



**RANCANG BANGUN APLIKASI *INVENTORY* PADA CV. HERI JAYA  
TEKNIK**

**KERJA PRAKTIK**



UNIVERSITAS  
**Dinamika**

**Oleh:**

**JOHANES ADITYA KURNIAWAN**

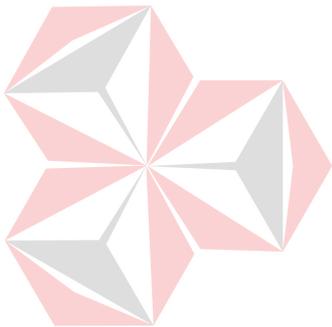
**11410100128**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA  
2015**

**RANCANG BANGUN APLIKASI PENGGAJIAN KARYAWAN  
OUTSOURCING PADA PT BANGUN PERSADA  
TRANSPORTATION**

**Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana**



UNIVERSITAS  
Disusun oleh :  
**Dinamika**

**Nama : ZAYED ELFASA**

**NIM : 11.41010.0127**

**Program : S1 (Strata Satu)**

**Jurusan : Sistem Informasi**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2015**

**SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH  
UNTUK KEPENTINGAN AKADEMIS**

Sebagai *civitas academica* Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Zayed Elfasa  
NIM : 11.41010.0127  
Program Studi : S1 Sistem Informasi  
Jurusan/Fakultas : Fakultas Teknologi dan Informatika

Demi pengembangan ilmu pengetahuan dan teknologi , menyetujui untuk memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya ***Hak Bebas Royalty Non Eksklusif (Non-Exclusive Royalty Free Right)*** atas karya ilmiah yang berjudul :

**RANCANG BANGUN APLIKASI PENGGAJIAN KARYAWAN  
OUTSOURCING PADA PT BANGUN PERSADA TRANSPORTATION**

Untuk disimpan, dialih mediakan, dikelola dalam bentuk pangkalan data (*database*), untuk didistribusikan atau dipublikasikan untuk kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

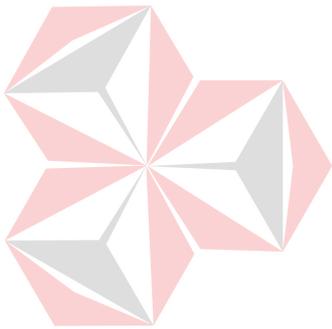
Surabaya , 22 Juni 2015

Zayed Elfasa  
NIM : 11410100127

## PERNYATAAN

Dengan ini saya menyatakan, bahwa kerja praktek ini adalah asli karya saya, bukan plagiat baik sebagian maupun keseluruhan. Karya atau pendapat orang lain yang ada dalam kerja praktek ini adalah semata hanya rujukan yang dicantumkan dalam daftar pustaka. Apabila dikemudian hari ditemukan adanya tindakan plagiat pada karya kerja praktek ini, saya bersedia untuk mengulang kerja praktek.

Surabaya, 22 Juni 2015



UNIVERSITAS  
**Dinamika**  
Zayed Elfasa

**LEMBAR PENGESAHAN**

**RANCANG BANGUN APLIKASI PENGGAJIAN KARYAWAN  
OUTSOURCING PADA PT BANGUN PERSADA TRANSPORTATION**

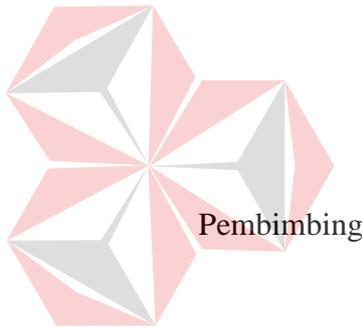
Laporan kerja praktek oleh

**ZAYED ELFASA**

**NIM : 11.41010.0127**

Telah diperiksa, diuji, dan disetujui

Surabaya, 22 Juni 2015



Disetujui:

UNIVERSITAS  
**Dinamika**  
Penyelia

Lilis Binawati, S.E., M.Ak.

NIDN. 0720056903

Juremi

Keuangan PT BPT

Mengetahui :

Ketua Program Studi

S1 Sistem Informasi

Vivine Nurcahyawati, M.Kom., OCP

NIDN. 0723018101

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas segala rahmat dan karunianya sehingga penulis dapat menyelesaikan laporan kerja praktek ini. Semoga laporan yang disusun oleh penulis ini dapat bermanfaat bagi pembaca.

Dalam penyusunan laporan kerja praktek ini, penulis ingin menyampaikan terima kasih kepada :

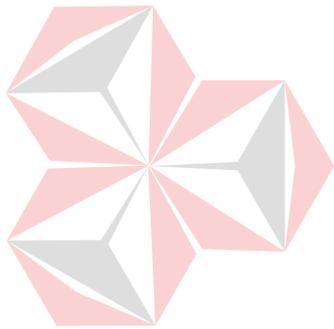
1. Orang tua yang telah mendoakan, membimbing, serta memberikan dukungan penuh sampai saat ini.
2. Ibu Vivine Nurcahyawati, M.Kom., OCP selaku Ketua Program Studi S1 Sistem Informasi.
3. Ibu Lilis Binawati, S.E., M.Ak., selaku dosen pembimbing yang selalu memberikan masukan dan arahan terkait dengan laporan dan program.
4. Bapak Atok Lindarto selaku *Manager* Sumber Daya Manusia PT Bangun Persada Transportation (PT BPT) yang telah memberikan kesempatan untuk kerja praktek di perusahaan tersebut.
5. KUPU Corp, UKM LUG (*Linux User Group*) Institus Bisnis dan Informatika Stikom Surabaya, dan teman-teman kampus yang telah memberikan masukan dan semangat hingga dapat terselesaikan laporan ini.

Pada kesempatan ini penulis menyadari banyak kekurangan dan kesalahan. Untuk itu penulis meminta maaf akan kesalahan baik dari penulisan maupun dari isi dan tidak lupa penulis mengharapkan kritik dan saran yang bersifat

membangun dari pembaca. Sehingga kritik dan saran yang membangun dari pembaca dapat menjadi pembelajaran bagi penulis.

Surabaya , 22 Juni 2015

Penulis

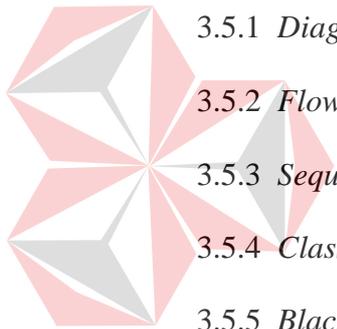


UNIVERSITAS  
**Dinamika**

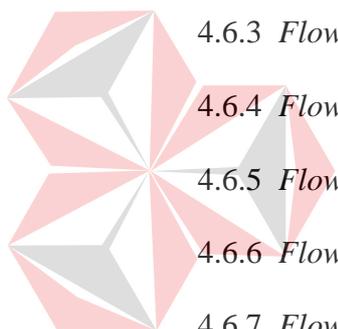
## DAFTAR ISI

	Halaman
ABSTRAK .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xvi
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II GAMBARAN UMUM PERUSAHAAN.....	7
2.1 Sejarah Perusahaan.....	7
2.2 Visi dan Misi Perusahaan .....	8
2.2.1 Visi Perusahaan .....	8
2.2.2 Misi Perusahaan .....	8
2.3 <i>Product &amp; Service</i> .....	8
2.3.1 Jasa Penyedia Tenaga Kerja.....	8
2.3.2 Pengadaan Barang dan Jasa.....	9
2.3.3 Jasa Konstruksi.....	9

2.4 Struktur Organisasi.....	9
<b>BAB III TINJAUAN PUSTAKA .....</b>	<b>11</b>
3.1 <i>Outsourcing</i> .....	11
3.2 Penggajian .....	12
3.3 Konsep Dasar Sistem Informasi dan Aplikasi.....	13
3.3.1 Sistem .....	13
3.3.2 Sistem Informasi .....	14
3.3.3 Aplikasi .....	14
3.4 Analisis dan Perancangan Sistem.....	15
3.5 <i>Undefined Modeling Language (UML)</i> .....	15
3.5.1 <i>Diagram Use Case</i> .....	15
3.5.2 <i>Flow Of Events</i> .....	18
3.5.3 <i>Sequence Diagram</i> .....	19
3.5.4 <i>Class Diagram</i> .....	19
3.5.5 <i>Black Box Testing</i> .....	20
3.5.6 <i>Entity Relational Diagram (ERD)</i> .....	20
3.6 Teori Yang Terkait Dengan Penyelesaian Masalah .....	25
3.6.1 <i>Microsoft Visual C# .NET</i> .....	25
3.6.2 <i>SQL Server 2008</i> .....	25
3.6.3 <i>Crystal Report</i> .....	26
<b>BAB IV METODE PENELITIAN .....</b>	<b>27</b>
4.1 <i>System Development Life Cycle (SDLC)</i> .....	27
4.1.1 <i>Fungsi System Development Life Cycle</i> .....	28
4.2 <i>Waterfall Programming</i> .....	28

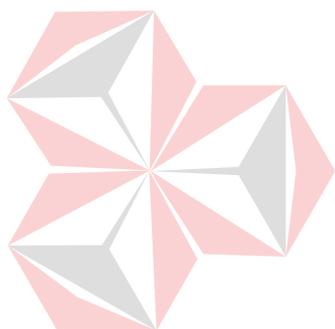


4.3 Model Pengembangan .....	29
4.4 Prosedur Penelitian.....	30
4.4.1 <i>Block Diagram</i> .....	31
4.5 Analisa Sistem.....	32
4.5.1 Identifikasi Masalah .....	33
4.5.2 Spesifikasi Aplikasi.....	33
4.5.3 Lingkungan Operasi .....	34
4.6 <i>Undefined Modeling Language</i> (UML) .....	34
4.6.1 <i>Use Case Diagram</i> .....	35
4.6.2 <i>Flow Of Events Use Case Login</i> .....	37
4.6.3 <i>Flow Of Events Manage Data Pengguna</i> .....	38
4.6.4 <i>Flow Of Events Data Master Profesi</i> .....	39
4.6.5 <i>Flow Of Events Manage Data Master Rekanan</i> .....	40
4.6.6 <i>Flow Of Events Manage Data Master Gaji Pokok</i> .....	41
4.6.7 <i>Flow Of Events Manage Data Master Tunjangan &amp; Profesi</i> .....	42
4.6.8 <i>Flow Of Events Manage Master Tenaga Kerja Outsourcing</i> .....	44
4.6.9 <i>Flow Of Events Manage Data Penggajian</i> .....	45
4.6.10 <i>Flow Of Events Cetak Slip Gaji</i> .....	47
4.6.11 <i>Flow Of Events Cetak Laporan Penggajian</i> .....	47
4.7 Sequence Diagram.....	48
4.7.1 <i>Sequence Diagram Login</i> .....	49
4.7.2 <i>Sequence Diagram Manage Data Pengguna</i> .....	49
4.7.3 <i>Sequence Diagram Manage Data Master Profesi</i> .....	50
4.7.4 <i>Sequence Diagram Manage Data Master Rekanan</i> .....	52



4.7.5	<i>Sequence Diagram Manage Data Master Gaji Pokok</i>	53
4.7.6	<i>Sequence Diagram Manage Data Master Tunjangan &amp; Profesi</i>	54
4.7.7	<i>Sequence Diagram Tenaga Kerja Outsourcing</i>	55
4.7.8	<i>Sequence Diagram Manage Data Penggajian</i>	56
4.7.9	<i>Sequence Diagram Cetak Slip Gaji</i>	57
4.7.10	<i>Sequence Diagram Cetak Laporan Penggajian</i>	57
4.8	<i>Class Diagram</i>	58
4.8.1	<i>Class Login</i>	58
4.8.2	<i>Class Pengguna</i>	59
4.8.3	<i>Class Admin</i>	58
4.8.4	<i>Class Master Profesi</i>	60
4.8.5	<i>Class Master Rekanan</i>	60
4.8.6	<i>Class Master Gaji Pokok</i>	61
4.8.7	<i>Class Master Tunjangan</i>	62
4.8.8	<i>Class Master OS</i>	62
4.8.9	<i>Class Penggajian</i>	63
4.8.10	<i>Relasi Class Diagram</i>	64
4.9	<i>Entity Relational Diagram (ERD)</i>	65
4.9.1	<i>Conceptual Data Model (CDM)</i>	65
4.9.2	<i>Physical Data Model (PDM)</i>	66
4.9.3	<i>Struktur Basis Data dan Tabel</i>	67
BAB V HASIL DAN PEMBAHASAN		73
5.1	<i>Disain Input Output</i>	73
5.1.1	<i>Implementasi dan Evaluasi</i>	83

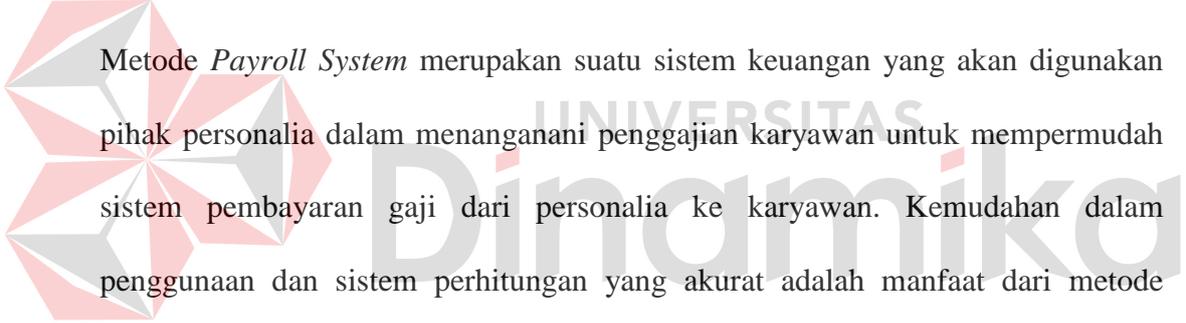
5.1.2 Kebutuhan Sistem .....	83
5.1.3 Penjelasan Program .....	84
BAB VI Penutup .....	95
6.1 Kesimpulan.....	95
6.2 Saran.....	96
DAFTAR PUSTAKA .....	97
BIODATA PENULIS .....	99
LAMPIRAN.....	100



UNIVERSITAS  
**Dinamika**

## ABSTRAK

Penggajian untuk karyawan *outsourcing* adalah hal vital pada PT Bangun Persada Transportation (PT BPT) dan bersifat penting karena menyangkut kelangsungan kinerja karyawan. Perhitungan yang akurat dan detil adalah hal mutlak yang diperlukan pada penggajian. Faktor-faktor yang menjadi komponen perhitungan gaji karyawan mulai dari jumlah absen, profesi, bermacam-macam tunjangan yang disesuaikan oleh rekanan, jamsostek, dan berbagai jenis potongan, dibuat secara manual menggunakan aplikasi *Microsoft Excel* yang cenderung mengakibatkan terjadinya kemungkinan ketidakakuratan perhitungan karena adanya data yang tidak masuk dalam parameter perhitungan penggajian.



Metode *Payroll System* merupakan suatu sistem keuangan yang akan digunakan pihak personalia dalam menanganani penggajian karyawan untuk mempermudah sistem pembayaran gaji dari personalia ke karyawan. Kemudahan dalam penggunaan dan sistem perhitungan yang akurat adalah manfaat dari metode *Payroll System* melalui pembuatan aplikasi penggajian yang sesuai dengan kebutuhan staff keuangan PT BPT dalam menghitung gaji karyawan *outsourcing* secara tepat dan akurat.

Perhitungan gaji kotor karyawan *outsourcing* berdasarkan jumlah absensi, profesi, dan bermacam-macam tunjangan. Apabila gaji kotor telah dikurangi dengan pinjaman, dan jamsostek maka akan didapatkan gaji bersih yang diterima oleh karyawan beserta slip gajinya.

Kata kunci: aplikasi, karyawan *outsourcing*, *payroll*, penggajian, PT Bangun Persada Transportation (PT BPT).

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Informasi merupakan salah satu kunci penting dalam kehidupan. Semua kegiatan manusia di zaman ini memerlukan informasi dan dapat dikatakan bahwa semua kegiatan manusia dituntut untuk menghasilkan informasi. Salah satu cara mendapatkan dan menghasilkan informasi adalah penggunaan komputer di era teknologi ini. Seiring perkembangan pesat di berbagai bidang maka tiap-tiap instansi sangat membutuhkan informasi yang cepat, tepat, dan akurat.

Pada era teknologi informasi ini sistem informasi banyak digunakan untuk menunjang kemajuan perusahaan tersebut agar lebih maju dan berkembang. Sistem informasi yang baik didukung oleh fasilitas dan sumber daya manusia yang dapat bersaing untuk mengelolanya.

PT Bangun Persada Transportation (PT BPT) adalah perusahaan yang bergerak dalam penyaluran karyawan (*outsourcing*) pada berbagai perusahaan yang membutuhkannya. Karyawan yang disediakan oleh perusahaan ini bermacam-macam seperti *costumer service*, *driver ambulance*, mekanik, *office boy*, dan masih banyak lainnya. Karyawan tersebut disalurkan kepada rekanan yang telah memiliki kontrak kerja dengan PT BPT selama beberapa tahun. Setiap bulan karyawan tersebut akan mendapatkan gaji yang dikontrol oleh PT BPT. PT BPT akan mendapatkan rekap data absensi dari rekanan terkait jumlah kehadiran, jumlah ketidakhadiran, dan

lembur, kemudian PT BPT mengolah data tersebut sebagai data *input* perhitungan gaji karyawan *outsourcing*. Selain rekap data absensi PT BPT juga akan mendapatkan rekap data berbagai tunjangan yang telah dimasukkan ke dalam kontrak kerja dan rekap data peminjaman sejumlah uang oleh para karyawan. Rekap data tersebut merupakan data yang akan di-*input*-kan pada perhitungan gaji karyawan *outsourcing*.

Selama ini PT BPT telah melakukan perhitungan gaji untuk karyawan *outsourcing* dengan menggunakan aplikasi perkantoran berupa *spreadsheet* dimana data karyawan *outsourcing* yang ditempatkan setiap rekanan dibuat dalam satu file *spreadsheet* sehingga setiap bulan staff PT BPT akan membuat file-file baru untuk melakukan pengolahan penggajian. Perhitungan gaji, angsuran, peminjaman, dan tunjangan dibuat manual dengan *spreadsheet* ini, yang mengakibatkan terdapat kemungkinan ketidak-akuratan perhitungan karena adanya data yang tidak masuk dalam parameter perhitungan penggajian.

Solusi dari permasalahan yang telah dipaparkan diatas adalah pembuatan aplikasi penggajian yang sesuai dengan kebutuhan staff keuangan PT BPT dalam menghitung gaji karyawan *outsourcing* secara tepat dan akurat.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka dapat dirumuskan permasalahan yang meliputi :

1. Bagaimana membangun dan merancang aplikasi penggajian karyawan *outsourcing* pada PT BPT ?

2. Bagaimana memproses perhitungan gaji secara otomatis sehingga mendapatkan hasil perhitungan penggajian yang tepat dan akurat untuk setiap karyawan *outsourcing*?
3. Bagaimana membuat laporan (*output*) penggajian karyawan *outsourcing* sesuai kebutuhan PT BPT?

### 1.3 Batasan Masalah

Berdasarkan pada rumusan masalah tersebut di atas, maka di dalam penelitian ini penulis memberikan batasan masalah terfokus pada masalah yang akan dibahas mengenai segala hal yang berkaitan dengan aplikasi penggajian pada PT BPT yaitu :

1. Aplikasi yang dibangun menghasilkan *output* laporan perhitungan penggajian karyawan *outsourcing* yang terkait data absensi, tunjangan hadir/transport, tunjangan *shift*, tunjangan nutrisi/makanan, tunjangan jabatan, peminjaman, jamsostek, dan pajak penghasilan. Output laporan yang dihasilkan oleh aplikasi ini disesuaikan dengan kebutuhan PT BPT yakni laporan rekap perhitungan gaji per bulan di setiap rekanan dan slip gaji untuk masing-masing karyawan *outsourcing*.
2. Aplikasi yang dirancang tidak terkait dengan pencatatan akuntansi.

### 1.4 Tujuan

Tujuan dari pembuatan aplikasi penggajian karyawan *outsourcing* pada PT BPT ini adalah :

1. Merancang aplikasi penggajian karyawan *outsourcing* pada PT BPT.

2. Menangani proses perhitungan gaji karyawan *outsourcing* di setiap rekanan dengan tepat dan akurat.
3. Menghasilkan laporan penggajian berupa laporan slip gaji untuk masing-masing karyawan *outsourcing* dan laporan rekap perhitungan gaji per bulan.

### **1.5 Manfaat**

Beberapa manfaat dari aplikasi penggajian karyawan *outsourcing* pada PT BPT ini antara lain:

1. Bagi Manajer PT BPT :
  - a. Memberi laporan penggajian karyawan *outsourcing* di setiap rekanan.
2. Bagi Staff PT BPT :
  - a. Meminimalisir kesalahan penghitungan gaji karyawan.
  - b. Mempermudah penghitungan gaji karyawan *outsourcing* karena diotomatisasi.
  - c. Mempermudah peng-*input*-an data-data yang diperlukan untuk penggajian karyawan *outsourcing*.
  - d. Memberi laporan penggajian dalam bentuk slip gaji untuk karyawan *outsourcing* dan laporan tanda terima gaji.

### **1.6 Sistematika penulisan**

Untuk memberikan gambaran menyeluruh tentang masalah yang sedang dibahas, maka sistematika penulisan laporan aplikasi penggajian karyawan *outsourcing* pada PT BPT adalah sebagai berikut:

## BAB I PENDAHULUAN

Pada bab ini dikemukakan hal-hal yang menjadi latar belakang, rumusan masalah, batasan masalah, tujuan yang ingin dicapai, manfaat aplikasi bagi perusahaan, serta sistematika penulisan laporan kerja praktek ini.

## BAB II GAMBARAN UMUM PERUSAHAAN

Pada bab ini membahas tentang gambaran umum PT BPT, sejarah perusahaan, visi dan misi perusahaan, dan struktur organisasi.

## BAB III TINJAUAN PUSTAKA

Pada bab ini membahas teori singkat yang berhubungan dengan pelaksanaan kerja praktek. Teori – teori ini dijadikan bahan acuan bagi penulis untuk menyelesaikan masalah.

## BAB IV METODE PENELITIAN

Pada bab membahas mengenai metode umum dalam pengembangan aplikasi. Pada kerja praktek ini akan menggunakan metode pengembangan aplikasi yaitu *System Development Life Cycle (SDLC)* menggunakan model *waterfall programming*. Pada bab ini juga dibahas mengenai gambaran aplikasi yang sedang dirancang dalam bentuk UML yang terdiri dari *use case bussiness*, *use case system*,

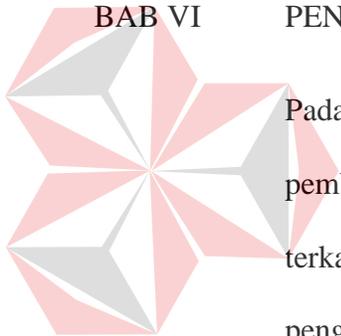
*activity diagram, integration diagram, sequence diagram, collaboration diagram, class diagram, statechart diagram, entity relationship diagram, conceptual data modelling, dan physical data modelling* mengenai perancangan aplikasi yang dibuat.

## BAB V HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai desain *input/output*, serta detail aplikasi. Detail aplikasi sudah dalam bentuk program jadi yang telah siap untuk dijalankan dan digunakan.

## BAB VI PENUTUP

Pada bab ini dibahas mengenai kesimpulan dari perancangan dan pembuatan aplikasi penggajian karyawan *outsourcing* pada PT BPT terkait dengan tujuan dan permasalahan yang ada, serta saran untuk pengembangan aplikasi di masa mendatang.



UNIVERSITAS  
Dinamika

## BAB II

### GAMBARAN UMUM PERUSAHAAN

#### 2.1 Sejarah Perusahaan

PT Bangun Persada Transportation (BPT) merupakan badan usaha swasta yang bergerak di bidang Jasa Penyedia dan Penyalur Tenaga Kerja (*outsourcing*), pengadaan barang, jasa transportasi serta konstruksi yang berkantor di Sidoarjo serta terdaftar resmi pada pemerintah, dan juga anggota dari beberapa asosiasi sesuai dengan Tanda Daftar Perusahaan (TDP) Nomor, 13.17.1.49.02113 dan Surat Ijin Disnaker Nomor.KEP:118/1164/404.3.3/XII/2011.

PT BPT telah berdiri sejak 21 Desember 2005 sesuai dengan akta notaris yang dikeluarkan oleh Sujayanto SH., M.Hum No. 116 pada 21 Desember 2005. Kantor pusat yang ditempati berada di Jl. Raya Ponokawan No. 07, Krian – Sidoarjo sesuai dengan dengan Direktur Utama adalah Herry Siswanto, ST. PT BPT memiliki banyak rekanan (*client*) baik dari instansi pemerintahan maupun swasta. Daftar rekanan yang ada diantaranya yaitu PT. Jasa Marga, PT. Nujyasumo Agung, PT. Marga Bumi Matra Raya, PA. Satya Ragam, PT. Jaya Kertas, PT Retdtroindo Nusantara, PT. Temprina Media Grafika, CV. Intan Transport dan lain - lain.

