

**PENGEMBANGAN SISTEM INFORMASI KEUANGAN
PEGAWAI BIRO ORGANISASI SEKRETARIAT DAERAH
PROVINSI JAWA TIMUR**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Luluk Mukarromah

NIM: 145150400111029



**PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI KEUANGAN PEGAWAI BIRO ORGANISASI
SEKRETARIAT DAERAH PROVINSI JAWA TIMUR

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Luluk Mukarromah
NIM: 145150400111029

Skripsi ini telah diuji dan dinyatakan lulus pada
28 November 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Fajar Pradana, S.ST., M.Eng
NIP. 19871121 201504 1 004

Dosen Pembimbing II

M. Chandra Saputra S.Kom., M.T., M.Eng.
NIK: 201609 860106 1 001

Mengetahui
Ketua Jurusan Sistem Informasi



Dic. Eng. Herman Tolle, 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, 10 Oktober 2018



Luluk Mukarromah

NIM: 145150400111029

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem Informasi Keuangan Pegawai Biro Organisasi Sekretariat Daerah Provinsi Jawa timur”. Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer dan bapak Herman Tolle, Dr. Eng., S.T, M.T selaku Ketua Jurusan Sistem Informasi yang telah memberikan izin dalam penulisan skripsi ini.
2. Bapak Fajar Pradana, S.ST, M.Eng selaku dosen Pembimbing 1 dan bapak M. Chandra Saputra , S.Kom., M.Eng selaku dosen Pembimbing 2, yang telah membimbing, memberi saran serta motivasi kepada penulis selama penyusunan laporan skripsi.
3. Seluruh Dosen dan seluruh civitas akademika Program Teknologi Informasi dan Ilmu Komputer yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Universitas Brawijaya.
4. Bapak Edi Hartanto, SE, bapak Moh. Aziz Al Basid, S.IP, bapak Ramli, ibu Happy Poerbodianti, SE dan seluruh keluarga besar Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur yang bersedia membimbing serta menerima penulis untuk dapat melaksanakan pengerjaan skripsi.
5. Bapak Mohamad Safi'i dan ibu Hatimah selaku orang tua dari penulis yang selama ini telah sangat berjasa dalam membimbing dan membesarkan penulis, selalu mencurahkan kasih sayang, doa, serta berbagai dukungan moral maupun materi kepada penulis. Serta saudara-saudara saya yang saya sayangi.
6. Seluruh teman-teman yang tidak dapat penulis sebutkan satu per satu yang telah memberikan semangat dan dukungan agar skripsi ini cepat terselesaikan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 10 Oktober 2018

Penulis

Luluk1300@gmail.com

ABSTRAK

Pengelolaan keuangan pegawai pada Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur masih kurang efisien. Pengelolaan keuangan seperti pembuatan dokumen-dokumen keuangan masih dikerjakan secara manual dengan bantuan *microsoft excel*. Hal tersebut memungkinkan terjadinya ketidakakuratan data yang disebabkan *human error* dan berdampak pada waktu penyelesaian yang lama. Peningkatan efisiensi pengelolaan keuangan pegawai dapat dicapai dengan dukungan sistem informasi. Penelitian pengembangan sistem informasi keuangan pegawai dilakukan untuk menghasilkan sistem informasi yang dapat meningkatkan efisiensi pengelolaan keuangan pegawai. Penelitian diawali dengan proses identifikasi permasalahan, studi literatur, kemudian pengumpulan data sebagai bahan untuk melakukan proses pengembangan sistem informasi. Pengembangan sistem informasi dilakukan dengan metode pengembangan *Rational Unified Process* (RUP). Pada fase *inception* dilakukan pemodelan proses bisnis *as-is* dan *to-be*, analisis kebutuhan fungsional dan nonfungsional, serta identifikasi fitur. Fase *elaboration* menghasilkan *use case diagram* sesuai dengan fitur yang telah diidentifikasi. Masing-masing *use case* memiliki skenario yang akan dijadikan dasar dalam memodelkan diagram perancangan seperti *activity diagram*, *sequence diagram*, dan perancangan algoritme. Sedangkan hasil pemodelan diagram kelas analisis akan dijadikan dasar dalam membuat kelas perancangan yang akan diimplementasikan kedalam *frame work codeigniter* serta *physical data model* yang akan diimplementasikan kedalam *Database Management System* (DBMS) *MySQL*. Setelah dilakukan implementasi sesuai dengan perancangan, akan dilakukan pengujian terhadap sistem informasi keuangan pegawai. Sistem informasi yang dihasilkan diuji menggunakan metode *validation testing*, *compatibility testing*, serta *usability testing*. Hasil pengujian menyatakan bahwa sistem informasi berjalan sesuai dengan kebutuhan dan dapat dengan mudah digunakan oleh pengguna.

Kata kunci: sistem informasi, keuangan pegawai, RUP, UML, pengujian sistem

ABSTRACT

The financial management of employees at Biro Organisasi of East Java Provincial Government is still less efficient. Financial document at biro organisasi was created manually with microsoft excel. This allows data inaccuracies caused by human error and takes a long time to finish the document. Increasing the efficiency of employee financial management can be achieved with the support of information systems. Research of employee financial information systems development is carried out to produce an information system that can improve the efficiency of employee financial management. Research begins with problem identification, literature study, then collecting data as the material to to develop informastion system. The development of information systems done by Rational Unified Process (RUP) development method. In the inception phase, as-is business process and to be business process will be modeled. Analysis of functional and non-functional needs, and feature identification will also be done in inception phase. The elaboration phase produces a use case diagram according to the features that have been identified. Each use case has a scenario that will be used as a basis for modeling design diagrams such as activity diagrams, sequence diagrams, and algorithmic design. While the results of the analysis class diagram modeling will be used as the basis for creating a design class that will be implemented into codeigniter frame work and physical data model that will be implemented into the MySql Database Management System (DBMS). After implementation in accordance with the design, testing of the employee's financial information system will be carried out. The information system was tested using validation testing, compatibility testing, and usability testing. The test results state that the information system runs according to needs and can be easily used by users.

Keywords: *information system, employee finances, RUP, UML, system testing*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN	xix
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	4
1.4 Manfaat.....	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur.....	6
2.1.1 Visi dan Misi	7
2.1.2 Struktur Organisasi.....	7
2.1.3 Uang Makan, Uang Sidang, Uang Lembur, Uang Perjalanan Dinas, dan Tambahan Penghasilan Pegawai.....	7
2.2 <i>Business Process Model and Notation</i> (BPMN)	8
2.3 Pengembangan Perangkat Lunak	12
2.3.1 <i>Rational Unified Process</i> (RUP)	12
2.3.2 Pemodelan <i>Use Case</i>	16
2.3.3 Basis Data	19
2.3.4 Pendekatan Berorientasi Objek	20
2.3.5 Unified Modeling Language	20
2.3.6 Pengujian.....	28

2.4 Teknologi Pengembangan Sistem	32
2.4.1 PHP	32
2.4.2 HTML	32
2.4.3 SQL.....	32
2.4.4 MySQL	32
2.4.5 <i>CodeIgniter</i> (CI)	32
BAB 3 METODOLOGI	34
3.1 Identifikasi Masalah	35
3.2 Studi Literatur	35
3.3 Pengumpulan Data	35
3.4 <i>Inception</i>	36
3.5 <i>Elaboration</i>	36
3.6 <i>Construction</i>	37
3.7 <i>Transition</i>	37
3.8 Kesimpulan dan Saran	38
BAB 4 ANALISIS PERSYARATAN	39
4.1 Proses Bisnis.....	39
4.1.1 Proses Bisnis Pencairan Uang Sidang.....	39
4.1.2 Proses Bisnis Pencairan Uang Perjalanan Dinas	41
4.1.3 Proses Bisnis Pencairan Uang Makan	45
4.1.4 Proses Bisnis Pencairan Uang Lembur	49
4.1.5 Proses Bisnis Pencairan Tunjangan Prestasi	53
4.1.6 Proses Bisnis Pencairan Tunjangan Daerah	57
4.1.7 Proses Bisnis <i>Recording</i> Absensi dan Lembur Pegawai	61
4.2 Analisis Persyaratan	61
4.2.1 Identifikasi Tipe Pemangku Kepentingan.....	61
4.2.2 Analisis Permasalahan.....	62
4.2.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna .	64
4.2.4 Identifikasi Pengguna	67
4.2.5 Identifikasi Fitur	68
4.2.6 Persyaratan Fungsional	69
4.2.7 Persyaratan Nonfungsional.....	73

4.3	Pemodelan <i>Use Case</i>	73
4.3.1	<i>Use Case Diagram</i>	74
4.3.2	Deskripsi Aktor	79
4.3.3	Spesifikasi <i>Use Case</i>	80
4.4	Pemodelan Aktivitas	109
4.4.1	Diagram Aktivitas Memasukkan Data Pengajuan Absensi.....	109
4.4.2	Diagram Aktivitas Memasukkan Data Perjalanan Dinas.....	110
4.4.3	Diagram Aktivitas Mengecek Data SPJ.....	111
4.4.4	Diagram Aktivitas Membuat Laporan Rekapitulasi.....	112
4.4.5	Diagram Aktivitas Memasukkan Data Sidang	113
4.4.6	Diagram Aktivitas Mencetak Daftar Penerima dan Kwitansi Uang Sidang.....	114
4.5	Diagram Kelas Analisis	115
BAB 5 PERANCANGAN.....		117
5.1	Sequence Diagram	117
5.1.1	<i>Sequence Diagram</i> Memasukkan Data Pengajuan Absensi.....	117
5.1.2	<i>Sequence Diagram</i> Memasukkan Data Perjalanan Dinas	118
5.1.3	<i>Sequence Diagram</i> Mengecek Data SPJ.....	119
5.1.4	<i>Sequence Diagram</i> Membuat Laporan Rekapitulasi.....	120
5.1.5	<i>Sequence Diagram</i> Memasukkan Data Sidang	121
5.1.6	<i>Sequence Diagram</i> Mencetak Daftar penerima dan Kwitansi Uang Sidang.....	122
5.2	Pemodelan Objek.....	123
5.2.1	Diagram Kelas Perancangan.....	123
5.2.2	Perancangan Basis Data	126
5.3	Perancangan Algoritme	127
5.3.1	Algoritme Memasukkan Data Pengajuan Absensi	127
5.3.2	Algoritme Memasukkan Data Perjalanan Dinas	128
5.3.3	Algoritme Mengecek Data SPJ	128
5.3.4	Algoritme Membuat Laporan Rekapitulasi	130
5.3.5	Algoritme Memasukkan Data Sidang.....	132
5.3.6	Algoritme Mencetak Daftar Penerima dan Kwitansi Uang Sidang	133

5.4 Perancangan Antarmuka Pengguna	135
5.4.1 Antarmuka <i>Login</i>	135
5.4.2 Antarmuka Daftar Perjalanan Dinas.....	135
5.4.3 Antarmuka Detail Perjalanan Dinas	136
5.4.4 Antarmuka Daftar Sidang.....	137
5.4.5 Antarmuka Daftar Laporan Rekapitulasi.....	138
BAB 6 IMPLEMENTASI	140
6.1 Spesifikasi Lingkungan Sistem Informasi	140
6.2 Implementasi Algoritma	141
6.2.1 Implementasi Algoritme Memasukkan Data Pengajuan Absensi	141
6.2.2 Implementasi Algoritme Memasukkan Data Perjalanan Dinas	142
6.2.3 Implementasi Algoritme Mengecek Data SPJ	143
6.2.4 Implementasi Algoritme Membuat Laporan rekapitulasi.....	145
6.2.5 Implementasi Algoritme Memasukkan Data Sidang.....	152
6.2.6 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang.....	153
6.3 Implementasi Antarmuka Pengguna	161
6.3.1 Antarmuka <i>Login</i>	161
6.3.2 Antarmuka Daftar Perjalanan Dinas.....	162
6.3.3 Antarmuka Detail Perjalanan Dinas	163
6.3.4 Antarmuka Daftar Sidang.....	164
6.3.5 Antarmuka Daftar Laporan Rekapitulasi.....	165
6.3.6 Penambahan Kebutuhan Fungsional	165
BAB 7 PENGUJIAN	166
7.1 Pengujian <i>Validation Testing</i>	166
7.2 Pengujian <i>Compatibility Testing</i>	174
7.3 Pengujian Usability Testing.....	176
7.4 <i>Time Analysis</i> Pada Proses Bisnis <i>As-is dan To-be</i>	180
7.4.1 Proses Bisnis Pencairan Uang Sidang.....	180
7.4.2 Proses Bisnis Pencairan Uang Perjalanan Dinas	181
7.4.3 Proses Bisnis Pencairan Uang Makan	181

7.4.4 Proses Bisnis Pencairan Uang Lembur	182
7.4.5 Proses Bisnis Pencairan Tunjangan Prestasi	183
7.4.6 Proses Bisnis Pencairan Tunjangan Daerah	185
BAB 8 KESIMPULAN.....	186
8.1 Kesimpulan.....	186
8.2 Saran	187
DAFTAR PUSTAKA.....	188
LAMPIRAN A HASIL WAWANCARA.....	190
LAMPIRAN B FORM PENGUJIAN USABILITAS.....	193



DAFTAR TABEL

Tabel 2. 1 Notasi <i>Flow Object</i>	8
Tabel 2. 2 Notasi <i>Connecting Object</i>	11
Tabel 2. 3 Notasi <i>Swimlanes</i>	12
Tabel 2. 4 Notasi <i>Artifact</i>	12
Tabel 2. 5 <i>Template</i> Dokumentasi Pernyataan Masalah	17
Tabel 2. 6 Contoh Tabel Fitur	18
Tabel 2. 7 Format Dokumentasi Spesifikasi <i>Use Case</i>	19
Tabel 2. 8 Notasi pada <i>Activity Diagram</i>	25
Tabel 2. 9 Notasi pada <i>Sequence Diagram</i>	27
Tabel 4. 1 Tipe Pemangku Kepentingan.....	62
Tabel 4. 2 <i>Problem Statement 1</i>	62
Tabel 4. 3 <i>Problem Statement 2</i>	63
Tabel 4. 4 Hasil Identifikasi Kebutuhan Pengguna.....	65
Tabel 4. 5 Hasil Identifikasi Pengguna.....	67
Tabel 4. 6 Hasil Identifikasi Fitur	68
Tabel 4. 7 Hubungan fitur dengan kebutuhan pengguna	69
Tabel 4. 8 Persyaratan Fungsional	70
Tabel 4. 9 Persyaratan Nonfungsional	73
Tabel 4. 10 Hubungan Aktivitas Proses Bisnis (<i>to-be</i>) Dengan <i>Use Case</i>	75
Tabel 4. 11 Hubungan <i>Use Case</i> Dengan Pemangku Kepentingan	77
Tabel 4. 12 Hubungan <i>Use Case</i> dengan Fitur	78
Tabel 4. 13 Deskripsi Aktor	79
Tabel 4. 14 Spesifikasi <i>Use Case</i> Masuk ke SIKAP	80
Tabel 4. 15 Spesifikasi <i>Use Case</i> Memasukkan Data Pengajuan Absensi	81
Tabel 4. 16 Spesifikasi <i>Use Case</i> Mengubah Data Pengajuan Absensi	82
Tabel 4. 17 Spesifikasi <i>Use Case</i> Menghapus Data Pengajuan Absensi.....	83
Tabel 4. 18 Spesifikasi <i>Use Case</i> Membuat Laporan Perjalanan Dinas	83
Tabel 4. 19 Spesifikasi <i>Use Case</i> Membuat Laporan Rekapitulasi	84
Tabel 4. 20 Spesifikasi <i>Use Case</i> Menyetujui Pengajuan Absensi.....	85
Tabel 4. 21 Spesifikasi <i>Use Case</i> Menolak Pengajuan Absensi	86

Tabel 4. 22 Spesifikasi <i>Use Case</i> Memasukkan Data Absensi	87
Tabel 4. 23 Spesifikasi <i>Use Case</i> Mengubah Data Absensi	88
Tabel 4. 24 Spesifikasi <i>Use Case</i> Menghapus Data Absensi.....	89
Tabel 4. 25 Spesifikasi <i>Use Case</i> Memasukkan Data Lembur	89
Tabel 4. 26 Spesifikasi <i>Use Case</i> Mengubah Data Lembur	90
Tabel 4. 27 Spesifikasi <i>Use Case</i> Menghapus Data Lembur.....	92
Tabel 4. 28 Spesifikasi <i>Use Case</i> Memasukkan Data perjalanan dinas.....	92
Tabel 4. 29 Spesifikasi <i>Use Case</i> Mengubah Data perjalanan dinas	93
Tabel 4. 30 Spesifikasi <i>Use Case</i> Menghapus Data perjalanan dinas	95
Tabel 4. 31 Spesifikasi <i>Use Case</i> Mencetak SPT	96
Tabel 4. 32 Spesifikasi <i>Use Case</i> Membuat SPPD.....	96
Tabel 4. 33 Spesifikasi <i>Use Case</i> Mengecek SPJ	97
Tabel 4. 34 Spesifikasi <i>Use Case</i> Membuat Kwitansi Uang Perjalanan Dinas.....	99
Tabel 4. 35 Spesifikasi <i>Use Case</i> Memasukkan Data Sidang.....	100
Tabel 4. 36 Spesifikasi <i>Use Case</i> Mengubah Data Sidang	101
Tabel 4. 37 Spesifikasi <i>Use Case</i> Menghapus Data Sidang.....	102
Tabel 4. 38 Spesifikasi <i>Use Case</i> Mencetak Daftar Penerima dan Kwitansi Uang Sidang.....	103
Tabel 4. 39 Spesifikasi <i>Use Case</i> Membuat SPP	104
Tabel 4. 40 Spesifikasi <i>Use Case</i> Memasukkan Data Presensi	105
Tabel 4. 41 Spesifikasi <i>Use Case</i> Memasukkan Data Pegawai	105
Tabel 4. 42 Spesifikasi <i>Use Case</i> Mengubah Data Pegawai	106
Tabel 4. 43 Spesifikasi <i>Use Case</i> Menghapus Data Pegawai.....	107
Tabel 4. 44 Spesifikasi <i>Use Case</i> Memasukkan Data Pegawai	108
Tabel 4. 45 Spesifikasi <i>Use Case</i> Menghapus Data Libur	109
Tabel 5. 1 <i>Pseudocode</i> Memasukkan Data Absensi	127
Tabel 5. 2 <i>Pseudocode</i> Memasukkan Data Perjalanan Dinas	128
Tabel 5. 3 <i>Pseudocode</i> Mengecek Data SPJ	129
Tabel 5. 4 <i>Pseudocode</i> Membuat Laporan Rekapitulasi	131
Tabel 5. 5 <i>Pseudocode</i> Memasukkan Data Sidang.....	132
Tabel 5. 6 <i>Pseudocode</i> Membuat Laporan Rekapitulasi	134
Tabel 6. 1 Spesifikasi Perangkat Keras	140

Tabel 6. 2 Spesifikasi Perangkat Lunak	140
Tabel 6. 3 Spesifikasi Minimal Lingkungan <i>Deployment</i>	141
Tabel 6. 4 Implementasi Algoritma Memasukkan Data Pengajuan Absensi	141
Tabel 6. 5 Implementasi Algoritma Memasukkan Data Perjalanan Dinas.....	142
Tabel 6. 6 Implementasi Algoritme Mengecek Data SPJ	143
Tabel 6. 7 Implementasi Algoritme Membuat Laporan Rekapitulasi	146
Tabel 6. 8 Implementasi Algoritme Memasukkan Data Sidang.....	152
Tabel 6. 9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang.....	154
Tabel 7. 1 Rancangan Pengujian Kasus Uji Memasukkan Data perjalanan dinas : <i>Basic Flow</i>	167
Tabel 7. 2 Rancangan Pengujian Kasus Uji Memasukkan Data perjalanan dinas : <i>Alternative Flow</i>	167
Tabel 7. 3 Rancangan Pengujian Kasus Uji Mengecek SPJ	168
Tabel 7. 4 Rancangan Pengujian Kasus Uji Memasukkan Data perjalanan dinas : <i>Alternative Flow</i>	168
Tabel 7. 5 Rancangan Pengujian Kasus Uji Membuat Laporan Rekapitulasi	169
Tabel 7. 6 Rancangan Pengujian Kasus Uji Memasukkan Data Sidang : <i>Basic Flow</i>	170
Tabel 7. 7 Rancangan Pengujian Kasus Uji Memasukkan Data Sidang : <i>Alternative Flow</i>	170
Tabel 7. 8 Rancangan Pengujian Kasus Uji Mencetak Daftar Penerima dan Kwitansi Uang Sidang.....	171
Tabel 7. 9 Hasil Pengujian <i>Validation Testing</i>	172
Tabel 7. 10 Rancangan Pengujian Kasus Uji Menguji Kompatibilitas Sistem Pada <i>Browser Mozilla Firefox, Opera, Google Chrome, IOS, dan Android</i>	175
Tabel 7. 11 <i>Form</i> Pernyataan SUS.....	177
Tabel 7. 12 Rancangan Kasus Uji Menguji Usabilitas Sistem Informasi Keuangan Pegawai (Pegawai Biro).....	178
Tabel 7. 13 Rancangan Kasus Uji Menguji Usabilitas Sistem Informasi Keuangan Pegawai (Bendahara Bagian)	178
Tabel 7. 14 Rancangan Kasus Uji Menguji Usabilitas Sistem Informasi Keuangan Pegawai (Bendahara Biro).....	179
Tabel 7. 15 Hasil Pengujian <i>Usability Testing</i>	179
Tabel 7. 16 Perubahan Aktivitas Proses Bisnis Pencairan Uang Sidang.....	180

Tabel 7. 17 Perubahan Aktivitas Proses Bisnis Pencairan Uang Perjalanan Dinas 181

Tabel 7. 18 Perubahan Aktivitas Proses Bisnis Pencairan Uang Makan 182

Tabel 7. 19 Perubahan Aktivitas Proses Bisnis Pencairan Uang Lembur 183

Tabel 7. 20 Perubahan Aktivitas Proses Bisnis Pencairan Tunjangan prestasi ... 184

Tabel 7. 21 Perubahan Aktivitas Proses Bisnis Pencairan Tunjangan daerah 185



DAFTAR GAMBAR

Gambar 2. 1 Bagan Struktur Organisasi Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur	7
Gambar 2. 2 Proses iterative RUP	13
Gambar 2. 3 Fase pada RUP	13
Gambar 2. 4 <i>Physical Data Model</i>	20
Gambar 2. 5 Class	21
Gambar 2. 6 Class Diagram	22
Gambar 2. 7 Contoh <i>Use Case Diagram</i>	23
Gambar 2. 8 <i>Generalization, include, dan extend</i>	24
Gambar 2. 9 Contoh <i>Activity Diagram</i>	24
Gambar 2. 10 Sequence Diagram	26
Gambar 2. 11 Kategori Penilaian Hasil Pengujian Kompatibilitas	29
Gambar 2. 12 System Usability Scale	30
Gambar 2. 13 Interpretasi grade skor SUS	31
Gambar 2. 14 Template test case	31
Gambar 3. 1 Metodologi Penelitian	34
Gambar 4. 1 Proses Bisnis Pencairan uang Sidang (<i>as-is</i>)	40
Gambar 4. 2 Proses Bisnis Pencairan Uang Sidang (<i>to-be</i>)	40
Gambar 4. 3 Proses Bisnis Pencairan Uang Perjalanan Dinas (<i>as-is</i>)	42
Gambar 4. 4 Proses Bisnis Pencairan Uang Perjalanan Dinas (<i>to-be</i>)	44
Gambar 4. 5 Proses Bisnis Pencairan Uang Makan (<i>as-is</i>)	46
Gambar 4. 6 Proses Bisnis Pencairan Uang Makan (<i>to-be</i>)	48
Gambar 4. 7 Proses Bisnis Pencairan Uang Lembur (<i>as-is</i>)	50
Gambar 4. 8 Proses Bisnis Pencairan Uang Lembur (<i>as-is</i>)	52
Gambar 4. 9 Proses Bisnis Pencairan Tunjangan Prestasi (<i>to-be</i>)	54
Gambar 4. 10 Proses Bisnis Pencairan Tunjangan Prestasi (<i>to-be</i>)	56
Gambar 4. 11 Proses Bisnis Pencairan Tunjangan Daerah (<i>as-is</i>)	58
Gambar 4. 12 Proses Bisnis Pencairan Tunjangan Daerah (<i>to-be</i>)	60
Gambar 4. 13 Proses Bisnis Recording Absensi dan Lembur (<i>to-be</i>)	61
Gambar 4. 14 Keterangan Kodifikasi Kebutuhan Pengguna	64

Gambar 4. 15 Kodefikasi Fitur	68
Gambar 4. 16 Kodefikasi Persyaratan Fungsional.....	69
Gambar 4. 17 Kodefikasi Persyaratan Nonfungsional	73
Gambar 4. 18 <i>Use Case Diagram</i> Sistem Informasi Keuangan Pegawai.....	74
Gambar 4. 19 Diagram Aktivitas Memasukkan Data Pengajuan Absensi.....	110
Gambar 4. 20 Diagram Aktivitas Memasukkan Data perjalanan dinas	111
Gambar 4. 21 Diagram Aktivitas Mengecek Data SPJ	112
Gambar 4. 22 Diagram Aktivitas Membuat Laporan Rekapitulasi.....	113
Gambar 4. 23 Diagram Aktivitas Membuat Rekapitulasi Uang Sidang.....	113
Gambar 4. 24 Diagram Aktivitas Mencetak Daftar Penerima dan Kwitansi Uang Sidang.....	114
Gambar 4. 25 Diagram Kelas Analisis.....	116
Gambar 5. 1 Sequence Diagram Memasukkan Data Pengajuan Absensi.....	117
Gambar 5. 2 <i>Sequence Diagram</i> Memasukkan Data Perjalanan Dinas	118
Gambar 5. 3 <i>Sequence Diagram</i> Mengecek Data SPJ	119
Gambar 5. 4 <i>Sequence Diagram</i> Membuat Laporan Rekapitulasi Uang Makan	120
Gambar 5. 5 <i>Sequence Diagram</i> Memasukkan Data Sidang.....	121
Gambar 5. 6 <i>Sequence Diagram</i> Mencetak Daftar Penerima dan Kwitansi Uang Sidang.....	122
Gambar 5. 7 Diagram Kelas Perancangan <i>Controller</i>	124
Gambar 5. 8 Diagram Kelas Perancangan Model	125
Gambar 5. 9 Physical Data Model SIKAP	126
Gambar 5. 10 Rancangan Antarmuka <i>Login</i>	135
Gambar 5. 11 Rancangan Antarmuka Daftar Perjalanan Dinas	136
Gambar 5. 12 Rancangan Antarmuka Detail Perjalanan Dinas.....	137
Gambar 5. 13 Rancangan Antarmuka Daftar Sidang	138
Gambar 5. 14 Rancangan Antarmuka Daftar Laporan Rekapitulasi	139
Gambar 6. 1 Antarmuka Login	161
Gambar 6. 2 Antarmuka Daftar Perjalanan Dinas	162
Gambar 6. 3 Antarmuka Detail Perjalanan Dinas	163
Gambar 6. 4 Antarmuka Daftar Sidang.....	164
Gambar 6. 5 Antarmuka Daftar Laporan Rekapitulasi	165

Gambar 7. 1 Kodefikasi hasil pengujian *validation testing*..... 166
Gambar 7. 2 Kodefikasi Pengan kompatibilitas 174
Gambar 7. 3 Hasil Pengujian Kompatibilitas 175
Gambar 7. 4 Kodefikasi Pengujian Usabilitas..... 176



DAFTAR LAMPIRAN

A.1 Hasil Wawancara Bendahara Biro	190
B.1 Pernyataan Usabilitas Oleh Bendahara Biro.....	193
B.2 Pernyataan Usabilitas Oleh Bendahara Bagian 1	194
B.3 Pernyataan Usabilitas Oleh Bendahara Bagian 2	195
B.4 Pernyataan Usabilitas Oleh Pegawai Biro.....	196



BAB 1 PENDAHULUAN

1.1 Latar belakang

Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur merupakan salah satu biro yang ada di Sekretariat Daerah Provinsi Jawa Timur. Biro Organisasi mempunyai tugas menyiapkan rumusan kebijakan, mengoordinasikan pelaksanaan tugas dan fungsi, serta pemantauan dan evaluasi program kegiatan dan penyelenggaraan pembinaan teknis administrasi dan sumber daya. Biro organisasi terdiri dari beberapa bagian yaitu Bagian Kelembagaan dan Analisis Jabatan, Bagian Pengembangan Kinerja dan Kepegawaian, serta Bagian Tata Laksana. Biro organisasi memiliki visi “Terdepan dalam mewujudkan organisasi Pemerintah Provinsi dan Kabupaten/Kota yang solid, efektif dan efisien”. Sedangkan misi dari Biro Organisasi yaitu “Penataan dan Penguatan Kelembagaan, Tatalaksana, Manajemen Sumber Daya Aparatur, Akuntabilitas dan pelayanan publik”.

Menurut pedoman kerja dan pelaksanaan tugas pemerintah daerah provinsi Jawa Timur tahun 2018 yang diatur dalam Peraturan Gubernur Jawa Timur nomor 64 tahun 2017, untuk meningkatkan kesejahteraan pegawai dapat diberikan tambahan penghasilan bagi pegawai negeri sipil (TPP) berupa tunjangan daerah dan tunjangan prestasi. Selain itu, pegawai mendapatkan uang makan dengan mempertimbangkan tingkat kehadiran. Pegawai juga mendapatkan uang sidang yang diberikan dalam rangka membahas sesuatu permasalahan yang dapat menghasilkan suatu *output*. Sedangkan untuk pekerjaan yang belum dapat diselesaikan dalam jam kerja dapat diselesaikan dengan cara lembur dan akan mendapatkan uang lembur. Proses pemberian tunjangan daerah, tunjangan prestasi, dan uang makan (keuangan pegawai) pada Biro Organisasi sering mengalami keterlambatan. Perhitungan keuangan pegawai dilakukan pada akhir bulan oleh masing-masing bendahara bagian dan membutuhkan waktu yang berbeda-beda untuk menyelesaikannya. Bendahara umum harus menunggu seluruh bendahara menyelesaikan tugasnya untuk mengajukan pencairan. Keuangan pegawai tersebut dihitung berdasarkan presensi pegawai selama satu bulan. Tidak adanya pencatatan absensi pegawai baik karena ijin sakit, cuti, atau perjalanan dinas menyulitkan proses perhitungan keuangan pegawai di Biro Organisasi.

Dalam Peraturan Gubernur Jawa Timur No 64 Tahun 2017, pegawai yang tidak masuk kerja akan dikenakan pemotongan tunjangan prestasi. Pemotongan tidak diberlakukan terhadap pegawai yang melaksanakan perjalanan dinas, diklat/kursus dan yang mengalami cuti (sakit, tahunan, alasan penting) yang dibuktikan dengan data pendukung. Karena tidak ada pencatatan perjalanan

dinas, ketika bendahara melakukan pendataan terhadap presensi yang kosong ada beberapa pegawai yang lupa bahwa pada saat presensi kosong tersebut mereka sedang melaksanakan perjalanan dinas. Dengan demikian presensi mereka dihitung alpa dan mendapatkan potongan tunjangan prestasi pada hari tersebut. Selain itu, pegawai yang sedang melakukan perjalanan dinas luar kota seharusnya tidak boleh melakukan presensi datang atau pulang, kecuali mereka yang melakukan perjalanan dinas dalam kota. Berkaitan dengan uang makan, pegawai yang sedang melakukan perjalanan dinas di dalam kota akan mendapatkan uang makan jika melakukan presensi datang atau pulang, sedangkan yang melakukan perjalanan dinas luar kota tidak. Saat ini belum ada kontrol absensi terhadap pegawai yang melakukan perjalanan dinas. Sehingga memungkinkan adanya pegawai yang melakukan perjalanan dinas luar kota tetapi melakukan presensi dan mendapatkan uang makan.

Terkait dengan perjalanan dinas, kadang terjadi ketidaksesuaian data antara Surat Perintah Tugas (SPT) dengan data keuangan perjalanan dinas. Hal tersebut disebabkan oleh tidak adanya integrasi antara pembuatan SPT dengan pembuatan kwitansi keuangan perjalanan dinas. Pembuatan SPT dapat dilakukan oleh pegawai yang bersangkutan atau bendahara. Pembuatan kwitansi uang saku dan transport perjalanan dinas dilakukan dengan memasukkan ulang data perjalanan dinas sesuai SPT yang telah dibuat. Tidak adanya integrasi antara data dokumen-dokumen perjalanan dinas serta kesalahan saat memasukkan data oleh bendahara dapat menyebabkan data perjalanan dinas yang tidak akurat. Permasalahan lain yaitu terjadinya perjalanan dinas ganda seorang pegawai pada hari yang sama. Penyebabnya yaitu tidak adanya pencatatan SPT yang telah dibuat agar dilakukan penolakan terhadap pengajuan SPT lain dengan pegawai yang sama pada hari yang sama.

Solusi terbaik dari permasalahan diatas yaitu dengan sebuah sistem informasi. sistem informasi (IS) merupakan kombinasi dari orang, perangkat keras, perangkat lunak, jaringan komunikasi, sumber data, kebijakan dan prosedur yang menyimpan, mengambil, mengubah, dan menyebarkan informasi dalam suatu organisasi (O'brien & Marakas, ed., 2010). Selama ini yang menyebabkan keterlambatan dan ketidakakuratan data keuangan di biro organisasi adalah kesalahan *input*, tidak adanya integrasi serta tidak adanya pencatatan presensi yang baik. Pemanfaatan sistem informasi secara efektif akan menghasilkan informasi yang dapat diterima secara tepat waktu, akurat, dan dapat dipercaya (Astuti & Dharmadiaksa, n.d.). Dengan adanya pencatatan data dan informasi pada sistem informasi yang akan dibangun akan mempermudah dan mempercepat proses perhitungan keuangan dan hasil yang diperoleh akan lebih akurat dibandingkan perhitungan manual oleh manusia.

Pada proses pengembangan sistem diharapkan dapat mengatasi perubahan dan menemukan kesalahan selama proses pengembangan sistem, tidak pada tahap akhir pengembangan sistem. Oleh karena itu pengembangan sistem menggunakan metode *Rational Unified Process* (RUP). RUP merupakan pendekatan pengembangan perangkat lunak yang dilakukan secara berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), dan lebih diarahkan pada penggunaan kasus (*use case driven*) (Kroll & Kruchten, 2003). Ada empat tahapan RUP yang akan dijalankan yaitu *inception* (permulaan), *elaboration* (perluasan/perencanaan), *construction* (konstruksi) dan *transition* (transisi). Analisis dan perancangan sistem dimodelkan dengan konsep OOAD (*Object Oriented Analysis and Design*). OOA (*Object Oriented Analysis*) adalah metode analisis yang memeriksa persyaratan (*requirement*) dari perspektif kelas dan objek yang ditemukan dalam domain masalah (Booch, 1994). Booch juga mengatakan bahwa OOD (*Object Oriented Design*) adalah metode desain yang meliputi proses dekomposisi berorientasi objek dan notasi untuk menggambarkan sistem yang sedang dirancang. Pemodelan sistem *object-oriented* biasanya menggunakan diagram UML (*Unified Modeling Language*) yang merupakan bahasa standar dalam menulis *blueprints* perangkat lunak yang digunakan untuk memvisualisasikan, menentukan, membuat, dan mendokumentasikan artefak dari sistem perangkat lunak (Booch, Rumbaugh & Jacobson, 2005).

Berdasarkan kebutuhan akan akurasi dan kecepatan proses administrasi keuangan pegawai serta pengontrolan terhadap perjalanan dinas di Biro Organisasi, penulis mengangkat topik dengan judul **“Pengembangan Sistem Informasi Keuangan Pegawai Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur”** dengan metode pengembangan *Rational Unified Process* (RUP). Dengan adanya sistem informasi dalam proses administrasi keuangan pegawai serta pencatatan perjalanan dinas pegawai diharapkan mampu meminimalisir kesalahan *input* dan mempercepat proses perhitungan keuangan pegawai. Sehingga dapat meningkatkan keakuratan perhitungan dan menghindari keterlambatan proses administrasi keuangan pegawai serta duplikasi perjalanan dinas di Biro Organisasi Setda Prov Jatim.

1.2 Rumusan masalah

1. Bagaimana sistem informasi yang dapat meningkatkan kecepatan dalam proses administrasi keuangan pegawai Biro Organisasi Setda Prov. Jatim?
2. Bagaimana sistem informasi yang dapat mengintegrasikan data dokumen-dokumen keuangan pegawai Biro Organisasi Setda Prov. Jatim?
3. Bagaimana sistem informasi yang dapat melakukan kontrol terhadap perjalanan dinas ganda pada Biro Organisasi Setda Prov. Jatim?
4. Bagaimana hasil analisis persyaratan Sistem Informasi Keuangan Pegawai?

5. Bagaimana hasil rancangan Sistem Informasi Keuangan Pegawai?
6. Bagaimana hasil pengujian Sistem Informasi keuangan Pegawai dengan metode *black box*?

1.3 Tujuan

1. Menghasilkan sistem informasi yang dapat meningkatkan kecepatan dalam proses administrasi keuangan pegawai Biro Organisasi Setda Prov. Jatim.
2. Menghasilkan sistem informasi yang dapat mengintegrasikan data dokumen-dokumen keuangan pegawai Biro Organisasi Setda Prov. Jatim.
3. Menghasilkan sistem informasi yang dapat melakukan kontrol terhadap perjalanan dinas Biro Organisasi Setda Prov. Jatim.
4. Mengetahui hasil analisis persyaratan Sistem Informasi Keuangan Pegawai.
5. Mengetahui hasil rancangan Sistem Informasi Keuangan Pegawai.
6. Mengetahui hasil pengujian Sistem Informasi keuangan Pegawai dengan metode *black box*.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat diantaranya:

1. Meningkatkan kecepatan administrasi keuangan pegawai Biro Organisasi Setda Prov. Jatim.
2. Mengintegrasikan data dokumen-dokumen keuangan pegawai Biro Organisasi Setda Prov. Jatim.
3. Menghindari perjalanan dinas ganda pegawai dalam hari yang sama.
4. Mengetahui hasil analisis persyaratan Sistem Informasi Keuangan Pegawai.
5. Mengetahui hasil rancangan Sistem Informasi Keuangan Pegawai.
6. Mengetahui hasil pengujian Sistem Informasi keuangan Pegawai dengan metode *black box*.

1.5 Batasan masalah

Berikut batasan masalah dalam penelitian ini

1. Dalam penelitian ini hanya mencakup permasalahan mengenai uang makan, tunjangan prestasi, tunjangan daerah, uang lembur, uang sidang presensi, serta data perjalanan dinas di Biro Organisasi Sekertariat daerah Provinsi Jawa Timur.
2. Pengembangan sistem dilakukan dari analisis kebutuhan, perancangan, implementasi, hingga pengujian.

1.6 Sistematika pembahasan

BAB I PENDAHULUAN

Pendahuluan berisikan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah serta sistematika penulisan dalam penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini akan membahas mengenai teori, temuan, atau bahan penelitian sebelumnya dari berbagai sumber yang berkaitan dengan topik penelitian. Teori yang diambil mengenai pengembangan perangkat lunak, *framework CodeIgniter*, dan bahasa pemrograman PHP.

BAB III METODOLOGI PENELITIAN

Bab ini berisi metodologi atau langkah-langkah yang dilakukan dalam penelitian ini.

BAB IV ANALISIS PERSYARATAN

Bab ini akan membahas mengenai pemodelan proses bisnis yang sedang diterapkan (*as-is*), analisis proses bisnis (*as-is*), pemodelan proses bisnis yang akan diterapkan (*to-be*) beserta analisis persyaratan sistem informasi keuangan pegawai di Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur

BAB V PERANCANGAN

Bagian ini akan membahas pemodelan diagram-diagram hasil perancangan serta desain antarmuka Sistem Informasi Keuangan Pegawai yang akan dibangun.

BAB VI IMPLEMENTASI

Bab ini akan membahas implementasi Sistem Informasi Keuangan Pegawai di Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur sesuai dengan hasil yang diperoleh dari proses analisis dan perancangan.

BAB VII PENGUJIAN

Bab ini akan membahas pengujian terhadap sistem yang telah diimplementasikan.

BAB VIII PENUTUP

Bab ini berisi kesimpulan berdasarkan hasil pengujian pada proses penelitian beserta saran untuk penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Penelitian serupa pernah dilakukan oleh Noviani Aisyah, A. B. Tjandrarini dan Pantjawati S. dengan judul “Rancang Bangun Sistem Informasi Penggajian Karyawan pada UD. New Sehati”. Penelitian tersebut bertujuan untuk menghasilkan sistem informasi penggajian pegawai secara terkomputerisasi dan menghindari *human error*. Penelitian lainnya dilakukan oleh Ahmat Josi dengan judul “Perancangan Aplikasi Penggajian pada Perguruan Tinggi (Studi Kasus Sekolah Tinggi XYZ)”. Penelitian ini bertujuan untuk menghasilkan rancangan aplikasi penghitungan gaji pegawai sehingga dapat mempersingkat waktu perhitungan serta menghindari *human error*. Dengan aplikasi yang dihasilkan, perhitungan gaji pegawai dapat dilakukan oleh pegawai yang bertugas dengan memasukkan data pegawai serta jumlah kehadiran. Akan tetapi pada penelitian ini aplikasi terbatas pada perhitungan gaji pegawai dan tidak memfasilitasi pembuatan struk gaji, laporan penggajian atau pengarsipan. Penelitian serupa lainnya juga dilakukan oleh Muhammad Ali Syakur dengan judul “Aplikasi Sistem Penggajian Guru Berbasis Web”. Tujuan dari penelitian tersebut yaitu untuk menghasilkan sebuah aplikasi berbasis web untuk melakukan perhitungan gaji pegawai. Dengan aplikasi ini dapat mengurangi *human error* dan mempersingkat waktu perhitungan gaji pegawai. Perbedaan dengan penelitian pada skripsi ini yaitu mengenai ruang lingkup penelitian. Pada penelitian ini sistem informasi yang akan dibangun merupakan sistem informasi keuangan pegawai menyangkut tunjangan prestasi, uang makan dan tunjangan daerah beserta hal yang mempengaruhinya. Hal yang mempengaruhi keuangan pegawai pada Biro Organisasi yaitu kehadiran pegawai.

Penelitian mengenai sistem informasi akuntansi pernah dilakukan oleh Yohannes Kurniawan, Marisa Karsen, Alvin Adiprasetyo, Hanny Juwitasary dan Diego Fernando Cabezas Tapia dengan judul “*Accounting Information System Implementation (A Case Study Approach)*”. Penelitian ini bertujuan untuk mengatasi permasalahan yang dihadapi oleh perusahaan XYZ mengenai akuntansi. Sebelumnya perusahaan XYZ menggunakan *microsoft excel* untuk menyelesaikan proses akuntansinya. Terdapat beberapa permasalahan ketika menggunakan *microsoft excel* sebagai alat akuntansi perusahaan XYZ salah satunya *excel* tidak terintegrasi satu dengan yang lain, sehingga proses perhitungan dilakukan secara manual dan membutuhkan waktu yang lama. Setelah diimplementasikan sistem informasi akuntansi “*accurate*” *set-up* data menjadi lebih cepat dan lebih mudah, data finansial dapat di-*update* secara *real time*, pengguna dapat menggunakan data kapan saja dan lebih cepat karena seluruh data pada sistem telah terintegrasi.

2.1 Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur

Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur merupakan salah satu biro yang ada di Sekretariat Daerah Provinsi Jawa Timur. Biro Organisasi mempunyai tugas menyiapkan rumusan kebijakan, mengoordinasikan pelaksanaan tugas dan fungsi, pemantauan dan evaluasi program kegiatan dan

penyelenggaraan pembinaan teknis administrasi dan sumber daya. Biro organisasi terdiri dari beberapa bagian yaitu Bagian Kelembagaan dan Analisis Jabatan, Bagian Pengembangan Kinerja dan Kepegawaian, dan Bagian Tata Laksana. Biro Organisasi merupakan unsur staf yang dipimpin oleh Kepala Biro serta berada di bawah dan bertanggungjawab kepada Gubernur melalui Sekretaris Daerah. Biro Organisasi beralamat di Jalan Pahlawan No. 110 Lantai 4 Surabaya Jawa Timur.

2.1.1 Visi dan Misi

Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur memiliki Visi “Terdepan dalam mewujudkan organisasi Pemerintah Provinsi dan Kabupaten / Kota yang solid, efektif dan efisien”. Sehingga untuk mencapai Visi tersebut, Biro organisasi memiliki Misi “Penataan dan Penguatan Kelembagaan, Tatalaksana, Manajemen Sumber Daya Aparatur, Akuntabilitas dan pelayanan publik”. Misi tersebut dilaksanakan pada periode 2014 s/d 2019. Sehingga visi yang dimiliki dapat tercapai sebelum periode tersebut berakhir.

2.1.2 Struktur Organisasi

Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur terdiri dari tiga bagian yaitu Bagian Kelembagaan dan Analisis Jabatan, Bagian Pengembangan Kinerja dan Kepegawaian, dan Bagian Tata Laksana. Sistem informasi yang dibangun terfokus pada subbagian Tata Usaha Biro yang menangani masalah keuangan pegawai di Biro Organisasi. Subbagian Tata Usaha Biro berada di bawah bagian Tatalaksana. Struktur organisasi Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Bagan Struktur Organisasi Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur

Sumber: Peraturan Gubernur Jawa Timur Nomor 58 Tahun 2016

2.1.3 Uang Makan, Uang Sidang, Uang Lembur, Uang Perjalanan Dinas, dan Tambahan Penghasilan Pegawai

Pegawai di biro organisasi mendapatkan hak beberapa diantaranya mendapatkan uang makan, uang sidang, uang lembur, uang perjalanan dinas, dan



tambahan penghasilan pegawai. Menurut pedoman kerja dan pelaksanaan tugas pemerintah daerah provinsi jawa timur tahun 2018 yang diatur dalam peraturan gubernur jawa timur nomor 64 tahun 2017, uang makan merupakan uang yang diberikan kepada PNS dengan mempertimbangkan tingkat kehadiran dan maksimal 22 hari kerja dalam satu bulan. Uang sidang adalah uang yang diberikan dalam rangka membahas sesuatu permasalahan yang dapat menghasilkan suatu *output*. Dalam satu hari uang sidang maksimal dapat dibayarkan paling banyak dua kali sidang. Pekerjaan yang belum dapat diselesaikan dalam jam kerja, dapat diselesaikan dengan cara lembur yang menghasilkan suatu output secara riil dan dapat dipertanggung jawabkan serta akan mendapat uang lembur. Uang perjalanan dinas merupakan uang yang diperoleh pegawai ketika melakukan perjalanan dinas. Pegawai yang melakukan perjalanan dinas akan mendapatkan uang harian, uang transport dan uang penginapan untuk melakukan perjalanan dinas. Tambahan penghasilan pegawai diberikan untuk meningkatkan kesejahteraan pegawai negeri sipil. Terdapat dua tambahan penghasilan yang diperoleh pegawai yaitu tunjangan daerah dan tunjangan prestasi.

2.2 Business Process Model and Notation (BPMN)

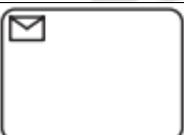
Business Process Model and Notation (BPMN) menyediakan beberapa diagram, yang dirancang untuk digunakan oleh orang-orang yang merancang dan mengelola Proses Bisnis. Proses bisnis terdiri dari serangkaian kegiatan yang dilakukan dalam koordinasi di lingkungan organisasi dan teknis (Weskey, 2007). Kegiatan-kegiatan ini dilakukan untuk mewujudkan tujuan bisnis. Proses bisnis suatu organisasi dapat berinteraksi dengan proses bisnis lainnya dari organisasi lain. BPMN dapat digunakan oleh perusahaan untuk memahami prosedur bisnis internal mereka dalam notasi grafis. BPMN akan digunakan untuk memodelkan proses bisnis yang ada saat ini (*as-is*) pada organisasi serta proses bisnis yang nantinya akan diterapkan (*to-be*). Sehingga dapat terlihat perbedaan proses bisnis sebelum dan sesudah adanya sistem informasi keuangan pegawai yang akan dikembangkan di Biro Organisasi.

Dalam pemodelan diagram proses bisnis dengan BPMN terdapat empat kategori dasar elemen yaitu *flow object*, *connecting object*, *swimlanes*, dan *artifacts*. *Flow object* terdiri dari *event*, *activity*, dan *gateway*. Notasi dan penjelasan mengenai *flow object* dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Notasi Flow Object

Flow Object	Notasi	Deskripsi
Task	 User Task	Task dimana <i>user</i> (pengguna) melakukan tugas dengan bantuan aplikasi perangkat lunak (sistem).

Tabel 2.1 Notasi *Flow Object* (Lanjutan)

<i>Flow Object</i>	Notasi	Deskripsi
	 Manual Task	<i>Task</i> yang dilaksanakan tanpa bantuan aplikasi (sistem).
	 Service Task	<i>Task</i> yang dilakukan menggunakan beberapa layanan seperti <i>web service</i> atau aplikasi otomatis.
	 Receive Task	<i>Task</i> menunggu untuk menerima pesan dari peserta luar (<i>external participant</i>).
	 Send	<i>Task</i> untuk mengirim pesan kepada <i>external participant</i> .
	 Script	<i>Task</i> yang dijalankan oleh aplikasi/sistem sesuai dengan skrip yang ada.
	 Business Rule	<i>Task</i> yang menyediakan mekanisme untuk memberikan masukan dan mendapatkan <i>output</i> dari perhitungan yang mungkin disediakan oleh <i>Business Rules Engine</i> .
	 Sub-Process	Jenis aktivitas dalam suatu proses untuk menunjukkan proses tingkat yang lebih rendah. Berguna untuk dekomposisi proses atau organisasi proses umum.
<i>Event</i>	 Event: Start	<i>Start event</i> menunjukkan di mana proses tertentu akan dimulai.
	 Event: Intermediate	<i>Intermediate event</i> terjadi antara <i>start event</i> dan <i>end event</i> .
	 Event: End	<i>End event</i> menunjukkan di mana proses akan berakhir.

Tabel 2.1 Notasi *Flow Object* (Lanjutan)

<i>Flow Object</i>	Notasi	Deskripsi
	<p>Receive Send</p>  <p>Event: Message</p>	Mengirim atau menerima pesan dari peserta
	<p>Receive Send</p>  <p>Event: Timer</p>	Digunakan untuk menandakan menunggu kondisi waktu tertentu untuk memulai Proses.
	<p>Receive Send</p>  <p>Event: Error</p>	Digunakan untuk melempar atau menangkap kesalahan dan untuk mengatasinya.
	<p>Receive Send</p>  <p>Event: Cancel</p>	Menunjukkan bahwa Transaksi harus dibatalkan.
	<p>Receive Send</p>  <p>Event: Compensation</p>	Menunjukkan bahwa kompensasi diperlukan.
	<p>Receive Send</p>  <p>Event: Conditional</p>	Digunakan untuk menentukan apakah akan memulai (atau melanjutkan) hanya jika kondisi tertentu benar.
	<p>Receive Send</p>  <p>Event: Link</p>	<i>Link</i> lempar/kirim sebagai "titik keluar" dari proses, dan <i>link</i> tangkap/terima sebagai "titik masuk" kedalam proses.
	<p>Receive Send</p>  <p>Event: Signal</p>	Digunakan untuk mengirim dan menerima sinyal
	<p>Receive Send</p>  <p>Event: Terminate</p>	Digunakan untuk "hentikan semua". Ketika mencapai <i>terminate event</i> , seluruh proses dihentikan, termasuk semua aktivitas paralel
	<p>Receive Send</p>  <p>Event: Terminate</p>	Digunakan untuk meringkas beberapa jenis peristiwa dengan simbol tunggal
<i>Gateway</i>	 <p>Exclusive Gateway</p>	Saat membelah, mengarahkan urutan aliran ke hanya salah satu cabang keluar, berdasarkan kondisi.
	 <p>Event-based Gateway</p>	Saat membelah, diikuti oleh <i>catching events</i> atau <i>receive tasks</i>

Tabel 2.1 Notasi *Flow Object* (Lanjutan)

<i>Flow Object</i>	Notasi	Deskripsi
Gateway	 Inclusive Gateway	Saat membelah, memungkinkan satu atau lebih cabang diaktifkan. Semua cabang yang aktif harus selesai sebelum bergabung.
	 Parallel Gateway	saat membelah, akan mengarahkan aliran ke semua cabang keluar. Ketika penggabungan, menunggu semua cabang selesai sebelum melanjutkan aliran.
	 Complex Gateway	Mendefinisikan perilaku yang tidak ditangkap oleh gateway lain.

Sumber: Diadaptasi dari Rosing, Scheer & Scheel (2015)

Connecting object terdiri dari beberapa notasi yang berfungsi untuk menghubungkan objek-objek lain pada diagram proses bisnis. Terdapat tiga jenis *connecting object* yaitu *sequence flow*, *message flow*, dan *association*. Notasi dan penjelasan mengenai *connecting object* dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Notasi *Connecting Object*

Nama	Notasi	Deskripsi
<i>Sequence Flow</i>		Digunakan untuk menunjukkan urutan kegiatan yang akan dilakukan dalam suatu proses.
<i>Message Flow</i>		Digunakan untuk menunjukkan aliran pesan antara dua peserta proses yang terpisah (entitas bisnis atau peran bisnis).
<i>Association</i>		digunakan untuk mengaitkan data, teks, dan Artefak lainnya dengan objek. Asosiasi digunakan untuk menunjukkan input dan output kegiatan.

Sumber: Diadaptasi dari Object Management Group (2008)

Swimlanes digunakan untuk menggambarkan tanggung jawab fungsional yang berbeda-beda. Terdapat dua jenis *swimlanes* yaitu *pool* dan *lane*. Notasi dan penjelasan mengenai *swimlanes* dapat dilihat pada Tabel 2.3.

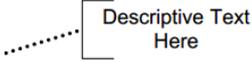
Tabel 2. 3 Notasi Swimlanes

Nama	Notasi	Deskripsi
<i>Pool</i>		Mewakili peserta dalam proses. Bertindak sebagai wadah mempartisi satu set kegiatan dari <i>pools</i> lainnya
<i>Lane</i>		subpartisi dalam <i>pool</i> dan akan memperpanjang seluruh panjang <i>pool</i> , baik secara vertikal maupun horizontal.

Sumber: Diadaptasi dari Object Management Group (2008)

Artifact digunakan untuk memberikan informasi tambahan mengenai proses. Terdapat tiga *artifact* standar dalam BPMN yaitu *data object*, *group*, dan *annotation*. Notasi dan penjelasan mengenai *artifact* dapat dilihat pada tabel 2.4.

Tabel 2. 4 Notasi Artifact

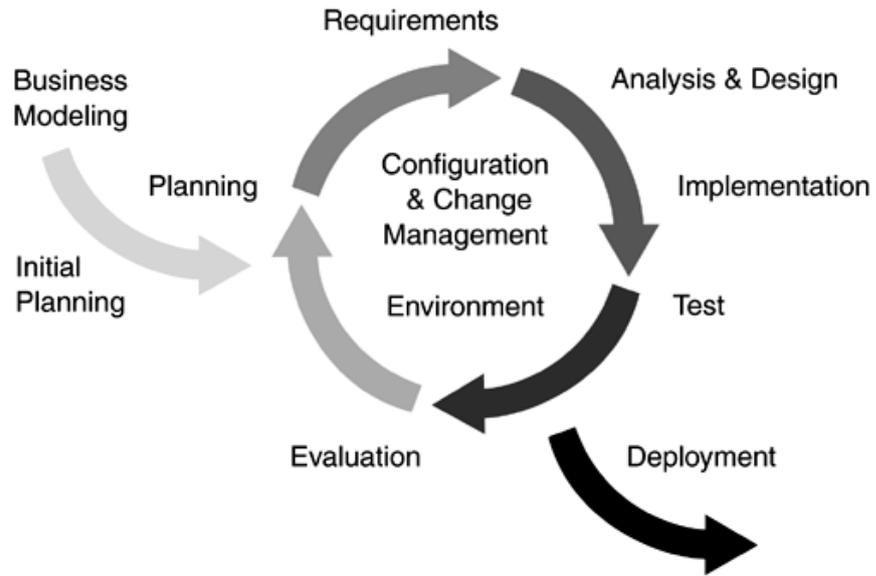
Nama	Notasi	Deskripsi
<i>Data Object</i>		Untuk menunjukkan bagaimana data diperlukan atau dihasilkan oleh kegiatan dan terhubung dengan aktivitas melalui Asosiasi.
<i>Group</i>		Pengelompokan ini dapat digunakan untuk tujuan dokumentasi atau analisis, tetapi tidak mempengaruhi Urutan Alur.
<i>Annotation</i>		Mekanisme bagi pemodel untuk menyediakan informasi teks tambahan bagi pembaca Diagram BPMN.

Sumber: Diadaptasi dari Object Management Group (2008)

2.3 Pengembangan Perangkat Lunak

2.3.1 Rational Unified Process (RUP)

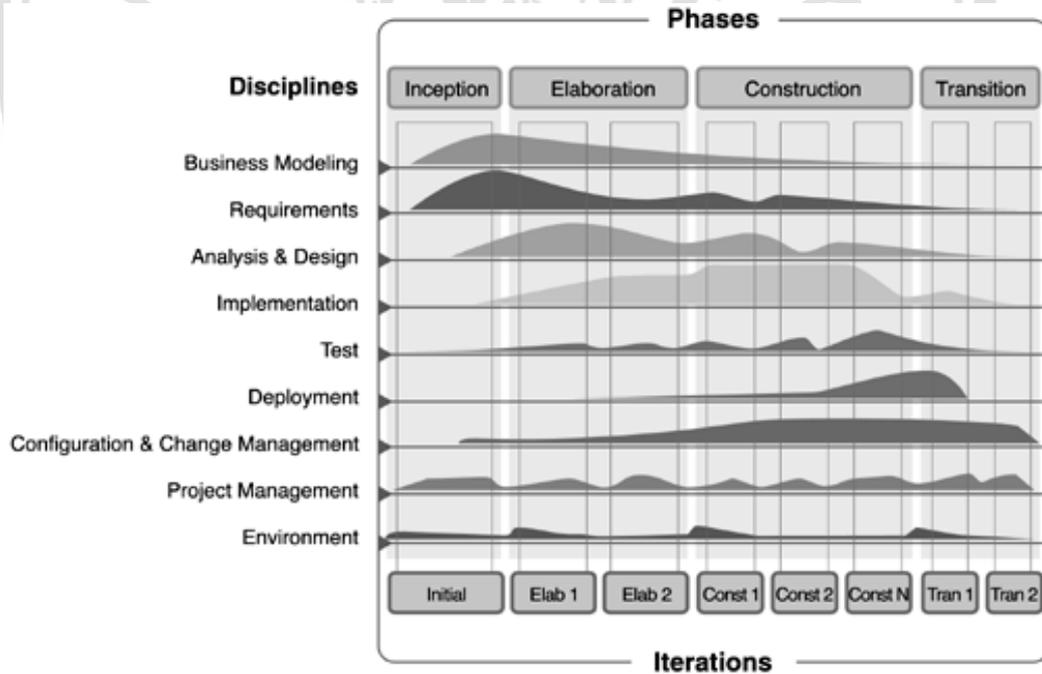
Rational Unified Process (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan secara berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), dan lebih diarahkan pada penggunaan kasus (*use case driven*) (Kroll & Kruchten, 2003). Dengan pendekatan pengembangan RUP, pengembangan sistem dapat dilakukan secara berulang-ulang apabila menemukan kesalahan pada saat pengujian. Iterasi juga dapat dilakukan pada setiap tahapan pada RUP. Iterasi dapat dilakukan apabila pada suatu tahapan RUP tidak memenuhi *milestone* (batas/capaian) yang telah ditetapkan. Proses perulangan pada RUP secara umum dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Proses iteratif RUP

Sumber: Kroll & Kruchten (2003)

RUP memiliki empat tahap atau fase yang dapat dilakukan secara iteratif. Keempat fase tersebut yaitu *Inception*, *Elaboration*, *Construction*, dan *Transition*. Fase pada RUP dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Fase pada RUP

Sumber: Kroll & Kruchten (2003)

2.3.1.1 Inception

Inception merupakan fase pertama dari keempat fase RUP. Tahap ini fokus pada memahami ruang lingkup dan tujuan proyek. Terdapat lima objektif pada fase ini. Penjelasan kelima objektif tersebut menurut Kroll & Kruchten (2003) yaitu:

1. Memahami tentang apa yang akan kita bangun, diantaranya yaitu menentukan visi, ruang lingkup sistem, batasan sistem, apa yang ada di dalam dan diluar sistem. Selain itu juga mengidentifikasi siapa yang menginginkan sistem ini serta apa yang bermanfaat bagi mereka.
2. Mengidentifikasi fungsionalitas kunci dari sistem. Memutuskan *use case* yang lebih diutamakan.
3. Mengidentifikasi setidaknya satu kandidat solusi.
4. Memahami biaya, jadwal, dan resiko berhubungan dengan proyek.
5. Menentukan proses apa yang harus diikuti serta *tools* yang akan digunakan.

menurut Kroll & Kruchten (2003) kebanyakan proyek hanya memiliki satu iterasi pada fase *inception*. Beberapa proyek mungkin akan memiliki lebih dari satu iterasi pada fase *inception*. Kroll dan Kruchten menambahkan jika pada suatu proyek terdapat lebih dari satu iterasi pada fase *inception*, maka iterasi pertama biasanya fokus utama pada objektif 1-3 (mengenai apa yang akan dibangun). Sedangkan iterasi lainnya lebih terfokus pada objektif 4-5 (mengenai bagaimana cara membangun). Terdapat beberapa alasan mengapa melakukan iterasi lebih dari satu kali pada fase *inception*. Beberapa diantaranya yaitu karena besarnya suatu proyek dan tim proyek kesulitan menentukan ruang lingkup sistem. Selain itu juga karena banyaknya pemangku kepentingan (*stake holder*) dan hubungan antara *stake holder* yang kompleks sehingga mengalami kesulitan saat melakukan negosiasi.

2.3.1.2 Elaboration

Tujuan dari fase elaborasi yaitu untuk mendefinisikan dan membuat dasar arsitektur sistem untuk desain yang akan diimplementasikan pada fase *construction* (Kroll & Kruchten, 2003). Kroll dan Kruchten menterjemahkan tujuan utama tadi ke empat tujuan pokok yang masing-masing menangani bidang risiko. Mengatasi risiko terkait persyaratan (apakah anda membangun aplikasi yang tepat?). Kemudian mengatasi risiko terkait arsitektur (apakah anda membangun solusi yang tepat?). Selain itu mengatasi risiko terkait biaya dan jadwal (apakah anda benar-benar di sesuai jalur?). Terakhir mengatasi risiko mengenai proses dan *tools* (apakah anda memiliki proses yang benar dan *tools* yang tepat untuk mengerjakan pekerjaan tersebut?).

Terdapat empat objektif pada fase *elaboration* menurut Kroll dan Kruchten yaitu:

1. Mendapatkan pemahaman yang lebih mendalam mengenai kebutuhan atau persyaratan yang nantinya akan membantu dalam membuat desain yang lebih mendetail.
2. Merancang, mengimplementasikan, dan menguji struktur kerangka sistem.
3. Mengurangi risiko, serta membuat estimasi jadwal dan biaya yang akurat.
4. Memperbaiki kasus pengembangan dan tempatkan lingkungan pengembangan.

Menurut Kroll dan Kruchten dalam fase Elaborasi, banyak risiko dikurangi dengan menghasilkan arsitektur yang dapat dieksekusi. Apabila sudah pernah membangun sistem dengan teknologi yang sama dengan teknologi yang digunakan pada proyek, maka biasanya hanya membutuhkan satu kali iterasi pada fase elaborasi. Tetapi jika tidak berpengalaman dalam domain aplikasi, sistemnya sangat kompleks, atau menggunakan teknologi baru, maka kemungkinan memerlukan dua atau tiga iterasi untuk mendapatkan arsitektur yang tepat dan untuk mengurangi risiko utama. Faktor lain yang menyebabkan diperlukannya beberapa iterasi yaitu pengembangan terdistribusi, memiliki banyak pemangku kepentingan atau perjanjian kontrak yang kompleks, atau perlu mematuhi peraturan keselamatan atau standar eksternal lainnya.

2.3.1.3 Construction

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Hasil perancangan pada tahap *elaboration* akan diimplementasikan pada tahap ini. Pengujian sistem yang fokus pada implementasi kode program juga dilakukan pada tahap ini. Sehingga kesalahan program dapat dideteksi lebih awal. Menurut Korll & Kruchten (2003) fase *construction* adalah mengenai pengembangan produk lengkap yang hemat biaya (versi operasional dari sistem) yang dapat diterapkan di lingkungan pengguna. Jumlah iterasi pada fase konstruksi biasanya lebih banyak daripada fase lain pada RUP. Korll dan Kruchten juga menambahkan mengenai dua objektif pada fase inception yaitu:

1. Meminimalisir biaya pengembangan dan mencapai beberapa derajat paralelisme dalam pekerjaan tim pengembangan dengan mengoptimalkan sumberdaya dan menghindari pengerjaan ulang.
2. Mengembangkan produk lengkap secara iteratif yang siap ditransisikan ke komunitas pengguna.

2.3.1.4 Transition

Tahap *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Aktifitas pada tahap ini termasuk pelatihan *user*, pemeliharaan dan pengujian sistem. Pada tahap ini akan dilakukan pengujian usabilitas (*Usability Testing*) apakah sistem bisa berguna dan dapat diterima oleh *user*. Menurut Korll &

Kruchten (2003) iterasi pada fase transisi bervariasi tergantung kompleksitas sistem. Pada sistem sederhana yang membutuhkan perbaikan *bug* minor, satu iterasi saja sudah cukup. Sedangkan sistem yang kompleks, seperti sistem kontrol lalu lintas udara, mungkin memerlukan beberapa iterasi untuk menambah fitur dan melakukan aktivitas *cutover* yang kompleks untuk mengalihkan bisnis dari menggunakan sistem lama ke sistem yang baru. Transisi dapat mencakup menjalankan sistem lama dan baru secara paralel, memigrasikan data, melatih pengguna, dan menyesuaikan proses bisnis.

2.3.2 Pemodelan Use Case

Untuk mencapai inti dari apa yang harus dilakukan sistem, pertama harus terlebih dahulu fokus pada siapa (atau apa) yang akan menggunakan sistem tersebut, serta apa saja yang akan digunakan oleh sistem (Bittner & Spence, 2003). Setelah itu baru melihat apa yang harus dilakukan sistem untuk para pengguna. *Use cases* merupakan cara untuk menggambarkan persyaratan atau kebutuhan sistem. Pemodelan *use case* dilakukan untuk mengetahui siapa saja pemangku kepentingan yang akan terkena dampak dari sebuah sistem, siapa saja yang akan menggunakan sistem, mengetahui kebutuhan pemangku kepentingan dan pengguna sistem, serta mengetahui kemungkinan solusi yang dapat ditawarkan kepada pemangku kepentingan dan pengguna untuk memenuhi kebutuhannya.

