

No dok : 6506

D 658.32 Nur . P

SUMBANGAN ALUMNI

**PERANCANGAN SISTEM INFORMASI
PENGHITUNGAN UPAH PEGAWAI HARIAN PADA
PT.KOREA NANO TECH INDONESIA**

TUGAS AKHIR

Diajukan Untuk Memenuhi Sebagian Persyaratan Penyelesaian Program Sarjana
Terapan Program Studi Sistem Informasi Industri Otomotif
pada Politeknik STMI Jakarta

OLEH

PRIDI NURWANTO

1315018



DATA BUKU PERPUSTAKAAN	
Tgl Terima	24/07/22
No Induk Buku	494/5110/SB/TA/22

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI**

JAKARTA

2019

POLITEKNIK STMI JAKARTA
KEMENTRIAN PERINDUSTRIAN REPUBLIK INDONESIA

TANDA PERSETUJUAN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR:

**PERANCANGAN SISTEM INFORMASI PENGHITUNGAN UPAH
PEGAWAI HARIAN PADA PT KOREA NANO TECH INDONESIA**

Disusun Oleh:

Nama : Pridi Nurwanto
NIM : 1315018
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 21 Agustus 2019
Tanggal Sidang : 17 September 2019
Tanggal Lulus : 17 September 2019

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam
Ujian Tugas Akhir Politeknik STMI Jakarta

Jakarta, 20 September 2019



Lucky Heriyanto, S.T., M.T.I
NIP : 197908202009011009

POLITEKNIK STMI JAKARTA
KEMENTRIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

JUDUL TUGAS AKHIR:

**PERANCANGAN SISTEM INFORMASI PENGHITUNGAN UPAH
PEGAWAI HARIAN PADA PT KOREA NANO TECH INDONESIA**

Disusun Oleh:

Nama : Pridi Nurwanto

NIM : 1315018

Program Studi : Sistem Informasi Industri Otomotif

Telah Diuji Oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia Pada Hari Selasa Tanggal 17 September 2019.

Jakarta, 20 September 2019

Dosen Pembimbing



Lucky Heriyanto, S.T., M.T.I
NIP : 197908202009011009

Ketua Penguji



Fifi L. Hadianastuti, S.Kom, M.Kes
NIP : 197310162005022001

Dosen Penguji



Dedy Trisanto, S.Kom, MMSI
NIP : 197805052005021002

Dosen Penguji



Triana Fatmawati, S.T., M.T.
NIP : 198005142005022001

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI**

TANDA PERSETUJUAN DOSEN PEMBIMBING

TUGAS AKHIR :

**PERANCANGAN SISTEM INFORMASI PENGHITUNGAN UPAH
PEGAWAI HARIAN PADA PT. KOREA NANO TECH INDONESIA**

DISUSUN OLEH

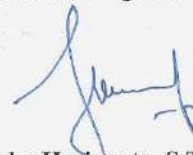
NAMA : PRIDI NURWANTO

NIM : 1315018

PROGRAM STUDI : SISTEM INFORMASI INDUSTRI OTOMOTIF

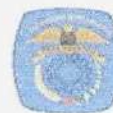
Telah diperiksa dan disetujui untuk memenuhi sebagian syarat penyelesaian Program Sarjana Terapan Program Studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Jakarta, 9 Agustus 2019



Lucky Heriyanto, S.T., M.T.I.

NIP.197908202009011009



LEMBAR BIMBINGAN PENYUSUNAN LAPORAN TUGAS AKHIR

Nama : Pridi Nurwanto
 NIM : 1315018
 Judul Laporan TA : Perancangan Sistem Informasi Penghitungan Upah
 Pegawai Harian Pada PT. Korea Nano Tech Indonesia
 Pembimbing : Lucky Heriyanto, S.T., M.T.I.

Tanggal	BAB	Keterangan	Paraf
28 Mei 2019	I & II	Bimbingan Bab I dan II	f
17 Juni 2019	I, II & III	Acc Bab I, II & Bimbingan Bab III	f
24 Juni 2019	III & IV	Acc Bab III & Bimbingan Bab IV	f
28 Juli 2019	IV	Bimbingan Bab IV	f
2 Juli 2019	IV & V	Revisi Bab IV, Bimbingan Bab V	f
9 Juli 2019	IV & V	Acc Bab IV dan Revisi V	f
16 Juli 2019	V	Revisi Bab V	f
23 Juli 2019	V	Revisi Bab V	f
29 Juli 2019	V	Revisi Bab V	f
5 Agustus 2019	V & VI	Acc Bab V & Masuk Bab IV	f
8 Agustus 2019	VI	Acc Seminar	f

Mengetahui,

Ketua Program Studi
 Sistem Informasi Industri
 Otomotif

Pembimbing

a.n. 

Noveriza Yuliasari, Ssi., MT
 NIP : 195404281986031002



Lucky Heriyanto, S.T., M.T.I.
 NIP : 197908202009011009

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Pridi Nurwanto

NIM : 1315018

Berstatus sebagai mahasiswa Program Studi Sistem Informasi Industri Otomotif di Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul: "PERANCANGAN SISTEM INFORMASI PENGHITUNGAN UPAH PEGAWAI HARIAN PADA PT KOREA NANO TECH INDONESIA"

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survey lapangan, dibantu oleh dosen pembimbing, serta buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya Tugas Akhir ini.
- **Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- **Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini. Jika terbukti saya tidak memenuhi apa yang telah saya nyatakan di atas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai dengan peraturan yang berlaku.

