

**PENGEMBANGAN SISTEM INFORMASI *MONITORING*  
PEKERJAAN KANTOR KONSULTAN PAJAK BERBASIS WEB  
PADA KANTOR KONSULTAN PAJAK I GEDE ARIANTA**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Diago Ariesandika  
NIM: 145150401111022



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

# PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI *MONITORING* PEKERJAAN  
KANTOR KONSULTAN PAJAK BERBASIS WEB PADA  
KANTOR KONSULTAN PAJAK I GEDE ARIANTA

## SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Diago Ariesandika  
NIM: 145150401111022

Skripsi ini telah diuji dan dinyatakan lulus pada  
25 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

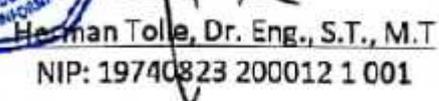
  
Satrio Agung Widaksono, S.Kom., M.Kom.  
NIP: 19860521 201212 1 001



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

Mengetahui  
Ketua Jurusan Sistem Informasi



  
Herman Tolle, Dr. Eng., S.T., M.T  
NIP: 19740823 200012 1 001

## IDENTITAS TIM PENGUJI

### Ketua Majelis (Penguji I)

Aryo Pinandito, S.T, M.MT



### Pembimbing I

Satrio Agung Wicaksono,  
S.Kom, M.Kom



### Anggota Majelis (Penguji II)

Rizal Setya Perdana, S.Kom, M.Kom



### Pembimbing II

Fajar Pradana, S.ST, M.Eng



## 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 unsurunsur 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).



## DAFTAR RIWAYAT HIDUP



### Data Pribadi

Nama : Diago Ariesandika  
Tempat, Tanggal Lahir : Probolinggo, 23 Maret 1996  
Agama : Islam  
Alamat Rumah : Perum Triwung Kidul Indah RT/RW 007/001 Kelurahan Triwung Kidul Kecamatan Kademangan Kota Probolinggo  
Nomor Telephone : 085235544433  
Email : diagoariesandika1@gmail.com

### Riwayat Pendidikan

2014 sampai dengan 2018 : Sistem Informasi Universitas Brawijaya Malang  
2011 sampai dengan 2014 : SMAN 2 Probolinggo  
2008 sampai dengan 2011 : SMPN 2 Probolinggo  
2002 sampai dengan 2008 : SDN Kademangan 1 Probolinggo

### Pengalaman

- Staff Ahli di Badan Perwakilan Mahasiswa Sistem Informasi Universitas Brawijaya Periode 2016/2017
- Ketua Divisi Perlengkapan Acara Pemilihan Wakil Mahasiswa Sistem Informasi Universitas Brawijaya 2016/2017
- Assistant of Database Programming 2016/2017 Fakultas Ilmu Komputer Universitas Brawijaya
- Assistant of Database System 2016/2017 Fakultas Ilmu Komputer Universitas Brawijaya
- Assistant of Analysis and Design of Information System 2016/2017 Fakultas Ilmu Komputer Universitas Brawijaya
- Tim dalam mengerjakan Proyek Sistem Monitoring Pembenahan Jaringan di TELKOM INDONESIA DIVRE V Surabaya tahun 2017



## KATA PENGANTAR

Dengan menyebut nama Allah SWT Yang Maha Pengasih dan Penyayang karena berkat Rahmat, hidayah dan Karunia-Nya kepada penulis sehingga dapat menyelesaikan penyusunan naskah skripsi ini dengan lancar. Shalawat serta salam semoga senantiasa terlimpah kepada junjungan Nabi Besar Muhammad SAW, kepada keluarga, serta para sahabatnya.

Penulisan naskah skripsi ini diajukan untuk memenuhi salah satu syarat untuk memperoleh gelar sarjana. Judul yang penulis ajukan sebagai skripsi adalah "PENGEMBANGAN SISTEM INFORMASI *MONITORING* PEKERJAAN KANTOR KONSULTAN PAJAK BERBASIS WEB PADA KANTOR KONSULTAN PAJAK I GEDE ARIANTA".

Dalam penyusunan dan penulisan naskah skripsi ini penulis mendapatkan banyak bantuan, bimbingan serta dukungan dari berbagai pihak baik secara moril maupun materil. Dalam kesempatan ini penulis ingin mengucapkan banyak terima kasih kepada yang terhormat

1. Bapak Satrio Agung Wicaksono, S.Kom., M.Kom. selaku Dosen pembimbing pertama yang telah membimbing dan memberikan banyak solusi dalam pengerjaan skripsi.
2. Bapak Fajar Pradana, S.ST., M.Eng. selaku Dosen pembimbing kedua yang telah membimbing dan memberikan banyak masukan selama pengerjaan skripsi.
3. Bapak Aditya Rachmadi, S.ST., M.TI, selaku pembimbing akademik yang telah meluangkan waktunya untuk memberikan bimbingan perihal akademik dari penulis selama berjalannya perkuliahan, sehingga penulis mampu menjalani perkuliahan sesuai dengan prosedur yang telah ditetapkan oleh akademik FILKOM.
4. Bapak Dr. Eng. Herman Tolle, S.T, M.T selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D selaku sekretaris Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Bapak Suprpto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya Malang.
7. Keluarga besar saya, mama Tatik Hartini, eyang Busini, mbak Arista Marhance, mas Sugeng, adik Mamad, om Totok, tante Rina yang telah memberi segenap dukungan dan sangat memotivasi saya sampai saat ini.
8. Segenap pimpinan dan staf kantor konsultan pajak I Gede Arianta yang telah mengizinkan serta membantu penulis untuk melakukan penelitian ini.
9. Sahabat-sahabat indra, savira, eki, azizah, lili, dony, danniar, candra, bayu, dimas, tedja, wira, audi, windi, edwin, firnanda, lutfhan, ista, ali, jawara,

fatih, munggarhanna, gladi, siti, moon, sekar, faishal, andi, winda, via, icha, anet, danu, yolanda, bima, idin, ghozi dan teman-teman lain seperjuangan yang tidak bisa penulis sebutkan satu-persatu yang selalu memberikan semangat disaat kesulitan, dan yang telah menemani saya dalam mengurus segala kepentingan skripsi.

10. Keluarga besar kos 42-B yang meramaikan suasana kos disaat siang maupun malam.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada proses penelitian ini. Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan, diskusi mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya.

Malang, 11 Juli 2018

Diago Ariesandika

diagoariesandika1@gmail.com



## ABSTRAK

Kantor konsultan pajak I Gede Arianta adalah sebuah kantor yang memberikan layanan jasa professional akuntan dan konsultan pajak dengan standar professional yang berlaku. Saat ini pencatatan kegiatan konsultasi antara staf dan klien tidak ada sehingga bisa berdampak pada permasalahan klien yang tidak kunjung selesai. Pengelolaan data klien yang tidak tersedia secara terpusat, hal ini dikarenakan data klien masih berupa lampiran kertas yang disimpan oleh staf dan berdampak untuk kehilangan berkas lampiran kertas. Pelaporan perpajakan bulanan atau perpajakan tahunan tidak tercatat, tidak tervalidasi dan sulit dipantau oleh pimpinan sehingga berdampak perpajakan klien tidak bisa terbayarkan dan dilaporkan. Pengelolaan tanda terima dokumen yang belum ada dan berdampak atas penanggung jawab kehilangan atau kesalahpahaman terhadap dokumen yang diberikan.

Berdasarkan masalah-masalah yang diuraikan diatas, maka dapat diberikan solusi untuk mengembangkan sistem informasi monitoring pekerjaan kantor konsultan pajak I Gede Arianta. Dalam pengerjaan sistem informasi menggunakan model pengembangan perangkat lunak *Waterfall*. Sistem yang dikembangkan memiliki 19 kebutuhan fungsional untuk mendukung proses bisnis pada kantor. Pada pengujian sistem informasi dilakukan dengan *black box testing* dan *user acceptance testing*. *Black box testing* terdapat 19 uji fungsionalitas sistem yang menghasilkan 100% valid. Kemudian pada *user acceptance testing* untuk mengetahui sistem diterima oleh pengguna akhir (pimpinan dan staf). Pimpinan menghasilkan nilai penerimaan sebesar 96,67% dan staf menghasilkan nilai penerimaan sebesar 87,5% yang artinya pengguna akhir sangat menerima sistem informasi yang telah dibuat.

Kata Kunci : sistem informasi, sistem informasi *monitoring*, *Waterfall model*, *monitoring* pekerjaan.

## ABSTRACT

*Tax Consultant Office I Gede Arianta is an office that gives professional accountant and tax consultant services with applicable professional standards. Currently, the recording of consultation activities between staff and clients does not exist so that it can impact on the client's problems are not completed. Client data management that is not centrally available, because client data is still a paper attachment stored by staff and has the effect of losing paper attachment files. Reporting of monthly taxation or annual taxation is unrecorded, unvalidated and difficult to monitor by the boss so as to impact the unpaid and reported taxation of clients. Management of receipt of documents that have not existed and impact on the person in charge of loss or misunderstanding of the documents provided.*

*Based on the problems described above, given a solution to develop information systems monitoring the work of tax consultant I Gede Arianta. In the process of information systems using the Waterfall model software development. The developed system has 19 functional requirements to support business processes at the office. In testing the information system is done with black-box testing and user acceptance testing. In black-box testing, there is 19 test system functionality that yields 100% valid. User acceptance testing used to know the system is accepted by the end users (leadership and staff). Leaders generate an acceptance value of 96.67% and staff generate an acceptance value of 87.5% which means the end users is very receptive to the information system that has been made.*

*Keywords: information system, information system monitoring, waterfall model, monitoring work.*

## DAFTAR ISI

PERSETUJUAN .....	ii
PERNYATAAN ORISINALITAS .....	iv
KATA PENGANTAR.....	vi
ABSTRAK.....	viii
<i>ABSTRACT</i> .....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xvii
DAFTAR LAMPIRAN .....	xix
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat .....	4
1.5 Batasan masalah .....	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Kajian Pustaka .....	6
2.2 Kantor Konsultan Pajak I Gede Arianta .....	7
2.3 <i>Monitoring</i> Pekerjaan .....	8
2.4 Sistem Informasi .....	8
2.5 Proses Bisnis .....	8
2.5.1 <i>Flow Object</i> .....	9
2.5.2 <i>Connections</i> .....	11
2.5.3 <i>Swimlanes</i> .....	11
2.5.4 <i>Artifacts</i> .....	11
2.6 Pengembangan Perangkat Lunak dengan Model <i>Waterfall</i> .....	12
2.6.1 <i>Communication</i> .....	12
2.6.2 <i>Planning</i> .....	13
2.6.3 <i>Modelling</i> .....	14



2.6.4 Construction .....	20
2.6.5 Deployment.....	25
BAB 3 METODOLOGI .....	26
3.1 Studi Literatur .....	27
3.2 Pengumpulan Data .....	27
3.3 Analisis Kebutuhan Sistem.....	27
3.4 Perancangan Sistem .....	27
3.5 Implementasi Sistem .....	28
3.6 Pengujian .....	28
3.7 Kesimpulan dan Saran .....	28
BAB 4 ANALISIS KEBUTUHAN .....	29
4.1 Proses Rekayasa Kebutuhan .....	29
4.1.1 Identifikasi Proses Bisnis AS-IS .....	29
4.1.2 Elisitasi Kebutuhan .....	37
4.1.3 Identifikasi Aktor .....	38
4.1.4 Pemodelan Proses Bisnis TO-BE.....	38
4.1.5 Analisis Kebutuhan Fungsional .....	50
4.2 Use Case Diagram .....	60
4.3 Use Case Scenario .....	62
4.3.1 Use Case Scenario Login .....	62
4.3.2 Use Case Scenario Mengelola Data Klien.....	63
4.3.3 Use Case Scenario Mengelola Staf .....	65
4.3.4 Use Case Scenario Mengelola Monitoring Aktivitas Klien .....	66
4.3.5 Use Case Scenario Mengelola Monitoring Bulanan Klien .....	69
4.3.6 Use Case Scenario Request Validasi Monitoring Bulanan Klien..	71
4.3.7 Use Case Scenario Memvalidasi Monitoring Bulanan Klien.....	72
4.3.8 Use Case Scenario Menambah Tanda Terima Dokumen .....	73
4.3.9 Use Case Scenario Mengelola Tanda Terima Dokumen .....	75
4.3.10 Use Case Scenario Mengelola Monitoring Tahunan Klien .....	76
4.3.11 Use Case Scenario Request Validasi Monitoring Tahunan Klien	78
4.3.12 Use Case Scenario Memvalidasi Monitoring Tahunan Klien.....	79
4.3.13 Use Case Scenario Mengelola Proposal .....	80

4.3.14 Use Case Scenario Request Persetujuan Proposal .....	81
4.3.15 Use Case Scenario Menyetujui Proposal.....	82
4.3.16 Use Case Scenario Logout .....	83
4.3.17 Use Case Scenario Melihat Grafik Aktivitas .....	84
4.3.18 Use Case Scenario Melihat Grafik <i>Monitoring</i> Bulanan .....	85
4.3.19 Use Case Scenario Melihat Grafik <i>Monitoring</i> Tahunan .....	86
BAB 5 PERANCANGAN.....	87
5.1 Perancangan Arsitektur Sistem .....	87
5.2 Perancangan <i>Sequence Diagram</i> .....	87
5.2.1 <i>Sequence Diagram</i> Kelola Data Klien .....	87
5.2.2 <i>Sequence Diagram</i> Melihat Grafik Aktivitas .....	89
5.2.3 <i>Sequence Diagram</i> Kelola <i>Monitoring</i> Aktivitas.....	89
5.2.4 <i>Sequence Diagram</i> Menambah <i>Monitoring</i> Bulanan .....	91
5.2.5 <i>Sequence Diagram</i> Request <i>Monitoring</i> Bulanan .....	91
5.2.6 <i>Sequence Diagram</i> Validasi Request <i>Monitoring</i> Bulanan.....	92
5.2.7 <i>Sequence Diagram</i> Menambah Tanda Terima Dokumen .....	93
5.3 Perancangan <i>Class Diagram</i> .....	94
5.3.1 Perancangan Class Diagram Staf .....	94
5.3.2 Perancangan <i>Class Diagram</i> Pimpinan .....	95
5.4 Perancangan Skema Basis Data .....	97
5.5 Perancangan Antarmuka Aplikasi .....	104
5.5.1 Rancangan Antarmuka Login Pengguna .....	104
5.5.2 Rancangan Antarmuka <i>Dashboard</i> Pimpinan .....	105
5.5.3 Rancangan Antarmuka <i>Dashboard</i> Staf .....	106
5.5.4 Rancangan Antarmuka Kelola Klien .....	107
5.5.5 Rancangan Antarmuka Kelola <i>Monitoring</i> Aktivitas .....	107
5.5.6 Rancangan Antarmuka Kelola <i>Monitoring</i> Bulanan.....	108
5.5.7 Rancangan Antarmuka Tambah Tanda Terima Dokumen .....	109
5.5.8 Rancangan Antarmuka Kelola Tanda Terima Dokumen .....	110
BAB 6 IMPLEMENTASI .....	111
6.1 Batasan Implementasi .....	111
6.2 Implementasi Kode Program .....	111

6.2.1 Implementasi Kelola Data Klien .....	111
6.2.2 Implementasi Kelola <i>Monitoring</i> Aktivitas.....	115
6.2.3 Implementasi Menambah <i>Monitoring</i> Bulanan.....	118
6.2.4 Implementasi <i>Request Monitoring</i> Bulanan .....	121
6.2.5 Implementasi Validasi <i>Request Monitoring</i> Bulanan .....	124
6.2.6 Implementasi Menambah Tanda Terima Dokumen .....	126
6.2.7 Implementasi Melihat Grafik Aktivitas .....	128
6.2.8 Implementasi Data Definition Language (DDL).....	129
6.3 Implementasi Antarmuka .....	134
6.3.1 Implementasi Antarmuka Login .....	134
6.3.2 Implementasi Antarmuka Dashboard Pimpinan.....	134
6.3.3 Implementasi Antarmuka Dashboard Staf.....	135
6.3.4 Implementasi Antarmuka Kelola Klien.....	135
6.3.5 Implementasi Antarmuka Kelola <i>Monitoring</i> Aktivitas .....	136
6.3.6 Implementasi Antarmuka <i>Monitoring</i> Bulanan .....	136
6.3.7 Implementasi Antarmuka Tambah Tanda Terima Dokumen....	137
6.3.8 Implementasi Antarmuka Kelola Tanda Terima Dokumen .....	137
BAB 7 PENGUJIAN DAN ANALISIS HASIL .....	138
7.1 Pengujian .....	138
7.1.1 Black-Box Testing .....	138
7.1.2 <i>User Acceptance Testing</i> .....	152
7.1.3 Pengujian Waktu Sebelum Dan Sesudah Hasil Implementasi ..	162
7.2 Analisis Hasil .....	163
BAB 8 KESIMPULAN DAN SARAN .....	165
8.1 Kesimpulan .....	165
8.2 Saran .....	166
DAFTAR PUSTAKA.....	167
LAMPIRAN A HASIL WAWANCARA PIMPINAN.....	169
LAMPIRAN B HASIL WAWANCARA STAF .....	172
LAMPIRAN C HASIL <i>USER ACCEPTANCE TESTING</i> .....	174



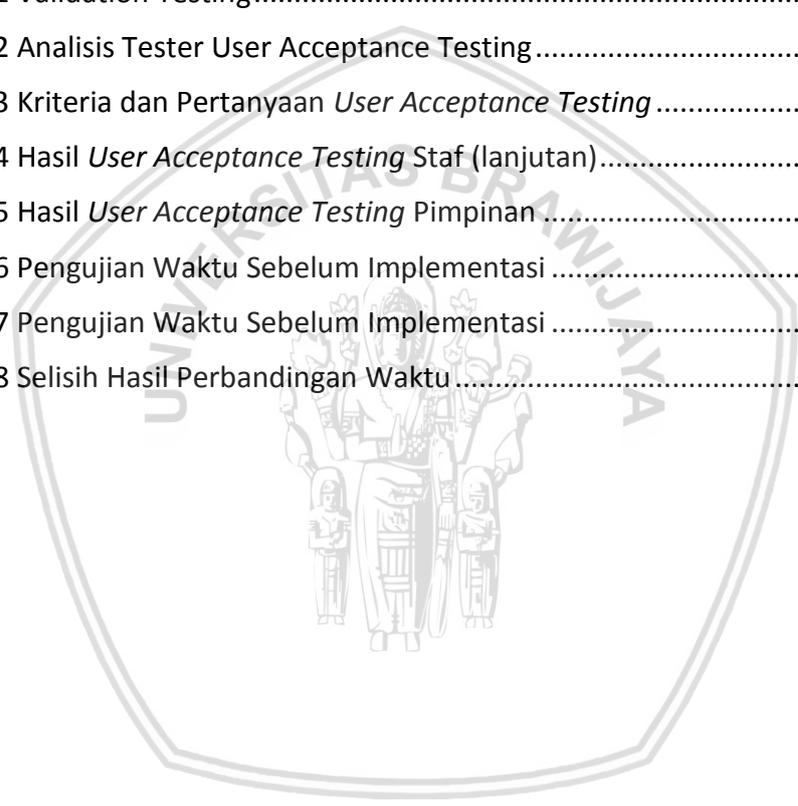
## DAFTAR TABEL

Tabel 2.1 Simbol-Simbol <i>Event</i> BPMN .....	9
Tabel 2.2 Simbol-Simbol <i>Activity</i> BPMN .....	10
Tabel 2.3 Simbol-Simbol <i>Gateway</i> BPMN .....	10
Tabel 2.4 Simbol-Simbol <i>Connections</i> BPMN .....	11
Tabel 2.5 Simbol-Simbol <i>Swimlanes</i> BPMN .....	11
Tabel 2.6 Simbol-Simbol <i>Artifacts</i> BPMN .....	12
Tabel 2.7 Analisis Permasalahan .....	14
Tabel 2.8 Simbol-Simbol Use Case Diagram .....	15
Tabel 2.9 Simbol-Simbol <i>Sequence Diagram</i> .....	18
Tabel 2.10 Simbol-Simbol <i>Class Diagram</i> .....	20
Tabel 2.11 Bobot Nilai Jawaban .....	24
Tabel 2.12 Persentase Nilai Jawaban .....	24
Tabel 4.1 Analisis Permasalahan Dalam Menangani <i>Monitoring</i> .....	36
Tabel 4.2 Temuan Kebutuhan .....	37
Tabel 4.3 Identifikasi Aktor .....	38
Tabel 4.4 Daftar Proses Bisnis <i>To-Be</i> .....	39
Tabel 4.5 Daftar Aktivitas Proses Bisnis PB-KKP-1 .....	46
Tabel 4.6 Daftar Aktivitas Proses Bisnis PB-KKP-2 .....	46
Tabel 4.7 Daftar Aktivitas Proses Bisnis PB-KKP-3 .....	47
Tabel 4.8 Daftar Aktivitas Proses Bisnis PB-KKP-4 .....	48
Tabel 4.9 Kebutuhan Fungsional Dengan Aktor Pengguna .....	50
Tabel 4.10 Kebutuhan Fungsional Dengan Aktor Staf .....	57
Tabel 4.11 Kebutuhan Fungsional Dengan Aktor Pimpinan .....	58
Tabel 4.12 <i>Use Case Scenario Login</i> .....	62
Tabel 4.13 <i>Use Case Scenario</i> Mengelola Data Klien .....	63
Tabel 4.14 <i>Use Case Scenario</i> Mengelola Staf .....	65
Tabel 4.15 <i>Use Case Scenario</i> Mengelola <i>Monitoring</i> Aktivitas Klien .....	66
Tabel 4.16 <i>Use Case Scenario</i> Mengelola <i>Monitoring</i> Bulanan Klien .....	69
Tabel 4.17 <i>Use Case Scenario Request</i> Validasi <i>Monitoring</i> Bulanan Klien .....	72
Tabel 4.18 <i>Use Case Scenario</i> Validasi <i>Monitoring</i> Bulanan Klien .....	72
Tabel 4.19 <i>Use Case Scenario</i> Menambah Tanda Terima Dokumen .....	73



Tabel 4.20 <i>Use Case Scenario</i> Mengelola Tanda Terima Dokumen.....	75
Tabel 4.21 <i>Use Case Scenario</i> Mengelola <i>Monitoring</i> Tahunan Klien .....	76
Tabel 4.22 <i>Use Case Scenario Request</i> Validasi <i>Monitoring</i> Tahunan Klien .....	78
Tabel 4.23 <i>Use Case Scenario</i> Memvalidasi <i>Monitoring</i> Tahunan Klien .....	79
Tabel 4.24 <i>Use Case Scenario</i> Mengelola Proposal .....	80
Tabel 4.25 <i>Use Case Scenario Request</i> Persetujuan Proposal .....	82
Tabel 4.26 <i>Use Case Scenario</i> Menyetujui Proposal .....	82
Tabel 4.27 <i>Use Case Scenario Logout</i> .....	83
Tabel 4.28 <i>Use Case Scenario</i> Melihat Grafik Aktivitas.....	84
Tabel 4.29 <i>Use Case Scenario</i> Melihat Grafik <i>Monitoring</i> Bulanan .....	85
Tabel 4.30 <i>Use Case Scenario</i> Melihat Grafik <i>Monitoring</i> Tahunan .....	86
Tabel 5.1 Struktur Tabel Pengguna .....	98
Tabel 5.2 Struktur Tabel Klien.....	99
Tabel 5.3 Struktur Tabel <i>Monitoring</i> Aktivitas.....	99
Tabel 5.4 Struktur Tabel Data Dokumen Klien.....	100
Tabel 5.5 Struktur Tabel Tipe Proker .....	100
Tabel 5.6 Struktur Tabel Proker.....	101
Tabel 5.7 Struktur Tabel Status.....	101
Tabel 5.8 Struktur Tabel Proposal.....	101
Tabel 5.9 Struktur Tabel <i>Monitoring</i> Proker Bulanan.....	102
Tabel 5.10 Struktur Tabel <i>Monitoring</i> Proker Tahunan.....	103
Tabel 5.11 Struktur Tabel Status.....	104
Tabel 6.1 Implementasi Kelola Data Klien .....	111
Tabel 6.2 Implementasi Kelola <i>Monitoring</i> Aktivitas.....	115
Tabel 6.3 Implementasi Menambah <i>Monitoring</i> Bulanan.....	118
Tabel 6.4 Implementasi <i>Request Monitoring</i> Bulanan.....	121
Tabel 6.5 Implementasi Validasi <i>Request Monitoring</i> Bulanan .....	124
Tabel 6.6 Implementasi Menambah Tanda Terima Dokumen .....	127
Tabel 6.7 Implementasi Melihat Grafik Aktivitas.....	128
Tabel 6.8 Implementasi DDL Tabel Bulan .....	129
Tabel 6.9 Implementasi DDL Tipe Proker.....	129
Tabel 6.10 Implementasi DDL Proker.....	129

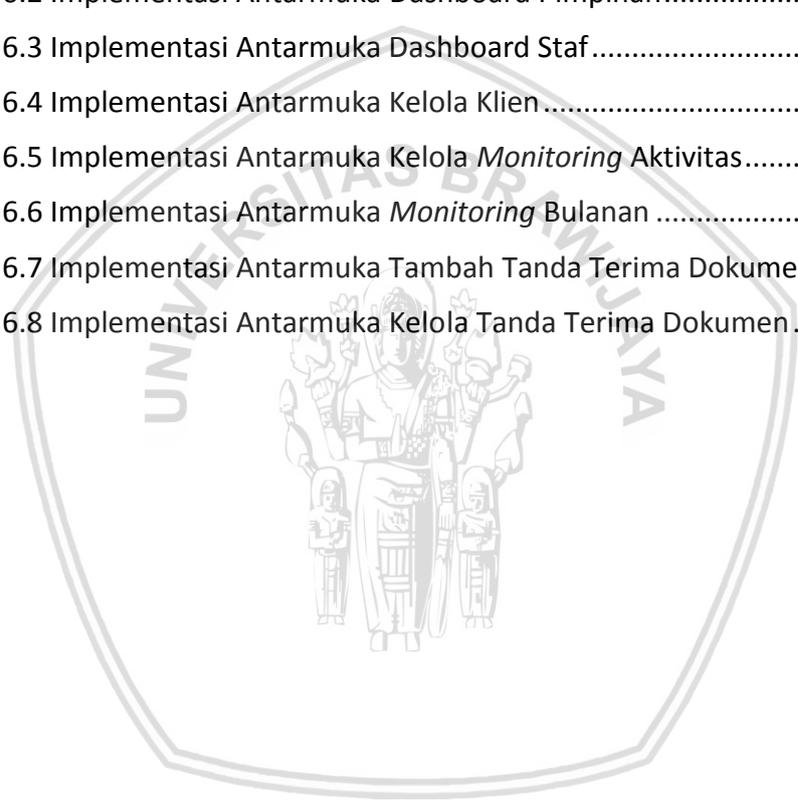
Tabel 6.11 Implementasi DDL Status .....	130
Tabel 6.12 Implementasi DDL Tabel Pengguna.....	130
Tabel 6.13 Implementasi DDL Tabel Klien.....	130
Tabel 6.14 Implementasi DDL Tabel <i>Monitoring</i> Aktivitas.....	131
Tabel 6.15 Implementasi DDL Tabel <i>Monitoring</i> Proker Bulanan.....	131
Tabel 6.16 Implementasi DDL Tabel <i>Monitoring</i> Proker Tahunan.....	132
Tabel 6.17 Implementasi DDL Tabel Proposal .....	133
Tabel 6.18 Implementasi DDL Tabel Data Dokumen Klien .....	133
Tabel 7.1 Validation Testing.....	139
Tabel 7.2 Analisis Tester User Acceptance Testing.....	152
Tabel 7.3 Kriteria dan Pertanyaan <i>User Acceptance Testing</i> .....	153
Tabel 7.4 Hasil <i>User Acceptance Testing</i> Staf (lanjutan).....	155
Tabel 7.5 Hasil <i>User Acceptance Testing</i> Pimpinan .....	159
Tabel 7.6 Pengujian Waktu Sebelum Implementasi .....	162
Tabel 7.7 Pengujian Waktu Sebelum Implementasi .....	163
Tabel 7.8 Selisih Hasil Perbandingan Waktu.....	164



## DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi.....	7
Gambar 2.2 <i>Waterfall Model</i> .....	12
Gambar 2.3 Contoh <i>Use Case Diagram</i> .....	15
Gambar 2.4 Contoh <i>Sequence Diagram</i> .....	17
Gambar 2.5 Contoh <i>Class Diagram</i> .....	19
Gambar 2.6 Hubungan <i>Class Diagram</i> .....	19
Gambar 2.7 Alur kerja framework CodeIgniter .....	21
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	26
Gambar 4.1 Proses Bisnis Registrasi Klien Saat Ini.....	30
Gambar 4.2 Proses Bisnis Konsultasi Saat Ini.....	31
Gambar 4.3 Proses Bisnis <i>Monitoring</i> Pajak Bulanan Saat Ini .....	32
Gambar 4.4 Proses Bisnis <i>Monitoring</i> Pajak Tahunan Saat Ini .....	34
Gambar 4.5 Proses Bisnis Usulan Registrasi Klien .....	39
Gambar 4.6 Proses Bisnis Usulan Konsultasi .....	40
Gambar 4.7 Proses Bisnis Usulan <i>Monitoring</i> Pajak Bulanan .....	42
Gambar 4.8 Proses Bisnis Usulan <i>Monitoring</i> Pajak Tahunan .....	44
Gambar 4.9 Use Case Diagram <i>Monitoring</i> Pekerjaan Kantor konsultan pajak ...	61
Gambar 5.1 Arsitektur Diagram Sistem Informasi <i>Monitoring</i> Pekerjaan.....	87
Gambar 5.2 <i>Class Diagram</i> Kelola Data Klien .....	88
Gambar 5.3 <i>Sequence Diagram</i> Melihat Grafik Aktivitas .....	89
Gambar 5.4 <i>Sequence Diagram</i> Kelola <i>Monitoring</i> Aktivitas.....	90
Gambar 5.5 <i>Sequence Diagram</i> Menambah <i>Monitoring</i> Bulanan.....	91
Gambar 5.6 <i>Sequence Diagram</i> Request <i>Monitoring</i> Bulanan .....	92
Gambar 5.7 <i>Sequence Diagram</i> Validasi Request <i>Monitoring</i> Bulanan.....	93
Gambar 5.8 <i>Sequence Diagram</i> Menambah Tanda Terima Dokumen .....	93
Gambar 5.9 <i>Class Diagram</i> Controller Staf .....	94
Gambar 5.10 <i>Class Diagram</i> Model Staf .....	95
Gambar 5.11 <i>Class Diagram</i> Controller Pimpinan .....	96
Gambar 5.12 <i>Class Diagram</i> Model Pimpinan .....	97
Gambar 5.13 Perancangan Skema Basis Data .....	98
Gambar 5.14 Rancangan Antarmuka Login Pengguna .....	105

Gambar 5.15 Rancangan Antarmuka <i>Dashboard</i> Pimpinan .....	106
Gambar 5.16 Rancangan Antarmuka <i>Dashboard</i> Staf .....	106
Gambar 5.17 Rancangan Antarmuka Kelola Klien .....	107
Gambar 5.18 Rancangan Antarmuka Kelola <i>Monitoring</i> Aktivitas .....	108
Gambar 5.19 Rancangan Antarmuka Kelola <i>Monitoring</i> Bulanan .....	109
Gambar 5.20 Rancangan Antarmuka Tambah Tanda Terima Dokumen .....	109
Gambar 5.21 Rancangan Antarmuka Kelola Tanda Terima .....	110
Gambar 6.1 Implementasi Antarmuka Login .....	134
Gambar 6.2 Implementasi Antarmuka Dashboard Pimpinan.....	134
Gambar 6.3 Implementasi Antarmuka Dashboard Staf.....	135
Gambar 6.4 Implementasi Antarmuka Kelola Klien.....	135
Gambar 6.5 Implementasi Antarmuka Kelola <i>Monitoring</i> Aktivitas.....	136
Gambar 6.6 Implementasi Antarmuka <i>Monitoring</i> Bulanan .....	136
Gambar 6.7 Implementasi Antarmuka Tambah Tanda Terima Dokumen.....	137
Gambar 6.8 Implementasi Antarmuka Kelola Tanda Terima Dokumen.....	137



## DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA PIMPINAN.....	169
LAMPIRAN B HASIL WAWANCARA STAF.....	172
LAMPIRAN C HASIL <i>USER ACCEPTANCE TESTING</i> .....	174
C.1 Responden 1 .....	174
C.2 Responden 2 .....	176



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Kebutuhan informasi dalam arus perkembangan teknologi yang terus menerus meningkat menjadi hal yang sangat penting. Adanya teknologi yang terus berkembang, maka penyebaran data dan informasi sangatlah mudah dan cepat. Banyak kantor, perusahaan besar ataupun kecil dalam mendukung kegiatan atau aktivitas di kantor atau perusahaannya menggunakan teknologi internet. Internet merupakan sistem terbuka yang tersedia bagi semua orang, dan perangkat lunak yang digunakan untuk membuat internet beroperasi terdapat banyak sekali program komputer yang kompleks sehingga bisa saling bekerjasama dan berkomunikasi dengan sangat baik (Comer, 2007). Teknologi internet mempunyai keuntungan dalam hal penyampaian informasi yang sangat cepat, bisa diakses dimana saja, kapan saja, oleh siapa saja.

Sistem informasi diartikan secara teknis sebagai seperangkat komponen yang saling terkait dalam pengumpulan (pengambilan), memproses, menyimpan, dan mendistribusikan suatu informasi agar mendukung dalam pengambilan keputusan dan pengendalian di dalam suatu organisasi (Laudon, 2014). Sehingga bisa dibidang bahwa sistem informasi merupakan suatu sistem yang bisa mengolah data dengan baik mulai dari yang sederhana hingga yang kompleks dan dijadikan sebuah informasi agar bisa dimanfaatkan dalam pengambilan keputusan. Dengan adanya fungsi sistem informasi yang sangat berpengaruh dalam kegiatan yang ada di organisasi, kantor atau perusahaan membuat kinerja ada menjadi jauh lebih efektif dan efisien. Dalam pengembangan sistem informasi terdapat banyak jenisnya seperti sistem informasi transaksi penjualan, sistem informasi akademik, sistem informasi pergudangan, sistem informasi kepegawaian, sistem informasi *monitoring* dan sebagainya.

Kantor konsultan pajak I Gede Arianta merupakan salah satu kantor di kota Malang yang memberikan layanan jasa professional Konsultan Pajak dan Akuntan dengan kompetensi tinggi, obyektif, itegritas dan sesuai dengan standar professional yang berlaku. Kantor ini sudah banyak memberikan jasa konsultasi perhitungan, penyetoran dan pelaporan berbagai macam jenis perpajakan. Dalam menangani klien, kantor ini melakukan sebuah aktivitas dimana satu klien dilayani oleh satu staf yang menjadi sebagai pengarah dan pemberi nasihat atas kasus-kasus yang sedang dihadapi.

Masalah timbul ketika klien berkonsultasi dengan staf beberapa kali dan pencatatan aktivitas yang masih belum ada, sehingga aktivitas yang dilakukan saat melakukan konsultasi tidak terekam oleh staf yang bersangkutan dan juga sangat sulit ketika pimpinan ingin memantau aktivitas apa yang sudah dilakukan oleh staf-stafnya. Masalah ini bisa berdampak buruk pada permasalahan klien yang tidak kunjung selesai dikarenakan staf tidak mampu melakukan pekerjaannya sesuai dengan masalah yang dihadapi klien, disisi lain juga mengakibatkan pimpinan tidak bisa mengevaluasi pekerjaan yang dilakukan staf yang berkaitan dengan

permasalahan konsultasi dengan klien yang bisa mengakibatkan citra buruk dari klien. Hal ini dikarenakan aktivitas konsultasi antara staf dan klien masih belum ada standar atau aturan baku yang dilakukan oleh staf dalam melakukan pencatatan aktivitas konsultasi yang nantinya disampaikan atau dilaporkan ke pada pimpinan. Selain itu data klien juga tidak tersedia secara terpusat, hal ini dikarenakan data klien masih berupa lampiran kertas yang disimpan oleh staf, sehingga mengakibatkan ketika pimpinan ingin melihat data klien harus menghubungi atau bertemu dengan staf yang melayani klien tersebut. Pencarian dokumen terkait data klien juga masih sulit dan membutuhkan waktu yang lama yaitu bisa lebih dari 1 jam. Pada masalah penyimpanan data klien juga berdampak negatif ketika berkas data klien hilang dan kantor tidak bisa mengetahui daftar data klien yang ada.

Selain masalah pencatatan dan pelaporan aktivitas konsultasi harian yang dilakukan oleh staf dan klien juga terdapat masalah terkait pelaporan pekerjaan yang dilakukan oleh staf tentang perpajakan bulanan atau perpajakan tahunan klien. Pelaporan perpajakan bulanan atau perpajakan tahunan klien tidak tercatat dan tervalidasi dengan baik karena dalam proses pelaporannya hanya melalui penyampaian lisan, padahal pimpinan seharusnya mengetahui dan melakukan *monitoring* dan validasi terkait setiap pekerjaan perpajakan bulanan dan tahunan klien yang dikerjakan oleh setiap staf. Sehingga saat ini pimpinan kesulitan dalam memantau ataupun meminta laporan progress *monitoring* pekerjaan masing-masing staf secara langsung. Pada masalah ini mengakibatkan dampak buruk bagi kantor dan klien ketika staf tidak melakukan pekerjaannya dalam perpajakan bulanan atau perpajakan tahunan klien dengan benar, sehingga perpajakan klien tidak bisa terbayarkan dan tidak ada laporan perpajakan klien yang dilampirkan kepada Kantor Pelayanan Pajak (KPP).

Dalam proses perpajakan bulanan atau perpajakan tahunan klien, staf membutuhkan beberapa dokumen yang akan digunakan sebagai bahan perpajakan bulanan atau perpajakan tahunan klien. Ketika klien menyerahkan dokumen tidak ada tanda terima atau bukti serah terima yang diserahkan kepada klien, masalah ini timbul karena masih belum dibuat format standar dalam pembuatan tanda terima dokumen yang diserahkan kepada klien. Pada setiap kegiatan serah terima dokumen belum terekam dengan baik, hal ini dikarenakan belum ada aturan baku yang dibuat oleh kantor konsultan pajak dalam proses serah terima dokumen klien. Pimpinanpun sulit memantau terkait dokumen apa saja yang sudah ada dan diberikan oleh klien kepada staf. Pada permasalahan tidak adanya standar tanda terima, memiliki dampak buruk ketika terjadi kecurangan atau kehilangan dokumen atau penanggung jawab atas dokumen yang diberikan ataupun adanya kesalahpahaman yang dilakukan oleh kedua pihak.

Berdasarkan masalah yang sudah diuraikan, maka untuk mempermudah dalam pengolahan data yang dilakukan oleh Kantor konsultan pajak I Gede Arianta dibutuhkan sebuah sistem informasi untuk pengolahan data agar bisa menyimpan semua data pekerjaan yang dilakukan oleh setiap staf dengan berbasis web yang terpusat dan dapat dioperasikan secara online. Hal ini dilakukan dengan cara

mengembangkan sistem informasi *monitoring* pekerjaan konsultasi pajak agar lebih meningkatkan kinerja Kantor konsultan pajak I Gede Arianta dengan menggunakan metode pengembangan *Waterfall Model* agar semua kebutuhan kantor dapat didefinisikan dengan baik. *Waterfall Model* termasuk dalam mengembangkan sebuah perangkat lunak model lama yang sifatnya berurutan atau sistematis. Dalam mengembangkan sebuah perangkat lunak terdapat 5 tahapan, dimulai dengan tahapan *Communication, planning, Modeling, Construction, Deployment* (Pressman, 2010). Dalam mengembangkan perangkat lunak dengan menggunakan *Waterfall Model* dilalui dengan tahap demi tahap berjalan berurutan dan setiap tahap harus menunggu tahap sebelumnya selesai. Agar bisa menghasilkan perangkat lunak yang baik dengan memakai *Waterfall Model* maka harus melewati 5 tahapan tersebut dan secara garis besar diawali dengan menganalisis kebutuhan pengguna, lalu melakukan perancangan sistem berdasarkan hasil dari analisis kebutuhan pengguna kemudian bisa dilakukan implementasi pengembangan dengan kode program sesuai dengan hasil dari perancangan sistem yang sudah terdefiniskan dengan baik. Setelah perangkat lunak dibuat maka dilakukan tahap pengujian untuk mengetahui kelayakan pada perangkat lunak yang telah dibuat. Setelah bisa dianggap layak dalam tahap pengujian maka tahap akhir adalah pengimplementasian perangkat lunak atau sistem informasi yang sudah dibuat kepada klien agar bisa membantu kinerja kantor konsultan pajak I Gede Arianta.

Dari permasalahan yang sudah dijelaskan sebelumnya, disarankan agar membangun sistem informasi yang akan dilakukan pada skripsi ini dengan judul **“PENGEMBANGAN SISTEM INFORMASI *MONITORING* PEKERJAAN KANTOR KONSULTAN PAJAK BERBASIS WEB PADA KANTOR KONSULTAN PAJAK I GEDE ARIANTA”**. Diharapkan dalam hasil penelitian ini bisa membantu kantor agar lebih memaksimalkan kinerja pekerjaan kantor.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, dapat dirumuskan permasalahan dalam penelitian ini adalah sebagai berikut :

1. Bagaimana hasil analisis proses bisnis dan kebutuhan sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta?
2. Bagaimana hasil perancangan dan hasil implementasi sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta?
3. Bagaimana hasil pengujian sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta?

## 1.3 Tujuan

Dari rumusan masalah yang sudah dijelaskan diatas terdapat tujuan yang ingin dicapai dalam penelitian sebagai berikut :

1. Melakukan analisis proses bisnis dan kebutuhan sistem sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta.

2. Membuat perancangan sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta dan membuat sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta sesuai dengan perancangan yang sudah dibuat.
3. Menguji hasil sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta dengan *black-box* dan *user acceptance testing*.

#### 1.4 Manfaat

Manfaat yang diperoleh dari hasil penelitian ini diharapkan dapat bermanfaat bagi beberapa pihak. Adapun manfaat yang diperoleh sebagai berikut:

1. Bagi Kantor konsultan pajak I Gede Arianta  
Memiliki sistem informasi *monitoring* pekerjaan agar meningkatkan kinerja kantor.
2. Bagi Penulis  
Menambah pengetahuan dalam menganalisis kebutuhan pengguna, merancang, membangun serta menguji sistem informasi berbasis *web*.

#### 1.5 Batasan masalah

Ruang lingkup dalam batasan masalah pada penelitian ini dibuat agar penelitian yang dilakukan oleh penulis tidak meluas, hal-hal yang terkait batasan masalah sebagai berikut :

1. Penelitian dilakukan di kantor konsultan pajak I Gede Arianta dan data yang diperoleh merupakan hasil dari wawancara langsung oleh pihak kantor konsultan pajak I Gede Arianta.
2. Pembuatan sistem yang dilakukan dalam penelitian ini mencakup perancangan, implementasi dan pengujian dengan memakai pendekatan *Software Defelopment Life Cycle* model *Waterfall*.
3. Pada tahap pengembangan perangkat lunak sistem informasi sampai pada tahap *deployment* (tahap *delivery*)

#### 1.6 Sistematika pembahasan

Sistematika yang disusun dalam penulisan skripsi ini adalah sebagai berikut :

##### **BAB I      Pendahuluan**

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

##### **BAB II     Landasan Kepustakaan**

Menjelaskan tentang teori dasar dan teori pendukung dalam pengembangan sistem informasi *monitoring* pekerjaan berbasis

*web* dan menjelaskan secara singkat perihal profil dari kantor terkait.

### **BAB III Metode Penelitian**

Berisi tentang pemaparan metode dan langkah-langkah yang akan dilakukan dalam penelitian.

### **BAB IV Analisis Kebutuhan**

Membahas terkait analisis kebutuhan sistem informasi *monitoring* pekerjaan kantor konsultan pajak berbasis *web* yang akan menjadi dasar dalam perancangan sistem.

### **BAB V Perancangan**

Membahas tentang perancangan sistem untuk sistem informasi *monitoring* pekerjaan berbasis *web* pada kantor konsultan pajak I Gede Arianta yang akan dibangun.

### **BAB VI Implementasi**

Membahas tentang implementasi sistem informasi *monitoring* pekerjaan berbasis *web* pada kantor konsultan pajak I Gede Arianta sesuai dengan perancangan sistem yang sudah dibuat.

### **BAB VII Pengujian**

Membahas terkait proses dan hasil dari pengujian sistem informasi *monitoring* pekerjaan berbasis *web* pada kantor konsultan pajak I Gede Arianta.

### **BAB VIII Penutup**

Memuat kesimpulan dan saran dari hasil penelitian yang dilakukan.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Penelitian yang dijadikan sebagai kajian pustaka dalam penelitian dan penulisan skripsi ini adalah penelitian yang sudah dilakukan oleh (Castillote, 2017) dengan judul '*Development of E-Toll Collection and Monitoring System*'. Peneliiian tersebut dilakukan penelitian mengenai proses pembayaran biaya tol pada terminal bus baru kota Tacloban memakan waktu yang lama. Castillote mengusulkan untuk menerapkan sistem baru untuk mengotomatisasi proses yang ada, usulan berupa *electronic toll collection* menggunakan RFID yang digunakan untuk mengumpulkan biaya tol secara otomatis. Sebuah kendaraan nantinya akan memegang chip RFID. Chip ini tidak lain berupa nomor identifikasi unik yang diberikan. Sistem yang nanti dibangun akan mengurangi waktu yang dihabiskan untuk membayar biaya tol di terminal bus baru kota Tacloban. Penelitian ini memakai kuisisioner dan wawancara dari 200 responden, 95,5% setuju untuk menerapkan sistem yang diusulkan di Terminal Bus Baru dalam perbaikan proses pembayaran tol. Hasil dari sistem yang dibangun ini nantinya juga bisa melakukan pengawasan (*monitoring*) terhadap kegiatan toll dan semua data kegiatan terekam ke dalam sistem, serta sistem ini juga membuat lebih efisien bagi pengemudi untuk membayar biaya tol mereka.

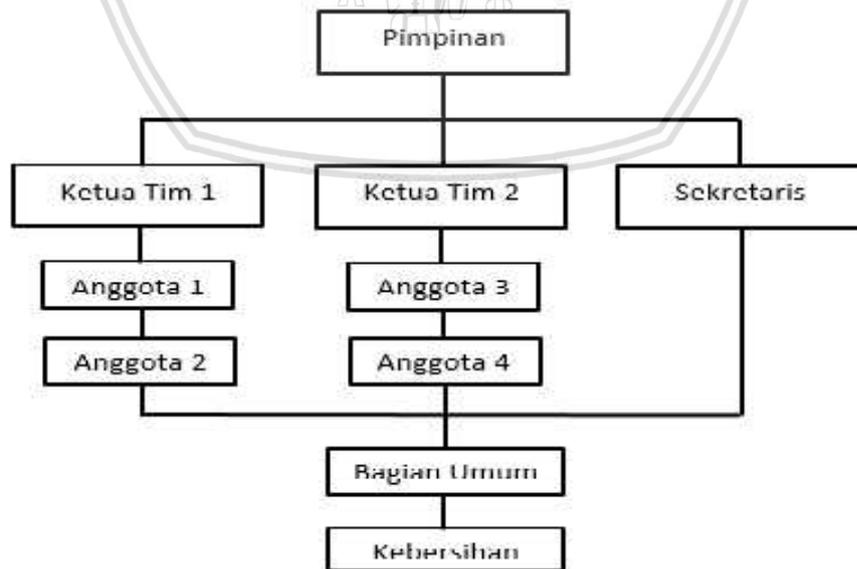
Penelitian lain yang dilakukan oleh (Rahni, 2015) yang berjudul '*DEVELOPMENT OF THE ONLINE STUDENT ATTENDANCE MONITORING SYSTEM (SAMST<sup>TM</sup>) BASED ON QR-CODES AND MOBILE DEVICES*'. Penelitian ini tentang sistem pemantauan (bernama SAMST<sup>TM</sup>) yang digunakan untuk menguraikan perkembangan kehadiran siswa berdasarkan kode QR dan perangkat seluler. Desain ini dipilih karena kesederhanaan dan efektivitas biaya, satu-satunya peralatan yang dibutuhkan oleh pengguna (siswa dan dosen) adalah perangkat internet seluler seperti komputer tablet dan *smartphone*. Banyak studi penelitian yang sudah menetapkan terkait kehadiran siswa selama kuliah dan kegiatan pembelajaran lainnya seperti lab, seminar dan lokakarya berkorelasi positif dengan kinerja siswa selama mereka studi sarjana. Beberapa penelitian bahkan telah menemukan peracikan tambahan faktor seperti bahwa efek negatif dari ketidakhadiran mempengaruhi kinerjan siswa yang membuatnya lebih rendah. Dengan demikian sangat penting bahwa kehadiran siswa perlu dipantau dan upaya untuk memperbaiki kehadiran siswa. Sistem saat ini dalam fase *prototyping* awal untuk menunjukkan bukti kerja dari konsepnya. Keuntungan utama dari sistem ini adalah metode pencatatan dan pemantauan kehadiran siswa yang lebih akurat dan cepat. Dengan sistem ini, secara kuantitatif akan lebih mudah untuk membedakan siswa berdasarkan ketekunan mereka dalam menghadiri kelas, dan dengan demikian juga memprediksi kinerja mereka karena korelasi antara kehadiran dan kinerja akademik.

Pada penelitian yang telah dilakukan oleh Adel Alshamrani dan Abdullah Bahattab yang membandingkan antara tiga model SDLC yang berjudul "A

*Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model*” menjelaskan bahwa kelebihan yang didapatkan dengan menggunakan *Waterfall Model* adalah sebagai berikut mudah dalam memahami dan pengimplementasian, lebih banyak diketahui, lebih sering digunakan, mendefinisikan desain dengan jelas sebelum diimplementasikan, bekerja dengan baik meskipun pada struktur tim yang tidak berpengalaman, meminimalkan overhead perencanaan dan tahapan diproses dan diselesaikan satu per satu (Alshamrani & Bahattab, 2015). *Waterfall Model* digunakan ketika semua persyaratan yang telah ada dipahami saat awal pengembangan perangkat lunak dengan baik, kebutuhannya bersifat stabil dan tidak ada perubahan. Pada penelitian yang dilakukan oleh penulis ini kebutuhan dari pengguna telah terdefinisi dengan baik, jelas dan tetap sehingga penulis menggunakan *Waterfall Model* dalam melakukan pengembangan perangkat lunak dalam melakukan penelitian.