2.3.2.1 Identifikasi Tipe Pemangku Kepentingan

Dalam memodelkan *use case*, pertama yang dilakukan adalah melakukan identifikasi pemangku kepentingan dan bagaimana keterlibatannya. Pemangku kepentingan adalah individu yang terkena dampak secara material terkena dampak dari hasil sebuah sistem atau proyek memproduksi sistem (Bittner & Spence, 2003). Tipe pemangku kepentingan bisa saja dikembangkan bergantung pada jenis organisasi atau perusahaan dan sebagainya. Bittner dan Spence menjelaskan beberapa kelompok pemangku kepentingan yaitu:

1. *Users* (pengguna): Pengguna merupakan jenis pemangku kepentingan yang secara langsung menggunakan atau berinteraksi dengan sistem. Mereka adalah orang-orang yang akan mengambil peran yang ditentukan oleh para aktor dalam model *use case*.
2. *Sponsors* (sponsor): Sponsor merupakan orang yang berinvestasi dalam produksi sistem. Para pemangku kepentingan ini hanya pengguna tidak langsung dari sistem atau hanya dipengaruhi oleh hasil bisnis yang dipengaruhi oleh sistem.
3. *Developers* (Pengembang): Merupakan pemangku kepentingan yang terlibat dalam kegiatan produksi atau mendukung pengembangan sistem dalam sebuah proyek.
4. *Authorities* (pihak berwenang): Merupakan seorang ahli dalam aspek tertentu dari masalah atau solusi domain. Dalam hal ini seorang yang memiliki keahlian khusus dalam membantu pengembangan sistem seperti dalam bidang otoritas legislatif, organisasi standar, departemen tata kelola

organisasi, badan pengatur eksternal dan internal, pakar domain, dan pakar teknologi.

5. *Customers* (pelanggan): Orang-orang dan/atau organisasi yang akan membeli sistem final (sistem yang sudah jadi). Diantaranya yaitu pembeli, evaluator, akuntan, dan agen yang bertindak atas nama organisasi pembeli.

2.3.2.2 Analisis Permasalahan

Sebelum masuk ke spesifikasi sistem dan proses membuat sistem yang ditawarkan sebagai solusi dari permasalahan, sebaiknya melakukan analisis permasalahan untuk mengetahui masalah apa yang ingin dipecahkan dengan solusi tertentu. Hal ini akan membantu untuk memastikan bahwa semua fungsi yang disediakan oleh sistem secara langsung berkontribusi untuk mengatasi permasalahan. Analisis permasalahan dilakukan untuk mengetahui permasalahan yang dihadapi, siapa yang terkena dampak dari permasalahan, apa dampak dari permasalahan, serta solusi yang disarankan akan dikatakan sukses jika bagaimana. Menurut Bittner dan Spence (2003) cara terbaik untuk mengetahui permasalahan yaitu dengan membuat sebuah pernyataan masalah (*problem statement*). Pernyataan masalah dapat didokumentasikan sesuai *template* dokumentasi pada Tabel 2.5.

Tabel 2. 5 Template Dokumentasi Pernyataan Masalah

<i>The problem of</i>	Mendesripsikan permasalahan.
<i>Affects</i>	Siapa saja pemangku kepentingan (<i>stakeholder</i>) yang terkena dampak dari permasalahan.
<i>The impact of which is</i>	Apa saja dampak dari permasalahan.
<i>A successful solution would</i>	Daftar syarat atau kunci solusi yang ditawarkan akan dikatakan berhasil apabila memenuhi kriteria dalam daftar ini.

Sumber: Bittner & Spence (2003)

2.3.2.3 Identifikasi Kebutuhan Pemangku Kepentingan dan pengguna

Biasanya para pemangku kepentingan akan memiliki perspektif yang berbeda tentang masalah dan kebutuhan yang berbeda. Kebutuhan merupakan sesuatu yang menurut stakeholder (pemangku kepentingan) harus dilakukan oleh sistem seperti permasalahan yang perlu mereka atasi (Bittner & Spence, 2003). Bittner dan Spence juga mengatakan bahwa mengetahui kebutuhan pemangku kepentingan membantu untuk melengkapi dan memberikan pemahaman yang lebih dalam mengenai pernyataan permasalahan para pemangku kepentingan. Leffingwell dan Widrig menjelaskan bahwa untuk melakukan identifikasi kebutuhan pemangku kepentingan dan pengguna serta identifikasi fitur dapat dilakukan dengan teknik wawancara (*interviewing*), *requirements workshops*, *brainstorming and idea reduction*, serta *storyboarding*.

2.3.2.4 Identifikasi Fitur

Fitur merupakan layanan yang disediakan sistem untuk memenuhi satu atau lebih kebutuhan pemangku kepentingan (Leffingwell & Widrig, 2003). Leffingwell dan Widrig juga menambahkan bahwa fitur merupakan cara mudah untuk menggambarkan fungsionalitas dari suatu sistem tanpa harus terhambat dengan penjelasan mengenai fungsionalitas yang terlalu mendetail. Contoh dokumentasi tabel fitur dapat dilihat pada Tabel 2.6. Hal-hal yang perlu diperhatikan dalam mendokumentasikan fitur menurut Bittner dan Spence yaitu:

1. Sertakan deskripsi fungsionalitas dan masalah kegunaan yang relevan yang harus ditangani.
2. Hindari desain. Lakukan deskripsi fitur di level umum. Fokus pada kemampuan yang dibutuhkan dan mengapa (bukan bagaimana) mereka harus dilaksanakan.
3. Tetapkan pengenal unik pada setiap fitur (kode fitur) untuk memudahkan referensi dan pelacakan fitur.

Tabel 2. 6 Contoh Tabel Fitur

Kode Fitur	Deskripsi Fitur
Berisi pengenal unik (kode) sebagai identitas fitur	Berisi penjelasan mengenai fitur atau hal yang dapat dilakukan oleh sistem

Sumber: Diadaptasi dari Bittner & Spence (2003)

2.3.2.5 Identifikasi Persyaratan Fungsional dan Non Fungsional

Terdapat dua jenis persyaratan yaitu persyaratan fungsional dan persyaratan nonfungsional. Persyaratan fungsional mengungkapkan bagaimana sistem berperilaku seperti *input*, *output*, dan fungsi yang diberikannya kepada penggunanya (Leffingwell & Widrig, 2003). Bittner dan Spence (2003) mengatakan bahwa persyaratan fungsional merupakan tindakan yang harus dapat dilakukan oleh sistem, tanpa mempertimbangkan kendala fisik. Persyaratan fungsional menentukan perilaku *input* dan *output* dari suatu sistem. Sedangkan persyaratan nonfungsional merupakan kualitas atau batasan-batasan yang harus dipenuhi oleh sistem (Bittner & Spence, 2003). Menurut Leffingwell dan Widrig persyaratan nonfungsional muncul karena perlunya menspesifikasikan aspek-aspek lain pada sistem seperti *usability*, *reliability*, *performance*, dan *supportability*.

2.3.2.6 Pemodelan Use Case

Terkadang hal yang paling sulit untuk digambarkan adalah bagaimana sistem akan bekerja. Pemodelan *use case* memberi cara menggambarkan bagaimana pengguna sistem (baik orang ataupun sistem lain) dan sistem akan berinteraksi (Bittner & Spence, 2003). Dengan pemodelan *use case* juga dapat melihat bagaimana sistem akan bereaksi ketika pengguna melakukan sesuatu. *Use case* menggambarkan apa yang dilakukan pengguna untuk memulai beberapa perilaku dalam sistem, dan apa yang sistem lakukan untuk memenuhi kebutuhan

pengguna. model *use case* dengan jelas mendefinisikan tanggung jawab sistem berkaitan dengan memuaskan tujuan pengguna, dan tanggung jawab pengguna untuk mendukung sistem.

Model *use case* berisi dua komponen utama yaitu aktor dan *use case*. Aktor merupakan manusia atau hal lain (contohnya sistem lain) yang berinteraksi dengan sistem. Aktor berada diluar sistem dan memiliki nama dan deskripsi singkat. Sedangkan *use case* mewakili hal-hal yang dilakukan sistem untuk para aktornya. *Use case* memiliki nama dan deskripsi singkat. *Use case* juga memiliki deskripsi rinci yang merupakan cerita atau alur tentang bagaimana para aktor menggunakan sistem untuk melakukan sesuatu (yang aktor butuhkan), dan apa yang sistem lakukan untuk memenuhi kebutuhan aktor. Model *use case* akan didokumentasikan kedalam dokumentasi spesifikasi *use case*. Dokumentasi spesifikasi *use case* dapat dilihat pada Tabel 2.7.

Tabel 2. 7 Format Dokumentasi Spesifikasi Use Case

<i>Name</i>	Nama <i>use case</i>
<i>Brief Description</i>	Deskripsi singkat tentang peran dan tujuan dari <i>use case</i> .
<i>Actor</i>	Aktor yang berinteraksi dengan <i>use case</i> .
<i>Pre-condition</i>	Syarat atau kondisi yang harus dipenuhi sebelum <i>use case</i> dapat dimulai.
<i>Post-condition</i>	kondisi yang dipenuhi pada saat <i>use case</i> berakhir atau selesai.
<i>Basic flow</i>	Mencakup apa yang "biasanya" terjadi ketika <i>use case</i> dilakukan. Alur normal yang dilalui pada saat <i>use case</i> dijalankan.
<i>Alternative flow</i>	<i>Alternative flow</i> meliputi alur yang bersifat opsional dan akan dijalankan pada kondisi tertentu ketika <i>use case</i> tidak berjalan sesuai alur normal (<i>basic flow</i>).
<i>Sub flow</i>	Flow yang rumit dapat dibagi lagi menjadi subflows. Penyederhanaan dari flow yang terlalu kompleks.

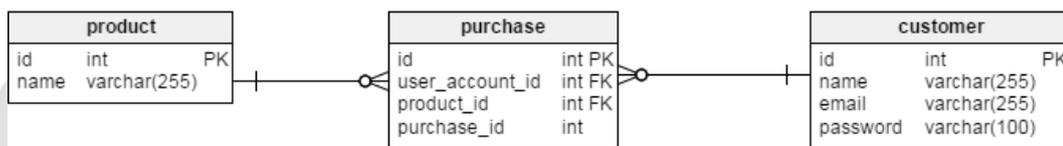
Sumber: Diadaptasi dari Bittner & Spence (2003)

2.3.3 Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan (Elmasri & Navathe, 2011). Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data baik berupa *file* teks ataupun *Database Management System* (DBMS). DBMS merupakan kumpulan program yang memungkinkan pengguna untuk membuat dan memelihara sebuah *database* (Elmasri & Navathe, 2011). Tujuan dari DBMS yaitu untuk memfasilitasi proses mendefinisikan, membangun, memanipulasi, dan berbagi database antara berbagai pengguna dan aplikasi. Hampir semua DBMS menggunakan SQL sebagai bahasa pengelola data.

2.3.3.1 Physical Data Model

Physical Data Model (PDM) digunakan untuk memodelkan atau menggambarkan objek data relasional (misalnya, tabel, kolom, *primary key* dan *foreign key*) serta hubungan diantara mereka. PDM menerangkan detail dari bagaimana data disimpan dalam *database*. PDM merupakan bentuk fisik perancangan basis data yang siap diimplementasikan ke dalam DBMS. Oleh karena itu nama tabel dalam PDM merupakan nama tabel asli yang akan diimplementasikan ke dalam DBMS. Setiap tabel pada PDM memiliki kolom dan masing-masing kolom memiliki nama yang unik. Masing-masing kolom pada tabel memiliki tipe data. Pada PDM terdapat kolom atau atribut yang merupakan *primary key* atau *foreign key*. *Primary key* adalah kolom atau kumpulan kolom yang berisi nilai-nilai yang secara unik mengidentifikasi setiap baris dalam sebuah tabel. Sedangkan *foreign key* merupakan kolom yang mengacu pada sebuah nilai yang unik pada tabel lain (table refrensi). Contoh dari *physical data model* dapat dilihat pada Gambar 2.4. *Primary key* ditandai dengan kode "PK" dan *foreign key* dengan kode "FK".



Gambar 2. 4 Physical Data Model

Sumber: Kozubek (2013)

2.3.4 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya sebagai suatu kumpulan objek (Rosa & Shalahuddin, 2016). Pada pengembangan berorientasi objek terdapat *Object Oriented Analysis* (OOA) dan *Object Oriented Design* (OOD) serta *Object Oriented Programming* (OOP). OOA adalah metode analisis yang memeriksa persyaratan (*requirement*) dari perspektif kelas dan objek yang ditemukan dalam domain masalah (Booch, 1994). Booch juga mengatakan bahwa OOD adalah metode desain yang meliputi proses dekomposisi berorientasi objek dan notasi untuk menggambarkan sistem yang sedang dirancang. Sedangkan OOP adalah metode implementasi di mana program disusun sebagai kumpulan objek kooperatif, yang masing-masing mewakili sebuah *instance* dari beberapa kelas, dan kelas-kelasnya adalah semua anggota dari hirarki kelas yang disatukan melalui relasi pewarisan. Hasil dari OOA akan digunakan sebagai model untuk memulai OOD. Sedangkan hasil OOD akan digunakan sebagai *blueprint* untuk menerapkan sistem dengan metode OOP.

2.3.5 Unified Modeling Language

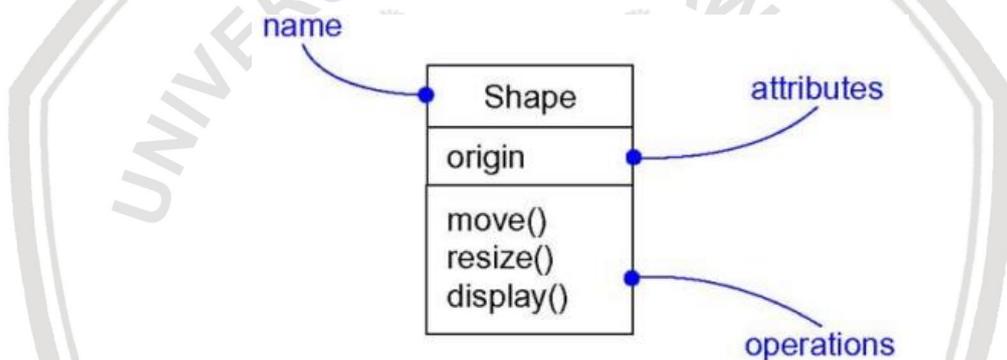
Pemodelan sistem *object-oriented* biasanya menggunakan beberapa diagram UML (*Unified Modeling Language*). UML merupakan bahasa standar dalam

menulis *blueprints* perangkat lunak yang digunakan untuk memvisualisasikan, menentukan, membuat, dan mendokumentasikan artefak dari sistem perangkat lunak (Booch, Rumbaugh & Jacobson, 2005). Contoh dari diagram UML yaitu *Use case Diagram*, *class diagram*, dan *sequence diagram*. Berikut penjelasan mengenai *use case diagram* dan *class diagram* menurut Booch, Rumbaugh & Jacobson (2005) serta *sequence diagram* menurut Dennis, Wixom & Tegarden (2009).

2.3.5.1 Class Diagram

Class diagram merupakan diagram UML yang paling sering digunakan dalam pemodelan sistem berorientasi objek. *Class diagram* menunjukkan sekumpulan kelas, antarmuka, dan kolaborasi serta hubungan diantara mereka. Contoh *class diagram* dapat dilihat pada Gambar 2.6. *Class diagram* biasanya berisi *class*, *interface*, *collaboration*, *relationship (dependency, generalization, dan association)*.

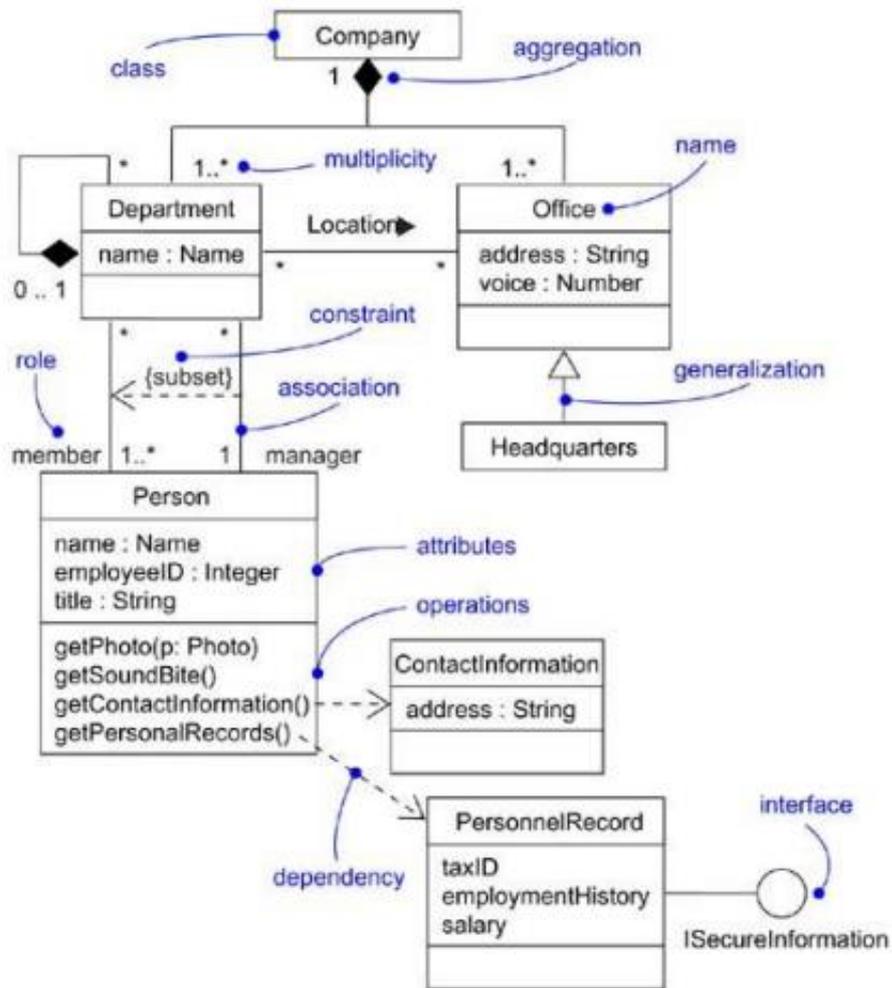
1. *Class* adalah deskripsi dari sekumpulan objek yang memiliki atribut, operasi, hubungan, dan semantik yang sama. Contoh *class* dapat dilihat pada gambar 2.5 yang memiliki nama kelas, atribut dan operasi.



Gambar 2. 5 Class

Sumber : Booch, Rumbaugh & Jacobson (2005)

2. *Interface* merupakan kumpulan operasi yang digunakan untuk menspesifikkan layanan dari sebuah kelas atau komponen.
3. *Colaboration* merupakan society dari kelas-kelas, antarmuka-antarmuka, dan element lainnya yang bekerja sama untuk menyediakan perilaku kooperatif yang lebih besar.
4. *Dependency* adalah “hubungan menggunakan” yang menyatakan bahwa perubahan dalam spesifikasi satu hal (misalnya, *class*) dapat mempengaruhi hal lain yang menggunakannya (misalnya, kelas Window), namun tidak harus sebaliknya. Dengan kata lain dapat bermakna ketergantungan antar kelas
5. *Generalization* adalah hubungan antara hal yang umum (disebut superclass atau orang tua) dan jenis yang lebih spesifik dari hal itu (disebut subkelas atau anak).
6. *Association* adalah hubungan struktural yang menentukan bahwa objek dari satu benda terhubung ke objek benda lain.

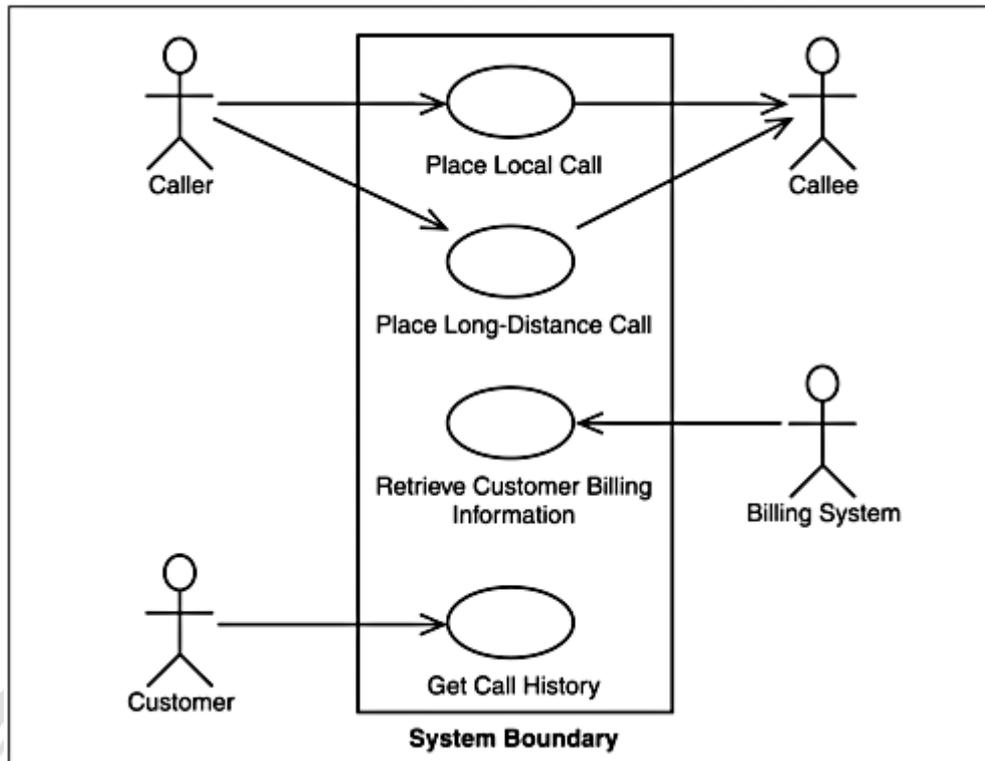


Gambar 2. 6 Class Diagram

Sumber: Booch, Rumbaugh & Jacobson (2005)

2.3.5.2 Use case Diagram

Use case diagram berfungsi untuk menggambarkan perilaku suatu sistem, subsistem, atau kelas yang menunjukkan hubungan antara aktor dengan sistem. *Use case* juga dapat digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem, dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Penamaan dari sebuah *use case* didefinisikan sesimpel mungkin dan dapat dipahami. Gambar 2.7 merupakan contoh dari *use case diagram*. Pada contoh diagram tersebut terdapat aktor yang digambarkan dengan simbol orang, *use case* digambarkan dengan simbol oval dan terdapat sebuah panah yang menghubungkan antara aktor dengan *use case* atau sebaliknya. Arah anak panah menunjukkan siapa yang memulai *case*. Aktor berada diluar sistem, maka dari itu pada Gambar 2.7 aktor berada di luar *system boundary*.

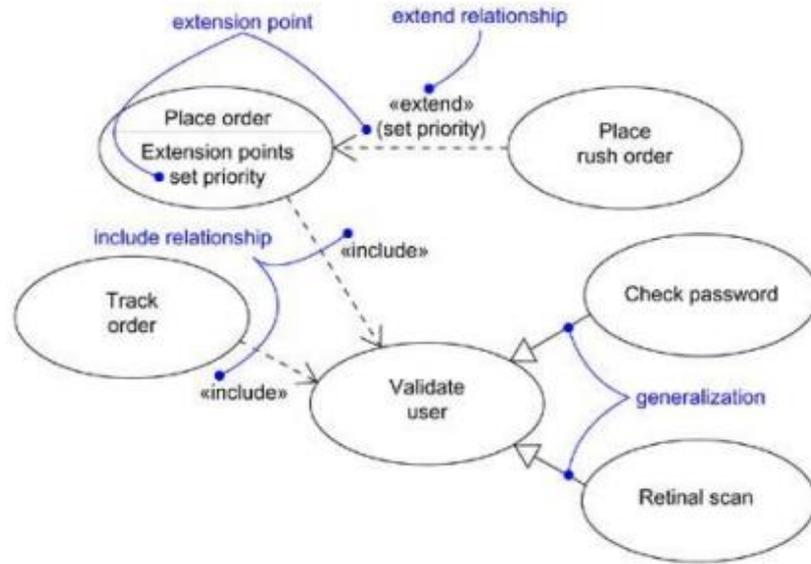


Gambar 2. 7 Contoh Use Case Diagram

Sumber: Bittner & Spence (2003)

Beikut beberapa keterangan mengenai notasi yang ada pada *use case diagram* menurut Booch, Rumbaugh & Jacobson (2005).

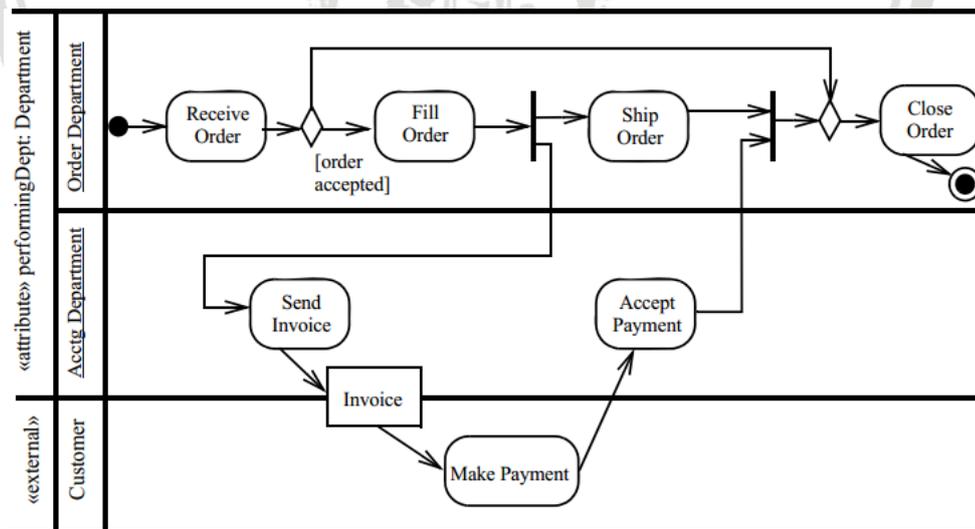
1. *Use case* adalah deskripsi sekumpulan urutan tindakan, termasuk varian, yang dilakukan sistem untuk menghasilkan hasil nilai yang dapat diamati kepada aktor.
2. *Actor* mewakili seperangkat peran yang dimainkan. Biasanya aktor merepresentasikan peran dari manusia, perangkat keras, atau sistem lain yang berinteraksi dengan sistem.
3. *Association* adalah hubungan antara aktor dan *use case*.
4. *Generalization* adalah hubungan umum-khusus antara dua *use case*.
5. *Include* merupakan relasi *use case* tambahan ke *use case* lain. Dimana *use case* tersebut memerlukan *use case* yang ditambahkan untuk menjalankan fungsinya.
6. *Extend* merupakan relasi *use case* tambahan ke *use case* lain. Dimana *use case* tersebut dapat berdiri sendiri walau tanpa *use case* yang ditambahkan. Notasi *generalization*, *include* dan *extend* pada *use case diagram* dapat dilihat pada Gambar 2.8.



Gambar 2. 8 Generalization, include, dan extend
 Sumber: Booch, Rumbaugh & Jacobson (2005)

2.3.5.3 Activity Diagram

Activity diagram merupakan diagram yang menunjukkan aliran kontrol dan data dari aktivitas ke aktivitas atau workflow (aliran kerja) dari sebuah sistem (Booch, Rumbaugh & Jacobson, 2005). Booch, et al menambahkan bahwa *activity diagram* digunakan untuk menggambarkan aspek dinamis dari sebuah sistem. Rosa dan Shalahuddin mengatakan bahwa *activity diagram* menggambarkan aktivitas sistem, bukan apa yang dilakukan oleh aktor. Contoh *activity diagram* dapat dilihat pada Gambar 2.9.

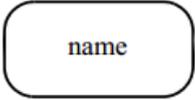
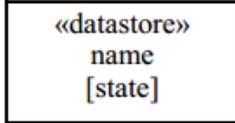
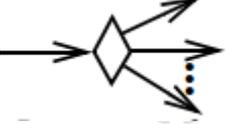
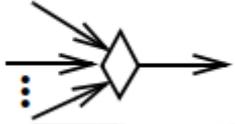
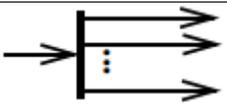


Gambar 2. 9 Contoh Activity Diagram
 Sumber: Object Management Group (2005)

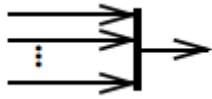


Activity diagram memiliki beberapa notasi seperti *action*, notasi *activity*, notasi *node*, dan sebagainya. Penjelasan masing-masing notasi pada *activity diagram* dapat dilihat pada Tabel 2.8.

Tabel 2. 8 Notasi pada Activity Diagram

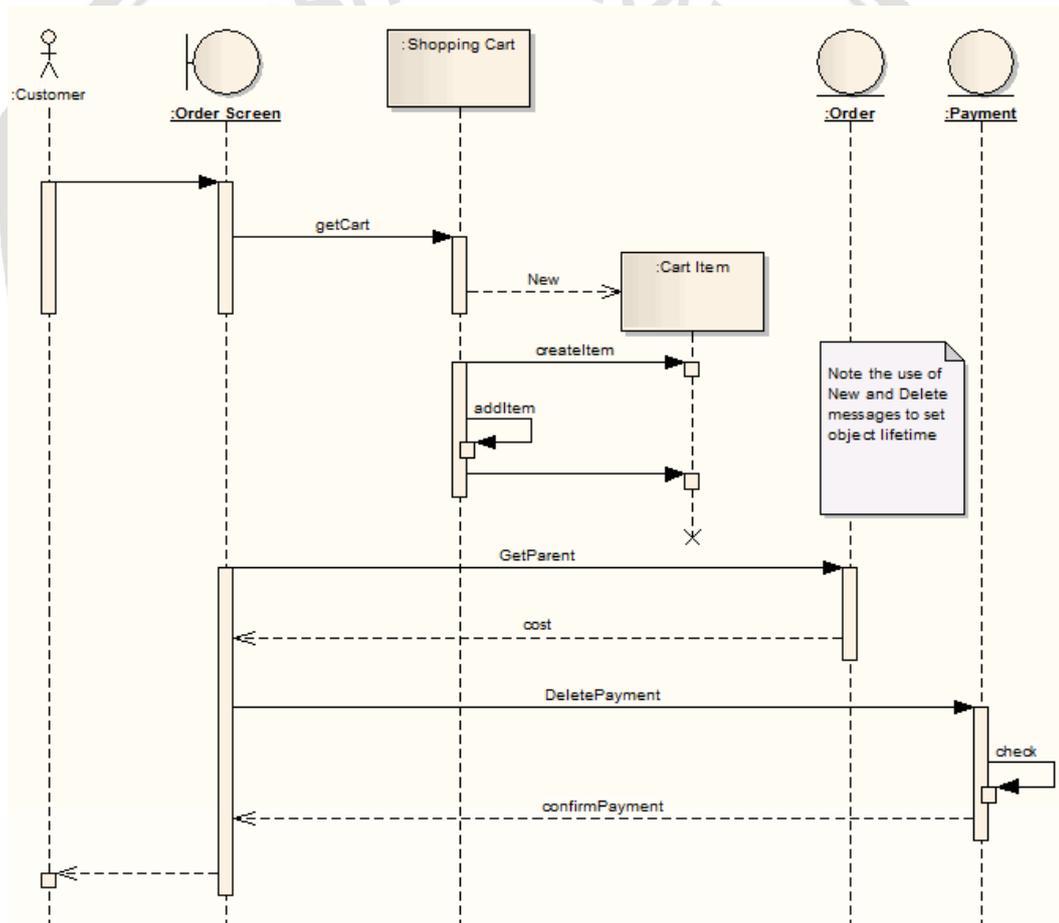
Nama	Notasi	Deskripsi
<i>Initial node</i>		Awal aliran mulai dalam suatu aktivitas.
<i>Action</i>		Merepresentasikan satu langkah dalam suatu aktivitas, atau menggambarkan suatu aktivitas dalam sistem atau suatu proses.
<i>Activity edge</i>		Menghubungkan atau mengkoneksikan notasi-notasi dalam <i>activity diagram</i> .
<i>Activity edge connector</i>		Menghubungkan <i>activity edge</i> . Setiap konektor dengan label sama akan dipasangkan. Salah satu dari konektor yang berpasangan harus memiliki satu <i>outcoming edge</i> dan pasangannya harus memiliki satu <i>incoming edge</i> .
<i>Activity final</i>		Memodelkan penghentian semua aliran dalam suatu aktivitas. Titik akhir aliran dalam aktivitas.
<i>Data Store</i>		Mendukung bentuk-bentuk pemodelan arus data.
<i>Decision node</i>		Mendukung alur kondisional dalam aktivitas. <i>Decision node</i> harus memiliki satu <i>incoming edge</i> , dan dua atau lebih <i>outcoming edge</i> .
<i>Merge node</i>		Menerima atau menggabungkan beberapa alur. <i>Merge node</i> harus memiliki dua atau lebih <i>incoming node</i> dan satu <i>outcoming node</i> .
<i>Fork node</i>		Mendukung paralelisme dalam aktivitas yaitu dengan membagi jalur menjadi beberapa jalur yang paralel (dapat berjalan bersamaan). <i>Fork node</i> memiliki satu <i>incoming edge</i> dan dua atau lebih <i>outcoming edge</i> .

Tabel 2.8 Notasi pada *Activity Diagram* (Lanjutan)

Nama	Notasi	Deskripsi
<i>Join node</i>		Mendukung paralelisme dalam aktivitas yaitu dengan menggabungkan beberapa jalur paralel (dapat berjalan bersamaan) menjadi satu jalur. <i>Fork node</i> memiliki dua atau lebih <i>incoming edge</i> dan satu <i>outcoming edge</i> .
<i>Activity Partition</i>		Untuk mempartisi atau mengelompokkan <i>action</i> berdasarkan domain spesifik (seperti unit sistem atau entitas) yang menjalankan action tersebut.

Sumber: Diadaptasi dari Object Management Group (2005)

2.3.5.4 Sequence Diagram

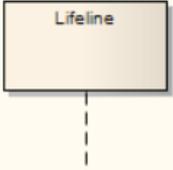
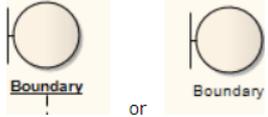
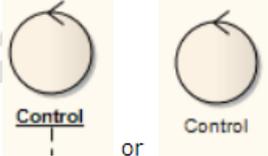
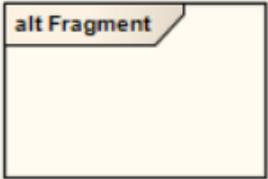


Gambar 2. 10 Sequence Diagram

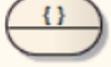
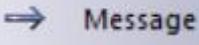
Sumber: Sparxsystems

Gambar 2.10 merupakan contoh *sequence diagram*. *Sequence diagram* merupakan salah satu jenis diagram interaksi yang menggambarkan objek yang berpartisipasi dalam *use case diagram* dan pesan yang melewatinya dari waktu ke waktu untuk satu *use case* (Dennis, Wixom, & Tegarden, 2009). *Sequence diagram* adalah model dinamis yang menunjukkan urutan pesan yang dilewatkan antar objek. *Sequence diagram* menekankan urutan aktivitas berbasis waktu yang terjadi di antara seperangkat objek. Dalam membuat *sequence diagram* bisa dengan melihat skenario yang ada pada *use case diagram*. Penjelasan mengenai notasi-notasi pada *sequence diagram* dapat dilihat pada tabel 2.9.

Tabel 2. 9 Notasi pada Sequence Diagram

Term and Definition	Symbol
<p>Actor:</p> <p>Merupakan pengguna sistem, bisa berupa manusia, mesin, sistem atau subsistem lain. Sesuatu yang berinteraksi dengan sistem disebut sebagai aktor. Aktor juga mewakili peran pengguna dalam <i>Sequence Diagrams</i>.</p>	
<p>Entity:</p> <p>Entitas adalah Objek stereotip yang memodelkan mekanisme yang menangkap informasi dalam suatu sistem.</p>	
<p>Lifeline:</p> <p><i>Lifeline</i> menunjukkan kehidupan objek selama suatu urutan. <i>Lifeline</i> mewakili <i>individual participant</i> dalam Interaksi.</p>	
<p>Boundary:</p> <p>Object stereotip yang memodelkan beberapa batasan sistem.</p>	
<p>Control:</p> <p>Merupakan objek stereotip yang memodelkan entitas pengendali atau manajer. Kontrol mengatur dan menjadwalkan aktivitas dan elemen lain.</p>	
<p>Combined Fragment:</p> <p>Fragmen gabungan mendefinisikan ekspresi fragmen interaksi. Fragmen gabungan didefinisikan oleh operator interaksi dan operan interaksi yang sesuai.</p>	

Tabel 2.9 Notasi pada Sequence Diagram (Lanjutan)

Term and Definition	Symbol
<p>Endpoint:</p> <p><i>Endpoint</i> digunakan untuk mencerminkan pesan yang hilang atau ditemukan secara berurutan. Pesan hilang adalah pesan dimana kejadian pengiriman diketahui, namun tidak ada kejadian penerimaan. Pesan ditemukan adalah pesan dimana kejadian penerima diketahui, namun tidak ada kejadian pengiriman yang diketahui.</p>	 <p>EndPoint</p>
<p>Diagram Gate:</p> <p><i>Gate</i> adalah titik koneksi untuk menghubungkan pesan di luar fragmen interaksi dengan pesan di dalam fragmen interaksi.</p>	 <p>Diagram Gate</p>
<p>State/Continuation:</p> <p><i>State/Continuation</i> menyediakan dua tujuan berbeda pada diagram interaksi.</p>	 <p>State/Continuation</p>
<p>Message:</p> <p>Pesan menunjukkan adanya arus informasi atau transisi kontrol antar elemen. Pesan mendefinisikan komunikasi antara Lifelines pada sebuah interaksi.</p>	 <p>Message</p>

Sumber: Diadaptasi dari sparxsystems

2.3.6 Pengujian

Terdapat dua teknik pengujian perangkat lunak yaitu *black box testing* dan *white box testing*. *White box testing* juga disebut pengujian struktural karena mempertimbangkan struktur (hirarki komponen, aliran kontrol, arus data) dari *test case* (Linz, Schaefer & Spillner, 2014). Sedangkan *Black box testing* juga disebut teknik pengujian fungsional, spesifikasi, atau perilaku karena pengamatan perilaku input / output merupakan fokus utama (Beizer, 1995 disitasi dalam Linz, Schaefer & Spillner, 2014). Pada pengujian *black box testing* yang diuji hanyalah fungsionalitas dari sistem tanpa mempedulikan *code* program. Pengujian dilakukan untuk melihat apakah fungsi-fungsi dari sistem yang dibangun sudah sesuai dengan perancangan sistem. Beberapa contoh *black box testing* yaitu *validation testing*, *compatibility testing* dan *usability testing*.

2.3.6.1 Validation Testing

Validation testing merupakan pengujian yang berfokus terhadap aksi dan keluaran dari sistem yang mampu dilihat oleh pengguna (Pressman, 2010). Pengujian sistem dengan serangkaian kegiatan untuk menunjukkan bahwa sistem yang dibangun sudah memenuhi persyaratan. Pengujian dapat dilakukan sesuai

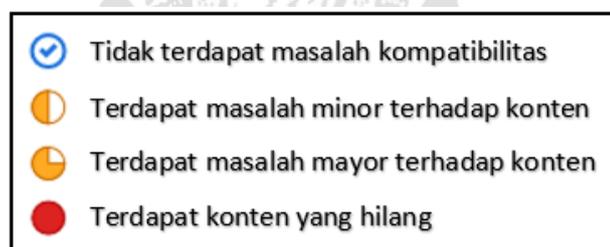


use case yang telah dimodelkan sesuai persyaratan. Masing- masing *use case* memiliki skenario yang memiliki kemungkinan untuk menemukan kesalahan sistem yang tidak sesuai dengan persyaratan. Alur *use case* dapat dijadikan panduan dalam melakukan pengujian validasi. Validasi dicapai melalui serangkaian tes yang menunjukkan kesesuaian sistem dengan persyaratan. Validation testing merupakan salah satu metode pengujian *black box testing* yaitu pengujian yang terfokus pada persyaratan fungsional sistem.

2.3.6.2 Compatibility Testing

Compatibility testing dilakukan untuk menguji apakah sistem yang dibangun dapat berjalan dengan baik di berbagai lingkungan yang berbeda. Komputer, sistem operasi, *browser*, serta kecepatan koneksi jaringan yang berbeda memiliki pengaruh yang signifikan terhadap berjalannya aplikasi berbasis web (Pressman, 2010). Pengujian kompatibilitas yang dilakukan yaitu kompatibilitas terhadap browser yang berbeda. Browser yang berbeda terkadang menghasilkan hasil atau tampilan yang berbeda. Salah satu contoh perbedaan yang dapat dilihat yaitu tata letak konten yang sedikit berbeda.

Pengujian kompatibilitas dilakukan menggunakan bantuan *sortSite*. *sortSite* dapat digunakan untuk mengecek format HTML, skrip, dan gambar yang tidak berfungsi di beberapa browser yang sering digunakan. Beberapa *browser* yang dapat diuji menggunakan *sortSite* yaitu IE, Edge, Firefox, Safari, Opera, Chrome, IOS, dan Android. Hasil pengujian menggunakan *sortSite* dikelompokkan kedalam empat kategori penilaian yang digambarkan pada Gambar 2.11.



Gambar 2. 11 Kategori Penilaian Hasil Pengujian Kompatibilitas

2.3.6.3 Usability Testing

Usability adalah atribut kualitas yang menilai seberapa mudah antarmuka pengguna digunakan (Nielsen, 2012). Nielsen juga menjelaskan bahwa kata "*Usability*" (Kegunaan) mengacu pada metode untuk meningkatkan kemudahan (*ease-of-use*) penggunaan selama proses perancangan. *Usability* didefinisikan oleh 5 komponen kualitas yaitu:

1. **Learnability:** Seberapa mudah bagi pengguna untuk menyelesaikan tugas dasar saat pertama kali mereka menggunakan sistem?
2. **Efficiency:** Begitu pengguna telah mempelajari sistem, seberapa cepat mereka bisa melakukan tugas?

3. **Memorability:** Ketika pengguna menggunakan kembali setelah lama tidak menggunakannya, seberapa mudah mereka kembali menggunakannya dengan baik?
4. **Errors:** Berapa banyak kesalahan yang dilakukan pengguna, seberapa parah kesalahan tersebut, dan seberapa mudah mereka pulih dari kesalahan?
5. **Satisfaction:** Seberapa menyenangkan menggunakan sistem tersebut?

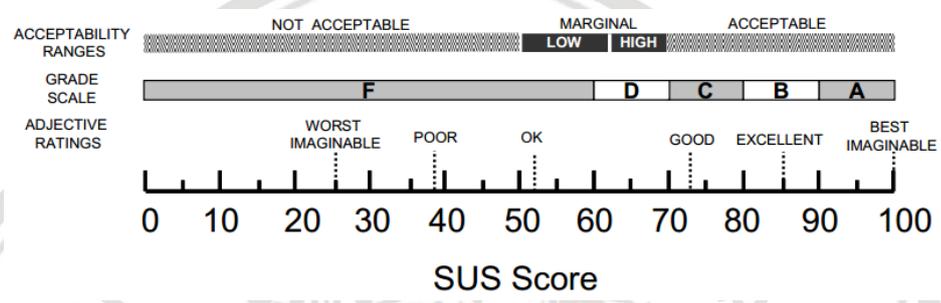
Usability dapat diukur menggunakan *System Usability Scale (SUS)* yang berupa skala Likert. Jonh Brooke mengembangkan *form* pernyataan SUS pada tahun 1986. *Form* pernyataan tersebut terdiri dari 10 pernyataan. Masing-masing pernyataan memiliki lima opsi jawaban yang dimulai dari skala 1 (sangat tidak setuju) hingga skala 5 (sangat setuju). Dari 10 pernyataan terdapat lima pernyataan positif (1, 3, 5, 7, 9) dan lima pernyataan negatif (2, 4, 6, 8, 10). *Form* pernyataan dapat dilihat pada Gambar 2.12.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Gambar 2. 12 System Usability Scale

Sumber: Brooke (1986)

Pertanyaan positif (1, 3, 5, 7, 9) kontribusi skornya dihitung dengan mengurangi poin dari responden dengan nilai 1 (-1). Sedangkan pertanyaan negatif (2, 4, 6, 8, 10) kontribusi skornya dihitung dengan mengurangi poin dari responden dengan nilai 5 (-5). Nilai *system usability* (SU) diperoleh dengan menjumlahkan skor yang diperoleh tadi kemudian dikalikan dengan 2,5. Sehingga nilai awal yang berkisar antara 0 hingga 40 berubah menjadi 0 hingga 100. Nilai *system usability* yang diperoleh akan dibandingkan dengan *range* nilai *grade* skor SUS. Berikut interpretasi *grade* dari skor SUS oleh Bangor, Kortum, & Miller (2009) dapat dilihat pada Gambar 2.13. Pada interpretasi *grade* dari skor SUS terdapat 5 *grade* yaitu F, D, C, B dan A. nilai 70 merupakan nilai rata-rata, sehingga *software* dikatakan baik apabila mendapatkan nilai lebih dari atau sama dengan 70 (grade C, B atau A).



Gambar 2. 13 Interpretasi grade skor SUS
 Sumber : Bangor, Kortum, & Miller (2009)

2.3.6.4 Test Case

Test case merupakan sekumpulan instruksi yang dirancang untuk menemukan jenis kesalahan atau cacat tertentu pada sistem perangkat lunak dengan menginduksi kegagalan (Agarwal, Tayal, Gupta, 2010). Setiap *test case* membutuhkan dokumentasi yang tepat, sebaiknya dalam format yang tetap. Terdapat banyak format *test case*. Gambar 2.14 merupakan contoh format pendokumentasian *test case*.

Test case name	Test Case ID
Purpose of test	Testing object (unit, application, module, etc.)
Test attribute	
Tests focus (function, feature, process, interface, validation, verification, etc.)	
Test type (alpha, beta, unit, integration, system)	
Test process	A set of instructions for conducting the test-initial stating condition-inputs-specifications-output expected
Test results	Expected and actual and comparison, error description, post-process state
Action	Correction, authorization, and feedback through retest
Action to initialize the pre-test status	

Gambar 2. 14 Template test case
 Sumber : Agarwal, Tayal, Gupta (2010)



2.4 Teknologi Pengembangan Sistem

2.4.1 PHP

PHP (*Hypertext preprocessor*) merupakan salah satu bahasa pemrograman web yang populer. PHP dapat disematkan kedalam HTML (*Hypertext Markup Language*). Kode PHP disertakan dalam instruksi pemrosesan awal dan akhir “<? Php” dan “?” yang mempermudah untuk masuk dan keluar dari "mode PHP". Semua yang ada di luar sepasang tag pembuka “<?php” dan penutup “?” akan diabaikan oleh PHP. Untuk memisahkan antara *statement* satu dengan yang lain pada PHP, setiap akhir *statement* diakhiri dengan tanda “;”. PHP dapat digunakan pada berbagai *Operating System* seperti *Linux, Unix, Windows, Mac OS* dan sebagainya.

2.4.2 HTML

HTML (*Hyper Text Markup Language*) adalah bahasa di mana sebagian besar situs web ditulis. *Hypertext* berarti dokumen berisi *links* yang mengijinkan pembaca untuk melompat dari satu tempat dalam dokumen ke dokumen lainnya. Sedangkan *Markup Language* merupakan cara bagaimana komputer-komputer saling berkomunikasi untuk mengatur bagaimana *text* diproses dan disajikan (Quality Nonsense Ltd). HTML digunakan untuk membuat halaman dan membuatnya berfungsi. Kode yang digunakan untuk membuat halaman yang menarik secara visual dikenal sebagai CSS. HTML dikembangkan oleh Tim Berners-Lee pada tahun 1990.

2.4.3 SQL

Structure Query Language (SQL) merupakan bahasa yang dapat digunakan untuk mengelola data pada DBMS (Rosa & Shalahuddin, 2016). Hampir sebagian DBMS yang ada menggunakan SQL sebagai bahasa pengolahan data mereka. SQL mulai berkembang pada tahun 1970an. SQL mulai digunakan sebagai standar resmi oleh ANSI (*American National Standards Institute*) pada tahun 1986 dan oleh ISO (*International Organization of Standardization*) pada tahun 1987.

2.4.4 MySQL

MySQL merupakan sebuah *database management system* (DBMS) yang berfungsi untuk menambah, mengakses, dan mengolah data yang tersimpan dalam database komputer, diperlukan sistem manajemen basis data seperti MySQL Server. MySQL merupakan *relational* DBMS. Sebuah *relational database* menyimpan data dalam tabel yang terpisah. MySQL merupakan *software* yang bersifat *open source*. Oleh karenanya MySQL dapat digunakan secara bebas (gratis).

2.4.5 CodeIgniter (CI)

CodeIgniter adalah *framework* untuk membangun *website* menggunakan PHP. Tujuannya adalah untuk memungkinkan proses pengembangan *website* yang lebih cepat daripada menulis kode dari nol, dengan menyediakan seperangkat *library*

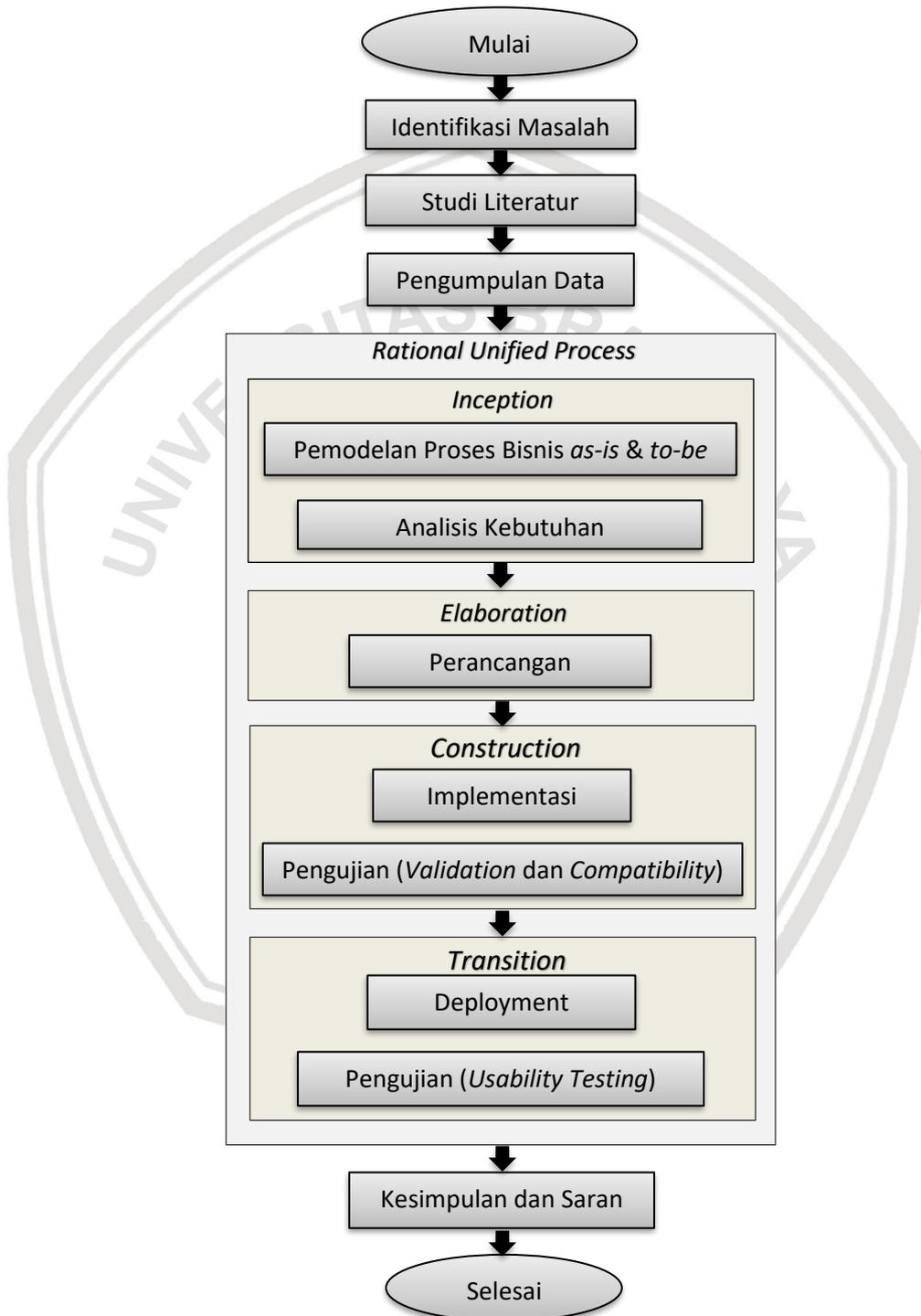
yang dapat digunakan pada proses pengembangan. *CodeIgniter* didasarkan pada pola pengembangan *Model-View-Controller* (MVC). MVC adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi. Berikut masing-masing pengertian *model*, *view*, dan *controller*:

1. *Model* mewakili struktur data. Biasanya kelas model akan berisi fungsi yang membantu dalam mengambil, memasukkan, dan memperbarui informasi di *database*.
2. *View* adalah informasi yang disajikan kepada pengguna. *view* biasanya berupa halaman web, tetapi di *CodeIgniter* tampilan juga bisa berupa fragmen halaman seperti *header* atau *footer*.
3. *Controller* berfungsi sebagai perantara antara *model*, *view*, dan sumber daya lain yang diperlukan untuk memproses permintaan HTTP dan menghasilkan halaman web.



BAB 3 METODOLOGI

Penelitian ini merupakan penelitian yang bertipe implementatif pengembangan. Pada bab ini penulis akan membahas langkah-langkah yang dilakukan dalam penelitian. Alur penelitian yang dilakukan tertera dalam gambar 3.1.



Gambar 3. 1 Metodologi Penelitian

Alur penelitian dimulai dari melakukan identifikasi masalah pada Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur, kemudian melakukan studi literatur yakni mencari landasan kerangka berfikir sesuai topik permasalahan yang didapat dari identifikasi masalah. Selanjutnya dilakukan pengumpulan data sebagai bahan untuk analisis kebutuhan dan perancangan sistem yang akan dibangun. Kemudian dilanjutkan dengan proses pengembangan sistem dengan metode RUP yang memiliki empat fase. Fase pertama yaitu *inception* yang didalamnya terdapat proses pemodelan proses bisnis *as-is* dan *to-be* serta analisis kebutuhan. Fase kedua yaitu *elaboration* yang didalamnya terdapat proses perancangan sistem. Fase ketiga yaitu *construction* yang didalamnya terdapat proses implementasi dan pengujian dengan *validation testing* dan *compatibility testing*. Fase keempat yaitu *Transition* yang berisi proses *deployment* dan pengujian *usability testing*. Kemudian akan dilanjutkan dengan membuat kesimpulan dan saran dari penelitian yang telah dilakukan. Alur penelitian digambarkan pada Gambar 3.1.

3.1 Identifikasi Masalah

Pada tahap ini dilakukan pengenalan tempat penelitian secara umum serta menemukan permasalahan yang dihadapi oleh organisasi. Pengenalan tempat serta proses menemukan permasalahan dilakukan dengan metode wawancara. Wawancara dilakukan dengan mengajukan pertanyaan kepada pegawai mengenai Biro organisasi serta permasalahan yang ada di Biro Organisasi. Permasalahan yang diperoleh nantinya akan disusun kedalam rumusan masalah penelitian.

3.2 Studi Literatur

Kegiatan ini dilakukan untuk mendapatkan landasan kerangka berfikir dalam penelitian yang dilakukan. Sumber referensi didapat dari beberapa buku, studi kasus, jurnal, dan artikel yang ada di media cetak maupun internet. Peneliti melakukan metode studi pustaka dengan cara membaca dan mempelajari buku-buku atau penelitian sejenis serta melakukan pencarian di beberapa situs internet untuk pengumpulan data dan informasi terkait dengan permasalahan yang dibahas dalam pembuatan Sistem Informasi Keuangan Pegawai. Hasil dari tahap ini yaitu landasan teori yang mendukung latar belakang serta proses penelitian yang akan dilakukan.

3.3 Pengumpulan Data

Pada tahap ini peneliti akan melakukan pengumpulan data. Pengumpulan data dilakukan dengan metode wawancara dan dokumentasi. Dalam proses wawancara ini menggunakan metode wawancara secara langsung dengan mengajukan beberapa pertanyaan mengenai proses melakukan perjalanan dinas hingga pembuatan SPJ perjalanan dinas, pemberian uang makan, uang lembur uang sidang dan tambahan penghasilan pegawai. Metode dokumentasi dilakukan dengan melihat atau mempelajari dokumen-dokumen yang berkaitan dengan perjalanan dinas, uang makan, uang sidang, uang lembur dan tambahan

penghasilan pegawai. Metode dokumentasi juga dilakukan dengan melihat pedoman kerja dan pelaksanaan tugas pemerintah daerah provinsi jawa timur tahun 2018 yang diatur dalam peraturan Gubernur Provinsi Jawa Timur tahun 2017. Hasil dari proses pengumpulan data nantinya akan digunakan untuk pemodelan proses bisnis, analisis kebutuhan dan perancangan sistem yang akan dibangun.

3.4 Inception

Dari data yang diperoleh pada tahap pengumpulan data, akan dilakukan pemodelan proses bisnis yang ada saat ini (*as-is*). Selanjutnya akan dilakukan analisis permasalahan yang ada pada proses bisnis *as-is*. Dari permasalahan yang diperoleh pada proses bisnis *as-is* akan dilakukan pemodelan proses bisnis setelah adanya sistem (*to-be*) yang dapat mengatasi atau mengurangi permasalahan yang ada pada proses bisnis *as-is*. Dari data yang diperoleh pada tahap pengumpulan data juga akan dilakukan analisis permasalahan pada Biro Organisasi. Analisis permasalahan dilakukan untuk mengetahui permasalahan yang dihadapi, siapa yang terkena dampak dari permasalahan, apa dampak dari permasalahan, serta solusi yang disarankan akan dikatakan sukses jika bagaimana. Kemudian akan dilakukan identifikasi pengguna, kebutuhan, serta fitur untuk sistem yang akan dibangun. Selanjutnya akan dilakukan analisis kebutuhan fungsional dan non fungsional. Hasil yang diperoleh dari tahap *inception* berupa hasil analisis kebutuhan dan identifikasi fitur yang nantinya akan digunakan pada tahap *elaboration* untuk dilakukan perancangan sistem. Pada tahap *inception* dilakukan satu kali iterasi karena proyek pengembangan sistem tidak terlalu besar sehingga lebih mudah untuk menentukan ruang lingkup sistem. Selain itu tidak banyak stake holder yang terlibat dalam proyek pengembangan sistem, sehingga hubungan antara stake holder tidak terlalu kompleks.

3.5 Elaboration

Hasil analisis kebutuhan dan identifikasi fitur yang diperoleh pada tahap *inception* akan dipadukan dengan proses bisnis *to-be* untuk membuat perancangan sistem informasi administrasi keuangan pegawai yang akan dibangun. Perancangan sistem akan digambarkan menggunakan beberapa diagram UML (*Unified Modeling Language*). Perancangan dimulai dari pemodelan *use case diagram* yang dilanjutkan dengan pembuatan *use case scenario*. *Use case scenario* dapat dijadikan dasar dalam memodelkan *activity diagram* yang akan dijadikan dasar dalam pemodelan *sequence diagram*. Perancangan kelas-kelas yang akan dibangun dimodelkan dalam bentuk *class diagram*. Pemodelan basis data akan digambarkan menggunakan *physical data model* (PDM). Pada tahap ini akan dihasilkan dokumentasi rancangan sistem yang nantinya akan diimplementasikan pada tahap *construction*. Pada fase *elaboration* dilakukan satu kali iterasi karena sebelumnya peneliti pernah mengembangkan sistem dengan teknologi serupa, sehingga mempermudah aktivitas yang dilakukan pada fase elaborasi.

3.6 Construction

Hasil perancangan pada tahap elaboration akan diimplementasikan pada tahap ini. Bahasa pemrograman yang digunakan pada tahap ini adalah bahasa pemrograman PHP, *javascript* dan HTML dengan metode OOP (*Object Oriented Programming*) menggunakan *framework CodeIgniter*. Pada tahap ini juga dilakukan pengujian *blackbox testing* dengan metode *validation testing*. Pengujian ini dilakukan untuk menguji apakah sistem yang dibuat sudah sesuai dengan rancangan awal yang sudah ditetapkan. Kemudian akan dilakukan pengujian kompatibilitas untuk memastikan bahwa sistem yang dikembangkan dapat berjalan dengan baik pada beberapa *browser* yang berbeda. Setelah melakukan pengujian sistem dan hasil pengujian yang diperoleh sesuai dengan rancangan awal maka dapat dilanjutkan ke tahap selanjutnya yaitu *transition*. Hasil dari tahap *construction* yaitu Sistem Informasi Keuangan Pegawai Biro Organisasi Setda Provinsi Jawa Timur yang telah diuji dengan metode *validation testing* dan *compatibility testing*. Pada fase ini juga dilakukan analisis perubahan waktu proses bisnis *as-is* dan *to-be* dengan menjalankan proses bisnis *as-is* dan *to-be* dengan masukan yang sama. Ketika pada proses bisnis *as-is* dilakukan pencairan uang sidang untuk 10 orang pegawai, maka pada proses bisnis *to-be* juga dilakukan pencairan untuk 10 orang pegawai. Fase *construction* dikerjakan dengan tiga kali proses iterasi. Iterasi pertama yaitu mengimplementasikan rancangan sistem utama bagian-per bagian yaitu mengimplementasikan *use case* sesuai dengan *basic flow*-nya kemudian melakukan pengujian agar dapat dilakukan pembetulan jika ada *use case* yang tidak berjalan sesuai *basic flow*-nya. Iterasi berikutnya yaitu membangun sistem yang lebih mendetail dengan mengimplementasikan *alternative flow* dari *use case*, kemudian dilakukan pengujian agar dapat dilakukan pembetulan terhadap fungsi yang tidak sesuai dengan *flow* pada spesifikasi *use case*. Iterasi terakhir yaitu mengintegrasikan bagian-bagian sistem yang telah dibangun serta melakukan perbaikan apabila terjadi kesalahan atau ketidaksesuaian saat melakukan integrasi bagian sistem untuk mendapatkan sistem yang telah terintegrasi dengan baik.

3.7 Transition

Pada tahap ini dilakukan *deployment* yaitu instalasi sistem informasi keuangan pegawai pada Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur menggunakan server lokal (*localhost*) dengan aplikasi *xampp* serta pengujian usability terhadap sistem menggunakan *tool SUS (System Usability Scale)* untuk memastikan bahwa sistem dapat digunakan dengan mudah oleh pengguna. Bendahara Biro, pegawai, serta bendahara bagian akan menggunakan sistem yang sudah di install. Pada tahap transisi hanya dilakukan satu kali iterasi karena sistem yang dibangun tidak terlalu kompleks sehingga setelah melakukan instalasi dan di coba oleh pengguna mungkin hanya memerlukan sedikit perbaikan sebelum sistem dapat digunakan sepenuhnya oleh pengguna.

3.8 Kesimpulan dan Saran

Rumusan masalah yang telah dirancang pada pendahuluan akan dijawab pada tahap pembuatan kesimpulan ini. Kesimpulan diperoleh dari hasil Analisis dan pengujian sistem informasi keuangan pegawai yang dibuat. Dari hasil analisis dan pengujian sistem yang telah dibuat dapat diperoleh perbandingan antara sebelum dan sesudah adanya sistem informasi keuangan pegawai di Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur. Dari perbandingan yang diperoleh akan dilihat apakah sistem yang dibuat dapat memenuhi tujuan dilakukannya penelitian. Kemudian dari hasil kesimpulan akan dilakukan pembuatan saran yang dapat menjadi acuan pengembangan yang lebih lanjut.



BAB 4 ANALISIS PERSYARATAN

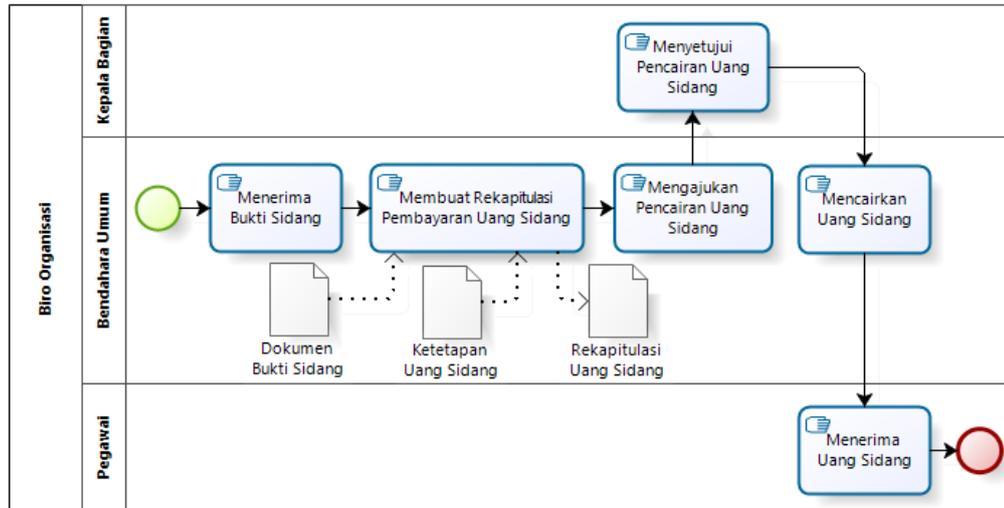
Bab ini membahas mengenai analisis persyaratan yang diperoleh dari analisis hasil pengumpulan data dengan wawancara dan studi dokumen. Hasil analisis persyaratan akan dimodelkan dalam bentuk proses bisnis, tipe pemangku kepentingan, pernyataan masalah, kebutuhan pemangku kepentingan dan pengguna, fitur, kebutuhan fungsional dan non fungsional, *use case* serta *activity diagram*.