Jakarta, 11 Agustus 2019



Pridi Nurwanto

ABSTRAK

Penghitungan upah pegawai merupakan hal yang sangat penting dalam kelangsungan berjalannya suatu perusahaan. Banyaknya pegawai khususnya pegawai harian membuat divisi HRD harus bekerja keras dalam melakukan proses penghitungan mulai dari kecepatan waktu dan keakuratan penghitungan. Dari hal tersebut, diperoleh permasalahan seperti belum adanya sistem yang dapat mengatur proses penghitungan upah pegawai harian yang berada di PT Korea Nano Tech Indonesia. Proses perhitungan yang masih menggunakan aplikasi standar membuat terhambatnya proses penghitungan ditambah dengan banyaknya pegawai harian yang mencapai 555 orang (periode 2018). Untuk memecahkan masalah tersebut, dibutuhkan sistem yang dapat menghitung upah pegawai secara otomatis. Sistem informasi ini dibuat menggunakan metode pengembangan sistem *waterfall*. Metode ini diawali dari identifikasi kebutuhan, analisis kebutuhan, merancang sistem dengan UML, merancang data dengan ERD, dan melakukan perancangan antar muka. Sistem ini dibangun menggunakan MySQL 5.0.12 dan *Java Standard Edition*. Dengan demikian dapat disimpulkan Divisi HRD PT. Korea Nano Tech Indonesia memiliki permasalahan dalam penghitungan gaji pegawai harian, hal tersebut dikarenakan banyaknya pegawai harian yang berkerja di dalam perusahaan. Berdasarkan permasalahan ini dibangun dan dirancang sistem aplikasi untuk penghitungan upah pegawai harian agar menjadi solusi yang tepat.

Kata Kunci : Sistem Informasi, Penghitungan Upah, Pegawai Harian, *Waterfall*, Penjadwalan Produksi, MySQL 5.0.12, *Java Standard Edition*.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir dengan judul **“Perancangan Sistem Informasi Penghitungan Upah Pegawai Harian Pada PT.Korea Nano Tech Indonesia”**.

Tugas akhir ini disusun guna memenuhi sebagian syarat yang harus dipenuhi dalam menempuh jenjang Sarjana Terapan program studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Dalam penyusunan laporan tugas akhir ini, penulis banyak mendapat bantuan serta bimbingan dari berbagai pihak. Pada kesempatan ini tidak lupa penulis sampaikan ucapan terima kasih kepada:

1. Allah Yang Maha Esa atas rahmat serta kemudahan yang diberikan.
2. Kedua orang tua serta keluarga tercinta yang telah memberikan do'a, dukungan, pengorbanan, semangat dan kasih sayang hingga saat ini.
3. Bapak Dr. Mustofa, S.T., M.T., selaku Direktur Politeknik STMI Jakarta.
4. Ibu Noveriza Yuliasari, Ssi., MT selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
5. Bapak Lucky Heriyanto, S.T., M.T.I., selaku dosen pembimbing yang telah memberikan arahan dan masukan dalam penyusunan Tugas Akhir.
6. Bapak Adi Priatna. selaku Manajer PT. Korea Nano Tech Indonesia.
7. Ibu Elida selaku supervisor HRD PT. Korea Nano Tech Indonesia yang telah membantu memberi arahan serta informasi untuk pengumpulan data.
8. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
9. Teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi Industri Otomotif terutama SA01 2015 atas kebersamaan, solidaritas, canda tawa dan motivasinya selama ini.

10. Teman-teman seperjuangan praktek kerja lapangan dimanapun terimakasih atas semangat, kebersamaan, solidaritas dan canda tawa.
11. Teman-teman seperjuangan SMG squad dan Warung empok yang selalu memberikan *support*, solidaritas tanpa batas dan membantu satu sama lain dalam mengerjakan tugas akhir ini.

Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan tugas akhir ini dengan melimpahkan rahmat dan karunia-Nya. Besar harapan Penulis bahwa laporan ini dapat memberikan manfaat serta pengetahuan bagi pembacanya. Terima kasih.

Jakarta, 8 Agustus 2019

Penulis,
Pridi Nurwanto

DAFTAR ISI

Judul Tugas Akhir	i
Lembar Persetujuan Dosen Pembimbing	ii
Lembar Bimbingan Tugas Akhir	iii
Lembar Pernyataan Keaslian.....	iv
Abstrak	v
Kata Pengantar	vi
Daftar Isi.....	viii
Daftar Gambar.....	xii
Daftar Tabel	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Pokok Permasalahan.....	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Manfaat Tugas Akhir	4
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI.....	6
2.1 Hakikat Sistem.....	6
2.1.1 Pengertian Sistem	6
2.1.2 Karakteristik Sistem	6
2.2 Klasifikasi Sistem	7
2.3 Hakikat Informasi	9
2.3.1 Pengertian Informasi.....	9
2.3.2 Fungsi Informasi.....	9
2.3.3 Siklus Informasi.....	9
2.3.4 Nilai Informasi.....	10
2.3.5 Kualitas Informasi	12
2.4 Pengertian Sistem Informasi.....	12
2.4.1 Komponen Informasi	13

2.5	Konsep Dasar Analisis Sistem.....	14
2.6	Sistem Pendukung Keputusan	15
2.6.1	Karakteristik Sistem Pendukung Keputusan	16
2.6.2	Tujuan Sistem Pendukung Keputusan.....	17
2.7	<i>System Development Life Cycle</i>	17
2.8	Model <i>Water Fall</i>	21
2.9	Kelebihan Model <i>WaterFall</i>	24
2.10	Definisi Absensi.....	24
2.10.1	Jenis-jenis Absensi	24
2.10.2	Pengelolaan Absensi.....	24
2.11	Pengertian Presensi	25
2.12	Definisi Jam Lembur	25
2.13	Definisi Gaji.....	26
2.14	Diagram Bagan Alis(<i>Flowmap</i>).....	26
2.15	Unified Modelling Language.....	29
2.15.1	<i>Use Case</i> Diagram	31
2.15.2	<i>Use Case Description</i>	32
2.15.3	<i>Activity</i> Diagram	33
2.15.4	<i>Sequance</i> Diagram	34
2.15.5	<i>Class</i> Diagram	35
2.15.6	<i>Deployment</i> Diagram	35
2.16	<i>Entity Relationship</i> Diagram.....	36
2.17	XAMPP	37
2.18	MYSQL	38
2.19	Tipe Data MySql.....	39
2.10	<i>Blackbox Testing</i>	40
2.21	<i>Java Standard Edition</i>	40
2.22	Mesin Absensi Kartu	41
2.23	Pengertian Time Card	41
BAB III	METODOLOGI PENELITIAN	42
3.1	Metodologi Penelitian.....	42