## 2.2 Kantor Konsultan Pajak I Gede Arianta

Kantor konsultan pajak I Gede Arianta merupakan sebuah kantor yang memberikan jasa professional konsultan pajak dengan integritas, obyektif, kompetensi tinggi dan sesuai dengan standar professional. Berdasarkan hasil wawancara pada pemilik kantor yang bernama I Gede Arianta, SH., SE., MAK., BKP., CA., ACPA., Ak. sebagai pimpinan telah memiliki 9 staf yaitu 2 orang ketua tim, ketua tim pertama memiliki 2 orang anggota staf dan ketua tim kedua memiliki 2 orang anggota staf, 1 orang sekretaris, 1 orang bagian umum dan 1 orang kebersihan. Pada struktur organisasi yang akan diselesaikan permasalahan pada penelitian ini adalah pihak pimpinan, ketua tim 1 dan ketua tim 2 beserta anggota. Berikut ini Gambar 2.1 adalah struktur organisasi dari kantor konsultan pajak I Gede Arianta.



**Gambar 2.1 Struktur Organisasi**

Sumber : (Hasil Wawancara, 2018)

## 2.3 Monitoring Pekerjaan

*Monitoring* pekerjaan dapat diartikan sebagai suatu kegiatan mengawasi atau memonitor seluruh pekerjaan yang dilakukan oleh satu orang ataupun banyak orang. Kegiatan *monitoring* dilakukan dengan cara langsung dan tidak langsung, kegiatan *monitoring* langsung merupakan kegiatan yang dilakukan dengan cara meninjau langsung untuk semua aktivitas yang sedang berjalan, sedangkan *monitoring* tidak langsung bisa dilakukan dengan cara melakukan kegiatan dalam penelaahan laporan secara tertulis, bisa juga dengan mencermati laporan lisan ataupun mewawancari satu orang atau banyak orang bagi yang terlibat dalam satu aktivitas (Rohayati, 2014). *Monitoring* pekerjaan bisa memberikan informasi mengenai status pekerjaan yang telah atau sedang dikerjakan dan juga kecenderungan bahwasannya dilakukan evaluasi secara berulang-ulang dari waktu ke waktu, dalam pemantauan biasanya dilakukan dengan tujuan tertentu agar bisa memeriksa proses atau mengevaluasi kondisi kemajuan untuk menuju hasil manajemen yang baik.

## 2.4 Sistem Informasi

Sistem informasi bisa didefinisikan sebagai gabungan dari orang, perangkat lunak, perangkat keras, sumber data, jaringan komunikasi, prosedur dan kebijakan yang digunakan untuk menyimpan, mengambil, mengubah serta menyebarkan suatu informasi kedalam organisasi (O'Brien, 2007).

Sistem informasi didefinisikan juga sebagai kombinasi dari perangkat lunak, perangkat keras dan juga jaringan telekomunikasi yang dibangun lalu dipakai oleh orang untuk pengumpulan, pembuatan dan pendistribusian data yang berguna dalam suatu pengaturan organisasi. Perangkat lunak mengacu pada rangkaian program yang dapat memberitahu komputer dalam melakukan tugas tertentu. Perangkat lunak mengacu pada alat-alat pada komputer fisik, misalnya keyboard dan monitor. Sedangkan jaringan telekomunikasi adalah serangkaian komputer yang terhubung secara bersama-sama dengan alat-alat komunikasi (Valacich, 2010)

Sehingga dari pengertian tersebut bisa disimpulkan bahwa sistem informasi merupakan suatu sistem yang bisa mengolah sebuah *input* data dan menghasilkan output berupa informasi yang bisa digunakan dalam membantu organisasi dalam mencapai tujuannya.

## 2.5 Proses Bisnis

Proses bisnis adalah kumpulan dari serangkaian kegiatan dalam melakukan suatu koordinasi yang berada pada organisasi dan lingkungan teknis. Aktivitas yang ada menunjukkan adanya aliran dari tujuan bisnis. Setiap proses bisnis diterapkan dan diberlakukan pada suatu organisasi, tetapi memungkinkan sebuah organisasi saling berinteraksi dengan proses bisnis yang ada dalam organisasi lainnya (Weske, 2007).

Pemodelan proses bisnis merupakan penggambaran suatu diagram yang mewakili dari urutan aktivitas yang secara implisit fokus pada suatu proses, kegiatan dan tindakan (Ramdhani, 2015). Dalam pemodelan proses bisnis biasanya gabungan dari pekerjaan dan dokumentasi dari satu atau lebih departemen dalam suatu institusi, perusahaan atau organisasi. Pada pemodelan proses bisnis terdapat manfaat untuk mempermudah dalam memahami alur proses secara terintegrasi dan tujuannya untuk mendefinisikan tahapan atau langkah apa yang akan diambil pada suatu organisasi dalam mencapai tujuan.

BPMN (*Business Process Modeling Notation*) digunakan untuk menstandarisasi model proses bisnis dan notasi dalam menghadapi berbagai notasi pemodelan dan sudut pandang yang berbeda. Dengan demikian, BPMN akan menyediakan sarana komunikasi yang sederhana untuk mengkomunikasikan informasi proses kepada pengguna bisnis lainnya, pelaksana proses, pelanggan, dan pemasok. BPMN dibatasi hanya untuk mendukung konsep pemodelan yang berlaku untuk Proses Bisnis. BPMN memiliki empat kategori elemen sebagai penunjang dalam pembuatan BPMN yang dijelaskan sebagai berikut (Object Management Group, 2011) :

**2.5.1 Flow Object**

*Flow Object* dibagi menjadi 3, yaitu *event*, *activity*, dan *gateway*. Berikut penjelasannya :

1. *Event*

*Event* merupakan sesuatu yang terjadi dan memiliki dampak dalam proses bisnis. Suatu *event* dapat berasal dari internal dan eksternal suatu proses. *Event* dibagi menjadi tiga yaitu *start event*, *intermediate event*, dan *end event*. Setiap proses selalu memiliki sebuah *start event* untuk menunjukkan awal dari proses bisnis. Tabel 2.1 merupakan penjelasan dari simbol *event*.

**Tabel 2.1 Simbol-Simbol Event BPMN**

No	Nama	Gambar	Deskripsi
1.1	<i>Start</i>		Mendeskripsikan dimana suatu proses dimulai
1.2	<i>Intermediate</i>		Mendeskripsikan dimana suatu terjadi diantara awal dan akhir proses. Akan mempengaruhi alur dari proses, tapi tidak akan memulai atau memberhentikan proses.
1.3	<i>End</i>		Mendeskripsikan suatu proses berakhir.

Sumber : (Object Management Group, 2011)

## 2. Activity

*Activity* merupakan tugas yang dilakukan dalam sebuah proses bisnis. *Activity* ditunjukkan dengan kotak dengan ujung bulat dengan nama yang menjelaskan aktivitas yang dilakukan. Terdapat dua macam *activity* yaitu *task* dan *sub process*. Tabel 2.2 merupakan penjelasan dari simbol *activity*.

**Tabel 2.2 Simbol-Simbol Activity BPMN**

No	Nama	Gambar	Deskripsi
1.	<i>Task</i>		Merupakan aktivitas yang dilakukan pada alur proses
2.	<i>Sub Process</i>		Merupakan sebuah aktivitas majemuk yang dimasukkan dalam proses. Aktivitas majemuk tersebut dapat dijelaskan dengan lebih detail.

Sumber : (Object Management Group, 2011)

## 3. Gateway

*Gateway* bertanggung jawab mengontrol bagaimana alur dari sebuah proses bisnis. Pada *gateway* dibagi menjadi 4 jenis yaitu *exclusive*, *parallel*, *inclusive*, *complex*. Tabel 2.3 merupakan penjelasan dari simbol *gateway*.

**Tabel 2.3 Simbol-Simbol Gateway BPMN**

No	Nama	Gambar	Deskripsi
1.	<i>Exclusive</i>		Sebagai <i>divergence</i> : digunakan untuk membuat jalur alternatif dalam sebuah proses, tapi hanya satu yang dipilih.  Sebagai <i>convergence</i> : digunakan untuk menggabungkan jalur alternatif.
2.	<i>Parallel</i>		Mendeskripsikan proses yang dijalankan secara bersamaan
3.	<i>Inclusive</i>		Mendeskripsikan sebuah proses yang dipecah menjadi beberapa jalur.
4.	<i>Complex</i>		Mendeksripsikan alur yang kompleks pada sebuah proses bisnis

Sumber : (Object Management Group, 2011)



### 2.5.2 Connections

*Connections* adalah elemen yang menghubungkan *flow objects*. Pada connection dibagi menjadi 3 jenis yaitu *sequence flow*, *association*, *message flow*. Tabel 2.4 merupakan penjelasan dari simbol *connections*.

Tabel 2.4 Simbol-Simbol *Connections* BPMN

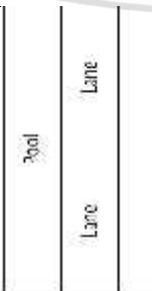
No	Nama	Gambar	Deskripsi
1.1	<i>Sequence Flow</i>		Menunjukkan urutan aktivitas yang dilakukan pada sebuah proses
1.2	<i>Association</i>		Menunjukkan hubungan antara data, teks, artifak lain, dan <i>flow object</i> pada sebuah proses
1.3	<i>Message Flow</i>		Menunjukkan alur pesan antara dua partisipan yang mampu mengirim dan menerima pesan

Sumber : (Object Management Group, 2011)

### 2.5.3 Swimlanes

*Swimlanes* merupakan wadah grafis yang membagi suatu set aktivitas dengan aktivitas lain. Pada swimlane dibagi menjadi 2 jenis yaitu *pool* dan *lane*. Tabel 2.5 merupakan penjelasan simbol *swimlanes*.

Tabel 2.5 Simbol-Simbol *Swimlanes* BPMN

No	Nama	Gambar	Deskripsi
1.1	<i>Pool</i>		Merupakan wadah yang berisi satu proses dan <i>sequence flow</i> yang menghubungkan aktivitas
1.2	<i>Lane</i>		Digunakan untuk mempresentasikan tanggungjawab aktivitas pada sebuah proses.

Sumber : (Object Management Group, 2011)

### 2.5.4 Artifacts

*Artifacts* mempresentasikan sebuah objek diluar sebuah proses. *Artifact* dapat mempresentasikan data atau catatan yang menjelaskan sebuah proses atau



dapat digunakan untuk mengelola tugas atau proses. Tabel 2.6 merupakan penjelasan simbol *artifacts*.

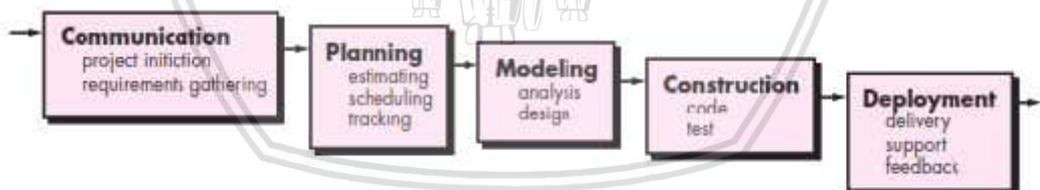
**Tabel 2.6 Simbol-Simbol *Artifacts* BPMN**

No	Nama	Gambar	Deskripsi
1.1	<i>Data Object</i>		Mempresentasikan informasi yang mengalir pada sebuah proses seperti dokumen bisnis, surat, <i>email</i> dan lain-lain.
1.2	<i>Data Store</i>		Tempat dimana proses dapat membaca atau menulis data
1.3	<i>Annotation</i>		Menunjukkan informasi tambahan kepada pembaca sebuah diagram BPMN.
1.4	<i>Group</i>		Memungkinkan untuk mengelompokkan elemen secara informal

Sumber : (Object Management Group, 2011)

## 2.6 Pengembangan Perangkat Lunak dengan Model *Waterfall*

*Waterfall* merupakan suatu model klasik atau lama yang sifatnya sistematis atau berurutan dalam melakukan pengembangan perangkat lunak (Pressman, 2010). Fase-fase pada *Waterfall Model* ditunjukkan pada Gambar 2.2 sebagai berikut :



**Gambar 2.2 *Waterfall Model***

Sumber : (Pressman, 2010)

### 2.6.1 *Communication*

Pada langkah ini dilakukan sebelum memulai aktivitas yang bersifat teknis, hal ini diperlukan untuk mendukung komunikasi dengan klien agar saling memahami dan tujuannya terpenuhi. Hasil dari komunikasi bisa berupa inisialisasi proyek seperti menganalisis masalah yang ada dan juga komunikasi ini bisa berupa pengumpulan data-data yang akan digunakan dalam mendukung dalam pendefinisian kebutuhan ataupun fitur yang ingin dikembangkan pada pengembangan perangkat lunak (Pressman, 2010).

Pada proses pengumpulan data dilakukan komunikasi secara lisan berupa wawancara kepada pihak kantor. Wawancara merupakan bentuk komunikasi yang sudah diakui sebagai tahnik pengumpulan data yang penting dan banyak dilakukan dalam pengembangan sistem informasi. Wawancara memungkinkan analisis sistem sebagai pewawancara untuk mengumpulkan data secara tatap muka langsung dengan orang yang diwawancarai (H.M., 1990).

Pada pengumpulan data menurut Jogiyanto H.M. dalam bukunya “Analisis & Desain Sistem Informasi Pendekatan terstruktur Teori dan Praktek Aplikasi Bisnis” terdapat beberapa domain yang digunakan ketika proses pengumpulan data, diantaranya menganalisis kelemahan sistem, menganalisis distribusi pekerjaan, menganalisis pengukuran pekerjaan, menganalisis keandalan, menganalisis dokumen, menganalisis laporan, menganalisis teknologi. Menganalisis masalah dilakukan untuk menemukan jawaban apa penyebab sebenarnya dari masalah yang timbul. Menganalisis distribusi pekerjaan menunjukkan beban dari masing-masing personil atau unit organisasi dalam menangani kegiatan yang sama. Menganalisis keandalan menunjukkan banyaknya kesalahan-kesalahan yang dilakukan dalam suatu kegiatan, semakin andal berarti semakin sedikit kesalahan yang dilakukan. Menganalisis dokumen dilakukan untuk menganalisis dokumen yang digunakan dalam sistem yang lama. Menganalisis teknologi dilakukan untuk menganalisis teknologi yang digunakan dalam sistem yang lama.

## **2.6.2 Planning**

Tahapan ini merupakan lanjutan dari tahap komunikasi berupa analisis kebutuhan dan menghasilkan kebutuhan user atau bisa dibilang sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan perangkat lunak, termasuk rencana yang dilakukan (Rohayati, 2014). Fase analisis kebutuhan bisnis melibatkan penentuan persyaratan fungsional dan non fungsional. Menurut (Wazlawick, 2013) dalam mewakili kebutuhan fungsional dan non fungsional yang digunakan untuk analisis kebutuhan direpresentasikan dengan menggunakan pemodelan *use case*.

### **2.6.2.1 Analisis Kebutuhan Fungsional dan Non Fungsional**

Kebutuhan fungsional merupakan pernyataan layanan yang harus disediakan dalam sistem, bagaimana reaksi sistem dengan masukan tertentu dan bagaimana perilaku sistem dengan situasi tertentu. Pada beberapa kasus lain di persyaratan fungsional juga bisa secara eksplisit menyatakan hal yang tidak dilakukan oleh sistem. Sedangkan kebutuhan non fungsional merupakan batasan dalam fungsi atau layanan yang disediakan oleh sistem seperti waktu, proses dan batasan yang diberlakukan standar (Sommerville, 2011).

### **2.6.2.2 Analisis Proses Bisnis**

Pada analisis proses bisnis dilakukan pemetaan terhadap peroses bisnis yang berlaku (*as-is*) pada organisasi. Dalam melakukan analisis proses bisnis *as-is* pertama-tama mengumpulkan informasi berkaitan dengan *stakeholder* yang terlibat dalam suatu proses kemudian mengumpulkan informasi mengenai kondisi

proses bisnis yang sedang berjalan (Brandenburg, 2017). Selanjutnya informasi proses bisnis yang didapatkan dimodelkan kedalam bentuk BPMN. Setelah itu informasi yang diperoleh berkaitan dengan masalah yang ada dalam proses bisnis yang sedang berjalan dilakukan analisis agar ada solusi untuk dilakukan perbaikan. Berikut Tabel 2.7 cara untuk melakukan analisis permasalahan.

**Tabel 2.7 Analisis Permasalahan**

Masalah	[Deskripsi masalah]
Mempengaruhi	[Pemangku kepentingan yang terpengaruh oleh masalah]
Dampak	[Dampak dari masalah]
Solusi	[Solusi beserta manfaatnya]

Sumber : (Ramadhani, 2018)

Setelah melakukan analisis permasalahan kemudian diperlukan pemodelan usulan proses bisnis (*to-be*) sebagai gambaran atau penjelasan mengenai perbaikan atau usulan terhadap permasalahan yang terdapat pada organisasi, apakah sudah memenuhi kebutuhan dari *stakeholder* yang terlibat dalam proses bisnis tersebut (Brandenburg, 2017).

### 2.6.3 Modelling

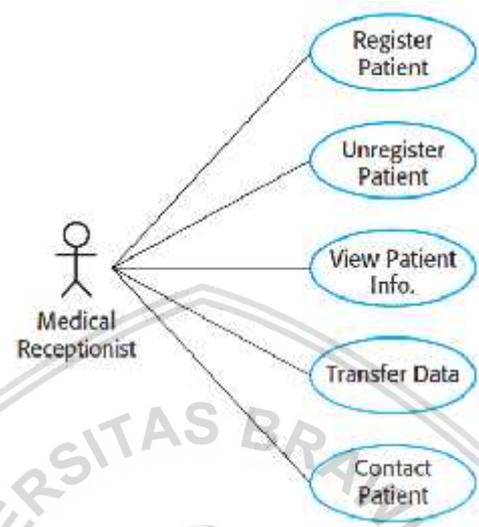
Pada langkah ini merupakan tahap perancangan dan permodelan perangkat lunak yang fokus dalam perancangan struktur data, arsitektur perangkat lunak, merancang tampilan antarmuka dan algoritma program (Pressman, 2010). Pada tahap pemodelan untuk memodelkan kerangka perangkat lunak yang akan dibangun menggunakan *Unified Modelling Language* (UML).

*Unified Modeling Language* (UML) adalah suatu standar bahasa dalam menulis *blueprints* perangkat lunak. UML bisa dipakai untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan suatu kerangka perangkat lunak. Sama halnya seperti arsitek bangunan dalam membangun rancangan *blueprints* agar digunakan oleh perusahaan konstruksi, pengembang perangkat lunak memuat suatu diagram UML agar bisa membantu dalam proses pengembangan perangkat lunak. Dengan menggunakan UML pengembang bisa lebih mudah memahami, menentukan sistem, dan menjelaskan perancangan sistem kepada orang lain (Pressman, 2010). Dalam UML tersedia *tools* yang akan digunakan untuk melakukan pemodelan sistem diantaranya yaitu *use case diagram*, *use case scenario*, *sequence diagram*, dan *class diagram*.

#### 2.6.3.1 Use Case Diagram

*Use Case Diagram* sebagai landasan dalam membantu menentukan fungsionalitas dan fitur apa saja dalam perangkat lunak dari sisi atau sudut pandang pengguna. *Use Case Diagram* menggambarkan bagaimana pengguna berinteraksi dengan suatu sistem dengan menentukan langkah-langkah yang dibutuhkan agar bisa mencapai tujuannya. *Use Case Diagram* memberikan Gambaran besar mengenai fungsionalitas suatu sistem dari semua kasus penggunaan dan bagaimana hubungan antar setiap *use case*. Pada *Use Case*

Diagram, simbol *use case* ditampilkan dengan bentuk oval, lalu aktor yang terhubung dengan setiap *use case* ditampilkan dengan garis. *Use Case Diagram* menampilkan semua *use case* sehingga bisa membantu dan memastikan bahwa semua fungsi yang ada pada sistem sudah tercakup. Contoh *Use Case Diagram* bisa dilihat pada Gambar 2.3 sebagai berikut:



**Gambar 2.3 Contoh Use Case Diagram**

Sumber : (Sommerville, 2011)

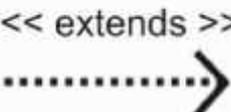
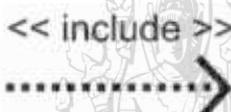
Pada Gambar 2.3 terdapat *Medical Receptionist* sebagai aktor yang memiliki beberapa hal yang bisa dilakukan terhadap perangkat lunak yang dikembangkan seperti *register patient*, *unregister patient*, *view patient info*, *transfer data*, *contact patient*.

Dalam membuat *Use Case Diagram* perlu memperhatikan simbol-simbol yang ada pada Tabel 2.8, disediakan semua simbol yang bisa digunakan dalam pembuatan *Use Case Diagram*.

**Tabel 2.8 Simbol-Simbol Use Case Diagram**

No	Nama	Gambar	Fungsi
1	<i>Uce Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
2	<i>Actor</i>		Orang, proses, atau sistem lain yang bisa berinteraksi dengan sistem yang akan dibangun.

Tabel 2.8 Simbol-Symbol Use Case Diagram (lanjutan)

No	Nama	Gambar	Fungsi
3	<i>Association</i>		Komunikasi yang berpartisipasi dalam <i>use case</i> , terdapat antara hubungan aktor dan <i>use case</i>
4	<i>Extend</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan
5	<i>Generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi salah satu usecase merupakan fungsi yang lebih umum dari lainnya
6	<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat jalannya <i>use case</i> tersebut

Sumber: (A.S & Shalahudin, 2015)

### 2.6.3.2 Use Case Scenario

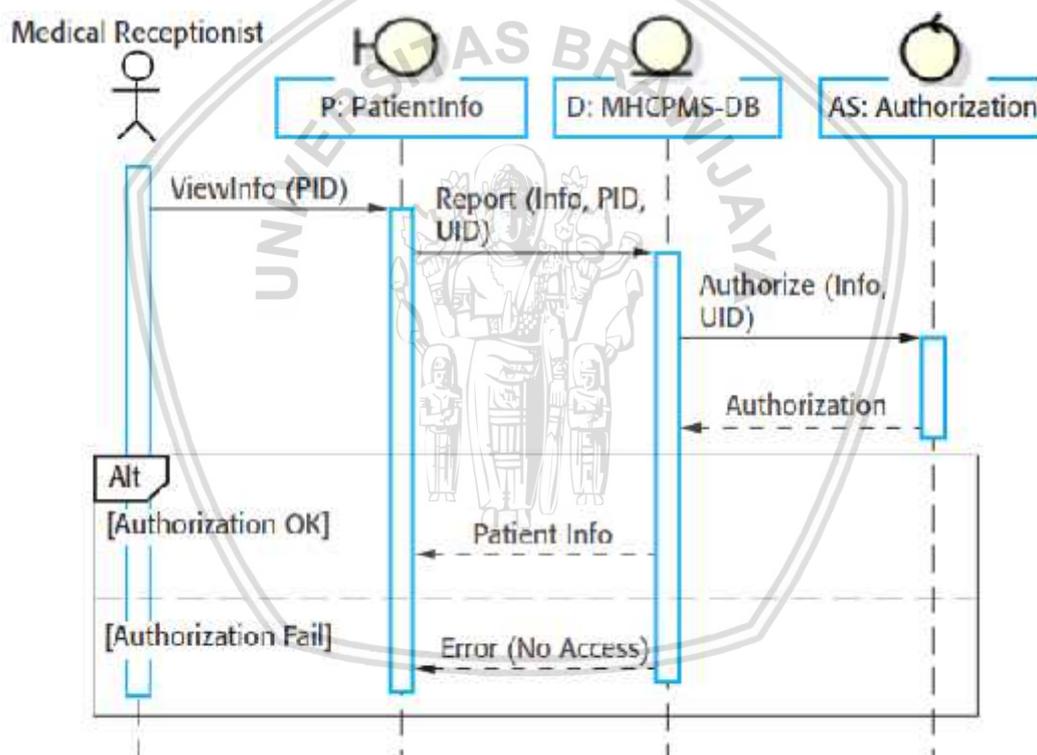
Use case scenario merupakan penjelasan secara tekstual dari sekumpulan skenario interaksi. Setiap skenario mendeskripsikan urutan aksi/ langkah yang dilakukan aktor ketika berinteraksi dengan sistem, baik yang berhasil maupun gagal (Kurniawan, 2018). Penjelasan use case scenario dilakukan secara tekstual pada beberapa format sesuai dengan kebutuhan dan penjelasannya bisa dibuat dengan tabel 1 kolom atau 2 kolom, yaitu singkat (*brief*), informal (*casual*) atau lengkap (*fully*). Penjelasan untuk format yang singkat cukup menuliskan 1 paragraf dengan mengacu pada skenario yang berhasil. Penjelasan untuk format informal bisa dibuat beberapa paragraf yang meliputi skenario berhasil dan gagal. Sedangkan penjelasan untuk format yang lengkap sering digunakan dalam banyak praktik, terdapat bagian penting yang terkandung dalam format lengkap diantaranya aktor primer, prakondisi, alur utama, alur alternatif, kondisi akhir.

Aktor primer merupakan inisiasi layanan sistem yang dilakukan oleh aktor supaya tujuan tercapai, jumlah aktor primer bisa lebih dari satu. Prakondisi merupakan kondisi spesifik yang harus dipenuhi sebelum sebuah use case dieksekusi oleh aktor primer. Alur utama merupakan jalur yang menjelaskan keberhasilan sebuah skenario hingga tujuannya tercapai. Alur alternatif

merupakan suatu jalur percabangan (pilihan) interaksi antar aktor dan sistem atau skenario gagal yang membuat tujuan aktor tidak terpenuhi. Kondisi akhir merupakan kondisi akhir saat sebuah use case berhasil dijalankan atau representasi dari tujuan yang ingin dicapai.

### 2.6.3.3 Sequence Diagram

*Sequence diagram* digunakan dalam menampilkan struktur komunikasi dinamis antar objek selama tugas atau fungsi dijalankan. Hal ini menunjukkan urutan proses pengiriman pesan antar objek dalam menyelesaikan tugas atau fungsi tertentu. Selain itu pada *sequence diagram* menunjukkan interaksi dari *use case* atau satu skenario pada sistem perangkat lunak. *Sequence diagram* juga menunjukkan pemanggilan metode yang digambarkan dengan panah horizontal dari pelaku ke target pelaku, diberi label menggunakan nama metode termasuk parameter, dan tipe kembalian. Terdapat contoh *sequence diagram* pada Gambar 2.4 berikut



Gambar 2.4 Contoh *Sequence Diagram*

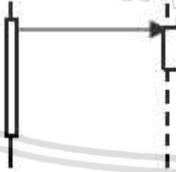
Sumber: (Sommerville, 2011)

Gambar 2.4 menunjukkan langkah-langkah yang terlibat saat dilakukan aktivitas tertentu pada sistem. Sumbu vertikal yang ada dalam diagram menunjukkan urutan waktu, dan setiap aliran waktu yang berjalan terus bergerak kebawah. Terdapat alternatif yang ditunjukkan untuk beberapa alur komunikasi dengan kondisi yang berbeda pada satu *sequence diagram* yang sama.



Dalam pembuatan stuktur *sequence diagram* harus memperhatikan dan mengerti simbol yang ada dikarenakan apabila ada kesalahan dalam penggambaran simbol maka akan terjadi kesalahpahaman dalam memahaminya dan sulit untuk tahap implementasi selanjutnya. Terdapat simbol yang bisa digunakan dalam pembuatan struktur *class diagram* pada Tabel 2.9 sebagai berikut :

**Tabel 2.9 Simbol-Simbol *Sequence Diagram***

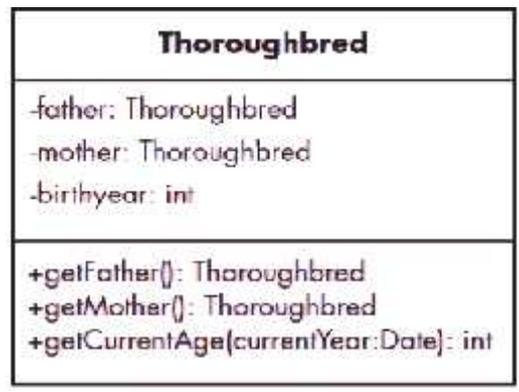
No	Nama	Gambar	Fungsi
1	<i>Actor</i>		Mendekripsikan orang yang berinteraksi dengan sistem
2	<i>Boundary</i>		Mendekripsikan <i>interface</i> dan interaksi satu atau lebih antara aktor dan sistem
3	<i>Controller</i>		Mendekripsikan perilaku mengatur, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem
4	<i>Entity</i>		Mendekripsikan data atau informasi yang disimpan dalam sistem
5	<i>Lifeline</i>		Mengindikasikan keberadaan sebuah obyek dalam basis waktu
6	<i>Message</i>		Mendekripsikan hubungan atau pesan antara objek satu dengan objek yang lain, dan juga meyakini urutan kejadian yang terjadi

Sumber: (A.S & Shalahudin, 2015)

#### 2.6.3.4 *Class Diagram*

Dalam memodelkan kelas, termasuk atributnya, operasi dan juga hubungan dengan kelas lain, UML menyediakan *class diagram model*. *Class diagram* menyediakan tampilan struktural atau statis dari suatu sistem. *Class diagram* tidak menyediakan sifat dinamis dari komunikasi antara objek kelas dalam diagram. Elemen utama pada *class diagram* adalah simbol kotak, merupaka ikon yang dipakai dalam mewakili antarmuka dan kelas. Pada setiap kotak dipisah menjadi bagian horizontal. Pada bagian tengah terdapat atribut tiap kelas. Setiap atribut mengarah pada sesuatu yang merupakan objek dalam kelas. Pada setiap atribut bisa mempunyai nama, tipe dan simbol. Terdapat simbol yang ditunjukkan dengan

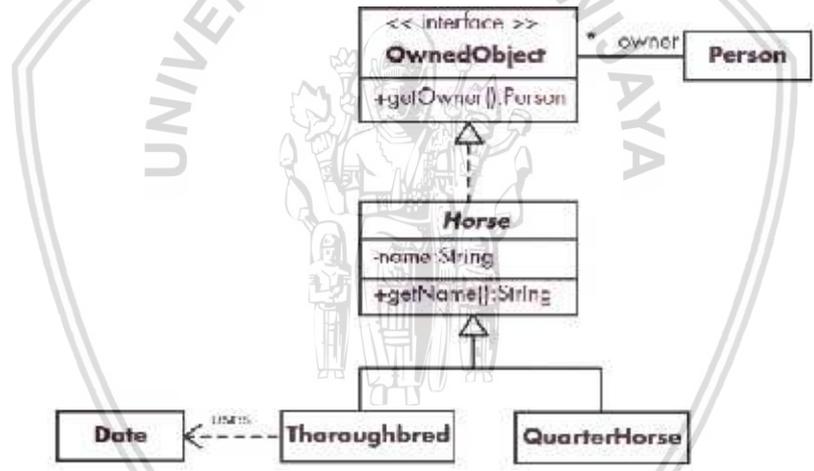
# (protected), - (private), + (public), ~ (package). Terdapat contoh *class diagram* pada Gambar 2.5 seperti berikut :



**Gambar 2.5 Contoh Class Diagram**

Sumber: (Pressman, 2010)

Pada Gambar 2.5 contoh *class diagram* semua atribut ditunjukkan sebagai *private* (-). Pada setiap operasi bisa ditampilkan dengan *public* (+), ada juga yang diberi parameter, jenis dan tipe pengembalian.



**Gambar 2.6 Hubungan Class Diagram**

Sumber: (Pressman, 2010)

Gambar 2.6 menunjukkan hubungan antar kelas. Kelas yang merupakan subkelas dari kelas lain terhubung dengan panah dengan garis yang membentuk segitiga. Tanda panah menunjuk dari *subclass* ke *superclass*. Pada UML, hubungan ini disebut *generalisasi*. Pada kelas *Thoroughbred* dan *QuarterHorse* ditunjukkan sebagai subclas dari kelas abstrak *Horse*.

Dalam pembuatan stuktur *class diagram* harus memperhatikan dan mengerti simbol yang ada dikarenakan apabila ada kesalahan dalam penggambaran simbol maka akan terjadi kesalahpahaman dalam memahaminya dan sulit untuk tahap implementasi selanjutnya. Terdapat simbol yang bisa digunakan dalam pembuatan struktur *class diagram* pada Tabel 2.10 sebagai berikut :

Tabel 2.10 Simbol-Simbol *Class Diagram*

No	Nama	Gambar	Fungsi
1.1	<i>Class</i>		<i>Class</i> di struktur sistem. <i>Class</i> mendefinisikan atribut, operasi dan <i>instance</i> .
1.2	<i>Association</i>		Relasi yang menunjukkan hubungan antara dua <i>class</i> .
1.3	<i>Generalization</i>		Relasi yang menunjukkan pewarisan dari <i>class</i> utama ( <i>parent</i> ) ke <i>class</i> anak ( <i>child</i> )
1.4	<i>Dependencies</i>		Relasi yang menunjukkan suatu <i>class</i> bergantung pada <i>class</i> yang lain
1.5	<i>Agregasi</i> / <i>Aggregation</i>		Relasi yang menunjukkan sebuah elemen yang terdiri dari beberapa komponen kecil

Sumber: (A.S & Shalahudin, 2015)

#### 2.6.4 Construction

Pada langkah ini dilakukan proses pembuatan kode program dan pengujian untuk menemukan kesalahan (Pressman, 2010). Pembuatan kode program merupakan penerjemahan desain kedalam sebuah bahasa yang bisa dibaca oleh komputer. Setelah pembuatan kode program selesai, maka selanjutnya dilakukan proses pengujian (*test*) perangkat lunak. Hal ini bertujuan agar membantu dalam proses perbaikan perangkat lunak apabila menemukan kesalahan yang ada dan memastikan perangkat lunak sudah berfungsi dengan layak.

Pengujian perangkat lunak adalah bagian penting dari suatu proses pengembangan perangkat lunak. Pentingnya pengujian ini untuk memastikan bahwa perangkat lunak yang telah dibangun sudah sesuai dengan yang direncanakan dan juga memastikan kualitas dari perangkat lunak. (Mustaqbal, et al., 2015) mengungkapkan bahwa pengujian perangkat lunak merupakan satu elemen dari suatu topik yang sering diartikan lebih luas sebagai verifikasi dan validasi. Verifikasi lebih mengarah pada kumpulan aktivitas untuk memastikan perangkat lunak telah mengimplementasikan fungsi-fungsi yang spesifik. Sedangkan validasi mengarah pada suatu kumpulan dari aktivitas yang berbeda agar memastikan bahwa perangkat lunak yang dibangun bisa ditelusuri terhadap kebutuhan klien.

##### 2.6.4.1 CodeIgniter

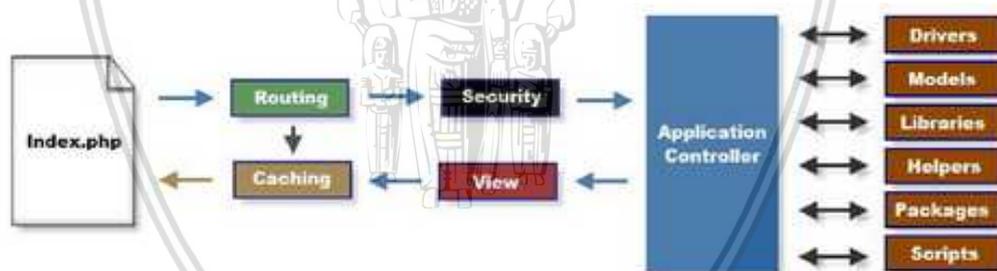
CodeIgniter adalah *framework* yang memakai model MVC (*model*, *view*, dan *controller*) dalam mengembangkan suatu *website* dinamis dengan memakai

bahasa pemrograman PHP (*Hypertext Preprocessor*), *framework* ini merupakan turunan dari bahasa pemrograman PHP oleh karena itu pembuatan program tidak jauh dari bahasa pemrograman PHP yang sudah di kenal sebelumnya. Dalam *model, view* dan *controller* ini para *developer* akan lebih gampang dalam mengembangkan suatu aplikasi website dengan desain tampilan terstruktur, sehingga dalam proses *maintenance website* akan lebih mudah (Tim Airputih, 2014).

Terdapat beberapa alasan dalam menggunakan *framework* diantaranya (Koespradono, 2013):

- 2.6.4.1.1.1 Mempermudah dan mempercepat dalam pembangunan suatu aplikasi *web*
- 2.6.4.1.1.2 Relatif mudah dalam melakukan proses *maintenance*, hal ini dikarenakan sudah terdapat pola tertentu yang ada di *framework*. Pada umumnya *framework* ini telah menyediakan fasilitas yang sudah umum digunakan sehingga pengembang tidak perlu membangunnya dari awal, seperti validasi, session, error handling, dan sebagainya
- 2.6.4.1.1.3 Lebih bebas dalam proses pengembangan apabila dibandingkan dengan *Content Management System (CMS)*

MVC merupakan suatu pola dasar yang memisahkan bagian program kedalam objek yang terpisah. Gambaran Interaksi pada bagian program CodeIgniter sebagai Gambar 2.8 berikut :



Gambar 2.7 Alur kerja framework CodeIgniter

Sumber: (Widodo, 2013)

Penjelasan Gambar 2.7 yaitu pertama *index.php* merupakan controller awal yang menginisialisasikan kebutuhan dalam menjalankan CodeIgniter. Selanjutnya data yang berasal dari *index.php* akan melewati *routing* untuk menentukan kegiatan apa yang harus dikerjakan saat diminta oleh *browser* atau *client*. Apabila cache sudah aktif, otomatis data yang ada akan langsung dikirimkan ke klien dan alur kerja utama akan diabaikan. Pada *security*, data akan difilter keamanannya. Pada *Controller* proses permintaan akan dikendalikan sesuai dengan permintaan klien seperti *models, helper, plugins, libraries, script* dan lain sebagainya. *View* siap menyajikan dan menampilkan informasi di *web browser* sesuai dengan permintaan (Widodo, 2013).

#### 2.6.4.2 Black Box Testing

*Black Box Testing* lebih fokus terhadap spesifikasi fungsional yang ada dalam perangkat lunak. Penguji bisa mendefinisikan kumpulan dari beberapa kondisi masukan dan menguji spesifikasi fungsional program. *Black box testing* bukan termasuk dari solusi alternatif selain dari *white box testing* namun lebih mengarah pada pelengkap untuk melakukan pengujian yang tidak dicakup dalam *white box testing* (Mustaqbal, et al., 2015). Proses pada *black box testing* merupakan pengujian yang mengacu pada spesifikasi sistem, dimana perilakunya hanya dapat ditentukan dengan memahami dan mempelajari hubungan antara masukan dan hasil keluaran.

*Black box testing* lebih difokuskan pada fungsionalitas dan bukan pengimplementasi dari perangkat lunak. Dalam *black box testing* lebih condong dalam menemukan beberapa hal seperti berikut (Mustaqbal, et al., 2015):

- 2.6.4.2.1.1.1 Fungsi yang tidak ada atau tidak benar
- 2.6.4.2.1.1.2 Terdapat kesalahan dalam antarmuka
- 2.6.4.2.1.1.3 Terdapat dalam struktur data dan juga akses basis data
- 2.6.4.2.1.1.4 Terdapat kesalahan dalam performansi
- 2.6.4.2.1.1.5 Terdapat kesalahan dalam inialisasi dan terminasi

Pada *black box testing* untuk menguji kebutuhan fungsional peneliti menggunakan *validation testing*.

##### 1. Validation Testing

*Validation testing* digunakan agar mengetahui apakah perangkat lunak yang dibangun telah benar sesuai dengan apa yang dibutuhkan. Item-item yang sudah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi tidak memerlukan konsentrasi terhadap alur algoritma program dan lebih ditekankan dalam menemukan konformitas antara daftar kebutuhan dan kinerja sistem (Indriati, 2010). Terdapat beberapa poin yang ada dalam pengujian validasi yaitu :

- a. Nama kasus uji
- b. Hasil yang diharapkan
- c. Hasil yang didapatkan
- d. Status Validitas

#### 2.6.4.3 User Acceptance Testing

Sebelum sebuah perangkat lunak yang telah dibangun diberikan kepada klien maka perlu dilakukan pengujian agar memastikan seluruh persyaratan pengguna yang ada pada *Specification Requirement Software* telah terpenuhi dan sesuai. Pengujian yang dilakukan pada tahap ini adalah *User Acceptance Testing* (UAT).

UAT dilakukan setelah semua sistem siap digunakan dan sebelum perangkat lunak digunakan oleh klien atau pengguna.

*User Acceptance Testing* (UAT) merupakan sarana formal dimana perusahaan mengkonfirmasi bahwa sistem yang telah dibangun sudah benar-benar memenuhi persyaratan pengguna. Tahap ini merupakan pengujian akhir yang dilakukan setelah melakukan pengujian fungsional. *User Acceptance Testing* (UAT) dilakukan oleh pengguna dan pengembang perangkat lunak, sebelum perangkat lunak ini dioperasikan pengujian ini dilakukan sebagai tahap akhir pengujian perangkat lunak. Tujuan utama dari *User Acceptance Testing* (UAT) adalah mengecek perangkat lunak terhadap kebutuhan bisnis, hal ini dilakukan oleh pengguna akhir yang sudah familiar dengan kebutuhan bisnis (Goel & Gupta, 2014).

Inti dari user acceptance testing terdapat pada seperangkat *acceptance criteria* yang harus dipenuhi oleh perangkat lunak agar bisa diterima (Naik, 2008). Terdapat acceptance criteria yang digunakan oleh peneliti berdasarkan buku "SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice" yang dibuat oleh Kshirasagar Naik diantaranya yaitu *Usability, Functional Correctness and Completeness, Confidentiality and Availability*.

*Usability* terdapat pertanyaan mengenai seberapa mudah dalam menggunakan sistem dan seberapa mudah dalam mempelajarinya. Tujuan dari *usability* untuk memastikan bahwa sistemnya fleksibel, mudah untuk mengonfigurasi, antarmuka *friendly*, bantuan *online* tersedia, pekerjaan tersedia (Naik, 2008).

*Functional Correctness and Completeness* terdapat pertanyaan mengenai apakah sistem melakukan apa yang diinginkan klien. Semua fitur yang dijelaskan dalam spesifikasi persyaratan harus ada dalam perangkat lunak yang di kirimkan (*delivered*) ke klien. Penting agar menunjukkan bahwa sistem bekerja dengan benar, setidaknya terdapat dua sampai tiga kondisi untuk fitur sebagai bagian dari penerimaan (Naik, 2008).

*Confidentiality and Availability* terdapat pertanyaan mengenai kriteria penerimaan *confidentiality* yang mengacu pada persyaratan bahwa data harus dilindungi dari pengungkapan yang tidak sah dan kriteria penerimaan *availability* dengan persyaratan bahwa data harus dilindungi dari *denial of service* (DoS) kepada pengguna yang berwenang (Naik, 2008).

#### **1.1.1.2 Skala Likert**

Skala *Likert* merupakan skala tanggapan psikometri yang sering digunakan dalam kuesioner untuk mengukur persepsi, pendapat atau sikap seseorang dalam mendapatkan tingkat persetujuan dari responden menggunakan pernyataan atau serangkaian pernyataan berdasarkan definisi operasional yang sudah ditetapkan oleh peneliti (Choizes, 2018). Umumnya *skala likert* menggunakan skala yang terdiri dari 5 poin secara terurut dimulai dari "Sangat Tidak Setuju" di salah satu ujung kemudian "Sangat Setuju" di ujung lainnya kemudian "Antara Setuju

maupun Tidak Setuju” di tengah-tengahnya. Terkadang juga digunakan skala 4 poin atau angka genap lainnya yang bertujuan untuk menghasilkan ukuran ipsatif atau pilihan paksa. Setiap tingkat pada skala diberikan nilai numerik yang biasanya dimulai dari 1 dan ditambah satu untuk tiap tingkatnya. Berikut pada Gambar 2.10 merupakan contoh dari skala *likert* dengan menggunakan skala 5 poin.



**Gambar 2.10 Contoh Skala *Likert***

Sumber : (Choizes, 2018)

Dalam implementasi skala *likert* terdapat beberapa pernyataan yang masing-masing memiliki bobot nilai tertentu. Proses perhitungan hasil dari skala *likert* diawali dengan memberikan bobot nilai pada tiap jawaban. Berikut pada Tabel 2.11 merupakan rincian bobot nilai dari jawaban. Sedangkan pada Tabel 2.12 merupakan persentase untuk setiap jawaban.

**Tabel 2.11 Bobot Nilai Jawaban**

Pilihan Jawaban	Bobot Nilai
A = Sangat Setuju	5
B = Setuju	4
C = Netral	3
D = Kurang	2
E = Sangat Kurang	1

Sumber: (Choizes, 2018)

**Tabel 2.12 Persentase Nilai Jawaban**

Interval Persentase	Interpretasi Skor
80 sampai 100%	Sangat Setuju
60 sampai 79,99%	Setuju
40 sampai 59,99%	Netral
20 sampai 39,99%	Tidak Setuju
0 sampai 19,99%	Sangat Kurang Setuju

Sumber: (Choizes, 2018)

Setelah bobot nilai dan persentase nilai telah ditentukan, selanjutnya adalah mengkalikan jumlah jawaban dengan bobot nilainya. Untuk perhitungannya akan dijelaskan pada Rumus berikut:

1. Jawaban Sangat Setuju (A)  $= n \times 5$

2. Jawaban Setuju (B) =  $n \times 4$   
 3. Jawaban Netral (C) =  $n \times 3$   
 4. Jawaban Tidak Setuju (D) =  $n \times 2$   
 5. Jawaban Sangat Tidak Setuju (E) =  $n \times 1$

$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1) \quad (2.3)$$

$n$  merupakan jumlah dari responden yang menjawab. Perhitungan selanjutnya adalah mencari nilai terendah ( $Y$ ) dan nilai tertinggi dari hasil skala *likert* tersebut. Rumus yang digunakan dijelaskan pada Rumus 2.4 dan Rumus 2.5, dimana  $N1$  merupakan nilai tertinggi skala *likert*,  $N2$  merupakan nilai terendah skala *likert*,  $n$  merupakan jumlah responden, dan  $U$  merupakan jumlah uji kasus.

$$Y = N1 \times n \times U \quad (2.4)$$

$$Y = N1 \times n \times U \quad (2.5)$$

Sedangkan untuk menghitung hasil dari UAT dengan menggunakan skala *likert* dijelaskan pada Rumus 2.6 berikut.

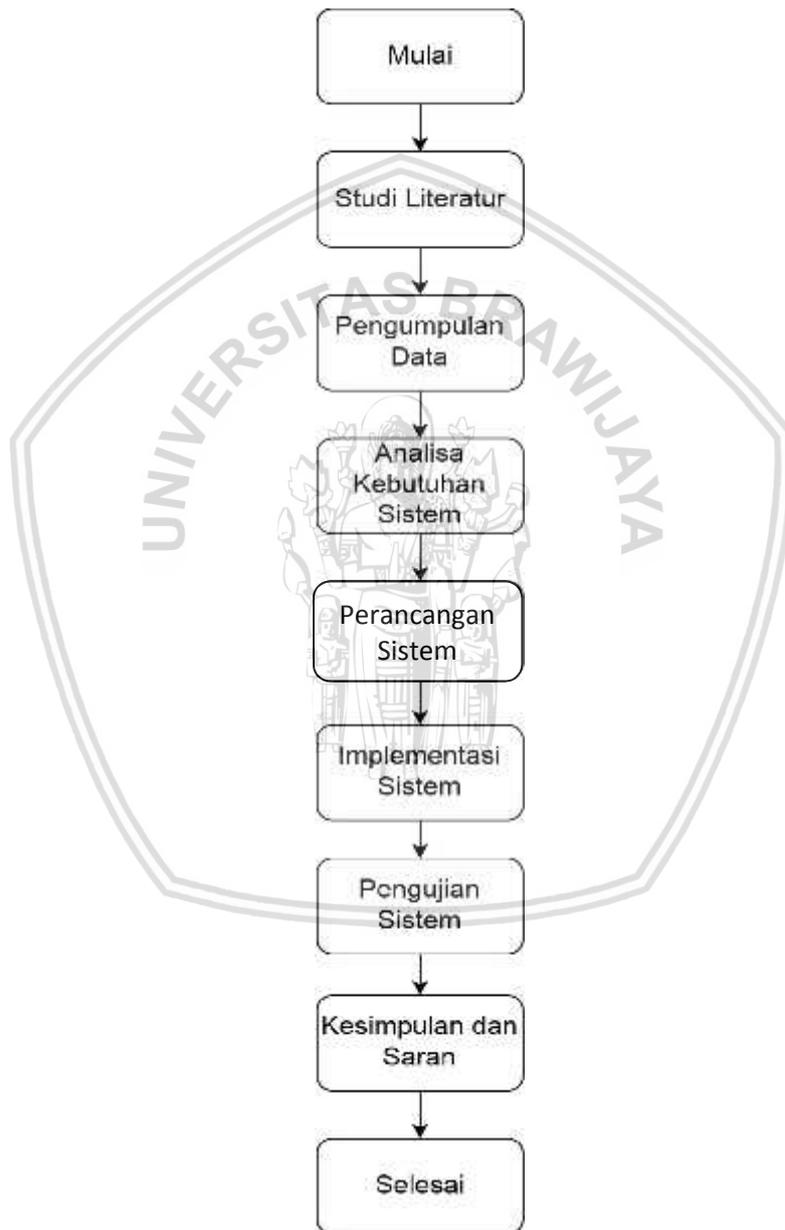
$$\text{Rumus Index \%} = (\text{Total Nilai} \div Y) \times 100 \quad (2.6)$$

### 2.6.5 Deployment

Pada langkah ini merupakan langkah akhir dalam pengembangan perangkat lunak. Perangkat lunak diserahkan (*delivery*) dan diimplementasikan kepada klien, selanjutnya dilakukan dukungan (*support*) dalam pemeliharaan perangkat lunak, dan menerima umpan balik (*feedback*) yang diberikan supaya perangkat lunak bisa tetap berjalan dengan baik (Pressman, 2010).

### BAB 3 METODOLOGI

Pada bab ini merupakan kerangka kerja yang membahas tentang penjabaran dari langkah-langkah yang dilakukan dalam penyusunan skripsi yang sesuai dengan latar belakang penelitian yang sudah diangkat. Supaya memberikan kemudahan dalam memberikan penjabaran tentang setiap proses yang akan dilakukan maka peneliti menyajikannya dalam diagram alir sebagai Gambar 3.1 berikut :



**Gambar 3.1 Diagram Alir Metodologi Penelitian**

### 3.1 Studi Literatur

Pada langkah ini dilakukan studi literatur atau pemahaman kepustakaan yang berkaitan dengan objek penelitian untuk mempelajari berbagai macam referensi yang diambil dari beragam sumber seperti: jurnal, buku, artikel dan laporan penelitian yang serupa, guna dibuat sebagai referensi dalam melakukan pengembangan sistem informasi *monitoring* pekerjaan kantor konsultan pajak berbasis web pada kantor konsultan pajak I Gede Arianta.

