <i>script</i> dalam bahasa yang dapat diterjemahkan oleh mesin.
Gateway		<i>Gateway</i> digunakan untuk mengontrol konvergensi dan divergensi dari <i>sequence flow</i> dalam proses dan dalam <i>Choreography</i> .

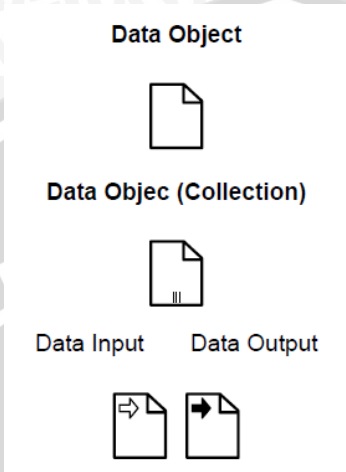
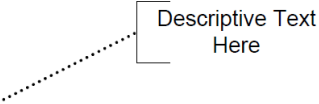
Tabel 2.1 Notasi BPMN (lanjutan)

Nama	Notasi	Deskripsi
Gateway Control Types	Exclusive  or  Event-Based   Parallel Event-Based  Inclusive  Complex  Parallel 	• Exclusive (Keputusan): digunakan untuk membuat jalur alternatif dalam sebuah aliran proses. • Event-based: • Inclusive: digunakan untuk membuat jalur alternatif maupun paralel dalam sebuah aliran proses, semua ekspresi kondisi dievaluasi. • Complex: digunakan untuk model perilaku sinkronisasi kompleks. • Parallel: digunakan untuk menggabungkan aliran paralel dan untuk membuat aliran paralel.
Sequence Flow		Sequence Flow digunakan untuk menunjukkan urutan Flow Elements dalam sebuah proses atau Choreography. Setiap Sequence Flow hanya memiliki satu sumber dan satu target. Flow Elements terdiri dari: Events (Start, Intermediate, dan End), Activities (Task dan Sub-Process; untuk Proses), Choreography Activities (Choreography Task dan Sub-Choreography; untuk Choreographies), dan Gateways.
Conditional Flow		Sequence flow yang mendefinisikan ekspresi kondisi, aliran akan berjalan hanya jika kondisi bernilai true.
Default Flow		Sequence flow yang digunakan hanya ketika semua conditional flow / aliran kondisi yang

Tabel 2.1 Notasi BPMN (lanjutan)

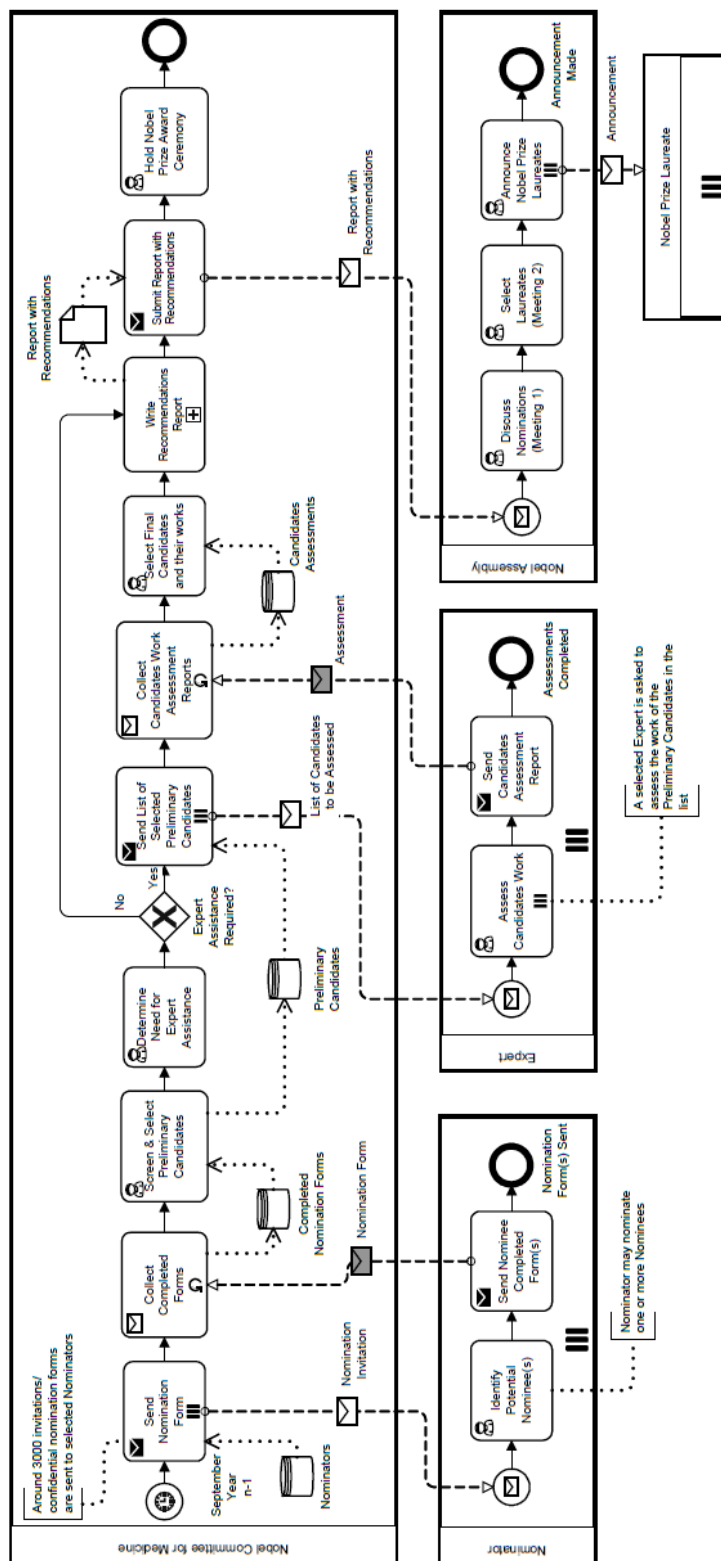
Nama	Notasi	Deskripsi
		keluar bernilai <i>false/not true</i> pada saat <i>runtime</i> .
Message Flow		Notasi yang digunakan untuk menunjukkan aliran pesan antara dua partisipan yang siap untuk mengirim dan menerima.
Association / Asosiasi		Notasi yang digunakan untuk menghubungkan informasi dan <i>artifact</i> dengan <i>flow object</i> .
Pool		<i>Pool</i> adalah representasi grafis dari partisipan dalam <i>Collaboration</i> . Partisipan dapat berupa <i>PartnerEntity</i> tertentu (misalnya, sebuah perusahaan) atau bisa berupa yang lebih umum (<i>PartnerRole</i> yang lebih umum (misalnya, pembeli, penjual, atau produsen)).
Lane		Lane adalah sebuah sub-partisi dalam sebuah proses, kadang-kadang dalam <i>Pool</i> , dan akan memanjang sejauh panjang Proses baik secara vertikal maupun horizontal. <i>Lane</i> digunakan untuk mengatur dan mengkategorikan aktivitas.

Tabel 2.1 Notasi BPMN (lanjutan)

Nama	Notasi	Deskripsi
Data object		• Data Object: menunjukkan aliran informasi yang ada pada sebuah proses seperti dokumen bisnis, <i>e-mail</i>, atau surat. • Collection: menunjukkan sekumpulan informasi seperti daftar permintaan item. • Data Input-Output: Data masukan adalah masukan luaran untuk seluruh proses, data keluaran adalah data hasil dari seluruh proses. • Data Input-Output: Data masukan adalah masukan luaran untuk seluruh proses, data keluaran adalah data hasil dari seluruh proses.
Message		Notasi digunakan untuk menggambarkan isi komunikasi antar dua partisipan.
Group		Notasi digunakan untuk pengelompokan elemen grafis yang berada dalam kategori yang sama. Jenis pengelompokan tidak mempengaruhi <i>sequence flow</i> dalam grup. Nama kategori muncul dalam diagram sebagai label grup.
Text Annotation (dihubungkan dengan Asosiasi)		Memberikan informasi teks tambahan untuk pembaca diagram BPMN

Sumber : Object Management Group / OMG (2011)

Gambar 2.2 berikut ini merupakan contoh pemodelan proses bisnis dengan BPMN.

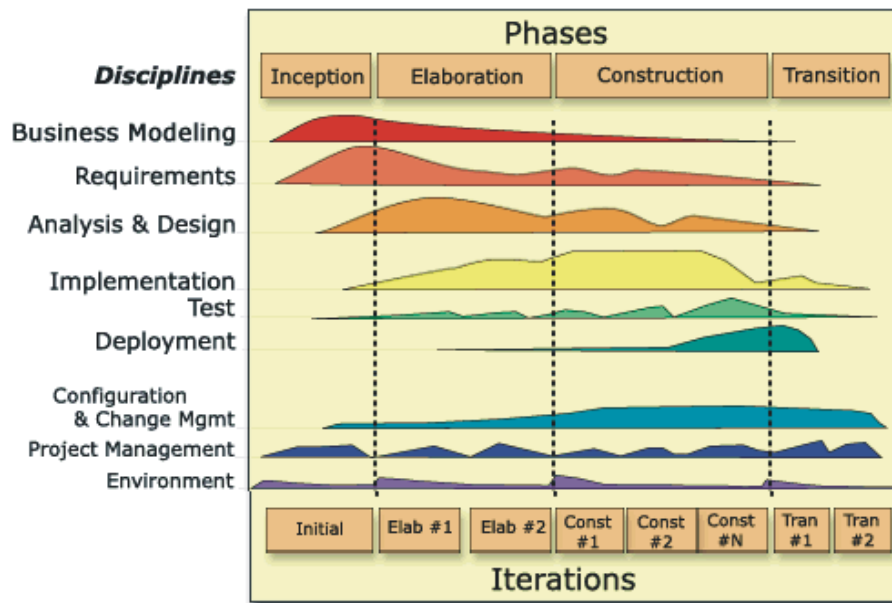


Gambar 2.2 Contoh BPMN

2.2.3 Rational Unified Process (RUP)

RUP adalah sebuah proses untuk mengembangkan suatu perangkat lunak. RUP menggunakan pendekatan disiplin untuk menetapkan tugas dan tanggung jawab dalam organisasi pengembang. Tujuannya adalah untuk memastikan membuat perangkat lunak yang berkualitas yang memenuhi kebutuhan pengguna dalam waktu dan biaya yang dapat diprediksi (IBM Corporation, 2003).

Gambar 2.3 menunjukkan arsitektur RUP. RUP terorganisir menjadi dua struktur/dimensi yaitu waktu dan isi/konten. Dimensi waktu merujuk pada aspek siklus hidup proses. Dimensi konten merujuk pada disiplin berupa aktivitas kelompok yang logis (IBM Corporation, 2003).



Gambar 2.3 Arsitektur RUP

Sumber : IBM Corporation (2003)

Berdasarkan dimensi waktu RUP memiliki empat fase (IBM Corporation, 2003):

1. *Inception*: pendefinisian ruang lingkup proyek, apa yang termasuk dalam proyek dan apa saja yang tidak termasuk ke dalam proyek.
2. *Elaboration*: merencanakan proyek, menspesifikasikan fitur-fitur dan menyusun dasar arsitektur.
3. *Construction*: membangun produk dengan beberapa iterasi.
4. *Transition*: melakukan transisi produk ke pengguna dengan melakukan pelatihan, instalasi dan dukungan.

Waktu yang dibutuhkan dalam setiap tahap/fase bervariasi. Untuk proyek yang kompleks dengan banyak tantangan teknis dan persyaratan yang belum jelas, fase elaborasi mungkin dilakukan 3-5 iterasi. Untuk proyek yang sangat sederhana, Elaborasi mungkin hanya terjadi iterasi tunggal (IBM Corporation, 2003).

Konten RUP diatur dalam dimensi disiplin. Disiplin yang dimaksud adalah kumpulan aktivitas terkait pada area yang bersangkutan. Disiplin ini terdiri dari (IBM Corporation, 2003):

1. *Business Modelling*
2. *Requirements*
3. *Analysis & Design*
4. *Implementation*
5. *Test*
6. *Deployment*
7. *Configuration & Change Management*
8. *Environment*

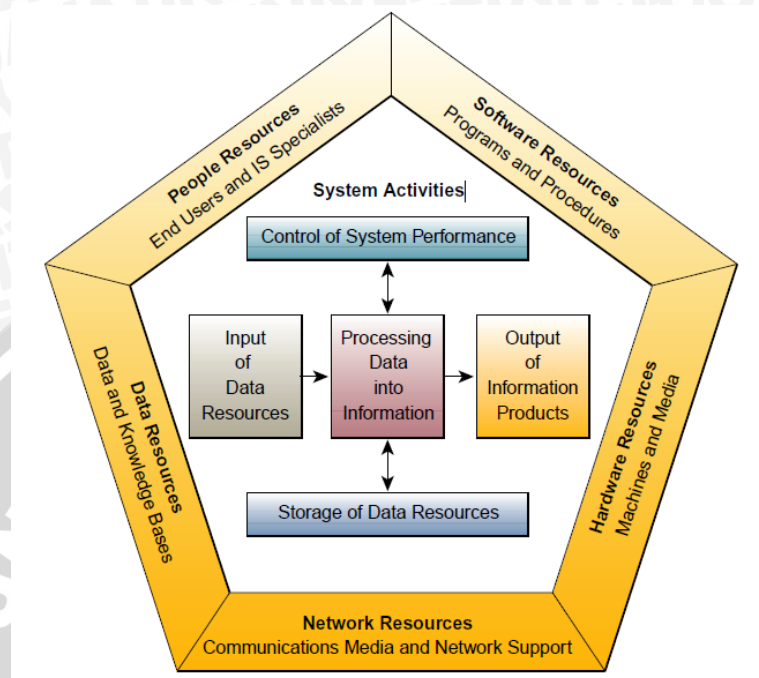
Pada penelitian ini, fase yang diterapkan berdasarkan RUP adalah fase *Inception* dan *Elaboration*. Selama fase *inception* dilakukan pendefinisian *scope*/lingkup dari proyek, apa yang termasuk dan tidak termasuk dalam proyek. Hal ini dilakukan dengan mengidentifikasi aktor dan *use case*. Pada fase *elaboration* yang dilakukan adalah mendapatkan pemahaman yang baik mengenai persyaratan dan membangun dasar *baseline*/ dasar arsitektur. Tetapi dalam penelitian ini, fase *elaboration* tidak dilakukan sampai analisis estimasi biaya dan sumber daya. Disiplin pada RUP yang diterapkan pada penelitian ini adalah *business modelling*, *requirement* serta *analysis and design*. Selain itu pada penelitian ini praktik-praktik baik dalam RUP digunakan dalam penyusunan model *use case* serta untuk analisis *use case* (mulai dari *use case* ke kelas-kelas analisis dan mekanisme analisis).

2.2.4 Sistem Informasi

Sebuah sistem informasi merupakan kombinasi yang terorganisir antara orang, *hardware*, *software*, komunikasi jaringan, sumber data dan kebijakan dan prosedur yang menyimpan, mengambil, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Sistem informasi digunakan oleh orang/user 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).

Sistem informasi adalah suatu sistem yang menerima data sebagai input dan memproses data tersebut menjadi hingga menghasilkan output berupa informasi. Untuk mencapai tujuannya sistem informasi melibatkan berbagai komponen dan aktifitas. 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 manusia, *hardware*, *software*, data, dan jaringan untuk melakukan input, proses, output,

penyimpanan dan kontrol dengan tujuan mengubah sumber data menjadi sebuah informasi (O'Brien & Marakas, 2010).



Gambar 2.4 Model Sistem Informasi

Sumber : O'Brien & Marakas (2010)

2.2.4.1 Analisis Sistem Informasi

Analisis sistem merupakan aktivitas memahami dan melakukan spesifikasi dengan detail mengenai apa saja yang harus dilakukan oleh sistem. Sedangkan perancangan sistem merupakan aktivitas menjelaskan dengan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan. Dapat disimpulkan bahwa analisis dan desain sistem informasi adalah proses organisasional kompleks di mana sistem informasi berbasis komputer diimplementasikan (Fatta, 2007).

Tahapan analisis merupakan tahapan pengembangan sistem untuk mempelajari sistem yang sedang berjalan dan mengusulkan sistem pengganti. Pada tahap analisis ini dilakukan deskripsi sistem yang sedang berjalan, definisi masalah dan kesempatan, serta usulan rekomendasi umum untuk memperbaiki, meningkatkan atau mengganti sistem yang sedang berjalan. Tujuan utama fase analisis adalah untuk mendokumentasikan kebutuhan bisnis (*business need*) dan persyaratan proses dari sistem baru (Fatta, 2007).

Ada lima aktivitas utama dalam fase analisis yaitu (Fatta, 2007):

1. Pengumpulan informasi
Langkah awal pada tahapan analisis adalah mengumpulkan informasi mengenai bagaimana proses-proses bisnis berjalan pada sistem lama.

Kemudian ditentukan pada titik-titik mana saja proses bisnis yang mengalami masalah dan dapat diselesaikan dengan sistem informasi. Pada tahap ini kelemahan-kelemahan pada sistem lama diidentifikasi dan diperbaiki dengan sistem baru.

2. Mendefinisikan sistem *requirement*

Setelah kelemahan sistem lama didapatkan, kemudian dilakukan pendefinisian apa saja yang sebenarnya dibutuhkan oleh sistem lama untuk mengatasi masalahnya. Inilah yang disebut sebagai persyaratan sistem (*system requirement*). Kebutuhan sistem ini nantinya dapat mengubah total keseluruhan proses bisnis pada sistem lama, ataupun hanya terjadi perubahan berupa penambahan beberapa prosedur baru.

3. Memprioritaskan kebutuhan

Saat dilakukan pendefinisian sistem bisa saja kita memperoleh kebutuhan yang sangat lengkap dan rumit. Ketersediaan waktu dan sumber daya untuk menyelesaikan keseluruhan kebutuhan bisa jadi tidak mencukupi. Oleh karena itu, pada tahap analisis ini analisis perlu memprioritaskan kebutuhan-kebutuhan yang dianggap kritis untuk diprioritaskan.

4. Menyusun dan mengevaluasi alternatif

Setelah menyusun dan memprioritaskan kebutuhan, analisis harus menyiapkan alternatif kedua jika susunan kebutuhan nantinya akan ditolak oleh klien.

5. Mengulas kebutuhan dengan pihak manajemen

Langkah terakhir dalam fase analisis adalah mengulas kebutuhan yang sudah ada dengan pihak klien, karena pihak klien adalah pihak yang paling mengetahui kebutuhan sistem mereka.

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 berorientasi objek (S. & Shalahuddin, 2013).

(a) MoSCoW

MoSCoW adalah teknik prioritisasi untuk membantu memahami dan mengelola prioritas. Prioritisasi dapat diterapkan pada persyaratan, tugas/*task*, produk, *use case*, kriteria penerimaan dan tes. Aturan prioritas MoSCoW ini terdiri dari *Must Have*(Mo), *Should Have*(S), *Could Have*(Co), dan *Won't Have this time*(W) (DSDM Consortium, 2015).

1. *Must Have*

Jika suatu persyaratan memiliki prioritas *Must Have*, maka tanpa persyaratan tersebut suatu solusi atas suatu permasalahan tidak dapat dikatakan layak. Untuk dapat mengetahui suatu persyaratan termasuk dalam prioritas *Must Have*, bisa menggunakan pertanyaan “Apa yang terjadi jika persyaratan tidak terpenuhi?”. Jika jawaban dari pertanyaan tersebut adalah “membatalkan proyek - tidak ada gunanya menerapkan solusi yang tidak memenuhi persyaratan ini” maka persyaratan tersebut termasuk dalam prioritas *Must Have* (DSDM Consortium, 2015).

2. *Should Have*

Should have digunakan ketika suatu persyaratan sistem merupakan persyaratan yang penting tetapi tidak vital. Untuk membedakan prioritas *Should Have* dan *Could Have* adalah dengan meninjau tingkat kegagalan yang disebabkan oleh persyaratan yang tidak terpenuhi, diukur dari nilai bisnis atau jumlah orang yang akan terkena dampaknya (DSDM Consortium, 2015).

3. *Could have*

Could have digunakan ketika suatu persyaratan sistem diinginkan untuk ada tetapi persyaratan tersebut kurang penting. Jika persyaratan yang berprioritas *could have* tidak dipenuhi maka akibat yang akan ditimbulkan pada proyek akan lebih kecil. Jika persyaratan berprioritas *should have* tidak dipenuhi maka akibat yang akan ditimbulkan pada proyek akan lebih besar. (DSDM Consortium, 2015).

4. *Won't Have this time*

Won't Have this time digunakan ketika suatu persyaratan disepakati untuk tidak dikerjakan pada suatu proyek pada jangka waktu ini. Persyaratan yang memiliki prioritas ini membantu memperjelas batasan/ruang lingkup proyek (DSDM Consortium, 2015).

2.2.4.2 Perancangan Sistem Informasi

Tahap desain / perancangan adalah tahapan mengubah kebutuhan yang masih berupa konsep menjadi sistem yang riil (Fatta, 2007). Desain 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, misalnya UML (*Unified Modeling Language*) (S. & Shalahuddin, 2013).

Dalam dimensi disiplin RUP analisis dan desain adalah disiplin ilmu untuk menunjukkan bagaimana *use case* dari sistem akan direalisasikan pada implementasi. Tujuan dari analisis dan desain adalah untuk mengubah persyaratan sistem ke dalam desain sistem, mengembangkan arsitektur untuk sistem, dan mengadaptasikan desain pada lingkungan implementasi yang cocok (IBM Corporation, 2004). IBM (*International Business Machine*) memiliki panduan dalam melakukan tahap analisis dan desain dengan menggunakan metode *Object Oriented Analysis and Design* (OOAD). Dalam memvisualisasikan model analisis dan desain IBM menggunakan UML 2.0. Berdasarkan panduan IBM dalam melakukan tahapan analisis dan perancangan dengan menggunakan metode OOAD, tahapan analisis dan perancangan yang dilakukan pada penelitian ini adalah :

1. Analisis arsitektur

Analisis arsitektur adalah aktivitas mendefinisikan arsitektur tingkat tinggi suatu proyek. Analisis arsitektur berfokus pada pola arsitektur yang akan

digunakan. Analisis arsitektur juga merupakan aktivitas awal mendefinisikan bagian dari sistem dan hubungan antar bagian tersebut dan juga mengorganisir bagian-bagian tersebut. Pada *use case* analisis arsitektur akan diperluas dengan mengidentifikasi kelas analisis dari *requirement* (IBM Corporation, 2004).

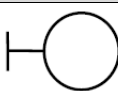
Pada tahap analisis arsitektur terdapat tahap identifikasi mekanisme analisis. Mekanisme analisis merupakan bagian dari mekanisme arsitektur. Mekanisme arsitektur terdiri dari tiga kategori yaitu (IBM Corporation, 2004):

- Mekanisme analisis. Tahap ini dilakukan dengan menyediakan perilaku khusus domain yang berkaitan dengan kelas atau komponen, atau sesuai dengan implementasi hubungan antar kelas dan/atau komponen. Contoh mekanisme analisis adalah *persistency*, distribusi, *message routing*, *security* dan lain-lain.
- Mekanisme desain. Tahap ini dilakukan dengan mengasumsikan detail dari lingkungan implementasi, tetapi tidak spesifik pada implementasi tertentu. Pada mekanisme desain beberapa teknologi dipilih seperti RDBMS vs. ODBMS.
- Mekanisme implementasi. Tahap ini dilakukan dengan menentukan implementasi yang tepat dari mekanisme, terikat pada teknologi tertentu, bahasa implementasi, vendor atau faktor lain. Pada mekanisme implementasi teknologi yang sangat spesifik dipilih, seperti Oracle vs. SYBASE.

2. Use case analysis

Pada tahap *use case analysis* dilakukan identifikasi kelas awal dari suatu sistem (kelas-kelas analisis). Kelas-kelas ini diidentifikasi dari *use case behaviour*. Jenis-jenis kelas analisis yang diidentifikasi dalam tahap ini adalah *boundary*, *controller* dan *entity*. Setiap tipe kelas ini akan didefinisikan dengan *stereotype* berbeda. Perbedaan ini akan digunakan selama tahap analisis dan akan menghilang pada tahap perancangan. Tabel 2.2 berikut ini adalah ikon yang digunakan pada tipe dari kelas analisis (IBM Corporation, 2004).

Tabel 2.2 Stereotype - ikon

Stereotype	Ikon
Boundary	
Entity	
Controller	

Sumber : IBM Corporation (2004)

Untuk menemukan kelas analisis ada tiga perspektif untuk mengidentifikasi kandidat kelas. Tiga perspektif tersebut adalah (IBM Corporation, 2004):

- *Boundary* antara sistem dan aktor.
- Informasi yang digunakan oleh sistem
- *Control logic* dari sistem

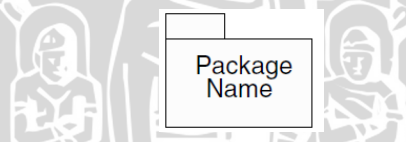
Setelah mengidentifikasi kelas-kelas analisis kemudian dilakukan identifikasi mekanisme analisis. Analisis mekanisme yang mungkin digunakan oleh kelas-kelas analisis diidentifikasi. Hal ini digunakan untuk mengetahui mekanisme analisis apa yang digunakan pada kelas-kelas analisis yang telah diidentifikasi sebelumnya (IBM Corporation, 2004).

Selain melakukan identifikasi kelas-kelas analisis dan mekanismenya, pada tahap ini juga dilakukan pemodelan interaksi kelas analisis. Pemodelan dapat dilakukan dengan menggunakan diagram interaksi (*sequence diagram* atau diagram komunikasi) (IBM Corporation, 2004).

3. Identifikasi elemen desain

Pada tahap identifikasi elemen desain kelas-kelas analisis disempurnakan ke dalam elemen desain. Elemen desain ini dapat berupa desain kelas, *packages* dan subsistem). Pada tahap ini beberapa kelas analisis mungkin dipecah, digabung, dihilangkan atau manipulasi lain (IBM Corporation, 2004).

Package adalah mekanisme untuk mengorganisir unsur ke dalam kelompok. *Package* merupakan model yang dapat berisi elemen model lain. Gambar 2.5 berikut ini merupakan notasi dalam menggambarkan *package*.



Gambar 2.5 Notasi Packages

4. *Use case design*

Use case design merupakan tahap memperbaiki realisasi *use case* yang telah didefinisikan pada tahap *use case analysis*. Sebuah realisasi *use case* merupakan kemungkinan realisasi dari sebuah *use case*. Realisasi *use case* dapat dipresentasikan dengan dengan satu set diagram. Contoh dari realisasi *use case* adalah diagram interaksi dan diagram kelas (IBM Corporation, 2004). Pada penelitian ini tahap *use case design* dilakukan dengan membuat pemodelan interaksi dengan *sequence diagram*.

5. Perancangan kelas

Perancangan kelas adalah tahap yang berfokus pada pendetailan kelas. Tahap perancangan kelas terdiri dari membuat inisialisasi kelas desain, mendefinisikan operasi, mendefinisikan *method/fungsi*, mendefinisikan *states*, mendefinisikan atribut, mendefinisikan dependensi, mendefinisikan asosiasi, mendefinisikan struktur internal, mendefinisikan generalisasi, *resolve use case collisions*, menangani persyaratan non fungsional secara

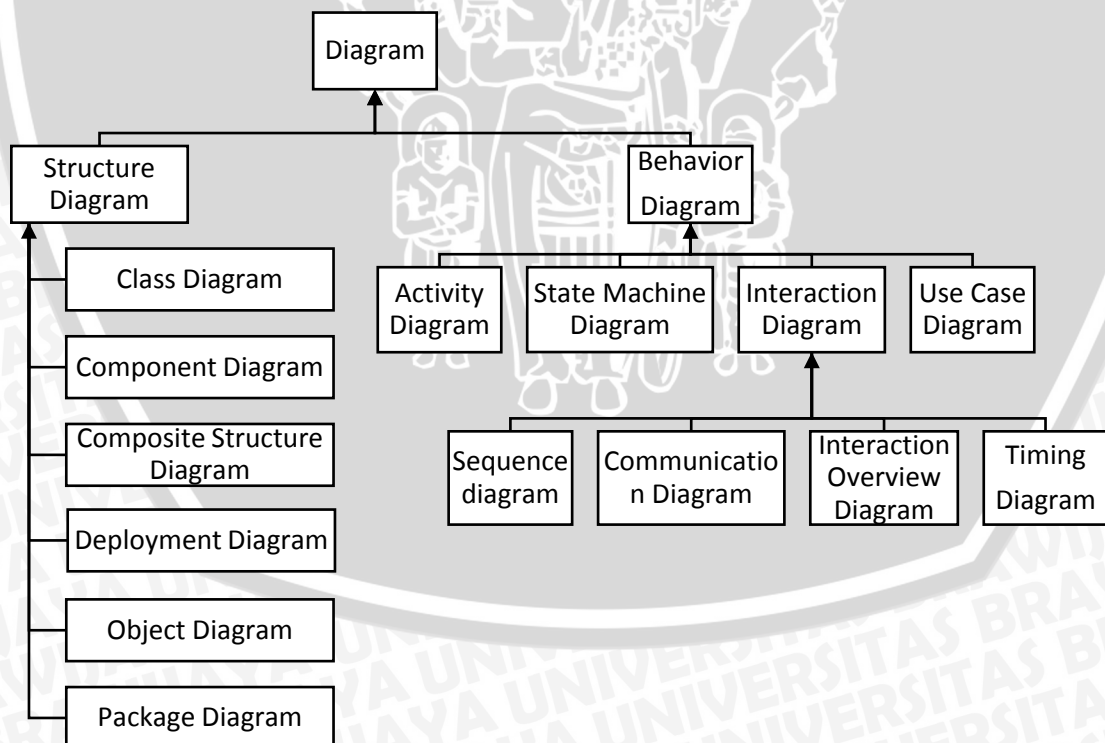
umum (IBM Corporation, 2004). Pada penelitian ini yang dilakukan pada perancangan kelas diantaranya adalah mengidentifikasi *method*, atribut dan juga hubungan antar kelas.

6. Perancangan *database*

Perancangan *database* adalah tahap yang bertujuan untuk memastikan bahwa data yang *persistent* disimpan secara konsisten dan efisien, dan untuk menentukan perilaku yang harus diimplementasikan di sistem (IBM Corporation, 2004). Pada penelitian ini perancangan *database* dilakukan dengan merancang model data berdasarkan diagram kelas model.

(b) UML

UML adalah bahasa standar yang digunakan dalam menjelaskan 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. 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. Dalam memodelkan suatu sistem UML menyediakan standar pada notasi dan diagram (Hermawan, 2015).



Gambar 2.6 Diagram UML

Sumber : Fowler (2003)

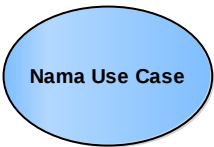
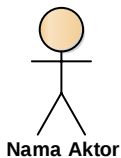
UML versi 2.0 memiliki 13 diagram dalam memodelkan/mendeskripsikan struktur atau tingkah laku sistem. Gambar 2.6 merupakan taksonomi dari 13 diagram UML (Fowler, 2003).

(i) Use Case

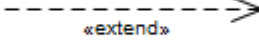
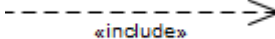
Diagram *use case* merupakan diagram dalam UML yang mendefinisikan kebutuhan yang harus dipenuhi oleh sistem. Diagram *use case* menggambarkan hubungan antara *user*/pengguna dan fungsi dari sistem tetapi tidak membahas bagaimana rincian spesifik pelaksanaan fungsi sistem (Seidl, et al., 2012). Gambar 2.7 merupakan contoh diagram *use case*.

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).

Tabel 2.3 Simbol-Simbol Use Case Diagram

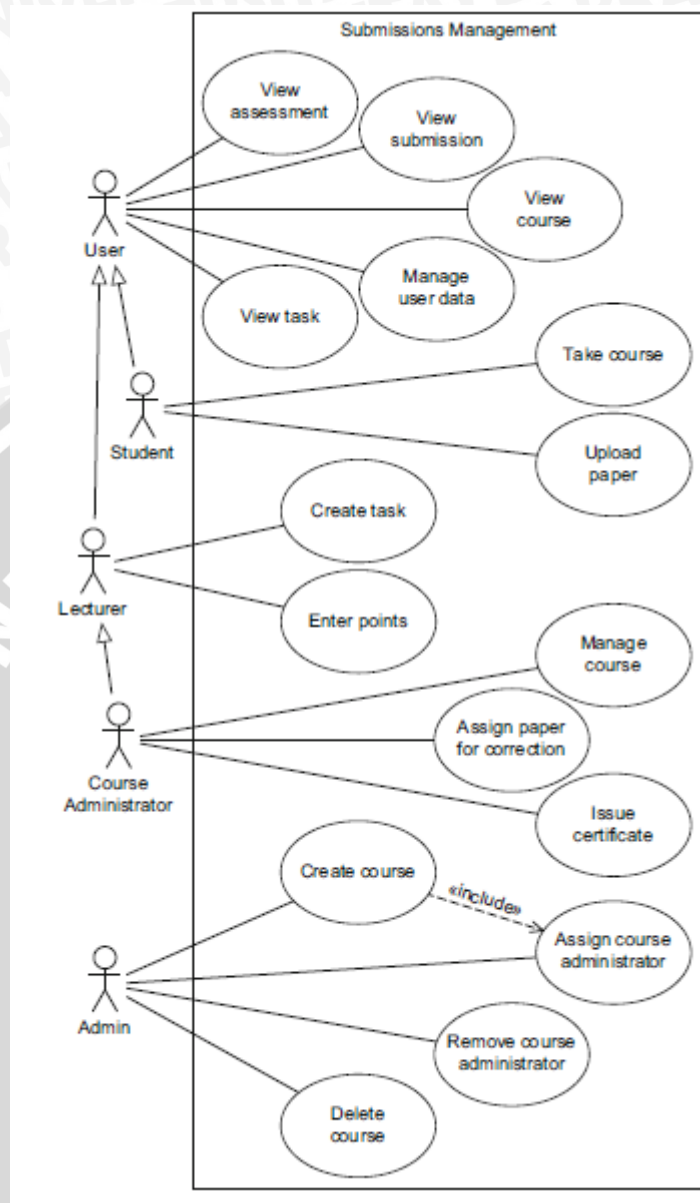
Nama	Notasi	Deskripsi
Use case		Notasi yang merepresentasikan persyaratan fungsional sistem. <i>Use case</i> merupakan sebuah deskripsi dari satu set urutan tindakan yang dilakukan sistem yang dapat digunakan oleh aktor. <i>Use case</i> mendeskripsikan apa yang dilakukan sistem tetapi tidak mendeskripsikan bagaimana sistem melakukannya.
Aktor		Notasi yang merepresentasikan seperangkat peran pengguna dari <i>use case</i> (manusia, perangkat keras, atau sistem lain) yang berinteraksi dengan sistem. Aktor berada di luar sistem tetapi masih di sekitar sistem.
Asosiasi / association		Aktor dan <i>use case</i> dapat dihubungkan dengan asosiasi. Asosiasi antara <i>use case</i> dan aktor

Tabel 2.3 Simbol-Simbol *Use Case Diagram* (lanjutan)

Nama	Notasi	Deskripsi
		menunjukkan bahwa aktor dan <i>use case</i> saling berkomunikasi, masing-masing (<i>use case</i> dan aktor) mungkin mengirim dan menerima pesan.
Extensi / <i>extend</i>		Notasi yang menggambarkan hubungan antar <i>use case</i> , <i>base use case</i> secara implisit menggabungkan perilaku/ <i>behaviour</i> dari <i>use case</i> lain pada lokasi yang ditentukan secara tidak langsung oleh <i>use case</i> yang meng- <i>extend</i> . <i>Base use case</i> dapat berdiri sendiri, tapi dalam kondisi tertentu dapat diperluas oleh <i>use case</i> lain. <i>Extend</i> digunakan untuk memodelkan bagian dari <i>use case</i> sebagai perilaku sistem yang bersifat opsional.
Generalisasi / <i>generalization</i>		Generalisasi menggambarkan hubungan umum khusus di mana elemen khusus (anak/ <i>child</i>) dibangun dari spesifikasi yang diturunkan dari <i>parent</i> .
<i>Include</i>		Notasi yang menggambarkan hubungan antar <i>use case</i> , <i>base use case</i> secara eksplisit menggabungkan perilaku/ <i>behaviour</i> use case lain pada lokasi yang ditentukan <i>base use case</i> . Use yang di <i>include</i> tidak berdiri sendiri, melainkan hanya dipakai sebagai bagian dari <i>use case</i> yang lebih besar (<i>use case</i> yang meng- <i>include</i>). <i>Include</i> digunakan untuk menghindari mendeskripsikan <i>flow of events</i> yang sama beberapa kali.

Sumber : Booch, et al. (2005)

Gambar 2.7 berikut ini merupakan contoh diagram *use case*.



Gambar 2.7 Use Case Diagram Submission System

Sumber: Seidl, et al. (2012)

Sebelum melakukan pemodelan *use case* perlu membuat dasar yang kuat. Dasar/pondasi sebelum melakukan pemodelan *use case* ini ada dua yaitu (Bittner & Spence, 2002):

1. Pemahaman pemangku kepentingan dan pengguna
Memahami pemangku kepentingan sangat penting karena pemangku kepentingan merupakan sumber utama persyaratan. Berikut ini adalah hal yang penting untuk didefinisikan dalam

memahami pemangku kepentingan dan pengguna (Bittner & Spence, 2002) :

- a. Tipe pemangku kepentingan : semua jenis pemangku kepentingan terpengaruh oleh proyek dan produk yang dihasilkan.
 - b. Tipe pengguna : Karakteristik dan kemampuan pengguna sistem.
2. Pembentukan *shared vision* untuk produk

Untuk membangun *shared vision* suatu proyek pengembangan perangkat lunak sangat penting untuk mendefinisikan hal-hal berikut ini (Bittner & Spence, 2002):

- a. Pernyataan masalah: ringkasan solusi dari masalah yang dipecahkan, berfokus pada dampak dari masalah serta manfaat dari setiap solusi yang sukses. Tabel 2.4 berikut ini merupakan *template* dalam mendefinisikan pernyataan masalah.

Tabel 2.4 Template pernyataan masaaah

<i>The problem of</i>	<i>[describe the problem]</i>
<i>Affects</i>	<i>[the stakeholders affected by the problem]</i>
<i>The impact of which is</i>	<i>[what is the impact of the problem?]</i>
<i>A successful solution would</i>	<i>[list some key benefits of a successful solution]</i>

Sumber : Bittner & Spence (2002)

- b. Kebutuhan pemangku kepentingan: kebutuhan dari pemangku kepentingan yang disajikan dengan solusi atas masalah yang mempengaruhi pemangku kepentingan secara individu. Kebutuhan merupakan sesuatu yang berasal dari pemangku kepentingan mengenai apa yang perlu dilakukan serta masalah yang harus diselesaikan oleh sistem. Dengan mengidentifikasi kebutuhan pemangku kepentingan kemungkinan kita akan memahami bagaimana dan sejauh mana masalah mempengaruhi berbagai jenis pemangku kepentingan (Bittner & Spence, 2002).
- c. Fitur, *constraint*, dan persyaratan tingkat tinggi

Fitur merupakan *high-level capabilities*/kemampuan tingkat tinggi (layanan atau kualitas) dari sistem yang diperlukan untuk memberikan manfaat bagi pengguna dan membantu memenuhi kebutuhan pemangku kepentingan dan kebutuhan pengguna (Bittner & Spence, 2002).

Persyaratan perangkat lunak merupakan pernyataan dari kondisi dan kemampuan yang harus dipenuhi oleh sistem.

Persyaratan menggambarkan apa yang harus dilakukan oleh sistem. Persyaratan terbagi menjadi dua kategori yaitu persyaratan fungsional dan persyaratan non fungsional.

Persyaratan fungsional merupakan tindakan/hal/perilaku yang harus mampu dilakukan oleh sistem. Sedangkan persyaratan non fungsional merupakan spesifikasi kualitas yang harus dimiliki oleh sistem, seperti *usability*, *reliability*, *performance*, dan *supportability*.

- d. Gambaran produk/pernyataan posisi produk: ringkasan aspek lain dari produk yang tidak langsung ditangkap oleh persyaratan tingkat tinggi. Pernyataan atau deskripsi ini penting karena akan memberikan penjelasan secara umum kepada setiap orang apa yang dilakukan oleh sistem.

Tabel 2.5 Template pernyataan posisi produk

<i>For</i>	<i>(target customer)</i>
<i>Who</i>	<i>(statement of the need or opportunity)</i>
<i>The</i>	<i>(product name) is a (product category)</i>
<i>That</i>	<i>(statement of key benefit, that is, compelling reason to buy)</i>
<i>Unlike</i>	<i>(primary competitive alternative)</i>
<i>Our product</i>	<i>(statement of primary differentiation)</i>

Sumber : Sumber : Bittner & Spence (2002)

(ii) Activity Diagram

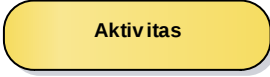
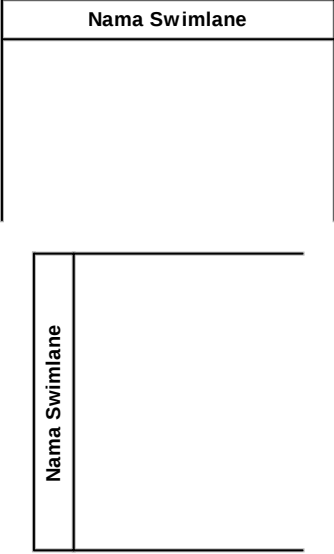
Diagram aktivitas atau *activity diagram* merupakan diagram yang menggambarkan aliran kerja (*work flow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas yang dapat dilakukan oleh sistem bukan menggambarkan apa yang dilakukan oleh aktor (S. & Shalahuddin, 2013).

Berikut ini adalah notasi-notasi / simbol-simbol yang digunakan dalam membuat *activity diagram*:

Tabel 2.6 Simbol-Simbol Activity Diagram

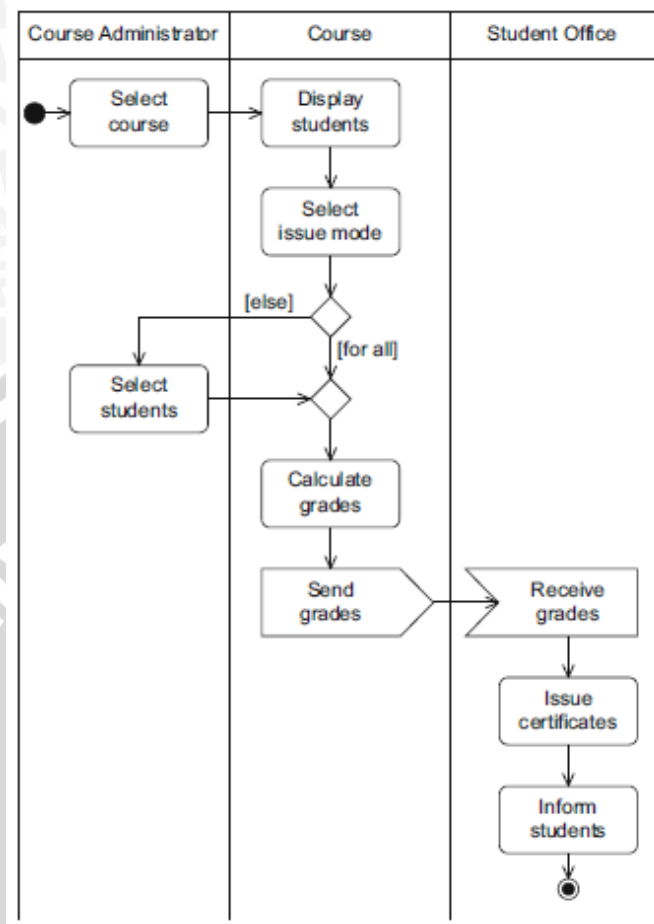
Nama	Notasi	Deskripsi
Status Awal	 Mulai	Notasi yang menggambarkan status awal dari aktivitas sistem. Sebuah diagram aktivitas memiliki sebuah status awal.

Tabel 2.6 Simbol-Simbol *Activity Diagram* (lanjutan)

Nama	Notasi	Deskripsi
Aktivitas		Notasi yang menggambarkan aktivitas yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan /decision		Notasi yang menggambarkan asosiasi percabangan di mana jika ada lebih dari satu pilihan aktivitas.
Penggabungan / join		Notasi yang menggambarkan asosiasi penggabungan. Notasi digunakan ketika ada lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir	 Selesai	Notasi yang menggambarkan status akhir yang dilakukan sistem. Sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane		Notasi yang digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: S. & Shalahuddin (2013)

Contoh activity diagram :



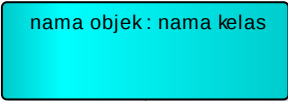
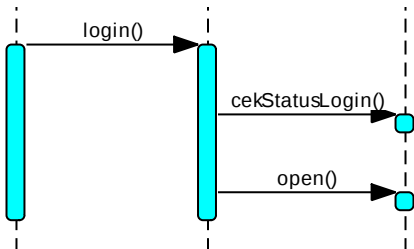
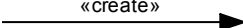
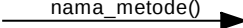
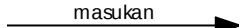
Gambar 2.8 Activity Diagram Issue certificate

Sumber: Seidl, et al. (2012)

(iii) Sequence Diagram

Sequence diagram adalah diagram yang menjelaskan secara detail mengenai urutan proses yang dilakukan di dalam sistem dalam mencapai tujuan *use case*. Detail proses yang dijelaskan di dalam *sequence diagram* diantaranya adalah interaksi yang terjadi antar kelas, operasi yang terlibat, urutan operasi, dan informasi yang diperlukan oleh masing-masing operasi. *Sequence diagram* biasanya tersusun atas objek, *interaction*, dan *message* (Hermawan, 2015).

Tabel 2.7 Simbol-Simbol *Sequence Diagram*

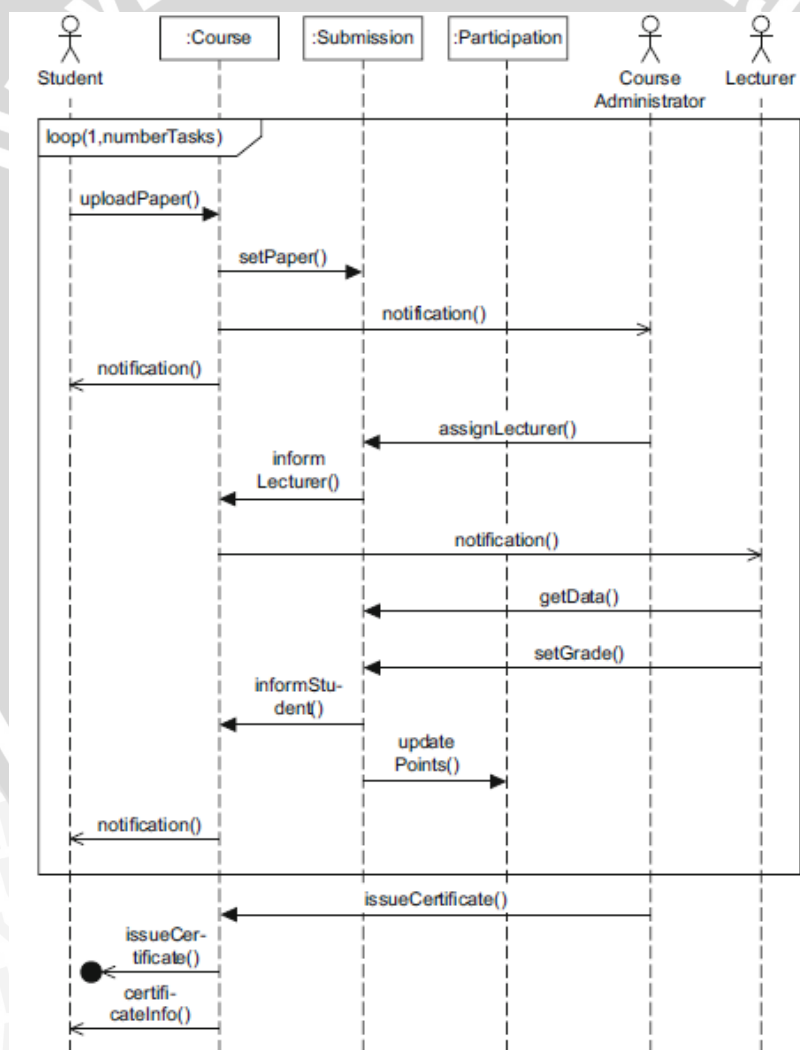
Nama	Notasi	Deskripsi
Aktor	 Nama Aktor	Notasi yang menggambarkan orang, proses, atau sistem lain yang berinteraksi dengan sistem. Notasi ini berada di luar sistem.
<i>Lifeline</i>		Notasi yang menyatakan kehidupan suatu objek.
Objek	 nama objek : nama kelas	Notasi yang menyatakan objek.
Waktu Aktif		Notasi yang menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misal :  Maka cekStatusLogin() dan open() dilakukan di dalam <i>method</i> login().
Pesan tipe <i>create</i>		Notasi yang menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i>		Notasi yang menyatakan suatu objek memanggil operasi / <i>method</i> yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi / <i>method</i> . Operasi / <i>method</i> harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe <i>send</i>		Notasi yang menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang menerima pesan.

Tabel 2.7 Simbol-Simbol *Sequence Diagram* (lanjutan)

Nama	Notasi	Deskripsi
Pesan tipe <i>return</i>		Notasi yang menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i>		Notasi yang menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : S. & Shalahuddin (2013)

Gambar 2.9 berikut ini contoh *sequence diagram* :



Gambar 2.9 *Sequence Diagram*

Sumber: Seidl, et al. (2012)

(iv) Class Diagram

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

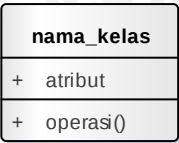
Suatu atribut dan metoda suatu kelas dapat memiliki salah satu dari sifat berikut ini (Yasin, 2012):

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh *class* manapun.

Hubungan antar *Class* (Dharwiyanti & Wahono, 2003):

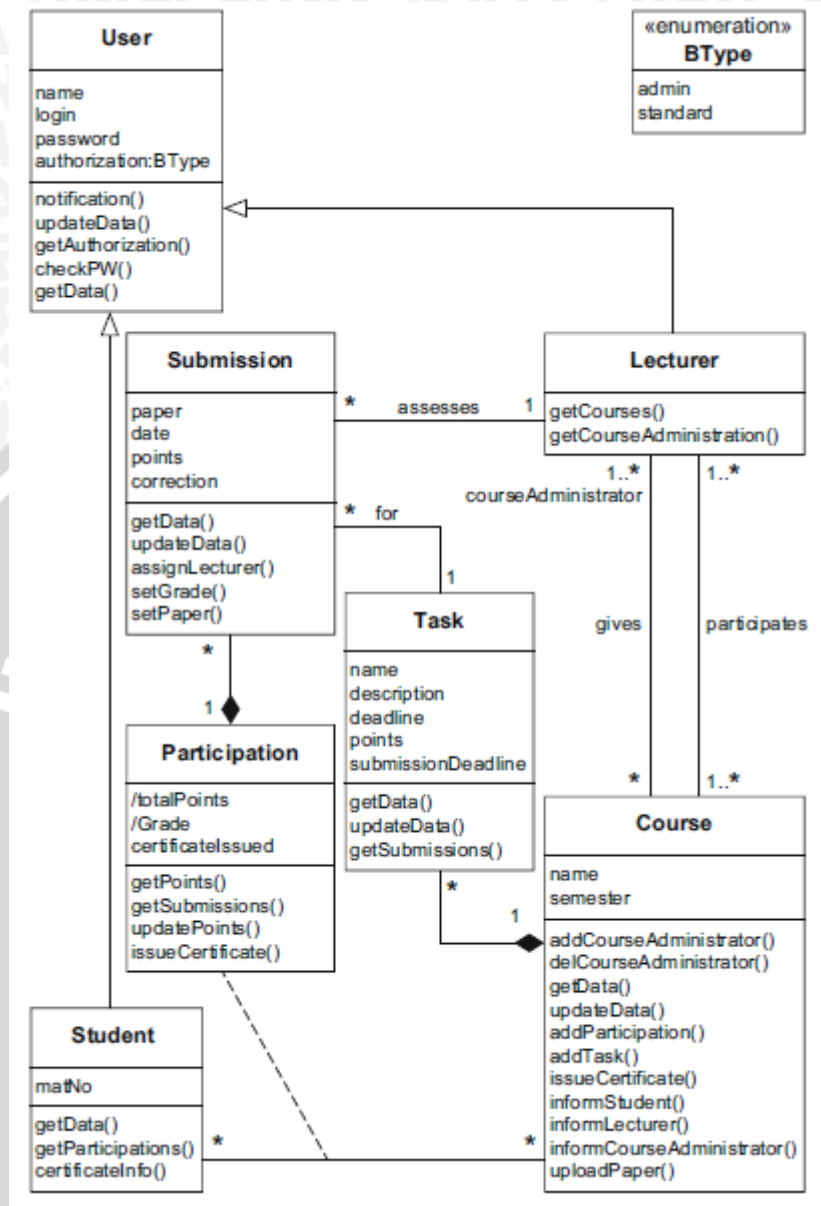
1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
3. Pewarisan, yaitu hubungan hierarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang *di-passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

Tabel 2.8 Simbol-Simbol *Class Diagram*

Nama	Notasi	Deskripsi
<i>Class</i> / kelas		Notasi yang menggambarkan kelas pada struktur sistem.
<i>Interface</i> / antarmuka		Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<i>Association</i> / asosiasi		Notasi yang menggambarkan relasi antar kelas dengan makna umum. Asosiasi biasanya disertai dengan <i>multiplicity</i> .
<i>Directed association</i> / asosiasi berarah		Notasi yang menggambarkan relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain.
Generalisasi		Notasi yang menggambarkan relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
<i>Dependency</i> / kebergantungan		Notasi yang menggambarkan relasi antar kelas dengan makna kebergantungan antar kelas.
<i>Aggregation</i> / agregasi		Notasi yang menggambarkan relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber : S. & Shalahuddin (2013)

Contoh class diagram :



Gambar 2.10 Class Diagram Submission System

Sumber: Seidl, et al. (2012)

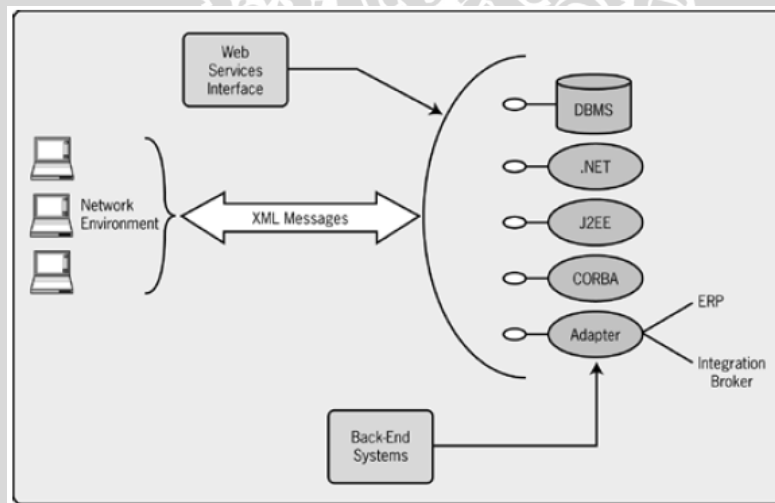
(c) Web Service

Web service adalah layanan antarmuka yang diakses melalui jaringan untuk fungsionalitas aplikasi yang dibangun dengan standar teknologi internet. (Tidwell, et al., 2001). Berikut ini merupakan karakteristik web service (Yasin, 2012):

1. Merupakan *application logic* yang dapat diakses dan dipublikasikan menggunakan standar Internet (TCP/ IP, HTTP, web).

2. Dideskripsikan dalam format XML.
3. Diidentifikasi dengan *Universal Resources Identifier* (URI)
4. Bersifat *Loosely coupled, self-contained*, modular dan terbuka (*nonproprietary*)
5. Digunakan untuk mendukung interoperabilitas interaksi *machine-to-machine* melalui jaringan Internet/Intranet.

Web service merupakan aplikasi *Extensible Markup Language* (XML) yang dipetakan ke program, objek, atau *database* atau ke fungsi bisnis yang komprehensif. Menggunakan dokumen XML yang dibuat dalam bentuk pesan, program mengirim *request* melalui jaringan dan menerima balasan / *reply* juga dalam bentuk dokumen XML. Standar *web service* mendefinisikan format dari pesan / *message*, menentukan antarmuka pesan yang dikirim, menjelaskan mekanisme untuk mempublikasikan dan untuk menemukan antarmuka *web service*. *Web service* mengubah dokumen XML ke dalam dan keluar dari sistem. Antarmuka *web service* menerima pesan XML standar dari lingkungan jaringan, mengubah data XML ke dalam format yang dipahami oleh *back-end software* sistem tertentu dan mengembalikan pesan balasan (Newcomer, 2002). Gambar 2.11 merupakan gambaran komunikasi *web service* dengan *back-end* sistem.