Sebagai perusahaan yang terus berkembang dan selalu unggul dari para pesaingnya, PT BPT bertujuan memberikan pelayanan secara profesional, terintegrasi dan optimal kepada rekanan yang membutuhkan, baik instansi pemerintahan maupun pihak swasta dan sangat spesifik dalam menjamin SDM yang dihasilkan, salahsatunya dengan memantau dari kinerja para karyawan *outsourcing*.

## 2.2 Visi dan Misi Perusahaan

Sebagai perusahaan yang bergerak di bidang penyalur jasa tenaga kerja , PT BPT (BPT) memiliki visi dan misi dalam menjalankan kegiatan oprasionalnya.

### 2.2.1 Visi Perusahaan

Mengutamakan pelayanan dan kerja keras tanpa batas.

### 2.2.2 Misi Perusahaan

Menjadikan PT BPT sebagai perusahaan terkemuka, maju, handal, dan profesional.

## 2.3 *Product & Service*

### 2.3.1 **Jasa Penyedia Tenaga Kerja**

1. Satuan Keamanan (*Security*)  
Perkantoran, pabrik, perumahan, proyek, mall, perbankan, gerbang tol, pengawalan khusus atau *body guard*.
2. Kebersihan (*Cleaning Service*)  
Perkantoran, pabrik, perumahan, proyek, mall, perbankan, gerbang tol, dan ruas jalan tol.
3. Tenaga produksi, administrasi, kasir, pengumpul tol dan Sopir.  
Perkantoran, pabrik, perumahan, proyek, mall, perbankan, gerbang tol.

### 2.3.2 Pengadaan Barang dan Jasa

#### 1. Peralatan Perlengkapan

Pertanian, peternakan, pertambangan, perikanan, pertamanan, kehutanan, kesehatan, percetakan, farmasi, peralatan medis, peralatan keamanan (HT, Kaca Miroll, CCTV, Metal detektor), peralatan *cleaning service* (mesin potong rumput, *vacum cleaner*, pengharum ruangan, dan lain-lain).

#### 2. Kendaraan

Mobil DEREK, kendaraan operasional kantor, kendaraan angkut barang, suku cadang (*sperre part*) kendaraan.

### 2.3.3 Jasa Konstruksi

#### 1. Arsitektur

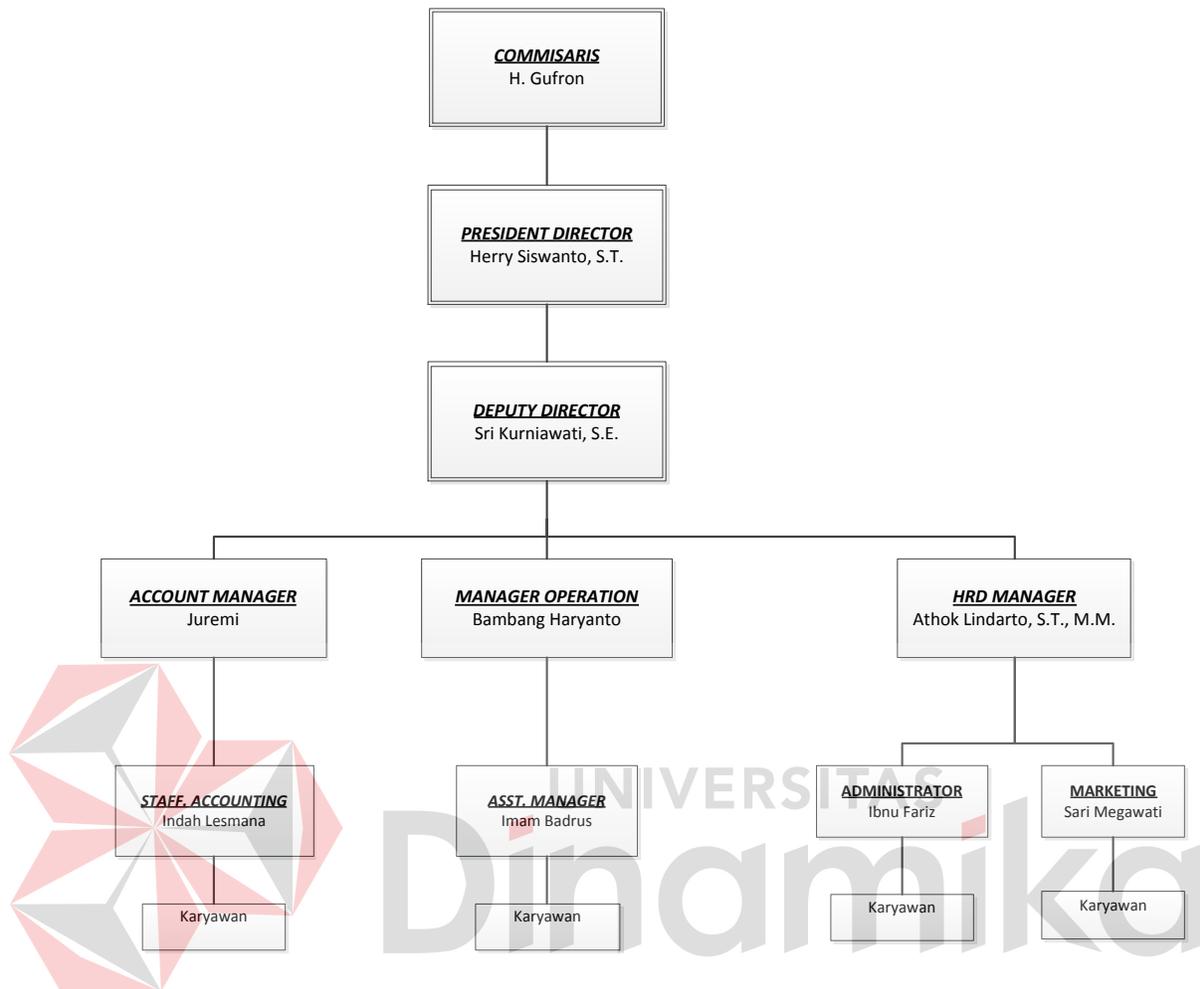
Perumahan dan pertamanan.

#### 2. Sipil

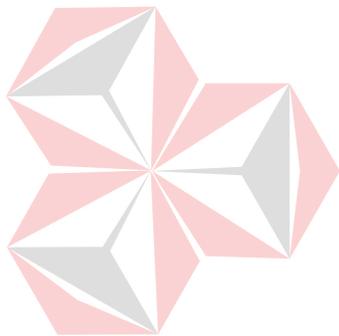
Jalan Raya, Irigasi termasuk perawatannya.

### 2.4. Struktur Organisasi

Struktur organisasi yang terdapat pada PT BPT ini memiliki bagian yang menangani semua kegiatan yang ada pada organisasi ini. Berikut bagan struktur organisasi yang dimiliki oleh PT BPT.



Gambar 2.1 Struktur Organisasi PT BPT



UNIVERSITAS  
**Dinamika**

## BAB III

### TINJAUAN PUSTAKA

Tinjauan pustaka digunakan untuk menyelesaikan masalah secara sistematis. Pada bab ini akan membahas tinjauan pustaka yang meliputi tinjauan pustaka mengenai hal-hal dari permasalahan yang ada dan tinjauan pustaka yang membahas tentang ilmu yang terkait dalam permasalahan tersebut.

#### 3.1 *Outsourcing*

Pengertian tentang *outsourcing* terus berkembang dari masa ke masa seiring dengan perkembangan teknologi, perkembangan industri, serta perkembangan pemikiran tentang *outsourcing*.

*Outsourcing* berasal dari bahasa Inggris yang berarti “alih daya”. *Outsourcing* memiliki nama lain, yaitu *contracting out*, yang merupakan sebuah pemindahan operasi dari satu perusahaan ke tempat lain. Hal ini biasanya dilakukan untuk memperkecil biaya produksi atau untuk memusatkan perhatian kepada hal lain. Di negara-negara maju seperti Eropa dan Amerika, pemanfaatan *outsourcing* sudah sangat mengglobal, sehingga menjadi sarana bagi perusahaan untuk lebih berkonsentrasi pada *core business*-nya dan fokus pada keunggulan produk servis.

Secara umum, *outsourcing* diartikan sebagai pemindahan atau pendelegasian beberapa proses bisnis kepada suatu badan penyedia jasa, di mana badan penyedia jasa tersebut melakukan proses administrasi dan manajemen berdasarkan definisi serta kriteria yang telah disepakati.

Menurut Sehat Damanik (2006), secara sederhana, *outsourcing* dipahami sebagai pendelegasian operasi dan manajemen harian dari suatu proses bisnis kepada pihak luar (perusahaan penyedia jasa *outsourcing*). Melalui pendelegasian, pengelolaan tidak lagi dilakukan oleh perusahaan, melainkan dilimpahkan kepada perusahaan jasa *outsourcing*.

Dari pengertian tersebut, dapat disimpulkan bahwa *outsourcing* adalah suatu bentuk perjanjian kerja antara perusahaan pengguna jasa dengan perusahaan penyedia jasa, di mana perusahaan pengguna jasa meminta kepada perusahaan penyedia jasa untuk menyediakan tenaga kerja yang diperlukan agar bekerja di perusahaan pengguna jasa dengan membayar sejumlah gaji tetap yang dibayarkan oleh perusahaan penyedia jasa.

### 3.2 Penggajian

Menurut Warren, Reeve, dan Fess (2006, p.552) dalam akuntansi, istilah gaji diartikan sebagai jumlah tertentu yang dibayarkan kepada karyawan untuk jasa yang diberikan selama periode tertentu.

Menurut Horngren, Harrison dan Bamber (2006, p.430) gaji merupakan pendapatan yang jumlahnya dihitung per tahun, per bulan, atau per minggu, sedangkan upah merupakan pendapatan yang dihitung berdasarkan tarif per jam. Gaji merupakan hal yang penting karena:

1. Para karyawan sangat sensitif terhadap kesalahan atau ketidakwajaran dalam gaji.

2. Untuk menjaga moral karyawan dengan cara membayar gaji tepat waktu dan dengan jumlah yang akurat.
3. Merupakan hal yang diatur dengan berbagai peraturan pemerintah federal atau negara bagian.
4. Mempunyai efek yang signifikan terhadap besar laba bersih pada sebagian besar usaha.

### **3.3 Konsep Dasar Sistem Informasi dan Aplikasi**

Konsep dasar dari Sistem Informasi terbagi atas dua pengertian yaitu pertama adalah sistem dan yang kedua adalah sistem informasi itu sendiri.

#### **3.3.1 Sistem**

Definisi sistem dapat dibagi menjadi dua pendekatan yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya (Herlambang, 2005).

### 3.3.2 Sistem Informasi

Data adalah fakta-fakta atau kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya. Sehingga sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya (Herlambang, 2005).

### 3.3.3 Aplikasi

Menurut Jogiyanto (2006) aplikasi merupakan sistem yang dirancang dan disusun sedemikian rupa untuk menghasilkan informasi yang terpadu dengan menggunakan sarana komputer sebagai penunjangnya.

Menurut Kamus Besar Bahasa Indonesia (1998 : 52) aplikasi adalah penerapan dari perancangan sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Aplikasi adalah program siap pakai. Program yang dapat difungsikan untuk berbagai macam kegiatan untuk mendukung bisnis. Contoh – contoh dari aplikasi adalah program pemrosesan kata dan *web browser*. Istilah aplikasi mulai masuk perlahan pada ranah teknologi informasi pada tahun 1993 yang biasa disebut dengan app.

### 3.4 Analisis dan Perancangan Sistem

Menurut (Jogiyanto, 2006) analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan dan pengembangannya. Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem.

### 3.5 *Undefined Modeling Language (UML)*

UML adalah sebuah bahasa pemodelan visual yang memungkinkan pengembangan sistem untuk membuat *blueprints* yang menggambarkan suatu visi dalam sebuah standar dengan cara yang mudah dimengerti serta menyediakan mekanisme untuk membagikan dan mengkomunikasikan visi tersebut kepada orang lain secara efektif (Schmuller,2004:7). Notasi UML dibuat sebagai kolaborasi dari Grady Booch, DR. James Rumbaugh, Ivar Jacobson, Rebecca Wirfs-Brock, Peter Yourdon, dan lainnya. Penggabungan beberapa metode menjadi UML dimulai tahun 1993.

#### 3.5.1 *Diagram Use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor dalam sistem yang akan dikembangkan. *Use case* sendiri adalah fungsionalitas atau persyaratan-persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan tersebut

menurut pandangan pemakai sistem sedangkan aktor bisa berupa orang, peralatan atau sistem lain yang berinteraksi terhadap sistem yang akan dibangun (Sholiq, 2010). Pada diagram *use case* terdapat elemen-elemen untuk menjelaskan maksud dari *use case*. Elemen-elemen yang digunakan dalam pemodelan *use case* sistem yaitu :

### 1. Aktor

Aktor merupakan semua yang ada di luar ruang lingkup sistem perangkat lunak dan berinteraksi dengan sistem perangkat lunak tersebut (Sholiq, 2010: 82).



Gambar 3.1 Contoh aktor *use case*

### 2. Notasi *Use Case*

*Use case* digunakan untuk menyatakan bagian tingkat tinggi fungsionalitas yang disediakan oleh sistem. Dengan kata lain, *use case* menggambarkan bagaimana seseorang sebagai pengguna berinteraksi dengan sistem (Sholiq, 2010: 85).



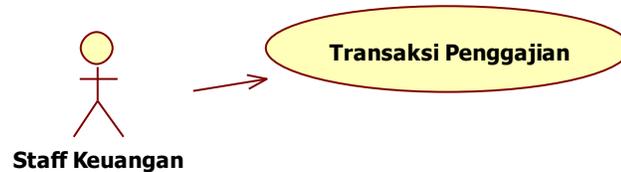
Gambar 3.2 Contoh notasi *use case*

### 3. Relasi

Relasi digunakan untuk membentuk diagram *use case* agar saling berhubungan.

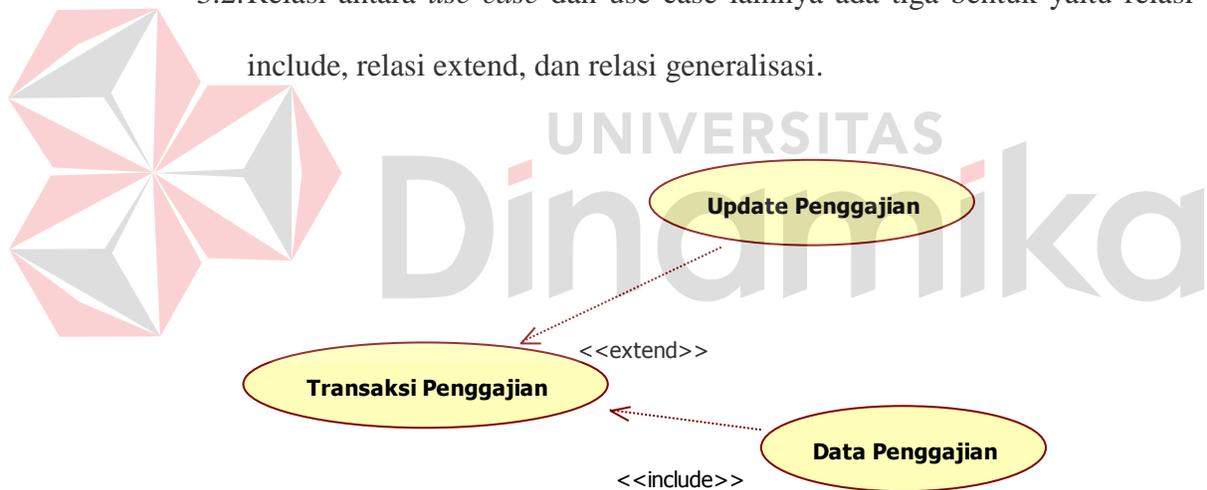
Secara umum relasi diagram dalam *use case* terbagi menjadi tiga macam yaitu :

#### 3.1. Relasi antara aktor dengan *use case* yang disebut relasi asosiasi.



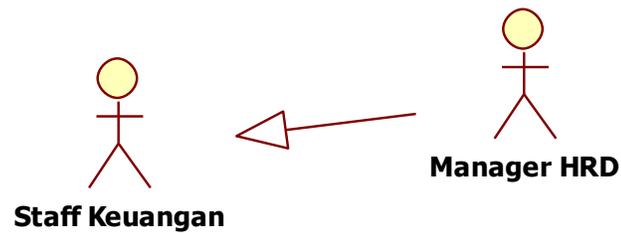
Gambar 3.3 Contoh relasi asosiasi

#### 3.2. Relasi antara *use case* dan use case lainnya ada tiga bentuk yaitu relasi include, relasi extend, dan relasi generalisasi.



Gambar 3.4 Contoh relasi *include* dan relasi *extends*

#### 3.3. Relasi antara aktor hanya digunakan untuk satu relasi yaitu relasi generalisasi (Sholih, 2010: 101-102).



Gambar 3.5 Contoh relasi Generalisasi

### 3.5.2 Flow Of Events

*Flow of events* digunakan untuk mendokumentasikan aliran logika dalam *use case*, yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan (Sholih, 2010:102). Namun pada bagian ini tidak dijelaskan apakah sistem dibangun menggunakan bahasa pemrograman Java, Visual Basic, atau PHP.

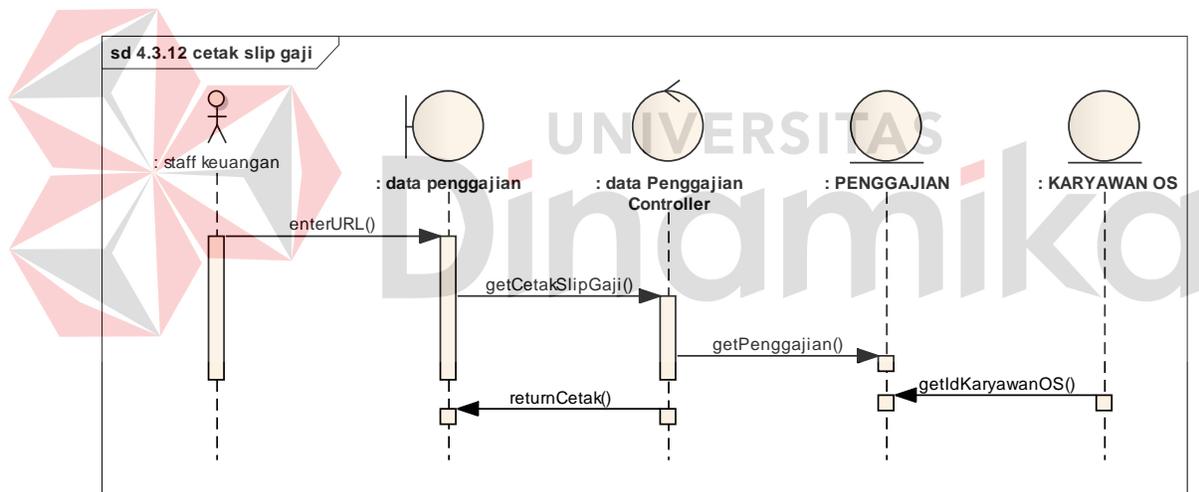
Tabel 3.1 Contoh *Flow of events*

<b>Nama Use Case</b>	[Ditulisakan nama <i>use case</i> ].
Kebutuhan terkait	A1 : [Menjelaskan kebutuhan terkait]
Tujuan	[Menjelaskan tujuan dari pembuatan <i>flow of event</i> ini].
Prasyarat	[Menjelaskan prasyarat yang dilakukan untuk menjalankan <i>flow of event</i> ini].
Kondisi akhir sukses	[Menjelaskan kondisi akhir apabila <i>flow of event</i> ini sukses dilaksanakan].
Kondisi akhir gagal	[Menjelaskan kondisi akhir apabila <i>flow of event</i> ini gagal dijalankan].
Aktor utama	[Menjelaskan siapa yang menjadi aktor utama pada <i>flow of event</i> ini].
Aktor sekunder	[Menjelaskan siapa yang menjadi aktor selain aktor utama pada <i>flow of event</i> ini].
Pemicu	[Menjelaskan pemicu pada flow of event].
Alur utama	Aksi 1 [Menjelaskan aksi langkah pertama] 2 [Menjelaskan aksi langkah kedua] 3 dst.

Nama Use Case	[Ditulisikan nama <i>use case</i> ].
Alur perluasan	Aksi 1 [Menjelaskan aksi langkah pertama] 2 [Menjelaskan aksi langkah kedua] 3 dst.

### 3.5.3 Sequence Diagram

Diagram sekuensial (*sequence diagram*) adalah diagram yang digunakan untuk menunjukkan alur (*flows*) fungsionalitas yang melalui sebuah *use case* yang disusun dalam urutan waktu (Sholih, 2010:23). Setiap diagram sekuensial merepresentasikan satu *flow* dari beberapa *flow* di dalam *use case*.

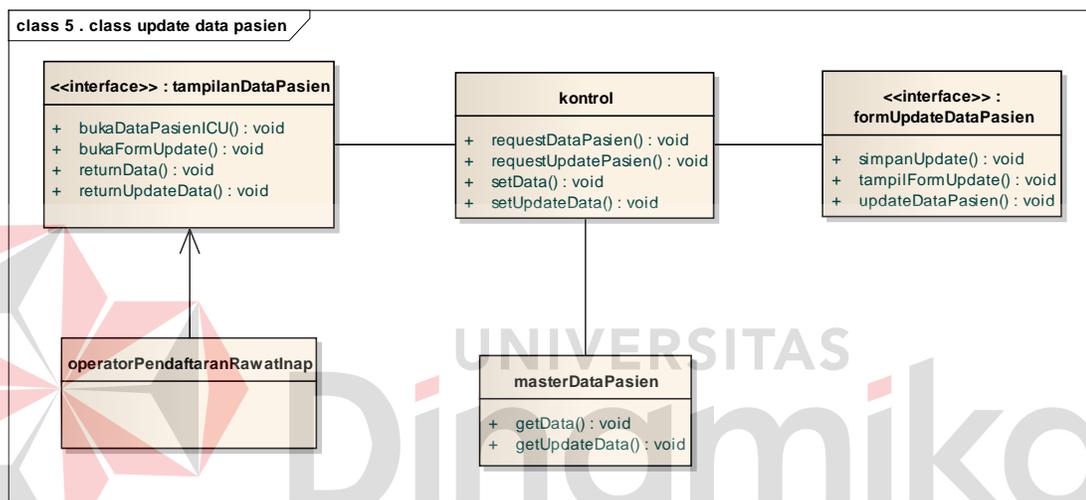


Gambar 3.6 Contoh *Sequence Diagram*

### 3.5.4 Class Diagram

Diagram kelas menunjukkan interaksi antar kelas-kelas di dalam sistem. Kelas juga dapat dianggap sebagai *blueprints* dari objek-objek di dalam sistem. Sebuah kelas dibuat dalam bentuk bujur sangkar yang terbagi dalam tiga bagian yaitu :

1. Bagian pertama menunjukkan nama kelas.
2. Bagian kedua menunjukkan anggota kelas yang memuat informasi atau atribut.
3. Bagian ketiga menunjukkan operasi-operasi dari sebuah kelas. Operasi dari sebuah kelas adalah tingkah laku yang disediakan oleh kelas (Sholihq 2010:28).



Gambar 3.7 Contoh Class Diagram

### 3.5.5 Black Box Testing

*Black box testing* merupakan pengujian yang dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang diuji (Romeo, 2003). Pengujian ini disebut juga sebagai *behavioral testing*, *spesification-based testing*, *input/output testing* atau *functional testing*. Black box testing dilakukan pada tahap akhir dan berfokus pada domain informasi.

### 3.5.6 Entity Relational Diagram ( ERD )

*Entity relationship diagram* (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. *Attribute* yaitu uraian dari entitas dimana mereka dihubungkan atau dapat dikatakan sebagai *identifier* atau *descriptors* dari entitas.

Entitas digolongkan menjadi *independent* atau *dependent entity*. *Independent entity* adalah apa yang tidak bersandar pada yang lain sebagai identifikasi. Suatu *dependent entity* adalah apa yang bersandar pada yang lain sebagai identifikasi.

Selain digolongkan menjadi *independent* atau *dependent entity*, terdapat jenis- jenis entitas khusus yaitu:

1. *Associative Entity*

*Associative Entity* (juga dikenal sebagai *intersection entity*) adalah entitas yang digunakan oleh rekanan dua entitas atau lebih untuk menyatukan suatu hubungan banyak - ke - banyak (*Many to Many*)

2. *Subtypes Entity*

*Subtypes Entity* digunakan di dalam hierarki generalisasi (*generalization hierarchies*) untuk menyajikan suatu subset kejadian dari entitas orangtua, yang disebut *supertype*, tetapi yang memiliki atribut atau hubungan yang berlaku hanya untuk *subset*.