4.1 Proses Bisnis

Pemodelan proses bisnis dilakukan untuk memberikan gambaran mengenai rangkaian aktivitas yang dilakukan oleh Biro Organisasi terkait proses pemberian keuangan pegawai di Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur. Terdapat enam proses bisnis saat ini (*as-is*) dan tujuh proses bisnis usulan (*to-be*) yang akan dimodelkan. Enam proses bisnis *as-is* yang akan dimodelkan yaitu proses bisnis mengenai pencairan uang sidang, pencairan uang perjalanan dinas, pencairan uang makan, pencairan uang lembur, pencairan tunjangan prestasi, serta pencairan tunjangan daerah. Sedangkan proses bisnis *to-be* yang akan dimodelkan yaitu proses bisnis mengenai pencairan uang sidang, pencairan uang perjalanan dinas, pencairan uang makan, pencairan uang lembur, pencairan tunjangan prestasi, pencairan tunjangan daerah, serta pencatatan absensi serta lembur pegawai. Pemodelan proses bisnis dilakukan setelah mengetahui informasi mengenai alur proses bisnis pada Biro Organisasi yang diperoleh dengan melakukan wawancara terhadap bendahara Biro Organisasi serta mempelajari dokumen dan peraturan terkait perjalanan dinas dan keuangan pegawai di Biro Organisasi.

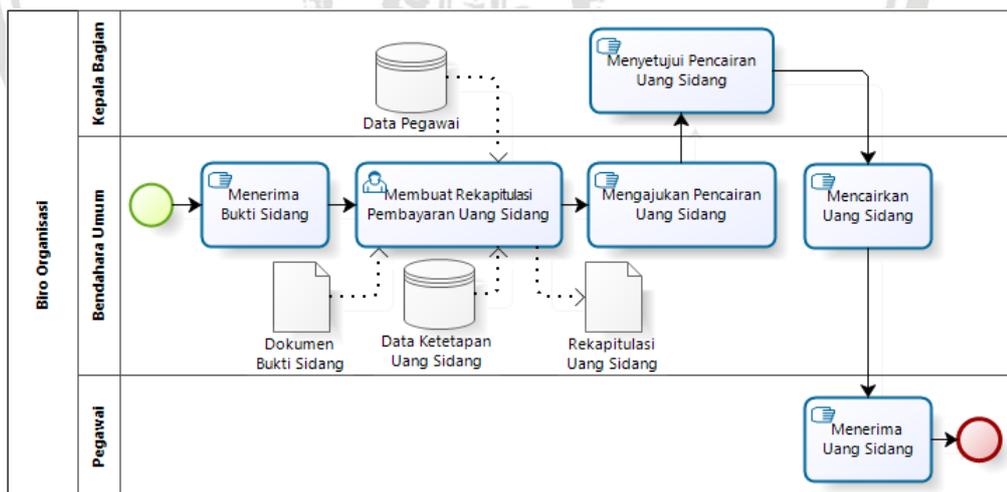
4.1.1 Proses Bisnis Pencairan Uang Sidang

Pencairan uang sidang dilakukan setelah dilaksanakannya sidang. Proses bisnis mengenai pencairan uang sidang yang ada di Biro organisasi (*as-is*) dapat dilihat pada Gambar 4.1. Proses bisnis dimulai ketika bendahara biro menerima bukti dilaksanakannya sidang (seperti berita acara dan presensi sidang). Kemudian bendahara biro akan membuat rekapitulasi pembayaran uang sidang sesuai daftar presensi dan berita acara sidang. Setelah membuat rekapitulasi pembayaran uang sidang, bendahara biro akan mengajukan pencairan uang sidang kepada kepala bagian, kemudian mencairkan uang sidang setelah disetujui oleh kepala bagian.



Gambar 4. 1 Proses Bisnis Pencairan uang Sidang (*as-is*)

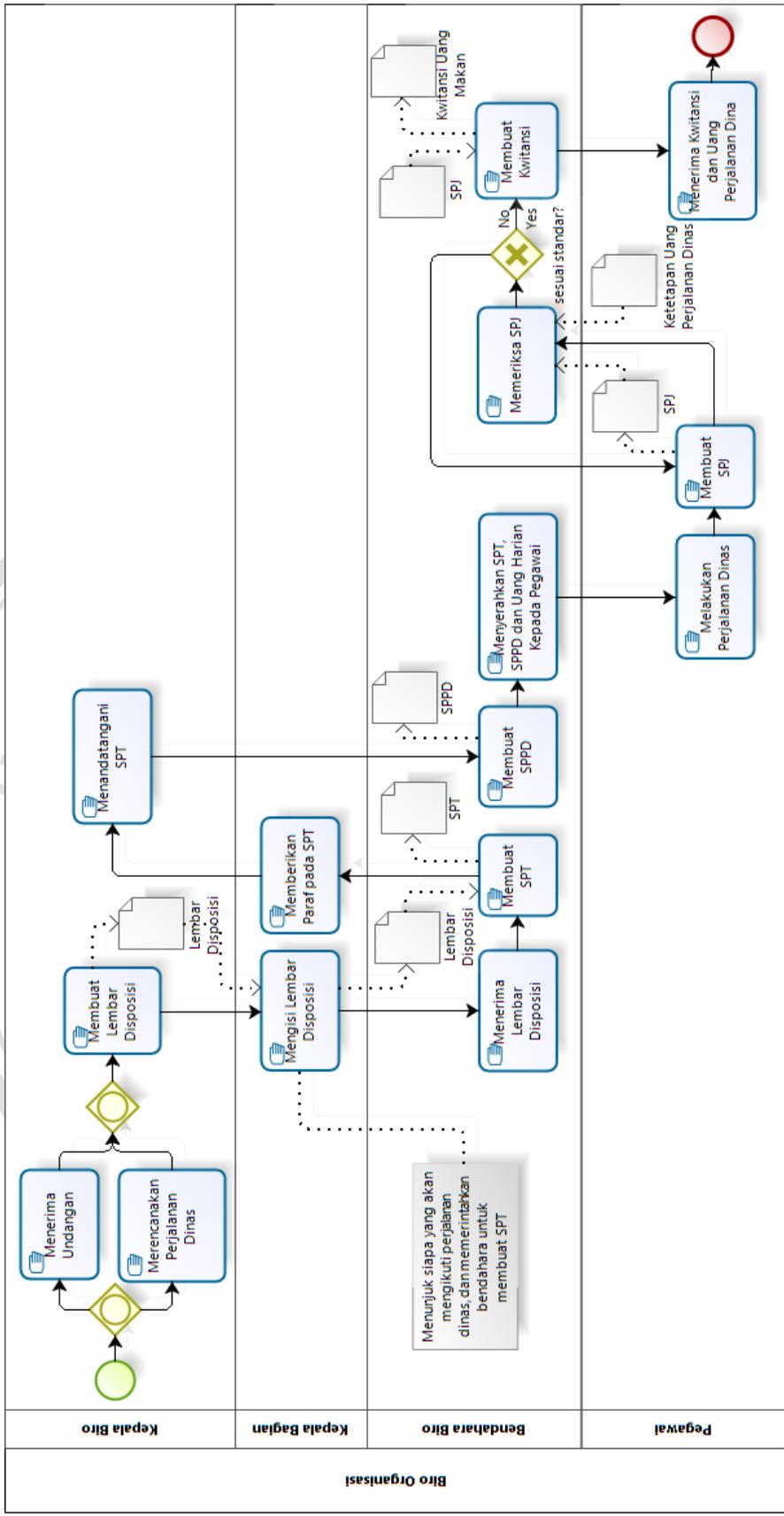
Setelah memodelkan proses bisnis *as-is*, dilakukan pemodelan proses bisnis usulan pencairan uang sidang setelah diterapkannya sistem informasi keuangan pegawai. Proses bisnis usulan dapat dilihat pada Gambar 4.2. Proses bisnis dimulai ketika bendahara biro menerima bukti dilaksanakannya sidang (seperti berita acara dan presensi sidang). Kemudian bendahara biro akan membuat rekapitulasi pembayaran uang sidang sesuai daftar presensi dan berita acara sidang dengan cara menginputkan nama pegawai yang mengikuti sidang. Penentuan serta perhitungan uang sidang dilakukan oleh sistem sesuai dengan tabel ketentuan uang sidang. Setelah membuat rekapitulasi pembayaran uang sidang, bendahara biro akan mengajukan pencairan uang sidang kepada kepala bagian kemudian mencairkan uang sidang setelah disetujui oleh kepala biro.



Gambar 4. 2 Proses Bisnis Pencairan Uang Sidang (*to-be*)

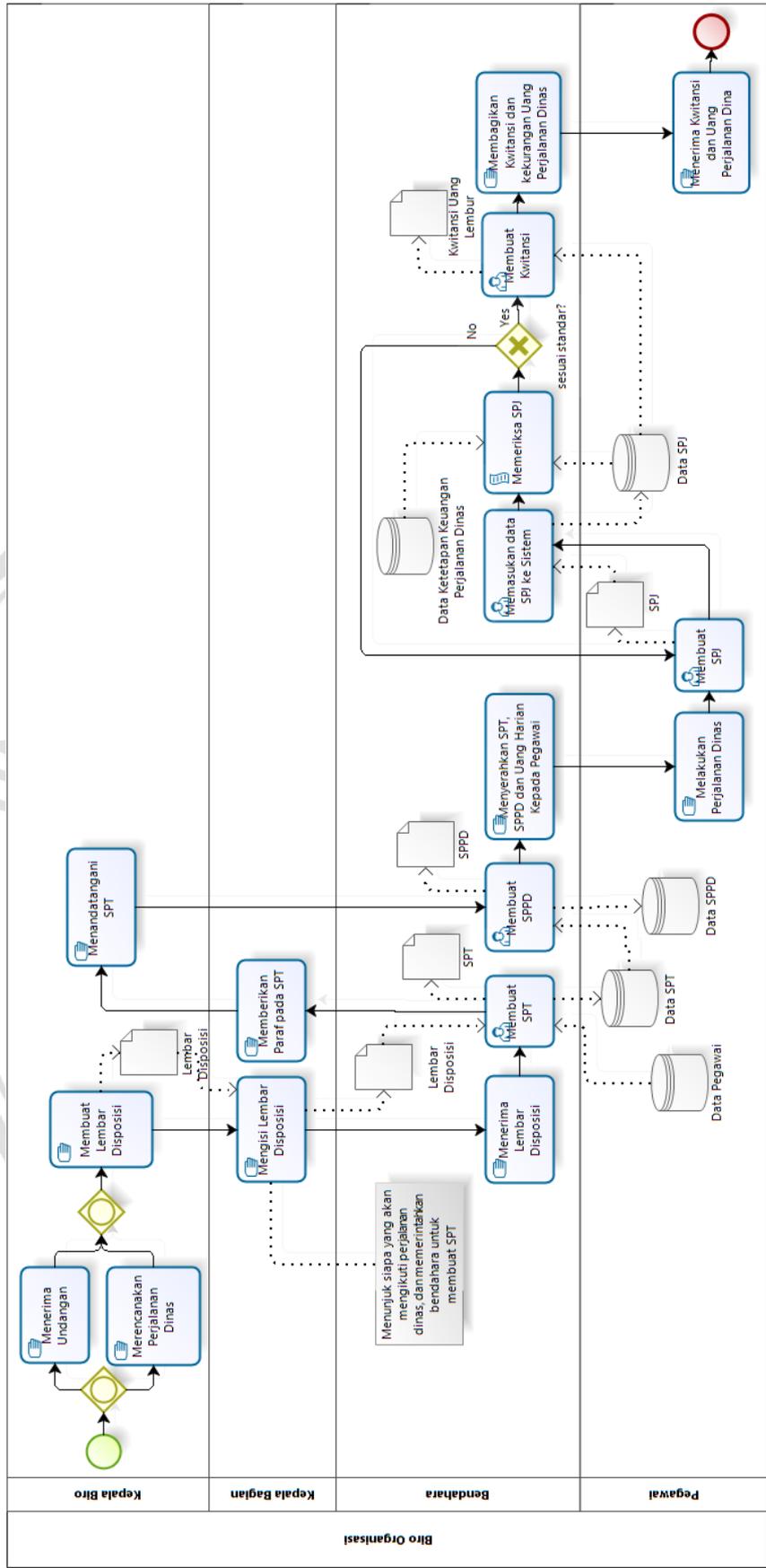
4.1.2 Proses Bisnis Pencairan Uang Perjalanan Dinas

Pencairan uang perjalanan dinas dilakukan setelah dilaksanakannya perjalanan dinas dan pegawai menyerahkan SPJ perjalanan dinas. Proses bisnis mengenai pencairan uang perjalanan dinas yang ada di Biro organisasi (as-is) dapat dilihat pada Gambar 4.3. Proses bisnis dimulai ketika kepala biro menerima surat undangan untuk melakukan perjalanan dinas atau kepala biro merencanakan perjalanan dinas. Kemudian kepala Biro akan membuat Lembar disposisi yang akan dilampiri lampiran surat undangan perjalanan dinas atau agenda perjalanan dinas yang akan diserahkan kepada kepala bagian. Setelah kepala bagian menerima lembar disposisi, kepala bagian akan mengisi lembar disposisi dengan nama-nama pegawai yang akan mengikuti perjalanan dinas, serta memerintahkan bendahara untuk membuat SPT. Setelah bendahara membuat SPT, kepala bagian akan memberikan paraf pada SPT dan mengajukannya kepada kepala biro untuk ditandatangani. Setelah mendapat SPT yang telah ditandatangani oleh kepala biro, bendahara bagian akan membuat SPPD kemudian menyerahkan SPPD, SPT, dan uang harian kepada pegawai yang ditugaskan untuk melakukan perjalanan dinas. Pegawai akan melakukan perjalanan dinas dengan membawa SPT dan SPPD. Pada tujuan perjalanan dinas, penanggung jawab acara akan menandatangani lembar 2 SPPD sebagai bukti bahwa pegawai bersangkutan telah melakukan perjalanan dinas. Setelah melakukan perjalanan dinas, pegawai akan membuat SPJ dan menyerahkannya kepada bendahara biro beserta lembar 2 SPPD. Bendahara biro akan memeriksa SPJ dan lembar 2 SPPD sesuai pedoman ketentuan uang perjalanan dinas. Jika SPJ sesuai dengan ketentuan, bendahara biro akan membuat kwitansi dan membagikan kekurangan uang perjalanan dinas kepada pegawai.



Gambar 4. 3 Proses Bisnis Pencairan Uang Perjalanan Dinas (as-is)

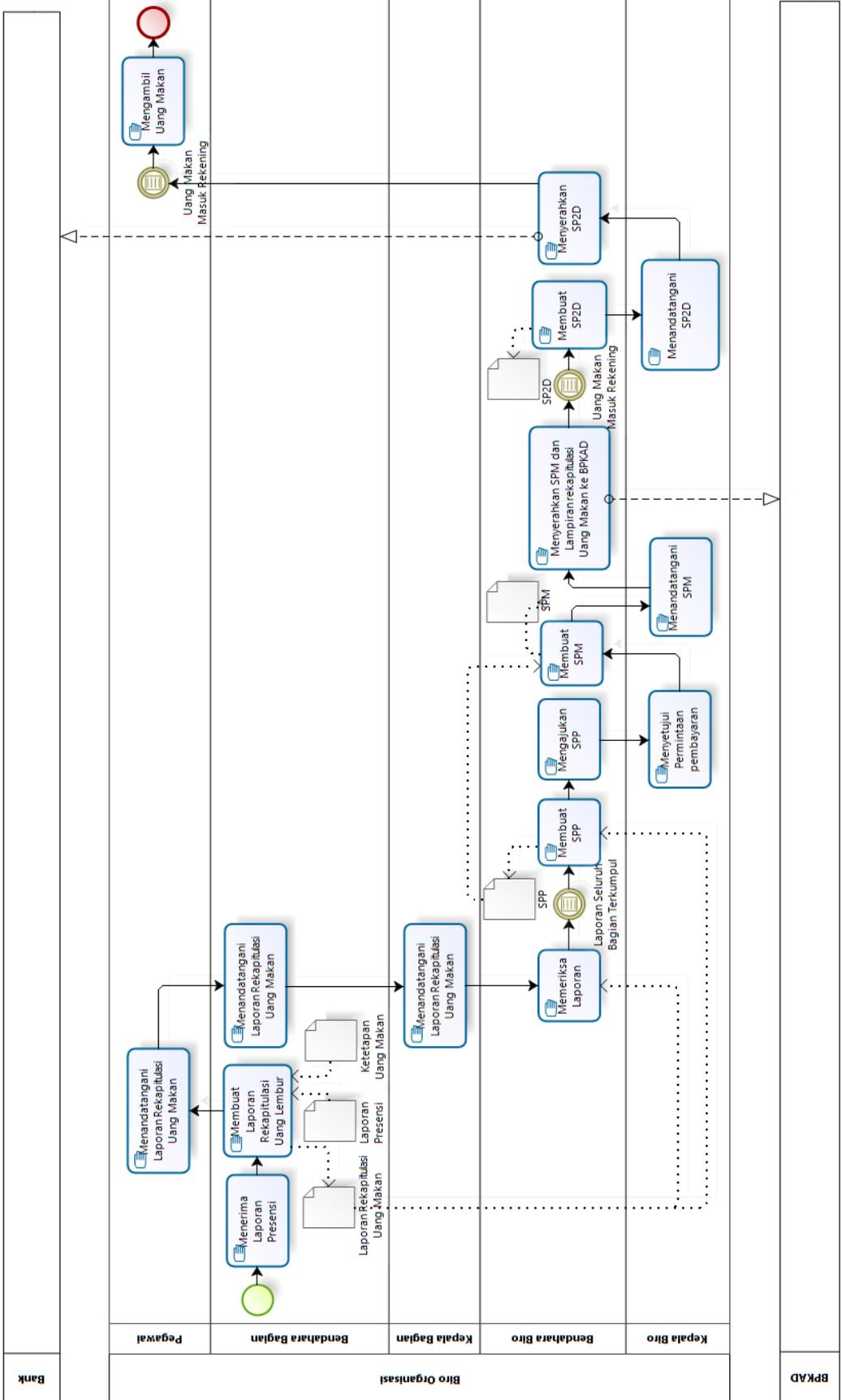
Setelah melakukan simulasi pada proses bisnis *as-is*, dilakukan pemodelan proses bisnis usulan pencairan uang perjalanan dinas setelah diterapkannya sistem informasi keuangan pegawai. Proses bisnis usulan dapat dilihat pada gambar 4.4. Proses bisnis dimulai ketika kepala biro menerima surat undangan untuk melakukan perjalanan dinas atau kepala biro merencanakan perjalanan dinas. Kemudian kepala Biro akan membuat Lembar disposisi yang akan dilampiri lampiran surat undangan perjalanan dinas atau agenda perjalanan dinas yang akan diserahkan kepada kepala bagian pada sistem. Setelah kepala bagian menerima lembar disposisi, kepala bagian akan mengisi lembar disposisi dengan nama-nama pegawai yang akan mengikuti perjalanan dinas, serta memerintahkan bendahara untuk membuat SPT. Bendahara membuat SPT pada sistem dengan menginputkan nama pegawai serta keterangan acara sesuai lembar disposisi. Kemudian sistem akan membuat SPT dan menyimpan data spt pada *database*. Setelah bendahara membuat SPT, kepala bagian akan memberikan paraf pada SPT dan mengajukannya kepada kepala biro untuk ditandatangani. Setelah mendapat SPT yang telah ditandatangani oleh kepala biro, bendahara bagian akan membuat SPPD melalui sistem sesuai dengan SPT yang sudah tersimpan dalam *database* tadi dan menyerahkan SPPD, SPT, dan uang harian kepada pegawai yang ditugaskan untuk melakukan perjalanan dinas. Kemudian pegawai akan melakukan perjalanan dinas dengan membawa SPT dan SPPD. Pada tujuan perjalanan dinas, penanggung jawab acara akan menandatangani lembar 2 SPPD sebagai bukti bahwa pegawai bersangkutan telah melakukan perjalanan dinas. Setelah melakukan perjalanan dinas, pegawai akan membuat SPJ dan menyerahkannya kepada bendahara biro beserta lembar 2 SPPD. Bendahara biro akan menginputkan data SPJ pegawai ke sistem. Kemudian sistem akan memeriksa SPJ sesuai pedoman ketentuan uang perjalanan dinas. Jika SPJ sesuai dengan ketentuan, bendahara biro akan membuat kwitansi dan membagikan kekurangan uang perjalanan dinas kepada pegawai.



Gambar 4. 4 Proses Bisnis Pencairan Uang Perjalanan Dinas (to-be)

4.1.3 Proses Bisnis Pencairan Uang Makan

Pencairan uang dilakukan setiap akhir bulan dengan mempertimbangkan kehadiran dan golongan pegawai. Proses bisnis mengenai pencairan uang makan yang ada di Biro organisasi (*as-is*) dapat dilihat pada Gambar 4.5. Proses bisnis dimulai ketika bendahara masing-masing bagian menerima laporan presensi pegawai. Kemudian bendahara bagian akan membuat laporan rekapitulasi uang makan seluruh pegawai bagiannya berdasarkan presensi dan ketentuan uang makan yang telah ditetapkan. Setelah bendahara bagian membuat laporan rekapitulasi uang makan, pegawai akan menandatangani laporan rekapitulasi tersebut yang dilanjutkan dengan penandatanganan laporan oleh bendahara bagian kemudian kepala bagian. Setelah itu Laporan rekapitulasi uang makan akan diserahkan kepada bendahara biro untuk dilakukan pengecekan. Bendahara biro akan menunggu seluruh bendahara bagian menyerahkan laporan rekapitulasi uang makan bagiannya untuk membuat surat permintaan pembayaran dan mengajukan pencairan uang makan kepada kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi uang makan dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi uang makan, BPKAD akan mentransfer uang sesuai rekapitulasi uang makan ke rekening biro. Bendahara biro akan menunggu hingga uang makan masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana uang makan ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer uang makan ke masing-masing pegawai. Pegawai akan menunggu hingga uang makan masuk ke rekening mereka masing-masing, kemudian pegawai dapat mengambil uang makan mereka selama satu bulan.

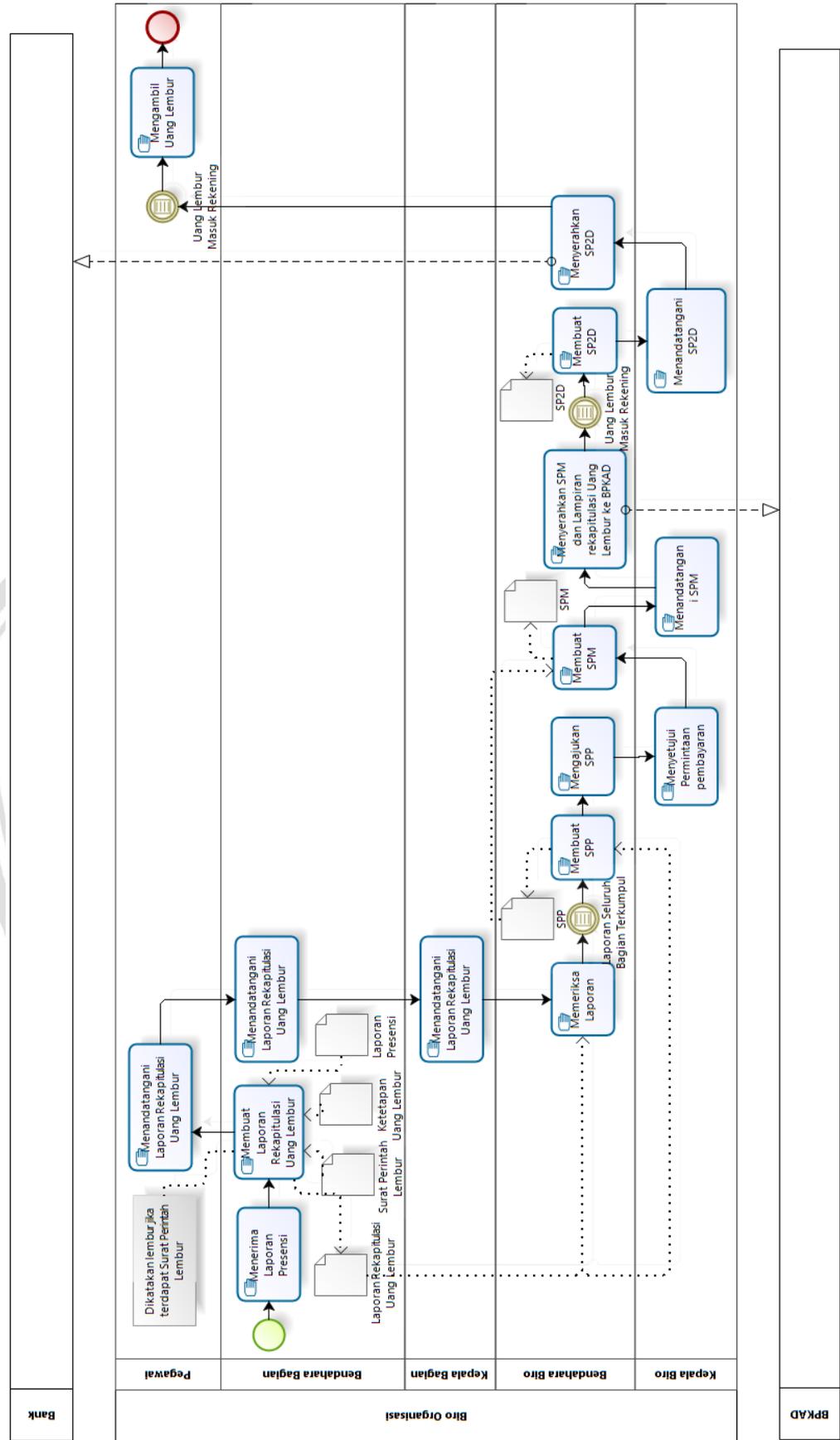


Gambar 4. 5 Proses Bisnis Pencairan Uang Makan (as-is)

Proses bisnis usulan mengenai pencairan uang makan yang ada di Biro organisasi (*to-be*) dapat dilihat pada Gambar 4.6. Proses bisnis dimulai ketika bendahara biro mengupload laporan presensi pegawai ke sistem informasi keuangan pegawai. Kemudian sistem akan secara otomatis *me-replace* presensi kosong pada daftar presensi dengan data absensi pegawai yang ada pada sistem. Setelah itu sistem akan secara otomatis menghitung uang makan pegawai dan memasukkan datanya ke *database*. Kemudian bendahara bagian akan membuat laporan rekapitulasi uang makan dengan mencetak laporan rekapitulasi uang makan menggunakan data yang ada pada sistem. Setelah bendahara bagian membuat laporan rekapitulasi uang makan, pegawai akan menandatangani laporan rekapitulasi tersebut yang dilanjutkan dengan penandatanganan laporan oleh bendahara bagian kemudian kepala bagian. Setelah itu Laporan rekapitulasi uang makan akan diserahkan kepada bendahara biro untuk dilakukan pengecekan. Pengecekan dilakukan oleh bendahara biro dengan mencocokkan nominal pada laporan bendahara bagian dengan nominal pada sistem, sehingga proses pengecekan tidak perlu dilakukan secara manual dan dapat diselesaikan lebih cepat. Bendahara biro akan menunggu seluruh bendahara bagian menyerahkan laporan rekapitulasi uang makan bagiannya untuk membuat surat permintaan pembayaran dan mengajukan pencairan uang makan kepada kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi uang makan dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi uang makan, BPKAD akan mentransfer uang sesuai rekapitulasi uang makan ke rekening biro. Bendahara biro akan menunggu hingga uang makan masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana uang makan ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer uang makan ke masing-masing pegawai. Pegawai akan menunggu hingga uang makan masuk ke rekening mereka masing-masing, kemudian pegawai dapat mengambil uang makan mereka selama satu bulan.

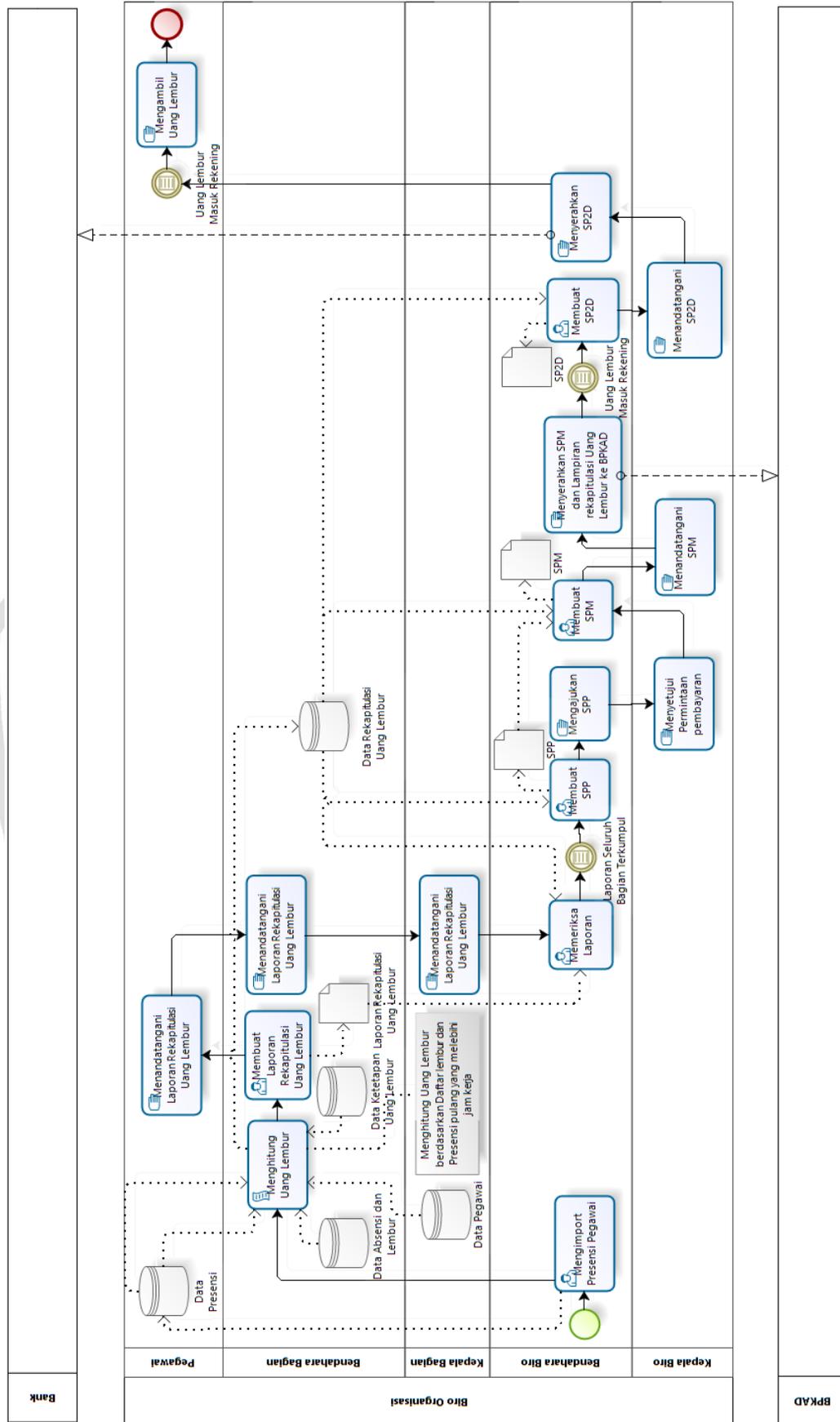
4.1.4 Proses Bisnis Pencairan Uang Lembur

Pencairan uang lembur dilakukan setiap akhir bulan dengan bukti berupa surat perintah lembur untuk masing-masing presensi pegawai yang melebihi jam pulang. Proses bisnis mengenai pencairan uang lembur yang ada di Biro organisasi (*as-is*) dapat dilihat pada Gambar 4.7. Proses bisnis dimulai ketika bendahara masing-masing bagian menerima laporan presensi pegawai. Kemudian bendahara bagian akan membuat laporan rekapitulasi uang lembur seluruh pegawai bagiannya berdasarkan presensi, surat perintah lembur dan ketentuan uang lembur yang telah ditetapkan. Setelah bendahara bagian membuat laporan rekapitulasi uang lembur, pegawai akan menandatangani laporan rekapitulasi tersebut yang dilanjutkan dengan penandatanganan laporan oleh bendahara bagian kemudian kepala bagian. Setelah itu Laporan rekapitulasi uang lembur akan diserahkan kepada bendahara biro untuk dilakukan pengecekan. Bendahara biro akan menunggu seluruh bendahara bagian menyerahkan laporan rekapitulasi uang lembur bagiannya untuk membuat surat permintaan pembayaran dan mengajukan pencairan uang lembur kepada kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi uang lembur dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi uang lembur, BPKAD akan mentransfer uang sesuai rekapitulasi uang lembur ke rekening biro. Bendahara biro akan menunggu hingga uang lembur masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana uang lembur ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer uang lembur ke masing-masing pegawai. Pegawai akan menunggu hingga uang lembur masuk ke rekening mereka masing-masing, kemudian pegawai dapat mengambil uang lembur mereka selama satu bulan.



Gambar 4. 7 Proses Bisnis Pencairan Uang Lembur (as-is)

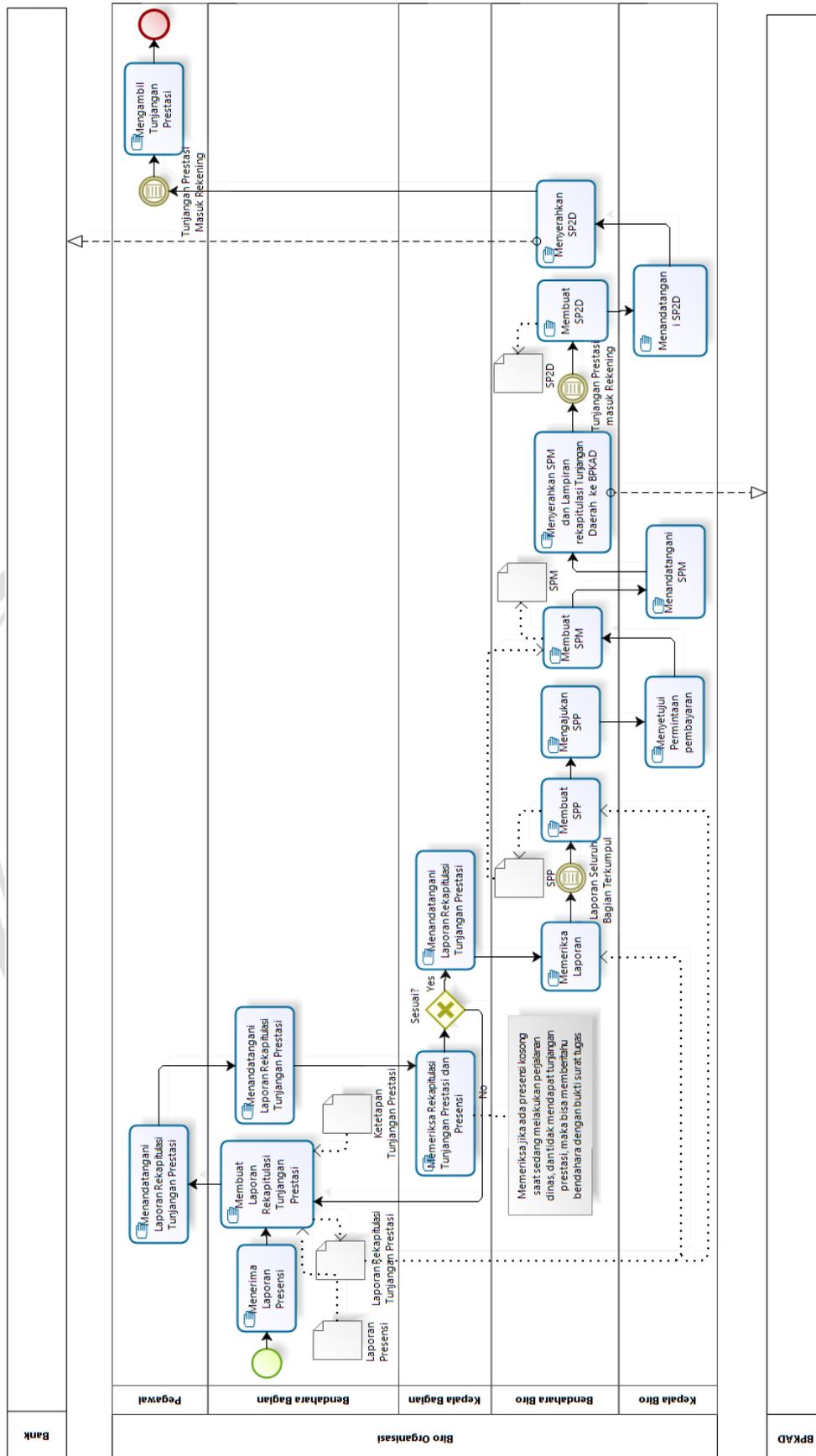
Proses bisnis usulan mengenai pencairan uang lembur yang ada di Biro organisasi (*to-be*) dapat dilihat pada Gambar 4.8. Proses bisnis dimulai ketika bendahara biro mengupload laporan presensi pegawai ke sistem informasi keuangan pegawai. Kemudian sistem akan secara otomatis *me-replace* presensi kosong pada daftar presensi dengan data absensi pegawai yang ada pada sistem. Setelah itu sistem akan secara otomatis menghitung uang lembur pegawai dan memasukkan datanya ke *database*. Kemudian bendahara bagian akan membuat laporan rekapitulasi uang lembur dengan mencetak laporan rekapitulasi uang lembur menggunakan data yang ada pada sistem. Setelah bendahara bagian membuat laporan rekapitulasi uang lembur, pegawai akan menandatangani laporan rekapitulasi tersebut yang dilanjutkan dengan penandatanganan laporan oleh bendahara bagian kemudian kepala bagian. Setelah itu Laporan rekapitulasi uang lembur akan diserahkan kepada bendahara biro untuk dilakukan pengecekan. Pengecekan dilakukan oleh bendahara biro dengan mencocokkan nominal pada laporan bendahara bagian dengan nominal pada sistem, sehingga proses pengecekan tidak perlu dilakukan secara manual dan dapat diselesaikan lebih cepat. Bendahara biro akan menunggu seluruh bendahara bagian menyerahkan laporan rekapitulasi uang lembur bagiannya untuk membuat surat permintaan pembayaran dan mengajukan pencairan uang lembur kepada kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi uang lembur dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi uang lembur, BPKAD akan mentransfer uang sesuai rekapitulasi uang lembur ke rekening biro. Bendahara biro akan menunggu hingga uang lembur masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana uang lembur ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer uang lembur ke masing-masing pegawai. Pegawai akan menunggu hingga uang lembur masuk ke rekening mereka masing-masing, kemudian pegawai dapat mengambil uang lembur mereka selama satu bulan.



Gambar 4. 8 Proses Bisnis Pencairan Uang Lembur (as-is)

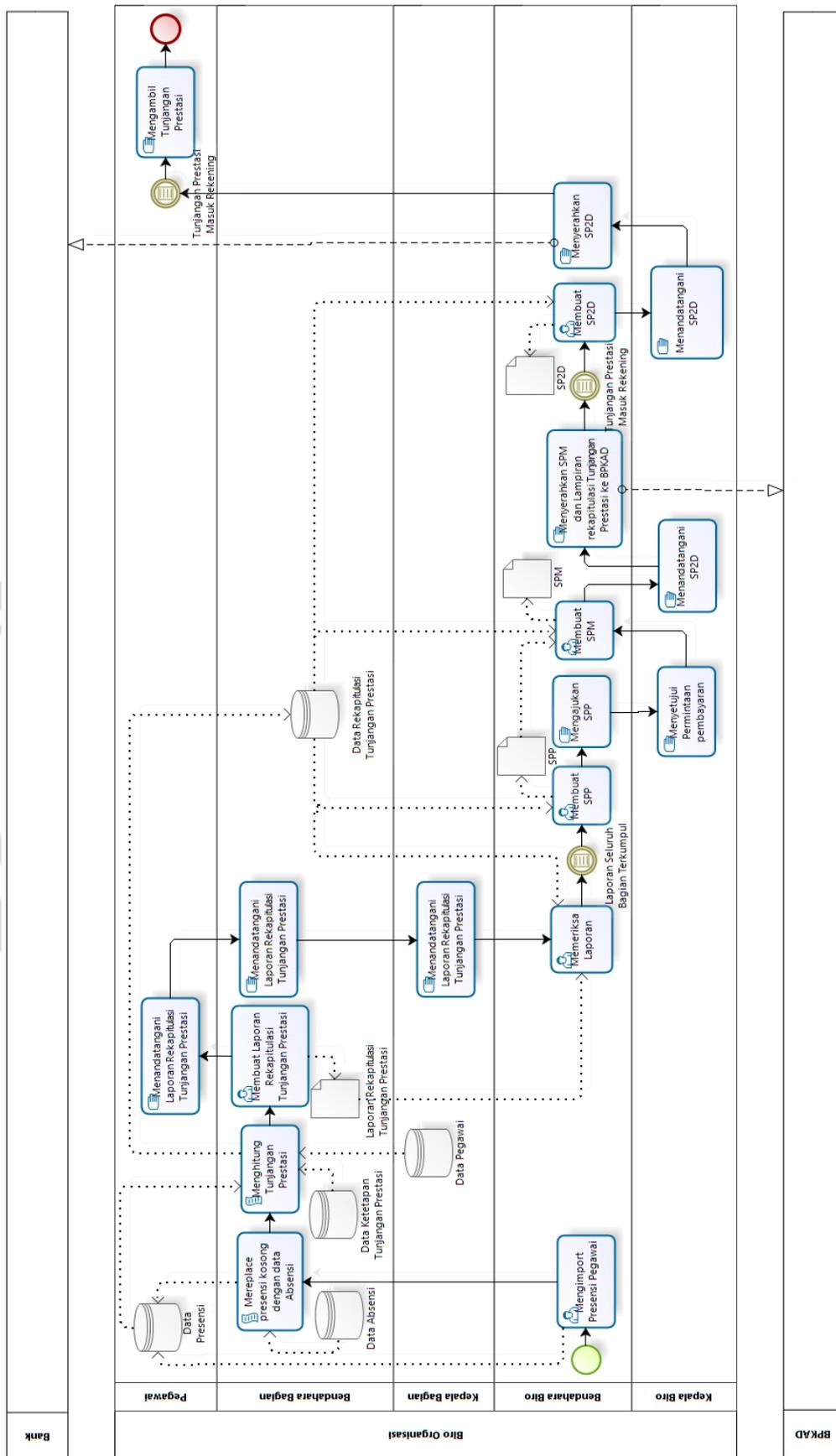
4.1.5 Proses Bisnis Pencairan Tunjangan Prestasi

Pencairan Tunjangan prestasi dilakukan setiap akhir bulan dengan mempertimbangkan kehadiran dan golongan pegawai. Proses bisnis mengenai pencairan tunjangan prestasi yang ada di Biro organisasi (*as-is*) dapat dilihat pada Gambar 4.9. Proses bisnis dimulai ketika bendahara masing-masing bagian menerima laporan presensi pegawai. Kemudian bendahara bagian akan membuat laporan rekapitulasi tunjangan prestasi seluruh pegawai bagiannya berdasarkan presensi dan ketentuan tunjangan prestasi yang telah ditetapkan. Setelah bendahara bagian membuat laporan rekapitulasi tunjangan prestasi, apabila rekapitulasi keuangan sudah sesuai pegawai akan menandatangani laporan rekapitulasi tersebut, jika tidak (misalnya ada presensi kosong saat pegawai sedang melakukan perjalanan dinas sehingga tidak mendapatkan tunjangan prestasi pada hari tersebut) pegawai dapat menyampaikan kepada bendahara bagian bahwa pada hari tersebut sedang melakukan perjalanan dinas dengan bukti SPT. Selanjutnya akan dilakukan penandatanganan laporan oleh bendahara bagian kemudian kepala bagian. Setelah itu Laporan rekapitulasi tunjangan prestasi akan diserahkan kepada bendahara biro untuk dilakukan pengecekan. Bendahara biro akan menunggu seluruh bendahara bagian menyerahkan laporan rekapitulasi tunjangan prestasi bagiannya untuk membuat surat permintaan pembayaran dan mengajukan pencairan tunjangan prestasi kepada kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi tunjangan prestasi dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi tunjangan prestasi, BPKAD akan mentransfer uang sesuai rakpitulasi tunjangan prestasi ke rekening biro. Bendahara biro akan menunggu hingga tunjangan prestasi masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana tunjangan prestasi ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer tunjangan prestasi ke masing-masing pegawai. Pegawai akan menunggu hingga tunjangan prestasi masuk ke rekening mereka masing-masing, kemudian pegawai dapat mengambil tunjangan prestasi mereka selama satu bulan.



Gambar 4. 9 Proses Bisnis Pencairan Tunjangan Prestasi (to-be)

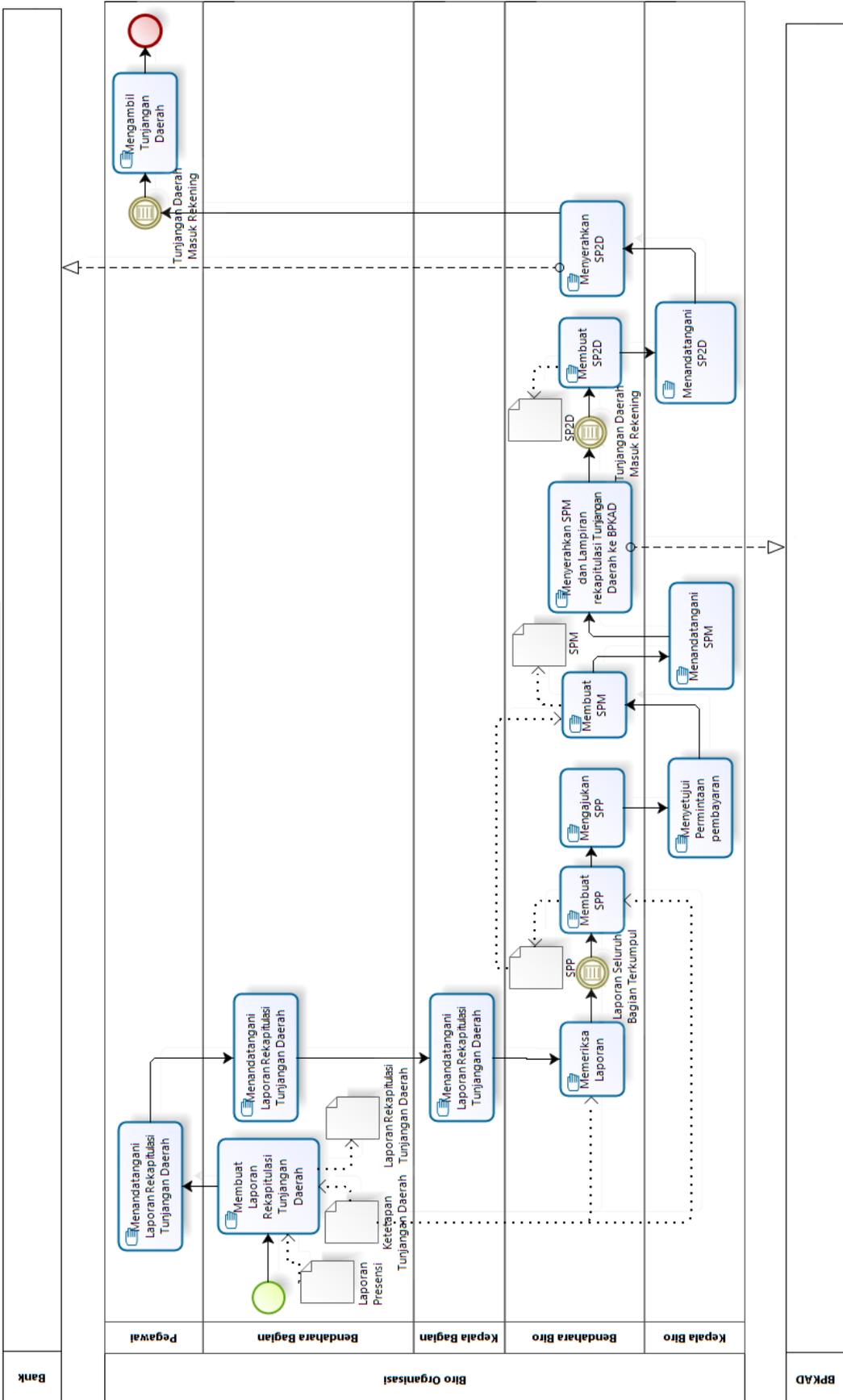
Proses bisnis usulan mengenai pencairan tunjangan prestasi yang ada di Biro organisasi (*to-be*) dapat dilihat pada Gambar 4.10. Proses bisnis dimulai ketika bendahara biro meng-*upload* laporan presensi pegawai ke sistem informasi keuangan pegawai. Kemudian sistem akan secara otomatis me-*replace* presensi kosong pada daftar presensi dengan data absensi pegawai yang ada pada sistem. Setelah itu sistem akan secara otomatis menghitung tunjangan prestasi pegawai dan memasukkan datanya ke *database*. Kemudian bendahara bagian akan membuat laporan rekapitulasi tunjangan prestasi dengan mencetak laporan rekapitulasi tunjangan prestasi menggunakan data yang ada pada sistem. Setelah bendahara bagian membuat laporan rekapitulasi tunjangan prestasi, pegawai akan menandatangani laporan rekapitulasi tersebut yang dilanjutkan dengan penandatanganan laporan oleh bendahara bagian kemudian kepala bagian. Setelah itu Laporan rekapitulasi tunjangan prestasi akan diserahkan kepada bendahara biro untuk dilakukan pengecekan. Pengecekan dilakukan oleh bendahara biro dengan mencocokkan nominal pada laporan bendahara bagian dengan nominal pada sistem, sehingga proses pengecekan tidak perlu dilakukan secara manual dan dapat diselesaikan lebih cepat. Bendahara biro akan menunggu seluruh bendahara bagian menyerahkan laporan rekapitulasi tunjangan prestasi bagiannya untuk membuat surat permintaan pembayaran dan mengajukan pencairan tunjangan prestasi kepada kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi tunjangan prestasi dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi tunjangan prestasi, BPKAD akan mentransfer uang sesuai rekapitulasi tunjangan prestasi ke rekening biro. Bendahara biro akan menunggu hingga tunjangan prestasi masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana tunjangan prestasi ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer tunjangan prestasi ke masing-masing pegawai. Pegawai akan menunggu hingga tunjangan prestasi masuk ke rekening mereka masing-masing, kemudian pegawai dapat mengambil tunjangan prestasi mereka selama satu bulan.



Gambar 4. 10 Proses Bisnis Pencairan Tunjangan Prestasi (to-be)

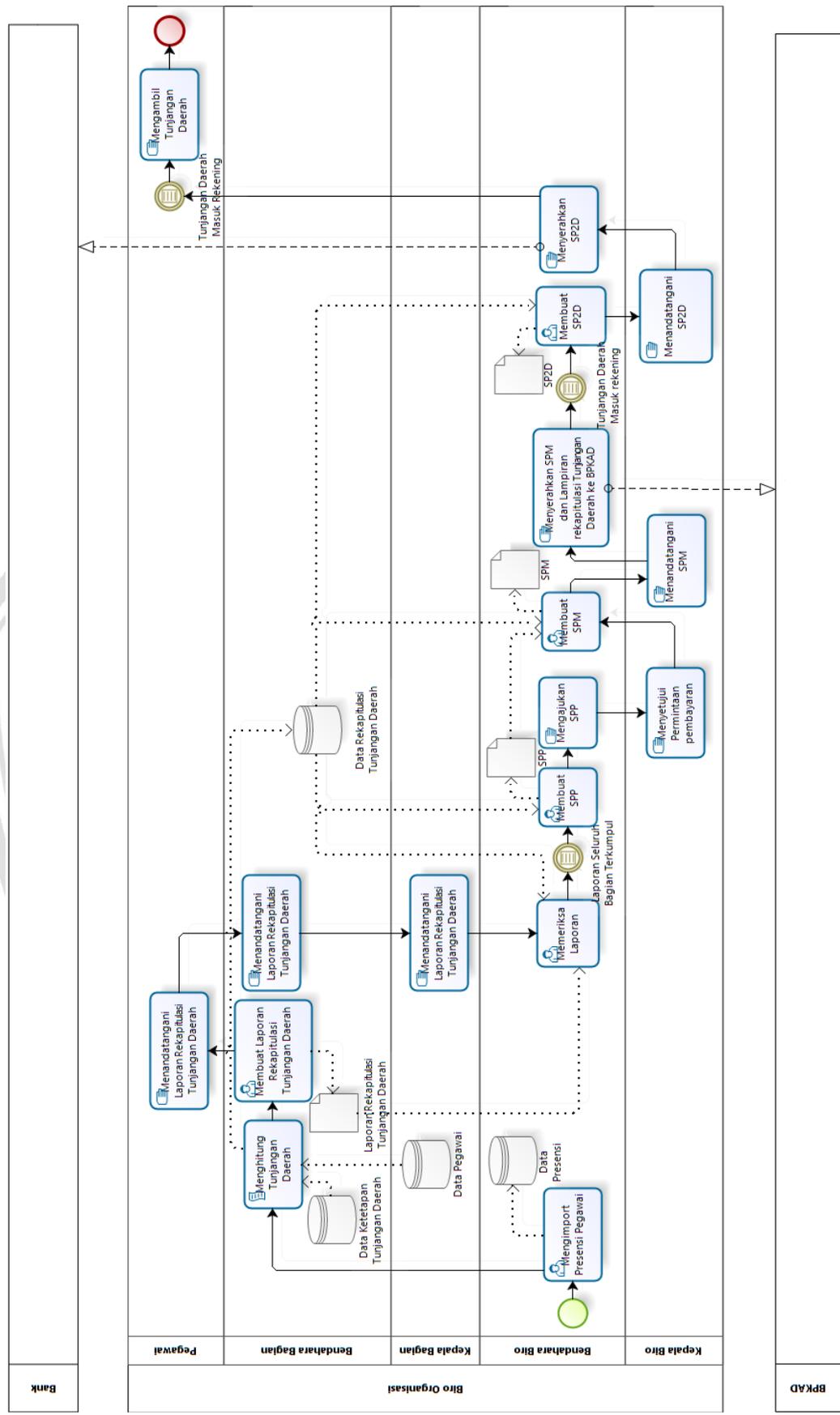
4.1.6 Proses Bisnis Pencairan Tunjangan Daerah

Pencairan tunjangan daerah dilakukan setiap akhir. Proses bisnis mengenai pencairan tunjangan daerah yang ada di Biro organisasi (as-is) dapat dilihat pada Gambar 4.11. Proses bisnis dimulai dengan pembuatan laporan rekapitulasi tunjangan daerah oleh bendahara bagian berdasarkan ketentuan tunjangan daerah yang telah ditetapkan. Setelah bendahara bagian membuat laporan rekapitulasi tunjangan daerah, pegawai akan menandatangani laporan rekapitulasi tersebut yang dilanjutkan dengan penandatanganan laporan oleh bendahara bagian kemudian kepala bagian. Setelah itu Laporan rekapitulasi tunjangan daerah akan diserahkan kepada bendahara biro untuk dilakukan pengecekan. Bendahara biro akan menunggu seluruh bendahara bagian menyerahkan laporan rekapitulasi tunjangan daerah bagiannya untuk membuat surat permintaan pembayaran dan mengajukan pencairan tunjangan daerah kepada kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi tunjangan daerah dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi tunjangan daerah, BPKAD akan mentransfer uang sesuai rekapitulasi tunjangan daerah ke rekening biro. Bendahara biro akan menunggu hingga tunjangan daerah masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana tunjangan daerah ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer tunjangan daerah ke masing-masing pegawai. Pegawai akan menunggu hingga tunjangan daerah masuk ke rekening mereka masing-masing, kemudian pegawai dapat mengambil tunjangan daerah mereka selama satu bulan.



Gambar 4. 11 Proses Bisnis Pencairan Tunjangan Daerah (as-is)

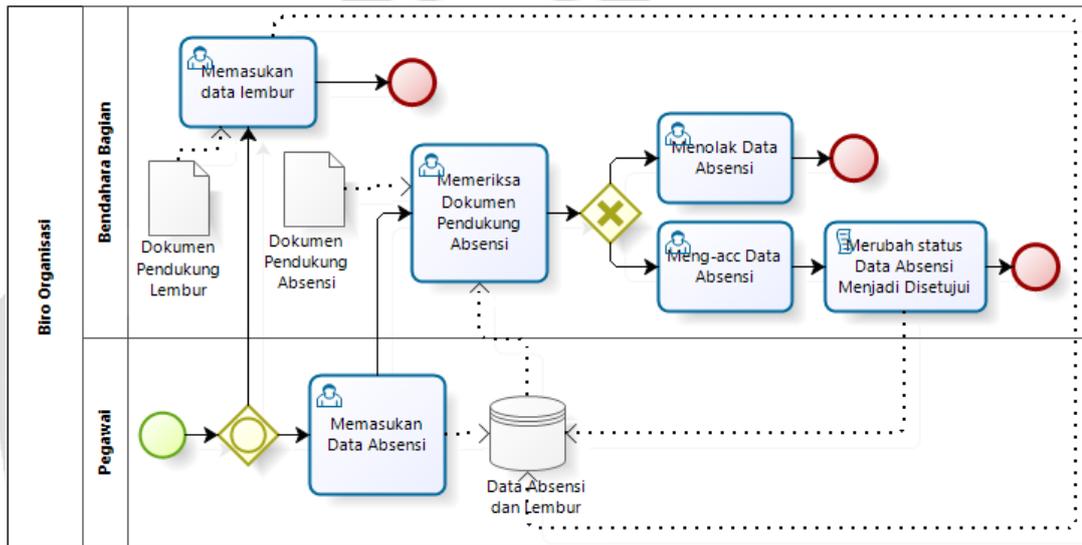
Proses bisnis usulan mengenai pencairan tunjangan daerah yang ada di Biro organisasi (*to-be*) dapat dilihat pada Gambar 4.12. Proses bisnis dimulai ketika bendahara biro mengupload laporan presensi pegawai ke sistem informasi keuangan pegawai. Kemudian sistem akan secara otomatis menghitung tunjangan daerah pegawai dan memasukkan datanya ke *database*. Kemudian bendahara bagian akan membuat laporan rekapitulasi tunjangan daerah dengan mencetak laporan rekapitulasi tunjangan daerah menggunakan data yang ada pada sistem. Setelah bendahara bagian membuat laporan rekapitulasi tunjangan daerah, pegawai akan menandatangani laporan rekapitulasi tersebut yang dilanjutkan dengan penandatanganan laporan oleh bendahara bagian kemudian kepala bagian. Setelah itu Laporan rekapitulasi tunjangan daerah akan diserahkan kepada bendahara biro untuk dilakukan pengecekan. Pengecekan dilakukan oleh bendahara biro dengan mencocokkan nominal pada laporan bendahara bagian dengan nominal pada sistem, sehingga proses pengecekan tidak perlu dilakukan secara manual dan dapat diselesaikan lebih cepat. Bendahara biro akan menunggu seluruh bendahara bagian menyerahkan laporan rekapitulasi tunjangan daerah bagiannya untuk membuat surat permintaan pembayaran dan mengajukan pencairan tunjangan daerah kepada kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi tunjangan daerah dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi tunjangan daerah, BPKAD akan mentransfer uang sesuai rekapitulasi tunjangan daerah ke rekening biro. Bendahara biro akan menunggu hingga tunjangan daerah masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana tunjangan daerah ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer tunjangan daerah ke masing-masing pegawai. Pegawai akan menunggu hingga tunjangan daerah masuk ke rekening mereka masing-masing, kemudian pegawai dapat mengambil tunjangan daerah mereka selama satu bulan.



Gambar 4. 12 Proses Bisnis Pencairan Tunjangan Daerah (to-be)

4.1.7 Proses Bisnis *Recording* Absensi dan Lembur Pegawai

Rancangan proses bisnis *recording* absensi pegawai setelah diterapkannya sistem informasi keuangan pegawai (*to-be*) dapat dilihat pada gambar 4.13. sebelumnya tidak ada proses pencatatan absensi, lembur ataupun perjalanan dinas pegawai, sehingga saat melakukan rekapitulasi keuangan pegawai pada akhir bulan sering mengalami kesulitan. Oleh karena itu dilakukan pemodelan proses bisnis usulan untuk pencatatan absensi dan Lembur pegawai. Proses bisnis dimulai dengan proses input data absensi atau lembur oleh pegawai, kemudian bendahara bagian akan mengecek dokumen pendukung absensi atau lembur (seperti surat perintah lembur). Apabila dokumen pendukung sesuai, maka data absensi atau lembur akan diinputkan ke sistem.



Gambar 4. 13 Proses Bisnis *Recording* Absensi dan Lembur (*to-be*)

4.2 Analisis Persyaratan

4.2.1 Identifikasi Tipe Pemangku Kepentingan

Identifikasi tipe pemangku kepentingan dilakukan untuk mengetahui dan mengelompokkan tipe pemangku kepentingan terkait sistem keuangan pegawai yang akan dikembangkan. Pengelompokan tipe pemangku kepentingan diperoleh dengan menganalisis hasil wawancara dengan bendahara Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur yaitu bapak Edi Hartono. Hasil identifikasi tipe pemangku kepentingan dapat dilihat pada Tabel 4.1. Hasil identifikasi tipe pemangku kepentingan akan dijadikan bahan pertimbangan dalam mengidentifikasi permasalahan pemangku kepentingan pada organisasi.

Tabel 4. 1 Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna	pemangku kepentingan yang secara langsung menggunakan atau berinteraksi dengan sistem.	Bendahara Biro
		Bendahara Bagian
		Pegawai
Pengembang	pemangku kepentingan yang terlibat dalam kegiatan produksi atau mendukung pengembangan sistem.	Peneliti
Pihak Berwenang	Pemangku kepentingan yang memiliki kemampuan untuk mendukung pengembangan dengan memberikan informasi mengenai regulasi yang berjalan agar sistem yang dikembangkan dapat berjalan sesuai dengan solusi yang berlaku.	Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur

4.2.2 Analisis Permasalahan

Analisis permasalahan dilakukan untuk mengetahui masalah apa yang ingin dipecahkan dengan solusi tertentu, dalam hal ini membangun sebuah Sistem Informasi Keuangan Pegawai di Biro Organisasi Sekretariat Daerah Profinsi Jawa Timur. Hal ini akan membantu untuk memastikan bahwa fungsi yang disediakan oleh sistem informasi keuangan pegawai secara langsung berkontribusi untuk mengatasi permasalahan yang ada di Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur. *The problem of* merupakan permasalahan yang dihadapi, *affects* merupakan siapa saja yang terkena dampak dari permasalahan, *the impact of which is* merupakan dampak permasalahan, dan *a successful solution would* merupakan syarat solusi yang ditawarkan akan dikatakan berhasil apabila memenuhi persyaratan. Pernyataan masalah (*problem statement*) yang diperoleh dari analisis hasil wawancara dengan bendahara Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur dapat dilihat pada Tabel 4.2 untuk *problem statement 1*, dan Tabel 4.3 untuk *problem statement 2*.

Tabel 4. 2 *Problem Statement 1*

<i>The problem of</i>	<ol style="list-style-type: none"> 1. Kesalahan input pada saat membuat laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi, atau tunjangan daerah. 2. Kesalahan perhitungan rekapitulasi uang makan, uang lembur, uang sidang, tunjangan prestasi, atau tunjangan daerah.
------------------------------	---

Tabel 4.2 Problem Statement 1 (Lanjutan)

	3. Ketidaksesuaian presensi pegawai yang digunakan sebagai dasar membuat laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi, atau tunjangan daerah.
Affects	Bendahara biro, bendahara bagian, pegawai
The impact of which is	<ol style="list-style-type: none"> 1. Pencairan uang makan, uang lembur, uang sidang tunjangan prestasi, tunjangan daerah atau uang perjalanan dinas menjadi lebih lama (terlambat). 2. Uang makan, uang lembur, uang sidang tunjangan prestasi, tunjangan daerah atau uang perjalanan dinas yang diterima pegawai menjadi tidak sesuai dengan yang seharusnya diterima.
A successful solution would	<ol style="list-style-type: none"> 1. Tidak ada atau mengurangi kesalahan data pegawai pada laporan rekapitulasi uang makan, uang lembur, uang sidang tunjangan prestasi, tunjangan daerah atau uang perjalanan dinas. 2. Tidak ada atau mengurangi kesalahan perhitungan uang makan, uang lembur, uang sidang tunjangan prestasi, tunjangan daerah atau uang perjalanan dinas. 3. Tidak ada data presensi yang tidak sesuai.

Tabel 4. 3 Problem Statement 2

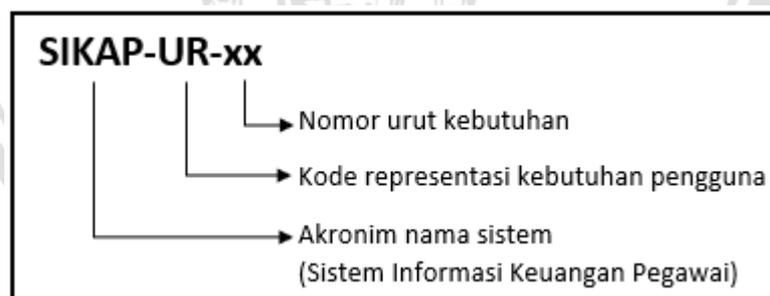
The problem of	<ol style="list-style-type: none"> 1. Kesalahan input pada saat membuat kwitansi uang perjalanan dinas. 2. Kesalahan perhitungan uang perjalanan dinas. 3. Ketidaksesuaian SPJ perjalanan dinas dengan ketentuan uang perjalanan dinas. 4. Tidak ada pencatatan perjalanan dinas.
Affects	Bendahara biro, pegawai
The impact of which is	<ol style="list-style-type: none"> 1. Data perjalanan dinas seperti SPT dengan SPPD menjadi tidak sesuai. 2. Uang perjalanan dinas yang diterima pegawai menjadi tidak sesuai dengan yang seharusnya diterima. 3. Adanya pegawai yang mengikuti sidang dan mendapat uang sidang serta uang perjalanan dinas sidang pada hari mereka seharusnya melaksanakan perjalanan dinas.

Tabel 4.3 Problem Statement 2 (Lanjutan)

	<ol style="list-style-type: none"> 4. Adanya pegawai yang mendapat uang makan serta uang perjalanan dinas sidang pada hari mereka seharusnya melaksanakan perjalanan dinas di luar kota. 5. Adanya pegawai yang melakukan perjalanan dinas lebih dari satu kali pada hari yang sama.
A successful solution would	<ol style="list-style-type: none"> 1. Sesuaiinya data perjalanan dinas. 2. Uang perjalanan dinas yang diterima pegawai sesuai dengan ketentuan yang telah ditetapkan. 3. Tidak ada pegawai yang mengikuti dan mendapat uang sidang serta perjalanan dinas pada hari yang sama. 4. Tidak ada pegawai yang mendapat uang makan serta perjalanan dinas pada saat melakukan perjalanan dinas luar kota. 5. Tidak ada pegawai yang melakukan perjalanan dinas lebih dari satu kali pada hari yang sama

4.2.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Kebutuhan pengguna diperoleh dari analisis hasil wawancara dan pernyataan masalah. Kebutuhan pengguna berkaitan dengan permasalahan yang diperoleh dari *problem statement* pada tahap analisis permasalahan. Berdasarkan permasalahan yang dihadapi oleh pemangku kepentingan akan diidentifikasi kebutuhan pengguna untuk mengatasi permasalahan tersebut. Masing-masing kebutuhan pengguna yang teridentifikasi akan diberikan kode sebagai identitas untuk mempermudah pelacakan nantinya. Keterangan kodefikasi kebutuhan pengguna dapat dilihat pada Gambar 4.14.