### 3.2 Pengumpulan Data

Pada langkah ini dilakukan tahap pengumpulan data dimana peneliti melakukan wawancara secara langsung guna untuk memperoleh informasi yang dibutuhkan dalam kebutuhan sistem dan sesuai dengan apa yang diinginkan oleh kantor konsultan pajak yaitu untuk membuat suatu sistem informasi *monitoring* pekerjaan dalam kantor supaya meningkatkan kinerja perusahaan. Data diperoleh dari pemilik kantor konsultan pajak yang mengatur manajemen yang ada di kantor dan ketua tim kantor konsultan pajak sebagai salah satu penanggung jawab bagi anggota staf yang lain. Data yang diperoleh adalah data yang bisa mendeskripsikan kebutuhan fungsional seperti fungsi-fungsi seperti apa yang berjalan dan kebutuhan non-fungsional seperti tampilan dari sistem yang akan dikembangkan.

### 3.3 Analisis Kebutuhan Sistem

Pada langkah ini dilakukan pemetaan hasil pengumpulan data untuk memperoleh kebutuhan fungsional. Dalam melakukan analisis kebutuhan memiliki tujuan untuk melakukan penyusunan kebutuhan agar bisa dilakukan dengan runtut dan terstruktur dan mempermudah dalam proses perancangan sistem yang akan dikembangkan. Hasil dari tahap ini menghasilkan pemodelan proses bisnis yang ada saat ini dan yang diajukan, *use case diagram* dan *use case scenario*.

### 3.4 Perancangan Sistem

Pada langkah ini mengacu pada hasil dari analisis kebutuhan yang sudah digali sebelumnya. Hasil dari langkah ini akan digunakan untuk referensi dalam merancang arsitektur pada proses implementasi. Pada perancangan sistem dalam penelitian ini dihasilkan *sequence diagram*, *class diagram*. Pembuatan diagram berdasarkan hasil dari analisis kebutuhan fungsional yang ada pada tahap analisis kebutuhan sistem sebelumnya.

Tahap setelah diagram dibuat adalah merancang *User Interface (UI)* atau tampilan yang nantinya akan diimplementasikan dalam sistem. Tampilan yang dirancang berupa tata letak terhadap setiap fungsi ataupun hal lain yang berkaitan dengan tampilan pada sistem informasi *monitoring* pekerjaan kantor konsultan pajak berbasis web pada kantor konsultan pajak I Gede Arianta.

### 3.5 Implementasi Sistem

Pada langkah ini dilakukan pengimplementasian sistem atau tahapan penulisan code program menggunakan bahasa pemrograman *Hypertext Markup Language (HTML)*, *Cascading Style Sheets (CSS)*, *Javascript*, *Asynchronous JavaScript and XML (AJAX)*, *Jquery*, *Hypertext Processor (PHP)*, dilengkapi dengan fasilitas yang telah disediakan oleh *framework CodeIgniter* serta menggunakan database MySQL. Dalam tahapan penulisan code program mengacu pada desain yang sudah dirancang pada tahap perancangan sistem sebelumnya agar sesuai dengan harapan dari kantor sebagai suatu sistem informasi *monitoring* pekerjaan kantor konsultan pajak yang baik.

### 3.6 Pengujian

Pada langkah ini dilakukan pengujian perangkat lunak yang telah selesai dibangun. Pengujian ini merupakan langkah penting karena menentukan kualitas baik dan buruknya dari suatu perangkat lunak. Pada pengujian ini dilakukan pembuktian dari proses pemenuhan kebutuhan yang sudah didefinisikan sebelumnya. Pengujian pertama yang dilakukan oleh peneliti dalam penelitian ini menggunakan pengujian secara non-fungsional berupa pengujian *black-box* untuk menguji keandalan dari perangkat lunak dan *user acceptance* untuk menguji bahwa perangkat lunak diterima dan memenuhi persyaratan pengguna. Pengujian *black-box* menggunakan pengujian *validation*.

Pengujian kedua berupa *black-box testing* menggunakan *validation testing* yang lebih mengedepankan terhadap fungsional suatu perangkat lunak dimana peneliti menyediakan satu paket masukan valid dan tidak valid, selanjutnya hasil keluaran akan diperiksa apakah sesuai dengan yang diharapkan atau tidak.

Pengujian ketiga akan mengukur waktu pada sebelum implementasi sistem informasi dan sesudah implementasi sistem informasi dan dihitung selisihnya. Pada pengujian waktu akan dihitung rata-rata waktu yang dibutuhkan. Hasil dari tiap pengujian akan dianalisis dan dijadikan masukan pada pengembangan tahap selanjutnya.

### 3.7 Kesimpulan dan Saran

Pada langkah ini merupakan suatu penilaian, evaluasi dan peninjauan ulang terhadap perangkat lunak yang dikembangkan agar bisa digunakan dengan baik, kebutuhan pengguna sudah terpenuhi dan sudah menjawab permasalahan yang telah dirumuskan di latar belakang. Pada penarikan kesimpulan didapatkan dari hasil analisis terhadap penelitian yang sudah dilakukan dan penulisan saran bisa disajikan kepada pengembang sistem selanjutnya agar memperbaiki kesalahan atau menambahkan kekurangan yang terdapat pada perangkat lunak yang telah dikembangkan.

## BAB 4 ANALISIS KEBUTUHAN

Bagian ini menjelaskan tentang proses analisis kebutuhan dari sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta yang akan dikembangkan. Tahap pertama menjelaskan terkait informasi yang didapatkan dari hasil wawancara dengan *stakeholder* dan memetakan informasi ke dalam tabel analisis permasalahan. Disamping itu informasi yang diperoleh berupa proses bisnis kantor juga digambarkan dengan diagram BPMN. Pada informasi yang diperoleh juga terdapat informasi tentang *stakeholder* yang terlibat dalam penggunaan sistem informasi *monitoring* pekerjaan kantor konsultan pajak nantinya.

Analisis kebutuhan dibuat agar mendapatkan kebutuhan yang umum kemudian dilanjutkan dengan spesifikasi kebutuhan yang digunakan agar mendapatkan kebutuhan fungsional yang akan dimanfaatkan dalam pembuatan *Use Case Diagram* dan *Use Case Scenario* sehingga bisa digunakan dalam tahap perancangan sistem yang ada pada bab selanjutnya.

### 4.1 Proses Rekayasa Kebutuhan

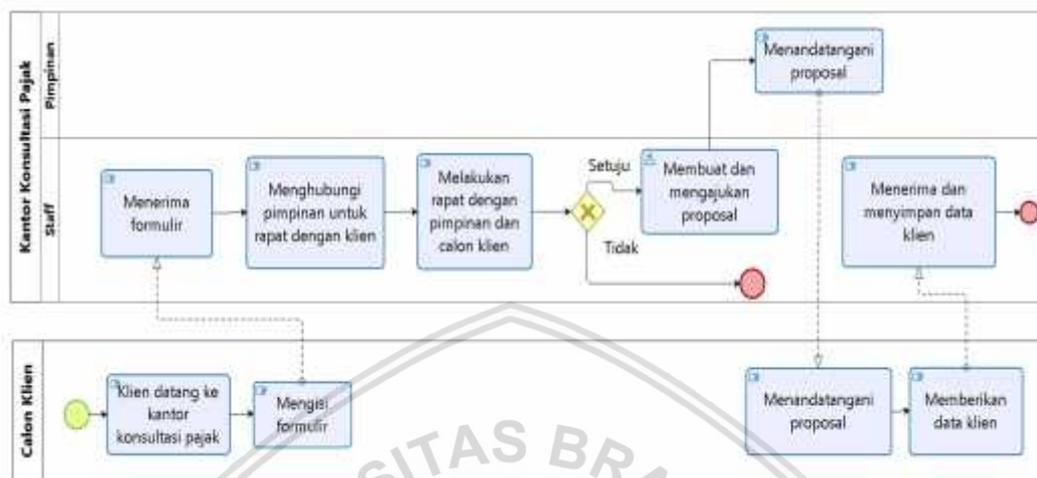
Proses rekayasa kebutuhan merupakan tahap untuk mengumpulkan kebutuhan-kebutuhan sistem. Tahap-tahap ini meliputi pemodelan proses bisnis *as-is*, elisitasi kebutuhan, identifikasi aktor pemodelan proses bisnis *to-be* dan spesifikasi kebutuhan.

Pemodelan proses bisnis *as-is* merupakan analisis mengenai kondisi perusahaan saat ini yang dilakukan dengan melakukan pemodelan menggunakan *Business Process Modelling Notation* (BPMN). Selanjutnya dilakukan tahapan elisitasi kebutuhan untuk mendefinisikan kebutuhan sistem dengan menggunakan teknik wawancara yang dilakukan kepada pihak kantor konsultan pajak I Gede Arianta. Kemudian setelah memperoleh permasalahan dan kebutuhan terhadap sistem saat ini maka selanjutnya melakukan identifikasi aktor yang terlibat pada sistem yang akan dibangun. Selanjutnya melakukan pemodelan proses bisnis *to-be* sebagai usulan perbaikan dari permasalahan sistem saat ini. Kemudian analisis kebutuhan fungsional, kebutuhan fungsional berupa spesifikasi kebutuhan untuk mendefinisikan interaksi antara aktor dan sistem yang akan dibangun.

#### 4.1.1 Identifikasi Proses Bisnis AS-IS

Berdasarkan hasil wawancara yang telah dilakukan, diperoleh bagaimana proses dalam menangani klien yang ada pada kantor konsultan pajak dari awal hingga akhir pelayanan kepada klien. Proses ini diawali dari pendaftaran sebagai klien pada kantor konsultan pajak dan dilakukan proses diskusi, setelah resmi menjadi klien selanjutnya bisa dilakukan konsultasi dan proses penyerahan data klien kepada kantor konsultan pajak. Kemudian terdapat proses perhitungan pajak dan pembuatan dokumen yang dibutuhkan untuk pembayaran pajak dan pelaporan pajak. Pada setiap tahap pembayaran pajak dan pelaporan pajak

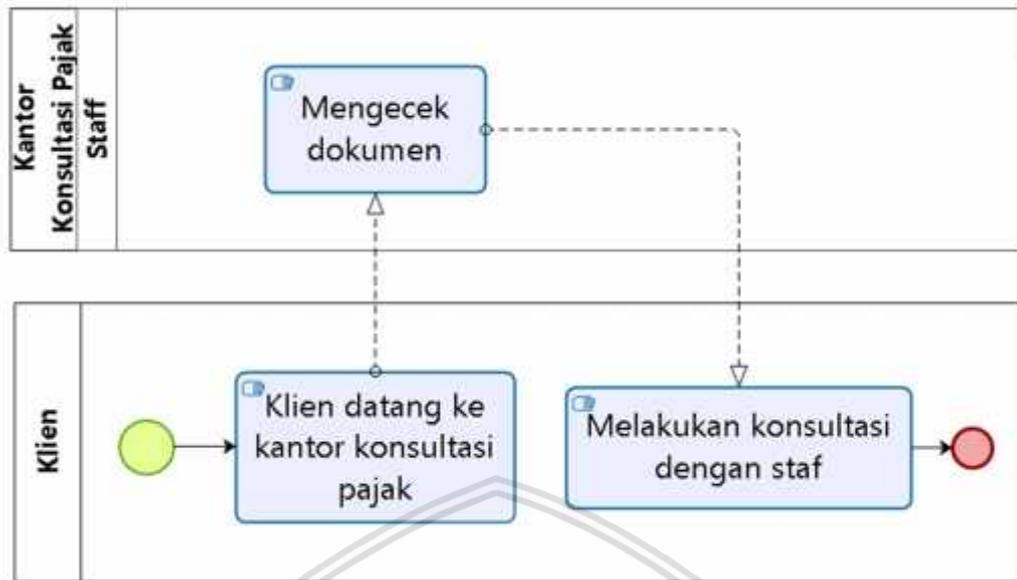
disampaikan kepada pimpinan kantor konsultan pajak untuk dicek dan divalidasi. Berikut merupakan penggambaran notasi BPMN yang digambarkan berdasarkan alur proses bisnis yang ada pada kantor konsultan pajak.



**Gambar 4.1 Proses Bisnis Registrasi Klien Saat Ini**

Penjabaran proses bisnis registrasi klien saat ini pada Gambar 4.1 sebagai berikut:

1. Calon klien datang ke kantor konsultan pajak I Gede Arianta
2. Kantor konsultan menyediakan dan memberikan formulir untuk klien baru
3. Calon klien mengisi formulir yang diberikan
4. Staf menerima data formulir calon klien
5. Staf menghubungi pimpinan agar dilakukan rapat dengan calon klien
6. Calon klien melakukan rapat dengan pimpinan dan juga staf yang nanti akan bertanggung jawab kepada klien tersebut
7. Setelah calon klien dan pihak kantor setuju maka staff membuat proposal sebagai kontrak kedua belah pihak, yaitu pihak calon klien dan pihak kantor konsultan pajak
8. Pimpinan menandatangani proposal
9. Calon klien menandatangani proposal
10. Staf meminta salinan data terkait data klien yang dibutuhkan oleh kantor konsultan pajak
11. Klien memberikan salinan data
12. Staf menerima dan menerima salinan data yang diberikan oleh klien



**Gambar 4.2 Proses Bisnis Konsultasi Saat Ini**

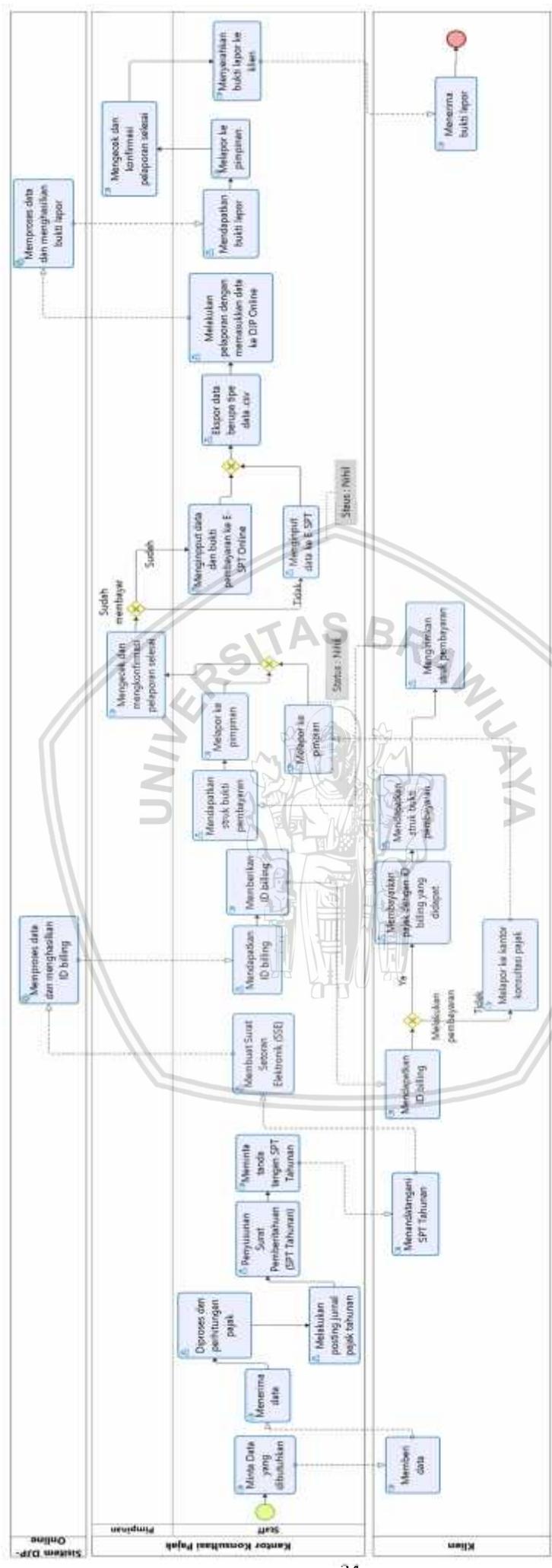
Penjabaran proses bisnis konsultasi saat ini pada Gambar 4.2 sebagai berikut:

1. Klien datang ke kantor konsultan pajak I Gede Arianta untuk berkonsultasi
2. Apabila klien membawa dokumen sebagai kebutuhan perpajakan atau sejenisnya maka diserahkan kepada staf, apabila membawa dokumen maka langsung berkonsultasi dengan staf
3. Apabila klien menyerahkan dokumen maka staf menerima dan menyimpan dokumen tersebut, kemudian melakukan konsultasi dengan klien



Penjabaran proses bisnis *monitoring* pajak bulanan saat ini pada Gambar 4.3 sebagai berikut:

1. Staf meminta data yang dibutuhkan untuk melakukan pembayaran pajak bulanan
2. Klien memberikan data kepada staf
3. Staf menerima data yang diberikan oleh klien
4. Staf melakukan proses perhitungan pajak klien
5. Staf melakukan penyusunan surat pemberitahuan (SPT) dan juga meminta tanda tangan klien untuk arsip
6. Klien menandatangani surat pemberitahuan (SPT)
7. Staf membuat surat setoran elektronik (SSE) di sistem DJP Online
8. DJP online memproses SSE dan menghasilkan ID billing untuk pembayaran
9. Staf memberikan ID billing kepada klien untuk dilakukan pembayaran
10. Klien membayarkan pajak sesuai dengan ID billing yang diberikan, selanjutnya mendapatkan bukti bayar (NTPN). Setelah itu menyerahkan bukti bayar kepada staf
11. Staf menerima bukti bayar dari klien
12. Staf melapor kepada pimpinan tentang pembayaran pajak bulanan klien, sehingga pimpinan bisa mengecek dan mengkonfirmasi
13. Memasukkan data perpajakan dan bukti bayar ke sistem E-SPT Online
14. Staf melakukan ekspor data (tipe data .csv) yang nantinya digunakan untuk melakukan pelaporan di DJP Online
15. Staf melakukan pelaporan pajak dengan memasukkan data ke DJP Online
16. DJP Online menerima selanjutnya memproses data lapor, setelah itu menghasilkan bukti lapor
17. Staf melapor kepada pimpinan tentang pelaporan pajak bulanan klien, sehingga pimpinan bisa mengecek dan mengkonfirmasi
18. Staf menyerahkan bukti lapor ke klien
19. Klien menerima bukti lapor dari staf



Gambar 4.4 Proses Bisnis Monitoring Pajak Tahunan Saat Ini

Penjabaran proses bisnis *monitoring* pajak tahunan saat ini pada Gambar 4.4 sebagai berikut:

1. Staf meminta data yang dibutuhkan untuk melakukan pembayaran pajak tahunan
2. Klien memberikan data kepada staf
3. Staf menerima data yang diberikan oleh klien
4. Staf melakukan proses perhitungan pajak klien
5. Staf melakukan posting jurnal perpajakan tahunan
6. Staf melakukan penyusunan surat pemberitahuan (SPT) dan juga meminta tanda tangan klien untuk arsip
7. Klien menandatangani surat pemberitahuan (SPT)
8. Staf membuat surat setoran elektronik (SSE) di sistem DJP Online
9. DJP online memproses SSE dan menghasilkan ID billing untuk pembayaran
10. Staf memberikan ID billing kepada klien untuk dilakukan pembayaran
11. Klien membayarkan pajak sesuai dengan ID billing yang diberikan, selanjutnya mendapatkan bukti bayar (NTPN). Setelah itu menyerahkan bukti bayar kepada staf
12. Staf menerima bukti bayar dari klien
13. Staf melapor kepada pimpinan tentang pembayaran pajak bulanan klien, sehingga pimpinan bisa mengecek dan mengkonfirmasi
14. Memasukkan data perpajakan dan bukti bayar ke sistem E-SPT Online
15. Staf melakukan ekspor data (tipe data .csv) yang nantinya digunakan untuk melakukan pelaporan di DJP Online
16. Staf melakukan pelaporan pajak dengan memasukkan data ke DJP Online
17. DJP Online menerima selanjutnya memproses data lapor, setelah itu menghasilkan bukti lapor
18. Staf melapor kepada pimpinan tentang pelaporan pajak bulanan klien, sehingga pimpinan bisa mengecek dan mengkonfirmasi
19. Staf menyerahkan bukti lapor ke klien
20. Klien menerima bukti lapor dari staf

#### 4.1.1.2 Analisis Permasalahan Dalam Menangani Monitoring

Dalam sub bab ini membahas tentang masalah yang ada saat ini dalam proses *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta sehingga bisa ditemukan solusi yang bisa mengatasi permasalahan tersebut. Berikut Tabel 4.1 menjelaskan mengenai analisis permasalahan yang ada.

**Tabel 4.1 Analisis Permasalahan Dalam Menangani Monitoring**

Masalah	1. Data klien belum terkelola dengan baik dan teratur
	2. Aktivitas konsultasi harian antara staf dan klien belum terekam dan sulit dimonitor oleh pimpinan
	3. Pelaporan <i>monitoring</i> pekerjaan perpajakan klien belum terekam dan sulit dimonitor oleh pimpinan
	4. Tidak ada bukti tanda terima yang tersimpan untuk penyerahan dokumen dari klien ke staf
	5. Proses validasi perpajakan bulanan, perpajakan tahunan yang dikerjakan oleh staf tentang pekerjaannya yang sudah selesai masih tidak teratur.
Mempengaruhi	1. Penyelesaian masalah klien yang dilakukan oleh staf kantor konsultan pajak
	2. Staf dalam melaporkan pekerjaan perpajakan klien kepada pimpinan
Dampak	1. Data klien hilang, sehingga tidak bisa mengetahui data klien
	2. Permasalahan klien tidak kunjung selesai
	3. Perpajakan bulanan klien atau tahunan klien tidak terbayarkan
	4. Terdapat kesalahpahaman yang terjadi terkait pertanggungjawaban dokumen yang telah diserahkan antara pihak kantor dan klien
Solusi	1. Menyediakan sistem yang bisa mendukung dalam mengelola data klien secara baik dan teratur
	2. Meyediakan sistem yang bisa memberikan informasi tentang aktivitas konsultasi harian yang dilakukan oleh staf
	3. Meyediakan sistem yang dapat meyimpan dan memberikan bukti tanda terima tentang penyerahan data klien

**Tabel 4.1 Analisis Permasalahan Dalam Menangani *Monitoring* (lanjutan)**

	4. Menyediakan sistem yang dapat melakukan proses <i>monitoring</i> pekerjaan dan validasi untuk pekerjaan yang dikerjakan oleh setiap staf kepada pimpinan
--	---

#### 4.1.2 Elisitasi Kebutuhan

Pada tahapan elisitasi kebutuhan, pengembang bekerjasama dengan pengguna akhir dalam sistem untuk menentukan lingkup domain dari sistem, fungsi apa saja yang disediakan oleh sistem, kinerja yang diperlukan oleh sistem, batasan pada sistem dan sebagainya. Tahap elisitasi dilakukan dengan menggunakan metode wawancara sehingga peneliti bisa berinteraksi dengan bertanya secara langsung dengan *stakeholder* yang terlibat untuk bisa merumuskan kebutuhan sistem. Berikut Tabel 4.2 menunjukkan hasil temuan kebutuhan dari tahap elisitasi

**Tabel 4.2 Temuan Kebutuhan**

No.	Temuan Kebutuhan
1.	Data klien bisa disimpan secara teratur, bisa diakses kapan saja dan dimana saja tanpa harus mendatangi staf yang bersangkutan untuk mencari berkas data klien
2.	Pimpinan bisa memantau aktivitas konsultasi harian yang dilakukan oleh staf dan klien untuk mengetahui bahasan atau permasalahan yang dikonsultasikan
3.	Staf bisa membuat tanda terima dokumen dan riwayat penyerahan data dokumen yang pernah diserahkan bisa diketahui
4.	Staf bisa melaporkan pekerjaan perpajakan bulanan atau perpajakan tahunan klien kepada pimpinan yang telah dikerjakan untuk divalidasi
5.	Pimpinan dan staf yang bersangkutan dengan klien bisa mencetak data klien yang dapat digunakan untuk keperluan kantor
6.	Pimpinan dan staf yang bersangkutan dengan klien bisa mencetak riwayat aktivitas konsultasi harian yang dapat digunakan untuk keperluan kantor
7.	Pimpinan dan staf yang bersangkutan dengan klien bisa mencetak riwayat aktivitas perpajakan bulanan atau perpajakan tahunan klien yang dapat digunakan untuk keperluan kantor
8.	Pimpinan dan staf yang bersangkutan dengan klien bisa mencetak riwayat serah terima dokumen yang dapat digunakan untuk keperluan kantor
9.	Pimpinan mendapatkan pemberitahuan jika ada pekerjaan perpajakan bulanan atau perpajakan tahunan klien dari staf yang akan divalidasi

Tabel 4.2 Temuan Kebutuhan (lanjutan)

No.	Temuan Kebutuhan
10.	Pimpinan bisa memantau pekerjaan perpajakan bulanan atau perpajakan tahunan klien yang dikerjakan oleh setiap staf
11.	Pimpinan dan staf yang bersangkutan dengan klien mendapatkan pemberitahuan jika ada pekerjaan perpajakan bulanan atau perpajakan tahunan klien yang masih belum terbayarkan atau dilaporkan pada batas waktu tertentu
12.	Pimpinan perlu mengecek dan memberikan persetujuan terkait proposal yang dibuat oleh staf, apakah sudah sesuai dengan kesepakatan antara klien dan pihak kantor

#### 4.1.3 Identifikasi Aktor

Identifikasi aktor merupakan orang atau sistem lain yang terlibat dalam sistem *monitoring* pekerjaan konsultasi pajak secara langsung. Pada Tabel 4.3 berikut menjelaskan tentang identifikasi aktor yang terlibat dengan sistem informasi *monitoring* pekerjaan kantor konsultan pajak.

Tabel 4.3 Identifikasi Aktor

No	Aktor	Deskripsi
1.	Staf	Seluruh staf yang berhubungan langsung dengan klien dan terlibat dalam tanggung jawab konsultasi harian, pekerjaan perpajakan dan serah terima dokumen klien
2.	Pimpinan	Orang yang memiliki otorisasi penuh dan mengelola pengguna yang memakai sistem dan memonitor pekerjaan yang dilakukan oleh staf
3.	Pengguna	Terdiri dari staf dan pimpinan

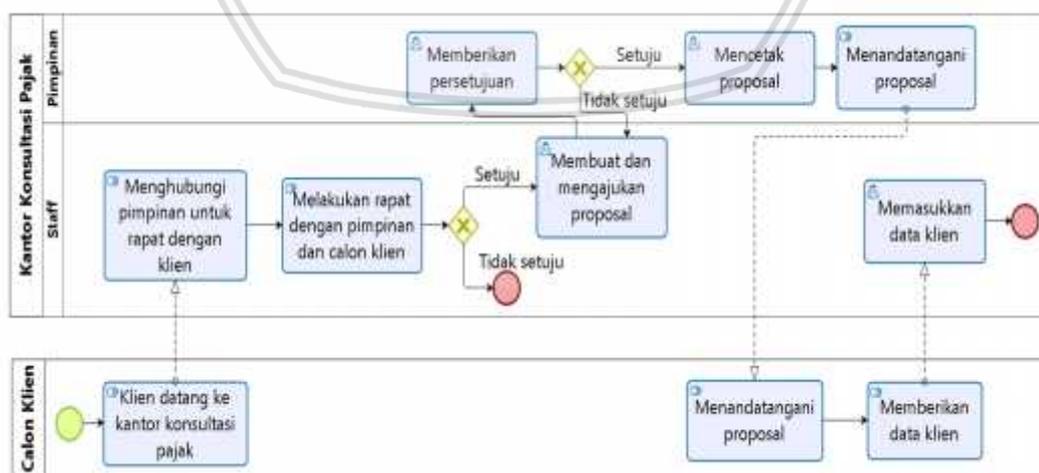
#### 4.1.4 Pemodelan Proses Bisnis TO-BE

Berdasarkan hasil analisis permasalahan yang telah dibahas sebelumnya, maka proses bisnis dalam menangani *monitoring* mengalami perubahan. Kode proses bisnis diwakilkan dengan kode PB-KKP-1 secara berurutan, PB merupakan singkatan dari Proses Bisnis, KKP singkatan dari Kantor Konsultan Pajak. Solusi terhadap permasalahan *monitoring* pekerjaan kantor konsultan pajak dijelaskan dalam Tabel 4.4 berikut

Tabel 4.4 Daftar Proses Bisnis To-Be

Kode Proses Bisnis	Nama	Deskripsi
PB-KKP-1	Proses Bisnis Usulan Registrasi Klien	Proses bisnis dari mulai registrasi calon klien sampai menjadi klien kantor konsultan pajak I Gede Arianta melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
PB-KKP-2	Proses Bisnis Usulan Konsultasi	Proses bisnis dari kegiatan konsultasi yang dilakukan oleh staf dan klien dipantau oleh pimpinan melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
PB-KKP-3	Proses Bisnis Usulan <i>Monitoring</i> Pajak Bulanan	Proses bisnis dari proses pelaporan dan validasi perpajakan bulanan klien yang dilakukan oleh staf kepada pimpinan melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
PB-KKP-4	Proses Bisnis Usulan <i>Monitoring</i> Pajak Tahunan	Proses bisnis dari proses pelaporan dan validasi perpajakan tahunan klien yang dilakukan oleh staf kepada pimpinan melalui Sistem Informasi <i>Monitoring</i> Pekerjaan

Berdasarkan Tabel 4.5 yang telah dijelaskan pada daftar proses bisnis to-be, bahwa telah diidentifikasi terdapat empat proses bisnis to-be yang perlu dimodelkan dalam diagram BPMN. Pemodelan proses bisnis PB-KKP-1 ditunjukkan pada Gambar 4.5 berikut:



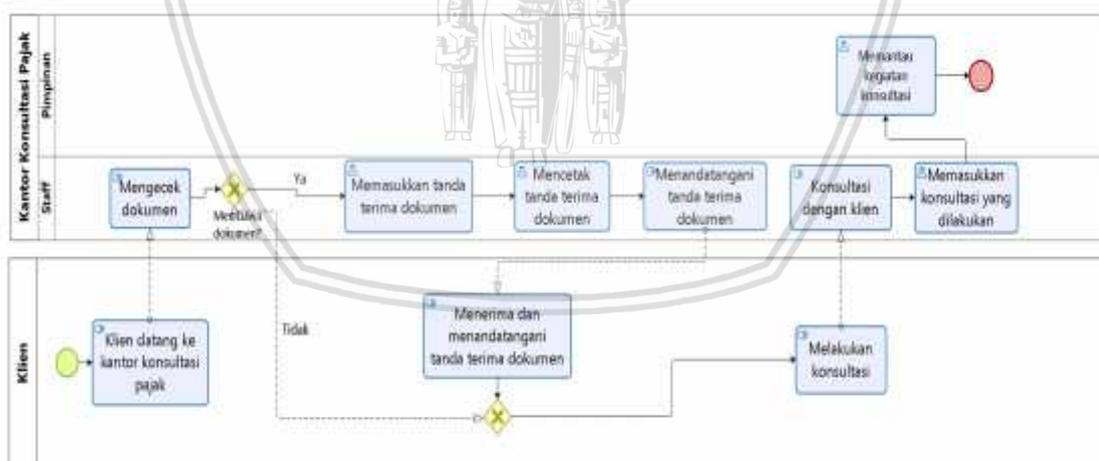
Gambar 4.5 Proses Bisnis Usulan Registrasi Klien



Penjabaran proses bisnis usulan registrasi klien pada Gambar 4.5 sebagai berikut:

1. Calon klien datang ke kantor konsultan pajak I Gede Arianta
2. Staf menghubungi pimpinan agar dilakukan rapat dengan calon klien
3. Calon klien melakukan rapat dengan pimpinan dan juga staf yang nanti akan bertanggung jawab kepada klien tersebut
4. Setelah calon klien dan pihak kantor setuju maka staff membuat proposal sebagai kontrak kedua belah pihak, yaitu pihak calon klien dan pihak kantor konsultan pajak
5. Pimpinan mengecek dan memberi persetujuan lalu mencetak dan menandatangani proposal kemudian diberikan kepada staf agar diserahkan kepada klien
6. Calon klien menandatangani proposal
7. Staf meminta data terkait data klien yang dibutuhkan oleh kantor konsultan pajak
8. Klien memberikan data yang diminta oleh staf
9. Staf menerima data yang diberikan oleh klien dan memasukkan data kesistem

Selanjutnya pada pemodelan proses bisnis PB-KKP-2 ditunjukkan pada Gambar 4.6 berikut:



**Gambar 4.6 Proses Bisnis Usulan Konsultasi**

Penjabaran proses bisnis usulan konsultasi pada Gambar 4.6 sebagai berikut:

1. Klien datang ke kantor konsultan pajak I Gede Arianta untuk berkonsultasi
2. Apabila klien membawa dokumen sebagai kebutuhan perpajakan atau sejenisnya maka diserahkan kepada staf, apabila membawa dokumen maka langsung berkonsultasi dengan staf

3. Apabila klien menyerahkan dokumen maka staf menerima dokumen tersebut dan membuat tanda terima dokumen serta data tanda terima dokumen nantinya akan disimpan kedalam sistem
4. Staf menandatangani tanda terima dokumen lalu menyerahkannya kepada klien
5. Klien menerima tanda terima dokumen yang diberikan oleh staf
6. Staf melakukan konsultasi dengan klien
7. Setelah melakukan konsultasi, staff mereview dan memasukkan bahasan konsultasi kedalam sistem
8. Pimpinan memantau kegiatan konsultasi yang dilakukan oleh staf

Kemudian pada pemodelan proses bisnis PB-KKP-3 ditunjukkan pada Gambar 4.7 dan pemodelan proses bisnis PB-KKP-4 ditunjukkan pada Gambar 4.8 berikut:





Penjabaran proses bisnis usulan *monitoring* pajak bulanan pada Gambar 4.7 sebagai berikut:

1. Staf meminta data yang dibutuhkan untuk melakukan pembayaran pajak bulanan
2. Klien memberikan data kepada staf
3. Staf melakukan proses perhitungan pajak klien
4. Staf melakukan penyusunan surat pemberitahuan (SPT) dan juga meminta tanda tangan klien untuk arsip
5. Klien menandatangani surat pemberitahuan (SPT)
6. Staf menginputkan data berupa massa pajak ke sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta
7. Staf membuat surat setoran elektronik (SSE) di sistem DJP Online
8. DJP online memproses SSE dan menghasilkan ID billing untuk pembayaran
9. Staf memberikan ID billing kepada klien untuk dilakukan pembayaran
10. Klien membayarkan pajak sesuai dengan ID billing yang diberikan, lalu mendapatkan bukti bayar (NTPN). Setelah itu menyerahkan bukti bayar kepada staf
11. Staf menerima bukti bayar dari klien, lalu staf merubah data *monitoring* bulanan klien yang berupa data pembayaran pajak
12. Staf *request* validasi terkait pembayaran perpajakan yang ditujukan kepada pimpinan
13. Pimpinan mengecek dan melakukan validasi dari pekerjaan staf
14. Memasukkan data perpajakan dan bukti bayar ke sistem E-SPT Online
15. Staf melakukan ekspor data (tipe data .csv) yang nantinya digunakan untuk melakukan pelaporan di DJP Online
16. Staf melakukan pelaporan pajak dengan memasukkan data ke DJP Online
17. DJP Online menerima lalu memproses data lapor, setelah itu menghasilkan bukti lapor
18. Staf menerima bukti lapor, lalu staf merubah data *monitoring* bulanan klien yang berupa data pelaporan perpajakan
19. Staf *request* validasi terkait pelaporan perpajakan yang ditujukan kepada pimpinan
20. Pimpinan mengecek dan melakukan validasi dari pekerjaan staf
21. Staf menyerahkan bukti lapor ke klien
22. Klien menerima bukti lapor dari staf



Penjabaran proses bisnis usulan *monitoring* pajak tahunan pada Gambar 4.8 sebagai berikut:

1. Staf meminta data yang dibutuhkan untuk melakukan pembayaran pajak tahunan
2. Klien memberikan data kepada staf
3. Staf melakukan proses perhitungan pajak klien
4. Staf melakukan posting jurnal perpajakan tahunan masukkan file ke sistem dan melakukan *request* validasi posting jurnal
5. Pimpinan mengecek dan memvalidasi terkait posting jurnal
6. Staf melakukan penyusunan surat pemberitahuan (SPT) tahunan dan juga meminta tanda tangan klien untuk arsip
7. Klien menandatangani surat pemberitahuan (SPT) tahunan
8. Staf menginputkan data berupa massa pajak ke sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta
9. Staf membuat surat setoran elektronik (SSE) di sistem DJP Online
10. DJP online memproses SSE dan menghasilkan ID billing untuk pembayaran
11. Staf memberikan ID billing kepada klien untuk dilakukan pembayaran
12. Klien membayarkan pajak sesuai dengan ID billing yang diberikan, lalu mendapatkan bukti bayar (NTPN). Setelah itu menyerahkan bukti bayar kepada staf
13. Staf menerima bukti bayar dari klien, lalu staf merubah data *monitoring* bulanan klien yang berupa data pembayaran pajak
14. Staf *request* validasi terkait pembayaran perpajakan yang ditujukan kepada pimpinan
15. Pimpinan mengecek dan melakukan validasi dari pekerjaan staf
16. Memasukkan data perpajakan dan bukti bayar ke sistem E-SPT Online
17. Staf melakukan ekspor data (tipe data .csv) yang nantinya digunakan untuk melakukan pelaporan di DJP Online
18. Staf melakukan pelaporan pajak dengan memasukkan data ke DJP Online
19. DJP Online memproses data lapor, lalu menghasilkan bukti lapor
20. Staf menerima bukti lapor, lalu staf merubah data *monitoring* bulanan klien yang berupa data pelaporan perpajakan
21. Staf *request* validasi terkait pelaporan perpajakan yang ditujukan kepada pimpinan
22. Pimpinan mengecek dan melakukan validasi dari pekerjaan staf

23. Staf menyerahkan bukti lapor ke klien

24. Klien menerima bukti lapor dari staf

Berdasarkan proses bisnis pada PB-KKP-1 yang ditunjukkan dalam Gambar 4.5, proses bisnis PB-KKP-2 yang ditunjukkan dalam Gambar 4.6, proses bisnis PB-KKP-3 yang ditunjukkan dalam Gambar 4.7 dan proses bisnis PB-KKP-4 yang ditunjukkan dalam Gambar 4.8, selanjutnya akan dijabarkan tentang aktivitas-aktivitas yang mendukung dalam penggunaan Sistem Informasi *Monitoring Pekerjaan* ke dalam daftar tabel aktivitas proses bisnis. Berikut Tabel 4.5 menunjukkan pemaparan daftar aktivitas proses bisnis PB-KKP-1

**Tabel 4.5 Daftar Aktivitas Proses Bisnis PB-KKP-1**

Aktor	Kode Aktivitas	Aktivitas	Keterangan
Pengguna	APB1-P1-1	Memasukkan data klien ke sistem	Pengguna menyimpan data klien kedalam sistem melalui Sistem Informasi <i>Monitoring Pekerjaan</i>
Staf	APB1-P2-1	Membuat dan mengajukan proposal	Staf membuat dan menyimpan proposal berkaitan dengan perjanjian antara klien dan kantor kemudian meminta persetujuan proposal kepada pimpinan melalui Sistem Informasi <i>Monitoring Pekerjaan</i> ,
Pimpinan	APB1-P3-1	Memberikan persetujuan	Pimpinan bisa memberikan persetujuan terkait proposal yang dibuat oleh staf
	APB1-P3-2	Mencetak proposal	Pimpinan mencetak dokumen proposal untuk ditandatangani bersama dengan klien

Kemudian pada daftar aktivitas proses bisnis PB-KKP-2 dijelaskan pada Tabel 4.6 berikut

**Tabel 4.6 Daftar Aktivitas Proses Bisnis PB-KKP-2**

Aktor	Kode Aktivitas	Aktivitas	Keterangan
Staf	APB2-P2-1	Memasukkan tanda terima dokumen	Staf memasukkan tanda terima dokumen beserta data yang diperlukan melalui Sistem Informasi <i>Monitoring Pekerjaan</i>

Tabel 4.6 Daftar Aktivitas Proses Bisnis PB-KKP-2 (lanjutan)

Aktor	Kode Aktivitas	Aktivitas	Keterangan
	APB2-P2-2	Mencetak tanda terima dokumen	Staf mencetak tanda terima dokumen yang akan diserahkan kepada klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
	APB2-P2-3	Memasukkan konsultasi yang dilakukan	Staf memasukkan bahasan konsultasi yang dilakukan dengan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
Pimpinan	APB2-P3-1	Memantau kegiatan konsultasi	Pimpinan melihat data kegiatan konsultasi yang dilakukan oleh setiap staf dan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan

Kemudian pada daftar aktivitas proses bisnis PB-KKP-3 dijelaskan pada Tabel 4.7 berikut

Tabel 4.7 Daftar Aktivitas Proses Bisnis PB-KKP-3

Aktor	Kode Aktivitas	Aktivitas	Keterangan
	APB3-P2-1	Memasukkan ke sistem(massa pajak)	Staf memasukkan massa pajak data perpajakan bulanan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
Staf	APB3-P2-2	<i>Update</i> data ke sistem memuat bukti bayar	Staf update data yang memuat bukti pembayaran pajak bulanan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
	APB3-P2-3	<i>Request</i> validasi bukti bayar	Staf melakukan <i>request</i> untuk validasi kepada pimpinan berkaitan dengan bukti pembayaran pajak bulanan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan

Tabel 4.7 Daftar Aktivitas Proses Bisnis PB-KKP-3 (lanjutan)

Aktor	Kode Aktivitas	Aktivitas	Keterangan
	APB3-P2-4	Update data ke sistem memuat bukti lapor	Staf update data yang yang memuat bukti lapor pajak bulanan klien melalui Sistem Informasi <i>Monitoring Pekerjaan</i>
	APB3-P2-5	Request validasi bukti lapor	Staf melakukan <i>request</i> untuk validasi kepada pimpinan berkaitan dengan bukti lapor pajak bulanan klien melalui Sistem Informasi <i>Monitoring Pekerjaan</i>
Pimpinan	APB3-P3-1	Mengecek dan memvalidasi bukti bayar	Pimpinan melakukan pengecekan dan memvalidasi pekerjaan staf yang sudah melakukan <i>request</i> untuk validasi dengan memuat bukti pembayaran pajak bulanan klien melalui Sistem Informasi <i>Monitoring Pekerjaan</i>
	APB3-P3-2	Mengecek dan memvalidasi bukti lapor	Pimpinan melakukan pengecekan dan memvalidasi pekerjaan staf yang sudah melakukan <i>request</i> untuk validasi dengan memuat bukti lapor pajak bulanan klien melalui Sistem Informasi <i>Monitoring Pekerjaan</i>

Kemudian pada daftar aktivitas proses bisnis PB-KKP-4 dijelaskan pada Tabel 4.8 berikut

Tabel 4.8 Daftar Aktivitas Proses Bisnis PB-KKP-4

Aktor	Kode Aktivitas	Aktivitas	Keterangan
Staf	APB4-P2-1	Memasukkan ke sistem(massa pajak)	Staf memasukkan massa pajak data perpajakan tahunan klien melalui Sistem Informasi <i>Monitoring Pekerjaan</i>

Tabel 4.8 Daftar Aktivitas Proses Bisnis PB-KKP-4 (lanjutan)

Aktor	Kode Aktivitas	Aktivitas	Keterangan
	APB4-P2-2	<i>Update</i> data ke sistem memuat bukti bayar	Staf update data yang yang memuat bukti pembayaran pajak tahunan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
	APB4-P2-3	<i>Request</i> validasi bukti bayar	Staf melakukan <i>request</i> untuk validasi kepada pimpinan berkaitan dengan bukti pembayaran pajak tahunan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
	APB4-P2-4	<i>Update</i> data ke sistem memuat bukti lapor	Staf update data yang yang memuat bukti lapor pajak tahunan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
	APB4-P2-5	<i>Request</i> validasi bukti lapor	Staf melakukan <i>request</i> untuk validasi kepada pimpinan berkaitan dengan bukti lapor pajak tahunan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
	APB4-P2-4	<i>Update</i> data ke sistem memuat posting jurnal	Staf update data yang yang memuat posting jurnal pajak tahunan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
	APB4-P2-7	<i>Request</i> validasi posting jurnal	Staf melakukan <i>request</i> untuk validasi kepada pimpinan berkaitan dengan posting jurnal pajak tahunan klien melalui Sistem Informasi <i>Monitoring</i> Pekerjaan
Pimpinan	APB4-P3-1	Mengecek dan memvalidasi bukti bayar	Pimpinan melakukan pengecekan dan memvalidasi pekerjaan staf yang sudah melakukan <i>request</i> untuk validasi dengan memuat bukti pembayaran pajak tahunan

Tabel 4.8 Daftar Aktivitas Proses Bisnis PB-KKP-4 (lanjutan)

Aktor	Kode Aktivitas	Aktivitas	Keterangan
			klien melalui Sistem Informasi <i>Monitoring Pekerjaan</i>
	APB4-P3-2	Mengecek dan memvalidasi bukti lapor	Pimpinan melakukan pengecekan dan memvalidasi pekerjaan staf yang sudah melakukan <i>request</i> untuk validasi dengan memuat bukti lapor pajak tahunan klien melalui Sistem Informasi <i>Monitoring Pekerjaan</i>
	APB4-P3-3	Mengecek dan memvalidasi posting jurnal	Pimpinan melakukan pengecekan dan memvalidasi pekerjaan staf yang sudah melakukan <i>request</i> untuk validasi dengan memuat bukti lapor pajak tahunan klien melalui Sistem Informasi <i>Monitoring Pekerjaan</i>

#### 4.1.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dalam pengembangan sistem informasi *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta dilakukan pengumpulan data dengan cara observasi dan wawancara langsung dengan staf dan pemilik perusahaan. Kode kebutuhan fungsional diwakilkan dengan kode KF-KKP-P1-1, KF-KKP-P2-1 dan KF-KKP-P3-1 secara berurutan. KF merupakan singkatan dari Kebutuhan Fungsional, KKP singkatan dari Kantor Konsultan Pajak, P1 mewakili aktor Pengguna, P2 mewakili aktor Staf dan P3 mewakili aktor Pimpinan. Berikut kebutuhan fungsional dijelaskan pada Tabel 4.9 dengan aktor pengguna

Tabel 4.9 Kebutuhan Fungsional Dengan Aktor Pengguna

Kode Kebutuhan	Nama Fungsi	Kebutuhan	Kode Aktivitas
KF-KKP-P1-1	Login	Sistem dapat melakukan login dengan menyediakan halaman login  <b>Spesifikasi Kebutuhan:</b> <ul style="list-style-type: none"> <li>Pengguna melakukan login dengan menggunakan</li> </ul>	-

Tabel 4.9 Kebutuhan Fungsional Dengan Aktor Pengguna (lanjutan)

Kode Kebutuhan	Nama Fungsi	Kebutuhan	Kode Aktivitas
		<p><i>username</i> dan <i>password</i>. (KF-KKP- P1-1-1)</p> <ul style="list-style-type: none"> <li>• Sistem menampilkan peringatan apabila <i>username</i> dan <i>password</i> ada yang kosong. (KF-KKP- P1-1-2)</li> <li>• Sistem menampilkan pemberitahuan agar memasukkan <i>username</i> dan <i>password</i> dengan benar apabila <i>username</i> dan <i>password</i> tidak sesuai. (KF-KKP- P1-1-3)</li> </ul>	
KF-KKP-P1-2	Kelola data klien	<p>Sistem dapat mengelola pendataan data klien dan mencetak data klien yang tersimpan</p> <p><b>Spesifikasi Kebutuhan:</b></p> <p><b>Menambah data klien</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk memasukkan data baru. Form memuat <i>efin</i>, nama klien, kode klien, nama direktur/pimpinan, <i>npwp</i>, jenis usaha, klu usaha, alamat, email, nomor telepon, tanggal lahir direktur/pimpinan, catatan khusus. (KF-KKP- P1-2-1)</li> </ul> <p><b>Mengubah data klien</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk mengubah data klien yang sebelumnya telah disimpan. (KF-KKP- P1-2-2)</li> </ul> <p><b>Menghapus data klien</b></p>	APB1-P1-1,



Tabel 4.9 Kebutuhan Fungsional Dengan Aktor Pengguna (lanjutan)

Kode Kebutuhan	Nama Fungsi	Kebutuhan	Kode Aktivitas
		<ul style="list-style-type: none"> <li>Sistem menyediakan fitur untuk menghapus data klien yang sebelumnya telah disimpan. (KF-KKP- P1-2-3)</li> </ul> <p><b>Mencetak data klien</b></p> <ul style="list-style-type: none"> <li>Sistem menyediakan fitur untuk mencetak data klien yang memuat efin, nama klien, kode klien, nama direktur/pimpinan, npwp, jenis usaha, klu usaha, alamat, email, nomor telepon, tanggal lahir direktur/pimpinan, staf yang melayani, catatan khusus. (KF-KKP- P1-2-4)</li> </ul>	
KF-KKP-P1-3	Kelola <i>monitoring</i> aktivitas klien	<p>Sistem dapat mengelola data konsultasi dengan klien yang dilakukan oleh staf dan mencetak data konsultasi yang tersimpan.</p> <p><b>Spesifikasi Kebutuhan:</b></p> <p><b>Menambah data <i>monitoring</i> aktivitas</b></p> <ul style="list-style-type: none"> <li>Sistem menampilkan form untuk memasukkan <i>monitoring</i> aktivitas klien baru. Form memuat tanggal, aktivitas dengan klien, staff yang melayani, keterangan. (KF-KKP-P1-3-1)</li> </ul> <p><b>Mengubah data <i>monitoring</i> aktivitas</b></p> <ul style="list-style-type: none"> <li>Sistem menampilkan form untuk mengubah data memasukkan <i>monitoring</i> aktivitas klien yang</li> </ul>	APB2-P2-3, APB2-P3-1

Tabel 4.9 Kebutuhan Fungsional Dengan Aktor Pengguna (lanjutan)

Kode Kebutuhan	Nama Fungsi	Kebutuhan	Kode Aktivitas
		<p>sebelumnya telah disimpan. (KF-KKP-P1-3-2)</p> <p><b>Menghapus data <i>monitoring</i> aktivitas</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur untuk menghapus <i>monitoring</i> aktivitas klien yang sebelumnya telah disimpan. (KF-KKP-P1-3-3)</li> </ul> <p><b>Mencetak data <i>monitoring</i> aktivitas</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur untuk mencetak data <i>monitoring</i> aktivitas klien yang memuat tanggal, aktivitas dengan klien, staff yang melayani, keterangan. (KF-KKP- P1-3-4)</li> </ul>	
KF-KKP-P1-4	Kelola <i>monitoring</i> bulanan klien	<p>Sistem dapat mengelola data <i>monitoring</i> perpajakan bulanan klien dan mencetak data <i>monitoring</i> perpajakan bulanan klien yang tersimpan</p> <p><b>Spesifikasi Kebutuhan:</b></p> <p>Staf bisa mengubah dan menghapus data <i>monitoring</i> bulanan yang tersimpan sebelum tervalidasi oleh pimpinan.</p> <p><b>Menambah data <i>monitoring</i> bulanan</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk memasukkan <i>monitoring</i> bulanan klien baru. Form memuat massa pajak, tanggal bayar bank, jumlah, ntpn, tanggal lapor kpp, no. bukti lapor, foto</li> </ul>	APB3-P2-1, APB3-P2-2, APB3-P2-4