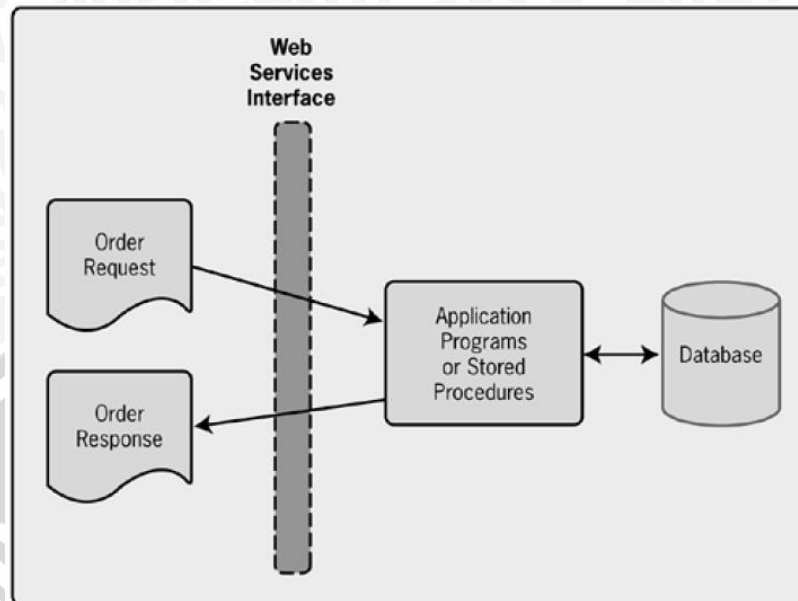
Gambar 2.11 Antarmuka *Web Service* dengan *back-end* sistem

Sumber : Newcomer (2002)

Teknologi *web service* umumnya mencakup dua jenis pola interaksi aplikasi :

1. *Remote procedure call (online)*
 Pada interaksi berorientasi *remote procedure call* (RPC), *request* pada *web service* berbentuk *method* atau *procedure call* dengan parameter input atau output. Berbeda dengan interaksi berorientasi dokumen, interaksi berorientasi RPC mengirim dokumen dalam format khusus

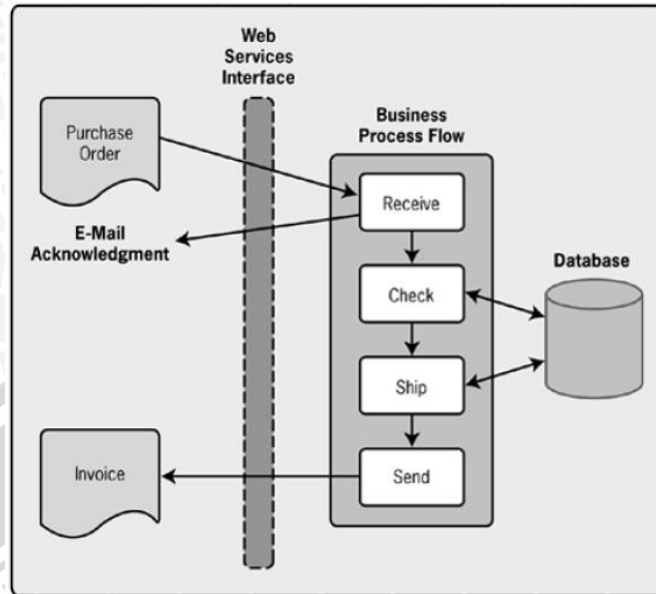
dipetakan ke satu program atau *database* (Newcomer, 2002), seperti pada Gambar 2.12.



Gambar 2.12 Web service request/response permintaan yang interaktif

2. Document oriented (batch)

Pada interaksi berorientasi dokumen, *request* pada *web service* berbentuk dokumen XML lengkap yang dimaksudkan untuk proses secara keseluruhan. Misalnya, *web service* untuk proses order pembelian lengkap, seperti pada Gambar 2.13. Produsen biasanya akan mengirim email atau lainnya kepada pengecer untuk menunjukkan bahwa order diterima dan akan diproses sesuai dengan aliran yang telah ditetapkan. Aliran ini termasuk langkah-langkah memeriksa *database* untuk pesanan sebelumnya dari pengecer yang sama untuk memastikan tidak melebihi batas kredit atau kapasitas atau penjadwalan pengiriman pesanan. Dalam aliran proses nyata tentu ada banyak langkah yang mungkin dilakukan sebelum pesanan dikirim dan faktur dikirim keluar, tetapi contoh pada Gambar 2.13 hanya menunjukkan langkah terakhir yaitu mengirim faktur XML kepada distributor untuk pembayaran setelah pesanan dikirim dan diterima (Newcomer, 2002).



Gambar 2.13 Web service proses order pembelian barang

Web service memerlukan beberapa teknologi berbasis XML untuk membawa dan untuk mengubah data yang masuk dan keluar dari program dan *database*. Teknologi berbasis XML tersebut adalah :

1. XML, sebagai fondasi dasar *web service* yang menyediakan bahasa untuk mendefinisikan data dan bagaimana memproses data tersebut.
2. WSDL (*Web Services Description Language*), sebuah teknologi berbasis XML yang mendefinisikan antarmuka, data dan jenis pesan, pola interaksi, serta pemetaan protokol *web service*.
3. SOAP (*Simple Object Access Protocol*), kumpulan teknologi berbasis XML yang mendefinisikan sebuah amplop untuk komunikasi pada *web service* dan menyediakan format serialisasi untuk transmisi dokumen XML atas jaringan dan konvendi untuk mewakili interaksi RPC.
4. UDDI (*Universal Description, Discovery, and Integration*), mekanisme register dan *discovery web service*, digunakan untuk memilah dan mengelompokkan informasi bisnis dan untuk mengambil *pointer* ke antarmuka *web service* (Newcomer, 2002).

(i) XML

XML mencakup beberapa komponen bahasa, struktur XML yang umum digunakan memiliki tiga komponen XML yaitu elemen, atribut dan entitas (Coyle, 2002).

1. Elemen

Elemen adalah sarana utama untuk menggambarkan data dalam XML. Aturan dalam menyusun elemen fleksibel, memungkinkan kombinasi yang berbeda antara teks, atribut, dan elemen lainnya. Namun, ada tiga cara elemen digunakan dalam dokumen XML.

a. Konten sederhana

Teks atau data lain ada di antara awal dan akhir *tag*. *Tag* awal memiliki nama yang sama dengan *tag* akhir kecuali *tag* akhir dimulai dengan garis miring/slash. Berikut ini adalah contoh konten sederhana yang mengandung *tag* awal, isi dan tak akhir.

```
<author>Stephen Hawking</author>
```

b. Elemen sebagai kontainer untuk elemen lain

Sebuah elemen dapat memuat elemen lainnya, menyediakan struktur data hierarkis. Berikut ini contoh elemen *book* yang berisi elemen *author* dan *title*.

```
<book>
  <author>Stephen Hawking</author>
  <title>A brief history of time</title>
</book>
```

c. Elemen kosong sebagai kontainer untuk atribut

Ketika suatu elemen tidak memiliki konten tetapi hanya memiliki atribut, ada cara singkat untuk menulis elemen. Elemen ditulis dengan disertai garis miring, contoh :

```
<book/>
```

Penggunaan paling umum dari elemen kosong adalah untuk menyimpan data atribut. Contoh :

```
<book title="A brief history of time"
author="Stephen Hawking"/>
```

Secara teknis sebuah elemen dapat berisi elemen lain, teks dan atribut, tetapi ini akan sulit dibaca dan dipahami atau memiliki *readability* rendah (Coyle, 2002). Contoh:

```
<book isbn="0-102-9393-3">
  A brief history of time
  <author>Stephen Hawking</author>
</book>
```

2. Atribut

Atribut memberikan informasi tambahan mengenai elemen. Dalam HTML, atribut digunakan untuk menentukan nama *file* gambar ketika *loading* sebuah dokumen HTML.

```

```

Atribut sering digunakan untuk menunjukkan informasi yang bukan merupakan bagian dari data yang dideskripsikan dalam elemen. Atribut sering digunakan untuk menjelaskan sesuatu tentang data itu sendiri. Misalnya, dalam XML berikut penggunaan atribut mungkin memberitahukan bahwa penggunaan data tidak *required* atau opsional (Coyle, 2002).

```
<file use="optional">computer.gif</file>
```

3. Entitas

Entitas digunakan untuk mengganti satu *string* untuk yang lain dalam dokumen XML. Sebagai contoh, jika sebuah frase "*XML and the Data Revolution*" sering diulang dalam dokumen, kita dapat

menentukan *shortcut* entitas deklarasi dari DTD (*Document Type Definition*) (Coyle, 2002).

```
<!ENTITY xdr " XML and the Data Revolution ">
```

Kemudian ketikan ingin menggunakan frase penuh kita menggunakan &xdr dan itu akan diganti dalam dokumen XML. Menggunakan entitas dapat membantu menghindari kesalahan ejaan ketikan mengetik hal yang sama berulang-ulang (Coyle, 2002).

XML mengadopsi lima *predefined entities* dari dunia HTML. *Ampersand* (&), lebih besar dari (>), kurang dari (<), *double-quote* ("), dan apostrof (') karakter yang diwakili dalam dokumen XML sebagai "&", "<", ">", """, dan "'" (Coyle, 2002).

Jika entitas panjang, kita dapat menyimpan informasi secara terpisah dalam *file* lain. Ini dapat dilakukan dengan suatu entitas referensi eksternal, menggunakan kata kunci SYSTEM dalam XML di antara nama entitas dan URL dari *file* (Coyle, 2002).

```
<!ENTITY text SYSTEM "http://my.url.here">
```

Parameter entitas berguna untuk membuat *string* substitusi dalam dokumen XML, hal ini sering berguna untuk menentukan *shortcut* dalam DTD untuk membuat tulisan DTD lebih mudah. Sebuah parameter entitas didefinisikan dengan memasukkan tanda persen sebelum nama entitas. Setelah ditentukan, parameter entitas dapat digantikan dengan nama parameter di antara tanda persen dan titik koma (Coyle, 2002).

(ii) SOAP

SOAP adalah suatu protokol berbasis XML yang digunakan untuk kebutuhan pertukaran informasi dalam suatu sistem terdistribusi dan terdesentralisasi. Dalam proses pertukaran pesan SOAP menggunakan protokol yang bersifat independen terhadap platform, model pemrograman, dan *transport protocol*. Pesan SOAP dikirimkan melalui HTTP, SMTP maupun FTP. Pesan SOAP berbentuk sekumpulan XML *schema* yang mendefinisikan format untuk mentransmisikan pesan XML melalui jaringan, termasuk tipe data dan cara menstrukturkan pesan secara tepat sehingga dapat mudah dipahami oleh server atau *end-point* lainnya. Pesan SOAP terdiri dari 3 bagian, yaitu (Yasin, 2012):

1. *Envelope*, suatu kerangka yang mendefinisikan apa yang ada dalam pesan dan bagaimana pesan harus diproses serta menunjukkan resipien dari pesan tersebut.
2. *Header*, berisi informasi yang berhubungan dengan keamanan dan *routing*. Keberadaan *header* dalam SOAP bersifat opsional.
3. *Body*, berisi data yang berhubungan dengan aplikasi tertentu yang sedang dipertukarkan. Data di-'mark-up' sebagai XML dan

dimasukkan dalam format yang spesifik dan didefinisikan dalam XML *schema*.

(iii) WSDL

WSDL merupakan bahasa standar yang menyediakan mekanisme untuk mendeskripsikan *service* yang disediakan oleh sistem (*web service*), lokasi keberadaan *service* dan bagaimana cara memperolehnya secara terstruktur dalam format XML. *Service* dideskripsikan sebagai koleksi dari *entry-point* atau *port* komunikasi. WSDL mendeskripsikan *service* dengan menggunakan elemen sebagai berikut (Yasin, 2012):

1. *Type*, tipe data yang digunakan sebagai argumen dan *return type*.
2. *Message*, merepresentasikan definisi data yang ditransmisikan.
3. *Port type*, sekumpulan operasi yang didukung oleh satu atau lebih *endpoint*.
4. *Binding*, mendefinisikan protokol dan format pertukaran data untuk operasi yang didefinisikan oleh *Port type*.
5. *Port*, mendefinisikan *end-point* yang digunakan untuk *binding*.
6. *Service*, koleksi *endpoint* yang berkaitan yang disediakan oleh *web service*.
7. *Operation*, mendefinisikan kemampuan yang didukung oleh *service* tertentu.

(iv) UDDI

UDDI merupakan sekumpulan spesifikasi yang menunjukkan *registry* informasi mengenai *web service*. UDDI menyediakan mekanisme untuk mempublikasikan informasi mengenai bisnis dan *service* pada suatu lokasi (*repository*) yang dikelola secara terpusat dan melakukan *query* mengenai informasi tersebut secara dinamis dan programatis. *Directory* pada UDDI bertindak seperti 'Yellow Pages' di mana *service* dikategorikan sesuai tujuan utamanya. *Directory* UDDI terdiri dari tiga bagian, yaitu:

1. *White pages*, menyediakan informasi rinci mengenai organisasi yang menawarkan *service*.
2. *Yellow pages*, mencakup pengkategorian jenis industri berdasarkan standar *taxonomy* industri.
3. *Green pages*, mendeskripsikan *interface* dan kebutuhan untuk memperoleh *service*, seperti *return type*.

UDDI merupakan *file XML schema* yang mendefinisikan struktur data mengenai karakteristik bisnis dan *service*. Deskripsi *service* didefinisikan menggunakan dokumen *Type Model (tModel)*. Secara umum UDDI berisi informasi mengenai siapa yang menyediakan *service* (*bussinessEntity*), *service* apa yang disediakan (*bussinessService*), di mana lokasi *service* tersedia (*bindingTemplate*),

referensi mengenai informasi bagaimana *service* tersebut diperoleh (tModel).

2.2.5 Evaluasi

Evaluasi merupakan tahap menilai perancangan dan menguji sistem untuk memastikan bahwa perancangan dan sistem yang dibuat sesuai dengan yang diharapkan dan memenuhi kebutuhan pengguna. Evaluasi memiliki tiga tujuan utama yaitu untuk menilai sejauh mana sebuah fungsi sistem dapat diakses, untuk menilai interaksi pengalaman pengguna dan untuk mengidentifikasi masalah yang spesifik dari sistem. Fungsional sistem merupakan hal yang penting dan harus sesuai dengan persyaratan pengguna. Dengan kata lain, desain atau perancangan sistem harus memungkinkan pengguna melakukan tugasnya dengan lebih mudah. Desain sistem tidak hanya sebatas membuat fungsional sistem, tetapi juga harus memastikan dengan jelas fungsional sistem tersebut dapat digunakan pengguna untuk melakukan tugas-tugas mereka (Dix, et al., 2004).

2.2.5.1 Verifikasi dan Validasi

Kebanyakan pengembang perangkat lunak memiliki pengalaman frustrasi dikarenakan kesulitan untuk mengimplementasikan kebutuhan pengguna yang rancu dan tidak menyeluruh. Jika penembang tidak memperoleh informasi yang mereka butuhkan mengenai kejelasan spesifikasi dari pengembangan perangkat lunak yang mereka butuhkan, maka pengembang akan membuat interpretasi sendiri. Karena itu dibutuhkan tahapan verifikasi dan validasi terhadap kebutuhan dalam penentuan spesifikasi perangkat lunak (Siahaan, 2012).

Verifikasi dan validasi merupakan proses melakukan pengecekan dan analisis untuk memastikan bahwa perangkat lunak sesuai dengan spesifikasi dan memenuhi kebutuhan pelanggan. Verifikasi dan validasi dimulai dengan tinjauan persyaratan, dilanjutkan tinjauan perancangan dan *code inspection* untuk pengujian produk (Agarwal, et al., 2010).

Verifikasi kebutuhan perangkat lunak adalah tahapan dari rekayasa perangkat lunak untuk memastikan produk yang dihasilkan dari aktivitas pengembang sesuai dengan spesifikasi yang ditentukan atau bisa disebut "*doing the thing right*". Validasi kebutuhan perangkat lunak adalah tahapan untuk memberikan penilaian produk sesuai dan memuaskan keinginan dari pemangku kepentingan atau "*doing the right thing*" (Siahaan, 2012). Salah satu teknik validasi adalah pembuatan prototipe. Teknik ini merupakan teknik dengan membangun prototipe berdasarkan pada spesifikasi kebutuhan fungsional, seperti interpretasi pengembang atas kebutuhan-kebutuhan sistem. Prototipe ini biasanya hanya diimplementasikan pada sebagian kebutuhan fungsional. Dalam teknik ini pengguna harus mengerti bahwa apa yang mereka observasi hanya sebuah prototipe bukan sistem akhir (Simarmata, 2010).

2.2.5.2 Review

Review atau tinjauan merupakan sebuah pertemuan menyajikan elemen perangkat lunak kepada personel proyek, manajer, pengguna, pelanggan atau

pihak lain yang berkepentingan untuk memberikan komentar atau persetujuan terhadap perangkat lunak tersebut. Tujuan *review* adalah untuk menemukan kesalahan dalam fase analisis, desain, *coding*, pengujian atau implementasi. *Review* memiliki dua tipe yaitu *informal technical review* dan *formal technical review*. *Informal technical review* dilakukan dengan sebuah pertemuan informal dan pengecekan secara informal. Sedangkan *formal technical review* merupakan aktivitas menjamin kualitas perangkat lunak dengan berbagai pendekatan, seperti *walkthrough*, *inspection*, dan lain-lain (Agarwal, et al., 2010).

2.2.5.3 Cognitive Walkthrough

Walkthrough merupakan tinjauan (*review*) secara detail mengenai urutan tindakan. Pada *cognitive walkthrough* urutan tindakan yang dimaksud adalah langkah-langkah yang dilakukan pengguna dalam menyelesaikan beberapa tugas. Evaluator kemudian melihat urutan tindakan untuk memeriksa masalah *usability*. Fokus utama *cognitive walkthrough* adalah melihat seberapa mudah suatu sistem untuk dipelajari. Dalam melakukan *cognitive walkthrough* diperlukan empat hal yaitu (Dix, et al., 2004):

1. Spesifikasi atau prototipe dari sistem.
2. Deskripsi tugas user yang dilakukan pada sistem.
3. Daftar tindakan yang diperlukan untuk menyelesaikan tugas user.
4. User yang melakukan dan pengalamannya serta pengetahuan evaluator untuk bisa berasumsi tentang mereka.

2.2.5.4 WebUse (website usability evaluation tool)

WebUse merupakan alat/*tool* yang digunakan untuk mengevaluasi *usability* dari suatu *website*. *Usability* merupakan salah satu faktor utama yang menentukan kesuksesan sebuah *website* (Chiew & Salim, 2003). Chiew dan Salim (2003) merumuskan 24 pertanyaan untuk kuesioner evaluasi *usability* yang dikategorikan menjadi empat yaitu *Content*, *Organization and Readability*; *Navigations and Links*; *User Interface Design*; serta *Performance and Effectiveness*. Tabel 2.1 merupakan daftar pertanyaan kuesioner WEBUSE berdasarkan kategorinya. Pada penelitian ini menggunakan beberapa pertanyaan kuesioner dari kategori *Navigations and Links* serta *User Interface Design*, dipilih sesuai kebutuhan.

Tabel 2.9 Kuesioner WEBUSE

Kategori	No.	Atribut
Content, Organization and Readability	1.	<i>This website contains most of my interest material and topics and they are up-to-date.</i>
	2.	<i>I can easily find what I want at this website</i>
	3.	<i>The content of this website is well organized.</i>
	4.	<i>Reading content at this website is easy.</i>
	5.	<i>I am comfortable and familiar with the language used.</i>
	6.	<i>I need not scroll left and right when reading at this website.</i>
	7.	<i>I can easily know where I am at this website.</i>

Navigations and Links	8	<i>This website provides useful cues and links for me to get the desired information.</i>
	9.	<i>It is easy to move around at this website by using the links or back button of the browser.</i>
	10.	<i>The links at this website are well maintained and updated.</i>
	11.	<i>The website does not open too many new browser windows when I am moving around.</i>
	12.	<i>Placement of links or menu is standard throughout the website and I can easily recognize them.</i>
User Interface Design	13.	<i>This website's interface design is attractive.</i>
	14.	<i>I am comfortable with the colours used at this website.</i>
	15.	<i>This website contains no feature that irritates me such as scrolling or blinking text and looping animations.</i>
	16.	<i>This website has a consistent feel and look.</i>
	17.	<i>This website does not contain too many Web advertisements.</i>
Performance and Effectiveness	18.	<i>The design of the website makes sense and it is easy to learn how to use it.</i>
	19.	<i>I need not wait too long to download a file or open a page.</i>
	20.	<i>I can easily distinguish between visited and not-visited links.</i>
	21.	<i>I can access this website most of the time.</i>
	22.	<i>This website responds to my actions as expected.</i>
	23.	<i>It is efficient to use this website.</i>
	24.	<i>This website always provides clear and useful messages when I don't know how to proceed.</i>

Sumber : Chiew & Salim (2003)

2.2.5.5 Traceability

Traceability adalah menelusuri persyaratan/*requirement* pada fitur atau kebutuhan pemangku kepentingan atau pada *design*, *code* dan *test*. Untuk tujuan pengujian kita harus dapat melacak *requirement* ke *test* untuk membuktikan bahwa *requirement* tercover oleh *test*. Salah satu tujuan *traceability* adalah untuk membantu memverifikasi bahwa semua *requirement* diimplementasikan dan bahwa aplikasi hanya mengimplementasikan *requirement*. (Fournier, 2009) *Traceability* penting untuk verifikasi perancangan. (Agarwal, et al., 2010) Beberapa kemungkinan tabel *traceability* adalah sebagai berikut (Pressman, 2001):

1. *Features traceability table*. Menunjukkan bagaimana *requirements* berelasi dengan pelanggan diamati fitur sistem / produk yang penting.
2. *Source traceability table*. Mengidentifikasi sumber kebutuhan masing-masing.
3. *Dependency traceability table*. Menunjukkan *requirement* terkait satu sama lain.
4. *Subsystem traceability table*. Mengkategorikan *requirement* dengan subsistem yang mereka perintah.
5. *Interface traceability table*. Menunjukkan bagaimana *requirement* berhubungan dengan antarmuka sistem internal dan eksternal.

BAB 3 METODOLOGI

Penelitian ini merupakan penelitian yang bertipe implementatif, di mana penelitian menitikberatkan pada luaran berupa produk baik software ataupun hardware. Dalam penelitian ini luaran produk yang dihasilkan adalah berupa software. Pendekatan implementatif yang digunakan dalam penelitian ini adalah pendekatan perancangan. Pendekatan perancangan merupakan pendekatan penelitian implementatif untuk membuat purwarupa (prototipe) melalui proses yang utuh mulai dari tahap analisis perancangan hingga pengujiannya. Pada bab ini peneliti mencoba menguraikan langkah-langkah yang akan dilakukan dalam penelitian. Tahap-tahap yang ada dalam penelitian ini adalah melakukan wawancara, studi literatur, pengumpulan data, pemodelan proses bisnis, analisis persyaratan sistem, perancangan sistem, evaluasi, evaluasi sistem, dan pengambilan kesimpulan. Gambar 3.1 menjelaskan alur penelitian yang akan digunakan dalam penelitian ini.



Gambar 3.1 Metodologi Penelitian

3.1 Studi Literatur

Studi literatur merupakan tahap mencari, memilih, mempelajari dan menyusun teori-teori dasar dan referensi yang mendukung penelitian yang akan dilakukan. Tahap yang dilakukan dalam studi literatur ini diantaranya melakukan mengumpulkan informasi di Dinas Pendapatan Kabupaten Madiun serta

melakukan pencarian dan pengumpulan literatur yang menunjang pengerjaan skripsi baik secara *online* maupun *offline*.

Teori-teori yang dijelaskan pada penelitian ini diantaranya adalah mengenai Dinas Pendapatan Kabupaten Madiun beserta pajak; teori mengenai proses bisnis beserta pemodelannya dengan BPMN; teori mengenai sistem informasi beserta analisis dan perancangannya; teori mengenai pemodelan berorientasi objek dengan UML yang meliputi *use case*, *activity diagram*, *sequence diagram* dan *class diagram*; teori mengenai *web service*; serta teori mengenai evaluasi untuk analisis dan perancangan sistem.

3.2 Pengumpulan Data

Tahap pengumpulan data merupakan tahap mencari dan mengumpulkan data yang diperlukan dalam penyusunan/pengerjaan skripsi. Tahap pengumpulan data pada penelitian ini dilakukan dengan wawancara, observasi dan studi dokumen di Dinas Pendapatan Daerah Madiun. Wawancara dilakukan secara langsung dilakukan di Dinas Pendapatan Kabupaten Madiun dengan bertanya kepada beberapa pegawai di setiap bidang yang ada. Dinas Pendapatan Kabupaten Madiun memiliki empat bidang yang saling bekerja sama yaitu Pendataan dan Penetapan, bidang Pembukuan dan Pelaporan, bidang Penagihan dan Keberatan, serta bidang Perencanaan. Data yang dikumpulkan adalah data-data yang mungkin akan dibutuhkan dalam tahap selanjutnya.

3.3 Pemodelan Proses Bisnis

Pemodelan proses bisnis dalam penelitian ini adalah tahap memodelkan proses bisnis penerimaan PAD yang saat ini berjalan di Dinas Pendapatan Kabupaten Madiun dan memodelkan proses bisnis usulan untuk penerimaan PAD di Dispenda Kabupaten Madiun. Pemodelan proses bisnis penerimaan PAD yang saat ini berjalan di Dispenda Kabupaten Madiun diperoleh dari hasil wawancara pada pihak Dispenda. Setelah dilakukan wawancara peneliti memodelkan proses bisnis saat ini dengan notasi BPMN. Kemudian dilakukan pemodelan proses bisnis usulan dari proses penerimaan PAD Kabupaten Madiun. Pada proses bisnis usulan ini akan ada aktivitas tambahan dan aktivitas yang mengalami perubahan dari proses bisnis yang saat ini berjalan di Dispenda Kabupaten Madiun.

3.4 Analisis Persyaratan Sistem

Analisis sistem merupakan tahap memahami dan menyusun persyaratan-persyaratan yang dibutuhkan sistem. Setelah dilakukan pemodelan proses bisnis usulan proses penerimaan PAD, dilakukan tahap analisis persyaratan sistem yang disesuaikan dengan perubahan-perubahan pada proses bisnis usulan. Proses analisis persyaratan yang dilakukan pada penelitian diantaranya adalah analisis permasalahan, melakukan analisis pemangku kepentingan, analisis peran dan representatif dari pemangku kepentingan yang telah didefinisikan sebelumnya, analisis tipe pengguna sistem, analisis persyaratan sistem, menggambarkan persyaratan sistem dengan diagram *use case*, mendeskripsikan setiap *use case* dan

menggambarkan alur *use case* dengan diagram aktivitas. Analisis persyaratan sistem ini nantinya akan digunakan untuk tahap selanjutnya yaitu tahap perancangan sistem.

3.5 Perancangan Sistem

Perancangan sistem adalah tahap yang dilakukan setelah analisis persyaratan sistem. Sebelum dilakukan tahap perancangan sistem dilakukan tahap analisis. Tahap analisis ini merupakan jembatan antara analisis persyaratan sistem dan perancangan. Analisis yang dilakukan pada penelitian ini diantaranya adalah mengidentifikasi kelas-kelas analisis, identifikasi mekanisme analisis, dan unifikasi kelas-kelas analisis. Perancangan yang dilakukan pada penelitian ini diantaranya mengidentifikasi elemen desain, perancangan *sequence diagram*, perancangan arsitektur, pemodelan data, perancangan antarmuka dan membuat purwarupa (prototipe).

3.6 Evaluasi

Evaluasi pada penelitian ini bertujuan untuk memperoleh saran dan masukan dari pengguna serta melihat apakah alur yang dispesifikasikan yang diimplementasikan pada prototipe sesuai dengan apa yang diharapkan oleh pengguna. Evaluasi yang dilakukan dalam penelitian ini adalah dengan melakukan verifikasi dan validasi. Verifikasi digunakan untuk memastikan perancangan dan prototipe yang dibuat sesuai dengan spesifikasi persyaratan yang telah ditentukan. Sedangkan validasi digunakan untuk memastikan spesifikasi persyaratan yang diimplementasikan dengan prototipe sesuai dan memuaskan keinginan pemangku kepentingan.

Verifikasi yang dilakukan pada penelitian ini diantaranya evaluasi *traceability* dan tinjauan skenario *use case*. Evaluasi *traceability* dilakukan dengan membuat matriks *traceability* dan tinjauan *traceability* dilakukan dengan mengikuti daftar tinjauan yang telah dibuat oleh Karl E. Wiegers. Tinjauan skenario *use case* dilakukan dengan membandingkan alur pengguna pada spesifikasi sistem dengan alur yang dilakukan oleh pengguna atas pengujian lima skenario *use case*. Validasi yang dilakukan pada penelitian ini adalah dengan tinjauan prototipe. Tinjauan prototipe ini dilakukan dengan menguji lima skenario *use case* dan kemudian pengguna melakukan *review* atas prototipe. Selain melakukan peninjauan atas prototipe berdasarkan skenario *use case* peninjauan prototipe juga dilakukan dengan meninjau navigasi dan antarmuka prototipe dengan bantuan kuesioner Webuse.

3.7 Kesimpulan dan Saran

Pembuatan kesimpulan dan saran adalah tahap akhir dalam penelitian ini. Pada akhir penulisan terdapat jawaban dari masalah yang telah dirumuskan serta saran yang bertujuan untuk memperbaiki kesalahan dan penyempurnaan terhadap penelitian skripsi ini.

BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN

Bagian ini menjelaskan mengenai pemodelan proses bisnis dan analisis persyaratan sistem yang meliputi analisis proses bisnis saat ini, analisis permasalahan, analisis proses bisnis usulan, analisis pemangku kepentingan analisis kebutuhan pengguna dan pemangku kepentingan, pernyataan posisi produk, fitur, persyaratan fungsional dan non fungsional, aktor, *use case diagram*, deskripsi *use case*, serta diagram aktivitas.

4.1 Proses bisnis saat ini

Proses bisnis saat ini pada gambar Gambar 4.1 merupakan proses bisnis penerimaan pajak daerah yang saat ini berjalan di Kabupaten Madiun. Proses bisnis dimulai dari pendataan wajib pajak oleh bidang pendataan dan penetapan, berikut ini penjabaran proses bisnis penerimaan pajak yang saat ini berjalan di Kabupaten Madiun :

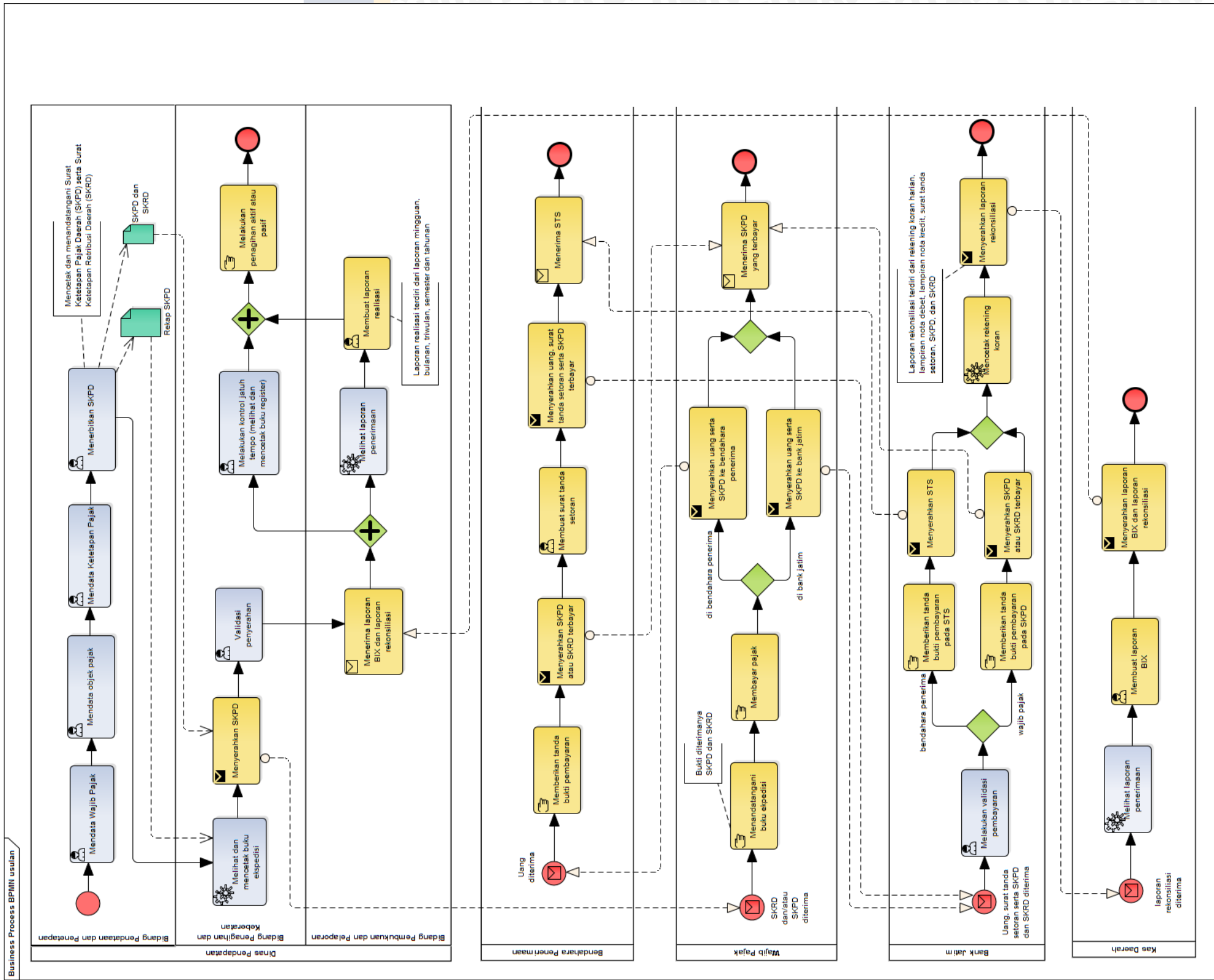
1. Bidang pendataan dan penetapan melakukan pendataan wajib pajak dan objek pajak pada sistem.
2. Bidang pendataan dan penetapan pendataan melakukan penetapan pajak pada sistem.
3. Bidang pendataan dan penetapan menerbitkan SKPD. Proses penerbitan ini merupakan proses mencetak dan menandatangani SKPD sebelum diserahkan ke wajib pajak.
4. Bidang penagihan dan keberatan melakukan rekap SKPD yang tercetak/telah terbit pada buku ekspedisi.
5. Bidang penagihan dan keberatan menyerahkan SKPD kepada wajib pajak.
6. Wajib pajak menerima SKPD dan menandatangani buku ekspedisi sebagai bukti bahwa SKPD telah diserahkan kepada wajib pajak.
7. Wajib pajak yang telah menerima SKPD akan membayar pajak ke bank jatim atau ke bendahara penerima.
 - a. Jika wajib pajak membayar di bendahara penerima
 1. Wajib pajak menyerahkan SKPD ke bendahara penerima
 2. Bendahara penerima memberikan tanda bukti pembayaran pajak pada SKPD
 3. Bendahara penerima menyerahkan SKPD terbayar/yang telah diberikan bukti pembayaran
 4. Wajib pajak menerima SKPD terbayar
 5. Wajib pajak selesai
 - b. Jika wajib pajak membayar di bank jatim
 1. Wajib pajak menyerahkan SKPD ke bank jatim
 2. Bank jatim menerima SKPD
 3. Bank jatim memasukkan uang ke rekening Dispenda Kabupaten Madiun.
 4. Bank jatim memberikan tanda bukti pembayaran pajak pada SKPD

5. Bank jatim memberikan SKPD terbayar pada wajib pajak
6. Wajib pajak menerima SKPD terbayar
7. Wajib pajak selesai
8. Bendahara penerima membuat surat tanda bukti setoran (STS) setiap harinya. STS merupakan rekap SKPD yang dibayar oleh wajib pajak ke bendahara penerima
9. Bendahara penerima menyerahkan uang, STS, dan SKPD terbayar ke bank jatim
10. Bank jatim memasukkan uang ke rekening Dispenda
11. Bank jatim memberikan tanda bukti pada STS
12. Bank jatim menyerahkan STS pada bendahara penerima
13. Bendahara penerima menerima STS
14. Bendahara penerima selesai
15. Bank jatim mencetak rekening koran setiap harinya. Rekening koran berisi rekap pembayaran pajak yang ada di rekening dispenda
16. Bank jatim menyerahkan laporan rekonsiliasi kepada kas daerah. Laporan rekonsiliasi terdiri dari rekening koran harian, lampiran nota detail debit, lampiran nota kredit, STD, SKPD.
17. Bank jatim selesai
18. Kas daerah menerima laporan rekonsiliasi dari bank jatim
19. Kas daerah membuat laporan BIX dengan merekap dari laporan rekonsiliasi
20. Kas daerah menyerahkan laporan BIX dan laporan rekonsiliasi ke bidang pembukuan dan pelaporan
21. Kas daerah selesai
22. Bidang pembukuan dan pelaporan menerima laporan BIX dan laporan rekonsiliasi
23. a. Bidang pembukuan dan pelaporan membuat laporan realisasi dengan merekap dari laporan rekonsiliasi dan laporan BIX. Laporan realisasi dibuat setiap minggu, setiap bulan, setiap semester dan setiap tahun.
b. Bidang penagihan dan keberatan merekap SKPD terbayar pada buku register
24. Bidang penagihan dan keberatan mencari wajib pajak yang hampir atau sudah jatuh tempo
25. Bidang penagihan dan keberatan melakukan penagihan aktif atau pasif pada wajib pajak
26. Dinas pendapatan selesai

Gambar 4.1 Proses bisnis penerimaan PAD saat ini

4.2 Proses bisnis usulan

Gambar 4.2 berikut ini merupakan proses bisnis usulan dari proses Penerimaan PAD di Kabupaten Madiun.



Gambar 4.2 Proses Bisnis Penerimaan PAD usulan

Proses bisnis usulan pada Gambar 4.2 merupakan proses bisnis yang akan berjalan setelah diterapkan sistem yang akan dirancang. Pada proses bisnis usulan ini ada aktivitas tambahan dan aktivitas yang mengalami perubahan dari proses yang bisnis saat ini berjalan di Dispenda Kabupaten Madiun (Gambar 4.1). Tabel 4.1 berikut ini adalah daftar perubahan dari proses bisnis saat ini ke proses bisnis usulan.

Tabel 4.1 Daftar tambahan dan perubahan aktivitas proses bisnis

Kode Aktivitas	Nama Aktivitas	Keterangan
A-BP-001	Mendata wajib pajak	Mendata wajib pajak yang mendaftar dilakukan oleh bidang pendataan dan penetapan pada sistem yang terintegrasi
A-BP-002	Mendata objek pajak	Mendata objek pajak yang didaftarkan wajib pajak dilakukan oleh bidang pendataan dan penetapan pada sistem yang terintegrasi
A-BP-003	Mendata ketetapan pajak	Mendata ketetapan pajak atas objek pajak yang telah terdaftar di sistem dilakukan oleh bidang pendataan dan penetapan pada sistem yang terintegrasi
A-BP-004	Menerbitkan SKPD	Bidang pendataan dan penetapan mencetak SKPD dan menandatangani SKPD tercetak
A-BP-005	Melihat dan mencetak buku ekspedisi	Bidang penagihan dan keberatan melihat dan mencetak daftar SKPD yang telah terbit/di cetak oleh bidang pendataan dan penetapan pada sistem
A-BP-006	Validasi penyerahan SKPD	Setelah menyerahkan SKPD bidang penagihan dan keberatan melakukan validasi penyerahan pada sistem
A-BP-007	Melakukan kontrol jatuh tempo (melihat dan mencetak buku register)	Bidang penagihan dan keberatan melihat dan mencetak data SKPD yang disertai status penyerahan dan status pembayaran pada sistem
A-BP-008	Melihat laporan penerimaan	Bidang pembukuan dan pelaporan melihat laporan penerimaan pajak pada sistem untuk membuat laporan realisasi
A-BP-009	Melakukan validasi pembayaran	Pihak bank melakukan validasi pembayaran dengan sistem yang terintegrasi dengan Dispenda

Tabel 4.1 Daftar tambahan dan perubahan aktivitas proses bisnis (lanjutan)

Kode Aktivitas	Nama Aktivitas	Keterangan
A-BP-010	Melihat laporan penerimaan	Kas daerah melihat laporan penerimaan pajak untuk membuat laporan BIX

4.3 Analisis permasalahan

Analisis permasalahan merupakan analisis secara ringkas dari masalah yang dipecahkan. Tabel 4.2 merupakan analisis permasalahan berdasarkan proses bisnis penerimaan PAD.

Tabel 4.2 Analisis Permasalahan

Masalah	1. Sistem yang ada hanya menyediakan informasi penetapan dan pendataan pajak bagi bidang penetapan dan pendataan. 2. Sistem hanya dapat digunakan oleh DISPENDA bidang pendataan dan penetapan. 3. Belum ada informasi rinci dan <i>real time</i> mengenai wajib pajak yang belum membayar maupun yang sudah membayar. 4. Belum ada sistem yang berhubungan dengan bank jatim.
Mempengaruhi	Dinas pendapatan, bidang perencanaan, bidang pendataan dan penetapan, bidang pembukuan dan pelaporan, penagihan dan keberatan, serta kas daerah.
Dampak	1. Bidang penagihan dan keberatan perlu melakukan rekap manual dari SKPD untuk buku membuat buku registrasi. 2. Perlu dilakukan pencarian manual mengenai status penyerahan SKPD kepada Wajib Pajak 3. Perlu dilakukan rekap manual mengenai pembayaran pajak yaitu dari SKPD terbayar. 4. Terjadi kesalahan ketika pencarian pajak yang belum dibayar oleh wajib pajak. 5. Target pendapatan tidak tercapai tepat waktu.
Solusi	Menyediakan pendataan pembayaran yang dapat mengubah status pembayaran pajak, menyediakan informasi secara <i>real-time</i> mengenai pajak yang sudah terbayar ataupun belum, menyediakan sistem yang berhubungan dengan bank jatim yang dapat memvalidasi pembayaran, dan menyediakan informasi yang terintegrasi antara pihak dalam maupun pihak luar.

4.4 Analisis Pemangku Kepentingan dan Pengguna

Pemangku kepentingan merupakan sumber utama persyaratan. Dalam penelitian ini dilakukan analisis tipe pemangku kepentingan dan tipe pengguna.

4.4.1 Tipe Pemangku Kepentingan

Tipe pemangku kepentingan merupakan semua jenis pemangku kepentingan yang terpengaruh oleh proyek dan produk yang dihasilkan. Tabel 4.3 merupakan penjelasan mengenai analisis pemangku kepentingan yang akan terlibat pada proyek sistem informasi penerimaan PAD Non PBB.

Tabel 4.3 Analisis Tipe Pemangku kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Contoh
Pengguna	Pengguna sebenarnya dari sistem. Orang yang langsung menggunakan sistem.	Kepala bidang pendataan dan penetapan, kepala bidang pembukuan dan pelaporan, kepala bidang penagihan dan keberatan, pegawai kas daerah
Pengembang	Orang yang akan melakukan pengembangan/pembuatan sistem dengan tahap-tahap tertentu dalam pengembangan.	Penulis
Pelanggan	Organisasi yang mengelola pendapatan daerah kabupaten Madiun yang akan menjadi pemilik dari sistem.	Dinas pendapatan Kabupaten Madiun

Tipe pemangku kepentingan pengguna dan pelanggan berperan sebagai *ambassador user* yang bertanggung jawab menyampaikan permasalahan, berpartisipasi dalam analisis dan perancangan sistem serta menyediakan informasi yang diperlukan dalam analisis dan perancangan sistem. Tipe pemangku kepentingan pengembang berperan sebagai analis sistem yang bertanggung jawab melakukan analisis dan perancangan sistem serta mendokumentasikan hasil analisis dan perancangan sistem.

4.4.2 Tipe Pengguna

Tipe pengguna menjelaskan mengenai karakteristik pengguna sistem. Tabel 4.4 merupakan penjelasan mengenai tipe-tipe pengguna yang terlibat dalam bisnis proses penerimaan PAD Non PBB.

Tabel 4.4 Tipe pengguna

Tipe Pengguna	Karakteristik	Contoh
Adapter Teknologi	- Umur antara 30 – 50 tahun. - Bekerja di lingkungan pemerintahan.	Kepala bidang pendataan dan penetapan, Kepala bidang pembukuan dan pelaporan, Kepala bidang penagihan dan keberatan, Pegawai kas daerah
Pengguna Standar	- Umur antara 30 – 50 tahun. - Bekerja di lingkungan pemerintahan.	Pegawai bidang pendataan dan penetapan, Pegawai bidang pembukuan dan pelaporan, Pegawai bidang penagihan dan keberatan, Pegawai kas daerah

4.5 Kebutuhan Pengguna dan Pemangku Kepentingan

Kebutuhan merupakan hal-hal yang perlu dilakukan sistem yang diperoleh dari pemangku kepentingan berdasarkan masalah yang ingin mereka selesaikan. Tabel 4.5 merupakan penjelasan mengenai kebutuhan yang diperoleh dari pengguna dan pemangku kepentingan pada sistem informasi penerimaan PAD Non PBB. Kebutuhan ini disajikan dengan solusi atas masalah yang mempengaruhi pemangku kepentingan. Solusi yang diberikan ini didapatkan berdasarkan proses bisnis usulan yang telah dimodelkan sebelumnya.

Setelah memperoleh kebutuhan berdasarkan wawancara mengenai masalah yang ingin diselesaikan oleh pengguna, kebutuhan ini akan diberikan prioritas. Prioritas pada kebutuhan pengguna dan pemangku kepentingan ini berdasarkan aturan MosCow. Jika suatu kebutuhan tidak dipenuhi mengakibatkan pembatalan proyek, maka kebutuhan tersebut memiliki prioritas *Must Have* (MO), karena tanpa kebutuhan tersebut suatu solusi tidak dapat menyelesaikan permasalahan. Jika suatu kebutuhan tidak dipenuhi oleh suatu solusi dan solusi tersebut masih dapat menyelesaikan permasalahan atau masih layak, maka kebutuhan tersebut memiliki prioritas *Should Have* (SO). Jika suatu kebutuhan suatu sistem diinginkan tetapi kurang penting, maka kebutuhan tersebut memiliki prioritas *Could Have* (CO). Untuk membedakan prioritas *Should Have* dan *Could Have* adalah dengan meninjau tingkat kegagalan yang disebabkan oleh persyaratan yang tidak terpenuhi, diukur dari nilai bisnis atau jumlah orang yang akan terkena dampaknya. Jika kebutuhan yang berprioritas *could have* tidak dipenuhi maka akibat yang akan ditimbulkan pada proyek akan lebih kecil. Jika kebutuhan

berprioritas *should have* tidak dipenuhi maka akibat yang akan ditimbulkan pada proyek akan lebih besar. Jika suatu kebutuhan disepakati untuk tidak dikerjakan pada proyek ini maka kebutuhan tersebut memiliki prioritas *Won't have this time* (W).

Tabel 4.5 Kebutuhan pengguna dan pemangku kepentingan

Kebutuhan	Pemangku kepentingan	Prioritas	Situasi saat ini	Solusi yang ditawarkan	Kode Aktivitas Proses Bisnis
Kemudahan mendapatkan informasi SKPD yang telah diterbitkan	Bidang penagihan dan keberatan	MO	Merekap manual dari SKPD yang telah dicetak oleh bidang pendataan dan penetapan	Sistem menyediakan informasi data SKPD yang telah diterbitkan yang terintegrasi dengan bidang pendataan dan penetapan	A-BP-005
Kemudahan mendapatkan informasi data pajak yang telah diserahkan dan telah dibayar	Bidang penagihan dan keberatan	MO	Merekap dari rekening koran dan SKPD terbayar	Sistem menyediakan informasi data pajak yang terintegrasi secara internal maupun eksternal	A-BP-007
Kemudahan mencari SKPD yang telah diserahkan	Bidang penagihan dan keberatan	S	Mencari secara manual dalam buku register	Data SKPD yang telah diserahkan dapat dicari di sistem yang menyediakan fitur untuk memvalidasi penyerahan SKPD	A-BP-006
Kemudahan mengelola data wajib pajak, objek pajak dan ketentuan	Bidang penetapan dan pendataan	MO	Data wajib pajak, objek pajak dan ketentuan pajak	Pengelolaan wajib pajak, objek pajak dan ketentuan pajak pada sistem dapat	A-BP-001, A-BP-002, A-BP-003, A-BP-004

Tabel 4.5 Kebutuhan pengguna dan pemangku kepentingan (lanjutan)

Kebutuhan	Pemangku kepentingan	Prioritas	Situasi saat ini	Solusi yang ditawarkan	Kode Aktivitas Proses Bisnis
pajak yang terintegrasi secara internal			belum terintegrasi	terintegrasi secara internal	
Informasi <i>real time</i> mengenai pembayaran pajak	Dispenda kabupaten Madiun	MO	-	Menyediakan <i>web service</i> untuk pembayaran pajak agar dapat terintegrasi dengan bank jatim	A-BP-009
Kemudahan mendapatkan informasi jumlah penerimaan yang bersumber dari pajak	Dispenda kabupaten Madiun, Kas Daerah	MO	Melakukan rekap manual dari rekening koran dan SKPD terbayar	Sistem menyediakan informasi penerimaan pajak	A-BP-008, A-BP-010

4.6 Fitur

Fitur merupakan pernyataan informal kemampuan yang disediakan sistem, sebagai penjelasan singkat atas perilaku/*behaviour* sistem. Tabel 4.6 merupakan penjelasan mengenai fitur-fitur yang ada pada sistem informasi penerimaan PAD Non PBB.

Setelah memperoleh fitur berdasarkan kebutuhan pengguna, fitur ini akan diberikan prioritas. Prioritas pada fitur ini berdasarkan aturan MosCow. Jika suatu fitur tidak dipenuhi mengakibatkan pembatalan proyek, maka fitur tersebut memiliki prioritas *Must Have* (MO), karena tanpa fitur tersebut suatu solusi tidak dapat menyelesaikan permasalahan. Jika suatu fitur tidak dipenuhi oleh suatu solusi dan solusi tersebut masih dapat menyelesaikan permasalahan atau masih layak, maka fitur tersebut memiliki prioritas *Should Have* (SO). Jika suatu fitur berdasarkan kebutuhan yang diinginkan tetapi kurang penting, maka fitur tersebut memiliki prioritas *Could Have* (CO). Untuk membedakan prioritas *Should Have* dan *Could Have* adalah dengan meninjau tingkat kegagalan yang disebabkan oleh fitur yang tidak terpenuhi, diukur dari nilai bisnis atau jumlah orang yang akan terkena dampaknya. Jika fitur yang berprioritas *could have* tidak dipenuhi maka akibat yang

akan ditimbulkan pada proyek akan lebih kecil. Jika fitur berprioritas *should have* tidak dipenuhi maka akibat yang akan ditimbulkan pada proyek akan lebih besar. Jika suatu fitur disepakati untuk tidak dikerjakan pada proyek ini, maka fitur tersebut memiliki prioritas *Won't have this time* (W).

Tabel 4.6 Fitur

Kode Fitur	Nama Fitur	Deskripsi	Prioritas
FITUR-001	Login	Sistem menyediakan fungsi yang dapat digunakan oleh kepala bidang pendataan dan penetapan, kepala bidang pembukuan dan pelaporan, kepala bidang penagihan dan keberatan, pegawai kas daerah	MO
FITUR-002	Kelola pengguna	Sistem menyediakan kemampuan melakukan pengelolaan pengguna sistem	MO
FITUR-003	Kelola wajib pajak	Sistem menyediakan kemampuan melakukan pengelolaan pendataan wajib pajak	MO
FITUR-004	Kelola objek pajak	Sistem menyediakan kemampuan melakukan pengelolaan pendataan objek pajak	MO
FITUR-005	Kelola penetapan pajak	Sistem menyediakan kemampuan melakukan pengelolaan penetapan pajak	MO
FITUR-006	Cetak SKPD	Sistem menyediakan kemampuan melakukan percetakan ketetapan pajak	MO
FITUR-007	Rekap SKPD	Sistem menyediakan kemampuan memberikan informasi SKPD	MO
FITUR-008	Kelola penyerahan SKPD	Sistem menyediakan kemampuan melakukan <i>pengelolaan</i> penyerahan SKPD	S
FITUR-009	Validasi pembayaran	Sistem menyediakan kemampuan mencatat pembayaran pajak	MO
FITUR-010	Rekap penerimaan pajak	Sistem menyediakan kemampuan memberikan informasi penerimaan pendapatan yang bersumber dari pajak	MO
FITUR-011	-	Sistem menyediakan kemampuan memberikan informasi secara <i>real-time</i> .	MO

4.7 Persyaratan Perangkat Lunak

Persyaratan perangkat lunak merupakan pernyataan berupa kondisi atau kemampuan yang harus dipenuhi oleh sistem. Persyaratan perangkat lunak ini terdiri dari persyaratan fungsional dan persyaratan non fungsional.

4.7.1 Persyaratan fungsional

Tabel 4.7 merupakan penjelasan mengenai persyaratan fungsional pada sistem informasi penerimaan PAD Non PBB.

Tabel 4.7 Persyaratan fungsional sistem

Kode Fitur	Kode Fungsi Dasar	Kode Lengkap Persyaratan Fungsional	Deskripsi
FITUR-001	PD-F-001	PD-F-001.1	Sistem memiliki fungsi bagi pengguna untuk masuk ke sistem dan menggunakan sistem
FITUR-002	PD-F-002	PD-F-002.1	Sistem dapat digunakan oleh admin untuk menambahkan data pengguna
		PD-F-002.2	Sistem dapat digunakan oleh admin untuk melakukan edit data pengguna
		PD-F-002.3	Sistem dapat digunakan oleh admin untuk menghapus data pengguna
		PD-F-002.4	Sistem dapat memberikan informasi pengguna yang sudah terdaftar di sistem kepada admin
FITUR-003	PD-F-003	PD-F-003.1	Sistem dapat digunakan oleh bidang pendataan dan ketetapan untuk menambahkan wajib pajak baru
		PD-F-003.2	Sistem dapat digunakan oleh bidang pendataan dan ketetapan untuk melakukan edit data wajib pajak
		PD-F-003.3	Sistem dapat digunakan oleh bidang pendataan dan ketetapan untuk menghapus data wajib pajak
		PD-F-003.4	Sistem dapat memberikan

Tabel 4.7 Persyaratan fungsional sistem (lanjutan)

Kode Fitur	Kode Fungsi Dasar	Kode Lengkap Persyaratan Fungsional	Deskripsi
			informasi wajib pajak yang telah terdaftar di sistem menambahkan objek pajak baru.
FITUR-004	PD-F-004	PD-F-004.1	Sistem dapat digunakan oleh bidang pendataan dan penetapan untuk
		PD-F-004.2	Sistem dapat digunakan oleh bidang pendataan dan penetapan untuk melakukan edit data objek pajak
		PD-F-004.3	Sistem dapat digunakan oleh bidang pendataan dan penetapan untuk menghapus data objek pajak
		PD-F-004.4	Sistem dapat memberikan informasi objek pajak yang telah terdaftar di sistem
FITUR-005	PD-F-005	PD-F-005.1	Sistem dapat digunakan oleh bidang pendataan dan penetapan untuk menetapkan pajak baru
		PD-F-005.2	Sistem dapat digunakan oleh bidang pendataan dan penetapan untuk melakukan edit data ketetapan pajak
		PD-F-005.3	Sistem dapat digunakan oleh bidang pendataan dan penetapan untuk menghapus data ketetapan pajak
		PD-F-005.4	Sistem dapat memberikan informasi ketetapan pajak yang telah terdata di sistem
FITUR-006	PD-F-006	PD-F-006.1	Sistem dapat digunakan oleh pendataan dan penetapan untuk melakukan percetakan SKPD atas pajak yang telah ditetapkan
FITUR-007	PD-F-007	PD-F-007.1	Sistem dapat memberikan informasi SKPD yang telah

Tabel 4.7 Persyaratan fungsional sistem (lanjutan)

Kode Fitur	Kode Fungsi Dasar	Kode Lengkap Persyaratan Fungsional	Deskripsi
			terbit disertai status pembayaran kepada bidang penagihan dan keberatan
FITUR-008	PD-F-008	PD-F-008.1	Sistem dapat digunakan oleh bidang penagihan dan keberatan untuk melakukan validasi penyerahan SKPD kepada bidang penagihan dan keberatan
		PD-F-008.2	Sistem dapat memberikan informasi mengenai SKPD yang telah terbit disertai status penyerahan SKPD kepada bidang penagihan dan keberatan
FITUR-009	PD-F-009	PD-F-009.1	Sistem dapat terintegrasi dengan bank untuk melakukan validasi pembayaran SKPD
FITUR-010	PD-F-010	PD-F-010.1	Sistem dapat memberikan informasi penerimaan pendapatan yang bersumber dari pajak kepada bidang pembukuan dan pelaporan, bidang penagihan dan keberatan serta kas daerah
FITUR-010	PD-F-010	PD-F-010.1	pembukuan dan pelaporan, bidang penagihan dan keberatan serta kas daerah

4.7.2 Persyaratan non fungsional

Tabel 4.8 merupakan penjelasan mengenai persyaratan non fungsional pada sistem informasi penerimaan PAD Non PBB. Pada penelitian ini persyaratan non fungsional hanya sebatas pendefinisian pada tahap analisis persyaratan saja dan tidak sampai tahap perancangan dan pengujian.