3.2	Pengumpulan Data.....	42
3.3	Metode Pengumpulan Data.....	43
3.4	Metode Pengembangan Sistem.....	43
3.5	<i>Literature Review</i>	45
3.6	Kerangka Pemecahan Masalah	48
BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA.....	51
4.1	Latar Belakang	51
4.2	Profil Perusahaan	52
4.3	Logo Perusahaan.....	52
4.4	Visi dan Misi Perusahaan	53
4.5	Kebijakan Mutu Perusahaan	53
4.6	Struktur Organisasi Perusahaan	54
4.6.1	Tugas dan Wewenang Setiap Divisi.....	54
4.7	Jumlah Karyawan dan Jam Kerja	56
4.8	Mesin Produksi	58
4.9	Produk Yang Dihasilkan.....	61
4.10	Jaringan Pemasaran	64
4.11	Jaringan Pemasok	65
4.12	Divisi HRD	65
4.13	Pengertian HRD.....	67
4.14	Struktur Organisasi HRD dan <i>Accounting</i>	67
4.15	Pengertian Time Card	68
4.16	Mesin Absensi Kartu	70
4.17	Proses Absensi Karyawan.....	71
4.18	Rekap Absen	73
4.18.1	Data Rekap Absen	74
4.19	Proses Upah Pegawai.....	74
4.20	Analisis Permasalahan	77
BAB V	ANALISIS DAN PEMBAHASAN.....	81
5.1	<i>System Request</i>	81
5.2	Proses Usulan Pengupahan Pegawai	81

5.3	Analisis Kebutuhan Sistem.....	83
5.3.1	<i>Functional Requirement</i>	85
5.3.2	<i>Non Functional Requirement</i>	86
5.4	Use Case Diagram	86
5.4.1	<i>Use Case Description</i>	87
5.5	Activity Diagram	91
5.6	Sequence Diagram	97
5.7	Class Diagram	103
5.8	Deployment Diagram	105
5.9	Windows Navigation Diagram.....	105
5.10	Entity Relationship Diagram.....	106
5.10.1	Kamus Data	107
5.11	Perancangan <i>Interface</i>	109
BAB VI	PENUTUP	115
6.1	Kesimpulan	115
6.2	Saran	115
	DAFTAR PUSTAKA	116
	DAFTAR LAMPIRAN	118

DAFTAR GAMBAR

Gambar II.1	Metode <i>Waterfall</i>	22
Gambar III.1	Kerangka Penelitian	50
Gambar IV.1	Logo PT. KOREA NANO TECH INDONESIA.....	52
Gambar IV.2	Kebijakan Mutu PT. Korea Nano Tech Indonesia.....	53
Gambar IV.3	Struktur Organisasi Perusahaan	54
Gambar IV.4	Struktur Organisasi HRD.....	68
Gambar IV.5	<i>Time Card</i> PT. KNT INDONESIA	69
Gambar IV.6	Rekonstuksi isi <i>Time Card</i> PT. KNT INDONESIA	70
Gambar IV.7	Mesin Absensi Kartu	71
Gambar IV.8	<i>Flowmap</i> Proses Absensi Karyawan	72
Gambar IV.9	<i>Flowmap</i> Rekap Daftar Hadir.....	73
Gambar IV.10	Rekap Daftar Hadir Pegawai Harian	74
Gambar IV.11	<i>Flowmap</i> Proses Upah Pegawai.....	75
Gambar IV.12	Rekonstruksi Laporan Upah Pegawai Harian.....	76
Gambar V.1	Flowmap Usulan Proses Penggajian Pegawai	82
Gambar V.2	<i>Use Case Diagram</i> Usulan	87
Gambar V.3	<i>Activity Diagram Login</i>	92
Gambar V.4	<i>Activity Diagram</i> Data Daftar Hadir dan Jam Lembur ...	93
Gambar V.5	<i>Activity Diagram</i> Data Penghitungan Upah	94
Gambar V.6	<i>Activity Diagram</i> Menghitung Upah Pegawai	95
Gambar V.7	<i>Activity Diagram</i> Evaluasi	96
Gambar V.8	<i>Activity</i> Mengelola Laporan Upah Pegawai	97
Gambar V.9	<i>Sequence Diagram Login</i>	98
Gambar V.10	<i>Sequence</i> Mengelola Daftar Hadir dan Jam Lembur	99
Gambar V.11	<i>Sequence</i> Mengelola Upah Pegawai	100
Gambar V.12	<i>Sequence</i> Menghitung Upah Pegawai	101
Gambar V.13	<i>Sequence</i> Evaluasi	102
Gambar V.14	<i>Sequence</i> Mengelola Laporan Upah Pegawai.....	103
Gambar V.15	<i>Class Diagram</i> Menghitung Upah Pegawai.....	104

Gambar V.16	<i>Deployment</i> Diagram Menghitung Upah Pegawai	105
Gambar V.17	<i>Windows Navigation</i> Diagram Upah Pegawai Harian....	106
Gambar V.18	<i>Entity Relationship</i> Diagram Upah Pegawai Harian	107
Gambar V.19	Rancangan <i>Form</i> Login	110
Gambar V.20	Rancangan Menu Utama.....	111
Gambar V.21	Rancangan Panel Rekap	111
Gambar V.22	Rancangan Panel Upah	112
Gambar V.23	Rancangan Panel Evaluasi	113
Gambar V.24	Rancangan Panel Report.....	113

DAFTAR TABEL

Tabel II.1	Simbol <i>Flow Direction</i>	27
Tabel II.2	Simbol Proses	27
Tabel II.3	Simbol <i>Input</i> dan <i>Output</i>	28
Tabel II.4	Jenis Diagram Pada UML dan fungsi-fungsinya.....	30
Tabel II.5	Simbol-Simbol Use Case Diagram.....	31
Tabel II.6	Simbol-Simbol Activity Diagram.....	33
Tabel II.7	Simbol-Simbol Sequence Diagram.....	34
Tabel II.8	Simbol-Simbol Class Diagram	35
Tabel II.9	Simbol-simbol <i>Deployment Diagram</i>	36
Tabel II.10	Simbol-simbol <i>ERD</i>	37
Tabel II.11	Tipe Data MySQL	39
Tabel III.1	Literature Review	46
Tabel VI.1	Jumlah Karyawan	57
Tabel VI.2	Waktu Kerja Bagian Produksi Shift 1	57
Tabel VI.3	Waktu Kerja Bagian Produksi Shift 2	57
Tabel VI.4	Waktu Kerja Bagian Produksi Shift 3	58
Tabel VI.5	Waktu Kerja bagian Kantor	58
Tabel VI.6	<i>Injection Machine</i>	58
Tabel VI.7	<i>Molding Machine</i>	59
Tabel VI.8	<i>Q/C EQUIPMENT MACHINE</i>	60
Tabel VI.9	Produk-produk Otomotif	61
Tabel VI.10	Daftar nama <i>customer</i>	64
Tabel VI.11	Daftar nama <i>Supplier</i>	65
Tabel VI.12	Analisis Permasalahan	77
Tabel V.1	<i>System Request</i>	81
Tabel V.2	Analisis Kebutuhan Sistem.....	83
Tabel V.3	<i>Case Description Login</i>	85
Tabel V.4	<i>Case Description</i> Data Daftar Hadir dan Jam Lembur...	88