Gambar 4. 14 Keterangan Kodefikasi Kebutuhan Pengguna

Hasil identifikasi kebutuhan pengguna dapat dilihat pada Tabel 4.4. hasil identifikasi kebutuhan pengguna ini nantinya akan dianalisis untuk melakukan identifikasi fitur. Kode kebutuhan merupakan kode atau identitas kebutuhan sesuai dengan kodefikasi pada Gambar 4.14. Kolom kebutuhan pengguna merupakan kebutuhan yang teridentifikasi. Kolom pemangku kepentingan merupakan pemangku kepentingan terkait dengan kebutuhan tersebut. Kolom

situasi saat ini merupakan keadaan yang ada di organisasi terkait dengan pemenuhan kebutuhan pengguna tersebut. Kolom solusi yang ditawarkan merupakan solusi yang ditawarkan untuk memenuhi kebutuhan pengguna.

Tabel 4. 4 Hasil Identifikasi Kebutuhan Pengguna

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan
SIKAP-UR-1	Menghitung laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi dan tunjangan daerah sesuai ketentuan	Bendahara biro, bendahara bagian	Pembuatan laporan rekapitulasi dilakukan secara manual	Sistem informasi yang menyediakan layanan perhitungan rekapitulasi secara otomatis sesuai ketentuan yang ditetapkan.
SIKAP-UR-2	Dapat mengelola presensi pegawai	Bendahara bagian, bendahara biro	Presensi pegawai dilihat dari data presensi fingerprint yang hanya berisi tanggal, waktu datang dan pulang pegawai	Sistem informasi yang menyediakan layanan untuk mengelola data presensi pegawai
SIKAP-UR-3	Dapat mencatat atau melakukan <i>recording</i> absensi pegawai	Bendahara bagian, bendahara biro, pegawai	Tidak ada pencatatan absensi pegawai	Sistem informasi yang menyediakan layanan untuk mengelola data absensi seperti pencatatan absensi dan perjalanan dinas
SIKAP-UR-4	Dapat melakukan pencatatan perjalanan dinas pegawai	Bendahara biro, bendahara bagian	Tidak ada pencatatan perjalanan dinas pegawai	Sistem informasi yang menyediakan layanan pencatatan atau <i>recording</i> perjalanan dinas

Tabel 4.4 Hasil Identifikasi Kebutuhan Pengguna (Lanjutan)

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan
SIKAP-UR-5	Melakukan perhitungan dan pengecekan uang perjalanan dinas pegawai	Bendahara biro	Dilakukan manual oleh bendahara biro	Sistem informasi yang menyediakan layanan perhitungan dan pengecekan spj serta pembuatan kwitansi perjalanan dinas sesuai ketentuan.
SIKAP-UR-6	Membuat dokumen persyaratan perjalanan dinas yang sesuai	Bendahara biro	Dokumen seperti SPT dan SPPD dibuat secara manual	Sistem informasi yang menyediakan layanan untuk membuat dokumen persyaratan perjalanan dinas
SIKAP-UR-7	Membuat rekapitulasi uang sidang yang sesuai dengan ketentuan.	Bendahara biro	Rekapitulasi uang sidang dibuat secara manual oleh bendahara biro	Sistem informasi yang menyediakan layanan untuk membuat rekapitulasi uang sidang sesuai dengan ketentuan
SIKAP-UR-8	Informasi yang ada hanya boleh diakses oleh yang berhak	Bendahara biro, bendahara bagian, pegawai	-	Sistem informasi yang menyediakan layanan untuk membatasi hak akses pengguna

Tabel 4.4 Hasil Identifikasi Kebutuhan Pengguna (Lanjutan)

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan
SIKAP-UR-9	Mengotrol perjalanan dinas dan presesnsi pegawai sesuai ketentuan	Bendahara biro, bendahara bagian	-	Sistem informasi menyediakan layanan untuk mencatat perjalanan dinas dan presensi pegawai berkaitan dengan keuangan pegawai.

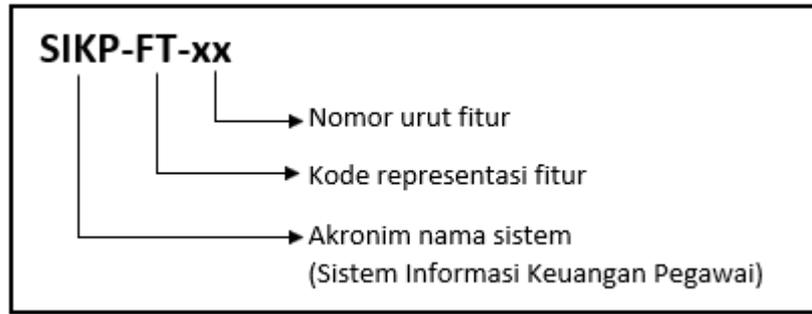
4.2.4 Identifikasi Pengguna

Identifikasi pengguna dilakukan untuk mengetahui siapa saja yang akan berinteraksi atau menggunakan sistem yang akan dibangun. Identifikasi pengguna diperoleh dari pemangku kepentingan yang masuk kelompok tipe pemangku kepentingan pengguna. Hasil identifikasi pengguna ada pada Tabel 4.5. Hasil identifikasi pengguna nantinya akan digunakan untuk identifikasi aktor dalam pemodelan *use case*. Kolom tipe pemangku kepentingan merupakan tipe pemangku kepentingan dari pengguna yang akan menggunakan sistem pada kolom tipe pengguna. Kolom deskripsi merupakan deskripsi dari pengguna.

Tabel 4. 5 Hasil Identifikasi Pengguna

Tipe pemangku kepentingan	Tipe Pengguna	Deskripsi
pengguna	Bendahara Biro	Bendahara biro bertugas dan bertanggung jawab untuk mengelola uang perjalanan dinas, uang sidang, dokumen perjalanan dinas serta melakukan pencairan uang makan, uang lembur, tunjangan prestasi, tunjangan daerah, uang sidang, dan uang perjalanan dinas pegawai sesuai ketentuan yang telah ditetapkan
	Bendahara Bagian	Bendahara bagian bertugas dan bertanggung jawab untuk mengelola dan membuat laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi dan tunjangan daerah sesuai ketentuan yang telah ditetapkan. Serta mengelola absensi dan data lembur pegawai.
	Pegawai	Pegawai biro bertugas memasukkan data absensi masing-masing serta membuat laporan perjalanan dinas

4.2.5 Identifikasi Fitur



Gambar 4. 15 Kodefikasi Fitur

Gambar 4.15 menunjukkan kodefikasi fitur. Pada hasil identifikasi fitur, setiap fitur akan diberi kode sebagai identitas. Pemberian identitas berupa kode fitur untuk mempermudah pelacakan fitur. Hasil identifikasi fitur dapat dilihat pada Tabel 4.6. Kolom kode fitur merupakan kode atau identitas fitur sesuai dengan kodefikasi pada Gambar 4.15. Sedangkan kolom deskripsi merupakan deskripsi dari fitur.

Tabel 4. 6 Hasil Identifikasi Fitur

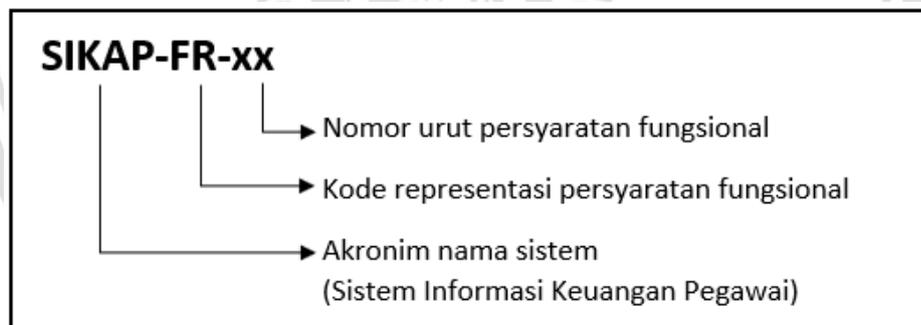
Kode Fitur	Deskripsi
SIKAP-FT-1	Sistem dapat mengenali identitas pengguna serta membatasi akses pengguna sesuai dengan hak yang dapat diakses oleh pengguna tersebut untuk mengelola atau melihat informasi (otentikasi).
SIKAP-FT-2	Sistem dapat mengelola data laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi dan tunjangan daerah yang terdiri dari data pegawai, data lembur, data presensi dan absensi pegawai serta data ketentuannya. Sistem juga dapat melakukan pencetakan laporan rekapitulasi keuangan tersebut.
SIKAP-FT-3	Sistem dapat mengelola data perjalanan dinas pegawai seperti data lembar disposisi, SPT dan SPPD serta melakukan pencetakan.
SIKAP-FT-4	Sistem dapat melakukan perhitungan dan pengecekan uang perjalanan dinas pegawai agar sesuai dengan ketentuan yang berlaku. Seperti pengecekan SPJ berdasarkan ketentuan serta melakukan pencetakan kwitansi uang perjalanan dinas sesuai SPJ.
SIKAP-FT-5	Sistem dapat mengelola data uang sidang seperti data bukti sidang dan ketentuan uang sidang serta dapat melakukan perhitungan dan pencetakan laporan rekapitulasi uang sidang.
SIKAP-FT-6	Sistem dapat melakukan kontrol terhadap kemungkinan adanya pegawai yang melakukan perjalanan dinas lebih dari satu kali pada hari yang sama, melakukan presensi, serta mendapatkan uang makan saat melakukan perjalanan dinas.

Fitur-fitur yang diidentifikasi berkaitan dengan pemenuhan kebutuhan pengguna yang diperoleh dari hasil identifikasi kebutuhan pengguna. Fitur merupakan solusi dari kebutuhan tertentu. Oleh karena itu akan dilakukan pengelompokan masing-masing fitur berdasarkan kebutuhan pengguna yang harus dipenuhi. Pengelompokan dan hubungan fitur dengan kebutuhan pengguna dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Hubungan fitur dengan kebutuhan pengguna

Kebutuhan Pengguna	Fitur
SIKAP-UR-8	SIKAP-FT-1
SIKAP-UR-1	SIKAP-FT-2
SIKAP-UR-2	
SIKAP-UR-3	
SIKAP-UR-4	
SIKAP-UR-6	SIKAP-FT-3
SIKAP-UR-5	SIKAP-FT-4
SIKAP-UR-7	SIKAP-FT-5
SIKAP-UR-9	SIKAP-FT-6

4.2.6 Persyaratan Fungsional



Gambar 4. 16 Kodefikasi Persyaratan Fungsional

Gambar 4.16 menunjukkan kodefikasi persyaratan fungsional. Masing-masing persyaratan fungsional memiliki kode sebagai identitas. Pemberian identitas berupa kode persyaratan fungsional untuk mempermudah pelacakan persyaratan fungsional. Persyaratan fungsional merupakan kemampuan atau tindakan yang harus dimiliki atau dapat dilakukan oleh sistem agar fitur sistem dapat berjalan, sehingga sistem dapat memenuhi kebutuhan pengguna. Tabel 4.8 menunjukkan hubungan antara persyaratan fungsional dengan fitur dari sistem yang akan dibangun. Kolom kode fitur merupakan kode atau identitas fitur yang berhasil diidentifikasi pada identifikasi fitur terkait dengan persyaratan fungsional yang

memiliki kode atau identitas pada kolom kode persyaratan fungsional. Kolom deskripsi merupakan deskripsi dari persyaratan fungsional.

Tabel 4. 8 Persyaratan Fungsional

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
SIKAP-FT-1	SIKAP-FR-1	Sistem dapat menampilkan halaman otentikasi
	SIKAP-FR-2	Sistem dapat mengotentikasi pengguna yang akan masuk ke sistem sesuai data pengguna
	SIKAP-FR-3	Sistem dapat mengganti informasi session sesuai dengan pengguna yang sedang aktif terkait dengan hak akses pengguna
	SIKAP-FR-4	Sistem dapat menghapus informasi pada session ketika pengguna sedang tidak aktif
SIKAP-FT-2	SIKAP-FR-5	Sistem dapat menampilkan antarmuka berkaitan dengan pengelolaan data pegawai
	SIKAP-FR-6	Sistem dapat menampilkan antarmuka berkaitan dengan pengelolaan data absensi dan presensi
	SIKAP-FR-7	Sistem dapat menampilkan antarmuka berkaitan dengan pembuatan laporan rekapitulasi uang makan
	SIKAP-FR-8	Sistem dapat menentukan dan menghitung uang makan serta total uang makan pegawai berdasarkan ketentuan uang makan
	SIKAP-FR-9	Sistem dapat menampilkan antarmuka berkaitan dengan pembuatan surat permintaan pembayaran uang makan sesuai data rekapitulasi uang makan
	SIKAP-FR-10	Sistem dapat menampilkan antar muka untuk mengelola data lembur pegawai
	SIKAP-FR-11	Sistem dapat menampilkan antarmuka berkaitan dengan pembuatan data laporan rekapitulasi uang lembur
	SIKAP-FR-12	Sistem dapat menentukan dan menghitung uang lembur serta total uang lembur pegawai berdasarkan ketentuan uang lembur
	SIKAP-FR-13	Sistem dapat menampilkan antarmuka berkaitan dengan pembuatan surat permintaan pembayaran uang lembur sesuai data rekapitulasi uang lembur

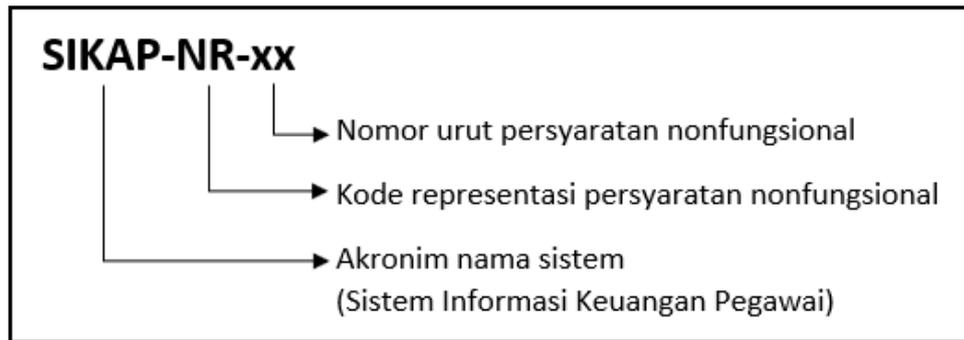
Tabel 4.8 Persyaratan Fungsional (Lanjutan)

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
SIKAP-FT-2	SIKAP-FR-14	Sistem dapat menampilkan antarmuka berkaitan dengan pembuatan laporan rekapitulasi tunjangan prestasi
	SIKAP-FR-15	Sistem dapat menentukan dan menghitung tunjangan prestasi serta total tunjangan prestasi pegawai berdasarkan ketentuan
	SIKAP-FR-16	Sistem dapat menentukan dan menghitung potongan tunjangan prestasi berdasarkan absensi atau keterlambatan pegawai serta melakukan pemotongan terhadap tunjangan prestasi pegawai
	SIKAP-FR-17	Sistem dapat menampilkan antarmuka berkaitan dengan pembuatan surat permintaan pembayaran tunjangan prestasi sesuai data rekapitulasi tunjangan prestasi
	SIKAP-FR-18	Sistem dapat menampilkan antarmuka berkaitan dengan pembuatan laporan rekapitulasi tunjangan daerah
	SIKAP-FR-19	Sistem dapat menampilkan antarmuka berkaitan dengan pembuatan surat permintaan pembayaran tunjangan daerah sesuai data rekapitulasi tunjangan daerah
	SIKAP-FR-20	Sistem dapat menentukan dan menghitung tunjangan daerah serta total tunjangan daerah pegawai berdasarkan ketentuan tunjangan daerah
	SIKAP-FR-21	Sistem dapat menampilkan antarmuka berkaitan dengan pengelolaan data libur nasional
SIKAP-FT-3	SIKAP-FR-22	Sistem dapat menampilkan formulir untuk memasukkan data perjalanan dinas dan melakukan pencetakan SPT
	SIKAP-FR-23	Sistem dapat menampilkan data pegawai pada SPT sesuai nama pegawai
	SIKAP-FR-24	Sistem dapat menampilkan antarmuka berkaitan dengan pengelolaan data perjalanan dinas
	SIKAP-FR-25	Sistem dapat mengaitkan antara SPT dan SPPD

Tabel 4.8 Persyaratan Fungsional (Lanjutan)

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
SIKAP-FT-3	SIKAP-FR-26	Sistem dapat menampilkan antarmuka berkaitan dengan pengelolaan dan pencetakan data SPPD
SIKAP-FT-4	SIKAP-FR-27	Sistem dapat menentukan uang harian pegawai berdasarkan pangkat, golongan, waktu dan tempat perjalanan dinas
	SIKAP-FR-28	Sistem dapat menentukan biaya penginapan untuk pegawai yang melakukan perjalanan dinas berdasarkan pangkat, golongan, waktu dan tempat perjalanan dinas
	SIKAP-FR-29	Sistem dapat menentukan biaya sewa kendaraan untuk pegawai yang melakukan perjalanan dinas sesuai dengan ketentuan berdasarkan pangkat, golongan, waktu dan tempat perjalanan dinas
	SIKAP-FR-30	Sistem dapat menentukan biaya transport untuk pegawai yang melakukan perjalanan dinas berdasarkan pangkat, golongan, waktu dan tempat perjalanan dinas
	SIKAP-FR-31	Sistem dapat melakukan pengecekan terhadap kesesuaian SPJ pegawai dengan ketentuan uang perjalanan dinas
	SIKAP-FR-32	Sistem dapat memberikan pemberitahuan apabila ada data SPJ yang tidak sesuai dengan ketentuan uang perjalanan dinas
	SIKAP-FR-33	Sistem dapat menampilkan antarmuka untuk membuat kwitansi uang perjalanan dinas
	SIKAP-FR-34	Sistem dapat menghitung uang perjalanan dinas pegawai sebagai data kwitansi
SIKAP-FT-5	SIKAP-FR-35	Sistem dapat menentukan dan menghitung uang sidang masing-masing pegawai terlibat dan menghitung total uang sidang pada suatu sidang
	SIKAP-FR-36	Sistem dapat menampilkan antarmuka untuk mencetak daftar penerima dan kwitansi uang sidang
SIKAP-FT-6	SIKAP-FR-37	Sistem dapat menampilkan antarmuka untuk memasukkan data sidang

4.2.7 Persyaratan Nonfungsional



Gambar 4. 17 Kodefikasi Persyaratan Nonfungsional

Gambar 4.17 menunjukkan kodefikasi persyaratan nonfungsional. Pada hasil identifikasi persyaratan nonfungsional, setiap persyaratan nonfungsional akan diberi kode sebagai identitas. Pemberian identitas berupa kode persyaratan nonfungsional untuk mempermudah pelacakan persyaratan nonfungsional. Persyaratan nonfungsional merupakan kualitas atau batasan-batasan yang harus dipenuhi oleh sistem. Tabel 4.9 menunjukkan persyaratan fungsional yang teridentifikasi. Kolom kode fitur merupakan kode atau identitas fitur yang berhasil diidentifikasi pada identifikasi fitur terkait dengan persyaratan nonfungsional yang memiliki kode atau identitas pada kolom kode persyaratan nonfungsional. Kolom deskripsi merupakan deskripsi dari persyaratan nonfungsional.

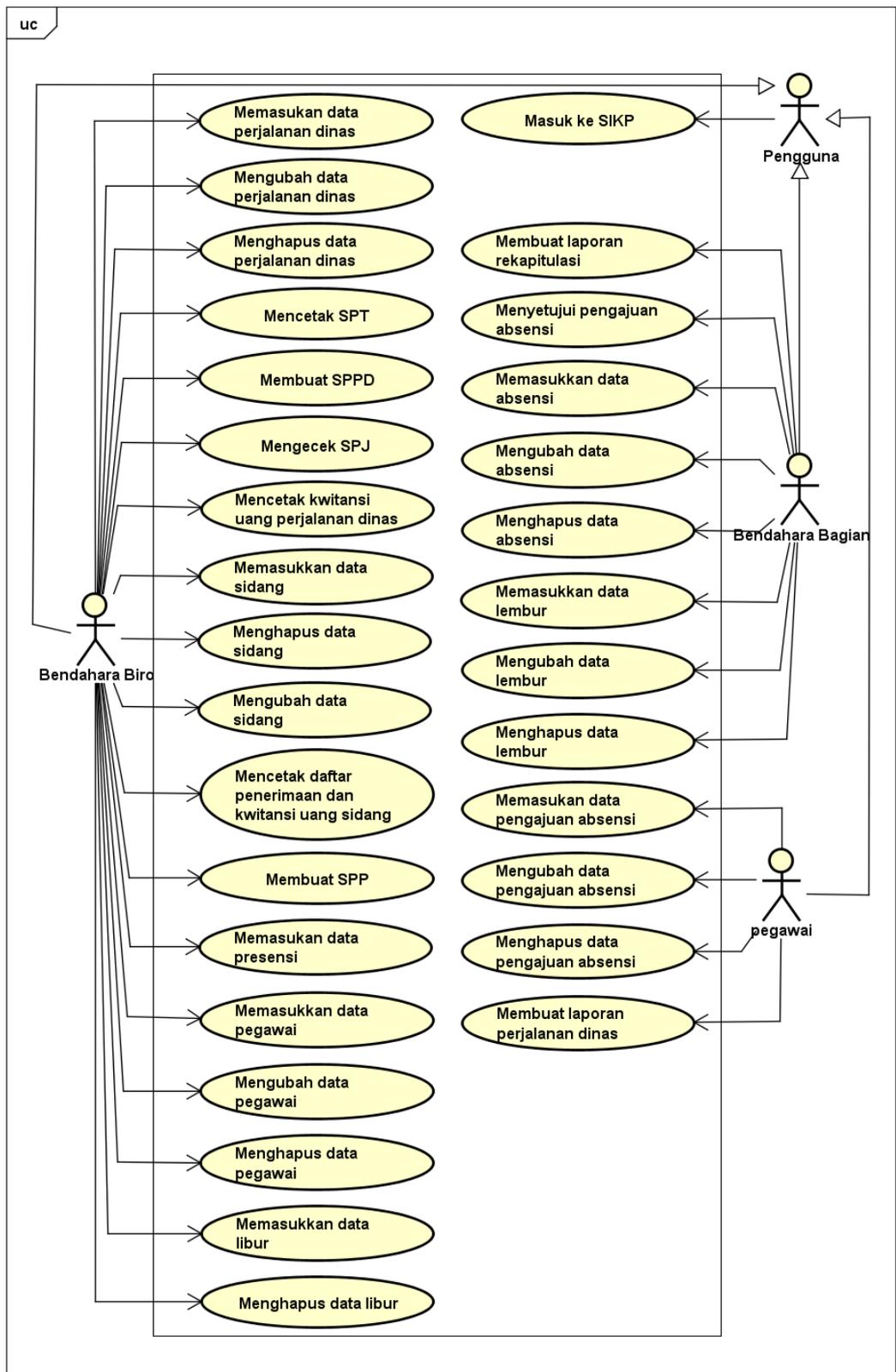
Tabel 4. 9 Persyaratan Nonfungsional

Kode Fitur	Kode Persyaratan Nonfungsional	Deskripsi
SIKAP-FT-1	SIKAP-NR-1	Sistem dapat berjalan pada beberapa <i>web browser</i> yang berbeda tanpa ada konten yang hilang
SIKAP-FT-2		
SIKAP-FT-3		
SIKAP-FT-4		
SIKAP-FT-5		
SIKAP-FT-6		

4.3 Pemodelan Use Case

Setelah mengidentifikasi kebutuhan fungsional dan nonfungsional, dilakukan pemodelan *use case* untuk mengetahui aktor dan *use case* serta menspesifikasikan *use case*. Pada pemodelan *use case* dilakukan identifikasi aktor, *use case*, dan spesifikasi masing-masing *use case*. Identifikasi aktor dilakukan untuk mengetahui siapa saja yang akan berinteraksi dengan sistem dari eksternal sistem. Identifikasi *use case* dilakukan untuk mengetahui tujuan dari aktor berinteraksi dengan sistem yang akan dibangun. Sedangkan spesifikasi *use case* dilakukan untuk membuat dokumentasi mengenai suatu *use case*.

4.3.1 Use Case Diagram



Gambar 4. 18 Use Case Diagram Sistem Informasi Keuangan Pegawai



Use case diagram digunakan untuk memodelkan bagaimana aktor dari eksternal sistem berinteraksi dengan sistem yang akan dibangun (memodelkan hubungan aktor dengan *use case*). Gambar 4.18 merupakan *use case diagram* dari Sistem Informasi Keuangan Pegawai yang akan dibangun. Identifikasi aktor dilakukan dengan mengelompokkan pengguna sistem kedalam beberapa peran. Setelah itu dilakukan identifikasi *use case* untuk menentukan tujuan aktor berinteraksi dengan sistem. Setelah *use case* teridentifikasi, *use case* akan dihubungkan dengan solusi yang ditawarkan dalam aktivitas pada proses bisnis solusi (*to-be*). Hubungan *use case* dengan aktivitas yang ada pada proses bisnis *to-be* dapat dilihat pada Tabel 4.10. Kolom *use case* merupakan *use case* yang terhubung dengan aktivitas tertentu (kolom aktivitas) pada suatu proses bisnis (kolom proses bisnis).

Tabel 4. 10 Hubungan Aktivitas Proses Bisnis (*to-be*) Dengan *Use Case*

Proses Bisnis	Aktivitas	<i>Use Case</i>
Pencairan uang sidang	Membuat rekapitulasi pembayaran uang sidang	Memasukkan data sidang
		Menghapus data sidang
		Mengubah data sidang
		Mencetak daftar penerima dan kwitansi uang sidang
Pencairan uang perjalanan dinas	Membuat SPT	Memasukan data perjalanan dinas
		Mengubah data perjalanan dinas
		Menghapus data perjalanan dinas
		Mencetak SPT
	Membuat SPPD	Membuat SPPD
	Memasukan data SPJ	Mengecek SPJ
	Membuat Kwitansi	Membuat Kwitansi perjalanan dinas
Pencairan uang makan	Memasukan data presensi pegawai	Memasukan data presensi pegawai
	Membuat laporan rekapitulasi uang makan	Membuat Laporan rekapitulasi
	Membuat surat permintaan pembayaran	Membuat surat permintaan pembayaran

Tabel 4.10 Hubungan Aktivitas Proses Bisnis (*to-be*) Dengan *Use Case* (Lanjutan)

Proses Bisnis	Aktivitas	Use Case
Pencairan Uang lembur	Memasukkan presensi pegawai	Memasukkan presensi pegawai
	Membuat laporan rekapitulasi uang lembur	Membuat laporan rekapitulasi
	Membuat surat permintaan pembayaran	Membuat surat permintaan pembayaran
Pencairan Tunjangan Prestasi	Memasukkan presensi pegawai	Memasukkan presensi pegawai
	Membuat laporan rekapitulasi tunjangan prestasi	Membuat laporan rekapitulasi
	Membuat surat permintaan pembayaran	Membuat surat permintaan pembayaran
Pencairan Tunjangan Daerah	Memasukkan presensi pegawai	Memasukkan presensi pegawai
	Membuat laporan rekapitulasi tunjangan daerah	Membuat laporan rekapitulasi
	Membuat surat permintaan pembayaran	Membuat surat permintaan pembayaran
Pencatatan absensi dan lembur	Memasukan data absensi atau lembur	Memasukan data pengajuan absensi
		Mengubah data pengajuan absensi
		Menghapus data pengajuan absensi
		Memasukkan data absensi
		Memasukkan data lembur
		Mengubah data lembur
		Menghapus data lembur
	Menyetujui pengajuan absensi	Menyetujui pengajuan absensi
	Menolak pengajuan absensi	Menyetujui pengajuan absensi

Kemudian *use case* akan dihubungkan dengan pemangku kepentingan untuk membantu pemangku kepentingan memahami apa saja yang dapat dilakukan pemangku kepentingan melalui sistem yang akan dibangun. Hubungan *use case* dengan pemangku kepentingan dapat dilihat pada Tabel 4.11. Kolom *use case* merupakan *use case* yang dilakukan oleh pengguna dalam kolom pengguna. Sedangkan kolom tipe pemangku kepentingan merupakan tipe pemangku kepentingan dari suatu pengguna.

Tabel 4. 11 Hubungan Use Case Dengan Pemangku Kepentingan

<i>Use Case</i>	Pengguna	Tipe Pemangku Kepentingan
Masuk ke SIKAP	Pengguna, pegawai, bendahara bagian, bendahara biro	Pengguna
Membuat laporan rekapitulasi	Bendahara Bagian	
Menyetujui pengajuan absensi		
Memasukkan data absensi		
Mengubah data absensi		
Menghapus data absensi		
Memasukkan data lembur		
Mengubah data lembur		
Menghapus data lembur		
Memasukan data perjalanan dinas	Bendahara Biro	
Mengubah data perjalanan dinas		
Menghapus data perjalanan dinas		
Mencetak SPT		
Membuat SPPD		
Mengecek SPJ		
Membuat kwitansi uang perjalanan dinas		
Memasukkan data sidang		
Menghapus data sidang		
Mengubah data sidang		
Mencetak daftar penerima dan kwitansi uang sidang		
Membuat SPP		

Tabel 4.11 Hubungan Use Case Dengan Pemangku Kepentingan (Lanjutan)

<i>Use Case</i>	Pengguna	Tipe Pemangku Kepentingan
Memasukan data presensi	Bendahara Biro	Pengguna
Memasukkan data pegawai		
Mengubah data pegawai		
Menghapus data pegawai		
Memasukkan data libur		
Menghapus data libur		
Memasukkan data pengajuan absensi	Pegawai	
Mengubah data pengajuan absensi		
Menghapus data pengajuan absensi		
Membuat laporan perjalanan dinas		

Kemudian masing-masing *use case* yang teridentifikasi akan dihubungkan dengan fitur untuk memastikan bahwa hasil *use case* yang teridentifikasi diperoleh dari hasil analisis persyaratan. Hubungan antara *use case* dengan fitur dapat dilihat pada Tabel 4.12.

Tabel 4. 12 Hubungan Use Case dengan Fitur

<i>Use Case</i>	Fitur
Masuk ke SIKAP	SIKAP-FT-1
Membuat laporan rekapitulasi	SIKAP-FT-2
Membuat surat permintaan pembayaran	
Memasukkan data presensi	SIKAP-FT-2, SIKAP-FT-6
Memasukan data pengajuan absensi	
Mengubah data pengajuan absensi	
Menghapus data pengajuan absensi	
Menyetujui pengajuan absensi	
Memasukkan data absensi	
Mengubah data absensi	
Menghapus data absensi	
Memasukkan data lembur	

Tabel 4.12 Hubungan *Use Case* dengan Fitur (Lanjutan)

<i>Use Case</i>	Fitur
Mengubah data lembur	SIKAP-FT-2, SIKAP-FT-6
Menghapus data lembur	
Memasukkan data libur	
Menghapus data libur	
Memasukan data perjalanan dinas	SIKAP-FT-3
Mengubah data perjalanan dinas	
Menghapus data perjalanan dinas	
Mencetak SPT	
Membuat SPPD	
Mengecek SPJ	SIKAP-FT-4
Membuat kwitansi uang perjalanan dinas	
Memasukkan data sidang	SIKAP-FT-5
Menghapus data sidang	
Mengubah data sidang	
Mencetak daftar penerima dan kwitansi uang sidang	
Memasukkan data pegawai	SIKAP-FT-2, SIKAP-FT-3, SIKAP-FT-4, SIKAP-FT-5
Mengubah data pegawai	
Menghapus data pegawai	

4.3.2 Deskripsi Aktor

Deskripsi aktor merupakan penjelasan mengenai peran dan tanggung jawab aktor. Tujuan aktor menggunakan sistem secara singkat juga dijelaskan pada deskripsi aktor. Deskripsi dari aktor yang telah teridentifikasi dapat dilihat pada Tabel 4.13. Kolom aktor merupakan peran dari sesuatu yang berinteraksi dengan sistem. Kolom deskripsi merupakan deskripsi dari aktor.

Tabel 4. 13 Deskripsi Aktor

Aktor	Deskripsi
Pengguna	Pengguna diperankan oleh seluruh pegawai biro organisasi, bendahara bagian, bendahara umum, dan kepala biro organisasi untuk mengakses dan mengelola informasi sesuai dengan hak aksesnya

Tabel 4.13 Deskripsi Aktor (Lanjutan)

Aktor	Deskripsi
Bendahara bagian	Bendahara bagian diperankan oleh bendahara masing-masing bagian yang ada di biro organisasi untuk mengelola laporan uang makan, uang lembur, tunjangan prestasi, dan tunjangan daerah serta mengelola data absensi dan data lembur pegawai
Bendahara Biro	Bendahara biro diperankan oleh bendahara biro organisasi untuk mengelola dokumen dan uang perjalanan dinas pegawai serta laporan rekapitulasi uang sidang pegawai
Pegawai	Pegawai diperankan oleh seluruh pegawai biro organisasi untuk mengelola data pengajuan absensi serta membuat laporan perjalanan dinas

4.3.3 Spesifikasi Use Case

Spesifikasi *use case* memuat informasi mengenai nama *use case* serta penjelasan singkat mengenai *use case* tersebut. Selain itu spesifikasi *use case* juga memuat informasi mengenai aktor yang terkait dengan *use case*. Kemudian informasi mengenai keadaan atau syarat yang harus dipenuhi untuk memulai suatu *use case* serta keadaan atau kondisi yang terpenuhi setelah *use case* dijalankan. Selain itu terdapat informasi mengenai alur yang biasanya terjadi ketika *use case* dijalankan, serta alur alternatif.

4.3.3.1 Spesifikasi Use Case Masuk ke SIKAP

Spesifikasi *use case* masuk ke SIKAP dapat dilihat pada Tabel 4.14.

Tabel 4. 14 Spesifikasi Use Case Masuk ke SIKAP

Name	Masuk ke SIKAP
Brief Description	Menjelaskan bagaimana aktor masuk ke dalam sistem melalui proses verifikasi identitas pengguna agar dapat mengakses dan mengelola informasi berdasarkan hak aksesnya.
Actor	Pengguna
Post-condition	Halaman login sudah tersedia
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memasukkan <i>username</i> dan <i>password</i> pengguna pada form <i>login</i>. 2. Aktor menekan tombol "<i>login</i>" untuk mengirimkan <i>username</i> dan <i>password</i> pengguna ke sistem 3. Sistem melakukan identifikasi terhadap identitas aktor

Tabel 4.14 Spesifikasi *Use Case* Masuk ke SIKAP (Lanjutan)

Basic flow	<p>{Identifikasi Pengguna}</p> <p>4. Sistem menyimpan informasi mengenai identitas aktor</p> <p>5. Sistem menampilkan informasi berdasarkan hak akses aktor</p> <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Kegagalan Identifikasi Pengguna</p> <p>Pada {Identifikasi Pengguna}, jika sistem tidak dapat mengidentifikasi pengguna, maka sistem akan menampilkan pesan bahwa <i>username</i> dan <i>password</i> yang dikirimkan oleh pengguna tidak sesuai dengan daftar pengguna yang dikenali sistem, kemudian <i>use case</i> selesai</p>
Pre-condition	Aktor berhasil dikenali oleh sistem dan dapat mengakses sistem sesuai hak aksesnya

4.3.3.2 Spesifikasi *Use Case* Memasukkan Data Pengajuan Absensi

Spesifikasi *use case* memasukkan data pengajuan absensi dapat dilihat pada Tabel 4.15.

Tabel 4. 15 Spesifikasi *Use Case* Memasukkan Data Pengajuan Absensi

Name	Memasukkan data pengajuan absensi
Brief Description	Menjelaskan bagaimana aktor memasukkan data pengajuan absensi
Actor	Pegawai
Pre-condition	<p>1. Aktor dikenali oleh sistem sebagai pegawai.</p> <p>2. Tersedia <i>form input</i> data pengajuan absensi</p>
Basic flow	<p>{Use Case Dimulai}</p> <p>1. <i>Use case</i> dimulai ketika aktor mengisi <i>form</i> pengajuan absensi berupa alasan absen, keterangan lain, lama absen, serta tanggal absen.</p> <p>2. Aktor mengajukan pengajuan absensi dengan menekan tombol “ajukan”.</p> <p>3. Sistem menyimpan data pengajuan absensi.</p> <p>{Menyimpan Data}</p> <p>4. Sistem menampilkan pesan bahwa data berhasil disimpan.</p> <p>5. <i>Use case</i> selesai.</p> <p>{Use Case Selesai}</p>

Tabel 4.15 Spesifikasi *Use Case* Memasukkan Data Pengajuan Absensi (Lanjutan)

Alternative flow	A1. Menangani Kegagalan Penyimpanan Data Pada {Menyimpan Data}, jika sistem gagal menyimpan data yang akan dimasukkan, sistem akan menampilkan pesan gagal melakukan penyimpanan data dan kembali menampilkan form pengajuan absensi. Kemudian <i>use case</i> selesai.
Post-condition	Data pengajuan absensi berhasil disimpan

4.3.3.3 Spesifikasi *Use Case* Mengubah Data Pengajuan Absensi

Spesifikasi *use case* mengubah data pengajuan absensi dapat dilihat pada Tabel 4.16.

Tabel 4. 16 Spesifikasi *Use Case* Mengubah Data Pengajuan Absensi

Name	Mengubah data pengajuan absensi
Brief Description	Menjelaskan bagaimana aktor mengubah data pengajuan absensi
Actor	Pegawai
Pre-condition	1. Aktor dikenali oleh sistem sebagai pegawai. 2. Tersedia halaman daftar pengajuan absensi.
Basic flow	{Use Case Dimulai} 1. <i>Use case</i> dimulai ketika aktor menekan tombol <i>edit</i> pada data pengajuan absensi yang ingin diubah. 2. Sistem akan menampilkan <i>form</i> untuk mengubah data pengajuan absensi. 3. Aktor mengisi <i>form</i> untuk mengubah data pengajuan absensi. 4. Aktor menekan tombol “simpan” untuk mengirim perubahan data kepada sistem. 5. Sistem akan menyimpan perubahan data pengajuan absensi. 6. <i>Use case</i> selesai {Use Case Selesai}
Alternative flow	-
Post-condition	Data pengajuan absensi berhasil diubah.

4.3.3.4 Spesifikasi *Use Case* Menghapus Data Pengajuan Absensi

Spesifikasi *use case* menghapus data pengajuan absensi dapat dilihat pada Tabel 4.17.

Tabel 4. 17 Spesifikasi *Use Case* Menghapus Data Pengajuan Absensi

Name	Menghapus data pengajuan absensi
Brief Description	Menjelaskan bagaimana aktor menghapus data pengajuan absensi
Actor	Pegawai
Pre-condition	1. Aktor dikenali oleh sistem sebagai pegawai. 2. Tersedia halaman daftar pengajuan absensi.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “hapus” pada data pengajuan absensi yang ingin dihapus. 2. Sistem akan menampilkan pertanyaan untuk melakukan validasi menghapus data kepada aktor. <p>{Validasi Perhapusan Data}</p> <ol style="list-style-type: none"> 3. Aktor memilih “ya” untuk memvalidasi penghapusan data. 4. Sistem akan menghapus data pengajuan. 5. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Pembatalan Penghapusan Data</p> <p>Pada {Validasi Penghapusan Data}, jika aktor memilih untuk membatalkan penghapusan data, maka sistem akan kembali menampilkan daftar pengajuan absensi sebelum dilakukan penghapusan. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data pengajuan absensi berhasil dihapus.

4.3.3.5 Spesifikasi *Use Case* Membuat Laporan Perjalanan Dinas

Spesifikasi *use case* membuat laporan perjalanan dinas dapat dilihat pada Tabel 4.18.

Tabel 4. 18 Spesifikasi *Use Case* Membuat Laporan Perjalanan Dinas

Name	Membuat laporan perjalanan dinas
Brief Description	Menjelaskan bagaimana aktor membuat laporan perjalanan dinas.

Tabel 4.18 Spesifikasi *Use Case* Membuat Laporan Perjalanan Dinas (Lanjutan)

Actor	Pegawai
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai pegawai. 2. Tersedia halaman daftar perjalanan dinas.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “detail” pada data perjalanan dinas. 2. Sistem akan menampilkan detail perjalanan dinas yaitu data perjalanan dinas, form laporan perjalanan dinas dan SPJ perjalanan dinas. 3. Aktor menekan tombol “Buat Laporan Perjalanan Dinas”. 4. Aktor mengisi <i>form</i> laporan perjalanan dinas berupa pejabat yang hadir, arahan, kendala, solusi, dan keterangan lain. 5. Aktor menekan tombol “Cetak” untuk mencetak laporan perjalanan dinas. 6. Sistem akan menyimpan data perjalanan dinas. 7. Sistem menerbitkan laporan perjananan dinas dalam bentuk file pdf. 8. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	-
Post-condition	Laporan perjalanan dinas berhasil dibuat.

4.3.3.6 Spesifikasi *Use Case* Membuat Laporan Rekapitulasi

Spesifikasi *use case* membuat laporan rekapitulasi dapat dilihat pada Tabel 4.19.

Tabel 4. 19 Spesifikasi *Use Case* Membuat Laporan Rekapitulasi

Name	Membuat laporan rekapitulasi
Brief Description	Menjelaskan bagaimana aktor membuat laporan rekapitulasi sesuai data rekapitulasi. Laporan rekapitulasi yang dapat dibuat yaitu laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi dan tunjangan daerah.
Actor	Bendahara bagian

Tabel 4.19 Spesifikasi *Use Case* Membuat Laporan Rekapitulasi (Lanjutan)

Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara bagian. 2. Tersedia halaman daftar laporan rekapitulasi yang ingin dibuat.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor memilih menu laporan rekapitulasi yang ingin dibuat, yaitu laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi atau tunjangan daerah. 2. Sistem menampilkan data laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi, atau tunjangan daerah yang pernah dibuat. 3. Aktor mengisi tahun dan bulan untuk mencetak laporan rekapitulasi kemudian menekan tombol “cetak”. 4. Aktor menekan tombol “cetak” pada data rekapitulasi tertentu untuk membuat laporan rekapitulasi. 5. Sistem akan menerbitkan laporan rekapitulasi dalam bentuk <i>file pdf</i>. 6. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	-
Post-condition	Laporan rekapitulasi berhasil dibuat yang diantaranya laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi, atau tunjangan daerah.

4.3.3.7 Spesifikasi *Use Case* Menyetujui Pengajuan Absensi

Spesifikasi *use case* menyetujui pengajuan absensi dapat dilihat pada Tabel 4.20.

Tabel 4. 20 Spesifikasi *Use Case* Menyetujui Pengajuan Absensi

Name	Menyetujui pengajuan absensi
Brief Description	Menjelaskan bagaimana aktor menyetujui pengajuan absensi pegawai.
Actor	Bendahara bagian
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara bagian. 2. Tersedia halaman daftar pengajuan absensi.

Tabel 4.20 Spesifikasi Use Case Menyetujui Pengajuan Absensi (Lanjutan)

Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “acc” untuk menyetujui pengajuan absensi. 2. Sistem menampilkan form validasi acc. <p>{Validasi Perubahan Data}</p> <ol style="list-style-type: none"> 3. Aktor memilih jenis absensi pegawai. 4. Aktor menekan tombol “acc” untuk menyetujui perubahan data. Dalam hal ini untuk menyetujui pengajuan absensi. 5. Sistem akan menyimpan perubahan status pada data pengajuan absensi dari “menunggu” menjadi “disetujui”. 6. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Pembatalan Perubahan Data</p> <p>Pada {Validasi Perubahan Data}, jika aktor memilih untuk membatalkan perubahan data, maka sistem akan menampilkan daftar absensi sebelum dilakukan perubahan data. Kemudian <i>use case</i> selesai.</p>
Post-condition	Pengajuan absensi berhasil disetujui.

4.3.3.8 Spesifikasi Use Case Menolak Pengajuan Absensi

Spesifikasi *use case* menolak pengajuan absensi dapat dilihat pada Tabel 4.21.

Tabel 4. 21 Spesifikasi Use Case Menolak Pengajuan Absensi

Name	Menolak pengajuan absensi
Brief Description	Menjelaskan bagaimana aktor menolak pengajuan absensi pegawai.
Actor	Bendahara bagian
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara bagian. 2. Tersedia halaman daftar pengajuan absensi.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “tolak” untuk menyetujui pengajuan absensi. 2. Sistem menampilkan <i>form</i> validasi penolakan absensi. <p>{Validasi Perubahan Data}</p>

Tabel 4.21 *Spesifikasi Use Case Menolak Pengajuan Absensi (Lanjutan)*

Basic flow	<p>3. Aktor menekan tombol “ya” untuk menyetujui perubahan data. Dalam hal ini untuk menolak pengajuan absensi.</p> <p>4. Sistem akan menyimpan perubahan status pada data pengajuan absensi dari “menunggu” menjadi “ditolak”.</p> <p>5. <i>Use case</i> selesai.</p> <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Pembatalan Perubahan Data</p> <p>Pada {Validasi Perubahan Data}, jika aktor memilih untuk membatalkan perubahan data, maka sistem akan menampilkan daftar absensi sebelum dilakukan perubahan data. Kemudian <i>use case</i> selesai.</p>
Post-condition	Pengajuan absensi berhasil ditolak.

4.3.3.9 *Spesifikasi Use Case Memasukkan Data Absensi*

Spesifikasi *use case* memasukkan data absensi dapat dilihat pada Tabel 4.22.

Tabel 4. 22 *Spesifikasi Use Case Memasukkan Data Absensi*

Name	Memasukkan data absensi
Brief Description	Menjelaskan bagaimana aktor memasukkan data absensi.
Actor	Bendahara Bagian
Pre-condition	<p>1. Aktor dikenali oleh sistem sebagai bendahara bagian.</p> <p>2. Tersedia <i>form</i> input data absensi.</p>
Basic flow	<p>{Use Case Dimulai}</p> <p>1. <i>Use case</i> dimulai ketika aktor mengisi <i>form</i> absensi berupa nama, nip, alasan absen, keterangan tambahan, lama absen, serta tanggal absen.</p> <p>2. Aktor meyimpan data absensi dengan menekan tombol “simpan”.</p> <p>3. Sistem menyimpan data absensi.</p> <p>{Menyimpan Data}</p> <p>4. Sistem menampilkan pesan bahwa data berhasil disimpan.</p> <p>5. <i>Use case</i> selesai.</p> <p>{Use Case Selesai}</p>

Tabel 4.22 Spesifikasi Use Case Memasukkan Data Absensi (Lanjutan)

Alternative flow	<p>A2. Menangani Kegagalan Penyimpanan Data</p> <p>Pada {Menyimpan Data}, jika sistem gagal menyimpan data yang akan dimasukkan, maka sistem akan menampilkan pesan bahwa gagal melakukan penyimpanan data dan kembali menampilkan <i>form</i> absensi. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data absensi berhasil disimpan

4.3.3.10 Spesifikasi Use Case Mengubah Data Absensi

Spesifikasi *use case* mengubah data absensi dapat dilihat pada Tabel 4.23.

Tabel 4. 23 Spesifikasi Use Case Mengubah Data Absensi

Name	Mengubah data absensi
Brief Description	Menjelaskan bagaimana aktor mengubah data absensi.
Actor	Bendahara bagian
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara bagian. 2. Tersedia halaman daftar absensi.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol edit pada data absensi yang ingin diubah. 2. Sistem akan menampilkan <i>form</i> untuk mengubah data absensi. 3. Aktor mengisi <i>form</i> untuk mengubah data absensi. 4. Aktor menekan tombol “simpan” untuk mengirim perubahan data kepada sistem. 5. Sistem akan menyimpan perubahan data absensi. 6. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	-
Post-condition	Data absensi berhasil diubah.

4.3.3.11 Spesifikasi Use Case Menghapus Data Absensi

Spesifikasi use case menghapus data absensi dapat dilihat pada Tabel 4.24.

Tabel 4. 24 Spesifikasi Use Case Menghapus Data Absensi

Name	Menghapus data absensi
Brief Description	Menjelaskan bagaimana aktor menghapus data absensi.
Actor	Bendahara bagian
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara bagian. 2. Tersedia halaman daftar absensi.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor menekan tombol “hapus” pada data absensi yang ingin dihapus. 2. Sistem akan menampilkan pertanyaan untuk melakukan validasi menghapus data kepada aktor. <p>{Validasi Penghapusan Data}</p> <ol style="list-style-type: none"> 3. Aktor memilih “ya” untuk memvalidasi penghapusan data. 4. Sistem akan menghapus data absensi. 5. Use case selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Pembatalan Penghapusan Data</p> <p>Pada {Validasi Penghapusan Data}, jika aktor memilih untuk membatalkan penghapusan data, maka sistem akan kembali menampilkan daftar absensi sebelum dilakukan penghapusan. Kemudian use case selesai.</p>
Post-condition	Data absensi berhasil dihapus.

4.3.3.12 Spesifikasi Use Case Memasukkan Data Lembur

Spesifikasi use case memasukkan data lembur dapat dilihat pada Tabel 4.25.

Tabel 4. 25 Spesifikasi Use Case Memasukkan Data Lembur

Name	Memasukkan data lembur
Brief Description	Menjelaskan bagaimana aktor memasukkan data lembur.
Actor	Bendahara bagian

Tabel 4.25 Spesifikasi *Use Case* Memasukkan Data Lembur (Lanjutan)

Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara bagian. 2. Tersedia <i>form input</i> data lembur.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor mengisi <i>form</i> lembur berupa tanggal, kegiatan lembur, serta pegawai yang mengikuti lembur. 2. Aktor meyimpan data lembur dengan menekan tombol “simpan”. 3. Sistem menyimpan data lembur. <p>{Menyimpan Data}</p> <ol style="list-style-type: none"> 4. Sistem menampilkan pesan bahwa data berhasil disimpan. 5. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Kegagalan Penyimpanan Data</p> <p>Pada {Menyimpan Data}, jika sistem gagal menyimpan data yang akan dimasukkan, maka sistem akan menampilkan pesan bahwa gagal melakukan penyimpanan data dan kembali menampilkan form lembur. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data lembur berhasil disimpan

4.3.3.13 Spesifikasi *Use Case* Mengubah Data Lembur

Spesifikasi *use case* mengubah data lembur dapat dilihat pada tabel 4.26.

Tabel 4. 26 Spesifikasi *Use Case* Mengubah Data Lembur

Name	Mengubah data lembur
Brief Description	Menjelaskan bagaimana aktor mengubah data lembur.
Actor	Bendahara bagian
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara bagian. 2. Tersedia halaman daftar lembur.

Tabel 4.26 Spesifikasi *Use Case* Mengubah Data Lembur (Lanjutan)

<p>Basic flow</p>	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol edit pada data lembur yang ingin diubah. 2. Sistem akan menampilkan <i>form</i> untuk mengubah data lembur. 3. Aktor mengisi <i>form</i> untuk mengubah data lembur. 4. Aktor menekan tombol “simpan” untuk mengirim perubahan data kepada sistem. 5. Sistem akan menyimpan perubahan data lembur. <p>{Menghapus Pegawai Lembur}</p> <ol style="list-style-type: none"> 6. Sistem Menghapus pegawai lembur. <p>{Menambahkan Pegawai Lembur}</p> <ol style="list-style-type: none"> 7. Aktor menekan tombol “Tambah pegawai”. 8. Sistem menampilkan <i>form input</i> data pegawai. 9. Aktor mengisi data nama dan nip pegawai. 10. Aktor menekan tombol “simpan”. 11. Sistem menyimpan data pegawai lembur. 12. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
<p>Alternative flow</p>	<p>A1. Tidak melakukan penambahan pegawai</p> <p>Pada {Menambah Pegawai Lembur}, jika aktor tidak melakukan penambahan pegawai maka aliran dilanjutkan ke {Use Case Selesai}. Kemudian <i>use case</i> selesai.</p> <p>A2. Tidak melakukan penghapusan pegawai</p> <p>Pada {Menghapus Pegawai Lembur}, jika aktor tidak melakukan penghapusan pegawai maka aliran dilanjutkan ke {Use Case Selesai}. Kemudian <i>use case</i> selesai.</p>
<p>Post-condition</p>	<p>Data lembur berhasil diubah.</p>

4.3.3.14 Spesifikasi Use Case Menghapus Data Lembur

Spesifikasi *use case* menghapus data lembur dapat dilihat pada Tabel 4.27.

Tabel 4. 27 Spesifikasi Use Case Menghapus Data Lembur

Name	Menghapus data lembur
Brief Description	Menjelaskan bagaimana aktor menghapus data lembur.
Actor	Bendahara bagian
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara bagian. 2. Tersedia halaman daftar lembur.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “hapus” pada data lembur yang ingin dihapus. 2. Sistem akan menampilkan pertanyaan untuk melakukan validasi menghapus data kepada aktor. <p>{Validasi Penghapusan Data}</p> <ol style="list-style-type: none"> 3. Aktor memilih “ya” untuk memvalidasi penghapusan data. 4. Sistem akan menghapus data lembur. 5. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Pembatalan Penghapusan Data</p> <p>Pada {Validasi Peghapusan Data}, jika aktor memilih untuk membatalkan penghapusan data, maka sistem akan kembali menampilkan daftar lembur sebelum dilakukan penghapusan. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data lembur berhasil dihapus.

4.3.3.15 Spesifikasi Use Case Memasukkan Data perjalanan dinas

Spesifikasi *use case* memasukkan data perjalanan dinas dapat dilihat pada Tabel 4.28.

Tabel 4. 28 Spesifikasi Use Case Memasukkan Data perjalanan dinas

Name	Memasukkan data perjalanan dinas
Brief Description	Menjelaskan bagaimana aktor memasukkan data perjalanan dinas.
Actor	Bendahara biro

Tabel 4.28 Spesifikasi *Use Case* Memasukkan Data perjalanan dinas (Lanjutan)

Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro 2. Tersedia halaman form input dinas.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor mengisi form dinas yang berupa no. spt, kegiatan, tempat, lama dinas, tanggal, dasar perjalanan dinas serta pegawai yang bertugas. 2. Aktor mengirim data perjalanan dinas ke sistem dengan menekan tombol “simpan”. 3. Sistem menyimpan data perjalanan dinas. <p>{Menyimpan Data}</p> <ol style="list-style-type: none"> 4. Sistem menampilkan pesan bahwa data perjalanan dinas berhasil disimpan. 5. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Kegagalan Penyimpanan Data</p> <p>Pada {Menyimpan Data}, jika sistem gagal menyimpan data yang akan dimasukkan yaitu data pegawai dinas yang telah tercatat melakukan perjalanan dinas pada tanggal yang sama, maka sistem akan menampilkan pesan bahwa gagal melakukan penyimpanan data pegawai dinas tersebut karena telah tercatat melakukan perjalanan dinas pada tanggal yang sama. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data perjalanan dinas berhasil dimasukkan.

4.3.3.16 Spesifikasi *Use Case* Mengubah Data perjalanan dinas

Spesifikasi *use case* mengubah data perjalanan dinas dapat dilihat pada Tabel 4.29.

Tabel 4. 29 Spesifikasi *Use Case* Mengubah Data perjalanan dinas

Name	Mengubah data dina
Brief Description	Menjelaskan bagaimana aktor mengubah data perjalanan dinas.
Actor	Bendahara biro
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro 2. Tersedia halaman daftar perjalanan dinas.

Tabel 4.29 Spesifikasi *Use Case* Mengubah Data perjalanan dinas (Lanjutan)

<p>Basic flow</p>	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol edit pada data perjalanan dinas yang ingin diubah. 2. Sistem akan menampilkan <i>form</i> untuk mengubah data perjalanan dinas. 3. Aktor mengisi <i>form</i> untuk mengubah data perjalanan dinas. 4. Aktor menekan tombol “simpan” untuk mengirim perubahan data kepada sistem. 5. Sistem akan menyimpan perubahan data perjalanan dinas. <p>{Menghapus Pegawai Dinas}</p> <ol style="list-style-type: none"> 6. Aktor menghapus pegawai dinas 7. Sistem Menghapus pegawai dinas. <p>{Menambahkan Pegawai Dinas}</p> <ol style="list-style-type: none"> 8. Aktor menekan tombol “Tambah pegawai”. 9. Sistem menampilkan form input data pegawai. 10. Aktor mengisi data nama dan nip pegawai. 11. Aktor menekan tombol “simpan”. 12. Sistem menyimpan data pegawai dinas. <p>{Menyimpan Data}</p> <ol style="list-style-type: none"> 13. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
<p>Alternative flow</p>	<p>A1. Menangani Kegagalan Penyimpanan Data</p> <p>Pada {Menyimpan Data}, jika sistem gagal menyimpan data yang akan dimasukkan yaitu data pegawai dinas yang telah tercatat melakukan perjalanan dinas pada tanggal yang sama, maka sistem akan menampilkan pesan bahwa gagal melakukan penyimpanan data pegawai dinas tersebut karena telah tercatat melakukan perjalanan dinas pada tanggal yang sama. Kemudian <i>use case</i> selesai.</p> <p>A2. Tidak melakukan penambahan pegawai</p> <p>Pada {Menambah Pegawai Dinas}, jika aktor tidak melakukan penambahan pegawai maka aliran dilanjutkan ke {Use Case Selesai}. Kemudian <i>use case</i> selesai.</p>

Tabel 4.29 Spesifikasi *Use Case* Mengubah Data perjalanan dinas (Lanjutan)

	<p>A3. Tidak melakukan penghapusan pegawai</p> <p>A4. Pada {Menghapus Pegawai Dinas}, jika aktor tidak melakukan penghapusan pegawai maka aliran dilanjutkan ke {Use Case Selesai}. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data perjalanan dinas berhasil diubah.

4.3.3.17 Spesifikasi *Use Case* Menghapus Data perjalanan dinas

Spesifikasi *use case* menghapus data perjalanan dinas dapat dilihat pada Tabel 4.30.

Tabel 4. 30 Spesifikasi *Use Case* Menghapus Data perjalanan dinas

Name	Menghapus data perjalanan dinas
Brief Description	Menjelaskan bagaimana aktor menghapus data perjalanan dinas perjalanan dinas.
Actor	Bendahara biro
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar perjalanan dinas.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “hapus” untuk melihat detail perjalanan dinas. 2. Sistem akan menampilkan pertanyaan untuk melakukan validasi penghapusan data kepada aktor. <p>{Validasi Penghapusan Data}</p> <ol style="list-style-type: none"> 3. Aktor memilih “ya” untuk memvalidasi penghapusan data. 4. Sistem akan menyimpan perubahan data perjalanan dinas. 5. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Pembatalan Penghapusan Data</p> <p>Pada {Validasi Peghapusan Data}, jika aktor memilih untuk membatalkan penghapusan data, maka sistem akan kembali menampilkan daftar lembur sebelum dilakukan penghapusan. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data perjalanan dinas berhasil dihapus.

4.3.3.18 Spesifikasi Use Case Mencetak SPT

Spesifikasi use case mencetak SPT dapat dilihat pada Tabel 4.31.

Tabel 4. 31 Spesifikasi Use Case Mencetak SPT

Name	Mencetak SPT
Brief Description	Menjelaskan bagaimana aktor mencetak SPT perjalanan dinas.
Actor	Bendahara biro
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar perjalanan dinas.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor menekan tombol “cetak spt” pada data perjalanan dinas tertentu untuk mencetak SPT perjalanan dinas. 2. Sistem menampilkan form cetak SPT. 3. Aktor mengisi tanggal cetak SPT kemudian menekan tombol “cetak”. 4. Sistem akan menerbitkan SPT perjalanan dinas dalam bentuk file pdf. 5. Use case selesai. <p>{Use Case Selesai}</p>
Alternative flow	-
Post-condition	SPT berhasil dicetak.

4.3.3.19 Spesifikasi Use Case Membuat SPPD

Spesifikasi use case membuat SPPD dapat dilihat pada Tabel 4.32.

Tabel 4. 32 Spesifikasi Use Case Membuat SPPD

Name	Membuat SPPD
Brief Description	Menjelaskan bagaimana aktor membuat SPPD (Surat Perintah Perjalanan Dinas).
Actor	Bendahara biro
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar perjalanan dinas.

Tabel 4.32 Spesifikasi *Use Case* Membuat SPPD (lanjutan)

Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “detail” untuk melihat detail perjalanan dinas. 2. Sistem menampilkan detail perjalanan dinas yang isinya keterangan mengenai perjalanan dinas dan daftar nama pegawai yang ditugaskan. <p>{Mengisi Data Umum SPPD}</p> <ol style="list-style-type: none"> 3. Aktor menekan tombol “detail” pada baris Data SPPD untuk memasukkan data umum SPPD. 4. Sistem menampilkan form detail SPPD. 5. Aktor mengisi form detail SPPD, kemudian menekan tombol simpan. 6. Sistem menyimpan data umum SPPD. <p>{Mencetak SPPD}</p> <ol style="list-style-type: none"> 7. Aktor menekan tombol “Cetak SPPD” pada pegawai yang SPPD-nya ingin dibuat. 8. Sistem menerbitkan SPPD dalam bentuk file PDF. 9. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Data Umum Telah Diisi</p> <p>Pada {Mengisi Data Umum SPPD}, jika telah mengisi data umum SPPD sebelumnya, maka aliran dilanjutkan ke {Mencetak SPPD}.</p>
Post-condition	SPPD berhasil dibuat.

4.3.3.20 Spesifikasi *Use Case* Mengecek SPJ

Spesifikasi *use case* mengecek SPJ dapat dilihat pada Tabel 4.33.

Tabel 4. 33 Spesifikasi *Use Case* Mengecek SPJ

Name	Mengecek SPJ
Brief Description	Menjelaskan bagaimana aktor mengecek SPJ perjalanan dinas masing-masing pegawai.
Actor	Bendahara biro

Tabel 4.33 Spesifikasi Use Case Mengecek SPJ (Lanjutan)

Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar perjalanan dinas.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor menekan tombol “detail” untuk melihat detail perjalanan dinas. 2. Sistem menampilkan detail perjalanan dinas yang isinya keterangan mengenai perjalanan dinas dan daftar nama pegawai yang ditugaskan. <p>{Mengisi Data Umum Keuangan Perjalanan Dinas}</p> <ol style="list-style-type: none"> 3. Aktor menekan tombol “detail” pada baris data keuangan untuk memasukkan data umum keuangan perjalanan dinas. 4. Sistem menampilkan form detail data umum keuangan perjalanan dinas. 5. Aktor mengisi form detail SPPD berupa data pulau, provinsi, tujuan, jenis transportasi, jenis kendaraan sewa dan jenis perjalanan dinas, kemudian menekan tombol simpan. 6. Sistem menyimpan data umum keuangan dinas. <p>{Memasukkan Data Keuangan Dinas}</p> <ol style="list-style-type: none"> 7. Aktor menekan tombol “Masukkan Data Keuangan Dinas” pada pegawai tertentu. 8. Sistem akan menampilkan form untuk memasukkan data keuangan dinas pegawai bersangkutan. 9. Aktor mengisi data keuangan perjalanan dinas pegawai bersangkutan yang berupa uang penginapan, lama menginap, lokasi dinas, jenis angkutan, biaya angkutan, biaya taxi dan angkutan lokal, kemudian menekan tombol “Cek Standar” untuk melakukan pengecekan kesesuaian data keuangan perjalanan dinas dengan standar yang ada. 10. Sistem melakukan pengecekan data SPJ. <p>{Data Tidak Sesuai}</p> <ol style="list-style-type: none"> 11. Sistem menampilkan pesan bawa data sesuai. 12. Use case selesai. <p>{Use Case Selesai}</p>

Tabel 4.33 Spesifikasi *Use Case* Mengecek SPJ (Lanjutan)

Alternative flow	<p>A1. Data Umum Keuangan Perjalanan Dinas Telah Diisi</p> <p>Pada {Mengisi Data Umum Keuangan Perjalanan Dinas}, jika telah mengisi data umum SPPD sebelumnya, maka aliran dilanjutkan ke {Memasukkan Data Keuangan Dinas}.</p> <p>A2. Data SPJ Tidak Sesuai</p> <p>Pada {Data Tidak Sesuai}, jika data yang diinputkan tidak sesuai dengan ketentuan, maka sistem akan menampilkan pesan bahwa data tidak sesuai, kemudian <i>use case</i> selesai.</p>
Post-condition	Hasil pengecekan data SPJ diketahui.

4.3.3.21 Spesifikasi *Use Case* Membuat Kwitansi Uang Perjalanan Dinas

Spesifikasi *use case* membuat kwitansi uang perjalanan dinas dapat dilihat pada Tabel 4.34.

Tabel 4. 34 Spesifikasi *Use Case* Membuat Kwitansi Uang Perjalanan Dinas

Name	Membuat kwitansi uang perjalanan dinas
Brief Description	Menjelaskan bagaimana aktor membuat kwitansi uang perjalanan dinas sesuai dengan SPJ.
Actor	Bendahara biro
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Data SPJ yang sesuai standar ketentuan uang perjalanan dinas telah diinputkan.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “detail” untuk melihat detail perjalanan dinas. 2. Sistem menampilkan detail perjalanan dinas yang salah satunya berupa daftar nama pegawai yang mengikuti perjalanan dinas tersebut beserta data SPJ yang telah diinputkan dan dilakukan pengecekan. <p>{Cetak Kwitansi}</p> <ol style="list-style-type: none"> 3. Aktor menekan tombol “cetak kwitansi” untuk mencetak kwitansi pada data pegawai bersangkutan. 4. Sistem akan menerbitkan kwitansi uang perjalanan dinas dalam bentuk <i>file</i> PDF. 5. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>

Tabel 4.34 Spesifikasi *Use Case* Membuat Kwitansi Uang Perjalanan Dinas (Lanjutan)

Alternative flow	A1. Menangani Data SPJ yang Belum Sesuai Standar Pada {Cetak SPJ}, apabila data SPJ belum sesuai standar maka tidak dapat melakukan pencetakan kwitansi karena sistem tidak menampilkan tombol “cetak kwitansi”. Kemudian <i>use case</i> selesai.
Post-condition	Kwitansi uang perjalanan dinas berhasil dibuat.

4.3.3.22 Spesifikasi *Use Case* Memasukkan Data Sidang

Spesifikasi *use case* memasukkan data sidang dapat dilihat pada Tabel 4.35.