Tabel 4.8 Persyaratan non fungsional sistem

Kode Persyaratan Non Fungsional	Nama Aspek Persyaratan Non Fungsional	Deskripsi
PD-NF-001	Performance	Data <i>ter-update</i> secara <i>real time</i> , waktu respon kurang dari 5 detik

4.8 Pernyataan kedudukan produk

Pernyataan posisi produk merupakan penjelasan/deskripsi singkat dan sederhana atas apa yang dilakukan oleh suatu sistem. Tabel 4.9 ini adalah pernyataan posisi produk dari Sistem Informasi Penerimaan PAD Non PBB Dinas Pendapatan Kabupaten Madiun :

Tabel 4.9 Pernyataan kedudukan produk

Untuk	Dinas Pendapatan Kabupaten Madiun dan Bank Jatim
Yang	Membutuhkan akses atas pendataan, penetapan dan pembayaran/penerimaan pajak.
Produk	Sistem informasi penerimaan pajak adalah sistem informasi terintegrasi.
Yang	Mampu mengelola informasi pendataan, penetapan dan penerimaan pajak serta melakukan validasi pembayaran pajak.
Tidak seperti	Mengakses informasi pendataan, penetapan dan pembayaran/penerimaan pajak dengan merekap manual dari rekening koran dan SKPD.
Produk kami	Menyediakan informasi yang <i>real time</i> dan dapat di akses secara <i>online</i> .

4.9 Aktor

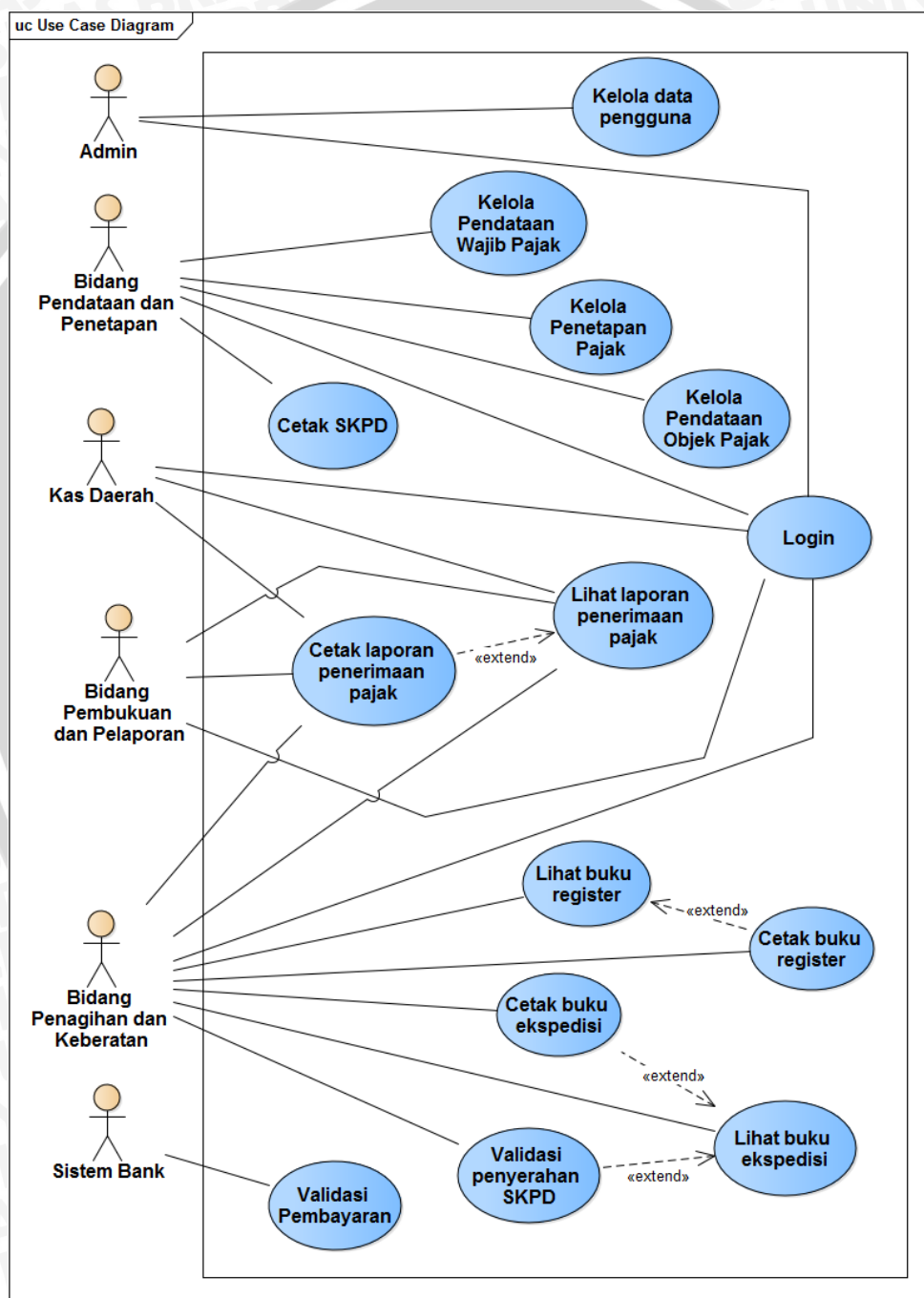
Setelah dilakukan analisis persyaratan dilakukan identifikasi aktor dari sistem dan perannya pada *use case*. Aktor merepresentasikan semua yang berada di luar sistem yang menggunakan sistem untuk melakukan sesuatu. Tabel 4.10 merupakan penjelasan mengenai aktor yang terlibat pada sistem informasi penerimaan PAD Non PBB.

Tabel 4.10 Aktor

Aktor	Tipe Pengguna	Contoh
Admin	Adapter teknologi, pengguna standar	Staf IT
Pegawai Bidang pendataan dan penetapan	Adapter teknologi, pengguna standar	Kepala bidang pendataan dan penetapan
Pegawai Kas daerah	Adapter teknologi, pengguna standar	Pegawai kas daerah
Pegawai Bidang pembukuan dan pelaporan	Adapter teknologi, pengguna standar	Kepala bidang pembukuan dan pelaporan
Pegawai Bidang penagihan dan keberatan	Adapter teknologi, pengguna standar	Kepala bidang penagihan dan keberatan
Sistem bank	-	Sistem bank

4.10 Diagram Use Case

Diagram *use case* menggambarkan hubungan antara pengguna dan fungsi dari sistem. Berdasarkan hasil analisis aktor dalam sistem ini terdapat enam aktor yaitu admin, bidang pendataan dan penetapan, kas daerah, bidang pembukuan dan pelaporan, bidang penagihan dan keberatan serta sistem bank. Setiap aktor memiliki peran masing-masing dalam sistem. Gambar 4.3 berikut ini merupakan hasil analisis diagram *use case* dari sistem informasi penerimaan PAD.



Gambar 4.3 Diagram *use case* Sistem Informasi Penerimaan PAD Non-PBB

Pemetaan *use case* terhadap persyaratan fungsional serta aktor dijelaskan pada Tabel 4.11 berikut ini.

Tabel 4.11 Pemetaan Diagram *Use Case*

Kode Fungsional	Kode <i>Use Case</i>	<i>Use Case</i>	Aktor
PD-F-002	A-UC-001	Kelola data pengguna	Admin
PD-F-003	A-UC-002	Kelola pendataan wajib pajak	Bidang pendataan dan penetapan
PD-F-004	A-UC-003	Kelola pendataan objek pajak	Bidang pendataan dan penetapan
PD-F-005	A-UC-004	Kelola penetapan pajak	Bidang pendataan dan penetapan
PD-F-006	A-UC-005	Cetak SKPD	Bidang pendataan dan penetapan
PD-F-010	A-UC-006	Lihat laporan penerimaan pajak	Kas daerah, bidang pembukuan dan pelaporan, bidang penagihan dan keberatan
PD-F-010	A-UC-007	Cetak laporan penerimaan pajak	Kas daerah, bidang pembukuan dan pelaporan, bidang penagihan dan keberatan
PD-F-007	A-UC-008	Lihat buku register	Bidang penagihan dan keberatan
PD-F-007	A-UC-009	Cetak buku register	Bidang penagihan dan keberatan
PD-F-008	A-UC-010	Lihat buku ekspedisi	Bidang penagihan dan keberatan
PD-F-008	A-UC-011	Cetak buku ekspedisi	Bidang penagihan dan keberatan
PD-F-008	A-UC-012	Validasi penyerahan SKPD	Bidang penagihan dan keberatan
PD-F-009	A-UC-013	Validasi pembayaran	Sistem bank
PD-F-001	A-UC-014	Login	Admin, bidang pendataan dan penetapan, kas daerah, bidang pembukuan dan pelaporan, bidang penagihan dan keberatan

4.11 Spesifikasi Use Case

Spesifikasi *use case* menjelaskan lebih detail setiap *use case* sistem. Penjelasan ini meliputi nama *use case*, *brief description*, *pre-condition*, *basic flow*, *alternatif flow*, *subflow*, *post condition* dan *glossary*. Berikut ini merupakan deskripsi dari empat belas *use case*.

1. Login

Spesifikasi *use case login* dijelaskan pada Tabel 4.12 berikut ini.

Tabel 4.12 Spesifikasi Use Case Login

Name	Login
Brief Description	<i>Use case</i> 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.
Pre-Condition	1. Admin, bidang pendataan dan penetapan, kas daerah, bidang pembukuan dan pelaporan, atau bidang penagihan dan keberatan membuka sistem 2. Sistem meminta pengguna untuk melakukan <i>login</i>
Basic Flow	{Mengisi formulir login} 1. <i>Use case</i> dimulai ketika Admin, bidang pendataan dan penetapan, kas daerah, bidang pembukuan dan pelaporan, atau bidang penagihan dan keberatan memasukkan <i>username</i> dan <i>password</i> {Memilih melakukan login} 2. Admin, bidang pendataan dan penetapan, kas daerah, bidang pembukuan dan pelaporan, atau bidang penagihan dan keberatan memilih untuk melakukan <i>login</i> {Melakukan validasi akun} 3. Sistem mengkonfirmasi ID dan <i>password</i> valid {Menampilkan notifikasi} 4. Sistem menampilkan pesan berhasil masuk ke sistem {Use case selesai} 5. <i>Use case</i> selesai.
Alternative Flows	1. ID 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. Sistem akan menampilkan notifikasi <i>username</i> dan <i>password</i> tidak sesuai, kemudian {<i>Use case</i> selesai} 2. <i>Username</i> atau <i>password</i> tidak diisi / formulir tidak diisi dengan lengkap

	Pada saat {mengisi formulir <i>login</i>} Admin, bidang pendataan dan penetapan, kas daerah, bidang pembukuan dan pelaporan, atau bidang penagihan dan keberatan tidak mengisi salah satu kolom formulir kemudian {melakukan <i>login</i>}. Sistem akan menampilkan pesan untuk melengkapi formulir, kemudian <i>use case</i> dilanjutkan {mengisi formulir <i>login</i>}. 3. Batal melakukan <i>login</i> Pada saat {mengisi formulir <i>login</i>} atau {melakukan <i>login</i>} Admin, bidang pendataan dan penetapan, kas daerah, bidang pembukuan dan pelaporan, atau bidang penagihan dan keberatan menutup program, kemudian {<i>use case</i> selesai}.
Subflow	-
Post-Condition	1. Admin, bidang pendataan dan penetapan, kas daerah, bidang pembukuan dan pelaporan, atau bidang penagihan dan keberatan berhasil masuk dan dapat menggunakan fitur yang disediakan sistem sesuai jenis pengguna

Glossary

Data *login* yang dimaksud adalah *username* dan *password*.

2. Kelola data pengguna

Spesifikasi *use case* kelola data pengguna dijelaskan pada Tabel 4.13 berikut ini.

Tabel 4.13 Spesifikasi Use Case Kelola Data Pengguna

Name	Kelola data pengguna
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana data pengguna sistem ditambahkan, diubah dan dihapus.
Pre-Condition	1. Admin menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Admin telah berhasil masuk ke sistem.
Basic Flow	{Memilih kelola pengguna} 1. <i>Use case</i> dimulai ketika admin memilih opsi kelola pengguna. {Memilih opsi pengelolaan} 2. Sistem akan menampilkan daftar/data pengguna yang terdaftar di sistem disertai opsi mengubah data.

	pengguna ataupun menghapus data pengguna dan menyediakan fungsi tambah pengguna baru. Jika admin memilih fungsi tambah pengguna baru, jalankan <i>subflow</i> Menambah data pengguna. Jika admin memilih opsi mengubah data pengguna, jalankan <i>subflow</i> Mengubah data pengguna. Jika admin memilih opsi menghapus data pengguna, jalankan <i>subflow</i> Menghapus data pengguna. {Use case selesai} 3. <i>Use case</i> selesai.
Alternative Flows	1. Terdapat kolom formulir penambahan data pengguna yang belum diisi Pada <i>subflow</i> menambahkan data pengguna saat {menyimpan data pengguna} sistem mendeteksi formulir belum diisi dengan lengkap. Sistem akan menampilkan notifikasi lengkapi formulir, kemudian dilanjutkan {Mengisi formulir pendaftaran pengguna}. 2. <i>Username</i> pengguna sudah ada / sudah digunakan Pada <i>subflow</i> menambahkan data pengguna saat {menyimpan data pengguna} sistem mendeteksi <i>username</i> yang dimasukkan oleh pengguna sudah digunakan. Sistem akan menampilkan notifikasi <i>username</i> sudah digunakan, kemudian {mengisi formulir pendaftaran pengguna} pada <i>subflow</i> menambahkan data pengguna. 3. Batal melakukan pendaftaran pengguna Pada <i>subflow</i> menambahkan data pengguna setelah {mengisi formulir pendaftaran pengguna} admin membatalkan pendaftaran pengguna. Sistem akan mereset formulir pendaftaran pengguna, kemudian {<i>use case</i> selesai}. 4. Terdapat kolom formulir edit data pengguna yang belum diisi Pada <i>subflow</i> mengubah data pengguna saat {menyimpan data pengguna} sistem mendeteksi ada kolom formulir yang kosong. Sistem akan menampilkan notifikasi lengkapi formulir edit data pengguna, kemudian {mengubah data pengguna} pada <i>subflow</i> mengubah data pengguna. 5. Batal melakukan edit data pengguna Pada <i>subflow</i> mengubah data pengguna sebelum {menyimpan data pengguna} admin melakukan batal edit pengguna. Sistem menutup/formulir edit pengguna, kemudian {<i>use case</i> selesai}.

	6. Batal melakukan hapus data pengguna Pada <i>subflow</i> menghapus data pengguna saat {konfirmasi hapus data pengguna} admin memilih batal hapus data pengguna. Sistem menutup notifikasi konfirmasi hapus data pengguna, kemudian {use case selesai}.
<i>Subflow</i>	1. <u>Menambahkan data pengguna</u> {Memilih tambah pengguna} - Admin memilih tambah data pengguna. {Mengisi formulir pendaftaran pengguna} - Sistem menampilkan formulir pendaftaran pengguna. - Admin mengisi formulir pendaftaran pengguna. {Menyimpan data pengguna} - Admin memilih simpan data pengguna. - Sistem menambahkan/memasukkan data baru pada penyimpanan/database. {Menampilkan notifikasi berhasil} - Sistem menampilkan notifikasi data pengguna berhasil ditambahkan. 2. <u>Mengubah data pengguna</u> {Memilih edit data pengguna} - Admin memilih pengguna yang akan diubah datanya. - Admin memilih edit data pengguna. {Mengubah data pengguna} - Sistem akan menampilkan formulir perubahan data. - Admin mengubah data pengguna. {Menyimpan data pengguna} - Admin memilih simpan data pengguna. - Sistem menyimpan data pengguna yang telah diubah. {Menampilkan notifikasi berhasil} - Sistem menampilkan notifikasi data pengguna berhasil diubah. 3. <u>Menghapus data pengguna</u> {Memilih hapus data pengguna} - Admin memilih data pengguna yang akan dihapus. - Admin memilih hapus data pengguna. {Konfirmasi hapus data pengguna} - Sistem menampilkan notifikasi konfirmasi penghapusan data pengguna.

	d. Admin melakukan konfirmasi penghapusan data pengguna. {Menghapus data pengguna} e. Sistem menghapus data pengguna. {Menampilkan notifikasi berhasil} f. Sistem menampilkan notifikasi data pengguna berhasil dihapus.
Post-Condition	1. Sistem menampilkan daftar pengguna yang terdaftar di sistem. 2. Ketika admin melakukan tambah data pengguna, data baru berhasil masuk ke sistem. 3. Ketika admin melakukan ubah data pengguna, data pengguna yang dipilih berhasil diubah sesuai dengan permintaan perubahan. 4. Ketika admin melakukan hapus data pengguna, data pengguna yang dipilih berhasil dihapus dari sistem.

Glossary

Data pengguna yang dimaksud adalah data dari pengguna sistem yang terdiri dari : *username*, *password*, nama lengkap, jabatan, status pengguna di sistem.

3. Kelola pendataan wajib pajak

Spesifikasi *use case* kelola pendataan wajib pajak dijelaskan pada Tabel 4.14 berikut ini.

Tabel 4.14 Spesifikasi Use Case Kelola Pendataan Wajib Pajak

Name	Kelola pendataan wajib pajak
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana data wajib pajak ditambahkan, diubah dan dihapus.
Pre-Condition	1. Bidang pendataan dan penetapan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang pendataan dan penetapan telah berhasil masuk ke sistem.
Basic Flow	{Memilih kelola wajib pajak} 1. <i>Use case</i> dimulai ketika Bidang pendataan dan penetapan memilih opsi kelola wajib pajak. {Memilih opsi pengelolaan wajib pajak} 2. Sistem akan menampilkan daftar wajib pajak yang terdaftar di sistem disertai opsi mengubah data wajib pajak ataupun menghapus data wajib pajak dan menyediakan fungsi tambah wajib pajak baru.

	Jika Bidang pendataan dan penetapan memilih fungsi tambah wajib pajak baru, jalankan <i>subflow</i> Menambahkan wajib pajak baru. Jika Bidang pendataan dan penetapan memilih opsi mengubah data wajib pajak, jalankan <i>subflow</i> Mengubah data wajib pajak. Jika Bidang pendataan dan penetapan memilih opsi menghapus data wajib pajak, jalankan <i>subflow</i> Menghapus data wajib pajak. {Use case selesai} 3. <i>Use case</i> selesai.
Alternative Flows	1. Terdapat kolom formulir penambahan data wajib pajak yang belum diisi Pada <i>subflow</i> menambahkan wajib pajak baru saat {menyimpan data wajib pajak} sistem mendeteksi formulir belum diisi dengan lengkap. Sistem akan menampilkan notifikasi lengkapi formulir, kemudian dilanjutkan {Mengisi formulir pendaftaran wajib pajak}. 2. Wajib pajak yang baru sudah terdaftar di sistem Pada <i>subflow</i> menambahkan wajib pajak baru saat {menyimpan data wajib pajak} sistem mendeteksi NPWPD yang dimasukkan oleh wajib pajak sudah digunakan. Sistem akan menampilkan notifikasi NPWPD sudah digunakan, kemudian {Mengisi formulir pendaftaran wajib pajak} pada <i>subflow</i> menambahkan wajib pajak baru. 3. Batal melakukan pendaftaran wajib pajak Pada <i>subflow</i> menambahkan wajib pajak baru setelah {mengisi formulir pendaftaran wajib pajak} bidang pendataan dan penetapan membatalkan pendaftaran wajib pajak baru. Sistem akan mereset formulir pendaftaran wajib pajak, kemudian {use case selesai}. 4. Terdapat kolom formulir edit data wajib pajak yang belum diisi Pada <i>subflow</i> mengubah data wajib pajak saat {menyimpan perubahan} sistem mendeteksi ada kolom formulir yang kosong. Sistem akan menampilkan notifikasi lengkapi formulir edit data wajib pajak, kemudian {mengubah data wajib pajak} pada <i>subflow</i> mengubah data wajib pajak. 5. Batal melakukan edit data wajib pajak Pada <i>subflow</i> mengubah data wajib pajak sebelum {menyimpan perubahan} bidang pendataan dan

	penetapan melakukan batal edit data wajib pajak. Sistem menutup/formulir edit data wajib pajak, kemudian {<i>use case</i> selesai}. 6. Batal melakukan hapus data wajib pajak Pada <i>subflow</i> menghapus data wajib pajak saat {konfirmasi hapus data wajib pajak} bidang pendataan dan penetapan memilih batal hapus data wajib pajak. Sistem menutup notifikasi konfirmasi hapus data wajib pajak, kemudian {<i>use case</i> selesai}.
Subflow	1. <u>Menambahkan wajib pajak baru</u> {Memilih tambah wajib pajak} a. Bidang pendataan dan penetapan memilih tambah wajib pajak baru. {Mengisi formulir pendaftaran wajib pajak} b. Sistem menampilkan formulir pendaftaran wajib pajak. c. Bidang pendataan dan penetapan mengisi formulir pendaftaran wajib pajak baru. {Menyimpan data wajib pajak} d. Bidang pendataan dan penetapan memilih simpan data wajib pajak. e. Sistem menambahkan/memasukkan data wajib pajak baru pada penyimpanan/<i>database</i>. {Menampilkan notifikasi berhasil} f. Sistem menampilkan notifikasi data wajib pajak berhasil ditambahkan. 2. <u>Mengubah data wajib pajak</u> {Memilih edit data wajib pajak} a. Bidang pendataan dan penetapan memilih data wajib pajak yang akan diubah datanya. b. Bidang pendataan dan penetapan memilih edit data wajib pajak. {Mengubah data wajib pajak} c. Sistem akan menampilkan formulir pengubahan data wajib pajak. d. Bidang pendataan dan penetapan mengubah data wajib pajak. {Menyimpan perubahan} e. Bidang pendataan dan penetapan memilih simpan data wajib pajak. f. Sistem menyimpan data wajib pajak yang telah diubah. {Menampilkan notifikasi berhasil}

	g. Sistem menampilkan notifikasi data wajib pajak berhasil diubah. 3. Menghapus data wajib pajak {Memilih hapus data wajib pajak} a. Bidang pendataan dan penetapan memilih data wajib pajak yang akan dihapus. b. Bidang pendataan dan penetapan memilih hapus data wajib pajak. {Konfirmasi hapus data} c. Sistem menampilkan notifikasi konfirmasi penghapusan data wajib pajak. d. Bidang pendataan dan penetapan melakukan konfirmasi penghapusan data wajib pajak. {Menghapus data wajib pajak} e. Sistem menghapus data wajib pajak. {Menampilkan notifikasi berhasil} f. Sistem menampilkan notifikasi data wajib pajak berhasil dihapus.
Post-Condition	1. Sistem menampilkan daftar wajib pajak yang terdaftar di sistem. 2. Ketika bidang penetapan dan pendataan melakukan tambah wajib pajak, data baru berhasil masuk ke sistem. 3. Ketika bidang penetapan dan pendataan melakukan ubah data wajib pajak, data wajib pajak yang dipilih berhasil diubah sesuai dengan permintaan perubahan. 4. Ketika bidang penetapan dan pendataan melakukan hapus data wajib pajak, data wajib pajak yang dipilih berhasil dihapus dari sistem.

Glossary

Data wajib pajak yang dimaksud adalah data dari orang yang wajib membayar pajak, terdiri dari : NPWPD (Nomor Pokok Wajib Pajak Daerah), nama wajib pajak, nomor HP WP, alamat wajib pajak (kabupaten/kodya, kecamatan dan desa/kelurahan).

4. Kelola pendataan objek pajak

Spesifikasi *use case* kelola pendataan objek pajak dijelaskan pada Tabel 4.15 berikut ini.

Tabel 4.15 Spesifikasi Use Case Kelola Pendataan Objek Pajak

Name	Kelola pendataan objek pajak
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana data objek pajak ditambahkan, diubah dan dihapus.
Pre-Condition	1. Bidang pendataan dan penetapan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang pendataan dan penetapan telah berhasil masuk ke sistem.
Basic Flow	{Memilih kelola pajak} 1. <i>Use case</i> dimulai Bidang pendataan dan penetapan memilih opsi kelola data objek pajak. {Memilih opsi pengelolaan pajak} 2. Sistem akan menampilkan daftar/data objek pajak yang terdata di sistem disertai opsi mengubah data objek pajak ataupun menghapus data objek pajak dan menyediakan fungsi tambah objek pajak baru. Jika Bidang pendataan dan penetapan memilih fungsi tambah objek pajak baru, jalankan <i>subflow</i> Menambah data objek pajak. Jika Bidang pendataan dan penetapan memilih opsi mengubah objek pajak, jalankan <i>subflow</i> Mengubah data objek pajak. Jika Bidang pendataan dan penetapan memilih opsi menghapus objek pajak, jalankan <i>subflow</i> Menghapus data objek pajak. {Use case selesai} 3. <i>Use case</i> selesai.
Alternative Flows	1. Terdapat kolom formulir penambahan data objek pajak yang belum diisi Pada <i>subflow</i> menambahkan objek pajak baru saat {menyimpan data objek pajak} sistem mendeteksi formulir belum diisi dengan lengkap. Sistem akan menampilkan notifikasi lengkapi formulir, kemudian dilanjutkan {Mengisi formulir pendaftaran objek pajak}. 2. Objek pajak sudah ada dalam sistem Pada <i>subflow</i> menambahkan objek pajak baru saat {menyimpan data objek pajak} sistem mendeteksi id objek pajak yang dimasukkan sudah digunakan.

	Sistem akan menampilkan notifikasi id objek pajak sudah digunakan, kemudian {Mengisi formulir pendaftaran objek pajak} pada <i>subflow</i> menambahkan objek pajak baru. 3. Batal melakukan pendataan objek pajak Pada <i>subflow</i> menambahkan objek pajak baru setelah {mengisi formulir pendaftaran objek pajak} bidang pendataan dan penetapan membatalkan pendaftaran objek pajak baru. Sistem akan mereset formulir pendaftaran objek pajak, kemudian {<i>use case</i> selesai}. 4. Terdapat kolom formulir edit data objek pajak yang belum diisi Pada <i>subflow</i> mengubah data objek pajak saat {menyimpan data objek pajak} sistem mendeteksi ada kolom formulir yang kosong. Sistem akan menampilkan notifikasi lengkapi formulir edit data objek pajak, kemudian {mengubah data objek pajak} pada <i>subflow</i> mengubah data objek pajak. 5. Batal melakukan edit objek data pajak Pada <i>subflow</i> mengubah data objek pajak sebelum {menyimpan data objek pajak} bidang pendataan dan penetapan melakukan batal edit data objek pajak. Sistem menutup/formulir edit data objek pajak, kemudian {<i>use case</i> selesai}. 6. Batal melakukan hapus data objek pajak Pada <i>subflow</i> menghapus data objek pajak saat {konfirmasi hapus data objek pajak} bidang pendataan dan penetapan memilih batal hapus data objek pajak. Sistem menutup notifikasi konfirmasi hapus data objek pajak, kemudian {<i>use case</i> selesai}.
Subflow	1. Menambahkan objek pajak baru {Memilih tambah data objek pajak} a. Bidang pendataan dan penetapan memilih tambah data objek pajak baru. {Mengisi formulir pendaftaran objek pajak} b. Sistem menampilkan formulir pendaftaran objek pajak baru. c. Bidang pendataan dan penetapan mengisi formulir pendaftaran objek pajak baru. {Menyimpan data objek pajak} d. Bidang pendataan dan penetapan memilih simpan data objek pajak baru. e. Sistem menambahkan/memasukkan data objek pajak baru pada penyimpanan/database.

	{Menampilkan notifikasi berhasil} f. Sistem menampilkan notifikasi data objek pajak berhasil ditambahkan. 2. Mengubah data objek pajak {Memilih edit data objek pajak} a. Bidang pendataan dan penetapan memilih data objek pajak yang akan diubah datanya. b. Bidang pendataan dan penetapan memilih edit data objek pajak. {Mengubah data objek pajak} c. Sistem akan menampilkan formulir perubahan data objek pajak. d. Bidang pendataan dan penetapan mengubah data objek pajak. {Menyimpan data objek pajak} e. Bidang pendataan dan penetapan memilih simpan data objek pajak. f. Sistem menyimpan data objek pajak yang telah diubah. {Menampilkan notifikasi berhasil} g. Sistem menampilkan notifikasi data objek pajak berhasil diubah. 3. Menghapus data objek pajak {Memilih hapus data objek pajak} a. Bidang pendataan dan penetapan memilih data objek pajak yang akan dihapus. b. Bidang pendataan dan penetapan memilih hapus data objek pajak. {Konfirmasi hapus data objek pajak} c. Sistem menampilkan konfirmasi penghapusan data objek pajak. d. Bidang pendataan dan penetapan melakukan konfirmasi penghapusan data objek pajak. {Menghapus data objek pajak} e. Sistem menghapus data objek pajak. {Menampilkan notifikasi berhasil} f. Sistem menampilkan notifikasi data objek pajak berhasil dihapus.
Post-Condition	1. Sistem menampilkan daftar objek pajak yang terdaftar di sistem. 2. Ketika bidang penetapan dan pendataan melakukan tambah objek pajak, data objek baru berhasil masuk ke sistem.

	3. Ketika bidang penetapan dan pendataan melakukan ubah data objek pajak, data objek pajak yang dipilih berhasil diubah sesuai dengan permintaan perubahan. 4. Ketika bidang penetapan dan pendataan melakukan hapus data objek pajak, data objek pajak yang dipilih berhasil dihapus dari sistem.
--	---

Glossary

Data objek pajak yang dimaksud adalah data dari objek milik wajib pajak yang dikenai pajak, terdiri dari : nama objek pajak, alamat objek pajak (kabupaten/kodya, kecamatan dan desa/kelurahan).

5. Kelola penetapan pajak

Spesifikasi *use case* kelola penetapan pajak dijelaskan pada Tabel 4.16 berikut ini.

Tabel 4.16 Spesifikasi Use Case Kelola Penetapan Pajak

Name	Kelola penetapan pajak
Pre-Condition	1. Bidang pendataan dan penetapan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang pendataan dan penetapan telah berhasil masuk ke sistem.
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana bidang penetapan dan pendataan melakukan penetapan pajak.
Basic Flow	{Memilih kelola penetapan pajak} 1. <i>Use case</i> dimulai ketika Bidang pendataan dan penetapan memilih opsi kelola penetapan pajak. {Memilih opsi pengelolaan pajak} 2. Sistem meminta data filter pencarian data pajak. 3. Bidang pendataan dan penetapan memasukkan data filter pencarian data pajak. 4. Bidang pendataan dan penetapan memilih cari 5. Sistem akan menampilkan daftar/data ketetapan pajak yang terdata di sistem sesuai filter disertai opsi mengubah data ketetapan pajak ataupun menghapus data ketetapan pajak dan menyediakan fungsi tambah penetapan pajak baru. Jika Bidang pendataan dan penetapan memilih fungsi tambah data ketetapan pajak baru, jalankan <i>subflow</i> Menambah data ketetapan pajak.

	Jika Bidang pendataan dan penetapan memilih opsi mengubah ketentuan pajak, jalankan <i>subflow</i> Mengubah data ketentuan pajak. Jika Bidang pendataan dan penetapan memilih opsi menghapus ketentuan pajak, jalankan <i>subflow</i> Menghapus data ketentuan pajak. {Use case selesai} 6. <i>Use case</i> selesai.
Alternative Flows	1. Terdapat kolom formulir penambahan data ketentuan pajak yang belum diisi Pada <i>subflow</i> menambahkan ketentuan pajak baru saat {menyimpan data ketentuan pajak} sistem mendeteksi formulir belum diisi dengan lengkap. Sistem akan menampilkan notifikasi lengkapi formulir, kemudian dilanjutkan {Mengisi formulir penetapan pajak}. 2. Ketentuan pajak sudah ada dalam sistem Pada <i>subflow</i> menambahkan ketentuan pajak baru saat {menyimpan data ketentuan pajak} sistem mendeteksi id ketentuan pajak sudah digunakan. Sistem akan menampilkan notifikasi id ketentuan pajak sudah digunakan, kemudian {Mengisi formulir penetapan pajak} pada <i>subflow</i> menambahkan ketentuan pajak baru. 3. Batal melakukan penetapan pajak Pada <i>subflow</i> menambahkan ketentuan pajak baru setelah {mengisi formulir penetapan pajak} bidang pendataan dan penetapan membatalkan penetapan pajak baru. Sistem akan mereset formulir penetapan pajak, kemudian {<i>use case</i> selesai}. 4. Terdapat kolom formulir edit data ketentuan pajak yang belum diisi Pada <i>subflow</i> mengubah data objek pajak saat {menyimpan data ketentuan pajak} sistem mendeteksi ada kolom formulir yang kosong. Sistem akan menampilkan notifikasi lengkapi formulir edit data ketentuan pajak, kemudian {mengubah data ketentuan pajak} pada <i>subflow</i> mengubah data ketentuan pajak. 5. Batal melakukan edit ketentuan pajak Pada <i>subflow</i> mengubah ketentuan pajak sebelum {menyimpan ketentuan pajak} bidang pendataan dan penetapan melakukan batal edit ketentuan pajak.

	Sistem menutup/formulir edit ketetapan pajak, kemudian {<i>use case</i> selesai}. 6. Batal melakukan hapus ketetapan pajak Pada <i>subflow</i> menghapus ketetapan pajak saat {konfirmasi hapus ketetapan pajak} bidang pendataan dan penetapan memilih batal hapus ketetapan pajak. Sistem menutup notifikasi konfirmasi hapus data objek pajak, kemudian {<i>use case</i> selesai}.
Subflow	1. <u>Menambahkan ketetapan pajak baru</u> {Memilih tambah ketetapan pajak} a. Bidang pendataan dan penetapan memilih tambah ketetapan pajak baru. {Mengisi formulir penetapan pajak} b. Sistem menampilkan formulir penetapan pajak baru. c. Bidang pendataan dan penetapan mengisi formulir penetapan pajak baru. {Menyimpan data ketetapan pajak} d. Bidang pendataan dan penetapan memilih simpan ketetapan pajak baru. e. Sistem menambahkan/memasukkan data ketetapan pajak baru pada penyimpanan/<i>database</i>. {Menampilkan notifikasi berhasil} f. Sistem menampilkan notifikasi data ketetapan pajak berhasil ditambahkan. 2. <u>Mengubah ketetapan pajak</u> {Memilih edit ketetapan pajak} a. Bidang pendataan dan penetapan memilih ketetapan pajak yang akan diubah datanya. b. Bidang pendataan dan penetapan memilih edit ketetapan pajak. {Mengubah ketetapan pajak} c. Sistem akan menampilkan formulir pengubahan ketetapan pajak. d. Bidang pendataan dan penetapan mengubah ketetapan pajak. {Menyimpan ketetapan pajak} e. Bidang pendataan dan penetapan memilih simpan ketetapan pajak. f. Sistem menyimpan ketetapan pajak yang telah diubah. {Menampilkan notifikasi berhasil}

	g. Sistem menampilkan notifikasi penetapan pajak berhasil diubah. 3. Menghapus penetapan pajak {Memilih hapus penetapan pajak} a. Bidang pendataan dan penetapan memilih penetapan pajak yang akan dihapus. b. Bidang pendataan dan penetapan memilih hapus penetapan pajak. {Konfirmasi hapus penetapan pajak} c. Sistem menampilkan konfirmasi penghapusan penetapan pajak. d. Bidang pendataan dan penetapan melakukan konfirmasi penghapusan penetapan pajak. {Menghapus penetapan pajak} e. Sistem menghapus penetapan pajak. {Menampilkan notifikasi berhasil} f. Sistem menampilkan notifikasi penetapan pajak berhasil dihapus.
Post-Condition	1. Sistem menampilkan daftar penetapan pajak yang terdata di sistem. 2. Ketika bidang penetapan dan pendataan melakukan tambah penetapan pajak, penetapan pajak baru berhasil masuk ke sistem. 3. Ketika bidang penetapan dan pendataan melakukan ubah penetapan pajak, penetapan pajak yang dipilih berhasil diubah sesuai dengan permintaan perubahan. 4. Ketika bidang penetapan dan pendataan melakukan hapus penetapan pajak, penetapan pajak yang dipilih berhasil dihapus dari sistem.

Glossary

Data penetapan pajak yang dimaksud adalah data penetapan pembayaran pajak dari suatu objek pajak, terdiri dari : Nomor penetapan, tahun pajak, periode pajak, tanggal jatuh tempo, kode rekening, tarif pajak dalam persen, jumlah penetapan pajak. Untuk pajak air tanah perlu ada variabel/data tambahan yaitu idx, harga per m3, volume tonase dan tipe pajak air tanah. Untuk pajak parkir perlu ada variabel/data tambahan yaitu jumlah omset per bulan.

6. Cetak SKPD

Spesifikasi *use case* cetak SKPD dijelaskan pada Tabel 4.17 berikut ini.

Tabel 4.17 Spesifikasi Use Case Cetak SKPD

Name	Cetak SKPD
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana sistem mencetak SKPD yang telah ditetapkan
Pre-Condition	1. Bidang pendataan dan penetapan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang pendataan dan penetapan telah berhasil masuk ke sistem.
Basic Flow	{Memilih opsi cetak SKPD} 1. <i>Use case</i> dimulai ketika Bidang pendataan dan penetapan memilih opsi cetak SKPD. {Mengisi formulir cetak SKPD} 2. Sistem menyediakan formulir SKPD mana saja yang akan dicetak. 3. Bidang pendataan dan penetapan mengisi formulir. {Konfirmasi cetak} 4. Bidang pendataan dan penetapan melakukan konfirmasi formulir. {Mencetak SKPD} 5. Sistem mencetak SKPD. 6. Sistem mengubah status SKPD menjadi tercetak. {Use case selesai} 7. <i>Use case</i> selesai.
Alternative Flows	1. Formulir cetak SKPD tidak diisi dengan lengkap Pada saat {mengisi formulir cetak SKPD} ada kolom yang tidak diisi. Sistem akan menampilkan pesan untuk melengkapi formulir, kemudian menjalankan {mengisi formulir cetak SKPD}. 2. Batal mencetak SKPD Setelah {mengisi formulir cetak SKPD} Bidang pendataan dan penetapan tidak mengkonfirmasi cetak SKPD atau membatalkan cetak SKPD, kemudian <i>use case</i> selesai.
Subflow	-
Post-Condition	1. Sistem mencetak SKPD sesuai kebutuhan pengguna. 2. Status SKPD menjadi tercetak.

Glossary

SKPD atau Surat Ketetapan Pajak Daerah merupakan surat yang berisi ketetapan pajak atas objek pajak milik wajib pajak yang diterbitkan oleh bidang pendataan dan penetapan untuk diserahkan ke wajib pajak sebagai pemberitahuan pembayaran pajak. Data SKPD pajak air tanah terdiri dari : nomor SKPD, periode pajak, NPWPD, Nama WP, Nama Objek Pajak, Alamat Objek pajak (kabupaten/kodya, kecamatan dan desa/kelurahan), tanggal jatuh tempo, tarif pajak dalam persen (%), kode rekening, idx, harga per m³, volume tonase, tipe pajak air tanah, dan jumlah ketetapan pajak. Data SKPD dari pajak parkir terdiri dari : nomor SKPD, periode pajak, NPWPD, Nama WP, Nama Objek Pajak, Alamat Objek pajak (kabupaten/kodya, kecamatan dan desa/kelurahan), tanggal jatuh tempo, tarif pajak dalam persen (%), kode rekening, jumlah omzet dan jumlah ketetapan pajak.

7. Lihat laporan penerimaan pajak

Spesifikasi *use case* lihat laporan penerimaan pajak dijelaskan pada Tabel 4.18 berikut ini.

Tabel 4.18 Spesifikasi Use Case Lihat Laporan Penerimaan Pajak

Name	Lihat laporan penerimaan pajak
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana sistem menyajikan data mengenai penerimaan pajak berdasarkan hari dan/atau berdasarkan jenis pajak.
Pre-Condition	1. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan telah berhasil masuk ke sistem
Basic Flow	{Memilih opsi penerimaan pajak} 1. <i>Use case</i> dimulai ketika Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan memilih opsi penerimaan pajak. {Mengisi formulir} 2. Sistem menyediakan formulir filter. 3. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan mengisi formulir filter. {Konfirmasi formulir} 4. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan memilih <i>submit</i> laporan penerimaan. {Menampilkan laporan}

	5. Sistem menampilkan laporan penerimaan pajak. {Use case selesai} 6. <i>Use case</i> selesai.
Alternative Flows	-
Subflow	-
Post-Condition	1. Sistem menampilkan laporan penerimaan pajak sesuai kebutuhan Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan.

Glossary

Laporan penerimaan pajak merupakan laporan atas pajak-pajak yang telah dibayar oleh wajib pajak. Tabel laporan penerimaan pajak terdiri dari kolom jenis pajak, objek pajak, kecamatan dari objek pajak, kelurahan/desa dari objek pajak, tanggal bayar pajak, dan jumlah pembayaran pajak. Kemudian jumlah bayar akan dihitung totalnya untuk ditampilkan. Laporan penerimaan pajak ini dapat difilter berdasarkan alamat, jenis pajak, dan tanggal pembayaran.

8. Cetak laporan penerimaan pajak

Spesifikasi *use case* cetak laporan penerimaan pajak dijelaskan pada Tabel 4.19 berikut ini.

Tabel 4.19 Spesifikasi Use Case Cetak Laporan Penerimaan Pajak

Name	Cetak laporan penerimaan pajak
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana mencetak laporan penerimaan pajak (pajak yang berstatus terbayar)
Pre-Condition	1. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan telah berhasil masuk ke sistem
Basic Flow	{Memilih opsi penerimaan pajak} 1. <i>Use case</i> dimulai ketika Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan memilih opsi penerimaan pajak. {Mengisi formulir} 2. Sistem menyediakan formulir filter. 3. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan mengisi formulir filter. {Konfirmasi formulir}

	4. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan memilih <i>submit</i> laporan penerimaan. {Menampilkan laporan} 5. Sistem menampilkan laporan penerimaan pajak. {Memilih cetak laporan penerimaan} 6. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan memilih cetak laporan penerimaan pajak. {Preview print} 7. Muncul <i>preview</i> laporan penerimaan pajak. {Mencetak laporan} 8. Kas daerah, bidang pembukuan dan pelaporan atau bidang penagihan dan keberatan memilih <i>print</i> 9. Sistem mencetak laporan penerimaan. {Use case selesai} 10. <i>Use case</i> selesai.
Alternative Flows	-
Subflow	-
Post-Condition	1. Laporan penerimaan pajak tercetak.

Glossary
Laporan penerimaan pajak merupakan laporan atas pajak-pajak yang telah dibayar oleh wajib pajak. Tabel laporan penerimaan pajak terdiri dari kolom jenis pajak, objek pajak, kecamatan dari objek pajak, kelurahan/desa dari objek pajak, tanggal bayar pajak, dan jumlah pembayaran pajak. Kemudian jumlah bayar akan dihitung totalnya untuk ditampilkan. Laporan penerimaan pajak ini dapat difilter berdasarkan alamat, jenis pajak, dan tanggal pembayaran.

9. Lihat buku register

Spesifikasi *use case* lihat buku register dijelaskan pada Tabel 4.20 berikut ini.

Tabel 4.20 Spesifikasi Use Case Lihat Buku Register

Name	Lihat buku register
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana sistem menyajikan data SKPD yang disertai status penyerahan (diserahkan atau belum) dan status pembayaran SKPD (terbayar atau belum).
Pre-Condition	1. Bidang penagihan dan keberatan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang penagihan dan keberatan telah berhasil masuk ke sistem

Basic Flow	{Memilih opsi buku register} 1. <i>Use case</i> dimulai ketika Bidang penagihan dan keberatan memilih opsi buku register. {Mengisi formulir} 2. Sistem menyediakan formulir filter. 3. Bidang penagihan dan keberatan mengisi formulir filter. {Melakukan konfirmasi} 4. Bidang penagihan dan keberatan memilih <i>submit</i> buku register. {Menampilkan buku register} 5. Sistem menampilkan buku register. {Use case selesai} 6. <i>Use case</i> selesai.
Alternative Flows	-
Subflow	-
Post-Condition	1. Sistem menampilkan buku register sesuai kebutuhan pengguna.

Glossary
Buku register merupakan buku yang berisi data-data ketetapan pajak yang telah didata oleh bidang pendataan dan penetapan yang memiliki status terbayar. Tabel buku register ini terdiri dari Nama wajib pajak, nomor HP wajib pajak, alamat objek pajak (kabupaten/kodya, kecamatan, kelurahan/desa), jenis pajak, tanggal jatuh tempo, status penyerahan SKPD ke wajib pajak, dan status pembayaran pajak. Buku register ini dapat difilter berdasarkan alamat, jenis pajak, tanggal jatuh tempo, tanggal terbit dan status pembayaran pajak.

10. Cetak buku register

Spesifikasi *use case* cetak buku register dijelaskan pada Tabel 4.21 berikut ini.

Tabel 4.21 Spesifikasi Use Case Cetak Buku Register

Name	Cetak buku register
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana mencetak buku register
Pre-Condition	1. Bidang penagihan dan keberatan menggunakan perangkat yang terhubung dengan jaringan internet 2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang penagihan dan keberatan telah berhasil masuk ke sistem
Basic Flow	{Memilih opsi buku register} 1. <i>Use case</i> dimulai ketika Bidang penagihan dan keberatan memilih opsi buku register.

	{Mengisi formulir} 2. Sistem menyediakan formulir filter. 3. Bidang penagihan dan keberatan mengisi formulir filter. {Melakukan konfirmasi} 4. Bidang penagihan dan keberatan memilih <i>submit</i> buku register. {Menampilkan buku register} 5. Sistem menampilkan buku register. {Memilih cetak buku register} 6. Bidang penagihan dan keberatan memilih cetak buku register. {Preview print} 7. Muncul <i>preview</i> buku register. {Memilih print} 8. Bidang penagihan dan keberatan memilih <i>print</i>. {Mencetak buku register} 9. Sistem mencetak buku register. {Use case selesai} 10. <i>Use case</i> selesai.
Alternative Flows	-
Subflow	-
Post-Condition	1. Buku register tercetak.

Glossary

Buku register merupakan buku yang berisi data-data ketetapan pajak yang telah didata oleh bidang pendataan dan penetapan yang memiliki status terbayar. Tabel buku register ini terdiri dari Nama wajib pajak, nomor HP wajib pajak, alamat objek pajak (kabupaten/kodya, kecamatan, kelurahan/desa), jenis pajak, tanggal jatuh tempo, status penyerahan SKPD ke wajib pajak, dan status pembayaran pajak. Buku register ini dapat difilter berdasarkan alamat, jenis pajak, tanggal jatuh tempo, tanggal terbit dan status pembayaran pajak.

11. Lihat buku ekspedisi

Spesifikasi *use case* lihat buku ekspedisi dijelaskan pada Tabel 4.22 berikut ini.

Tabel 4.22 Spesifikasi Use Case Lihat Buku Ekspedisi

Name	Lihat buku ekspedisi
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana sistem menyajikan data SKPD yang telah tercetak disertai status penyerahan (sudah diserahkan atau belum diserahkan) kepada wajib pajak.

Pre-Condition	1. Bidang penagihan dan keberatan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang penagihan dan keberatan telah berhasil masuk ke sistem
Basic Flow	{Memilih opsi buku ekspedisi} 1. <i>Use case</i> dimulai ketika Bidang penagihan dan keberatan memilih opsi buku ekspedisi. {Mengisi formulir} 2. Sistem menyediakan formulir filter. 3. Bidang penagihan dan keberatan mengisi formulir filter. {Melakukan konfirmasi} 4. Bidang penagihan dan keberatan memilih <i>submit</i> buku ekspedisi. {Menampilkan buku ekspedisi} 5. Sistem menampilkan buku ekspedisi. {Use case selesai} 6. <i>Use case</i> selesai.
Alternative Flows	-
Subflow	-
Post-Condition	1. Sistem menampilkan buku ekspedisi sesuai kebutuhan pengguna.

Glossary

Buku ekspedisi merupakan buku yang berisi bukti bahwa SKPD (Surat Ketetapan Pajak Daerah) yang dicetak oleh bidang pendataan dan penetapan sudah diserahkan ke wajib pajak oleh bidang penagihan dan keberatan. Tabel buku ekspedisi terdiri dari Nama wajib pajak, nama objek pajak, kecamatan objek pajak, kelurahan/desa objek pajak, tanggal jatuh tempo, dan tombol untuk melakukan validasi penyerahan SKPD. Buku ekspedisi ini dapat difilter berdasarkan alamat, jenis pajak, tanggal jatuh tempo, dan tanggal terbit.

12. Cetak buku ekspedisi

Spesifikasi *use case* cetak buku ekspedisi dijelaskan pada Tabel 4.23 berikut ini.

Tabel 4.23 Spesifikasi Use Case Cetak Buku Ekspedisi

Name	Cetak buku ekspedisi
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana mencetak buku ekspedisi.
Pre-Condition	1. Bidang penagihan dan keberatan menggunakan perangkat yang terhubung dengan jaringan internet.

	2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang penagihan dan keberatan telah berhasil masuk ke sistem
Basic Flow	{Memilih opsi buku ekspedisi} 1. <i>Use case</i> dimulai ketika Bidang penagihan dan keberatan memilih opsi buku ekspedisi. {Mengisi formulir} 2. Sistem menyediakan formulir filter. 3. Bidang penagihan dan keberatan mengisi formulir. {Melakukan konfirmasi} 4. Bidang penagihan dan keberatan memilih <i>submit</i> buku ekspedisi. {Menampilkan buku ekspedisi} 5. Sistem menampilkan buku ekspedisi. {Memilih cetak buku ekspedisi} 6. Bidang penagihan dan keberatan memilih cetak buku ekspedisi. {Preview print} 7. Muncul <i>preview</i> buku ekspedisi. {Memilih print} 8. Bidang penagihan dan keberatan memilih <i>print</i>. {Mencetak buku ekspedisi} 9. Sistem mencetak buku ekspedisi. {Use case selesai} 10. <i>Use case</i> selesai.
Alternative Flows	-
Subflow	-
Post-Condition	1. Buku register tercetak.

Glossary

Buku ekspedisi merupakan buku yang berisi bukti bahwa SKPD (Surat Ketetapan Pajak Daerah) yang dicetak oleh bidang pendataan dan penetapan sudah diserahkan ke wajib pajak oleh bidang penagihan dan keberatan. Tabel buku ekspedisi terdiri dari Nama wajib pajak, nama objek pajak, kecamatan objek pajak, kelurahan/desa objek pajak, tanggal jatuh tempo, dan tombol untuk melakukan validasi penyerahan SKPD. Buku ekspedisi ini dapat difilter berdasarkan alamat, jenis pajak, tanggal jatuh tempo, dan tanggal terbit.

13. Validasi penyerahan SKPD

Spesifikasi *use case* validasi penyerahan SKPD dijelaskan pada Tabel 4.24 berikut ini.

Tabel 4.24 Spesifikasi Use Case Validasi Penyerahan SKPD

Name	Validasi penyerahan SKPD
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana melakukan validasi penyerahan SKPD pada wajib pajak atau bagaimana sistem mengubah status SKPD menjadi diserahkan.
Pre-Condition	1. Bidang penagihan dan keberatan menggunakan perangkat yang terhubung dengan jaringan internet. 2. Sistem tidak sedang dalam proses perbaikan. 3. Bidang penagihan dan keberatan telah berhasil masuk ke sistem
Basic Flow	{Memilih opsi buku ekspedisi} 1. <i>Use case</i> dimulai ketika Bidang penagihan dan keberatan memilih opsi buku ekspedisi. {Mengisi formulir} 2. Sistem menyediakan formulir filter. 3. Bidang penagihan dan keberatan mengisi formulir filter. {Melakukan konfirmasi} 4. Bidang penagihan dan keberatan memilih <i>submit</i> buku ekspedisi. {Menampilkan buku ekspedisi} 5. Sistem menampilkan buku ekspedisi. {Memilih data SKPD} 6. Bidang penagihan dan keberatan memilih data SKPD yang akan divalidasi. {Melakukan validasi penyerahan} 7. Bidang penagihan memilih validasi penyerahan SKPD pada SKPD yang dipilih. 8. Sistem mengubah status SKPD menjadi diserahkan. {Use case selesai} 9. <i>Use case</i> selesai.
Alternative Flows	1. Membatalkan validasi penyerahan Setelah {melakukan validasi penyerahan} Bidang penagihan dan keberatan melakukan pembatalan validasi. Sistem mengubah status pajak menjadi belum diserahkan, kemudian <i>use case</i> selesai.
Subflow	-
Post-Condition	1. Status pajak berubah menjadi diserahkan.

Glossary

SKPD atau Surat Ketetapan Pajak Daerah merupakan surat yang berisi ketetapan pajak atas objek pajak milik wajib pajak yang diterbitkan oleh bidang pendataan dan penetapan untuk diserahkan ke wajib pajak sebagai pemberitahuan pembayaran pajak.

14. Validasi pembayaran

Spesifikasi *use case* validasi pembayaran dijelaskan pada Tabel 4.25 berikut ini.

Tabel 4.25 Spesifikasi Use Case Validasi Pembayaran

Name	Validasi pembayaran
Brief Description	<i>Use case</i> ini mendeskripsikan tentang bagaimana sistem dapat terintegrasi dengan sistem bank untuk melakukan validasi pembayaran pajak.
Pre-Condition	1. Sistem bank dijalankan.
Basic Flow	{Mengirim request data pajak} 1. <i>Use case</i> dimulai ketika sistem bank memilih untuk melakukan validasi pembayaran. 2. Sistem bank mengirimkan <i>request</i> data pajak yang divalidasi ke sistem milik Dispenda. {Mengirim data pajak} 3. Sistem milik Dispenda melakukan pencarian data. 4. Sistem milik Dispenda mengirim data pajak pada sistem bank. {Konfirmasi validasi pembayaran} 5. Sistem bank melakukan konfirmasi data pembayaran. 6. Sistem bank mengirim mengirim pesan perubahan status pembayaran ke sistem milik Dispenda. 7. Sistem milik Dispenda mengubah status pembayaran pajak. {Mengirim pesan berhasil} 8. Sistem milik Dispenda mengirim pesan berhasil ke sistem bank. {Use case selesai} 9. <i>Use case</i> selesai.
Alternative Flows	1. Data pajak tidak ditemukan Setelah {mengirim data pajak} sistem tidak menemukan data pajak yang <i>direquest</i>. Sistem akan mengirimkan pesan data pajak tidak ditemukan, kemudian {<i>use case</i> selesai}.
Subflow	-
Post-Condition	1. Sistem mengubah status pembayaran pajak.

- | |
|---|
| 2. Sistem mengirim pesan berhasil ke sistem virtual bank. |
|---|

Glossary

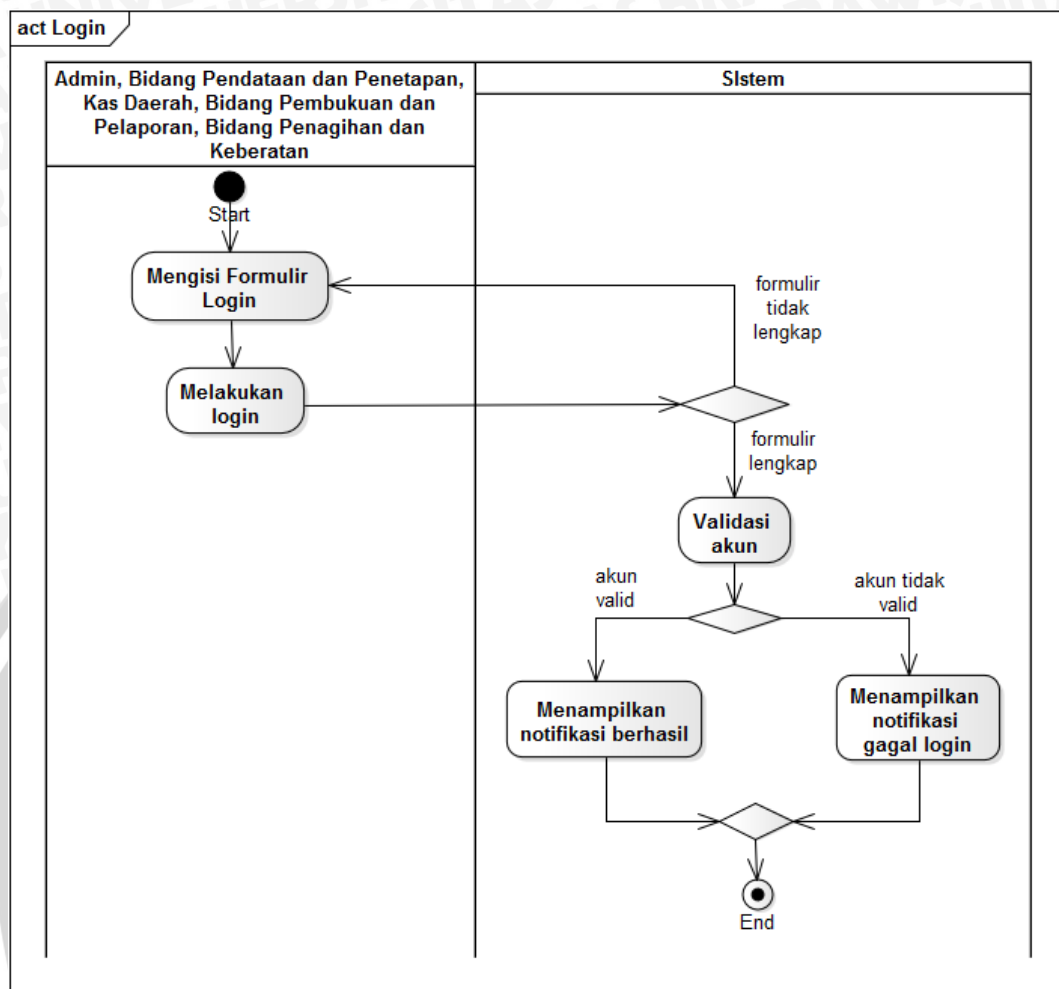
Data pajak yang dimaksud adalah data penetapan pajak dari objek pajak yang dimiliki oleh wajib pajak. Pembayaran yang dimaksud adalah pembayaran pajak atas penetapan pada SKPD. Data SKPD pajak air tanah terdiri dari : nomor SKPD, periode pajak, NPWPD, Nama WP, Nama Objek Pajak, Alamat Objek pajak (kabupaten/kodya, kecamatan dan desa/kelurahan), tanggal jatuh tempo, tarif pajak dalam persen (%), kode rekening, idx, harga per m³, volume tonase, tipe pajak air tanah, dan jumlah penetapan pajak. Data SKPD dari pajak parkir terdiri dari : nomor SKPD, periode pajak, NPWPD, Nama WP, Nama Objek Pajak, Alamat Objek pajak (kabupaten/kodya, kecamatan dan desa/kelurahan), tanggal jatuh tempo, tarif pajak dalam persen (%), kode rekening, jumlah omzet dan jumlah penetapan pajak. Validasi pembayaran pajak mengirimkan data berupa tanggal pembayaran dan jumlah pembayaran pajak oleh wajib pajak.