Tabel V.5	<i>Case Description</i> Mengelola Upah Pegawai	89
Tabel V.6	<i>Case Description</i> Menghitung Upah Pegawai.....	89
Tabel V.7	<i>Case Description</i> Mengelola Laporan	90
Tabel V.8	<i>Case Description</i> Evaluasi.....	91
Tabel V.9	Tabel Data Pegawai	108
Tabel V.10	Tabel Data rekap.....	108
Tabel V.11	Tabel Data upah.....	109

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI**

TANDA PERSETUJUAN DOSEN PEMBIMBING

TUGAS AKHIR :

**PERANCANGAN SISTEM INFORMASI PENGHITUNGAN UPAH
PEGAWAI HARIAN PADA PT. KOREA NANO TECH INDONESIA**

DISUSUN OLEH

NAMA : PRIDI NURWANTO

NIM : 1315018

PROGRAM STUDI : SISTEM INFORMASI INDUSTRI OTOMOTIF

Telah diperiksa dan disetujui untuk memenuhi sebagian syarat penyelesaian Program Sarjana Terapan Program Studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Jakarta, 9 Maret 2019

Lucky Heriyanto, S.T., M.T.I.

NIP.197908202009011009

DAFTAR HADIR BIMBINGAN

TUGAS AKHIR

Nama Dosen : Lucky Heriyanto S.Kom, M.TI
Nama Mahasiswa : Pridi Nurwanto
NIM : 1315018
No.Telepon : 08568539051
Judul : Perancangan Sistem Informasi Perhitungan Upah Pegawai
Harian Pada PT, Korea Nano Tech Indonesia

NO	Tanggal	Kegiatan	Tanda Tangan
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
13			
14			

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Pridi Nurwanto

NIM : 1315018

Berstatus sebagai mahasiswa Program Studi Sistem Informasi Industri Otomotif di Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul: “PERANCANGAN SISTEM INFORMASI PENGHITUNGAN UPAH PEGAWAI HARIAN PADA PT KOREA NANO TECH INDONESIA”

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survey lapangan, dibantu oleh dosen pembimbing, serta buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya Tugas Akhir ini.
- **Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- **Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini. Jika terbukti saya tidak memenuhi apa yang telah saya nyatakan di atas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai dengan peraturan yang berlaku.

Jakarta, 11 Agustus 2019

Pridi Nurwanto

ABSTRAK

Penghitungan upah pegawai merupakan hal yang sangat penting dalam kelangsungan berjalannya suatu perusahaan. Banyaknya pegawai khususnya pegawai harian membuat divisi HRD harus bekerja keras dalam melakukan proses penghitungan mulai dari kecepatan waktu dan keakuratan penghitungan. Dari hal tersebut, diperoleh permasalahan seperti belum adanya sistem yang dapat mengatur proses penghitungan upah pegawai harian yang berada di PT Korea Nano Tech Indonesia. Proses perhitungan yang masih menggunakan aplikasi standar membuat terhambatnya proses penghitungan ditambah dengan banyaknya pegawai harian yang mencapai 555 orang (periode 2018). Untuk memecahkan masalah tersebut, dibutuhkan sistem yang dapat menghitung upah pegawai secara otomatis. Sistem informasi ini dibuat menggunakan metode pengembangan sistem *waterfall*. Metode ini diawali dari identifikasi kebutuhan, analisis kebutuhan, merancang sistem dengan UML, merancang data dengan ERD, dan melakukan perancangan antar muka. Sistem ini dibangun menggunakan MySQL 5.0.12 dan *Java Standard Edition*. Dengan demikian dapat disimpulkan Divisi HRD PT. Korea Nano Tech Indonesia memiliki permasalahan dalam penghitungan gaji pegawai harian, hal tersebut dikarenakan banyaknya pegawai harian yang berkerja di dalam perusahaan. Berdasarkan permasalahan ini dibangun dan dirancang sistem aplikasi untuk penghitungan upah pegawai harian agar menjadi solusi yang tepat.

Kata Kunci : Sistem Informasi, Penghitungan Upah, Pegawai Harian, *Waterfall*, Penjadwalan Produksi, MySQL 5.0.12 , *Java Standard Edition*.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir dengan judul **“Perancangan Sistem Informasi Penghitungan Upah Pegawai Harian Pada PT.Korea Nano Tech Indonesia”**.

Tugas akhir ini disusun guna memenuhi sebagian syarat yang harus dipenuhi dalam menempuh jenjang Sarjana Terapan program studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Dalam penyusunan laporan tugas akhir ini, penulis banyak mendapat bantuan serta bimbingan dari berbagai pihak. Pada kesempatan ini tidak lupa penulis sampaikan ucapan terima kasih kepada:

1. Allah Yang Maha Esa atas rahmat serta kemudahan yang diberikan.
2. Kedua orang tua serta keluarga tercinta yang telah memberikan do'a, dukungan, pengorbanan, semangat dan kasih sayang hingga saat ini.
3. Bapak Dr. Mustofa, S.T., M.T., selaku Direktur Politeknik STMI Jakarta.
4. Ibu Noveriza Yuliasari, Ssi., MT selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
5. Bapak Lucky Heriyanto, S.T., M.T.I., selaku dosen pembimbing yang telah memberikan arahan dan masukan dalam penyusunan Tugas Akhir.
6. Bapak Adi Priatna. selaku Manajer PT. Korea Nano Tech Indonesia.
7. Ibu Elida selaku supervisor HRD PT. Korea Nano Tech Indonesia yang telah membantu memberi arahan serta informasi untuk pengumpulan data.
8. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
9. Teman-teman mahasiswa/i Politeknik STMI Jakarta Program Studi Sistem Informasi Industri Otomotif terutama SA01 2015 atas kebersamaan, solidaritas, canda tawa dan motivasinya selama ini.