Tabel 4. 35 Spesifikasi *Use Case* Memasukkan Data Sidang

Name	Memasukkan data sidang
Brief Description	Menjelaskan bagaimana aktor memasukkan data sidang.
Actor	Bendahara Biro
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia form input data sidang.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> <i>Use case</i> dimulai ketika aktor mengisi <i>form</i> input sidang berupa hari, tanggal, waktu, tempat, acara, dan peserta sidang. Aktor meyimpan data sidang dengan menekan tombol “simpan”. Sistem menyimpan data sidang. <p>{Menyimpan Data}</p> <ol style="list-style-type: none"> Sistem menampilkan pesan bahwa data berhasil disimpan. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	A1. Menangani Kegagalan Penyimpanan Data Pada {Menyimpan Data}, jika sistem gagal menyimpan data yang akan dimasukan, maka sistem akan menampilkan pesan bahwa gagal melakukan penyimpanan data dan kembali menampilkan form sidang. Kemudian <i>use case</i> selesai.
Post-condition	Data sidang berhasil disimpan.



4.3.3.23 Spesifikasi Use Case Mengubah Data Sidang

Spesifikasi *use case* mengubah data sidang dapat dilihat pada tabel 4.36.

Tabel 4. 36 Spesifikasi Use Case Mengubah Data Sidang

Name	Mengubah data sidang
Brief Description	Menjelaskan bagaimana aktor mengubah data sidang.
Actor	Bendahara biro
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar sidang.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol edit pada data sidang yang ingin diubah. 2. Sistem akan menampilkan <i>form</i> untuk mengubah data sidang. 3. Aktor mengisi <i>form</i> untuk mengubah data sidang. 4. Aktor menekan tombol “simpan” untuk mengirim perubahan data kepada sistem. 5. Sistem akan menyimpan perubahan data sidang. <p>{Menghapus Pegawai Sidang}</p> <ol style="list-style-type: none"> 6. Sistem menghapus pegawai sidang. <p>{Menambahkan Pegawai Sidang}</p> <ol style="list-style-type: none"> 7. Aktor menekan tombol “Tambah pegawai”. 8. Sistem menampilkan form input data pegawai. 9. Aktor mengisi data nama dan nip pegawai. 10. Aktor menekan tombol “simpan”. 11. Sistem menyimpan data pegawai sidang. <p>{Menyimpan Data}</p> <ol style="list-style-type: none"> 12. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>

Tabel 4.36 Spesifikasi Use Case Mengubah Data Sidang (Lanjutan)

<p>Alternative flow</p>	<p>A1. Menangani Kegagalan Penyimpanan Data</p> <p>Pada {Menyimpan Data}, jika sistem gagal menyimpan data pegawai sidang, maka sistem akan menampilkan pesan bahwa gagal melakukan penyimpanan data pegawai sidang dan kembali menampilkan daftar pegawai sidang sebelum dilakukan penambahan. Kemudian use case selesai.</p> <p>A2. Tidak melakukan penambahan pegawai</p> <p>Pada {Menambah Pegawai Sidang}, jika aktor tidak melakukan penambahan pegawai maka aliran dilanjutkan ke {Use Case Selesai}. Kemudian use case selesai.</p> <p>A3. Tidak melakukan penghapusan pegawai</p> <p>Pada {Menghapus Pegawai Sidang}, jika aktor tidak melakukan penghapusan pegawai maka aliran dilanjutkan ke {Use Case Selesai}. Kemudian use case selesai.</p>
<p>Post-condition</p>	<p>Data sidang berhasil diubah.</p>

4.3.3.24 Spesifikasi Use Case Menghapus Data Sidang

Spesifikasi use case menghapus data sidang dapat dilihat pada Tabel 4.37.

Tabel 4. 37 Spesifikasi Use Case Menghapus Data Sidang

<p>Name</p>	<p>Menghapus data sidang</p>
<p>Brief Description</p>	<p>Menjelaskan bagaimana aktor menghapus data sidang.</p>
<p>Actor</p>	<p>Bendahara biro</p>
<p>Pre-condition</p>	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar sidang.
<p>Basic flow</p>	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor menekan tombol “hapus” pada data sidang yang ingin dihapus. 2. Sistem akan menampilkan pertanyaan untuk melakukan validasi menghapus data kepada aktor. <p>{Validasi penghapusan Data}</p> <ol style="list-style-type: none"> 3. Aktor memilih “ya” untuk memvalidasi penghapusan. 4. Sistem akan menghapus data sidang. 5. Use case selesai. <p>{Use Case Selesai}</p>



Tabel 4.37 Spesifikasi *Use Case* Menghapus Data Sidang (Lanjutan)

Alternative flow	A1. Menangani Pembatalan Penghapusan Data Pada {Validasi Penghapusan Data}, jika aktor memilih untuk membatalkan penghapusan data, maka sistem akan kembali menampilkan daftar sidang sebelum dilakukan penghapusan. Kemudian <i>use case</i> selesai.
Post-condition	Data sidang berhasil dihapus.

4.3.3.25 Spesifikasi *Use Case* Mencetak Daftar Penerima dan Kwitansi Uang Sidang

Spesifikasi *use case* mencetak daftar penerima dan kwitansi uang sidang dapat dilihat pada Tabel 4.38.

Tabel 4. 38 Spesifikasi *Use Case* Mencetak Daftar Penerima dan Kwitansi Uang Sidang

Name	Mencetak daftar penerima dan kwitansi uang sidang
Brief Description	Menjelaskan bagaimana aktor mencetak daftar penerima dan kwitansi uang sidang sesuai dengan data sidang.
Actor	Bendahara biro
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar sidang.
Basic flow	{Use Case Dimulai} 1. <i>Use case</i> dimulai ketika aktor menekan tombol “detail” untuk melihat detail sidang. 2. Sistem menampilkan detail sidang. 3. Aktor mengisi data tanggal pembayaran dan kode kwitansi. 4. Aktor menekan tombol “cetak daftar penerima dan kwitansi uang sidang” untuk mencetak daftar penerima dan kwitansi uang sidang. 5. Sistem akan menerbitkan daftar penerima dan kwitansi uang sidang dalam bentuk <i>file</i> pdf. 6. <i>Use case</i> selesai. {Use Case Selesai}
Alternative flow	-
Post-condition	Daftar penerima dan kwitansi uang sidang berhasil dicetak.

4.3.3.26 Spesifikasi Use Case Membuat SPP

Spesifikasi use case membuat SPP dapat dilihat pada Tabel 4.39.

Tabel 4. 39 Spesifikasi Use Case Membuat SPP

Name	Membuat SPP
Brief Description	Menjelaskan bagaimana aktor membuat SPP (Surat Permintaan pembayaran), diantaranya SPP uang makan, uang lembur, tunjangan prestasi atau tunjangan daerah.
Actor	Bendahara biro
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman laporan rekapitulasi.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor memilih menu laporan rekapitulasi yang SPPnya ingin dibuat, yaitu laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi atau tunjangan daerah. 2. Sistem menampilkan data laporan rekapitulasi uang makan, uang lembur, tunjangan prestasi, atau tunjangan daerah yang pernah dibuat. 3. Aktor mengisi tahun, bulan dan tanggal cetak untuk mencetak SPP kemudian menekan tombol “cetak”. <p>{Menerbitkan SPP}</p> <ol style="list-style-type: none"> 4. Sistem akan menerbitkan SPP ke perangkat aktor dalam bentuk file pdf. 5. Use case selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Pada {Menerbitkan SPP}, jika ada bagian yang belum membuat laporan rekapitulasi bagiannya pada bulan dan tahun yang dipilih, maka sistem akan menampilkan pesan bahwa tidak dapat melakukan pencetakan SPP karena ada bagian yang belum membuat laporan rekapitulasi.</p>
Post-condition	SPP berhasil dibuat.

4.3.3.27 Spesifikasi Use Case Memasukkan Data Presensi

Spesifikasi use case memasukkan data presensi dapat dilihat pada Tabel 4.40.

Tabel 4. 40 Spesifikasi Use Case Memasukkan Data Presensi

Name	Memasukkan data presensi
Brief Description	Menjelaskan bagaimana aktor melakukan <i>import</i> laporan presensi ke sistem.
Actor	Bendahara biro
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara biro 2. Tersedia halaman <i>import</i> presensi.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor menekan tombol "<i>choose file</i>" untuk memasukan data presensi ke sistem. 2. Sistem menampilkan kotak dialog untuk aktor memilih file laporan presensi yang akan dimasukkan. 3. Aktor memilih file presensi kemudian menekan tombol "ok" kemudian aktor akan menekan tombol "<i>upload</i>" untuk memasukan file ke sistem. 4. Sistem memasukan dan menyimpan data presensi dari file yang dimasukkan oleh aktor. 5. Use case selesai. <p>{Use Case Selesai}</p>
Alternative flow	-
Post-condition	Data presensi berhasil masuk ke sistem.

4.3.3.28 Spesifikasi Use Case Memasukkan Data Pegawai

Spesifikasi use case memasukkan data pegawai dapat dilihat pada Tabel 4.41.

Tabel 4. 41 Spesifikasi Use Case Memasukkan Data Pegawai

Name	Memasukkan data pegawai
Brief Description	Menjelaskan bagaimana aktor memasukkan data pegawai.
Actor	Bendahara Biro
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia form input data pegawai.

Tabel 4.41 Spesifikasi Use Case Memasukkan Data Pegawai (Lanjutan)

Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor mengisi <i>form input</i> pegawai berupa nama, nip, jenis kelamin, pangkat, golongan/ruang, jabatan pegawai, username, password, serta level. 2. Aktor meyimpan data pegawai dengan menekan tombol “simpan”. 3. Sistem menyimpan data pegawai. <p>{Menyimpan Data}</p> <ol style="list-style-type: none"> 4. Sistem menampilkan pesan bahwa data berhasil disimpan. 5. Use case selesai. <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Kegagalan Penyimpanan Data</p> <p>Pada {Menyimpan Data}, jika sistem gagal menyimpan data yang akan dimasukan, maka sistem akan menampilkan pesan bahwa gagal melakukan penyimpanan data dan kembali menampilkan form pegawai. Kemudian use case selesai.</p>
Post-condition	Data pegawai berhasil disimpan.

4.3.3.29 Spesifikasi Use Case Mengubah Data Pegawai

Spesifikasi use case mengubah data pegawai dapat dilihat pada Tabel 4.42.

Tabel 4. 42 Spesifikasi Use Case Mengubah Data Pegawai

Name	Mengubah data pegawai
Brief Description	Menjelaskan bagaimana aktor mengubah data pegawai.
Actor	Bendahara biro
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar pegawai.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. Use case dimulai ketika aktor mengisi <i>form input</i> pegawai berupa nama, nip, jenis kelamin, pangkat, golongan/ruang, jabatan pegawai, username, password, serta level. 2. Aktor meyimpan data pegawai dengan menekan tombol “simpan”. 3. Sistem menyimpan data pegawai.



Tabel 4.42 Spesifikasi *Use Case* Mengubah Data Pegawai (Lanjutan)

	<p>{Menyimpan Data}</p> <p>4. Sistem menampilkan pesan bahwa data berhasil disimpan.</p> <p>5. <i>Use case</i> selesai.</p> <p>{Use Case Selesai}</p>
Alternative flow	<p>A1. Menangani Kegagalan Penyimpanan Data</p> <p>Pada {Menyimpan Data}, jika sistem gagal menyimpan data yang akan dimasukkan, maka sistem akan menampilkan pesan bahwa gagal melakukan penyimpanan data dan kembali menampilkan form pegawai. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data pegawai berhasil diubah.

4.3.3.30 Spesifikasi *Use Case* Menghapus Data Pegawai

Spesifikasi *use case* menghapus data pegawai dapat dilihat pada Tabel 4.43.

Tabel 4. 43 Spesifikasi *Use Case* Menghapus Data Pegawai

Name	Menghapus data pegawai
Brief Description	Menjelaskan bagaimana aktor menghapus data pegawai.
Actor	Bendahara biro
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar pegawai.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “hapus” pada data pegawai yang ingin dihapus. 2. Sistem akan menampilkan pertanyaan untuk melakukan validasi menghapus data kepada aktor. <p>{Validasi penghapusan Data}</p> <ol style="list-style-type: none"> 3. Aktor memilih “ya” untuk memvalidasi penghapusan data pegawai. 4. Sistem akan menghapus data pegawai. 5. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>

Tabel 4.43 Spesifikasi *Use Case* Menghapus Data Pegawai (Lanjutan)

Alternative flow	<p>A1. Menangani Pembatalan Penghapusan Data</p> <p>Pada {Validasi Penghapusan Data}, jika aktor memilih untuk membatalkan penghapusan data, maka sistem akan kembali menampilkan daftar pegawai sebelum dilakukan penghapusan. Kemudian <i>use case</i> selesai.</p>
Post-condition	Data pegawai berhasil dihapus.

4.3.3.31 Spesifikasi *Use Case* Memasukkan Data Libur

Spesifikasi *use case* memasukkan data libur dapat dilihat pada Tabel 4.44.

Tabel 4. 44 Spesifikasi *Use Case* Memasukkan Data Pegawai

Name	Memasukkan data libur
Brief Description	Menjelaskan bagaimana aktor memasukkan data libur.
Actor	Bendahara Biro
Pre-condition	<ol style="list-style-type: none"> 1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia form input data libur.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor mengisi <i>form</i> input data libur berupa tanggal libur. 2. Aktor meyimpan data pegawai dengan menekan tombol “simpan”. 3. Sistem menyimpan data libur. 4. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	-
Post-condition	Data libur berhasil disimpan

4.3.3.32 Spesifikasi Use Case Menghapus Data Libur

Spesifikasi *use case* menghapus data libur dapat dilihat pada Tabel 4.45.

Tabel 4. 45 Spesifikasi Use Case Menghapus Data Libur

Name	Menghapus data libur
Brief Description	Menjelaskan bagaimana aktor menghapus data libur.
Actor	Bendahara biro
Pre-condition	1. Aktor dikenali oleh sistem sebagai bendahara biro. 2. Tersedia halaman daftar libur.
Basic flow	<p>{Use Case Dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor menekan tombol “hapus” pada data libur yang ingin dihapus. 2. Sistem akan menampilkan pertanyaan untuk melakukan validasi menghapus data kepada aktor. <p>{Validasi penghapusan Data}</p> <ol style="list-style-type: none"> 3. Aktor memilih “ya” untuk memvalidasi penghapusan data libur. 4. Sistem akan menghapus data libur. 5. <i>Use case</i> selesai. <p>{Use Case Selesai}</p>
Alternative flow	-
Post-condition	Data libur berhasil dihapus.

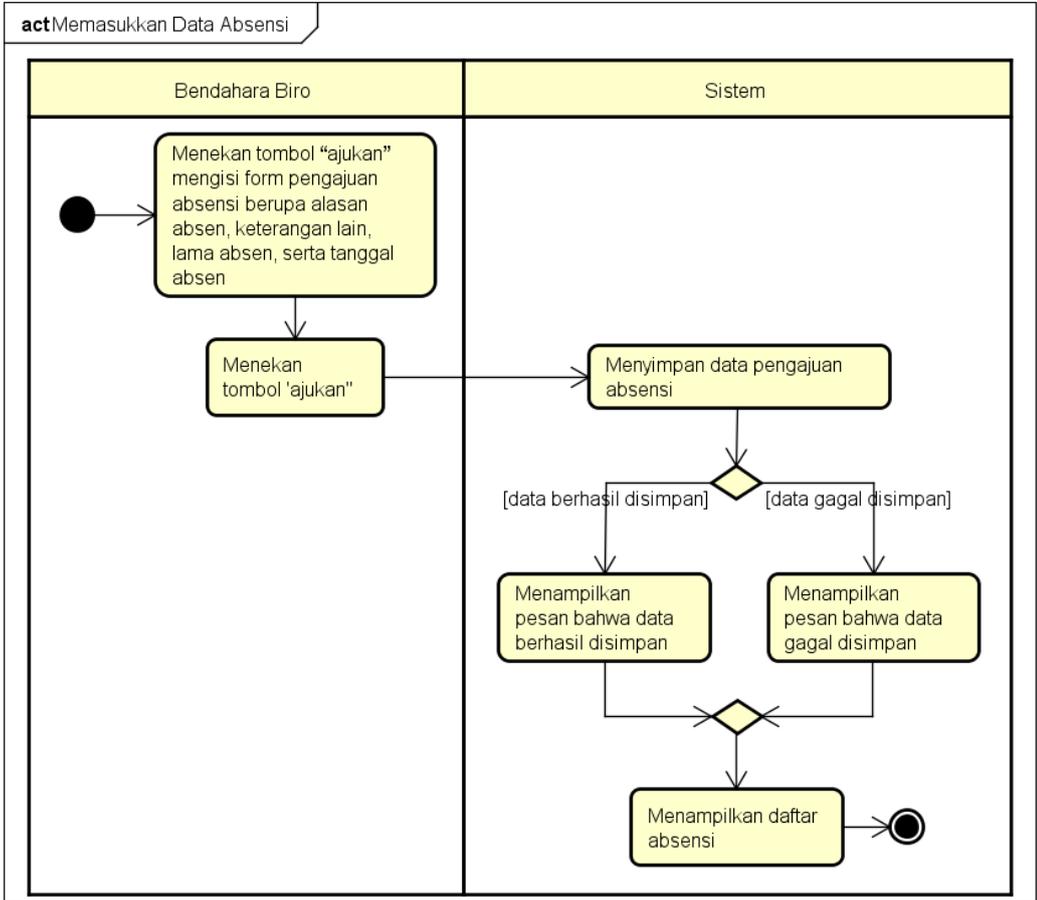
4.4 Pemodelan Aktivitas

Pemodelan aktifitas dimodelkan dalam bentuk diagram aktivitas berdasarkan alur yang ada pada spesifikasi *use case*. Beberapa aktivitas yang dimodelkan dalam penelitian ini yaitu aktivitas memasukkan data perjalanan dinas, mengecek data SPJ, membuat laporan rekapitulasi, memasukkan data sidang, mencetak daftar penerima dan kwitansi uang sidang.

4.4.1 Diagram Aktivitas Memasukkan Data Pengajuan Absensi

Diagram aktivitas memasukkan data pengajuan absensi menunjukkan aktivitas-aktivitas yang dilakukan oleh aktor pegawai dalam memasukkan data pengajuan absensi. Aktivitas dimodelkan berdasarkan alur yang ada pada spesifikasi *use case* pada Tabel 4.15. Diagram aktivitas memasukkan data

pengajuan absensi dapat dilihat pada Gambar 4.19. alur dimulai dengan pegawai yang mengisi kemudian menyimpan data pengajuan absensi, kemudian sistem akan menyimpan data tersebut. Sistem akan menampilkan pesan bahwa berhasil melakukan penyimpanan jika data berhasil disimpan dan sebaliknya.

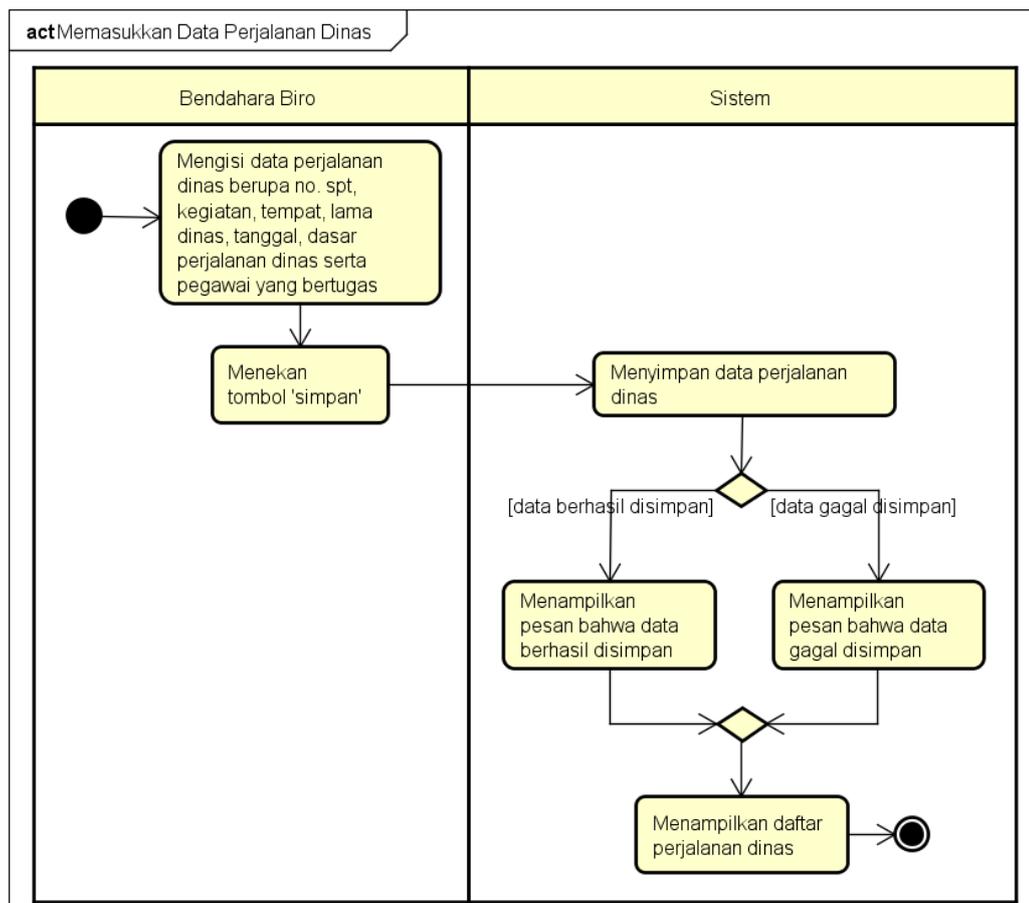


Gambar 4. 19 Diagram Aktivitas Memasukkan Data Pengajuan Absensi

4.4.2 Diagram Aktivitas Memasukkan Data Perjalanan Dinas

Diagram aktivitas memasukkan data perjalanan dinas menunjukkan aktivitas-aktivitas yang dilakukan oleh aktor bendahara biro dalam memasukkan data perjalanan dinas beserta pegawai yang bertugas. Aktivitas dimodelkan berdasarkan alur yang ada pada spesifikasi *use case* pada Tabel 4.28. Diagram aktivitas memasukkan data perjalanan dinas dapat dilihat pada Gambar 4.20. alur dimulai dengan bendahara biro yang mengisi kemudian menyimpan data perjalanan dinas, kemudian sistem akan menyimpan data tersebut. Sistem akan menampilkan pesan bahwa berhasil melakukan penyimpanan jika data berhasil disimpan dan sebaliknya.

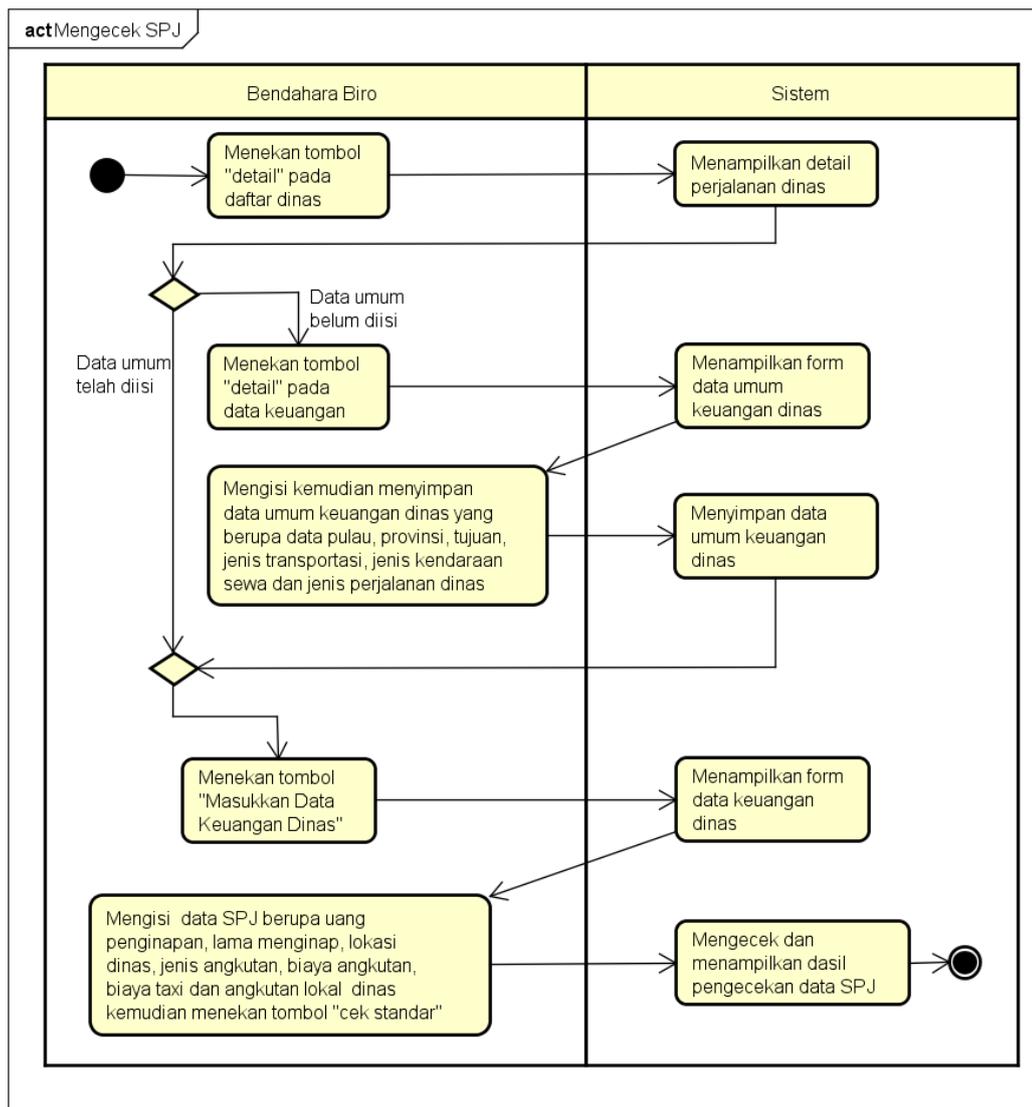




Gambar 4. 20 Diagram Aktivitas Memasukkan Data perjalanan dinas

4.4.3 Diagram Aktivitas Mengecek Data SPJ

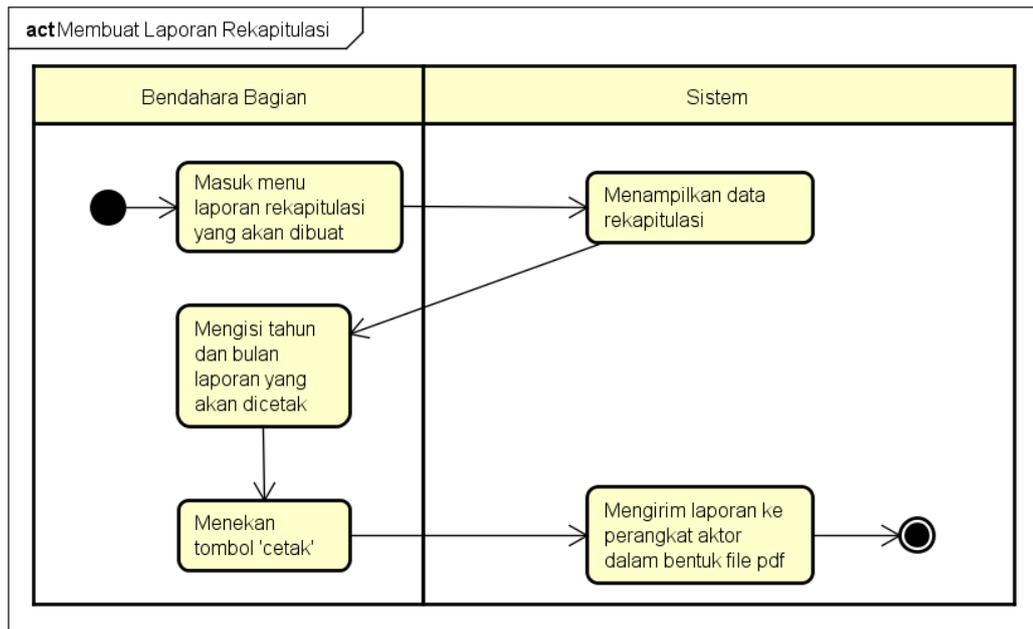
Diagram aktivitas mengecek data SPJ menunjukkan aktivitas-aktivitas yang dilakukan oleh aktor bendahara biro dalam mengecek kesesuaian data SPJ dengan standar yang telah ditetapkan. Aktivitas dimodelkan berdasarkan alur yang ada pada spesifikasi *use case* pada Tabel 4.33. Diagram aktivitas mengecek SPJ dapat dilihat pada Gambar 4.21. alur dimulai dengan bendahara biro yang melihat detail perjalanan dinas. Kemudian sistem akan menampilkan detail perjalanan dinas seperti data pegawai yang ditugaskan dalam perjalanan dinas tersebut. Apabila aktor belum mengisi data umum keuangan dinas, maka aktor akan mengisi data umum keuangan dinas terlebih dahulu kemudian mengisi data keuangan perjalanan dinas masing-masing pegawai dan mengeceknya. Apabila aktor telah mengisi data umum keuangan dinas, maka aktor dapat langsung mengisi dan melakukan pengecekan pada data keuangan dinas masing-masing pegawai.



Gambar 4. 21 Diagram Aktivitas Mengecek Data SPJ

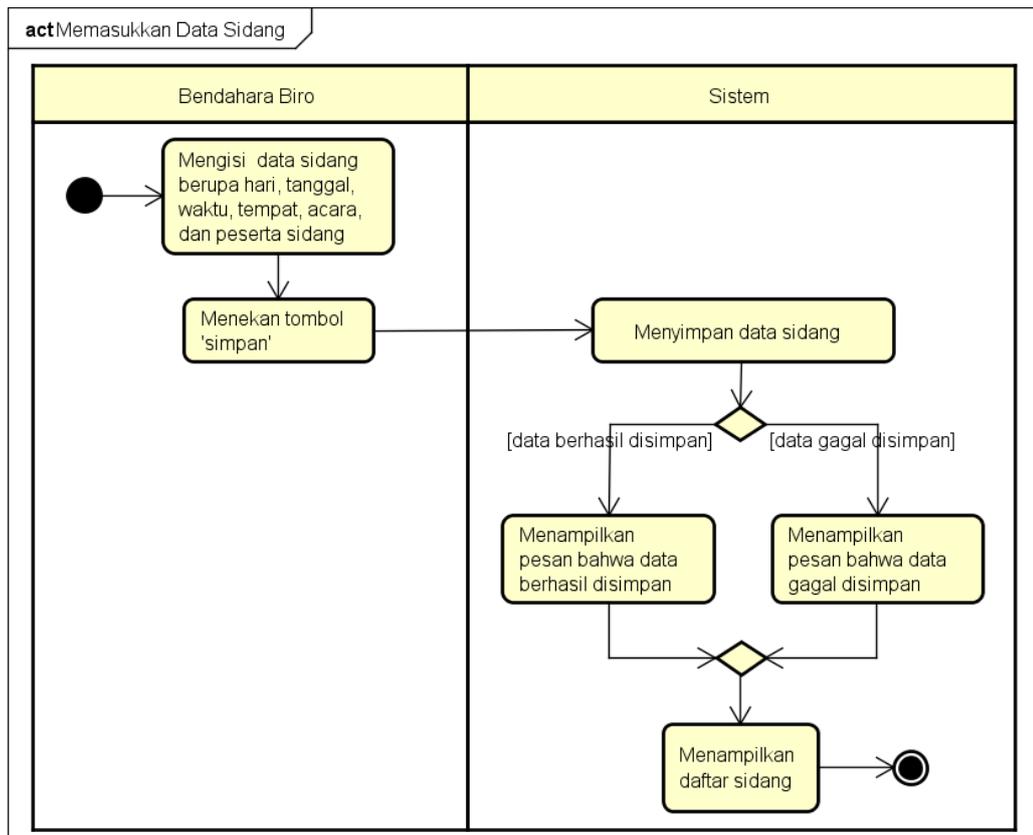
4.4.4 Diagram Aktivitas Membuat Laporan Rekapitulasi

Diagram aktivitas membuat laporan rekapitulasi menunjukkan aktivitas-aktivitas yang dilakukan oleh aktor bendahara bagian untuk membuat laporan rekapitulasi, baik laporan rekapitulasi uang makan, uang lembur, tunjangan daerah, atau tunjangan prestasi. Aktivitas dimodelkan berdasarkan alur yang ada pada spesifikasi *use case* membuat laporan rekapitulasi pada Tabel 4.19. Diagram aktivitas membuat laporan rekapitulasi dapat dilihat pada Gambar 4.22. Aktivitas dimulai dengan masuknya aktor pada menu rekapitulasi sesuai dengan jenis laporan rekapitulasi yang ingin dibuat. Kemudian sistem akan menampilkan data rekapitulasi seluruh bagian. Untuk mencetak laporan rekapitulasi, aktor perlu memasukkan bulan dan tahun laporan rekapitulasi yang ingin dicetak kemudian sistem akan mengirim laporan ke perangkat aktor dalam bentuk *file* pdf.



Gambar 4. 22 Diagram Aktivitas Membuat Laporan Rekapitulasi

4.4.5 Diagram Aktivitas Memasukkan Data Sidang

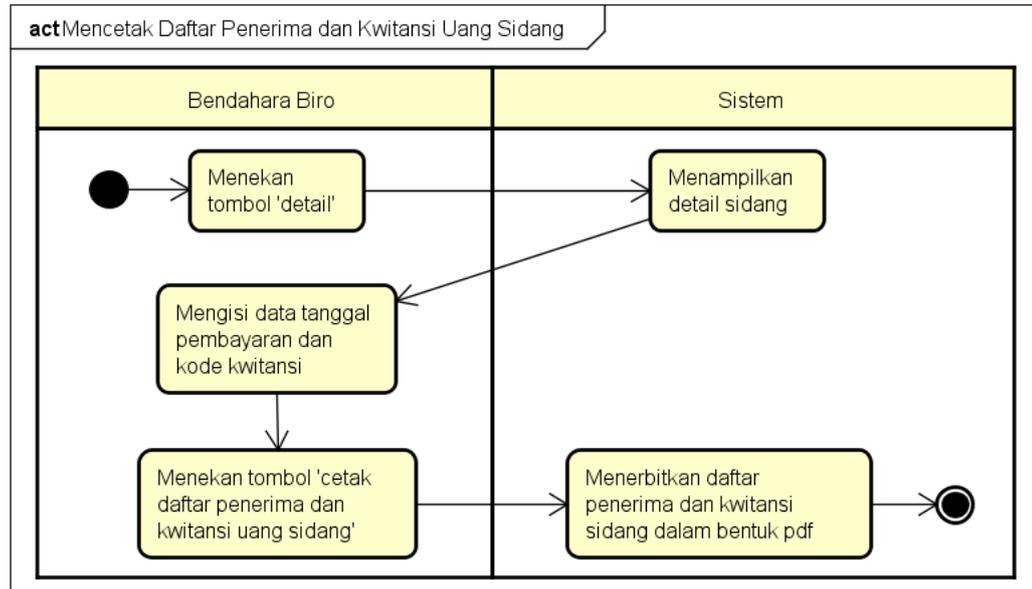


Gambar 4. 23 Diagram Aktivitas Membuat Rekapitulasi Uang Sidang

Diagram aktivitas memasukkan data sidang menunjukkan aktivitas-aktivitas yang dilakukan oleh aktor bendahara biro untuk memasukkan data sidang. Aktivitas dimodelkan berdasarkan alur yang ada pada spesifikasi *use case* memasukkan data sidang yang ada pada Tabel 4.35. Diagram aktivitas memasukkan data sidang dapat dilihat pada Gambar 2.23. Aktivitas dimulai dengan aktor yang memasukkan data sidang kemudian menekan tombol simpan. Sistem akan menyimpan data sidang dan memberikan keterangan kepada aktor apakah data tersebut berhasil disimpan atau tidak.

4.4.6 Diagram Aktivitas Mencetak Daftar Penerima dan Kwitansi Uang Sidang

Diagram aktivitas mencetak daftar penerima dan kwitansi uang sidang menunjukkan aktivitas-aktivitas yang dilakukan oleh aktor bendahara biro untuk mencetak daftar penerima dan kwitansi uang sidang. Aktivitas dimodelkan berdasarkan alur yang ada pada spesifikasi *use case* mencetak daftar penerima dan kwitansi uang sidang pada Tabel 4.38. Diagram aktivitas mencetak daftar penerima dan kwitansi uang sidang dapat dilihat pada Gambar 2.24. Aktivitas dimulai dengan aktor menekan tombol “detail” pada salah satu data sidang untuk melihat detail sidang. Sistem akan menampilkan detail sidang, kemudian aktor akan mengisi tanggal pembayaran serta kode kwitansi dilanjutkan dengan menekan tombol “cetak daftar penerima dan kwitansi uang sidang”. Sistem akan menerbitkan daftar penerima dan kwitansi uang sidang dalam bentuk *pdf*.



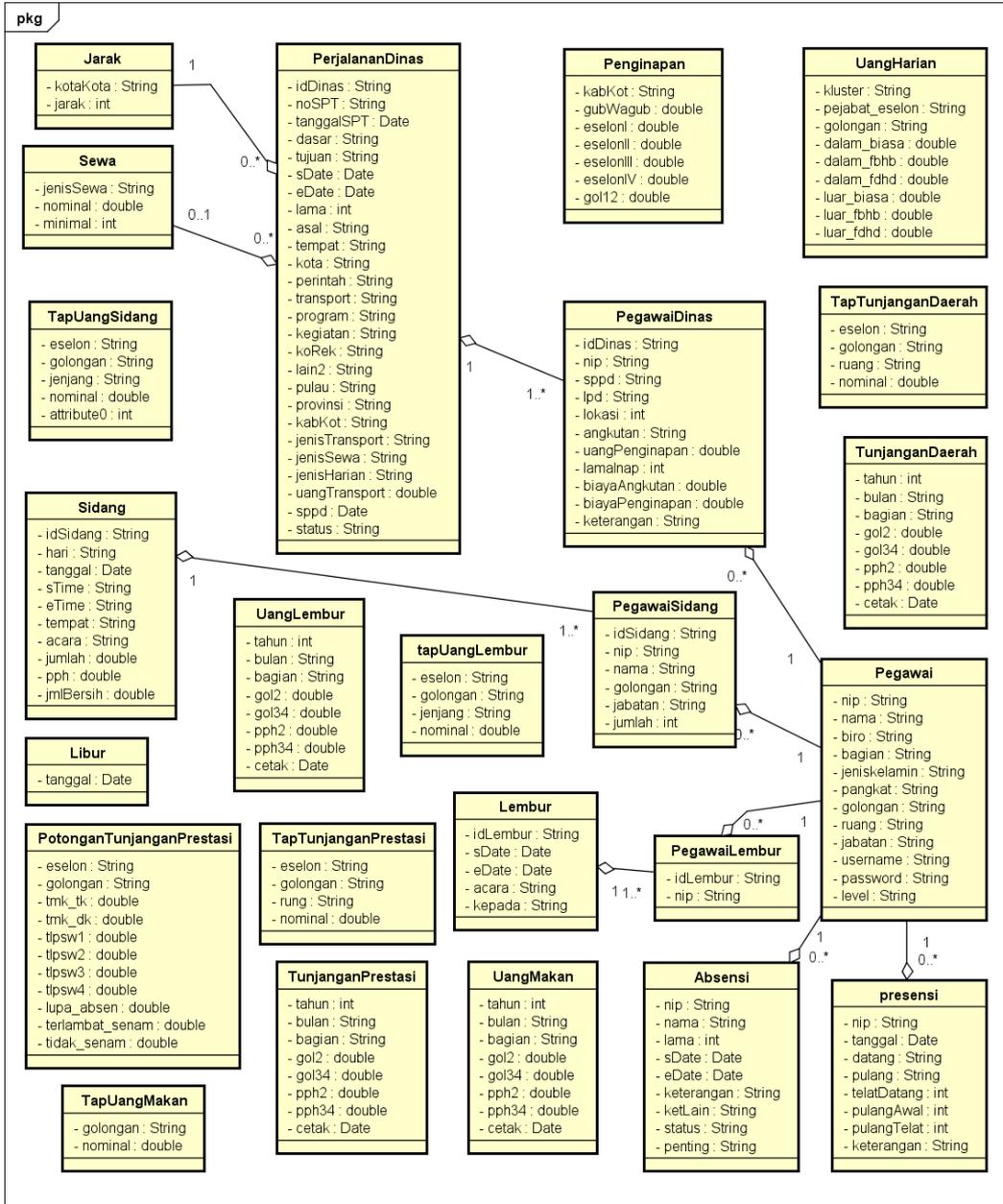
Gambar 4. 24 Diagram Aktivitas Mencetak Daftar Penerima dan Kwitansi Uang Sidang



4.5 Diagram Kelas Analisis

Gambar 4.25 merupakan diagram kelas analisis dari sistem yang akan dibangun. Diagram tersebut menunjukkan kelas-kelas, atribut kelas, serta hubungan antar kelas dari sistem yang akan dibangun. Diagram kelas analisis dapat dijadikan acuan untuk membuat rancangan basis data nantinya. Pada diagram kelas analisis yang dibuat terdapat 24 kelas yang memiliki atribut masing-masing. Objek pegawaiDinas yang direpresentasikan oleh kelas PegawaiDinas merupakan penyusun dari objek dinas yang direpresentasikan oleh kelas dinas. Hubungan multiplisitas diantara keduanya yaitu satu objek pegawaiDinas dapat menjadi penyusun dari beberapa objek dinas atau beberapa objek dinas dapat disusun dari satu objek pegawaiDinas yang sama.

Kelas Dinas merepresentasikan objek yang menyimpan informasi mengenai perjalanan dinas dan disusun oleh beberapa kelas lain, diantaranya kelas Jarak, Sewa, PegawaiDinas, Pegawai, Penginapan, dan UangHarian. Kelas Sidang merepresentasikan objek yang menyimpan informasi mengenai pelaksanaan sidang dan disusun oleh beberapa kelas lain, diantaranya kelas PegawaiSidang, Pegawai, dan TapUangSidang. Sedangkan kelas Lidang merepresentasikan objek yang menyimpan informasi mengenai pelaksanaan lembur dan disusun oleh beberapa kelas lain, diantaranya kelas PegawaiLembur dan Pegawai. Selain itu terdapat beberapa kelas yang merepresentasikan suatu objek yang terlibat dalam kegiatan pembuatan laporan rekapitulasi yaitu TapTunjanganDaerah, TunjanganDaerah, TapUangLembur, UangLembur, TapTunjanganPrestasi, TapPotonganPrestasi, TunjanganPrestasi, TapUangMakan, UangMakan, Absensi, libur, dan Presensi. Kemudian terdapat objek pengguna yang direpresentasikan oleh kelas User yang menyimpan informasi pengguna yang dapat sistem.



Gambar 4. 25 Diagram Kelas Analisis



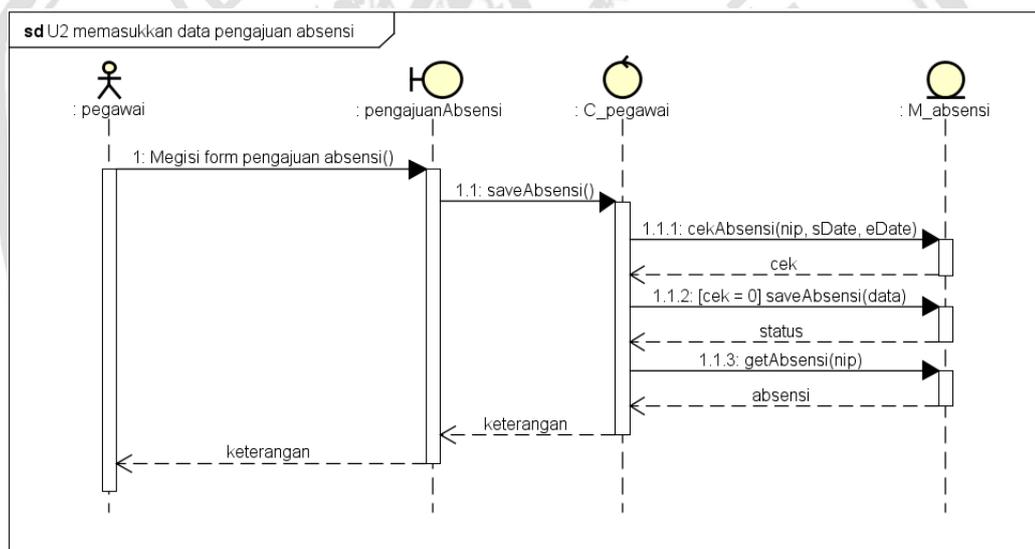
BAB 5 PERANCANGAN

Bab ini berisi dokumentasi perancangan sistem informasi keuangan pegawai yang akan dibangun. Dokumentasi perancangan dimodelkan dengan diagram UML yaitu *sequence diagram* dan *class diagram*. Pemodelan rancangan basis data menggunakan *physical data model*. Pada bab ini juga akan dilakukan perancangan algoritme dan antarmuka pengguna.

5.1 Sequence Diagram

Sequence diagram digunakan untuk menunjukkan urutan pesan yang dilewatkan antar objek. *Sequence diagram* menekankan urutan aktivitas berbasis waktu yang terjadi di antara seperangkat objek. *Sequence diagram* dimodelkan berdasarkan aliran pada spesifikasi *use case* yang telah didokumentasikan pada bab sebelumnya. Pemodelan *sequence diagram* dilakukan untuk mengetahui pertukaran pesan antara aktor, *boundary*, *control*, dan entitas pada sistem informasi keuangan pegawai yang akan dibangun.

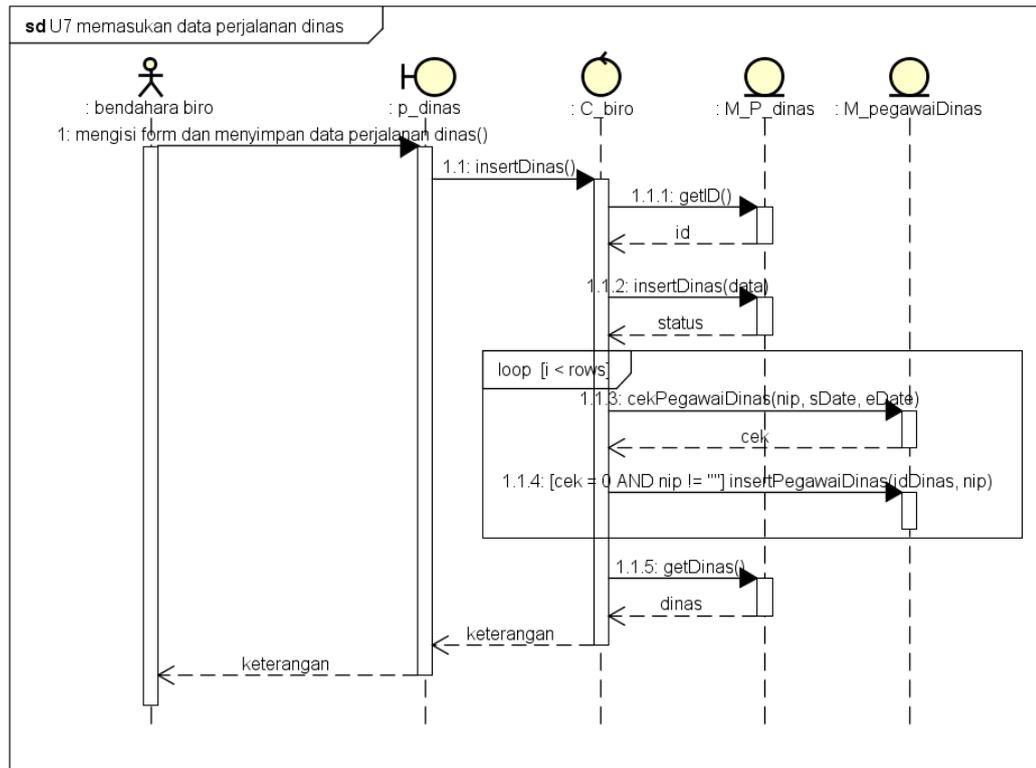
5.1.1 Sequence Diagram Memasukkan Data Pengajuan Absensi



Gambar 5. 1 Sequence Diagram Memasukkan Data Pengajuan Absensi

Gambar 5.1 merupakan visualisasi dari pertukaran pesan pada aktivitas memasukkan data Pengajuan Absensi yang dilakukan oleh aktor pegawai. Objek yang terlibat dalam interaksi ini yaitu aktor pegawai, pengajuanAbsensi sebagai *boundary*, C_pegawai sebagai objek *control*, serta M_absensi sebagai objek *model*.

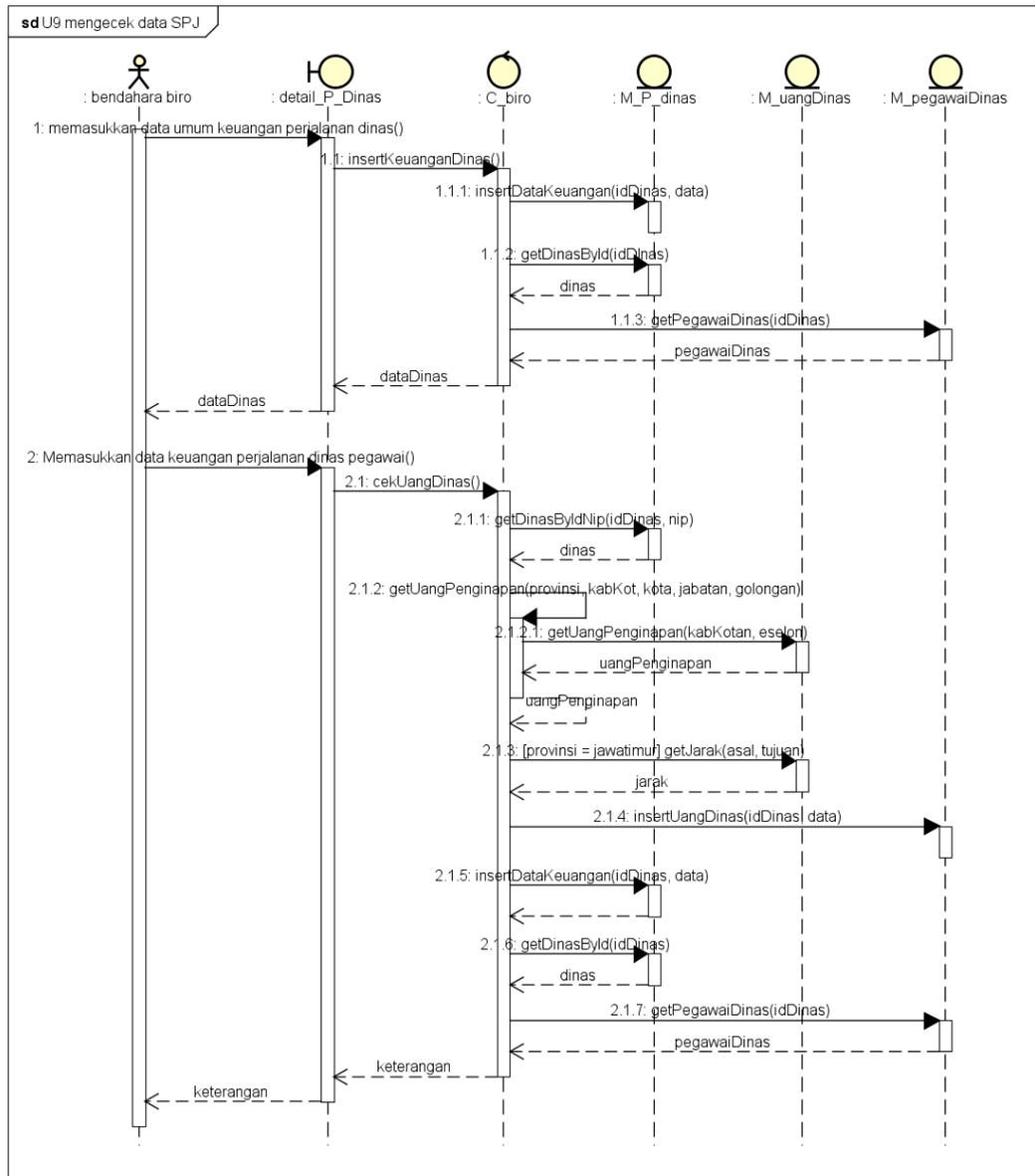
5.1.2 Sequence Diagram Memasukkan Data Perjalanan Dinas



Gambar 5. 2 Sequence Diagram Memasukkan Data Perjalanan Dinas

Gambar 5.2 merupakan visualisasi dari pertukaran pesan pada aktivitas memasukkan data SPT yang dilakukan oleh aktor bendahara biro. Objek yang terlibat dalam interaksi ini yaitu aktor bendahara bagian, editDinas sebagai *boundary*, C_biro sebagai objek *control*, serta M_dinas dan M_pegawaiDinas sebagai objek *model*.

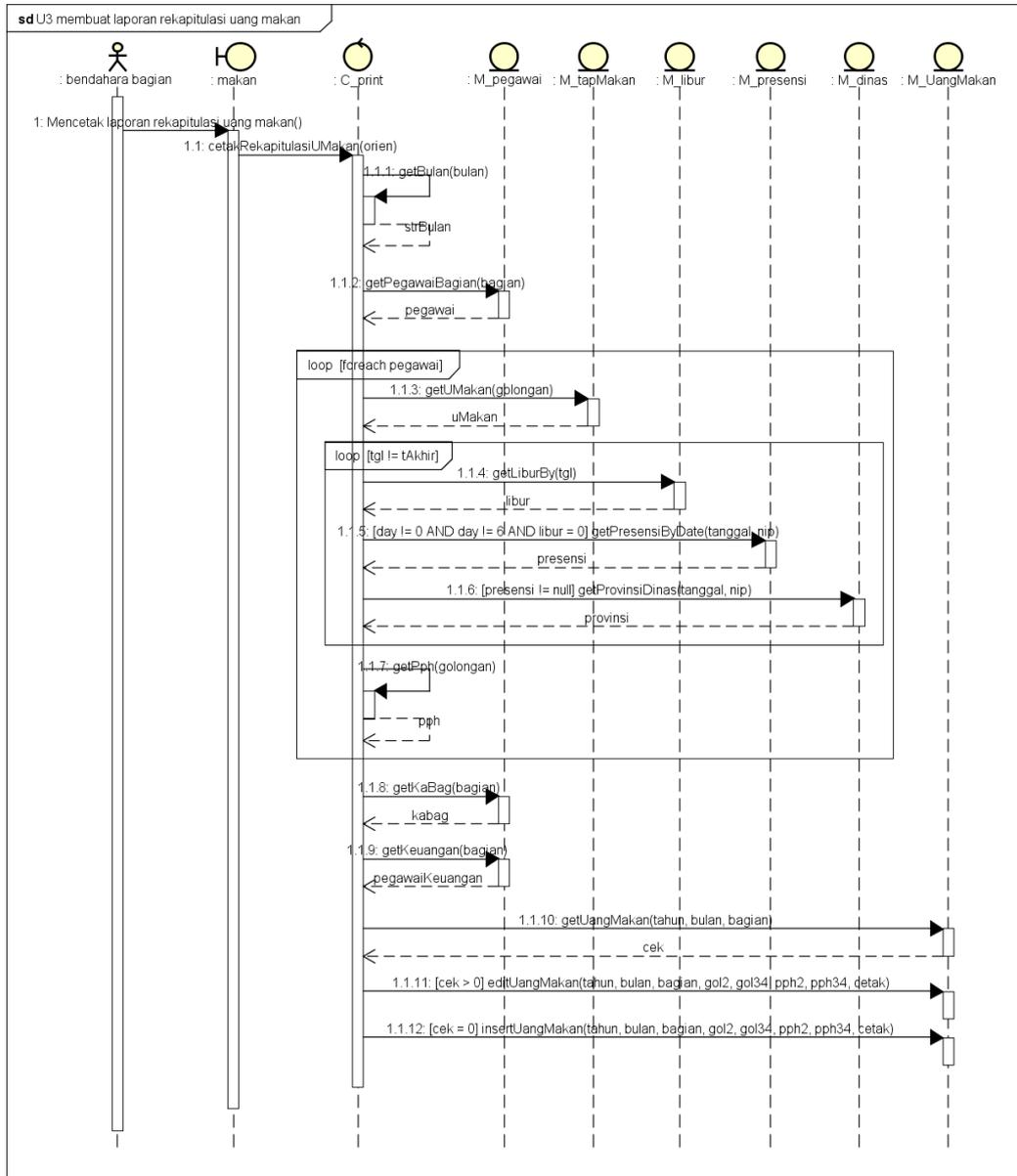
5.1.3 Sequence Diagram Mengecek Data SPJ



Gambar 5. 3 Sequence Diagram Mengecek Data SPJ

Gambar 5.3 merupakan visualisasi dari pertukaran pesan pada aktivitas mengecek data SPJ yang dilakukan oleh aktor bendahara biro. Objek yang terlibat dalam interaksi ini yaitu aktor bendahara biro, detailDinas sebagai *boundary*, C_biro sebagai objek *control*, serta M_dinas, M_pegawaiDinas dan M_uangDinas sebagai objek *model*.

5.1.4 Sequence Diagram Membuat Laporan Rekapitulasi

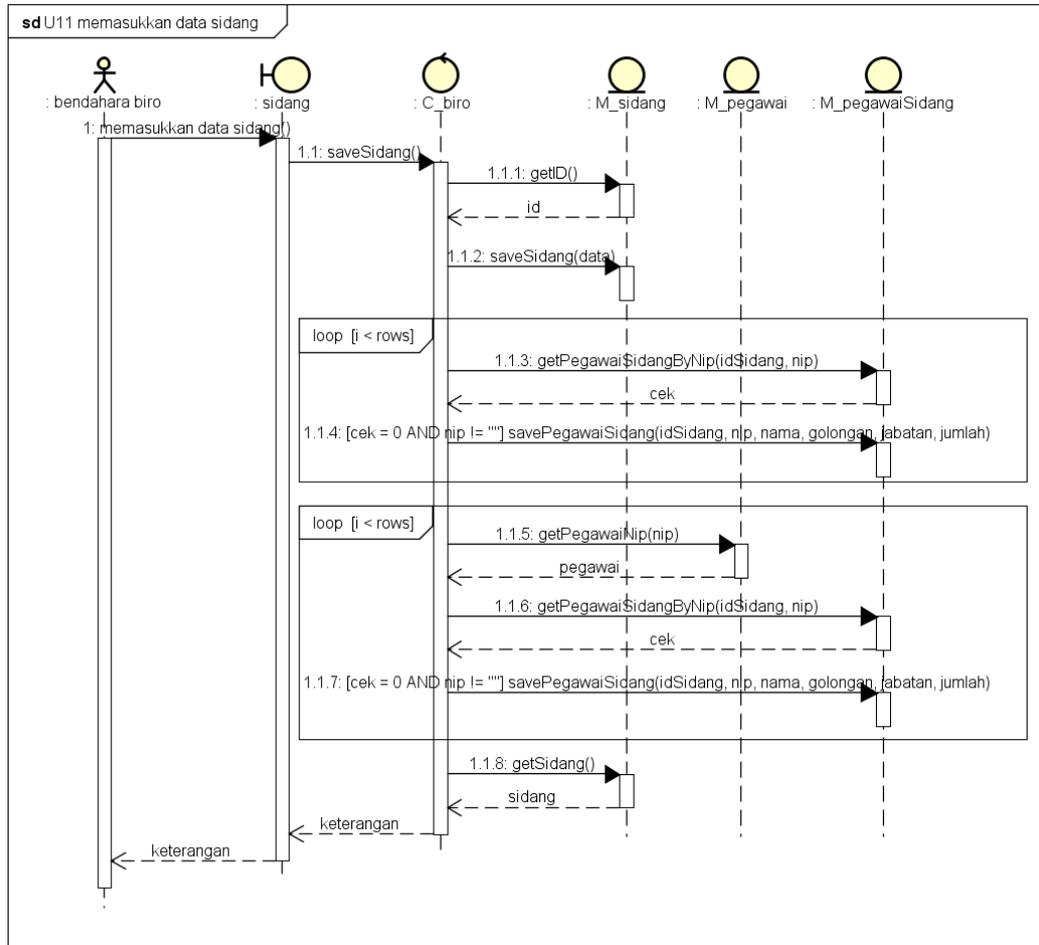


Gambar 5. 4 Sequence Diagram Membuat Laporan Rekapitulasi Uang Makan

Gambar 5.4 merupakan visualisasi dari pertukaran pesan pada aktivitas membuat laporan rekapitulasi yang dilakukan oleh aktor bendahara biro. Terdapat beberapa jenis laporan rekapitulasi yang dapat dibuat yaitu laporan rekapitulasi uang makan, laporan rekapitulasi uang lembur, laporan rekapitulasi tunjangan daerah dan laporan rekapitulasi tunjangan prestasi. *Sequence diagram* yang akan digambarkan yaitu *sequence diagram* membuat laporan rekapitulasi uang makan. Objek yang terlibat dalam interaksi ini yaitu aktor bendahara bagian, makan sebagai *boundary*, C_bagian sebagai objek *control*, serta M_pegawai, M_tapMakan, M_libur, M_presensi, M_dinas, dan M_uangMakan sebagai objek *model*.



5.1.5 Sequence Diagram Memasukkan Data Sidang

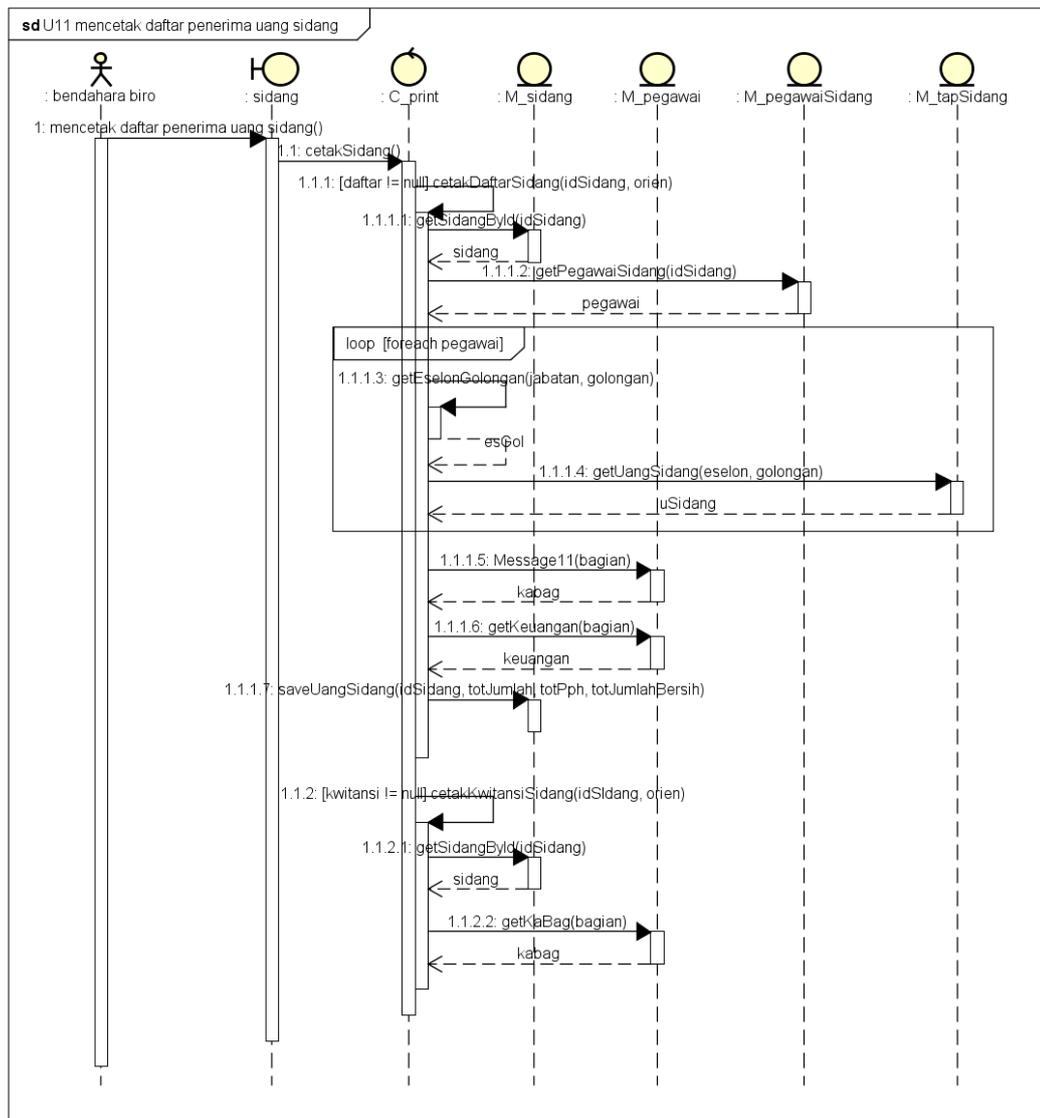


Gambar 5. 5 Sequence Diagram Memasukkan Data Sidang

Gambar 5.5 merupakan visualisasi dari pertukaran pesan pada aktivitas memasukkan data sidang yang dilakukan oleh aktor bendahara biro. Objek yang terlibat dalam interaksi ini yaitu aktor bendahara biro, sidang sebagai *boundary*, C_biro sebagai objek *control*, serta M_sidang, M_pegawaiSidang dan M_pegawai sebagai objek *model*.



5.1.6 Sequence Diagram Mencetak Daftar penerima dan Kwitansi Uang Sidang



Gambar 5. 6 Sequence Diagram Mencetak Daftar Penerima dan Kwitansi Uang Sidang

Gambar 5.6 merupakan visualisasi dari pertukaran pesan pada aktivitas mencetak daftar penerima dan kwitansi uang sidang yang dilakukan oleh aktor bendahara biro. Objek yang terlibat dalam interaksi ini yaitu aktor bendahara biro, sidang sebagai *boundary*, C_priny sebagai objek *control*, serta M_sidang, M_pegawaiSidang, M_tapSidang dan M_pegawai sebagai objek *model*.