4.12 Activity Diagram

Diagram aktivitas atau *activity diagram* merupakan diagram yang menggambarkan aliran kerja (*work flow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Aliran aktivitas ini disesuaikan dengan spesifikasi use case yang telah dibuat sebelumnya.

1. Activity diagram login

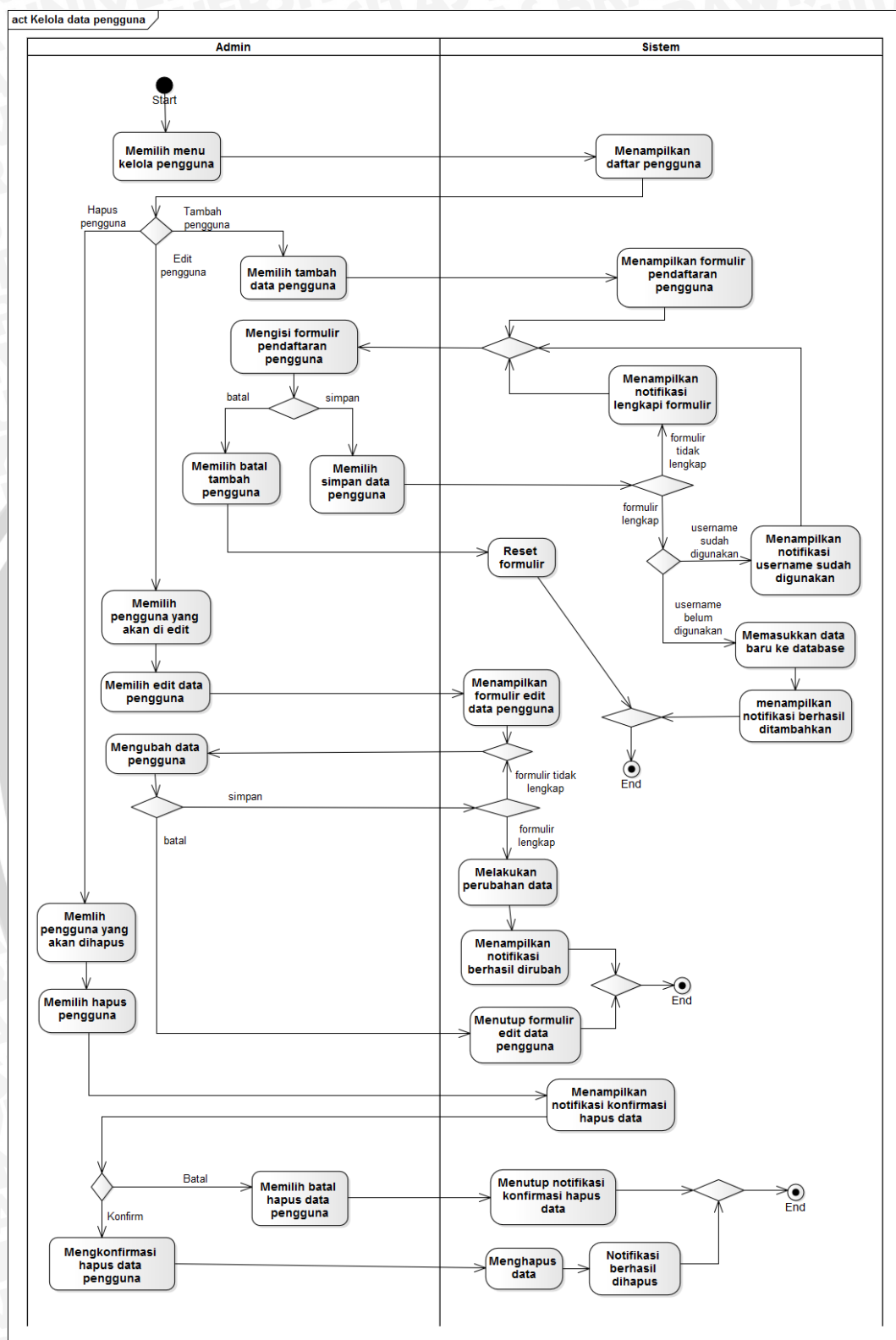
Activity diagram login menggambarkan aliran aktivitas ketika aktor melakukan *login*. Pada *activity diagram login* ini melibatkan aktor dan sistem. Aktor yang terlibat diantaranya Admin, Bidang Pendataan dan Penetapan, Kas Daerah, Bidang Pembukuan dan Pelaporan, atau Bidang Penagihan dan Keberatan. Aktivitas dimulai ketika aktor mengisi formulir *login* dan aktivitas berakhir ketika sistem menampilkan notifikasi berhasil atau tidak masuk ke dalam sistem. Gambar 4.4 menunjukkan *activity diagram* dari proses *login*.



Gambar 4.4 Activity diagram login

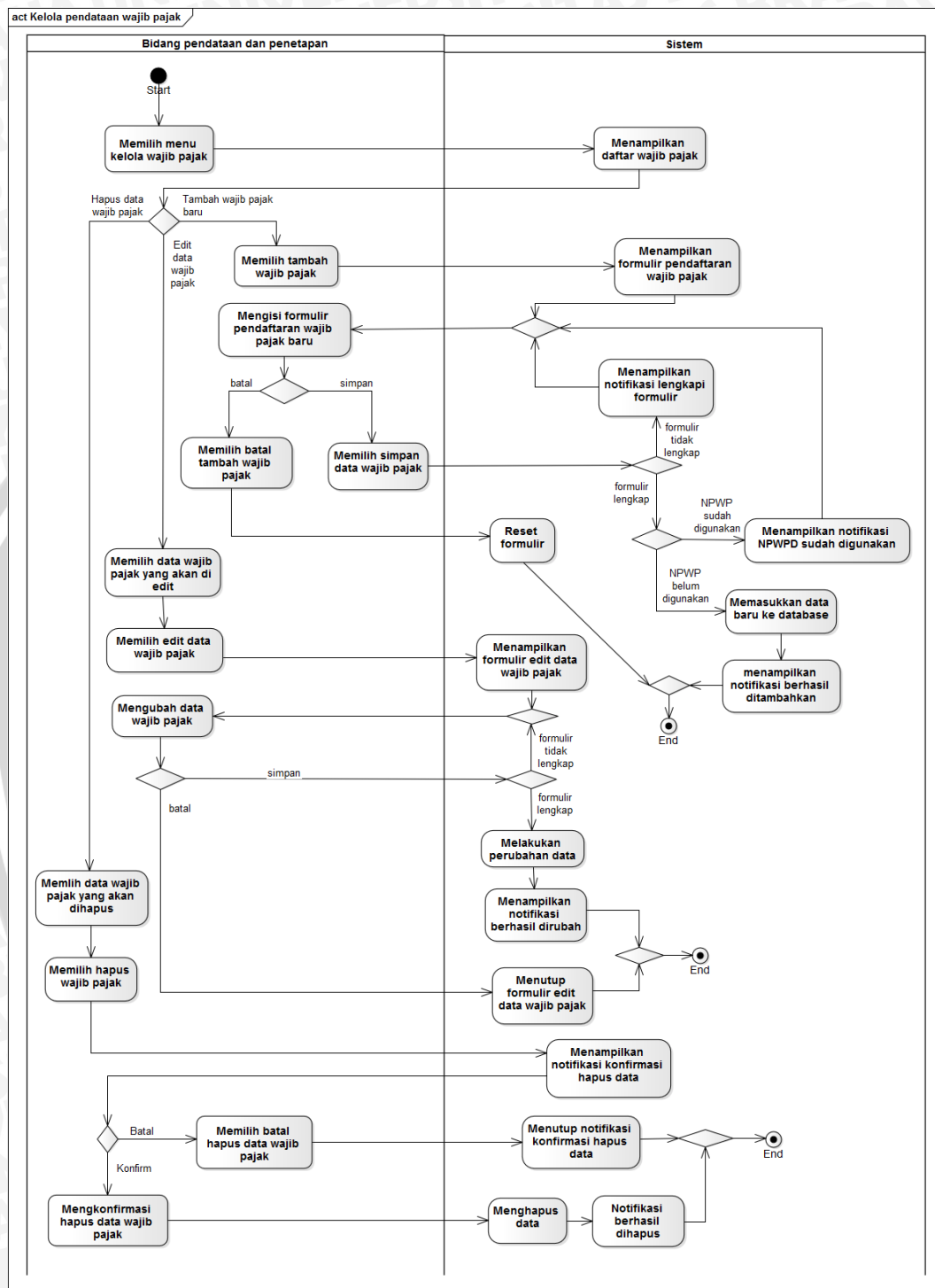
2. Activity diagram kelola data pengguna

Activity diagram kelola data pengguna menggambarkan aliran aktivitas ketika aktor melakukan pengelolaan data pengguna. Pada activity diagram kelola data pengguna ini melibatkan aktor admin dan sistem. Aktivitas kelola data pengguna dimulai ketika admin membuka sistem. Aktivitas akhir ketika admin memilih melakukan tambah pengguna adalah sistem melakukan reset formulir atau sistem menampilkan notifikasi berhasil menambahkan data pengguna. Aktivitas akhir ketika admin memilih melakukan edit data pengguna adalah sistem menampilkan notifikasi berhasil diubah atau sistem menutup formulir edit data pengguna. Aktivitas akhir ketika admin memilih menghapus data pengguna adalah sistem menutup notifikasi konfirmasi hapus data pengguna atau sistem menampilkan notifikasi berhasil dihapus. Gambar 4.5 menunjukkan activity diagram dari proses kelola data pengguna.



Gambar 4.5 Activity diagram kelola data pengguna

3. Activity diagram kelola pendataan wajib pajak

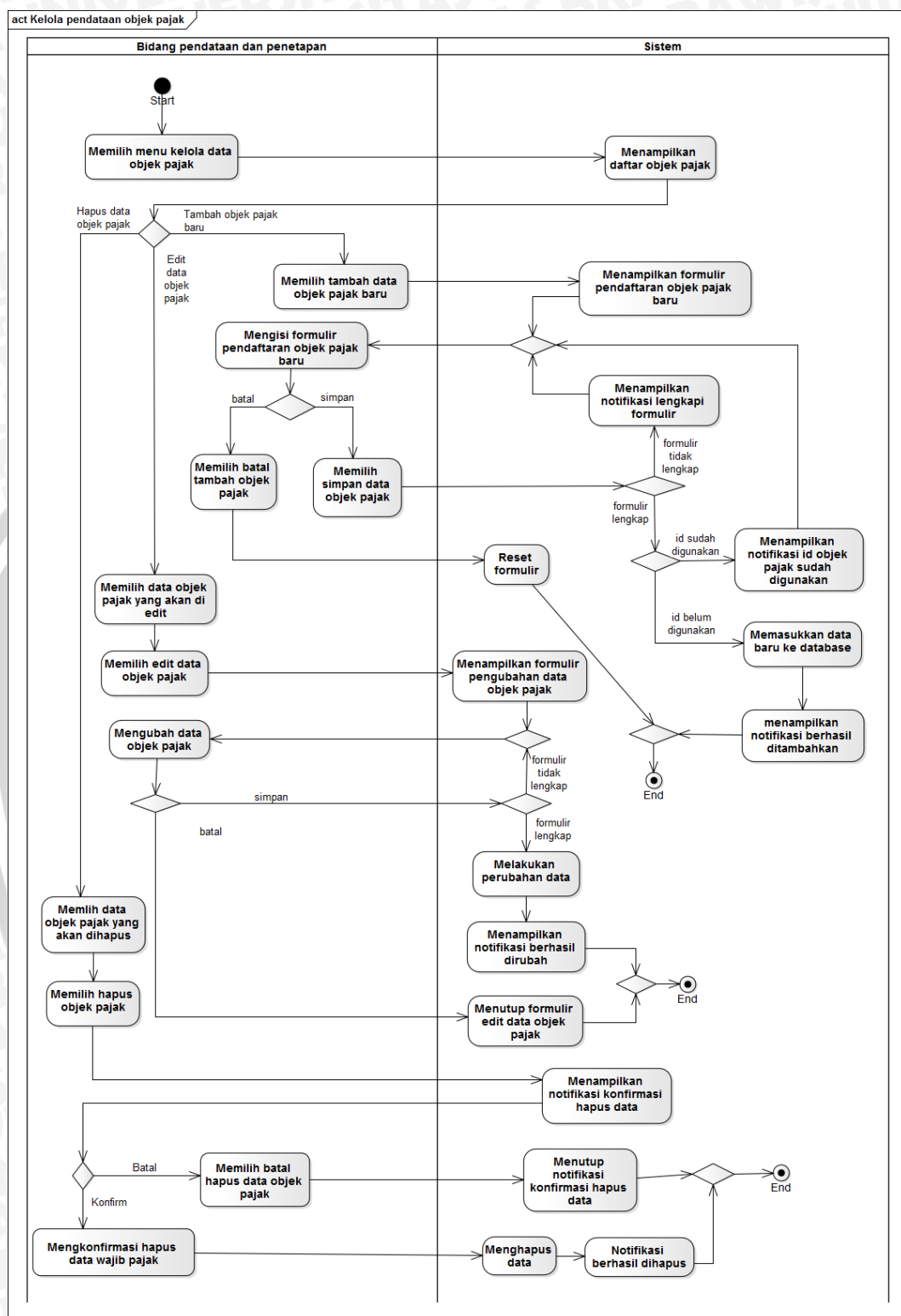


Gambar 4.6 Activity diagram kelola pendataan wajib pajak

Activity diagram kelola pendataan wajib pajak menggambarkan aliran aktivitas ketika aktor melakukan pengelolaan data wajib pajak. Pada *activity diagram* kelola pendataan wajib pajak ini melibatkan sistem serta aktor bidang pendataan dan penetapan. Aktivitas kelola pendataan wajib pajak dimulai ketika bidang pendataan dan penetapan membuka sistem. Aktivitas akhir ketika bidang pendataan dan penetapan memilih melakukan tambah wajib pajak baru adalah sistem melakukan reset formulir atau sistem menampilkan notifikasi data berhasil ditambahkan. Aktivitas akhir ketika bidang pendataan dan penetapan memilih melakukan edit data wajib pajak adalah sistem menampilkan notifikasi berhasil diubah atau sistem menutup formulir edit data wajib pajak. Aktivitas akhir ketika bidang pendataan dan penetapan memilih menghapus data wajib pajak adalah sistem menutup notifikasi konfirmasi hapus data atau sistem menampilkan notifikasi berhasil dihapus. Gambar 4.6 menunjukkan *activity diagram* dari proses/aktivitas pengelolaan wajib pajak.

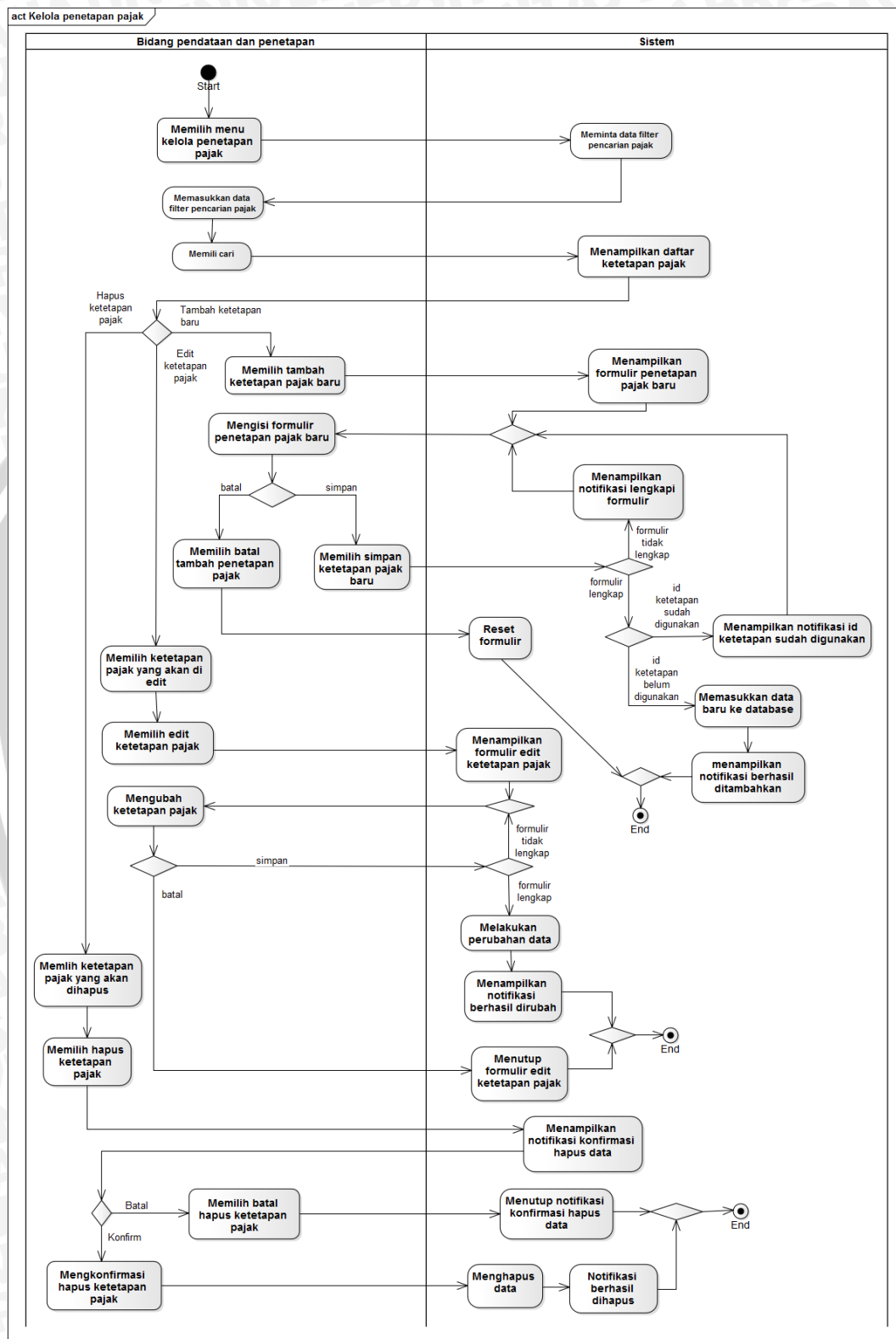
4. *Activity diagram* kelola pendataan objek pajak

Activity diagram kelola data objek pajak menggambarkan aliran aktivitas ketika aktor melakukan pengelolaan data objek pajak. Pada *activity diagram* kelola pendataan objek pajak ini melibatkan sistem serta aktor bidang pendataan dan penetapan. Aktivitas kelola pendataan objek pajak dimulai ketika bidang pendataan dan penetapan membuka sistem. Aktivitas akhir ketika bidang pendataan dan penetapan memilih melakukan tambah objek pajak baru adalah sistem melakukan reset formulir atau sistem menampilkan notifikasi data berhasil ditambahkan. Aktivitas akhir ketika bidang pendataan dan penetapan memilih melakukan edit data objek pajak adalah sistem menampilkan notifikasi berhasil diubah atau sistem menutup formulir edit data objek pajak. Aktivitas akhir ketika bidang pendataan dan penetapan memilih menghapus data objek pajak adalah sistem menutup notifikasi konfirmasi hapus data atau sistem menampilkan notifikasi berhasil dihapus. Gambar 4.7 menunjukkan *activity diagram* proses pengelolaan pendataan objek pajak.



Gambar 4.7 Activity diagram kelola pendataan objek pajak

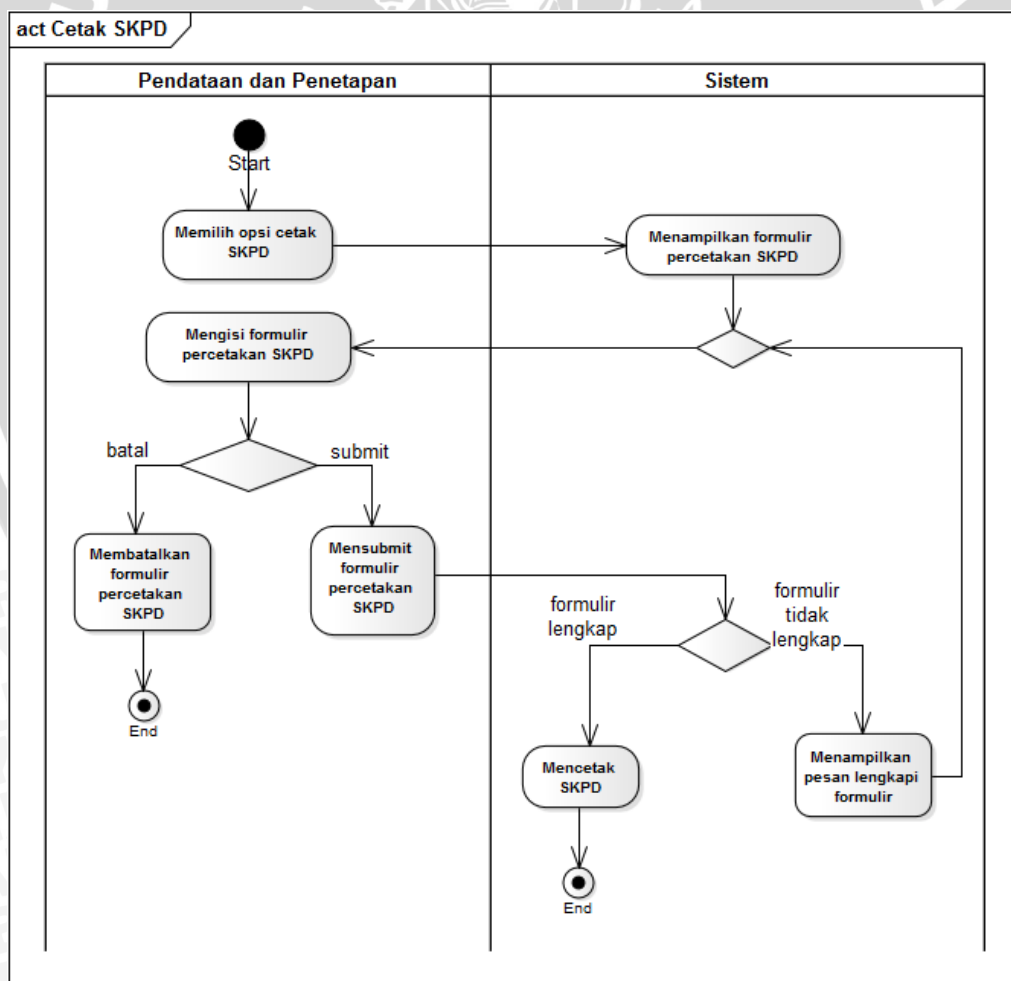
5. Activity diagram kelola penetapan pajak



Gambar 4.8 Activity diagram kelola penetapan pajak

Activity diagram kelola penetapan pajak menggambarkan aliran aktivitas ketika aktor melakukan pengelolaan penetapan pajak. Pada *activity diagram* kelola penetapan pajak ini melibatkan sistem serta aktor bidang pendataan dan penetapan. Aktivitas kelola penetapan pajak dimulai ketika bidang pendataan dan penetapan membuka sistem. Aktivitas akhir ketika bidang pendataan dan penetapan memilih melakukan tambah ketetapan pajak baru adalah sistem melakukan reset formulir atau sistem menampilkan notifikasi data berhasil ditambahkan. Aktivitas akhir ketika bidang pendataan dan penetapan memilih melakukan edit ketetapan pajak adalah sistem menampilkan notifikasi berhasil diubah atau sistem menutup formulir edit ketetapan pajak. Aktivitas akhir ketika bidang pendataan dan penetapan memilih menghapus data objek pajak adalah sistem menutup notifikasi konfirmasi hapus data atau sistem menampilkan notifikasi berhasil dihapus. Gambar 4.8 menunjukkan *activity diagram* aktivitas pengelolaan penetapan pajak.

6. Activity diagram cetak SKPD

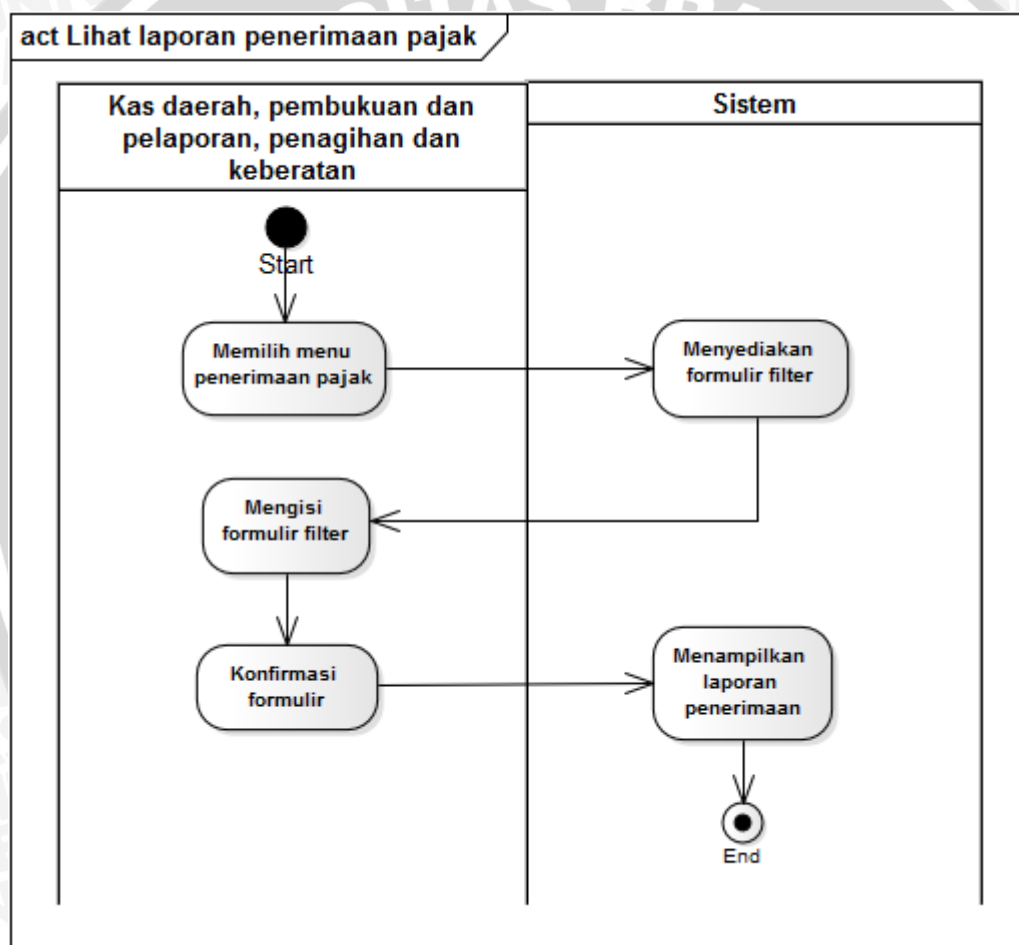


Gambar 4.9 Activity diagram cetak SKPD

Activity diagram cetak SKPD menggambarkan aliran aktivitas ketika aktor mencetak Surat Ketetapan Pajak yang telah ditetapkan. Pada *activity diagram* cetak SKPD ini melibatkan sistem serta aktor bidang pendataan dan penetapan. Aktivitas cetak SKPD dimulai ketika bidang pendataan dan penetapan membuka sistem dan aktivitas berakhir ketika aktor membatalkan formulir percetakan SKPD atau sistem mencetak SKPD.

Gambar 4.9 menunjukkan *activity diagram* proses cetak SKPD.

7. *Activity diagram* lihat laporan penerimaan pajak



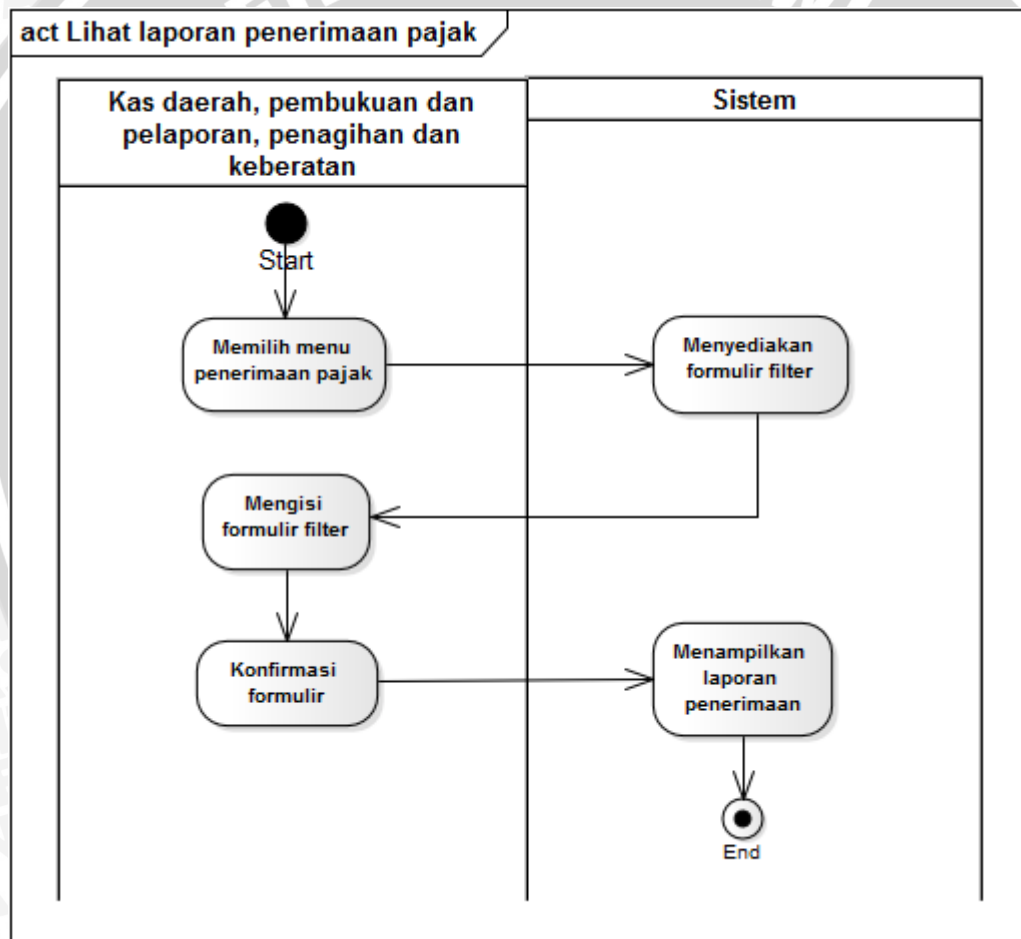
Gambar 4.10 *Activity diagram* lihat laporan penerimaan

Activity diagram lihat laporan penerimaan pajak menggambarkan aliran aktivitas ketika aktor melihat laporan penerimaan pajak. Pada *activity diagram* lihat laporan penerimaan pajak ini melibatkan aktor dan sistem. Aktor yang mungkin terlibat diantaranya Kas Daerah, Pembukuan dan Pelaporan, atau Penagihan dan Keberatan. Aktivitas dimulai ketika aktor membuka sistem dan aktivitas berakhir ketika sistem menampilkan laporan

penerimaan. Gambar 4.10 merupakan *activity diagram* dari aktivitas melihat laporan penerimaan pajak.

8. *Activity diagram* cetak laporan penerimaan pajak

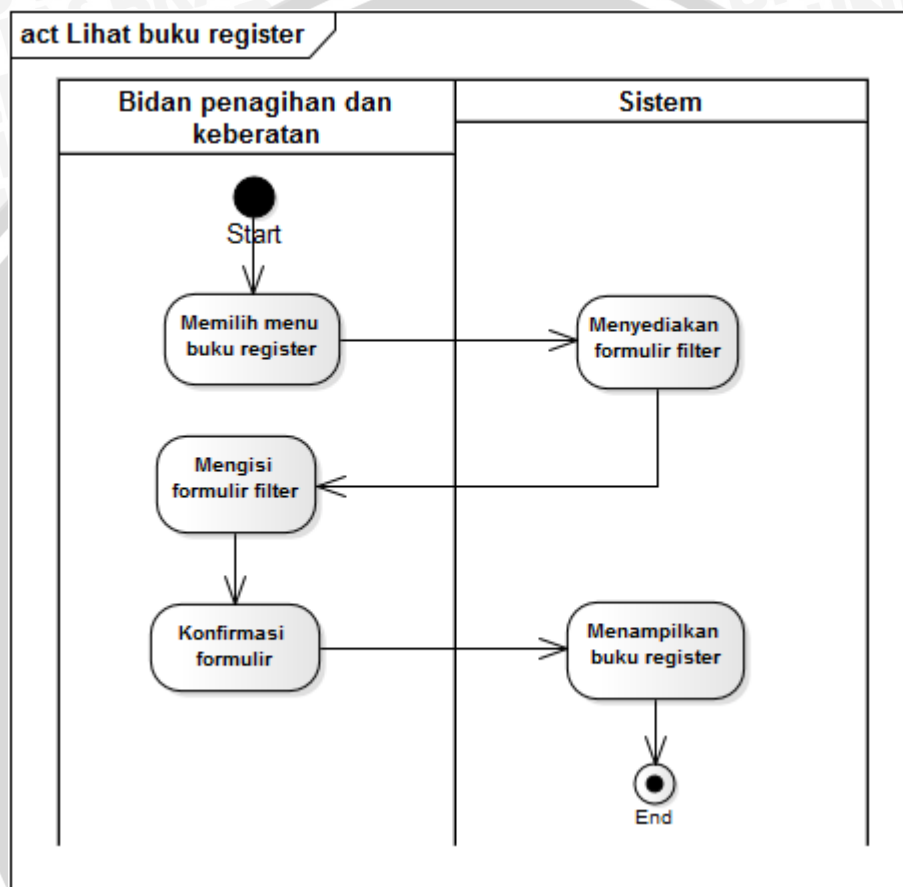
Activity diagram cetak laporan penerimaan pajak menggambarkan aliran aktivitas ketika aktor mencetak laporan penerimaan pajak sesuai kebutuhan. Pada *activity diagram* cetak laporan penerimaan pajak ini melibatkan aktor dan sistem. Aktor yang terlibat diantaranya Kas Daerah, Bidang Pembukuan dan Pelaporan, atau Bidang Penagihan dan Keberatan. Aktivitas dimulai ketika aktor membuka sistem dan aktivitas berakhir ketika sistem mencetak laporan penerimaan sesuai kebutuhan aktor. Gambar 4.11 menunjukkan *activity diagram* dari proses cetak laporan penerimaan pajak.



Gambar 4.11 *Activity diagram* cetak laporan penerimaan pajak

9. Activity diagram lihat buku register

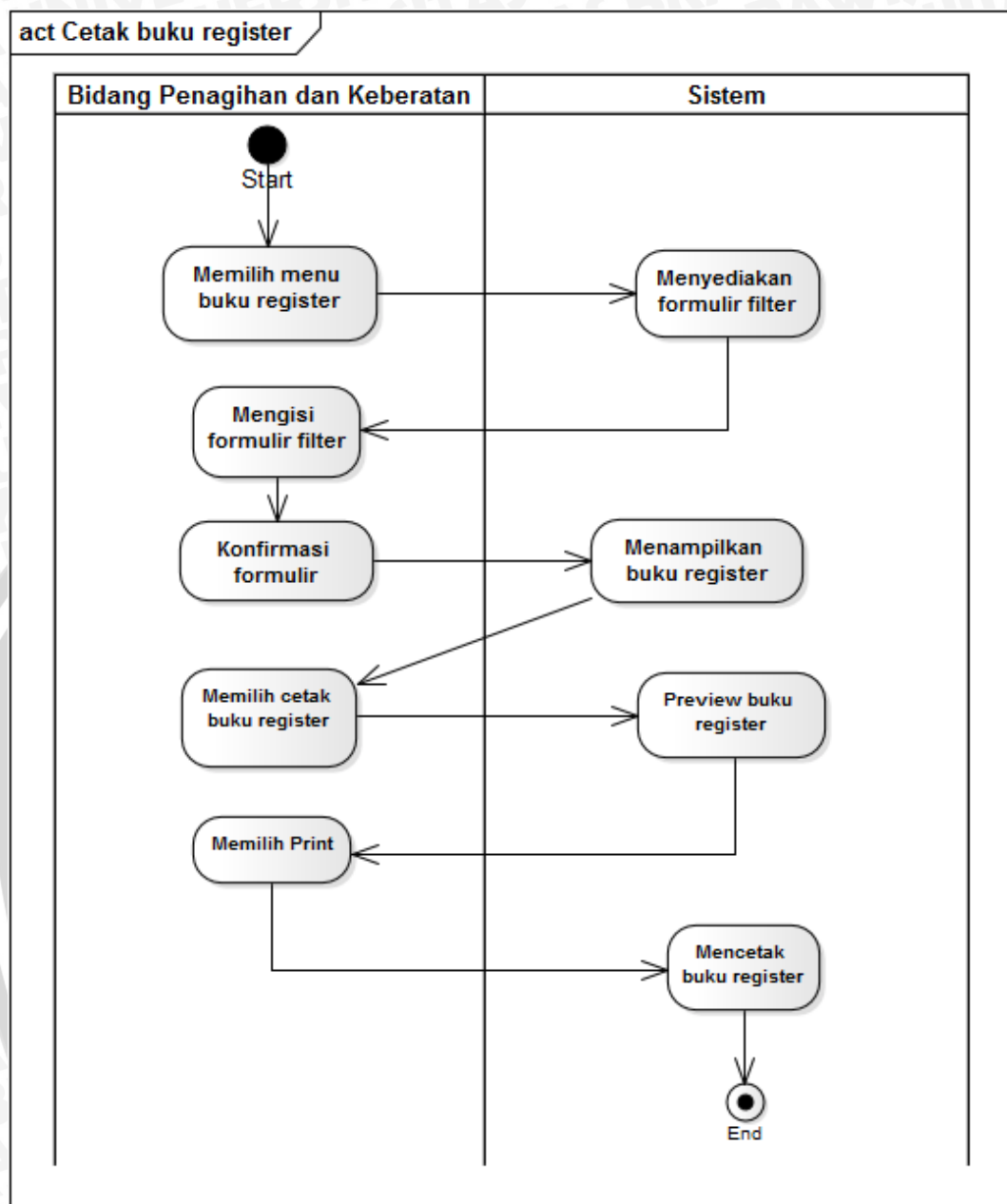
Activity diagram lihat buku register menggambarkan aliran aktivitas ketika aktor melihat buku register. Pada activity diagram lihat buku register ini melibatkan sistem serta aktor bidang penagihan dan keberatan. Aktivitas dimulai ketika aktor membuka sistem dan aktivitas berakhir ketika sistem menampilkan buku register. Gambar 4.12 menunjukkan activity diagram proses melihat buku register.



Gambar 4.12 Activity diagram lihat buku register

10. Activity diagram cetak buku register

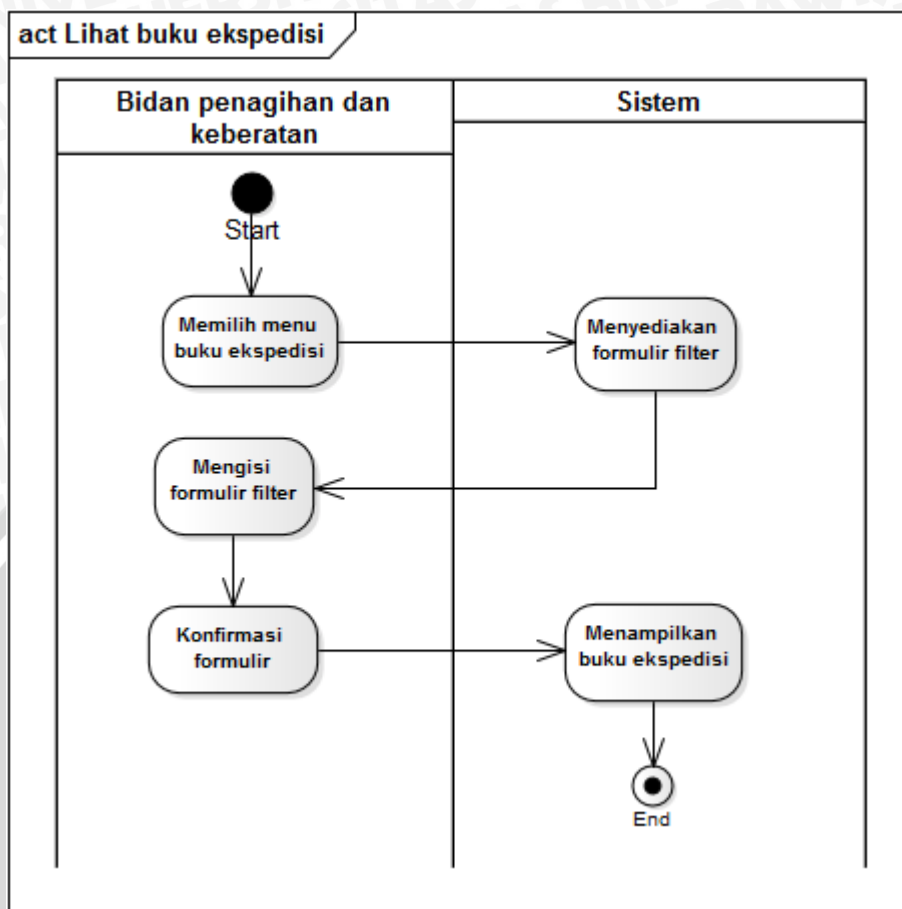
Activity diagram cetak buku register menggambarkan aliran aktivitas ketika aktor mencetak buku register. Pada activity diagram cetak buku register ini melibatkan sistem serta aktor bidang penagihan dan keberatan. Aktivitas dimulai ketika aktor membuka sistem dan aktivitas berakhir ketika sistem mencetak buku ekspedisi. Gambar 4.13 menunjukkan activity diagram proses cetak buku register.



Gambar 4.13 Activity diagram cetak buku register

11. Activity diagram lihat buku ekspedisi

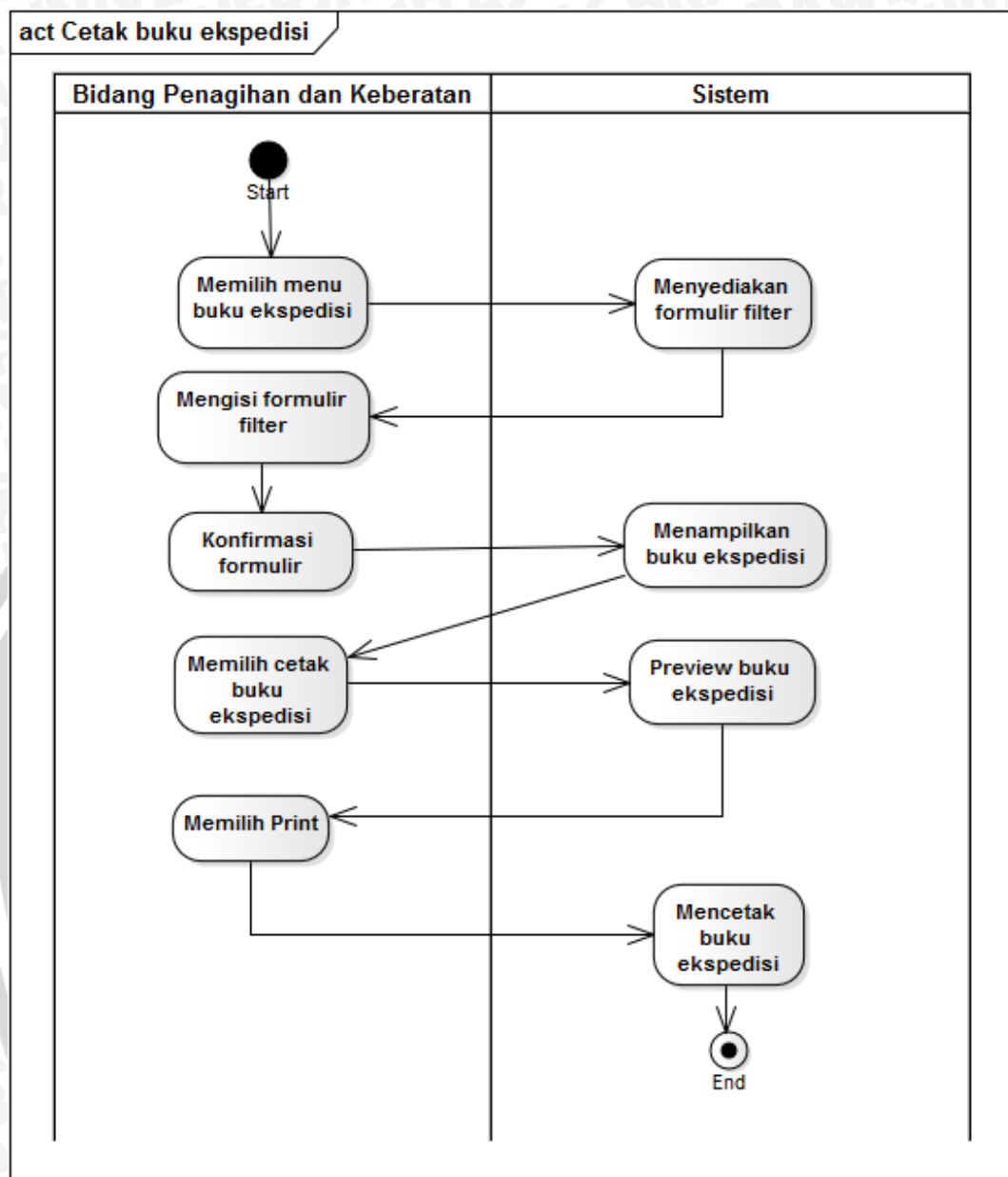
Activity diagram lihat buku ekspedisi menggambarkan aliran aktivitas ketika aktor melihat buku ekspedisi. Pada activity diagram lihat buku ekspedisi ini melibatkan sistem serta aktor bidang penagihan dan keberatan. Aktivitas dimulai ketika aktor membuka sistem dan aktivitas berakhir ketika sistem menampilkan buku ekspedisi. Gambar 4.14 menunjukkan activity diagram dari proses lihat buku ekspedisi.



Gambar 4.14 Activity diagram lihat buku ekspedisi

12. Activity diagram cetak buku ekspedisi

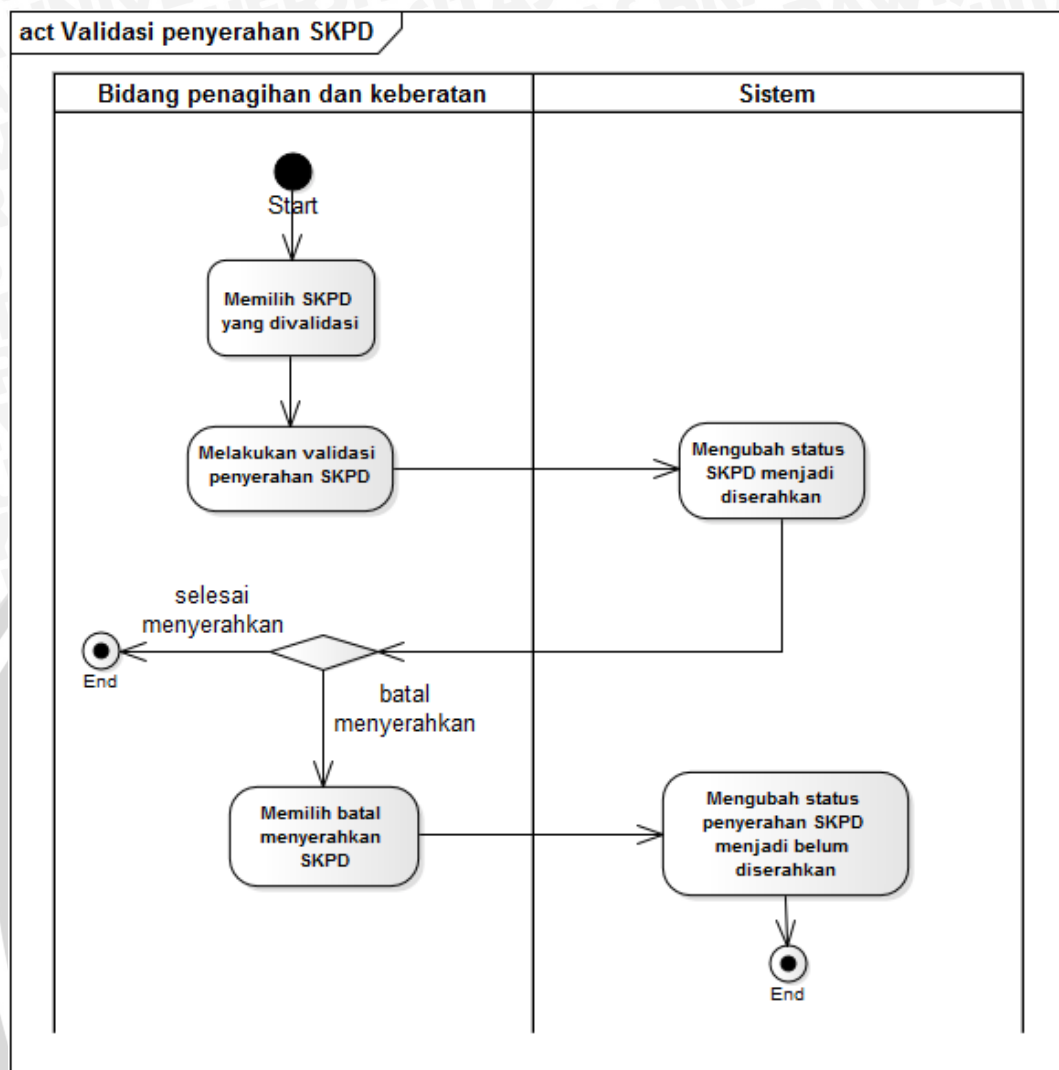
Activity diagram cetak buku ekspedisi menggambarkan aliran aktivitas ketika aktor mencetak buku ekspedisi. Pada activity diagram cetak buku ekspedisi ini melibatkan sistem serta aktor bidang penagihan dan keberatan. Aktivitas dimulai ketika aktor membuka sistem dan aktivitas berakhir ketika sistem mencetak buku ekspedisi. Gambar 4.15 menunjukkan activity diagram proses cetak buku ekspedisi.



Gambar 4.15 Activity diagram cetak buku ekspedisi

13. Activity diagram validasi penyerahan SKPD

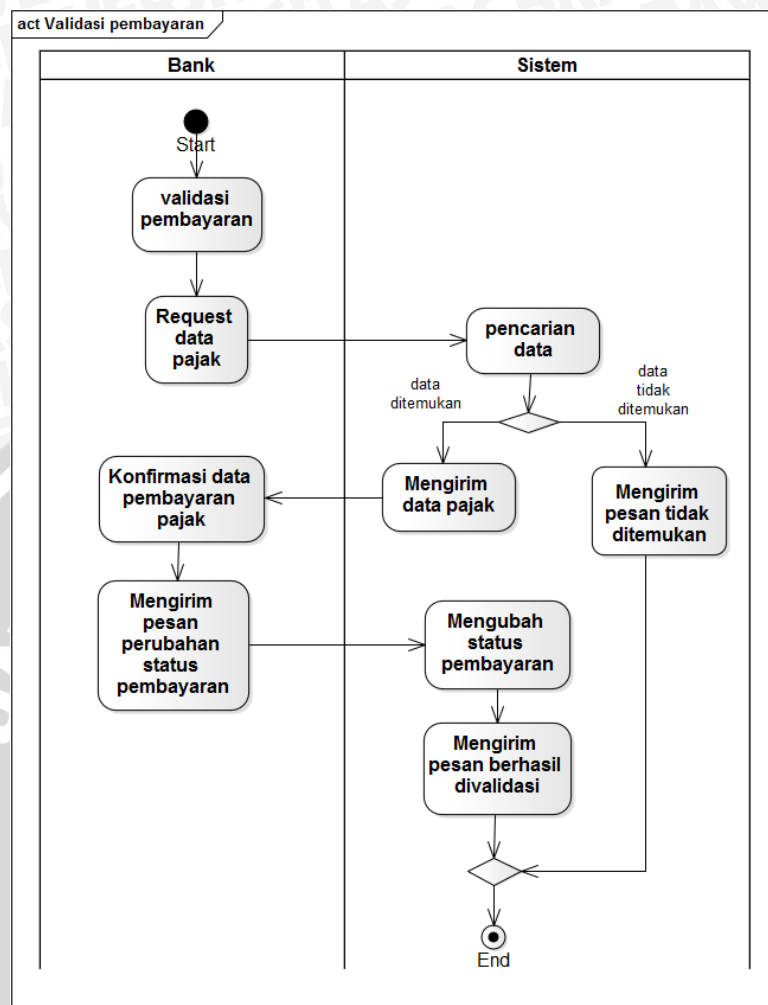
Activity diagram validasi penyerahan SKPD menggambarkan aliran aktivitas ketika aktor memvalidasi penyerahan SKPD pada sistem. Pada activity diagram validasi penyerahan SKPD ini melibatkan sistem serta aktor bidang penagihan dan keberatan. Aktivitas dimulai ketika aktor mengisi membuka sistem dan aktivitas berakhir ketika aktor selesai menyerahkan SKPD ke wajib pajak atau sistem mengubah status penyerahan SKPD menjadi belum diserahkan. Gambar 4.16 menunjukkan diagram activity proses validasi penyerahan SKPD.



Gambar 4.16 Activity diagram validasi penyerahan SKPD

14. Activity diagram validasi pembayaran

Activity diagram validasi pembayaran menggambarkan aliran aktivitas ketika aktor melakukan validasi pembayaran. Pada activity diagram validasi pembayaran ini melibatkan sistem serta aktor bank. Aktivitas dimulai ketika aktor melakukan validasi pembayaran dan aktivitas berakhir ketika sistem mengirim pesan tidak ditemukan atau sistem mengirim pesan berhasil divalidasi. Gambar 4.17 menunjukkan activity diagram proses validasi pembayaran.



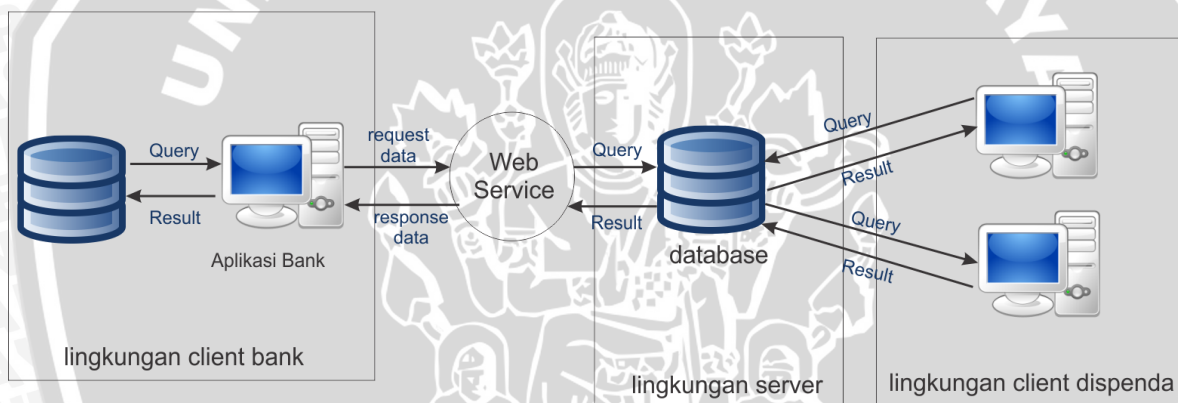
Gambar 4.17 Activity diagram validasi pembayaran

BAB 5 PERANCANGAN

Bagian ini menjelaskan mengenai perancangan sistem yang meliputi kelas-kelas analisis, mekanisme analisis, unifikasi kelas-kelas analisis, elemen desain yang terdiri dari *package* dan diagram kelas, *sequence diagram*, perancangan arsitektur, pemodelan data, perancangan antarmuka, dan hasil prototipe.

5.1 Analisis Arsitektur

Analisis arsitektur ini menggambarkan komunikasi internal di Dinas Pendapatan dan juga eksternal antara Dinas Pendapatan dengan pihak bank. Komunikasi internal di Dispenda menerapkan arsitektur 2-tier yang terdiri dari *client* dan *server*. Sedangkan komunikasi eksternal antara Dispenda dengan Bank Jatim menerapkan arsitektur 3-tier yang terdiri dari *client*, *middleware* dan *server*. *Web service* berperan sebagai *middleware*. *Web service* yang digunakan adalah *web service SOAP*.