Tabel 4.9 Kebutuhan Fungsional Dengan Aktor Pengguna (lanjutan)

Kode Kebutuhan	Nama Fungsi	Kebutuhan	Kode Aktivitas
		<p>bukti bayar, foto bukti lapor. (KF-KKP- P1-4-1)</p> <p><b>Mengubah data <i>monitoring</i> bulanan</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk mengubah data memasukkan <i>monitoring</i> bulanan klien yang sebelumnya telah disimpan. (KF-KKP- P1-4-2)</li> </ul> <p><b>Menghapus data <i>monitoring</i> bulanan</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur untuk menghapus <i>monitoring</i> bulanan klien yang sebelumnya telah disimpan. (KF-KKP- P1-4-3)</li> </ul> <p><b>Mencetak data <i>monitoring</i> bulanan</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur untuk mencetak data <i>monitoring</i> bulanan klien yang memuat massa pajak, tanggal bayar bank, jumlah, ntpn, status, tanggal lapor kpp, no. bukti lapor, foto bukti bayar, foto bukti lapor, status. (KF-KKP- P1-4-1)</li> </ul>	
KF-KKP-P1-5	Kelola tanda terima dokumen	<p>Sistem dapat melihat dan menghapus data dokumen tanda terima yang sebelumnya sudah tersimpan dan mencetak riwayat data tanda terima sebelumnya.</p> <p><b>Spesifikasi Kebutuhan:</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur untuk tanda terima dokumen klien yang</li> </ul>	-



Tabel 4.9 Kebutuhan Fungsional Dengan Aktor Pengguna (lanjutan)

Kode Kebutuhan	Nama Fungsi	Kebutuhan	Kode Aktivitas
		<p>sebelumnya telah disimpan. (KF-KKP- P1-5-1)</p>	
KF-KKP-P1-6	<p>Kelola <i>monitoring</i> tahunan klien</p>	<p>Sistem dapat mengelola data <i>monitoring</i> perpajakan tahunan klien dan mencetak data <i>monitoring</i> perpajakan tahunan klien yang tersimpan</p> <p><b>Spesifikasi Kebutuhan:</b></p> <p>Staf bisa mengubah dan menghapus data <i>monitoring</i> tahunan yang tersimpan sebelum tervalidasi oleh pimpinan.</p> <p><b>Menambah data <i>monitoring</i> tahunan</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk memasukkan <i>monitoring</i> tahunan klien baru. Form memuat jenis pekerjaan, tahun, tanggal bayar pajak, jumlah, ntpn, tanggal lapor spt, no bukti lapor. (KF-KKP-P1-6-1)</li> </ul> <p><b>Mengubah data <i>monitoring</i> tahunan</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk mengubah data memasukkan <i>monitoring</i> tahunan klien yang sebelumnya telah disimpan. (KF-KKP-P1-6-2)</li> </ul> <p><b>Menghapus data <i>monitoring</i> tahunan</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur untuk menghapus memasukkan <i>monitoring</i> tahunan klien yang</li> </ul>	<p>APB4-P2-1, APB4-P2-2, APB4-P2-4, APB4-P2-6</p>



Tabel 4.9 Kebutuhan Fungsional Dengan Aktor Pengguna (lanjutan)

Kode Kebutuhan	Nama Fungsi	Kebutuhan	Kode Aktivitas
		sebelumnya telah disimpan. (KF-KKP-P1-6-3)	
KF-KKP-P1-7	Kelola proposal	<p>Sistem dapat mengelola proposal yang sudah tersimpan</p> <p><b>Spesifikasi Kebutuhan:</b></p> <p>Staf bisa mengubah proposal ketika belum mendapat persetujuan pimpinan. Staf bisa mencetak proposal ketika sudah mendapat persetujuan dari pimpinan.</p> <p><b>Mengubah proposal</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk mengubah data proposal yang sebelumnya telah disimpan. (KF-KKP-P1-7-1)</li> </ul> <p><b>Mencetak proposal</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur untuk mencetak data proposal . (KF-KKP-P1-7-1)</li> </ul>	APB1-P2-1, APB1-P3-2
KF-KKP-P1-8	Logout	<p>Sistem dapat keluar dari sistem yang berguna untuk keamanan informasi.</p> <p><b>Spesifikasi Kebutuhan:</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur logout untuk keluar dari sistem. (KF-KKP-P1-8-1)</li> <li>• Sistem menampilkan halaman login ketika aktor sudah logout. (KF-KKP-P1-8-2)</li> </ul>	-

Berikut Tabel 4.10 merupakan penjabaran dari kebutuhan fungsional sistem dengan aktor staf

Tabel 4.10 Kebutuhan Fungsional Dengan Aktor Staf

Kode	Nama Fungsi	Kebutuhan	Kode Aktivitas
KF-KKP-P2-1	<i>Request</i> validasi <i>monitoring</i> bulanan klien	Sistem dapat melakukan validasi dari staf kepada pimpinan atas pekerjaan perpajakan <i>monitoring</i> bulanan klien <b>Spesifikasi Kebutuhan:</b> <ul style="list-style-type: none"> <li>Sistem menyediakan fitur validasi yang berkaitan dengan perpajakan bulanan klien yang dikerjakan oleh staf berkaitan dengan bukti bayar dan bukti lapor sehingga pimpinan bisa mendapatkan pemberitahuan tentang pekerjaan perpajakan bulanan klien. (KF-KKP-P2-1-1)</li> </ul>	APB3-P2-3
KF-KKP-P2-2	Tambah tanda terima dokumen	Sistem dapat menyimpan data dokumen yang diberikan oleh klien dan sekaligus membuat form tanda terima <b>Spesifikasi Kebutuhan:</b> <ul style="list-style-type: none"> <li>Sistem menyediakan fitur untuk membuat dan mencetak tanda terima dokumen yang memuat nama dokumen, tahun, keterangan, yang menyerahkan, yang menerima. (KF-KKP-P2-2-1)</li> <li>Sistem menyimpan tanda terima dokumen yang dibuat kedalam database. (KF-KKP-P2-2-2)</li> </ul>	APB2-P2-1, APB2-P2-2, APB2-S-1
KF-KKP-P2-3	<i>Request</i> validasi <i>monitoring</i> tahunan klien	Sistem dapat melakukan validasi dari staf kepada pimpinan atas pekerjaan perpajakan <i>monitoring</i> tahunan klien <b>Spesifikasi Kebutuhan:</b> <ul style="list-style-type: none"> <li>Sistem menyediakan fitur validasi yang berkaitan dengan perpajakan bulanan klien yang dikerjakan oleh staf berkaitan dengan posting jurnal, bukti</li> </ul>	APB4-P2-3, APB4-P2-5, APB4-P2-7

**Tabel 4.10 Kebutuhan Fungsional Dengan Aktor Staf (lanjutan)**

		bayar dan bukti lapor sehingga pimpinan bisa mendapatkan pemberitahuan tentang pekerjaan perpajakan bulanan klien. (KF-KKP-P2-3-1)	
KF-KKP-P2-4	<i>Request</i> persetujuan proposal	<p>Sistem dapat meminta persetujuan terkait proposal perjanjian klien dari staf kepada pimpinan</p> <p><b>Spesifikasi Kebutuhan:</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fitur persetujuan yang berkaitan dengan pembuatan proposal perjanjian klien yang dikerjakan oleh staf sehingga pimpinan bisa mengecek isi dari proposal perjanjian klien. (KF-KKP-P2-4-1)</li> </ul>	APB1-P2-1

Berikut Tabel 4.11 merupakan penjabaran dari kebutuhan fungsional sistem dengan aktor pimpinan

**Tabel 4.11 Kebutuhan Fungsional Dengan Aktor Pimpinan**

Kode	Nama Fungsi	Kebutuhan	Kode Aktivitas
KF-KKP-P3-1	Kelola staf	<p>Sistem dapat mengelola otoritas kepada staf yang menggunakan sistem</p> <p><b>Spesifikasi Kebutuhan:</b></p> <p><b>Menambah staf</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk memasukkan data staf baru. Form memuat no staff, nama, password. (KF-KKP-P3-1-1)</li> </ul> <p><b>Mengubah staf</b></p> <ul style="list-style-type: none"> <li>• Sistem menampilkan form untuk mengubah data staf yang sebelumnya telah disimpan. (KF-KKP-P3-1-2)</li> </ul>	-



Tabel 4.11 Kebutuhan Fungsional Dengan Aktor Pimpinan (lanjutan)

Kode	Nama Fungsi	Kebutuhan	Kode Aktivitas
		<p><b>Menghapus staf</b></p> <ul style="list-style-type: none"> <li>Sistem menyediakan fitur untuk menghapus data staf yang sebelumnya telah disimpan. (KF-KKP-P3-1-3)</li> </ul>	
KF-KKP-P3-2	Validasi <i>monitoring</i> bulanan klien	<p>Sistem dapat melakukan proses validasi oleh pimpinan kepada staf atas pekerjaan perpajakan <i>monitoring</i> bulanan klien</p> <p><b>Spesifikasi Kebutuhan:</b></p> <ul style="list-style-type: none"> <li>Sistem menyediakan fitur pemberitahuan tentang pekerjaan perpajakan bulanan klien kemudian memvalidasi <i>request</i> dari staf ke pimpinan yang berkaitan dengan bukti bayar dan bukti lapor. (KF-KKP-P3-2-1)</li> </ul>	APB3-P3-1, APB3-P3-2, APB4-P3-3
KF-KKP-P3-3	Validasi <i>monitoring</i> tahunan klien	<p>Sistem dapat melakukan proses validasi oleh pimpinan kepada staf atas pekerjaan perpajakan <i>monitoring</i> tahunan klien</p> <p><b>Spesifikasi Kebutuhan:</b></p> <ul style="list-style-type: none"> <li>Sistem menyediakan fitur pemberitahuan tentang pekerjaan perpajakan tahunan klien kemudian memvalidasi <i>request</i> dari staf ke pimpinan yang berkaitan dengan posting jurnal, bukti bayar dan bukti lapor. (KF-KKP-P3-3-1)</li> </ul>	APB4-P3-1, APB4-P3-2
KF-KKP-P3-4	Persetujuan proposal	<p>Sistem dapat melakukan proses persetujuan oleh pimpinan kepada staf atas proposal perjanjian klien</p> <p><b>Spesifikasi Kebutuhan:</b></p> <ul style="list-style-type: none"> <li>Sistem menyediakan fitur pemberitahuan tentang</li> </ul>	APB1-P3-1

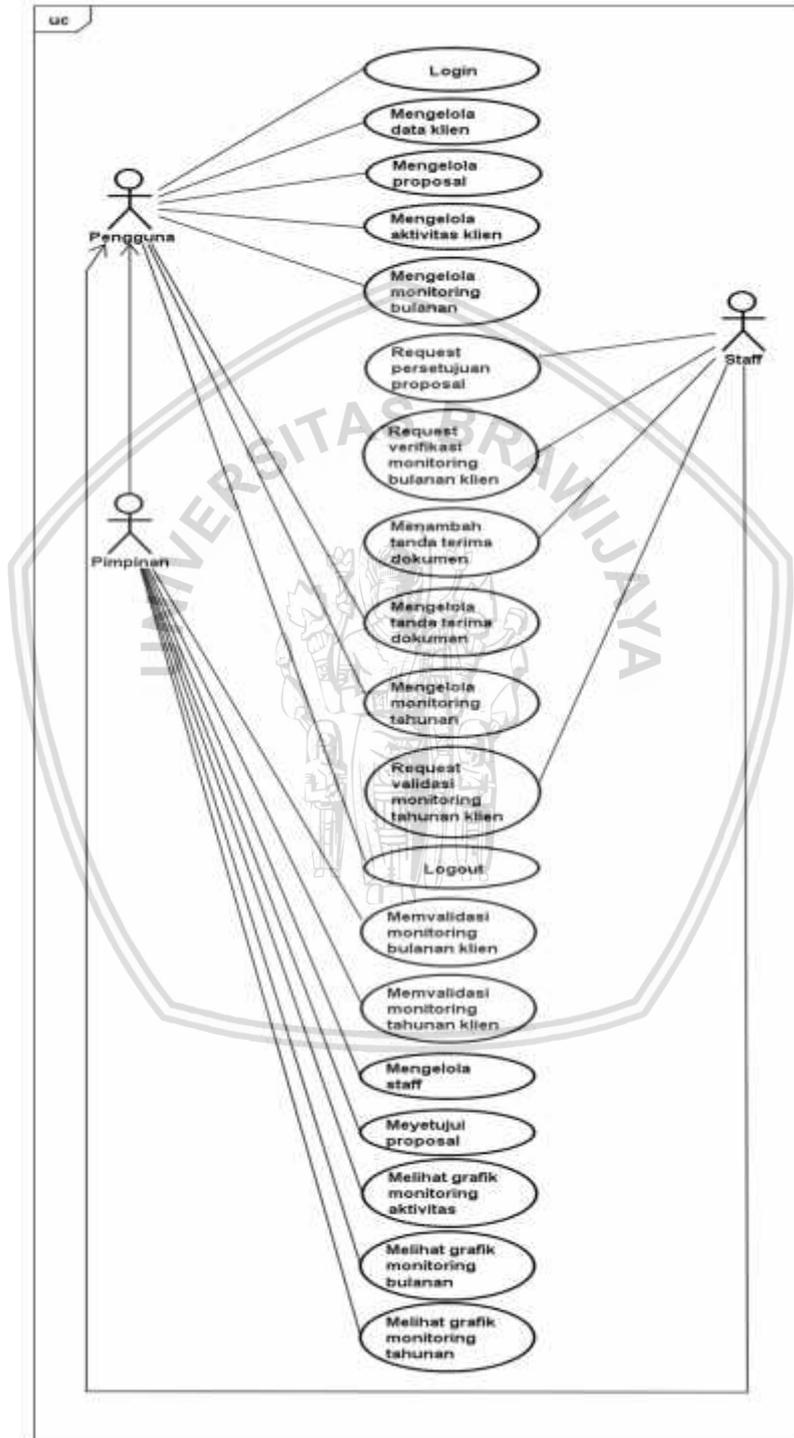
Tabel 4.11 Kebutuhan Fungsional Dengan Aktor Pimpinan (lanjutan)

Kode	Nama Fungsi	Kebutuhan	Kode Aktivitas
		permintaan persetujuan proposal perjanjian kemudian pimpinan bisa mengecek dan menyetujui proposal. (KF-KKP-P3-4-1)	
KF-KKP-P3-5	Melihat grafik aktivitas	Sistem dapat menampilkan grafik aktivitas yang sudah dilakukan oleh staf  <b>Spesifikasi Kebutuhan:</b> <ul style="list-style-type: none"> <li>Sistem menyediakan grafik yang bisa memantau secara umum jumlah aktivitas yang sudah dilakukan oleh staf</li> </ul>	-
KF-KKP-P3-6	Melihat grafik <i>monitoring</i> bulanan	Sistem dapat menampilkan grafik <i>monitoring</i> bulanan yang sudah dikerjakan oleh staf  <b>Spesifikasi Kebutuhan:</b> <ul style="list-style-type: none"> <li>Sistem menyediakan grafik yang bisa memantau secara umum jumlah perpajakan bulanan yang sudah dilakukan oleh staf</li> </ul>	-
KF-KKP-P3-7	Melihat grafik <i>monitoring</i> tahunan	Sistem dapat menampilkan grafik <i>monitoring</i> tahunan yang sudah dikerjakan oleh staf  <b>Spesifikasi Kebutuhan:</b> <ul style="list-style-type: none"> <li>Sistem menyediakan grafik yang bisa memantau secara umum jumlah perpajakan tahunan yang sudah dilakukan oleh staf</li> </ul>	-

## 4.2 Use Case Diagram

*Use case diagram* merepresentasikan siapa saja aktor yang terlibat dalam sistem beserta peran yang dijalankan dalam sistem, *use case diagram* mengacu terhadap kebutuhan fungsionalitas yang sudah dijabarkan sebelumnya. Setiap *use*

case yang ada akan dijelaskan secara mendetail dengan menggunakan *use case scenario*. Pada use case scenario dijabarkan dalam bentuk Tabel yang memuat nama use case, aktor yang terlibat, tujuan use case skenario alternatif dan juga skenario utama. Berikut Gambar 4.9 menunjukkan *use case diagram* pada sistem *monitoring* pekerjaan kantor konsultan pajak I Gede Arianta :



Gambar 4.9 Use Case Diagram *Monitoring* Pekerjaan Kantor konsultan pajak



### 4.3 Use Case Scenario

Pada use case scenario digunakan untuk menjelaskan lebih rinci kebutuhan fungsional yang sudah didefinisikan. Kode *use case scenario* diwakilkan dengan kode UC-KKP-1 secara berurutan, UC merupakan singkatan dari *Use Case*, KKP singkatan dari Kantor Konsultan Pajak. Berikut merupakan use case scenario dari masing-masing use case yang sudah digambarkan pada Gambar 4.9 sebelumnya :

#### 4.3.1 Use Case Scenario Login

Penjelasan tentang kebutuhan fungsional login akan dijelaskan dalam Tabel 4.13 berikut :

**Tabel 4.12 Use Case Scenario Login**

<b>Kode Use Case</b>	<b>UC-KKP-1</b>
<b>Nama</b>	Login
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Sistem dapat mencocokkan <i>username</i> dan <i>password</i> ke dalam <i>database</i> supaya pengguna bisa masuk kedalam sistem sebagai identitas yang telah diketahui agar menjaga keamanan informasi perangkat lunak
<b>Kondisi Awal</b>	Halaman login ditampilkan oleh sistem dan terdapat <i>form username</i> dan <i>password</i>
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memasukkan username dan password dengan benar lalu memilih tombol login	2. Sistem memberikan akses dan menampilkan dashboard awal
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memasukkan username atau password yang salah	2. Sistem akan menampilkan pemberitahuan berupa "Masukkan Username dan Password dengan Benar !".
<b>Kondisi Akhir</b>	Pengguna masuk ke dalam sistem dan halaman dashboard awal ditampilkan oleh sistem
<b>Kode Kebutuhan</b>	KF-KKP-P1-1

### 4.3.2 Use Case Scenario Mengelola Data Klien

Penjelasan tentang kebutuhan fungsional mengelola data klien pada skenario utama yaitu jika pengguna memilih fungsi tambah maka akan menjalankan skenario menambah data klien, jika pengguna memilih fungsi ubah maka akan menjalankan skenario mengubah data klien, jika pengguna memilih fungsi hapus maka akan menjalankan skenario menghapus data klien, jika pengguna memilih fungsi cetak maka akan menjalankan skenario mencetak data klien. Berikut Tabel 4.14 menjelaskan lebih detail kebutuhan fungsional mengelola data klien:

**Tabel 4.13 Use Case Scenario Mengelola Data Klien**

<b>Kode Use Case</b>	<b>UC-KKP-2</b>
<b>Nama</b>	Mengelola data klien
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Sistem dapat mengelola data klien berupa menambah, mengubah, menghapus atau mencetak data klien
<b>Kondisi Awal</b>	Halaman menu klien ditampilkan sistem
<b>Skenario Utama</b>	
<b>Menambah data klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih fungsi tambah klien  3. Pengguna mengisikan data klien, lalu pilih <i>submit</i>	2. Sistem menampilkan halaman form untuk memasukkan data klien baru  4. Sistem menyimpan data klien dan menampilkan halaman menu klien lalu menampilkan pesan bahwa data berhasil dimasukkan
<b>Mengubah data klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih fungsi ubah klien  3. Pengguna mengisi atau mengubah form klien lalu setelah selesai tekan submit	2. Sistem menampilkan halaman form ubah klien  4. Data klien yang mengalami perubahan akan disimpan dan menampilkan halaman menu klien lalu menampilkan pesan bahwa data berhasil diubah
<b>Menghapus data klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>



**Tabel 4.13 Use Case Scenario Mengelola Data Klien (lanjutan)**

1. Pengguna memilih fungsi hapus klien	2. Sistem melakukan proses penghapusan data klien yang dipilih oleh pengguna 3. Sistem melakukan proses penghapusan data klien yang dipilih oleh pengguna
<b>Mencetak data klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih daftar klien yang akan dicetak dan memilih fungsi cetak klien 3. pengguna menyesuaikan pengaturan cetak lalu pilih tombol "OK"	2. Sistem menampilkan halaman preview untuk mencetak data klien 4. Sistem mencetak data klien
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
a. Data yang dimasukkan kurang lengkap 1. Pengguna memilih menambah data klien atau mengubah data klien, lalu pengguna tidak mengisi halaman form tertentu yang disediakan secara lengkap	2. Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi
b. Batal cetak data klien 1. Pengguna memilih daftar klien yang akan dicetak dan memilih fungsi cetak klien 3. Pengguna membatalkan fungsi cetak dengan menekan tombol "Cancel"	2. Sistem menampilkan halaman preview untuk mencetak data 4. Sistem membatalkan cetak data
<b>Kondisi Akhir</b>	Data klien berhasil dikelola dengan sistem
<b>Kode Kebutuhan</b>	KF-KKP-P1-2



### 4.3.3 Use Case Scenario Mengelola Staf

Penjelasan tentang kebutuhan fungsional mengelola staf pada skenario utama yaitu jika pengguna memilih fungsi tambah maka akan menjalankan skenario menambah data staf, jika pengguna memilih fungsi ubah maka akan menjalankan skenario mengubah data staf, jika pengguna memilih fungsi hapus maka akan menjalankan skenario menghapus data staf, jika pengguna memilih fungsi cetak maka akan menjalankan skenario mencetak data staf. Berikut Tabel 4.15 menjelaskan lebih detail kebutuhan fungsional mengelola staf:

**Tabel 4.14 Use Case Scenario Mengelola Staf**

<b>Kode Use Case</b>	<b>UC-KKP-3</b>
<b>Nama</b>	Mengelola staf
<b>Aktor</b>	Pimpinan
<b>Deskripsi</b>	Sistem dapat mengelola staf berupa menambah, mengubah atau menghapus staf yang akan memiliki hak akses kedalam sistem
<b>Kondisi Awal</b>	Halaman menu staf ditampilkan sistem
<b>Skenario Utama</b>	
<b>Menambah data staf</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pimpinan memilih fungsi tambah staf  3. Pengguna mengisikan form data staf	2. Sistem menampilkan halaman form untuk memasukkan staf baru  4. Sistem menyimpan data staf dan menampilkan halaman menu staf lalu menampilkan pesan bahwa data berhasil dimasukkan
<b>Mengubah data staf</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pimpinan memilih fungsi ubah staf  3. Pimpinan mengisi atau mengubah form klien lalu setelah selesai tekan submit	2. Sistem menampilkan halaman form ubah staf  4. Data staf yang mengalami perubahan akan disimpan lalu menampilkan pesan bahwa data berhasil diubah
<b>Menghapus data staf</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>

**Tabel 4.14 Use Case Scenario Mengelola Staf (lanjutan)**

1. Pimpinan memilih fungsi hapus staf	2. Sistem melakukan proses penghapusan staf yang dipilih oleh pimpinan 3. Data staf terhapus dan menampilkan halaman menu klien lalu menampilkan pesan bahwa data berhasil dihapus
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
a. Data yang dimasukkan kurang lengkap 1. Pimpinan memilih menambah staf atau mengubah staf, lalu pimpinan tidak mengisi halaman form yang disediakan secara lengkap	2. Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi
<b>Kondisi Akhir</b>	Data staf berhasil dikelola dengan sistem
<b>Kode Kebutuhan</b>	KF-KKP-P3-1

#### 4.3.4 Use Case Scenario Mengelola *Monitoring* Aktivitas Klien

Penjelasan tentang kebutuhan fungsional mengelola *monitoring* aktivitas klien pada skenario utama yaitu jika pengguna memilih fungsi tambah maka akan menjalankan skenario menambah data *monitoring* aktivitas klien, jika pengguna memilih fungsi ubah maka akan menjalankan skenario mengubah data *monitoring* aktivitas klien, jika pengguna memilih fungsi hapus maka akan menjalankan skenario menghapus data *monitoring* aktivitas klien, jika pengguna memilih fungsi cetak maka akan menjalankan skenario mencetak data *monitoring* aktivitas klien. Berikut Tabel 4.16 menjelaskan lebih detail kebutuhan fungsional mengelola *monitoring* aktivitas klien:

**Tabel 4.15 Use Case Scenario Mengelola *Monitoring* Aktivitas Klien**

<b>Kode Use Case</b>	UC-KKP-4
<b>Nama</b>	Mengelola <i>monitoring</i> aktivitas klien
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Sistem dapat mengelola aktivitas konsultasi harian klien berupa menambah, mengubah, menghapus



Tabel 4.15 Use Case Scenario Mengelola Monitoring Aktivitas Klien (lanjutan)

	atau mencetak data aktivitas konsultasi klien
<b>Kondisi Awal</b>	Halaman menu <i>monitoring</i> aktivitas klien ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Menambah data <i>monitoring</i> aktivitas klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih daftar klien yang akan dikelola	2. Sistem menampilkan halaman detail <i>monitoring</i> aktivitas klien yang terdapat tombol tambah, ubah dan hapus
3. Pengguna memilih fungsi tambah <i>monitoring</i> aktivitas klien	4. Sistem menampilkan halaman form untuk memasukkan <i>monitoring</i> aktivitas klien baru
5. Pengguna mengisikan form-form <i>monitoring</i> aktivitas klien	6. Sistem menyimpan data <i>monitoring</i> aktivitas klien dan menampilkan halaman menu <i>monitoring</i> aktivitas klien lalu menampilkan pesan bahwa data berhasil dimasukkan
<b>Mengubah data <i>monitoring</i> aktivitas klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih daftar klien yang akan dikelola	2. Sistem menampilkan halaman detail <i>monitoring</i> aktivitas klien yang terdapat tombol tambah, ubah dan hapus
3. Pengguna memilih fungsi ubah data <i>monitoring</i> aktivitas klien	4. Sistem menampilkan halaman form ubah data <i>monitoring</i> aktivitas klien
5. Pengguna mengisi atau mengubah form data <i>monitoring</i> aktivitas klien lalu setelah selesai tekan submit	6. Data <i>monitoring</i> aktivitas klien yang mengalami perubahan akan disimpan dan menampilkan pesan bahwa data berhasil diubah
<b>Menghapus data <i>monitoring</i> aktivitas klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>

**Tabel 4.15 Use Case Scenario Mengelola Monitoring Aktivitas Klien (lanjutan)**

<p>1. Pengguna memilih daftar klien yang akan dikelola</p> <p>3. Pengguna memilih fungsi hapus data <i>monitoring</i> aktivitas klien</p>	<p>2. Sistem menampilkan halaman detail <i>monitoring</i> aktivitas klien yang terdapat tombol tambah, ubah dan hapus</p> <p>4. Sistem melakukan proses penghapusan data <i>monitoring</i> aktivitas klien yang dipilih oleh pengguna dan menampilkan halaman menu <i>monitoring</i> aktivitas klien lalu menampilkan pesan bahwa data berhasil dihapus</p>
<p><b>Mencetak data <i>monitoring</i> aktivitas klien</b></p>	
<p>1. Pengguna memilih daftar klien yang akan dikelola</p> <p>3. Pengguna memilih fungsi cetak <i>monitoring</i> aktivitas klien</p> <p>5. pengguna menyesuaikan pengaturan cetak lalu pilih tombol "OK"</p>	<p>2. Sistem menampilkan halaman detail <i>monitoring</i> aktivitas klien yang terdapat tombol tambah, ubah dan hapus</p> <p>4. Sistem menampilkan halaman preview untuk mencetak data <i>monitoring</i> aktivitas klien dan</p> <p>6. Sistem mencetak data <i>monitoring</i> aktivitas klien</p>
<p><b>Skenario Alternatif</b></p>	
<p><b>Aksi Aktor</b></p>	<p><b>Respon Sistem</b></p>
<p>a. Data yang dimasukkan kurang lengkap</p> <p>1. Pada skenario utama pengguna memilih menambah data <i>monitoring</i> aktivitas klien atau mengubah data <i>monitoring</i> aktivitas klien, lalu pengguna tidak mengisi halaman form yang disediakan secara lengkap</p> <p>b. Batal cetak data klien</p> <p>1. Pengguna memilih daftar klien yang akan dicetak dan memilih fungsi cetak klien</p>	<p>2. Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi</p> <p>2. Sistem menampilkan halaman preview untuk mencetak data</p>



**Tabel 4.15 Use Case Scenario Mengelola Monitoring Aktivitas Klien (lanjutan)**

3. Pengguna membatalkan fungsi cetak dengan menekan tombol "Cancel"	4. Sistem membatalkan cetak data
<b>Kondisi Akhir</b>	<i>monitoring</i> aktivitas klien berhasil dikelola dengan sistem oleh pengguna
<b>Kode Kebutuhan</b>	KF-KKP-P1-3

#### 4.3.5 Use Case Scenario Mengelola Monitoring Bulanan Klien

Penjelasan tentang kebutuhan fungsional mengelola *monitoring* bulanan klien pada skenario utama yaitu jika pengguna memilih fungsi tambah maka akan menjalankan skenario menambah data *monitoring* bulanan klien, jika pengguna memilih fungsi ubah maka akan menjalankan skenario mengubah data *monitoring* bulanan klien, jika pengguna memilih fungsi hapus maka akan menjalankan skenario menghapus data *monitoring* bulanan klien, jika pengguna memilih fungsi cetak maka akan menjalankan skenario mencetak data *monitoring* bulanan klien. Berikut Tabel 4.17 menjelaskan lebih detail kebutuhan fungsional mengelola *monitoring* bulanan klien:

**Tabel 4.16 Use Case Scenario Mengelola Monitoring Bulanan Klien**

<b>Kode Use Case</b>	UC-KKP-5
<b>Nama</b>	Mengelola <i>monitoring</i> bulanan klien
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Use case ini digunakan untuk mengelola <i>monitoring</i> bulanan klien berupa menambah, mengubah atau menghapus data <i>monitoring</i> bulanan klien yang digunakan untuk melakukan <i>memonitoring</i> proses pembayaran perpajakan bulanan klien
<b>Kondisi Awal</b>	Status login sukses dan menu <i>monitoring</i> bulanan klien ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Menambah data <i>monitoring</i> bulanan klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih daftar klien yang akan dikelola	2. Sistem menampilkan halaman detail <i>monitoring</i> bulanan klien



**Tabel 4.16 Use Case Scenario Mengelola *Monitoring* Bulanan Klien (lanjutan)**

<p>4. Pengguna memilih fungsi tambah <i>monitoring</i> bulanan klien</p> <p>6. Pengguna mengisikan form-form <i>monitoring</i> bulanan klien</p>	<p>yang terdapat tombol tambah, ubah dan hapus</p> <p>5. Sistem menampilkan halaman form untuk memasukkan <i>monitoring</i> bulanan klien baru</p> <p>7. Sistem menyimpan data <i>monitoring</i> bulanan klien dan menampilkan halaman menu <i>monitoring</i> bulanan klien lalu menampilkan pesan bahwa data berhasil dimasukkan</p>
<b>Mengubah data <i>monitoring</i> bulanan klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
<p>1. Pengguna memilih daftar klien yang akan dikelola</p> <p>3. Pengguna memilih fungsi ubah data <i>monitoring</i> bulanan klien</p> <p>5. Pengguna mengisi atau mengubah form data <i>monitoring</i> bulanan klien lalu setelah selesai tekan submit</p>	<p>2. Sistem menampilkan halaman detail <i>monitoring</i> bulanan klien yang terdapat tombol tambah, ubah dan hapus</p> <p>4. Sistem menampilkan halaman form ubah data <i>monitoring</i> bulanan klien</p> <p>6. Data <i>monitoring</i> bulanan klien yang mengalami perubahan akan disimpan dan menampilkan pesan bahwa data berhasil diubah</p>
<b>Menghapus data <i>monitoring</i> bulanan klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
<p>1. Pengguna memilih daftar klien yang akan dikelola</p> <p>3. Pengguna memilih fungsi hapus data <i>monitoring</i> bulanan klien</p>	<p>2. Sistem menampilkan halaman detail <i>monitoring</i> bulanan klien yang terdapat tombol tambah, ubah dan hapus</p> <p>4. Sistem melakukan proses penghapusan data <i>monitoring</i> bulanan klien yang dipilih oleh pengguna dan menampilkan halaman menu <i>monitoring</i> bulanan klien lalu menampilkan pesan bahwa data berhasil dihapus</p>
<b>Mencetak data <i>monitoring</i> bulanan klien</b>	



**Tabel 4.16 Use Case Scenario Mengelola Monitoring Bulanan Klien (lanjutan)**

1. Pengguna memilih daftar klien yang akan dikelola	2. Sistem menampilkan halaman detail <i>monitoring</i> bulanan klien yang terdapat tombol tambah, ubah dan hapus
3. Pengguna memilih fungsi cetak <i>monitoring</i> bulanan klien	4. Sistem menampilkan halaman preview untuk mencetak data <i>monitoring</i> bulanan klien dan
5. pengguna menyesuaikan pengaturan cetak lalu pilih tombol "OK"	6. Sistem mencetak data <i>monitoring</i> bulanan klien
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
<p>a. Data yang dimasukkan kurang lengkap</p> <p>1. Pada skenario utama pengguna memilih menambah data <i>monitoring</i> bulanan klien atau mengubah data <i>monitoring</i> bulanan klien, lalu pengguna tidak mengisi halaman form yang disediakan secara lengkap</p> <p>b. Batal cetak data klien</p> <p>1. Pengguna memilih daftar klien yang akan dicetak dan memilih fungsi cetak klien</p> <p>3. Pengguna membatalkan fungsi cetak dengan menekan tombol "Cancel"</p>	<p>2. Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi</p> <p>2. Sistem menampilkan halaman preview untuk mencetak data</p> <p>4. Sistem membatalkan cetak data</p>
<b>Kondisi Akhir</b>	<i>Monitoring</i> bulanan klien berhasil dikelola dengan sistem
<b>Kode Kebutuhan</b>	KF-KKP-P1-4

#### 4.3.6 Use Case Scenario Request Validasi Monitoring Bulanan Klien

Penjelasan tentang kebutuhan fungsional *request validasi monitoring* bulanan klien akan dijelaskan dalam Tabel 4.18 berikut :



Tabel 4.17 Use Case Scenario Request Validasi Monitoring Bulanan Klien

<b>Kode Use Case</b>	<b>UC-KKP-6</b>
<b>Nama</b>	<i>Request validasi monitoring</i> bulanan klien
<b>Aktor</b>	Staf
<b>Deskripsi</b>	Sistem memproses <i>request validasi monitoring</i> bulanan yang dilakukan oleh staf kepada pimpinan
<b>Kondisi Awal</b>	Halaman menu <i>monitoring</i> bulanan klien ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Staf memilih daftar klien yang akan dikelola 3. Staf memilih fungsi <i>request validasi selesai monitoring</i> bulanan	2. Sistem menampilkan halaman detail <i>monitoring</i> bulanan klien 4. Sistem memproses dengan merubah status menjadi 'proses' dan meneruskan agar bisa dicek dan divalidasi oleh pimpinan
<b>Skenario Alternatif</b>	
Tidak ada	Tidak ada
<b>Kondisi Akhir</b>	<i>Request validasi monitoring</i> bulanan klien berhasil diproses sistem dan status dirubah menjadi 'proses'
<b>Kode Kebutuhan</b>	KF-KKP-P2-1

#### 4.3.7 Use Case Scenario Memvalidasi Monitoring Bulanan Klien

Penjelasan tentang kebutuhan fungsional memvalidasi *monitoring* bulanan klien akan dijelaskan dalam Tabel 4.19 berikut :

Tabel 4.18 Use Case Scenario Validasi Monitoring Bulanan Klien

<b>Kode Use Case</b>	<b>UC-KKP-7</b>
<b>Nama</b>	Memvalidasi <i>monitoring</i> bulanan klien
<b>Aktor</b>	Pimpinan
<b>Deskripsi</b>	Sistem memproses validasi <i>monitoring</i> bulanan yang dilakukan oleh pimpinan kepada staf

**Tabel 4.18 Use Case Scenario Validasi Monitoring Bulanan Klien (lanjutan)**

<b>Kondisi Awal</b>	Halaman menu <i>dashboard</i> ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pimpinan memilih <i>request</i> validasi yang akan diproses	2. Sistem menampilkan halaman <i>monitoring</i> bulanan klien berdasarkan <i>request</i> validasi yang dipilih
3. Pimpinan mengecek data dan memilih menerima validasi	4. Sistem merubah statusnya menjadi 'selesai'
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
a. Pimpinan menolak <i>request</i> validasi	
1. Pimpinan memilih <i>request</i> validasi yang akan diproses	2. Sistem menampilkan halaman <i>monitoring</i> bulanan klien berdasarkan <i>request</i> validasi yang dipilih
3. Pimpinan memilih menolak validasi	4. Sistem merubah statusnya menjadi 'belum selesai'
<b>Kondisi Akhir</b>	Validasi <i>monitoring</i> bulanan klien berhasil diproses sistem dan status dirubah menjadi 'selesai'
<b>Kode Kebutuhan</b>	KF-KKP-P3-2

#### 4.3.8 Use Case Scenario Menambah Tanda Terima Dokumen

Penjelasan tentang kebutuhan fungsional menambah tanda terima dokumen akan dijelaskan dalam Tabel 4.20 berikut :

**Tabel 4 19 Use Case Scenario Menambah Tanda Terima Dokumen**

<b>Kode Use Case</b>	<b>UC-KKP-8</b>
<b>Nama</b>	Menambah tanda terima dokumen
<b>Aktor</b>	Staf
<b>Deskripsi</b>	Sistem bisa menambah tanda terima dokumen dan data tanda terima



**Tabel 4 19 Use Case Scenario Menambah Tanda Terima Dokumen (lanjutan)**

	dokumen akan disimpan kedalam sistem
<b>Kondisi Awal</b>	Halaman menu tanda terima dokumen ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
<ol style="list-style-type: none"> <li>1. Staf memilih fungsi tambah tanda terima dokumen</li> <li>3. Staf mengisi form tanda terima dokumen, lalu memilih submit</li> <li>5. Staf memilih fungsi cetak dengan menekan tombol "OK"</li> </ol>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan form tanda terima dokumen</li> <li>4. Sistem menyimpan data dan menampilkan halaman preview untuk mencetak data tanda terima dokumen</li> <li>6. Sistem mencetak tanda terima</li> </ol>
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
<p>a. Data yang dimasukkan kurang lengkap</p> <ol style="list-style-type: none"> <li>1. Staf tidak mengisi halaman form tanda terima dokumen yang disediakan secara lengkap, lalu memilih submit</li> </ol> <p>b. Batal cetak data <i>monitoring</i> bulanan klien</p> <ol style="list-style-type: none"> <li>1. Staff mengisi form tanda terima dokumen lalu menekan submit</li> <li>3. Staf membatalkan fungsi cetak dengan menekan tombol "Cancel"</li> </ol>	<ol style="list-style-type: none"> <li>2. Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi</li> <li>2. Sistem menampilkan halaman preview untuk mencetak data tanda terima dokumen</li> <li>4. Data tanda terima dokumen tetap tersimpan kedalam sistem dan tidak mencetak form tanda terima</li> </ol>
<b>Kondisi Akhir</b>	Data tanda terima dokumen berhasil disimpan sistem dan data tanda terima dokumen dicetak sistem
<b>Kode Kebutuhan</b>	KF-KKP-P2-2



### 4.3.9 Use Case Scenario Mengelola Tanda Terima Dokumen

Penjelasan tentang kebutuhan fungsional mengelola tanda terima dokumen pada skenario utama yaitu jika pengguna memilih fungsi hapus tanda terima maka akan menjalankan skenario menghapus data tanda terima dokumen klien, jika pengguna memilih fungsi cetak riwayat tanda terima maka akan menjalankan skenario mencetak data riwayat tanda terima dokumen klien. Berikut Tabel 4.21 menjelaskan lebih detail kebutuhan fungsional mengelola tanda terima dokumen:

**Tabel 4.20 Use Case Scenario Mengelola Tanda Terima Dokumen**

<b>Kode Use Case</b>	<b>UC-KKP-9</b>
<b>Nama</b>	Mengelola tanda terima dokumen
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Sistem dapat mengelola tanda terima dokumen berupa menghapus atau mencetak data tanda terima dokumen klien sebelumnya yang telah tersimpan
<b>Kondisi Awal</b>	Halaman menu tanda terima dokumen ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Menghapus data tanda terima dokumen</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
<ol style="list-style-type: none"> <li>1. Pengguna memilih daftar klien yang akan dikelola</li> <li>2. Pengguna memilih fungsi hapus data tanda terima dokumen</li> </ol>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan halaman detail tanda terima dokumen yang terdapat tombol hapus dan cetak</li> <li>3. Sistem melakukan proses penghapusan data tanda terima dokumen yang dipilih oleh pengguna</li> <li>5. Data tanda terima dokumen terhapus dan menampilkan halaman riwayat tanda terima dokumen lalu menampilkan pesan bahwa data berhasil dihapus</li> </ol>
<b>Mencetak data riwayat tanda terima dokumen</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
<ol style="list-style-type: none"> <li>2. Pengguna memilih daftar klien yang akan dikelola</li> </ol>	<ol style="list-style-type: none"> <li>3. Sistem menampilkan halaman detail tanda terima dokumen yang terdapat tombol hapus dan cetak</li> </ol>



**Tabel 4.20 Use Case Scenario Mengelola Tanda Terima Dokumen (lanjutan)**

3. Pengguna memilih fungsi cetak riwayat tanda terima dokumen	4. Sistem menampilkan halaman preview untuk mencetak data <i>monitoring</i> bulanan klien
4. Pengguna menyesuaikan pengaturan cetak lalu pilih tombol "OK"	5. Sistem mencetak data <i>monitoring</i> bulanan klien
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
a. Batal cetak data <i>monitoring</i> bulanan klien 1. Staff memilih fungsi cetak riwayat tanda terima dokumen 3. pengguna membatalkan fungsi cetak dengan menekan tombol "Cancel"	2. Sistem menampilkan halaman preview untuk mencetak data tanda terima dokumen 4. Sistem membatalkan proses cetak
<b>Kondisi Akhir</b>	Tanda terima dokumen berhasil dikelola dengan sistem
<b>Kode Kebutuhan</b>	KF-KKP-P1-5

#### 4.3.10 Use Case Scenario Mengelola *Monitoring* Tahunan Klien

Penjelasan tentang kebutuhan fungsional Mengelola *monitoring* tahunan klien pada skenario utama yaitu jika pengguna memilih fungsi tambah maka akan menjalankan skenario menambah data *monitoring* tahunan klien, jika pengguna memilih fungsi ubah maka akan menjalankan skenario mengubah data *monitoring* tahunan klien, jika pengguna memilih fungsi hapus maka akan menjalankan skenario menghapus data *monitoring* tahunan klien, jika pengguna memilih fungsi cetak maka akan menjalankan skenario mencetak data *monitoring* tahunan klien. Berikut Tabel 4.22 menjelaskan lebih detail kebutuhan fungsional mengelola *monitoring* tahunan klien:

**Tabel 4.21 Use Case Scenario Mengelola *Monitoring* Tahunan Klien**

<b>Kode Use Case</b>	<b>UC-KKP-10</b>
<b>Nama</b>	Mengelola <i>monitoring</i> tahunan klien
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Sistem dapat mengelola <i>monitoring</i> tahunan klien berupa menambah, mengubah atau menghapus data



Tabel 4.21 Use Case Scenario Mengelola Monitoring Tahunan Klien (lanjutan)

	<i>monitoring</i> tahunan klien yang digunakan untuk melakukan <i>memonitoring</i> proses pembayaran perpajakan tahunan klien
<b>Kondisi Awal</b>	Halaman menu <i>monitoring</i> tahunan klien ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Menambah data <i>monitoring</i> tahunan klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih daftar klien yang akan dikelola	2. Sistem menampilkan halaman detail <i>monitoring</i> tahunan klien yang terdapat tombol tambah, ubah dan hapus
3. Pengguna memilih fungsi tambah <i>monitoring</i> tahunan klien	4. Sistem menampilkan halaman form untuk memasukkan <i>monitoring</i> tahunan klien baru
5. Pengguna mengisikan form-form <i>monitoring</i> tahunan klien	6. Sistem menyimpan data <i>monitoring</i> tahunan klien dan menampilkan pesan bahwa data berhasil dimasukkan
<b>Mengubah data <i>monitoring</i> tahunan klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih daftar klien yang akan dikelola	2. Sistem menampilkan halaman detail <i>monitoring</i> tahunan klien yang terdapat tombol tambah, ubah dan hapus
3. Pengguna memilih fungsi ubah data <i>monitoring</i> tahunan klien	4. Sistem menampilkan halaman form ubah data <i>monitoring</i> tahunan klien
5. Pengguna mengisi atau mengubah form data <i>monitoring</i> tahunan klien lalu setelah selesai tekan submit	6. Data <i>monitoring</i> tahunan klien yang mengalami perubahan akan disimpan dan menampilkan halaman menu <i>monitoring</i> tahunan klien lalu menampilkan pesan bahwa data berhasil diubah
<b>Menghapus data <i>monitoring</i> tahunan klien</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>

**Tabel 4.21 Use Case Scenario Mengelola Monitoring Tahunan Klien (lanjutan)**

1. Pengguna memilih daftar klien yang akan dikelola	2. Sistem menampilkan halaman detail <i>monitoring</i> tahunan klien yang terdapat tombol tambah, ubah dan hapus
3. Pengguna memilih fungsi hapus data <i>monitoring</i> tahunan klien	4. Sistem melakukan proses penghapusan data <i>monitoring</i> tahunan klien yang dipilih oleh pengguna dan menampilkan halaman menu <i>monitoring</i> tahunan klien lalu menampilkan pesan bahwa data berhasil dihapus
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
a. Data yang dimasukkan kurang lengkap  1. Pada skenario utama pengguna memilih menambah data <i>monitoring</i> tahunan klien atau mengubah data <i>monitoring</i> tahunan klien, lalu pengguna tidak mengisi halaman form yang disediakan secara lengkap	2. Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi
<b>Kondisi Akhir</b>	<i>Monitoring</i> tahunan klien berhasil dikelola dengan sistem oleh pengguna
<b>Kode Kebutuhan</b>	KF-KKP-P1-6

#### 4.3.11 Use Case Scenario Request Validasi Monitoring Tahunan Klien

Penjelasan tentang kebutuhan fungsional *request* validasi *monitoring* tahunan klien akan dijelaskan dalam Tabel 4.23 berikut :

**Tabel 4.22 Use Case Scenario Request Validasi Monitoring Tahunan Klien**

<b>Kode Use Case</b>	<b>UC-KKP-11</b>
<b>Nama</b>	<i>Request</i> validasi <i>monitoring</i> tahunan klien
<b>Aktor</b>	Staf
<b>Deskripsi</b>	Sistem memproses <i>request</i> validasi <i>monitoring</i> tahunan yang dilakukan oleh staf kepada pimpinan

**Tabel 4.22 Use Case Scenario *Request Validasi Monitoring Tahunan* Klien (lanjutan)**

<b>Kondisi Awal</b>	Halaman menu <i>monitoring</i> tahunan klien ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Staf memilih daftar klien yang akan dikelola  3. Staf memilih fungsi <i>request validasi</i> selesai <i>monitoring</i> tahunan	2. Sistem menampilkan halaman detail <i>monitoring</i> tahunan klien  4. Sistem memproses dengan merubah status menjadi 'proses' dan meneruskan agar bisa dicek dan divalidasi oleh pimpinan
<b>Skenario Alternatif</b>	
Tidak ada	Tidak ada
<b>Kondisi Akhir</b>	<i>Request validasi monitoring</i> tahunan klien berhasil diproses sistem dan status dirubah menjadi 'proses'
<b>Kode Kebutuhan</b>	KF-KKP-P2-3

#### 4.3.12 Use Case Scenario Memvalidasi *Monitoring Tahunan* Klien

Penjelasan tentang kebutuhan fungsional memvalidasi *monitoring* tahunan klien akan dijelaskan dalam Tabel 4.24 berikut :

**Tabel 4.23 Use Case Scenario Memvalidasi *Monitoring Tahunan* Klien**

<b>Kode Use Case</b>	<b>UC-KKP-12</b>
<b>Nama</b>	Memvalidasi <i>monitoring</i> tahunan klien
<b>Aktor</b>	Pimpinan
<b>Deskripsi</b>	Sistem memproses validasi <i>monitoring</i> tahunan yang dilakukan oleh pimpinan kepada staf
<b>Kondisi Awal</b>	Halaman menu dashboard ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pimpinan memilih <i>request validasi</i> yang akan diproses	2. Sistem menampilkan halaman <i>monitoring</i> tahunan klien berdasarkan <i>request validasi</i> yang dipilih

**Tabel 4.23 Use Case Scenario Memvalidasi Monitoring Tahunan Klien (lanjutan)**

3. Pimpinan mengecek data dan memilih menerima validasi	4. Sistem merubah statusnya menjadi 'selesai'
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
a. Pimpinan menolak <i>request</i> validasi	
1. Pimpinan memilih <i>request</i> validasi yang akan diproses	2. Sistem menampilkan halaman <i>monitoring</i> tahunan klien berdasarkan <i>request</i> validasi yang dipilih
3. Pimpinan memilih menolak validasi	4. Sistem merubah statusnya menjadi 'belum selesai'
<b>Kondisi Akhir</b>	Validasi <i>monitoring</i> tahunan klien berhasil diproses sistem dan status dirubah menjadi 'selesai'
<b>Kode Kebutuhan</b>	KF-KKP-P3-3

#### 4.3.13 Use Case Scenario Mengelola Proposal

Penjelasan tentang kebutuhan fungsional mengelola proposal pada skenario utama yaitu jika pengguna memilih fungsi ubah maka akan menjalankan skenario mengubah data proposal klien, jika pengguna memilih fungsi cetak maka akan menjalankan skenario mencetak data proposal klien. Berikut Tabel 4.25 menjelaskan lebih detail kebutuhan fungsional mengelola proposal:

**Tabel 4.24 Use Case Scenario Mengelola Proposal**

<b>Kode Use Case</b>	<b>UC-KKP-13</b>
<b>Nama</b>	Mengelola proposal
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Sistem dapat mengelola proposal berupa mengubah atau mencetak data tanda terima dokumen klien
<b>Kondisi Awal</b>	Halaman menu klien ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Mengubah data proposal</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>