Menurut Marlinda (2004: 28), *attribute* sebagai kolom di sebuah relasi mempunyai macam-macam jenis *attribute* yaitu :

a. *Key Attribute*

*Attribute* ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya entity mahasiswa yang *attribute*-nya NIM.



Gambar 3.8 *Key Attribute*

b. *Particial key Attribute*

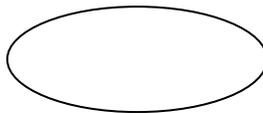
Adalah *Attribute* yang tidak menjadi atau merupakan anggota dari *Key Primer*. Misalnya antara Cabang (toko) dan kode cabang.



Gambar 3.9 *Particial Key Attribute*

c. *Single Vallue Attribute*

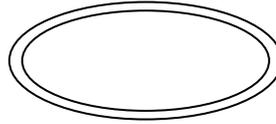
*Attribute* yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya Umur (Tanggal lahir).



Gambar 3.10 *Single Value Attribute*

d. *Multi Value Attribute*

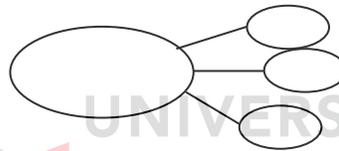
*Attribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya pendidikan (SD, SMP, SMA).



Gambar 3.11 *Multi Value Attribute*

e. *Composite Attribute*

*Attribute* yang memiliki dua harga, misalnya nama besar (nama kerja) dan nama kecil (nama asli)



Gambar 3.12 *Composite Attribute*

f. *Derived Attribute*

*Attribute* yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari table *Attribute* atau table lain yang berhubungan.



Gambar 3.13 *Derived Attribute*

Model *Entity - Relationship* (ER) mula-mula diusulkan oleh Peter pada tahun 1976 sebagai cara untuk mempersatukan pandangan basis data jaringan dan relasional. Langkah sederhana dari model ER adalah model data konseptual yang memandang dunia nyata sebagai kesatuan (*entities*) dan hubungan (*relationship*).

Komponen dasar model merupakan diagram *entity-relationship* yang digunakan untuk menyajikan objek data secara *visual*. *Entity Relationship Diagram* mengilustrasikan struktur logis dari basis data yang mempunyai metodologi sebagai berikut:

Tabel 3.2 Ilustrasi Pembuatan ERD

<b>Proses</b>	<b>Keterangan</b>
1. Menentukan Entitas	Menentukan peran, kejadian, lokasi, hal nyata, dan konsep dimana pengguna akan menyimpan data.
2. Menentukan Relasi	Tentukan hubungan antara pasangan entitas menggunakan matriks relasi.
3. Gambar ERD Sementara	Entitas digambarkan dengan kotak dan relasi dengan garis yang menghubungkan entitas.
4. Isi Kardinalitas	Tentukan jumlah kejadian dari satu entitas untuk sebuah kejadian pada entitas yang berhubungan.
5. Tentukan Kunci Utama	Tentukan atribut yang mengidentifikasi satu dan hanya satu kejadian pada masing-masing entitas.
6. Gambar ERD berdasar Kunci	Hilangkan relasi <i>Many-to-Many</i> dan masukkan <i>primary</i> dan kunci tamu pada masing-masing entitas.
7. Menentukan Atribut	Tuliskan <i>field-field</i> yang diperlukan oleh sistem.
8. Pemetaan Atribut	Pasangkan atribut dengan satu entitas yang sesuai pada masing-masing atribut.
9. Gambar ERD dengan Atribut	Aturlah ERD dari langkah 6 dengan menambahkan entitas atau relasi yang ditemukan pada langkah 8.
10. Periksa Hasil	Apakah ERD sudah menggambar sistem yang akan dibangun.

*Entity Relationship Diagram* ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh

perandang *database*. Untuk itu, *entity relationship diagram* dibagi menjadi dua jenis model, yaitu:

1. *Conceptual Data model*

*Conceptual Data model* (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

2. *Physical Data Model*

*Physical Data Model* (PDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara fisik.

### 3.6 Teori Yang Terkait Dengan Penyelesaian Masalah

#### 3.6.1 Microsoft Visual C# .NET

*Microsoft Visual C# .NET* adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem *.NET Framework*, dengan menggunakan bahasa *C#*. Dengan menggunakan alat ini, para pembuat program dapat membangun aplikasi *Windows Forms*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual Basic*, atau *visual F#*) atau juga dapat diperoleh secara terpadu dalam *Microsoft visual Studio .NET* (Yuswanto, 2007).

Bahasa *Visual C# .NET* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsot Visual C#* versi sebelumnya yang dimplementasikan diatas *.NET Framework*. Peluncurannya mengundang kontrovensi, mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft*, dan versi baru ini tidak kompatibel dengan versi terdahulu.

### 3.6.2 SQL Server 2008

*Microsoft SQL Server* adalah sebuah sistem manajemen basis data relasional (RDBMS) produk *Microsoft*. Bahasa kueri utamanya adalah *transact-SQL* yang merupakan implementasi dari *SQL standar ANSI/ISO* yang digunakan oleh *Microsoft* dan *Sybase*. *SQL (Structured Query Language)* adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional (Yuswanto, 2007).

Umumnya *SQL Server* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar. Penulis menggunakan *SQL Server 2008* untuk merancang database yang digunakan pada sistem.

### 3.6.3 Crystal Report

Merupakan *software* yang digunakan untuk pembuatan laporan. Dengan cara mengoneksi nama tabel yang akan dibuatkan laporannya. Setelah tampilan data ada maka klik dan *drag* semua *field* yang ada sesuai dengan tampilan yang diinginkan. Biasanya *crystal report* adalah komponen dari *C# .NET*.

## BAB IV

### METODE PENELITIAN

Metode penelitian digunakan untuk metode dalam pengembangan aplikasi. Metode yang akan dijelaskan pada bab ini adalah metode *System Development Life Cycle* (SDLC) menggunakan model *waterfall programming method*.

#### 4.1 *System Development Life Cycle* (SDLC)

Menurut Carol (2001), *System Development Life Cycle* merupakan siklus hidup pengembangan sistem. Dalam rekayasa sistem dan rekayasa perangkat lunak, SDLC berupa suatu proses pembuatan dan perubahan aplikasi serta model dan metode yang digunakan untuk mengembangkan aplikasi - aplikasi tersebut.

Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metode pengembangan perangkat lunak. Metode-metode ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan aplikasi, yaitu proses pengembangan perangkat lunak.

Pengembangan SDLC adalah proses yang digunakan oleh analis sistem untuk mengembangkan aplikasi, termasuk persyaratan, validasi, pelatihan, dan pengguna (*stakeholder*) kepemilikan. Setiap SDLC harus menghasilkan aplikasi berkualitas tinggi yang memenuhi atau melampaui harapan pelanggan, mencapai penyelesaian dalam waktu dan perkiraan biaya, bekerja secara efektif dan efisien di saat ini dan

direncanakan Teknologi Informasi infrastruktur, dan murah untuk mempertahankan dan biaya efektif.

#### **4.1.1 Fungsi *System Development Life Cycle* (SDLC)**

Untuk menggambarkan tahapan-tahapan utama dan langkah-langkah dari setiap tahapan yang secara garis besar terbagi dalam fase-fase utama dalam SDLC, yaitu :

1. Perencanaan : Mengapa mengembangkan aplikasi?
2. Analisis : Siapa, apa, kapan dan dimana aplikasi diterapkan?
3. Perancangan : Bagaimana kerja aplikasi?
4. Implementasi : Bagaimana aplikasi dipasang?

#### **4.2 *Waterfall Programming***

*Waterfall Development Method* adalah suatu cara pengembangan *software* yang fase – fasenya berurutan. Sebuah fase tidak bisa dikerjakan sebelum fase sebelumnya telah selesai dikerjakan.

a) Kelebihan dari *Waterfall Development Method* adalah :

1. Proses pengidentifikasian aplikasi memerlukan waktu yang lama sebelum fase programming dimulai.
2. Meminimalisasi perubahan aplikasi pada saat proses pengembangan perangkat lunak.

b) Kekurangan dari *Waterfall Development Method* adalah :

1. Fase perancangan / design harus dilakukan pada paper yang khusus sebelum fase *programming* dimulai.
2. Terjadi selisih waktu yang cukup lama antara pengajuan sistem dan pembaharuan sistem.

### 4.3 Model Pengembangan

Model pengembangan yang digunakan pada rancang bangun ini adalah menggunakan model *waterfall*. Model *waterfall* merupakan salah satu model dalam SDLC. Menurut Kassem (2011), tahap-tahap dari model *waterfall* adalah sebagai berikut:

#### 1. Analisis

Tahap analisis yang akan dilakukan meliputi identifikasi permasalahan, prosedur operasional, dan mendefinisikan kebutuhan fungsional dan nonfungsional yang terkait dalam pembuatan perangkat lunak. Pada tahap ini juga mendefinisikan modul-modul yang ada pada perangkat lunak yang akan dibuat.

#### 2. Desain

Tahap ini akan membuat desain arsitektural aplikasi mulai dari desain aplikasi, database, dan antarmuka pengguna.

#### 3. Implementasi

Tahap implementasi ini mencakup pembuatan aplikasi dan pendokumentasian pada tiap proses. Pembuatan aplikasi akan dilakukan per modul.

#### 4. Uji coba & integrasi

Pada tahap ini merupakan tahap uji coba pada aplikasi yang telah dikembangkan. Uji coba ini dilakukan pada masing-masing modul. Tahap ini juga akan melakukan integrasi antar modul sehingga setiap modul dapat berjalan dengan baik dan terintegrasi.

#### 5. Instalasi

Setelah uji coba tiap modul berhasil dan masing-masing modul dapat terintegrasi, maka akan dilakukan instalasi aplikasi pada perusahaan.

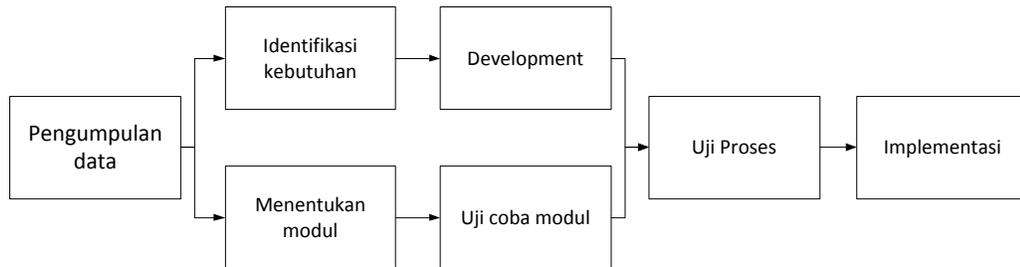
### 4.4 Prosedur Penelitian

Secara garis besar, langkah-langkah yang akan dilakukan dalam pembuatan perangkat lunak ini adalah sebagai berikut :

1. Melakukan pengumpulan data dengan wawancara, observasi, dan studi literatur untuk mendukung pembuatan perangkat lunak.
2. Melakukan identifikasi kebutuhan fungsional dan nonfungsional dari aplikasi penggajian karyawan *outsourcing* untuk dapat menentukan modul-modul yang sudah ditentukan.
3. Melakukan perancangan aplikasi penggajian karyawan *outsourcing*, membuat alur sistem, dan disain *database* dari modul-modul yang telah ditentukan.
4. Melakukan pembuatan aplikasi penggajian karyawan *outsourcing*, melakukan uji coba pada tiap modul.
5. Melakukan uji coba aplikasi pada sisi penugasan (*workload*) pada tiap pengguna aplikasi untuk menguji coba aplikasi penggajian karyawan *outsourcing*. Hasil uji

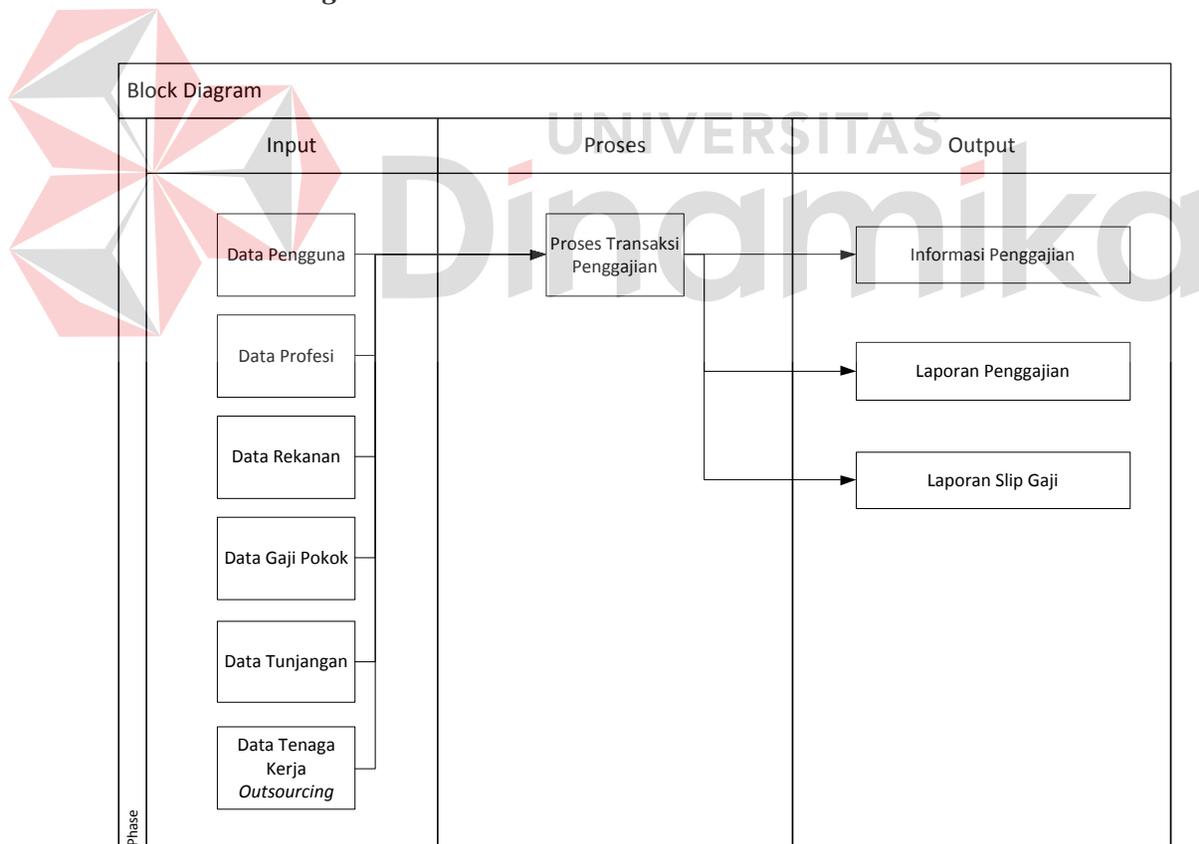
coba ini akan dianalisis untuk mengetahui apakah aplikasi yang dibuat telah sesuai dengan yang diharapkan.

6. Melakukan implementasi dan membuat laporan hasil penelitian.



Gambar 4.1 Kerangka prosedur penelitian

#### 4.4.1 Block Diagram



Gambar 4.2 Block Diagram Aplikasi Penggajian Karyawan Outsourcing

Gambar 4.2 diatas merupakan gambar *block diagram* aplikasi penggajian karyawan *outsourcing* pada PT Bangun Persada Transportation (BPT). *Input* yang dibutuhkan untuk diproses antara lain data pengguna, data profesi, data rekanan, data gaji pokok, data tunjangan, dan data tenaga kerja *outsourcing*. Pada *block* proses terdapat satu proses pengolahan data yaitu proses transaksi penggajian. Pada *block output* hasil informasi dan laporan yang ditampilkan adalah sebagai berikut :

#### 1. Informasi Penggajian

Informasi yang ditampilkan yaitu informasi dari gaji yang didapatkan oleh karyawan *outsourcing* setelah dilakukan perhitungan gaji dan potongannya.

#### 2. Laporan Penggajian

Laporan penggajian disini adalah laporan tanda terima gaji.

#### 3. Laporan Slip Gaji

Laporan slip gaji ini adalah slip gaji yang akan diberikan kepada karyawan *outsourcing* setelah karyawan *outsourcing* menerima gaji.

### 4.5 Analisa Aplikasi

Analisa aplikasi adalah langkah pertama untuk membuat suatu aplikasi baru. Langkah awal yang dilakukan adalah melakukan wawancara, dengan tujuan untuk mendapatkan informasi tentang mekanisme pendataan pegawai, data absensi, dan data perhitungan gaji pegawai. Selanjutnya dilakukan analisa terhadap permasalahan yang ada pada PT Bangun Persada Transportation (BPT), khususnya mengenai penggajian.

Berdasarkan hasil wawancara dan analisis proses pada PT BPT, selama ini perhitungan gaji untuk karyawan *outsourcing* menggunakan aplikasi perkantoran berupa *Microsoft Office Excel*. Pada perhitungan penggajian tersebut ditemukan hal-hal yang perlu dibenahi, yaitu sering kali mengalami kesalahan dalam melakukan pengecekan angsuran, peminjaman, perhitungan tunjangan, dan ditemukan kesalahan dalam proses perhitungan penggajian karena masih dilakukan secara manual.

Mengacu pada permasalahan yang telah disebutkan, PT BPT membutuhkan aplikasi penggajian yang terkomputerisasi agar lebih efektif sehingga dapat mengetahui berapa jumlah gaji yang seharusnya diberikan kepada karyawan *outsourcing*.

Oleh karena itu, dirancanglah sebuah aplikasi penggajian yang diharapkan dapat mengatasi permasalahan tersebut.

#### **4.5.1 Identifikasi Masalah**

Permasalahan-permasalahan yang dapat diidentifikasi pada proses penggajian ini adalah sebagai berikut:

1. Masih sering ditemukan kesalahan dalam pencatatan peminjaman, angsuran, tunjangan, dan penjumlahan penggajian.
2. Pembuatan laporan penggajian pegawai yang membutuhkan waktu lama karena menggunakan cara manual, yakni menggunakan aplikasi *Microsoft Office Excel*.

#### **4.5.2 Spesifikasi Aplikasi**

Pembuatan aplikasi ini diharapkan dapat:

1. Mengelola data penggajian.
2. Memberikan laporan kepegawaian, bentuk dalam aplikasi absensi dan penggajian, antara lain:
  - a. Laporan gaji pegawai tiap bulan untuk setiap rekanan.
  - b. Slip gaji / tanda terima gaji untuk masing-masing karyawan *outsourcing*.

### 4.5.3 Lingkungan Operasi

Untuk mengembangkan aplikasi sesuai dengan spesifikasi kebutuhan, dibutuhkan lingkungan operasi sebagai berikut:

- a. Sistem Operasi Windows

Sistem operasi yang disarankan adalah Windows 7 atau Windows 8.

- b. SQL Server 2008

SQL Server 2008 digunakan karena software database ini bisa digunakan untuk membuat sistem berbasis *client-server*.

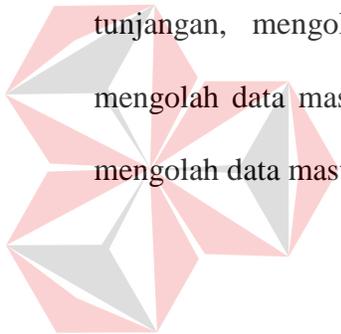
## 4.6 Undefined Modeling Language (UML)

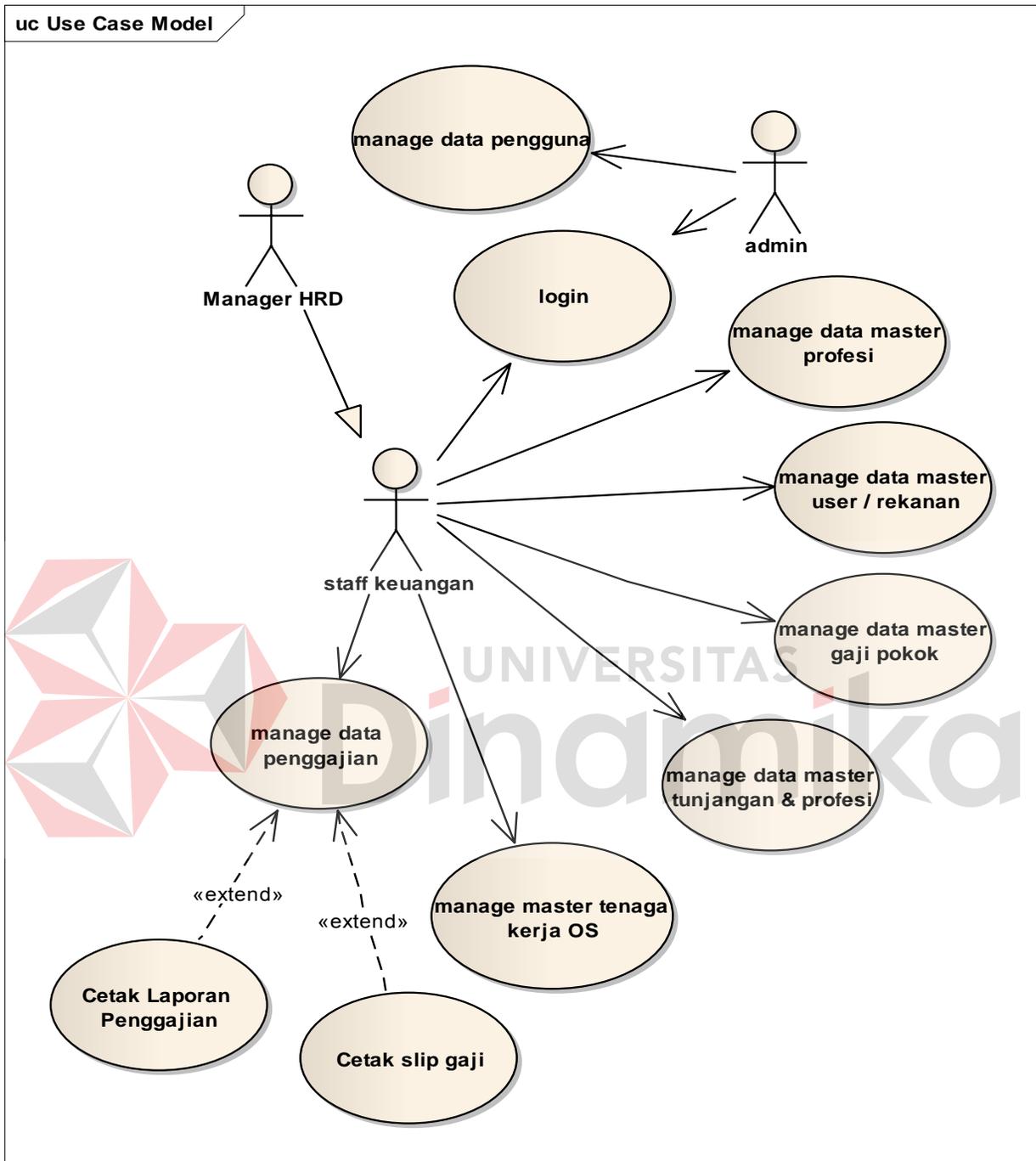
Dalam era teknologi informasi saat ini dibutuhkan analisa dan perancangan aplikasi pengelolaan penggajian yang diharapkan mampu meningkatkan keakuratan *output* yang dihasilkan.

Dokumen UML menggambarkan proses yang sudah ada menurut hasil analisis yang dibuat berdasarkan hasil *survey* pada perusahaan PT BPT.

#### 4.6.1 Use Case Diagram

Diagram *use case* ini akan menunjukkan interaksi antara pengguna dan aplikasi. Seperti yang terlihat pada gambar 4.1, terdapat tiga pengguna yang akan berinteraksi dengan aplikasi yaitu admin, staff keuangan, dan manager HRD. Ketiga pengguna tersebut merupakan orang yang berada di dalam atau terlibat langsung pada aplikasi penggajian. Admin memiliki tugas yaitu mengolah data pengguna, menampilkan data *level* pengguna, dan *login* ke aplikasi. Kemudian ada staff keuangan dan manager HRD yang memiliki tugas yang sama. Tugasnya terdiri dari *login* aplikasi, mengolah data master user/rekanan, mengolah data master nama tunjangan, mengolah data karyawan *outsourcing*, mengolah data penggajian, mengolah data master tunjangan & profesi, mengolah data master gaji pokok, dan mengolah data master profesi.





Gambar 4.3 Use Case aplikasi penggajian *outsourcing* PT BPT

#### 4.6.2 Flow of Events Use Case Login

Proses login dibutuhkan untuk mengidentifikasi pengunjung aplikasi apakah pengguna terdaftar dalam aplikasi atau tidak. Pengguna yang akan login dibagi menjadi tiga yaitu admin, staff keuangan, dan manager HRD.