Gambar 5.1 Analisis Arsitektur

Gambar 5.1 menjelaskan arsitektur Sistem Informasi Penerimaan PAD non-PBB. Arsitektur sistem ini dibagi menjadi bagian yaitu :

1. *Client 1* (Bank)

Aplikasi bank melakukan *request* terhadap *server* melalui *web service*. *Web service* akan melakukan *query* pada *database server* untuk mendapatkan data. *Database* kemudian mengirim hasil ke *web service* untuk dikirim ke aplikasi bank.

2. *Client 2* (Dispenda)

Lingkungan *client* Dispenda merupakan *client* yang dapat melakukan *request* terhadap *server* secara langsung melalui *browser*. Pada *client* ini tidak perlu menggunakan *web service* dalam pengaksesan data ke lingkungan *server*.

3. *Server*

Server pada sistem ini bertugas untuk merespon semua permintaan *client*. Baik permintaan *client* secara langsung maupun permintaan *client* melalui

web service. Selain merespon permintaan *client*, *server* juga bertugas untuk mengakses data terhadap *database* dan dikirimkan kembali ke *client*.

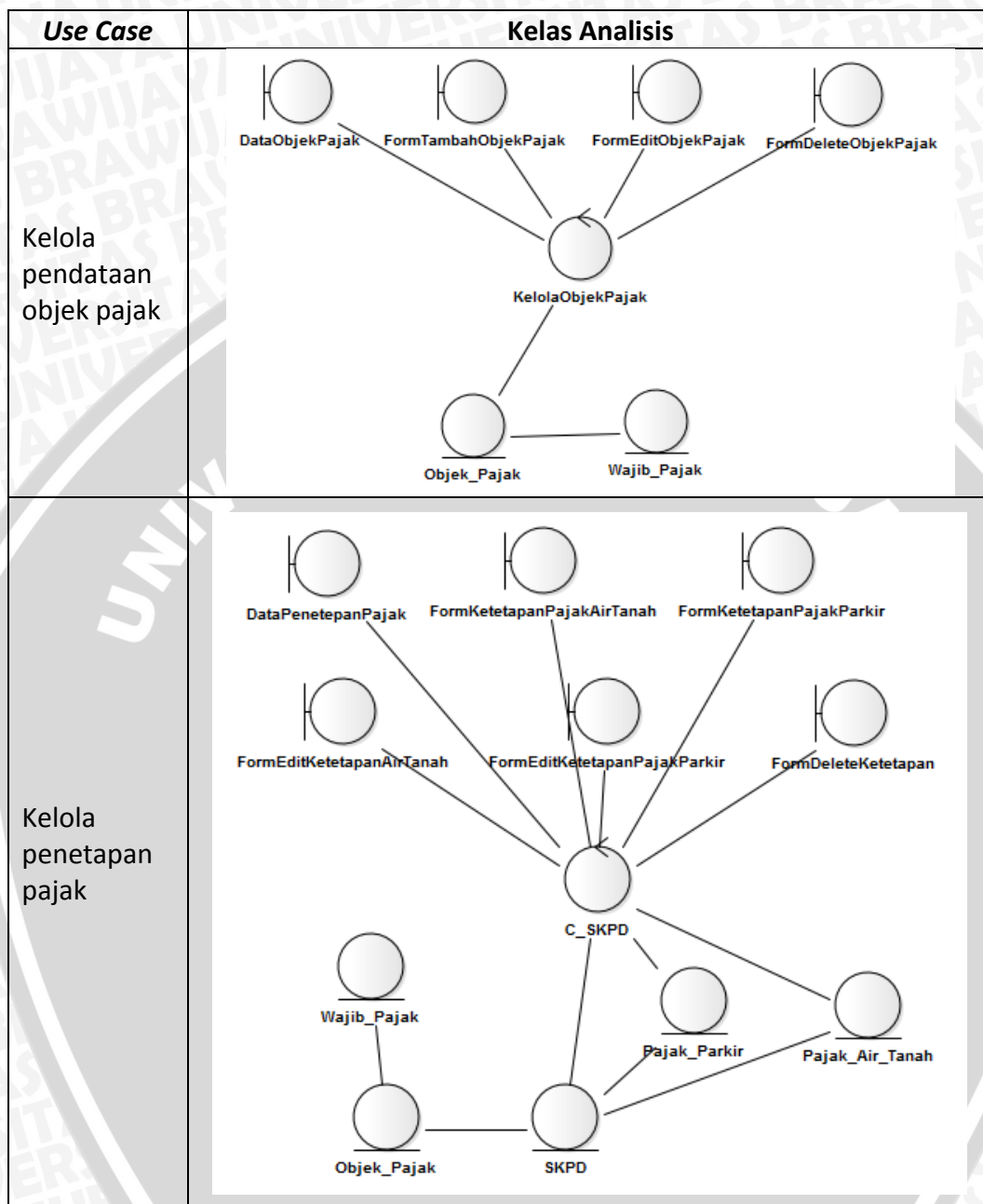
5.2 Kelas-Kelas Analisis

Setelah melakukan pendeskripsian *use case* secara lengkap pada tahap analisis *requirement*, kemudian dilakukan identifikasi kandidat kelas-kelas analisis dari sistem. Identifikasi kandidat kelas-kelas analisis ini diperoleh dengan menemukan kelas-kelas dari *use case behaviour*. Kelas analisis ini merupakan dugaan awal komposisi dari sistem. Tabel 5.1 berikut ini merupakan kelas-kelas analisis dari setiap *use case*.

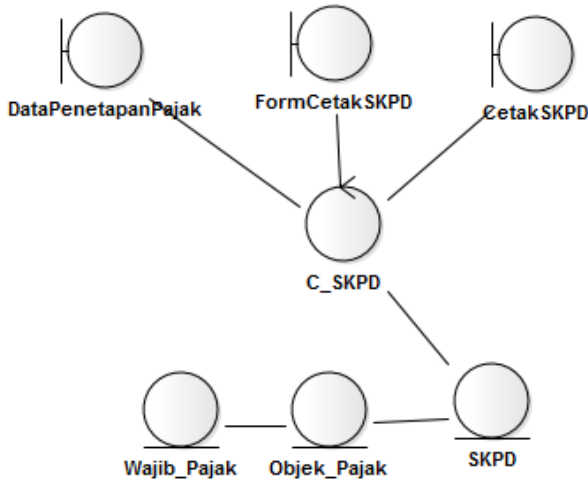
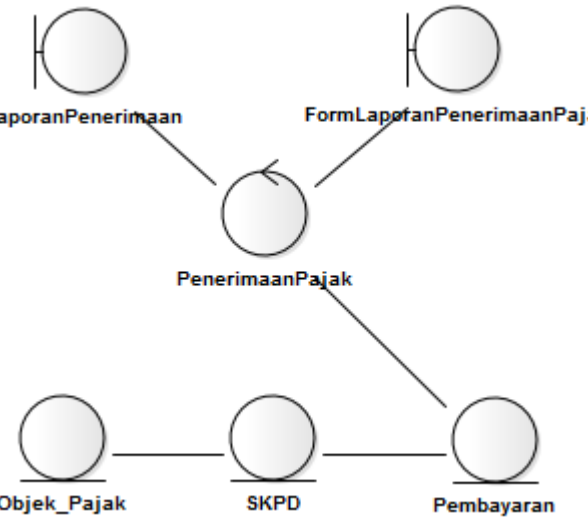
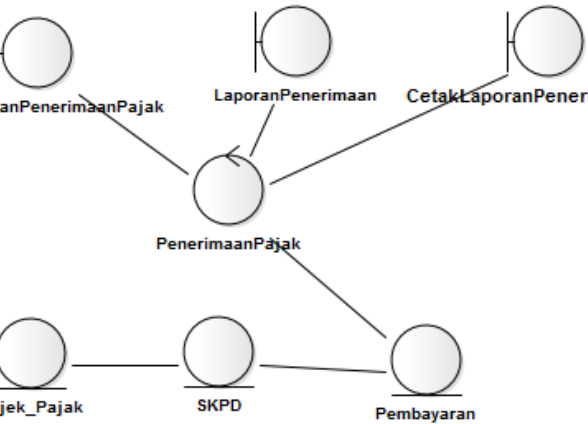
Tabel 5.1 Kelas-Kelas Analisis

Use Case	Kelas Analisis
Login	<pre> sequenceDiagram participant FormLogin participant Login participant Pengguna FormLogin-->>Login Login-->>Pengguna </pre>
Kelola data pengguna	<pre> classDiagram KelolaPengguna -- DataPengguna KelolaPengguna -- FormTambahPengguna KelolaPengguna -- FormEditPengguna KelolaPengguna -- FormDeletePengguna KelolaPengguna -- Pengguna </pre>
Kelola pendataan wajib pajak	<pre> classDiagram KelolaWP -- DataWajibPajak KelolaWP -- FormTambahWP KelolaWP -- FormEditWP KelolaWP -- FormDeleteWP KelolaWP -- Wajib_Pajak </pre>

Tabel 5.1 Kelas-Kelas Analisis (lanjutan)



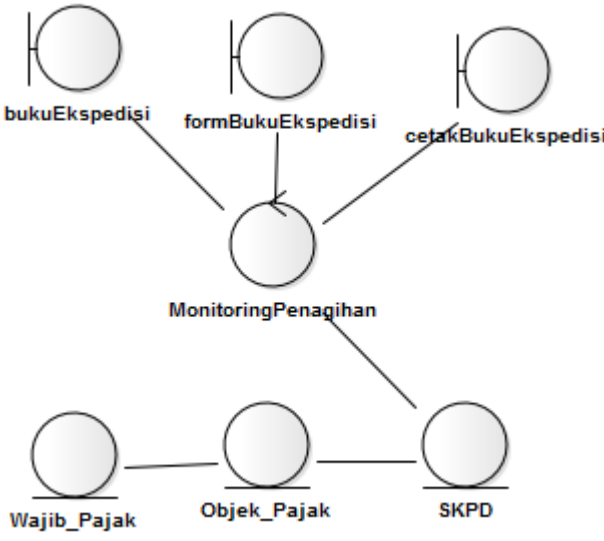
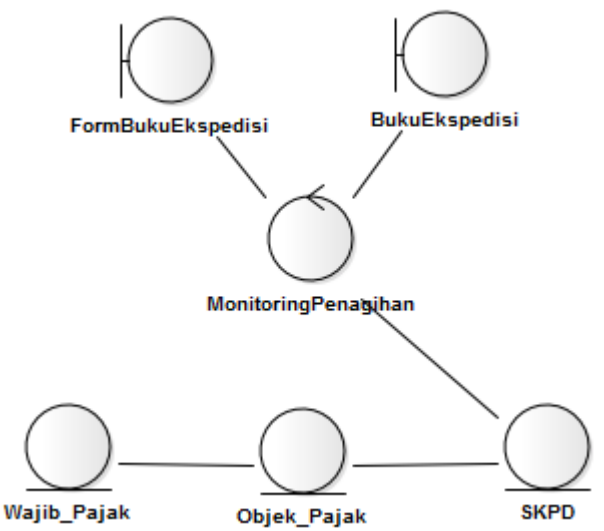
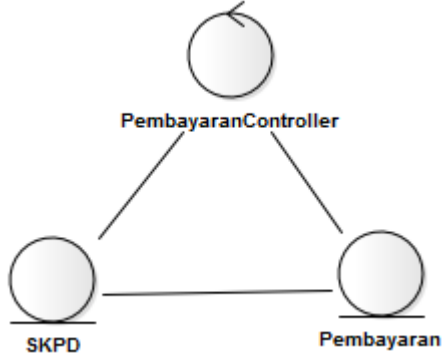
Tabel 5.1 Kelas-Kelas Analisis (lanjutan)

Use Case	Kelas Analisis
Cetak SKPD	 <pre> graph TD DP((DataPenetapanPajak)) --> CSKPD((C_SKPD)) FCSKPD((FormCetakSKPD)) --> CSKPD CSKPD --> ZSKPD((CetakSKPD)) WJ((Wajib_Pajak)) --- OP((Objek_Pajak)) OP --- SKPD((SKPD)) </pre> The diagram for 'Cetak SKPD' shows a central use case 'C_SKPD' receiving input from 'DataPenetapanPajak' and 'FormCetakSKPD', and producing 'CetakSKPD'. It is associated with entities 'Wajib_Pajak', 'Objek_Pajak', and 'SKPD'.
Lihat laporan penerimaan pajak	 <pre> graph TD LP((LaporanPenerimaan)) --> PJP((PenerimaanPajak)) FLPP((FormLaporanPenerimaanPajak)) --> PJP PJP --- OP((Objek_Pajak)) PJP --- SKPD((SKPD)) PJP --- P((Pembayaran)) </pre> The diagram for 'Lihat laporan penerimaan pajak' shows a central use case 'PenerimaanPajak' receiving input from 'LaporanPenerimaan' and 'FormLaporanPenerimaanPajak'. It is associated with entities 'Objek_Pajak', 'SKPD', and 'Pembayaran'.
Cetak laporan penerimaan pajak	 <pre> graph TD FLPP((FormLaporanPenerimaanPajak)) --> PJP((PenerimaanPajak)) LP((LaporanPenerimaan)) --> PJP PJP --> CLPP((CetakLaporanPenerimaanPajak)) PJP --- OP((Objek_Pajak)) PJP --- SKPD((SKPD)) PJP --- P((Pembayaran)) </pre> The diagram for 'Cetak laporan penerimaan pajak' shows a central use case 'PenerimaanPajak' receiving input from 'FormLaporanPenerimaanPajak' and 'LaporanPenerimaan', and producing 'CetakLaporanPenerimaanPajak'. It is associated with entities 'Objek_Pajak', 'SKPD', and 'Pembayaran'.

Tabel 5.1 Kelas-Kelas Analisis (lanjutan)

Use Case	Kelas Analisis
Lihat buku register	<pre> classDiagram class MonitoringPenagihan class FormBukuRegister class BukuRegister class Wajib_Pajak class Objek_Pajak class SKPD MonitoringPenagihan --> FormBukuRegister MonitoringPenagihan --> BukuRegister Wajib_Pajak --> Objek_Pajak Objek_Pajak --> SKPD MonitoringPenagihan --> SKPD </pre>
Cetak buku register	<pre> classDiagram class MonitoringPenagihan class bukuRegister class formBukuRegister class CetakBukuRegister class Wajib_Pajak class Objek_Pajak class SKPD MonitoringPenagihan --> bukuRegister MonitoringPenagihan --> formBukuRegister MonitoringPenagihan --> CetakBukuRegister Wajib_Pajak --> Objek_Pajak Objek_Pajak --> SKPD MonitoringPenagihan --> SKPD </pre>
Lihat buku ekspedisi	<pre> classDiagram class MonitoringPenagihan class FormBukuEkspedisi class BukuEkspedisi class Wajib_Pajak class Objek_Pajak class SKPD MonitoringPenagihan --> FormBukuEkspedisi MonitoringPenagihan --> BukuEkspedisi Wajib_Pajak --> Objek_Pajak Objek_Pajak --> SKPD MonitoringPenagihan --> SKPD </pre>

Tabel 5.1 Kelas-Kelas Analisis (lanjutan)

Use Case	Kelas Analisis
Cetak buku ekspedisi	
Validasi penyerahan SKPD	
Validasi pembayaran	

Dari hasil identifikasi kelas analisis pada Tabel 5.1 diperoleh :

- a. Kelas yang terlibat pada *use case login* terdiri dari *boundary* FormLogin, *controller* login dan *entity* Pengguna.
- b. Kelas yang terlibat pada *use case* kelola data pengguna terdiri dari *boundary* DataPengguna, FormTambahPengguna, FormEditPengguna, FormDeletePengguna; *controller* KelolaPengguna; serta *entity* Pengguna.
- c. Kelas yang terlibat pada *use case* kelola pendataan wajib pajak adalah *boundary* DataWajibPajak, FormTambahWP, FormEditWP, FormDeleteWP; *controller* KelolaWP; serta *entity* Wajib_Pajak.
- d. Kelas yang terlibat pada *use case* kelola pendataan objek pajak adalah *boundary* DataObjekPajak, FormTambahObjekPajak, FormEditObjekPajak, FormDeleteObjekPajak; *controller* KelolaObjekPajak; serta *entity* Objek_Pajak dan Wajib_Pajak.
- e. Kelas yang terlibat pada *use case* kelola penetapan pajak diantaranya adalah *boundary* DataPenetapanPajak, FormKetetapanPajakAirTanah, FormKetetapanPajakParkir, FormEditKetetapanAirTanah, FormEdirKetetapanPajakParkir, FormDeleteKetetapan; *controller* C_SKPD; serta *entity* SKPD, Pajak_Parkir, Pajak_Air_Tanah, Objek_Pajak, dan Wajib_Pajak.
- f. Kelas yang terlibat pada *use case* Cetak SKPD adalah *boundary* DataPenetapanPajak, FormCetakSKPD, CetakSKPD; *controller* C_SKPD; serta *entity* SKPD, Objek_Pajak, dan Wajib_Pajak.
- g. Kelas yang terlibat pada *use case* lihat laporan penerimaan adalah *boundary* LaporanPenerimaan, FormLaporanPenerimaanPajak; *controller* PenerimaanPajak; serta *entity* Pembayaran, SKPD dan Objek_Pajak.
- h. Kelas yang terlibat pada *use case* cetak laporan penerimaan pajak adalah *boundary* FormLaporanPenerimaanPajak, LaporanPenerimaan dan CetakLaporanPenerimaan; *controller* PenerimaanPajak; serta *entity* Pembayaran, SKPD dan Objek_Pajak.
- i. Kelas yang terlibat pada *use case* lihat buku register terdiri dari *boundary* FormBukuRegister dan BukuRegister; *controller* MonitoringPenagihan; serta *entity* SKPD, Objek_Pajak, dan Wajib_Pajak.
- j. Kelas yang terlibat pada *use case* cetak buku register terdiri dari *boundary* FormBukuRegister, BukuRegister dan CetakBukuRegister; *controller* MonitoringPenagihan; serta *entity* SKPD, Objek_Pajak, dan Wajib_Pajak.
- k. Kelas yang terlibat pada *use case* lihat buku ekspedisi adalah *boundary* FormBukuEkspedisi dan BukuEkspedisi; *controller* MonitoringPenagihan; serta *entity* SKPD, Objek_Pajak dan Wajib_Pajak.

- l. Kelas yang terlibat pada *use case* cetak buku ekspedisi adalah *boundary* FormBukuEkspedisi, BukuEkspedisi dan cetakBukuEkspedisi; *controller* MonitoringPenagihan; serta *entity* SKPD, Objek_Pajak dan Wajib_Pajak.
- m. Kelas yang terlibat pada *use case* validasi penyerahan SKPD adalah *boundary* FormBukuEkspedisi dan BukuEkspedisi; *controller* MonitoringPenagihan; serta *entity* SKPD, Objek_Pajak dan Wajib_Pajak
- n. Kelas yang terlibat pada *use case* validasi pembayaran adalah *controller* PembayaranController yang nantinya akan menjadi kelas bertipe *service* serta *entity* Pembayaran dan SKPD.

5.3 Mekanisme Analisis

Mekanisme analisis merupakan analisis arsitektur untuk mengurangi kompleksitas analisis dan untuk meningkatkan konsistensi. Mekanisme analisis yang digunakan dalam penelitian ini yaitu *persistence* dan *distribution*. Tabel 5.2 berikut ini adalah mekanisme analisis yang digunakan oleh kelas-kelas analisis yang telah diidentifikasi sebelumnya.

Tabel 5.2 Mekanisme Analisis

Kelas analisis	Mekanisme Analisis
BukuEkspedisi	<i>None</i>
FormBukuEkspedisi	<i>None</i>
CetakBukuEkspedisi	<i>None</i>
MonitoringPenagihan	<i>None</i>
Wajib_Pajak	<i>Persistence</i>
Objek_Pajak	<i>Persistence</i>
SKPD	<i>Persistence</i>
BukuRegister	<i>None</i>
FormBukuRegister	<i>None</i>
CetakBukuRegister	<i>None</i>
MonitoringPenagihan	<i>None</i>
FormLaporanPenerimaanPajak	<i>None</i>
LaporanPenerimaan	<i>None</i>
CetakLaporanPenerimaanPajak	<i>None</i>
PenerimaanPajak	<i>None</i>
Pembayaran	<i>Persistence</i>
DataPenetapanPajak	<i>None</i>
FormCetakSKPD	<i>None</i>
CetakSKPD	<i>None</i>
C_SKPD	<i>None</i>
DataPengguna	<i>None</i>
FormTambahPengguna	<i>None</i>
FormEditPengguna	<i>None</i>

Tabel 5.2 Mekanisme Alanisis (lanjutan)

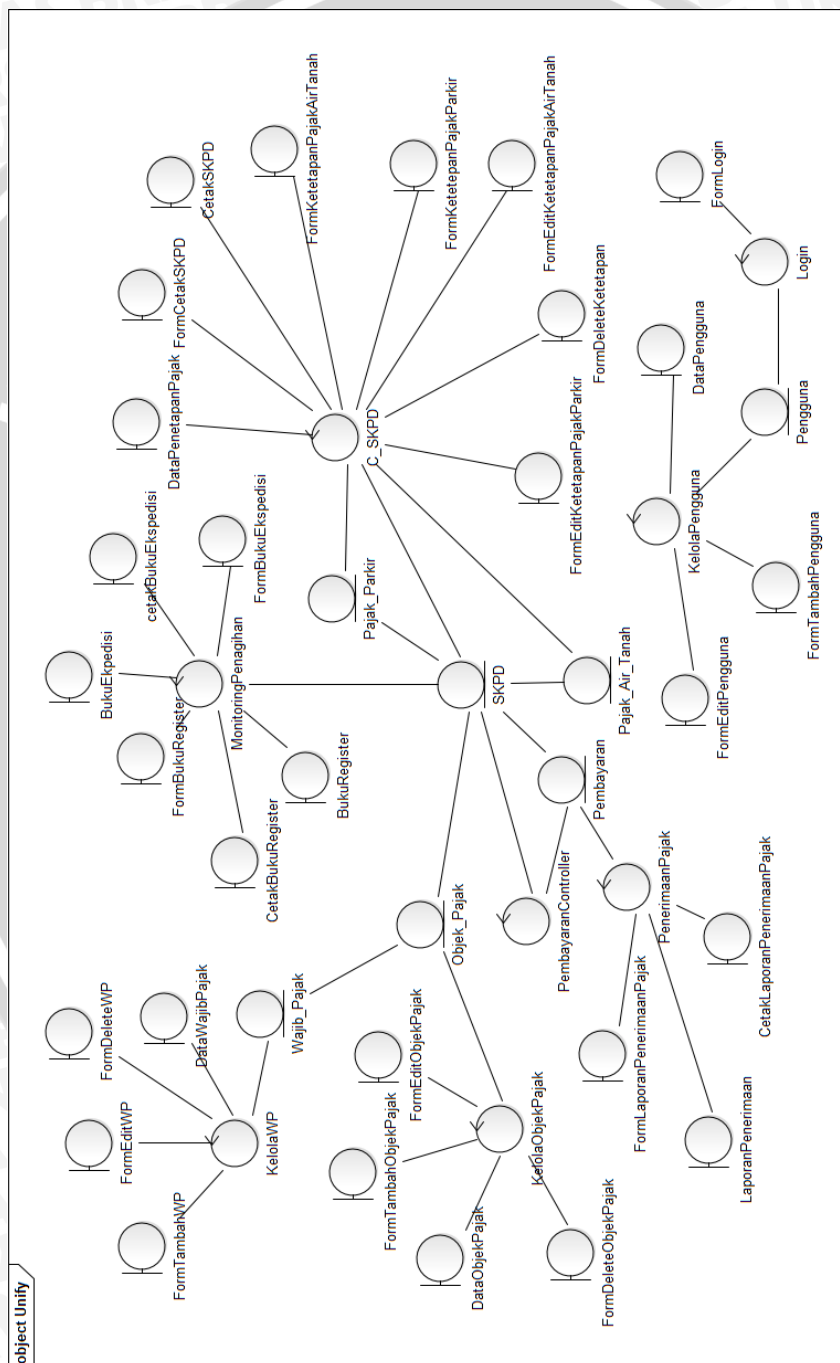
Kelas analisis	Mekanisme Analisis
FormDeletePengguna	<i>None</i>
KelolaPengguna	<i>None</i>
Pengguna	<i>Persistency</i>
DataObjekPajak	<i>None</i>
FormTambahObjekPajak	<i>None</i>
FormEditObjekPajak	<i>None</i>
FormDeleteObjekPajak	<i>None</i>
KelolaObjekPajak	<i>None</i>
DataWajibPajak	<i>None</i>
FormTambahWP	<i>None</i>
FormEditWP	<i>None</i>
FormDeleteWP	<i>None</i>
KelolaWP	<i>None</i>
DataPenetapanPajak	<i>None</i>
FormKetetapanPajakAirTanah	<i>None</i>
FormKetetapanPajakParkir	<i>None</i>
FormEditKetetapanAirTanah	<i>None</i>
FormEditKetetapanPajakParkir	<i>None</i>
FormDeleteKetetapan	<i>None</i>
Pajak_Parkir	<i>Persistency</i>
Pajak_Air_Tanah	<i>Persistency</i>
FormLogin	<i>None</i>
Login	<i>None</i>
PembayaranController	<i>Distribution</i>

Mekanisme analisis *persistency* merupakan mekanisme analisis dari kelas yang memiliki kemampuan untuk tetap menyediakan data selama suatu sistem masih ada. Kelas analisis yang memiliki mekanisme analisis *persistency* merupakan kelas yang berhubungan dengan data yang akan disimpan dan dapat diakses lagi. Dari kelas analisis yang telah diidentifikasi sebelumnya, mekanisme analisis *persistency* dimiliki oleh kelas Wajib_Pajak, Objek_Pajak, SKPD, Pembayaran, Pengguna, Pajak_Parkir, dan Pajak_Air_Tanah.

Mekanisme analisis *distribution* merupakan mekanisme analisis dimana kelas memiliki kemampuan untuk mengirim/mendistribusikan elemen melintasi *node* yang ada dari sistem. Dari identifikasi kelas analisis sebelumnya kelas PembayaranController memiliki mekanisme analisis *Distribution*. Kelas ini nantinya menjadi bagian dari mekanisme layanan milik Dispenda yang dapat digunakan oleh Bank Jatim.

5.4 Unifikasi kelas-kelas analisis

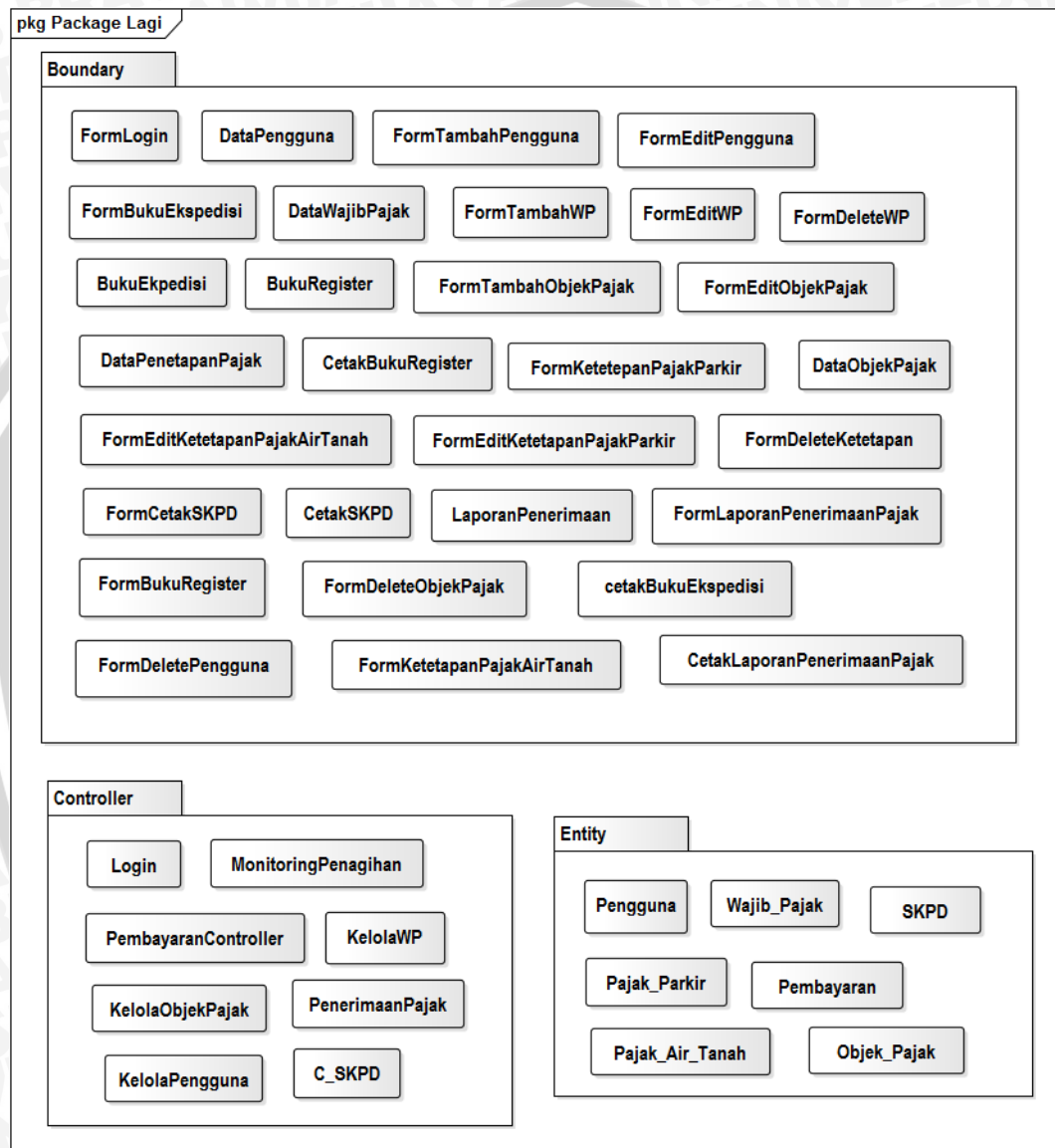
Sebelum dilakukan perancangan, perlu dilakukan filter kelas-kelas analisis yang telah didefinisikan sebelumnya. Ada kemungkinan beberapa *use case* menggunakan kelas yang sama. Sebuah kelas dapat digunakan untuk beberapa *use case*. Oleh karena itu diperlukan unifikasi atau penggabungan kelas-kelas analisis. Gambar 5.2 merupakan hasil unifikasi dari kelas-kelas analisis yang telah didefinisikan sebelumnya.



Gambar 5.2 Unifikasi Kelas-Kelas Analisis

5.5 Package diagram

Package diagram merupakan diagram yang menyediakan cara mengumpulkan elemen-elemen menjadi kelompok-kelompok. Gambar 5.3 berikut ini merupakan pengelompokan kelas analisis dengan *package diagram* yang terdiri dari *package boundary*, *controller*, dan *entity*.

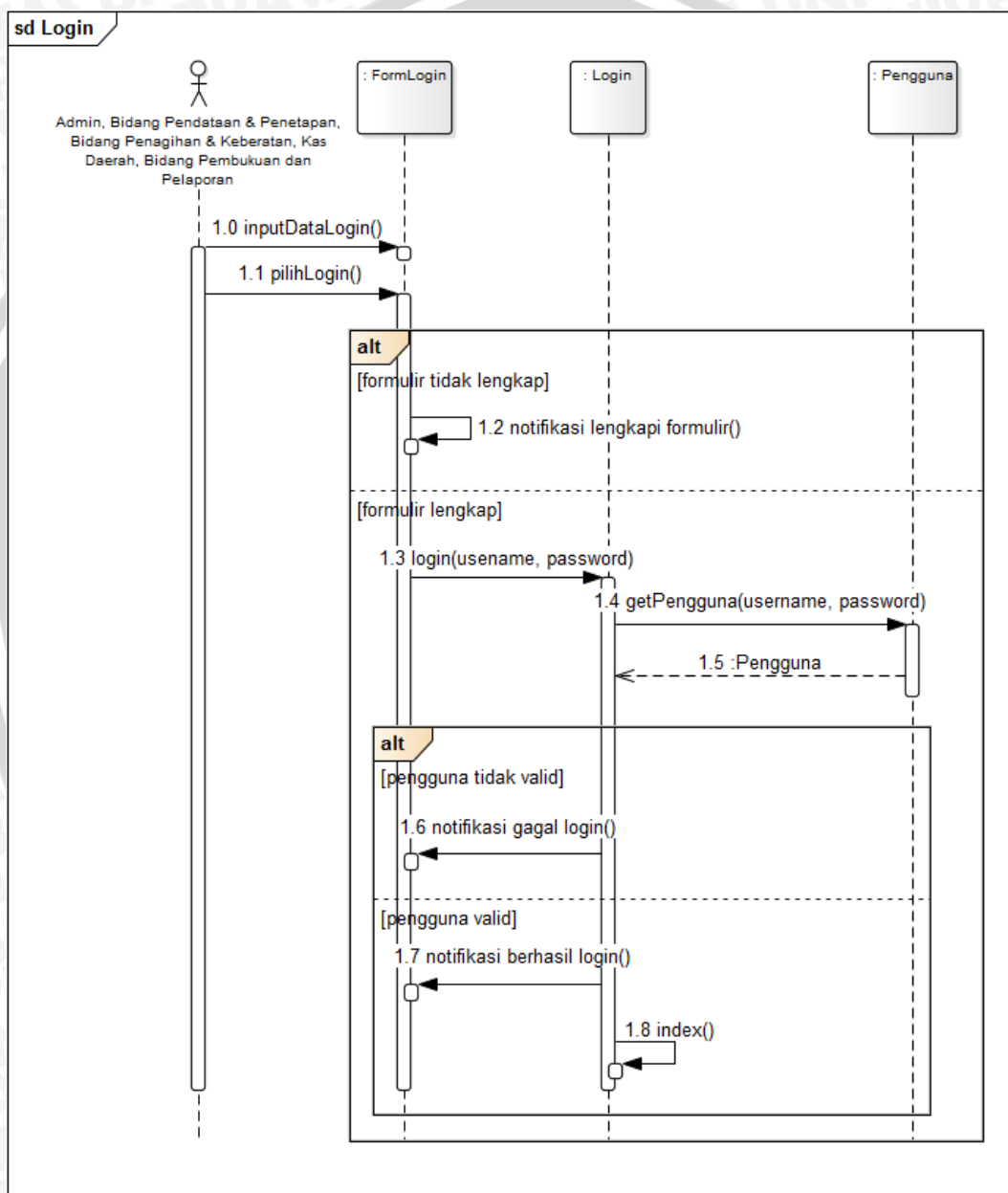


Gambar 5.3 Package

5.6 Diagram *sequence*

Diagram *sequence* menggambarkan interaksi antar objek dan menjelaskan mengenai urutan proses yang dilakukan dalam sistem untuk mencapai tujuan *use case*. Penggambaran interaksi ini disesuaikan dengan spesifikasi *use case*. Berikut ini adalah diagram *sequence* dari setiap *use case* :

1. Diagram *sequence use case login*



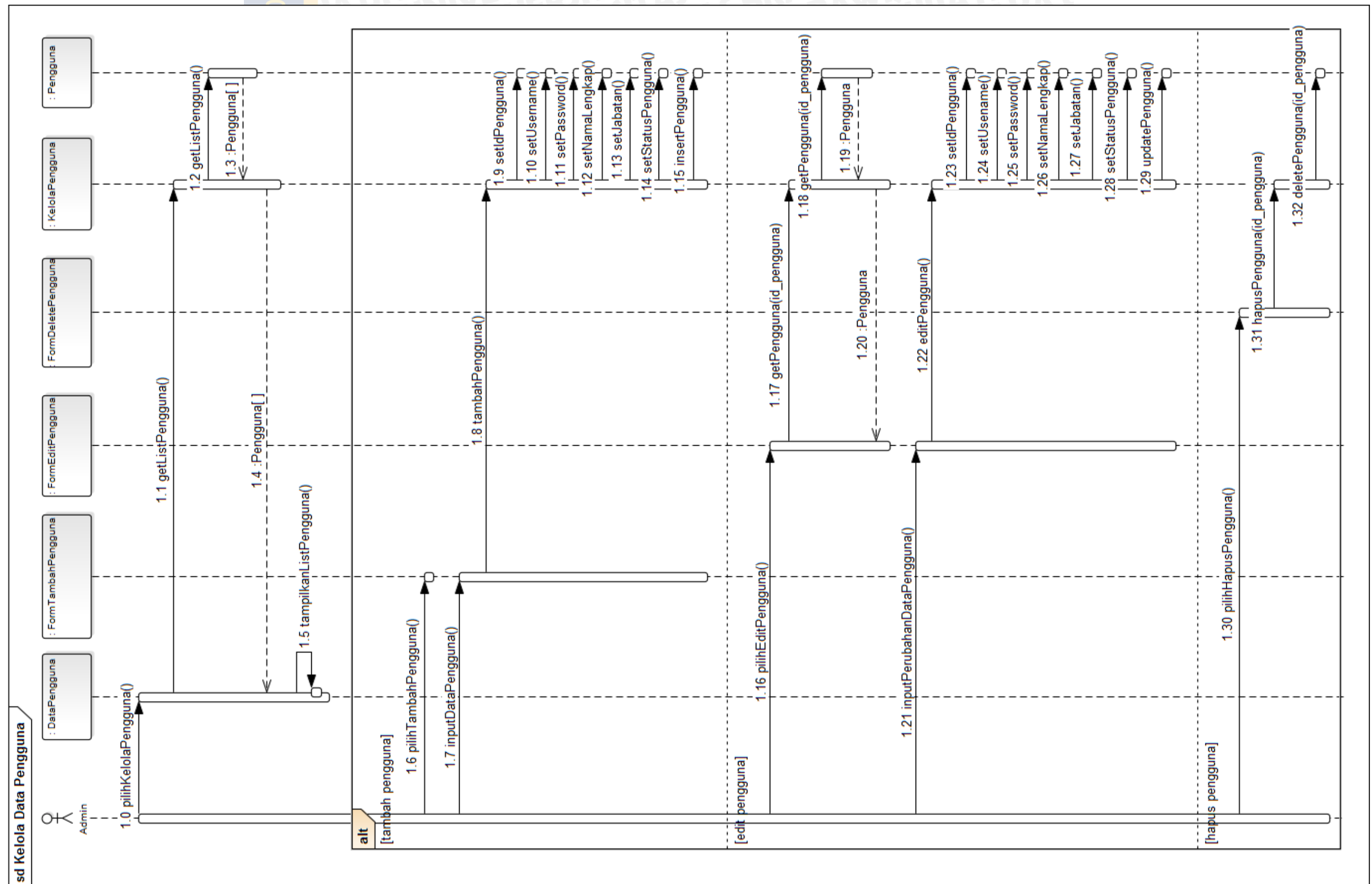
Gambar 5.4 *Sequence diagram login*

Gambar 2.1 menunjukkan pemodelan interaksi pada aktivitas *login*. Interaksi antar objek pada aktivitas ini diawali ketika aktor yaitu admin, bidang pendataan & penetapan, bidang penagihan & keberatan, kas daerah atau bidang pembukuan & pelaporan melakukan input data *login* dan memilih untuk melakukan *login* pada *view* FormLogin. Ketika aktor memilih untuk melakukan *login* sistem akan melakukan pengecekan apakah data *login* yang dimasukkan sudah lengkap. Jika data yang dimasukkan tidak lengkap (formulir tidak lengkap), maka objek FormLogin akan menampilkan notifikasi lengkapi formulir. Jika data yang dimasukkan sudah lengkap (formulir lengkap), maka objek FormLogin akan memanggil fungsi *login* milik objek *Login*. Objek *login* akan memanggil fungsi *getPengguna* pada objek *Pengguna* dan objek *Pengguna* akan mengembalikan data *Pengguna*. Setelah itu sistem akan melakukan pengecekan apakah data pengguna valid. Jika data pengguna tidak valid, maka sistem akan menampilkan notifikasi gagal *login*. Jika data pengguna valid, sistem akan menampilkan notifikasi berhasil *login* dan objek *login* akan menjalankan fungsi *index*.

2. Diagram *sequence use case* kelola data pengguna

Gambar 5.5 menunjukkan pemodelan interaksi dari aktivitas kelola data pengguna. Interaksi antar objek pada aktivitas kelola pengguna ini diawali ketika aktor admin memilih opsi kelola pengguna. Saat aktor memilih untuk melakukan kelola pengguna, objek *DataPengguna* memanggil fungsi *getListPengguna* ke objek *KelolaPengguna*. Objek *KelolaPengguna* akan memanggil fungsi *getListPengguna* ke objek *Pengguna* untuk mendapatkan data *Pengguna* yang terdaftar. Objek *Pengguna* kemudian akan mengembalikan data berupa *listPengguna* ke objek *KelolaPengguna*. Objek *KelolaPengguna* akan mengembalikan data ini ke objek *DataPengguna* untuk ditampilkan. Setelah sistem menampilkan data, pengguna aktor akan diberikan tiga opsi yang bisa dilakukan yaitu tambah pengguna, edit pengguna dan hapus pengguna.

Ketika aktor memilih opsi tambah pengguna maka aktor diharuskan memasukkan data pengguna yang akan ditambahkan pada *FormTambahPengguna*. Setelah aktor memasukkan data pengguna dan melakukan *submit*, *FormTambahPengguna* akan memanggil fungsi *tambahPengguna* ke objek *KelolaPengguna*. Objek *KelolaPengguna* kemudian akan memanggil fungsi *setIdPengguna*, *setUsername*, *setPassword*, *setNamaLengkap*, *setJabatan*, *setStatusPengguna* ke objek *Pengguna* untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek *Pengguna*. Untuk menambahkan data pengguna ke *database* objek *KelolaPengguna* kemudian memanggil fungsi *insertPengguna* ke objek *Pengguna*.



Gambar 5.5 Sequence diagram kelola data pengguna

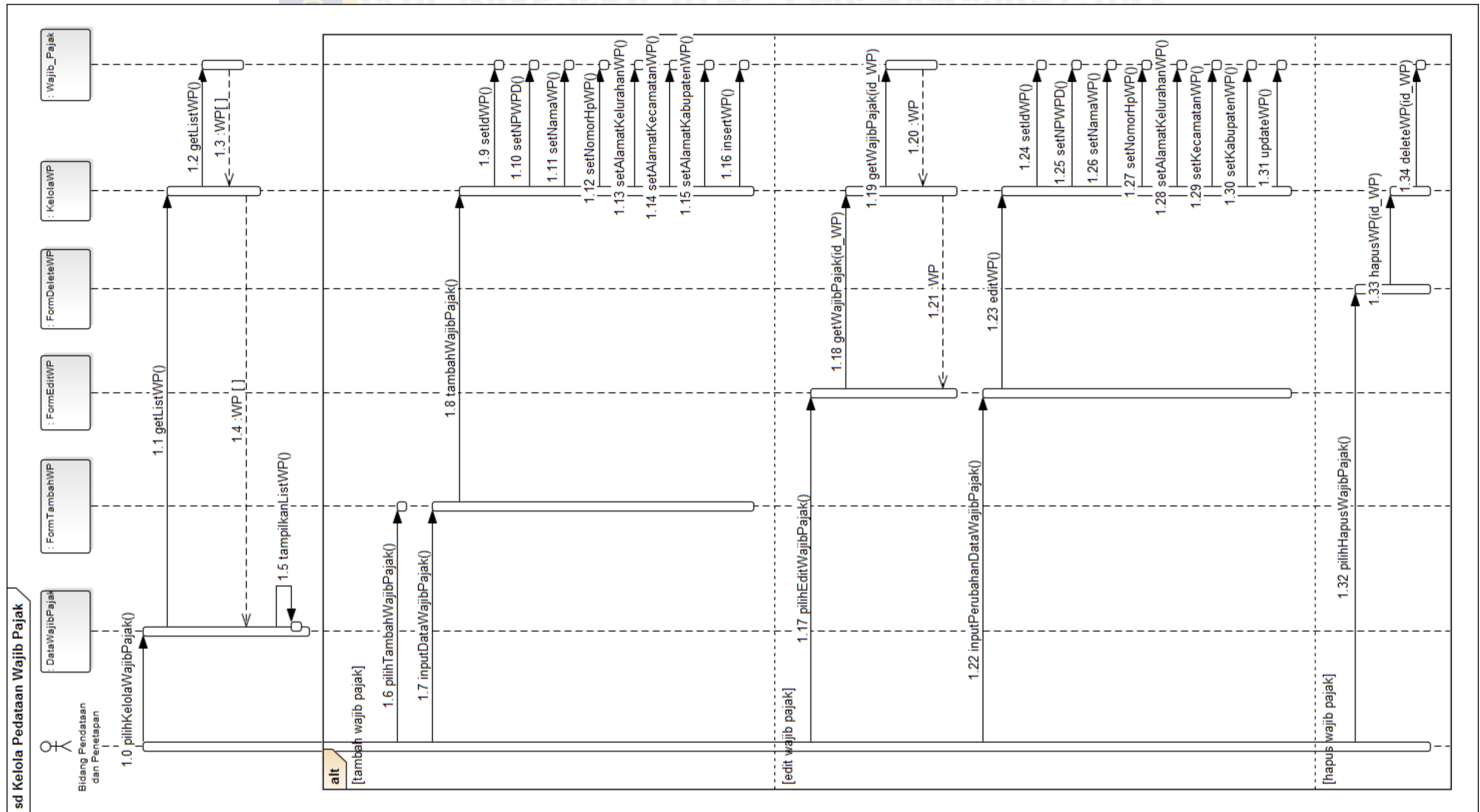
Ketika aktor memilih opsi edit pengguna pada salah satu pengguna maka `FormEditPengguna` akan memanggil fungsi `getPengguna` ke objek `KelolaPengguna`. Objek `KelolaPengguna` akan memanggil fungsi `getPengguna` ke objek `Pengguna` untuk mendapatkan data pengguna dari *database*. Objek `Pengguna` kemudian mengembalikan data pengguna ke objek `KelolaPengguna` untuk dikembalikan ke objek `FormEditPengguna`. Kemudian aktor akan mengubah data pada `FormEditPengguna`. Setelah selesai melakukan perubahan dan memilih untuk *submit*, objek `FormEditPengguna` akan memanggil fungsi `editPengguna` ke objek `KelolaPengguna`. Objek `KelolaPengguna` kemudian akan memanggil fungsi `setIdPengguna`, `setUsername`, `setPassword`, `setNamaLengkap`, `setJabatan`, `setStatusPengguna` ke objek `Pengguna` untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek `Pengguna`. Untuk mengubah data pengguna di *database* objek `KelolaPengguna` kemudian memanggil fungsi `updatePengguna` ke objek `Pengguna`.

Ketika aktor memilih untuk menghapus salah satu pengguna, `FormDeletePengguna` akan memanggil fungsi `hapusPengguna` ke objek `KelolaPengguna`. Untuk menghapus data pengguna pada *database* `KelolaPengguna` kemudian memanggil fungsi `deletePengguna` ke objek `Pengguna`.

3. Diagram *sequence use case* kelola pendataan wajib pajak

Gambar 5.6 menunjukkan interaksi antar objek pada aktivitas kelola pendataan wajib pajak. Aktor yang terlibat dalam aktivitas ini adalah Bidang Pendataan & Penetapan. Interaksi pada aktivitas ini diawali ketika aktor memilih opsi Kelola Wajib Pajak. Saat aktor memilih untuk melakukan Kelola Wajib Pajak, objek `DataWajibPajak` memanggil fungsi `getListWP` ke objek `KelolaWP`. Untuk mendapatkan data Wajib Pajak objek `KelolaWP` kemudian memanggil fungsi `getListWP` ke objek `Wajib_Pajak`. Objek `Wajib_Pajak` akan mengembalikan data-data Wajib Pajak ke objek `KelolaWP`. Objek `kelolaWP` akan mengirim data-data Wajib Pajak tersebut ke objek `DataWajibPajak` untuk ditampilkan. Setelah sistem menampilkan data Wajib Pajak, aktor akan diberikan tiga opsi yang dapat dilakukan yaitu tambah wajib pajak, edit wajib pajak dan hapus wajib pajak.

Ketika aktor memilih opsi tambah wajib pajak, maka aktor diharuskan memasukkan data wajib pajak yang akan ditambahkan pada `FormTambahWP`. Setelah aktor memasukkan data wajib pajak dan melakukan *submit*, `FormTambahWP` akan memanggil fungsi `tambahWajibPajak` ke objek `kelolaWP`. Objek `kelolaWP` kemudian akan memanggil fungsi `setIdWP`, `setNPWP`, `setNamaWP`, `setNomorHpWP`, `setAlamatKelurahanWP`, `setAlamatKecamatanWP` dan `setAlamatKabupatenWP` ke objek `Wajib_Pajak` untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek `Wajib_Pajak`. Untuk menambahkan data wajib pajak ke *database* objek `kelolaWP` kemudian memanggil fungsi `insertWP` ke objek `Wajib_Pajak`.



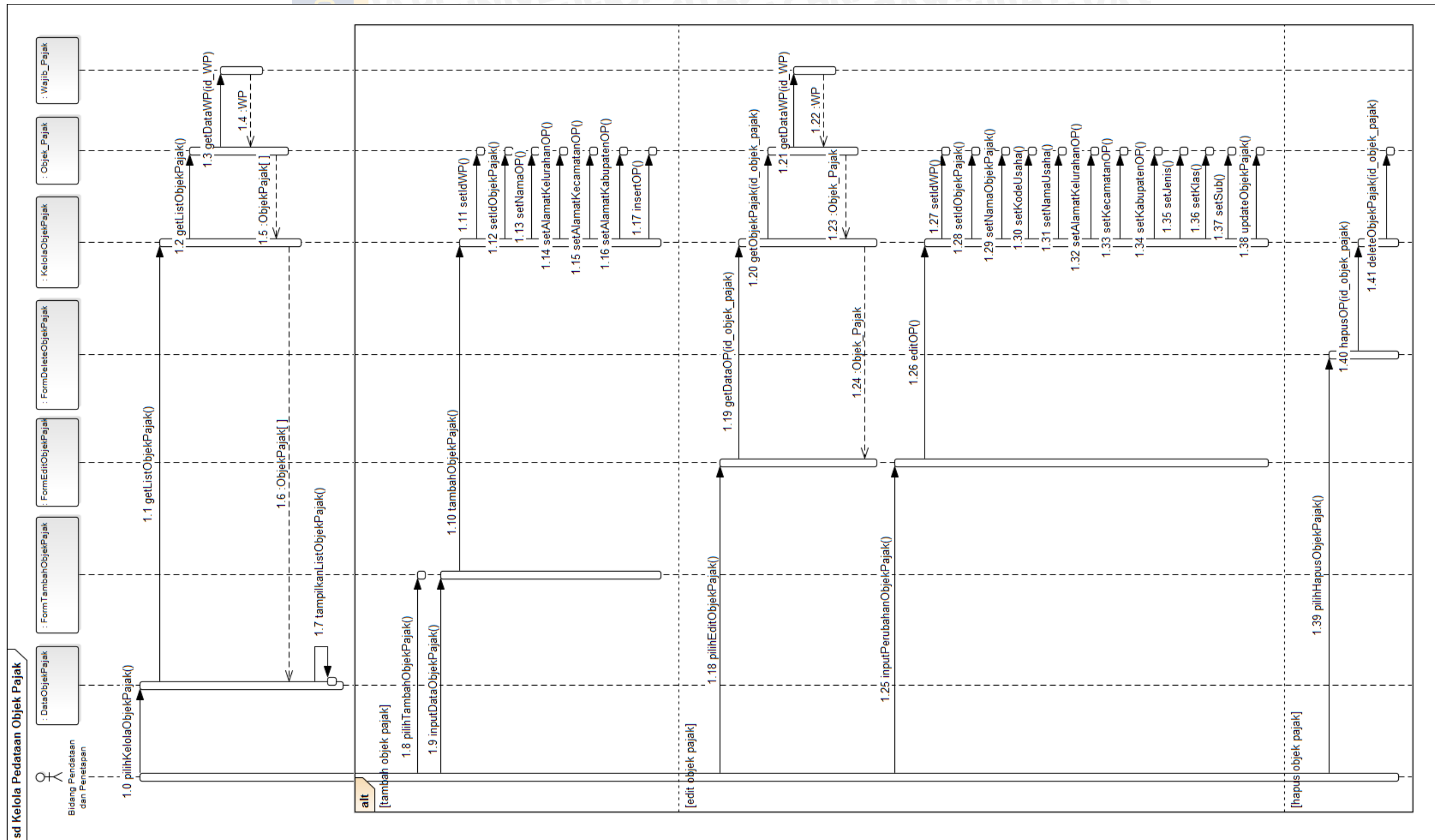
Gambar 5.6 Sequence diagram kelola pendataan wajib pajak

Ketika aktor memilih opsi edit wajib pajak pada salah satu wajib pajak dalam *list*, FormEditWP akan memanggil fungsi `getWajibPajak` ke objek KelolaWP. Objek kelolaWP akan memanggil fungsi `getWajibPajak` untuk mendapatkan data Wajib Pajak dari *database*. Objek Wajib_Pajak kemudian mengembalikan data wajib pajak ke objek kelolaWP untuk dikembalikan ke objek FormEditWP. Kemudian aktor akan mengubah data pada FormEditWP. Setelah selesai melakukan perubahan dan memilih untuk *submit*, objek FormEditWP akan memanggil fungsi `editWP` ke objek kelolaWP. Objek kelolaWP kemudian akan memanggil fungsi `setIdWP`, `setNPWPD`, `setNamaWP`, `setNomorHpWP`, `setAlamatKelurahanWP`, `setAlamatKecamatanWP` dan `setAlamatKabupatenWP` ke objek Wajib_Pajak untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek Wajib_Pajak. Untuk mengubah data wajib pajak di *database* objek kelolaWP kemudian memanggil fungsi `updateWP` ke objek Wajib_Pajak.

Ketika aktor memilih untuk menghapus salah satu wajib pajak, FormDeleteWP akan memanggil fungsi `hapusWP` ke objek kelolaWP. Untuk menghapus data wajib pajak pada *database* kelolaWP kemudian memanggil fungsi `deleteWP` ke objek Wajib_Pajak.

4. Diagram *sequence use case* kelola pendataan objek pajak

Gambar 5.7 menunjukkan interaksi antar objek pada aktivitas kelola pendataan objek pajak. Aktor yang terlibat dalam aktivitas ini adalah Bidang Pendataan & Penetapan. Interaksi pada aktivitas ini diawali ketika aktor memilih opsi Kelola Objek Pajak. Saat aktor memilih untuk melakukan Kelola Objek Pajak, objek DataObjekPajak memanggil fungsi `getListObjekPajak` ke objek KelolaObjekPajak. Untuk mendapatkan data Objek Pajak objek KelolaObjekPajak kemudian memanggil fungsi `getListObjekPajak` ke objek Objek_Pajak. Kemudian objek Objek_Pajak akan memanggil fungsi `getWajibPajak` ke objek Wajib_Pajak untuk mengetahui data wajib pajak dari setiap objek pajak. Objek Wajib_Pajak akan mengembalikan data-data Wajib Pajak ke objek Objek_Pajak dan objek Objek_Pajak akan mengembalikan list data Objek Pajak ke objek KelolaObjekPajak. Objek KelolaObjekPajak akan mengirim data-data Objek Pajak tersebut ke objek DataObjekPajak untuk ditampilkan. Setelah sistem menampilkan data Objek Pajak, aktor akan diberikan tiga opsi yang dapat dilakukan yaitu tambah objek pajak, edit objek pajak dan hapus objek pajak.



Gambar 5.7 Sequence diagram kelola pendataan objek pajak

Ketika aktor memilih opsi tambah objek pajak, maka aktor diharuskan memasukkan data objek pajak yang akan ditambahkan pada FormTambahObjekPajak. Setelah aktor memasukkan data objek pajak dan melakukan *submit*, FormTambahObjekPajak akan memanggil fungsi tambahObjekPajak ke objek kelolaObjekPajak. Objek kelolaObjekPajak kemudian akan memanggil fungsi setIdWP, setIdObjekPajak, setNameOP, setAlamatKelurahanOP, setAlamatKecamatanOP dan setAlamatKabupatenOP ke objek Objek_Pajak untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek Objek_Pajak. Untuk menambahkan data objek pajak ke *database* objek kelolaObjekPajak kemudian memanggil fungsi insertOP ke objek Objek_Pajak.

Ketika aktor memilih opsi edit objek pajak pada salah satu objek pajak dalam *list*, FormEditObjekPajak akan memanggil fungsi getDataOP ke objek KelolaObjekPajak. Objek kelolaObjekPajak akan memanggil fungsi getObjectPajak untuk mendapatkan data Objek Pajak dari *database*. Sedangkan untuk mengetahui wajib pajak dari objek pajak tersebut objek Objek_Pajak memanggil fungsi getWajibPajak ke objek Wajib_Pajak. Objek Wajib_Pajak akan mengembalikan data wajib pajak ke objek ObjekPajak dan objek Objek_Pajak akan mengembalikan data Objek Pajak untuk dikembalikan ke objek FormEditObjekPajak. Kemudian aktor akan mengubah data pada FormEditObjekPajak. Setelah selesai melakukan perubahan dan memilih untuk *submit*, objek FormEditObjekPajak akan memanggil fungsi editOP ke objek kelolaObjekPajak. Objek kelolaObjekPajak kemudian akan memanggil fungsi setIdWP, setIdObjekPajak, setNameOP, setAlamatKelurahanOP, setAlamatKecamatanOP dan setAlamatKabupatenOP ke objek Objek_Pajak untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek Objek_Pajak. Untuk mengubah data wajib pajak di *database* objek kelolaObjekPajak kemudian memanggil fungsi updateObjekPajak ke objek Objek_Pajak.