10. Teman-teman seperjuangan praktek kerja lapangan dimanapun terimakasih atas semangat, kebersamaan, solidaritas dan canda tawa.
11. Teman-teman seperjuangan SMG squad dan Warung empok yang selalu memberikan *support*, solidaritas tanpa batas dan membantu satu sama lain dalam mengerjakan tugas akhir ini.

Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan tugas akhir ini dengan melimpahkan rahmat dan karunia-Nya. Besar harapan Penulis bahwa laporan ini dapat memberikan manfaat serta pengetahuan bagi pembacanya. Terima kasih.

Jakarta, 8 Agustus 2019

Penulis,
Pridi Nurwanto

**PERANCANGAN SISTEM INFORMASI
PENGHITUNGAN UPAH PEGAWAI HARIAN PADA
PT.KOREA NANO TECH INDONESIA**

TUGAS AKHIR

Diajukan Untuk Memenuhi Sebagian Persyaratan Penyelesaian Program Sarjana
Terapan Program Studi Sistem Informasi Industri Otomotif
pada Politeknik STMI Jakarta

**OLEH
PRIDI NURWANTO
1315018**



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI
JAKARTA
2019**

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem informasi merupakan salah satu komponen penting dalam dunia informasi dan teknologi yang menjadi kebutuhan penting dalam berbagai bidang kehidupan, salah satunya adalah bidang bisnis. Kebutuhan informasi yang aktual serta akurat saat ini sudah menjadi sebuah keperluan yang sangat penting dalam memenangkan persaingan di dalam dunia bisnis. Tanpa disadari, perkembangan teknologi informasi dan kebutuhan informasi telah menjadi sumber daya utama bagi pelaku bisnis yang juga memiliki peranan penting dalam membangun kerajaan bisnis. Kebutuhan informasi juga telah menjadi salah satu sumber daya utama bagi pelaku bisnis yang tidak kalah pentingnya dengan sumber daya yang lain.

Kemampuan untuk mengakses, mengelola data dan menyediakan informasi secara cepat, tepat dan akurat dalam mengatur sumber daya manusia yang ada sudah menjadi salah satu kebutuhan penting bagi pelaku bisnis dan konsumen. Untuk menunjang hal tersebut maka perlu menerapkan sistem informasi yang memiliki integrasi yang baik. Mengelola data penghitungan upah pegawai merupakan salah satu hal yang penting untuk mendukung kelancaran aktivitas suatu bisnis. Sehingga dengan sistem informasi ini diharapkan mampu manajemen seluruh aktifitas penghitungan upah pegawai khususnya pegawai harian.

PT. Korea Nano Tech Indonesia adalah salah satu perusahaan yang bergerak di bidang industri manufaktur otomotif, dimana proses penghitungan upah pegawai masih sedikit memiliki kendala. Dalam penghitungan upah PT. Korea Nano Tech Indonesia menggunakan aplikasi *microsoft excel* sebagai aplikasi utamanya, hal ini sangat rentan terhadap keterlambatan pemberian upah, dikarenakan masih sering terjadinya kesalahan dalam melakukan proses penghitungan upah. Untuk kasus ini divisi HRD dituntut dalam kecepatan dan

keakuratan melakukan penghitungan upah pada pegawai harian, yang mana pegawai harian tersebut mendapatkan upah atau bayaran berdasarkan daftar hadir dan jam lembur, dengan jumlah pegawai harian mencapai 555 orang . Maka dengan ini divisi HRD membutuhkan aplikasi sistem yang lebih cepat dalam penghitungan upah dan lebih akurat untuk mencegah terjadinya keterlambatan dalam penghitungan upah pegawai harian berdasarkan daftar hadir dan jam lembur. Ditambah dengan *database* yang memiliki penyimpanan yang cukup besar, hal ini dibutuhkan untuk menyimpan rekaman daftar upah seluruh pegawai disetiap bulannya, yang mana apabila laporan daftar upah dibutuhkan pihak divisi HRD tak perlu kesulitan dalam mencarinya. Dalam pembuatan aplikasi sistem banyak sekali metode yang sudah tersedia salah satunya dengan menggunakan metode *waterfall* pembuatan aplikasi sistem akan lebih terstruktur mulai dari perencanaan sampai pemeliharaan program, dengan metode ini juga *user* akan dengan mudah *update* program yang lebih baru.

Divisi HRD PT. Korea Nano Tech Indonesia melakukan proses pengupahan berdasarkan pencatatan daftar hadir dan jam lembur pegawai harian. Mulai dari daftar hadir menggunakan mesin *check clock* dan *time card*, melakukan rekap daftar hadir dengan menyortir *time card* karyawan berdasarkan *shift* kerja dan jam lembur, apa bila sudah di sortir dan direkap, lalu dilakukan penghitungan upah menggunakan aplikasi *microsoft excel* dengan memasukan rumus *microsoft excel* sebagai metode penghitungannya dengan demikian divisi HRD mendapatkan hasil yang akurat dalam penghitungan upah seluruh pegawai, akan tetapi proses tersebut membutuhkan waktu yang cukup lama ditambah dengan pegawai harian pada PT. Korea Nano Tech Indonesia yang cukup banyak.

Berdasarkan penjelasan di atas maka diperlukan pengembangan suatu aplikasi yang dapat membantu perusahaan dalam penghitungan upah pegawai berdasarkan daftar hadir dan jam lembur, yang menjadi suatu laporan yang berguna bagi perusahaan. Adapun judul Tugas Akhir ini adalah “PERANCANGAN SISTEM INFORMASI PENGHITUNGAN UPAH PEGAWAI HARIAN PADA PT. KOREA NANO TECH INDONESIA”.

1.2. Pokok Permasalahan

Permasalahan yang terjadi di Divisi *Human Resources Department* pada PT. Korea Nano Tech Indonesia adalah sebagai berikut:

1. Proses atau penginputan data penghitungan upah pegawai harian masih menggunakan aplikasi yang masih standar dan belum terhubung secara otomatis kedalam *database*.
2. Membutuhkan waktu yang cukup lama dalam proses penghitungan upah pegawai, yang dilakukan sebulan sekali di akhir bulan. Sehingga memungkinkan terjadinya keterlambatan pemberian upah khususnya pada pegawai harian.