**Tabel 4.24 Use Case Scenario Mengelola Proposal (lanjutan)**

1. Pengguna memilih daftar klien yang akan dikelola dan memilih fungsi proposal	2. Sistem menampilkan halaman detail proposal
3. Pengguna mengubah data proposal dengan memilih ubah proposal	4. Sistem menampilkan form-form data proposal
5. Pengguna mengisi atau mengubah form yang tersedia untuk merubah data proposal lalu menekan submit	6. Sistem menyimpan perubahan data proposal
<b>Mencetak data proposal</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pengguna memilih daftar klien yang akan dikelola dan memilih fungsi proposal	2. Sistem menampilkan halaman detail proposal
3. Pengguna memilih fungsi cetak proposal	4. Sistem menampilkan halaman preview untuk mencetak data proposal
5. Pengguna menyesuaikan pengaturan cetak lalu pilih tombol "OK"	6. Sistem mencetak data proposal
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
a. Batal cetak data proposal	
2. Staff memilih fungsi cetak riwayat tanda terima dokumen	3. Sistem menampilkan halaman preview untuk mencetak data tanda terima dokumen
4. pengguna membatalkan fungsi cetak dengan menekan tombol "Cancel"	5. Sistem membatalkan proses cetak
<b>Kondisi Akhir</b>	Data proposal berhasil dikelola dengan sistem
<b>Kode Kebutuhan</b>	KF-KKP-P1-7

#### 4.3.14 Use Case Scenario Request Persetujuan Proposal

Penjelasan tentang kebutuhan fungsional *request* persetujuan proposal akan dijelaskan dalam Tabel 4.26 berikut :



**Tabel 4.25 Use Case Scenario Request Persetujuan Proposal**

<b>Kode Use Case</b>	<b>UC-KKP-14</b>
<b>Nama</b>	<i>Request</i> persetujuan proposal
<b>Aktor</b>	Staf
<b>Deskripsi</b>	Sistem memproses <i>request</i> persetujuan proposal yang dilakukan oleh staf kepada pimpinan
<b>Kondisi Awal</b>	Halaman menu klien ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
<ol style="list-style-type: none"> <li>1. Staf memilih daftar klien yang akan diproses persetujuan proposal</li> <li>3. Staf memilih fungsi <i>request</i> persetujuan proposal selesai</li> </ol>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan halaman menu proposal klien yang dipilih</li> <li>4. Sistem memproses dengan merubah status menjadi 'proses' dan meneruskan agar bisa dicek dan setuju oleh pimpinan</li> </ol>
<b>Skenario Alternatif</b>	
Tidak ada	Tidak ada
<b>Kondisi Akhir</b>	<i>Request</i> persetujuan proposal berhasil diproses sistem dan status dirubah menjadi 'proses'
<b>Kode Kebutuhan</b>	KF-KKP-P2-4

#### 4.3.15 Use Case Scenario Menyetujui Proposal

Penjelasan tentang kebutuhan fungsional menyetujui proposal akan dijelaskan dalam Tabel 4.27 berikut :

**Tabel 4.26 Use Case Scenario Menyetujui Proposal**

<b>Kode Use Case</b>	<b>UC-KKP-15</b>
<b>Nama</b>	Menyetujui proposal
<b>Aktor</b>	Pimpinan
<b>Deskripsi</b>	Sistem memproses persetujuan propsoal yang dilakukan oleh pimpinan kepada staf



**Tabel 4.26 Use Case Scenario Menyetujui Proposal (lanjutan)**

<b>Kondisi Awal</b>	Halaman menu dashboard ditampilkan oleh sistem
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pimpinan memilih <i>request</i> persetujuan yang akan diproses 3. Pimpinan mengecek data dan menyetujui	2. Sistem menampilkan halaman proposal klien berdasarkan <i>request</i> persetujuan proposal yang dipilih 4. Sistem merubah statusnya menjadi 'Disetujui'
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
a. Pimpinan menolak persetujuan 1. Pimpinan memilih <i>request</i> persetujuan proposal 3. Pimpinan memilih menolak persetujuan proposal	2. Sistem menampilkan halaman propsoal klien berdasarkan <i>request</i> 4. Sistem merubah statusnya menjadi 'Tidak disetujui'
<b>Kondisi Akhir</b>	Persetujuan proposal berhasil diproses sistem dan status dirubah menjadi 'Disetujui'
<b>Kode Kebutuhan</b>	KF-KKP-P3-4

**4.3.16 Use Case Scenario Logout**

Penjelasan tentang kebutuhan fungsional *logout* akan dijelaskan dalam Tabel 4.28 berikut :

**Tabel 4.27 Use Case Scenario Logout**

<b>Kode Use Case</b>	<b>UC-KKP-16</b>
<b>Nama</b>	<i>Logout</i>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Use case ini digunakan oleh pengguna untuk keluar dari sistem yang sudah teridentifikasi dengan cara menghapus session agar menjaga keamanan informasi perangkat lunak



Tabel 4.27 Use Case Scenario Logout (lanjutan)

<b>Kondisi Awal</b>	Status <i>login</i> sukses dan pengguna berada dalam sistem
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
3. Pengguna memilih fungsi <i>logout</i>	4. Sistem akan menghapus <i>session</i> yang teridentifikasi dan sistem menampilkan halaman login
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
Tidak ada	Tidak ada
<b>Kondisi Akhir</b>	Session yang teridentifikasi dihapus dan halaman login kembali ditampilkan oleh sistem
<b>Kode Kebutuhan</b>	KF-KKP-P1-8

#### 4.3.17 Use Case Scenario Melihat Grafik Aktivitas

Penjelasan tentang kebutuhan fungsional melihat grafik aktivitas akan dijelaskan dalam Tabel 4.29 berikut :

Tabel 4.28 Use Case Scenario Melihat Grafik Aktivitas

<b>Kode Use Case</b>	<b>UC-KKP-17</b>
<b>Nama</b>	Melihat grafik aktivitas
<b>Aktor</b>	Pimpinan
<b>Deskripsi</b>	Use case ini digunakan oleh pimpinan untuk menampilkan grafik data aktivitas
<b>Kondisi Awal</b>	Status <i>login</i> sukses halaman menu dashboard ditampilkan
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pimpinan memilih menu dashboard	2. Sistem akan menampilkan halaman dashboard dan menampilkan grafik aktivitas
<b>Skenario Alternatif</b>	

**Tabel 4.28 Use Case Scenario Melihat Grafik Aktivitas (lanjutan)**

Aksi Aktor	Respon Sistem
Tidak ada	Tidak ada
Kondisi Akhir	Halaman dashboard dan informasi grafik aktivitas ditampilkan
Kode Kebutuhan	KF-KKP-P3-5

#### 4.3.18 Use Case Scenario Melihat Grafik Monitoring Bulanan

Penjelasan tentang kebutuhan fungsional melihat grafik *monitoring* bulanan akan dijelaskan dalam Tabel 4.30 berikut :

**Tabel 4.29 Use Case Scenario Melihat Grafik Monitoring Bulanan**

Kode Use Case	UC-KKP-18
Nama	Melihat grafik <i>monitoring</i> bulanan
Aktor	Pimpinan
Deskripsi	Use case ini digunakan oleh pimpinan untuk menampilkan grafik data <i>monitoring</i> bulanan
Kondisi Awal	Status <i>login</i> sukses halaman menu dashboard ditampilkan
Skenario Utama	
Aksi Aktor	Respon Sistem
1. Pimpinan memilih menu dashboard	2. Sistem akan menampilkan halaman dashboard
3. Pimpinan memilih tombol <i>monitoring</i> perpajakan	4. Sistem menampilkan halaman <i>monitoring</i> perpajakan dan informasi grafik monitoring bulanan
Skenario Alternatif	
Aksi Aktor	Respon Sistem
Tidak ada	Tidak ada
Kondisi Akhir	Halaman <i>monitoring</i> perpajakan dan informasi grafik <i>monitoring</i> bulanan ditampilkan
Kode Kebutuhan	KF-KKP-P3-6



#### 4.3.19 Use Case Scenario Melihat Grafik *Monitoring* Tahunan

Penjelasan tentang kebutuhan fungsional melihat grafik *monitoring* tahunan akan dijelaskan dalam Tabel 4.31 berikut :

**Tabel 4.30 Use Case Scenario Melihat Grafik *Monitoring* Tahunan**

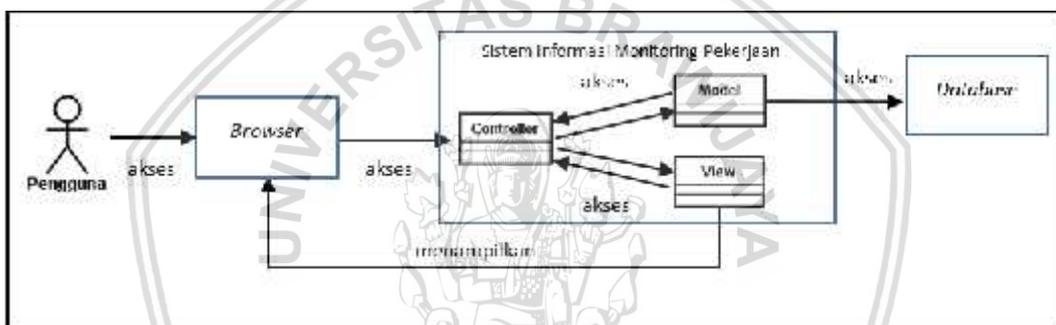
<b>Kode Use Case</b>	<b>UC-KKP-19</b>
<b>Nama</b>	Melihat grafik <i>monitoring</i> tahunan
<b>Aktor</b>	Pimpinan
<b>Deskripsi</b>	Use case ini digunakan oleh pimpinan untuk menampilkan grafik data <i>monitoring</i> tahunan
<b>Kondisi Awal</b>	Status <i>login</i> sukses halaman menu dashboard ditampilkan
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
1. Pimpinan memilih menu dashboard	2. Sistem akan menampilkan halaman dashboard
3. Pimpinan memilih tombol <i>monitoring</i> perpajakan	4. Sistem menampilkan halaman <i>monitoring</i> perpajakan dan informasi grafik monitoring bulanan
<b>Skenario Alternatif</b>	
<b>Aksi Aktor</b>	<b>Respon Sistem</b>
Tidak ada	Tidak ada
<b>Kondisi Akhir</b>	Halaman <i>monitoring</i> perpajakan dan informasi grafik <i>monitoring</i> tahunan ditampilkan
<b>Kode Kebutuhan</b>	KF-KKP-P3-7

## BAB 5 PERANCANGAN

Pada bab ini menjelaskan tentang tahapan perancangan Sistem Informasi *Monitoring* Pekerjaan Kantor konsultan pajak. Tahap perancangan adalah tahapan awal dalam mendesain sistem dan merancang antarmuka yang akan dibangun. Perancangan dibuat berdasarkan hasil dari analisis kebutuhan yang sebelumnya telah ditentukan. Tahap perancangan mencakup perancangan arsitektur sistem, perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data dan perancangan antarmuka pengguna.

### 5.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem bertujuan untuk menggambarkan bagaimana interaksi antar komponen-komponen Sistem Informasi *Monitoring* Pekerjaan Kantor konsultan pajak. Perancangan arsitektur sistem bisa dilihat pada Gambar 5.1 berikut.



**Gambar 5.1** Arsitektur Diagram Sistem Informasi *Monitoring* Pekerjaan

Pada Gambar 5.1 menjelaskan bahwa sistem diakses oleh pengguna melalui *browser*. Sistem yang dibangun menggunakan *framework* CodeIgniter, sehingga browser akan memproses sistem sesuai dengan akses file yang ada pada CodeIgniter dengan menerapkan pola *Mode-View-Controller* (MVC).

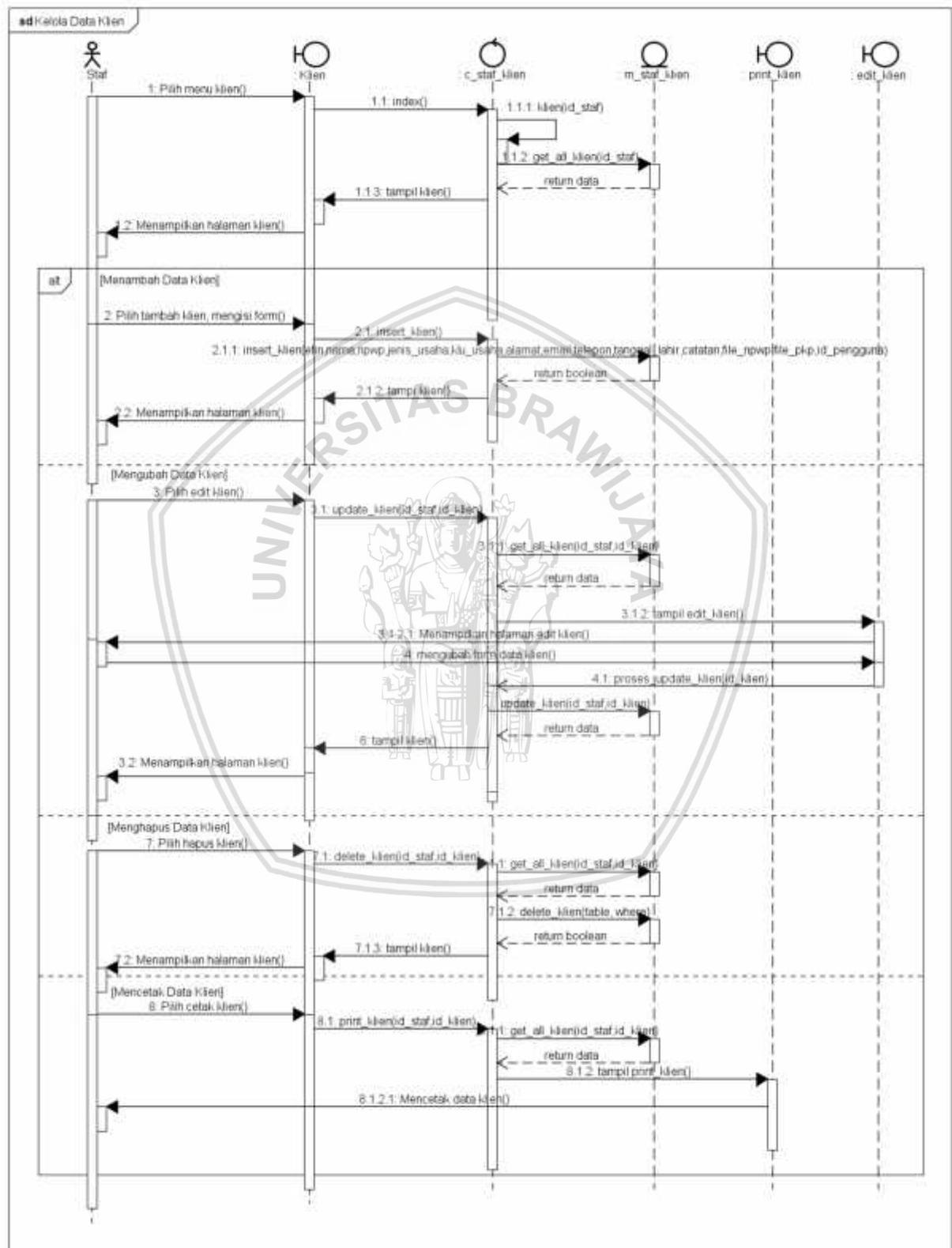
### 5.2 Perancangan *Sequence Diagram*

*Sequence diagram* berisi penggambaran alur interaksi yang berfokus pada urutan pertukaran pesan yang terjadi antar objek dalam perangkat lunak. *Sequence diagram* terdiri dari rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu *event* dan digunakan untuk mengidentifikasi *method* yang terdapat dalam sistem. Berikut ini merupakan *sequence diagram* berdasarkan hasil analisis kebutuhan sistem yang sudah ditetapkan.

#### 5.2.1 *Sequence Diagram* Kelola Data Klien

*Sequence diagram* kelola data klien mengacu pada *use case scenario* dengan kode UC-KKP-2. Alur ini diawali dengan pengguna memilih menu klien untuk menampilkan daftar klien yang tersimpan selanjutnya pengguna bisa memilih tombol tambah klien untuk membuat klien baru dengan mengisi form-form

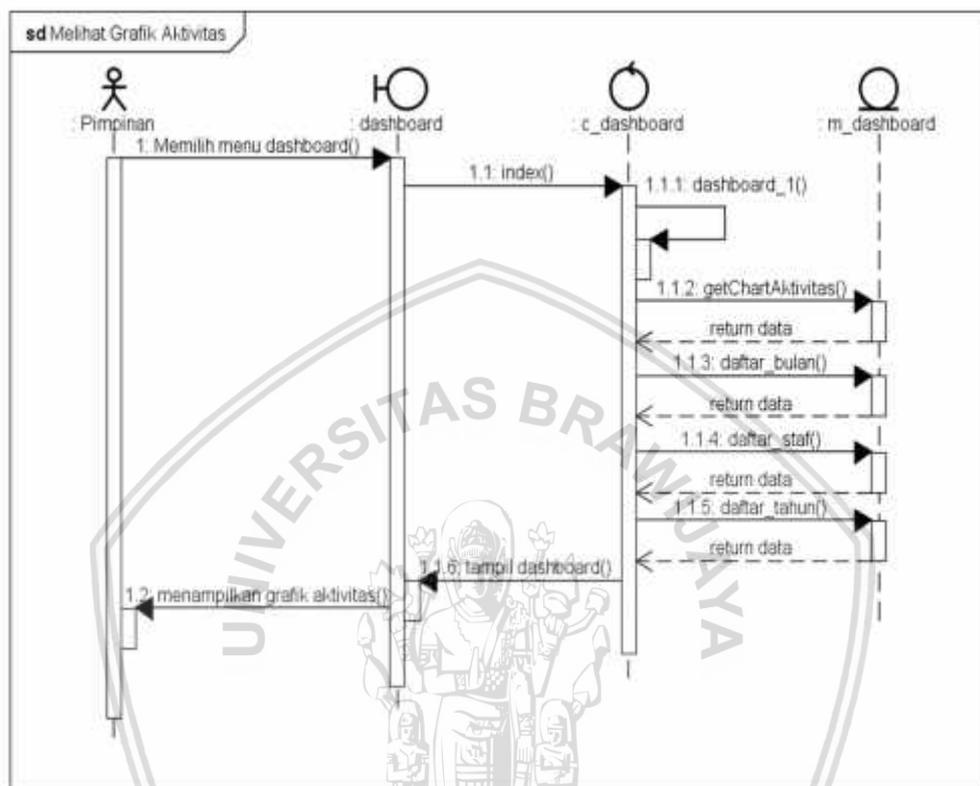
yang telah tersedia, kemudian pilih submit dengan menjalankan fungsi insert\_klien() dan menyimpan data kedalam database, selanjutnya akan menampilkan halaman daftar klien beserta data klien yang sudah dibuat.



Gambar 5.2 Class Diagram Kelola Data Klien

### 5.2.2 Sequence Diagram Melihat Grafik Aktivitas

Sequence diagram melihat grafik aktivitas mengacu pada use case scenario dengan kode UC-KKP-17. Alur ini diawali dengan pengguna memilih menu dashboard, kemudian akan menampilkan grafik jumlah aktivitas yang dilakukan oleh staf.

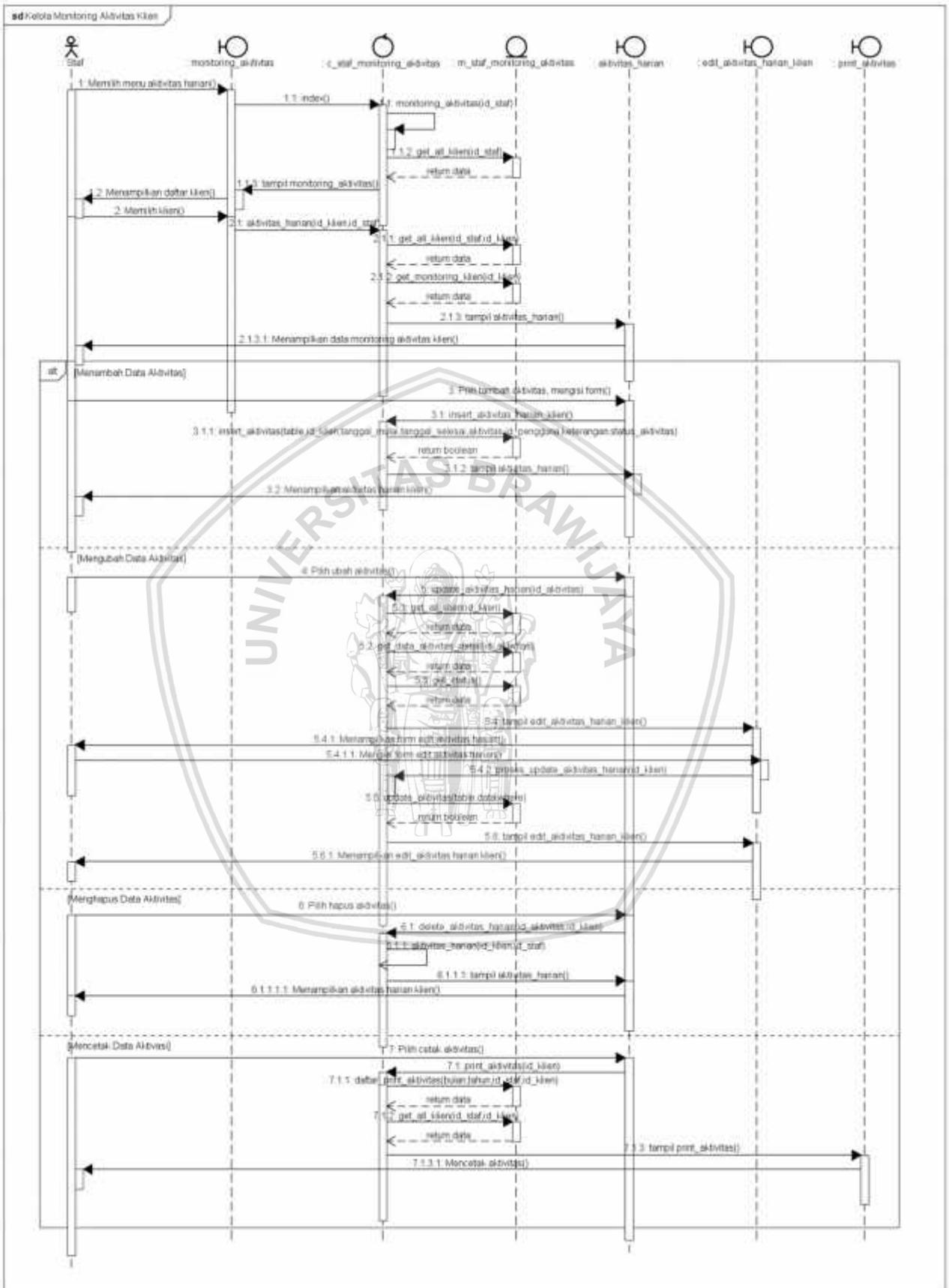


Gambar 5.3 Sequence Diagram Melihat Grafik Aktivitas

### 5.2.3 Sequence Diagram Kelola Monitoring Aktivitas

Sequence diagram kelola data monitoring aktivitas mengacu pada use case scenario dengan kode UC-KKP-4. Alur ini diawali dengan pengguna memilih klien yang akan dikelola, kemudian menampilkan data konsultasi yang ada. Pengguna bisa memilih menambahkan data konsultasi harian lalu mengisi form-form yang tersedia lalu pilih submit, kemudian sistem akan menjalankan fungsi insert\_aktivitas\_harian\_klien() dan menyimpan datanya ke dalam database. Selanjutnya sistem akan menampilkan daftar data konsultasi harian.



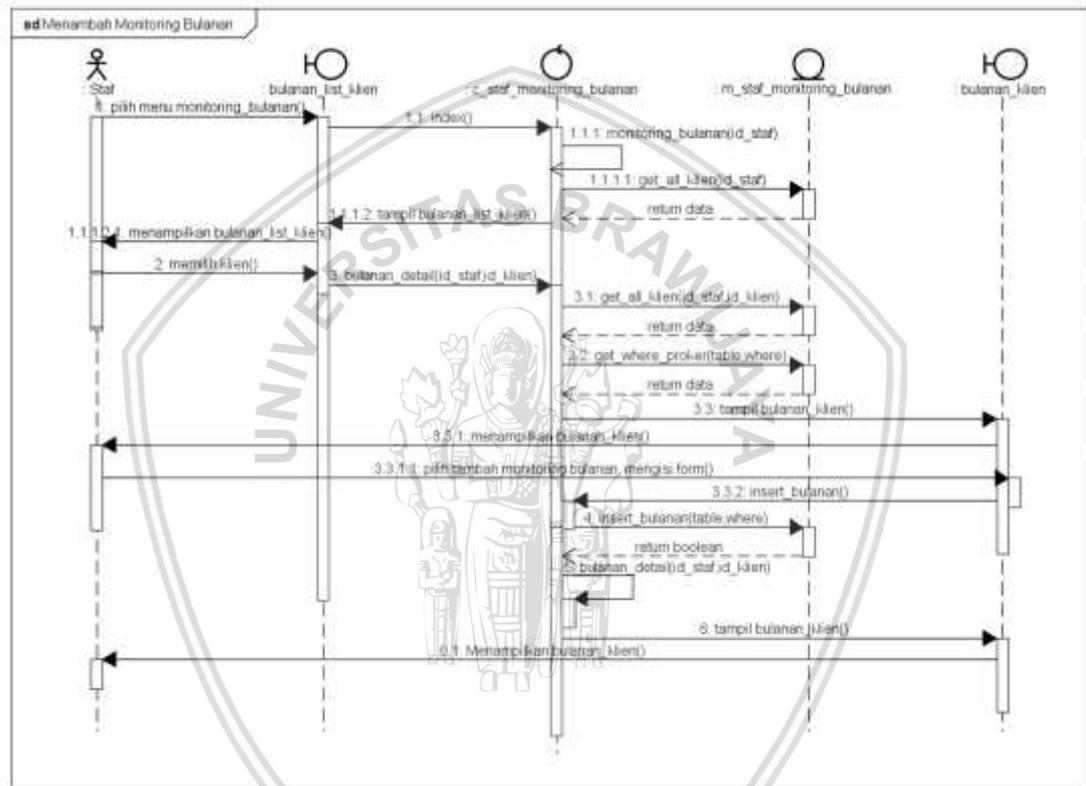


Gambar 5.4 Sequence Diagram Kelola Monitoring Aktivitas



### 5.2.4 Sequence Diagram Menambah Monitoring Bulanan

Sequence diagram kelola *monitoring* bulanan mengacu pada *use case scenario* dengan kode UC-KKP-5. Alur ini diawali dengan pengguna memilih klien yang akan dikelola pada halaman menu *monitoring* bulanan klien yang menampilkan daftar klien. Kemudian sistem akan menampilkan data *monitoring* bulanan klien. Pengguna bisa menambahkan data *monitoring* bulanan klien dengan mengisikan form-form yang tersedia selanjutnya pilih submit. Kemudian sistem akan memanggil fungsi `insert_bulanan()` dan menyimpan datanya ke dalam *database*. Kemudian sistem akan menampilkan data *monitoring* bulanan klien yang sudah ditambahkan.

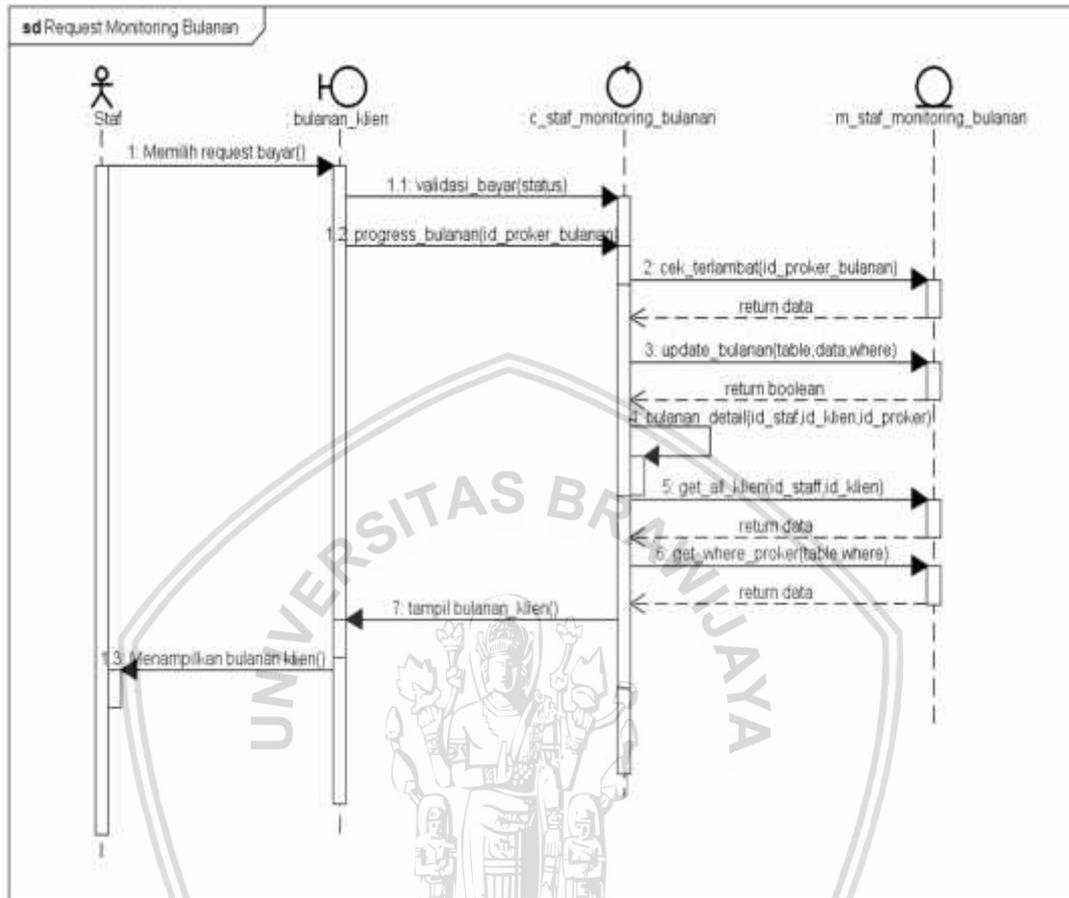


Gambar 5.5 Sequence Diagram Menambah Monitoring Bulanan

### 5.2.5 Sequence Diagram Request Monitoring Bulanan

Sequence diagram *request monitoring* bulanan mengacu pada *use case scenario* dengan kode UC-KKP-6. Bagian ini menjelaskan mengenai *request* validasi status pembayaran perpajakan bulanan klien. Alur ini diawali dengan pengguna telah mengisi daya pembayaran, data pelaporan dan data lainnya yang berkaitan dengan *monitoring* bulanan. Kemudian staf melakukan *request* validasi atas pekerjaan perpajakan bulanan yang dikerjakan yang ditujukan kepada pimpinan. Selanjutnya sistem akan menjalankan fungsi `validasi_bayar(status)`, selanjutnya menjalankan fungsi `progress_bulanan(id_proker_bulanan)`, kemudian di model akan menjalankan fungsi `cek_terlambat(id_proker_bulanan)`, kemudian memanggil model dengan menjalankan fungsi `update_bulanan(table,data,where)`

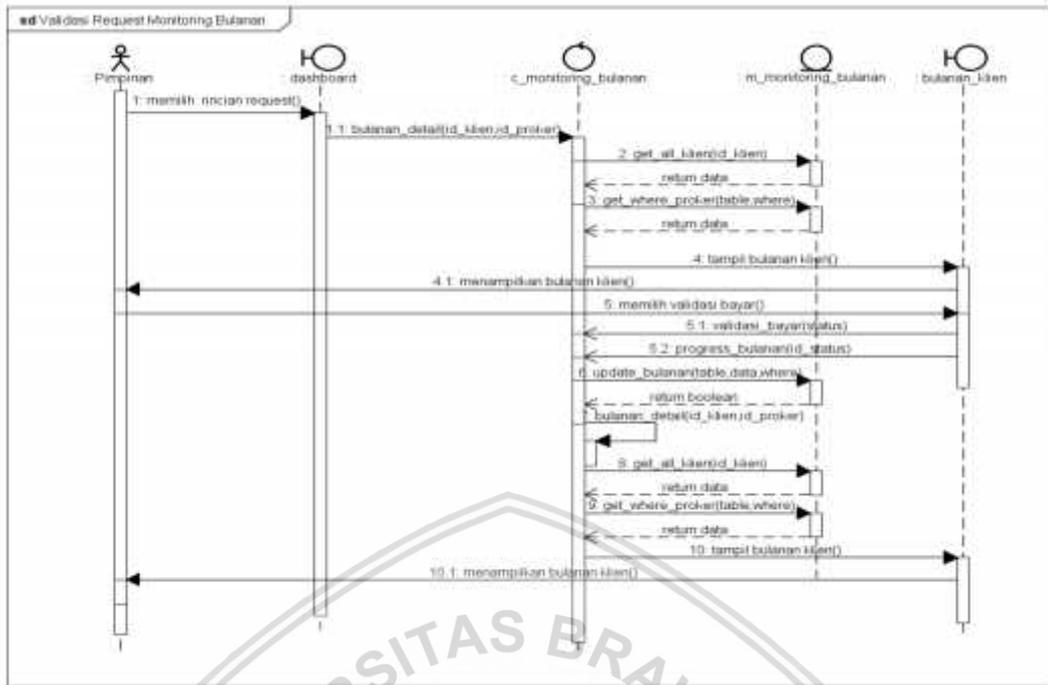
selanjutnya sistem akan kembali memanggil fungsi `bulanan_detail(id_staf,id_klien_id_proker)` dengan menampilkan *monitoring* bulanan detail dan perubahan status *request monitoring* bulanan.



Gambar 5.6 Sequence Diagram Request Monitoring Bulanan

### 5.2.6 Sequence Diagram Validasi Request Monitoring Bulanan

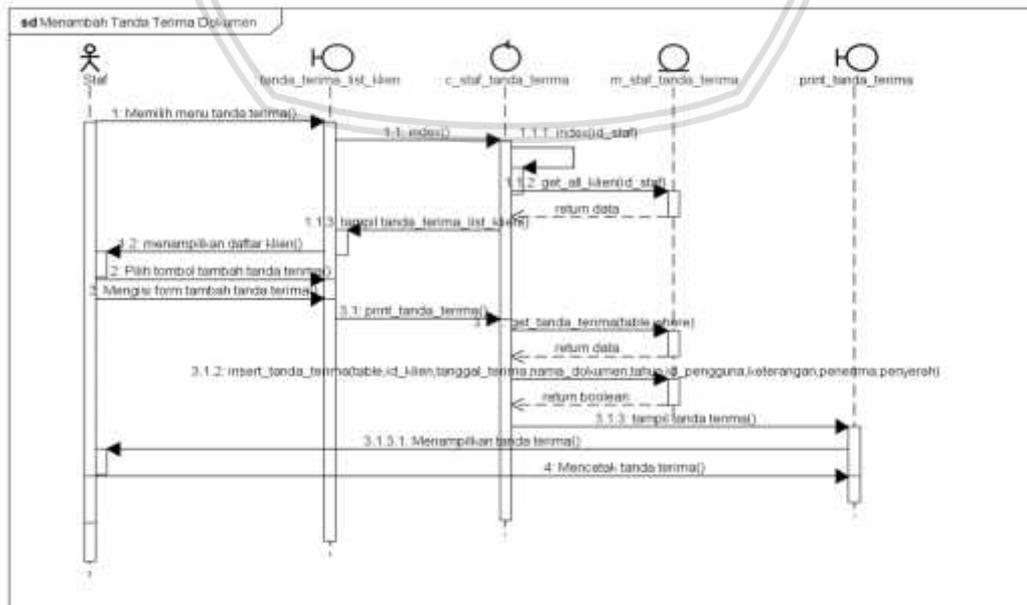
Sequence diagram validasi *request monitoring* bulanan mengacu pada *use case scenario* dengan kode UC-KKP-7. Pimpinan bisa melihat daftar *request* validasi perpajakan bulanan atau tahunan pada menu dashboard. Pengguna memilih rincian klien yang akan divalidasi, kemudian sistem menampilkan halaman *monitoring* bulanan detail. Pimpinan mengecek data-data dan melakukan validasi (pembayaran), selanjutnya sistem menjalankan fungsi `validasi_bayar(status)` kemudian fungsi `progress_bulanan(id_status)`. kemudian memanggil model dengan menjalankan fungsi `update_bulanan(table,data,where)` selanjutnya sistem akan kembali memanggil fungsi `bulanan_detail(id_klien,id_proker)` dengan menampilkan *monitoring* bulanan detail dan perubahan status validasi *request monitoring* bulanan.



Gambar 5.7 Sequence Diagram Validasi Request Monitoring Bulanan

### 5.2.7 Sequence Diagram Menambah Tanda Terima Dokumen

Sequence diagram menambah tanda terima dokumen mengacu pada use case scenario dengan kode UC-KKP-8. Alur ini diawali dengan pengguna memilih menu tanda terima. Kemudian pengguna bisa membuat tanda terima dengan memilih tombol tambah, kemudian mengisi form-form dan menekan tombol submit. Selanjutnya sistem akan menjalankan fungsi print\_tanda\_terima(), selanjutnya data tanda terima akan disimpan kedalam database dan sistem menampilkan dan mencetak tanda terima yang dibuat



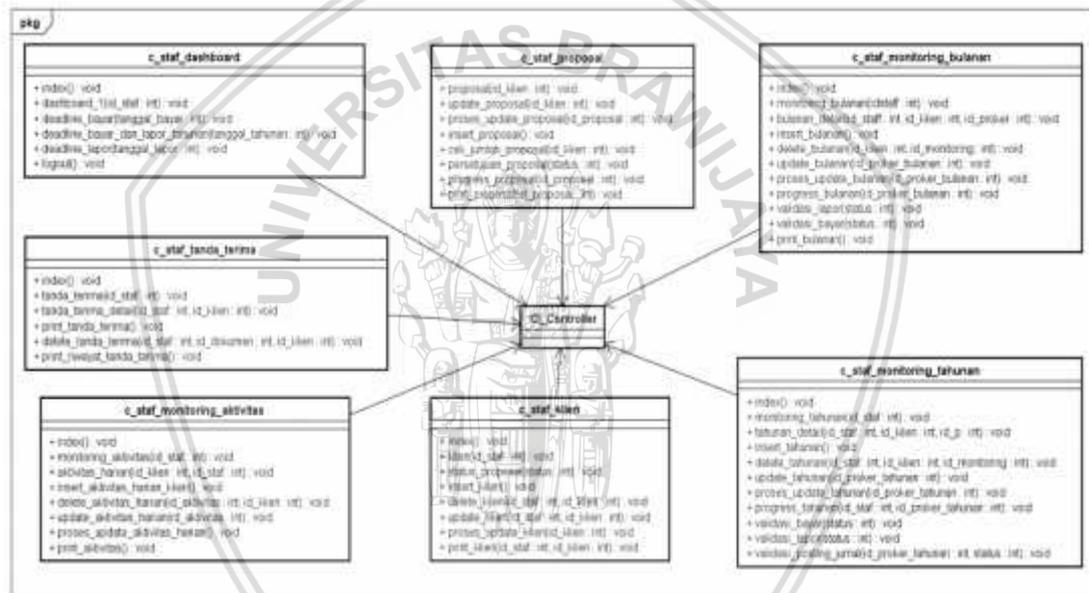
Gambar 5.8 Sequence Diagram Menambah Tanda Terima Dokumen

### 5.3 Perancangan Class Diagram

Class diagram berisi penggambaran objek atau kelas apa saja yang saling berhubungan, berkaitan dan saling bertukar informasi pada Sistem Informasi Monitoring Pekerjaan. Perancangan class diagram pada penelitian ini terbagi dalam dua bagian yaitu perancangan class diagram staf dan perancangan class diagram pimpinan. Selain itu pada bagian ini juga digambarkan class diagram yang digunakan untuk domain model dalam pemodelan skema basis data. Class diagram merupakan gambaran yang akan dijadikan acuan dari kelas-kelas yang akan digunakan dalam membangun Sistem Informasi Monitoring Pekerjaan.

#### 5.3.1 Perancangan Class Diagram Staf

Pada bagian ini menggambarkan class diagram dengan aktor staf. Class diagram di bagian ini dijadikan sebagai acuan untuk membangun kelas-kelas dalam Sistem Informasi Monitoring Pekerjaan.

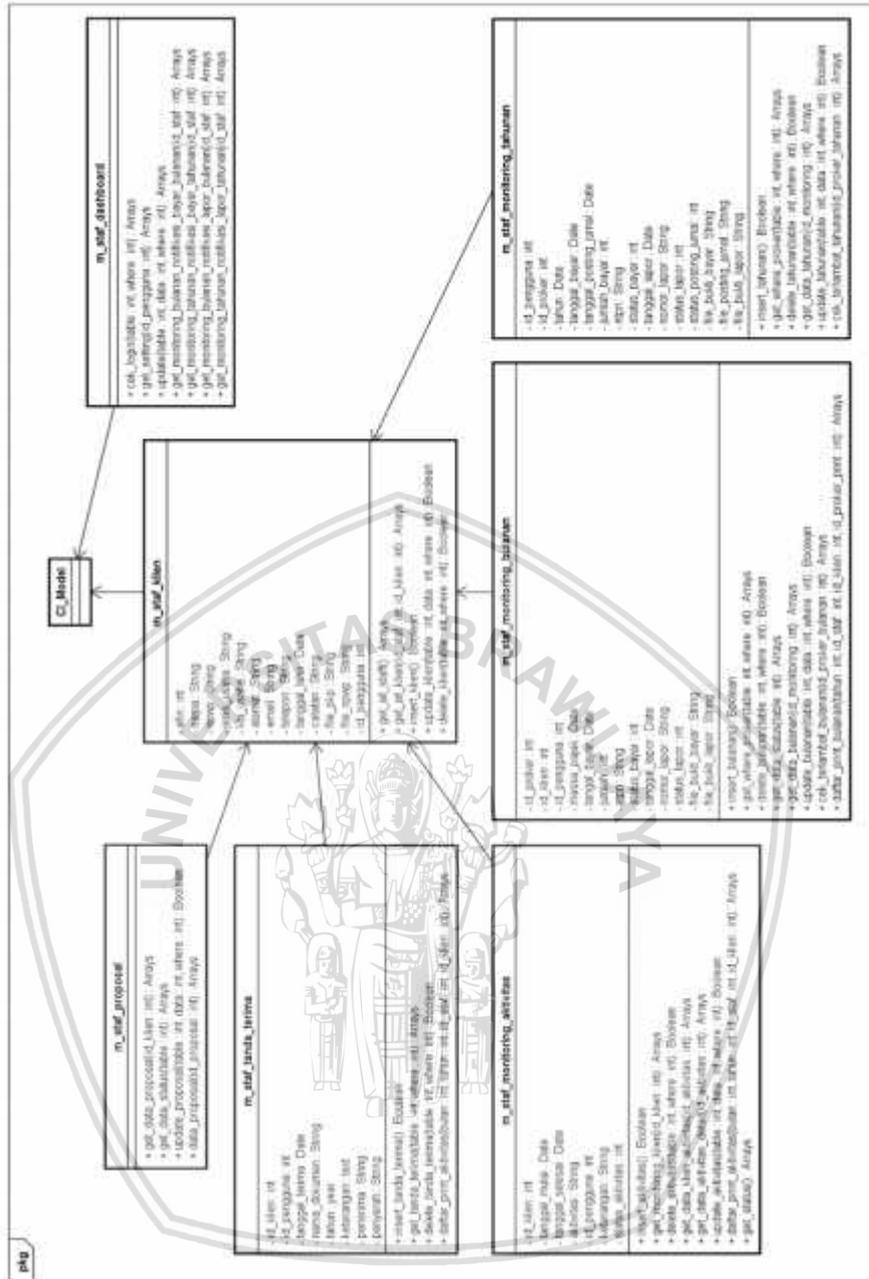


Gambar 5.9 Class Diagram Controller Staf

Gambar 5.9 menunjukkan relasi antara kelas-kelas controller dengan aktor staf. Terdapat 7 kelas untuk aktor pimpinan dalam membangun Sistem Informasi Monitoring Pekerjaan, yaitu

1. c\_staf\_dashboard
2. c\_staf\_proposal
3. c\_staf\_monitoring\_bulanan
4. c\_staf\_monitoring\_tahunan
5. c\_staf\_tanda\_terima
6. c\_staf\_monitoring\_aktivitas
7. c\_staf\_klien



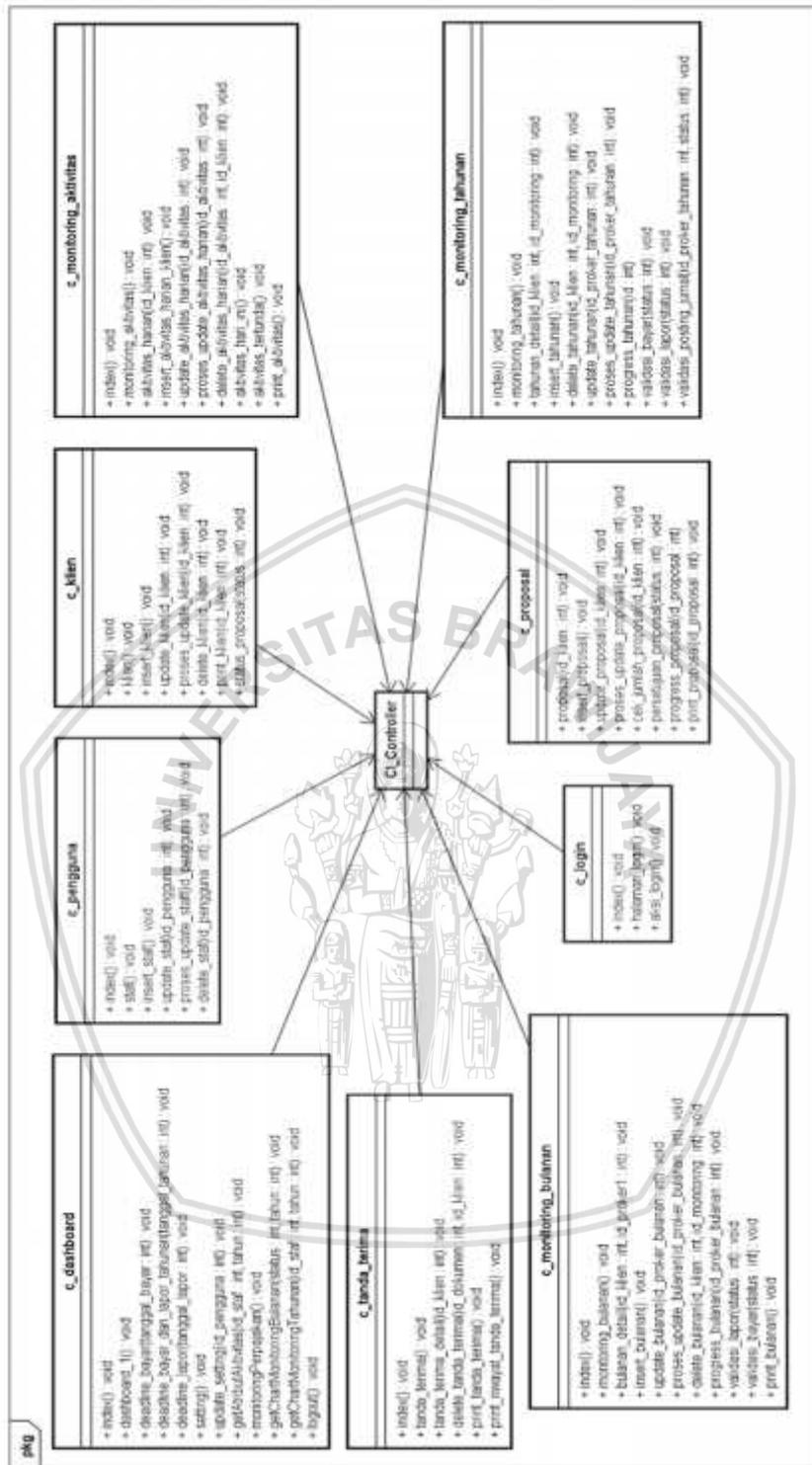


Gambar 5.10 Class Diagram Model Staf

Gambar 5.10 menunjukkan relasi antara kelas-kelas *model* dengan aktor staf. Terdapat 7 kelas untuk aktor staf dalam membangun Sistem Informasi *Monitoring Pekerjaan*, yaitu *m\_staf\_dashboard*, *m\_staf\_klien*, *m\_staf\_tanda\_terima*, *m\_staf\_monitoring\_aktivitas*, *m\_staf\_monitoring\_bulanan*, *m\_staf\_monitoring\_tahunan*, *m\_staf\_proposal*, dan *m\_staf\_tanda\_terima*.

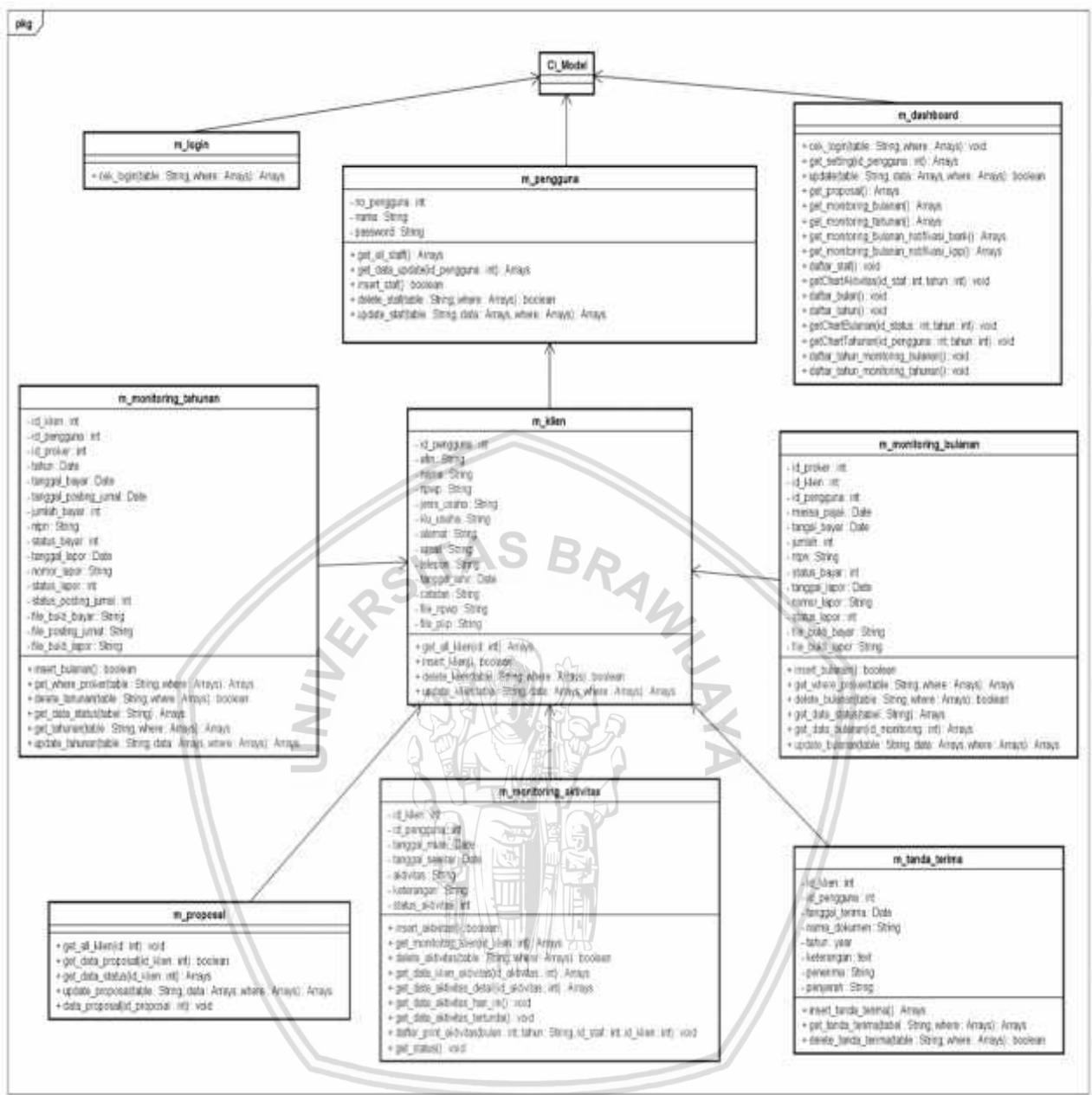
### 5.3.2 Perancangan Class Diagram Pimpinan

Pada bagian ini menggambarkan *class diagram* dengan aktor pimpinan. *Class diagram* di bagian ini dijadikan sebagai acuan untuk membangun kelas-kelas dalam Sistem Informasi *Monitoring Pekerjaan*.