Tabel 4.1 *Flow of Events Use Case Login*

Nama Use Case	Login	
Kebutuhan terkait	Proses untuk menentukan apakah pengguna dapat menggunakan aplikasi ini atau tidak dengan syarat pengguna terdaftar dan memiliki <i>username</i> dan <i>password</i> .	
Tujuan	Menangani proses login pengunjung aplikasi.	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Pengguna dapat masuk ke halaman utama dari aplikasi yang telah disesuaikan dengan level pengguna.	
Kondisi akhir gagal	Apabila login gagal maka pengguna harus meng- <i>input</i> -kan kembali <i>username</i> dan <i>password</i> dengan benar.	
Aktor utama	Pengguna aplikasi.	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman login.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Aktor masuk ke halaman login	1. Aplikasi menampilkan halaman login.
	2. Input <i>username</i>	2. Menampilkan <i>username</i>
	3. Input <i>password</i>	3. Menampilkan <i>password</i>
	4. Klik “ <i>Login</i> ”	4. Mengirim data ke <i>database</i> kemudian mengecek kebenaran <i>username</i> dan <i>password</i>
		5. Menampilkan halaman utama aplikasi. <i>Use case</i> selesai.
Alur Perluasan	4.1 Jika poin 4 pada alur utama aplikasi gagal melakukan login maka akan kembali ke poin 1 pada alur utama aplikasi.	

### 4.6.3 Flow of Event Use Case Manage Data Pengguna

*Flow of event* ini mendeskripsikan cara menampilkan data pengguna yang hanya dapat dilakukan oleh admin. Admin dapat melihat pengguna-pengguna yang akan menggunakan aplikasi penggajian ini.

Tabel 4.2 *Flow of Event Use Case manage data pengguna*

Nama Use Case	Manage data pengguna	
Kebutuhan terkait	Menampilkan data-data pengguna yang akan menjadi pengguna dari aplikasi penggajian ini.	
Tujuan	Mengetahui data pengguna yang akan menggunakan aplikasi penggajian ini.	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Tampil data pengguna, <i>edit</i> , dan hapus data pengguna pada aplikasi sukses ditampilkan.	
Kondisi akhir gagal	Apabila data pengguna tidak tampil maka ada kesalahan pada <i>database</i> atau isi dari <i>database</i> kosong.	
Aktor utama	Admin	
Aktor sekunder	Tidak ada.	
Pemicu	Aktor membuka halaman master pengguna.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Admin membuka halaman master pengguna.	1. Aplikasi menampilkan halaman master pengguna.
	2. Admin menambah data di form tambah data pengguna.	2. Aplikasi menampilkan form yang telah terisi data pengguna baru.
	3. <i>Input</i> "Nama lengkap"	
	4. <i>Input</i> "Nama pengguna"	
	5. <i>Input</i> "Kata Sandi"	
	6. <i>Input</i> "Level Pengguna"	
	7. Admin menekan tombol "save".	7. Aplikasi mengirim data ke <i>server</i> kemudian disimpan.
	8. Admin mengubah data pengguna dengan menekan tombol <i>edit</i> .	8. Aplikasi membuka form <i>edit</i> data pengguna.
	9. Admin mulai <i>edit</i> data.	9. Menampilkan hasil <i>edit</i> sementara.
10. Setelah data diubah, admin menekan tombol <i>update</i> .	10. Aplikasi akan menyimpan perubahan data di <i>server</i> .	

<b>Nama Use Case</b>	<b>Manage data pengguna</b>	
	11. Admin menghapus data pengguna aplikasi dengan cara menekan tombol <i>delete</i> .	11. Aplikasi akan menghapus data pengguna di <i>server</i> . <i>Use case</i> selesai.
Alur perluasan	7.1 Jika di aplikasi masih ada <i>field</i> yang kosong, maka aplikasi akan mengeluarkan pesan ada <i>field</i> yang masih kosong dan kembali ke langkah 2 pada aplikasi.	

#### 4.6.4 Flow of Event Manage Data Master Profesi

*Flow of event* ini mendeskripsikan cara menampilkan data master profesi. Profesi yang dimaksud pada *flow of event* adalah profesi yang dimiliki oleh karyawan *outsourcing*. *Manage* data master profesi ini hanya bisa dilakukan oleh staf keuangan dan *manager* HRD. Staf keuangan dapat melihat, menambah, *edit*, dan menghapus data lokasi.

Tabel 4.3 *Flow of Event Use Case* manage data master profesi

<b>Nama Use Case</b>	<b>Manage Data Master Profesi</b>	
Kebutuhan terkait	Menampilkan data master profesi dari karyawan <i>outsourcing</i> .	
Tujuan	Mengetahui profesi-profesi yang dimiliki oleh karyawan <i>outsourcing</i> .	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Tampil data master profesi, <i>edit</i> , dan hapus data profesi pada aplikasi sukses ditampilkan.	
Kondisi akhir gagal	Apabila data master profesi tidak tampil maka ada kesalahan pada <i>database</i> atau isi dari <i>database</i> kosong.	
Aktor utama	Staff keuangan	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman master profesi.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Aktor membuka halaman master profesi.	1. Aplikasi menampilkan halaman master profesi.
	2. Aktor menambah data di form tambah data profesi.	2. Aplikasi menampilkan form yang telah terisi data profesi yang baru.
	3. Aktor menekan tombol	3. Aplikasi mengirim data ke

<b>Nama Use Case</b>	<b>Manage Data Master Profesi</b>	
	“save”.	<i>server</i> kemudian disimpan.
	4. Aktor mengubah data dengan menekan tombol <i>edit</i> .	4. Aplikasi membuka form <i>edit</i> data profesi.
	5. Aktor mulai <i>edit</i> data.	5. Menampilkan hasil <i>edit</i> sementara.
	6. Setelah data diubah, admin menekan tombol <i>update</i> .	6. Aplikasi akan menyimpan perubahan data di <i>server</i> .
	7. Aktor menghapus data profesi dengan cara menekan tombol <i>delete</i> .	7. Aplikasi akan menghapus data profesi di <i>server</i> . <i>Use case</i> selesai.
Alur perluasan	3.1 Jika di aplikasi masih ada <i>field</i> yang kosong, maka aplikasi akan mengeluarkan pesan ada <i>field</i> yang masih kosong dan kembali ke langkah 2 pada aplikasi.	

#### 4.6.5 Flow of Event Manage Data Master Rekanan

*Flow of event* ini mendeskripsikan cara menampilkan data master rekanan. *Manage* data master rekanan ini hanya bisa dilakukan oleh staf keuangan dan *manager* HRD. Staf keuangan dapat melihat, menambah, *edit*, dan menghapus data rekanan.

Tabel 4.4 *Flow of Event Use Case* manage data master rekanan

<b>Nama Use Case</b>	<b>Manage Data Master Rekanan</b>	
Kebutuhan terkait	Menampilkan data rekanan dari karyawan <i>outsourcing</i> .	
Tujuan	Mengetahui rekanan yang ditempati oleh karyawan <i>outsourcing</i> .	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Tampil data master rekanan, <i>edit</i> , dan hapus data rekanan pada aplikasi sukses ditampilkan.	
Kondisi akhir gagal	Apabila data master rekanan tidak tampil maka ada kesalahan pada <i>database</i> atau isi dari <i>database</i> kosong.	
Aktor utama	Staff keuangan	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman master profesi.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>

<b>Nama Use Case</b>	<b>Manage Data Master Rekanan</b>	
	1. Aktor membuka halaman master user/rekanan.	1. Aplikasi menampilkan halaman master rekanan.
	2. Aktor menambah data di form tambah data rekanan.	2. Aplikasi menampilkan form yang telah terisi data rekanan yang baru.
	3. Input “Nama Rekanan”	
	4. Input “Alamat Rekanan”	
	5. Input “No Telp”	
	6. Aktor menekan tombol “save”.	6. Aplikasi mengirim data ke <i>server</i> kemudian disimpan.
	7. Aktor mengubah data dengan menekan tombol <i>edit</i> .	7. Aplikasi membuka form <i>edit</i> data rekanan.
	8. Aktor mulai <i>edit</i> data.	8. Menampilkan hasil <i>edit</i> sementara.
	9. Setelah data diubah, admin menekan tombol <i>update</i> .	9. Aplikasi akan menyimpan perubahan data di <i>server</i> .
	10. Aktor menghapus data profesi dengan cara menekan tombol <i>delete</i> .	10. Aplikasi akan menghapus data rekanan di <i>server</i> . <i>Use case</i> selesai.
Alur perluasan	6.1 Jika di aplikasi masih ada <i>field</i> yang kosong, maka aplikasi akan mengeluarkan pesan ada <i>field</i> yang masih kosong dan kembali ke langkah 2 pada aplikasi.	

#### 4.6.6 Flow of Event Manage Data Master Gaji Pokok

*Flow of event* ini mendeskripsikan cara menampilkan data master gaji pokok. *Manage data* master gaji pokok ini hanya bisa dilakukan oleh staf keuangan dan *manager* HRD. Staf keuangan dapat melihat, menambah, *edit*, dan menghapus data gaji pokok.

Tabel 4.5 *Flow of Event Use Case* manage data master gaji pokok

<b>Nama Use Case</b>	<b>Manage Data Master Gaji Pokok</b>
Kebutuhan terkait	Menampilkan data gaji pokok dari karyawan <i>outsourcing</i> .
Tujuan	Mengetahui gaji pokok yang dimiliki oleh karyawan <i>outsourcing</i> .

<b>Nama Use Case</b>	<b>Manage Data Master Gaji Pokok</b>	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Tampil data master gaji pokok, <i>edit</i> , dan hapus data gaji pokok pada aplikasi sukses ditampilkan.	
Kondisi akhir gagal	Apabila data master gaji pokok tidak tampil maka ada kesalahan pada <i>database</i> atau isi dari <i>database</i> kosong.	
Aktor utama	Staff keuangan	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman master gaji pokok.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Aktor membuka halaman master gaji pokok.	1. Aplikasi menampilkan halaman master gaji pokok.
	2. Aktor menambah data di form tambah data gaji pokok.	2. Aplikasi menampilkan form yang telah terisi data gaji pokok yang baru.
	3. Input “Profesi”	
	4. Input “Rekanan”	
	5. Input “Gaji Pokok”	
	6. Aktor menekan tombol “ <i>save</i> ”.	6. Aplikasi mengirim data ke <i>server</i> kemudian disimpan.
	7. Aktor mengubah data dengan menekan tombol <i>edit</i> .	7. Aplikasi membuka form <i>edit</i> data gaji pokok.
	8. Aktor mulai <i>edit</i> data.	8. Menampilkan hasil <i>edit</i> sementara.
	9. Setelah data diubah, admin menekan tombol <i>update</i> .	9. Aplikasi akan menyimpan perubahan data di <i>server</i> .
10. Aktor menghapus data gaji pokok dengan cara menekan tombol <i>delete</i> .	10. Aplikasi akan menghapus data gaji pokok di <i>server</i> . <i>Use case</i> selesai.	
Alur perluasan	6.1 Jika di aplikasi masih ada <i>field</i> yang kosong, maka aplikasi akan mengeluarkan pesan ada <i>field</i> yang masih kosong dan kembali ke langkah 2 pada aplikasi.	

#### 4.6.7 Flow of Event Manage Data Master Tunjangan & Profesi

*Flow of event* ini mendeskripsikan cara menampilkan data master tunjangan & profesi. *Manage data master* ini hanya bisa dilakukan oleh staf keuangan dan

*manager* HRD. Staf keuangan dapat melihat, menambah, *edit*, dan menghapus data tunjangan & profesi.

Tabel 4.6 *Flow of Event Use Case manage data master tunjangan & profesi*

Nama Use Case	Manage Data Master Tunjangan & Profesi	
Kebutuhan terkait	Menampilkan data tunjangan & profesi dari karyawan <i>outsourcing</i> .	
Tujuan	Mengetahui macam-macam tunjangan yang dilihat dari profesi yang dimiliki oleh karyawan <i>outsourcing</i> .	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Tampil data master tunjangan & profesi, <i>edit</i> , dan hapus data pada aplikasi sukses ditampilkan.	
Kondisi akhir gagal	Apabila data master tunjangan & profesi tidak tampil maka ada kesalahan pada <i>database</i> atau isi dari <i>database</i> kosong.	
Aktor utama	Staff keuangan	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman master tunjangan & profesi.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Aktor membuka halaman master tunjangan & profesi.	1. Aplikasi menampilkan halaman master tunjangan & profesi.
	2. Aktor menambah data di form tambah data tunjangan & profesi.	2. aplikasi menampilkan form yang telah terisi data yang baru.
	3. Input "Nama Tunjangan"	
	4. Input "Nama Rekanan"	
	5. Input "Profesi"	
	6. Input "Jumlah Tunjangan"	
	7. Input "Status kehadiran berdasarkan tunjangan"	
	8. Aktor menekan tombol "save".	8. Aplikasi mengirim data ke <i>server</i> kemudian disimpan.
	9. Aktor mengubah data dengan menekan tombol <i>edit</i> .	9. Aplikasi membuka form <i>edit</i> data tunjangan & profesi.
	10. Aktor mulai <i>edit</i> data.	10. Menampilkan hasil <i>edit</i> sementara.
	11. Setelah data diubah, admin menekan tombol <i>update</i> .	11. Aplikasi akan menyimpan perubahan data di <i>server</i> .
	12. Aktor menghapus data	12. Aplikasi akan

Nama Use Case	Manage Data Master Tunjangan & Profesi	
	tunjangan & profesi dengan cara menekan tombol <i>delete</i> .	menghapus data tunjangan & profesi di <i>server</i> . <i>Use case</i> selesai.
Alur perluasan	8.1 Jika di aplikasi masih ada <i>field</i> yang kosong, maka aplikasi akan mengeluarkan pesan ada <i>field</i> yang masih kosong dan kembali ke langkah 2 pada aplikasi.	

#### 4.6.8 Flow of Event Manage Master Karyawan Outsourcing

*Flow of event* ini mendeskripsikan cara menampilkan data master dari karyawan *outsourcing* (OS). *Manage* data master ini hanya bisa dilakukan oleh staf keuangan dan *manager* HRD. Staf keuangan dapat melihat, menambah, *edit*, dan menghapus data karyawan *outsourcing*.

Tabel 4.7 Flow of Event Use Case manage data master karyawan OS

Nama Use Case	Manage Data Master Karyawan OS	
Kebutuhan terkait	Menampilkan data master dari karyawan OS.	
Tujuan	Mengengola data-data karyawan OS yang bekerja di berbagai rekanan.	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Tampil data master karyawan OS, <i>edit</i> , dan hapus data pada aplikasi sukses ditampilkan.	
Kondisi akhir gagal	Apabila data master karyawan OS tidak tampil maka ada kesalahan pada <i>database</i> atau isi dari <i>database</i> kosong.	
Aktor utama	Staff keuangan	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman master karyawan OS.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Aktor membuka halaman master karyawan OS.	1. Aplikasi menampilkan halaman master karyawan OS.
	2. Aktor menambah data di form tambah data master karyawan OS.	2. Aplikasi menampilkan form yang telah terisi data yang baru.
	3. Input "Nama Karyawan"	
	4. Input "Alamat"	
	5. Input "No telp"	

<b>Nama Use Case</b>	<b>Manage Data Master Karyawan OS</b>	
	6. Input “Tgl Lahir”	
	7. Input “Status”	
	8. Input “Rekanan”	
	9. Input “Profesi”	
	10. Input “Tgl Mulai kerja”	
	11. Input “Tgl Selesai Kerja”	
	12. Aktor menekan tombol “save”.	12. Aplikasi mengirim data ke <i>server</i> kemudian disimpan.
	13. Aktor mengubah data dengan menekan tombol <i>edit</i> .	13. Aplikasi membuka form <i>edit</i> data nama tunjangan.
	14. Aktor mulai <i>edit</i> data.	14. Menampilkan hasil <i>edit</i> sementara.
	15. Setelah data diubah, admin menekan tombol <i>update</i> .	15. Aplikasi akan menyimpan perubahan data di <i>server</i> .
	16. Aktor menghapus data master karyawan OS dengan cara menekan tombol <i>delete</i> .	16. Aplikasi akan menghapus data master karyawan OS di <i>server</i> . <i>Use case</i> selesai.
Alur perluasan	12.1 Jika di aplikasi masih ada <i>field</i> yang kosong, maka aplikasi akan mengeluarkan pesan ada <i>field</i> yang masih kosong dan kembali ke langkah 2 pada aplikasi.	

#### 4.6.9 Flow of Event Manage Data Penggajian

*Flow of event* ini mendeskripsikan cara menampilkan data transaksi penggajian. *Manage data master* ini hanya bisa dilakukan oleh staf keuangan dan *manager HRD*. Staf keuangan dapat melihat, menambah, *edit*, dan menghapus data karyawan *outsourcing*.

Tabel 4.8 *Flow of Event Use Case manage data penggajian*

<b>Nama Use Case</b>	<b>Manage Data Penggajian</b>
Kebutuhan terkait	Menampilkan data penggajian.
Tujuan	Mengengola data-data penggajian para karyawan <i>outsourcing</i> .

<b>Nama Use Case</b>	<b>Manage Data Penggajian</b>	
Prasyarat	Tidak ada	
Kondisi akhir sukses	Tampil data transaksi penggajian, <i>edit</i> , dan hapus data pada aplikasi sukses dieksekusi.	
Kondisi akhir gagal	Apabila data penggajian tidak tampil maka ada kesalahan pada <i>database</i> atau isi dari <i>database</i> kosong.	
Aktor utama	Staff keuangan	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman data penggajian.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Aktor membuka halaman data penggajian.	1. Aplikasi menampilkan halaman data penggajian.
	2. Aktor menambah data di form tambah data transaksi penggajian.	2. Aplikasi menampilkan form yang telah terisi data yang baru.
	3. Input jamsostek	
	4. Input jumlah pinjaman	
	5. Input total jumlah pinjaman	
	6. Input angsuran	
	7. Input sisa	
	8. Input kehadiran	
	9. Input tidak masuk	
	10. Input lembur	
	11. Input Paket Lembur	
	12. Input Penalti	
	13. Input tambahan gaji	
	14. Input keterangan tambahan gaji	
	15. Aktor menekan tombol “ <i>save</i> ”.	15. Aplikasi mengirim data ke <i>server</i> kemudian disimpan.
	16. Aktor mengubah data dengan menekan tombol <i>edit</i> .	16. Aplikasi membuka form <i>edit</i> data penggajian.
	17. Aktor mulai <i>edit</i> data.	17. Menampilkan hasil <i>edit</i> sementara.
	18. Setelah data diubah, admin menekan tombol <i>update</i> .	18. Aplikasi akan menyimpan perubahan data di <i>server</i> .
19. Aktor menghapus data transaksi dengan cara menekan tombol <i>delete</i> .	19. Aplikasi akan menghapus data penggajian di <i>server</i> . <i>Use case</i> selesai.	
Alur perluasan	15.1 Jika di Aplikasi masih ada <i>field</i> yang kosong, maka	

<b>Nama Use Case</b>	<b>Manage Data Penggajian</b>
	aplikasi akan mengeluarkan pesan ada <i>field</i> yang masih kosong dan kembali ke langkah 2 pada aplikasi.

#### 4.6.10 Flow of Event Cetak Slip Gaji

*Flow of event* ini mendeskripsikan cara mencetak slip gaji karyawan *outsourcing*. Cetak slip gaji ini hanya bisa dilakukan oleh staf keuangan dan *manager* HRD.

Tabel 4.9 *Flow of Event Use Case* cetak slip gaji

<b>Nama Use Case</b>	<b>Cetak Slip Gaji</b>	
Kebutuhan terkait	Slip gaji merupakan bukti fisik dari gaji yang akan diberikan pada karyawan <i>outsourcing</i> .	
Tujuan	Mencetak slip gaji para karyawan <i>outsourcing</i> .	
Prasyarat	Transaksi penggajian pada aplikasi telah dilaksanakan.	
Kondisi akhir sukses	Slip gaji sukses dicetak.	
Kondisi akhir gagal	Apabila tidak dapat cetak slip gaji, maka ada kesalahan pada aplikasi operasinya.	
Aktor utama	Staff keuangan	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman data penggajian.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Aktor membuka halaman data penggajian.	1. Aplikasi menampilkan halaman data penggajian.
	2. Aktor mencari data karyawan <i>outsourcing</i> yang slip gajinya akan dicetak.	2. Aplikasi menampilkan hasil pencarian karyawan <i>outsourcing</i> .
	3. Aktor menekan tombol “cetak slip”.	3. Aplikasi membuka file slip gaji dalam format PDF. <i>Use case</i> selesai.

#### 4.6.11 Flow of Event Cetak Laporan Penggajian

*Flow of event* ini mendeskripsikan cara mencetak laporan penggajian karyawan *outsourcing*. Cetak laporan penggajian ini hanya bisa dilakukan oleh staf keuangan dan *manager* HRD.

Tabel 4.10 *Flow of Event Use Case* cetak laporan penggajian

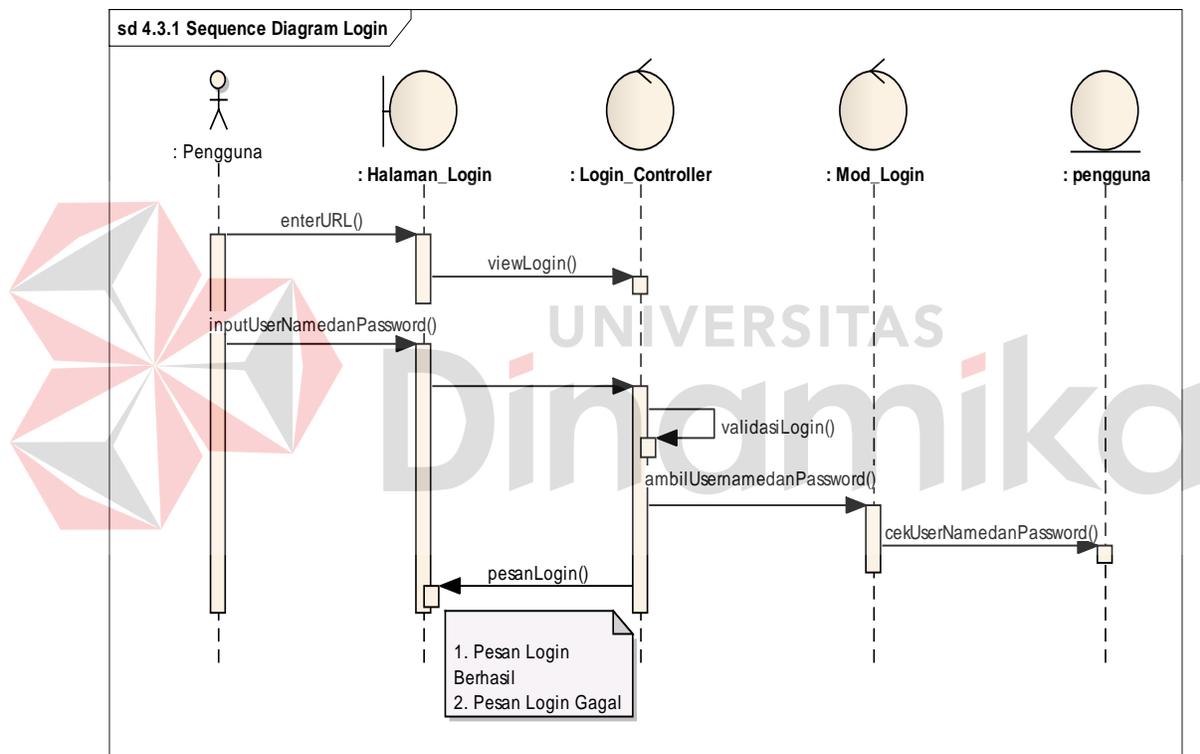
Nama Use Case	Cetak Laporan Penggajian	
Kebutuhan terkait	Laporan penggajian ini dicetak sebagai arsip kantor ataupun permintaan dari direktur.	
Tujuan	Mencetak laporan gaji para karyawan <i>outsourcing</i> .	
Prasyarat	Transaksi penggajian pada aplikasi telah dilaksanakan.	
Kondisi akhir sukses	Laporan penggajain sukses dicetak.	
Kondisi akhir gagal	Apabila tidak dapat cetak laporan penggajian, maka ada kesalahan pada aplikasi operasinya.	
Aktor utama	Staff keuangan	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman data penggajian.	
Alur utama	<b>Aktor</b>	<b>Aplikasi</b>
	1. Aktor membuka halaman data penggajian.	1. Aplikasi menampilkan halaman data penggajian.
	2. Aktor mencari data karyawan <i>outsourcing</i> yang laporan penggajiannya akan dicetak.	2. Aplikasi menampilkan hasil pencarian karyawan <i>outsourcing</i> .
	3. Aktor menekan tombol “cetak laporan”.	3. Aplikasi membuka file slip gaji dalam format *xlsx. <i>Use case</i> selesai.