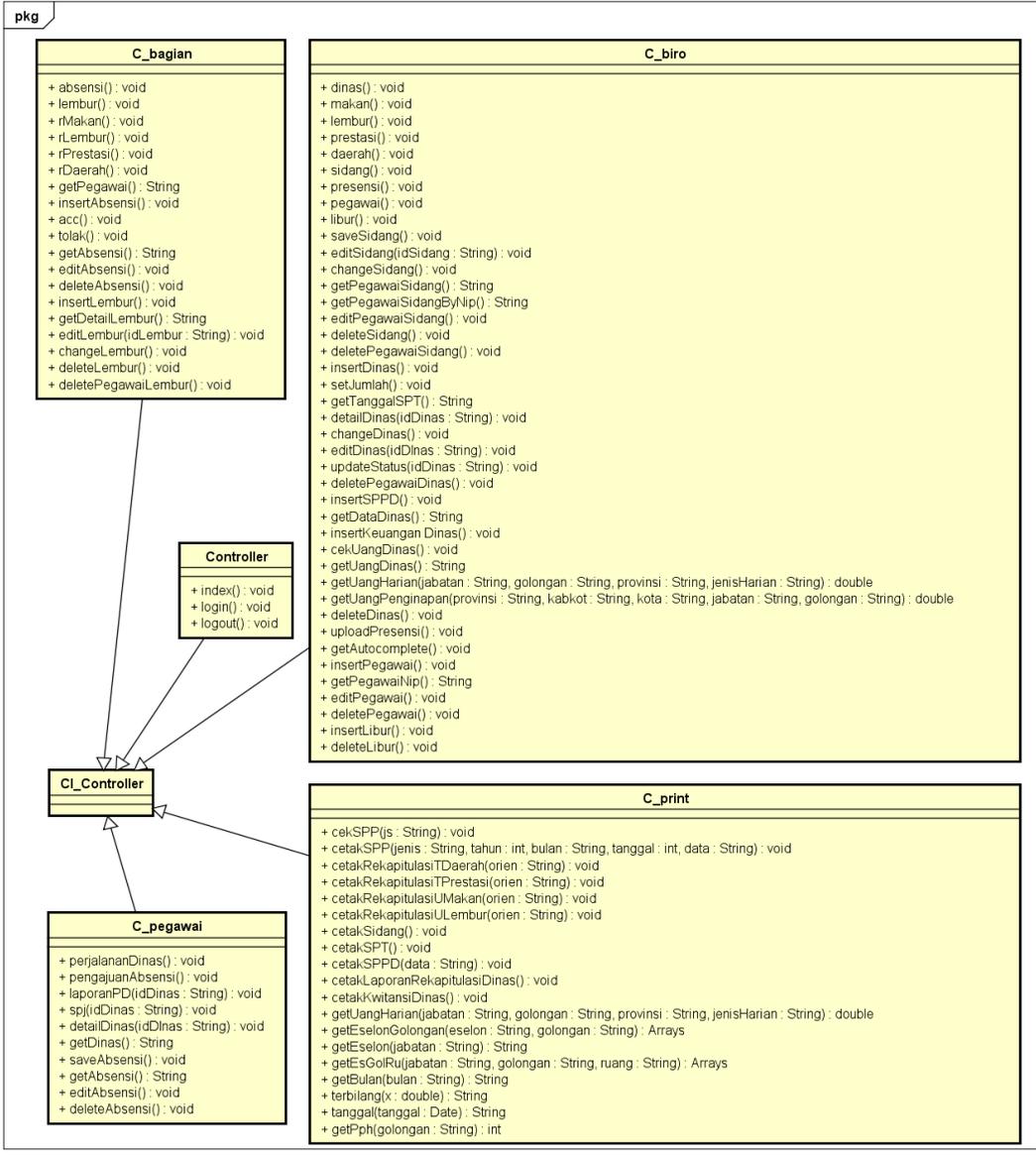
5.2 Pemodelan Objek

Pemodelan objek dilakukan untuk mendokumentasikan objek-objek yang terlibat dalam sistem yang akan dikembangkan. Pemodelan objek didokumentasikan menggunakan diagram kelas. Diagram kelas analisis pada bab analisis persyaratan akan menjadi panduan untuk membuat diagram kelas objek dan perancangan basis data yang nantinya akan diimplementasikan ke *database* sistem. Diagram kelas perancangan merupakan visualisasi kelas dan hubungan antara setiap kelas sebagai panduan untuk melakukan pembuatan sistem.

5.2.1 Diagram Kelas Perancangan

Diagram kelas analisis yang telah dibuat sebelumnya akan dikembangkan kedalam diagram kelas perancangan. Sistem informasi keuangan pegawai akan dikembangkan menggunakan *framework codeigniter* sehingga akan dimodelkan dalam dua kelas diagram. Diagram kelas perancangan yang akan dibuat yaitu kelas diagram *model* dan kelas diagram *controller*. Diagram kelas perancangan akan dibuat berdasarkan diagram kelas analisis dan objek-objek yang teridentifikasi pada *sequence diagram* dengan pola *MVC* pada *framework codeigniter*.

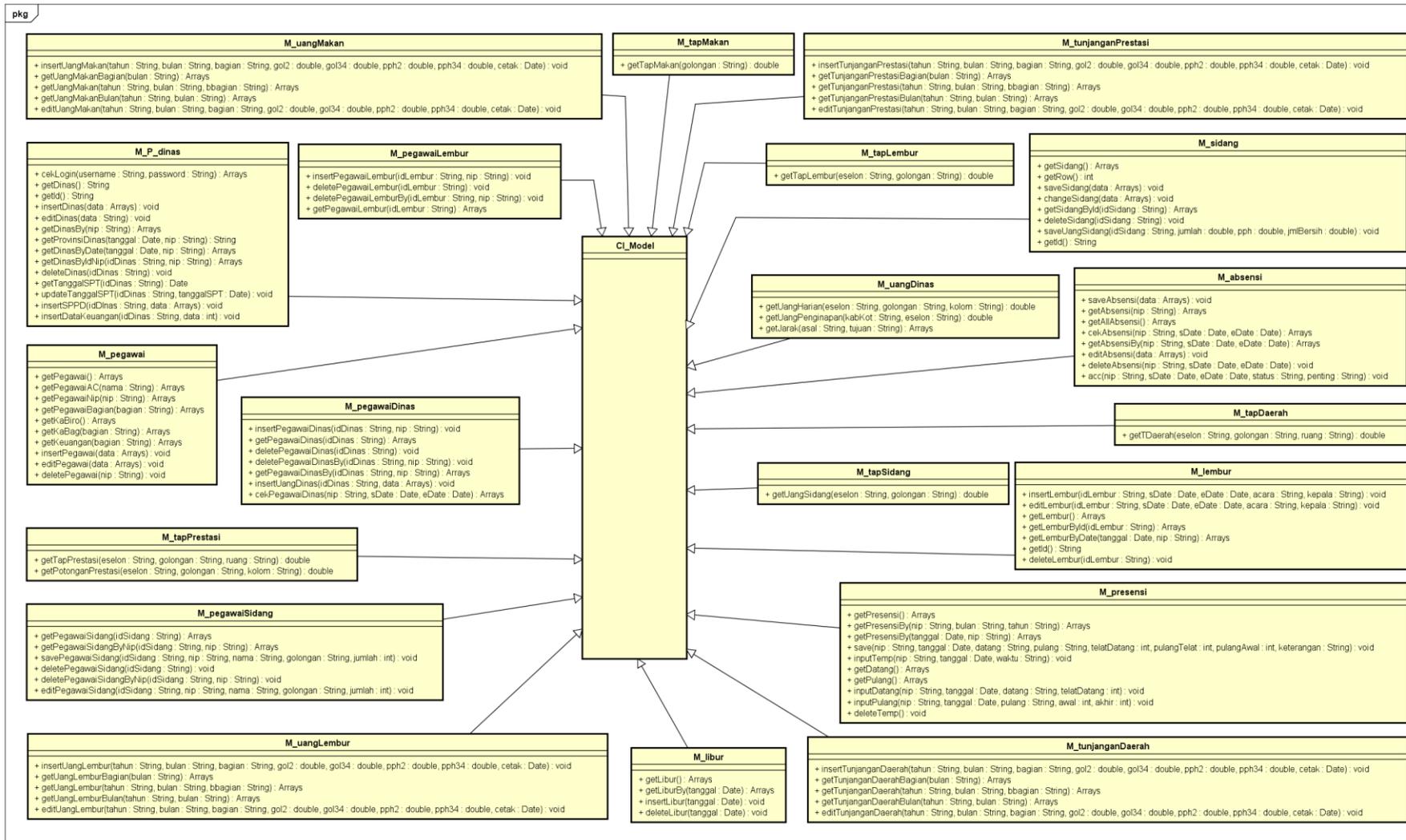
Diagram kelas perancangan *controller* dapat dilihat pada Gambar 5.7. Terdapat enam kelas yang satu diantaranya merupakan kelas induk yaitu kelas *CI_Controller* yang merupakan kelas bawaan dari *framework codeigniter*. Lima kelas lainnya merupakan kelas turunan yang diturunkan dari kelas *CI_controller* yaitu kelas *C_bagian*, *C_pegawai*, *C_biro*, *C_controller* dan *C_print*. Kelas *C_bagian* memuat fungsi logika yang dapat dilakukan oleh aktor bendahara bagian. Fungsi logika yang dapat dilakukan aktor bendahara biro terdapat dalam kelas *C_biro*. Kelas *C_pegawai* memuat fungsi logika yang dapat dilakukan oleh aktor pegawai. Fungsi logika mengenai pencetakan dokumen terdapat dalam kelas *C_print*. Sedangkan kelas *C_controller* memuat fungsi logika yang dapat dilakukan oleh seluruh aktor seperti otentikasi untuk masuk ke sistem.



Gambar 5. 7 Diagram Kelas Perancangan Controller

Diagram kelas perancangan model dapat dilihat pada Gambar 5.8. Terdapat 21 kelas yang satu diantaranya merupakan kelas induk yaitu kelas CI_Model yang merupakan kelas bawaan dari *framework codeigniter*. Dua puluh kelas lainnya merupakan kelas turunan yang diturunkan dari kelas CI_Model yaitu kelas M_absensi, M_dinas, M_pegawaiDinas, M_lembur, M_pegawaiLembur, M_sidang, M_pegawaiSidang, M_presensi, M_libur, M_pegawai, M_uangDinas, M_uangMakan, M_tunjanganPrestasi, M_tunjanganDaerah, M_uangLembur, M_tapMakan, M_tapDaerah, M_tapSidang, M_tapPrestasi dan M_tapLembur. Masing-masing kelas berisi fungsi untuk berinteraksi dengan basis data dan mengakses informasi didalamnya. Misalnya M_absensi untuk mengakses tabel absensi, M_uang dinas untuk mengakses tabel terkait uang dinas seperti tabel pd_bbm, pd_penginapan, pd_sewa dan pd_uangHarian.

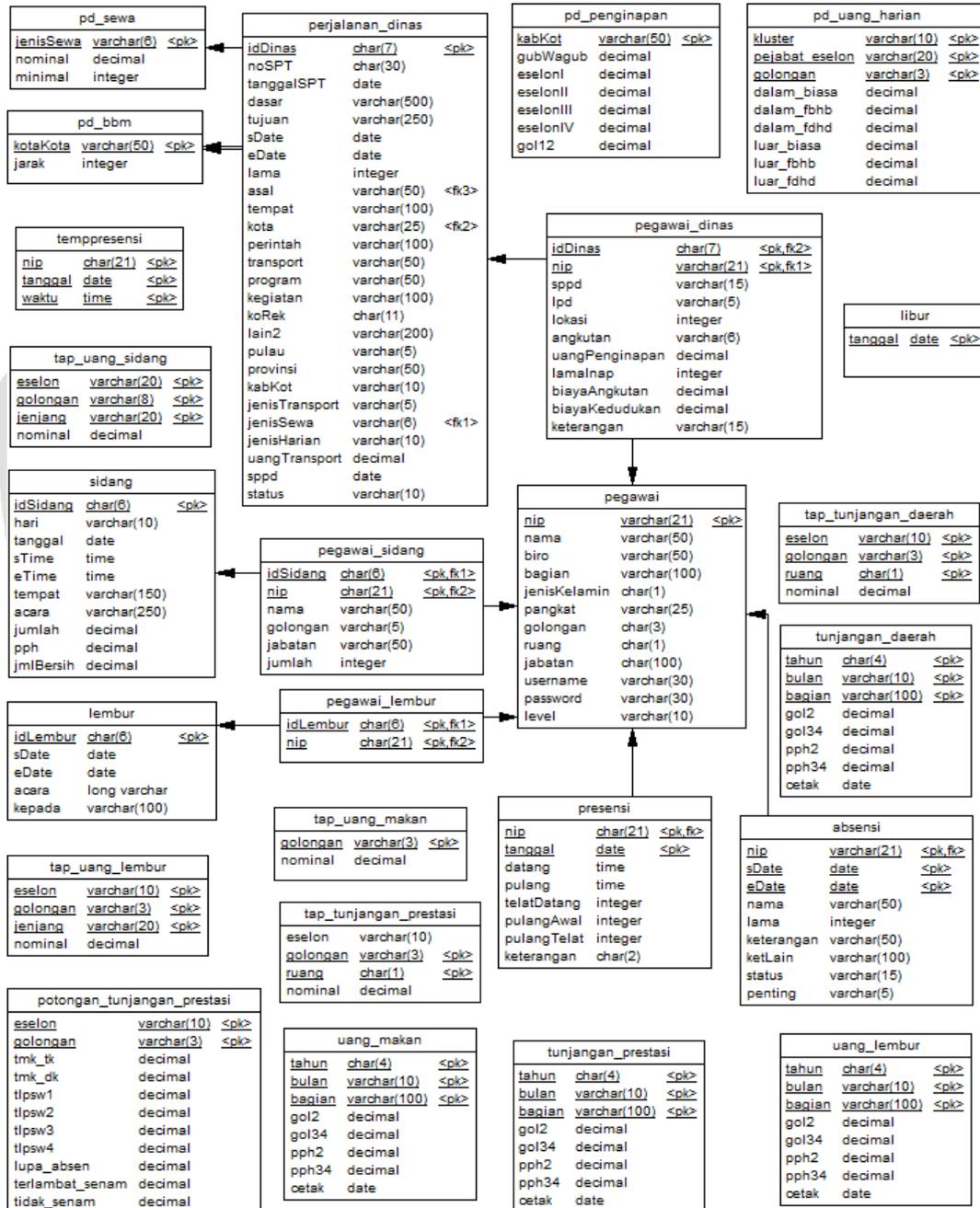




Gambar 5. 8 Diagram Kelas Perancangan Model

5.2.2 Perancangan Basis Data

Rancangan basis data dalam bentuk *physical data model* (PDM) pada Gambar 5.9 dibuat berdasarkan hasil pemodelan diagram kelas analisis pada bab analisis persyaratan. Masing-masing tabel pada PDM merepresentasikan kelas-kelas pada diagram analisis. Sedangkan atribut pada masing-masing kelas diagram analisis direpresentasikan kedalam nama kolom. Hubungan antar tabel PDM juga dibuat berdasarkan hubungan antar kelas pada diagram kelas analisis. Rancangan basis data ini nantinya akan diimplementasikan ke database sistem yang akan dibangun.



Gambar 5. 9 Physical Data Model SIKAP

Basis data sistem yang akan dibuat terdiri dari 25 tabel. Data mengenai perjalanan dinas pegawai tersimpan dalam tabel dinas, sedangkan ketentuan keuangan dinas dan keterangan lainnya tersimpan pada tabel pd_bbm, pd_penginapan, pd_sewa, pd_uang_harian, dan pegawai_dinas. Data terkait sidang tersimpan pada tabel sidang, pegawai_sidang, dan tap_uang_sidang. Tabel lembur, pegawai_lembur, tap_uang_lembur dan uang_lembur menyimpan data terkait lembur pegawai. Data terkait uang makan disimpan pada tabel tap_uang_makan dan uang_makan. Data terkait tunjangan prestasi disimpan pada tabel tap_tunjangan_prestasi, potongan_tunjangan_prestasi, dan tunjangan_prestasi. Sedangkan tabel tap_tunjangan_daerah dan tunjangan_daerah menyimpan data mengenai tunjangan daerah pegawai. Tabel absensi, presensi, pegawai dan libur menyimpan data pendukung yang dibutuhkan dalam perhitungan keuangan pegawai.

5.3 Perancangan Algoritme

Perancangan algoritme dilakukan untuk menyusun sekumpulan operasi logika sebagai konsep pembangunan fungsi-fungsi dari sistem yang akan dikembangkan. Perancangan algoritme akan didokumentasikan ke dalam *pseudocode*. *Pseudocode* yang telah dibuat akan menjadi acuan dalam implementasi sistem yang akan dikembangkan dalam tahap implementasi.

5.3.1 Algoritme Memasukkan Data Pengajuan Absensi

Menjelaskan algoritme dari fungsi yang akan dijalankan pada sistem ketika aktor akan melakukan aktivitas memasukkan data pengajuan absensi. Tabel 5.1 merupakan rancangan algoritme fungsi memasukkan data pengajuan absensi dalam bentuk *pseudocode*. Algoritme dimulai dengan menyimpan data yang diinputkan oleh aktor kedalam sebuah array kemudian mengecek apakah pegawai bersangkutan telah tercatat absen pada tanggal tersebut atau tidak. Jika tidak tercatat, maka akan memasukkan pengajuan absensi ke *database*. Apabila data pengajuan absensi gagal disimpan akan ada keterangan bahwa data pengajuan absensi gagal disimpan begitu pula sebaliknya

Tabel 5. 1 Pseudocode Memasukkan Data Absensi

Baris	Pseudocode
1	START
2	data <- READ (nip, nama, lama, sDate, eDate, keterangan, ketLain), 'menunggu', ''
3	cek <- READ (absensi) from database
4	IF (cek = 0) THEN
5	INSERT data to database
6	Ket <- 'Berhasil Diajukan'
7	ELSE
8	Ket <- 'Tidak dapat mengajukan Ijin! Anda telah mengajukan ijin pada tanggal tersebut.'
9	END IF
10	CALL function pengajuanAbsensi() to DISPLAY web page daftar pengajuan absensi
11	END

5.3.2 Algoritme Memasukkan Data Perjalanan Dinas

Menjelaskan algoritme dari fungsi yang akan dijalankan pada sistem ketika aktor akan melakukan aktivitas memasukkan data perjalanan dinas. Tabel 5.2 merupakan rancangan algoritme fungsi memasukkan data perjalanan dinas dalam bentuk *pseudocode*. Algoritme dimulai dengan pengambilan nilai idDinas terbesar dari *database*, kemudian menentukan nilai idDinas data yang akan dimasukkan. Selanjutnya data yang dimasukkan oleh aktor akan disimpan ke *database*. Dilanjutkan dengan menghitung jumlah pegawai yang mengikuti perjalanan dinas. Jika pegawai yang mengikuti perjalanan dinas lebih dari 0, maka akan memasukkan seluruh data pegawai dinas ke *database*. Apabila data pegawai dinas gagal disimpan akan ada keterangan bahwa data pegawai dinas gagal disimpan begitu pula sebaliknya.

Tabel 5. 2 Pseudocode Memasukkan Data Perjalanan Dinas

Baris	Pseudocode
1	START
2	noUrurt <- READ (maxIdDinas) from database
3	noUrut <- noUrut + 1
4	idDinas <- "D" + noUrut
5	data <- idDinas, READ (noSPT, tujuan, tempat, lama, sDate, eDate, dasar)
6	INSERT data to database
7	nip <- READ (nipP)
8	nama <- READ (namaP)
9	jumlah <- READ (jumlahP)
10	rows <- COUNT (nip)
11	ket <- "tidak dapat memasukkan pegawai"
12	cek2 <- 1
13	FOR (i <- 0 until (rows - 1))
14	cek <- READ pegawaiDinas from database
15	IF (cek = 0 AND nip i != "") THEN
16	INSERT idDinas, nip to database
17	ELSE IF (cek != 0) THEN
18	ket <- ket + nama
19	cek2 <- 0
20	END IF
21	Increment (i)
22	END FOR
23	IF (cek2 = 1) THEN
24	session ket = "Data Berhasil Disimpan"
25	END IF
26	CALL function dinas() to DISPLAY web page daftar perjalanan dinas
27	END

5.3.3 Algoritme Mengecek Data SPJ

Menjelaskan algoritme dari fungsi yang akan dijalankan pada sistem ketika aktor akan melakukan aktivitas memasukkan data perjalanan dinas. Tabel 5.3 merupakan rancangan algoritme fungsi memasukkan data SPPD dalam bentuk *pseudocode*. Algoritme dimulai dengan menangkap nilai p_idDinas kemudian di



simpan ke dalam variabel idDinas. Selanjutnya menyimpan data keuangan dinas yang diinputkan aktor kedalam *array* data. Sistem mengambil data perjalanan dinas dari *database* kemudian mengimpor nilainya ke *array* dinas. Selanjutnya mengambil data uangPenginapan dari *database* dan menyimpannya ke variabel tUangPenginapan. Kemudian dilakukan pengecekan kesesuaian antara uang penginapan yang diinputkan oleh aktor dengan standar pada *database*. Pengecekan pertama yaitu kesesuaian uang penginapan, kemudian pengecekan uang transportasi unuk wilayah D.K.I Jakarta, dilanjutkan dengan pengecekan uang transportasi dalam provinsi dan luar provinsi, yang terakhir yaitu pengecekan uang transport dalam pulau dan luar pulau.

Tabel 5. 3 Pseudocode Mengecek Data SPJ

Baris	Pseudocode
1	START
2	idDinas <- READ (p_idDinas)
3	data <- read (uangPenginapan, lamaInap, lokasi, angkutan, biayaAngkutan, biayaTaxi, biayaLokal)
4	keterangan <- sesuai
5	dinas <- READ (dinas) from database
6	tUangPenginapan <- READ (uangPenginapan) from database
7	session ket <- ""
8	session tId <- idDinas 1
9	IF (data uangPenginapan > tUangPenginapan) THEN
10	keterangan <- "tidak sesuai"
11	session ket <- session ket + "Uang penginapan melebihi standar sebesar" + (data uangPenginapan - tUangPenginapan)
12	END IF
13	IF (data provinsi = "D.K.I Jakarta") THEN
14	IF (data angkutan = "udara") THEN
15	tTransportasi = 5000000
16	ELSE
17	tTransportasi = 1200000
18	END IF
19	IF (data biayaAngkutan + data biayaTaxi + data biayaLokal > tTransportasi) THEN
20	keterangan <- "tidak sesuai"
21	session ket <- session ket + "Uang transport melebihi standar sebesar " + (data biayaAngkutan + data biayaTaxi + data biayaLokal - tTransportasi)
22	END IF
23	ELSE IF (dinas provinsi = "Jawa Timur") THEN
24	jarak <- READ (jarak) FROM DATABASE
25	IF (dinas kota CONTAIN "surabaya") THEN
26	IF (data lokasi = 1) THEN
27	tBiayaKedudukan <- 150000
28	ELSE IF (data lokasi = 2) THEN
29	tBiayaKedudukan <- 200000
30	ELSE
31	tBiayaKedudukan <- 250000
32	END IF
33	IF (data biayaTaxi + data biayaLokal > tBiayaKedudukan) THEN
34	keteranga <- "tidak sesuai"



Tabel 5.3 *Pseudocode* Mengecek Data SPJ (Lanjutan)

Baris	<i>Pseudocode</i>
35	session ket <- "Uang transport melebihi standar sebesar " + (data biayaTaxi + data biayaLokal - tBiayaKedudukan)
36	END IF
37	ELSE
38	IF (data angkutan = "darat") THEN
39	tBiayaAngkutan = 700000
40	ELSE
41	tBiayaAngkutan = 1500000
42	END IF
43	IF (data biayaAngkutan > tBiayaAngkutan) THEN
44	keterangan <- "tidak sesuai" session ket <- "Uang biayaAngkutan melebihi standar sebesar " + (data biayaAngkutan - tBiayaAngkutan)
45	END IF
46	IF (data biayaTaxi + data biayaLokal > tBiayaKedudukan) THEN
47	keterangan <- "tidak sesuai"
48	session ket <- "Uang transport melebihi standar sebesar " + (data biayaTaxi + data biayaLokal - tBiayaKedudukan)
49	END IF
50	END IF
51	ELSE
52	IF (dinas pulau = "dalam") THEN
53	tTransport = 700000
54	ELSE
55	tTransport = 1500000
56	END IF
57	IF (data biayaAngkutan + data biayaTaxi + data biayaLokal > tTransport) THEN
58	keterangan <- "tidak sesuai"
59	session ket <- "Uang transport melebihi standar sebesar " + (data biayaAngkutan + data biayaTaxi + data biayaLokal - tTransport)
60	END IF
61	END IF
62	data keterangan <- keterangan
63	INSERT idDinas, data to database
64	CALL function detailDinas() to DISPLAY web page detail dinas
65	END

5.3.4 Algoritme Membuat Laporan Rekapitulasi

Menjelaskan algoritme dari fungsi yang akan dijalankan pada sistem ketika aktor akan melakukan aktivitas membuat laporan rekapitulasi. Rancangan algoritme yang akan dibuat adalah rancangan algoritme membuat laporan rekapitulasi uang makan. Tabel 5.4 merupakan rancangan algoritme fungsi mencetak laporan rekapitulasi uang makan dalam bentuk *pseudocode*. Algoritme dimulai dengan membuat dokumen *pdf* untuk menampilkan laporan rekapitulasi uang makan. Selanjutnya menuliskan kop laporan rekapitulasi uang makan sesuai

bulan tahun dan bagian tertentu. Kemudian mengambil data pegawai pada bagian tertentu dari database dan menghitung uang makan masing-masing pegawai. Perhitungan uang makan pegawai berdasarkan kehadiran pegawai pada hari masuk kerja, kehadiran di hari libur baik hari sabtu, minggu atau hari libur nasional tidak mendapatkan uang makan. Dalam proses perhitungan uang makan akan ditentukan uang makan yang diperoleh masing-masing pegawai berdasarkan jabatan dan golongan serta potongan pph sesuai golongan pegawai bersangkutan. Setelah itu sistem akan menuliskan rekapitulasi uang makan seluruh pegawai bergolongan 2, kemudian pegawai bergolongan 3 dan 4, serta rekapitulasi seluruh pegawai pada bagian tersebut.

Tabel 5. 4 Pseudocode Membuat Laporan Rekapitulasi

Baris	Pseudocode
1	START
2	Make pdf document to preview laporan rekapitulasi uang makan
3	temp <- session level
4	bagian <- temp 1
5	tahun <- READ (tahun)
6	bulan <- READ (bulan)
7	strBulan <- CALL function getBulan(bulan)
8	WRITE header
9	WRITE colom name
10	No, totJumlah, totPph, totJumlahBersih, jmlGol2, pphGol2, jmlBersihGol2, jmlGol34, pphGol34, jmlBersihGol34, cek2 <- 0
11	pegawai <- READ data pegawai form database
12	Hari <- "01"
13	tAwal <- tahun + "-" + bulan + "-" + hari
14	tAkhir <- last day at that month
15	FOREACH pegawai DO
16	makan <- READ uang makan from database
17	uMakan, masuk <- 0
18	tgl <- tAwal
19	hari <- "01"
20	WHILE (tgl != tAkhir)DO
21	tgl <- tahun + "-" + bulan + "-" + hari
22	day <- cek day position in that week
23	libur <- READ data libur from database
24	IF (day != 0 AND day != 6 AND libur = 0) THEN
25	presensi <- READ presensi from database
26	IF (presensi = null) THEN
27	cek <- READ provinsi dinas from database
28	IF (cek = null OR cek != "Jawa Timur")
29	uMakan <- uMakan + makan
30	increment (masuk)
31	END IF
32	END IF
33	END IF
34	END WHILE
35	pph <- uMakan * getPph(pegawai golongan) / 100
36	jumlahBersih <- uMakan - pph
37	IF (pegawai golongan = "II") THEN
38	jmlGol2 <- uMakan
39	pphGol2 <- pph



Tabel 5.4 Pseudocode Membuat Laporan Rekapitulasi (Lanjutan)

Baris	Pseudocode
40	jmlBersihGol2 <- jumlahBersih
41	ELSE IF (pegawai golongan = "III" OR pegawai golongan = "IV") THEN
42	jmlGol34 <- uMakan
43	pphGol34 <- pph
44	jmlBersihGol34 <- jumlahBersih
45	END IF
46	totJumlah <- totJumlah + jumlahBersih
47	totPph <- totPph + pph
48	totJumlahBersih <- totJumlahBersi + jumlahBersih - pph
49	IF (pegawai golongan = "II" AND cek2 = 0) THEN
50	WRITE rekapitulation of golongan 2
51	END IF
52	WRITE the details of uang makan pegawai
53	END FOREACH
54	WRITE rekapitulation of golongan 3 and 4
55	WRITE rekapitulation of golongan 2
56	WRITE rekapitulation of golongan 3 and 4
57	WRITE rekapitulation of all golongan
58	kaBag <- READ data kepala bagian from database
59	keuang <- READ data pengelola keuangan from database
60	WRITE tandatangan pejabat
61	cek <- READ uang makan from database
62	cetak <- date of today
63	IF (cek > 0) THEN
64	EDIT data uang makan on database
65	ELSE
66	INSERT tahun, strBulan, bagian, jmlGol2, jmlGol34, pphGol2, pphGol34, cetak to database
67	END IF
68	DISPLAY pdf document
69	END

5.3.5 Algoritme Memasukkan Data Sidang

Menjelaskan algoritme dari fungsi yang akan dijalankan pada sistem ketika aktor akan melakukan aktivitas memasukkan data sidang. Tabel 5.5 merupakan rancangan algoritme fungsi memasukkan data sidang dalam bentuk *pseudocode*. Algoritme dimulai dengan pengambilan nilai idSidang terbesar dari *database*, kemudian menentukan nilai idSidang data yang akan dimasukkan. Selanjutnya data yang dimasukkan oleh aktor akan disimpan ke *database*. Dilanjutkan dengan menghitung jumlah pegawai yang mengikuti Sidang. Jika pegawai yang mengikuti sidang lebih dari 0, maka akan memasukkan seluruh data pegawai sidang ke *database*. Apabila data pegawai sidang gagal disimpan akan ada keterangan bahwa data pegawai gagal disimpan begitu pula sebaliknya.

Tabel 5. 5 Pseudocode Memasukkan Data Sidang

Baris	Pseudocode
1	START
2	noUrurt <- READ (maxIdDinas) from database
3	noUrut <- noUrut + 1



Tabel 5. 5 *Pseudocode* Memasukkan Data Sidang (Lanjutan)

4	idSidang <- "D" + noUrut
5	data <- idDinas, READ (tanggal, sTime, eTime, tempat, acara)
6	INSERT data to database
7	nipL <- READ (nipLT)
8	rows <- COUNT (nipL)
9	namaL <- READ (namaLT)
10	golonganL <- READ (golonganLT)
11	jabatanL <- READ (jabatanLT)
12	jumlahL <- READ (jumlahLT)
13	ket <- "Data Berhasil Disimpan"
14	ket2 <- ""
15	FOR (i <- 0 until (rows - 1))
16	cek <- READ pegawaiDinas from database
17	IF (cek = 0 AND nip i != "") THEN
18	INSERT idDinas, nipLT, namaLT, golonganLT, jabatanLT, jumlahLT to database
19	ELSE IF (cek != 0) THEN
20	Ket2 <- namaL i + nipL i
21	END IF
22	Increment (i)
23	END FOR
24	nip <- READ (nipP)
25	jumlah <- READ (jumlahP)
26	rows <- COUNT (nip)
27	FOR (i <- 0 until (rows - 1))
28	pegawai <- READ data pegawai from database
29	cek <- READ pegawai dinas from database
30	IF (cek = 0 AND nip i != "") THEN
31	Gol = pegawai golongan + "/" + pegawai ruang
32	INSERT idDinas, nip to database
33	ELSE IF (cek != 0) THEN
34	ket <- pegawai nama + "(" + nip i + ")"
35	END IF
36	Increment (i)
37	END FOR
38	IF (ket2 != "") THEN
39	session ket = ket2
40	ELSE
41	session ket = ket
42	END IF
43	CALL function sidang() to DISPLAY web page daftar sidang
44	END

5.3.6 Algoritme Mencetak Daftar Penerima dan Kwitansi Uang Sidang

Menjelaskan algoritme dari fungsi yang akan dijalankan pada sistem ketika aktor akan melakukan aktivitas mencetak daftar penerima dan kwitansi uang sidang. Tabel 5.6 merupakan rancangan algoritme fungsi mencetak daftar penerima dan kwitansi uang sidang dalam bentuk *pseudocode*. Algoritme dimulai dengan membuat dokumen *pdf* untuk menampilkan daftar penerima dan kwitansi uang sidang. Selanjutnya menuliskan kop daftar penerima uang. Kemudian menulis nama kolom tabel daftar penerima uang sidang. Sistem akan mengambil data pegawai yang mengikuti sidang tersebut dari database. Selanjutnya sistem akan menghitung uang sidang yang diterima oleh masing-masing pegawai. Setelah

sistem melakukan perhitungan uang sidang dan potongan pph pegawai serta menuliskan rinciannya pada tabel daftar penerima uang sidang, sistem akan mencetak tandatangan pejabat bersangkutan. Selanjutnya membuat halaman baru untuk kwitansi uang sidang. Kwitansi diawali dengan *header* kwitansi yang berisi kode kwitansi dan sebagainya. Kemudian di bagian konten terdapat nominal yang diterima dan diakhiri dengan tandatangan pejabat bersangkutan.

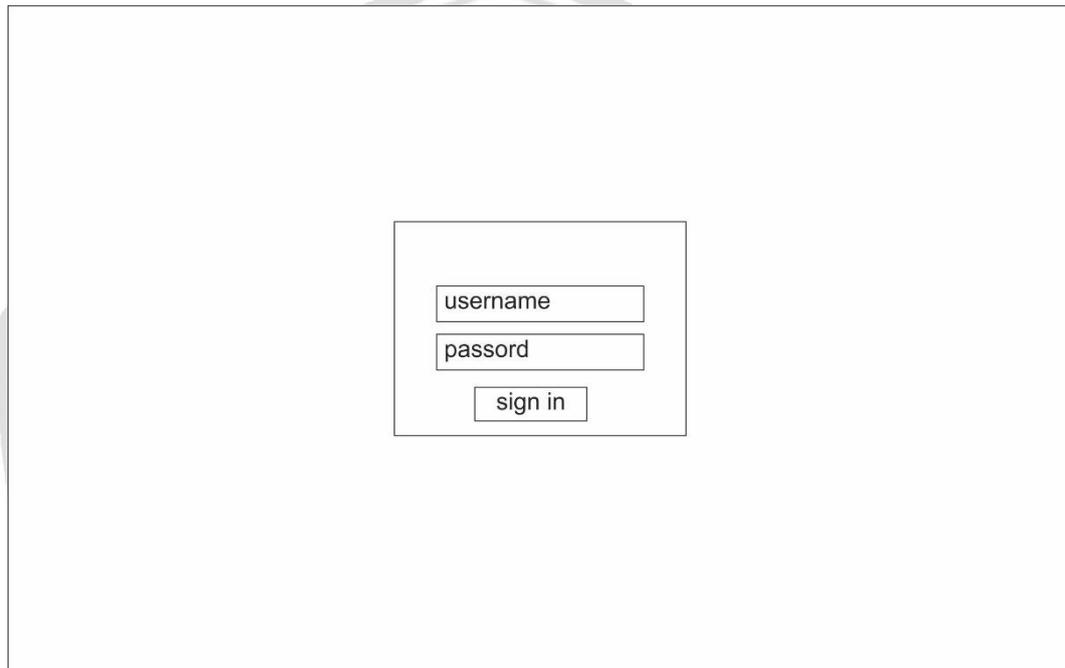
Tabel 5. 6 Pseudocode Membuat Laporan Rekapitulasi

Baris	Pseudocode
1	START
2	Make pdf document to preview daftar penerima dan kwitansi uang sidang
3	Data <- READ data sidang from database
4	WRITE header
5	WRITE colom name
6	no, totJumlah, totPph, totJumlahBersih, jmlGol2, pphGol2, jmlBersihGol2, jmlGol34, pphGol34, jmlBersihGol34, cek2 <- 0
7	pegawai <- READ data pegawai form database
8	FOREACH pegawai DO
9	Gol <- SPLIT pegawai golongan WITH "/"
10	esGol <- getEselonGolongan(pegawai jabatan, gol 0)
11	uangSidang <- Read uang sidang from database
12	jumlahUangSidang <- uangSidang * pegawai jumlah
13	pph <- jumlahUangSidang * getPph(gol 0) / 100
14	jumlahBersih <- jumlahUangSidang - pph
15	totJumlah <- totJumlah + jumlahBersih
16	totPph <- totPph + pph
17	totJumlahBersih = totJumlahBersih + jumlahBersih - pph
18	WRITE detail uang sidang pegawai
19	END FOREACH
20	WRITE rekapitulation of all pegawai
21	kaBag <- READ data kepala bagian from database
22	keuang <- READ data pengelola keuangan from database
23	WRITE tandatangan pejabat
24	INSERT idSidang, totJumlah, totPph, totJumlahBersih inti database
25	ADD new page
26	data <- READ data sidang from database
27	WRITE header
28	WRITE content
29	WRITE tandatangan pejabat
30	DISPLAY pdf document
31	END

5.4 Perancangan Antarmuka Pengguna

5.4.1 Antarmuka *Login*

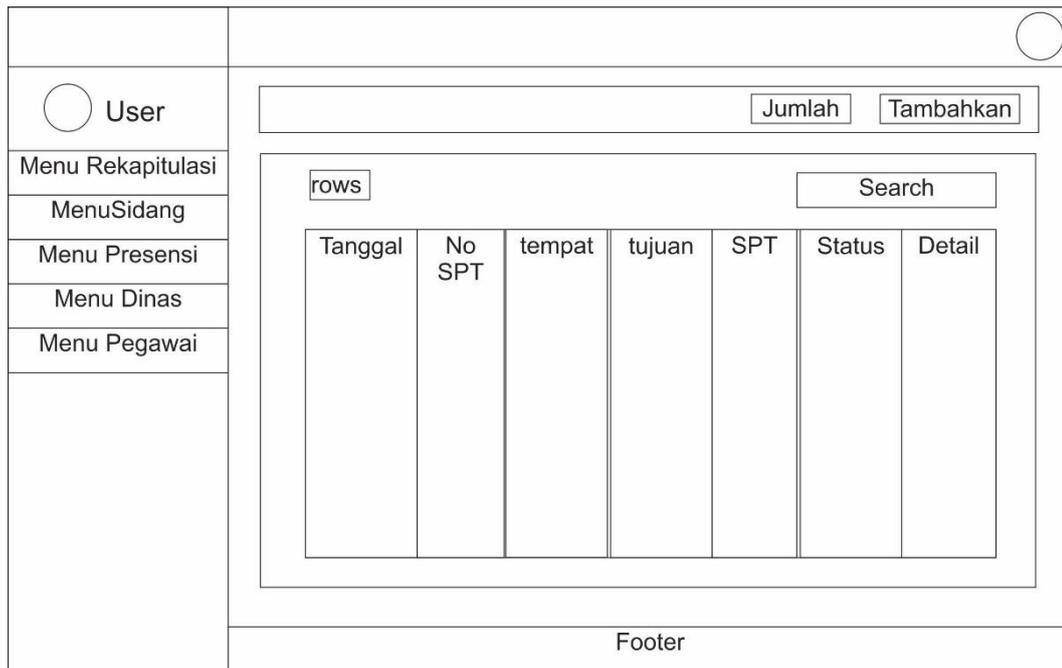
Rancangan antarmuka *login* pada Gambar 5.10 merupakan tampilan yang akan dilihat oleh pengguna ketika pertama membuka sistem untuk masuk kedalam sistem. Pada antarmuka ini pengguna akan melakukan proses otentikasi agar dapat mengakses informasi dalam sistem sesuai dengan hak aksesnya. Antarmuka *login* memiliki beberapa komponen yaitu kolom input *username* untuk memasukkan *username*, kolom input *password* untuk memasukkan *password* dan tombol login untuk menjalankan fungsi otentikasi pengguna dan masuk ke dalam sistem.

The image shows a simple login form centered on a white background. The form consists of three rectangular input fields stacked vertically. The top field is labeled 'username', the middle field is labeled 'password', and the bottom field is a button labeled 'sign in'. The labels are placed to the left of each respective field.

Gambar 5. 10 Rancangan Antarmuka *Login*

5.4.2 Antarmuka Daftar Perjalanan Dinas

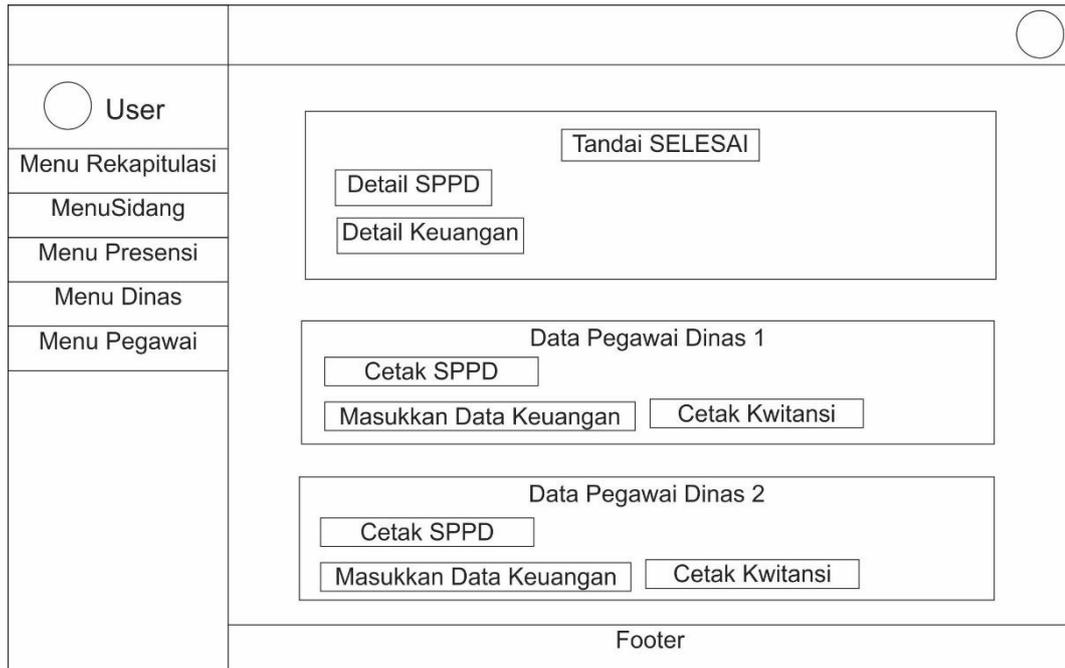
Rancangan antarmuka daftar perjalanan dinas pada Gambar 5.11 merupakan tampilan yang akan dilihat oleh pengguna ketika melihat daftar perjalanan dinas yang telah tercatat. Selain itu antarmuka ini juga dibuka oleh pengguna ketika akan memasukkan data perjalanan dinas atau akan melihat detail data perjalanan dinas. Antarmuka daftar perjalanan dinas memiliki beberapa komponen yaitu *icon user* di pojok kanan atas untuk melakukan *logout* dan *side bar* menu di bagian kiri untuk masuk ke menu yang ada pada sistem. Kemudian di bagian konten terdapat kolom jumlah untuk memasukkan jumlah pegawai yang akan melakukan perjalanan dinas, dan tombol “tambahkan” untuk menampilkan *form input* data perjalanan dinas dan menyimpannya ke *database* sistem. Kolom “rows” untuk mengatur banyak data yang ditampilkan dalam satu tampilan layar. Kemudian kolom “*search*” untuk memasukkan kata kunci dari data yang akan dicari pada tabel. Pada bagian konten juga terdapat tabel yang berisi daftar perjalanan dinas.



Gambar 5. 11 Rancangan Antarmuka Daftar Perjalanan Dinas

5.4.3 Antarmuka Detail Perjalanan Dinas

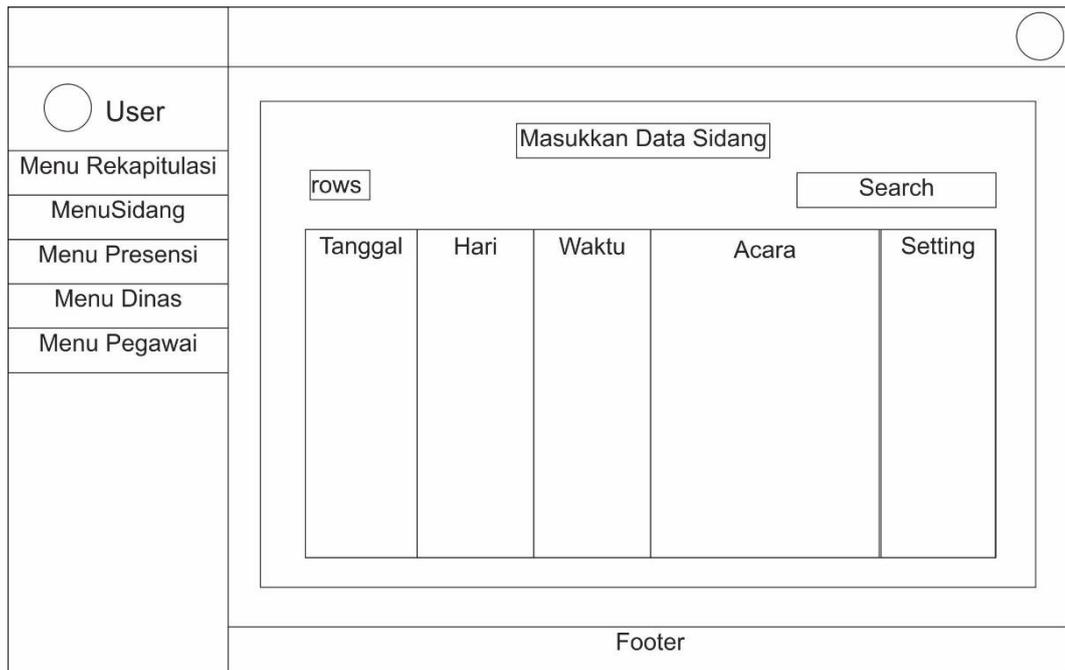
Rancangan antarmuka detail perjalanan dinas pada Gambar 5.12 merupakan tampilan yang akan dilihat oleh pengguna ketika melihat detail suatu perjalanan dinas. Selain itu antarmuka ini juga dibuka oleh pengguna ketika akan memasukkan data SPPD atau uang dinas, dan mencetak SPPD atau kwitansi uang perjalanan dinas. Antarmuka detail perjalanan dinas memiliki beberapa komponen yaitu *icon user* di pojok kanan atas untuk melakukan *logout* dan *side bar* menu di bagian kiri untuk masuk ke menu yang ada pada sistem. Kemudian di bagian konten terdapat tombol “tanda Selesai” untuk mengubah status perjalanan dinas menjadi selesai, kemudian tombol “detail SPPD” untuk menampilkan data SPPD dan *form input* data SPPD yang akan disimpan. Selain itu terdapat tombol “detail keuangan” untuk menampilkan data umum keuangan dinas dan *form input* data umum keuangan dinas yang akan disimpan. Pada bagian data pegawai dinas terdapat tombol “cetak SPPD” untuk mencetak SPPD masing-masing pegawai. Tombol “Masukkan data keuangan” untuk menampilkan *form input* data keuangan dinas masing-masing pegawai dan melakukan pengecekan terhadap data tersebut. Kemudian tombol “cetak kwitansi” untuk melakukan pencetakan kwitansi uang perjalanan dinas masing-masing pegawai. Jumlah tabel pegawai sesuai dengan jumlah pegawai yang bertugas pada suatu perjalanan dinas.



Gambar 5. 12 Rancangan Antarmuka Detail Perjalanan Dinas

5.4.4 Antarmuka Daftar Sidang

Rancangan antarmuka daftar sidang pada Gambar 5.12 merupakan tampilan yang akan dilihat oleh pengguna ketika melihat daftar sidang yang telah tercatat. Selain itu antarmuka ini juga dibuka oleh pengguna ketika akan memasukkan data sidang atau akan melihat detail data sidang. Antarmuka daftar sidang memiliki beberapa komponen yaitu *icon user* di pojok kanan atas untuk melakukan *logout* dan *side bar* menu di bagian kiri untuk masuk ke menu yang ada pada sistem. Kemudian di bagian konten terdapat tombol “Masukkan Data Sidang” untuk menampilkan *form input* data sidang dan menyimpannya ke *database* sistem. Kolom “rows” untuk mengatur banyak data yang ditampilkan dalam satu tampilan layar. Kemudian kolom “search” untuk memasukkan kata kunci dari data yang akan dicari pada tabel. Pada bagian konten juga terdapat tabel yang berisi daftar sidang.



Gambar 5. 13 Rancangan Antarmuka Daftar Sidang

5.4.5 Antarmuka Daftar Laporan Rekapitulasi

Rancangan antarmuka daftar laporan rekapitulasi pada Gambar 5.13 merupakan tampilan yang akan dilihat oleh pengguna ketika melihat daftar laporan rekapitulasi uang makan, uang lembur, tunjangan daerah, atau tunjangan prestasi yang telah tercatat. Selain itu antarmuka ini juga dibuka oleh pengguna ketika akan mencetak laporan rekapitulasi. Antarmuka daftar laporan rekapitulasi memiliki beberapa komponen yaitu *icon user* di pojok kanan atas untuk melakukan *logout* dan *side bar* menu di bagian kiri untuk masuk ke menu yang ada pada sistem. Kemudian di bagian konten terdapat komponen cetak laporan rekapitulasi yang didalamnya terdapat kolom “tahun” untuk memasukkan tahun laporan, kolom “bulan” untuk memasukkan bulan laporan dan tombol “*cetak*” untuk mencetak laporan rekapitulasi sesuai bilan dan tahun. Pada komponen daftar laporan rekapitulasi terdapat kolom “rows” untuk mengatur banyak data yang ditampilkan dalam satu tampilan layar. Kemudian kolom “*search*” untuk memasukkan kata kunci dari data yang akan dicari pada tabel. Pada bagian ini juga terdapat tabel yang berisi daftar laporan rekapitulasi.

<input type="radio"/> User Menu Absensi Menu Lembur Menu Rekapitulasi	○									
	Upload Laporan Rekapitulasi									
	Tahun			Bulan						
	Cetak									
Daftar Rekapitulasi										
rows			Search							
Tahun	Bulan	Gol II	Gol III,IV	Jumlah	Pph Gol II	Pph Gol III,IV	Jumlah	Jumlah Bersih	Tanggal Cetak	
Footer										

Gambar 5. 14 Rancangan Antarmuka Daftar Laporan Rekapitulasi



BAB 6 IMPLEMENTASI

Bab implementasi menjelaskan mengenai kebutuhan dan hasil implementasi sistem informasi yang dikembangkan. Bab ini menjelaskan kegiatan pada fase konstruksi dari metode pengembangan RUP dalam mengembangkan Sistem Informasi Keuangan Pegawai (SIKAP). Implementasi dilakukan berdasarkan hasil perancangan pada bab perancangan. Bab implementasi terdiri dari subbab yang membahas mengenai spesifikasi lingkungan implementasi, implementasi dari beberapa algoritme, dan implementasi dari beberapa antarmuka pengguna.

6.1 Spesifikasi Lingkungan Sistem Informasi

Subbab ini menjelaskan mengenai spesifikasi perangkat keras dan perangkat lunak yang digunakan selama proses pengembangan Sistem Informasi Keuangan Pegawai (SIKAP). Spesifikasi perangkat keras dijelaskan pada Tabel 6.1. Tabel 6.2 menjelaskan spesifikasi perangkat lunak. Sedangkan spesifikasi minimal lingkungan *deployment* dijelaskan pada Tabel 6.3

Tabel 6. 1 Spesifikasi Perangkat Keras

Unit Komputasi	Laptop Asus X550D
CPU	AMD A10-5750M 2.5 GHz
Kapasitas RAM	4 GB
Kapasitas Penyimpanan	1 TB
Kartu Grafis	Radeon Graphics HD 8670M
Resolusi Layar	1366 x 768 pixels

Tabel 6. 2 Spesifikasi Perangkat Lunak

Sistem Operasi	Windows 10 Pro 64-bit
Bahasa Pemrograman	PHP HTML Javascript
Editor Kode Program	Sublime text 3
Perangkat Lunak Pendukung	Xampp Google Chrome Astah PowerDesigner Bizagi Corel Draw

Tabel 6.2 Spesifikasi Perangkat Lunak (Lanjutan)

Librari Pendukung	JQuery ExcelReader Fpdf bootstrap font-owsome Ionicons datetimepicker datatable CodeIgniter (CI) JQuery-ui
--------------------------	---

Tabel 6. 3 Spesifikasi Minimal Lingkungan *Deployment*

Kapasitas RAM minimal	64 MB
Kapasitas Memori	750 MB

6.2 Implementasi Algoritma

Subbab implementasi algoritma berisi hasil implementasi program berdasarkan algoritme yang telah dibuat pada bab perancangan. Beberapa implementasi algoritme yaitu algoritme memasukkan data pengajuan absensi, memasukkan data perjalanan dinas, memasukkan data SPPD, dan mengecek data SPJ.

6.2.1 Implementasi Algoritme Memasukkan Data Pengajuan Absensi

Algoritme memasukkan data pengajuan absensi dalam bentuk *pseudocode* pada Tabel 5.1 diimplementasikan kedalam kode program dalam bahasa pemrograman PHP. Hasil implementasi dapat dilihat pada Tabel 6.4.

Tabel 6. 4 Implementasi Algoritma Memasukkan Data Pengajuan Absensi

Baris	Kode Program
1	<code>public function saveAbsensi(){</code>
2	<code> \$data = array(</code>
3	<code> 'nip' => \$this->input->post('nip'),</code>
4	<code> 'nama' => \$this->input->post('nama'),</code>
5	<code> 'lama' => \$this->input->post('lama'),</code>
6	<code> 'sDate' => \$this->input->post('sDate'),</code>
7	<code> 'eDate' => \$this->input->post('eDate'),</code>
8	<code> 'keterangan' => \$this->input->post('keterangan'),</code>
9	<code> 'ketLain' => \$this->input->post('ketLain'),</code>
10	<code> 'status' => 'menunggu',</code>
11	<code> 'penting' => ''</code>
12	<code>);</code>

Tabel 6.4 Implementasi Algoritma Memasukkan Data Pengajuan Absensi (Lanjutan)

Baris	Kode Program
13	<code>\$cek = \$this->M_absensi->cekAbsensi(\$data['nip'],</code>
14	<code>\$data['sDate'],\$data['eDate']->num_rows();</code>
15	<code>if (\$cek == 0) {</code>
16	<code> \$this->M_absensi->saveAbsensi(\$data);</code>
17	<code> \$this->session->set_userdata('ket', "Berhasil</code>
18	<code> diajukan");</code>
19	<code> }else{</code>
20	<code> \$this->session->set_userdata('ket', "Tidak dapat</code>
21	<code> mengajukan Ijin! Anda telah mengajukan ijin pada</code>
22	<code> tanggal tersebut.");</code>
23	<code> }</code>
24	<code> \$this->pengajuanAbsensi();</code>
25	<code>}</code>

6.2.2 Implementasi Algoritme Memasukkan Data Perjalanan Dinas

Algoritme memasukkan data perjalanan dinas dalam bentuk *pseudocode* pada Tabel 5.2 diimplementasikan kedalam kode program dalam bahasa pemrograman PHP. Hasil implementasi dapat dilihat pada Tabel 6.5.

Tabel 6.5 Implementasi Algoritma Memasukkan Data Perjalanan Dinas

Baris	Kode Program
1	<code>Function insertDinas() {</code>
2	<code> \$noUrut = (int) substr(\$this->M_dinas->getID(), 1);</code>
3	<code> \$noUrut++;</code>
4	<code> \$char = "D";</code>
5	<code> \$idDinas = \$char . sprintf("%06s", \$noUrut);</code>
6	<code> \$data = array(</code>
7	<code> 'idDinas' => \$idDinas,</code>
8	<code> 'noSPT' => \$this->input->post('noSPT'),</code>
9	<code> 'tujuan' => \$this->input->post('tujuan'),</code>
10	<code> 'tempat' => \$this->input->post('tempat'),</code>
11	<code> 'lama' => \$this->input->post('lama'),</code>
12	<code> 'sDate' => \$this->input->post('sDate'),</code>
13	<code> 'eDate' => \$this->input->post('eDate'),</code>
14	<code> 'dasar' => \$this->input->post('dasar')</code>
15	<code>);</code>
16	<code> \$this->M_dinas->insertDinas(\$data);</code>
17	<code> \$ket = "Tidak dapat memasukkan pegawai ";</code>
18	<code> \$cek2 = 1;</code>
19	<code> for (\$i=1; \$i <= \$this->input->post('jml'); \$i++) {</code>
20	<code> \$nip = \$this->input->post('nip' . \$i);</code>
21	<code> \$cek = \$this->M_pegawaiDinas->cekPegawaiDinas(\$nip,</code>
22	<code> \$data['sDate'],\$data['eDate']->num_rows();</code>
23	<code> if (\$cek == 0 AND \$nip != "") {</code>

Tabel 6.5 Implementasi Algoritma Memasukkan Data Perjalanan Dinas (Lanjutan)

Baris	Kode Program
23	<code>\$this->M_pegawaiDinas-</code>
	<code>>insertPegawaiDinas(\$idDinas, \$nip);</code>
24	<code>}else{</code>
25	<code>\$ket = \$ket.' '.\$this->input->post('nama'.\$i);</code>
26	<code>\$this->session->set_userdata('ket', \$ket);</code>
27	<code>\$cek2 = 0;</code>
28	<code>}</code>
29	<code>}</code>
30	<code>if (\$cek2 == 1) {</code>
31	<code>\$this->session->set_userdata('ket', 'Data Berhasil</code>
	<code>Disimpan');</code>
32	<code>}</code>
33	<code>\$this->dinas();</code>
34	<code>}</code>

6.2.3 Implementasi Algoritme Mengecek Data SPJ

Algoritme mengecek data SPJ dalam bentuk *pseudocode* pada Tabel 5.3 diimplementasikan kedalam kode program dalam bahasa pemrograman PHP. Hasil implementasi dapat dilihat pada Tabel 6.6.

Tabel 6. 6 Implementasi Algoritme Mengecek Data SPJ

Baris	Kode Program
1	<code>function cekUangDinas(){</code>
2	<code>\$idDinas = explode('@', \$this->input-</code>
3	<code>>post('p_idDinas');</code>
4	<code>\$data = array(</code>
5	<code>'uangPenginapan' => \$this->input-</code>
	<code>>post('uangPenginapan'),</code>
6	<code>'lamaInap' => \$this->input->post('lamaInap'),</code>
7	<code>'lokasi' => \$this->input->post('lokasi'),</code>
8	<code>'angkutan' => \$this->input->post('angkutan'),</code>
9	<code>'biayaAngkutan' => \$this->input-</code>
	<code>>post('biayaAngkutan'),</code>
10	<code>'biayaTaxi' => \$this->input->post('biayaTaxi'),</code>
11	<code>'biayaLokal' => \$this->input->post('biayaLokal')</code>
12	<code>);</code>
13	<code>\$keterangan = 'sesuai';</code>
14	<code>\$dinas = \$this->M_dinas->getDinasByIdNip(\$idDinas[0],</code>
	<code>\$idDinas[1]);</code>
15	<code>\$tUangPenginapan = \$this->getUangPenginapan(</code>
	<code>\$dinas['provinsi'],\$dinas['kabKot'],\$dinas['kota'],\$din</code>
16	<code>as['jabatan'],\$dinas['golongan']);</code>
17	<code>\$this->session->set_userdata('ket','');</code>



Tabel 6.6 Implementasi Algoritme Mengecek Data SPJ (Lanjutan)

Baris	Kode Program
18	<code>\$this->session->set_userdata('tId',\$idDinas[1]);</code>
19	<code>if (\$data['uangPenginapan'] > \$tUangPenginapan) {</code>
20	<code> \$keterangan = 'tidak sesuai';</code> <code> \$this->session->ket += ' ' . 'Uang penginapan melebihi standar sebesar ' . (\$data['uangPenginapan'] - \$tUangPenginapan) . ' ';</code>
21	<code>}</code>
22	<code>if (\$dinas['provinsi'] == 'D.K.I Jakarta') {</code>
23	<code> if (\$data['angkutan'] == 'udara') {</code>
24	<code> \$tTransportasi = 5000000;</code>
25	<code> }else{</code>
26	<code> \$tTransportasi = 1200000;</code>
27	<code> }</code>
28	<code> if (\$data['biayaAngkutan']+\$data['biayaLokal']+\$data['biayaTaxi'] > \$tTransportasi) {</code>
29	<code> \$keterangan = 'tidak sesuai';</code>
30	<code> \$this->session->ket = \$this->session->ket . ' ' . 'Uang transport melebihi standar sebesar ' . ((\$data['biayaAngkutan']+\$data['biayaLokal']+\$data['biayaTaxi'])-\$tTransportasi) . ' ';</code>
31	<code> }</code>
32	<code> }elseif (\$dinas['provinsi'] == 'Jawa Timur') {</code>
33	<code> \$jarak = \$this->M_uangDinas->getJarak(\$dinas['asal'], \$dinas['kota']);</code>
34	<code> if ((int)\$jarak <= 40) {</code>
35	<code> if (\$data['lokasi'] == '1') {</code>
36	<code> \$tBiayaKedudukan = 150000;</code>
37	<code> }elseif (\$data['lokasi'] == '2') {</code>
38	<code> \$tBiayaKedudukan = 200000;</code>
39	<code> }else{</code>
40	<code> \$tBiayaKedudukan = 250000;</code>
41	<code> }</code>
42	<code> if (\$data['biayaLokal']+\$data['biayaTaxi'] > \$tBiayaKedudukan) {</code>
43	<code> \$keterangan = 'tidak sesuai';</code>
44	<code> \$this->session->ket = \$this->session->ket . ' ' . 'Uang transport melebihi standar sebesar ' . (\$data['biayaLokal']+\$data['biayaTaxi']-\$tBiayaKedudukan) . ' ';</code>
45	<code> }</code>
46	<code> }else{</code>
47	<code> if (\$data['angkutan'] == 'darat') {</code>
48	<code> \$tBiayaAngkutan = 700000;</code>
49	<code> }else{</code>
50	<code> \$tBiayaAngkutan = 1500000;</code>
51	<code> }</code>
52	<code> if (\$data['biayaAngkutan'] > \$tBiayaAngkutan) {</code>
53	<code> \$keterangan = 'tidak sesuai';</code>



Tabel 6.6 Implementasi Algoritme Mengecek Data SPJ (Lanjutan)

Baris	Kode Program
54	<code>\$this->session->ket = \$this->session->ket.' '.'Uang biaya angkutan melebihi standar sebesar '.(\$tBiayaAngkutan- \$data['biayaAngkutan']).' ';</code>
55	<code>}</code>
56	<code>if (\$data['biayaLokal']+\$data['biayaTaxi'] > 400000) {</code>
57	<code>\$keterangan = 'tidak sesuai';</code>
58	<code>\$this->session->ket = \$this->session->ket.' '.'Uang biaya kedudukan-tujuan melebihi standar sebesar '.(400000- \$data['biayaLokal']+\$data['biayaTaxi']).' ';</code>
59	<code>}</code>
60	<code>}</code>
61	<code>}else{</code>
62	<code>if (\$dinas['pulau'] = 'dalam') {</code>
63	<code>\$tTransport = 700000;</code>
64	<code>}else{</code>
65	<code>\$tTransport = 1500000;</code>
66	<code>}</code>
67	<code>if (\$data['biayaAngkutan']+\$data['biayaLokal']+ \$data['biayaTaxi'] > \$tTransport) {</code>
68	<code>\$keterangan = 'tidak sesuai';</code>
69	<code>\$this->session->ket = \$this->session->ket.' '.'Uang transport melebihi standar sebesar '.(\$tTransport-(\$data['biayaAngkutan']+ \$data['biayaLokal']+\$data['biayaTaxi']).' ';</code>
70	<code>}</code>
71	<code>}</code>
72	<code>\$data['keterangan'] = \$keterangan;</code>
73	<code>\$this->M_pegawaiDinas->insertUangDinas(\$idDinas, \$data);</code>
74	<code>\$this->detailDinas(\$idDinas[0]);</code>
75	<code>}</code>
76	<code>}</code>

6.2.4 Implementasi Algoritme Membuat Laporan rekapitulasi

Algoritme membuat laporan rekapitulasi dalam bentuk *pseudocode* pada Tabel 5.4 diimplementasikan kedalam kode program dalam bahasa pemrograman PHP. Hasil implementasi dapat dilihat pada Tabel 6.7.