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian tugas akhir ini yang dilakukan terhadap Divisi *Human Resources Department* pada PT. Korea Nano Tech Indonesia dalam melakukan penghitungan upah pegawai harian adalah untuk merancang dan membangun sistem aplikasi penghitungan upah:

1. Membuat aplikasi yang dapat memudahkan penghitungan upah pegawai harian lebih cepat dan juga akurat dalam penghitungan.
2. Menyediakan fasilitas penyimpanan data pengupahan pegawai dengan menggunakan *database* agar dapat menyimpan rekaman daftar upah dalam satu lokasi dan juga dapat membantu proses pencarian dokumen upah menjadi lebih mudah.
3. Aplikasi yang mudah digunakan serta dapat menghasilkan *output* berupa laporan upah pegawai dan dapat langsung di dokumentasikan.

1.4 Batasan Masalah

Agar dalam penulisan Tugas Akhir ini lebih fokus dan lebih terarah, maka perlu diadakan batasan-batasan sebagai berikut:

1. Tempat melakukan penelitian adalah PT. Korea Nano Tech Indonesia, mulai dari 15 September sampai 24 Desember 2018.

2. Analisis dan penelitian hanya sebatas mengenai informasi penghitungan upah pegawai harian hingga menjadi suatu *report* atau laporan upah pada PT. Korea Nano Tech Indonesia.
3. Menggunakan NetBeans IDE 8.2, MySQL 127.0.0.1 dan Xampp 3.2.4.
4. Sistem ini hanya di rancang bangun tetapi belum di implementasikan.

1.5 Manfaat Tugas Akhir

Adapun manfaat yang dapat diambil dari penelitian pada Tugas Akhir ini adalah sebagai berikut:

1. Membantu mempercepat dan mempermudah proses penghitungan upah pegawai harian berdasarkan daftar hadir dan jam lembur pada PT. Korea Nano Tech Indonesia.
2. Merancang *database* untuk laporan upah pegawai harian.
3. Membantu PT. Korea Nano Tech Indonesia dengan mencegah keterlambatan dalam pengupah pegawai harian.

1.6 Sistematika Penulisan

Tugas Akhir ini disusun secara sistematis guna memberikan gambaran yang jelas mengenai isi dan pembahasan yang ada di dalamnya. Adapun sistematika penulisan Tugas Akhir ini diuraikan dalam enam bab, diantaranya sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini memuat latar belakang, pokok permasalahan, tujuan penelitian, batasan masalah, manfaat tugas akhir, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini membahas tentang berbagai teori yang diperoleh dari buku-buku, literatur, ataupun berbagai macam referensi yang berkaitan dengan tema yang diambil. Teori-teori yang dipaparkan

pada laporan ini adalah seputar sistem informasi, pengendalian kualitas, *Unified Modelling Language* (UML), serta teori-teori lain yang berhubungan dengan perancangan sistem informasi.

BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan metode ilmiah dalam mencari dan menjelaskan kerangka pemecahan masalah yang menguraikan tahap-tahap untuk mengatasi permasalahan yang telah diidentifikasi.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini membahas tentang data yang telah diperoleh selama melaksanakan penelitian di PT. Korea Nano Tech Indonesia seperti alur proses pengupahan berdasarkan daftar hadir, proses bisnis yang berjalan, dan hal lain yang mendukung sistem informasi proses pengupahan.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari analisis kebutuhan sistem, pemodelan sistem dengan *Unified Modelling Language* (UML), pemodelan data dan *Entity Relationship Diagram* (ERD), sampai perancangan *interface*.

BAB VI PENUTUP

Pada bab ini berisikan kesimpulan dari hasil penelitian yang dilakukan, serta saran-saran dalam penerapan sistem informasi rekrutmen karyawan untuk perusahaan dan pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Hakikat Sistem

Kata sistem berasal dari bahasa Yunani, yaitu *systema* yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan.

2.1.1 Pengertian Sistem

Menurut Jogiyanto (2005) mengemukakan bahwa sistem adalah kumpulan dari elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem menggambarkan suatu kejadian dan kesatuan yang nyata seperti tempat, benda, serta orang-orang yang ada dan terjadi.

Sedangkan menurut Ramdhani (2008), sistem adalah hubungan satu unit dengan unit-unit lainnya yang tidak dapat dipisahkan serta menuju suatu kesatuan dalam rangka mencapai tujuan yang telah ditetapkan.

Dengan demikian, sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai tujuan dari sistem tersebut. Maksud dari suatu sistem adalah untuk mencapai tujuan dan sasaran dalam ruang lingkup yang sempit.

2.1.2 Karakteristik Sistem

Karakteristik sistem menurut Nurchmad (2009), yaitu sebagai berikut:

1. Komponen (*Component*)

Segala sesuatu yang menjadi bagian penyusunan sistem. Komponen sistem dapat berupa benda nyata ataupun abstrak. Komponen sistem disebut sebagai subsistem.

2. Batas (*Boundary*)

Diperlukan untuk membedakan satu sistem dengan sistem yang lain. Tanpa adanya batas sistem, sangat sulit untuk memberikan batasan *scope* tinjauan terhadap sistem.

3. Lingkungan (*Environment*)

Segala sesuatu yang berada di luar sistem yang dapat menguntungkan ataupun merugikan. Umumnya lingkungan yang menguntungkan akan selalu dipertahankan untuk menjaga keberlangsungan sistem, sedangkan lingkungan sistem yang merugikan akan diupayakan agar mempunyai pengaruh seminimal mungkin, bahkan ditiadakan.

4. Penghubung (*Interface*)

Segala sesuatu yang berfungsi menjembatani hubungan antarkomponen dalam sistem. Penghubung merupakan sarana setiap komponen saling berinteraksi dan berkomunikasi.

5. Masukan (*Input*)

Merupakan komponen sistem, yaitu segala sesuatu yang perlu dimasukkan ke dalam sistem sebagai bahan yang akan diolah lebih lanjut untuk menghasilkan keluaran (*output*) yang berguna.