Ketika aktor memilih untuk menghapus salah satu objek pajak, FormDeleteObjekPajak akan memanggil fungsi hapusOP ke objek kelolaObjekPajak. Untuk menghapus data objek pajak pada *database* kelolaObjekPajak kemudian memanggil fungsi deleteObjekPajak ke objek Objek_Pajak.

5. Diagram *sequence use case* kelola penetapan pajak

Gambar 5.8 menunjukkan interaksi antar objek pada aktivitas kelola penetapan pajak. Aktor yang terlibat dalam aktivitas ini adalah Bidang Pendataan & Penetapan. Interaksi pada aktivitas ini diawali ketika aktor memilih opsi Kelola Penetapan Pajak kemudian memasukkan filter pencarian pajak dan memilih untuk melakukan pencarian. Objek DataPenetapanPajak memanggil fungsi getListSKPD ke objek C_SKPD. Untuk mendapatkan data Ketetapan Pajak objek C_SKPD kemudian memanggil fungsi getListSKPD ke objek SKPD. Kemudian objek SKPD akan memanggil fungsi getObjectPajak

pada objek Objek_Pajak untuk mengetahui detail objek pajak pada setiap SKPD dan Objek Objek_Pajak akan memanggil fungsi `getDataWP` ke objek Wajib_Pajak untuk mengetahui data wajib pajak dari setiap objek pajak. Objek Wajib_Pajak akan mengembalikan data-data Wajib Pajak ke objek Objek_Pajak, objek Objek_Pajak akan mengembalikan data Objek Pajak ke objek SKPD dan objek SKPD akan mengembalikan *list* SKPD ke `C_SKPD`. Objek `C_SKPD` akan mengirim data-data SKPD tersebut ke objek `DataPenetapanPajak` untuk ditampilkan. Setelah sistem menampilkan data Objek Pajak, aktor akan diberikan tiga opsi yang dapat dilakukan yaitu tambah ketentuan pajak, edit ketentuan pajak dan hapus ketentuan pajak.

Ketika aktor memilih opsi tambah ketentuan pajak, aktor diberikan pilihan akan menetapkan pajak air tanah atau pajak parkir.

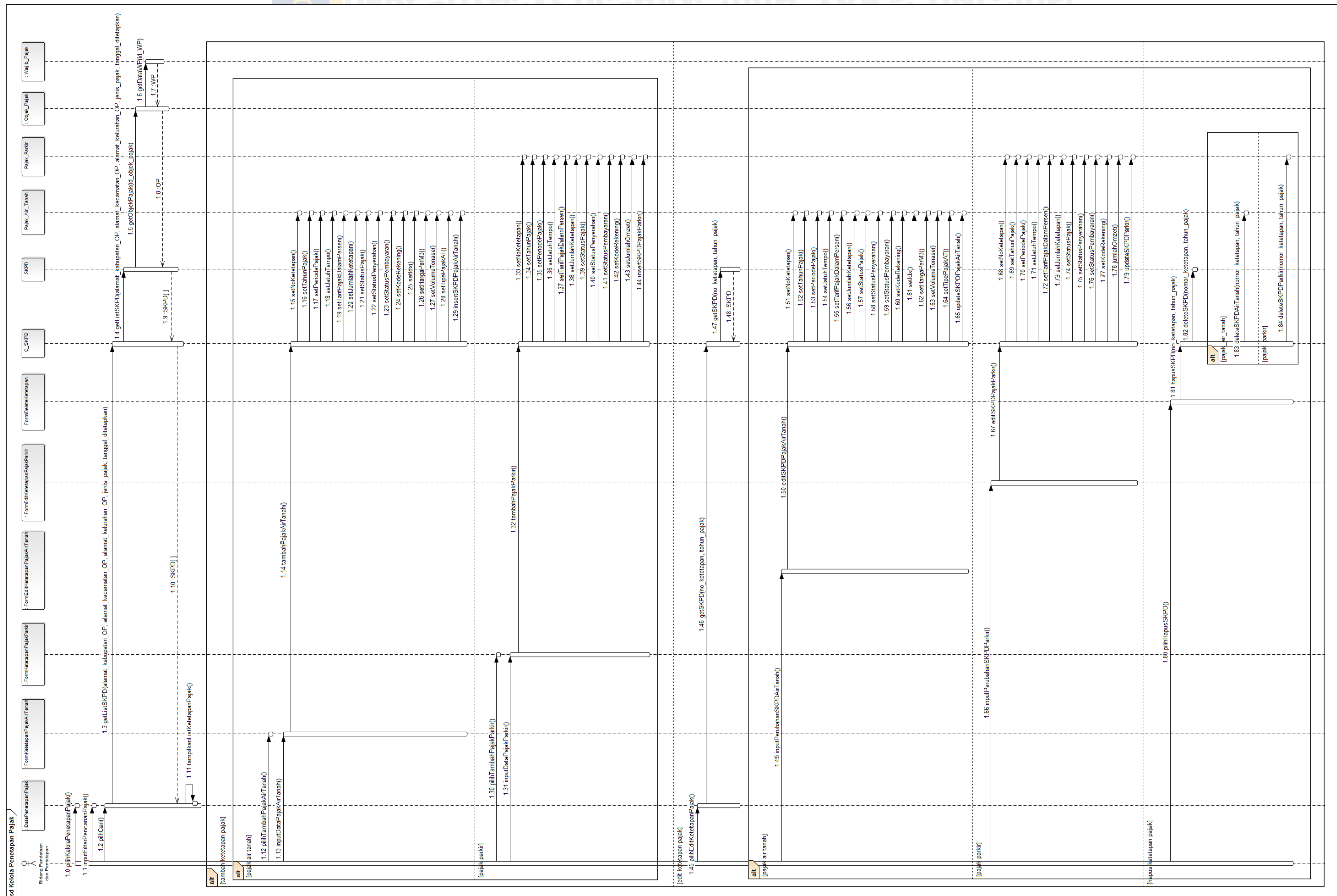
1. Jika aktor memilih menambah ketentuan pajak air tanah maka aktor diharuskan memasukkan data ketentuan pajak air tanah yang akan ditambahkan pada `FormKetetapanPajakAirTanah`. Setelah aktor memasukkan data ketentuan pajak air tanah dan melakukan *submit*, `FormKetetapanPajakAirTanah` akan memanggil fungsi `tambahPajakAirTanah` ke objek `C_SKPD`. Objek `C_SKPD` kemudian akan memanggil fungsi `setNoKetetapan`, `setTahunPajak`, `setPeriodePajak`, `setJatuhTempo`, `setTarifPajakDalamPersen`, `setJumlahKetetapan`, `setStatusPajak`, `setStatusPenyerahan`, `setStatusPembayaran`, `setKodeRekening`, `setIdx`, `setHargaPerM3`, `setVolumeTonase` dan `setTipePajakAT` ke objek `Pajak_Air_Tanah` untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek `Pajak_Air_Tanah`. Untuk menambahkan data ketentuan pajak air tanah ke *database* objek `C_SKPD` kemudian memanggil fungsi `insertSKPDPajakAirTanah` ke objek `SKPD`.
2. Jika aktor memilih menambah ketentuan pajak parkir maka aktor diharuskan memasukkan data ketentuan pajak parkir yang akan ditambahkan pada `FormKetetapanPajakParkir`. Setelah aktor memasukkan data ketentuan pajak air tanah dan melakukan *submit*, `FormKetetapanPajakParkir` akan memanggil fungsi `tambahPajakParkir` ke objek `C_SKPD`. Objek `C_SKPD` kemudian akan memanggil fungsi `setNoKetetapan`, `setTahunPajak`, `setPeriodePajak`, `setJatuhTempo`, `setTarifPajakDalamPersen`, `setJumlahKetetapan`, `setStatusPajak`, `setStatusPenyerahan`, `setStatusPembayaran`, `setKodeRekening`, dan `setJumlahOmzet` ke objek `Pajak_Parkir` untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek `Pajak_Parkir`. Untuk menambahkan data ketentuan pajak parkir ke *database* objek `C_SKPD` kemudian memanggil fungsi `insertSKPDPajakParkir` ke objek `SKPD`.

Ketika aktor memilih opsi edit ketentuan pajak pada salah satu ketentuan pajak dalam *list*, objek `DataPenetapanPajak` akan memanggil fungsi `getSKPD` ke objek `C_SKPD`. Untuk mendapatkan data dari *database* `C_SKPD` kemudian akan memanggil fungsi `getSKPD` pada objek `SKPD`. Objek `SKPD` akan

mengembalikan data SKPD ke C_SKPD. Data SKPD tersebut dapat berupa ketetapan pajak air tanah atau pajak parkir.

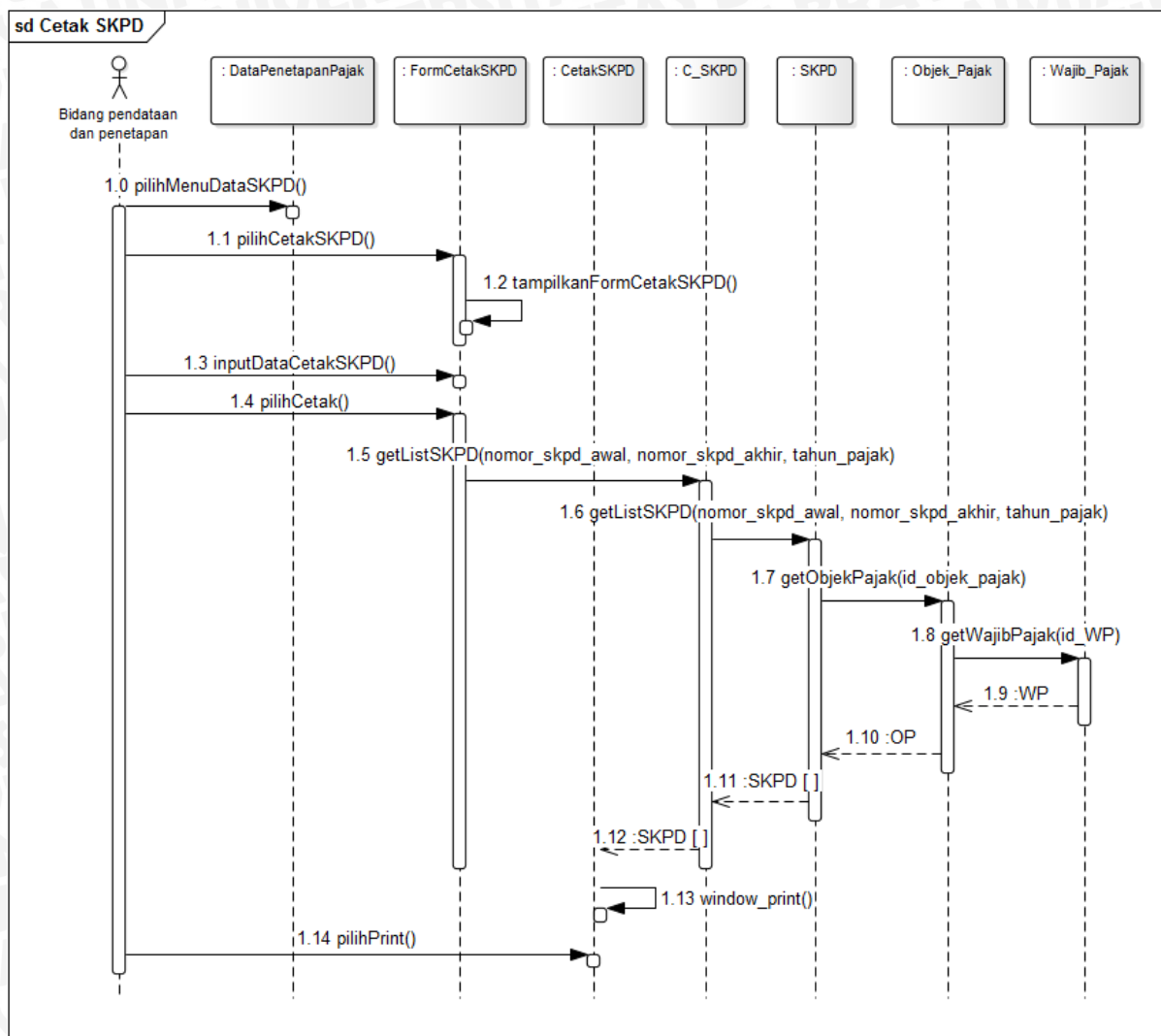
1. Jika data SKPD merupakan data ketetapan pajak air tanah, maka aktor akan mengubah data pada `FormEditKetetapanPajakAirTanah`. Setelah selesai melakukan perubahan dan memilih untuk *submit*, objek `FormEditKetetapanPajakAirTanah` akan memanggil fungsi `editSKPDPajakAirTanah` ke objek `C_SKPD`. Objek `C_SKPD` kemudian akan memanggil fungsi `setNoKetetapan`, `setTahunPajak`, `setPeriodePajak`, `setJatuhTempo`, `setTarifPajakDalamPersen`, `setJumlahKetetapan`, `setStatusPajak`, `setStatusPenyerahan`, `setStatusPembayaran`, `setKodeRekening`, `setIdx`, `setHargaPerM3`, `setVolumeTonase` dan `setTipePajakAT` ke objek `Pajak_Air_Tanah` untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek `Pajak_Air_Tanah`. Untuk mengubah data ketetapan pajak air tanah di *database* objek `C_SKPD` kemudian memanggil fungsi `updateSKPDAirTanah` ke objek `Pajak_Air_Tanah`.
2. Jika data SKPD merupakan data ketetapan pajak parkir, maka aktor akan mengubah data pada `FormEditKetetapanPajakParkir`. Setelah selesai melakukan perubahan dan memilih untuk *submit*, objek `FormEditKetetapanPajakParkir` akan memanggil fungsi `editSKPDPajakAirTanah` ke objek `C_SKPD`. Objek `C_SKPD` kemudian akan memanggil fungsi `setNoKetetapan`, `setTahunPajak`, `setPeriodePajak`, `setJatuhTempo`, `setTarifPajakDalamPersen`, `setJumlahKetetapan`, `setStatusPajak`, `setStatusPenyerahan`, `setStatusPembayaran`, `setKodeRekening`, dan `setJumlahOmzet` ke objek `Pajak_Parkir` untuk memberikan nilai pada setiap variabel yang dimiliki oleh objek `Pajak_Parkir`. Untuk mengubah data ketetapan pajak parkir di *database* objek `C_SKPD` kemudian memanggil fungsi `updateSKPDPajakParkir` ke objek `Pajak_Air_Tanah`.

Ketika aktor memilih untuk menghapus salah satu ketetapan pajak, `FormDeleteKetetapan` akan memanggil fungsi `hapusSKPD` ke objek `C_SKPD`. Jika SKPD yang dihapus adalah SKPD Pajak Air Tanah, untuk menghapus data ketetapan pajak air tanah pada *database* objek `C_SKPD` memanggil fungsi `deleteSKPDAirTanah` ke objek `Pajak_Air_Tanah`. Jika SKPD yang dihapus adalah SKPD Pajak Parkir, untuk menghapus data ketetapan pajak parkir pada *database* objek `C_SKPD` memanggil fungsi `deleteSKPDParkir` ke objek `Pajak_Parkir`.



Gambar 5.8 *Sequence diagram* kelola penetapan pajak

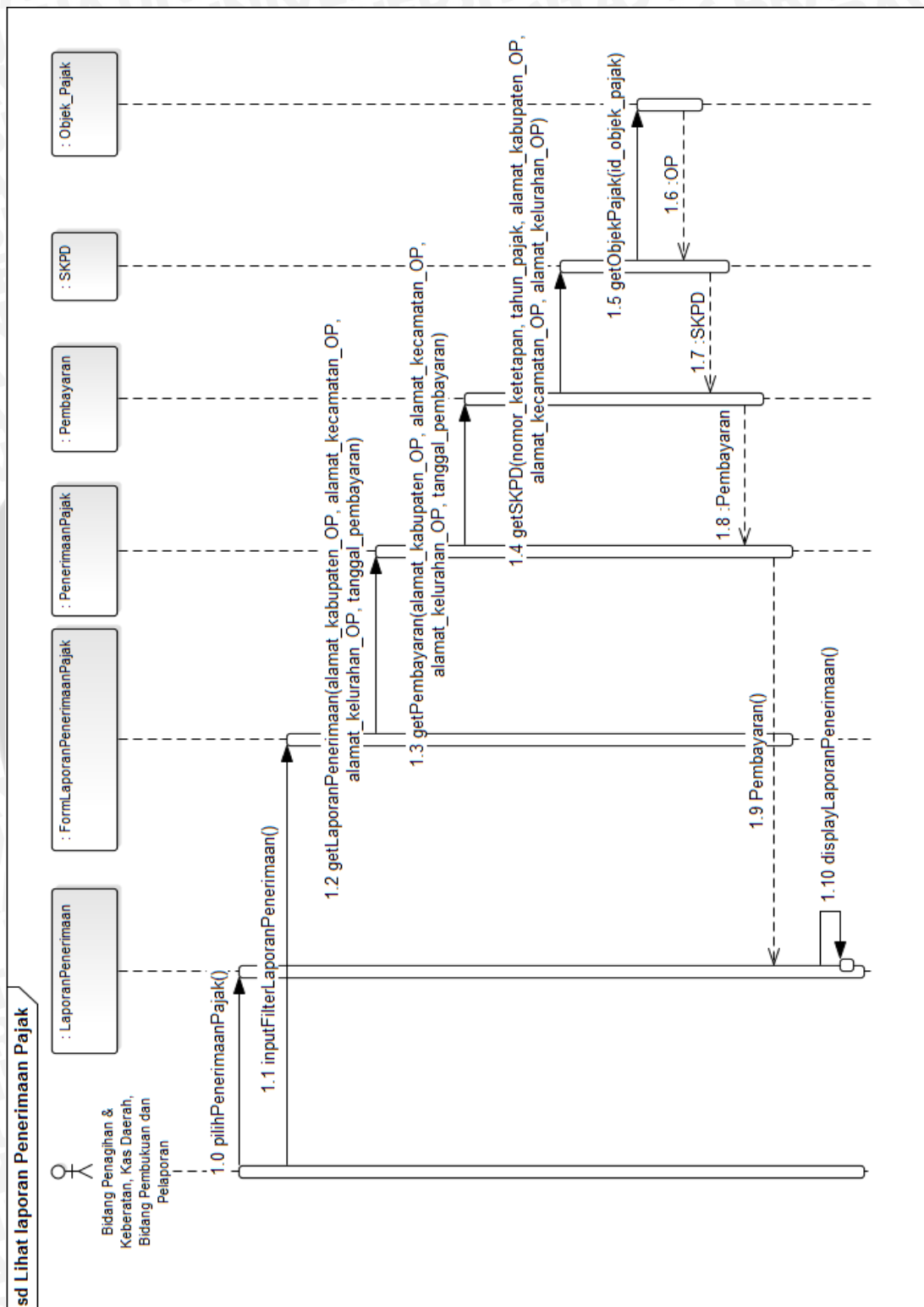
6. Diagram *sequence use case* cetak SKPD



Gambar 5.9 Sequence diagram cetak SKPD

Gambar 5.9 menunjukkan pemodelan interaksi dari aktivitas cetak SKPD. Aktor yang terlibat dalam aktivitas ini adalah bidang pendataan dan penetapan. Interaksi pada aktivitas ini dimulai ketika aktor memilih menu Data SKPD kemudian memilih opsi Cetak SKPD dan objek FormCetakSKPD akan menampilkan formulir untuk mencetak SKPD. Kemudian aktor memasukkan data yang dibutuhkan untuk mencetak SKPD pada FormCetakSKPD. Setelah itu aktor memilih cetak dan FormCetakSKPD akan memanggil fungsi getListSKPD pada objek C_SKPD. C_SKPD kemudian memanggil fungsi getListSKPD ke objek SKPD untuk mendapatkan data SKPD. Objek SKPD akan memanggil getObjectPajak ke objek Objek_Pajak untuk mendapatkan detail ObjekPajak dari SKPD dan ObjekPajak akan memanggil getWajibPajak ke objek Wajib_Pajak untuk mendapatkan detail Wajib Pajak. Wajib_Pajak akan mengembalikan data Wajib Pajak ke Objek_Pajak. Objek Pajak akan mengembalikan data Objek Pajak ke SKPD dan SKPD akan mengembalikan *list* SKPD ke objek SKPD. *List* SKPD akan dikirim ke objek CetakSKPD untuk bisa di cetak.

7. Diagram sequence use case lihat laporan penerimaan pajak

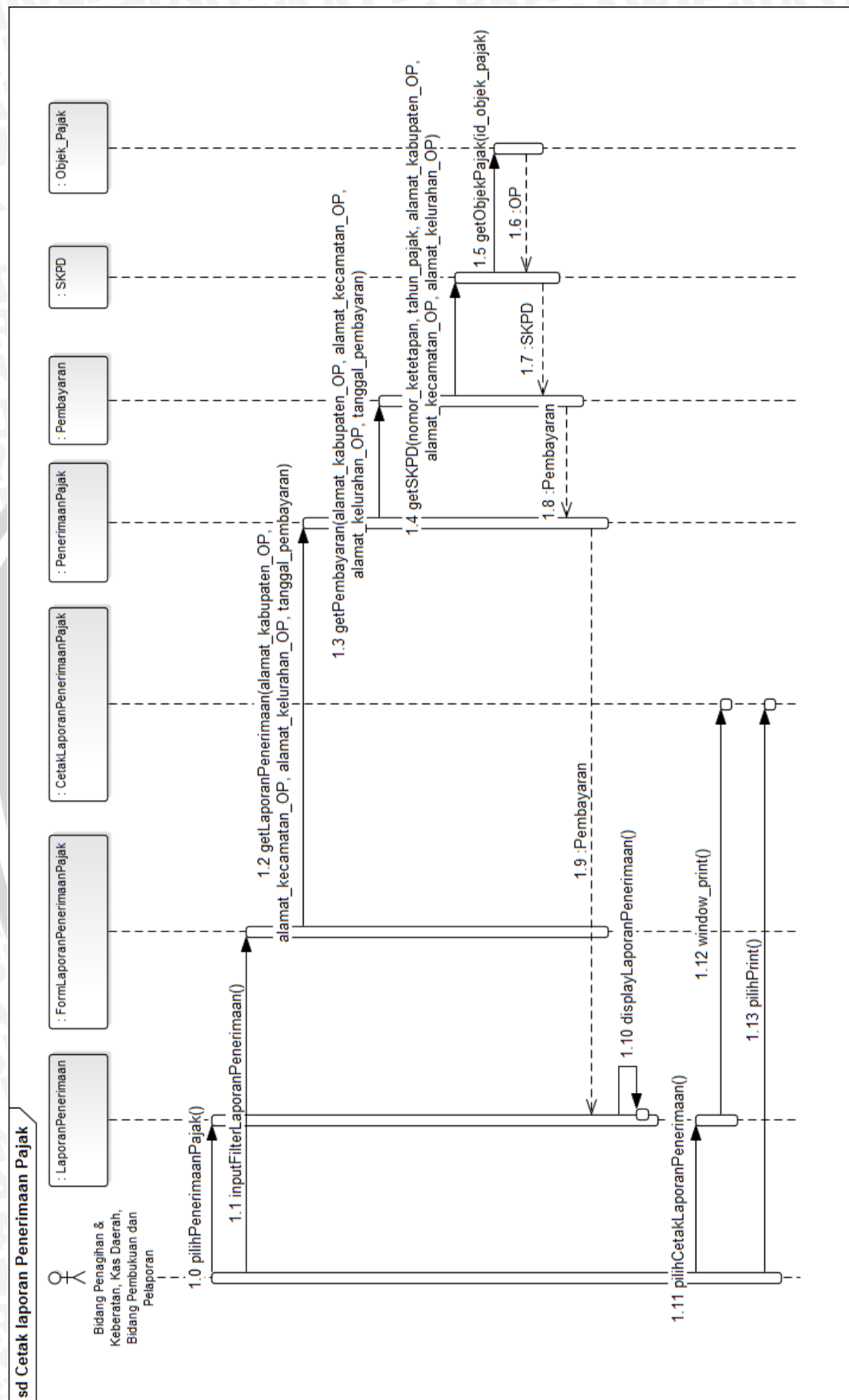


Gambar 5.10 Sequence diagram lihat laporan penerimaan pajak

Gambar 5.10 menunjukkan pemodelan interaksi dari aktivitas laporan penerimaan pajak. Aktor yang terlibat dalam aktivitas ini adalah Bidang Penagihan & Keberatan, Kas Daerah, Bidang Pembukuan & Pelaporan. Interaksi antar objek pada aktivitas ini diawali dengan aktor memilih untuk melihat Penerimaan Pajak kemudian melakukan filter laporan penerimaan pada FormLaporanPenerimaanPajak. FormLaporanPenerimaanPajak akan memanggil fungsi getLaporanPenerimaan ke objek PenerimaanPajak. Untuk mendapatkan data dari *database* objek PenerimaanPajak kemudian memanggil fungsi getPembayaran ke objek Pembayaran. Objek Pembayaran kemudian memanggil fungsi getSKPD ke SKPD untuk melengkapi detailnya. Objek SKPD kemudian akan memanggil fungsi getObjectPajak untuk melengkapi detail SKPD. Objek Pajak akan mengembalikan data Objek Pajak ke objek SKPD, objek SKPD akan mengembalikan data SKPD ke objek Pembayaran dan objek Pembayaran akan mengembalikan data Pembayaran ke objek PenerimaanPajak. Data Pembayaran ini akan dikirim ke objek LaporanPenerimaan untuk ditampilkan.

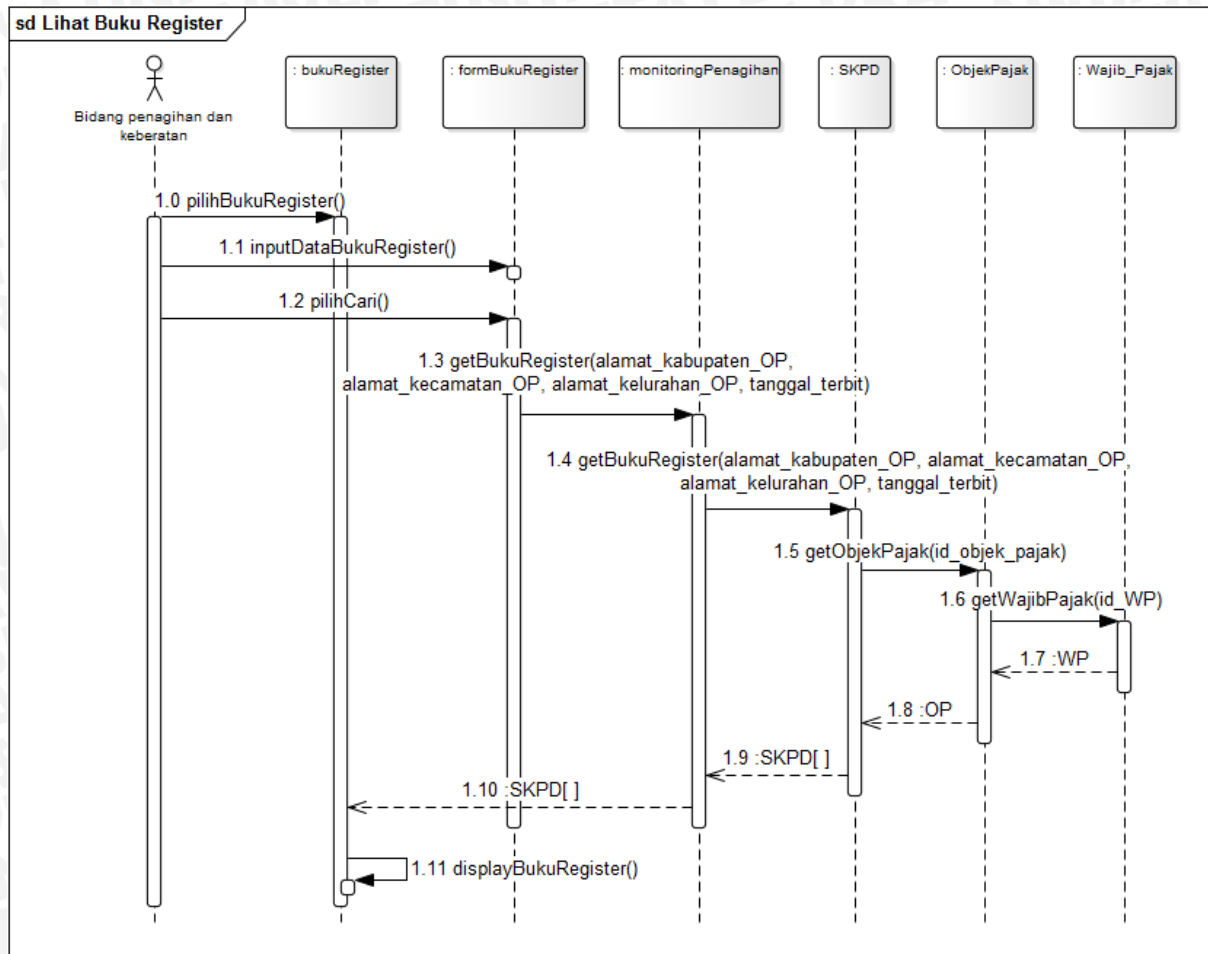
8. Diagram *sequence use case* cetak laporan penerimaan pajak

Gambar 5.11 menunjukkan pemodelan interaksi dari aktivitas cetak laporan penerimaan pajak. Aktor yang terlibat dalam aktivitas ini adalah Bidang Penagihan & Keberatan, Kas Daerah, Bidang Pembukuan & Pelaporan. Interaksi antar objek pada aktivitas ini diawali dengan aktor memilih untuk melihat Penerimaan Pajak kemudian melakukan filter laporan penerimaan pada FormLaporanPenerimaanPajak. FormLaporanPenerimaanPajak akan memanggil fungsi getLaporanPenerimaan ke objek PenerimaanPajak. Untuk mendapatkan data dari *database* objek PenerimaanPajak kemudian memanggil fungsi getPembayaran ke objek Pembayaran. Objek Pembayaran kemudian memanggil fungsi getSKPD ke SKPD untuk melengkapi detailnya. Objek SKPD kemudian akan memanggil fungsi getObjectPajak untuk melengkapi detail SKPD. Objek Pajak akan mengembalikan data Objek Pajak ke objek SKPD, objek SKPD akan mengembalikan data SKPD ke objek Pembayaran dan objek Pembayaran akan mengembalikan data Pembayaran ke objek PenerimaanPajak. Data Pembayaran ini akan dikirim ke objek LaporanPenerimaan untuk ditampilkan. Kemudian aktor akan memilih cetak laporan penerimaan. Objek LaporanPenerimaan akan memanggil objek CetakLaporanPenerimaanPajak untuk mencetak laporan penerimaan.



Gambar 5.11 Sequence diagram cetak laporan penerimaan pajak

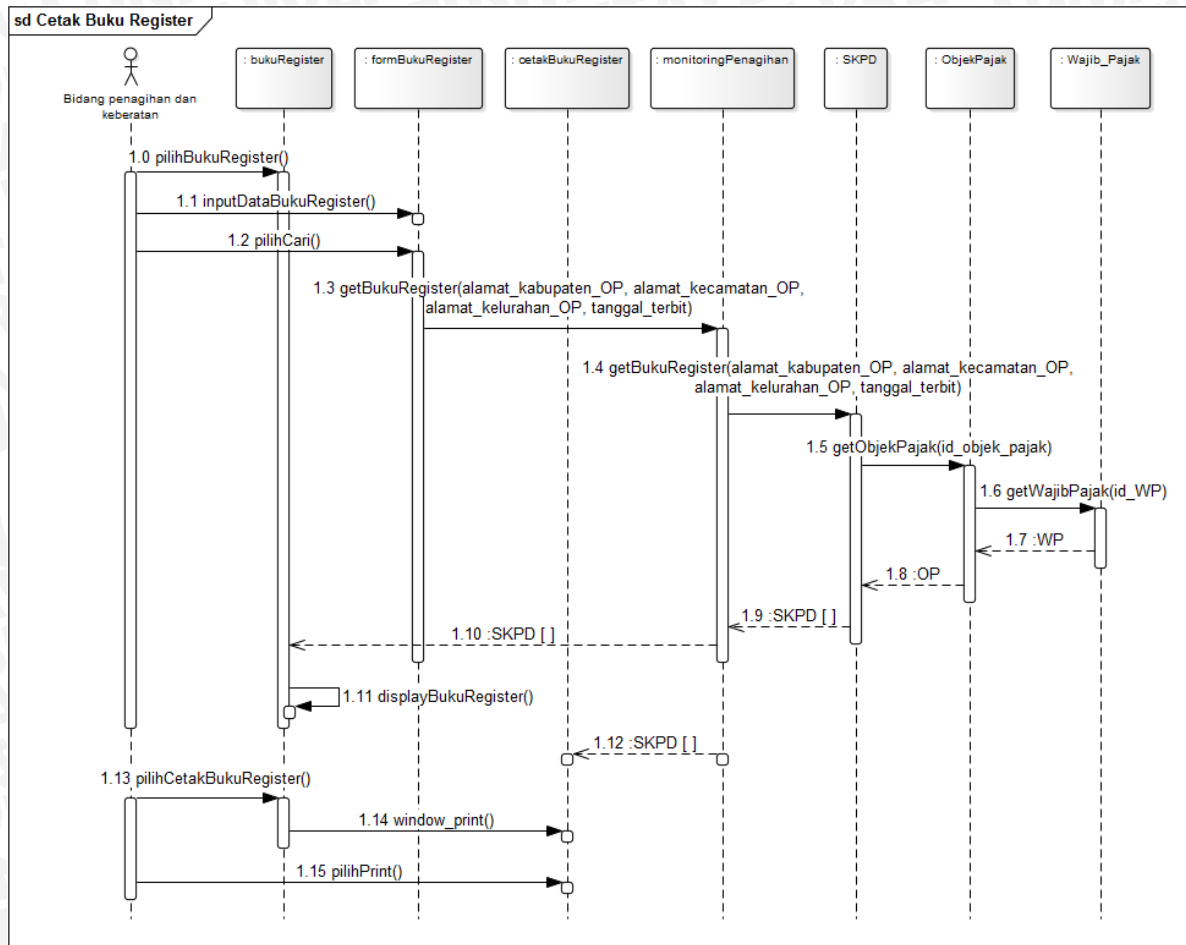
9. Diagram *sequence use case* lihat buku register



Gambar 5.12 Sequence diagram lihat buku register

Gambar 5.12 menunjukkan interaksi antar objek pada aktivitas lihat buku register. Aktor yang terlibat dalam aktivitas ini adalah bidang penagihan dan keberatan. Aktivitas diawali ketika aktor memilih opsi Buku Register kemudian memasukkan filter data buku register pada FormBukuRegister dan memilih melakukan pencarian. FormBukuRegister kemudian akan memanggil fungsi `getBukuRegister` pada objek `monitoringPenagihan`. Untuk mendapatkan data dari *database* objek `monitoringPenagihan` akan memanggil fungsi `getBukuRegister` pada objek `SKPD`. Objek `SKPD` akan memanggil fungsi `getObjectPajak` ke Objek_Pajak untuk melengkapi detailnya dan Objek_Pajak akan memanggil fungsi `getWajibPajak` ke objek `Wajib_Pajak` untuk melengkapi detail siapa yang wajib pajaknya. `Wajib_Pajak` akan mengembalikan data `Wajib Pajak` ke Objek_Pajak, Objek_Pajak akan mengembalikan data Objek Pajak ke SKPD dan SKPD akan mengembalikan daftar SKPD sesuai filter ke objek `monitoringPenagihan`. `monitoringPenagihan` akan mengirim daftar SKPD ke objek `bukuRegister` untuk ditampilkan.

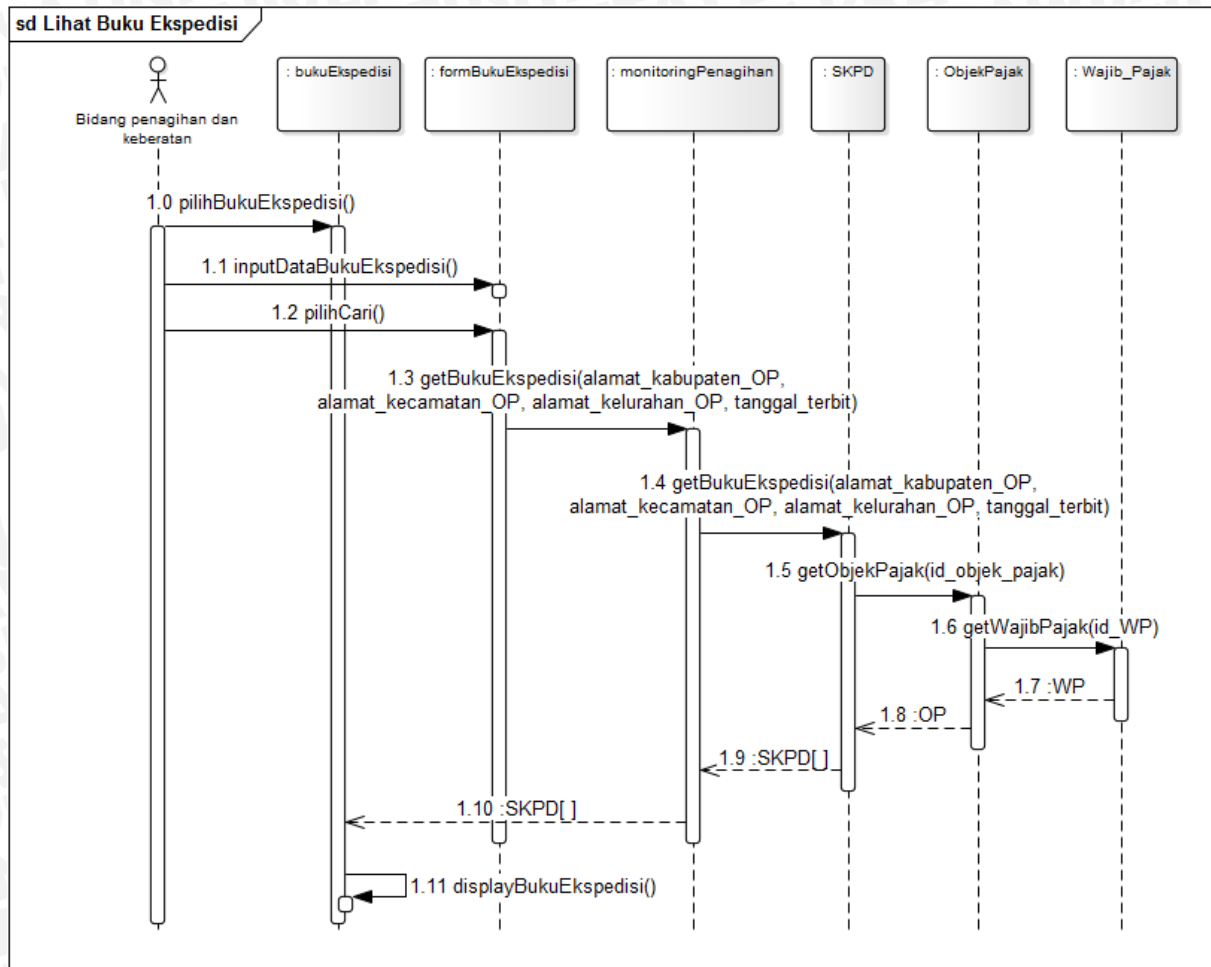
10. Diagram *sequence use case* cetak buku register



Gambar 5.13 Sequence diagram cetak buku register

Gambar 5.13 menunjukkan interaksi antar objek pada aktivitas cetak buku register. Aktor yang terlibat dalam aktivitas ini adalah bidang penagihan dan keberatan. Aktivitas diawali ketika aktor memilih opsi Buku Register kemudian memasukkan filter data buku register pada FormBukuRegister dan memilih melakukan pencarian. FormBukuRegister kemudian akan memanggil fungsi `getBukuRegister` pada objek `monitoringPenagihan`. Untuk mendapatkan data dari *database* objek `monitoringPenagihan` akan memanggil fungsi `getBukuRegister` pada objek `SKPD`. Objek `SKPD` akan memanggil fungsi `getObjectPajak` ke Objek_Pajak untuk melengkapi detailnya dan Objek_Pajak akan memanggil fungsi `getWajibPajak` ke objek `Wajib_Pajak` untuk melengkapi detail siapa yang wajib pajaknya. `Wajib_Pajak` akan mengembalikan data `Wajib Pajak` ke `Objek_Pajak`, `Objek_Pajak` akan mengembalikan data `Objek Pajak` ke `SKPD` dan `SKPD` akan mengembalikan daftar `SKPD` sesuai filter ke objek `monitoringPenagihan`. `monitoringPenagihan` akan mengirim daftar `SKPD` ke objek `bukuRegister` untuk ditampilkan. Kemudian aktor memilih untuk mencetak Register. Objek `bukuRegister` akan memanggil `cetakBukuRegister` untuk melakukan cetak buku register.

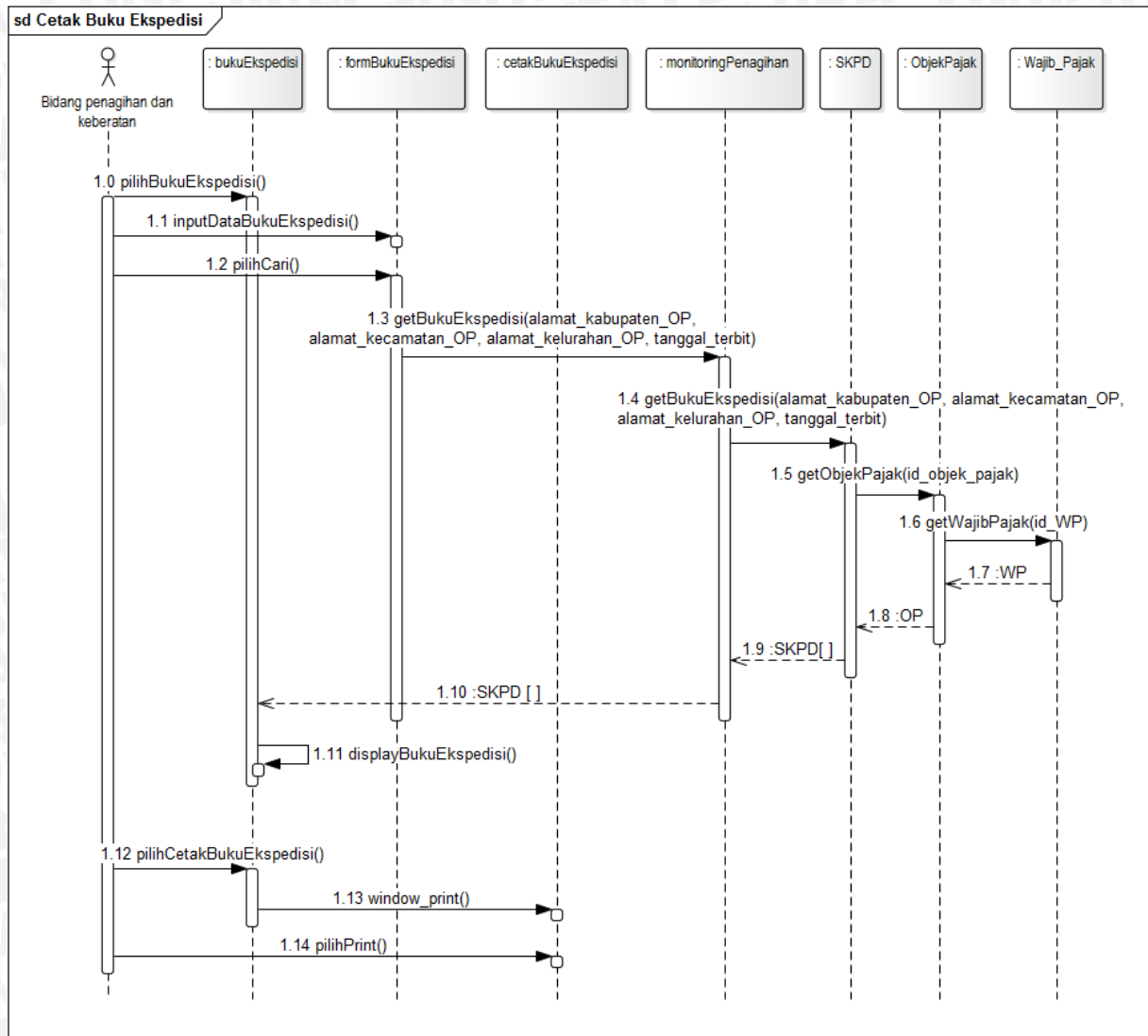
11. Diagram *sequence use case* lihat buku ekspedisi



Gambar 5.14 Sequence diagram lihat buku ekspedisi

Gambar 5.14 menunjukkan interaksi antar objek pada aktivitas lihat buku ekspedisi. Aktor yang terlibat dalam aktivitas ini adalah bidang penagihan dan keberatan. Aktivitas diawali ketika aktor memilih opsi Buku Ekspedisi kemudian memasukkan filter data buku ekspedisi pada FormBukuEkspedisi dan memilih melakukan pencarian. FormBukuEkspedisi kemudian akan memanggil fungsi `getBukuEkspedisi` pada objek `monitoringPenagihan`. Untuk mendapatkan data dari *database* objek `monitoringPenagihan` akan memanggil fungsi `getBukuEkspedisi` pada objek `SKPD`. Objek `SKPD` akan memanggil fungsi `getObjectPajak` ke Objek_Pajak untuk melengkapi detailnya dan Objek_Pajak akan memanggil fungsi `getWajibPajak` ke objek `Wajib_Pajak` untuk melengkapi detail siapa yang wajib pajaknya. `Wajib_Pajak` akan mengembalikan data `Wajib Pajak` ke `Objek_Pajak`, `Objek_Pajak` akan mengembalikan data `Objek Pajak` ke `SKPD` dan `SKPD` akan mengembalikan daftar `SKPD` sesuai filter ke objek `monitoringPenagihan`. `monitoringPenagihan` akan mengirim daftar `SKPD` ke objek `bukuEkspedisi` untuk ditampilkan.

12. Diagram *sequence use case* cetak buku ekspedisi

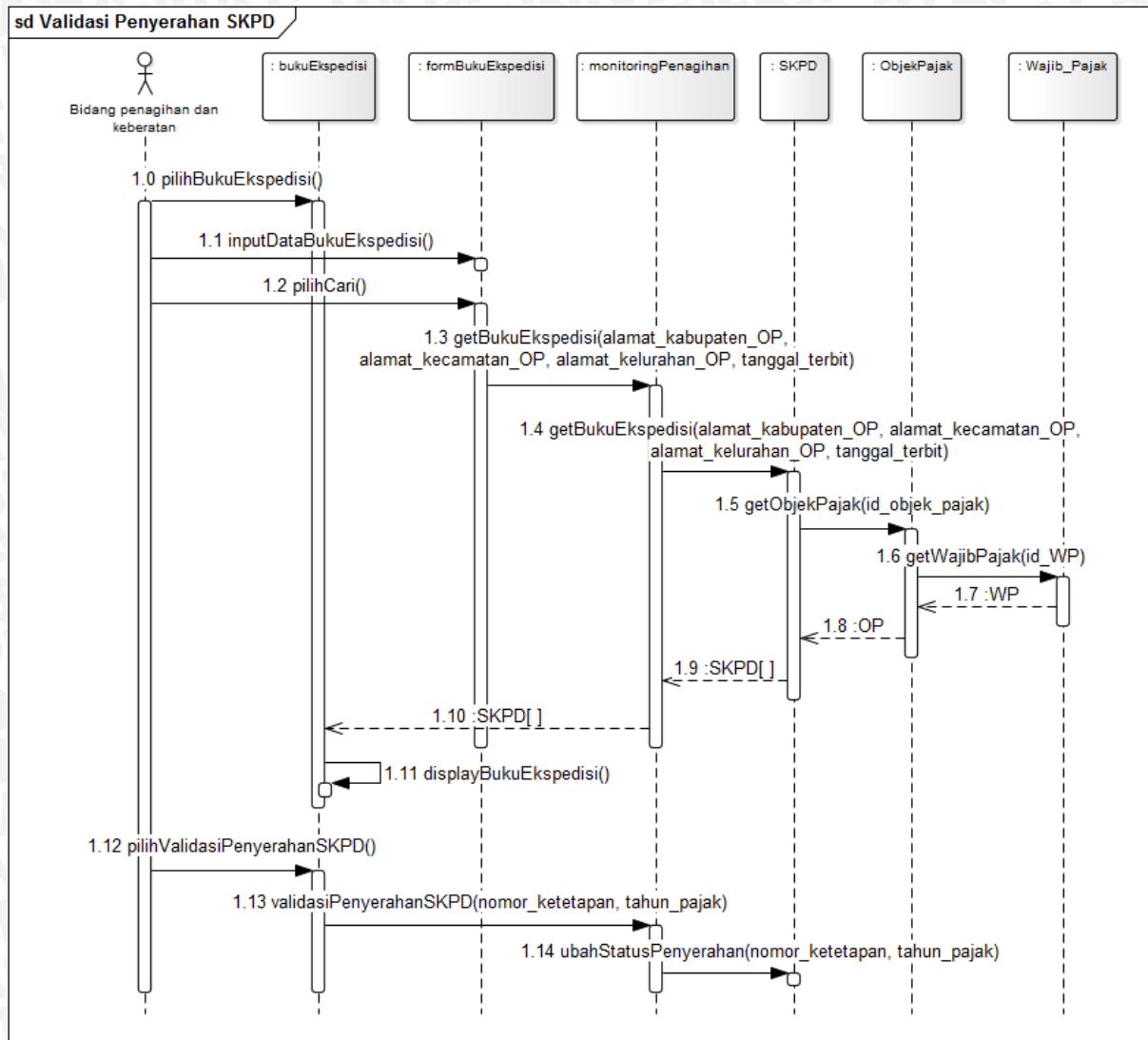


Gambar 5.15 Sequence diagram cetak buku ekspedisi

Gambar 5.15 menunjukkan interaksi antar objek pada aktivitas cetak buku ekspedisi. Aktor yang terlibat dalam aktivitas ini adalah bidang penagihan dan keberatan. Aktivitas diawali ketika aktor memilih opsi Buku Ekspedisi kemudian memasukkan filter data buku ekspedisi pada FormBukuEkpedisi dan memilih melakukan pencarian. FormBukuEkspedisi kemudian akan memanggil fungsi `getBukuEkspedisi` pada objek `monitoringPenagihan`. Untuk mendapatkan data dari *database* objek `monitoringPenagihan` akan memanggil fungsi `getBukuEkspedisi` pada objek `SKPD`. Objek `SKPD` akan memanggil fungsi `getObjectPajak` ke Objek_Pajak untuk melengkapi detailnya dan Objek_Pajak akan memanggil fungsi `getWajibPajak` ke objek `Wajib_Pajak` untuk melengkapi detail siapa yang wajib pajaknya. `Wajib_Pajak` akan mengembalikan data `Wajib Pajak` ke Objek_Pajak, Objek_Pajak akan mengembalikan data Objek Pajak ke SKPD dan SKPD akan mengembalikan daftar SKPD sesuai filter ke objek `monitoringPenagihan`. `monitoringPenagihan`

akan mengirim daftar SKPD ke objek bukuEkspedisi untuk ditampilkan. Kemudian aktor memilih untuk mencetak bukuEkspedisi. Objek bukuEkspedisi akan memanggil cetakBukuEkspedisi untuk melakukan cetak buku ekspedisi.

13. Diagram *sequence use case* validasi penyerahan SKPD

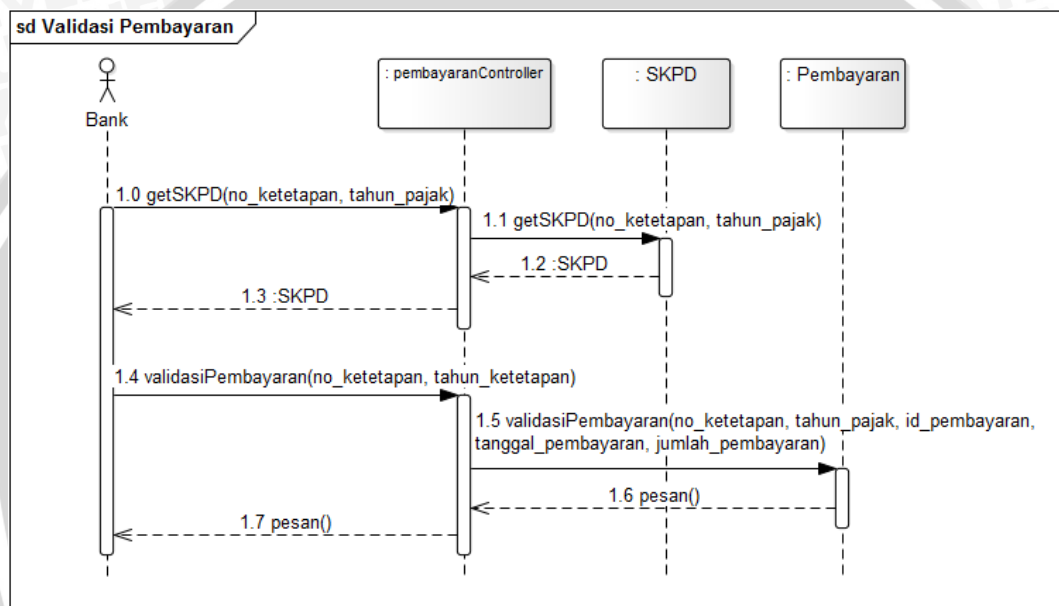


Gambar 5.16 Sequence Diagram Validasi Penyerahan SKPD

Gambar 5.16 menunjukkan interaksi antar objek pada proses validasi penyerahan SKPD. Aktor yang terlibat dalam aktivitas ini adalah bidang penagihan & keberatan. Aktivitas diawali ketika aktor memilih opsi Buku Ekspedisi kemudian memasukkan filter data buku ekspedisi pada FormBukuEkspedisi dan memilih melakukan pencarian. FormBukuEkspedisi kemudian akan memanggil fungsi getBukuEkspedisi pada objek monitoringPenagihan. Untuk mendapatkan data dari *database* objek monitoringPenagihan akan memanggil fungsi getBukuEkspedisi pada objek SKPD. Objek SKPD akan memanggil fungsi getObjectPajak ke Objek_Pajak untuk melengkapi detailnya dan Objek_Pajak akan memanggil fungsi getWajibPajak ke objek Wajib_Pajak untuk melengkapi detail siapa yang wajib

pajaknya. Wajib_Pajak akan mengembalikan data Wajib Pajak ke Objek_Pajak, Objek_Pajak akan mengembalikan data Objek Pajak ke SKPD dan SKPD akan mengembalikan daftar SKPD sesuai filter ke objek monitoringPenagihan. monitoringPenagihan akan mengirim daftar SKPD ke objek bukuEkspedisi untuk ditampilkan. Setelah tampil buku Ekspedisi aktor akan memilih validasi Penyerahan SKPD pada salah satu SKPD pada buku ekspedisi. Objek bukuEkspedisi kemudian akan memanggil fungsi validasiPenyerahanSKPD ke objek monitoringPenagihan. Kemudian objek monitoringPenagihan memanggil fungsi ubahStatusPenyerahan ke objek SKPD untuk mengubah status SKPD pada *database*.

14. Diagram *sequence use case* validasi pembayaran



Gambar 5.17 Sequence diagram validasi pembayaran

Gambar 5.17 menunjukkan pemodelan interaksi pada proses validasi pembayaran SKPD oleh pihak Bank. Pada pemodelan interaksi ini Sistem Bank mengakses *service* yang disediakan oleh Dispenda. Interaksi dimulai ketika Bank menggunakan layanan *getSKPD* pada objek *pembayaranController* untuk mencari SKPD yang akan divalidasi pembayarannya. *PembayaranController* kemudian memanggil fungsi *getSKPD* pada objek *SKPD* dan objek *SKPD* mengembalikan data *SKPD* ke *PembayaranController*. *PembayaranController* kemudian akan mengirim data tersebut ke Bank. Setelah itu Bank menggunakan layanan *validasiPembayaran* pada *PembayaranController* untuk melakukan validasi pembayaran pajak. Kemudian *PembayaranController* akan memanggil fungsi *validasiPembayaran* milik objek *Pembayaran*. Objek *Pembayaran* akan mengirimkan pesan berhasil di validasi ke *PembayaranController* dan *PembayaranController* akan mengirim pesan tersebut ke Bank.