Gambar 5.11 Class Diagram Controller Pimpinan

Gambar 5.11 menunjukkan relasi antara kelas-kelas *controller*. Terdapat 9 kelas untuk aktor pimpinan dalam membangun Sistem Informasi *Monitoring Pekerjaan*, yaitu *c\_login*, *c\_dashboard*, *c\_pengguna*, *c\_proposal*, *c\_klien*, *c\_monitoring\_aktivitas*, *c\_monitoring\_bulanan*, *c\_monitoring\_tahunan*, *c\_proposal*, *c\_tanda\_terima*.

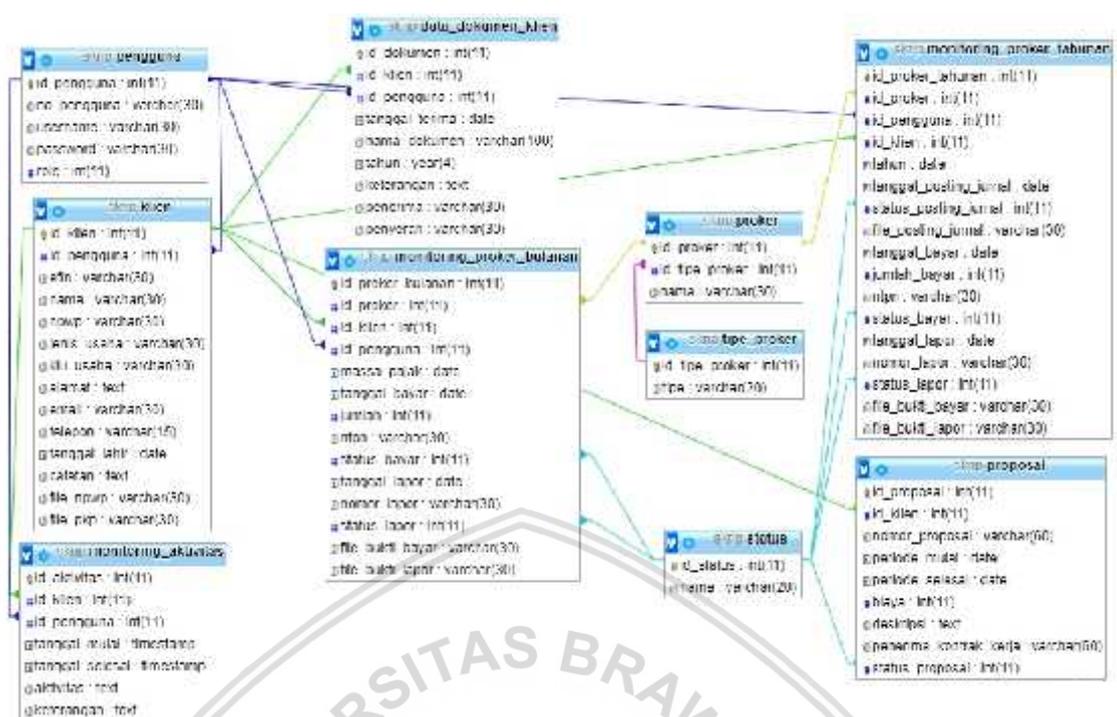


Gambar 5.12 Class Diagram Model Pimpinan

Gambar 5.12 menunjukkan relasi antara kelas-kelas *model*. Terdapat 9 kelas untuk aktor pimpinan dalam membangun Sistem Informasi *Monitoring Pekerjaan*, yaitu *m\_login*, *m\_dashboard*, *m\_staf*, *m\_klien*, *m\_monitoring\_aktivitas*, *m\_monitoring\_bulanan*, *m\_monitoring\_tahunan*, *m\_proposal*, *m\_tanda\_terima*.

**5.4 Perancangan Skema Basis Data**

Skema basis data merupakan penggambaran untuk menunjukkan tabel-tabel, relasi dan atribut di setiap tabel yang digunakan sebagai acuan untuk membangun basis data Sistem Informasi *Monitoring Pekerjaan*



Gambar 5.13 Perancangan Skema Basis Data

Pada Gambar 5.13 terdapat 10 tabel dalam basis data Sistem Informasi *Monitoring Pekerjaan* yang terdiri dari tabel pengguna, klien, *monitoring* aktivitas, data dokumen klien, *monitoring* proker bulanan, proker, tipe proker, status, *monitoring* proker tahunan. Keterangan untuk masing-masing tabel dan atribut dijelaskan sebagai berikut.

Tabel 5.1 Struktur Tabel Pengguna

Nama Atribut	Tipe	Lebar	Keterangan
id_pengguna	int	11	Sebagai <i>primary key</i>
no_pengguna	varchar	30	Berisi nomor identitas pengguna
username	varchar	30	Berisi nama dari pengguna
password	varchar	30	Berisi password dari pengguna
role	int	11	Berisi peran yang membedakan hak akses dari pengguna

Tabel klien merupakan penyimpanan dari data klien, terdapat kolom sebagai *primary key* adalah id\_klien. Keterangan lebih lengkap untuk struktur tabel klien ditunjukkan pada Tabel 5.2 sebagai berikut.

Tabel 5.2 Struktur Tabel Klien

Nama	Tipe	Lebar	Keterangan
id_klien	int	11	Sebagai <i>primary key</i>
id_pengguna	int	11	Merupakan <i>foreign key</i> dari tabel pengguna
efin	varchar	30	Berisi no efin klien sebagai nomor identitas wajib pajak
nama	varchar	30	Berisi nama dari klien
npwp	varchar	30	berisi npwp dari klien
jenis_usaha	varchar	30	Berisi jenis usaha milik klien
klu_usaha	varchar	30	Berisi klu (klasifikasi lapangan usaha)
alamat	text		Berisi alamat dari klien
email	varchar	30	Berisi email dari klien
telepon	varchar	30	Berisi nomer telepon dari klien
tanggal_lahir	date		Berisi tanggal lahir pimpinan klien
catatan	text		Berisi catatan tertentu yang dibutuhkan oleh kantor konsultan pajak
file_npwp	varchar	30	Sebagai penamaan file npwp untuk upload file npwp
file_pkp	varchar	30	Sebagai penamaan file pkp untuk upload file pkp

Tabel *monitoring* aktivitas merupakan penyimpanan dari data *monitoring* konsultasi klien, terdapat kolom sebagai *primary key* adalah id\_konsultasi. Keterangan lebih lengkap untuk struktur tabel *monitoring* aktivitas ditunjukkan pada Tabel 5.3 sebagai berikut.

Tabel 5.3 Struktur Tabel *Monitoring* Aktivitas

Nama	Tipe	Lebar	Keterangan
id_aktivitas	int	11	Sebagai <i>primary key</i>
id_klien	int	11	Merupakan <i>foreign key</i> dari tabel klien
id_pengguna	int	11	Merupakan <i>foreign key</i> dari tabel pengguna
tanggal_mulai	timestamp		Berisi tanggal mulai ketika konsultasi

**Tabel 5.3 Struktur Tabel *Monitoring* Aktivitas (lanjutan)**

Tanggal_selesai	timestamp		Berisi tanggal selesai ketika konsultasi
aktivitas	text		Kegiatan, pembahasan ataupun masalah ketika konsultasi dilakukan
keterangan	text		Berisi keterangan tambahan yang diperlukan oleh kantor atau bisa berupa penjelasan pembahasan masalah yang dikonsultasikan

Tabel data dokumen klien merupakan penyimpanan dari data serah terima dokumen klien, terdapat kolom sebagai *primary key* adalah *id\_dokumen*. Keterangan lebih lengkap untuk struktur tabel data dokumen klien ditunjukkan pada Tabel 5.4 sebagai berikut.

**Tabel 5.4 Struktur Tabel Data Dokumen Klien**

Nama	Tipe	Lebar	Keterangan
id_dokumen	int	11	Sebagai <i>primary key</i>
id_klien	int	11	Merupakan <i>foreign key</i> dari tabel klien
id_pengguna	int	11	Merupakan <i>foreign key</i> dari tabel pengguna
tanggal_terima	date		Berisi tanggal serah terima dokumen
nama_dokumen	text		Berisi nama dokumen
tahun	year	4	Berisi tahun jenis dokumen
keterangan	text		Berisi keterangan tambahan yang dibutuhkan untuk penjelasan dokumen serah terima
penerima	varchar	30	Berisi nama penerima
peserah	varchar	30	Berisi nama orang yang menyerahkan

Tabel tipe proker merupakan penyimpanan dari tipe perpajakan bulanan atau perpajakan tahunan, terdapat kolom sebagai *primary key* adalah *id\_tipe\_proker*. Keterangan lebih lengkap untuk struktur tabel tipe proker ditunjukkan pada Tabel 5.5 sebagai berikut.

**Tabel 5.5 Struktur Tabel Tipe Proker**

Nama	Tipe	Lebar	Keterangan
id_tipe_proker	int	11	Merupakan <i>primary key</i>
tipe	varchar	30	Berisi nama perpajakan bulanan atau perpajakan tahunan

Tabel proker merupakan penyimpanan dari jenis pekerjaan yang berkaitan dengan perpajakan bulanan atau perpajakan tahunan, terdapat kolom sebagai *primary key* adalah id\_proker. Keterangan lebih lengkap untuk struktur tabel proker ditunjukkan pada Tabel 5.6 sebagai berikut.

**Tabel 5.6 Struktur Tabel Proker**

Nama	Tipe	Lebar	Keterangan
id_proker	int	11	Sebagai <i>primary key</i>
id_tipe_proker	int	11	Merupakan <i>foreign key</i> dari tabel tipe proker
nama	varchar	30	Berisi nama untuk jenis perpajakan bulanan atau perpajakan tahunan

Tabel status merupakan penyimpanan dari validasi pekerjaan perpajakan oleh staf dan pimpinan, terdapat kolom sebagai *primary key* adalah id\_status. Keterangan lebih lengkap untuk struktur tabel status ditunjukkan pada Tabel 5.7 sebagai berikut.

**Tabel 5.7 Struktur Tabel Status**

Nama	Tipe	Lebar	Keterangan
id_status	int	11	Sebagai <i>primary key</i>
nama	varchar	30	Berisi nama perubahan validasi seperti belum selesai, proses dan selesai

Tabel proposal merupakan penyimpanan dari data persetujuan proposal antara klien dan kantor konsultan pajak, terdapat kolom sebagai *primary key* adalah id\_proposal. Keterangan lebih lengkap untuk struktur tabel proposal ditunjukkan pada Tabel 5.8 sebagai berikut.

**Tabel 5.8 Struktur Tabel Proposal**

Nama	Tipe	Lebar	Keterangan
id_proposal	int	11	Sebagai <i>primary key</i>
id_klien	int	11	Merupakan <i>foreign key</i> dari tabel klien
Nomor_proposal	varchar	60	Berisi nomor proposal
periode_mulai	date		Berisi tanggal awal perjanjian proposal

Tabel 5.8 Struktur Tabel Proposal (lanjutan)

periode_selesai	date		Berisi tanggal akhir perjanjian proposal
biaya	int	11	Berisi biaya yang dibayarkan klien kepada kantor konsultan pajak
deskripsi	text		Berisi rincian penjelasan yang lebih lengkap untuk perjanjian dalam proposal
penerima_kontrak_kerja	varchar	30	Berisi nama orang yang bekerjasama dengan pihak kantor konsultan pajak dalam proposal
status_proposal	int	1	Berisi status persetujuan yang dari pembuatan proposal oleh staf kepada pimpinan

Tabel *monitoring* proker bulanan merupakan penyimpanan dari *monitoring* perpajakan bulanan klien yang dilakukan oleh staf, terdapat kolom sebagai *primary key* adalah *id\_proker\_bulanan*. Keterangan lebih lengkap untuk struktur tabel *monitoring* proker bulanan ditunjukkan pada Tabel 5.9 sebagai berikut.

Tabel 5.9 Struktur Tabel *Monitoring Proker Bulanan*

Nama	Tipe	Lebar	Keterangan
id_proker_bulanan	int	11	Sebagai <i>primary key</i>
id_proker	int	11	Merupakan <i>foreign key</i> dari tabel proker
id_klien	int	11	Merupakan <i>foreign key</i> dari tabel klien
id_pengguna	int	11	Merupakan <i>foreign key</i> dari tabel pengguna
massa_pajak	date		Berisi massa pajak yang sudah ditentukan
tanggal_bayar	date		Berisi tanggal maksimal pembayaran pajak
jumlah	int	11	Berisi nominal jumlah pajak
ntpn	varchar	30	berisi NTPN (Nomor Transaksi Penerimaan Negara)
status_bayar	int	11	berisi status validasi pembayaran perpajakan bulanan yang dilakukan oleh staf kepada pimpinan

Tabel 5.9 Struktur Tabel *Monitoring Proker Bulanan* (lanjutan)

tanggal_lapor	date		Berisi tanggal maksimal pelaporan pajak
nomor_lapor	varchar	30	Berisi nomor lapor pajak yang sudah dilaporkan tentang perpajakan bulanan
status_lapor	int	11	berisi status validasi pelaporan perpajakan bulanan yang dilakukan oleh staf kepada pimpinan
file_bukti_bayar	varchar	30	Sebagai penamaan file bukti bayar untuk upload file bukti bayar
file_bukti_lapor	varchar	30	Sebagai penamaan file bukti lapor untuk upload file bukti lapor

Tabel *monitoring* proker tahunan merupakan penyimpanan dari *monitoring* perpajakan tahunan klien yang dilakukan oleh staf, terdapat kolom sebagai *primary key* adalah *id\_proker\_tahunan*. Keterangan lebih lengkap untuk struktur tabel *monitoring* proker tahunan ditunjukkan pada Tabel 5.10 sebagai berikut.

Tabel 5.10 Struktur Tabel *Monitoring Proker Tahunan*

Nama	Tipe	Lebar	Keterangan
id_proker_tahunan	int	11	Sebagai <i>primary key</i>
id_proker	int	11	Merupakan <i>foreign key</i> dari tabel proker
id_klien	int	11	Merupakan <i>foreign key</i> dari tabel klien
id_pengguna	int	11	Merupakan <i>foreign key</i> dari tabel pengguna
tahun	year	4	Berisi massa pajak yang sudah ditentukan
tanggal_posting_jurnal	date		Berisi tanggal membuat jurnal
status_posting_jurnal	int	11	Berisi status validasi posting jurnal yang dilakukan oleh staf dan pimpinan
file_posting_jurnal	varchar	30	Sebagai penamaan file untuk upload file posting jurnal
tanggal_bayar	date		Berisi tanggal maksimal pembayaran pajak
jumlah_bayar	int	11	Berisi nominal jumlah pajak

Tabel 5.10 Struktur Tabel *Monitoring Proker Tahunan* (lanjutan)

ntpn	varchar	30	berisi NTPN (Nomor Transaksi Penerimaan Negara)
status_bayar	int	11	Berisi status validasi pembayaran perpajakan bulanan yang dilakukan oleh staf kepada pimpinan
tanggal_lapor	date		Berisi tanggal maksimal pelaporan pajak
nomor_lapor	varchar	30	Berisi nomor lapor pajak yang sudah dilaporkan tentang perpajakan bulanan
status_lapor	int	11	Berisi status validasi pelaporan perpajakan bulanan yang dilakukan oleh staf kepada pimpinan
file_bukti_bayar	varchar	30	Sebagai penamaan file bukti bayar untuk upload file bukti bayar
file_bukti_lapor	varchar	30	Sebagai penamaan file bukti lapor untuk upload file bukti lapor

Tabel bulan merupakan untuk pembuatan daftar bulan di chart, terdapat kolom sebagai *primary key* adalah *id\_bulan*. Keterangan lebih lengkap untuk struktur tabel status ditunjukkan pada Tabel 5.11 sebagai berikut.

Tabel 5.11 Struktur Tabel Status

Nama	Tipe	Lebar	Keterangan
id_bulan	int	11	Sebagai <i>primary key</i>
nama_bulan	varchar	30	Berisi nama perubahan validasi seperti belum selesai, proses dan selesai

## 5.5 Perancangan Antarmuka Aplikasi

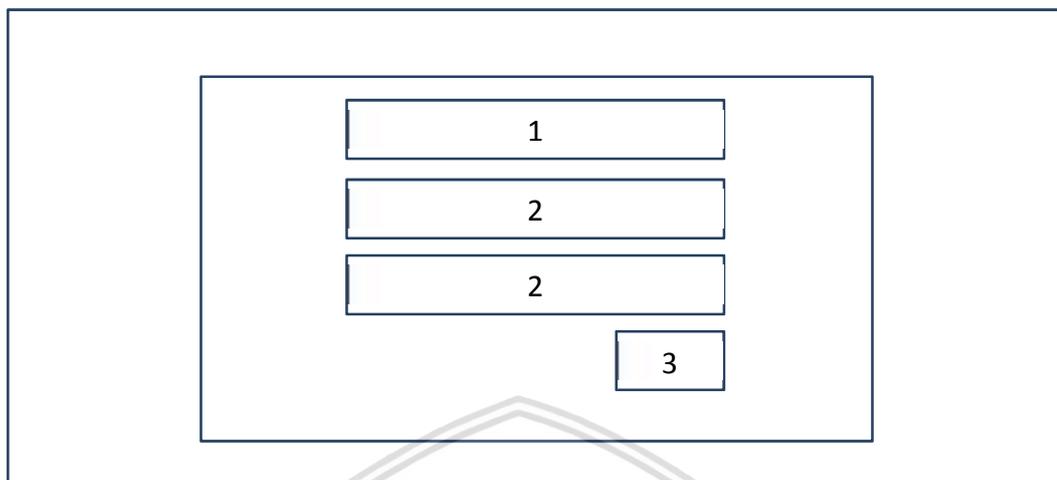
Bagian ini merupakan perancangan yang dilakukan untuk menggambarkan tampilan dari perangkat lunak yang akan dibangun. Berikut ini rancangan antarmuka Sistem Informasi *Monitoring* Pekerjaan:

### 5.5.1 Rancangan Antarmuka Login Pengguna

Halaman *login* merupakan halaman yang pertama kali dilihat oleh pengguna, login dilakukan untuk memberikan akses kepada pengguna untuk masuk ke sistem utama. Berikut Gambar 5.14 merupakan rancangan antarmuka login untuk pengguna beserta keterangan sebagai berikut :

1. Logo Kantor konsultan pajak I Gede Arianta

2. Form login (Username dan Password)
3. Tombol Submit

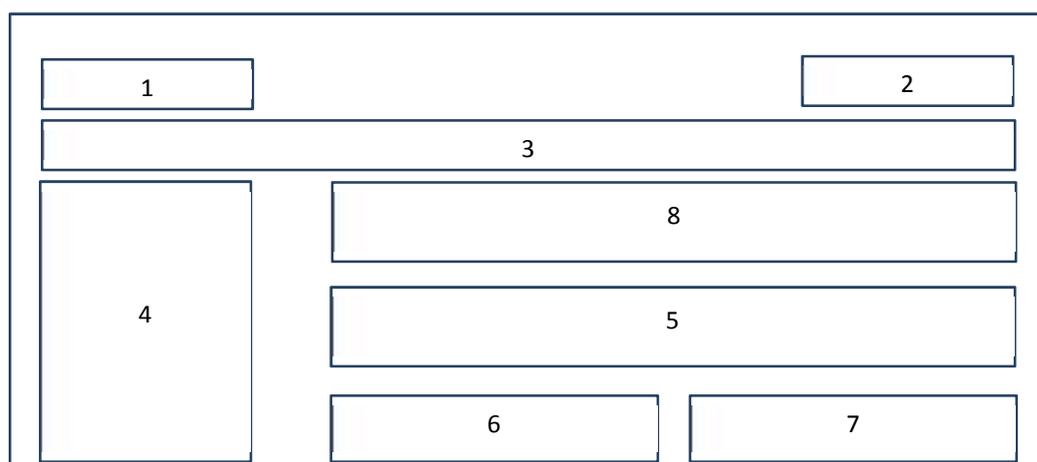


Gambar 5.14 Rancangan Antarmuka Login Pengguna

### 5.5.2 Rancangan Antarmuka *Dashboard* Pimpinan

Halaman *dashboard* bagi pimpinan merupakan halaman untuk menampilkan *request* pekerjaan dari staf dan notifikasi jadwal pemberitahuan atas pembayaran dan pelaporan pajak. Berikut Gambar 5.15 merupakan rancangan antarmuka *dashboard* beserta keterangan sebagai berikut:

1. Logo Kantor konsultan pajak I Gede Arianta
2. Status login, menu *logout* dan *setting*
3. *Icon* dan keterangan *dashboard*
4. Daftar menu
5. Menampilkan *request* proses pekerjaan dari staf
6. Menampilkan notifikasi bank yang belum bayar sesuai dengan batas jadwal yang sudah ditentukan beserta staf yang bertanggung jawab
7. Menampilkan notifikasi lapor yang belum lapor sesuai dengan batas jadwal yang sudah ditentukan beserta staf yang bertanggung jawab
8. Menampilkan grafik *monitoring* jumlah konsultasi yang dilakukan oleh staf

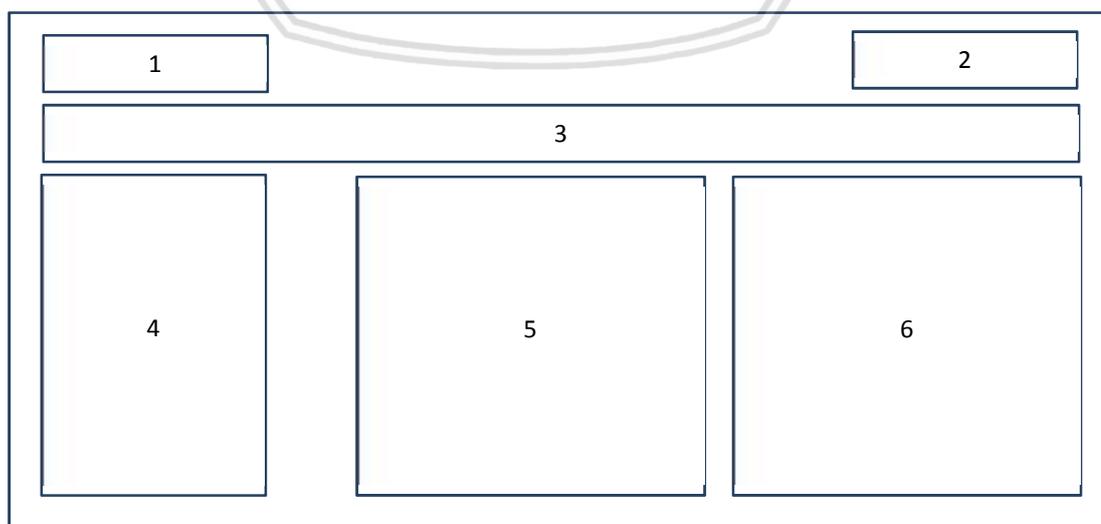


**Gambar 5.15 Rancangan Antarmuka *Dashboard* Pimpinan**

### 5.5.3 Rancangan Antarmuka *Dashboard* Staf

Halaman *dashboard* bagi staf merupakan halaman untuk menampilkan notifikasi jadwal pemberitahuan atas pembayaran dan pelaporan pajak klien. Berikut Gambar 5.16 merupakan rancangan antarmuka *dashboard* beserta keterangan sebagai berikut:

1. Logo Kantor konsultan pajak I Gede Arianta
2. Status login dan menu *logout*
3. *Icon* dan keterangan *dashboard*
4. List menu
5. Menampilkan notifikasi bank yang belum bayar sesuai dengan batas jadwal yang sudah ditentukan
6. Menampilkan notifikasi lapor yang belum lapor sesuai dengan batas jadwal yang sudah ditentukan

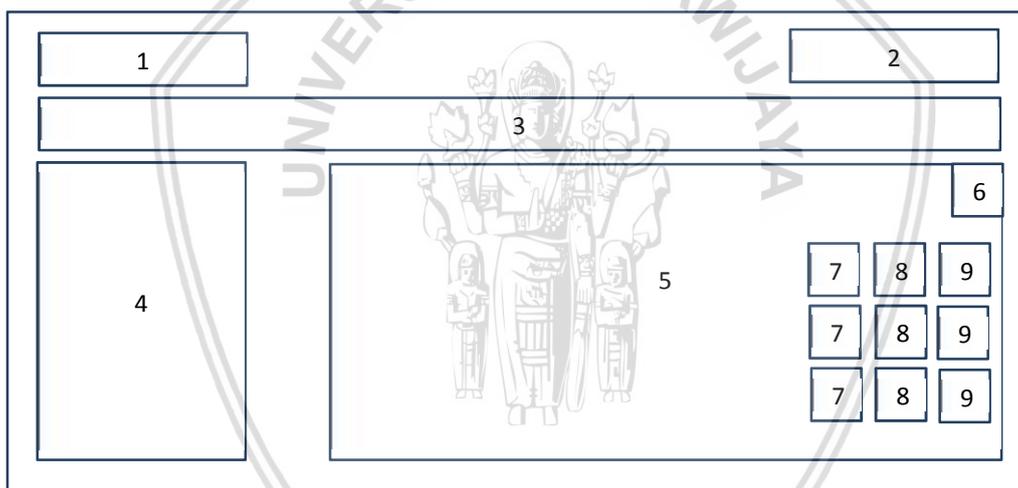


**Gambar 5.16 Rancangan Antarmuka *Dashboard* Staf**

### 5.5.4 Rancangan Antarmuka Kelola Klien

Halaman kelola klien merupakan halaman yang digunakan oleh pengguna untuk mengelola data klien yang ada di kantor konsultan pajak. Berikut Gambar 5.17 merupakan rancangan antarmuka kelola klien beserta sebagai berikut:

1. Logo Kantor konsultan pajak I Gede Arianta
2. Status login, menu *logout* ( jika pimpinan ada tambahan menu pengaturan)
3. *Icon* dan keterangan klien
4. List menu
5. Tabel daftar data klien
6. Menambah data klien
7. Mengubah data klien sesuai dengan yang dipilih
8. Menghapus data klien sesuai dengan yang dipilih
9. Mencetak data klien sesuai dengan yang dipilih



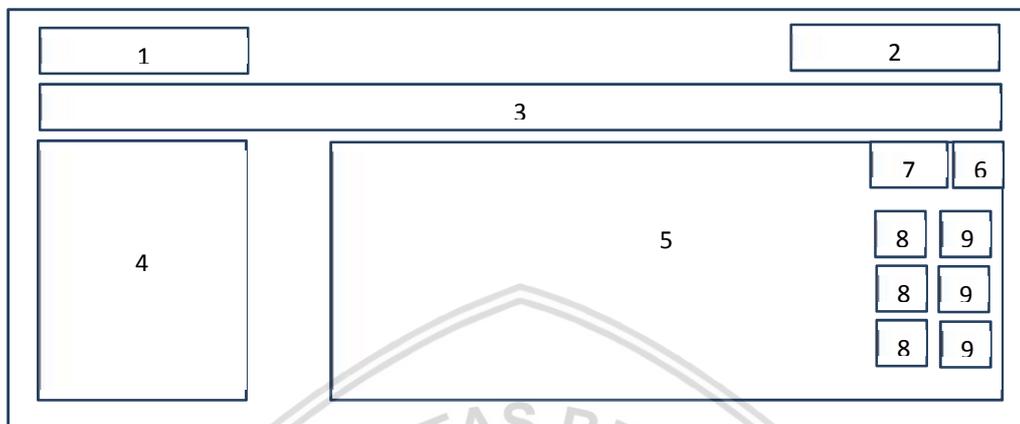
Gambar 5.17 Rancangan Antarmuka Kelola Klien

### 5.5.5 Rancangan Antarmuka Kelola *Monitoring* Aktivitas

Halaman kelola *monitoring* aktivitas yang digunakan oleh pengguna untuk mengelola data konsultasi klien yang ada di kantor konsultan pajak. Berikut Gambar 5.18 merupakan rancangan antarmuka kelola *monitoring* aktivitas beserta keterangan sebagai berikut:

1. Logo Kantor konsultan pajak I Gede Arianta
2. Status login, menu *logout* (jika pimpinan ada tambahan menu pengaturan)
3. *Icon* dan keterangan *monitoring* aktivitas
4. List menu
5. Tabel daftar data konsultasi

6. Menambah data konsultasi
7. Mencetak data konsultasi
8. Mengubah data konsultasi sesuai dengan yang dipilih
9. Menghapus data konsultasi sesuai dengan yang dipilih

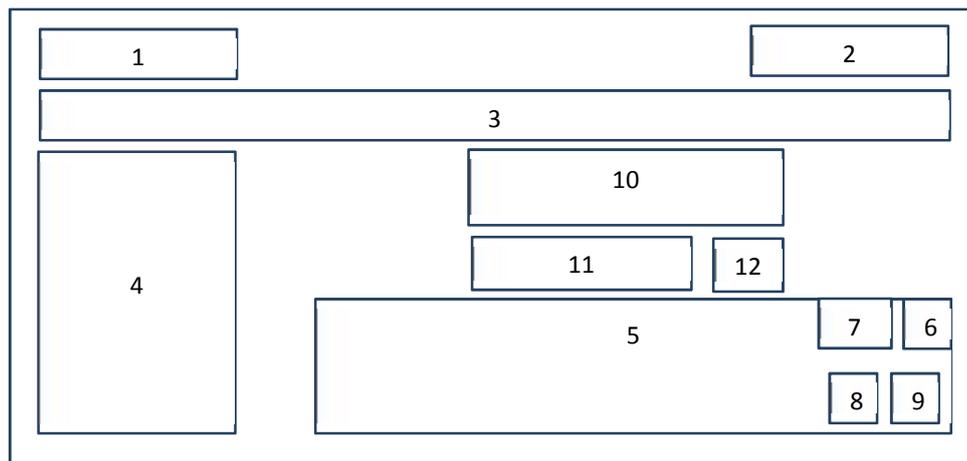


**Gambar 5.18 Rancangan Antarmuka Kelola *Monitoring* Aktivitas**

### 5.5.6 Rancangan Antarmuka Kelola *Monitoring* Bulanan Dan Tahunan

Halaman kelola *monitoring* bulanan dan tahunan yang digunakan oleh pengguna untuk mengelola data perpajakan bulanan atau perpajakan tahunan klien yang ada di kantor konsultan pajak. Berikut Gambar 5.19 merupakan rancangan antarmuka kelola *monitoring* bulanan dan tahunan beserta keterangan sebagai berikut:

1. Logo Kantor konsultan pajak I Gede Arianta
2. Status login, menu *logout* (jika pimpinan ada tambahan menu pengaturan)
3. *Icon* dan keterangan *monitoring* bulanan atau tahunan
4. List menu
5. Tabel daftar data *monitoring* perpajakan bulanan dan terdapat proses *request* validasi *monitoring* perpajakan bulanan atau tahunan
6. Menambah data *monitoring* perpajakan bulanan atau tahunan
7. Mencetak data *monitoring* perpajakan bulanan atau tahunan
8. Mengubah data *monitoring* perpajakan bulanan atau tahunan sesuai dengan yang dipilih
9. Menghapus data *monitoring* perpajakan bulanan atau tahunan sesuai dengan yang dipilih
10. Menampilkan beberapa data klien sesuai dengan yang diproses
11. Daftar jenis pekerjaan perpajakan bulanan atau tahunan
12. Tombol submit

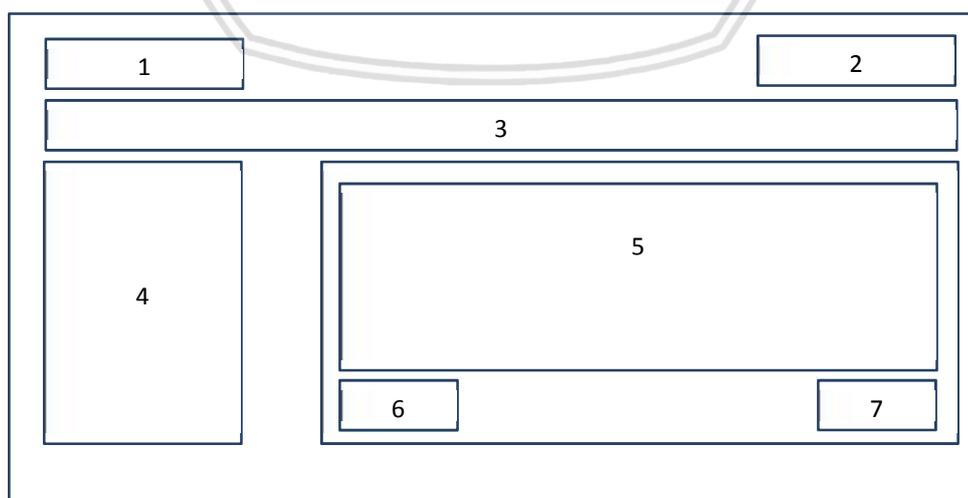


Gambar 5.19 Rancangan Antarmuka Kelola *Monitoring* Bulanan Dan Tahunan

### 5.5.7 Rancangan Antarmuka Tambah Tanda Terima Dokumen

Halaman tambah tanda terima dokumen digunakan oleh staf untuk membuat dan menyimpan tanda terima dokumen yang berkaitan dengan serah terima dokumen klien. Berikut Gambar 5.20 merupakan rancangan antarmuka tambah tanda terima dokumen beserta keterangan sebagai berikut:

1. Logo Kantor konsultan pajak I Gede Arianta
2. Status login dan menu *logout*
3. *Icon* dan keterangan tanda terima dokumen
4. List menu
5. Form-form untuk data tanda terima berupa nama klien, nama dokumen, tahun, keterangan, klien yang menyerahkan, staf yang menerima
6. Tombol back untuk membatalkan pembuatan tanda terima
7. Submit



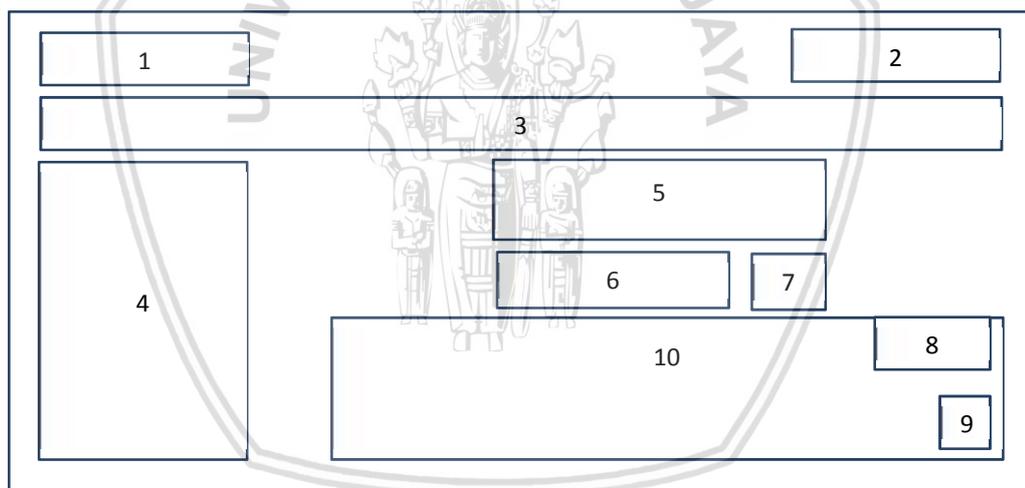
Gambar 5.20 Rancangan Antarmuka Tambah Tanda Terima Dokumen



### 5.5.8 Rancangan Antarmuka Kelola Tanda Terima Dokumen

Halaman kelola tanda terima dokumen digunakan oleh pengguna untuk mengelola data tanda terima dokumen yang telah tersimpan. Berikut Gambar 5.21 merupakan rancangan antarmuka kelola tanda terima dokumen beserta keterangan dari sebagai berikut:

1. Logo Kantor konsultan pajak I Gede Arianta
2. Status login, menu *logout* (jika pimpinan ada tambahan menu pengaturan)
3. *Icon* dan keterangan tanda terima dokumen
4. List menu
5. Menampilkan beberapa data klien sesuai dengan yang diproses
6. Daftar jenis pekerjaan perpajakan bulanan atau tahunan
7. Tombol submit
8. Mencetak riwayat daftar tanda terima yang telah tersimpan
9. Menghapus data tanda terima dokumen sesuai dengan yang dipilih
10. Tabel daftar tanda terima dokumen klien yang sudah tersimpan



Gambar 5.21 Rancangan Antarmuka Kelola Tanda Terima

## BAB 6 IMPLEMENTASI

Pada bab implementasi membahas hasil implementasi Sistem Informasi *Monitoring Pekerjaan* yang merupakan proses realisasi dari analisis kebutuhan dan perancangan yang telah dilakukan oleh peneliti. Pembahasan implementasi akan diuraikan menjadi batasan implementasi, implementasi kode program dan implementasi antarmuka

### 6.1 Batasan Implementasi

Berikut batasan-batasan dalam implementasi Sistem Informasi *Monitoring Pekerjaan Kantor Konsultan Pajak I Gede Arianta* :

1. Sistem Informasi *Monitoring Pekerjaan Kantor Konsultan Pajak I Gede Arianta* dibangun dengan bahasa pemrograman PHP dan menggunakan *framework* CodeIgniter
2. Basis data yang dirancang menggunakan Relational Database Management System MySQL

### 6.2 Implementasi Kode Program

Pada bagian implementasi kode program pada Sistem Informasi *Monitoring Pekerjaan Kantor Konsultan Pajak I Gede Arianta* yang dibangun akan sesuai dengan *sequence diagram* yang sudah dibuat beserta aktor yang teridentifikasi dalam penggunaan sistem.

#### 6.2.1 Implementasi Kelola Data Klien

Pada bagian implementasi kelola data klien sesuai dengan *sequence* kelola data klien dibuat kedalam implementasi kode program pada kelas dan potongan kode program pada Tabel 6.1 berikut

**Tabel 6.1 Implementasi Kelola Data Klien**

No	C_staf_klien.php
1	function index(){
2	\$this->klien();
3	}
4	function klien(\$id_staf){
5	\$data=\$this->m_staf_klien->get_all_klien(\$id_staf);
6	\$this->load->view('staf/klien',array('data'=>\$data));
7	}
8	function status_proposal(\$status){
9	if(\$status==5){
10	?><p style="color:#820303"> Belum Disetujui </p><?php
11	}else if(\$status==6){
12	?><p style="color:#b5a61e"> Proses</p><?php

Tabel 6.1 Implementasi Kelola Data Klien (lanjutan)

No	C_staf_klien.php
13	}else if(\$status==7){
14	?><p style="color:#359356"> Disetujui</p><?php
15	} else{
16	?><p style="color:#b2b2b2"> Belum Disetujui
17	</p><?php
18	}
19	}
20	function insert_klien(){
21	\$this->load->library('upload');
22	\$config['upload_path'] = './file_upload/';
23	\$config['allowed_types'] = 'gif jpg png jpeg';
24	\$config['file_name'] = random_string('alnum',10);
25	\$this->upload->initialize(\$config);
26	\$efin=\$_POST['efin'];
27	\$nama=\$_POST['nama'];
28	\$npwp=\$_POST['npwp'];
29	\$jenis_usaha=\$_POST['jenisUsaha'];
30	\$klu_usaha=\$_POST['klu'];
31	\$alamat=\$_POST['alamat'];
32	\$email=\$_POST['email'];
33	\$telepon=\$_POST['telpon'];
34	\$tanggal_lahir=\$_POST['lahir'];
35	\$catatan=\$_POST['catatan'];
36	\$id_pengguna=\$this->session->userdata('id_pengguna');
37	if(\$_FILES['file_npwp']['size']!=0 ) {
38	if (\$this->upload->do_upload('file_npwp')){
39	\$data_image = \$this->upload->data();
40	\$data['file_npwp']=\$config['upload_path'].\$data
41	a_image['file_name'];
42	}
43	}
44	if(\$_FILES['file_pkp']['size']!=0 ){
45	if (\$this->upload->do_upload('file_pkp')){
46	\$data_image2 = \$this->upload->data();
47	\$data['file_pkp']=\$config['upload_path'].\$data
48	_image2['file_name'];
49	}
50	}
51	
52	\$res = \$this->m_staf_klien->insert_klien(\$efin,\$nama,\$npwp,
53	\$jenis_usaha,\$klu_usaha,\$alamat,\$email,\$telepon,\$tanggal_la

Tabel 6.1 Implementasi Kelola Data Klien (lanjutan)

No	C_staf_klien.php
54	hir,\$catatan,\$data['file_npwp'],\$data['file_pkp'],\$sid_pengg
55	una);
56	if(\$this->db->affected_rows(>0){
57	\$this->session->set_flashdata('pesan','Data berhasil
58	dimasukkan.');
59	redirect('staf/c_staf_klien');
60	}else {
61	\$this->session->set_flashdata('pesan','Data gagal
62	dimasukkan');
63	redirect('staf/c_staf_klien');
64	}
65	}
66	function delete_klien(\$sid_staf,\$sid_klien){
67	\$get_foto_lama = \$this->m_staf_klien-
68	>get_all_klien(\$sid_staf,\$sid_klien);
69	unlink(\$get_foto_lama[0]['file_pkp']);
70	unlink(\$get_foto_lama[0]['file_npwp']);
71	\$where = array('id_klien' => \$sid_klien);
72	\$res = \$this->m_staf_klien->delete_klien('klien',\$where);
73	if(\$res){
74	\$this->session->set_flashdata('pesan','Data berhasil
75	Dihapus.');
76	redirect('staf/c_staf_klien');
77	}else {
78	\$this->session->set_flashdata('pesan','Data gagal
79	Dihapus');
80	redirect('staf/c_staf_klien');
81	}
82	}
83	function update_klien(\$sid_staf,\$sid_klien){
84	\$data = \$this->m_staf_klien-
85	>get_all_klien(\$sid_staf,\$sid_klien);
86	\$this->load->view('staf/edit_klien',array('data'=>\$data));
87	}
88	function proses_update_klien(\$sid_klien){
89	\$this->load->library('upload');
90	\$config['upload_path'] = './file_upload/';
91	\$config['allowed_types'] = 'gif jpg png pdf docx doc';
92	\$config['file_name'] = random_string('alnum',10);
93	\$this->upload->initialize(\$config);
94	if (\$_FILES['file_npwp']['size']>0) {

Tabel 6.1 Implementasi Kelola Data Klien (lanjutan)

No	C_staf_klien.php
95	<code>\$get_foto_lama=\$this-&gt;m_staf_klien-</code>
96	<code>&gt;get_all_klien(\$id_klien);</code>
97	<code>unlink(\$get_foto_lama[0]['file_npwp']);</code>
98	<code>}</code>
99	<code>if (\$_FILES['file_pkp']['size']&gt;0) {</code>
100	<code>\$get_foto_lama=\$this-&gt;m_staf_klien-</code>
101	<code>&gt;get_all_klien(\$id_klien);</code>
102	<code>unlink(\$get_foto_lama[0]['file_pkp']);</code>
103	<code>}</code>
104	<code>\$data = array(</code>
105	<code>'efin'=&gt;\$_POST['efin'],</code>
106	<code>'nama'=&gt;\$_POST['nama'],</code>
107	<code>'npwp'=&gt;\$_POST['npwp'],</code>
108	<code>'jenis_usaha'=&gt;\$_POST['jenis_usaha'],</code>
109	<code>'klu_usaha'=&gt;\$_POST['klu_usaha'],</code>
110	<code>'alamat'=&gt;\$_POST['alamat'],</code>
111	<code>'email'=&gt;\$_POST['email'],</code>
112	<code>'telepon'=&gt;\$_POST['telepon'],</code>
113	<code>'tanggal_lahir'=&gt;\$_POST['tgl_lahir_pimpinan'],</code>
114	<code>'catatan'=&gt;\$_POST['catatan'],</code>
115	<code>'id_pengguna'=&gt;\$this-&gt;session-&gt;userdata('id_pengguna')</code>
116	<code>);</code>
117	<code>if(\$_FILES['file_npwp']['size']!=0){</code>
118	<code>if (\$this-&gt;upload-&gt;do_upload('file_npwp')){</code>
119	<code>\$data_image = \$this-&gt;upload-&gt;data();</code>
120	<code>\$data['file_npwp']=\$config['upload_path'].\$dat</code>
121	<code>a_image['file_name'];</code>
122	<code>}</code>
123	<code>}</code>
124	<code>if(\$_FILES['file_pkp']['size']!=0)</code>
125	<code>{</code>
126	<code>if (\$this-&gt;upload-&gt;do_upload('file_pkp')){</code>
127	<code>\$data_image2 = \$this-&gt;upload-&gt;data();</code>
128	<code>\$data['file_PKP']=\$config['upload_path'].\$data</code>
129	<code>_image2['file_name'];</code>
130	<code>}</code>
131	<code>}</code>
132	<code>}</code>
133	<code>\$res = \$this-&gt;m_staf_klien-</code>
134	<code>&gt;update_klien('klien',\$data,array('id_klien'=&gt;\$id_kl</code>
135	<code>ien));</code>
136	<code>if (\$res) {</code>

**Tabel 6.1 Implementasi Kelola Data Klien (lanjutan)**

No	C_staf_klien.php
137	<code>\$this-&gt;session-&gt;set_flashdata('pesan', 'Data Berhasil</code>
138	<code>Diubah');</code>
139	<code>redirect('staf/c_staf_klien/update_klien/'. \$this-</code>
140	<code>&gt;session-</code>
141	<code>&gt;userdata('id_pengguna')." / ". \$_POST['id_klien']);</code>
142	<code>}</code>
143	<code>else {</code>
144	<code>\$this-&gt;session-&gt;set_flashdata('pesan', 'Data Tidak</code>
145	<code>Berubah');</code>
146	<code>redirect('staf/c_staf_klien/update_klien/'. \$this-</code>
147	<code>&gt;session-&gt;userdata('id_pengguna')." / ". \$_POST['id_klien']);</code>
148	<code>}</code>
149	<code>}</code>
150	<code>function print_klien(\$id_staf, \$id_klien){</code>
151	<code>\$data=\$this-&gt;m_staf_klien-</code>
152	<code>&gt;get_all_klien(\$id_staf, \$id_klien);</code>
153	<code>\$this-&gt;load-&gt;view('staf/print_klien', array('data'=&gt;\$data));</code>
154	<code>}</code>

Kode program pada Tabel 6.1 merupakan kode program yang terdapat pada kelas `c_klien` yang digunakan untuk mengelola data klien dan terdapat fungsi-fungsi yang dibuat untuk mendukung pengelolaan data klien diantaranya untuk melihat, menambah, mengubah, menghapus dan mencetak data klien. Pada baris 8-19 digunakan untuk menunjukkan status dari proposal yang sedang dikerjakan oleh staf. Pada baris 20-65 digunakan untuk mendaftarkan klien baru. Baris 66-87 digunakan untuk menghapus data klien yang tersimpan. 83-19 digunakan untuk mengubah data klien yang sudah disimpan sebelumnya di database. Sedangkan baris 150-154 digunakan untuk mencetak data-data klien untuk kepentingan kantor.