6. Pengolahan (*Processing*)

Komponen sistem yang mempunyai peran utama mengolah masukan agar menghasilkan *output* yang berguna bagi para pemakainya.

7. Keluaran (*Output*)

Komponen sistem yang berupa berbagai macam bentuk keluaran yang dihasilkan oleh komponen pengolahan.

8. Sasaran (*Objective*)

Setiap komponen dalam sistem perlu dijaga agar saling bekerja sama agar mampu mencapai sasaran sistem.

9. Kendali (*Control*)

Setiap komponen dalam sistem perlu dijaga agar tetap bekerja sesuai dengan peran dan fungsinya masing-masing.

10. Umpan Balik (*Feedback*)

2.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, di antaranya sebagai berikut:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi, dan sebagainya.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia merupakan sistem yang melibatkan interaksi antara manusia dan mesin, disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*. Sistem informasi akuntansi merupakan contoh *man-machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem Tertentu dan Sistem Tidak Tentu

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi di antara bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tidak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem Tertutup dan Sistem Terbuka

Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem lainnya. Karena sistem bersifat terbuka dan terpengaruh oleh lingkungan luarnya, suatu sistem harus mempunyai sistem pengendalian yang baik.

2.3 Hakikat Informasi

Pada hakikatnya, informasi adalah salah satu sumber utama dari perusahaan dan dapat dikelola seperti halnya sumber-sumber lain. Informasi adalah sumber konseptual yang menggambarkan sumber-sumber fisik yang harus dikelola oleh manajer.

2.3.1 Pengertian Informasi

Menurut Sutabri (2005) menyatakan bahwa informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

Sedangkan menurut Jogiyanto (2005), informasi adalah sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian nyata yang digunakan untuk pengambilan keputusan. Lalu menurut Mcleod (2008), informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Dari pernyataan para ahli mengenai pengertian informasi tersebut, dapat disimpulkan bahwa informasi adalah suatu data atau objek yang diproses terlebih dahulu sedemikian rupa sehingga dapat tersusun dan terklasifikasi dengan baik sehingga memiliki arti bagi penerimanya, yang selanjutnya menjadi pengetahuan bagi penerima tentang suatu hal tertentu yang membantu pengambilan keputusan secara tepat.

2.3.2 Fungsi Informasi

Menurut Sutabri (2005), fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Ketika berbekal informasi seseorang dapat mengambil keputusan dengan baik. Akan tetapi, dalam pengambilan keputusan yang kompleks, informasi hanya dapat menambah kemungkinan kepastian atau mengurangi berbagai macam pilihan.

2.3.3 Siklus Informasi

Pada umumnya banyak bagian informasi yang mungkin berguna dan dengan suatu cara tertentu dapat mempengaruhi tanggapan penerima informasi. Informasi dapat berasal dari pengamatan, percakapan, rapat, media cetak dan elektronik, laporan, dan bahkan dari sistem informasi itu sendiri. Sistem informasi hanya memberikan sebagian dari informasi yang dipergunakan oleh pengambil keputusan dan bahwa sistem informasi ini merupakan informasi formal yang dapat ditentukan banyaknya. Oleh karena itu penentuan banyaknya informasi yang dapat ditangani atau dihasilkan oleh fungsi organisasi sangatlah penting.

Menurut Sutabri (2005), data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk, suara, gambar, dan sebagainya.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.

2.3.4 Nilai Informasi

Menurut Sutabri (2005), nilai dari informasi ditentukan dari dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai apabila manfaat yang diperoleh lebih berharga dibandingkan dengan biaya untuk mendapatkannya. Berikut ini adalah nilai informasi berdasarkan atas 10 (sepuluh) sifat, yaitu:

1. Mudah diperoleh

Sifat ini menunjukkan kemudahan dan kecepatan untuk memperoleh informasi. Kecepatannya dapat diukur, misalnya 1 menit versus 24 jam. Akan tetapi berapa nilainya bagi pemakai informasi sulit untuk mengukurnya.

2. Luas dan lengkap

Sifat ini menunjukkan kelengkapan isi informasi. Hal ini tidak hanya mengenai volumenya, akan tetapi juga mengenai keluaran informasinya. Sifat ini sangat kabur dan karena itu sulit untuk mengukurnya.

3. Ketelitian

Sifat ini berhubungan dengan tingkat kebebasan dari kesalahan keluaran informasi. Pada volume data yang besar biasanya terdapat dua jenis kesalahan, yakni kesalahan pencatatan dan kesalahan perhitungan.

4. Kecocokan

Sifat ini menunjukkan seberapa baik keluaran informasi dalam hubungannya dengan permintaan para pemakai. Isi informasi harus ada hubungannya dengan masalah yang sedang dihadapi sedangkan semua keluaran yang lainnya tidak berguna. Sifat ini sulit mengukurnya.

5. Ketepatan waktu

Sifat ini berhubungan dengan waktu yang dilalui, yang lebih pendek dari siklus untuk mendapatkan informasi. Masukan pengolahan dan pelaporan keluaran kepada para pemakai biasanya tepat waktu. Dalam beberapa hal, ketepatan waktu dapat diukur. Misalnya berapa banyak penjualan dapat ditingkatkan dengan menanggapi permintaan pelanggan mengenai ketersediaan barang-barang inventaris.

6. Kejelasan

Sifat ini menunjukkan tingkat kejelasan informasi. Informasi hendaknya terbebas dari istilah-istilah yang tidak jelas.

7. Keluwesan

Sifat ini berhubungan dengan apakah informasi tersebut dapat digunakan untuk membuat lebih dari satu keputusan, tetapi apakah juga dapat digunakan untuk lebih dari seorang pengambil keputusan. Sifat ini sulit mengukurnya, akan tetapi dalam beberapa hal dapat diukur dengan suatu nilai tertentu.

8. Dapat dibuktikan

Sifat ini menunjukkan sejauh mana informasi itu dapat diuji oleh beberapa pemakai hingga sampai didapatkan kesimpulan yang sama.

9. Tidak ada prasangka

Sifat ini berhubungan dengan ada tidaknya keinginan untuk mengubah informasi tersebut guna mendapatkan kesimpulan yang telah diarahkan sebelumnya.