5.7 Diagram Kelas

Perancangan kelas diagram digunakan untuk menggambarkan keadaan (atribut/properti) suatu sistem dan layanan untuk memanipulasi atribut/properti tersebut dalam bentuk *method/fungsi*. Kelas diagram ini diperoleh dari menyempurnakan kelas analisis yang telah didefinisikan sebelumnya. Berikut ini adalah perancangan kelas diagram sistem yang terdiri dari diagram kelas bertipe model, diagram kelas bertipe *controller*, dan diagram kelas bertipe *view*.

a. Diagram kelas bertipe model

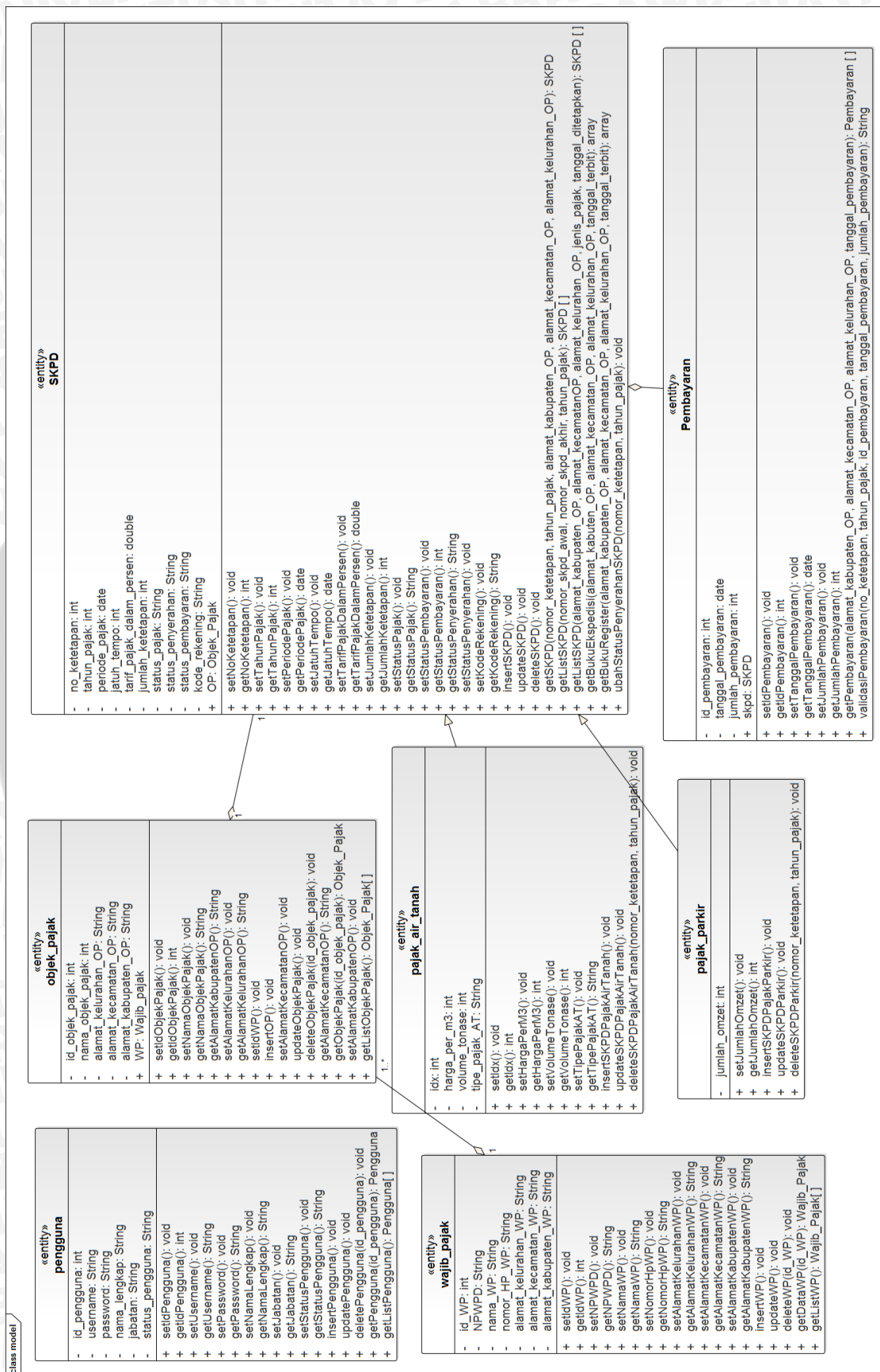
Kelas bertipe model merupakan kelas yang berhubungan dengan *database*. *Class diagram* model ini diperoleh dari perbaikan kelas analisis yang memiliki *stereotype* <<entity>>. Kelas-kelas *entity* ini berasal dari identifikasi objek pada *use case*, atribut-atribut yang dimiliki setiap kelas diperoleh dari *glossary* pada spesifikasi *use case* dan fungsi-fungsi yang dimiliki berasal dari *sequence diagram* yang telah dirancang sebelumnya. Kelas diagram model/*entity* pada Gambar 5.18 Diagram kelas model berikut ini terdiri dari kelas pengguna, wajib_pajak, objek_pajak, SKPD, pajak_air_tanah, pajak_parkir, dan Pembayaran.

b. Diagram kelas bertipe *controller*

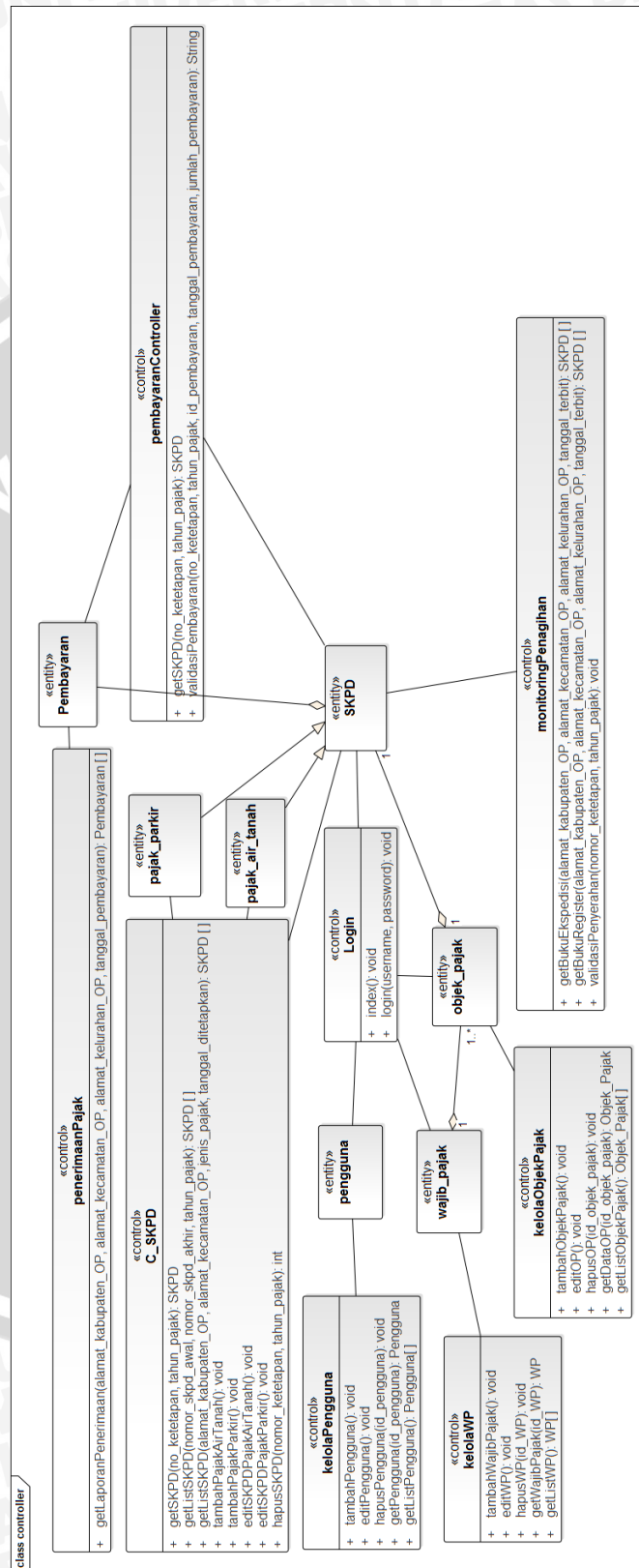
Kelas bertipe *controller* merupakan kelas yang menghubungkan antara model atau *entity* dengan *view*. Biasanya satu *use case* hanya memiliki satu kali kelas *controller*, tetapi tidak menutup kemungkinan *use case* yang kompleks membutuhkan satu atau lebih kelas *controller*. Kelas diagram *controller* Gambar 5.19 terdiri dari kelas penerimaanPajak, monitoringPenagihan, kelolaObjekPajak, kelolaWP, pembayaranController, C_SKPD, dan kelolaPengguna. pembayaranController merupakan kelas *service* yang disediakan oleh Dispenda yang dapat diakses oleh bank.

c. Diagram kelas bertipe *view*

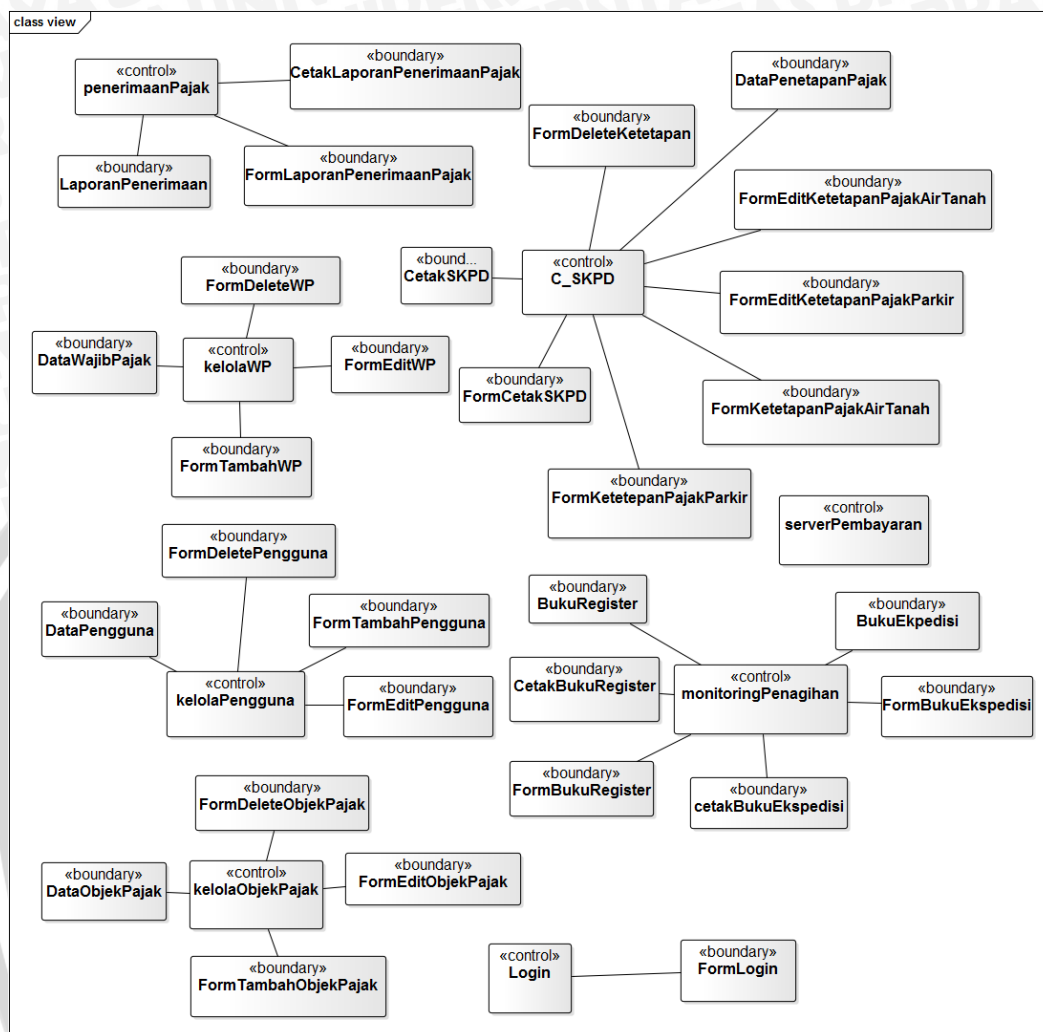
Kelas bertipe *view* merupakan kelas yang terkait antarmuka yang berhubungan dengan sesuatu di luar sistem (misal : pengguna). Kelas *view* ini didapatkan dari mengidentifikasi spesifikasi *use case*. Gambar 5.20 berikut ini adalah perancangan kelas pada sistem informasi penerimaan PAD Kabupaten Madiun serta hubungannya dengan *controller*.



Gambar 5.18 Diagram kelas model



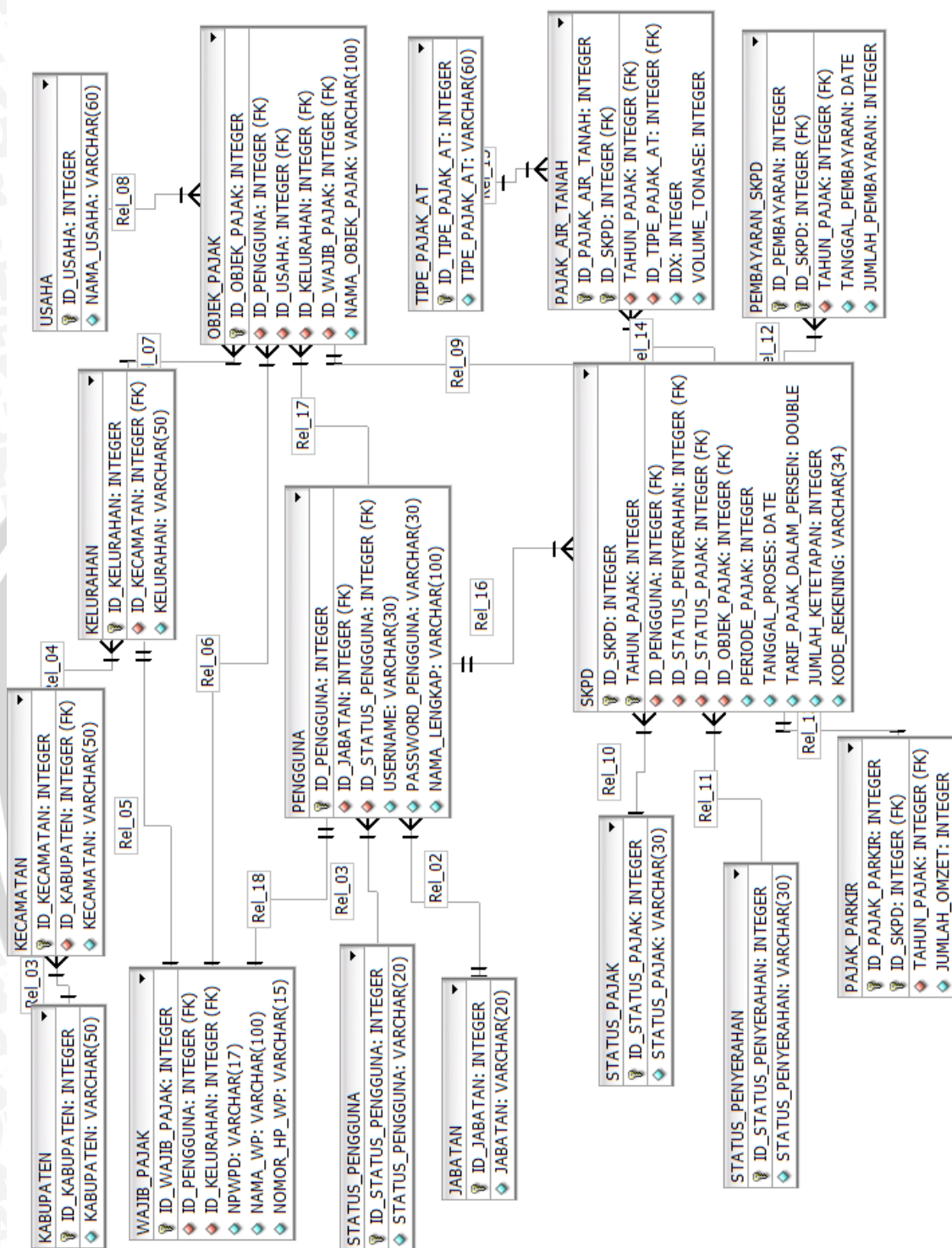
Gambar 5.19 Diagram kelas *controller*



Gambar 5.20 Diagram kelas view

5.8 Pemodelan Data

Pemodelan data merupakan tahap perancangan basis data yang akan digunakan untuk menyimpan data hasil proses dari sistem. Pemodelan data pada penelitian ini direpresentasikan dalam bentuk PDM (*Physical Data Model*). Gambar 5.21 menunjukkan pemodelan data sistem dengan menggunakan PDM.



Gambar 5.21 Pemodelan data

Berikut ini merupakan penjelasan rancangan masing-masing tabel :

1. Nama Tabel : PENGGUNA

No.	Nama field	Type	Lebar	Keterangan
1	ID_PENGGUNA	INTEGER	-	Id_pengguna sebagai <i>primary key</i>
2	ID_JABATAN	INTEGER	-	Jabatan pengguna
3	ID_STATUS_PENGGUNA	INTEGER	-	Status pengguna pada sistem
4	USERNAME	VARCHAR	30	Username milik pengguna
5	PASSWORD_PENGGUNA	VARCHAR	30	Password pengguna
6	NAMA LENGKAP	VARCHAR	100	Nama lengkap pengguna

2. Nama Tabel : STATUS_PENGGUNA

No.	Nama field	Type	Lebar	Keterangan
1	ID_STATUS_PENGGUNA	INTEGER	-	Id dari jenis status pengguna
2	STATUS_PENGGUNA	VARCHAR	20	Jenis status pengguna

3. Nama Tabel : JABATAN

No.	Nama field	Type	Lebar	Keterangan
1	ID_JABATAN	INTEGER	-	Id dari jabatan
2	JABATAN	VARCHAR	20	Jabatan pengguna

4. Nama Tabel : WAJIB_PAJAK

No.	Nama field	Type	Lebar	Keterangan
1	ID_WAJIB_PAJAK	INTEGER	-	Id wajib pajak sebagai <i>primary key</i>
2	ID_PENGGUNA	INTEGER	-	Id pengguna yang melakukan penambahan atau perubahan data wajib pajak
3	ID_KELURAHAN	INTEGER	-	Alamat kelurahan wajib pajak
4	NPWPD	VARCHAR	17	Nomor Pokok Wajib Pajak Daerah
5	NAMA_WP	VARCHAR	100	Nama wajib pajak
6	NOMOR_HP_WP	VARCHAR	15	Nomor HP milik wajib pajak

5. Nama Tabel : OBJEK_PAJAK

No.	Nama field	Type	Lebar	Keterangan
1	ID_OBJEK_PAJAK	INTEGER	-	Id objek pajak
2	ID_PENGGUNA	INTEGER	-	Id pengguna yang menambah atau mengubah data objek pajak
3	ID_USAHA	INTEGER	-	Ide jenis usaha objek pajak
4	ID_KELURAHAN	INTEGER	-	Id alamat kelurahan objek pajak
5	ID_WAJIB PAJAK	INTEGER	-	Id pemilik objek pajak
6	NAMA_OBJEK_PAJAK	VARCHAR	100	Nama dari objek pajak

6. Nama Tabel : KABUPATEN

No.	Nama field	Type	Lebar	Keterangan
1	ID_KABUPATEN	INTEGER	-	Id dari setiap kabupaten
2	KABUPATEN	VARCHAR	50	Nama kabupaten

7. Nama Tabel : KECAMATAN

No.	Nama field	Type	Lebar	Keterangan
1	ID_KECAMATAN	INTEGER	-	Id dari kecamatan
2	ID_KABUPATEN	INTEGER	-	Id kabupaten dari kecamatan
3	KECAMATAN	VARCHAR	50	Nama kecamatan

8. Nama Tabel : KELURAHAN

No.	Nama field	Type	Lebar	Keterangan
1	ID_KELURAHAN	INTEGER	-	Id dari kelurahan
2	ID_KECAMATAN	INTEGER	-	Id kecamatan dari kelurahan
3	KELURAHAN	VARCHAR	50	Nama kelurahan

9. Nama Tabel : USAHA

No.	Nama field	Type	Lebar	Keterangan
1	ID_USAHA	INTEGER	-	Id dari nama usaha
2	NAMA_USAHA	VARCHAR	60	Jenis atau nama usaha

10. Nama Tabel : SKPD

No.	Nama field	Tipe	Lebar	Keterangan
1	ID_SKPD	INTEGER	-	Id dari surat ketetapan pajak daerah
2	TAHUN_PAJAK	INTEGER	-	Tahun periode pajak
3	ID_PENGGUNA	INTEGER	-	Id dari pengguna yang menambah atau mengubah data SKPD
4	ID_STATUS_PENYERAHAN	INTEGER	-	Id dari status penyerahan
5	ID_STATUS_PAJAK	INTEGER	-	Id dari status pajak
6	ID_OBJEK_PAJAK	INTEGER	-	Id objek pajak dari SKPD
7	PERIODE_PAJAK	DATE	-	Periode pajak
8	TANGGAL_PROSES	DATE	-	Tanggal pendataan
9	TARIF_PAJAK_DALAM_PERSEN	DOUBLE	-	Tarif pajak dalam bentuk persen
10	JUMLAH_KETETAPAN	INTEGER	-	Jumlah ketetapan pajak
11	KODE_REKENING	VARCHAR	34	Kode rekening pajak

11. Nama Tabel : PAJAK_PARKIR

No.	Nama field	Tipe	Lebar	Keterangan
1	ID_PAJAK_PARKIR	INTEGER	-	Id dari pajak parkir
2	PERIODE_PAJAK	INTEGER	-	Tahun periode pajak
3	ID_SKPD	INTEGER	-	Id SKPD dari pajak parkir
4	JUMLAH_OMZET	INTEGER	-	Jumlah omzet per bulan

12. Nama Tabel : PAJAK_AIR_TANAH

No.	Nama field	Tipe	Lebar	Keterangan
1	ID_PAJAK_AIR_TANAH	INTEGER	-	Id dari pajak air tanah
2	PERIODE_PAJAK	INTEGER	-	Tahun periode pajak

3	ID_SKPD	INTEGER	-	Id SKPD dari pajak air tanah
4	ID_TPE_PAJAK_AT	INTEGER	-	Id jenis pajak dari pajak air tanah
5	IDX	INTEGER	-	Idx
6	VOLUME_TONASE	INTEGER	-	Volume tonase

13. Nama Tabel : TIPE_PAJAK_AT

No.	Nama field	Tipe	Lebar	Keterangan
1	ID_TPE_PAJAK_AT	INTEGER	-	Id dari tipe pajak air tanah
2	TIPE_PAJAK_AT	VARCHAR	30	Nama tipe pajak air tanah

14. Nama Tabel : STATUS_PAJAK

No.	Nama field	Tipe	Lebar	Keterangan
1	ID_STATUS_PAJAK	INTEGER	-	Id dari status pajak
2	STATUS_PAJAK	VARCHAR	30	Status pajak

15. Nama Tabel : STATUS_PENYERAHAN

No.	Nama field	Tipe	Lebar	Keterangan
1	ID_STATUS_PENYERAHAN	INTEGER	-	Id dari status penyerahan
2	STATUS_PENYERAHAN	VARCHAR	50	Status penyerahan SKPD

16. Nama Tabel : PEMBAYARAN_SKPD

No.	Nama field	Tipe	Lebar	Keterangan
1	ID_PEMBAYARAN	INTEGER	-	Id dari pembayaran
2	ID_SKPD	INTEGER	-	Id SKPD dari pembayaran pajak
3	PERIODE_PAJAK	INTEGER	-	Tahun periode SKPD
4	TANGGAL_PEMBAYARAN	DATE	-	Tanggal pembayaran pajak
5	JUMLAH_PEMBAYARAN	INTEGER	-	Jumlah pembayaran pajak

5.9 Perancangan antarmuka pengguna

Perancangan antarmuka pengguna ini menjelaskan mengenai rancangan antarmuka pengguna sistem dalam berinteraksi dengan penggunanya. Perancangan antarmuka pengguna dalam penelitian ini nantinya akan digunakan sebagai panduan dalam melakukan pembuatan prototipe dari sistem. Berikut ini merupakan perancangan antarmuka pengguna dari sistem :

A. Perancangan antarmuka *Login*

Antarmuka *login* merupakan antarmuka pengguna untuk masuk ke sistem sebelum pengguna dapat menggunakan fitur sesuai hak aksesnya. Gambar 5.22 berikut merupakan antarmuka *login* untuk pengguna sistem.

The diagram illustrates the login interface for the Dinas Pendapatan Kabupaten Madiun. It features a central white rectangular area with a light gray border. At the top, the text "DINAS PENDAPATAN KABUPATEN MADIUN" is displayed in a bold, black, sans-serif font, followed by an arrow pointing to the number "1". Below this, a circular logo containing the text "Logo Kabupaten Madiun" is shown, with an arrow pointing to the number "2". Underneath the logo, there are two input fields: the first is labeled "username" and the second is labeled "password". A large curly bracket to the left of these two fields is labeled with the number "3". At the bottom of the input fields, there is a button labeled "Log in" with an arrow pointing to the number "4".

Gambar 5.22 Perancangan antarmuka *login*

Keterangan perancangan antarmuka *login* :

1. Nama dinas
2. Gambar logo Kabupaten Madiun
3. Formulir untuk melakukan *login*, terdiri dari :
 - a. *Username*
 - b. *Password*
4. Tombol untuk melakukan *login* ke sistem

B. Perancangan antarmuka kelola pengguna

Antarmuka kelola pengguna merupakan antarmuka pengguna untuk pengelolaan pengguna yang dapat menggunakan sistem oleh admin. Selain

The screenshot displays the web application interface for DISPENDA KAB MADIUN. The interface includes a header with the application name and a 'Log out' button. A sidebar on the left contains a logo, the user's name 'Putri Mutiara T Admin', and a 'Kelola Pengguna' (Manage Users) button. The main content area features a 'Data Pengguna Terdaftar' (Registered Users Data) table and a 'Tambah Pengguna' (Add User) button. Numbered annotations (1-8) highlight specific UI elements: 1 points to the application name, 2 to the 'Log out' button, 3 to the user's name, 4 to the 'Kelola Pengguna' button, 5 to the 'Data Pengguna Terdaftar' table header, 6 to the 'Tambah Pengguna' button, 7 to the table body, and 8 to the sidebar area.

DISPENDA KAB MADIUN → 1

2 ← Log out

3 ← Logo Kab Madiun Putri Mutiara T Admin

4 ← Kelola Pengguna

5 ← Data Pengguna Terdaftar

6 ← Tambah Pengguna

7 ←

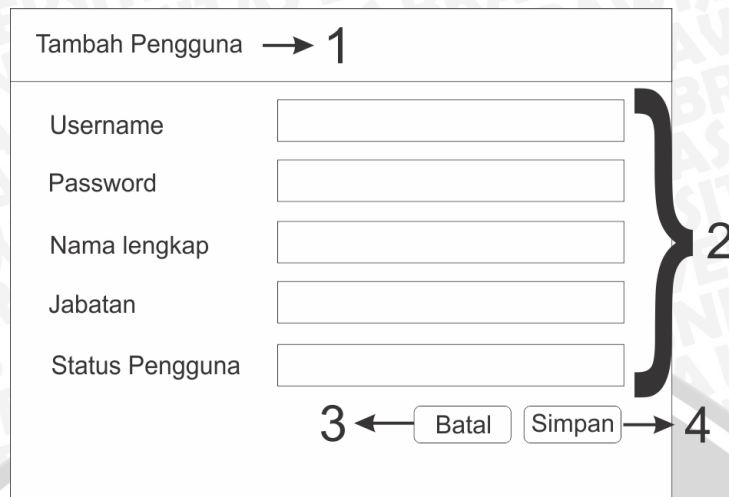
8 ←

No	Username	Nama lengkap	Jabatan	Status user	Ubah/hapus

Keterangan antarmuka kelola pengguna :

1. Nama dinas
2. Tombol untuk melakukan *logout*
3. Gambar logo Kabupaten Madiun
4. Nama pengguna dan status pengguna
5. Judul
6. Tombol untuk menambah data pengguna baru
7. Tabel daftar pengguna yang terdaftar pada sistem
8. Pilihan menu untuk pengguna admin

Antarmuka tambah data pengguna merupakan antarmuka yang menggambarkan rancangan formulir untuk menambahkan data pengguna baru oleh admin. Gambar 5.24 berikut merupakan rancangan antarmuka pengguna tambah data pengguna.



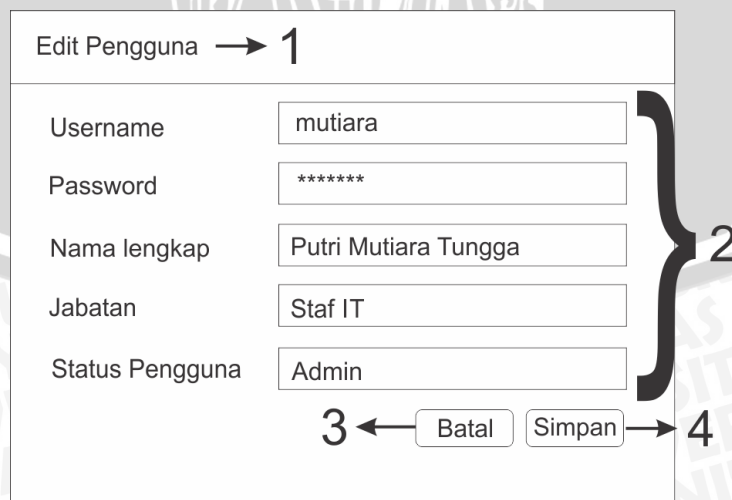
Gambar 5.24 Perancangan antarmuka tambah pengguna

Keterangan antarmuka tambah pengguna :

1. Judul formulir
2. Formulir untuk menambahkan pengguna baru, terdiri dari :
 - a. *Username*
 - b. *Password*
 - c. Nama lengkap
 - d. Jabatan
 - e. Status pengguna
3. Tombol untuk membatalkan proses tambah pengguna baru
4. Tombol untuk menyimpan data pengguna baru

D. Perancangan antarmuka edit pengguna

Antarmuka edit pengguna merupakan antarmuka yang menggambarkan formulir untuk melakukan edit data pengguna oleh admin. Gambar 5.25 berikut merupakan rancangan antarmuka pengguna edit data pengguna.



Gambar 5.25 Perancangan antarmuka edit pengguna

Keterangan antarmuka edit data pengguna :

1. Judul formulir
2. Formulir untuk edit data pengguna, terdiri dari :
 - a. Username
 - b. Password
 - c. Nama lengkap
 - d. Jabatan
 - e. Status pengguna
3. Tombol untuk membatalkan proses edit data pengguna
4. Tombol untuk menyimpan perubahan data pengguna

E. Perancangan antarmuka konfirmasi hapus pengguna

Antarmuka konfirmasi hapus pengguna merupakan antarmuka ketika admin melakukan hapus dan sistem memberikan opsi apakah admin benar-benar ingin menghapus data pengguna. Gambar 5.26 berikut merupakan rancangan antarmuka konfirmasi hapus data pengguna.

Konfirmasi hapus data pengguna → 1

Apakah anda yakin akan menghapus pengguna yang dipilih? → 2

3 ← → 4

Gambar 5.26 Perancangan antarmuka konfirmasi hapus pengguna

Keterangan antarmuka konfirmasi hapus pengguna :

1. Judul
2. Pertanyaan konfirmasi
3. Tombol untuk membatalkan hapus data pengguna
4. Tombol untuk mengkonfirmasi hapus data pengguna

F. Perancangan antarmuka laporan penerimaan pajak

Antarmuka laporan penerimaan pajak merupakan antarmuka pengguna untuk melihat laporan penerimaan pajak berdasarkan filter yang disediakan sistem. Gambar 5.27 berikut merupakan hasil rancangan antarmuka laporan penerimaan pajak.

The wireframe shows a web interface for 'DISPENSA KAB MADIUN'. At the top right is a 'Log out' button (2). On the left is a sidebar with a logo (3) and a menu item 'Laporan Penerimaan' (8). The main area has a header 'Laporan Penerimaan Pajak' (5). Below this are input fields for 'Alamat' (4), 'Jenis Pajak' (6), and 'Tanggal Pembayaran' (6), with a 'Cari' button (7). A table (9) displays the results with columns: No, Jenis Pajak, Objek Pajak, Kecamatan, Kelurahan/Desa, Tanggal Bayar, and Jumlah Bayar. A 'Total' row is at the bottom. A 'Cetak' button (10) is at the bottom right.

No	Jenis Pajak	Objek Pajak	Kecamatan	Kelurahan/Desa	Tanggal Bayar	Jumlah Bayar
Total						

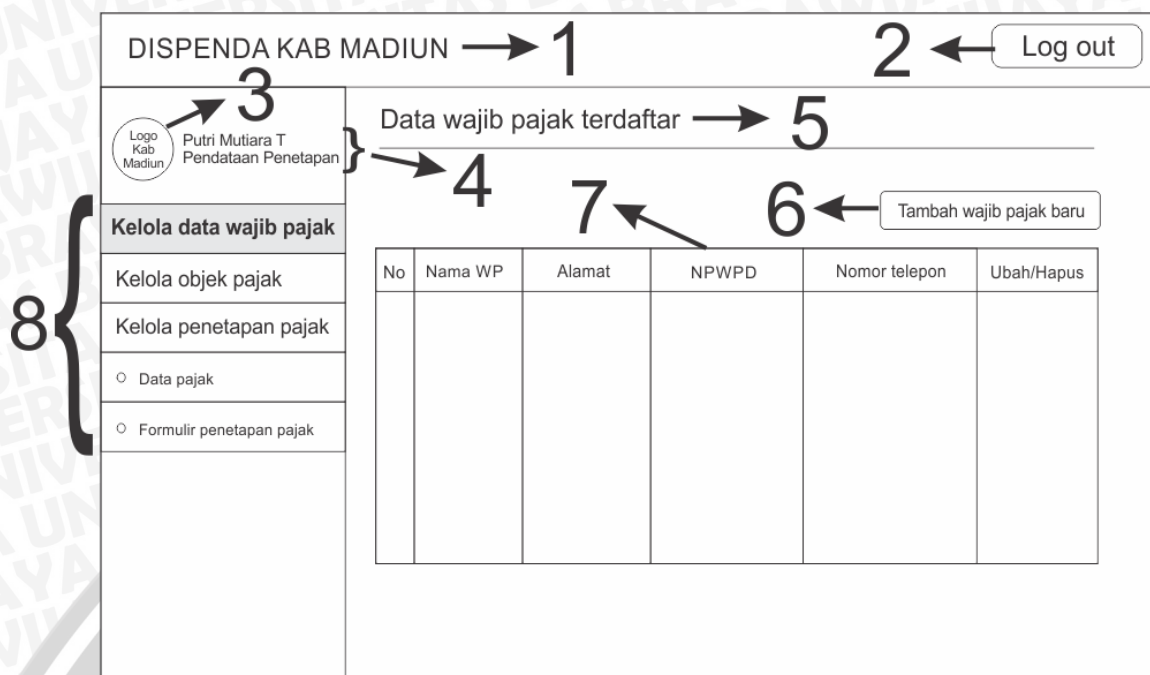
Gambar 5.27 Perancangan antarmuka laporan penerimaan pajak

Keterangan antarmuka laporan penerimaan pajak :

1. Nama dinas
2. Tombol untuk *logout* dari sistem
3. Gambar logo Kabupaten Madiun
4. Nama dan status pengguna
5. Judul
6. Formulir filter lihat laporan penerimaan pajak, terdiri dari : alamat, jenis pajak, dan tanggal pembayaran
7. Tombol untuk melakukan pencarian laporan penerimaan berdasarkan filter
8. Pilihan menu untuk pengguna kas daerah dan bidang pembukuan pelaporan
9. Tabel laporan penerimaan berdasarkan filter
10. Tombol untuk mencetak laporan penerimaan yang ditampilkan

G. Perancangan antarmuka kelola data wajib pajak

Antarmuka kelola wajib pajak merupakan antarmuka pengguna untuk pengelolaan wajib pajak oleh Bidang Pendataan dan Penetapan. Selain itu juga ditampilkan daftar wajib pajak yang telah terdaftar di sistem. Gambar 5.28 berikut merupakan hasil rancangan antarmuka kelola wajib pajak.



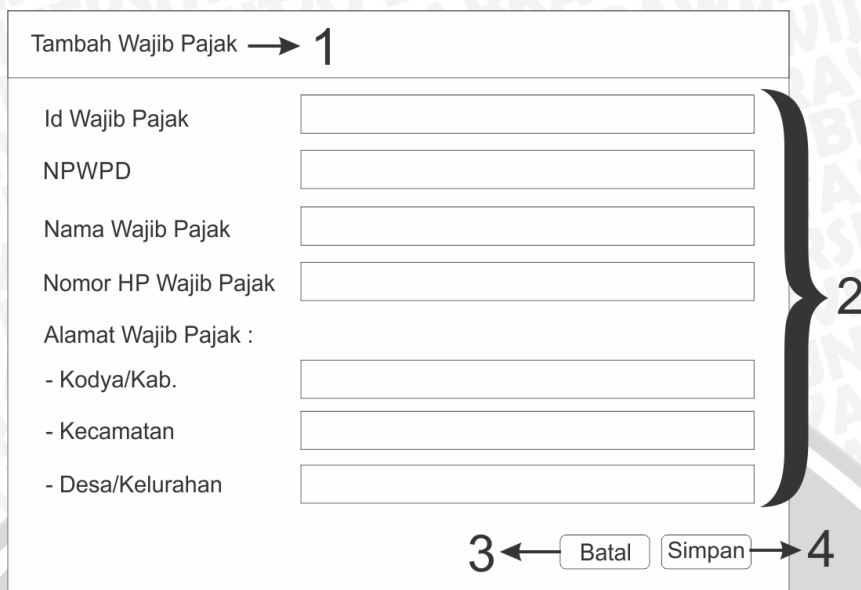
Gambar 5.28 Perancangan antarmuka kelola data wajib pajak

Keterangan antarmuka laporan penerimaan pajak :

1. Nama dinas
2. Tombol untuk *logout* dari sistem
3. Gambar logo Kabupaten Madiun
4. Nama dan status pengguna
5. Judul
6. Tombol untuk menambahkan data wajib pajak baru
7. Tabel data wajib pajak yang terdaftar di sistem
8. Pilihan menu untuk bidang pendataan penetapan. Terdapat menu yang aktif dan tidak aktif. Kelola data wajib pajak merupakan menu yang sedang aktif/digunakan oleh pengguna.

H. Perancangan antarmuka tambah wajib pajak

Antarmuka tambah wajib pajak merupakan antarmuka yang menggambarkan rancangan formulir untuk menambahkan data wajib pajak baru oleh bidang Pendataan dan Penetapan. Gambar 5.29 menunjukkan antarmuka ketika Bidang Pendataan dan Penetapan akan menambahkan wajib pajak baru.



Tambah Wajib Pajak → 1

Id Wajib Pajak

NPWPD

Nama Wajib Pajak

Nomor HP Wajib Pajak

Alamat Wajib Pajak :

- Kodya/Kab.

- Kecamatan

- Desa/Kelurahan

2

3 ← → 4

Gambar 5.29 Perancangan antarmuka tambah wajib pajak

Keterangan antarmuka tambah wajib pajak :

1. Judul formulir
2. Formulir untuk tambah data wajib pajak, terdiri dari :
 - a. Id Wajib Pajak
 - b. NPWPD
 - c. Nama Wajib Pajak
 - d. Nomor HP Wajib Pajak
 - e. Alamat Wajib Pajak terdiri dari : Kodya/Kabupaten, Kecamatan, dan Desa/Kelurahan
3. Tombol untuk membatalkan proses tambah data wajib pajak
4. Tombol untuk menyimpan data wajib pajak baru

1. Perancangan antarmuka edit data wajib pajak

Antarmuka edit data wajib pajak merupakan antarmuka yang menggambarkan formulir untuk melakukan edit data wajib oleh Bidang Pendataan dan Penetapan. Gambar 5.30 berikut merupakan rancangan antarmuka pengguna edit data wajib pajak.

The screenshot shows a web form titled "Edit Data Wajib Pajak" with the following fields and annotations:

- Annotation 1:** Points to the title "Edit Data Wajib Pajak".
- Annotation 2:** A bracket on the right side groups the input fields: "Id Wajib Pajak" (value: 14), "NPWPD" (value: P.2.0078981.12.01), "Nama Wajib Pajak" (value: 085735791947), "Nomor HP Wajib Pajak" (value: Putri Mutiara Tungga), and "Alamat Wajib Pajak" (which includes sub-fields for "Kodya/Kab." (Madiun), "Kecamatan" (Kecamatan 1), and "Kelurahan/Desa" (Kelurahan 3)).
- Annotation 3:** Points to the "Batal" (Cancel) button.
- Annotation 4:** Points to the "Simpan" (Save) button.

Gambar 5.30 Perancangan antarmuka edit data wajib pajak

Keterangan antarmuka edit data wajib pajak :

1. Judul formulir
2. Formulir untuk edit data wajib pajak, terdiri dari :
 - a. Id Wajib Pajak
 - b. NPWPD
 - c. Nama Wajib Pajak
 - d. Nomor HP Wajib Pajak
 - e. Alamat Wajib Pajak terdiri dari : Kodya/Kabupaten, Kecamatan, dan Desa/Kelurahan
3. Tombol untuk membatalkan proses edit data wajib pajak
4. Tombol untuk menyimpan perubahan data wajib pajak

J. Perancangan antarmuka konfirmasi hapus data wajib pajak

Antarmuka konfirmasi hapus data wajib pajak merupakan antarmuka ketika Bidang Pendataan dan Penetapan melakukan hapus data wajib pajak dan sistem memberikan opsi apakah Bidang Pendataan dan Penetapan benar-benar ingin menghapus data wajib pajak. Gambar 5.31 berikut merupakan rancangan antarmuka konfirmasi hapus data wajib pajak.

The screenshot shows a confirmation dialog box titled "Konfirmasi hapus wajib pajak" with the following elements and annotations:

- Annotation 1:** Points to the title "Konfirmasi hapus wajib pajak".
- Annotation 2:** Points to the question text "Apakah anda yakin akan menghapus Wajib Pajak yang dipilih?".
- Annotation 3:** Points to the "Tidak" (No) button.
- Annotation 4:** Points to the "Ya" (Yes) button.

Gambar 5.31 Perancangan antarmuka konfirmasi hapus data wajib pajak

157

L. Perancangan antarmuka tambah objek pajak

Antarmuka tambah objek pajak merupakan antarmuka yang menggambarkan rancangan formulir untuk menambahkan objek pajak baru. Gambar 5.33 menunjukkan antarmuka ketika Bidang Pendataan dan Penetapan akan menambahkan objek pajak baru.

Gambar 5.33 Perancangan antarmuka tambah objek pajak

Keterangan antarmuka tambah objek pajak :

1. Judul formulir
2. Formulir untuk tambah data objek pajak, terdiri dari :
 - a. Id Objek Pajak
 - b. NPWPD
 - c. Nama Objek Pajak
 - d. Alamat Objek Pajak terdiri dari : Kodya/Kabupaten, Kecamatan, dan Desa/Kelurahan
3. Tombol untuk membatalkan proses tambah data objek pajak
4. Tombol untuk menyimpan data objek pajak baru

M. Perancangan antarmuka edit data objek pajak

Antarmuka edit data objek pajak merupakan antarmuka yang menggambarkan formulir untuk melakukan edit data objek oleh Bidang Pendataan dan Penetapan. Gambar 5.34 berikut merupakan rancangan antarmuka pengguna edit data objek pajak.

Gambar 5.34 Perancangan antarmuka edit data objek pajak

Keterangan antarmuka edit objek pajak :

1. Judul formulir
2. Formulir untuk edit data objek pajak, terdiri dari :
 - a. Id Objek Pajak
 - b. NPWPD
 - c. Nama Objek Pajak
 - d. Alamat Objek Pajak terdiri dari : Kodya/Kabupaten, Kecamatan, dan Desa/Kelurahan
3. Tombol untuk membatalkan proses edit data objek pajak
4. Tombol untuk menyimpan perubahan data objek pajak

N. Perancangan antarmuka konfirmasi hapus objek pajak

Antarmuka konfirmasi hapus data objek pajak merupakan antarmuka ketika Bidang Pendataan dan Penetapan melakukan hapus data objek pajak dan sistem memberikan opsi apakah Bidang Pendataan dan Penetapan benar-benar ingin menghapus data objek pajak. Gambar 5.35 berikut merupakan rancangan antarmuka konfirmasi hapus data objek pajak.

Gambar 5.35 Perancangan antarmuka konfirmasi hapus objek pajak

Keterangan antarmuka konfirmasi hapus objek pajak :

1. Judul
2. Pertanyaan konfirmasi
3. Tombol untuk membatalkan hapus data objek pajak
4. Tombol untuk mengkonfirmasi hapus data objek pajak

O. Perancangan antarmuka kelola penetapan pajak – data pajak

Antarmuka kelola penetapan pajak sub menu data pajak merupakan antarmuka pengguna untuk pengelolaan data pajak oleh Bidang Pendataan dan Penetapan. Selain itu juga ditampilkan data pajak yang telah didata pada sistem. Gambar 5.36 berikut merupakan hasil rancangan antarmuka kelola penetapan pajak.

The screenshot shows a web application interface for managing tax data. It includes a header with the title 'DISPENDA KAB MADIUN' and a 'Log out' button. A sidebar on the left contains a menu with options like 'Kelola data wajib pajak', 'Kelola objek pajak', and 'Kelola penetapan pajak'. The main content area has a section for 'Data objek pajak terdaftar' with a 'Cetak SKPD' button. Below this is a 'Lihat Daftar Ketetapan Pajak' section with search filters for 'Alamat', 'Jenis pajak', and 'Tanggal ditetapkan', and a 'Cari' button. At the bottom is a table with 9 columns: 'No SKPD', 'Nama WP', 'Nama objek pajak', 'Alamat objek pajak', 'Jenis pajak', 'Jatuh tempo', 'Jumlah ketetapan', 'Status SKPD', and 'Ubah/hapus'.

Gambar 5.36 Perancangan antarmuka kelola penetapan pajak – data pajak

Keterangan antarmuka laporan penerimaan pajak :

1. Nama dinas
2. Tombol untuk *logout* dari sistem
3. Gambar logo Kabupaten Madiun
4. Nama dan status pengguna
5. Judul
6. Tombol untuk mencetak SKPD
7. Formulir filter data SKPD, terdiri dari : alamat, jenis pajak, dan tanggal ditetapkan
8. Tombol untuk mencari data SKPD sesuai dengan filter
9. Tabel data pajak yang telah ditetapkan sesuai filter
10. Pilihan menu untuk bidang pendataan penetapan. Terdapat menu yang aktif dan tidak aktif. Kelola penetapan pajak - data pajak merupakan menu yang sedang aktif/ digunakan oleh pengguna.

P. Perancangan antarmuka cetak SKPD

Antarmuka cetak SKPD merupakan antarmuka yang menggambarkan rancangan formulir untuk mencetak Surat Ketetapan Pajak Daerah atas pajak yang telah didata dan ditetapkan di sistem oleh Bidang Pendataan dan Penetapan. Gambar 5.37 berikut ini merupakan antarmuka cetak SKPD.

The screenshot shows a window titled 'Cetak SKPD → 1'. It contains two input fields: 'Nomor SKPD' followed by 's/d' and another input field, and 'Periode pajak' with a single input field. These two rows are grouped by a bracket labeled '2'. At the bottom, there are two buttons: 'Batal' (labeled '3' with an arrow pointing left) and 'Cetak' (labeled '4' with an arrow pointing right).

Gambar 5.37 Perancangan antarmuka cetak SKPD

Keterangan antarmuka cetak SKPD :

1. Judul
2. Formulir untuk cetak SKPD, terdiri dari : Nomor SKPD awal, Nomor SKPD akhir, dan periode pajak
3. Tombol untuk membatalkan proses cetak SKPD
4. Tombol untuk memproses cetak SKPD

Q. Perancangan antarmuka penetapan pajak air tanah

The screenshot shows a complex interface for 'DISPENDA KAB MADIUN'. At the top right is a 'Log out' button (labeled '2'). On the left is a sidebar menu (labeled '7') with items: 'Kelola data wajib pajak', 'Kelola objek pajak', 'Kelola penetapan pajak' (selected), 'Data pajak', and 'Formulir penetapan pajak'. The main area is titled 'Penetapan pajak → 5'. It has two tabs: 'Pajak air tanah' (selected) and 'Pajak parkir' (labeled '6'). The 'Pajak air tanah' tab contains several input fields: 'No SKPD', 'Periode Pajak', 'NPWPD', 'Nama WP', 'Nama Objek Pajak', 'Alamat Objek Pajak' (with sub-fields for Kelurahan/Desa, Kecamatan, and Kodya/Kab.), 'Jatuh Tempo', 'Tarif %', 'Kode rekening', 'Idx', 'Harga per m³', 'Volume tonase', 'Tipe Pajak Air Tanah', and 'Jumlah Ketetapan'. A 'Cari' button (labeled '8') is next to the 'NPWPD' field. A large bracket labeled '9' groups the bottom part of the form fields. At the bottom are 'Simpan' (labeled '10' with an arrow pointing left) and 'Batal' (labeled '11' with an arrow pointing right) buttons.

Gambar 5.38 Perancangan antarmuka penetapan pajak air tanah

Antarmuka penetapan pajak air tanah merupakan antarmuka yang menggambarkan rancangan formulir untuk menambahkan data ketetapan pajak air tanah oleh bidang Pendataan dan Penetapan. Gambar 5.39 menunjukkan antarmuka ketika Bidang Pendataan dan Penetapan akan menambahkan data ketetapan pajak air tanah.

Keterangan antarmuka penetapan pajak air tanah :

1. Nama dinas
2. Tombol untuk *logout* dari sistem
3. Gambar logo Kabupaten Madiun
4. Nama dan status pengguna
5. Judul
6. Tab untuk formulir pajak air tanah dan pajak parkir
7. Pilihan menu untuk bidang pendataan penetapan. Terdapat menu yang aktif dan tidak aktif. Kelola penetapan pajak – formulir penetapan pajak merupakan menu yang sedang aktif/ digunakan oleh pengguna.
8. Tombol untuk mencari data wajib pajak berdasarkan NPWPD
9. Formulir penetapan pajak air tanah, terdiri dari :
 - a. Nomor SKPD (nomor ketetapan)
 - b. Periode Pajak
 - c. NPWPD (Nomor Pokok Wajib Pajak Daerah)
 - d. Nama Wajib Pajak
 - e. Nama Objek Pajak
 - f. Alamat Objek Pajak yang terdiri dari kelurahan atau desa, kecamatan, dan kabupaten
 - g. Tanggal jatuh tempo
 - h. Tarif pajak dalam % (persen)
 - i. Kode Rekening
 - j. Idx
 - k. Harga per m³
 - l. Volume tonase
 - m. Tipe Pajak air tanah
 - n. Jumlah Ketetapan
10. Tombol untuk menyimpan penetapan pajak air tanah
11. Tombol untuk membatalkan proses penetapan pajak air tanah

R. Perancangan antarmuka penetapan pajak parkir

Antarmuka penetapan pajak parkir merupakan antarmuka yang menggambarkan rancangan formulir untuk menambahkan data ketetapan pajak parkir oleh bidang Pendataan dan Penetapan. Gambar 5.39 menunjukkan antarmuka ketika Bidang Pendataan dan Penetapan akan menambahkan data ketetapan pajak parkir.

Gambar 5.39 Perancangan antarmuka penetapan pajak parkir

Keterangan antarmuka penetapan pajak parkir :

1. Nama dinas
2. Tombol untuk *logout* dari sistem
3. Gambar logo Kabupaten Madiun
4. Nama dan status pengguna
5. Judul
6. Tab untuk formulir pajak air tanah dan pajak parkir
7. Pilihan menu untuk bidang pendataan penetapan. Terdapat menu yang aktif dan tidak aktif. Kelola penetapan pajak – formulir penetapan pajak merupakan menu yang sedang aktif/ digunakan oleh pengguna.
8. Tombol untuk mencari data wajib pajak berdasarkan NPWPD
9. Formulir penetapan pajak parkir, terdiri dari :
 - a. Nomor SKPD (nomor ketetapan)
 - b. Periode Pajak
 - c. NPWPD (Nomor Pokok Wajib Pajak Daerah)
 - d. Nama Wajib Pajak
 - e. Nama Objek Pajak
 - f. Alamat Objek Pajak yang terdiri dari kelurahan atau desa, kecamatan, dan kabupaten
 - g. Tanggal jatuh tempo
 - h. Tarif pajak dalam % (persen)
 - i. Kode Rekening
 - j. Jumlah Omset
 - k. Jumlah Ketetapan
10. Tombol untuk menyimpan penetapan pajak parkir

11. Tombol untuk membatalkan proses penetapan pajak parkir

S. Perancangan antarmuka buku ekspedisi

Antarmuka buku ekspedisi merupakan antarmuka pengguna untuk melihat buku ekspedisi berdasarkan filter yang disediakan sistem oleh Bidang Penagihan dan Keberatan. Buku ekspedisi ini berisi data pajak yang terdaftar beserta status penyerahan SKPD ke wajib pajak. Gambar 5.40 berikut merupakan hasil rancangan antarmuka buku ekspedisi.

Gambar 5.40 Perancangan antarmuka buku ekspedisi

Keterangan antarmuka penetapan pajak parkir :

1. Nama dinas
2. Tombol untuk logout dari sistem
3. Gambar logo Kabupaten Madiun
4. Nama dan status pengguna
5. Judul
6. Formulir untuk filter buku ekspedisi, terdiri dari : alamat, jenis pajak, jatuh tempo, dan tanggal terbit
7. Tombol untuk melakukan pencarian buku ekspedisi berdasarkan formulir filter
8. Pilihan menu untuk bidang penagihan keberatan. Terdapat menu yang aktif dan tidak aktif. Buku ekspedisi merupakan menu yang sedang aktif/ digunakan oleh pengguna.
9. Tabel buku ekspedisi berdasarkan formulir filter
10. Tombol untuk melakukan cetak buku ekspedisi yang tampil (sesuai tabel nomor 9)
11. Tombol untuk melakukan validasi penyerahan SKPD

T. Perancangan antarmuka buku register

Antarmuka buku register merupakan antarmuka pengguna untuk melihat buku register berdasarkan filter yang disediakan sistem oleh Bidang Penagihan dan Keberatan. Buku register ini berisi data pajak yang terdaftar beserta status pembayaran pajak yang dilakukan oleh wajib pajak. Gambar 5.41 berikut merupakan hasil rancangan antarmuka buku ekspedisi.

No SKPD	Nama WP	Objek Pajak	Kecamatan objek pajak	Kelurahan/Desa Objek Pajak	Jatuh tempo	Validasi pembayaran
						Validasi

Gambar 5.41 Perancangan antarmuka buku register

Keterangan antarmuka penetapan pajak parkir :

1. Nama dinas
2. Tombol untuk *logout* dari sistem
3. Gambar logo Kabupaten Madiun
4. Nama dan status pengguna
5. Judul
6. Formulir untuk filter buku register, terdiri dari : alamat, jenis pajak, jatuh tempo, tanggal terbit, dan status pembayaran
7. Tombol untuk melakukan pencarian buku ekspedisi berdasarkan formulir filter
8. Pilihan menu untuk bidang penagihan keberatan. Terdapat menu yang aktif dan tidak aktif. Buku ekspedisi merupakan menu yang sedang aktif/ digunakan oleh pengguna.
9. Tabel buku ekspedisi berdasarkan formulir filter
10. Tombol untuk melakukan cetak buku ekspedisi yang tampil (sesuai tabel nomor 9)
11. Tombol untuk melakukan validasi penyerahan SKPD

5.10 Prototipe

Setelah perancangan antarmuka selesai dilakukan, kemudian dilakukan pembuatan prototipe dari sistem. Prototipe ini nantinya akan ditunjukkan kepada pengguna untuk dilakukan evaluasi sehingga diperoleh masukan dan saran. Berikut ini adalah hasil dari pembuatan prototipe sistem :

A. Prototipe halaman *login*

Gambar 5.42 Halaman *login*

Gambar 5.42 merupakan hasil pembuatan prototipe halaman *login* yang digunakan oleh pengguna untuk *login*/masuk ke sistem sebelum dapat menggunakan fitur-fitur sesuai hak aksesnya.

B. Prototipe halaman kelola pengguna

No	Username	Nama Lengkap	Jabatan	Status User	Ubah/Hapus
1	mutiara	Putri Mutiara Tungga	IT	Admin	Edit Hapus
2	pendataan_penetapan	Mustofa	Kepala Bidang Pendataan dan Penetapan	Pendataan Penetapan	Edit Hapus
3	penagihan_keberatan	Ari Nursurahmat	Kepala Bidang Penagihan dan Keberatan	Penagihan Keberatan	Edit Hapus
4	pembukuan_pelaporan	Suprihardi	Kepala Bidang Pembukuan dan Pelaporan	Pembukuan Pelaporan	Edit Hapus
5	kas_daerah	Nama	Pegawai Kas Daerah	Kas Daerah	Edit Hapus

Gambar 5.43 Halaman Kelola Pengguna

Gambar 5.43 merupakan tampilan prototipe dari halaman kelola pengguna yang digunakan oleh admin untuk melakukan pengelolaan data-data pengguna sistem.