## 4.7 Sequence Diagram

### 4.7.1 Sequence Diagram Login

Pengguna aplikasi ini akan melakukan *login* sebelum masuk ke sistem. Pengguna pada aplikasi penggajian ini memiliki tiga pengguna utama yaitu *manager* HRD, *staff*, dan admin. Pada awal pengguna membuka aplikasi ini akan dihadapkan

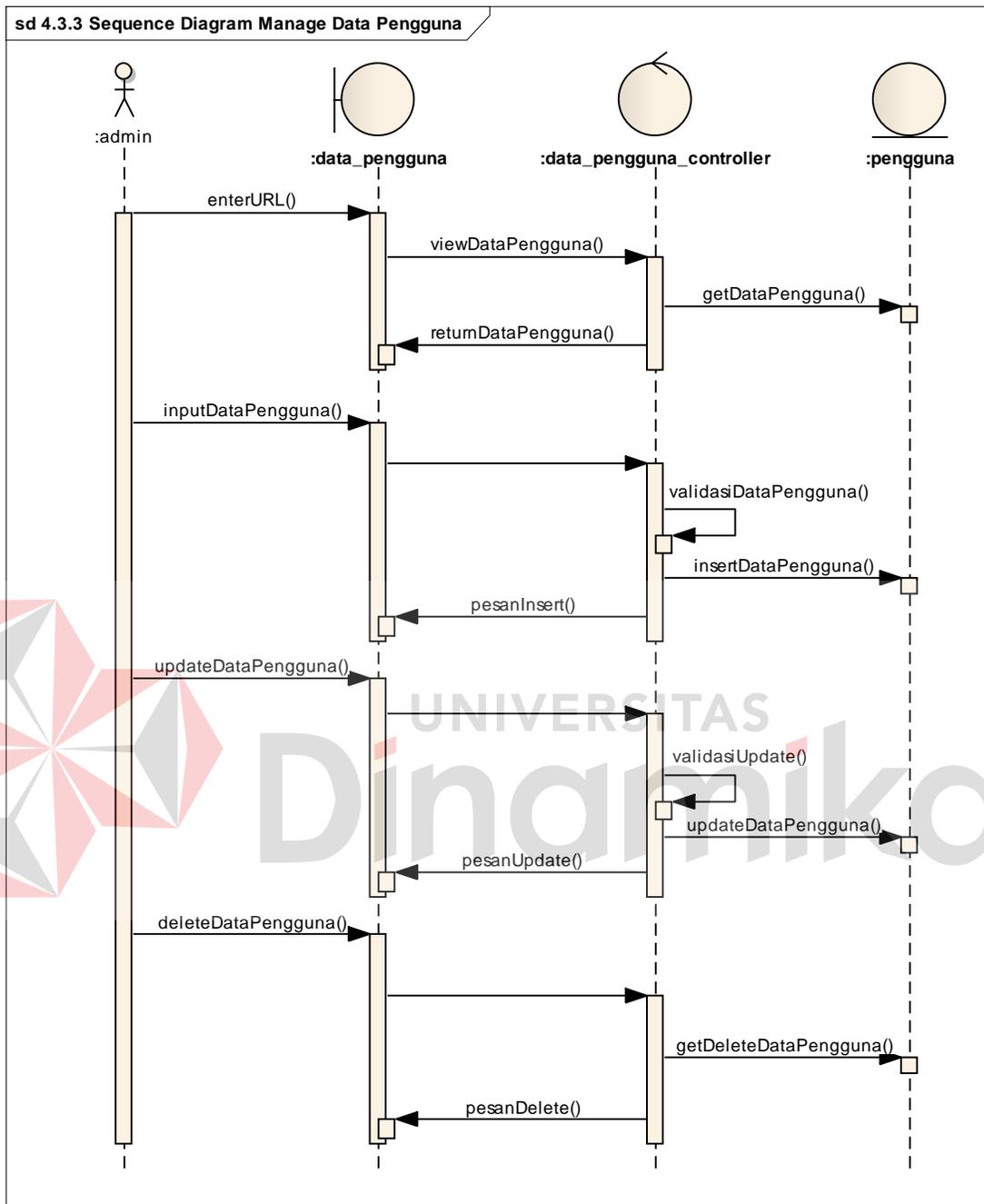
dengan *form login*. Pengguna akan memasukkan *username* dan *password* yang benar agar dapat masuk ke dalam halaman utama aplikasi. Setelah pengguna memasukkan *username* dan *password* sistem akan mengecek kebenarannya di dalam *database*. Setelah dicek maka sistem akan mengembalikan nilai ke dalam form. Jika benar maka sistem akan masuk ke halaman utama aplikasi. Jika salah maka sistem akan memberikan pesan kesalahan *username* atau *password*.



Gambar 4.4 Sequence Diagram Login

#### 4.7.2 Sequence Diagram Manage Data Pengguna

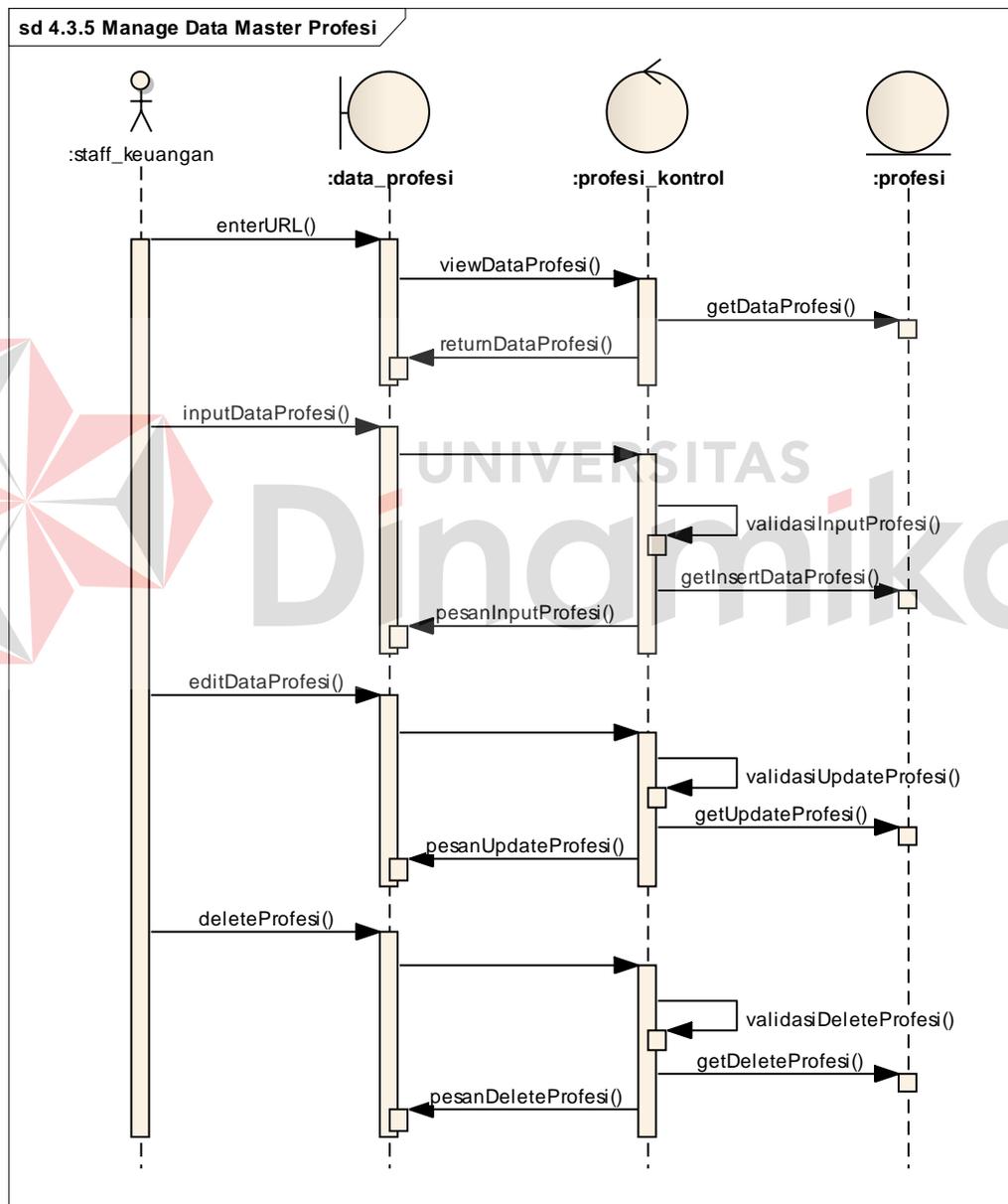
Data pengguna berada pada halaman data pengguna. Data pengguna ini memiliki beberapa atribut yang harus dimasukkan ke dalam *database*. Atribut-atribut tersebut akan terlihat pada bagian disain I/O.



Gambar 4.5 Sequence Diagram Manage Data Pengguna

### 4.7.3 Sequence Diagram Manage Data Master Profesi

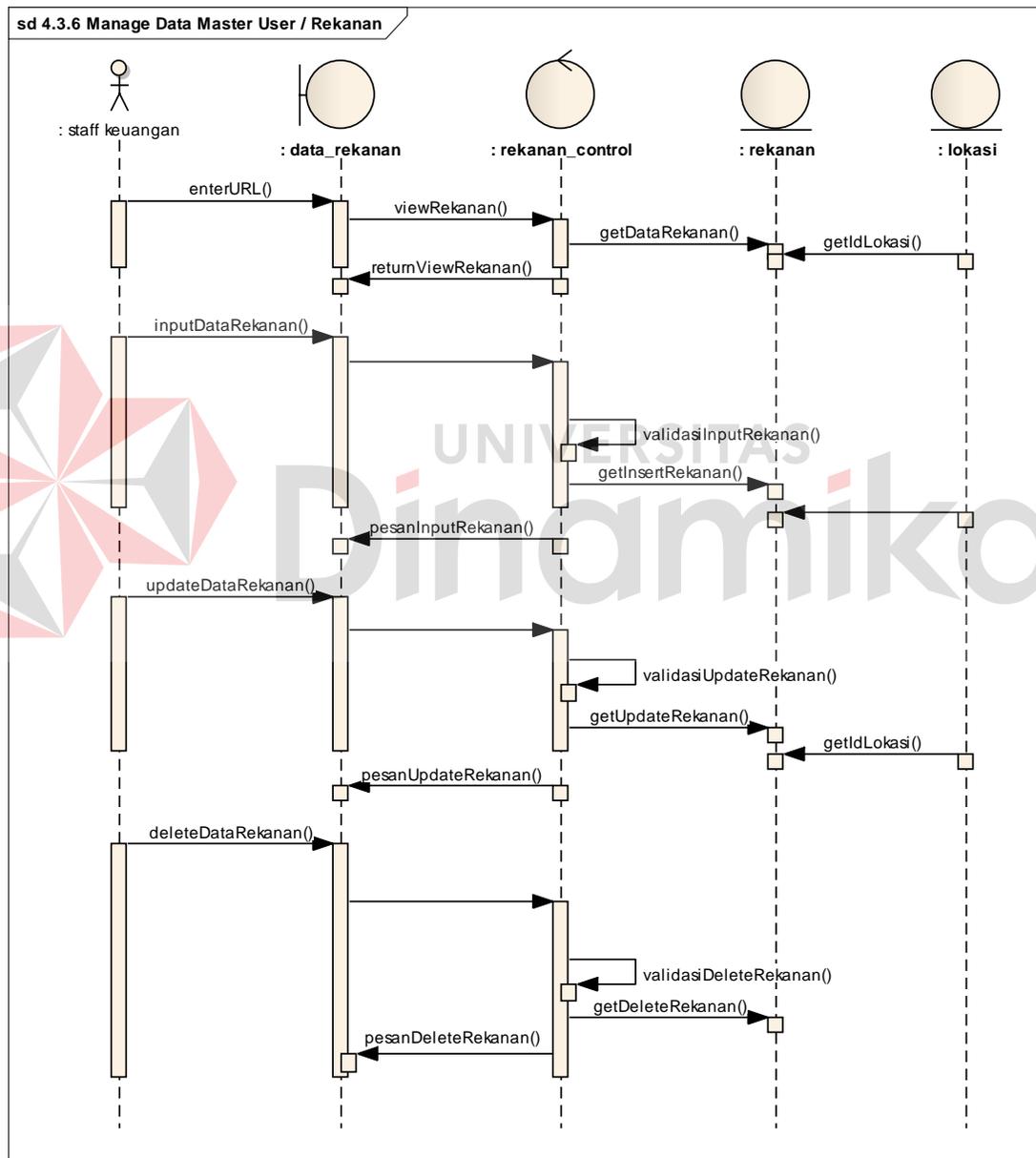
Data master profesi akan digunakan untuk menyimpan profesi-profesi karyawan *outsourcing*. Data master profesi dapat ditambahkan, diubah, dan dihapus seperti diagram dibawah ini.



Gambar 4.6 Sequence Diagram Manage Data Master Profesi

#### 4.7.4 Sequence Diagram Manage Data Master Rekanan

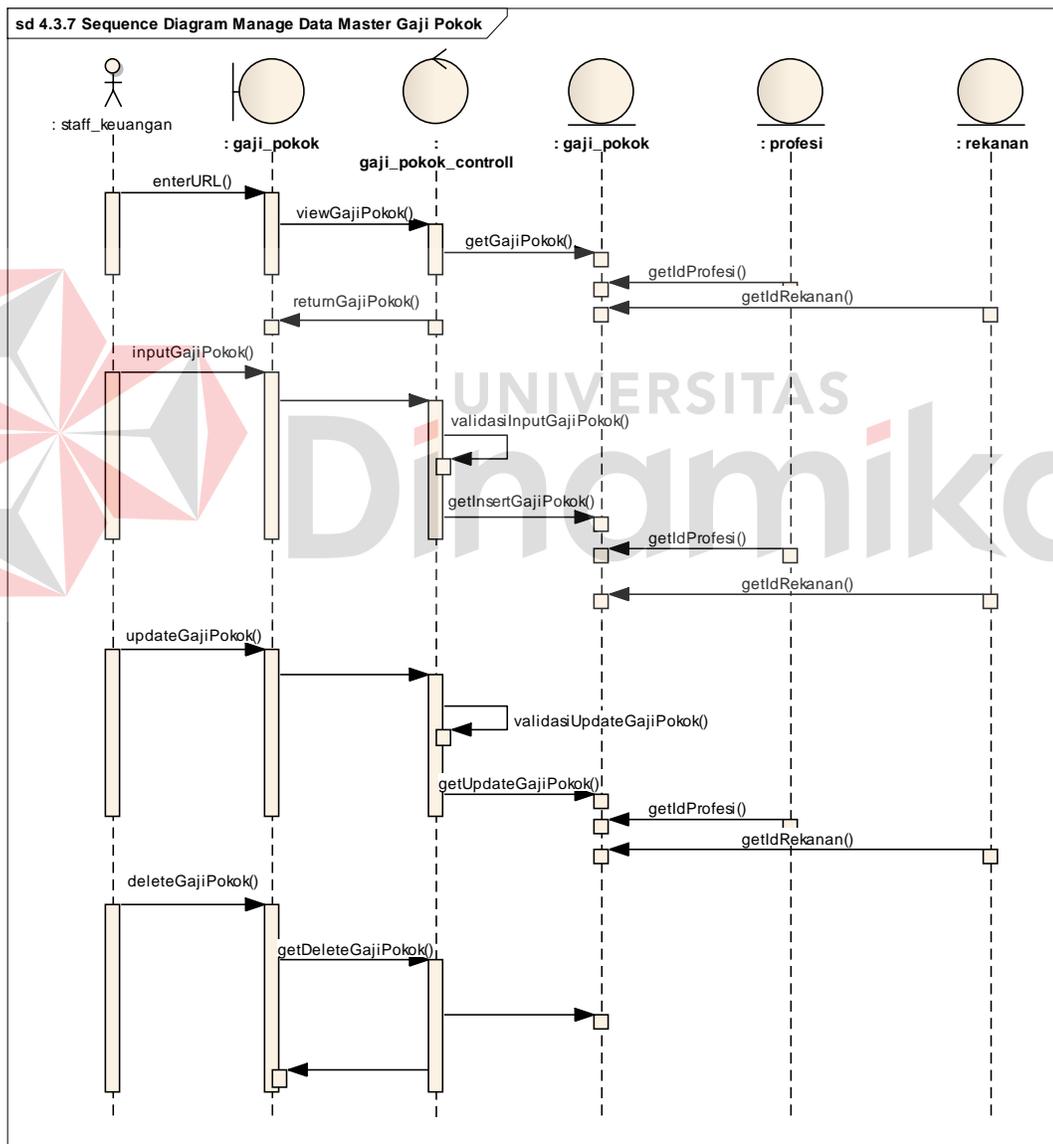
Data master rekanan digunakan untuk menyimpan data rekanan di dalam database. Data rekanan ini dapat ditambah, diubah, ataupun dihapus seperti diagram yang ada di bawah ini.



Gambar 4.7 Sequence Diagram Manage Data Rekanan

#### 4.7.5 Sequence Diagram Manage Data Master Gaji Pokok

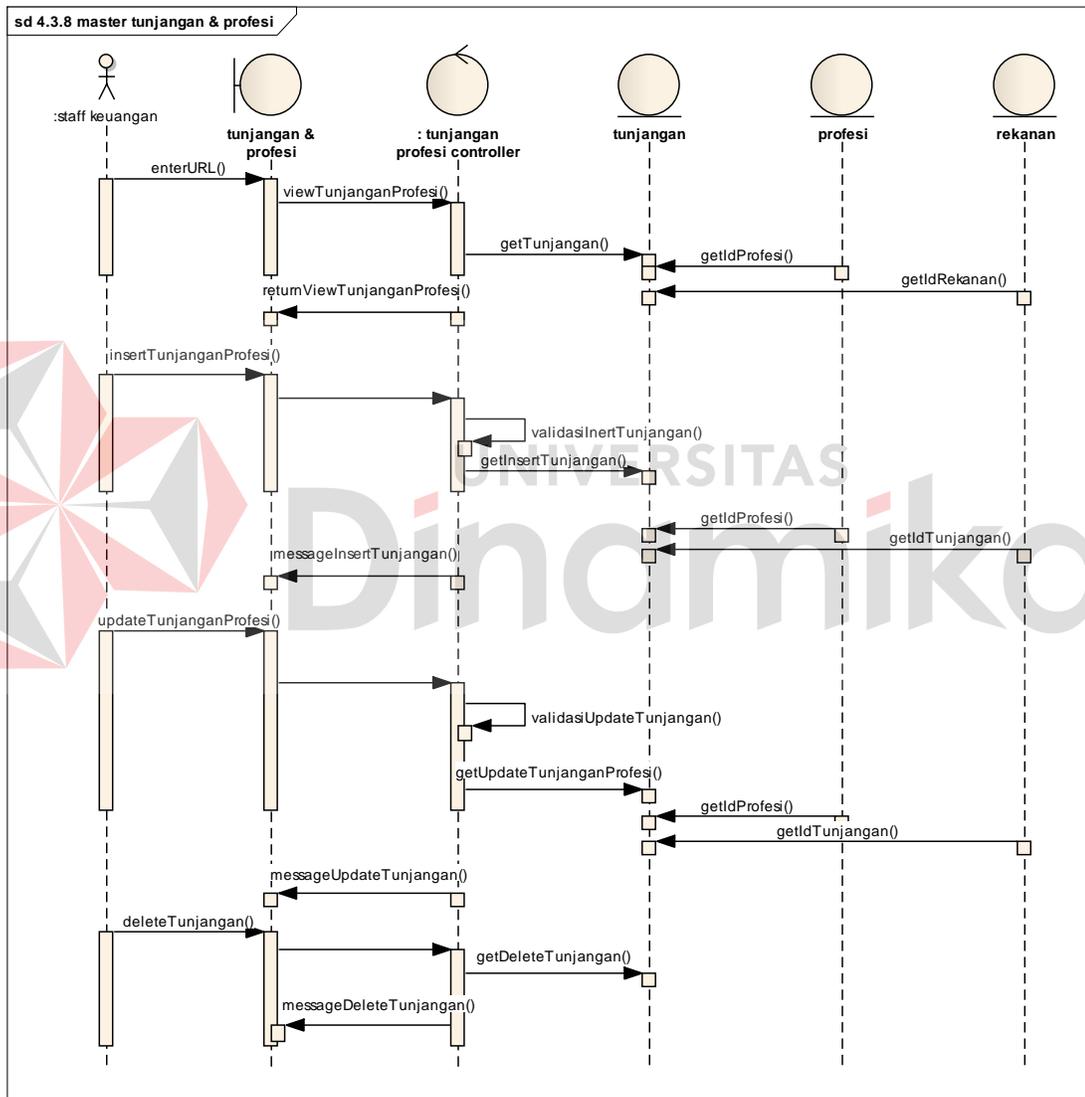
Data master gaji pokok akan disimpan di dalam database dengan tujuan untuk menyimpan data gaji pokok disetiap rekanan berdasarkan profesi yang ada di rekanan tersebut. Data master gaji pokok ini dapat ditambah, diubah, ataupun dihapus seperti gambar dibawah ini.



Gambar 4.8 Sequence Diagram Manage Data Master Gaji Pokok

#### 4.7.6 Sequence Diagram Manage Data Master Tunjangan & Profesi

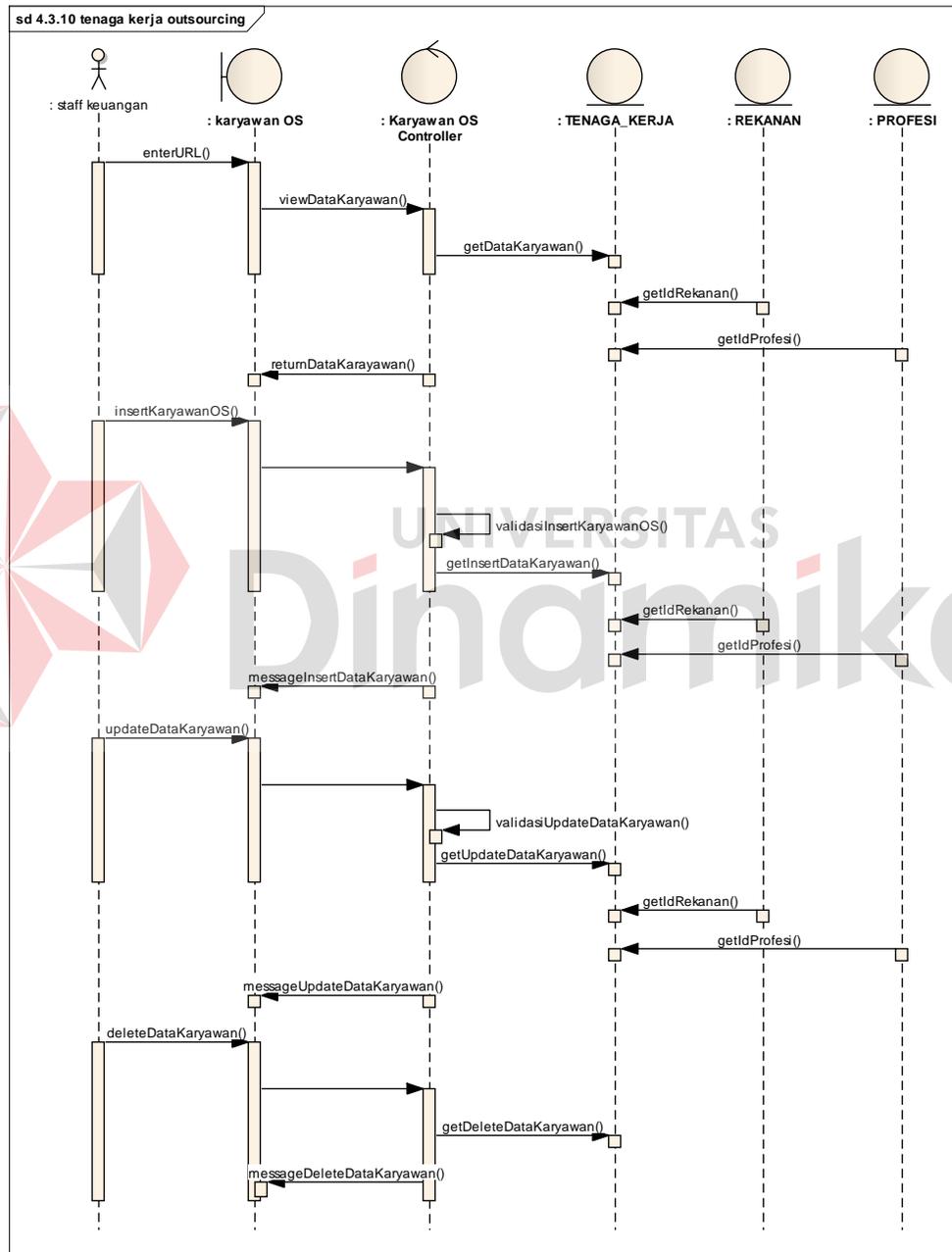
Data master tunjangan dan profesi adalah data tunjangan berdasarkan profesi pada setiap rekanan. Data master tunjangan ini dapat ditambah, diubah, ataupun dihapus seperti yang terlihat pada diagram dibawah ini.