### 6.2.2 Implementasi Kelola *Monitoring* Aktivitas

Pada bagian implementasi kelola *monitoring* aktivitas sesuai dengan *sequence* kelola *monitoring* aktivitas dibuat kedalam implementasi kode program pada kelas dan potongan kode program pada tabel 6.2 berikut

**Tabel 6.2 Implementasi Kelola *Monitoring* Aktivitas**

No	C_staf_monitoring_aktivitas.php
1	<code>function index(){</code>
2	<code>\$this-&gt;monitoring_aktivitas(\$this-&gt;session-</code>
3	<code>&gt;userdata('id_pengguna'));</code>
4	<code>}</code>
5	
6	<code>function monitoring_aktivitas(\$id_staf){</code>

Tabel 6.2 Implementasi Kelola *Monitoring* Aktivitas (lanjutan)

No	C_staf_monitoring_aktivitas.php
7	<code>    \$data=\$this-&gt;m_staf_monitoring_aktivitas-</code>
8	<code>&gt;get_all_klien(\$id_staf);</code>
9	<code>    \$this-&gt;load-</code>
10	<code>&gt;view('staf/aktivitas_list_klien',array('data'=&gt;\$data));</code>
11	<code>    }</code>
12	
13	<code>function aktivitas_harian(\$id_klien,\$id_staf){</code>
14	<code>    \$data_klien=\$this-&gt;m_staf_monitoring_aktivitas-</code>
15	<code>&gt;get_all_klien(\$id_staf,\$id_klien);</code>
16	<code>    \$data_monitoring=\$this-&gt;m_staf_monitoring_aktivitas-</code>
17	<code>&gt;get_monitoring_klien(\$id_klien);</code>
18	<code>    \$this-&gt;load-</code>
19	<code>&gt;view('staf/aktivitas_harian_klien',array('klien'=&gt;\$data_klien,'mo-</code>
20	<code>nitoning'=&gt;\$data_monitoring));</code>
21	<code>    }</code>
22	<code>function insert_aktivitas_harian_klien(){</code>
23	<code>    \$data = array(</code>
24	<code>        'id_klien'=&gt;\$_POST['id'],</code>
25	<code>        'tanggal_mulai'=&gt;\$_POST['tanggal']."</code>
26	<code>        ".\$_POST['jam_mulai'].":00",</code>
27	<code>        'tanggal_selesai'=&gt;\$_POST['tanggal']."</code>
28	<code>        ".\$_POST['jam_selesai'].":00",</code>
29	<code>        'aktivitas'=&gt;\$_POST['aktivitas'],</code>
30	<code>        'id_pengguna'=&gt;\$_POST['id_staff'],</code>
31	<code>        'keterangan'=&gt;\$_POST['keterangan'],</code>
32	<code>        'status_aktivitas'=&gt;\$_POST['status_aktivitas']</code>
33	<code>    );</code>
34	<code>    \$res=\$this-&gt;m_staf_monitoring_aktivitas-</code>
35	<code>&gt;insert_aktivitas('monitoring_aktivitas',\$data['id_klien'],\$data['</code>
36	<code>tanggal_mulai'],\$data['tanggal_selesai'],\$data['aktivitas'],\$data[</code>
37	<code>'id_pengguna'],\$data['keterangan'],\$data['status_aktivitas']);</code>
38	<code>    if(\$res){</code>
39	<code>        \$this-&gt;session-&gt;set_flashdata('pesan','Data berhasil</code>
40	<code>dimasukkan.');</code>
41	<code>        redirect('staf/c_staf_monitoring_aktivitas/aktivitas</code>
42	<code>_harian/'.\$_POST['id']."/".\$this-&gt;session-</code>
43	<code>&gt;userdata('id_pengguna');</code>
44	<code>    }else {</code>
45	<code>        \$this-&gt;session-&gt;set_flashdata('pesan','Data</code>
46	<code>gagal dimasukkan');</code>
47	<code>        redirect('staf/c_staf_monitoring_aktivitas/aktivitas</code>
48	<code>_harian/'.\$_POST['id']."/".\$this-&gt;session-</code>
49	<code>&gt;userdata('id_pengguna');</code>
50	<code>    }</code>
51	<code>}</code>

Tabel 6.2 Implementasi Kelola *Monitoring* Aktivitas (lanjutan)

No	C_staf_monitoring_aktivitas.php
52	
53	function delete_aktivitas_harian(\$id_aktivitas,\$id_klien){
54	\$where = array('id_aktivitas' => \$id_aktivitas);
55	\$res = \$this->m_staf_monitoring_aktivitas-
56	>delete_aktivitas('monitoring_aktivitas',\$where);
57	if(\$res){
58	\$this->session->set_flashdata('pesan','Data berhasil
59	Dihapus.');
60	redirect('staf/c_staf_monitoring_aktivitas/aktivitas
61	_harian/'.\$id_klien.'/'.\$this->session-
62	>userdata('id_pengguna'));
63	}else {
64	\$this->session->set_flashdata('pesan','Data gagal
65	Dihapus');
66	redirect('staf/c_staf_monitoring_aktivitas/aktivitas
67	_harian/'.\$id_klien.'/'.\$this->session-
68	>userdata('id_pengguna'));
69	}
70	}
71	
72	function update_aktivitas_harian(\$id_aktivitas){
73	\$data2=\$this->m_staf_monitoring_aktivitas-
74	>get_data_klien_aktivitas(\$id_aktivitas);
75	\$data = \$this->m_staf_monitoring_aktivitas-
76	>get_data_aktivitas_detail(\$id_aktivitas);
77	\$status=\$this->m_staf_monitoring_aktivitas->get_status();
78	\$this->load-
79	>view('staf/edit_aktivitas_harian_klien',array('data'=>\$data,'data
80	2'=>\$data2,'status'=>\$status));
81	}
82	
83	function proses_update_aktivitas_harian(){
84	\$data = array(
85	'tanggal_mulai'=>\$_POST['tanggal']."
86	".\$_POST['jam_mulai'].":00",
87	'tanggal_selesai'=>\$_POST['tanggal']."
88	".\$_POST['jam_selesai'].":00",
89	'aktivitas'=>\$_POST['aktivitas'],
90	'keterangan'=>\$_POST['keterangan'],
91	'status_aktivitas'=>\$_POST['status_aktivitas']
92	);
93	\$res = \$this->m_staf_monitoring_aktivitas-
94	>update_aktivitas('monitoring_aktivitas',\$data,array('id_aktivitas
95	'=>\$_POST['id_aktivitas']));

**Tabel 6.2 Implementasi Kelola *Monitoring* Aktivitas (lanjutan)**

No	C_staf_monitoring_aktivitas.php
96	if (\$res) {
97	\$this->session->set_flashdata('pesan', 'Data Berhasil
98	Diubah');
99	redirect('staf/c_staf_monitoring_aktivitas/update_aktivitas_harian/'. \$_POST['id_aktivitas']);
100	
101	}
102	else {
103	\$this->session->set_flashdata('pesan', 'Data Gagal
104	Diubah');
105	redirect('staf/c_staf_monitoring_aktivitas/update_aktivitas_harian/'. \$_POST['id_aktivitas']);
106	
107	}
108	}
109	
110	function print_aktivitas(){
111	\$print=\$this->m_staf_monitoring_aktivitas->daftar_print_aktivitas(\$_POST['bulan_aktivitas'],\$_POST['tahun_aktivitas'],\$_POST['id_staf'],\$_POST['id_klien']);
112	
113	
114	\$klien=\$this->m_staf_monitoring_aktivitas->get_all_klien(\$_POST['id_staf'],\$_POST['id_klien']);
115	
116	\$this->load->view('staf/print_aktivitas',array('print'=>\$print,'klien'=>\$klien));
117	
118	
119	}

Kode program pada Tabel 6.2 merupakan kode program yang terdapat pada kelas *c\_staf\_monitoring\_aktivitas* yang digunakan untuk mengelola kegiatan konsultasi dan terdapat fungsi-fungsi yang dibuat untuk mendukung pengelolaan data klien diantaranya untuk melihat, menambah, mengubah, menghapus dan mencetak data konsultasi. Pada baris 13-21 digunakan untuk menampilkan aktivitas harian yang dilakukan oleh klien staf tertentu. Baris 22-51 digunakan untuk menambah data aktivitas atau konsultasi harian. Baris 53-70 digunakan untuk menghapus data aktivitas yang tersimpan. Baris 72-108 digunakan untuk mengubah data aktivitas yang sudah tersimpan sebelumnya di database. Baris 110-119 digunakan untuk mencetak data aktivitas atau konsultasi klien berdasarkan bulan dan tahun sesuai dengan kebutuhan kantor.

### 6.2.3 Implementasi Menambah *Monitoring* Bulanan

Pada bagian implementasi menambah *monitoring* bulanan sesuai dengan *sequence* menambah *monitoring* bulanan dibuat kedalam implementasi kode program pada kelas dan potongan kode program pada tabel 6.3 berikut

**Tabel 6.3 Implementasi Menambah *Monitoring* Bulanan**

No	C_staf_monitoring_bulanan.php
1	function index(){

Tabel 6.3 Implementasi Menambah *Monitoring* Bulanan (lanjutan)

No	C_staf_monitoring_bulanan.php
2	<code>\$this-&gt;monitoring_bulanan(\$this-&gt;session-</code>
3	<code>&gt;userdata('id_pengguna'));</code>
4	<code>}</code>
5	
6	<code>function monitoring_bulanan(\$idstaff){</code>
7	<code>    \$data=\$this-&gt;m_staf_monitoring_bulanan-</code>
8	<code>&gt;get_all_klien(\$idstaff);</code>
9	<code>    \$this-&gt;load-</code>
10	<code>&gt;view('staf/bulanan_list_klien',array('data'=&gt;\$data));</code>
11	<code>}</code>
12	
13	<code>function bulanan_detail(\$id_staff,\$id_klien,\$id_proker=null){</code>
14	<code>    \$data_klien=\$this-&gt;m_staf_monitoring_bulanan-</code>
15	<code>&gt;get_all_klien(\$id_staff,\$id_klien);</code>
16	<code>    \$data_proker=\$this-&gt;m_staf_monitoring_bulanan-</code>
17	<code>&gt;get_where_proker('proker',array('id_tipe_proker'=&gt;1));</code>
18	<code>    if (isset(\$_POST['proker'])) { //masuk dari pilih jenis</code>
19	<code>pekerjaan di view</code>
20	<code>        \$data_monitoring=\$this-&gt;m_staf_monitoring_bulanan-</code>
21	<code>&gt;get_where_proker('monitoring_proker_bulanan',array('id_klien'=&gt;\$i</code>
22	<code>d_klien,'id_proker'=&gt;\$_POST['proker']));</code>
23	<code>        \$id_proker1=\$_POST['proker'];</code>
24	<code>    } else if (\$id_proker!=null) {</code>
25	<code>//masuk dari controller delete bulanan dan masuk dari dashboard jg</code>
26	<code>        \$data_monitoring=\$this-&gt;m_staf_monitoring_bulanan-</code>
27	<code>&gt;get_where_proker('monitoring_proker_bulanan',array('id_klien'=&gt;\$i</code>
28	<code>d_klien,'id_proker'=&gt;\$id_proker));</code>
29	<code>        \$id_proker1=\$id_proker;</code>
30	<code>    } else { //masuk awal dari list bulanan klien</code>
31	<code>        \$data_monitoring=\$this-&gt;m_staf_monitoring_bulanan-</code>
32	<code>&gt;get_where_proker('monitoring_proker_bulanan',array('id_klien'=&gt;\$i</code>
33	<code>d_klien,'id_proker'=&gt;1));</code>
34	<code>        \$id_proker1=1;</code>
35	<code>    }</code>
36	<code>    \$this-&gt;load-</code>
37	<code>&gt;view('staf/bulanan_klien',array('klien'=&gt;\$data_klien,</code>
38	<code>'proker'=&gt;\$data_proker,</code>
39	<code>'monitoring'=&gt;\$data_monitoring,'id_proker'=&gt;\$id_proker1));</code>
40	<code>}</code>
41	
42	<code>function insert_bulanan(){</code>
43	<code>    \$this-&gt;load-&gt;library('upload');</code>
44	<code>    \$config['upload_path'] = './file_upload/';</code>
45	<code>    \$config['allowed_types'] = 'gif jpg png jpeg pdf';</code>
46	<code>    \$config['file_name'] = random_string('alnum',10);</code>

Tabel 6.3 Implementasi Menambah *Monitoring* Bulanan (lanjutan)

No	C_staf_monitoring_bulanan.php
47	<code>\$this-&gt;upload-&gt;initialize(\$config);</code>
48	<code>\$replaced_number = str_replace('.', '', \$_POST['jumlah']);</code>
49	<code>\$id_proker=\$_POST['proker'];</code>
50	<code>\$id_klien=\$_POST['id_klien'];</code>
51	<code>\$id_pengguna=\$this-&gt;session-&gt;userdata('id_pengguna');</code>
52	<code>\$massa_pajak=\$_POST['tahun_massa_pajak'].'-</code>
53	<code>\$_POST['bulan_massa_pajak'].'-20';</code>
54	<code>\$tanggal_bayar=\$_POST['tanggal_bayar'];</code>
55	<code>\$jumlah=\$replaced_number;</code>
56	<code>\$ntpn=\$_POST['ntpn'];</code>
57	<code>\$status_bayar=0;</code>
58	<code>\$tanggal_lapor=\$_POST['tanggal_lapor'];</code>
59	<code>\$nomor_lapor=\$_POST['nomor_lapor'];</code>
60	<code>\$status_lapor=0;</code>
61	<code>if(\$_FILES['file_bukti_bayar']['size']!=0 ){</code>
62	<code>    if (\$this-&gt;upload-&gt;do_upload('file_bukti_bayar')){</code>
63	<code>        \$data_image = \$this-&gt;upload-</code>
64	<code>        -&gt;data();\$data['file_bukti_bayar']=\$config['u</code>
65	<code>        pload_path'].\$data_image['file_name'];</code>
66	<code>    }</code>
67	<code>    }</code>
68	<code>if(\$_FILES['file_bukti_lapor']['size']!=0 ){</code>
69	<code>    if(\$this-&gt;upload-&gt;do_upload('file_bukti_lapor')){</code>
70	<code>        \$data_image2 = \$this-&gt;upload-</code>
71	<code>        -&gt;data();\$data['file_bukti_lapor']=\$config['u</code>
72	<code>        pload_path'].\$data_image2['file_name'];</code>
73	<code>    }</code>
74	<code>    }</code>
75	<code>\$res = \$this-&gt;m_staf_monitoring_bulanan-</code>
76	<code>&gt;insert_bulanan('monitoring_proker_bulanan',\$id_proker,\$id_klien,\$</code>
77	<code>id_pengguna,\$massa_pajak,\$tanggal_bayar,\$jumlah,\$ntpn,\$status_baya</code>
78	<code>r,\$tanggal_lapor,\$nomor_lapor,\$status_lapor,\$data['file_bukti_baya</code>
79	<code>r'],\$data['file_bukti_lapor']);</code>
80	<code>if(\$this-&gt;db-&gt;affected_rows(&gt;0){</code>
81	<code>    \$this-&gt;session-&gt;set_flashdata('pesan','Data berhasil</code>
82	<code>dimasukkan.');</code>
83	<code>    redirect('staf/c_staf_monitoring_bulanan/bulanan_det</code>
84	<code>    ail/'.\$this-&gt;session-</code>
85	<code>    -&gt;userdata('id_pengguna')."/".\$_POST['id_klien']."/".</code>
86	<code>    \$_POST['id_proker']);</code>
87	<code>    }else {</code>
88	<code>        \$this-&gt;session-&gt;set_flashdata('pesan','Data gagal</code>
89	<code>dimasukkan.');</code>
90	<code>        redirect('staf/c_staf_monitoring_bulanan/bulanan_det</code>
91	<code>        ail/'.\$this-&gt;session-</code>

**Tabel 6.3 Implementasi Menambah *Monitoring* Bulanan (lanjutan)**

No	C_staf_monitoring_bulanan.php
92	>userdata('id_pengguna')." / ". \$_POST['id_klien']. " / ".
93	\$_POST['id_proker']);
94	}
95	}

Kode program pada Tabel 6.3 merupakan kode program yang terdapat pada kelas *c\_staf\_monitoring\_bulanan* yang digunakan untuk mengelola data *monitoring* perpajakan bulanan dan terdapat fungsi-fungsi yang dibuat untuk mendukung pengelolaan data klien untuk melihat dan menambah data. Pada baris 13-40 digunakan untuk menampilkan data perpajakan bulanan klien tertentu. Baris 42-95 digunakan untuk menambahkan data *monitoring* perpajakan bulanan klien.

### 6.2.4 Implementasi *Request Monitoring* Bulanan

Pada bagian implementasi *request monitoring* bulanan sesuai dengan *sequence request monitoring* bulanan dibuat kedalam implementasi kode program pada kelas dan potongan kode program pada tabel 6.4 berikut

**Tabel 6.4 Implementasi *Request Monitoring* Bulanan**

No	C_staf_monitoring_bulanan.php
1	function progress_bulanan(\$id_proker_bulanan){
2	\$data = \$this->m_staf_monitoring_bulanan-
3	>cek_terlambat_bulanan(\$id_proker_bulanan);
4	if(\$data[0]['deadline']<10 and
5	isset(\$_POST['status_bayar'])) {
6	\$res=\$this->m_staf_monitoring_bulanan-
7	>update_bulanan('monitoring_proker_bulanan',array('status_bayar'=>
8	3),array('id_proker_bulanan'=>\$id_proker_bulanan));
9	if(\$res){
10	\$this->session->set_flashdata('pesan','Data
11	berhasil Diproses.');
12	redirect('staf/c_staf_monitoring_bulanan/bulan
13	an_detail/'. \$this->session-
14	>userdata('id_pengguna')." / ". \$_POST['id_klien'
15	]." / ". \$_POST['id_proker']);
16	} else {
17	\$this->session->set_flashdata('pesan','Data
18	gagal Diproses');
19	redirect('staf/c_staf_monitoring_bulanan/bulan
20	an_detail/'. \$this->session-
21	>userdata('id_pengguna')." / ". \$_POST['id_klien'
22	]." / ". \$_POST['id_proker']);
23	}
24	} else if (isset(\$_POST['status_bayar']) and
25	\$data[0]['deadline']>=10) {
26	\$res=\$this->m_staf_monitoring_bulanan-
27	>update_bulanan('monitoring_proker_bulanan',array('status_bayar'=>



Tabel 6.4 Implementasi *Request Monitoring* Bulanan (lanjutan)

No	C_staf_monitoring_bulanan.php
28	<code>\$_POST['status_bayar']),array('id_proker_bulanan'=&gt;\$id_proker_bula</code>
29	<code>nan));</code>
30	<code>if(\$res){</code>
31	<code>    \$this-&gt;session-&gt;set_flashdata('pesan','Data</code>
32	<code>berhasil Diproses.');</code>
33	<code>    redirect('staf/c_staf_monitoring_bulanan/bulan</code>
34	<code>an_detail/'. \$this-&gt;session-</code>
35	<code>&gt;userdata('id_pengguna')."/". \$_POST['id_klien'</code>
36	<code>']."/". \$_POST['id_proker']);</code>
37	<code>    }else {</code>
38	<code>        \$this-&gt;session-&gt;set_flashdata('pesan','Data</code>
39	<code>gagal Diproses');</code>
40	<code>        redirect('staf/c_staf_monitoring_bulanan/bulan</code>
41	<code>an_detail/'. \$this-&gt;session-</code>
42	<code>&gt;userdata('id_pengguna')."/". \$_POST['id_klien'</code>
43	<code>']."/". \$_POST['id_proker']);</code>
44	<code>    }</code>
45	<code>    }else if (\$data[0]['deadline']&gt;=0 and</code>
46	<code>isset(\$_POST['status_lapor'])) {</code>
47	<code>        //lebih dari tanggal 20 maka proses terlambat</code>
48	<code>        \$res=\$this-&gt;m_staf_monitoring_bulanan-</code>
49	<code>&gt;update_bulanan('monitoring_proker_bulanan',array('status_lapor'=&gt;</code>
50	<code>\$_POST['status_lapor']),array('id_proker_bulanan'=&gt;\$id_proker_bula</code>
51	<code>nan));</code>
52	<code>        if(\$res){</code>
53	<code>            \$this-&gt;session-&gt;set_flashdata('pesan','Data</code>
54	<code>berhasil Diproses.');</code>
55	<code>            redirect('staf/c_staf_monitoring_bulanan/bulan</code>
56	<code>an_detail/'. \$this-&gt;session-</code>
57	<code>&gt;userdata('id_pengguna')."/". \$_POST['id_klien'</code>
58	<code>']."/". \$_POST['id_proker']);</code>
59	<code>        }else {</code>
60	<code>            \$this-&gt;session-&gt;set_flashdata('pesan','Data</code>
61	<code>gagal Diproses');</code>
62	<code>            redirect('staf/c_staf_monitoring_bulanan/bulan</code>
63	<code>an_detail/'. \$this-&gt;session-</code>
64	<code>&gt;userdata('id_pengguna')."/". \$_POST['id_klien'</code>
65	<code>']."/". \$_POST['id_proker']);</code>
66	<code>        }</code>
67	<code>    }else if (\$data[0]['deadline']&lt;0 and</code>
68	<code>isset(\$_POST['status_lapor'])) {</code>
69	<code>        \$res=\$this-&gt;m_staf_monitoring_bulanan-</code>
70	<code>&gt;update_bulanan('monitoring_proker_bulanan',array('status_lapor'=&gt;</code>
71	<code>3),array('id_proker_bulanan'=&gt;\$id_proker_bulanan));</code>
72	<code>        if(\$res){</code>
73	<code>            \$this-&gt;session-&gt;set_flashdata('pesan','Data</code>
74	<code>berhasil Diproses.');</code>
75	<code>            redirect('staf/c_staf_monitoring_bulanan/bulan</code>
76	<code>an_detail/'. \$this-&gt;session-</code>

Tabel 6.4 Implementasi *Request Monitoring* Bulanan (lanjutan)

No	C_staf_monitoring_bulanan.php
77	<code>&gt;userdata('id_pengguna')."/".\$_POST['id_klien'</code>
78	<code>]./".\$_POST['id_proker']);</code>
79	<code>    }else {</code>
80	<code>        \$this-&gt;session-&gt;set_flashdata('pesan', 'Data</code>
81	<code>gagal Diproses');</code>
82	<code>        redirect('staf/c_staf_monitoring_bulanan/bulan</code>
83	<code>an_detail/' . \$this-&gt;session-</code>
84	<code>&gt;userdata('id_pengguna')."/".\$_POST['id_klien'</code>
85	<code>]./".\$_POST['id_proker']);</code>
86	<code>    }</code>
87	<code>}</code>
88	<code>}</code>
89	
90	<code>function validasi_lapor(\$status){</code>
91	<code>    if(\$status==0){</code>
92	<code>        ?&gt;&lt;button        type="submit"        name="status_lapor"</code>
93	<code>value="1"        class="btn        btn-danger        btn-sm        onclick="return</code>
94	<code>confirm('Apakah Anda Yakin untuk Memproses Selesai?')"&gt;Belum</code>
95	<code>Selesai&lt;/button&gt;&lt;?php</code>
96	<code>    }else if(\$status==1){</code>
97	<code>        echo "Proses";</code>
98	<code>    }else if(\$status==2){</code>
99	<code>        echo "Selesai";</code>
100	<code>    }else if(\$status==3){</code>
101	<code>        echo "Proses Terlambat";</code>
102	<code>    }else if(\$status==4){</code>
103	<code>        echo "Terlambat";</code>
104	<code>    }</code>
105	<code>}</code>
106	
107	<code>function validasi_bayar(\$status){</code>
108	<code>    if(\$status==0){</code>
109	<code>        ?&gt;&lt;button        type="submit"        name="status_bayar"</code>
110	<code>value="1"        class="btn        btn-danger        btn-sm        onclick="return</code>
111	<code>confirm('Apakah Anda Yakin untuk Memproses Selesai?')"&gt;Belum</code>
112	<code>Selesai&lt;/button&gt;&lt;?php</code>
113	<code>    }else if(\$status==1){</code>
114	<code>        echo "Proses";</code>
115	<code>    }else if(\$status==2){</code>
116	<code>        echo "Selesai";</code>
117	<code>    }else if(\$status==3){</code>
118	<code>        echo "Proses Terlambat";</code>
119	<code>    }else if(\$status==4){</code>

**Tabel 6.4 Implementasi Request Monitoring Bulanan (lanjutan)**

No	C_staf_monitoring_bulanan.php
120	echo "Terlambat";
121	}
122	}

Kode program pada Tabel 6.4 merupakan kode program yang terdapat pada kelas *c\_staf\_monitoring\_bulanan* yang digunakan untuk melakukan proses *request* validasi oleh staf yang ditujukan kepada pimpinan mengenai *monitoring* perpajakan bulanan. Pada baris 1-88 digunakan untuk mengubah status validasi *monitoring* bulanan yang dilakukan oleh staf. Baris 90-122 digunakan untuk menampilkan status validasi yang berkaitan dengan validasi laporan dan validasi bayar *monitoring* bulanan yang dikerjakan oleh staf.

### 6.2.5 Implementasi Validasi Request Monitoring Bulanan

Pada bagian implementasi validasi *request monitoring* bulanan sesuai dengan *sequence* validasi *request monitoring* bulanan dibuat kedalam implementasi kode program pada kelas dan potongan kode program pada tabel 6.5 berikut

**Tabel 6.5 Implementasi Validasi Request Monitoring Bulanan**

No	C_monitoring_bulanan.php
1	function progress_bulanan(\$id_proker_bulanan){
2	if (isset(\$_POST['status_bayar'])) {
3	\$res=\$this->m_monitoring_bulanan-
4	>update_bulanan('monitoring_proker_bulanan',array('status_bayar'=>
5	\$_POST['status_bayar']),array('id_proker_bulanan'=>\$id_proker_bula-
6	nan));
7	if(\$res){
8	\$this->session->set_flashdata('pesan','Data
9	berhasil Diproses.');
10	redirect('kepala/c_monitoring_bulanan/bulanan_
11	detail/' .\$_POST['id_klien']."/" .\$_POST['id_pro
12	ker']);
13	}else {
14	\$this->session->set_flashdata('pesan','Data
15	gagal Diproses');
16	redirect('kepala/c_monitoring_bulanan/bulanan_
17	detail/' .\$_POST['id_klien']."/" .\$_POST['id_pro
18	ker']);
19	}
20	}else if (isset(\$_POST['status_lapor'])) {
21	\$res=\$this->m_monitoring_bulanan-
22	>update_bulanan('monitoring_proker_bulanan',array('status_lapor'=>
23	\$_POST['status_lapor']),array('id_proker_bulanan'=>\$id_proker_bula-
24	nan));
25	if(\$res){
26	\$this->session->set_flashdata('pesan','Data
27	berhasil Diproses.');
28	redirect('kepala/c_monitoring_bulanan/bulanan_

Tabel 6.5 Implementasi Validasi *Request Monitoring* Bulanan (lanjutan)

No	C_monitoring_bulanan.php
29	detail/' .\$_POST['id_klien']. '/' .\$_POST['id_pro
30	ker']);
31	}else {
32	\$this->session->set_flashdata('pesan', 'Data
33	gagal Diproses');
34	redirect('kepala/c_monitoring_bulanan/bulanan_
35	detail/' .\$_POST['id_klien']. '/' .\$_POST['id_pro
36	ker']);
37	}
38	}
39	}
40	
41	function validasi_bayar(\$status){
42	if(\$status==0){
43	echo "Belum";
44	}else if(\$status==1){ ?>
45	<p>Validasi : Selesai</p>
46	<button type="submit" name="status_bayar" value="0"
47	class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda
48	Yakin untuk Tidak Memvalidasi?')"><span class="lnr lnr-cross-
49	circle"></span></button>
50	<button type="submit" name="status_bayar" value="2"
51	class="btn btn-success btn-sm" onclick="return confirm('Apakah Anda
52	Yakin untuk Memproses Validasi?')"><span class="lnr lnr-checkmark-
53	circle"></span></button>
54	<?php
55	}else if(\$status==2){
56	echo "Selesai";
57	}else if(\$status==3){?>
58	<p>Validasi : Terlambat</p>
59	<button type="submit" name="status_bayar" value="0"
60	class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda
61	Yakin untuk Tidak Memvalidasi?')"><span class="lnr lnr-cross-
62	circle"></span></button>
63	<button type="submit" name="status_bayar" value="4"
64	class="btn btn-success btn-sm" onclick="return confirm('Apakah Anda
65	Yakin untuk Memproses Validasi?')"><span class="lnr lnr-checkmark-
66	circle"></span></button>
67	<?php
68	}else if(\$status==4){
69	echo "Terlambat";
70	}
71	}
72	
73	function validasi_lapor(\$status){

Tabel 6.5 Implementasi Validasi *Request Monitoring* Bulanan (lanjutan)

No	C_monitoring_bulanan.php
74	<code>if(\$status==0){</code>
75	<code>    echo "Belum";</code>
76	<code>}else if(\$status==1){ ?&gt;</code>
77	<code>    &lt;p&gt;Validasi : Selesai&lt;/p&gt;</code>
78	<code>    &lt;button type="submit" name="status_lapor" value="0"</code>
79	<code>class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda</code>
80	<code>Yakin untuk Tidak Memvalidasi?')"&gt;&lt;span class="lnr lnr-cross-</code>
81	<code>circle"&gt;&lt;/span&gt;&lt;/button&gt;</code>
82	<code>    &lt;button type="submit" name="status_lapor" value="2"</code>
83	<code>class="btn btn-success btn-sm" onclick="return confirm('Apakah Anda</code>
84	<code>Yakin untuk Memproses Validasi?')"&gt;&lt;span class="lnr lnr-checkmark-</code>
85	<code>circle"&gt;&lt;/span&gt;&lt;/button&gt;</code>
86	<code>    &lt;?php</code>
87	<code>    }else if(\$status==2){</code>
88	<code>        echo "Selesai";</code>
89	<code>    }else if(\$status==3){?&gt;</code>
90	<code>    &lt;p&gt;Validasi : Terlambat&lt;/p&gt;</code>
91	<code>    &lt;button type="submit" name="status_lapor" value="0"</code>
92	<code>class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda</code>
93	<code>Yakin untuk Tidak Memvalidasi?')"&gt;&lt;span class="lnr lnr-cross-</code>
94	<code>circle"&gt;&lt;/span&gt;&lt;/button&gt;</code>
95	<code>    &lt;button type="submit" name="status_lapor" value="4"</code>
96	<code>class="btn btn-success btn-sm" onclick="return confirm('Apakah Anda</code>
97	<code>Yakin untuk Memproses Validasi?')"&gt;&lt;span class="lnr lnr-checkmark-</code>
98	<code>circle"&gt;&lt;/span&gt;&lt;/button&gt;</code>
99	<code>    &lt;?php</code>
100	<code>    }else if(\$status==4){</code>
101	<code>        echo "Terlambat";</code>
102	<code>    }</code>
103	<code>}</code>

Kode program pada Tabel 6.5 merupakan kode program yang terdapat pada kelas *c\_monitoring\_bulanan* yang digunakan untuk melakukan proses validasi oleh pimpinan mengenai *monitoring* perpajakan bulanan yang dikerjakan oleh staf. Pada baris 1-39 digunakan untuk mengubah status validasi *monitoring* bulanan yang dilakukan oleh pimpinan. Baris 41-103 digunakan untuk menampilkan status validasi yang berkaitan dengan validasi lapor dan validasi bayar *monitoring* bulanan pada sisi pimpinan.

### 6.2.6 Implementasi Menambah Tanda Terima Dokumen

Pada bagian implementasi menambah tanda terima dokumen sesuai dengan *sequence* menambah tanda terima dokumen dibuat kedalam implementasi kode program pada kelas dan potongan kode program pada tabel 6.6 berikut

Tabel 6.6 Implementasi Menambah Tanda Terima Dokumen

No	C_staf_tanda_terima.php
1	function tanda_terima(\$id_staf){
2	\$data_klien=\$this->m_staf_tanda_terima-
3	>get_all_klien(\$id_staf);
4	\$this->load-
5	>view('staf/tanda_terima_list_klien',array('data'=>\$data_klien));
6	}
7	
8	function tanda_terima_detail(\$id_staf,\$id_klien){
9	\$data_klien=\$this->m_staf_tanda_terima-
10	>get_all_klien(\$id_staf,\$id_klien);
11	\$data_monitoring=\$this->m_staf_tanda_terima-
12	>get_tanda_terima('data_dokumen_klien',array('id_klien'=>\$id_klien
13	));
14	\$this->load-
15	>view('staf/tanda_terima_detail',array('klien'=>\$data_klien,'monit
16	oring'=>\$data_monitoring));
17	}
18	
19	function print_tanda_terima(){
20	\$id_staff=\$this->m_staf_tanda_terima-
21	>get_tanda_terima('klien',array('id_klien'=>\$_POST['id_klien']));
22	\$counter=0;
23	while (\$counter<=\$_POST['counter']) {
24	\$thn="tahun_{\$counter}";
25	\$ket="keterangan_{\$counter}";
26	\$doc="dokumen_{\$counter}";
27	\$data = array(
28	'id_klien'=>\$_POST['id_klien'],
29	'tanggal_terima'=> date("Y-m-d"),
30	'nama_dokumen'=>\$_POST[\$doc],
31	'tahun'=>\$_POST[\$thn],
32	'id_pengguna'=>\$id_staff[0]['id_pengguna'],
33	'keterangan'=>\$_POST[\$ket],
34	'penerima'=>\$_POST['penerima'],
35	'penyerah'=>\$_POST['penyerah']
36	);
37	\$this->m_staf_tanda_terima-
38	>insert_tanda_terima('data_dokumen_klien',\$data['id_klien'],\$data[
39	'tanggal_terima'],\$data['nama_dokumen'],\$data['tahun'],\$data['id_p
40	engguna'],\$data['keterangan'],\$data['penerima'],\$data['penyerah']
41	);
42	\$counter++;
43	};

**Tabel 6.6 Implementasi Menambah Tanda Terima Dokumen (lanjutan)**

No	C_staf_tanda_terima.php
44	<code>\$this-&gt;load-</code>
45	<code>&gt;view('staf/print_tanda_terima',array('data'=&gt;\$_POST,'klien'=&gt;\$id_</code>
46	<code>staff));</code>
47	<code>}</code>

Kode program pada Tabel 6.6 merupakan kode program yang terdapat pada kelas `c_staf_tanda_terima` yang digunakan untuk membuat dan menyimpan data tanda terima dokumen dan terdapat fungsi-fungsi yang dibuat untuk mendukung pengelolaan data klien untuk pembuatan tanda terima dokumen. Pada baris 8-17 digunakan untuk melihat data tanda terima yang pernah dibuat sebelumnya dan tersimpan di database. Baris 19-47 digunakan untuk mencetak tanda terima yang dibuat oleh staf dan menyimpan datanya ke dalam database.

### 6.2.7 Implementasi Melihat Grafik Aktivitas

Pada bagian implementasi melihat grafik aktivitas sesuai dengan *sequence* melihat grafik aktivitas dibuat kedalam implementasi kode program pada kelas dan potongan kode program pada tabel 6.7 berikut

**Tabel 6.7 Implementasi Melihat Grafik Aktivitas**

No	C_dashboard.php
1	<code>function dashboard_1(){</code>
2	<code>    // Chart</code>
3	<code>    \$dataChart=\$this-&gt;m_dashboard-</code>
4	<code>&gt;getChartAktivitas("0","2018"); //awal masuk dashboard</code>
5	<code>    \$bulan = \$this-&gt;m_dashboard-&gt;daftar_bulan();</code>
6	<code>    \$staf = \$this-&gt;m_dashboard-&gt;daftar_staf();</code>
7	<code>    \$tahun = \$this-&gt;m_dashboard-&gt;daftar_tahun();</code>
8	<code>    //END Chart</code>
9	<code>    \$data_proposal=\$this-&gt;m_dashboard-&gt;get_proposal();</code>
10	<code>    \$data_bulanan=\$this-&gt;m_dashboard-</code>
11	<code>&gt;get_monitoring_bulanan();</code>
12	<code>    \$data_tahunan=\$this-&gt;m_dashboard-</code>
13	<code>&gt;get_monitoring_tahunan();</code>
14	<code>    \$data_notifikasi_bayar_bulanan=\$this-&gt;m_dashboard-</code>
15	<code>&gt;get_monitoring_bulanan_notifikasi_bayar_bulanan();</code>
16	<code>    \$data_notifikasi_bayar_tahunan=\$this-&gt;m_dashboard-</code>
17	<code>&gt;get_monitoring_tahunan_notifikasi_bayar_tahunan();</code>
18	<code>    \$data_notifikasi_lapor_bulanan=\$this-&gt;m_dashboard-</code>
19	<code>&gt;get_monitoring_bulanan_notifikasi_lapor_bulanan();</code>
20	<code>    \$data_notifikasi_lapor_tahunan=\$this-&gt;m_dashboard-</code>
21	<code>&gt;get_monitoring_tahunan_notifikasi_lapor_tahunan();</code>
22	<code>    \$this-&gt;load-</code>
23	<code>&gt;view('kepala/dashboard',array('data_bulanan'=&gt;\$data_bulanan,'data</code>
24	<code>_tahunan'=&gt;\$data_tahunan,'data_notifikasi_bayar_bulanan'=&gt;\$data_no</code>
25	<code>tifikasi_bayar_bulanan,'data_notifikasi_bayar_tahunan'=&gt;\$data_noti</code>
26	<code>fikasi_bayar_tahunan,'data_notifikasi_lapor_bulanan'=&gt;\$data_notifi</code>

**Tabel 6.7 Implementasi Melihat Grafik Aktivitas (lanjutan)**

No	C_dashboard.php
27	kasi_lapor_bulanan,'data_notifikasi_lapor_tahunan'=>\$data_notifika
28	si_lapor_tahunan,'data_proposal'=>\$data_proposal,'staf'=>\$staf,'ta
29	hun'=>\$tahun,'dataChart'=>\$dataChart,'bulan'=>\$bulan));
30	}

Kode program pada Tabel 6.7 merupakan kode program yang terdapat pada kelas `c_dashboard` yang fungsi salah satunya digunakan untuk melihat jumlah dari konsultasi setiap staf yang disajikan berupa gambar grafik. Pada baris 3-8 digunakan untuk menampilkan data grafik.

### 6.2.8 Implementasi Data Definition Language (DDL)

Pada bagian ini merupakan implementasi tabel-tabel yang digunakan dalam mengolah basis data Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta. Bagian ini mengacu pada tahap perancangan skema basis data sebelumnya. Berikut *query* dalam membuat tabel-tabel basis data

**Tabel 6.8 Implementasi DDL Tabel Bulan**

No	Tabel Bulan
	<pre>CREATE TABLE `bulan` (   `id_bulan` int(11) NOT NULL PRIMARY KEY,   `nama_bulan` varchar(30) NOT NULL )</pre>

Tabel 6.8 merupakan DDL yang digunakan untuk menyimpan data bulan yang dipakai dalam pembuatan grafik informasi.

**Tabel 6.9 Implementasi DDL Tipe Proker**

No	Tabel Bulan
	<pre>CREATE TABLE `tipe_proker` (   `id_tipe_proker` int(11) NOT NULL PRIMATY KEY,   `tipe` varchar(20) NOT NULL )</pre>

Tabel 6.9 merupakan DDL yang digunakan untuk menyimpan data tipe-tipe proker.

**Tabel 6.10 Implementasi DDL Proker**

No	Tabel Proker
	<pre>CREATE TABLE `proker` (   `id_proker` int(11) NOT NULL PRIMARY KEY,   `id_tipe_proker` int(11) NOT NULL,   `nama` varchar(30) NOT NULL,   FOREIGN KEY (id_tipe_proker) REFERENCES tipe_proker   (id_tipe_proker) )</pre>

Tabel 6.10 merupakan DDL yang digunakan untuk menyimpan dan mengelompokkan data tipe-tipe proker yang ada pada perpajakan bulanan dan tahunan.

**Tabel 6.11 Implementasi DDL Status**

No	Tabel Status
	<pre>CREATE TABLE `status` (   `id_status` int(11) NOT NULL PRIMATY KEY,   `nama` varchar(20) NOT NULL )</pre>

Tabel 6.11 merupakan DDL yang digunakan untuk menyimpan tipe status yang terdapat pada sistem.

**Tabel 6.12 Implementasi DDL Tabel Pengguna**

No	Tabel Pengguna
	<pre>CREATE TABLE `pengguna` (   `id_pengguna` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,   `no_pengguna` varchar(30) NOT NULL,   `username` varchar(30) NOT NULL,   `password` varchar(30) NOT NULL,   `role` int(11) NOT NULL )</pre>

Tabel 6.12 merupakan DDL yang digunakan untuk menyimpan data pengguna yang bisa mengakses sistem.

**Tabel 6.13 Implementasi DDL Tabel Klien**

No	Tabel Klien
	<pre>CREATE TABLE `klien` (   `id_klien` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,   `id_pengguna` int(11) NOT NULL,   `efin` varchar(30) NOT NULL,   `nama` varchar(30) NOT NULL,   `npwp` varchar(30) NOT NULL,   `jenis_usaha` varchar(60) NOT NULL,   `klu_usaha` varchar(30) NOT NULL,   `alamat` text NOT NULL,   `email` varchar(30) NOT NULL,   `telepon` varchar(15) NOT NULL,   `tanggal_lahir` date NOT NULL,   `catatan` text NOT NULL,   `file_npwp` varchar(30) NOT NULL,   `file_pkp` varchar(30) NOT NULL, )</pre>



**Tabel 6.13 Implementasi DDL Tabel Klien (lanjutan)**

No	Tabel Klien
	FOREIGN KEY (id_pengguna) REFERENCES pengguna (id_pengguna) )

Tabel 6.13 merupakan DDL yang digunakan untuk menyimpan data klien yang ada pada kantor konsultan pajak.

**Tabel 6.14 Implementasi DDL Tabel *Monitoring* Aktivitas**

No	Tabel <i>Monitoring</i> Aktivitas
	<pre>CREATE TABLE `monitoring_aktivitas` (   `id_aktivitas` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,   `id_klien` int(11) NOT NULL,   `id_pengguna` int(11) NOT NULL,   `tanggal_mulai` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,   `tanggal_selesai` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,   `aktivitas` text NOT NULL,   `keterangan` text NOT NULL,   `status_aktivitas` int(11) NOT NULL,   FOREIGN KEY (id_klien) REFERENCES klien (id_klien),   FOREIGN KEY (id_pengguna) REFERENCES pengguna (id_pengguna),   FOREIGN KEY (status_aktivitas) REFERENCES status (id_status) )</pre>

Tabel 6.14 merupakan DDL yang digunakan untuk menyimpan data *monitoring* aktivitas yang berupa data konsultasi antara klien dan staf.

**Tabel 6.15 Implementasi DDL Tabel *Monitoring* Proker Bulanan**

No	Tabel <i>Monitoring</i> Proker <b>Bulanan</b>
	<pre>CREATE TABLE `monitoring_proker_bulanan` (   `id_proker_bulanan` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,   `id_proker` int(11) NOT NULL,   `id_klien` int(11) NOT NULL,   `id_pengguna` int(11) NOT NULL,   `massa_pajak` date NOT NULL,   `tanggal_bayar` date NOT NULL,   `jumlah` int(11) NOT NULL,   `ntpn` varchar(30) NOT NULL,   `status_bayar` int(11) NOT NULL,   `tanggal_lapor` date NOT NULL,   `nomor_lapor` varchar(30) NOT NULL,   `status_lapor` int(11) NOT NULL,   `file_bukti_bayar` varchar(30) NOT NULL,   `file_bukti_lapor` varchar(30) NOT NULL, )</pre>



**Tabel 6 15 Implementasi DDL Tabel *Monitoring Proker Bulanan* (lanjutan)**

No	Tabel <i>Monitoring Proker Bulanan</i>
	<pre>FOREIGN KEY (id_klien) REFERENCES klien (id_klien), FOREIGN KEY (id_pengguna) REFERENCES pengguna (id_pengguna), FOREIGN KEY (id_proker) REFERENCES proker (id_proker), FOREIGN KEY (status_bayar) REFERENCES status (id_status), FOREIGN KEY (status_lapor) REFERENCES status (id_status) )</pre>

Tabel 6.15 merupakan DDL yang digunakan untuk menyimpan data *monitoring* perpajakan bulanan klien

**Tabel 6.16 Implementasi DDL Tabel *Monitoring Proker Tahunan***

No	Tabel <i>Monitoring Proker Tahunan</i>
	<pre>CREATE TABLE `monitoring_proker_tahunan` (   `id_proker_tahunan` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,   `id_proker` int(11) NOT NULL,   `id_pengguna` int(11) NOT NULL,   `id_klien` int(11) NOT NULL,   `tahun` date NOT NULL,   `tanggal_posting_jurnal` date NOT NULL,   `status_posting_jurnal` int(11) NOT NULL,   `file_posting_jurnal` varchar(30) NOT NULL,   `tanggal_bayar` date NOT NULL,   `jumlah_bayar` int(11) NOT NULL,   `ntpn` varchar(30) NOT NULL,   `status_bayar` int(11) NOT NULL,   `tanggal_lapor` date NOT NULL,   `nomor_lapor` varchar(30) NOT NULL,   `status_lapor` int(11) NOT NULL,   `file_bukti_bayar` varchar(30) NOT NULL,   `file_bukti_lapor` varchar(30) NOT NULL,   FOREIGN KEY (id_klien) REFERENCES klien (id_klien),   FOREIGN KEY (id_pengguna) REFERENCES pengguna (id_pengguna),   FOREIGN KEY (id_proker) REFERENCES proker (id_proker),   FOREIGN KEY (status_bayar) REFERENCES status (id_status),   FOREIGN KEY (status_lapor) REFERENCES status (id_status),   FOREIGN KEY (status_posting_jurnal) REFERENCES status (id_status) )</pre>

Tabel 6.16 merupakan DDL yang digunakan untuk menyimpan data *monitoring* perpajakan tahunan klien



**Tabel 6.17 Implementasi DDL Tabel Proposal**

No	Tabel Proposal
	<pre>CREATE TABLE `proposal` (   `id_proposal` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,   `id_klien` int(11) NOT NULL,   `nomor_proposal` varchar(60) NOT NULL,   `periode_mulai` date NOT NULL,   `periode_selesai` date NOT NULL,   `biaya` int(11) NOT NULL,   `deskripsi` text NOT NULL,   `penerima_kontrak_kerja` varchar(60) NOT NULL,   `status_proposal` int(11) NOT NULL,   FOREIGN KEY (id_klien) REFERENCES klien (id_klien),   FOREIGN KEY (status_proposal) REFERENCES status (id_proposal) )</pre>

Tabel 6.17 merupakan DDL yang digunakan untuk menyimpan data proposal tiap klien.

**Tabel 6.18 Implementasi DDL Tabel Data Dokumen Klien**

No	Tabel Data Dokumen Klien
	<pre>CREATE TABLE `data_dokumen_klien` (   `id_dokumen` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,   `id_klien` int(11) NOT NULL,   `id_pengguna` int(11) NOT NULL,   `tanggal_terima` date NOT NULL,   `nama_dokumen` varchar(100) NOT NULL,   `tahun` year(4) NOT NULL,   `keterangan` text NOT NULL,   `penerima` varchar(30) NOT NULL,   `penyerah` varchar(30) NOT NULL,   FOREIGN KEY (id_klien) REFERENCES klien (id_klien),   FOREIGN KEY (id_pengguna) REFERENCES pengguna (id_pengguna) )</pre>

Tabel 6.18 merupakan DDL yang digunakan untuk menyimpan data setiap tanda terima yang dibuat.

## 6.3 Implementasi Antarmuka

Pada bagian ini terdapat hasil implementasi antarmuka Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta.

### 6.3.1 Implementasi Antarmuka Login

**Gambar 6.1 Implementasi Antarmuka Login**

Pada Gambar 6.1 merupakan tampilan halaman login pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta, terdiri dari username, password dan tombol login.

### 6.3.2 Implementasi Antarmuka Dashboard Pimpinan



**Gambar 6.2 Implementasi Antarmuka Dashboard Pimpinan**

Pada Gambar 6.2 merupakan tampilan halaman menu dashboard pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta yang bisa melihat grafik rekapan jumlah kegiatan *monitoring* konsultasi dan *monitoring* perpajakan, serta terdapat pemberitahuan notifikasi untuk pengingat batas pembayaran dan pelaporan perpajakan dan juga terdapat notifikasi *request* validasi dari klien.

### 6.3.3 Implementasi Antarmuka Dashboard Staf



Gambar 6.3 Implementasi Antarmuka Dashboard Staf

Pada Gambar 6.3 merupakan tampilan halaman menu dashboard pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta yang bisa melihat notifikasi batas untuk pembayaran dan pelaporan perpajakan.

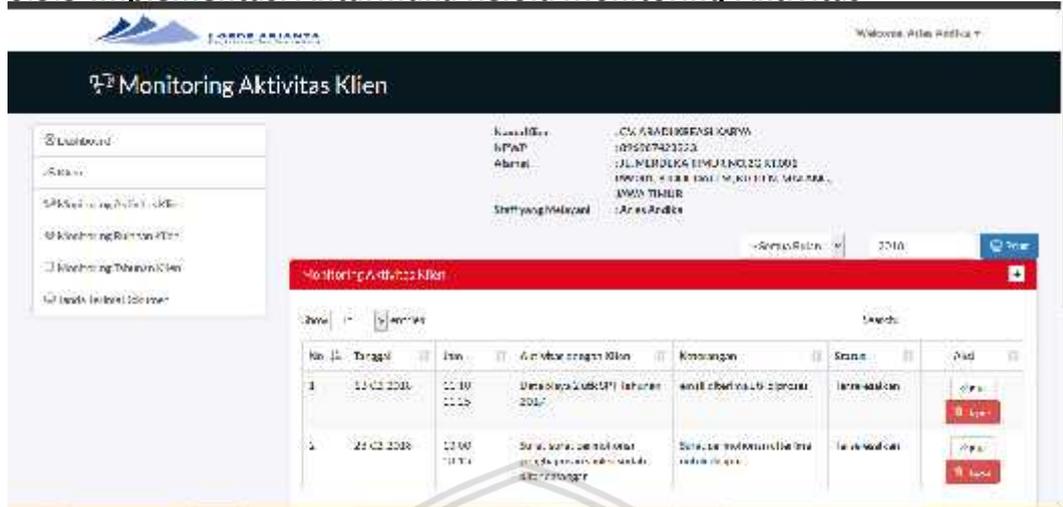
### 6.3.4 Implementasi Antarmuka Kelola Klien



Gambar 6.4 Implementasi Antarmuka Kelola Klien

Pada Gambar 6.4 merupakan tampilan halaman menu klien pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta yang bisa mengelola data klien yang terdiri dari menambah, mengubah, menghapus, dan mencetak data klien.

### 6.3.5 Implementasi Antarmuka Kelola *Monitoring* Aktivitas



Gambar 6.5 Implementasi Antarmuka Kelola *Monitoring* Aktivitas

Pada Gambar 6.5 merupakan tampilan halaman menu *monitoring* aktivitas klien pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta yang bisa mengelola data konsultasi harian yang terdiri dari menambah, mengubah, menghapus, dan mencetak data klien.

### 6.3.6 Implementasi Antarmuka *Monitoring* Bulanan



Gambar 6.6 Implementasi Antarmuka *Monitoring* Bulanan

Pada Gambar 6.6 merupakan tampilan halaman menu *monitoring* bulanan klien pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta yang bisa mengelola data *monitoring* perpajakan bulanan klien yang dilakukan oleh staf dan dalam perpajakan bulanan terdapat beberapa jenis SPT bulanan. Tiap jenis pekerjaan perpajakan bulanan terdiri dari menambah, mengubah, menghapus, mencetak dan melakukan proses *request* validasi kepada pimpinan.

### 6.3.7 Implementasi Antarmuka Tambah Tanda Terima Dokumen



**Gambar 6.7 Implementasi Antarmuka Tambah Tanda Terima Dokumen**

Pada Gambar 6.7 merupakan tampilan halaman menu tanda terima dokumen pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta yang bisa mencetak dan menyimpan data tanda terima dokumen.

### 6.3.8 Implementasi Antarmuka Kelola Tanda Terima Dokumen



**Gambar 6.8 Implementasi Antarmuka Kelola Tanda Terima Dokumen**

Pada Gambar 6.8 merupakan tampilan halaman menu tanda terima dokumen pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta yang bisa mencetak riwayat dokumen yang disimpan untuk keperluan kantor dan juga bisa menghapus data tanda terima.

## BAB 7 PENGUJIAN DAN ANALISIS HASIL

### 7.1 Pengujian

Pada bab ini menjelaskan tentang pengujian yang akan dilakukan pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta. Pengujian ini dilakukan pengujian pengujian *black-box testing* yang berupa *validation testing*, *user acceptance testing* dan pengujian waktu. Tujuan dari bab pengujian adalah untuk mengetahui bagaimana implementasi sistem apakah terdapat cacat atau error yang terjadi. Pengujian juga bertujuan untuk mengetahui apakah sistem yang telah dikembangkan sudah sesuai dan mampu menyelesaikan permasalahan yang ada. Selain itu pengujian bertujuan untuk mengetahui penerimaan sistem oleh pengguna akhir.