C. Prototipe halaman laporan penerimaan pajak

DISPENDA KAB MADIUN Putri Mutiara Tungga

Laporan Penerimaan Pajak

Alamat: Pilih Kecamatan Pilih Kelurahan atau Desa

Jenis Pajak: Pilih Jenis Pajak

Tanggal pembayaran:

Cari

Semua Pajak **Cetak**

No	Jenis Pajak	Objek Pajak	Kecamatan	Kelurahan/Desa	Tanggal bayar	Jumlah Bayar
1.	Pajak Parkir	Objek Pajak 1	Kecamatan 1	Kelurahan 1	14 April 2016	100000
2.	Pajak Parkir	Objek Pajak 2	Kecamatan 4	Kelurahan 3	14 April 2016	100000
3.	Pajak Air Tanah	Objek Pajak 3	Kecamatan 2	Kelurahan 3	14 April 2016	100000
4.	Pajak Air Tanah	Objek Pajak 2	Kecamatan 1	Kelurahan 1	14 April 2016	100000
Total						10000000

Gambar 5.44 Halaman Laporan Penerimaan Pajak

Gambar 5.44 merupakan tampilan prototipe pada halaman laporan penerimaan pajak yang digunakan oleh Kas Daerah, Bidang Pembukuan Pelaporan, serta Bidang Penagihan Keberatan untuk melihat laporan penerimaan pajak berdasarkan alamat dan/atau jenis pajak dan/atau tanggal pembayaran pajak.

D. Prototipe halaman kelola data wajib pajak

DISPENDA KAB MADIUN Putri Mutiara Tungga

Data Wajib Pajak Terdaftar **Tambah Wajib Pajak Baru**

Show 10 entries Search:

No	Nama	Alamat	NPWPD	Nomor Telepon	Ubah/Hapus
1	mutiara	Kecamatan 2, Kelurahan 1	P.2.0078981.12.01	08723462347	Edit Hapus
2	putri	Kecamatan 2, Kelurahan 4	P.2.0078981.12.02	08488219321	Edit Hapus
3	randi	Kecamatan 4, Kelurahan 6	P.2.0078981.12.03	087578834234	Edit Hapus
4	rayyan	Kecamatan 3, Kelurahan 7	P.2.0078981.12.04	08579332423	Edit Hapus
5	rizal	Kecamatan 5, Kelurahan 1	P.2.0078981.12.05	0981232138123	Edit Hapus
6	hasanah	Kecamatan 3, Kelurahan 4	P.2.0078981.12.06	08566123812	Edit Hapus
7	krisdhamara	Kecamatan 4, Kelurahan 4	P.2.0078981.12.07	08767123123	Edit Hapus

Gambar 5.45 Halaman Kelola Data Wajib Pajak

Gambar 5.45 merupakan tampilan prototipe pada halaman kelola data wajib pajak yang digunakan oleh bidang Pendataan dan Penetapan untuk mengelola data-data wajib pajak yang ada di Kabupaten Madiun.

E. Prototipe halaman tambah data wajib pajak

DISPENDA KAB MADILIN

Putri Mutiara Tungga
Pendataan & Penetapan

MAIN NAVIGATION

- Kelola Data Wajib Pajak
- Kelola Objek Pajak
- Kelola Penetapan Pajak

Tambah Data Wajib Pajak

Id Wajib Pajak

NPWP

Nama Wajib Pajak

Nomor HP Wajib Pajak

ALAMAT WAJIB PAJAK :

Kodya/Kabupaten

Kecamatan

Kelurahan

Batal Simpan

No	Nama	Alamat	NPWP	Nomor Telepon	Ubah/Hapus
5	rizal	Kecamatan 5, Kelurahan 1	P.2.0078981.12.05	0981232138123	Ubah/Hapus
6	hasanah	Kecamatan 3, Kelurahan 4	P.2.0078981.12.06	08566123812	Ubah/Hapus
7	krisdhamara	Kecamatan 4, Kelurahan 4	P.2.0078981.12.07	08767123123	Ubah/Hapus

Gambar 5.46 Halaman Tambah Data Wajib Pajak

Gambar 5.46 adalah tampilan hasil pembuatan prototipe pada halaman tambah wajib pajak yang merupakan formulir yang digunakan oleh Bidang Pendataan dan Penetapan untuk mendata wajib pajak baru.

F. Prototipe halaman edit data wajib pajak

DISPENDA KAB MADILIN

Putri Mutiara Tungga
Pendataan & Penetapan

MAIN NAVIGATION

- Kelola Data Wajib Pajak
- Kelola Objek Pajak
- Kelola Penetapan Pajak

Edit Data Wajib Pajak

Id Wajib Pajak

NPWP

Nama Wajib Pajak

Nomor HP Wajib Pajak

ALAMAT WAJIB PAJAK :

Kodya/Kabupaten

Kecamatan

Kelurahan

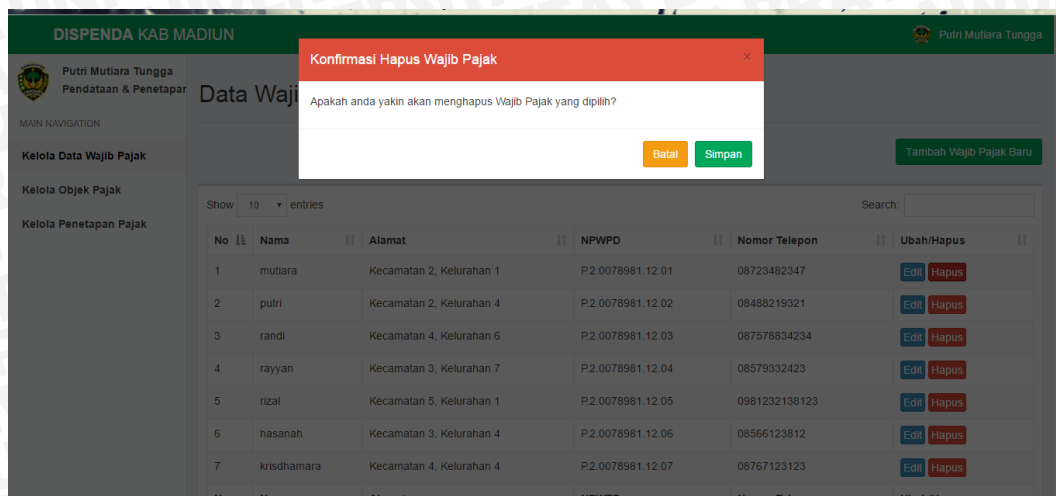
Batal Simpan

No	Nama	Alamat	NPWP	Nomor Telepon	Ubah/Hapus
5	rizal	Kecamatan 5, Kelurahan 1	P.2.0078981.12.05	0981232138123	Ubah/Hapus
6	hasanah	Kecamatan 3, Kelurahan 4	P.2.0078981.12.06	08566123812	Ubah/Hapus
7	krisdhamara	Kecamatan 4, Kelurahan 4	P.2.0078981.12.07	08767123123	Ubah/Hapus

Gambar 5.47 Halaman Edit Data Wajib Pajak

Gambar 5.47 menunjukkan tampilan prototipe pada halaman edit data wajib pajak yang berupa formulir yang digunakan oleh Bidang Pendataan dan Penetapan untuk melakukan perubahan data wajib pajak yang telah terdaftar di sistem.

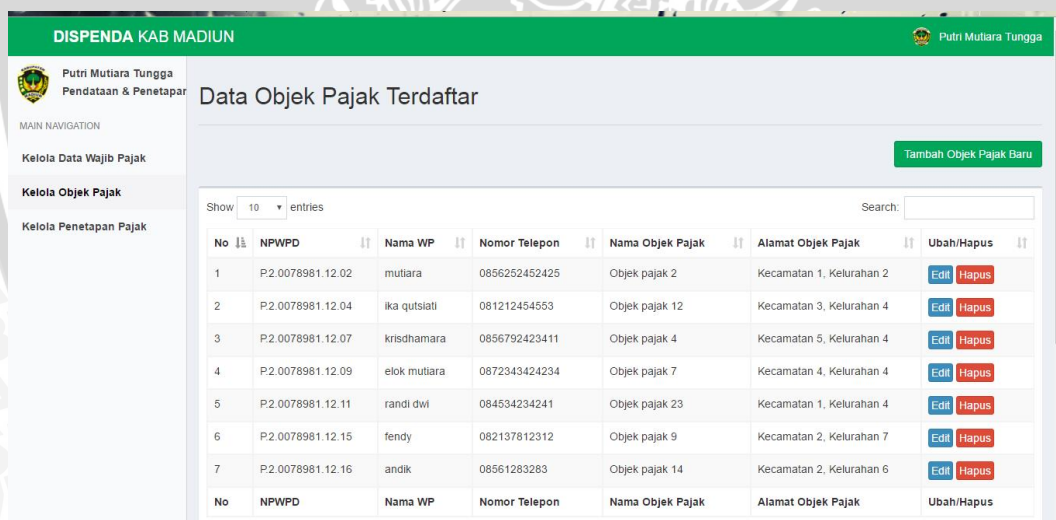
G. Prototipe halaman konfirmasi hapus wajib pajak



Gambar 5.48 Halaman Konfirmasi Hapus Wajib Pajak

Gambar 5.48 menunjukkan tampilan prototipe halaman konfirmasi hapus wajib pajak yang merupakan formulir persetujuan penghapusan data wajib pajak yang dipilih oleh Bidang Pendataan dan Penetapan.

H. Prototipe halaman kelola objek pajak



Gambar 5.49 Halaman Kelola Objek Pajak

Gambar 5.49 menunjukkan tampilan prototipe halaman kelola objek pajak yang merupakan tampilan ketika Bidang Pendataan dan Penetapan mengelola data-data objek pajak pada sistem.

I. Prototipe halaman kelola penetapan pajak – data SKPD

No SKPD	Nama WP	Nama Objek Pajak	Alamat Objek Pajak	Jenis Pajak	Jatuh Tempo	Jumlah Ketetapan	Status SKPD	Ubah/Hapus
1	Putri Mutiara Tungga	Objek Pajak 1	Kecamatan 1, Kelurahan 3	Parkir	Telah ditetapkan	90000	Telah ditetapkan	Edit Hapus
2	Elok Mutiara Niwana	Objek Pajak 11	Kecamatan 5, Kelurahan 1	Air Tanah	20-04-2016	50000	Telah diterbitkan	Edit Hapus

Gambar 5.50 Halaman Kelola Penetapan Pajak-Data SKPD

Gambar 5.50 menunjukkan tampilan prototipe pada halaman kelola penetapan pajak pada sub menu data SKPD yang merupakan halaman yang digunakan oleh Bidang Pendataan dan Penetapan untuk melihat data pajak yang telah ditetapkan sesuai filter yang disediakan sistem. Halaman ini juga menyediakan pilihan cetak SKPD yang digunakan oleh Bidang Pendataan dan Penetapan ketika akan mencetak SKPD.

J. Prototipe halaman formulir cetak SKPD

Gambar 5.51 Halaman Formulir Cetak SKPD

Gambar 5.51 menunjukkan halaman formulir cetak SKPD yang digunakan oleh Bidang Pendataan dan Penetapan untuk mencetak SKPD sesuai masukan pada formulir yang disediakan sistem.

K. Prototipe halaman kelola penetapan pajak – formulir penetapan pajak air tanah

Gambar 5.52 Halaman Kelola Penetapan Pajak – Formulir Penetapan Pajak Air Tanah

Gambar 5.52 menunjukkan tampilan prototipe halaman kelola penetapan pajak pada sub menu formulir penetapan pajak pada jenis pajak air tanah. Halaman berisi formulir yang digunakan oleh Bidang Pendataan dan Penetapan untuk melakukan penetapan pajak atas objek pajak air tanah yang telah terdata di sistem.

L. Prototipe halaman buku ekspedisi

No SKPD	Nama WP	Objek Pajak	Kecamatan objek pajak	Kelurahan/Desa objek pajak	Jatuh Tempo	Validasi Penyerahan
1	mutiara	Objek pajak 1	Kecamatan 1	Kelurahan 3	20-04-2016	Validasi
2	putri	Objek pajak 2	Kecamatan 2	Kelurahan 5	14-05-2016	Batalkan Validasi
3	septiana	Objek pajak 3	Kecamatan 3	Kelurahan 7	05-03-2016	Validasi
4	nurul	Objek pajak 4	Kecamatan 4	Kelurahan 2	12-03-2016	Validasi

Gambar 5.53 Halaman Buku Ekspedisi

Gambar 5.53 menunjukkan tampilan prototipe pada halaman buku ekspedisi yang digunakan oleh Bidang Penagihan dan Keberatan ketika ingin melihat buku ekspedisi atau akan mencetak buku ekspedisi.

M. Prototipe halaman buku register

DISPENDA KAB MADIUN

Putri Mutiara Tungga
Penagihan & Keberatan

Putri Mutiara Tungga

Buku Register

MAIN NAVIGATION

Buku Ekspedisi

Buku Register

Laporan Penerimaan

Alamat

Pilih Kecamatan

Pilih Kelurahan atau Desa

Jenis Pajak

Pilih Jenis Pajak

Jatuh tempo

Tanggal terbit

Status Pembayaran

Semua

Cari

Pajak Air Tanah

Cetak

No SKPD	Nama WP	Nomor HP WP	Objek Pajak	Alamat Objek Pajak	Jenis Pajak	Jatuh Tempo	Status Penyerahan SKPD	Status Pembayaran
1	mutiara	0864239284	Objek pajak 2	Kecamatan 2, Kelurahan 3	Pajak Air tanah	20-04-2016	Sudah serahkan	Sudah dibayar
2	putri	028323467732	Objek pajak	Kecamatan 1, Kelurahan	Pajak Air	20-04-2016	Belum Diserahkan	Belum Dibayar

Gambar 5.54 Halaman Buku Register

Gambar 5.54 menunjukkan tampilan prototipe pada halaman buku register yang digunakan oleh Bidang Penagihan dan Keberatan ketika ingin melihat buku register atau akan mencetak buku register.

BAB 6 EVALUASI

Bab ini membahas tentang evaluasi dari analisis dan perancangan sistem yang telah dibuat. Evaluasi yang dilakukan berupa validasi dan verifikasi. Validasi berfungsi untuk memastikan perancangan yang dihasilkan sesuai dengan spesifikasi yang telah didefinisikan. Verifikasi berfungsi untuk memastikan apakah spesifikasi perangkat lunak yang diimplementasikan dengan prototipe sudah sesuai dengan harapan pengguna.

6.1 Tinjauan

Tinjauan pada penelitian ini dilakukan dengan menguji pengguna untuk melakukan lima skenario *use case* terhadap prototipe yang telah dibuat, tinjauan navigasi dan tinjauan antarmuka pengguna. Tinjauan skenario yang diujikan dalam penelitian ini adalah skenario berhasil tambah pengguna, skenario data pengguna tidak diisi dengan lengkap, skenario edit data wajib pajak, skenario berhasil melakukan cetak SKPD dan skenario berhasil melakukan lihat buku ekspedisi. Pengguna akan melakukan tinjauan mengenai alur, jenis-jenis data, kemudahan pengoperasian dan penanganan kesalahan atas skenario yang telah ditentukan tersebut. Setelah itu pengguna akan melakukan tinjauan navigasi dan antarmuka prototipe secara keseluruhan.

6.1.1 Tinjauan skenario *use case*

Pada tinjauan skenario *use case* ini pengguna akan diminta menjalankan skenario yang diminta kemudian peneliti melakukan tinjauan terhadap skenario tersebut. Tinjauan skenario *use case* pada penelitian ini dilakukan oleh dua pengguna dari Dinas Pendapatan Daerah Kabupaten Madiun yaitu oleh Ari Nursurahmat, S. Sos (Kepala bidang penagihan dan keberatan) dan Budi Muratno, S. Sos (Kepala bidang perencanaan). Tujuan dari tinjauan *use case* ini bertujuan untuk mengetahui apakah spesifikasi perangkat lunak sudah sesuai dengan perancangan prototipe yang telah dibuat. Berikut ini adalah hasil pengujian dan tinjauan lima skenario *use case* oleh dua orang pengguna.

1. Pengguna : Ari Nursurahmat, S. Sos

Tabel 6.1 Hasil pengujian prototipe skenario berhasil tambah pengguna oleh Ari Nursurahmat, S. Sos

<i>Use case</i>	Kelola pengguna	
Skenario	Berhasil tambah pengguna	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu kelola pengguna 2. Memilih tombol tambah pengguna baru	1. Memilih menu kelola pengguna 2. Memilih tombol tambah pengguna baru	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.

3. Mengisi formulir pendaftaran pengguna	3. Mengisi formulir pendaftaran pengguna	
4. Memilih tombol simpan	4. Memilih tombol simpan	

Tabel 6.2 Hasil pengujian prototipe skenario data pengguna tidak diisi dengan lengkap oleh Ari Nursurahmat, S. Sos

<i>Use case</i>	Kelola pengguna	
Skenario	Data pengguna tidak diisi dengan lengkap	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu kelola pengguna	1. Memilih menu kelola pengguna	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.
2. Memilih tombol tambah pengguna baru	2. Memilih tombol tambah pengguna baru	
3. Mengisi formulir pendaftaran pengguna dengan tidak lengkap	3. Mengisi formulir pendaftaran pengguna dengan tidak lengkap	
4. Memilih tombol simpan	4. Memilih tombol simpan	

Tabel 6.3 Hasil pengujian prototipe skenario berhasil melakukan edit data wajib pajak oleh Ari Nursurahmat, S. Sos

<i>Use case</i>	Kelola wajib pajak	
Skenario	Berhasil melakukan edit data wajib pajak	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu kelola data wajib pajak	1. Memilih menu kelola data wajib pajak	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.
2. Memilih tombol edit	2. Memilih tombol edit	
3. Mengedit data wajib pajak	3. Mengedit data wajib pajak	
4. Memilih tombol simpan	4. Memilih tombol simpan	

**Tabel 6.4 Hasil pengujian prototipe skenario berhasil melakukan cetak SKPD
oleh Ari Nursurahmat, S. Sos**

<i>Use case</i>	Cetak SKPD	
Skenario	Berhasil melakukan cetak SKPD	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu kelola penetapan pajak 2. Memilih menu data SKPD 3. Memilih tombol cetak SKPD 4. Memasukkan nomor SKPD awal dan akhir, tahun SKPD 5. Memilih tombol cetak	1. Memilih menu kelola penetapan pajak 2. Memilih menu data SKPD 3. Memilih tombol cetak SKPD 4. Memasukkan nomor SKPD awal dan akhir, tahun SKPD 5. Memilih tombol cetak	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.

Tabel 6.5 Hasil pengujian prototipe skenario berhasil melakukan lihat buku ekspedisi oleh Ari Nursurahmat, S. Sos

<i>Use case</i>	Lihat buku ekspedisi	
Skenario	Berhasil melakukan lihat buku ekspedisi	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu buku ekspedisi 2. Mengisi formulir filter 3. Memilih tombol cari	1. Memilih menu buku ekspedisi 2. Mengisi formulir filter 3. Memilih tombol cari	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.

2. Pengguna : Budi Muratno, S. Sos

**Tabel 6.6 Hasil pengujian prototipe skenario berhasil tambah pengguna oleh
Budi Muratno, S. Sos**

<i>Use case</i>	Kelola pengguna	
Skenario	Berhasil tambah pengguna	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu kelola pengguna 2. Memilih tombol tambah pengguna baru	1. Memilih menu kelola pengguna 2. Memilih tombol tambah pengguna baru	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.

3. Mengisi formulir pendaftaran pengguna	3. Mengisi formulir pendaftaran pengguna	
4. Memilih tombol simpan	4. Memilih tombol simpan	

Tabel 6.7 Hasil pengujian prototipe skenario data pengguna tidak diisi dengan lengkap oleh Budi Muratno, S. Sos

<i>Use case</i>	Kelola pengguna	
Skenario	Data pengguna tidak diisi dengan lengkap	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu kelola pengguna 2. Memilih tombol tambah pengguna baru 3. Mengisi formulir pendaftaran pengguna dengan tidak lengkap 4. Memilih tombol simpan	1. Memilih menu kelola pengguna 2. Memilih tombol tambah pengguna baru 3. Mengisi formulir pendaftaran pengguna dengan tidak lengkap 4. Memilih tombol simpan	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.

Tabel 6.8 Hasil pengujian prototipe skenario berhasil melakukan edit data wajib pajak Budi Muratno, S. Sos

<i>Use case</i>	Kelola wajib pajak	
Skenario	Berhasil melakukan edit data wajib pajak	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu kelola data wajib pajak 2. Memilih tombol edit 3. Mengedit data wajib pajak 4. Memilih tombol simpan	1. Memilih menu kelola data wajib pajak 2. Memilih tombol edit 3. Mengedit data wajib pajak 4. Memilih tombol simpan	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.

**Tabel 6.9 Hasil pengujian prototipe skenario berhasil melakukan cetak SKPD
Budi Muratno, S. Sos**

<i>Use case</i>	Cetak SKPD	
Skenario	Berhasil melakukan cetak SKPD	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu kelola penetapan pajak	1. Memilih menu kelola penetapan pajak	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.
2. Memilih menu data SKPD	2. Memilih menu data SKPD	
3. Memilih tombol cetak SKPD	3. Memilih tombol cetak SKPD	
4. Memasukkan nomor SKPD awal dan akhir, tahun SKPD	4. Memasukkan nomor SKPD awal dan akhir, tahun SKPD	
5. Memilih tombol cetak	5. Memilih tombol cetak	

Tabel 6.10 Hasil pengujian prototipe skenario berhasil melakukan lihat buku ekspedisi oleh Budi Muratno, S. Sos

<i>Use case</i>	Lihat buku ekspedisi	
Skenario	Berhasil melakukan lihat buku ekspedisi	
Alur pengguna	Alur seharusnya	Ketersesuaian
1. Memilih menu buku ekspedisi	1. Memilih menu buku ekspedisi	Alur yang dilakukan pengguna sudah sesuai dengan alur seharusnya.
2. Mengisi formulir filter	2. Mengisi formulir filter	
3. Memilih tombol cari	3. Memilih tombol cari	

Berdasarkan review atas 5 use case dan 2 pengguna dapat disimpulkan bahwa alur yang dilakukan oleh kedua pengguna sesuai dengan alur yang seharusnya atau alur yang dispesifikasikan.

6.1.2 Tinjauan Prototipe

Tinjauan prototipe pada penelitian ini adalah tinjauan yang bertujuan untuk mengetahui apakah spesifikasi perangkat lunak yang diimplementasikan dengan menggunakan prototipe sudah sesuai dengan harapan pengguna, mendapatkan masukan dari pengguna atas spesifikasi perangkat lunak dan perancangan perangkat lunak, serta mengetahui kualitas navigasi dan antarmuka pengguna. Tinjauan prototipe dilakukan dengan pengguna menjalankan lima skenario *use case* pada prototipe kemudian menilai kesesuaian prototipe dengan harapan mereka dan memberikan masukan.

Dari tinjauan prototipe atas lima skenario terhadap dua pengguna diperoleh hasil sebagai berikut :

1. Dari 2 pengguna yang melakukan tinjauan terhadap 5 skenario *use case* semua menyatakan bahwa alur 5 skenario sudah sesuai dengan yang diharapkan, beberapa jenis-jenis data pada formulir pada 5 skenario *use case* perlu ditambahkan dan disempurnakan serta proses 5 skenario *use case* mudah untuk dijalankan.
2. Masukan dari pengguna :
 - a. Pada formulir tambah pengguna kolom status pengguna seharusnya diberikan pilihan status pengguna.
 - b. Pada formulir edit data wajib pajak perlu ditambah kolom alamat lengkap.
 - c. Pada formulir cetak SKPD perlu ditambah periode pajak berupa tanggal dan bulan.
 - d. Peletakan menu data SKPD seharusnya diletakkan setelah menu formulir penetapan pajak.

Hasil tinjauan prototipe oleh pengguna telah dilampirkan pada Lampiran B Gambar B.1, Gambar B.2, Gambar B.5, dan Gambar B.6.

6.1.2.1 Tinjauan Navigasi

Tinjauan navigasi pada penelitian ini merupakan evaluasi subjektif melalui prototipe untuk mendapatkan masukan mengenai navigasi sistem serta mengetahui kualitas navigasi dari sistem. Hasil tinjauan navigasi kedua pengguna menyetujui setiap butir-butir pertanyaan mengenai navigasi sistem. Hasil tinjauan navigasi ini telah dilampirkan pada lampiran B Gambar B.3 dan Gambar B.7.

6.1.2.2 Tinjauan Antarmuka

Tinjauan antarmuka pada penelitian ini merupakan evaluasi subjektif melalui prototipe untuk mendapatkan masukan mengenai antarmuka sistem serta mengetahui kualitas antarmuka dari sistem. Hasil tinjauan navigasi kedua pengguna menyetujui setiap butir-butir pertanyaan mengenai antarmuka pengguna sistem. Hasil tinjauan antarmuka ini telah dilampirkan pada lampiran B Gambar B.4 dan Gambar B.8.

6.2 Evaluasi Traceability

Evaluasi *traceability* dilakukan dengan menelusuri mulai dari hasil dari pemodelan proses bisnis, analisis sampai dengan hasil dari perancangan yang telah dilakukan sebelumnya. Selain itu juga dilakukan tinjauan *traceability*.

6.2.1 Matriks Traceability

Matriks *traceability* digunakan untuk menelusuri hasil dari pemodelan proses bisnis, analisis persyaratan sampai dengan perancangan yang telah dibuat. Tabel 6.11 menunjukkan matriks *traceability* mulai dari pemodelan proses bisnis sampai dengan perancangan.

Tabel 6.11 Matriks Traceability

Kode aktivitas BPMN	Kode Fungsi	Kode Use Case	Nama Use Case	Diagram aktivitas	Diagram sequence	Antarmuka
A-BP-XXX	PD-F-002	A-UC-001	Kelola data pengguna	Gambar 4.5	Gambar 5.5	Gambar 5.23, Gambar 5.24, Gambar 5.25, Gambar 5.26
A-BP-001	PD-F-003	A-UC-002	Kelola pendataan wajib pajak	Gambar 4.6	Gambar 5.6	Gambar 5.28, Gambar 5.29, Gambar 5.30, Gambar 5.31
A-BP-002	PD-F-004	A-UC-003	Kelola pendataan objek pajak	Gambar 4.7	Gambar 5.7	Gambar 5.32, Gambar 5.33, Gambar 5.34, Gambar 5.35
A-BP-003	PD-F-005	A-UC-004	Kelola penetapan pajak	Gambar 4.8	Gambar 5.8	Gambar 5.36, Gambar 5.38, Gambar 5.39
A-BP-004	PD-F-006	A-UC-005	Cetak SKPD	Gambar 4.9	Gambar 5.9	Gambar 5.37
A-BP-008, A-BP-010	PD-F-010	A-UC-006	Lihat laporan penerimaan pajak	Gambar 4.10	Gambar 5.10	Gambar 5.27
A-BP-008, A-BP-010	PD-F-010	A-UC-007	Cetak laporan penerimaan pajak	Gambar 4.11	Gambar 5.11	Gambar 5.27

A-BP-007	PD-F-007	A-UC-008	Lihat buku register	Gambar 4.12	Gambar 5.12	Gambar 5.41
A-BP-007	PD-F-007	A-UC-009	Cetak buku register	Gambar 4.13	Gambar 5.13	Gambar 5.41
A-BP-005	PD-F-008	A-UC-010	Lihat buku ekspedisi	Gambar 4.14	Gambar 5.14	Gambar 5.40
A-BP-005	PD-F-008	A-UC-011	Cetak buku ekspedisi	Gambar 4.15	Gambar 5.15	Gambar 5.40
A-BP-006	PD-F-008	A-UC-012	Validasi penyerahan SKPD	Gambar 4.16	Gambar 5.16	Gambar 5.40
A-BP-009	PD-F-009	A-UC-013	Validasi pembayaran	Gambar 4.17	Gambar 5.17	-
A-BP-XXY	PD-F-001	A-UC-014	Login	Gambar 4.4	Gambar 5.4	Gambar 5.22

6.2.2 Tinjauan Traceability

Tinjauan *traceability* merupakan tinjauan yang dilakukan oleh pengembang atas *traceability* yang telah dilakukan. Tabel 6.12 berikut ini merupakan hasil dari tinjauan *traceability*.

Tabel 6.12 Tinjauan *traceability*

No.	Pertanyaan	Ya	Tidak	Keterangan
1	Apakah setiap persyaratan didefinisikan dengan benar dan unik?	✓		Setiap persyaratan memiliki kode unik dengan kombinasi huruf dan angka serta telah didefinisikan dengan benar dan tidak ambigu.
2	Apakah setiap persyaratan fungsional dari perangkat lunak dapat dilacak sampai persyaratan tingkat tinggi? (seperti <i>use case</i>)	✓		Setiap persyaratan fungsional dapat dilacak sampai persyaratan tingkat tinggi dibuktikan dengan tabel <i>traceability</i>
3	Dapatkah semua bagian desain antarmuka tingkat tinggi dapat dilacak kembali sesuai persyaratan?	✓		Semua desain antarmuka tingkat tinggi dapat dilacak kembali sesuai persyaratan dibuktikan dengan tabel <i>traceability</i> .

BAB 7 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap penelitian yang dilakukan. Kesimpulan berisi jawaban atas rumusan masalah/pertanyaan penelitian. Saran berisi pernyataan-pernyataan ringkas dan jelas mengenai masalah-masalah atau hal-hal yang dapat dilakukan untuk mengembangkan penelitian ini secara lebih lanjut.

7.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Analisis proses bisnis yang dilakukan dalam penelitian ini mencakup proses bisnis yang saat ini berjalan dan proses bisnis yang diusulkan. Pada proses bisnis saat ini terdapat aktivitas-aktivitas yang masih dilakukan secara manual dan tidak terintegrasi. Pada proses bisnis usulan terdapat 10 perubahan dari proses bisnis yang saat ini berjalan. Perubahan pada proses bisnis usulan ini berupa tambahan aktivitas, perubahan aktivitas dan output data.
2. Analisis persyaratan pada penelitian ini dimulai dari analisis pemangku kepentingan sampai diagram aktivitas. Hasil dari analisis persyaratan pada penelitian ini terdiri dari 3 pemangku kepentingan, 2 tipe pengguna, 11 fitur, 11 persyaratan fungsional yang didetailkan menjadi 23, 1 persyaratan non fungsional, 6 aktor dan 14 *use case*. Analisis persyaratan ini dibuat dengan mempertimbangkan dengan proses bisnis usulan yang telah dibuat.
3. Perancangan sistem pada penelitian ini terdiri dari kelas analisis, elemen desain berupa *package*, perancangan *sequence diagram*, perancangan kelas diagram, pemodelan data, perancangan antarmuka serta perancangan prototipe yang disesuaikan dengan analisis persyaratan yang telah dilakukan.
4. Evaluasi penelitian ini terdiri dari evaluasi *traceability*, evaluasi skenario *use case* dan evaluasi terhadap prototipe.

Hasil evaluasi *traceability* yaitu :

- a. Setiap persyaratan sudah memiliki kode unik dan telah didefinisikan dengan benar,
- b. Persyaratan fungsional dapat dilacak sampai persyaratan tingkat tinggi,
- c. Semua desain antarmuka tingkat tinggi dapat dilacak kembali sesuai persyaratan dibuktikan dengan tabel *traceability*.

Hasil dari evaluasi skenario *use case* dan prototipe yaitu :

- a. Alur yang dilakukan dua pengguna yang melakukan pengujian skenario sudah sesuai dengan alur pada spesifikasi *use case* sistem.
- b. Dari 2 pengguna yang melakukan tinjauan terhadap 5 skenario *use case* semua menyatakan bahwa alur 5 skenario sudah sesuai dengan yang diharapkan pengguna, beberapa jenis-jenis data pada formulir pada 5 skenario *use case* perlu ditambahkan dan disempurnakan serta proses 5 skenario *use case* mudah untuk dijalankan.

- c. Hasil tinjauan navigasi kedua pengguna menyetujui setiap butir-butir pertanyaan mengenai navigasi sistem.
- d. Hasil tinjauan navigasi kedua pengguna menyetujui setiap butir-butir pertanyaan mengenai antarmuka pengguna sistem.

7.2 Saran

Saran yang dapat diberikan untuk penelitian lebih lanjut adalah sebagai berikut:

1. Melakukan perbaikan berdasarkan hasil evaluasi prototipe kepada pengguna.
2. Melakukan tahap pengembangan sistem berikutnya yaitu tahap perancangan detail dan implementasi sistem berdasarkan hasil analisis persyaratan dan perancangan sistem.
3. Mengembangkan sistem dengan menambahkan pajak lain selain pajak air tanah dan pajak parkir.



DAFTAR PUSTAKA

- Agarwal, B. B., Tayal, S. P. & Gupta, M., 2010. *Software engineering & testing*. United States: Jones and Bartlett.
- Amin, V., 2014. *Sistem Pelaporan dan Pengolahan Data Pajak Bumi dan Bangunan (PBB) Kabupaten Jombang Berbasis Web Menggunakan Framework Codeigniter*, Malang: Universitas Brawijaya.
- Badan Pusat Statistik, 2013. *Badan Pusat Statistik*. [Online]
Available at: <http://madiunkab.bps.go.id/linkTabelStatis/view/id/32>
[Diakses 4 Oktober 2015].
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. s.l.:Addison Wesley.
- Booch, G., Rumbaugh, J. & Jacobson, I., 2005. *The Unified Modeling Language User Guide*. 2nd penyunt. s.l.: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.
- Coyle, F. P., 2002. *XML, Web Services, and the Data Revolution*. s.l.:Addison Wesley.
- Dharwiyanti, S. & Wahono, R. S., 2003. *Pengantar Unified Modeling Language (UML)*. [Online]
Available at: <setia.staff.gunadarma.ac.id>
[Diakses 2015 Oktober 4].
- Dix, A., Finlay, J., Abowd, G. C. & Beale, R., 2004. *HUMAN-COMPUTER INTERACTION*. 3th penyunt. Harlow, England: Pearson/Prentice-Hall..
- DSDM Consortium, 2015. *DSDM Agile Project Framework*. 1st penyunt. United Kingdom: DSDM Consortium.
- Fatta, H. A., 2007. *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. [Online]
Available at:
<https://books.google.co.id/books?id=oHi8C1W4N7wC&pg=PA8&lpg=PA8&dq=koehsi+dan+coupling&source=bl&ots=3MTkEDb2Lo&sig=FmrkeTa72BcZ2K2C30cAxh-pZhC&hl=id&sa=X&ved=0ahUKEwjn75Dc7qrKAhXYc44KHbsIBSM4ChDoAQhFMAc#v=onepage&q=koehsi%20dan%20coupling&f=true>
[Diakses 19 Januari 2016].
- Fournier, G., 2009. *Essential Software Testing A Use-Case Approach*. Boca Raton, FL: Taylor & Francis Group.
- Fowler, M., 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd penyunt. s.l.:Addison Wesley.

- Hermawan, J., 2015. *Analisa & Desain Pemrograman Berorientasi Obyek dengan UML dan VISUAL BASIC. NET.* [Online]
Available at:
https://books.google.co.id/books?id=INoYnOCIRBkC&pg=PA118&dq=analisa+dan+desain+pemrograman&hl=id&sa=X&ved=0CBkQ6AEwAGoVChMI4_SYmqCoyAIVDsK0Ch2-jwBf#v=onepage&q=analisa%20dan%20desain%20pemrograman&f=false
[Diakses 4 Oktober 2015].
- IBM Corporation, 2003. *PRJ270: Essentials of Rational Unified Process Student Manual.* s.l.:International Business Machines Corporation,.
- IBM Corporation, 2004. *DEV475 Mastering Object-Oriented Analysis and Design with UML 2.0.* United States: IBM Rational Software.
- Kementrian Kesehatan RI Inspektorat Jenderal, 2014. UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 33 TAHUN 2004 TENTANG PERIMBANGAN KEUANGAN ANTARA PEMERINTAH PUSAT DAN PEMERINTAHAN DAERAH.[pdf] Kementrian Kesehatan RI Inspektorat Jenderal. Tersedia di:<
[http://www.itjen.depkes.go.id/public/upload/unit/pusat/files/Undang-undang/uu2004_33\(imbangkeuPusDa\).pdf](http://www.itjen.depkes.go.id/public/upload/unit/pusat/files/Undang-undang/uu2004_33(imbangkeuPusDa).pdf)> [Diakses 4 Oktober 2015]
- Newcomer, E., 2002. *Understanding Web Services- XML, WSDL, SOAP and UDDI.* s.l.:s.n.
- 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.* 15th penyunt. s.l.:s.n.
- Pant, K. & Juric, M., 2008. *Business Process Driven SOA using BPMN and BPEL.* s.l.:Packt Publishing Ltd..
- Pressman, R. S., 2001. *Software Engineering - A practitioner's approach.* 5th penyunt. Boston: McGraw-Hill.
- Pressman, R. S., 2010. *Software Engineering - A PRACTITIONER'S APPROACH.* 7 penyunt. s.l.:s.n.
- S., R. A. & Shalahuddin, M., 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek.* Bandung: Informatika Bandung.
- Seidl, M., Scholz, M., Huemer, C. & Kappel, G., 2012. *UML @ Classroom An Introduction to Object-Oriented Modeling.* Heidelberg: s.n.
- Siahaan, D., 2012. *Analisa Kebutuhan dalam Rekayasa Perangkat Lunak.* Yogyakarta: ANDI.

Simarmata, J., 2010. *Rekayasa Web : analisis dan design sistem, rekayasa informasi, rekayasa hypermedia, interaksi manusia dan komputer, rekayasa kebutuhan, data mining, manajemen proyek*. Yogyakarta: Andi.

Sugiyanto, S. M., 2007. *Pajak & Retribusi Daerah*. Jakarta: Grasindo.

Tantra, R., 2012. *Manajemen Proyek Sistem Informasi*. Yogyakarta: ANDI.

Tidwell, D., Snell, J. & Kulchenko, P., 2001. *Programming Web Services with SOAP*. s.l.:O'Reilly.

Weske, M., 2007. *Business Process Management*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg New York.

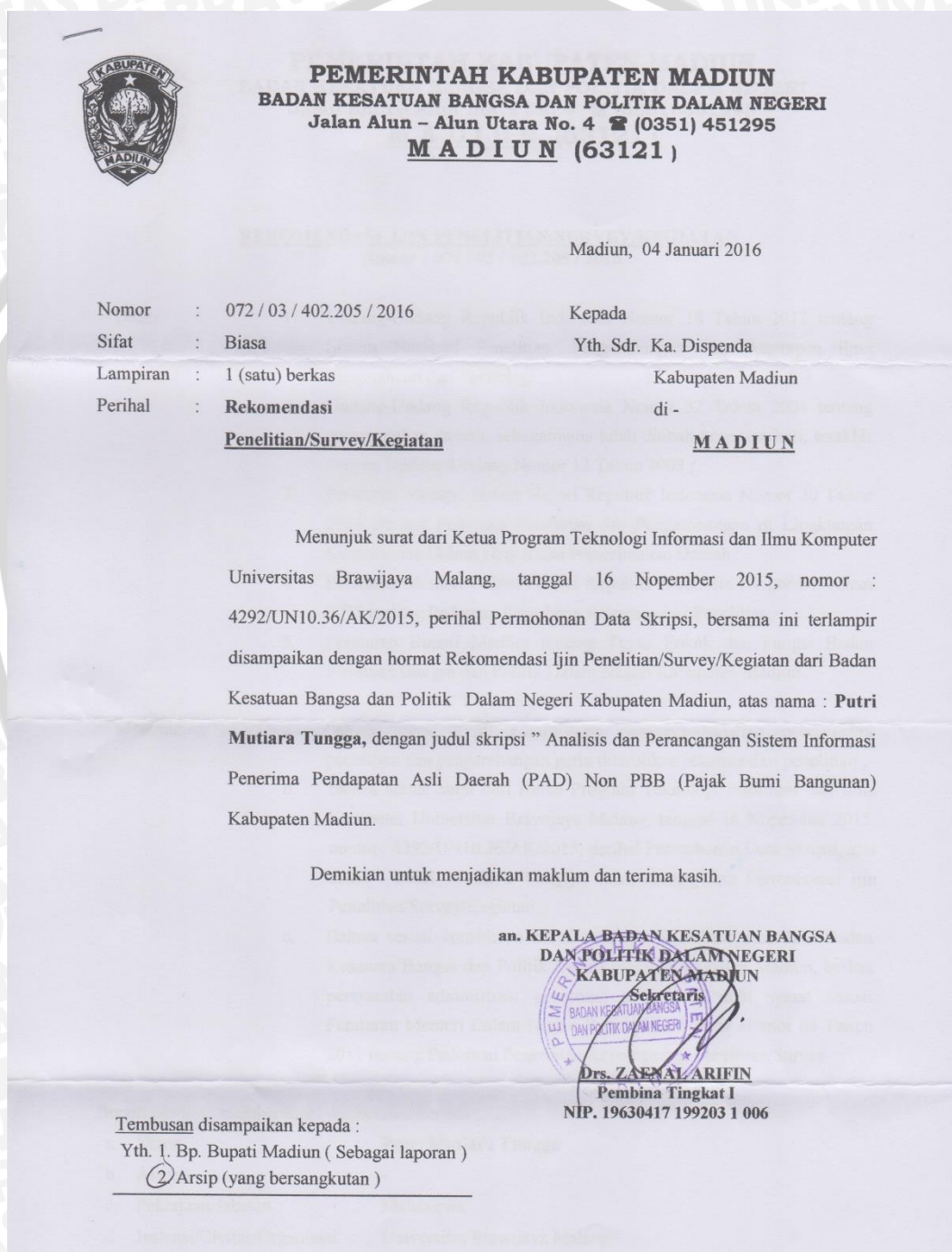
Yasin, V., 2012. *Rekayasa Perangkat Lunak Berorientasi Objek Pemodelan, Arsitektur dan Perancangan (Modelling, Architecture dan Design)*. Jakarta: Mitra Wacana Media.



LAMPIRAN A SURAT IZIN PENELITIAN

A.1 Surat Izin Penelitian Lembar Pertama

Gambar A.1 berikut menunjukkan lembar pertama surat izin untuk melakukan penelitian di DISPENDA Kabupaten Madiun dari Badan Kesatuan Bangsa dan Politik Dalam Negeri Kabupaten Madiun.



PEMERINTAH KABUPATEN MADIUN
BADAN KESATUAN BANGSA DAN POLITIK DALAM NEGERI
 Jalan Alun - Alun Utara No. 4 ☎ (0351) 451295
MADIUN (63121)

Madiun, 04 Januari 2016

Nomor : 072 / 03 / 402.205 / 2016 Kepada
 Sifat : Biasa Yth. Sdr. Ka. Dispenda
 Lampiran : 1 (satu) berkas Kabupaten Madiun
 Perihal : **Rekomendasi** di -
Penelitian/Survey/Kegiatan **MADIUN**

Menunjuk surat dari Ketua Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang, tanggal 16 Nopember 2015, nomor : 4292/UN10.36/AK/2015, perihal Permohonan Data Skripsi, bersama ini terlampir disampaikan dengan hormat Rekomendasi Ijin Penelitian/Survey/Kegiatan dari Badan Kesatuan Bangsa dan Politik Dalam Negeri Kabupaten Madiun, atas nama : **Putri Mutiara Tungga**, dengan judul skripsi " Analisis dan Perancangan Sistem Informasi Penerima Pendapatan Asli Daerah (PAD) Non PBB (Pajak Bumi Bangunan) Kabupaten Madiun.

Demikian untuk menjadikan maklum dan terima kasih.

an. KEPALA BADAN KESATUAN BANGSA DAN POLITIK DALAM NEGERI KABUPATEN MADIUN
 Sekretaris
Drs. ZAENAL ARIFIN
 Pembina Tingkat I
 NIP. 19630417 199203 1 006

Tembusan disampaikan kepada :
 Yth. 1. Bp. Bupati Madiun (Sebagai laporan)
 ② Arsip (yang bersangkutan)

Gambar A.1 Surat izin peneltian lembar pertama

A.2 Surat Izin Penelitian Lembar kedua

Gambar A.2 berikut menunjukkan lembar kedua surat izin untuk melakukan penelitian di DISPENDA Kabupaten Madiun dari Badan Kesatuan Bangsa dan Politik Dalam Negeri Kabupaten Madiun.

 **PEMERINTAH KABUPATEN MADIUN**
BADAN KESATUAN BANGSA DAN POLITIK DALAM NEGERI
Jalan Alun – Alun Utara No. 4 ☎ (0351) 451295
MADIUN (63121)

REKOMENDASI IJIN PENELITIAN/SURVEY/KEGIATAN
Nomor : 072 / 03 / 402.205 / 2016

Dasar : 1. Undang-Undang Republik Indonesia Nomor 18 Tahun 2012 tentang Sistem Nasional Penelitian, Pengembangan dan Penerapan Ilmu Pengetahuan dan Teknologi ;
2. Undang-Undang Republik Indonesia Nomor 32 Tahun 2004 tentang pemerintahan daerah, sebagaimana telah diubah beberapa kali, terakhir dengan Undang-Undang Nomor 12 Tahun 2008 ;
3. Peraturan Menteri Dalam Negeri Republik Indonesia Nomor 20 Tahun 2011 tentang Pedoman Penelitian dan Pengembangan di Lingkungan Kementerian Dalam Negeri dan Pemerintahan Daerah ;
4. Peraturan Menteri Dalam Negeri Republik Indonesia Nomor 64 Tahun 2011 tentang Pedoman Penerbitan Rekomendasi Penelitian;
5. Peraturan Bupati Madiun tentang Tugas Pokok dan Fungsi Badan Kesatuan Bangsa dan Politik Dalam Negeri Kabupaten Madiun.

Menimbang : a. bahwa untuk tertib administrasi dan pengendalian pelaksanaan penelitian dan pengembangan perlu diterbitkan rekomendasi penelitian ;
b. bahwa sesuai surat dari Ketua Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang, tanggal 16 Nopember 2015, nomor : 4292/UN10.36/AK/2015, perihal Permohonan Data Skripsi, atas nama : **Putri Mutiara Tungga**, telah mengajukan Permohonan Ijin Penelitian/Survey/Kegiatan ;
c. Bahwa sesuai konsideran huruf a dan b, serta hasil ferivikasi Badan Kesatuan Bangsa dan Politik Dalam Negeri Kabupaten Madiun, berkas persyaratan administrasi penelitian telah memenuhi syarat sesuai Peraturan Menteri Dalam Negeri Republik Indonesia Nomor 64 Tahun 2011 tentang Pedoman Penerbitan Rekomendasi Penelitian/Survey.

Bupati Madiun, memberikan rekomendasi kepada :

a. Nama : **Putri Mutiara Tungga**
b. Alamat : -
c. Pekerjaan/Jabatan : Mahasiswa
d. Instansi/Civitas/Organisasi : Universitas Brawijaya Malang
e. Kebangsaan : Indonesia

Gambar A.2 Surat izin penelitian lembar kedua

A.3 Surat Izin Penelitian Lembar ketiga

Gambar A.3 berikut menunjukkan lembar ketiga surat izin untuk melakukan penelitian di DISPENDA Kabupaten Madiun dari Badan Kesatuan Bangsa dan Politik Dalam Negeri Kabupaten Madiun.

Untuk mengadakan kegiatan/penelitian/survey/research dengan :

a. Judul	: Analisis dan Perancangan Sistem Informasi Penerima Pendapatan Asli Daerah (PAD) Non PBB (Pajak Bumi Bangunan) Kabupaten Madiun
b. Bidang Penelitian	: Sistem Informasi
c. Tujuan	: Permintaan Data
d. Pembimbing	: - Ismiarta Aknuranda, S.T,M.Sc, Ph.D - Satrio Agung Wicaksono, S.Kom, M.Kom
e. Anggota/Peserta	: -
f. Tanggal (Waktu)	: 11 November 2015 s/d 11 Februari 2016
g. Tempat/Lokasi	: Dispenda Kabupaten Madiun

Dengan Ketentuan

1. Berkewajiban menghormati dan mentaati peraturan dan tata tertib di daerah setempat / lokasi Penelitian/Survey/ Kegiatan ;
2. Pelaksanaan penelitian/Kegiatan agar tidak disalahgunakan untuk tujuan tertentu yang dapat mengganggu kestabilan keamanan dan ketertiban di daerah setempat/lokasi Penelitian/Survey/Kegiatan ;
3. Melaporkan hasil penelitian dan sejenisnya kepada Gubernur Jawa Timur melalui Badan Kesatuan Bangsa dan Politik Dalam Negeri Kabupaten Madiun.

Demikian rekomendasi ini dibuat untuk dipergunakan seperlunya.

Madiun, 04 Januari 2016

**an. KEPALA BADAN KESATUAN BANGSA
DAN POLITIK DALAM NEGERI
KABUPATEN MADIUN**

Sekretaris
Drs. ZAENAL ARIFIN
Pembina Tingkat I
NIP. 19630417 199203 1 006

Tembusan disampaikan kepada :

Yth. 1. Bp. Bupati Madiun (Sebagai laporan)
2. Arsip (yang bersangkutan)

Gambar A.3 Surat Izin Penelitian lembar ketiga

LAMPIRAN B HASIL EVALUASI

B.1 Hasil tinjauan prototipe oleh Ari Nursurahmat, S. Sos lembar pertama

Gambar B.1 berikut ini menunjukkan lembar pertama hasil tinjauan prototipe atas skenario *use case* yang dilakukan oleh Bapak Ari Nursurahmat, S. Sos .

REVIEW PROTOTYPE (SKENARIO USE CASE)	
Use case : Kelola Pengguna	
Skenario : Berhasil Tambah Pengguna	
Pertanyaan	Jawaban
Apakah alur tambah pengguna sudah sesuai dengan yang diharapkan?	Ya
Apakah jenis-jenis data pada formulir tambah pengguna sudah benar dan lengkap?	Sudah lengkap
Apakah proses tambah pengguna mudah dioperasikan?	Mudah
Saran :	
Ditambah pilihan status pengguna	
Use case : Kelola Pengguna	
Skenario : Data pengguna tidak diisi dengan lengkap	
Pertanyaan	Jawaban
Apakah penanganan kesalahan sudah sesuai dengan yang diharapkan?	Sudah
Saran :	
Use case : Kelola Wajib Pajak	
Skenario : Edit Data Wajib Pajak	
Pertanyaan	Jawaban
Apakah alur edit data wajib pajak sudah sesuai dengan yang diharapkan?	Sudah
Apakah jenis-jenis data pada formulir edit data wajib pajak sudah benar dan lengkap?	Perlu ditambah
Apakah proses edit data wajib pajak mudah dioperasikan?	Mudah
Saran :	
Ditambah alamat lengkap	

Gambar B.1 Hasil tinjauan prototipe oleh Ari Nursurahmat, S. Sos lembar pertama

B.2 Hasil tinjauan prototipe oleh Ari Nursurahmat, S. Sos lembar kedua

Gambar B.2 berikut ini menunjukkan lembar kedua hasil tinjauan prototipe atas skenario *use case* yang dilakukan oleh Bapak Ari Nursurahmat, S. Sos .

Use case : Cetak SKPD	
Skenario : Berhasil melakukan cetak SKPD	
Pertanyaan	Jawaban
Apakah alur cetak SKPD sudah sesuai dengan yang diharapkan?	Sudah
Apakah jenis-jenis data pada formulir cetak SKPD sudah benar dan lengkap?	perlu ditambah
Apakah proses cetak SKPD mudah dioperasikan?	mudah
Saran :	
1) ditambah periode tel & bulan 2) mau data skpd sebaiknya ditaruh setelah formulir penetapan.	
Use case : Lihat buku ekspedisi	
Skenario : Berhasil melakukan Lihat Buku Ekspedisi	
Pertanyaan	Jawaban
Apakah alur lihat buku ekspedisi sudah sesuai dengan yang diharapkan?	Sudah
Apakah jenis-jenis data pada tabel buku ekspedisi sudah benar dan lengkap?	Sudah
Apakah proses lihat buku ekspedisi mudah dioperasikan?	mudah
Saran :	
—	

Gambar B.2 Hasil tinjauan prototipe oleh Ari Nursurahmat, S. Sos lembar kedua

B.3 Hasil tinjauan navigasi prototipe oleh Ari Nursurahmat, S. Sos

Gambar B.3 berikut ini menunjukkan hasil evaluasi navigasi pada prototipe yang dilakukan oleh Bapak Ari Nursurahmat, S. Sos .

REVIEW PROTOTYPE (NAVIGASI)

Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui posisi 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	✓			
Navigasi menu pada sistem sudah menunjukkan fungsi dengan jelas	✓			
Masukan dan Saran				

Hari, 12 Februari 2019
 NABID PERKASIPAN
 DAN PERKASIPAN

Gambar B.3 Hasil tinjauan navigasi prototipe oleh Ari Nursurahmat, S. Sos

B.4 Hasil tinjauan antarmuka prototipe oleh Ari Nursurahmat, S. Sos

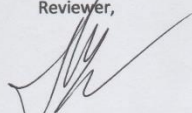
Gambar B.4 berikut ini menunjukkan hasil evaluasi antarmuka pada prototipe yang dilakukan oleh Bapak Ari Nursurahmat, S. Sos .

REVIEW PROTOTYPE (ANTARMUKA)

Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓			
Sistem memiliki antar muka yang konsisten	✓			
Masukan dan Saran	Saran pengirim			

Madiun, 9 Mei 2016

Reviewer,


ARI NURSURAHMAT
 KABID PEMAGI HARI
 DAN KEBERATAN

Gambar B.4 Hasil tinjauan antarmuka prototipe oleh Ari Nursurahmat, S. Sos

B.5 Hasil tinjauan prototipe oleh Budi Muratno, S. Sos lembar pertama

Gambar B.5 berikut ini menunjukkan lembar pertama hasil tinjauan prototipe atas skenario *use case* yang dilakukan oleh Bapak Budi Muratno, S. Sos.

REVIEW PROTOTYPE (SKENARIO USE CASE)	
Use case : Kelola Pengguna Skenario : Berhasil Tambah Pengguna	
Pertanyaan	Jawaban
Apakah alur tambah pengguna sudah sesuai dengan yang diharapkan?	Sudah sesuai harapan.
Apakah jenis-jenis data pada formulir tambah pengguna sudah benar dan lengkap?	Sudah benar & lengkap.
Apakah proses tambah pengguna mudah dioperasikan?	Sangat mudah.
Saran :	
Use case : Kelola Pengguna Skenario : Data pengguna tidak diisi dengan lengkap	
Pertanyaan	Jawaban
Apakah penanganan kesalahan sudah sesuai dengan yang diharapkan?	Sudah benar, untuk data yang salah satu kolom tidak diisi, sistem tidak berjalan (error).
Saran :	
Use case : Kelola Wajib Pajak Skenario : Edit Data Wajib Pajak	
Pertanyaan	Jawaban
Apakah alur edit data wajib pajak sudah sesuai dengan yang diharapkan?	Sudah baik, untuk edit data yang diharapkan.
Apakah jenis-jenis data pada formulir edit data wajib pajak sudah benar dan lengkap?	Sudah benar & lengkap.
Apakah proses edit data wajib pajak mudah dioperasikan?	Mudah sekali.
Saran :	

Gambar B.5 Hasil tinjauan prototipe oleh Budi Muratno, S. Sos lembar pertama

B.6 Hasil tinjauan prototipe oleh Budi Muratno, S. Sos lembar kedua

Gambar B.6 berikut ini merupakan lembar kedua hasil tinjauan prototipe atas skenario *use case* yang dilakukan oleh Bapak Budi Muratno, S. Sos.

Use case : Cetak SKPD	
Skenario : Berhasil melakukan cetak SKPD	
Pertanyaan	Jawaban
Apakah alur cetak SKPD sudah sesuai dengan yang diharapkan?	Sangat mudah dan proses cetak buku 1 lembar selesai.
Apakah jenis-jenis data pada formulir cetak SKPD sudah benar dan lengkap?	Sudah benar sesuai cetak SKPD.
Apakah proses cetak SKPD mudah dioperasikan?	Mudah selesai dan proses cetak.
Saran :	
Use case : Lihat buku ekspedisi	
Skenario : Berhasil melakukan Lihat Buku Ekspedisi	
Pertanyaan	Jawaban
Apakah alur lihat buku ekspedisi sudah sesuai dengan yang diharapkan?	Sangat mudah dan selesai melihat ke dalam buku ekspedisi.
Apakah jenis-jenis data pada tabel buku ekspedisi sudah benar dan lengkap?	Sangat lengkap.
Apakah proses lihat buku ekspedisi mudah dioperasikan?	Mudah dan selesai & operasi jalan.
Saran :	

Gambar B.6 Hasil tinjauan prototipe oleh Budi Muratno, S. Sos lembar kedua

B.7 Hasil tinjauan navigasi prototipe oleh Budi Muratno, S. Sos

Gambar B.7 berikut ini menunjukkan hasil evaluasi navigasi pada prototipe yang dilakukan oleh Bapak Budi Muratno, S. Sos.

REVIEW PROTOTYPE (NAVIGASI)

Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Saya dapat dengan mudah mengetahui posisi 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	✓	—	—	
Navigasi menu pada sistem sudah menunjukan fungsi dengan jelas	✓	—	—	
Masukan dan Saran	— Budi Muratno, S. Sos 14/05/2023			

Gambar B.7 Hasil tinjauan navigasi prototipe oleh Budi Muratno, S. Sos

B.8 Hasil tinjauan antarmuka prototipe oleh Budi Muratno, S. Sos

Gambar B.8 berikut ini menunjukkan hasil evaluasi antarmuka pada prototipe yang dilakukan oleh Bapak Budi Muratno, S. Sos.

REVIEW PROTOTYPE (ANTARMUKA)

Pertanyaan	Setuju	Netral	Tidak Setuju	Keterangan
Perancangan antar muka sistem mudah untuk dipelajari	✓	—	—	
Sistem memiliki antar muka yang konsisten	✓	—	—	
Masukan dan Saran	—			

Madiun, 9 Mei 2016

Reviewer,

Budi Muratno
Budi Muratno

Gambar B.8 Hasil tinjauan antarmuka prototipe oleh Budi Muratno, S. Sos