Gambar 4.9 Sequence Diagram Manage Data Master Tunjangan & Profesi

#### 4.7.7 Sequence Diagram Manage Master Karyawan Outsourcing

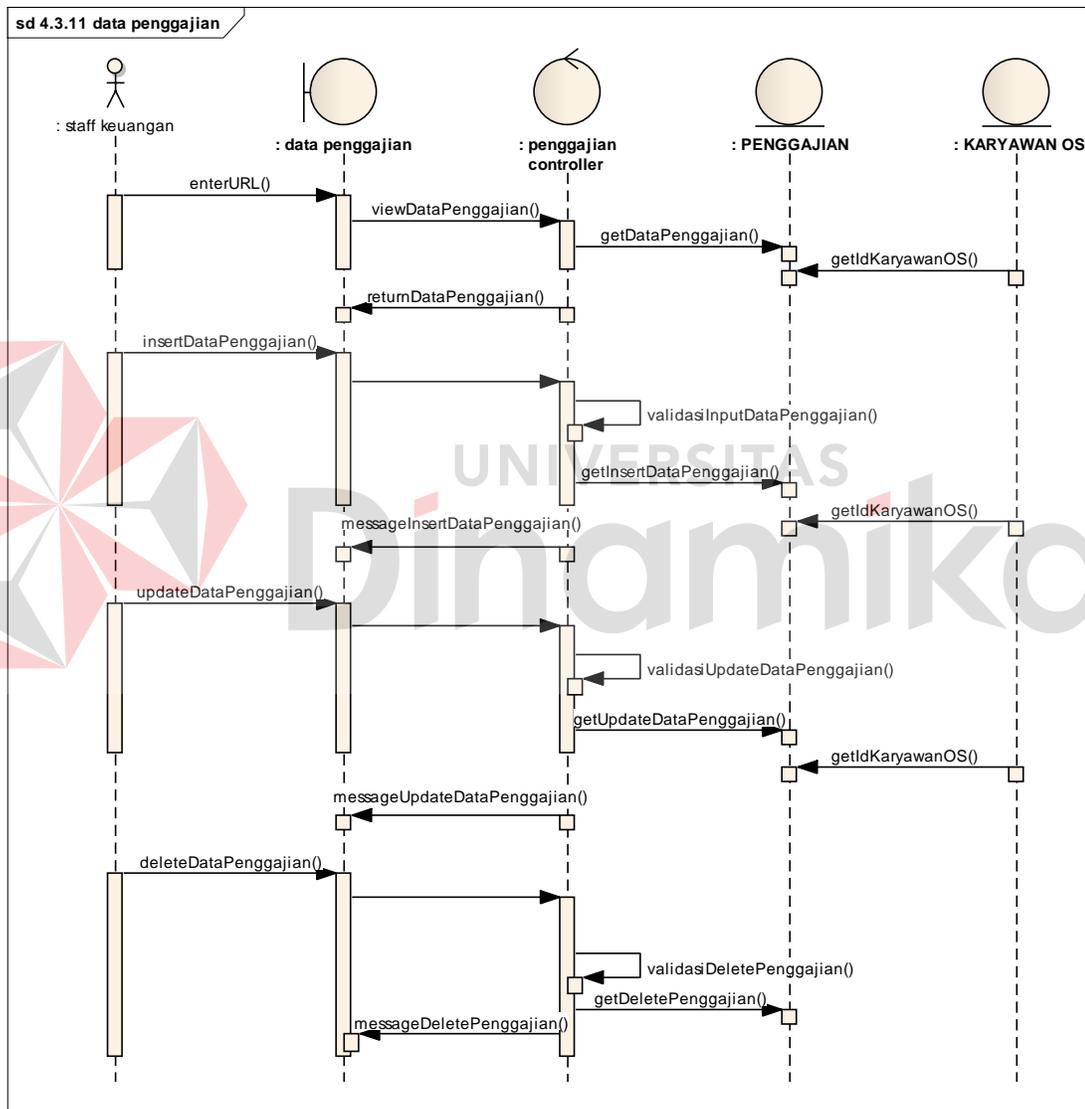
Data master karyawan *outsourcing* ini dapat ditambah, diubah, ataupun dihapus seperti pada gambar diagram dibawah ini.



Gambar 4.10 Sequence Diagram Manage Data Karyawan *Outsourcing*

#### 4.7.8 Sequence Diagram Manage Data Penggajian

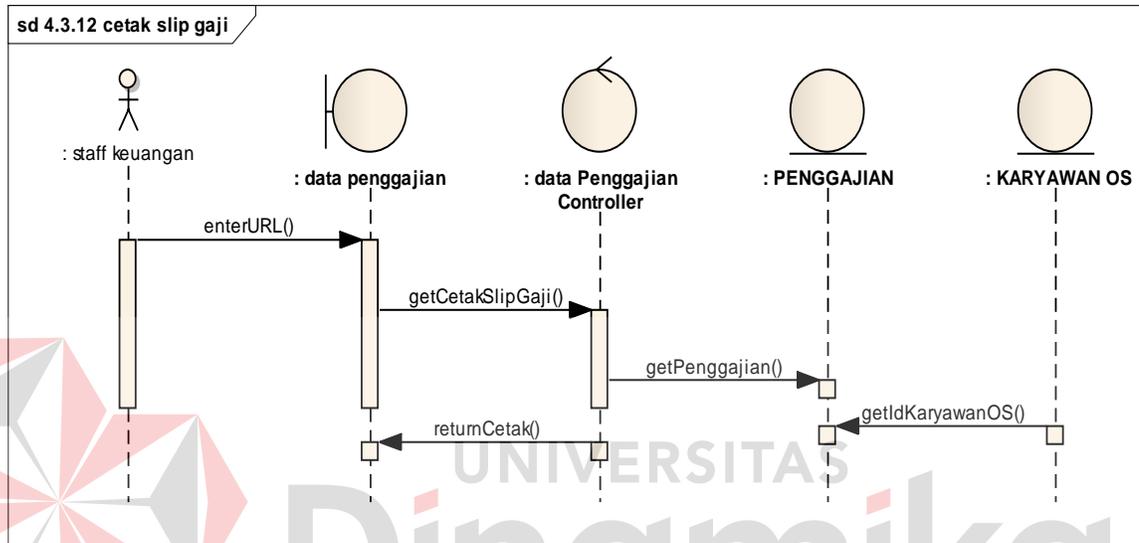
Data penggajian ini adalah data transaksi yang dimasukkan oleh staff. *Input* data pada transaksi perlu ketelitian agar keakuratan data tetap terjaga. Proses *input*, *ubah*, dan *hapus* dapat dilihat pada diagram dibawah ini.



Gambar 4.11 Sequence Diagram Manage Data Penggajian

#### 4.7.9 Sequence Diagram Cetak Slip Gaji

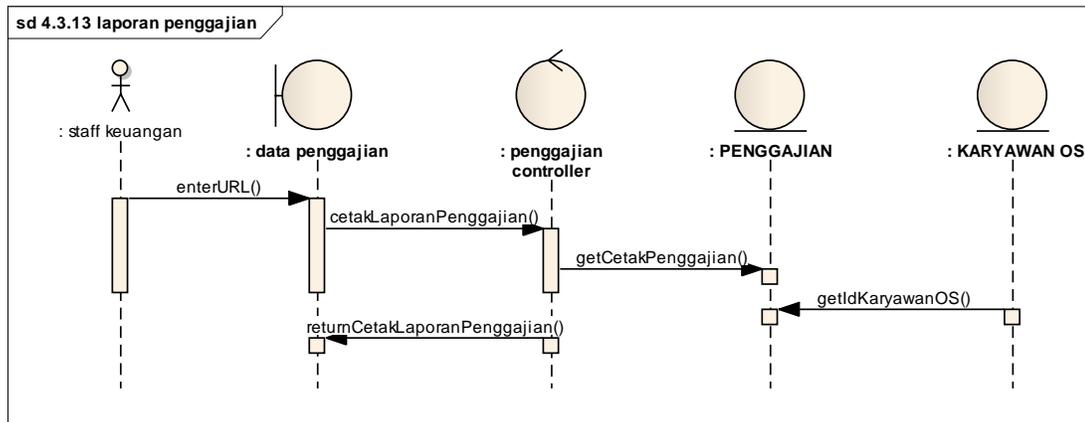
Cetak slip gaji ini akan dilakukan oleh staff. Slip gaji akan dicetak dan akan disimpan dalam format *\*.pdf* kemudian dicetak. Proses pencetakan tersebut dapat dilihat pada proses dibawah ini.



Gambar 4.12 Sequence Diagram Cetak Slip Gaji

#### 4.7.10 Sequence Diagram Cetak Laporan Penggajian

Cetak laporan penggajian ini dilakukan oleh staff. Laporan penggajian ini akan disimpan dalam bentuk format *\*.xls* dan dibuka menggunakan aplikasi excel. Proses cetak laporan penggajian ini dijelaskan dalam diagram dibawah ini.



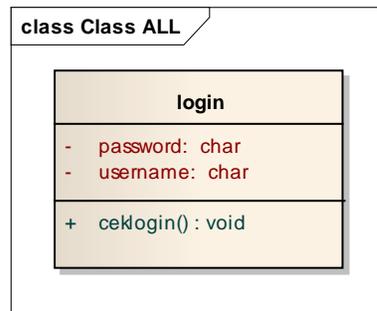
Gambar 4.13 *Sequence Diagram* Cetak Laporan Penggajian

## 4.8 Class Diagram

*Class diagram* digunakan untuk menampilkan kelas-kelas dalam sistem dan relasi antara mereka. Berdasarkan perancangan sistem pada diagram *use case*, dibutuhkan *class-class* untuk membangun dan mendukung jalannya aplikasi. Pada diagram *use-case* telah dijabarkan kegiatan-kegiatan yang akan dilakukan oleh sistem. Kemudian pada *sequence-diagram* juga telah diberikan penjelasan alur yang dilakukan oleh sistem. *Use-case* dan *sequence-diagram* tersebut akan dibuat *class-class* yang akan digunakan dalam kode pemrograman.

### 4.8.1 Class login

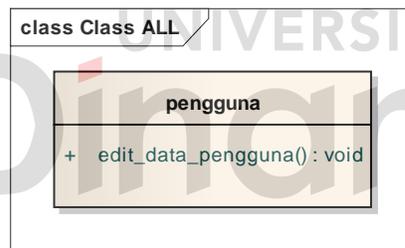
*Class login* merupakan *class* yang mengatur pengguna mana saja yang boleh mengakses aplikasi penggajian ini. *Class* ini memiliki *method* bernama *ceklogin* untuk mengecek apakah pengguna terdaftar untuk mengakses sistem.

Gambar 4.14 *Class login*

### 4.8.2 Class Pengguna

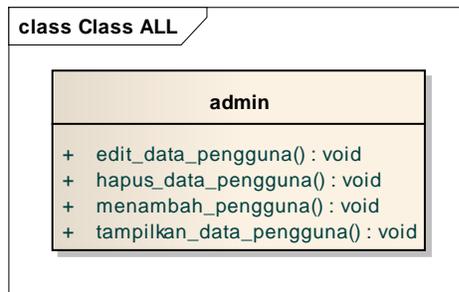
*Class pengguna* merupakan *class* yang mengatur data-data seorang pengguna.

*Class* ini memiliki *method* bernama *edit\_data\_pengguna* untuk mengganti nama pengguna dan kata sandi pengguna.

Gambar 4.15 *Class pengguna*

### 4.8.3 Class Admin

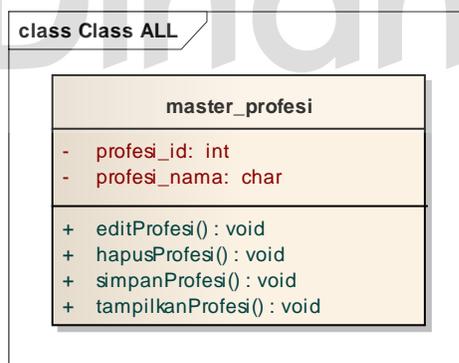
*Class admin* merupakan *class* yang mengatur data-data pengguna yang akan mengakses aplikasi ini. *Class* ini memiliki *method* bernama *edit\_data\_pengguna* untuk mengganti nama pengguna dan kata sandi pengguna, *hapus\_data\_pengguna* untuk menghapus data pengguna, *menambah\_pengguna* untuk menambah pengguna, dan *tampilkan\_data\_pengguna* untuk menampilkan data pengguna.



Gambar 4.16 Class admin

#### 4.8.4 Class master\_profesi

*Class master\_profesi* merupakan *class* yang mengatur data-data yang berhubungan dengan profesi karyawan *outsourcing*. *Class* ini memiliki *method* bernama *editProfesi* untuk mengganti nama profesi, *hapusProfesi* untuk menghapus profesi, *simpanProfesi* untuk menambah data profesi, dan *tampilkanProfesi* untuk menampilkan profesi.



Gambar 4.17 Class master\_profesi

#### 4.8.5 Class master\_rekanan

*Class master\_rekanan* merupakan *class* yang mengatur data-data yang berhubungan dengan rekanan tempat karyawan *outsourcing* bekerja. *Class* ini memiliki *method* bernama *editRekanan* untuk mengganti nama rekanan,

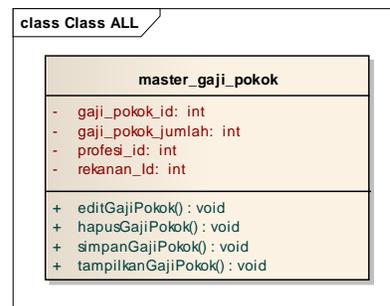
*hapusRekanan* untuk menghapus rekanan, *simpanRekanan* untuk menambah data rekanan, dan *tampilkanRekanan* untuk menampilkan data rekanan.



Gambar 4.18 Class master\_rekanan

#### 4.8.6 Class master\_gaji\_pokok

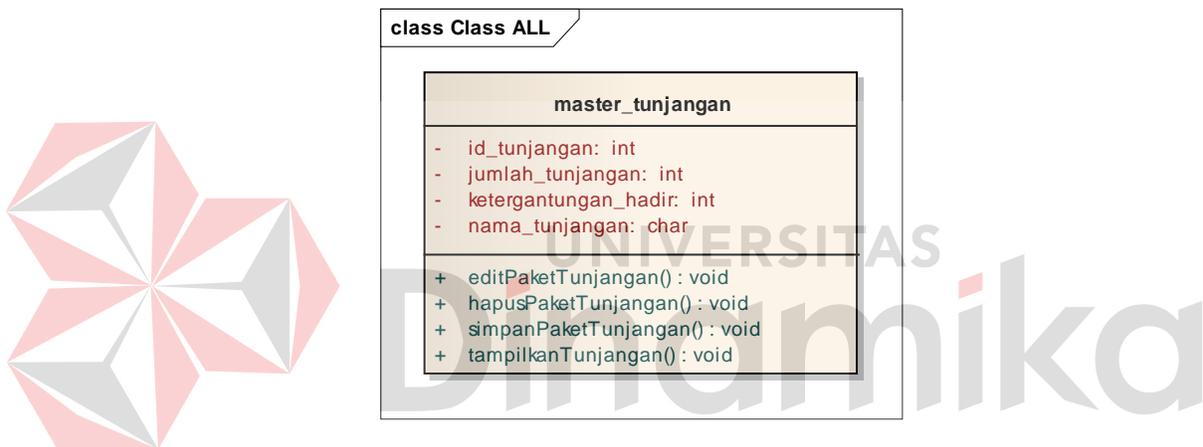
*Class master\_gaji\_pokok* merupakan *class* yang mengatur data-data yang berhubungan dengan gaji pokok karyawan *outsourcing* bekerja. *Class* ini memiliki *method* bernama *editGajiPokok* untuk mengganti jumlah gaji pokok, *hapusGajiPokok* untuk menghapus gaji pokok yang tidak digunakan, *simpanGajiPokok* untuk menambah data gaji pokok, dan *tampilGajiPokok* untuk menampilkan data gaji pokok.



Gambar 4.19 Class master\_gaji\_pokok

#### 4.8.7 Class master\_tunjangan

*Class master\_tunjangan* merupakan *class* yang mengatur data-data yang berhubungan dengan tunjangan karyawan *outsourcing* bekerja. *Class* ini memiliki *method* bernama *editPaketTunjangan* untuk mengganti jumlah tunjangan, *hapusPaketTunjangan* untuk menghapus tunjangan yang tidak digunakan, *simpanTunjangan* untuk menambah data tunjangan, dan *tampilkanTunjangan* untuk menampilkan data tunjangan.



Gambar 4.20 *Class* master\_tunjangan

#### 4.8.8 Class master\_os

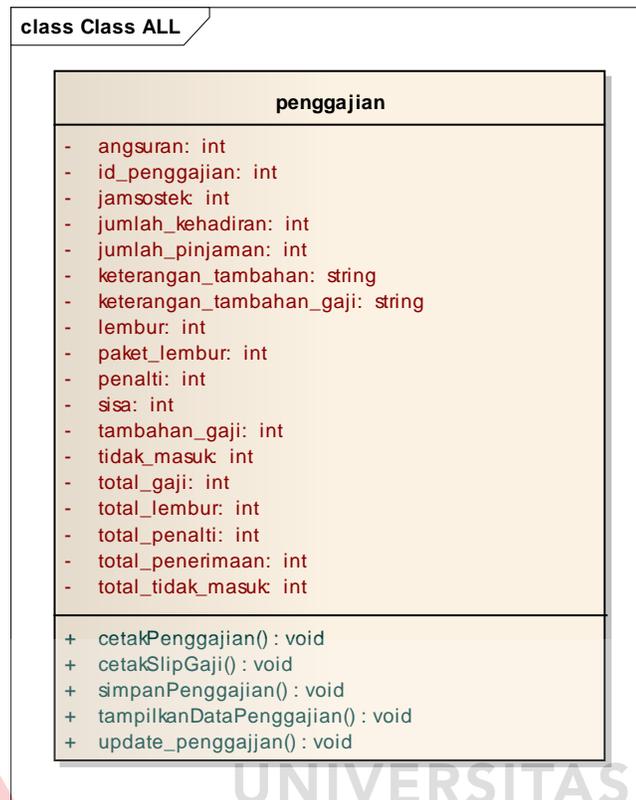
*Class master\_os* merupakan *class* yang mengatur data-data yang berhubungan dengan karyawan *outsourcing*. *Class* ini memiliki *method* bernama *edit* karyawan untuk mengganti data-data karyawan *outsourcing*, *hapus* karyawan untuk menghapus karyawan *outsourcing*, *simpan* karyawan untuk menambah data karyawan *outsourcing*, dan *tampilkanKaryawan* untuk menampilkan karyawan *outsourcing*.



Gambar 4.21 Class master\_os

#### 4.8.9 Class penggajian

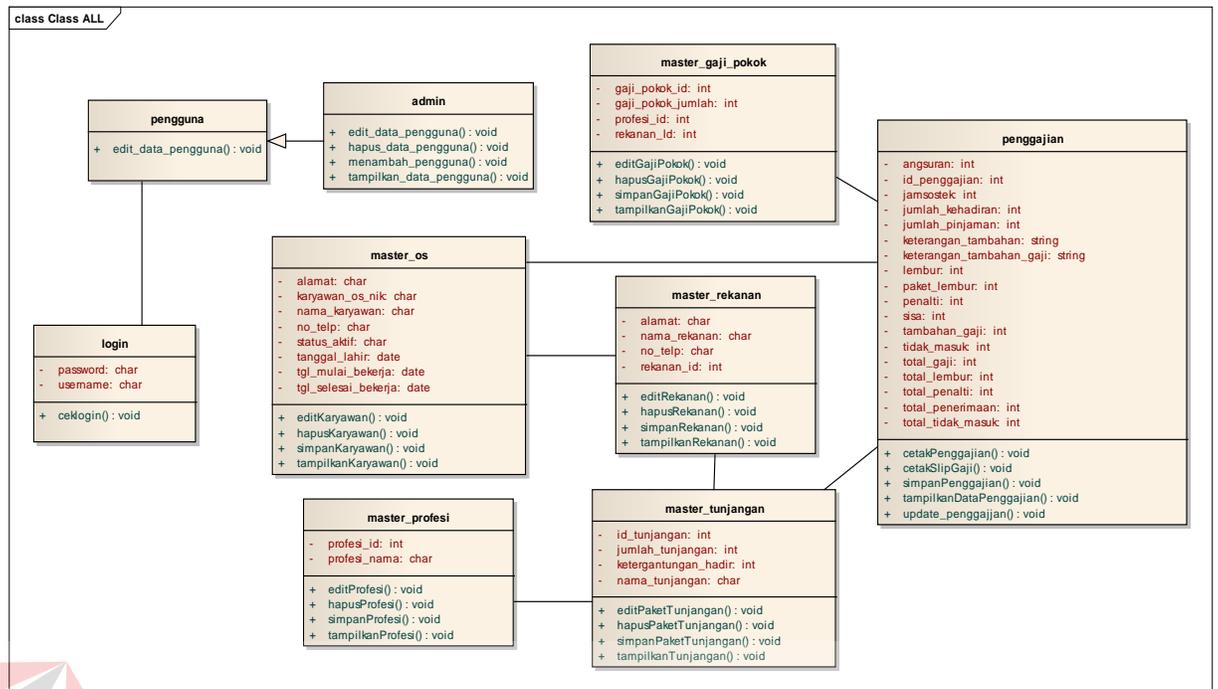
*Class penggajian* merupakan *class* yang mengatur data-data yang berhubungan dengan penggajian karyawan *outsourcing*. *Class* ini memiliki *method* bernama *cetakpenggajian* untuk mencetak data penggajian perrekanan tiap bulan dalam bentuk format pdf, *cetakslipgaji* untuk cetak daftar gaji karyawan *outsourcing*, *update\_penggajian* untuk mengganti data-data penggajian, *simpanPenggajian* untuk menambah data penggajian, dan *tampilkanDataPenggajian* untuk menampilkan data penggajian.



Gambar 4.22 Class penggajian

#### 4.8.10 Relasi Class Diagram

*Class-class* yang dibuat memiliki relasi *class diagram*. Relasi ini bertujuan untuk menjelaskan hubungan antara *class* satu dengan yang lainnya. *Class* admin berhubungan dengan *class* pengguna. Hubungan *class* tersebut adalah *class* generalisasi. *Class* login berhubungan dengan *class* pengguna. Hubungan kedua *class* itu disebut dengan asosiasi. Kemudian ada *class* master\_gaji\_pokok yang berhubungan, *class* master\_tunjangan, *class* master\_os yang berhubungan dengan *class* penggajian. Hubungan *class*-nya disebut dengan asosiasi. Kemudian ada *class* rekanan dan *class* master\_produksi yang berhubungan dengan *class* master\_tunjangan. *Class* rekanan berhubungan dengan *class* master\_os



Gambar 4.23 Relasi class diagram

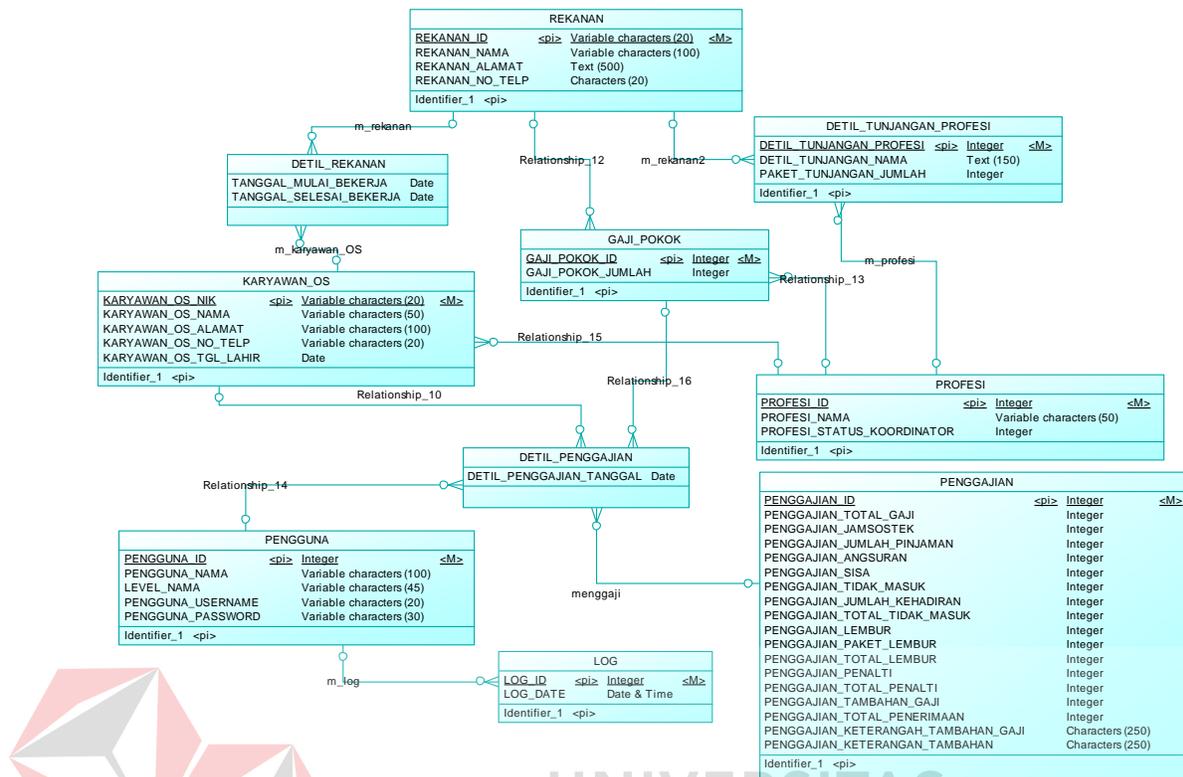
## 4.9 Entity Relational Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan proses yang menunjukkan hubungan antar entitas dan relasinya. ERD terbagi menjadi *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM).