#### 7.1.1 Black-Box Testing

*Black-Box Testing* termasuk pengujian fungsional dimana penguji hanya dikonsentrasikan kepada fungsionalitas sistem dan tidak mencakup pengimplementasian sistem. Pengujian *black-box* sangat penting dilakukan untuk melihat langkah awal dari jalannya fungsionalitas sistem berupa inputan pengguna, aksi dari pengguna ataupun hal lainnya. Sehingga Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta harus dilakukan pengujian *black-box* berupa *validation testing*.

##### 7.1.1.1 Validation Testing

Penggunaan *validation testing* untuk mengetahui apakah Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta yang dibangun telah sesuai dengan kebutuhan kantor. Kode *validation testing* diwakilkan dengan kode VT\_KKP\_1, VT\_KKP\_2 dan VT\_KKP\_3 secara berurutan. VT merupakan singkatan dari *Validation Testing*, KKP singkatan dari Kantor Konsultan Pajak. Terdapat beberapa kolom yang akan digunakan dalam *validation testing* diantaranya:

- a. Kode Pengujian
- b. Kode Use Case
- c. Fungsional Sistem
- d. Kasus Uji
- e. Yang diharapkan
- f. Hasil yang didapat
- g. Status

Penjelasan secara lengkap mengenai *validation testing* dijelaskan pada Tabel 7.1 sebagai berikut :

**Tabel 7.1 Validation Testing**

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
VT_KKP_1	UC-KKP-1	Login	Pengguna melakukan login dengan mengisi <i>username</i> dan <i>password</i>	Halaman menu dashboard ditampilkan	Halaman menu dashboard ditampilkan	Valid
			Pengguna melakukan login dengan mengisi <i>username</i> dan <i>password</i> yang salah	Halaman login ditampilkan dengan notifikasi 'Masukkan Username dan Password dengan Benar !'	Halaman login ditampilkan dengan notifikasi 'Masukkan Username dan Password dengan Benar !'	Valid
VT_KKP_2	UC-KKP-2	Mengelola data klien	Pengguna memilih menu klien	Halaman menu klien ditampilkan	Halaman menu klien ditampilkan	Valid
			Pengguna memilih tombol tambah klien. Kemudian mengisi form-form yang disediakan terkait data klien	Halaman tambah klien ditampilkan dan data klien ditambahkan sesuai dengan masukan pengguna. Kemudian halaman klien yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	Halaman tambah klien ditampilkan dan data klien ditambahkan sesuai dengan masukan pengguna. Kemudian halaman klien yang diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	Valid
			Pengguna memilih tombol edit. Kemudian mengubah data dalam form-form yang	Halaman form edit klien ditampilkan dan data klien diubah sesuai dengan masukan pengguna. Kemudian halaman klien	Halaman form edit klien ditampilkan dan data klien diubah sesuai dengan masukan pengguna. Kemudian halaman klien	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
			disediakan terkait data klien	yang datanya diubah ditampilkan dengan notifikasi 'Data Berhasil Diubah'	yang datanya diubah ditampilkan dengan notifikasi 'Data Berhasil Diubah'	
			Pengguna memilih tombol hapus pada salah satu klien	Data klien dihapus sesuai dengan pilihan pengguna dan halaman klien ditampilkan dengan notifikasi 'Data berhasil Dihapus'	Data klien dihapus sesuai dengan pilihan pengguna dan halaman klien ditampilkan dengan notifikasi 'Data berhasil Dihapus'	Valid
			Pengguna memilih tombol cetak data klien	Halaman cetak data klien ditampilkan sesuai dengan data yang dimasukkan pengguna	Halaman cetak data klien ditampilkan sesuai dengan data yang dimasukkan pengguna	Valid
			Form tambah klien atau update klien kosong	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Valid
			File npwp atau file pkp yang diupload selain tipe gif, jpg, png, jpeg	Data npwp atau pkp tidak berhasil tersimpan	Data npwp atau pkp tidak berhasil tersimpan	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
VT_KKP_3	UC-KKP-3	Mengelola data staf	Pimpinan memilih tombol tambah staf. Kemudian mengisikan form-form yang disediakan terkait data staf	Halaman tambah staf ditampilkan dan data staf ditambahkan sesuai dengan masukan pimpinan. Kemudian halaman staf yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	Halaman tambah staf ditampilkan dan data staf ditambahkan sesuai dengan masukan pimpinan. Kemudian halaman staf yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	Valid
			Pimpinan memilih tombol edit. Kemudian mengubah data dalam form-form yang disediakan terkait data klien	Halaman form edit staf ditampilkan dan data staf diubah sesuai dengan masukan pimpinan. Kemudian menampilkan notifikasi 'Data Berhasil Diubah'	Halaman form edit staf ditampilkan dan data staf diubah sesuai dengan masukan pimpinan. Kemudian menampilkan notifikasi 'Data Berhasil Diubah'	Valid
			Pimpinan memilih tombol hapus pada salah satu staf	Data staf dihapus sesuai dengan pilihan pimpinan dan halaman staf ditampilkan dengan notifikasi 'Data berhasil Dihapus'	Data staf dihapus sesuai dengan pilihan pimpinan dan halaman staf ditampilkan dengan notifikasi 'Data berhasil Dihapus'	Valid
			Form tambah staf atau update staf kosong	Sistem akan menampilkan pesan bahwa terdapat	Sistem akan menampilkan pesan bahwa terdapat	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
				beberapa data yang masih belum terisi	beberapa data yang masih belum terisi	
VT_KKP_4	UC-KKP-4	Mengelola <i>monitoring</i> aktivitas	Pengguna memilih menu <i>monitoring</i> aktivitas klien, kemudian memilih salah satu klien untuk dikelola	Halaman <i>monitoring</i> aktivitas ditampilkan dengan berisi daftar klien. Kemudian menampilkan rincian data klien yang dipilih oleh pengguna	Halaman <i>monitoring</i> aktivitas ditampilkan dengan berisi daftar klien. Kemudian menampilkan rincian data klien yang dipilih oleh pengguna	Valid
			Pengguna memilih tombol tambah aktivitas. Kemudian mengisikan form-form yang disediakan terkait data klien	Halaman tambah aktivitas ditampilkan dan data aktivitas ditambahkan sesuai dengan masukan pengguna. Kemudian halaman aktivitas klien yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	Halaman tambah aktivitas ditampilkan dan data aktivitas ditambahkan sesuai dengan masukan pengguna. Kemudian halaman aktivitas klien yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	Valid
			Pengguna memilih tombol edit. Kemudian mengubah data dalam form-form yang disediakan terkait data aktivitas klien	Halaman form edit aktivitas klien ditampilkan dan data klien diubah sesuai dengan masukan pengguna. Kemudian menampilkan	Halaman form edit aktivitas klien ditampilkan dan data klien diubah sesuai dengan masukan pengguna. Kemudian menampilkan	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
				notifikasi 'Data Berhasil Diubah'	notifikasi 'Data Berhasil Diubah'	
			Pengguna memilih tombol hapus pada salah satu aktivitas klien	Data aktivitas klien dihapus sesuai dengan pilihan pengguna dan halaman aktivitas klien ditampilkan dengan notifikasi 'Data berhasil Dihapus'	Data aktivitas klien dihapus sesuai dengan pilihan pengguna dan halaman aktivitas klien ditampilkan dengan notifikasi 'Data berhasil Dihapus'	Valid
			Pengguna mengisi bulan dan tahun kemudian memilih tombol cetak data aktivitas klien	Halaman cetak data aktivitas klien ditampilkan sesuai dengan data yang dimasukkan pengguna	Halaman cetak data aktivitas klien ditampilkan sesuai dengan data yang dimasukkan pengguna	Valid
			Form tambah aktivitas atau update aktivitas kosong	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Valid
VT_KKP_5	UC-KKP-5	Mengelola <i>monitoring</i> bulanan	Pengguna memilih menu <i>monitoring</i> bulanan klien, kemudian memilih salah satu klien untuk dikelola	Halaman <i>monitoring</i> bulanan ditampilkan dengan berisi daftar klien. Kemudian menampilkan rincian data klien yang dipilih oleh pengguna	Halaman <i>monitoring</i> bulanan ditampilkan dengan berisi daftar klien. Kemudian menampilkan rincian data klien yang dipilih oleh pengguna	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
			Pengguna memilih tombol tambah <i>monitoring</i> bulanan. Kemudian mengisikan form-form yang disediakan terkait data klien	Halaman tambah <i>monitoring</i> bulanan klien ditampilkan dan data ditambahkan sesuai dengan masukan pengguna. Kemudian halaman <i>monitoring</i> bulanan klien yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	Halaman tambah <i>monitoring</i> bulanan klien ditampilkan dan data ditambahkan sesuai dengan masukan pengguna. Kemudian halaman <i>monitoring</i> bulanan klien yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	Valid
			Pengguna memilih tombol edit. Kemudian mengubah data dalam form-form yang disediakan terkait data <i>monitoring</i> bulanan klien	Halaman form edit <i>monitoring</i> bulanan klien ditampilkan dan data diubah sesuai dengan masukan pengguna. Kemudian menampilkan notifikasi 'Data Berhasil Diubah'	Halaman form edit <i>monitoring</i> bulanan klien ditampilkan dan data diubah sesuai dengan masukan pengguna. Kemudian menampilkan notifikasi 'Data Berhasil Diubah'	Valid
			Pengguna memilih tombol hapus pada salah satu <i>monitoring</i> bulanan klien	Data <i>monitoring</i> bulanan klien dihapus sesuai dengan pilihan pengguna dan halaman <i>monitoring</i> bulanan klien ditampilkan dengan	Data <i>monitoring</i> bulanan klien dihapus sesuai dengan pilihan pengguna dan halaman <i>monitoring</i> bulanan klien ditampilkan dengan	Valid

**Tabel 7.1 Validation Testing (lanjutan)**

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
				notifikasi 'Data berhasil Dihapus'	notifikasi 'Data berhasil Dihapus'	
			Pengguna mengisi bulan dan tahun kemudian memilih tombol cetak data <i>monitoring</i> bulanan klien	Halaman cetak data <i>monitoring</i> bulanan klien ditampilkan sesuai dengan data yang dimasukkan pengguna	Halaman cetak data <i>monitoring</i> bulanan klien ditampilkan sesuai dengan data yang dimasukkan pengguna	Valid
			Form massa pajak pada tambah atau update <i>monitoring</i> bulanan kosong	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Valid
			Jumlah nominal pajak bulanan berupa huruf	Isi form jumlah tidak akan tampil dan tidak tersimpan ketika mengetikkan huruf	Isi form jumlah tidak akan tampil dan tidak tersimpan ketika mengetikkan huruf	Valid
			Tahun massa pajak berupa huruf	Menampilkan peringatan supaya memasukkan nilai berupa angka	Menampilkan peringatan supaya memasukkan nilai berupa angka	Valid
VT_KKP_6	UC-KKP-6	<i>Request</i> validasi <i>monitoring</i> bulanan klien	Staf memilih tombol belum selesai untuk memproses <i>request</i> validasi	Status data <i>request</i> validasi dirubah menjadi 'proses'	Status data <i>request</i> validasi dirubah menjadi 'proses'	Valid

**Tabel 7.1 Validation Testing (lanjutan)**

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
VT_KKP_7	UC-KKP-7	Validasi <i>monitoring</i> bulanan klien	Pimpinan memilih tombol setuju terhadap <i>request</i> validasi	Status data validasi dirubah menjadi 'selesai'	Status data validasi dirubah menjadi 'selesai'	Valid
VT_KKP_8	UC-KKP-8	Menambah tanda terima dokumen	Staf memilih tombol tambah tanda terima dokumen. Kemudian mengisikan form-form yang disediakan terkait data tanda terima dokumen	Halaman tambah tanda terima dokumen ditampilkan dan data ditambahkan sesuai dengan masukan staf. Kemudian halaman cetak tanda terima sesuai dengan masukan staf ditampilkan dan siap dicetak	Halaman tambah tanda terima dokumen ditampilkan dan data ditambahkan sesuai dengan masukan staf. Kemudian halaman cetak tanda terima sesuai dengan masukan staf ditampilkan dan siap dicetak	Valid
			Form tambah tanda terima dokumen kosong	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Valid
			Mengisikan form tahun berupa huruf	Menampilkan peringatan supaya memasukkan nilai berupa angka	Menampilkan peringatan supaya memasukkan nilai berupa angka	Valid
VT_KKP_9	UC-KKP-9	Mengelola tanda	Pengguna memilih tombol hapus pada salah	Data tanda terima dokumen dihapus sesuai dengan pilihan pengguna dan halaman tanda terima	Data tanda terima dokumen dihapus sesuai dengan pilihan pengguna dan halaman tanda terima	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
		terima dokumen	satu data tanda terima dokumen	dokumen ditampilkan dengan notifikasi 'Data berhasil Dihapus'	dokumen ditampilkan dengan notifikasi 'Data berhasil Dihapus'	
			Pengguna mengisi bulan dan tahun kemudian memilih tombol cetak data riwayat tanda terima dokumen	Halaman cetak riwayat tanda terima dokumen ditampilkan sesuai dengan data yang dimasukkan pengguna	Halaman cetak riwayat tanda terima dokumen ditampilkan sesuai dengan data yang dimasukkan pengguna	Valid
VT_KKP_10	UC-KKP-10	Mengelola <i>monitoring</i> tahunan klien	Pengguna memilih menu <i>monitoring</i> tahunan klien, kemudian memilih salah satu klien untuk dikelola	Halaman <i>monitoring</i> tahunan ditampilkan dengan berisi daftar klien. Kemudian menampilkan rincian data <i>monitoring</i> tahunan klien yang dipilih oleh pengguna	Halaman <i>monitoring</i> tahunan ditampilkan dengan berisi daftar klien. Kemudian menampilkan rincian data <i>monitoring</i> tahunan klien yang dipilih oleh pengguna	Valid
			Pengguna memilih tombol tambah <i>monitoring</i> tahunan. Kemudian mengisikan form-form yang disediakan terkait data klien	Halaman tambah <i>monitoring</i> tahunan klien ditampilkan dan data ditambahkan sesuai dengan masukan pengguna. Kemudian halaman <i>monitoring</i> tahunan klien yang datanya diperbaharui ditampilkan dengan	Halaman tambah <i>monitoring</i> tahunan klien ditampilkan dan data ditambahkan sesuai dengan masukan pengguna. Kemudian halaman <i>monitoring</i> tahunan klien yang datanya diperbaharui ditampilkan dengan	Valid

**Tabel 7.1 Validation Testing (lanjutan)**

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
				notifikasi 'Data berhasil dimasukkan'	notifikasi 'Data berhasil dimasukkan'	
			Pengguna memilih tombol edit. Kemudian mengubah data dalam form-form yang disediakan terkait data <i>monitoring</i> tahunan klien	Halaman form edit <i>monitoring</i> tahunan klien ditampilkan dan data diubah sesuai dengan masukan pengguna. Kemudian menampilkan notifikasi 'Data Berhasil Diubah'	Halaman form edit <i>monitoring</i> tahunan klien ditampilkan dan data diubah sesuai dengan masukan pengguna. Kemudian menampilkan notifikasi 'Data Berhasil Diubah'	Valid
			Pengguna memilih tombol hapus pada salah satu <i>monitoring</i> tahunan klien	Data <i>monitoring</i> tahunan klien dihapus sesuai dengan pilihan pengguna dan halaman <i>monitoring</i> tahunan klien ditampilkan dengan notifikasi 'Data berhasil Dihapus'	Data <i>monitoring</i> tahunan klien dihapus sesuai dengan pilihan pengguna dan halaman <i>monitoring</i> tahunan klien ditampilkan dengan notifikasi 'Data berhasil Dihapus'	Valid
			Form tahun pada tambah <i>monitoring</i> tahunan kosong	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Sistem akan menampilkan pesan bahwa terdapat beberapa data yang masih belum terisi	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
			Mengisikan form tahun berupa huruf	Menampilkan peringatan supaya memasukkan nilai berupa angka	Menampilkan peringatan supaya memasukkan nilai berupa angka	Valid
			Jumlah nominal pajak tahunan berupa huruf	Isi form jumlah tidak akan tampil dan tidak tersimpan ketika mengetikkan huruf	Isi form jumlah tidak akan tampil dan tidak tersimpan ketika mengetikkan huruf	Valid
VT_KKP_11	UC-KKP-11	<i>Request</i> validasi <i>monitoring</i> tahunan klien	Staf memilih tombol belum selesai untuk memproses <i>request</i> validasi tahunan	Status data <i>request</i> validasi dirubah menjadi 'proses'	Status data <i>request</i> validasi dirubah menjadi 'proses'	Valid
VT_KKP_12	UC-KKP-12	Validasi <i>monitoring</i> tahunan klien	Pimpinan memilih tombol setuju terhadap <i>request</i> validasi tahunan	Status data validasi dirubah menjadi 'selesai'	Status data validasi dirubah menjadi 'selesai'	Valid
VT_KKP_13	UC-KKP-13	Mengelola proposal	Pengguna memilih tombol proposal di halaman klien	Halaman proposal ditampilkan dengan rincian data proposal klien yang dipilih	Halaman proposal ditampilkan dengan rincian data proposal klien yang dipilih	Valid
			Pengguna memilih tombol tambah proposal. Kemudian	Halaman tambah proposal ditampilkan dan data ditambahkan sesuai dengan	Halaman tambah proposal ditampilkan dan data ditambahkan sesuai dengan	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
			mengisikan form-form yang disediakan terkait data proposal	masukan pengguna. Kemudian halaman proposal yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	masukan pengguna. Kemudian halaman proposal yang datanya diperbaharui ditampilkan dengan notifikasi 'Data berhasil dimasukkan'	
			Pengguna memilih tombol edit. Kemudian mengubah data dalam form-form yang disediakan terkait data proposal	Halaman form edit proposal ditampilkan dan data diubah sesuai dengan masukan pengguna. Kemudian menampilkan notifikasi 'Data Berhasil Diubah'	Halaman form edit proposal ditampilkan dan data diubah sesuai dengan masukan pengguna. Kemudian menampilkan notifikasi 'Data Berhasil Diubah'	Valid
			Pengguna memilih tombol cetak data proposal	Halaman cetak data proposal ditampilkan sesuai dengan data yang dimasukkan pengguna	Halaman cetak data proposal ditampilkan sesuai dengan data yang dimasukkan pengguna	Valid
			Jumlah nominal biaya berupa huruf	Isi form jumlah tidak akan tampil dan tidak tersimpan ketika mengetikkan huruf	Isi form jumlah tidak akan tampil dan tidak tersimpan ketika mengetikkan huruf	Valid
VT_KKP_14	UC-KKP-14	<i>Request</i> persetujuan proposal	Staf memilih tombol belum disetujui untuk	Status data <i>request</i> persetujuan proposal	Status data <i>request</i> persetujuan proposal	Valid

Tabel 7.1 Validation Testing (lanjutan)

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
			memproses <i>request</i> persetujuan proposal	dirubah menjadi 'proses persetujuan'	dirubah menjadi 'proses persetujuan'	
VT_KKP_15	UC-KKP-15	Persetujuan proposal	Pimpinan memilih tombol setuju terhadap <i>request</i> persetujuan proposal	Status data persetujuan proposal dirubah menjadi 'disetujui'	Status data persetujuan proposal dirubah menjadi 'disetujui'	Valid
VT_KKP_16	UC-KKP-17	Melihat grafik aktivitas	Pimpinan memilih menu dashboard	Halaman dashboard dan informasi grafik aktivitas ditampilkan	Halaman dashboard dan informasi grafik aktivitas ditampilkan	Valid
VT_KKP_17	UC-KKP-18	Melihat grafik <i>monitoring</i> bulanan	Pimpinan memilih tombol <i>monitoring</i> perpajakan pada halaman dashboard	Halaman <i>monitoring</i> perpajakan dan informasi grafik <i>monitoring</i> bulanan ditampilkan	Halaman <i>monitoring</i> perpajakan dan informasi grafik <i>monitoring</i> bulanan ditampilkan	Valid
VT_KKP_18	UC-KKP-19	Melihat grafik <i>monitoring</i> tahunan	Pimpinan memilih tombol <i>monitoring</i> perpajakan pada halaman dashboard	Halaman <i>monitoring</i> perpajakan dan informasi grafik <i>monitoring</i> tahunan ditampilkan	Halaman <i>monitoring</i> perpajakan dan informasi grafik <i>monitoring</i> tahunan ditampilkan	Valid
VT_KKP_19	UC-KKP-16	<i>Logout</i>	Pengguna memilih tombol <i>logout</i>	Halaman login ditampilkan	Halaman login ditampilkan	Valid

### 7.1.2 User Acceptance Testing

Pada bagian ini menjelaskan *User Acceptance Testing* untuk memeriksa sistem informasi yang sudah dibuat dalam kebutuhan bisnis yang sudah teridentifikasi pada tahap analisis kebutuhan sehingga bisa memperoleh kepercayaan pengguna bahwa sistem informasi yang sudah dibuat bebas dari kesalahan dan telah memenuhi persyaratan pengguna. Oleh karena itu diperlukan *user acceptance testing* pada Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta.

*User acceptance testing* dilakukan dengan cara membuat daftar pertanyaan yang bisa mewakili seluruh kebutuhan yang sudah didefinisikan di bab analisis kebutuhan. Daftar pertanyaan ditanyakan ketika pengguna mencoba menggunakan Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta secara langsung. Berikut Tabel 7.2 sebagai perwakilan pengguna yang akan mengisi daftar pertanyaan. Pengujian Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta menggunakan tiga kriteria yang diuraikan pada Tabel 7.3.

**Tabel 7.2 Analisis Tester User Acceptance Testing**

No	Peran	Fungsi Sistem Yang Digunakan	Jumlah Populasi	Jumlah Tester (>=50%)
1.	Pimpinan (Pemilik Kantor)	1. Login	1	1
		2. Kelola data klien		
		3. Kelola <i>monitoring</i> aktivitas klien		
		4. Kelola <i>monitoring</i> bulanan klien		
		5. Kelola tanda terima dokumen		
		6. Kelola <i>monitoring</i> tahunan klien		
		7. Kelola proposal		
		8. Kelola staf		
		9. Validasi <i>monitoring</i> bulanan klien		
		10. Validasi <i>monitoring</i> tahunan klien		
		11. Persetujuan proposal		
		12. Melihat grafik aktivitas		
		13. Melihat grafik <i>monitoring</i> bulanan		
		14. Melihat grafik <i>monitoring</i> tahunan		
		15. Logout		

**Tabel 7.2 Analisis Tester User Acceptance Testing (lanjutan)**

No	Peran	Fungsi Sistem Yang Digunakan	Jumlah Populasi	Jumlah Tester (>=50%)
2.	Staf	1. Login	7	4
		2. Kelola data klien		
		3. Kelola <i>monitoring</i> aktivitas klien		
		4. Kelola <i>monitoring</i> bulanan klien		
		5. Kelola tanda terima dokumen		
		6. Kelola <i>monitoring</i> tahunan klien		
		7. Kelola proposal		
		8. <i>Request</i> validasi <i>monitoring</i> bulanan klien		
		9. Tambah tanda terima dokumen		
		10. <i>Request</i> validasi <i>monitoring</i> tahunan klien		
		11. <i>Request</i> persetujuan proposal		
		12. Logout		
Total Tester				5

Pada Tabel 7.2 menjelaskan jumlah responden yang terlibat dalam *user acceptance testing* dan terdapat beberapa fungsi sistem yang diuji.

**Tabel 7.3 Kriteria dan Pertanyaan User Acceptance Testing**

No	Kriteria	Definisi (Naik & Kshirasagar, 2008 )	Daftar Pertanyaan
1.	<i>Usability</i>	Evaluasi layanan atau produk melalui pengujian dengan pengguna yang mewakili. Bertujuan untuk memastikan bahwa sistem bisa mudah digunakan dan dipelajari oleh pengguna.	<ol style="list-style-type: none"> <li>1. Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta mudah digunakan?</li> <li>2. Apakah tampilan Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta nyaman ketika digunakan?</li> <li>3. Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta</li> </ol>



Tabel 7.3 Kriteria dan Pertanyaan *User Acceptance Testing* (lanjutan)

			mempermudah penyampaian informasi antar staf dan pimpinan?
			4. Apakah penyaringan informasi <i>monitoring</i> pekerjaan dapat mempermudah dalam mencari informasi <i>monitoring</i> pekerjaan tertentu?
			5. Apakah laporan informasi perpajakan mempermudah dalam melaporkan ke pimpinan?
2.	<i>Functional Correctness and Completeness</i>	<i>Functional Correctness and Completeness</i> adalah pengujian untuk mengetahui apakah sistem yang dibuat sudah sesuai kebutuhan yang diinginkan pengguna dan telah didefinisikan pada spesifikasi kebutuhan	<p>1. Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta membantu dalam pengelolaan data klien dengan baik?</p> <p>2. Apakah laporan <i>monitoring</i> pekerjaan berupa konsultasi, perpajakan dan tanda terima dalam setiap bulan dan tahun membantu untuk kemajuan kantor?</p> <p>3. Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta membantu dalam pengelolaan data konsultasi yang dilakukan klien dan staf?</p> <p>4. Apakah proses validasi <i>monitoring</i> perpajakan membantu sebagai pembuktian bahwa setiap pekerjaan selesai dikerjakan sesuai dengan proses yang ada?</p> <p>5. Apakah pembuatan bukti tanda terima membantu dalam dokumentasi proses serah terima dokumen?</p> <p>6. Apakah cetak data klien dan laporan pekerjaan staf membantu untuk arsip kantor sehingga dapat dimanfaatkan untuk masa depan?</p>



**Tabel 7.3 Kriteria dan Pertanyaan User Acceptance Testing (lanjutan)**

			7. Apakah grafik <i>monitoring</i> aktivitas dan <i>monitoring</i> perpajakan dapat membantu pimpinan dalam memantau jumlah data konsultasi dan perpajakan tiap staf dengan baik?
3.	<i>Confidentially and Availability</i>	<p><i>Confidentiality</i> adalah satu set aturan yang membatasi hak akses terhadap informasi.</p> <p><i>Availability</i> adalah jaminan akses yang dapat dipercaya terhadap informasi oleh orang-orang yang berwenang</p>	1. Apakah login user dibutuhkan untuk membantu keamanan informasi kantor dalam Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta ?

Setelah mendapatkan kriteria dan pertanyaan yang sesuai dengan kriteria yang diharapkan. Tahap selanjutnya adalah melakukan analisis hasil dari daftar pertanyaan yang sudah dijawab oleh staf sebagai *tester*. Kode UAT diwakilkan dengan kode UAT-P2-1, UAT-P2-2 dan UAT-P2-3 secara berurutan. UAT merupakan singkatan dari *User Acceptance Testing*, P2 menunjukkan aktor staf. Berikut UAT dengan aktor staf ditunjukkan pada Tabel 7.4:

**Tabel 7.4 Hasil User Acceptance Testing Staf**

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta mudah digunakan?	UAT-P2-1				2	2
2	Apakah tampilan Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta	UAT-P2-2			1	2	1



Tabel 7.4 Hasil *User Acceptance Testing* Staf (lanjutan)

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	nyaman ketika digunakan?						
3	Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta mempermudah penyampaian informasi antar staf dan pimpinan?	UAT-P2-3				3	1
4	Apakah penyaringan informasi <i>monitoring</i> pekerjaan dapat mempermudah dalam mencari informasi <i>monitoring</i> pekerjaan tertentu?	UAT-P2-4				2	2
5	Apakah laporan informasi perpajakan mempermudah dalam melaporkan ke pimpinan?	UAT-P2-5				3	1
6	Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta membantu dalam pengelolaan data klien dengan baik?	UAT-P2-6				1	3
7	Apakah laporan <i>monitoring</i> pekerjaan berupa konsultasi, perpajakan dan tanda terima dalam setiap bulan dan tahun	UAT-P2-7				3	1

Tabel 7.4 Hasil *User Acceptance Testing* Staf (lanjutan)

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	membantu untuk kemajuan kantor?						
8	Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta membantu dalam pengelolaan data konsultasi yang dilakukan klien dan staf?	UAT-P2-8			1	1	2
9	Apakah proses validasi <i>monitoring</i> perpajakan membantu sebagai pembuktian bahwa setiap pekerjaan selesai dikerjakan sesuai dengan proses yang ada?	UAT-P2-9			1	2	1
10	Apakah pembuatan bukti tanda terima membantu dalam dokumentasi proses serah terima dokumen?	UAT-P2-10					4
11	Apakah cetak data klien dan laporan pekerjaan staf membantu untuk arsip kantor sehingga dapat dimanfaatkan untuk masa depan?	UAT-P2-11			1	2	1



Tabel 7.4 Hasil *User Acceptance Testing* Staf (lanjutan)

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
12	Apakah login user dibutuhkan untuk membantu keamanan informasi kantor dalam Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta ?	UAT-P2-12					4
Total			0	0	4	21	23

Pada perhitungan skala likert dari hasil kuisisioner user acceptance testing staf diawali dengan menghitung nilai berdasarkan dari 4 responden staf yang sudah dipilih.

$$A = 23 \times 5 = 115$$

$$B = 21 \times 4 = 84$$

$$C = 4 \times 3 = 12$$

$$D = 0 \times 2 = 0$$

$$E = 0 \times 1 = 0$$

$$\text{Total Nilai} = A + B + C + D + E$$

$$\text{Total Nilai} = 115 + 84 + 12 + 0 + 0 = 211$$

Selanjutnya pada perhitungan skala *likert* yang diperoleh dari hasil kuisisioner *user acceptance testing* staf dengan menghitung nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 4 \times 12 = 240$$

Kemudian pada akhir perhitungan skala *likert* yang diperoleh dari hasil kuisisioner *user acceptance testing* staf dengan menggunakan rumus index untuk menghitung nilai persentase

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (211 \div 240) \times 100\% = 87,5\% (\text{Sangat Setuju})$$

Tabel 7.5 Hasil *User Acceptance Testing* Pimpinan

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta mudah digunakan?	UAT-P3-1				1	
2	Apakah tampilan Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta nyaman ketika digunakan?	UAT-P3-2				1	
3	Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta mempermudah penyampaian informasi antar staf dan pimpinan?	UAT-P3-3					1
4	Apakah penyaringan informasi <i>monitoring</i> pekerjaan dapat mempermudah dalam mencari informasi <i>monitoring</i> pekerjaan tertentu?	UAT-P3-4					1
5	Apakah laporan informasi perpajakan mempermudah dalam melaporkan ke pimpinan?	UAT-P3-5					1
6	Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor	UAT-P3-6					1



Tabel 7.5 Hasil *User Acceptance Testing* Pimpinan (lanjutan)

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	Konsultan Pajak I Gede Arianta membantu dalam pengelolaan data klien dengan baik?						
7	Apakah laporan <i>monitoring</i> pekerjaan berupa konsultasi, perpajakan dan tanda terima dalam setiap bulan dan tahun membantu untuk kemajuan kantor?	UAT-P3-7					1
8	Apakah Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta membantu dalam pengelolaan data konsultasi yang dilakukan klien dan staf?	UAT-P3-8					1
9	Apakah proses validasi <i>monitoring</i> perpajakan membantu sebagai pembuktian bahwa setiap pekerjaan selesai dikerjakan sesuai dengan proses yang ada?	UAT-P3-9					1



Tabel 7.5 Hasil *User Acceptance Testing* Pimpinan (lanjutan)

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
10	Apakah cetak data klien dan laporan pekerjaan staf membantu untuk arsip kantor sehingga dapat dimanfaatkan untuk masa depan?	UAT-P3-10					1
11	Apakah grafik <i>monitoring</i> aktivitas dan <i>monitoring</i> perpajakan dapat membantu pimpinan dalam memantau jumlah data konsultasi dan perpajakan tiap staf dengan baik?	UAT-P3-11					1
12	Apakah login user dibutuhkan untuk membantu keamanan informasi kantor dalam Sistem Informasi <i>Monitoring</i> Pekerjaan Kantor Konsultan Pajak I Gede Arianta ?	UAT-P3-12					1
Total			0	0	0	2	10

Pada perhitungan skala likert dari hasil kuisioner user acceptance testing staf diawali dengan menghitung nilai berdasarkan dari 1 responden pimpinan.

$$A = 10 \times 5 = 50$$

$$B = 2 \times 4 = 8$$

$$C = 0 \times 3 = 0$$

$$D = 0 \times 2 = 0$$

$$E = 0 \times 1 = 0$$

$$\text{Total Nilai} = A + B + C + D + E$$



$$\text{Total Nilai} = 50 + 8 + 0 + 0 + 0 = 58$$

Selanjutnya pada perhitungan skala *likert* yang diperoleh dari hasil kuisisioner *user acceptance testing* pimpinan dengan menghitung nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 1 \times 12 = 60$$

Kemudian pada akhir perhitungan skala *likert* yang diperoleh dari hasil kuisisioner *user acceptance testing* pimpinan dengan menggunakan rumus index untuk menghitung nilai persentase

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (58 \div 60) \times 100\% = 96,67\% \text{ (Sangat Setuju)}$$

### 7.1.3 Pengujian Waktu Sebelum Dan Sesudah Hasil Implementasi

Pengujian Perbandingan Waktu dilakukan untuk melihat hasil perbedaan waktu yang diperlukan oleh pimpinan menyelesaikan kebutuhan dalam mendapatkan data klien lebih cepat. Berikut tabel 7.6 merupakan hasil pengujian waktu sebelum implementasi sistem informasi pada proses pencarian data klien.

**Tabel 7.6 Pengujian Waktu Sebelum Implementasi**

No	Aktor	Aktivitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
1	Pimpinan	Menghubungi staf untuk mencari data klien	1 menit	5 menit	3 menit
2	Staf	Mencari data klien	5 menit	1 jam	32 menit 30 detik
3	Pimpinan	Mendapatkan data klien dari staf	1 menit	5 menit	20 menit 30 detik
Total Rata-Rata Waktu			7 menit	1 jam 10 menit	38 menit 30 detik

Berdasarkan Tabel 7.6 tersebut diketahui bahwa untuk aktivitas pencarian data klien didapatkan waktu penyelesaian terbaiknya memerlukan waktu 7 menit, waktu penyelesaian terburuknya memerlukan 1 jam 10 menit dan rata-rata dari waktu penyelesaiannya memerlukan 38 menit 30 detik.

Terjadinya perbedaan waktu dari waktu penyelesaian terbaik dan terburuk diakibatkan oleh faktor waktu dari aktivitas-aktivitas yang dilakukan oleh aktor staf. Waktu penyelesaian dari aktivitas pencarian data klien yang dilakukan oleh aktor staf bergantung kepada pencarian dokumen terkait letak dokumen dan terdapat aktivitas pekerjaan lain yang sedang dikerjakan, apabila staf tersebut lupa akan letak dokumen yang disimpan maka pencarian data yang diterima membutuhkan waktu yang lama lebih dari 1 jam dan bahkan bisa dianggap bahwa dokumen terkait data klien tersebut hilang.



Berikut pada Tabel 7.7 menunjukkan hasil pengujian waktu untuk aktivitas-aktivitas pencarian data klien setelah implementasi sistem yang mengacu kepada pencarian data klien sebelum implementasi sistem. Berdasarkan Tabel 7.7 tersebut diketahui bahwa untuk pencarian data klien setelah implementasi sistem didapatkan waktu penyelesaian terbaiknya memerlukan waktu 10 detik, waktu penyelesaian terburuknya memerlukan 50 detik dan rata-rata dari waktu penyelesaiannya memerlukan 30 detik. Perbedaan waktu ini dapat terjadi ketika pimpinan perlu memilih terlebih dahulu data klien mana yang ingin dilihat informasi lengkapnya.

**Tabel 7.7 Pengujian Waktu Sebelum Implementasi**

No	Aktor	Aktivitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
1	Pimpinan	Mencari dan mendapatkan data klien	10 detik	50 detik	30 detik
Total Rata-Rata Waktu			10 detik	50 detik	30 detik

## 7.2 Analisis Hasil

Pada analisis hasil pengujian *black box* berupa validasi sistem berdasarkan dari hasil pengujian validasi pada Tabel 7.1 yang menjelaskan bahwa nomor kode uji VT\_KKP\_1 hingga VT\_KKP\_19 adalah valid, artinya dari semua kasus uji yang sudah dilakukan dalam penelitian pada hasil pengujian yang didapatkan sudah tidak terdapat kesalahan atau *error* dan sesuai dengan apa yang diharapkan oleh peneliti untuk membangun sistem informasi. Pengujian validasi dilakukan secara keseluruhan yang merujuk pada *use case* yang sudah dibuat. Hasil yang didapatkan dari *validation testing* dapat disimpulkan bahwa sistem informasi yang dibangun dapat berjalan sesuai dengan hasil yang diharapkan.

Pada analisis hasil perhitungan kriteria *user acceptance testing* pimpinan yang telah dihitung dengan menggunakan *skala likert* menghasilkan nilai 96,67% yang berarti dalam interpretasi nilai berarti pihak pimpinan sangat setuju dengan sistem informasi *monitoring* pekerjaan yang sudah dibangun, sedangkan untuk *user acceptance testing* staf menghasilkan nilai 87,5% yang berarti dalam interpretasi nilai dari pihak staf juga sangat setuju dengan sistem informasi *monitoring* pekerjaan yang sudah dibangun.

Pada kriteria *usability* yang diuji oleh pimpinan sudah dianggap bagus untuk tampilan dan kemudahan menggunakan sistem. Menurut staf untuk kriteria *usability* pada sisi tampilan umumnya juga sudah dianggap bagus dan untuk kemudahan dalam menggunakan sistem tidak terlalu rumit karena menu atau tombol fungsi bisa dengan mudah digunakan, namun terdapat 1 staf yang memilih netral dikarenakan terdapat fungsi tombol yang ukurannya terlalu kecil dan samar untuk ditemukan seperti tombol tambah.

Pada kriteria *functional correctness and completeness* menurut pimpinan secara keseluruhan sudah sesuai dengan kebutuhan kantor. Menurut staf untuk kriteria *functional correctness and completeness* juga sudah diterima dengan baik dan sesuai dengan kebutuhan kantor, namun ada juga 1 staf yang bersifat netral terhadap kriteria yang berhubungan dengan pencatatan konsultasi dikarenakan menurut staf ini kegiatan konsultasi dirasa tidak terlalu bermasalah sehingga dianggap kurang perlu dilakukan pencatatan kegiatan konsultasi akan tetapi hal ini bukan berarti staf menolak akan fitur pencatatan ini namun bersifat netral untuk mendukung kemajuan dan kepentingan kantor. Pada kriteria *confidentially and availability* dianggap sangat perlu oleh pimpinan dan staf supaya data kantor lebih aman dan akses untuk tiap pengguna dibatasi sesuai kebutuhan.

Pada Tabel 7.8 menunjukkan tabel hasil perbandingan waktu antara aktivitas sebelum implementasi sistem dan setelah implementasi sistem. Aktivitas yang akan dibandingkan adalah aktivitas untuk kebutuhan dalam mendapatkan data klien. Waktu yang dijelaskan terdiri dari waktu terbaik, waktu terburuk dan waktu rata-ratanya untuk penyelesaian aktivitas.

**Tabel 7.8 Selisih Hasil Perbandingan Waktu**

Proses Bisnis	Total Rata-Rata Mendapatkan Data Klien		
	Terbaik	Terburuk	Rata-Rata
<b>Sebelum Implementasi</b>	7 menit	1 jam 10 menit	38 menit 30 detik
<b>Setelah Implementasi</b>	10 detik	50 detik	30 detik
<b>Selisih</b>	6 menit 50 detik	1 jam 9 menit 10 detik	38 menit

Tabel 7.8 tersebut diketahui bahwa untuk waktu dari aktivitas mendapatkan data klien sebelum implementasi sistem dibandingkan dengan setelah implementasi sistem didapatkan selisih waktu sebanyak 6 menit 50 detik untuk waktu penyelesaian terbaiknya, selisih 1 jam 9 menit 10 detik untuk waktu penyelesaian terburuknya dan selisih 38 menit untuk waktu rata-ratanya.

Berdasarkan data waktu yang dipaparkan pada Tabel 7.8 dapat diketahui bahwa hasil pengujian waktu proses dari aktivitas setelah implementasi sistem informasi dapat diselesaikan lebih cepat dibandingkan dengan waktu proses dari aktivitas sebelum implementasi sistem informasi.

## BAB 8 KESIMPULAN DAN SARAN

### 8.1 Kesimpulan

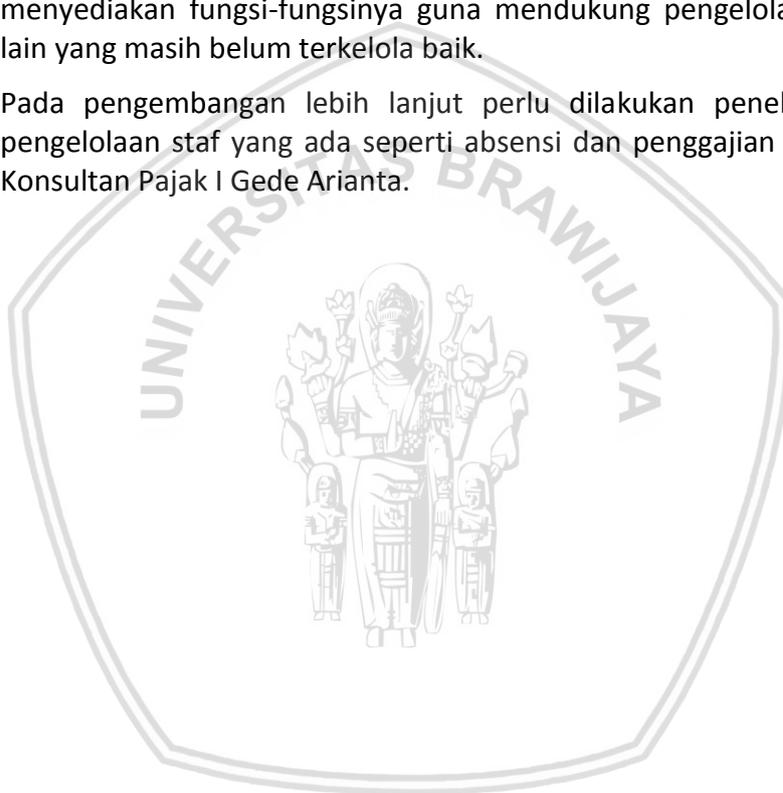
Berdasarkan dari penelitian yang sudah dilakukan, bisa diambil kesimpulan dari pembangunan dan pengujian Sistem Informasi *Monitoring* Pekerjaan Kantor Konsultan Pajak I Gede Arianta antara lain adalah sebagai berikut :

1. Berdasarkan pada tahap proses perancangan bisnis menghasilkan usulan proses bisnis yang mempermudah aktivitas dalam mengelola data klien, mengelola kegiatan konsultasi, mengelola perpajakan bulanan dan tahunan dan mengelola tanda terima dokumen. Kemudian pada permasalahan pencarian data klien menghasilkan kebutuhan sistem berupa kelola data klien, kebutuhan kelola *monitoring* aktivitas untuk menyelesaikan permasalahan pemantauan konsultasi, kebutuhan kelola *monitoring* perpajakan menyelesaikan permasalahan pemantauan perpajakan dan kelola tanda terima dokumen menyelesaikan permasalahan pertanggungjawaban serah terima dokumen.
2. Berdasarkan hasil penelitian pada proses perancangan menghasilkan *use case* mengelola data, *sequence diagram* kelola data klien, *class diagram* c\_klien dan m\_klien untuk menyelesaikan kebutuhan pengelolaan data klien. *Use case* melihat grafik *monitoring* aktivitas, *sequence diagram* kelola monitoring aktivitas, *class diagram* c\_monitoring\_aktivitas dan m\_monitoring\_aktivitas untuk menyelesaikan kebutuhan pemantauan konsultasi. Proses implementasi sistem dilakukan berdasarkan hasil dari perancangan sistem yang sudah didefinisikan. Implementasi sistem menggunakan bahasa pemrograman HTML, CSS, PHP, Javascript, Codeigniter sebagai *framework* PHP dan MySQL sebagai database sistem.
3. a. Pengujian yang digunakan menggunakan *black-box testing* berupa *validation testing* dan *user acceptance testing*. *Validation testing* membuktikan bahwa sistem telah diuji sebanyak 19 kasus uji dengan hasil uji 100% valid yang berarti sudah tidak ada *error* pada sistem informasi yang dibuat.  
b. UAT membuktikan bahwa telah menyelesaikan kebutuhan dalam mempermudah pengelolaan data klien, mempermudah pemantauan konsultasi, mempermudah proses pertanggungjawaban serah terima dokumen, mempermudah pemantauan perpajakan. *User acceptance testing* (UAT) menghasilkan nilai pimpinan 96,67% dapat menerima sistem yang dibuat dan staf 87,5% dapat menerima sistem yang dibuat  
c. Berdasarkan hasil pengujian perbandingan waktu telah menyelesaikan permasalahan pencarian data klien dan didapatkan bahwa waktu yang diperlukan oleh pimpinan untuk mendapatkan data klien ketika sebelum implementasi sistem rata-rata adalah 38 menit 30 detik, sedangkan proses tersebut dapat dipercepat setelah implementasi

sistem yang mana pimpinan dapat mendapatkan data klien dalam waktu rata-rata 30 detik. Selisih waktu antara sebelum dan sesudah implementasi sistem rata-rata 38 menit.

## 8.2 Saran

1. Pada implementasi lebih lanjut bisa meningkatkan keamanan sistem informasi sehingga bisa memberikan jaminan terhadap setiap data yang tersimpan dan tidak mudah dilihat oleh orang yang tidak memiliki hak akses.
2. Pada pengembangan lebih lanjut perlu dilakukan penelitian terkait jasa-jasa lain yang ada di Kantor Konsultan Pajak I Gede Arianta dan menyediakan fungsi-fungsinya guna mendukung pengelolaan jasa-jasa lain yang masih belum terkelola baik.
3. Pada pengembangan lebih lanjut perlu dilakukan penelitian terkait pengelolaan staf yang ada seperti absensi dan penggajian pada Kantor Konsultan Pajak I Gede Arianta.



## DAFTAR PUSTAKA

- A.S, R. & Shalahudin, M., 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Alshamrani, A. & Bahattab, A., 2015. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science*, XII(1), pp. 106-111.
- Brandenburg, L., 2017. *Bridging The Gap*. [Online]. Available at: <http://www.bridging-the-gap.com/as-is-business-process/>. [Diakses 29 03 2018].
- Castillote, G. A., 2017. Development of E-Toll Collection and Monitoring System. *Journal of Computing and Innovation (JCI)*, 1(1), pp. 15-20.
- Choizes, 2018. *Pengertian Skala Likert dan Contoh Cara Hitung Kuesionernya*. [Online]. Available at: <https://www.diedit.com/skala-likert/> [Diakses 10 Mei 2018].
- Comer, D. E., 2007. *The Internet book : everything you need to know about computer networking and how the internet works*. 4th ed. United States of America: Pearson Education.
- Goel, R. & Gupta, D. N., 2014. Survey on Acceptance Testing Technique. *International Journal of Software and Web Sciences (IJSWS)*, 8(1), pp. 20-23.
- H.M., J., 1990. *Analisis & Disain Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. 1 penyunt. Yogyakarta: Andi Offset.
- Indriati, 2010. [Online]. Available at: <http://indryz.lecture.ub.ac.id/files/2010/11/Pengujian-Validasi.docx> [Diakses 03 April 2018].
- Koespradono, 2013. SISTEM INFORMASI PENGOLAHAN DATA PERTUMBUHAN EKONOMI DAN KETIMPANGAN DI KABUPATEN KLATEN (TAHUN 2003-2012) MENGGUNAKAN FRAMEWORK CODEIGNITER. *Jurnal SCRIPT*, 1(1), pp. 46-53.
- Kumar, M., Singh, S. K. & Dwivedi, D. R. K., 2015. A Comparative Study of Black Box Testing and White Box Testing Techniques. *International Journal of Advance Research in Computer Science and Management Studies*, 3(10), pp. 32-44.
- Kurniawan, T. A., 2018. Pemodelan Use Case (UML) Evaluasi Terhadap beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 5(1), pp. 77-86.
- Laudon, K. C. a. J. P., 2014. *Management Information Systems: Managing the Digital Firm*. 13th ed. United States of America: Pearson Education.
- Mustaqbal, M. S., Firdaus, R. F. & Rahmadi, H., 2015. PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi

- Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), pp. 31-36.
- Naik & Kshirasagar, 2008 . *SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice*. 1st penyunt. United States : John Wiley & Sons, Inc., Hoboken, New Jersey.
- Object Management Group, 2011. *The Object Management Group® (OMG®)*. [Online]. Available at: <http://www.omg.org/spec/BPMN/2.0/PDF> [Diakses 10 Februari 2018].
- O'Brien, J. A. a. G. M., 2007. *Introduction to Information System*. 5th ed. New York: McGraw-Hill/Irwin.
- Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 7th ed. New York: McGraw-Hill.
- Rahni, A. A. A., 2015. DEVELOPMENT OF THE ONLINE STUDENT ATTENDANCE MONITORING SYSTEM (SAMSTM) BASED ON QR-CODES AND MOBILE DEVICES. *Journal of Engineering Science and Technology*, 1(2), pp. 29-40.
- Ramadhani, M. R., 2018. Analisis dan Perancangan Sistem Informasi Manajemen Donasi, Kegiatan, dan Relawan bagi Komunitas Sosial di Kota Malang (Studi Kasus: Komunitas TurunTangan Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(9), pp. 3102-3109.
- Ramdhani, M. A., 2015. PEMODELAN PROSES BISNIS SISTEM AKADEMIK MENGGUNAKAN PENDEKATAN BUSINESS PROCESS MODELLING NOTATION (BPMN) (STUDI KASUS INSTITUSI PERGURUAN TINGGI XYZ). *Jurnal Informasi*, VII(2), pp. 83-92.
- Rohayati, M., 2014. MEMBANGUN SISTEM INFORMASI MONITORING DATA INVENTORY DI VIO HOTEL INDONESIA. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 1(1), pp. 1-8.
- Sommerville, I., 2011. *Software Engeenering*. 9 ed. New York: Addison-Wesley.
- Tim Airputih, 2014. *CodeIgniter a powerful PHP framework with a very small footprint*. Jakarta Selatan: s.n.
- Valacich, J. a. C. S., 2010. *Information Systems Today: Managing in the Digital World*. 4th ed. United States of America: Pearson Education.
- Wazlawick, R. S., 2013. *Object-Oriented Analysis and Design for Information Systems*. Waltham : Morgan Kaufmann.
- Weske, M., 2007. *Business Process Management : Concepts, Languages, Architectures*. Berlin: Springer.
- Widodo, I. W., 2013. *Membangun Web Super Cepat dengan CodeIgniter GroceryCRUD dan TankAuth*. [e-book] In Wahyu Widodo. Tersedia di: Google books [Diakses 16 Januari 2018].