Tabel 6. 7 Implementasi Algoritme Membuat Laporan Rekapitulasi

Baris	Kode Program
1	function cetakRekapitulasiUMakan(\$orien) {
2	\$params = array('orientation' => \$orien, 'unit' => 'mm', 'size' => array(210, 330));
3	\$this->load->library('fpdf_gen', \$params);
4	\$temp = explode('/', \$this->session->userdata('level'));
5	\$bagian = 'Bagian '.\$temp[1];
6	\$tahun = \$this->input->post('tahun');
7	\$bulan = \$this->input->post('bulan');
8	\$strBulan = \$this->getBulan(\$bulan);
9	\$this->fpdf->SetAutoPageBreak(true,1);
10	\$this->fpdf->SetFont('Arial','B',12);
11	\$this->fpdf->setXY(20,10);
12	\$this->fpdf->MultiCell(175,5,'UANG MAKAN KARYAWAN / KARYAWATI PADA '.strtoupper(\$bagian).' BULAN : ' .strtoupper(\$strBulan).' '.\$tahun,0,'C'); \$this->fpdf->Ln(4);
13	\$yStart = \$this->fpdf->GetY();
14	\$this->fpdf->SetFont("Arial","",8);
15	\$y = \$yStart;
16	\$this->fpdf->Line(10,\$y,204,\$y);
17	\$this->fpdf->Line(9.5,\$y-0.5,204.5,\$y-0.5);
18	\$this->fpdf->MultiCell(8,16,'NO',0,'C');
19	\$this->fpdf->SetXY(18,\$y);
20	\$this->fpdf->MultiCell(55,16,'NAMA',0,'C');
21	\$this->fpdf->SetXY(73,\$y);
22	\$this->fpdf->MultiCell(15,16,'GOL',0,'C');
23	\$this->fpdf->SetXY(88,\$y+4);
24	\$this->fpdf->MultiCell(12,4,'JMLH HARI',0,'C');
25	\$this->fpdf->SetXY(100,\$y);
26	\$this->fpdf->MultiCell(18,4,'UANG MAKAN 1 (SATU) HARI',0,'C');
27	\$this->fpdf->SetXY(118,\$y+2); \$this->fpdf->MultiCell(22,4,'JUMLAH UANG MAKAN (4X5)',0,'C');
28	\$this->fpdf->SetXY(140,\$y+2);
29	\$this->fpdf->MultiCell(18,4,'POTONGAN PAJAK PPh 21',0,'C');
30	\$this->fpdf->SetXY(158,\$y+2);
31	\$this->fpdf->MultiCell(21,4,'JUMLAH PENERIMAAN BERSIH',0,'C');
32	\$this->fpdf->SetXY(179,\$y);
33	\$this->fpdf->MultiCell(25,16,'TANDA TANGAN',0,'C');
34	
35	\$ss = 4;
36	\$y = \$this->fpdf->GetY();
37	\$this->fpdf->Line(10,\$y,204,\$y);



Tabel 6.7 Implementasi Algoritme Membuat Laporan Rekapitulasi (Lanjutan)

Baris	Kode Program
38	<code>\$this->fpdf->MultiCell(8,\$s,'1',0,'C');</code>
39	<code>\$this->fpdf->SetXY(18,\$y);</code>
40	<code>\$this->fpdf->MultiCell(55,\$s,'2',0,'C');</code>
41	<code>\$this->fpdf->SetXY(73,\$y);</code>
42	<code>\$this->fpdf->MultiCell(15,\$s,'3',0,'C');</code>
43	<code>\$this->fpdf->SetXY(88,\$y);</code>
44	<code>\$this->fpdf->MultiCell(12,\$s,'4',0,'C');</code>
45	<code>\$this->fpdf->SetXY(100,\$y);</code>
46	<code>\$this->fpdf->MultiCell(18,\$s,'5',0,'C');</code>
47	<code>\$this->fpdf->SetXY(118,\$y);</code>
48	<code>\$this->fpdf->MultiCell(22,\$s,'6',0,'C');</code>
49	<code>\$this->fpdf->SetXY(140,\$y);</code>
50	<code>\$this->fpdf->MultiCell(18,\$s,'7',0,'C');</code>
51	<code>\$this->fpdf->SetXY(158,\$y);</code>
52	<code>\$this->fpdf->MultiCell(21,\$s,'8',0,'C');</code>
53	<code>\$this->fpdf->SetXY(179,\$y);</code>
54	<code>\$this->fpdf->MultiCell(25,\$s,'9',0,'C');</code>
55	<code>\$y = \$this->fpdf->GetY();</code>
56	<code>\$this->fpdf->Line(10,\$y,204,\$y);</code>
57	<code>\$this->fpdf->SetFont('Arial','',10);</code>
58	<code>\$no = 1;</code>
59	<code>\$totJumlah = 0;</code>
60	<code>\$totPph = 0;</code>
61	<code>\$totJumlahBersih = 0;</code>
62	<code>\$jmlGol2 = 0;</code>
63	<code>\$pphGol2 = 0;</code>
64	<code>\$jmlBersihGol2 = 0;</code>
65	<code>\$jmlGol34 = 0;</code>
66	<code>\$pphGol34 = 0;</code>
67	<code>\$jmlBersihGol34 = 0;</code>
68	<code>\$pegawai = \$this->M_pegawai->getPegawaiBagian(\$bagian);</code>
69	<code>\$cek2 = 0;</code>
70	<code>\$y = \$this->fpdf->GetY()+4;</code>
71	<code>\$hari = '01';</code>
72	<code>\$tAwal = \$tahun.'-'. \$bulan.'-'. \$hari;</code>
73	<code>\$tAkhir = date('Y-m-t', strtotime(\$tAwal));</code>
74	<code>foreach (\$pegawai->result_array() as \$key) {</code>
75	<code> \$makan = \$this->M_tapMakan->getUMakan(\$key['golongan']);</code>
76	<code> \$uMakan = 0;</code>
77	<code> \$masuk = 0;</code>
78	<code> \$tgl = \$tAwal;</code>
79	<code> \$hari = '01';</code>
80	<code> while (\$tgl != \$tAkhir) {</code>
81	<code> \$tgl = \$tahun.'-'. \$bulan.'-'. \$hari;</code>
82	<code> \$day = date("w", mktime(0, 0, 0, \$bulan, \$hari++, \$tahun));</code>

Tabel 6.7 Implementasi Algoritme Membuat Laporan Rekapitulasi (Lanjutan)

Baris	Kode Program
83	<code>\$libur = \$this->M_libur->getLiburBy(\$tgl)->num_rows();</code>
84	<code>if (\$day != 0 AND \$day != 6 AND \$libur == 0) {</code>
85	<code> \$presensi = \$this->M_presensi->GetPresensiByDate(\$tgl, \$key['nip']);</code>
86	<code> if (\$presensi != null) {</code>
	<code> \$cek = \$this->M_dinas->getProvinsiDinas(\$tgl,\$key['nip']);</code>
87	<code> if (\$cek == null OR \$cek != 'Jawa Timur') {</code>
88	<code> \$uMakan += \$makan;</code>
89	<code> \$masuk++;</code>
90	<code> }</code>
91	<code> }</code>
92	<code>}</code>
93	<code>}</code>
94	<code>\$pph = \$uMakan * \$this->getPph(\$key['golongan']) / 100;</code>
95	<code>\$jumlahBersih = \$uMakan - \$pph;</code>
96	<code>if (\$key['golongan'] == 'II') {</code>
97	<code> \$jmlGol2 += \$uMakan;</code>
98	<code> \$pphGol2 += \$pph;</code>
99	<code> \$jmlBersihGol2 += \$jumlahBersih;</code>
100	<code>}elseif (\$key['golongan'] == 'III' \$key['golongan'] ==</code>
101	<code>'IV') {</code>
	<code> \$jmlGol34 += \$uMakan;</code>
102	<code> \$pphGol34 += \$pph;</code>
103	<code> \$jmlBersihGol34 += \$jumlahBersih;</code>
104	<code> }</code>
105	<code>\$totJumlah += \$jumlahBersih;</code>
106	<code>\$totPph += \$pph;</code>
107	<code>\$totJumlahBersih += \$jumlahBersih;</code>
108	<code>if (\$key['golongan'] == 'II' AND \$cek2 == 0) {</code>
109	<code> \$this->fpdf->SetFont('Arial','BI',10);</code>
110	<code> \$y = \$this->fpdf->GetY()+9;</code>
111	<code> \$this->fpdf->SetXY(18,\$y);</code>
112	<code> \$this->fpdf->MultiCell(55,\$s,"JUMLAH GOL. III & IV",0,'L');</code>
113	<code> \$this->fpdf->SetXY(118,\$y);</code>
114	<code> \$this->fpdf->MultiCell(22,\$s,\$jmlGol34,0,'R');</code>
115	<code> \$this->fpdf->SetXY(140,\$y);</code>
116	<code> \$this->fpdf->MultiCell(18,\$s,\$pphGol34,0,'R');</code>
117	<code> \$this->fpdf->SetXY(158,\$y);</code>
118	<code> \$this->fpdf->MultiCell(21,\$s,\$jmlBersihGol34,0,'R');</code>
119	<code> \$this->fpdf->SetFont('Arial','',10);</code>
120	<code> \$cek2 = 1;</code>
121	<code> \$y = \$this->fpdf->GetY()+5;</code>
122	<code> }</code>
123	<code>if (\$key['singkat'] != "") {</code>
124	<code> \$nama = \$key['singkat'];</code>

Tabel 6.7 Implementasi Algoritme Membuat Laporan Rekapitulasi (Lanjutan)

Baris	Kode Program
125	}else{
126	\$nama = \$key['nama'];
127	}
128	\$this->fpdf->SetY(\$y);
129	\$this->fpdf->MultiCell(8,\$s,\$no,0,'C');
130	\$this->fpdf->SetXY(18,\$y);
131	\$this->fpdf->MultiCell(55,\$s,\$nama,0,'L');
132	\$this->fpdf->SetXY(18,\$y+4);
133	\$this->fpdf->MultiCell(55,\$s,'NIP. '.\$key['nip'],0,'L');
134	\$this->fpdf->SetXY(73,\$y);
135	\$this->fpdf->MultiCell(15,\$s, \$key['golongan'].'/'.\$key['ruang'],0,'C');
136	\$this->fpdf->SetXY(88,\$y);
137	\$this->fpdf->MultiCell(12,\$s,\$masuk,0,'C');
138	\$this->fpdf->SetXY(100,\$y);
139	\$this->fpdf->MultiCell(18,\$s,\$makan,0,'R');
140	\$this->fpdf->SetXY(118,\$y);
141	\$this->fpdf->MultiCell(22,\$s,\$uMakan,0,'R');
142	\$this->fpdf->SetXY(140,\$y);
143	\$this->fpdf->MultiCell(18,\$s,\$pph,0,'R');
144	\$this->fpdf->SetXY(158,\$y);
145	\$this->fpdf->SetFont('Arial','B',10);
146	\$this->fpdf->MultiCell(21,\$s,\$jumlahBersih,0,'R');
147	\$this->fpdf->SetFont('Arial','',10);
148	if (\$no % 2) {
149	\$this->fpdf->SetXY(179,\$y);
150	\$this->fpdf->MultiCell(25,\$s,\$no.'.....',0,'L');
151	}else{
152	\$this->fpdf->SetXY(189,\$y);
153	\$this->fpdf->MultiCell(25,\$s,\$no.'.....',0,'L');
154	}
155	\$no++;
156	\$y = \$this->fpdf->GetY()+7.25;
157	}
158	\$this->fpdf->SetFont('Arial','BI',10);
159	\$y = \$this->fpdf->GetY()+9;
160	\$this->fpdf->SetXY(18,\$y);
161	\$this->fpdf->MultiCell(55,\$s,"JUMLAH GOL. II",0,'L');
162	\$this->fpdf->SetXY(118,\$y);
163	\$this->fpdf->MultiCell(22,\$s,\$jmlGol2,0,'R');
164	\$this->fpdf->SetXY(140,\$y);
165	\$this->fpdf->MultiCell(18,\$s,\$pphGol2,0,'R');
166	\$this->fpdf->SetXY(158,\$y);
167	\$this->fpdf->MultiCell(21,\$s,\$jmlBersihGol2,0,'R');
168	\$y = \$this->fpdf->GetY()+5;
169	\$this->fpdf->Line(10,\$y,204,\$y);

Tabel 6.7 Implementasi Algoritme Membuat Laporan Rekapitulasi (Lanjutan)

Baris	Kode Program
170	<code>\$this->fpdf->SetXY(18, \$y);</code>
171	<code>\$this->fpdf->MultiCell(55, \$s+1, "JUMLAH GOL. II", 0, 'L');</code>
172	<code>\$this->fpdf->SetXY(118, \$y);</code>
173	<code>\$this->fpdf->MultiCell(22, \$s+1, \$jmlGol2, 0, 'R');</code>
174	<code>\$this->fpdf->SetXY(140, \$y);</code>
175	<code>\$this->fpdf->MultiCell(18, \$s+1, \$pphGol2, 0, 'R');</code>
176	<code>\$this->fpdf->SetXY(158, \$y);</code>
177	<code>\$this->fpdf->MultiCell(21, \$s+1, \$jmlBersihGol2, 0, 'R');</code>
178	<code>\$y = \$this->fpdf->GetY();</code>
179	<code>\$this->fpdf->Line(10, \$y, 204, \$y);</code>
180	<code>\$this->fpdf->SetXY(18, \$y);</code>
181	<code>\$this->fpdf->MultiCell(55, \$s+1, "JUMLAH GOL. III dan IV", 0, 'L');</code>
182	<code>\$this->fpdf->SetXY(118, \$y);</code>
183	<code>\$this->fpdf->MultiCell(22, \$s+1, \$jmlGol34, 0, 'R');</code>
184	<code>\$this->fpdf->SetXY(140, \$y);</code>
185	<code>\$this->fpdf->MultiCell(18, \$s+1, \$pphGol34, 0, 'R');</code>
186	<code>\$this->fpdf->SetXY(158, \$y);</code>
187	<code>\$this->fpdf->MultiCell(21, \$s+1, \$jmlBersihGol34, 0, 'R');</code>
188	<code>\$y = \$this->fpdf->GetY();</code>
189	<code>\$this->fpdf->Line(10, \$y, 204, \$y);</code>
190	<code>\$this->fpdf->SetXY(18, \$y);</code>
191	<code>\$this->fpdf->MultiCell(55, \$s+1, "JUMLAH SELURUHNYA", 0, 'L');</code>
192	<code>\$this->fpdf->SetXY(118, \$y);</code>
193	<code>\$this->fpdf->MultiCell(22, \$s+1, \$totJumlah, 0, 'R');</code>
194	<code>\$this->fpdf->SetXY(140, \$y);</code>
195	<code>\$this->fpdf->MultiCell(18, \$s+1, \$totPph, 0, 'R');</code>
196	<code>\$this->fpdf->SetXY(158, \$y);</code>
197	<code>\$this->fpdf->MultiCell(21, \$s+1, \$totJumlahBersih, 0, 'R');</code>
198	<code>\$y = \$this->fpdf->GetY();</code>
199	<code>\$this->fpdf->Line(10, \$y, 204, \$y);</code>
200	<code>\$this->fpdf->Line(9.5, \$y+0.5, 204.5, \$y+0.5);</code>
201	<code>\$this->fpdf->Line(9.5, \$yStart-0.5, 9.5, \$y+0.5);</code>
202	<code>\$this->fpdf->Line(10, \$yStart, 10, \$y);</code>
203	<code>\$this->fpdf->Line(18, \$yStart, 18, \$y);</code>
204	<code>\$this->fpdf->Line(73, \$yStart, 73, \$y);</code>
205	<code>\$this->fpdf->Line(88, \$yStart, 88, \$y);</code>
206	<code>\$this->fpdf->Line(100, \$yStart, 100, \$y);</code>
207	<code>\$this->fpdf->Line(118, \$yStart, 118, \$y);</code>
208	<code>\$this->fpdf->Line(140, \$yStart, 140, \$y);</code>
209	<code>\$this->fpdf->Line(158, \$yStart, 158, \$y);</code>
210	<code>\$this->fpdf->Line(179, \$yStart, 179, \$y);</code>
211	<code>\$this->fpdf->Line(204, \$yStart, 204, \$y);</code>
212	<code>\$this->fpdf->Line(204.5, \$yStart-0.5, 204.5, \$y+0.5);</code>



Tabel 6.7 Implementasi Algoritme Membuat Laporan Rekapitulasi (Lanjutan)

Baris	Kode Program
213	<code>\$kaBag = \$this->M_pegawai->getKaBag(\$bagian);</code>
214	<code>\$keuangan = \$this->M_pegawai->getKeuangan(\$bagian);</code>
215	<code>\$this->fpdf->Ln(2);</code>
216	<code>\$this->fpdf->SetFont('Arial','',10);</code>
217	<code>\$this->fpdf->SetX(105);</code>
218	<code>\$this->fpdf->MultiCell(95,\$s,'Surabaya, '.\$this->tanggal(date('Y-m-d')),0,'C');</code>
219	<code>\$this->fpdf->Ln(4);</code>
220	<code>\$y = \$this->fpdf->GetY();</code>
221	<code>\$this->fpdf->MultiCell(95,\$s,'Kuasa Pengguna Anggaran',0,'C');</code>
222	<code>\$this->fpdf->SetXY(105,\$y);</code>
223	<code>\$this->fpdf->MultiCell(95,\$s,'Bendaharawan Pengeluaran Pembantu',0,'C');</code>
224	<code>\$y = \$this->fpdf->GetY();</code>
225	<code>\$this->fpdf->MultiCell(95,\$s,\$bagian,0,'C');</code>
226	<code>\$this->fpdf->SetXY(105,\$y);</code>
227	<code>\$this->fpdf->MultiCell(95,\$s,\$bagian,0,'C');</code>
228	<code>\$this->fpdf->Ln(10);</code>
229	<code>\$y = \$this->fpdf->GetY();</code>
230	<code>\$this->fpdf->SetFont('Arial','UB',10);</code>
231	<code>\$this->fpdf->MultiCell(95,\$s,\$kaBag['nama'],0,'C');</code>
232	<code>\$this->fpdf->SetXY(105,\$y);</code>
233	<code>\$this->fpdf->MultiCell(95,\$s,\$keuangan['nama'],0,'C');</code>
234	<code>\$y = \$this->fpdf->GetY();</code>
235	<code>\$this->fpdf->SetFont('Arial','',10);</code>
236	<code>\$this->fpdf->MultiCell(95,\$s,'NIP. '.\$kaBag['nip'],0,'C');</code>
237	<code>\$this->fpdf->SetXY(105,\$y);</code>
238	<code>\$this->fpdf->MultiCell(95,\$s,'NIP. '.\$keuangan['nip'],0,'C');</code>
239	<code>\$cek = \$this->M_uangMakan->getUangMakan(\$tahun, \$strBulan, \$bagian)->num_rows();</code>
240	<code>\$cetak = date('Y-m-d');</code>
241	<code>if (\$cek > 0) {</code>
242	<code> \$this->M_uangMakan->editUangMakan(\$tahun, \$strBulan, \$bagian, \$jmlGol2, \$jmlGol34, \$pphGol2, \$pphGol34, \$cetak);</code>
243	<code>}else{</code>
244	<code> \$this->M_uangMakan->insertUangMakan(\$tahun, \$strBulan, \$bagian, \$jmlGol2, \$jmlGol34, \$pphGol2, \$pphGol34, \$cetak);</code>
245	<code>}</code>
246	<code>\$this->fpdf->Output();</code>
247	<code>}</code>



6.2.5 Implementasi Algoritme Memasukkan Data Sidang

Algoritme memasukkan data sidang dalam bentuk *pseudocode* pada Tabel 5.5 diimplementasikan kedalam kode program dalam bahasa pemrograman PHP. Hasil implementasi dapat dilihat pada Tabel 6.8.

Tabel 6. 8 Implementasi Algoritme Memasukkan Data Sidang

Baris	Kode Program
1	<code>public function saveSidang(){</code>
2	<code> \$noUrut = (int) substr(\$this->M_sidang->getID(), 1);</code>
3	<code> \$char = "S";</code>
4	<code> \$idSidang = \$char . sprintf("%05s", \$noUrut);</code>
5	<code> \$data = array(</code>
6	<code> 'idSidang' => \$idSidang,</code>
7	<code> 'hari' => \$this->input->post('hari'),</code>
8	<code> 'tanggal' => \$this->input->post('tanggal'),</code>
9	<code> 'sTime' => \$this->input->post('sTime'),</code>
10	<code> 'eTime' => \$this->input->post('eTime'),</code>
11	<code> 'tempat' => \$this->input->post('tempat'),</code>
12	<code> 'acara' => \$this->input->post('acara')</code>
13	<code>);</code>
14	<code> \$this->M_sidang->saveSidang(\$data);</code>
15	<code> \$nipL = \$this->input->post('nipL');</code>
16	<code> \$rows = count(\$nipL);</code>
17	<code> \$namaL = \$this->input->post('namaL');</code>
18	<code> \$golonganL = \$this->input->post('golonganL');</code>
19	<code> \$jabatanL = \$this->input->post('jabatanL');</code>
20	<code> \$jumlahL = \$this->input->post('jumlahL');</code>
21	<code> \$ket = "Data Berhasil Disimpan";</code>
22	<code> \$ket2 = "";</code>
23	<code> for (\$i=0; \$i < \$rows; \$i++) {</code>
24	<code> \$ccek = \$this->M_pegawaiSidang->getPegawaiSidangByNip(</code>
25	<code> \$data['idSidang'], \$nipL[\$i])->num_rows();</code>
26	<code> if (\$ccek == 0 AND \$nipL[\$i] != "") {</code>
27	<code> \$this->M_pegawaiSidang-></code>
	<code> savePegawaiSidang(\$data['idSidang'], \$nipL[\$i],</code>
	<code> \$namaL[\$i], \$golonganL[\$i], \$jabatanL[\$i],</code>
	<code> \$jumlahL[\$i]);</code>
28	<code> }elseif(\$ccek != 0){</code>
29	<code> \$ket2 = \$namaL[\$i].' ('.\$nipL[\$i].') ; ';</code>
30	<code> }</code>
31	<code> }</code>
32	<code> \$nip = \$this->input->post('nip');</code>
33	<code> \$jumlah = \$this->input->post('jumlah');</code>
34	<code> \$rows = count(\$nip);</code>
35	<code> for (\$i=0; \$i < \$rows; \$i++) {</code>
36	<code> \$pegawai = \$this->M_pegawai->getPegawaiNip(\$nip[\$i])-></code>
	<code> result_array();</code>



Tabel 6.8 Implementasi Algoritme Memasukkan Data Sidang (Lanjutan)

Baris	Kode Program
37	<code>\$cek = \$this->M_pegawaiSidang->getPegawaiSidangByNip</code>
	<code>(\$data['idSidang'], \$nip[\$i])->num_rows();</code>
38	<code>if (\$cek == 0 AND \$nip[\$i] != "") {</code>
39	<code> \$gol = \$pegawai[0]['golongan'].'/'.\$pegawai[0]['ruang'];</code>
40	<code>}elseif(\$cek != 0){</code>
41	<code> \$ket2 = \$namaL[\$i].' ('.\$nipL[\$i].') ; '</code>
42	<code> }</code>
43	<code>}</code>
44	<code>\$nip = \$this->input->post('nip');</code>
45	<code>\$jumlah = \$this->input->post('jumlah');</code>
46	<code>\$rows = count(\$nip);</code>
47	<code>for (\$i=0; \$i < \$rows; \$i++) {</code>
48	<code> \$pegawai = \$this->M_pegawai->getPegawaiNip(\$nip[\$i])-</code>
	<code>>result_array();</code>
49	<code> \$cek = \$this->M_pegawaiSidang->getPegawaiSidangByNip</code>
	<code>(\$data['idSidang'], \$nip[\$i])->num_rows();</code>
50	<code> if (\$cek == 0 AND \$nip[\$i] != "") {</code>
51	<code> \$gol = \$pegawai[0]['golongan'].'/'.\$pegawai[0]['ruang'];</code>
52	<code> \$this->M_pegawaiSidang->savePegawaiSidang(</code>
	<code> \$data['idSidang'], \$nip[\$i], \$pegawai[0]['nama'], \$gol,</code>
	<code> \$pegawai[0]['jabatan'], \$jumlah[\$i]);</code>
53	<code> }elseif(\$cek != 0){</code>
54	<code> \$ket2 = \$pegawai[0]['nama'].' ('.\$nip[\$i].') ; '</code>
55	<code> }</code>
56	<code>}</code>
57	<code>if (\$ket2 != "") {</code>
58	<code> \$this->session->set_userdata('ket', \$ket2);</code>
59	<code>}else{</code>
60	<code> \$this->session->set_userdata('ket', \$ket);</code>
61	<code>}</code>
62	<code>\$this->sidang();</code>
63	<code>}</code>

6.2.6 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang

Algoritme mencetak daftar penerima dan kwitansi uang sidang dalam bentuk *pseudocode* pada Tabel 5.6 diimplementasikan kedalam kode program dalam bahasa pemrograman PHP. Hasil implementasi dapat dilihat pada Tabel 6.9.

Tabel 6. 9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang

Baris	Kode Program
1	function cetakSidang() {
2	\$param = array(
3	'orientation' => 'L',
4	'unit' => 'mm',
5	'size' => array(210, 330)
6);
7	\$this->load->library('fpdf_gen', \$param);
8	\$idSidang = \$this->input->post('id_detail');
9	\$this->fpdf->SetAutoPageBreak(true, 60);
10	\$this->fpdf->SetFont('Arial', 'B', 11);
11	\$data = \$this->M_sidang->getSidangById(\$idSidang);
12	foreach (\$data->result_array() as \$key) {
13	\$sidang = array(
14	'idSidang' => \$key['idSidang'],
15	'hari' => \$key['hari'],
16	'tanggal' => \$key['tanggal'],
17	'tempat' => \$key['tempat'],
18	'sTime' => \$key['sTime'],
19	'eTime' => \$key['eTime'],
20	'tempat' => \$key['tempat'],
21	'acara' => \$key['acara']
22);
23	}
24	\$s = 5;
25	\$this->fpdf->SetX(50);
26	\$this->fpdf->MultiCell(230, 6, 'DAFTAR PENERIMAAN UANG SIDANG DALAM RANGKA '.strtoupper(\$sidang['acara']), 0, 'C');
27	\$this->fpdf->Ln(5);
28	\$y = \$this->fpdf->GetY();
29	\$this->fpdf->Line(10, \$y-0.5, 320, \$y-0.5);
30	\$this->fpdf->SetFont('Arial', 'B', 8);
31	\$this->fpdf->MultiCell(8, \$s*2, 'NO', 1, 'C');
32	\$this->fpdf->SetXY(18, \$y);
33	\$this->fpdf->MultiCell(67, \$s*2, 'N A M A / N I P', 1, 'C');
34	\$this->fpdf->SetXY(85, \$y);
35	\$this->fpdf->MultiCell(12, \$s*2, 'GOL', 1, 'C');
36	\$this->fpdf->SetXY(97, \$y);
37	\$this->fpdf->MultiCell(48, \$s*2, 'JABATAN', 1, 'C');
38	\$this->fpdf->SetXY(145, \$y);
39	\$this->fpdf->MultiCell(21, \$s, 'BANYAKNYA SIDANG', 1, 'C');
40	\$this->fpdf->SetXY(166, \$y);
41	\$this->fpdf->MultiCell(26, \$s, 'BESAR UANG SIDANG', 1, 'C');
42	\$this->fpdf->SetXY(192, \$y);
43	\$this->fpdf->MultiCell(28, \$s*2, 'JUMLAH', 1, 'C');
44	\$this->fpdf->SetXY(220, \$y);

Tabel 6.9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang (Lanjutan)

Baris	Kode Program
45	<code>\$this->fpdf->MultiCell(28,\$s*2,'PPh. 21',1,'C');</code>
46	<code>\$this->fpdf->SetXY(248,\$y);</code>
47	<code>\$this->fpdf->MultiCell(35,\$s,'JUMLAH PENERIMAAN BERSIH',1,'C');</code>
48	<code>\$this->fpdf->SetXY(283,\$y);</code>
49	<code>\$this->fpdf->MultiCell(35,\$s*2,'TANDATANGAN',1,'C');</code>
50	<code>\$s = 3.5;</code>
51	<code>\$y = \$this->fpdf->GetY();</code>
52	<code>\$this->fpdf->SetFont('Arial', '', 8);</code>
53	<code>\$this->fpdf->MultiCell(8,\$s,'1',1,'C');</code>
54	<code>\$this->fpdf->SetXY(18,\$y);</code>
55	<code>\$this->fpdf->MultiCell(67,\$s,'2',1,'C');</code>
56	<code>\$this->fpdf->SetXY(85,\$y);</code>
57	<code>\$this->fpdf->MultiCell(12,\$s,'3',1,'C');</code>
58	<code>\$this->fpdf->SetXY(97,\$y);</code>
59	<code>\$this->fpdf->MultiCell(48,\$s,'4',1,'C');</code>
60	<code>\$this->fpdf->SetXY(145,\$y);</code>
61	<code>\$this->fpdf->MultiCell(21,\$s,'5',1,'C');</code>
62	<code>\$this->fpdf->SetXY(166,\$y);</code>
63	<code>\$this->fpdf->MultiCell(26,\$s,'6',1,'C');</code>
64	<code>\$this->fpdf->SetXY(192,\$y);</code>
65	<code>\$this->fpdf->MultiCell(28,\$s,'7',1,'C');</code>
66	<code>\$this->fpdf->SetXY(220,\$y);</code>
67	<code>\$this->fpdf->MultiCell(28,\$s,'8',1,'C');</code>
68	<code>\$this->fpdf->SetXY(248,\$y);</code>
69	<code>\$this->fpdf->MultiCell(35,\$s,'9',1,'C');</code>
70	<code>\$this->fpdf->SetXY(283,\$y);</code>
71	<code>\$this->fpdf->MultiCell(35,\$s,'10',1,'C');</code>
72	<code>\$no = 1;</code>
73	<code>\$totJumlah = 0;</code>
74	<code>\$totPph = 0;</code>
75	<code>\$totJumlahBersih = 0;</code>
76	<code>\$jmlGol2 = 0;</code>
77	<code>\$pphGol2 = 0;</code>
78	<code>\$jmlBersihGol2 = 0;</code>
79	<code>\$jmlGol34 = 0;</code>
80	<code>\$pphGol34 = 0;</code>
81	<code>\$jmlBersihGol34 = 0;</code>
82	<code>\$cek2 = 0;</code>
83	<code>\$pegawaiSidang = \$this->M_pegawaiSidang->getPegawaiSidang(\$idSidang);</code>
84	<code>\$y = \$this->fpdf->GetY();</code>
85	<code>\$s = 5;</code>
86	<code>\$this->fpdf->SetY(\$y);</code>
87	<code>foreach (\$pegawaiSidang->result_array() as \$key) {</code>
88	<code>\$gol = explode('/', \$key['golongan']);</code>
89	



Tabel 6.9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang (Lanjutan)

Baris	Kode Program
90	<code>\$esGol = \$this->getEselonGolongan(\$key['jabatan'], \$gol[0]);</code>
91	<code>\$uangSidang = \$this->M_tapSidang->getUangSidang(\$esGol['eselon'], \$esGol['golongan']);</code>
92	<code>\$jumlahUangSidang = (int)\$uangSidang * (int)\$key['jumlah'];</code>
93	<code>\$pph = \$jumlahUangSidang*\$this->getPph(\$gol[0])/100;</code>
94	<code>\$jumlahBersih = \$jumlahUangSidang - \$pph;</code>
95	<code>\$totJumlah += \$jumlahBersih;</code>
96	<code>\$totPph += \$pph;</code>
97	<code>\$totJumlahBersih += \$jumlahBersih - \$pph;</code>
98	<code>\$this->fpdf->SetY(\$y);</code>
99	<code>\$this->fpdf->SetFont('Arial', '', 9);</code>
100	<code>\$this->fpdf->MultiCell(8, \$s*2, \$no, 1, 'C');</code>
101	<code>\$this->fpdf->SetXY(18, \$y);</code>
102	<code>\$this->fpdf->MultiCell(67, \$s, \$key['nama'], 'LTR', 'L');</code>
103	<code>\$this->fpdf->SetXY(18, \$y+5);</code>
104	<code>\$this->fpdf->MultiCell(67, \$s, 'NIP. '.\$key['nip'], 'LBR', 'L');</code>
105	<code>\$this->fpdf->SetXY(85, \$y);</code>
106	<code>\$this->fpdf->MultiCell(12, \$s*2, \$key['golongan'], 1, 'C');</code>
107	<code>\$this->fpdf->SetXY(97, \$y);</code>
108	<code>\$this->fpdf->MultiCell(48, \$s*2, \$key['jabatan'], 1, 'L');</code>
109	<code>\$this->fpdf->SetXY(145, \$y);</code>
110	<code>\$this->fpdf->MultiCell(21, \$s*2, \$key['jumlah'].' kali', 1, 'C');</code>
111	<code>\$this->fpdf->SetXY(166, \$y);</code>
112	<code>\$this->fpdf->MultiCell(26, \$s*2, \$uangSidang, 1, 'R');</code>
113	<code>\$this->fpdf->SetXY(192, \$y);</code>
114	<code>\$this->fpdf->MultiCell(28, \$s*2, \$jumlahUangSidang, 1, 'R');</code>
115	<code>\$this->fpdf->SetXY(220, \$y);</code>
116	<code>\$this->fpdf->MultiCell(28, \$s*2, \$pph, 1, 'R');</code>
117	<code>\$this->fpdf->SetXY(248, \$y);</code>
118	<code>\$this->fpdf->MultiCell(35, \$s*2, \$jumlahBersih, 1, 'R');</code>
119	<code>if (\$no % 2) {</code>
120	<code> \$this->fpdf->SetXY(283, \$y);</code>
121	<code> \$this->fpdf->MultiCell(35, \$s*2, \$no.'.....', 1, 'L');</code>
122	<code>}else{</code>
123	<code> \$this->fpdf->SetXY(300, \$y);</code>
124	<code> \$this->fpdf->MultiCell(25, \$s*2, \$no.'.....', 0, 'L');</code>
125	<code> \$this->fpdf->SetXY(283, \$y);</code>
126	<code> \$this->fpdf->MultiCell(35, \$s*2, '', 1, 'L');</code>
127	<code>}</code>



Tabel 6.9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang (Lanjutan)

Baris	Kode Program
128	<code>\$no++;</code>
129	<code>\$y = \$this->fpdf->GetY();</code>
130	<code>}</code>
131	<code>\$this->fpdf->SetFont('Arial','BI',9);</code>
132	<code>\$y = \$this->fpdf->GetY();</code>
133	<code>\$this->fpdf->SetXY(10,\$y);</code>
134	<code>\$this->fpdf->MultiCell(182,\$s,"JUMLAH SELURUHNYA",1,'C');</code>
135	<code>\$this->fpdf->SetXY(192,\$y);</code>
136	<code>\$this->fpdf->MultiCell(28,\$s,\$totJumlah,1,'R');</code>
137	<code>\$this->fpdf->SetXY(220,\$y);</code>
138	<code>\$this->fpdf->MultiCell(28,\$s,\$totPph,1,'R');</code>
139	<code>\$this->fpdf->SetXY(248,\$y);</code>
140	<code>\$this->fpdf->MultiCell(35,\$s,\$totJumlahBersih,1,'R');</code>
141	<code>\$this->fpdf->SetXY(283,\$y);</code>
142	<code>\$this->fpdf->MultiCell(35,\$s','',1,'R');</code>
143	<code>\$kaBag = \$this->M_pegawai->getKaBag('Bagian Tata Laksana');</code>
144	<code>\$keuangan = \$this->M_pegawai->getKeuangan('Bagian Tata</code>
145	<code>Laksana');</code>
146	<code>\$s = 4;</code>
147	<code>\$this->fpdf->Ln(5);</code>
148	<code>\$this->fpdf->SetFont('Arial','',9);</code>
149	<code>\$this->fpdf->SetX(165);</code>
150	<code>\$this->fpdf->MultiCell(155,\$s,'Surabaya, '.\$this-></code>
151	<code>tanggal(date('Y-m-d')),0,'C');</code>
152	<code>\$y = \$this->fpdf->GetY();</code>
153	<code>\$this->fpdf->MultiCell(155,\$s,'Kuasa Pengguna</code>
154	<code>Anggaran',0,'C');</code>
155	<code>\$this->fpdf->SetXY(165,\$y);</code>
156	<code>\$this->fpdf->MultiCell(155,\$s,'Bendaharawan Pengeluaran</code>
157	<code>Pembantu',0,'C');</code>
158	<code>\$y = \$this->fpdf->GetY();</code>
159	<code>\$this->fpdf->MultiCell(155,\$s,'Bagian Tatalaksana',0,'C');</code>
160	<code>\$this->fpdf->SetXY(165,\$y);</code>
161	<code>\$this->fpdf->MultiCell(155,\$s,'Bagian Tatalaksana',0,'C');</code>
162	<code>\$this->fpdf->SetXY(165,\$y);</code>
163	<code>\$this->fpdf->MultiCell(155,\$s,'Bagian Tatalaksana',0,'C');</code>
164	<code>\$this->fpdf->Ln(13);</code>
165	<code>\$y = \$this->fpdf->GetY();</code>
166	<code>\$this->fpdf->SetFont('Arial','UB',10);</code>
167	<code>\$this->fpdf->MultiCell(155,\$s,\$kaBag['nama'],0,'C');</code>
168	<code>\$this->fpdf->SetXY(165,\$y);</code>
169	<code>\$this->fpdf->MultiCell(155,\$s,'EDI HARTONO, SE',0,'C');</code>
170	<code>\$y = \$this->fpdf->GetY();</code>
171	<code>\$this->fpdf->SetFont('Arial','',10);</code>
172	<code>\$this->fpdf->MultiCell(155,\$s,'NIP. '.\$kaBag['nip'],0,'C');</code>
173	<code>\$this->fpdf->SetXY(165,\$y);</code>



Tabel 6.9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang (Lanjutan)

Baris	Kode Program
169	<code>\$this->fpdf->MultiCell(155,\$s,'NIP. 19730225 008012 1</code>
170	<code>011',0,'C');</code>
171	<code>\$this->M_sidang->saveUangSidang(\$idSidang, \$totJumlah,</code>
172	<code>\$totPph, \$totJumlahBersih);</code>
173	<code>\$size = array(210, 160);</code>
174	<code>\$this->fpdf->AddPage('L', \$size);</code>
175	<code>\$this->fpdf->SetAutoPageBreak(true,1);</code>
176	<code>\$this->fpdf->SetFont('Arial','B',11);</code>
177	<code>\$data = \$this->M_sidang->getSidangById(\$idSidang);</code>
178	<code>foreach (\$data->result_array() as \$key) {</code>
179	<code> \$sidang = array(</code>
180	<code> 'acara' => \$key['acara'],</code>
181	<code> 'jumlah' => \$key['jumlah'],</code>
182	<code> 'pph' => \$key['pph'],</code>
183	<code> 'jmlBersih' => \$key['jmlBersih']</code>
184	<code>);</code>
185	<code> }</code>
186	<code> \$s = 4;</code>
187	<code> \$this->fpdf->SetFont('Arial','',10);</code>
188	<code> \$this->fpdf->SetXY(22,17);</code>
189	<code> \$this->fpdf->MultiCell(40,\$s,\$this->input->post('kode'));</code>
190	<code> \$this->fpdf->Rect(20,15,59,7);</code>
191	<code> \$this->fpdf->SetFont('Times','',12);</code>
192	<code> \$this->fpdf->SetXY(127,16);</code>
193	<code> \$this->fpdf->MultiCell(35,\$s,'NO. BKU / HAL');</code>
194	<code> \$this->fpdf->SetXY(127,24);</code>
195	<code> \$this->fpdf->MultiCell(35,5,'NO. PROGRAM / KEGIATAN');</code>
196	<code> \$this->fpdf->Line(125,13,204,13);</code>
197	<code> \$this->fpdf->Line(125,23,204,23);</code>
198	<code> \$this->fpdf->Line(125,34,204,34);</code>
199	<code> \$this->fpdf->Line(125,13,125,34);</code>
200	<code> \$this->fpdf->Line(165,13,165,34);</code>
201	<code> \$this->fpdf->Line(185,13,185,34);</code>
202	<code> \$this->fpdf->Line(204,13,204,34);</code>
203	<code> \$this->fpdf->SetFont('Times','BU',17);</code>
204	<code> \$this->fpdf->SetXY(10,39);</code>
205	<code> \$this->fpdf->MultiCell(190,\$s,'K W I T A N S I',0,'C');</code>
206	<code> \$h = 9;</code>
207	<code> \$this->fpdf->SetFont('Times','',12);</code>
208	<code> \$this->fpdf->SetXY(10,50);</code>
209	<code> \$y = \$this->fpdf->GetY();</code>
210	<code> \$this->fpdf->MultiCell(40,\$h,'Telah Terima Dari',0,'L');</code>
211	<code> \$this->fpdf->SetXY(50,\$y);</code>
212	<code> \$this->fpdf->MultiCell(5,\$h,':',0,'L');</code>



Tabel 6.9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang (Lanjutan)

Baris	Kode Program
212	<code>\$this->fpdf->SetXY(55,\$y);</code>
213	<code>\$this->fpdf->SetFont('Arial','B',11);</code>
214	<code>\$this->fpdf->MultiCell(140,\$h,'KUASA PENGGUNA ANGGARAN BAGIAN TATALAKSANA',0,'l');</code>
215	<code>\$y = \$this->fpdf->GetY();</code>
216	<code>\$this->fpdf->SetFont('Times','',12);</code>
217	<code>\$this->fpdf->MultiCell(40,\$h,'Jumlah Uang',0,'l');</code>
218	<code>\$this->fpdf->SetXY(50,\$y);</code>
219	<code>\$this->fpdf->MultiCell(5,\$h,':',0,'L');</code>
220	<code>\$this->fpdf->SetXY(55,\$y);</code>
221	<code>\$this->fpdf->SetFont('Arial','B',10);</code>
222	<code>\$this->fpdf->MultiCell(155,\$h,' ==='.strtoupper(\$this->terbilang(\$sidang['jumlah'])).' RUPIAH ===',0,'l');</code>
223	<code>\$this->fpdf->Line(58,\$y+1,204,\$y+1);</code>
224	<code>\$this->fpdf->Line(53,\$y+8,199,\$y+8);</code>
225	<code>\$this->fpdf->Line(58,\$y+1,53,\$y+8);</code>
226	<code>\$this->fpdf->Line(204,\$y+1,199,\$y+8);</code>
227	<code>\$this->fpdf->SetFont('Times','',12);</code>
228	<code>\$y = \$this->fpdf->GetY();</code>
229	<code>\$this->fpdf->MultiCell(40,\$h,'Untuk Pembayaran',0,'L');</code>
230	<code>\$this->fpdf->SetXY(50,\$y);</code>
231	<code>\$this->fpdf->MultiCell(5,\$h,':',0,'L');</code>
232	<code>\$this->fpdf->SetFont('Arial','',10);</code>
233	<code>\$this->fpdf->SetXY(55,\$y+3);</code>
234	<code>\$this->fpdf->MultiCell(140,\$s,'Uang sidang dalam rangka '.\$sidang['acara'].' dengan rincian :',0,'L');</code>
235	<code>\$this->fpdf->SetX(55);</code>
236	<code>\$y = \$this->fpdf->GetY();</code>
237	<code>\$this->fpdf->MultiCell(70,\$s,'Uang sidang : 1 x Rp. '.number_format(\$sidang['jumlah'],2,',','.').' :',0,'L');</code>
238	<code>\$this->fpdf->SetXY(145,\$y);</code>
239	<code>\$this->fpdf->MultiCell(15,\$s,'= Rp. ',0,'L');</code>
240	<code>\$this->fpdf->SetXY(160,\$y);</code>
241	<code>\$this->fpdf->MultiCell(27,\$s,number_format(\$sidang['jumlah'],2,',','.'),0,'R');</code>
242	<code>\$this->fpdf->SetX(55);</code>
243	<code>\$y = \$this->fpdf->GetY();</code>
244	<code>\$this->fpdf->MultiCell(70,\$s,'- PPh. 21',0,'L');</code>
245	<code>\$this->fpdf->SetXY(145,\$y);</code>
246	<code>\$this->fpdf->MultiCell(15,\$s,'= Rp. ',0,'L');</code>
247	<code>\$this->fpdf->SetXY(160,\$y);</code>
248	<code>\$this->fpdf->MultiCell(27,\$s,</code>
249	<code>number_format(\$sidang['pph'],2,',','.'),0,'R');</code>
250	<code>\$this->fpdf->SetX(110);</code>
251	<code>\$y = \$this->fpdf->GetY();</code>
252	<code>\$this->fpdf->Line(146,\$y-0.5,186,\$y-.5);</code>
253	<code>\$this->fpdf->MultiCell(35,\$s,'Jumlah diterima',0,'L');</code>



Tabel 6.9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang (Lanjutan)

Baris	Kode Program
254	<code>\$this->fpdf->SetXY(145,\$y);</code>
255	<code>\$this->fpdf->MultiCell(15,\$s,'= Rp. ',0,'L');</code>
256	<code>\$this->fpdf->SetXY(160,\$y);</code>
257	<code>\$this->fpdf->MultiCell(27,\$s,</code>
258	<code>number_format(\$sidang['jumlah']-</code> <code>\$sidang['pph'],2,',','.'),0,'R');</code>
259	<code>\$y=\$this->fpdf->GetY()+8;</code>
260	<code>\$this->fpdf->Image('assets/image/terbilang.png',10,\$y,39,7);</code>
261	<code>\$this->fpdf->Ln();</code>
262	<code>\$this->fpdf->Line(58,\$y,110,\$y);</code>
263	<code>\$this->fpdf->Line(53,\$y+7,105,\$y+7);</code>
264	<code>\$this->fpdf->Line(58,\$y,53,\$y+7);</code>
265	<code>\$this->fpdf->Line(110,\$y,105,\$y+7);</code>
266	<code>\$this->fpdf->SetXY(60,\$y+1.5);</code>
267	<code>\$this->fpdf->SetFont('Arial','B',11);</code>
268	<code>\$this->fpdf->MultiCell(50,\$s,</code> <code>number_format(\$sidang['jumlah'],2,',','.'),0,'L');</code>
269	<code>\$this->fpdf->SetXY(130,\$y+10);</code>
270	<code>\$this->fpdf->SetFont('Times','',12);</code>
271	<code>\$this->fpdf->MultiCell(60,7,'Surabaya, '.\$this-</code> <code>>tanggal(\$this->input->post('tgl')),0,'R');</code>
272	<code>\$this->fpdf->SetX(130);</code>
273	<code>\$this->fpdf->MultiCell(60,5,'Yang Menerima,',0,'R');</code>
274	<code>\$y = \$this->fpdf->GetY();</code>
275	<code>\$this->fpdf->SetFont('Arial','',9);</code>
276	<code>\$this->fpdf->SetXY(15,\$y);</code>
277	<code>\$this->fpdf->MultiCell(60,\$s,'Setuju dibayar',0,'C');</code>
278	<code>\$this->fpdf->SetXY(80,\$y);</code>
279	<code>\$this->fpdf->MultiCell(60,\$s,'Lunas dibayar tgl',0,'L');</code>
280	<code>\$y = \$this->fpdf->GetY();</code>
281	<code>\$this->fpdf->SetXY(15,\$y);</code>
282	<code>\$this->fpdf->MultiCell(60,\$s,'Kuasa Pengguna</code> <code>Anggaran',0,'C');</code>
283	<code>\$this->fpdf->SetXY(75,\$y);</code>
284	<code>\$this->fpdf->MultiCell(60,\$s,'Bendahara Pengeluaran</code> <code>pembantu',0,'C');</code>
285	<code>\$y = \$this->fpdf->GetY();</code>
286	<code>\$this->fpdf->SetXY(15,\$y);</code>
287	<code>\$this->fpdf->MultiCell(60,\$s,'Bagian Tatalaksana',0,'C');</code>
288	<code>\$this->fpdf->SetXY(75,\$y);</code>
289	<code>\$this->fpdf->MultiCell(60,\$s,'Bagian Tatalaksana',0,'C');</code>
290	<code>\$y = \$this->fpdf->GetY()+13;</code>
291	<code>\$this->fpdf->SetXY(15,\$y);</code>
292	<code>\$this->fpdf->SetFont('Arial','BU',9);</code>
293	<code>\$this->fpdf->MultiCell(60,\$s,\$kaBag['nama'],0,'C');</code>
294	<code>\$this->fpdf->SetXY(75,\$y);</code>

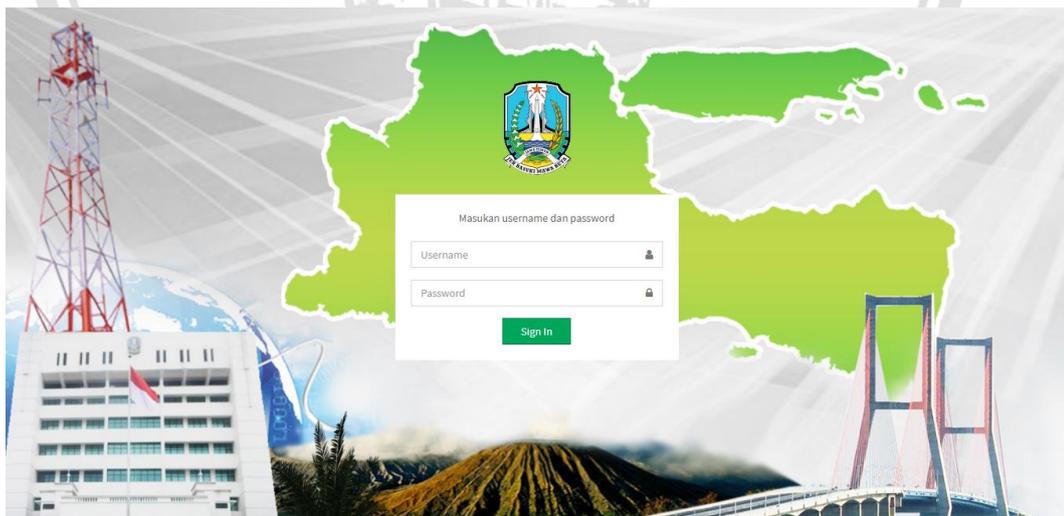
Tabel 6.9 Implementasi Algoritme Mencetak DaftarPenerima dan Kwitansi Uang Sidang (Lanjutan)

Baris	Kode Program
295	<code>\$this->fpdf->MultiCell(60,\$s,\$keuangan['nama'],0,'C');</code>
296	<code>\$this->fpdf->SetXY(135,\$y);</code>
297	<code>\$this->fpdf->MultiCell(60,\$s,\$keuangan['nama'],0,'C');</code>
298	<code>\$y = \$this->fpdf->GetY();</code>
299	<code>\$this->fpdf->SetXY(15,\$y);</code>
300	<code>\$this->fpdf->SetFont('Arial', '',9);</code>
301	<code>\$this->fpdf->MultiCell(60,\$s,'NIP. '.\$kaBag['nip'],0,'C');</code>
302	<code>\$this->fpdf->SetXY(75,\$y);</code>
303	<code>\$this->fpdf->MultiCell(60,\$s,'NIP.'.\$keuangan['nip'],0,'C');</code>
304	<code>\$this->fpdf->SetXY(135,\$y);</code>
305	<code>\$this->fpdf->MultiCell(60,\$s,'NIP.'.\$keuangan['nip'],0,'C');</code>
306	<code>\$this->fpdf->Output();</code>
307	<code>}</code>

6.3 Implementasi Antarmuka Pengguna

Subbab implementasi antarmuka pengguna menggambarkan hasil implementasi rancangan antarmuka pengguna Sistem Informasi Keuangan Pegawai pada bab perancangan. Hanya beberapa antarmuka pengguna yang digambarkan pada subbab ini. Antar muka pengguna tersebut diantaranya antarmuka login, antarmuka daftar perjalanan dinas, antarmuka detail perjalanan dinas, antarmuka daftar sidang, antarmuka presensi, antarmuka absensi, antarmuka lembur, dan antarmuka laporan rekapitulasi.

6.3.1 Antarmuka Login



Gambar 6. 1 Antarmuka Login

Rancangan antarmuka *login* pada Gambar 5.9 diimplementasikan kedalam antarmuka login pada Gambar 6.1. antar muka login merupakan tampilan yang akan dilihat oleh pengguna ketika pertama membuka sistem untuk masuk kedalam sistem. Pada antarmuka ini pengguna akan melakukan proses otentikasi agar dapat mengakses informasi dalam sistem sesuai dengan hak aksesnya. Antarmuka *login* memiliki beberapa komponen yaitu kolom input *username* untuk memasukkan *username*, kolom input *password* untuk memasukkan *password* dan tombol login untuk menjalankan fungsi otentikasi pengguna dan masuk ke dalam sistem.

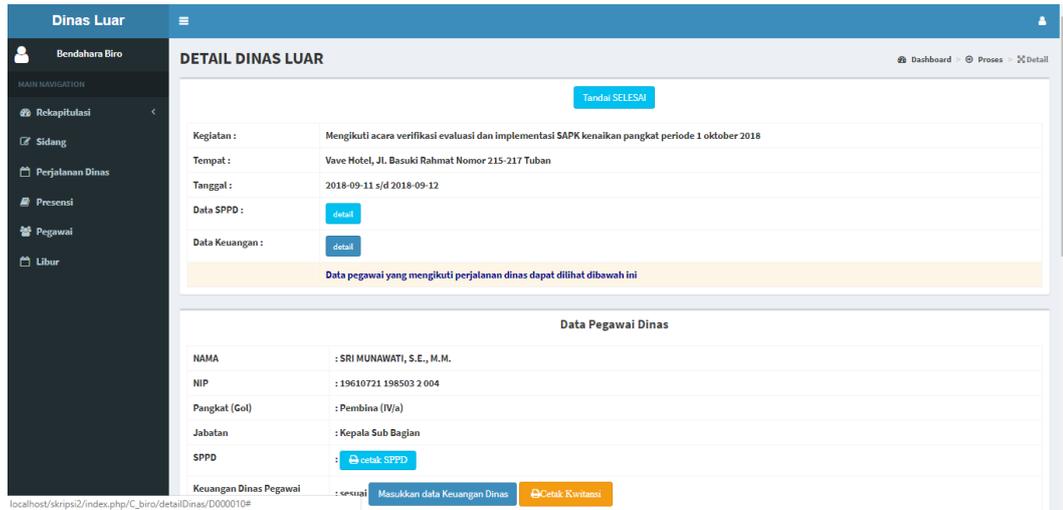
6.3.2 Antarmuka Daftar Perjalanan Dinas

Tanggal	No SPT	Tempat	Tujuan	SPT	Status	Detail
2018-09-11 s/d 2018-09-12	094/031.2/2018	Vave Hotel, Jl. Basuki Rahmat Nomor 215-217 Tuban	Mengikuti acara verifikasi evaluasi dan implementasi SAPK kenaikan pangkat periode 1 oktober 2018		diproses	
2018-09-12 s/d 2018-09-12	094/031.3/2018	Kementrian PAN RB, Jakarta	Menghadiri undangan rapat persiapan penyelenggaraan TOP 99 inovasi pelayanan publik		diproses	
2018-09-05 s/d 2018-09-05	094/031.2/2018	BKD Provinsi Jawa Timur, Jl. Jemur Andayani 1, Surabaya	Melaksanakan koordinasi terkait mutasi kerja PNS dan berkas tidak lengkap (BTL) kenaikan pangkat periode 1 Oktober 2018 di lingkungan sekretariat Daerah Provinsi Jawa Timur		diproses	

Gambar 6. 2 Antarmuka Daftar Perjalanan Dinas

Rancangan antarmuka daftar perjalanan dinas pada Gambar 5.10 diimplementasikan ke antarmuka daftar perjalanan dinas pada Gambar 6.2. Antarmuka daftar perjalanan dinas merupakan tampilan yang akan dilihat oleh pengguna ketika melihat daftar perjalanan dinas yang telah tercatat. Selain itu antarmuka ini juga dibuka oleh pengguna ketika akan memasukkan data perjalanan dinas atau akan melihat detail data perjalanan dinas. Antarmuka daftar perjalanan dinas memiliki beberapa komponen yaitu *icon user* di pojok kanan atas untuk melakukan *logout* dan *side bar* menu di bagian kiri untuk masuk ke menu yang ada pada sistem. Kemudian di bagian konten terdapat kolom “jumlah” untuk memasukkan jumlah pegawai yang akan melakukan perjalanan dinas, dan tombol “tambahkan” untuk menampilkan *form input* data perjalanan dinas dan menyimpannya ke *database* sistem. Kolom “rows” untuk mengatur banyak data yang ditampilkan dalam satu tampilan layar. Kemudian kolom “*search*” untuk memasukkan kata kunci dari data yang akan dicari pada tabel. Pada bagian konten juga terdapat tabel yang berisi daftar perjalanan dinas.

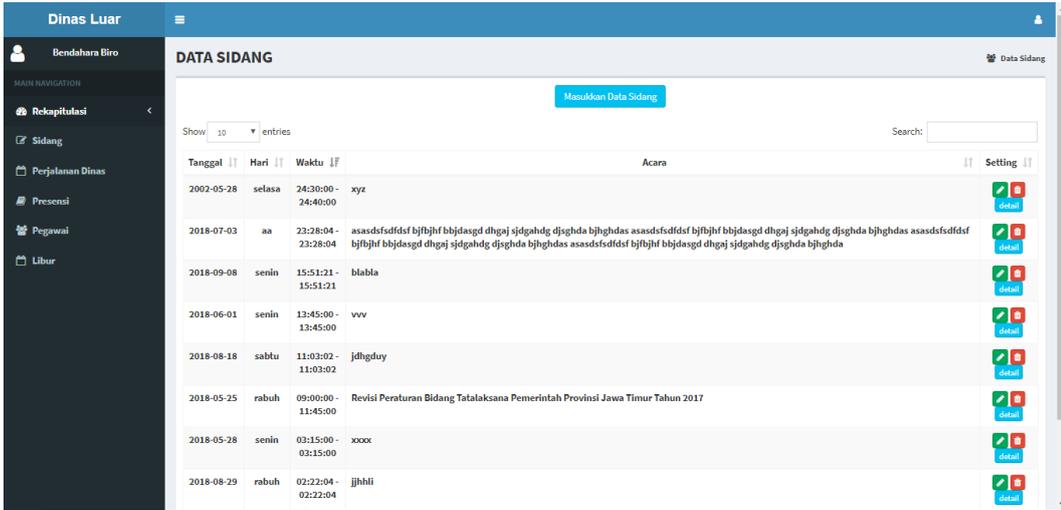
6.3.3 Antarmuka Detail Perjalanan Dinas



Gambar 6. 3 Antarmuka Detail Perjalanan Dinas

Rancangan antarmuka detail perjalanan dinas pada Gambar 5.11 diimplementasikan kedalam antarmuka detail perjalanan dinas pada Gambar 6.3. Antarmuka detail perjalanan dinas merupakan tampilan yang akan dilihat oleh pengguna ketika melihat detail suatu perjalanan dinas. Selain itu antarmuka ini juga dibuka oleh pengguna ketika akan memasukkan data SPPD atau uang dinas, dan mencetak SPPD atau kwitansi uang perjalanan dinas. Antarmuka detail dinas memiliki beberapa komponen yaitu *icon user* di pojok kanan atas untuk melakukan *logout* dan *side bar* menu di bagian kiri. Kemudian di bagian konten terdapat tombol “tandai Selesai” untuk mengubah status perjalanan dinas menjadi selesai, kemudian tombol “detail SPPD” untuk menampilkan data SPPD dan *form input* data SPPD yang akan disimpan. Selain itu terdapat tombol “detail keuangan” untuk menampilkan data umum keuangan dinas dan *form input* data umum keuangan dinas yang akan disimpan. Pada bagian data pegawai dinas terdapat tombol “cetak SPPD” untuk mencetak SPPD masing-masing pegawai. Tombol “Masukkan data keuangan” untuk menampilkan *form input* data keuangan dinas masing-masing pegawai dan melakukan pengecekan terhadap data tersebut. Kemudian tombol “cetak kwitansi” untuk melakukan pencetakan kwitansi uang perjalanan dinas masing-masing pegawai. Jumlah tabel pegawai sesuai dengan jumlah pegawai yang bertugas pada suatu perjalanan dinas.

6.3.4 Antarmuka Daftar Sidang



Gambar 6. 4 Antarmuka Daftar Sidang

Rancangan antarmuka daftar sidang pada Gambar 5.12 diimplementasikan ke antarmuka daftar sidang pada Gambar 6.4. Antarmuka daftar sidang merupakan tampilan yang akan dilihat oleh pengguna ketika melihat daftar sidang yang telah tercatat. Selain itu antarmuka ini juga dibuka oleh pengguna ketika akan memasukkan data sidang atau akan melihat detail data sidang. Antarmuka daftar sidang memiliki beberapa komponen yaitu *icon user* di pojok kanan atas untuk melakukan *logout* dan *side bar* menu di bagian kiri untuk masuk ke menu yang ada pada sistem. Kemudian di bagian konten terdapat tombol “Masukkan Data Sidang” untuk menampilkan *form input* data sidang dan menyimpannya ke *database* sistem. Kolom “rows” untuk mengatur banyak data yang ditampilkan dalam satu tampilan layar. Kemudian kolom “search” untuk memasukkan kata kunci dari data yang akan dicari pada tabel. Pada bagian konten juga terdapat tabel yang berisi daftar sidang.



6.3.5 Antarmuka Daftar Laporan Rekapitulasi

The screenshot shows a web application interface for generating a 'UANG MAKAN' (Food Money) report. The interface is titled 'Dinas Luar' and 'Luluk Mukarramah'. It features a sidebar menu with options like 'Absensi', 'Lembur', and 'Rekapitulasi'. The main content area is titled 'UANG MAKAN' and contains a 'Cetak Rekapitulasi UANG MAKAN' section. This section has input fields for 'Masukkan tahun' (Year) and 'Pilih Bulan:' (Select Month), with 'Januari' selected. A 'Cetak' button is present. Below this is a 'DAFTAR REKAPITULASI UANG MAKAN' table. The table has columns for 'Tahun', 'Bulan', 'COL. II', 'COL. III, IV', 'Jumlah', 'PPH GOL II', 'PPH GOL III, IV', 'Jumlah', 'Jumlah Bersih', and 'Tanggal Cetak'. The table displays two rows of data for the year 2018. The first row is for August (2018-08-30) and the second row is for March (2018-09-12). A search bar and pagination controls are also visible.

Tahun	Bulan	GOL. II	GOL. III, IV	Jumlah	PPH GOL II	PPH GOL III, IV	Jumlah	Jumlah Bersih	Tanggal Cetak
2018		1200000	7230000	8430000	0	0	0	8430000	2018-08-30
2018	Maret	1140000	6915000	8055000	0	407250	407250	7647750	2018-09-12

Gambar 6. 5 Antarmuka Daftar Laporan Rekapitulasi

Rancangan antarmuka daftar laporan rekapitulasi pada Gambar 5.13 diimplementasikan kedalam antarmuka daftar laporan rekapitulasi pada Gambar 6.5. Antarmuka daftar laporan rekapitulasi merupakan tampilan yang akan dilihat oleh pengguna ketika melihat daftar laporan rekapitulasi uang makan, uang lembur, tunjangan daerah, atau tunjangan prestasi yang telah tercatat. Selain itu antarmuka ini juga dibuka oleh pengguna ketika akan mencetak laporan rekapitulasi. Antarmuka daftar laporan rekapitulasi memiliki beberapa komponen yaitu *icon user* di pojok kanan atas untuk melakukan *logout* dan *side bar* menu di bagian kiri untuk masuk ke menu yang ada pada sistem. Kemudian di bagian konten terdapat komponen cetak laporan rekapitulasi yang didalamnya terdapat kolom “tahun” untuk memasukkan tahun laporan, kolom “bulan” untuk memasukkan bulan laporan dan tombol “cetak” untuk mencetak laporan rekapitulasi sesuai bilan dan tahun. Pada komponen daftar laporan rekapitulasi terdapat kolom “rows” untuk mengatur banyak data yang ditampilkan dalam satu tampilan layar. Kemudian kolom “search” untuk memasukkan kata kunci dari data yang akan dicari pada tabel. Pada bagian ini juga terdapat tabel yang berisi daftar laporan rekapitulasi.

6.3.6 Penambahan Kebutuhan Fungsional

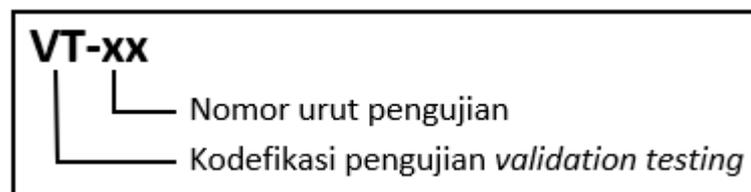
Saat dilakukan implementasi pada fase *construction*, dibutuhkan penambahan kebutuhan fungsional untuk mendukung fitur SIKAP-FT-2 yaitu sistem dapat mengelola data laporan rekapitulasi melakukan pencetakan laporan rekapitulasi keuangan tersebut. Penentuan uang lembur, uang makan, dan tunjangan prestasi membutuhkan data hari libur nasional. Sehingga ditambahkan kebutuhan fungsional SIKAP-FR-21 pada tabel kebutuhan fungsional.

BAB 7 PENGUJIAN

Bab pengujian berisi hasil tahap pengujian yang dilakukan pada Sistem Informasi Keuangan Pegawai. Pengujian yang dilakukan yaitu pengujian *validation testing* untuk menguji apakah sistem yang dibangun sesuai dengan spesifikasi atau kebutuhan awal. Selain itu terdapat pengujian *compatibility testing* untuk menguji apakah sistem yang dibangun memenuhi kebutuhan nonfungsional yang telah ditetapkan. Selanjutnya pengujian *usability testing* untuk menilai seberapa mudah antarmuka pengguna digunakan.

7.1 Pengujian *Validation Testing*

Pengujian *validation testing* merupakan salah satu metode pengujian *black box testing* untuk menguji apakah sistem yang dibangun dapat berjalan sesuai dengan spesifikasi awal yang telah ditetapkan (harapan pengguna). Agar sistem dapat diterima oleh organisasi maka sistem harus memenuhi harapan pengguna dalam hal ini organisasi. Selain itu fungsionalitas sistem harus berjalan dengan baik agar tidak menghambat kinerja organisasi ketika digunakan. Oleh karena itu dilakukan pengujian validasi untuk memastikan fungsionalitas sistem berjalan sesuai spesifikasi dan dapat diterima oleh organisasi. Subbab ini menjelaskan pengujian *validation testing* terhadap masing-masing *use case* pada bab analisis persyaratan. Menurut Pressman (2010) *use case* menyediakan skenario yang memiliki kemungkinan tinggi untuk menemukan kesalahan dalam persyaratan interaksi pengguna. Oleh karena itu, pengujian dilakukan berdasarkan *use case scenario* pada masing-masing *use case*.



Gambar 7. 1 Kodefikasi hasil pengujian *validation testing*

Gambar 7.1 menunjukkan kodefikasi pengujian *validation testing*. Terdapat beberapa fungsional yang akan diuji diantaranya memasukkan data perjalanan dinas, mengecek data SPJ, membuat laporan rekapitulasi, memasukkan data sidang, serta mencetak daftar penerima dan kwitansi uang sidang. Alur pengujian sesuai dengan *basic flow* dan *alternative flow* pada *use case scenario* masing-masing *use case*. Rancangan pengujian dapat dilihat pada Tabel 7.1 hingga Tabel 7.8. template dokumentasi *test case* mengikuti template dari Agarwal, Tayal, Gupta (2010) dengan beberapa penyesuaian. *Test attribute* digantikan dengan kode persyaratan untuk mengetahui pengujian dilakukan untuk mengetahui persyaratan sistem yang ingin dipenuhi. *Test process* dipecah menjadi test case dan prosedur uji dan hasil yang diharapkan. Hasil pengujian akan disatukan dalam tabel hasil pengujian seluruh kasus uji.

**Tabel 7. 1 Rancangan Pengujian Kasus Uji Memasukkan Data perjalanan dinas :
Basic Flow**

Kode Pengujian	VT-01
Nama Kasus Uji	Kasus uji memasukkan data perjalanan dinas
Kode Persyaratan	SIKAP-FR-22
Fokus pengujian	Validation
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan penambahan data perjalanan dinas.
Test Case	Menambahkan data perjalanan dinas dengan memasukkan data pegawai dinas yang belum tercatat ditugaskan melakukan perjalanan dinas pada tanggal perjalanan dinas yang akan dimasukkan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi data perjalanan dinas pada formulir perjalanan dinas. Pada kolom nama dan nip pegawai diisi dengan data pegawai yang belum tercatat melakukan perjalanan dinas lain. 2. Penguji menekan tombol “simpan”.
Hasil yang Diharapkan	Sistem berhasil menambahkan data dan menampilkan pesan bahwa data berhasil disimpan.

**Tabel 7. 2 Rancangan Pengujian Kasus Uji Memasukkan Data perjalanan dinas :
Alternative Flow**

Kode Pengujian	VT-02
Nama Kasus Uji	Kasus uji memasukkan data perjalanan dinas
Kode Persyaratan	SIKAP-FR-22
Fokus pengujian	Validation
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan penambahan data perjalanan dinas namun tidak dapat memasukkan data pegawai yang telah tercatat ditugaskan melakukan perjalanan dinas pada tanggal perjalanan dinas yang akan dimasukkan.
Test Case	menambahkan data perjalanan dinas dengan memasukkan data pegawai dinas yang telah tercatat ditugaskan melakukan perjalanan dinas pada tanggal perjalanan dinas yang akan dimasukkan.