### 4.9.1 Conceptual Data Model (CDM)

*Conceptual Data Model* (CDM) adalah gambaran secara keseluruhan struktur aplikasi. CDM dapat membangun disain awal sistem dan tidak perlu khawatir dengan detail implementasinya secara fisik. Dan melalui prosedur *generation* yang mudah, kita bisa melakukan *generate* CDM ke *Physical Data Model* (PDM).

Bentuk *Conceptual Data Model* dari aplikasi penggajian PT BPT ada pada gambar 4.22

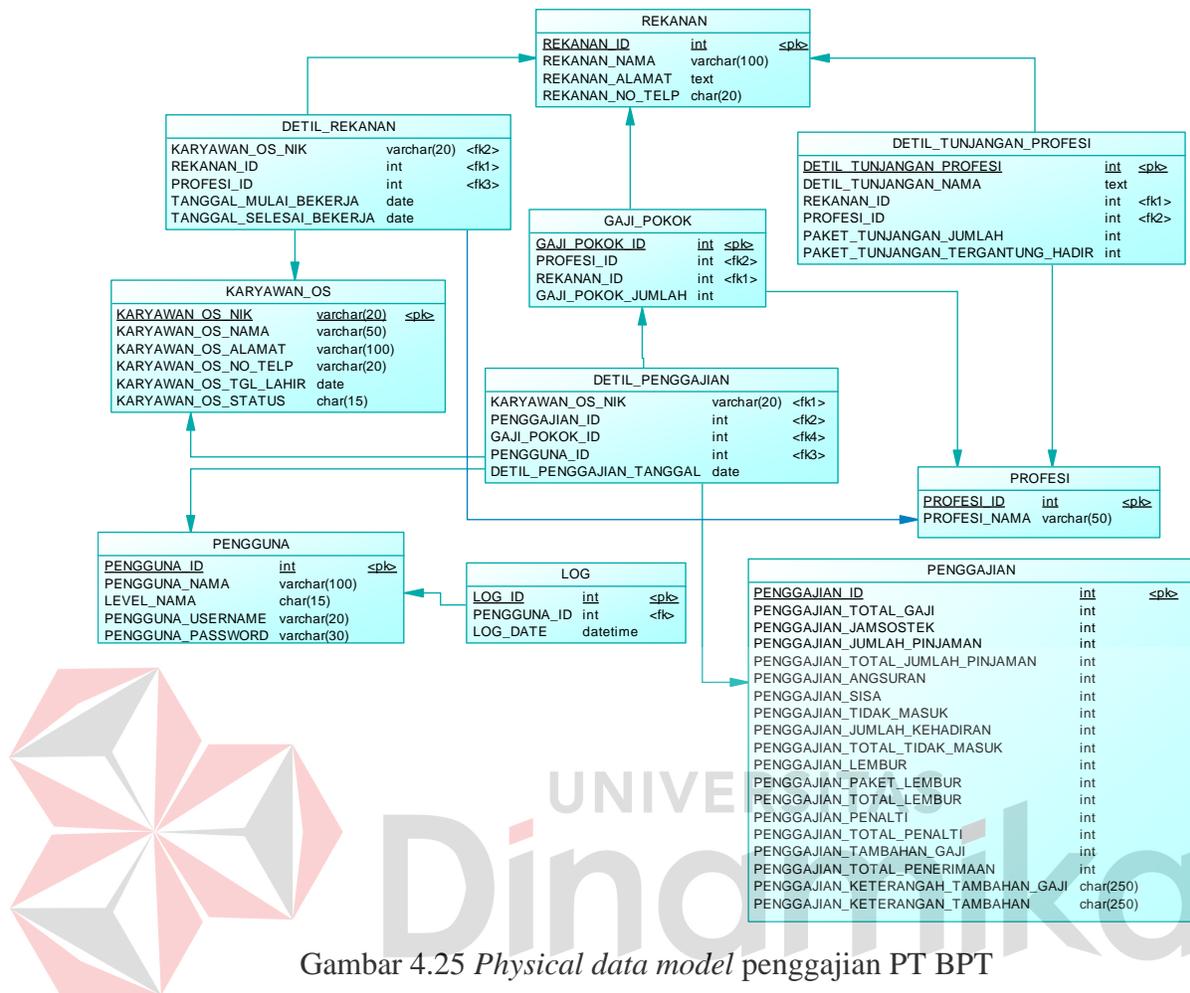


Gambar 4.24 *Conceptual Data Model* Penggajian PT BPT

## 4.9.2 *Physical Data Model (PDM)*

Physical Data Model menggambarkan struktur data sebagaimana akan di implementasikan oleh DBMS. PDM dapat mengoptimalkan *database* dengan memodifikasi tabel, kolom, *index*, *refrential integrity*, *view*, *physical storage*, *trigger and stored procedure*. *Procedure database generation* menerapkan hal itu dengan cara menyesuaikan dengan DBMS yang kita pilih.

Bentuk *Physical data model* dari *generate conceptual data model* untuk aplikasi penggajian PT BPT ada pada gambar 4.23



Gambar 4.25 Physical data model penggajian PT BPT

### 4.9.3 Struktur Basis data dan Tabel

Struktur tabel digunakan dalam pembuatan aplikasi penggajian PT BPT. Data-data dibawah ini akan menjelaskan satu per satu secara detil dari struktur tabel sistem.

**a. Nama Tabel : PENGGUNA**

Primary Key : PENGGUNA\_ID

Foreign Key : LEVEL\_ID

Fungsi : Untuk menyimpan data pengguna aplikasi penggajian yang ada dalam PT BPT.

Tabel 4.11 PENGGUNA

No.	Name	Type	Size	Keterangan
1.	PENGGUNA_ID	int	-	Primary Key
2.	LEVEL_NAMA	int	-	Foreign Key
3.	PENGGUNA_NAMA	varchar	100	-
4.	PENGGUNA_USERNAME	varchar	20	-
5.	PENGGUNA_PASSWORD	varchar	30	-

**b. Nama Tabel : LOG**

Primary Key : LOG\_ID

Foreign Key : PENGGUNA\_ID

Fungsi : Untuk menyimpan log pengguna.

Tabel 4.12 LOG

No.	Name	Type	Size	Keterangan
1.	LOG_ID	Int	-	Primary Key
2.	PENGGUNA_ID	Int	-	Foreign Key
3.	LOG_DATE	datetime	-	-

**c. Nama Tabel : REKANAN**

Primary Key : REKANAN\_ID

Foreign Key :

Fungsi : Untuk menyimpan data rekanan.

Tabel 4.13 REKANAN

No.	Name	Type	Size	Keterangan
1.	REKANAN_ID	Int	-	Primary Key
2.	REKANAN_NO_TELP	char	100	
3.	REKANAN_NAMA	Varchar	100	-
4.	REKANAN_ALAMAT	text	-	-

**d. Nama Tabel : DETIL\_REKANAN**

Primary Key : -

Foreign Key : KARYAWAN\_OS\_NIK, REKANAN\_ID

Fungsi : Untuk menyimpan detail rekanan.

Tabel 4.14 DETIL\_REKANAN

No.	Name	Type	Size	Keterangan
1.	KARYAWAN_OS_NIK	Varchar	20	Foreign Key
2.	REKANAN_ID	Int	-	Foreign Key
3.	PROFESI_ID	int	-	-
3.	TANGGAL_MULAI_BEKERJA	Date	-	-
4.	TANGGAL_SELESAI_BEKERJA	Date	-	-

**e. Nama Tabel : DETIL\_TUNJANGAN\_PROFESI**

Primary Key : DETIL\_TUNJANGAN\_PROFESI

Foreign Key : PAKET\_TUNJANGAN\_ID, REKANAN\_ID, PROFESI\_ID,  
PAKET\_TUNJANGAN\_JUMLAH,  
PAKET\_TUNJANGAN\_TERGANTUNG\_HARDIR

Fungsi : Untuk menyimpan nama paket tunjangan.

Tabel 4.15 TUNJANGAN\_PROFESI

No.	Name	Type	Size	Keterangan
1.	DETIL_TUNJANGAN_PROFESI	Int	-	Primary Key
2.	DETIL_TUNJANGAN_NAMA	text	-	-
3.	REKANAN_ID	Int	-	Foregin Key
4.	PROFESI_ID	Int	-	Foregin Key
5.	PAKET_TUNJANGAN_JUMLAH	Int	-	-
6.	PAKET_TUNJANGAN_TERGANTUNG_HADIR	Int	-	-

**f. Nama Tabel : PROFESI**

Primary Key : PROFESI\_ID

Foreign Key : -

Fungsi : Untuk menyimpan nama-nama profesi karyawan  
*outsourcing*.

Tabel 4.16 PROFESI

No.	Name	Type	Size	Keterangan
1.	PROFESI_ID	Int	-	Primary Key
2.	PROFESI_NAMA	Varchar	50	-

**g. Nama Tabel : GAJI\_POKOK**

Primary Key : GAJI\_POKOK\_ID

Foreign Key : PROFESI\_ID, REKANAN\_ID

Fungsi : Untuk menyimpan gaji pokok karyawan *outsourcing*.

Tabel 4.17 GAJI\_POKOK

No.	Name	Type	Size	Keterangan
1.	GAJI_POKOK_ID	Int	-	Primary Key
2.	PROFESI_ID	Int	-	Foreign Key
3.	REKANAN_ID	Int	-	Foreign Key
4.	GAJI_POKOK_JUMLAH	Int	-	-

**h. Nama Tabel : DETIL\_PENGGAJIAN**

Primary Key : -

Foreign Key : KARYAWAN\_OS\_NIK, PENGGAJIAN\_ID,  
GAJI\_POKOK\_ID, PENGGUNA\_ID,  
DETIL\_PENGGAJIAN\_TANGGAL

Fungsi : Untuk menyimpan nama-nama profesi karyawan  
*outsourcing*.

Tabel 4.18 DETIL\_PENGGAJIAN

No.	Name	Type	Size	Keterangan
1.	KARYAWAN_OS_NIK	Varchar	20	Foreign Key
2.	PENGGUNA_ID	int	-	Foreign Key
3.	GAJI_POKOK_ID	int	-	Foreign Key
4.	PENGGUNA_ID	int	-	Foreign Key
5.	DETIL_PENGGAJIAN_TANGGAL	Date	-	-

**i. Nama Tabel : KARYAWAN\_OS**

Primary Key : KARYAWAN\_OS\_NIK

Foreign Key : -

Fungsi : Untuk menyimpan nama-nama karyawan

*outsourcing.*

Tabel 4.19 KARYAWAN\_OS

No.	Name	Type	Size	Keterangan
1.	KARYAWAN_OS_NIK	Varchar	20	Primary Key
2.	KARYAWAN_OS_NAMA	Varchar	50	-
3.	KARYAWAN_OS_ALAMAT	Varchar	100	-
4.	KARYAWAN_OS_NO_TELP	Varchar	20	-
5.	KARYAWAN_OS_TGL_LAHIR	date	-	-
6.	KARYAWAN_OS_STATUS	char	15	-

**j. Nama Tabel : PENGGAJIAN**

Primary Key : PENGGAJIAN\_ID

Foreign Key : -

Fungsi : Untuk menyimpan data penggajian

Tabel 4.20 PENGGAJIAN

No.	Name	Type	Size	Keterangan
1.	PENGGAJIAN_ID	Int	-	Primary Key

No	Name	Type	Size	Keterangan
2.	PENGAJIAN_TOTAL_GAJI	Int	-	-
3.	PENGGAJIAN_JAMSOSTEK	Int	-	-
4.	PENGGAJIAN_JUMLAH_PINJAMAN	Int	-	-
5.	PENGGAJIAN_TOTAL_JUMLAH_PINJAMAN	Int	-	-
6.	PENGGAJIAN_ANGSURAN	Int	-	-
7.	PENGGAJIAN_SISA	Int	-	-
8.	PENGGAJIAN_TIDAK_MASUK	Int	-	-
9.	PENGGAJIAN_LEMBUR	Int	-	-
10.	PENGGAJIAN_PAKET_LEMBUR	Int	-	-
11.	PENGGAJIAN_TOTAL_LEMBUR	Int	-	-
12.	PENGGAJIAN_LEMBUR	Int	-	-
13.	PENGGAJIAN_TOTAL_LEMBUR	Int	-	-
14.	PENGGAJIAN_PENALTI	Int	-	-
15.	PENGGAJIAN_TOTAL_PENALTI	Int	-	-
16.	PENGGAJIAN_TAMBAHAN_GAJI	Int	-	-
17.	PENGGAJIAN_TOTAL_PENERIMAAN	Int	-	-
18.	PENGGAJIAN_KETERANGAN_TAMBAHAN_GAJI	Char	250	-
19.	PENGGAJIAN_KETERANGAN_TAMBAHAN	Char	250	-

## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1 Disain *Input Output*

Disain *Input Output* ini dapat kita lakukan sebelum disain *interface* yang sesungguhnya kita buat dengan melalui program. Dengan desain ini, para pengguna dapat membayangkan apakah aplikasi yang akan dibuat tersebut sesuai dengan kebutuhan yang ada di perusahaan tersebut. Dalam aplikasi ini ini terdapat beberapa desain input dan output, antara lain:



Gambar 5.1 *Form Login*

*Form login* ini digunakan untuk masuk ke dalam aplikasi dan menentukan hak akses pengguna dalam aplikasi, antara lain sebagai staff keuangan, admin, atau *manager HRD*. *Form* ini terdapat *button* “Masuk” dan dua *textbox* yaitu nama pengguna dan kata sandi untuk mengolah semua proses *login*.

**b. Disain Form Menu Utama**

Gambar 5.2 *Form* Menu Utama

*Form* utama ini memiliki tiga menu utama yang terdiri dari data master, transaksi, dan laporan. Di setiap *menu* terdapat *sub menu*. *Menu* yang pertama yaitu data master memiliki *sub menu* yang terdiri dari *menu* pengguna, profesi, rekanan, gaji pokok, tunjangan, dan karyawan *outsourcing*. *Menu* yang kedua yaitu transaksi memiliki *sub menu* yaitu penggajian. *Menu* yang ketiga yaitu *menu* laporan memiliki *sub menu* yaitu slip gaji dan laporan penggajian.

**c. Disain Form Pengguna**

No	Nama Lengkap	Nama Pengguna	Level
1	Zayed Elfasa	staff	staff
2	Juremi	staff01	staff

Gambar 5.3 Disain *Form* Pengguna

*Form* pengguna terdapat pada *menu* data master. *Form* pengguna berfungsi untuk mengolah data pengguna yang akan mengkases aplikasi penggajian ini. Pada *form* ini terdapat *textbox* yang terdiri dari nama lengkap, nama pengguna, kata sandi, dan level pengguna. *Form* ini juga terdapat *table datagrid* yang memperlihatkan data pengguna. Selain itu terdapat *button save* , *edit*, dan *delete*. *Form* ini hanya akan digunakan oleh admin untuk mengelola data pengguna.

#### d. Disain *Form* Profesi

No	Nama Profesi
1	Sapu Jalan
2	Supir

Gambar 5.4 Disain *Form* Profesi

*Form* profesi terdapat pada *menu* data master. *Form* profesi berfungsi untuk mengolah data profesi sebagai profesi karyawan *outsourcing* bekerja. Pada *form* ini terdapat *textbox* yang terdiri dari nama profesi. *Form* ini juga terdapat *table datagrid* yang memperlihatkan data profesi. Selain itu terdapat *button save* , *edit*, dan *delete*. *Form* ini hanya akan digunakan oleh staff keuangan dan *manager* HRD untuk mengelola data profesi.

e. **Disain Form Rekanan**

FORM REKANAN

DATA MASTER    TRANSAKSI    LAPORAN

PENGGUNA    PROFESI    REKANAN    GAJI POKOK    TUNJANGAN    KARYAWAN OS

NAMA REKANAN    Text box

ALAMAT    Text box

NO TELP    Text box

SAVE    EDIT    DELETE

Gambar 5.5 Disain Form Rekanan

*Form* rekanan terdapat pada *menu* data master. *Form* profesi berfungsi untuk mengolah data rekanan dimana tempat karyawan *outsourcing* bekerja. Pada *form* ini terdapat *textbox* yang terdiri dari nama rekanan, alamat, dan nomor telepon. *Form* ini juga terdapat *table datagrid* yang memperlihatkan data rekanan. Selain itu terdapat *button save* , *edit*, dan *delete*. *Form* ini hanya akan digunakan oleh staff keuangan dan *manager* HRD untuk mengelola data rekanan.

f. **Disain Form Gaji Pokok**

No	Nama Profesi	Rekanan	Jumlah Gaji Pokok
1	Sapu Jalan	PT ABC	1.500.000,00
2	Supir	PT MNA	1.800.000,00

Gambar 5.6 Disain *Form Gaji Pokok*

*Form Gaji Pokok* terdapat pada *menu data master*. *Form profesi* berfungsi untuk mengolah data gaji pokok karyawan *outsourcing*. Pada *form* ini terdapat *textbox* yang terdiri dari nama profesi, nama rekanan, dan jumlah gaji pokok. *Form* ini juga terdapat *table datagrid* yang memperlihatkan data gaji pokok. Selain itu terdapat *button save*, *edit*, dan *delete*. *Form* ini hanya akan digunakan oleh staff keuangan dan *manager HRD* untuk mengelola data gaji pokok.

**g. Disain Form Tunjangan & Profesi**

No	Nama Tunjangan	Nama Profesi	Rekanan	Jumlah Gaji Pokok
1	Tunjangan Transportasi	Sapu Jalan	PT ABC	15.000,00
2	Tunjangan Makan	Supir	PT MNA	30.000,00

Gambar 5.7 Disain Form Tunjangan & Profesi

*Form* tunjangan terdapat pada *menu* data master. *Form* profesi berfungsi untuk mengolah data tunjangan sebagai tunjangan karyawan *outsourcing*. Pada *form* ini terdapat *textbox* yang terdiri dari nama tunjangan, nama rekanan, nama profesi, jumlah tunjangan, dan status kehadiran. *Form* ini juga terdapat *table datagrid* yang memperlihatkan data tunjangan. Selain itu terdapat *button save*, *edit*, dan *delete*. *Form* ini hanya akan digunakan oleh staff keuangan dan *manager* HRD untuk mengelola data tunjangan.

### h. Disain *Form Karyawan Outsourcing*

Gambar 5.8 Disain *Form Karyawan Outsourcing*

*Form karyawan outsourcing* terdapat pada *menu data master*. *Form profesi* berfungsi untuk mengolah data karyawan *outsourcing*. Pada *form* ini terdapat *textbox* yang terdiri dari NIK Karyawan, Nama Karyawan, Alamat, No Telp, Tanggal Lahir, Status Aktif, Profesi, Rekanan, Tanggal Bekerja, dan Tanggal selesai bekerja. *Form* ini juga terdapat *table datagrid* yang memperlihatkan data karyawan OS. Selain itu terdapat *button save*, *edit*, dan *delete*. *Form* ini hanya akan digunakan oleh staff keuangan dan *manager HRD* untuk mengelola data karyawan OS.

i. **Disain Form Penggajian**

The screenshot shows a software window titled "FORM PENGGAJIAN" with a blue title bar. It contains several tabs: "DATA MASTER", "TRANSAKSI", and "LAPORAN". Below the tabs are two buttons: "PENGGAJIAN" and "VIEW PENGGAJIAN".

The form fields include:

- NIK: 11.4.100.01983
- PROFESI: Office Boy (dropdown)
- NAMA KARYAWAN: Ahmad Mucharror (dropdown)
- GAJI POKOK: 1.400.000
- JAMSOSTEK: 2.5% RP. 35.000
- TAMBAHAN GAJI: 0
- KETERANGAN: -
- KETERANGAN LAINNYA: -
- DETAIL TUNJANGAN table:
 

Nama Tunjangan	Tarif Tunja...	Total Tunj...
Tunjangan Hadir	3.400	95.200
Tunjangan Shift	1.500	42.000
Tunjangan Nutrisi	3.000	84.000
- TOTAL HADIR: 28
- TIDAK MASUK: 2 (JUMLAH: 3000, TOTAL TIDAK MASUK: 6000)
- PENALTI: 0 (JUMLAH: 0, TOTAL PENALTI: 0)
- JUMLAH PINJAMAN: 0
- PAYROLL: 0
- TUNJANGAN RP. 221.200
- TOTAL POTONGAN RP. 41.000
- LEMBUR: 1 (LEMBUR: 15000, TOTAL LEMBUR: 15000)
- ANGSURAN: 0
- SISA: 0
- TOTAL GAJI dan TUNJANGAN RP. 1.621.200
- TOTAL PENERIMAAN RP. 1.595.200

At the bottom right, there are four buttons: "PROSES", "SAVE", "EDIT", and "DELETE". A large watermark "Dinamika" is visible across the center of the form.

Gambar 5.9 Disain Form Penggajian

*Form* penggajian terdapat pada *menu* transaksi. *Form* penggajian berfungsi untuk mengolah data penggajian karyawan *outsourcing*. Pada *form* ini terdiri dari banyak *textbox* seperti yang terlihat pada gambar. Selain itu terdapat *button save*, *edit*, dan *delete*.

### j. Disain Form *View* Penggajian

FORM PENGGAJIAN

DATA MASTER | TRANSAKSI | LAPORAN

PENGGAJIAN | VIEW PENGGAJIAN

Nama Rekanan: PT MNA

Profesi: OFFICE BOY

Bulan: JUNI Tahun: 2015

PROSES

NIK Karyaw...	Nama Karyaw...	Total Gaji	Tunj Hadir	Tunj Shift	Total Tunj	Angsuran Pinj Sisa	Jamsostek Tdk Masuk	Tot. Tidak Masuk

FORM PENGGAJIAN

DATA MASTER | TRANSAKSI | LAPORAN

PENGGAJIAN | VIEW PENGGAJIAN

Nama Rekanan: PT MNA

Profesi: OFFICE BOY

Bulan: JUNI Tahun: 2015

PROSES

Penalti	Total Penalti	Total Potongan	Total Lembur	Keterangan Tambahan Gaji	Total Penerimaan

Gambar 5.10 Disain *Form View* Penggajian

*Form view* penggajian juga terdapat *table datagrid* yang memperlihatkan data penggajian yang dapat dikelola berdasarkan tempat karyawan *outsourcing* bekerja. *Form* ini hanya akan digunakan oleh staff keuangan dan *manager* HRD untuk mengelola data profesi.

### k. Disain Slip Gaji

 PT. BANGUN PERSADA TRANSPORTATION Jl. Raya Ponokawan 07 Krian - Sidoarjo	
<b>Nama</b>	: Ahmad Mucharror
<b>Jabatan</b>	: Office Boy
Rincian Penghasilan	
Periode Juni 2015	
<b>I Penerimaan</b>	<b>II Potongan</b>
<b>Gaji</b>	Tidak Masuk : 6,000.00
Gaji Pokok : 1,400,000.00	Pinjaman : 0,00
<b>Tunjangan</b>	Jamsostek : 35,000.00
Tunjangan Hadir : 95.200,00	Penalti : 0
Tunjangan B : 42.000,00	Lain - lain (Payroll) : 0
Tunjangan C : 84.000,00	
<b>III Lembur 1 x 15.000</b>	15,000.00
<b>IV Lain - lain</b>	0
<b>Total Gaji + Tunj + Lembur + Lain-lain</b>	<b>Total Potongan</b>
: 1.636.200,00	: 41.000,00
<b>V Penerimaan Bersih</b>	1.595.200,00

Gambar 5.11 Disain Slip Gaji

Slip gaji ini akan dicetak sebagai bukti bahwa karyawan *outsourcing* telah mendapatkan gaji. Pada slip gaji ini terdapat data nama lengkap karyawan *outsourcing*, jabatan, gaji pokok, berbagai jenis tunjangan, potongan, total gaji, total potongan, dan penerimaan bersih.

### l. Disain Laporan Penggajian

 LAPORAN TANDA TERIMA GAJI PT MNA SATPAM PERIODE: JUNI 2015								
Nama	Total Gaji & Tunjangan	Lembur	Jumlah Potongan				Total Penerimaan	TTD
			Angsuran	Tidak Masuk	Pinalty	Jamsostek		
Bagus Sudiro B.	1.629.100	0	0	2.000	0	35.000	1.219.825	
Abdul Rozaq	1.629.100	75.000	0	1.000	0	35.000	1.668.100	
Akmad Hadi S	1.621.200	15.000	0	6.000	0	35.000	1.595.200	
Ahmad Mucharror	1.621.200	15.000	0	6.000	0	35.000	1.595.200	
<b>TOTAL</b>	<b>6.500.600</b>	<b>105.000</b>	<b>0</b>	<b>15.000</b>	<b>0</b>	<b>140.000</b>	<b>6.078.325</b>	

Gambar 5.12 Disain Laporan Penggajian

Laporan penggajian ini akan dicetak sebagai laporan kepada manager HRD. Laporan penggajian ini berisi daftar penggajian karyawan *outsourcing* tiap bulannya di setiap rekanan. Laporan akan memiliki kolom-kolom yang terdiri dari nama karyawan *outsourcing* gaji pokok, lembur, jumlah potongan untuk angsuran, ketidakhadiran, pinalty, jamsostek, total penerimaan, dan paraf karyawan *outsourcing*.

### 5.1.1 Implementasi dan Evaluasi

Implementasi aplikasi ini akan menjelaskan detail aplikasi absensi dan penggajian di PT BPT, serta menjelaskan form-form yang ada.

Mengimplementasikan aplikasi merupakan tahap pengujian dimana disain aplikasi dapat berjalan dengan baik. Implementasi harus sesuai dengan hasil analisis aplikasi.

### 5.1.2 Kebutuhan Aplikasi

*Hardware* dan *software* yang dibutuhkan untuk menggunakan program aplikasi penggajian pada PT BPT, yaitu:

#### a. *Hardware*

1. Microprocessor Pentium IV
2. VGA dengan resolusi 800 x 600 dan mendukung Microsoft Widows 7.
3. RAM 512 MB.

#### b. *Software*

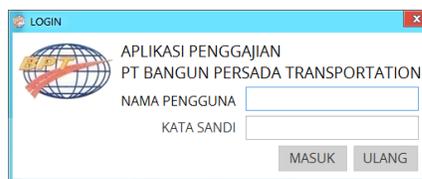
1. Sistem Oprasi Micrososft Windows 7
2. Sql Server 2008 R2

3. .NET Framework 4
4. Crystal Report Engine

### 5.1.3 Penjelasan Program

Di bawah ini merupakan penjelasan mengenai penggunaan masing-masing form yang ada pada aplikasi penggajian pada PT BPT.

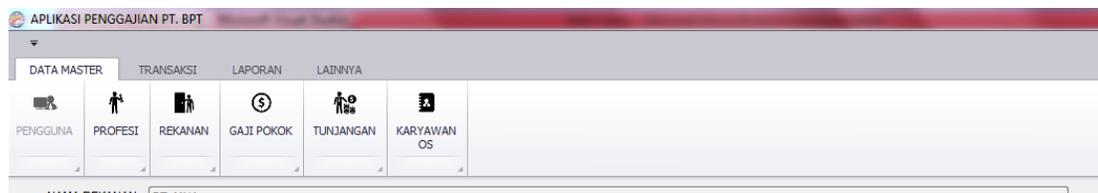
#### a. *Form Login*



Gambar 5.13 *Form Login*

*Form Login* ini berfungsi sebagai keamanan aplikasi, sehingga siapapun tidak dapat masuk ke dalam aplikasi jika tidak memiliki hak akses berupa nama pengguna dan kata sandi kemudian hak akses aplikasi disesuaikan dengan level pengguna.

#### b. *Form Menu Utama*



Gambar 5.14 *Form Menu Utama*

Form menu utama ini akan menampilkan menu-menu yang dapat digunakan dalam aplikasi. Menu utama terdiri dari “DATA MASTER”, “TRANSAKSI”,

“LAPORAN”, “LAINNYA”. “DATA MASTER” memiliki sub menu yaitu pengguna (hanya bisa diakses oleh admin), profesi, rekanan, gaji pokok, tunjangan, dan karyawan OS (*Outsourcing*). “TRANSAKSI” memiliki sub menu yang terdiri dari penggajian dan view data penggajian. “LAPORAN” memiliki sub menu yang terdiri dari slip gaji dan laporan penggajian. “LAINNYA” memiliki sub menu yaitu *Logout*.

### c. Form Pengguna

ID	NAMA LENGKAP	LEVEL PENGGUNA	NAMA PENGGUNA	KATA SANDI
1	Zayed Elfasa	admin	admin	12345
2	M Oby	direktur	direktur	12345
3	Atok Lindarto	manager	manager	12345
4	Juremi	staff	staff	12345

This Program Created By : Zayed Elfasa (STIKOM Surabaya)

Gambar 5.15 *Form Pengguna*

Form Pengguna ini adalah form yang digunakan oleh admin aplikasi untuk mengelola pengguna yang diizinkan untuk mengakses aplikasi penggajian ini. Level yang ada pada form aplikasi ini adalah admin.

#### d. Form Profesi

ID	NAMA PROFESI
1	Operator derek
2	Korlap Derek
3	Driver Derek
4	Assisten Driver Derek
5	Pul Tol
6	Assisten Patroli 2
7	Driver Ambulance
8	Receptionist
9	Satpam Aktif
10	Satpam
11	Koordinator Lapangan
12	Office Boy
13	Operator Head Trailer
14	Sapu Jalan
15	IT
16	Helper
17	Mekanik
18	Costumer Service (CS)
19	Office Boy Koord.
20	Satpam 2

This Program Created By : Zayed Elfasa (STIKOM Surabaya)

Gambar 5.16 Form Profesi

Form Profesi ini digunakan untuk mengelola profesi. Karyawan *outsourcing* yang dikelola oleh PT BPT memiliki nama profesi yang disesuaikan dengan rekanan tempat karyawan *outsourcing* akan ditempatkan. Level yang ada pada form aplikasi ini adalah *manager* dan staff keuangan.

### e. Form Rekanan

ID	NAMA REKANAN	ALAMAT	NO TELP
1	PT. KURNIA MEGA	Jl. Semampir tengah	082375236
2	PT. BIMA	Jl. Surabaya Selatan	08235976
3	PT. HUSADA PUTRA	Jl. Gemah Ripah Loh Jinawi	023858726
4	PT. KARYA ABADI	Jl. Prapanca Selatan	02375235
5	PT. ANUGERAH PERKASA	Jl. Soekarno	0823765892
6	PT. HUSADA ADI UTAMA	Jl. Panglima Adi Wijaya	02835986436
7	PT. MNA	Jl. Panglima Patamon	0238765

This Program Created By : Zayed Elfasa (STIKOM Surabaya)

Gambar 5.17 Form Rekanan

Form Rekanan ini digunakan untuk mengelola data rekanan yang melakukan kontrak dengan PT BPT. Karyawan *outsourcing* yang dikelola oleh PT BPT akan ditempatkan pada rekanan-rekanan sesuai dengan profesi yang diambil oleh karyawan *outsourcing*. Level yang ada pada form aplikasi ini adalah *manager* dan staff keuangan.

## f. Form Gaji Pokok

**INPUT GAJI POKOK**

NAMA REKANAN

NAMA PROFESI

GAJI POKOK (RP.)

**GRIDVIEW GAJI POKOK**

ID	NAMA PROFESI	NAMA REKANAN	JUMLAH GAJI POKOK (RP.)
1	Office Boy Koord.	PT. MNA	1,450,000
2	Office Boy	PT. MNA	1,400,000
3	Sapu Jalan	PT. MNA	1,400,000
4	Operator derek	PT. KURNIA MEGA	1,600,000
5	Satpam 2	PT. HUSADA ADI UTAMA	1,900,000

This Program Created By : Zayed Elfasa (STIKOM Surabaya)

Gambar 5.18 *Form Gaji Pokok*

Form Gaji Pokok digunakan untuk mengelola gaji pokok dimana gaji pokok disesuaikan dengan nama profesi yang ada di setiap rekanan. Level yang ada pada form aplikasi ini adalah *manager* dan staff keuangan

### g. Form Tunjangan & Profesi

PT. KURNIA MEGA Operator derek TAMPILKAN -

ID	NAMA REKANAN	NAMA PROFESI	NAMA TUNJANGAN	JUMLAH TUNJANGAN	TERGANTUNG KEHADIRAN
1	PT. MNA	Office Boy Koord.	Tunjangan Hadir / Trans...	3,500	YA
2	PT. MNA	Office Boy Koord.	Tunjangan shift	1,500	YA
3	PT. MNA	Office Boy Koord.	Tunjangan Nutrisi / Mak...	3,000	YA
4	PT. MNA	Office Boy	Tunjangan Hadir / Trans...	3,400	YA
5	PT. MNA	Office Boy	Tunjangan shift	1,500	YA
6	PT. MNA	Office Boy	Tunjangan Nutrisi / Mak...	3,000	YA
7	PT. MNA	Sapu Jalan	Tunjangan Hadir / Trans...	3,400	YA
8	PT. MNA	Sapu Jalan	Tunjangan shift	1,500	YA
9	PT. MNA	Sapu Jalan	Tunjangan Nutrisi / Mak...	30,000	TIDAK

This Program Created By : Zayed Elfasa (STIKOM Surabaya)

Gambar 5.19 Form Tunjangan & Profesi

Form Tunjangan ini digunakan untuk mengelola tunjangan dari karyawan *outsourcing* yang dikelola oleh PT BPT. Tunjangan disesuaikan dengan nama profesi dan rekanan yang ditempati oleh karyawan *outsourcing*. Level yang ada pada form aplikasi ini adalah *manager* dan *staff* keuangan

## h. Form Karyawan OS (*Outsourcing*)

**INPUT KARYAWAN OS**

NIK KARYAWAN  \*

NAMA KARYAWAN  \*

ALAMAT  \*

NO TELP  \*

TANGGAL LAHIR 30 Maret 2015  \*

STATUS AKTIF  AKTIF  TIDAK AKTIF

PROFESI Operator derek

REKANAN PT. KURNIA MEGA

TGL BEKERJA 30 Maret 2015  \*

TGL SELESAI BEKERJA 30 Maret 2015  \*

SAVE EDIT DELETE

**GRIDVIEW KARYAWAN OS**

NIK	NAMA KAR...	ALAMAT	NO TELP	TGL LAHIR	STATUS	NAMA PRO...	NAMA REK...	TGL MULAI...	TGL SELESAI
03.06.11.0...	Nuryanto	---	08098999	11/04/189...	AKTIF	Office Boy...	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Achmad M...	---	08099000	12/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Hariyanto	---	08099001	13/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Fendik Isty...	---	08099002	14/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Akhmad Ir...	---	08099003	15/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Nurdi Kristi...	---	08099004	16/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Bagus Sudi...	---	08099005	17/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Akmad Had...	---	08099006	18/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Dwi Januardi	---	08099007	19/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Suyandik	---	08099008	20/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Muchlisin	---	08099009	21/04/189...	AKTIF	Sapu Jalan	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Edi Rusdianto	---	08099010	22/04/189...	AKTIF	Sapu Jalan	PT. MNA	11/04/201...	11/04/2015...
03.06.11.0...	Abdul Rozaq	---	08099011	23/04/189...	AKTIF	Office Boy	PT. MNA	11/04/201...	11/04/2015...

This Program Created By : Zayed Elfasa (STIKOM Surabaya)

Gambar 5.20 Form Karyawan OS

Form Karyawan OS digunakan untuk mengelola karyawan *outsourcing* yang akan ditempatkan pada rekanan dan disesuaikan dengan profesinya. Karyawan *outsourcing* ini akan dikontrak oleh rekanan dan di-*input*-kan ke dalam aplikasi. Level yang ada pada form aplikasi ini adalah *manager* dan *staff* keuangan.

## i. Form Penggajian

**INPUT DATA PENGGAJIAN**

DETIIL KARYAWAN

NIK	NAMA KARYAWAN	NAMA PROFESI	NAMA REKANAN
03.06.11.0002	Achmad Mucharror	Office Boy	PT. MNA
03.06.11.0002	Achmad Mucharror	Office Boy	PT. MNA

Starts with([NAMA KARYAWAN], 'Achmad Mucharror') And Starts with([NIK], '03.06.11.0002') And Starts with([NAMA PROFESI], 'Office Boy') And Starts with([NAMA REKANAN], 'PT. MNA')

GAJI POKOK

JAMSOSTEK (%)  **Rp. 35,000** TANGGAL GAJIAN

DETIIL TUNJANGAN

NAMA TUNJANGAN	TARIF TUNJANGAN	TOTAL	STATUS
Tunjangan Hadir / Tr...	3,400	95,200	YA
Tunjangan shift	1,500	42,000	YA

**TOTAL TUNJANGAN RP. 221,200**

**TOTAL GAJI + TUNJANGAN (RP.)**

**POTONGAN**

JUMLAH PINJAMAN	PAYROLL	ADMINISTRASI	TOTAL PINJAMAN
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

ANGSURAN  SISA

TOTAL POTONGAN RP. 41,000

LEMBUR  JUMLAH  TOTAL LEMBUR **15,000**

TAMBAHAN GAJI  KETERANGAN TAMBAHAN GAJI

TOTAL PENERIMAAN **RP. 1,595,200**

Gambar 5.21 Form Penggajian

Form penggajian ini digunakan untuk mengelola transaksi penggajian dari karyawan *outsourcing*. Dalam form ini terdapat detil karyawan, detil tunjangan, gaji pokok, pinjaman, dan potongan-potongan gaji. Level yang ada pada form aplikasi ini adalah *manager* dan *staff* keuangan

## j. Form View Penggajian

NAMA REKANAN

PROFESI

BULAN  TAHUN

NIK

NIK KARYAWAN	NAMA KARYAWAN	GAJI POKOK	TUNJANGAN HADIR / TRANSPORT	TUNJANGAN SHIFT	TUNJANGAN NUTRISI / MAKANAN	TOTAL TUNJANGAN	TOTAL GAJI POKOK & TUNJANGAN	ANGSURAN	SISA	JAMSOSTEK	TIDAK MASUK	TOTAL TIDAK MASUK	PEI
03.06.11.0007	Bagus Sudiro B.	1,400,000	98,600	43,500	87,000	229,100	1,629,100	0	0	350,000	1	2,000	0
03.06.11.0013	Abdul Rozaq	1,400,000	98,600	43,500	87,000	229,100	1,629,100	0	0	35,000	1	1,000	0
03.06.11.0008	Akmad Hadi S.	1,400,000	95,200	42,000	84,000	221,200	1,621,200	0	0	35,000	2	6,000	0
03.06.11.0002	Achmad Mucharror	1,400,000	95,200	42,000	84,000	221,200	1,621,200	0	0	35,000	2	6,000	0

	TOTAL GAJI POKOK & TUNJANGAN	ANGSURAN	SISA	JAMSOSTEK	TIDAK MASUK	TOTAL TIDAK MASUK	PENALTI	TOTAL PENALTI	TOTAL POTONGAN	TOTAL LEMBUR	TAMBAHAN GAJI	KETERANGAN TAMBAHAN GAJI	TOTAL PENERIMAAN
	1.629.100	0	0	350.000	1	2.000	0	0	352.000	0	0	---	1.219.825
	1.629.100	0	0	35.000	1	1.000	0	0	36.000	75.000	0	---	1.668.100
	1.621.200	0	0	35.000	2	6.000	0	0	41.000	15.000	0	---	1.595.200
	1.621.200	0	0	35.000	2	6.000	0	0	41.000	15.000	0	---	1.595.200

Gambar 5.22 Form View Penggajian

Form View Penggajian digunakan untuk menampilkan data penggajian karyawan *outsourcing* PT BPT. Data penggajian akan tampil sesuai dengan nama rekanan dan profesi yang diambil oleh karyawan *outsourcing*. Untuk melihat data penggajian pengguna dapat memilih nama rekanan, profesi, bulan penggajian, dan tahun penggajian kemudian klik proses maka aplikasi akan menampilkan data penggajian.

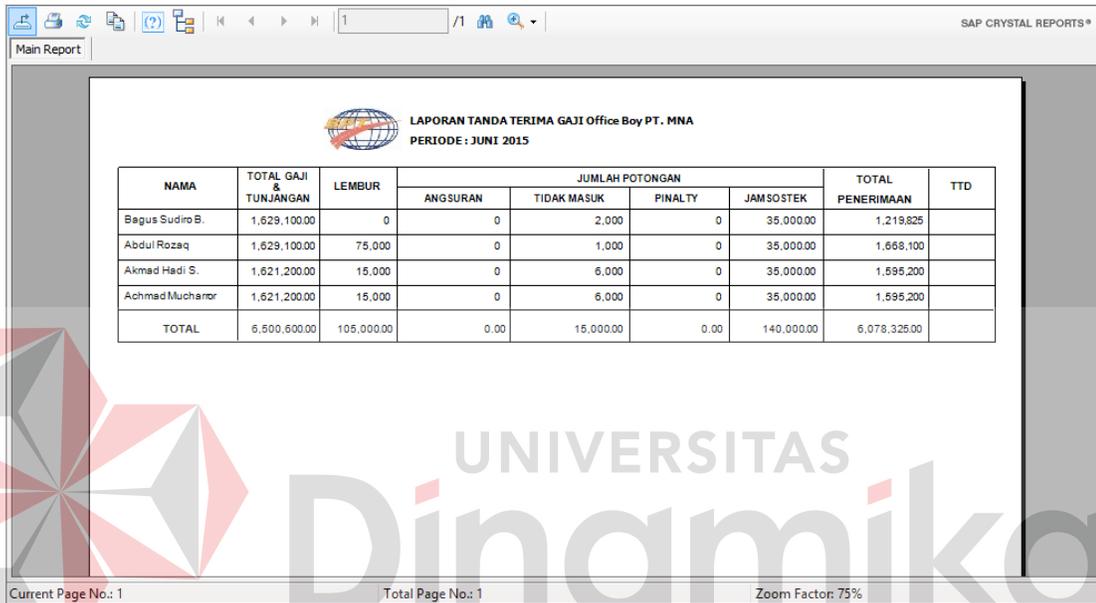
### k. Laporan Slip Gaji

I Penerimaan		II Potongan	
<b>Gaji</b>			
Gaji Pokok	: 1,400,000	Tidak Masuk	: 6,000
		Pinjaman	: 0,00
		Jamsostek	: 35,000,00
		Penalti	: 0
		Lain-lain (payroll)	: 0
<b>Tunjangan</b>			
Tunjangan Hadir / Transport	: 95,200,00		
Tunjangan shift	: 42,000,00		
Tunjangan Nutrisi / Makanan	: 84,000,00		
III Lembur 1 X 15,000	: 15,000		
IV Lain-lain	: 0		
<b>Total Gaji + Tunj. + Lembur + Lainnya</b>	<b>: 1,636,200.00</b>	<b>Total Potongan</b>	<b>: 41,000.00</b>
<b>V Penerimaan Bersih</b>	<b>: 1,595,200</b>		

Gambar 5.23 Form Laporan Slip Gaji

Laporan slip gaji ini akan diberikan pada karyawan *outsourcing*. Pengguna aplikasi ini meng-*input*-kan nama rekanan, profesi, NIK, nama, bulan, dan tahun penggajian maka aplikasi akan menampilkan slip gaji sesuai dengan *input* pengguna.

### 1. Laporan Tanda Terima Gaji



**LAPORAN TANDA TERIMA GAJI Office Boy PT. MNA**  
PERIODE : JUNI 2015

NAMA	TOTAL GAJI & TUNJANGAN	LEMBUR	JUMLAH POTONGAN				TOTAL PENERIMAAN	TTD
			ANGSURAN	TIDAK MASUK	PINALTY	JAMSOSTEK		
Bagus Sudiro B.	1,629,100.00	0	0	2,000	0	35,000.00	1,219,825	
Abdul Rozaq	1,629,100.00	75,000	0	1,000	0	35,000.00	1,668,100	
Akmad Hadi S.	1,621,200.00	15,000	0	6,000	0	35,000.00	1,595,200	
Achmad Muchanor	1,621,200.00	15,000	0	6,000	0	35,000.00	1,595,200	
<b>TOTAL</b>	<b>6,500,600.00</b>	<b>105,000.00</b>	<b>0.00</b>	<b>15,000.00</b>	<b>0.00</b>	<b>140,000.00</b>	<b>6,078,325.00</b>	

Current Page No.: 1      Total Page No.: 1      Zoom Factor: 75%

Gambar 5.24 *Form* Laporan Tanda Terima Gaji

Laporan tanda terima gaji ini akan diberikan kepada pimpinan dan rekanan. Aplikasi akan mencetak laporan tanda terima gaji ketika pengguna meng-*input*-kan nama rekanan, profesi, bulan, dan tahun penggajian.

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

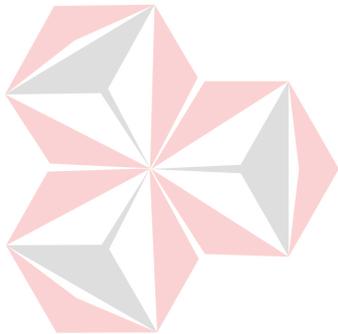
Berdasarkan proses perancangan dan implementasi Aplikasi Penggajian pada PT Bangun Persada Transportation (BPT), maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Dengan adanya aplikasi penggajian ini, maka proses pengelolaan data karyawan *outsourcing*, profesi, rekanan, gaji pokok, tunjangan & profesi, penggajian dan laporan-laporannya menjadi lebih cepat dengan tampilan yang sesuai dengan pengguna, serta proses transaksi dapat dilakukan dengan lebih efektif dan akurat, sehingga bisa mengurangi kesalahan atau *error* yang terjadi.
2. Aplikasi penggajian ini mampu menghasilkan laporan slip gaji per karyawan dan laporan tanda terima gaji per rekanan secara cepat, mampu menghitung penggajian secara akurat, sehingga informasi yang dihasilkan bisa meningkatkan kinerja pengguna aplikasi PT BPT.

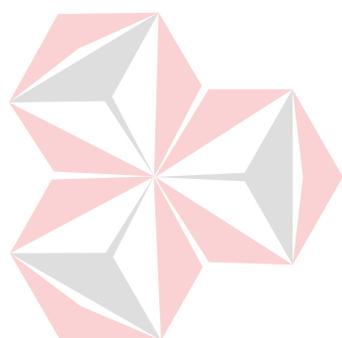
## 6.2 Saran

Apabila aplikasi absensi dan penggajian ini digunakan pada perusahaan, maka disarankan beberapa hal sebagai berikut:

1. Diperlukan infrastruktur komputer yang tepat untuk menjalankan aplikasi penggajian ini.
2. Pengguna aplikasi harus mengikuti prosedur yang telah ditentukan oleh manajemen agar implementasi aplikasi penggajian ini berjalan dengan baik.
3. Aplikasi ini dapat dikembangkan dengan memasukkan fasilitas perhitungan PPh pasal 21.



UNIVERSITAS  
**Dinamika**



UNIVERSITAS  
**Dinamika**

## DAFTAR PUSTAKA

Britton, Carol; Jill Doake. 2001. *Object-Oriented Systems Development*. McGraw-Hill. hlm. 27-34, 268.

Damanik, Sehat. 2006. *Outsourcing & Perjanjian Kerja: Menurut UU No. 13 Tahun 2003 tentang Ketenagakerjaan (Sebagai Penuntun untuk Merencanakan – Melaksanakan Bisnis Outsourcing dan Perjanjian Kerja)*. DSS Publishing : Jakarta.

Darmawan H., Erico & Risal, Laurentius. 2011. *Pemrograman Berorientasi Objek C#*. Informatika: Bandung.

Herlambang, Soendoro, dan Tanuwijaya, Haryanto. 2005. *Sistem Informasi: Konsep, Teknologi, dan Manajemen*. Graha Ilmu : Yogyakarta.

Horngren, Harrison dan Bamber. *Akuntansi* Alih bahasa: Barlian muhammad. 2006 Edisi keenam jakarta, Indeks

Jogiyanto. 2006. *Analisa & Desain Sistem Informasi*. Yogyakarta : Andi Yogyakarta.

Kendall, dan Kendall. 2003. *Analisis dan Perancangan Sistem Jilid 1*. Jakarta : Prenhallindo.

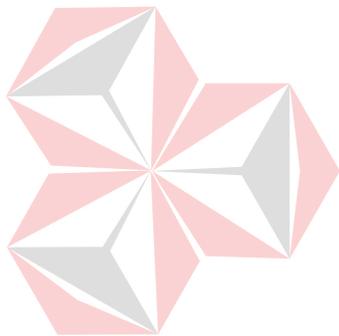
Marlinda, Linda. 2004. *Sistem Basis Data*. Yogyakarta : Andi Ofset.

Schmuller, Joseph. 2004. *SAMS Teach Yourself UML in 24 Hours Third Edition*. Indianapolis : Sams Publishing.

Sholiq. 2010. *Analisis dan Perancangan Berorientasi Objek*. Bandung: CV. Muara Indah.

Warren, Reeve, dan Fess. 2006. *Accounting “Pengantar Akuntansi”*, buku 2, edisi 21. Salemba Empat : Jakarta.

Yuswanto, Subari. 2007. *Pemrograman Database Visual Basic.Net*. Surabaya : Prestasi Pustaka Publisher.



UNIVERSITAS  
**Dinamika**