**Tabel 7.2 Rancangan Pengujian Kasus Uji Memasukkan Data perjalanan dinas :
Alternative Flow (Lanjutan)**

Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi data perjalanan dinas pada formulir perjalanan dinas. Pada kolom nama dan nip salah satu pegawai diisi dengan data pegawai yang telah tercatat melakukan perjalanan dinas lain. 2. Penguji menekan tombol “simpan”.
Hasil yang Diharapkan	Sistem menampilkan pesan bahwa data pegawai yang telah tercatat di perjalanan dinas lain tidak dapat disimpan.

Tabel 7. 3 Rancangan Pengujian Kasus Uji Mengecek SPJ

Kode Pengujian	VT-03
Nama Kasus Uji	Kasus uji mengecek SPJ
Kode Persyaratan	SIKAP-FR-32 dan SIKAP-FR-33
Fokus pengujian	Validation
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pengecekan data SPJ dan menampilkan pesan bahwa data telah sesuai untuk data yang sesuai dengan ketentuan uang perjalanan dinas.
Test Case	Mengecek data SPJ yang sesuai dengan ketentuan uang perjalanan dinas.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi data keuangan perjalan dinas pada formulir keuangan perjalanan dinas dengan data yang sesuai ketentuan uang perjalanan dinas. 2. Penguji menekan tombol “cek”.
Hasil yang Diharapkan	Sistem menampilkan pesan bahwa data keuang dinas sesuai dengan ketentuan.

**Tabel 7. 4 Rancangan Pengujian Kasus Uji Memasukkan Data perjalanan dinas :
Alternative Flow**

Kode Pengujian	VT-04
Nama Kasus Uji	Kasus uji mengecek SPJ
Kode Persyaratan	SIKAP-FR-32 dan SIKAP-FR-33

**Tabel 7.4 Rancangan Pengujian Kasus Uji Memasukkan Data perjalanan dinas :
Alternative Flow (Lanjutan)**

Fokus pengujian	Validation
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pengecekan data SPJ dan menampilkan pesan bahwa data tidak sesuai untuk data yang tidak sesuai dengan ketentuan uang perjalanan dinas.
Test Case	Mengecek data SPJ yang tidak sesuai dengan ketentuan uang perjalanan dinas.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi data keuangan perjalan dinas pada formulir keuangan perjalanan dinas dengan data yang tidak sesuai ketentuan uang perjalanan dinas. 2. Penguji menekan tombol “cek”.
Hasil yang Diharapkan	Sistem menampilkan pesan bahwa data keuang dinas tidak sesuai dengan ketekapan.

Tabel 7. 5 Rancangan Pengujian Kasus Uji Membuat Laporan Rekapitulasi

Kode Pengujian	VT-05
Nama Kasus Uji	Kasus uji membuat laporan rekapitulasi
Kode Persyaratan	SIKAP-FR-8
Fokus pengujian	Validation
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pembuatan laporan rekapitulasi uang makan sesuai dengan ketentuan uang makan.
Test Case	Mencetak laporan rekapitulasi uang makan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji masuk ke menu laporan rekapitulasi uang makan. 2. Penguji mengisi tahun dan bulan laporan rekapitulasi uang makan. 3. Penguji menekan tombol “cetak”.
Hasil yang Diharapkan	Sistem menampilkan laporan rekapitulasi uang makan dalam bentuk <i>file</i> pdf.

Tabel 7. 6 Rancangan Pengujian Kasus Uji Memasukkan Data Sidang : *Basic Flow*

Kode Pengujian	VT-06
Nama Kasus Uji	Kasus uji memasukkan data sidang
Kode Persyaratan	SIKAP-FR-38
Fokus pengujian	Validation
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan penambahan data sidang.
Test Case	Menambahkan data sidang dengan memasukkan data pegawai sidang yang belum tercatat mengikuti sidang pada tanggal sidang yang akan dimasukkan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi data sidang pada formulir sidang. Pada kolom nama dan nip pegawai diisi dengan data pegawai yang belum tercatat mengikuti sidang pada tanggal tersebut. 2. Penguji menekan tombol “simpan”.
Hasil yang Diharapkan	Sistem berhasil menambahkan data dan menampilkan pesan bahwa data berhasil disimpan.

Tabel 7. 7 Rancangan Pengujian Kasus Uji Memasukkan Data Sidang : *Alternative Flow*

Kode Pengujian	VT-07
Nama Kasus Uji	Kasus uji memasukkan data sidang
Kode Persyaratan	SIKAP-FR-38
Fokus pengujian	Validation
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan penambahan data sidang namun tidak dapat menambahkan pegawai yang telah tercatat mengikuti sidang pada tanggal yang sama.
Test Case	Menambahkan data sidang dengan memasukkan data pegawai sidang yang telah tercatat mengikuti sidang pada tanggal sidang yang akan dimasukkan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji mengisi data sidang pada formulir sidang. Pada salah satu kolom nama dan nip pegawai diisi dengan data pegawai yang telah tercatat mengikuti sidang pada tanggal tersebut. 2. Penguji menekan tombol “simpan”.

**Tabel 7.7 Rancangan Pengujian Kasus Uji Memasukkan Data Sidang :
Alternative Flow (Lanjutan)**

Hasil yang Diharapkan	Sistem menampilkan pesan bahwa data pegawai yang telah tercatat mengikuti sidang di hari yang sama tidak dapat dimasukkan.
------------------------------	--

Tabel 7. 8 Rancangan Pengujian Kasus Uji Mencetak Daftar Penerima dan Kwitansi Uang Sidang

Kode Pengujian	VT-08
Nama Kasus Uji	Mencetak daftar penerima dan kwitansi uang sidang
Kode Persyaratan	SIKAP-FR-37
Fokus pengujian	Validation
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pencetakan daftar penerima dan kwitansi uang sidang.
Test Case	Mencetak daftarpenerima dan kwitansi uang sidang.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji menekan tombol “detail” pada data sidang. 2. Penguji mengisi tanggal dan nomor kwitansi pembayaran uang sidang. 3. Penguji menekan tombol “cetak daftar penerima dan kwitansi uang sidang”.
Hasil yang Diharapkan	Sistem menampilkan daftar penerima dan kwitansi uang sidang dalam bentuk file pdf.

Hasil pengujian *Validation* testing dijelaskan pada Tabel 7.9. Kolom kode pengujian merupakan kode dari pengujian yang dilakukan. Kolom kasus uji merupakan kasus uji yang dilakukan untuk menguji sistem berdasarkan *use case* tertentu. Kolom hasil yang diharapkan merupakan hasil atau keadaan yang akan tercapai dalam suatu kasus uji sesuai dengan *use case scenario*. Sedangkan kolom hasil pengujian merupakan hasil atau keadaan yang didapatkan setelah melakukan pengujian. Kolom status merupakan status pengujian yang didapat dengan membandingkan hasil yang diharapkan dengan hasil pengujian. Status valid berarti hasil pengujian sesuai dengan hasil yang diharapkan.

Tabel 7. 9 Hasil Pengujian *Validation Testing*

Kode Pengujian	Kasus Uji	Hasil yang Diharapkan	Hasil Pengujian	Status
VT-01	Menambahkan data perjalanan dinas dengan memasukkan data pegawai dinas yang belum tercatat ditugaskan melakukan perjalanan dinas pada tanggal perjalanan dinas yang akan dimasukkan	Sistem menampilkan pesan bahwa data berhasil disimpan.	Sistem menampilkan pesan bahwa data berhasil disimpan.	Valid
VT-02	Menambahkan data perjalanan dinas dengan memasukkan data pegawai dinas yang telah tercatat ditugaskan melakukan perjalanan dinas pada tanggal perjalanan dinas yang akan dimasukkan	Sistem menampilkan pesan bahwa data pegawai yang telah tercatat di perjalanan dinas lain tidak dapat disimpan.	Sistem menampilkan pesan bahwa data pegawai yang telah tercatat di perjalanan dinas lain tidak dapat disimpan.	Valid
VT-03	Mengecek data SPJ yang sesuai dengan ketentuan uang perjalanan dinas	Sistem menampilkan pesan bahwa data keuangan dinas sesuai dengan ketentuan	Sistem menampilkan pesan bahwa data keuangan dinas sesuai dengan ketentuan	Valid
VT-04	Mengecek data SPJ yang tidak sesuai dengan ketentuan uang perjalanan dinas	Sistem menampilkan pesan bahwa data keuangan dinas tidak sesuai	Sistem menampilkan pesan bahwa data keuangan dinas tidak sesuai	Valid

Kode Pengujian	Kasus Uji	Hasil yang Diharapkan	Hasil Pengujian	Status
		dengan ketekapan	dengan ketekapan	

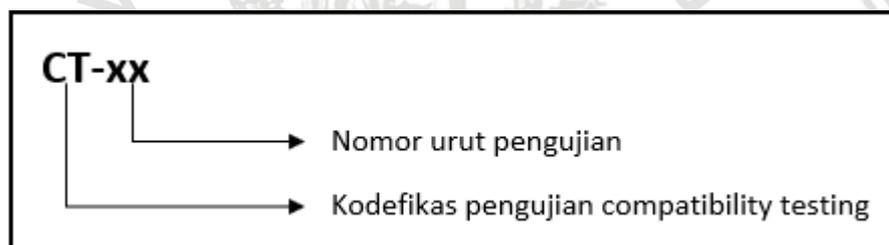
Tabel 7.9 Hasil Pengujian *Validation Testing* (Lanjutan)

Kode Pengujian	Kasus Uji	Hasil yang Diharapkan	Hasil Pengujian	Status
VT-05	Mencetak laporan rekapitulasi uang makan	Sistem menampilkan laporan rekapitulasi uang makan dalam bentuk file pdf	Sistem menampilkan laporan rekapitulasi uang makan dalam bentuk file pdf	Valid
VT-06	Menambahkan data sidang dengan memasukkan data pegawai sidang yang belum tercatat mengikuti sidang pada tanggal sidang yang akan dimasukkan	Sistem menampilkan pesan bahwa data berhasil disimpan.	Sistem menampilkan pesan bahwa data berhasil disimpan.	Valid
VT-07	Menambahkan data sidang dengan memasukkan data pegawai sidang yang telah tercatat mengikuti sidang pada tanggal sidang yang akan dimasukkan	Sistem menampilkan pesan bahwa data pegawai yang telah tercatat mengikuti sidang di hari yang sama tidak dapat dimasukkan.	Sistem menampilkan pesan bahwa data pegawai yang telah tercatat mengikuti sidang di hari yang sama tidak dapat dimasukkan.	Valid
VT-08	Mencetak daftarpenerima dan kwitansi uang sidang	Sistem menampilkan daftar penerima dan kwitansi uang sidang dalam bentuk file pdf.	Sistem menampilkan daftar penerima dan kwitansi uang sidang dalam bentuk file pdf.	Valid

Hasil pengujian Sistem Informasi Keuangan Pegawai terhadap beberapa fungsionalitas dengan metode *validation testing* menunjukkan bahwa seluruh kasus uji yang dilakukan menghasilkan status valid. Hal tersebut menunjukkan bahwa fungsionalitas sistem berjalan dengan baik dan sesuai dengan spesifikasi. Fungsionalitas sistem dibuat berdasarkan kebutuhan pengguna, sehingga dengan terpenuhinya fungsionalitas sistem dengan baik maka kebutuhan pengguna juga terpenuhi.

7.2 Pengujian *Compatibility Testing*

Compatibility testing dilakukan untuk menguji apakah sistem yang dibangun dapat berjalan dengan baik di berbagai lingkungan yang berbeda. Pengguna yang akan mengakses Sistem Informasi Keuangan Pegawai di Biro Organisasi menggunakan beberapa *browser* yang berbeda. Browser yang berbeda terkadang menghasilkan hasil atau tampilan yang berbeda. Oleh karena itu sistem harus dapat dijalankan di beberapa browser yang berbeda dengan baik. Pada bab analisis persyaratan terdapat kebutuhan non fungsional sistem yaitu sistem dapat dijalankan di berbagai *browser*. Sehingga dilakukan pengujian kompatibilitas sistem terhadap beberapa *browser* untuk memenuhi kebutuhan nonfungsional tersebut. Gambar 7.2 merupakan kodefikasi pengujian kompatibilitas.



Gambar 7. 2 Kodefikasi Penguian kompatibilitas

Pengujian kompatibilitas dilakukan dengan bantuan aplikasi *sortSite* yang menghasilkan temuan dengan beberapa kategori permasalahan seperti yang dijelaskan pada Gambar 2.11 bab Landasan Teori. Beberapa komponen penyusun Sistem Informasi Keuangan Pegawai yang akan diuji yaitu *script* html, php, dan javascript serta beberapa *library* pendukung sistem yaitu *Jquery*, *ExcelReader*, *Fpdf*, *bootstrap*, *font-owsome*, *Ionicons*, *datetimepicker*, *datatable*, *CodeIgniter* (CI), dan *Jquery-ui*. Pengujian dilakukan dengan memasukkan alamat *website* Sistem Informasi Keuangan Pegawai pada aplikasi *sortSite* kemudian dijalankan. Setelah itu akan dilakukan pengecekan hasil pengujian terhadap seluruh sistem dan hasilnya akan ditampilkan kategori permasalahan pada masing-masing browser. Rancangan pengujian dapat dilihat pada Tabel 7.10. template dokumentasi *test case* mengikuti template dari Agarwal, Tayal, Gupta (2010) dengan beberapa penyesuaian. *Test attribute* digantikan dengan kode persyaratan untuk mengetahui pengujian dilakukan untuk mengetahui persyaratan sistem yang ingin dipenuhi. *Test proses* dipecah menjadi test case dan prosedur uji dan hasil yang diharapkan. Hasil pengujian akan disatukan dalam tabel hasil pengujian seluruh kasus uji. *Browser* yang diuji adalah *mozilla firefox*, *opera*, *google chrome*, *IOS*, dan *android* karena browser tersebut yang digunakan oleh Biro Organisasi.

Tabel 7. 10 Rancangan Pengujian Kasus Uji Menguji Kompatibilitas Sistem Pada Browser Mozilla Firefox, Opera, Google Chrome, IOS, dan Android

Kode Pengujian	CT-01
Nama Kasus Uji	Kasus uji kompatibilitas pada <i>browser mozilla firefox</i>
Kode Persyaratan	SIKAP-NR-1
Fokus pengujian	Kompatibilitas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat berjalan dengan baik pada <i>browser mozilla firefox, opera, google chrome, IOS, dan android</i>
Test Case	Menguji kompatibilitas sistem pada <i>browser mozilla firefox, opera, google chrome, IOS, dan android</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji menjalankan aplikasi <i>sortsite</i> 2. Penguji memasukkan url sistem pada menu bar address 3. Penguji melakukan checking dengan menekan tombol check 4. Aplikasi <i>sortsite</i> melakukan pemindaian setiap halaman situs dengan menjalankan <i>checkpoints</i>, kemudian memeriksa halaman yang terhubung dengan halaman tersebut. Kemudian <i>sortsite</i> akan memindai halam baru yang ditemukan dan mengulang proses ini hingga halaman terakhir. 5. Aplikasi <i>sortsite</i> menampilkan hasil pengujian pada beberapa browser.
Hasil yang Diharapkan	Sistem dapat berjalan dengan baik pada <i>browser mozilla firefox, opera, google chrome, IOS, dan android</i>

Compatibility 0 pages with browser specific issues 0% have issues, better than average

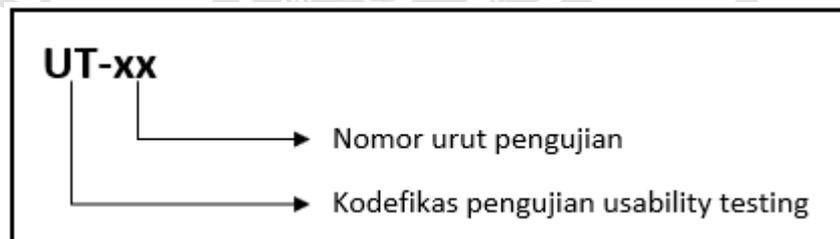
Browser Version	Firefox	Opera	Chrome	iOS	Android
Version	61	54	68	11	4*
Critical Issues	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓

Gambar 7. 3 Hasil Pengujian Kompatibilitas

Hasil pengujian pada Gambar 7.3. menunjukkan tidak terdapat *critical issues*, *major issues* ataupun *minor issues* pada *broser mozilla firefox* versi 61, *opera* versi 54, *google chrome* versi 68, *IOS* versi 11, dan *android* versi 4. Hasil pengujian juga menunjukkan bahwa pengujian kompatibilitas SIKAP mendapatkan hasil diatas rata-rata (*better than average*). Berdasarkan hasil pengujian tersebut sistem dapat diterima karena tidak terdapat *critical issues*, *major issues* ataupun *minor issues* pada seluruh *browser* yang diuji dan dapat menggagu jalanya atau tampilan sistem. Browser yang biasanya digunakan oleh pengguna yang akan mengakses Sistem informasi keuangan pegawai adalah *mozilla firefox*, *opera*, *google chrome*, *IOS*, dan *android*.

7.3 Pengujian Usability Testing

Pengujian *usability* dilakukan untuk menilai seberapa mudah antarmuka pengguna digunakan. *Usability* dapat diukur menggunakan *System Usability Scale* (SUS) yang berupa skala likert. Kebanyakan pengguna yang akan menggunakan Sistem Informasi Keuangan Pegawai pada Biro Organisasi belum terbiasa dengan sistem informasi. Aktifitas pengelolaan keuangan yang ada belum menggunakan sistem informasi dan masih dilakukan secara manual. Oleh karena itu antarmuka sistem informasi yang dikembangkan harus mudah digunakan dan dimengerti oleh pengguna. Seberapa mudah antarmuka pengguna dapat digunakan bisa diuji menggunakan pengujian usabilitas. Gambar 7.4 merupakan kodefikasi pengujian usabilitas.



Gambar 7. 4 Kodefikasi Pengujian Usabilitas

Pengujian akan dilakukan oleh empat orang responden. Masing-masing responden akan mencoba menggunakan Sistem Informasi Keuangan Pegawai sesuai dengan perannya masing-masing. Satu orang sebagai bendahara biro, dua orang sebagai bendahara bagian, dan satu orang sebagai pegawai biro. Kemudian masing-masing responden akan mengisi *form* pernyataan untuk menilai *usability* dari Sistem informasi keuangan pegawai menggunakan *form* pernyataan SUS yang dikembangkan pada tahun 1986 oleh Jonh Brooke. *Form* pernyataan SUS untuk Sistem Informasi Keuangan Pegawai dapat dilihat pada Tabel 7.10.

Tabel 7. 11 Form Pernyataan SUS

NO	PERTANYAAN	JAWABAN (sangat tidak setuju - sangat setuju)				
		1	2	3	4	5
1	Saya akan sering menggunakan Sistem Informasi Keuangan Pegawai (SIKAP)					
2	Sistem Informasi Keuangan Pegawai (SIKAP) terlalu kompleks					
3	Sistem Informasi Keuangan Pegawai (SIKAP) mudah digunakan					
4	Saya membutuhkan bantuan orang yang mengerti untuk menggunakan Sistem Informasi Keuangan Pegawai (SIKAP)					
5	Fungsi-fungsi pada Sistem Informasi Keuangan Pegawai (SIKAP) sudah terintegrasi dengan baik					
6	Tertalu banyak ketidakkonsistenan dalam Sistem Informasi Keuangan Pegawai (SIKAP)					
7	Saya rasa ada banyak orang yang dapat belajar menggunakan Sistem Informasi Keuangan Pegawai (SIKAP) dengan cepat					
8	Sistem Informasi Keuangan Pegawai (SIKAP) sulit untuk digunakan					
9	Saya merasa percaya diri saat menggunakan Sistem Informasi Keuangan Pegawai (SIKAP)					
10	Saya harus belajar banyak hal sebelum dapat menggunakan Sistem Informasi Keuangan Pegawai (SIKAP)					

Rancangan pengujian dapat dilihat pada Tabel 7.12 hingga Tabel 7.14. template dokumentasi *test case* mengikuti template dari Agarwal, Tayal, Gupta (2010) dengan beberapa penyesuaian. *Test attribute* digantikan dengan kode persyaratan untuk mengetahui pengujian dilakukan untuk mengetahui persyaratan sistem yang ingin dipenuhi. *Test proses* dipecah menjadi test case dan

prosedur uji dan hasil yang diharapkan. Hasil pengujian akan disatukan dalam tabel hasil pengujian seluruh kasus uji.

Tabel 7. 12 Rancangan Kasus Uji Menguji Usabilitas Sistem Informasi Keuangan Pegawai (Pegawai Biro)

Kode Pengujian	UT-01
Nama Kasus Uji	Kasus uji usabilitas Sistem Informasi Keuangan Pegawai
Fokus pengujian	Usabilitas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem mudah digunakan oleh pegawai biro.
Test Case	Menguji usabilitas Sistem Informasi Keuangan Pegawai yang berperan sebagai pegawai biro.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji masuk ke Sistem Informasi Keuangan Pegawai sebagai pegawai. 2. Menjalankan fungsi-fungsi yang dapat dijalankan oleh pegawai biro. 3. Mengisi form pernyataan SUS.
Hasil yang Diharapkan	Sistem informasi mudah digunakan.

Tabel 7. 13 Rancangan Kasus Uji Menguji Usabilitas Sistem Informasi Keuangan Pegawai (Bendahara Bagian)

Kode Pengujian	UT-02
Nama Kasus Uji	Kasus uji usabilitas Sistem Informasi Keuangan Pegawai
Fokus pengujian	Usabilitas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem mudah digunakan oleh bendahara bagian.
Test Case	Menguji usabilitas Sistem Informasi Keuangan Pegawai yang berperan sebagai bendahara bagian.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji masuk ke Sistem Informasi Keuangan Pegawai sebagai bendahara bagian. 2. Menjalankan fungsi-fungsi yang dapat dijalankan oleh bendahara bagian. 3. Mengisi form pernyataan SUS.
Hasil yang Diharapkan	Sistem informasi mudah digunakan.

Tabel 7. 14 Rancangan Kasus Uji Menguji Usabilitas Sistem Informasi Keuangan Pegawai (Bendahara Biro)

Kode Pengujian	UT-03
Nama Kasus Uji	Kasus uji usabilitas Sistem Informasi Keuangan Pegawai
Fokus pengujian	Usabilitas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem mudah digunakan oleh bendahara biro.
Test Case	Menguji usabilitas Sistem Informasi Keuangan Pegawai yang berperan sebagai bendahara biro.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji masuk ke Sistem Informasi Keuangan Pegawai sebagai pegawai. 2. Menjalankan fungsi-fungsi yang dapat dijalankan oleh bendahara biro. 3. Mengisi form pernyataan SUS.
Hasil yang Diharapkan	Sistem informasi mudah digunakan.

Skor masing-masing responden terhadap usabilitas Sistem Informasi Keuangan Pegawai akan dihitung untuk mendapatkan nilai usabilitas sistem. Pernyataan ganjil merupakan pernyataan positif dan skornya didapat dari nilai yang diberikan responden dikurangi dengan 1. Sedangkan pernyataan genap merupakan pernyataan negatif dan skornya didapat dari 5 dikurangi dengan nilai yang diberikan oleh responden. Tabel 7.15 merupakan hasil dari pengujian *usability*.

Tabel 7. 15 Hasil Pengujian *Usability Testing*

Nama	Jabatan	Skor	Jumlah Skor	Nilai SU (skor x 2.5)
Edi Hartono, SE	Bendahara Biro	4+4+3+2+2 +4+4+4+4+4	36	90
Ramli	Pengadministrasi Keuangan	4+3+4+1+3 +3+3+3+3+1	28	70
Happy Poerbodianti, SE	Penata Laporan Keuangan	4+4+4+2+3 +4+4+4+4+4	37	92.5
Moh. Aziz A. Basid, S.IP	Pengelola Kepegawaian	3+4+4+2+3 +4+4+4+4+4	36	90

Berdasarkan hasil pengujian usabilitas pada Tabel 7.15, *grade* dari nilai *System Usability*

$$(90 + 70 + 92.5 + 90) / 4 = 342.5 / 4 = 85.625$$

Berdasarkan interpretasi *grade* skor SUS (Bangor, Kortum, & Miller, 2009), Nilai *System Usability* 85.625 mendapatkan *grade* B. Bangor, Kortum dan Miller (2009) juga mengatakan bahwa sistem dikatakan baik apabila mendapat skor 70 keatas yaitu dengan *grade* C keatas. Hasil pengujian usabilitas Sistem Informasi keuangan pegawai mendapat *grade* B yang berarti Sistem Informasi keuangan pegawai dikatakan baik dan dapat digunakan oleh pengguna dengan mudah.

7.4 Time Analysis Pada Proses Bisnis *As-is* dan *To-be*

Setelah dilakukan implementasi sistem informasi keuangan pegawai, dilakukan analisis perubahan waktu antara proses bisnis *as-is* dan *to-be*. *Time analysis* dilakukan dengan memasukkan waktu penyelesaian masing-masing aktivitas pada proses bisnis *as-is* dan *to-be*. Proses bisnis *as-is* dan *to-be* dijalankan dengan *input* yang sama. Ketika pada proses bisnis *as-is* dimasukkan 10 orang pegawai, maka pada proses bisnis *to-be* juga dimasukkan 10 orang pegawai.

7.4.1 Proses Bisnis Pencairan Uang Sidang

Perubahan Aktivitas pada proses bisnis pencairan uang sidang setelah dilakukan simulasi *time analysis* dengan waktu 60 menit pada aktivitas membuat rekapitulasi pembayaran uang sidang proses bisnis *as-is* serta 20 menit pada proses bisnis *to-be* dapat dilihat pada Tabel 7.16. Pada proses bisnis *as-is* pembuatan rekapitulasi uang sidang dilakukan oleh bendahara biro dengan memasukkan data pegawai yang mengikuti sidang satu persatu. Kemudian penentuan nominal uang sidang dilakukan dengan menginputkan secara manual nominal uang sidang berdasarkan pedoman ketentuan uang sidang. Proses perhitungan total uang sidang seluruh pegawai yang mengikuti sidang juga dilakukan secara manual. Pada proses bisnis *to-be* aktivitas pembuatan rekapitulasi uang sidang dilakukan oleh bendahara biro dengan bantuan sistem informasi keuangan pegawai. Setelah bendahara biro menginputkan nama pegawai, sistem akan membuat rekapitulasi uang makan dengan data pegawai dari *database* serta menentukan uang sidang masing-masing pegawai sesuai tabel ketentuan yang ada pada *database*. Perhitungan total uang sidang seluruh pegawai yang mengikuti sidang juga dilakukan oleh sistem.

Tabel 7. 16 Perubahan Aktivitas Proses Bisnis Pencairan Uang Sidang

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses Bisnis <i>to-be</i>	Keterangan
Bendahara biro	Membuat rekapitulasi uang sidang (waktu 60 menit)	Membuat rekapitulasi uang sidang (waktu 20 menit)	Mempercepat waktu selama 40 menit



7.4.2 Proses Bisnis Pencairan Uang Perjalanan Dinas

Pada proses bisnis *as-is* dan *to-be* dilakukan simulasi pada level time analisis. Estimasi waktu pada simulasi masing-masing aktivitas pada proses bisnis *as-is* dan *to-be* yaitu 5 dan 3 menit pada aktivitas membuat SPT oleh bendahara biro, 10 dan 3 menit membuat SPPD oleh bendahara biro, 60 dan 60 menit membuat SPJ oleh pegawai, 60 dan 5 menit memeriksa SPJ oleh bendahara biro, serta 10 dan 3 menit membuat kwitansi oleh bendahara biro. Waktu tercepat yang dibutuhkan untuk menyelesaikan aktivitas-aktivitas tersebut pada proses bisnis *as-is* adalah 160 menit sedangkan pada poses bisnis *to-be* adalah 90 menit. Waktu paling lama yang dibutuhkan untuk menyelesaikan aktivitas-aktivitas tersebut pada proses bisnis *as-is* adalah 275 menit sedangkan pada proses bisnis *to-be* adalah 224 menit, dengan skenario terjadi ditemukan kesalahan pada proses memeriksa SPJ, sehingga pegawai perlu membuat ulang SPJ mereka. Persentase kemungkinan ditemukannya kesalahan dan tidak ditemukannya kesalahan pada aktivitas memeriksa SPJ adalah 50% dan 50%. Perubahan Aktivitas pada proses bisnis pencairan uang makan setelah dilakukan simulasi *time analysis* dapat dilihat pada Tabel 7.17.

Tabel 7. 17 Perubahan Aktivitas Proses Bisnis Pencairan Uang Perjalanan Dinas

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses Bisnis <i>to-be</i>	Keterangan
Bendahara Bagian	Membuat SPT (5 menit).	Membuat SPT (3 menit).	Mempercepat waktu dari 5 menit menjadi 3 menit.
Bendahara Bagian	Membuat SPPD (5 menit).	Membuat SPPD (3 menit).	Mempercepat waktu dari 5 menit menjadi 3 menit.
Bendahara Biro	Mengecek SPJ (60 menit).	Meng- <i>input</i> data SPJ (5 menit).	Mempercepat waktu dari 60 menit menjadi 8 menit.
		Mengecek data SPJ (3 menit).	
Bendahara Biro	Membuat Kwitansi (5 menit).	Membuat Kwitansi (3 menit).	Mempercepat waktu dari 5 menit menjadi 3 menit.

7.4.3 Proses Bisnis Pencairan Uang Makan

Pada proses bisnis *as-is* dan *to-be* dilakukan simulasi pada level time analisis. Estimasi waktu simulasi proses bisnis masing-masing aktivitas yaitu 120 menit pada aktivitas membuat laporan rekapitulasi uang makan oleh bendahara bagian, 15 menit pada aktivitas memeriksa laporan rekapitulasi uang makan oleh bendahara biro, 5 menit membuat surat permintaan pembayaran oleh bendahara biro, 5 menit membuat surat perintah pembayaran oleh bendahara biro, dan 5 menit membuat surat kuara untuk transfer uang makan oleh bendahara biro. Sedangkan

pada proses bisnis *to-be*, estimasi waktu yaitu 3 menit pada aktivitas meng-*import* presensi pegawai ke dalam sistem oleh bendahara biro, 1 menit me-*replace* presensi kosong dengan data absensi yang ada pada *database* secara otomatis oleh sistem, 1 menit menghitung uang makan secara otomatis oleh sistem, 3 menit membuat laporan uang makan oleh bendahara menggunakan data yang ada pada *database* sistem, 3 menit memeriksa laporan rekapitulasi uang makan bagian oleh bendahara biro dengan mencocokkan data laporan bagian dengan data di *database* sistem, 3 menit membuat surat permintaan pembayaran oleh bendahara biro, 3 menit membuat surat perintah pembayaran oleh bendahara biro, dan 3 menit membuat surat perintah pencairan dana uang makan oleh bendahara biro. Perubahan Aktivitas pada proses bisnis pencairan uang makan setelah dilakukan simulasi *time analysis* dapat dilihat pada Tabel 7.18.

Tabel 7. 18 Perubahan Aktivitas Proses Bisnis Pencairan Uang Makan

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses Bisnis <i>to-be</i>	Keterangan
Bendahara Biro	-	Mengimport Laporan Presensi (3 menit).	Mempercepat waktu dari 120 menit menjadi 8 menit.
Bendahara Bagian	Membuat Laporan Rekapitulasi Uang Makan (120 menit).	Me-replace data presensi kosong dengan data absensi (1 menit). Dilakukan otomatis oleh sistem.	
		Menghitung uang makan (1 menit). Dilakukan otomatis oleh sistem.	
		Membuat Laporan Rekapitulasi Uang Makan (3 menit)	
Bendahara Biro	Memeriksa Laporan Rekapitulasi Uang Makan (15 menit).	Memeriksa Laporan Rekapitulasi Uang Makan (3 menit).	Mempercepat waktu dari 15 menit menjadi 3 menit.
Bendahara Biro	Membuat Surat Permintaan Pembayaran (5 menit).	Membuat Surat Permintaan Pembayaran (3 menit).	Mempercepat waktu dari 5 menit menjadi 3 menit.

7.4.4 Proses Bisnis Pencairan Uang Lembur

Pada proses bisnis *as-is* dan *to-be* dilakukan simulasi level time analisis. Estimasi waktu simulasi yaitu 120 menit pada aktivitas membuat laporan rekapitulasi uang lembur oleh bendahara bagian, 15 menit pada aktivitas

memeriksa laporan rekapitulasi uang lembur oleh bendaha biro, 5 menit membuat surat permintaan pembayaran oleh bendahara biro, 5 menit membuat surat perintah pembayaran oleh bendahara biro, dan 5 menit membuat surat kuasa untuk transfer uang lembur oleh bendahara biro. Sedangkan pada *to-be* estimasi waktu simulasi yaitu 3 menit pada aktivitas meng-*import* presensi pegawai ke dalam sistem oleh bendahara biro, 2 menit menghitung uang lembur secara otomatis oleh sistem dengan mencocokkan data presensi yang melebihi jam pulang dengan data lembur yang ada pada sistem, 3 menit membuat laporan uang lembur oleh bendahara menggunakan data yang ada pada *database* sistem, 3 menit memeriksa laporan rekapitulasi uang lembur bagian oleh bendahara biro dengan mencocokkan data laporan bagian dengan data di *database* sistem, 3 menit membuat surat permintaan pembayaran oleh bendahara biro, 3 menit membuat surat perintah pembayaran oleh bendahara biro, dan 3 menit membuat surat perintah pencairan dana uang lembur oleh bendahara biro. Perubahan Aktivitas pada proses bisnis pencairan uang lembur setelah dilakukan simulasi *time analysis* dapat dilihat pada Tabel 7.19.

Tabel 7. 19 Perubahan Aktivitas Proses Bisnis Pencairan Uang Lembur

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses Bisnis <i>to-be</i>	Keterangan
Bendahara Biro	-	Mengimport Laporan Presensi (3 menit).	Mempercepat waktu dari 120 menit menjadi 8 menit.
Bendahara Bagian	Membuat Laporan Rekapitulasi Uang Lembur (120 menit).	Menghitung uang lembur (2 menit). Dilakukan otomatis oleh sistem. Membuat Laporan Rekapitulasi Uang Lembur (3 menit)	
Bendahara Biro	Memeriksa Laporan Rekapitulasi Uang Lembur (15 menit).	Memeriksa Laporan Rekapitulasi Uang Lembur (3 menit).	Mempercepat waktu dari 15 menit menjadi 3 menit.
Bendahara Biro	Membuat Surat Permintaan Pembayaran (5 menit).	Membuat Surat Permintaan Pembayaran (3 menit).	Mempercepat waktu dari 5 menit menjadi 3 menit.

7.4.5 Proses Bisnis Pencairan Tunjangan Prestasi

Pada proses bisnis *as-is* dan *to-be* dilakukan simulasi pada level *time analysis*. Estimasi waktu simulasi yaitu 120 menit pada aktivitas membuat laporan rekapitulasi tunjangan prestasi oleh bendahara bagian, 15 menit pada aktivitas memeriksa laporan rekapitulasi tunjangan prestasi oleh bendaha biro, 5 menit membuat surat permintaan pembayaran oleh bendahara biro, 5 menit membuat

surat perintah pembayaran oleh bendahara biro, dan 5 menit membuat surat kuasa untuk transfer tunjangan prestasi oleh bendahara biro. Sedangkan estimasi waktu simulasi *to-be* 3 menit pada aktivitas meng-*import* presensi pegawai ke dalam sistem oleh bendahara biro, 1 menit me-*replace* presensi kosong dengan data absensi yang ada pada *database* secara otomatis oleh sistem, 1 menit menghitung tunjangan prestasi secara otomatis oleh sistem, 3 menit membuat laporan tunjangan prestasi oleh bendahara menggunakan data yang ada pada *database* sistem, 3 menit memeriksa laporan rekapitulasi tunjangan prestasi bagian oleh bendahara biro dengan mencocokkan data laporan bagian dengan data di *database* sistem, 3 menit membuat surat permintaan pembayaran oleh bendahara biro, 3 menit membuat surat perintah pembayaran oleh bendahara biro, dan 3 menit membuat surat perintah pencairan dana tunjangan prestasi oleh bendahara biro. Perubahan Aktivitas pada proses bisnis pencairan tunjangan prestasi setelah dilakukan simulasi *time analysis* dapat dilihat pada 7.20.

Tabel 7. 20 Perubahan Aktivitas Proses Bisnis Pencairan Tunjangan prestasi

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses Bisnis <i>to-be</i>	Keterangan
Bendahara Biro	-	Mengimport Laporan Presensi (3 menit).	Mempercepat waktu dari 120 menit menjadi 8 menit.
Bendahara Bagian	Membuat Laporan Rekapitulasi Tunjangan prestasi (120 menit).	Me-replace data presensi kosong dengan data absensi (1 menit). Dilakukan otomatis oleh sistem.	
		Menghitung tunjangan prestasi (1 menit). Dilakukan otomatis oleh sistem.	
		Membuat Laporan Rekapitulasi Tunjangan prestasi (3 menit)	
Bendahara Biro	Memeriksa Laporan Rekapitulasi Tunjangan prestasi (15 menit).	Memeriksa Laporan Rekapitulasi Tunjangan prestasi (3 menit).	Mempercepat waktu dari 15 menit menjadi 3 menit.
Bendahara Biro	Membuat Surat Permintaan Pembayaran (5 menit).	Membuat Surat Permintaan Pembayaran (3 menit).	Mempercepat waktu dari 5 menit menjadi 3 menit.

7.4.6 Proses Bisnis Pencairan Tunjangan Daerah

Pada proses bisnis *as-is* dan *to-be* dilakukan simulasi pada level time analisis. Estimasi waktu simulasi proses bisnis yaitu 120 menit pada aktivitas membuat laporan rekapitulasi tunjangan daerah oleh bendahara bagian, 15 menit pada aktivitas memeriksa laporan rekapitulasi tunjangan daerah oleh bendahara biro, 5 menit membuat surat permintaan pembayaran oleh bendahara biro, 5 menit membuat surat perintah pembayaran oleh bendahara biro, dan 5 menit membuat surat kuasa untuk transfer tunjangan daerah oleh bendahara biro. Sedangkan estimasi waktu simulasi proses bisnis *to-be* yaitu 3 menit pada aktivitas mengimport presensi pegawai ke dalam sistem oleh bendahara biro, 1 menit menghitung tunjangan daerah secara otomatis oleh sistem, 3 menit membuat laporan tunjangan daerah oleh bendahara menggunakan data yang ada pada *database* sistem, 3 menit memeriksa laporan rekapitulasi tunjangan daerah bagian oleh bendahara biro dengan mencocokkan data laporan bagian dengan data di *database* sistem, 3 menit membuat surat permintaan pembayaran oleh bendahara biro, 3 menit membuat surat perintah pembayaran oleh bendahara biro, dan 3 menit membuat surat perintah pencairan dana tunjangan daerah oleh bendahara biro.

Perubahan Aktivitas pada proses bisnis pencairan tunjangan daerah setelah dilakukan simulasi *time analysis* dapat dilihat pada Tabel 7.21.

Tabel 7. 21 Perubahan Aktivitas Proses Bisnis Pencairan Tunjangan daerah

Unit Bisnis	Proses Bisnis <i>as-is</i>	Proses Bisnis <i>to-be</i>	Keterangan
Bendahara Biro	-	Mengimport Laporan Presensi (3 menit).	Mempercepat waktu dari 120 menit menjadi 8 menit.
Bendahara Bagian	Membuat Laporan Rekapitulasi Tunjangan daerah (120 menit).	Menghitung tunjangan daerah (1 menit). Dilakukan otomatis oleh sistem. Membuat Laporan Rekapitulasi Tunjangan daerah (3 menit)	
Bendahara Biro	Memeriksa Laporan Rekapitulasi Tunjangan daerah (15 menit).	Memeriksa Laporan Rekapitulasi Tunjangan daerah (3 menit).	Mempercepat waktu dari 15 menit menjadi 3 menit.
Bendahara Biro	Membuat Surat Permintaan Pembayaran (5 menit).	Membuat Surat Permintaan Pembayaran (3 menit).	Mempercepat waktu dari 5 menit menjadi 3 menit.

BAB 8 KESIMPULAN

8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat diambil kesimpulan bahwa jika Sistem Informasi Keuangan Pegawai (SIKAP) diterapkan, diperkirakan akan mampu mengatasi permasalahan yang terdapat pada rumusan masalah.

1. Proses administrasi keuangan pegawai seperti pencairan uang perjalanan dinas, uang sidang, tunjangan prestasi, tunjangan daerah, uang makan, dan uang lembur di biro organisasi dapat dipercepat dengan adanya Sistem Informasi Keuangan Pegawai yang dapat dilihat dari perbandingan perubahan proses bisnis pada bab analisis persyaratan. Proses pencairan uang makan dapat dipercepat pada aktivitas membuat daftar penerima dan kwitansi uang sidang sebesar 40 menit. Proses pencairan uang perjalanan dinas pada beberapa aktivitas dapat dipercepat sebesar 62 menit. Proses pencairan uang makan, uang lembur, tunjangan daerah dan tunjangan prestasi pada beberapa aktivitas dapat dipercepat sebesar 126 menit.
2. Dengan adanya Sistem Informasi Keuangan Pegawai dapat mengintegrasikan data dokumen-dokumen keuangan pegawai. Pada proses bisnis *to-be* pencairan uang perjalanan dinas, dapat dilihat bahwa aktivitas pembuatan SPT akan menyimpan data SPT yang diinputkan pengguna ke sebuah database, kemudian SPPD serta kwitansi akan dibuat berdasarkan data yang tersimpan dalam *database*. Selain itu, pada proses bisnis *to-be* pencairan uang makan, uang lembur, serta tunjangan prestasi juga laporan rekapitulasi sama-sama dibuat berdasarkan data absensi yang tersimpan didalam *database*, kemudian hasil perhitungan keuangan akan disimpan ke dalam *database*. SPP dibuat sesuai dengan data laporan keuangan yang ada pada *database*.
3. Perjalanan dinas dapat dilakukan apabila ada Surat Perintah Tugas (SPT) dan Surat Perintah Perjalanan Dinas (SPPD) yang memerintahkan pegawai bersangkutan untuk melakukan perjalanan dinas. Dengan adanya Sistem Informasi Keuangan Pegawai pada Biro Organisasi, SPT dan SPPD akan dibuat melalui SIKAP. Pada skenario *use case* memasukkan data perjalanan dinas dan mengubah data perjalanan dinas, terdapat *alternative flow* A1 (menangani kegagalan penyimpanan data). Pada *alternatif flow* A1 apabila terdapat pegawai yang telah tercatat melakukan perjalanan dinas pada tanggal yang sama, maka sistem tidak akan memasukkan data pegawai bersangkutan ke *database* dan memberikan pemberitahuan kepada pengguna bahwa data pegawai tersebut ditolak. Setelah dilakukan pengujian *validation testing* pada *alternatif flow* A1 *use case* memasukkan data perjalanan dinas hasil yang diperoleh adalah valid yang berarti sistem informasi keuangan pegawai dapat menolak pegawai yang tercatat melakukan perjalanan dinas pada tanggal yang sama.

4. Analisis persyaratan menghasilkan 6 fitur, 39 persyaratan fungsional, serta satu persyaratan nonfungsional. Berdasarkan persyaratan fungsional diperoleh 33 *use case*.
5. Setelah persyaratan sistem berhasil diidentifikasi, dilakukan perancangan sistem yang menghasilkan *use case scenario* untuk masing-masing *use case*, *activity diagram*, *sequence diagram*, rancangan algoritme dalam bentuk *pseudocode*, dan rancangan antarmuka pengguna.
6. Setelah dilakukan pengujian validasi (*validation testing*) terhadap beberapa fungsi utama seperti memasukkan data perjalanan dinas, mengecek SPJ, membuat laporan rekapitulasi, memasukkan data sidang dan membuat kwitansi sidang, diperoleh hasil pengujian valid dan sesuai dengan spesifikasi atau kebutuhan pengguna. Pengujian kompatibilitas menunjukkan hasil bahwa Sistem Informasi Keuangan Pegawai tidak mengalami masalah yang serius ketika dijalankan pada *browser* yang berbeda. Sehingga SIKAP sesuai untuk digunakan oleh pegawai biro organisasi yang kebanyakan menggunakan beberapa *browser* berbeda seperti google chrome, mozilla firefox, dan opera. Hasil pengujian usability juga mendapatkan nilai *System Usability Scale* 85.625 yang berarti mendapatkan *grade* B dan dinilai mudah untuk digunakan. Sehingga pegawai biro organisasi akan dapat menggunakan SIKAP dengan mudah.

8.2 Saran

Saran yang dapat diberikan sebagai bahan pertimbangan untuk melakukan pengembangan lebih lanjut pada Sistem Informasi Keuangan Pegawai Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur diantaranya adalah:

1. Perlu dilakukan pengembangan lebih lanjut terhadap Sistem Informasi Keuangan Pegawai yang dapat terintegrasi secara langsung dengan *fingerprint*, sehingga tidak perlu melakukan *export* dan *import* data presensi sebagai bahan perhitungan keuangan pegawai.
2. Perlu dilakukan pengembangan lebih lanjut terhadap Sistem Informasi Keuangan Pegawai yang dapat terintegrasi secara langsung dengan *fingerprint* secara *online*, sehingga dapat menampilkan daftar hadir pegawai secara *real time* untuk menghindari terjadinya lupa absen yang akan berpengaruh terhadap perhitungan keuangan pegawai.

DAFTAR PUSTAKA

- Kroll, P. dan Kruchten, P., 2003. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. [e-book] Boston: Addison Wesley. Tersedia di: <<http://karenbell.blogdetik.com>> [Diakses 12 Januari 2018]
- Elmasri, R. dan Navathe, P. B., 2011. *Fundamentals of Database System 6th ed.* [e-book] Boston: Addison Wesley. Tersedia di: <<http://iips.icci.edu.iq/images/exam/databases-ramaz.pdf>> [Diakses 12 Januari 2018]
- S. R. A. & Shalahuddin, M., 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. 2nd ed. Bandung: Informatika.
- Booch, G., Jacobson, I. dan Rumbaugh, J., 1999. *The Unified Modeling Language User Guide 2nd ed.* [e-book] Boston: Addison Wesley. Tersedia di: <https://www.researchgate.net/publication/234785986_Unified_Modeling_Language_User_Guide_The_2nd_Edition_Addison-Wesley_Object_Technology_Series> [Diakses 12 Januari 2018]
- Nixon, R., 2015. *Learning PHP, MySQL & JavaScript With jQuery, CSS & HTML5 4th ed.* [e-book] Sebastopol: O'Reilly Media. Tersedia di: <<http://file.allitebooks.com/20160417/Learning%20PHP,%20MySQL%20&%20JavaScript,%204th%20Edition.pdf>> [Diakses 13 Januari 2018]
- Dennis, A., Wixom, B. H. dan Tegarden, D., 2009. *System Analysis Design UML Version 2.0 An Object-Oriented Approach 3th ed.* [e-book] Hoboken: John Wiley & Sons. Tersedia di: <<http://www.freebookspot.es>> [Diakses 13 Januari 2018]
- Spiller, A., Linz, T. dan Schaefer, H., 2014. *Software Testing Foundations A Study Guide for the Certified Tester Exam 4th ed.* [e-book] Santa Barbara: Rocky Nook. Tersedia di: <http://prof.mau.ac.ir/images/Uploaded_files/Software%20Testing%20Foundations%20A%20Study%20Guide%20for%20the%20Certified%20Tester%20Exam%5B5309302%5D.PDF> [Diakses 17 Januari 2018]
- Tegarden, D., 2011. *Business Process Model and Notation (BPMN) Version 2.0*. [e-book] Object Management Group. Tersedia di: <http://www.oatsolutions.com.br/artigos/SpecBPMN_v2.pdf> [Diakses 20 Januari 2018]
- British Columbia Institute of Technology, 2014. *Welcome to CodeIgniter*. [online] Tersedia di: <https://codeigniter.com/user_guide/general/welcome.html> [Diakses 21 Januari 2018]
- Oracle Corporation, 2018. *What is MySQL?*. [online] Tersedia di: <<https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>> [Diakses 21 Januari 2018]

- IBM, 2018. *The Sequence Diagram*. [online] Tersedia di: <<https://www.ibm.com/developerworks/rational/library/3101.html>> [Diakses 21 Januari 2018]
- Sparxsystem, 2018. *UML 2 Tutorial – Sequence Diagram*. [online] Tersedia di: <http://www.sparxsystems.com/resources/uml2_tutorial/uml2_sequencediagram.html> [Diakses 21 Januari 2018]
- The PHP Group, 2018. *What is PHP?*. [online] Tersedia di: <<http://php.net/manual/en/intro-what-is.php>> [Diakses 21 Januari 2018]
- Bittner, K. & Spence, I. 2002. *Use Case Modeling*. Pearson Education, Inc., Boston.
- Booch, G. 1994. *Object-Oriented Analysis and Design With Applications*. 2nd ed. Addison Wesley Longman, Inc., California.
- Pressman, R. 2010. *Software Engineering: A Practitioner's Approach*. 7th ed. McGraw-Hill Companies, Inc., New York.
- Bangor, A. Kortum, P. & Miller, J. 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale, [e-journal], 114-123, Tersedia melalui Journal of Usability Studies <http://uxpajournal.org/wp-content/uploads/sites/8/pdf/JUS_Bangor_May2009.pdf> [Diakses 3 Maret 2018]
- Brooke, J. 2013. SUS: A Retrospective, [e-journal], 114-123, Tersedia melalui Journal of Usability Studies <http://uxpajournal.org/wp-content/uploads/sites/8/pdf/JUS_Brooke_February_2013.pdf> [Diakses 8 Maret 2018]
- Astuti, N. M. M. P. & Dharmadiaksa, I. B. 2014. Pengaruh Efektifitas Penerapan Sistem Informasi Akuntansi, Pemanfaatan dan Kesesuaian tugas pada Kinerja Karyawan, [e-journal], 373-384. Tersedia melalui *E-Jurnal Akuntansi Universitas Udayana* <<https://ojs.unud.ac.id/index.php/Akuntansi/article/download/9148/7783>> [Diakses 28 Maret 2018]
- Object Management Group, 2011. Business Process Model and Notation (BPMN). [Online] Tersedia di: <<http://www.omg.org/spec/BPMN/2.0/PDF>> [Diakses 20 Maret 2018]
- PowerMapper, 2016. *powermapper*. [Online] Tersedia di: <http://www.powermapper.com/products/sortsite/> [Diakses 20 Februari 2017].
- Aisyah, N. Pantjawati, S. & Tjandrarini, A. B. 2016. Rancang Bangun Sistem Informasi Penggajian Pegawai UD Sehati, [e-journal], 5(8), 373-384. Tersedia melalui <<https://jurnal.stikom.edu/index.php/jsika/article/viewFile/1150/773>> [diakses 28 Maret 2018]

LAMPIRAN A HASIL WAWANCARA

A.1 Hasil Wawancara Bendahara Biro

Berikut hasil wawancara dengan bendahara umum Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur

Tanggal : 29 Januari 2018
Tempat : Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur
Narasumber : Edi Hartono, SE
Jabatan : Bendahara Pengeluaran Pembantu Biro Organisasi

Pertanyaan:

Apakah tugas yang dilakukan oleh bendahara pada Biro Organisasi Provinsi Jawa Timur?

Jawaban:

Salah satu tugas yang dilakukan oleh bendahara pada biro organisasi yaitu membuat laporan rekapitulasi uang makan, uang lembur, tunjangan daerah, serta tunjangan prestasi yang dilakukan oleh bendahara bagian. Bendahara biro memiliki tugas terkait pembayaran uang perjalanan dinas serta uang sidang dan mengajukan SPP (Surat Permintaan Pembayaran) terkait uang lembur, uang makan, tunjangan daerah serta tunjangan prestasi.

Pertanyaan:

Apakah terdapat kendala yang dihadapi dalam melaksanakan tugas tersebut?

Jawaban:

Pembuatan laporan rekapitulasi uang makan, uang lembur, tunjangan daerah dan tunjangan prestasi dilakukan menggunakan bantuan *microsoft excel* yang ditentukan berdasarkan kehadiran pegawai kecuali untuk tunjangan daerah. Pada akhir bulan, masing-masing bendahara bagian akan menggunakan rekapitulasi presensi pegawai yang diperoleh dari *database fingerprint* sebagai bahan untuk membuat laporan rekapitulasi tersebut. Setelah laporan rekapitulasi dibuat, bendahara akan melakukan verifikasi kepada masing-masing pegawai bagian terkait dengan presensi kosong. Apabila terjadi kesalahan atau ketidaksesuaian, bendahara bagian akan merubah laporan rekapitulasi tersebut. Hal tersebut membutuhkan waktu, sehingga akan menyebabkan keterlambatan proses pencairan. Selain itu, bagi pegawai yang melakukan perjalanan dinas dan tidak melakukan presensi menyebabkan presensi pegawai bersangkutan kosong pada tanggal tersebut. Pegawai yang sedang melakukan perjalanan dinas tidak akan mendapatkan potongan tunjangan prestasi. Akan tetapi, karena kemungkinan pegawai tersebut lupa bahwa telah melakukan perjalanan dinas pada hari tersebut, maka presensi akan tetap kosong tanpa keterangan dan pegawai bersangkutan akan dikenakan potongan tunjangan prestasi. Pegawai

tidak boleh melakukan perjalanan dinas lebih dari satu kali pada tanggal yang sama. Ada kemungkinan pegawai ditugaskan melakukan perjalanan dinas lebih dari satu kali pada tanggal yang sama karena tidak ada pencatatan perjalanan dinas. Kesalahan penulisan baik nama, nip atau jabatan dapat terjadi pada dokumen-dokumen keuangan atau Surat Perintah Tugas yang disebabkan kesalahan pada saat mengetik data pegawai atau melakukan *copy paste* data pegawai ke dokumen tertentu.

Pertanyann :

Bagaimana proses pencairan uang makan?

Jawaban :

Bendahara bagian akan membuat laporan rekapitulasi uang makan masing-masing pegawai bagian pada akhir bulan sesuai dengan kehadiran pegawai bersangkutan. Kemudian dilakukan verifikasi terhadap masing-masing pegawai terkait kehadiran pegawai. Setelah seluruh bagian menyerahkan laporannya, bendahara biro akan melakukan pengecekan dan membuat SPP (Surat Permintaan Pembayaran) dan mengajukannya ke kepala biro. Setelah kepala biro menyetujui permintaan pembayaran, bendahara biro akan membuat surat perintah pembayaran yang kemudian akan ditandatangani oleh kepala biro dan diserahkan ke BPKAD dengan melampirkan laporan rekapitulasi uang makan dari seluruh bagian. Setelah BPKAD menerima surat perintah pembayaran beserta lampiran laporan rekapitulasi uang makan, BPKAD akan mentransfer uang sesuai rekapitulasi uang makan ke rekening biro. Bendahara biro akan menunggu hingga uang makan masuk ke rekening biro, kemudian bendahara biro akan membuat surat perintah pencairan dana uang makan ke masing-masing pegawai. Setelah surat kuasa ditandatangani oleh kepala biro, bendahara akan menyerahkan surat kuasa ke bank agar bank dapat melakukan transfer uang makan ke masing-masing pegawai. Proses tersebut juga berlaku untuk pencairan uang lembur, tunjangan daerah dan tunjangan prestasi. Tunjangan daerah tidak memperhitungkan kehadiran pegawai, uang lembur disesuaikan dengan presensi dan surat perintah lembur.

Pertanyaan :

Bagaimana proses pencairan uang sidang?

Jawaban :

bendahara menerima bukti dilaksanakannya sidang (seperti berita acara dan presensi sidang). Kemudian bendahara membuat rekapitulasi pembayaran uang sidang sesuai daftar presensi dan berita acara sidang. Setelah membuat rekapitulasi pembayaran uang sidang, bendahara mengajukan pencairan uang sidang kepada kepala bagian, kemudian mencairkan uang sidang setelah disetujui oleh kepala bagian.

Pertanyaan :

Bagaimana proses pencairan uang perjalanan dinas?

Jawaban :

Perjalanan dinas harus didasari dengan SPT (Surat Perintah Tugas). Berdasarkan SPT, bendahara akan membuat SPPD dan pegawai dapat melakukan perjalanan dinas dengan membawa SPPD. Setelah pegawai melaksanakan perjalanan dinas, pegawai yang bertugas akan membuat SPJ perjalanan dinas dan menyerahkannya ke bendahara. Bendahara melakukan pengecekan terhadap SPJ dan membuat kwitansi serta membagikan kekurangan uang perjalanan dinas kepada pegawai

Mengetahui,
 Penata Usaha Keuangan
 Biro Organisasi

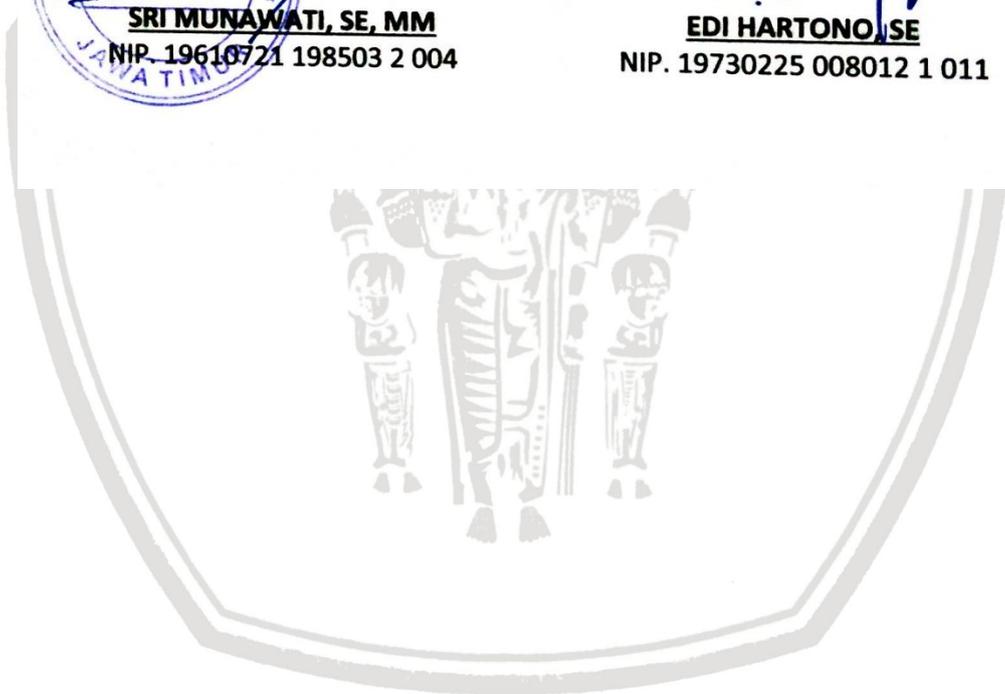


SRI MUNAWATI, SE, MM
 NIP. 19610721 198503 2 004

Bendahara Pengeluaran Pembantu
 Biro Organisasi



EDI HARTONO, SE
 NIP. 19730225 008012 1 011



LAMPIRAN B FORM PENGUJIAN USABILITAS

B.1 Pernyataan Usabilitas Oleh Bendahara Biro

FORM PERTANYAAN SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI KEUANGAN PEGAWAI

(Studi Kasus Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur)

Nama : EDI. H
 Jabatan : BPP Biro
 Tanggal Penilaian : 12 SEPT 2018

Kuisisioner ini digunakan untuk mengetahui seberapa mudah antarmuka pengguna dapat digunakan. Beri tanda centang (✓) pada salah satu kolom jawaban pada setiap pertanyaan dibawah. Nilai 1 berarti sangat tidak setuju hingga nilai 5 yang berarti sangat setuju.

NO	PERTANYAAN	JAWABAN (sangat tidak setuju - sangat setuju)				
		1	2	3	4	5
1	Saya akan sering menggunakan Sistem Informasi Keuangan Pegawai (SIKP)					✓
2	Sistem Informasi Keuangan Pegawai (SIKP) terlalu kompleks	✓				
3	Sistem Informasi Keuangan Pegawai (SIKP) mudah digunakan				✓	
4	Saya membutuhkan bantuan orang yang mengerti untuk menggunakan Sistem Informasi Keuangan Pegawai (SIKP)			✓		
5	Fungsi-fungsi pada Sistem Informasi Keuangan Pegawai (SIKP) sudah terintegrasi dengan baik			✓		
6	Tertalu banyak ketidakkonsistenan dalam Sistem Informasi Keuangan Pegawai (SIKP)	✓				
7	Saya rasa ada banyak orang yang dapat belajar menggunakan Sistem Informasi Keuangan Pegawai (SIKP) dengan cepat					✓
8	Sistem Informasi Keuangan Pegawai (SIKP) sulit untuk digunakan	✓				
9	Saya merasa percaya diri saat menggunakan Sistem Informasi Keuangan Pegawai (SIKP)					✓
10	Saya harus belajar banyak hal sebelum dapat menggunakan Sistem Informasi Keuangan Pegawai (SIKP)	✓				

Responden


 EDI. H

B.2 Pernyataan Usabilitas Oleh Bendahara Bagian 1

FORM PERTANYAAN SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI KEUANGAN PEGAWAI

(Studi Kasus Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur)

Nama : *Ramli*
 Jabatan : *ADm. Keuangan*
 Tanggal Penilaian : *12-9-2018*

Kuisisioner ini digunakan untuk mengetahui seberapa mudah antarmuka pengguna dapat digunakan. Beri tanda centang (✓) pada salah satu kolom jawaban pada setiap pertanyaan dibawah. Nilai 1 berarti sangat tidak setuju hingga nilai 5 yang berarti sangat setuju.

NO	PERTANYAAN	JAWABAN (sangat tidak setuju - sangat setuju)				
		1	2	3	4	5
1	Saya akan sering menggunakan Sistem Informasi Keuangan Pegawai (SIKP)					✓
2	Sistem Informasi Keuangan Pegawai (SIKP) terlalu kompleks		✓			
3	Sistem Informasi Keuangan Pegawai (SIKP) mudah digunakan					✓
4	Saya membutuhkan bantuan orang yang mengerti untuk menggunakan Sistem Informasi Keuangan Pegawai (SIKP)				✓	
5	Fungsi-fungsi pada Sistem Informasi Keuangan Pegawai (SIKP) sudah terintegrasi dengan baik				✓	
6	Tertalu banyak ketidakkonsistenan dalam Sistem Informasi Keuangan Pegawai (SIKP)		✓			
7	Saya rasa ada banyak orang yang dapat belajar menggunakan Sistem Informasi Keuangan Pegawai (SIKP) dengan cepat				✓	
8	Sistem Informasi Keuangan Pegawai (SIKP) sulit untuk digunakan		✓			
9	Saya merasa percaya diri saat menggunakan Sistem Informasi Keuangan Pegawai (SIKP)				✓	
10	Saya harus belajar banyak hal sebelum dapat menggunakan Sistem Informasi Keuangan Pegawai (SIKP)				✓	

Responden

Ramli
Ramli.....

B.3 Pernyataan Usabilitas Oleh Bendahara Bagian 2

**FORM PERTANYAAN SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI
KEUANGAN PEGAWAI**
(Studi Kasus Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur)

Nama : **HAPPY POERBODJATI**
 Jabatan : **PENATA LAP. KEUANGAN**
 Tanggal Penilaian : **12-9-2018**

Kuisisioner ini digunakan untuk mengetahui seberapa mudah antarmuka pengguna dapat digunakan. Beri tanda centang (✓) pada salah satu kolom jawaban pada setiap pertanyaan dibawah. Nilai 1 berarti sangat tidak setuju hingga nilai 5 yang berarti sangat setuju.

NO	PERTANYAAN	JAWABAN (sangat tidak setuju - sangat setuju)				
		1	2	3	4	5
1	Saya akan sering menggunakan Sistem Informasi Keuangan Pegawai (SIKP)					✓
2	Sistem Informasi Keuangan Pegawai (SIKP) terlalu kompleks	✓				
3	Sistem Informasi Keuangan Pegawai (SIKP) mudah digunakan					✓
4	Saya membutuhkan bantuan orang yang mengerti untuk menggunakan Sistem Informasi Keuangan Pegawai (SIKP)			✓		
5	Fungsi-fungsi pada Sistem Informasi Keuangan Pegawai (SIKP) sudah terintegrasi dengan baik				✓	
6	Tertalu banyak ketidakkonsistenan dalam Sistem Informasi Keuangan Pegawai (SIKP)	✓				
7	Saya rasa ada banyak orang yang dapat belajar menggunakan Sistem Informasi Keuangan Pegawai (SIKP) dengan cepat					✓
8	Sistem Informasi Keuangan Pegawai (SIKP) sulit untuk digunakan	✓				
9	Saya merasa percaya diri saat menggunakan Sistem Informasi Keuangan Pegawai (SIKP)					✓
10	Saya harus belajar banyak hal sebelum dapat menggunakan Sistem Informasi Keuangan Pegawai (SIKP)	✓				

Responden


 HAPPY.....

B.4 Pernyataan Usabilitas Oleh Pegawai Biro

FORM PERTANYAAN SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI KEUANGAN PEGAWAI

(Studi Kasus Biro Organisasi Sekretariat Daerah Provinsi Jawa Timur)

Nama : MOH. AZIZ AL BASID
 Jabatan : Pengelola Kepegawaian
 Tanggal Penilaian : 12 September 2018

Kuisiner ini digunakan untuk mengetahui seberapa mudah antarmuka pengguna dapat digunakan. Beri tanda centang (✓) pada salah satu kolom jawaban pada setiap pertanyaan dibawah. Nilai 1 berarti sangat tidak setuju hingga nilai 5 yang berarti sangat setuju.

NO	PERTANYAAN	JAWABAN (sangat tidak setuju - sangat setuju)				
		1	2	3	4	5
1	Saya akan sering menggunakan Sistem Informasi Keuangan Pegawai (SIKP)				✓	
2	Sistem Informasi Keuangan Pegawai (SIKP) terlalu kompleks	✓				
3	Sistem Informasi Keuangan Pegawai (SIKP) mudah digunakan					✓
4	Saya membutuhkan bantuan orang yang mengerti untuk menggunakan Sistem Informasi Keuangan Pegawai (SIKP)			✓		
5	Fungsi-fungsi pada Sistem Informasi Keuangan Pegawai (SIKP) sudah terintegrasi dengan baik			✓	✓	
6	Tertalu banyak ketidakkonsistenan dalam Sistem Informasi Keuangan Pegawai (SIKP)	✓				
7	Saya rasa ada banyak orang yang dapat belajar menggunakan Sistem Informasi Keuangan Pegawai (SIKP) dengan cepat					✓
8	Sistem Informasi Keuangan Pegawai (SIKP) sulit untuk digunakan	✓				
9	Saya merasa percaya diri saat menggunakan Sistem Informasi Keuangan Pegawai (SIKP)					✓
10	Saya harus belajar banyak hal sebelum dapat menggunakan Sistem Informasi Keuangan Pegawai (SIKP)	✓				

Responden


 MOH. AZIZ AL BASID
 NIP. 19900303 2010 1 001