10. Dapat diukur

Sifat ini menunjukkan hakikat informasi yang dihasilkan oleh sistem informasi formal. Meskipun kabar angin, desas-desus, dugaan-dugaan, klenik, dan lainnya juga sering dianggap sebagai informasi, namun hal-hal tersebut berada di luar lingkup pembahasan.

Informasi bernilai sempurna apabila pengambil keputusan dapat mengambil keputusan secara optimal dalam setiap hal, dan bukan keputusan yang rata-rata akan menjadi optimal dan untuk menghindari kejadian-kejadian yang akan mendatangkan kerugian.

2.3.5 Kualitas Informasi

Agar informasi dapat mempunyai manfaat dalam proses pengambilan keputusan, informasi harus mempunyai kualitas dan nilai. Kriteria kualitas informasi adalah:

1. Akurat (*accuracy*) yang berarti informasi harus tidak bias atau menyesatkan dan bebas dari kesalahan.
2. Tepat waktu (*timeliness*) yang berarti informasi yang sampai kepada penerima tidak boleh terlambat. Mahalnya nilai informasi saat ini adalah karena harus cepatnya informasi tersebut didapatkan, sehingga diperlukan teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkan.
3. Relevan (*relevancy*) yang berarti informasi harus mempunyai manfaat bagi pihak yang menerimanya.

2.4 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen pembentuk sistem yang memiliki keterkaitan antara satu komponen dan komponen lain yang bertujuan menghasilkan informasi dalam bidang tertentu.

Pengertian sistem informasi menurut Lucas (2007) adalah kegiatan dari prosedur-prosedur yang diorganisasikan, apabila dieksekusi akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian dalam organisasi.

2.4.1 Komponen Sistem Informasi

Sistem informasi terdiri atas komponen-komponen yang disebut blok bangunan, yaitu komponen *input*, komponen model, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain dan membentuk suatu kesatuan untuk mencapai sasaran.

1. **Komponen *input***

Input mewakili data yang masuk dalam sistem informasi. *Input* di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.

2. **Komponen model**

Komponen ini terdiri atas kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan *output* yang diinginkan.

3. **Komponen *output***

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. **Komponen teknologi**

Teknologi merupakan *tool box* dalam sistem informasi, Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, serta membantu pengendalian dari sistem secara keseluruhan.

5. **Komponen *hardware***

Hardware berperan penting sebagai media penyimpanan vital bagi sistem informasi. Fungsinya sebagai tempat untuk menampung sumber data dan

informasi untuk memperlancar serta mempermudah kerja dari sistem informasi.

6. Komponen *software*

Software berfungsi sebagai tempat untuk mengolah, menghitung, dan memanipulasi data yang diambil dari *hardware* untuk menciptakan informasi.

7. Komponen basis data

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya.

8. Komponen kontrol

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah. Apabila terlanjur terjadi kesalahan, dapat cepat diatasi.

2.5 Konsep Dasar Analisis Sistem

Dalam buku Analisa Sistem Informasi oleh Sutabri (2005), berikut ini merupakan pengertian analisis sistem menurut para ahli, yaitu:

1. Menurut McLeod (2008), analisis sistem adalah penelitian atas sistem yang telah ada dengan tujuan merancang sistem yang baru.
2. Menurut Laudon (2007), analisis sistem adalah memeriksa sebuah masalah yang ada yang akan diselesaikan oleh perusahaan dengan menggunakan sistem informasi. Analisis sistem mencakup beberapa langkah yang harus dilakukan, yaitu:
 - a. Menentukan masalah.
 - b. Mengidentifikasi penyebab dari masalah tersebut.
 - c. Menentukan pemecahan masalahnya.
 - d. Mengidentifikasi kebutuhan informasi yang dibutuhkan untuk memecahkan masalah tersebut.

3. Menurut Bentley dan Whitten (2007), analisis sistem adalah suatu teknik untuk menyelesaikan suatu masalah yang ada pada suatu sistem dengan cara membagi masalah tersebut ke beberapa bagian dengan maksud agar mudah dicari penyelesaiannya.

Langkah-langkah yang harus dilakukan dalam menganalisis sistem menurut Jogiyanto (2005), yaitu:

1. *Identify*

Mengidentifikasi masalah merupakan langkah pertama yang dilakukan dalam tahap analisis. Masalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai. Oleh karena itulah pada tahap analisis sistem, langkah yang pertama dilakukan adalah mengidentifikasi terlebih dahulu masalah-masalah yang terjadi.

2. *Understand*

Langkah kedua dari tahap analisis sistem adalah memahami sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari operasi secara terperinci bagaimana sistem yang ada beroperasi. Untuk mempelajari operasi yang berjalan diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.

3. *Analyze*

Langkah ini dilakukan berdasarkan data yang diperoleh dari hasil penelitian yang telah dilakukan.

4. *Report*

Setelah proses analisis sistem selesai dilakukan, langkah terakhir adalah membuat laporan hasil analisis sistem yang menjelaskan temuan-temuan yang dapat dijadikan acuan alternatif penyelesaian masalah.

2.6 Sistem Pendukung Keputusan (SPK)

Konsep SPK pertama kali diperkenalkan pada awal tahun 1970-an oleh Scott Morton. Definisi SPK oleh Scott Morton dalam Herdiyanti dan Widiyanti, (2014) adalah sebagai "sistem berbasis komputer interaktif, yang membantu para

pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah tidak terstruktur”. SPK dirancang untuk menunjang seluruh tahapan pembuatan keputusan yang dimulai dari tahap mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pembuatan keputusan, sampai pada kegiatan mengevaluasi pemilihan alternatif.

Pada dasarnya SPK merupakan pengembangan lebih lanjut dari Sistem Informasi Manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Interaktif dengan tujuan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan, analisis, pengalaman dan wawasan manajer untuk mengambil keputusan yang lebih baik.

Menurut McLeod dalam Eniyati (2011) Sistem pendukung keputusan adalah sistem penghasil informasi yang ditujukan pada suatu masalah tertentu yang harus dipecahkan oleh manajer dan dapat membantu manajer dalam pengambilan keputusan. Pada dasarnya konsep DSS hanyalah sebatas pada kegiatan membantu para manajer melakukan pembuatan keputusan dan tidak melakukan penilaian serta menggantikan posisi dan peran manajer.

2.6.1 Karakteristik Sistem Pendukung Keputusan

Alters Keen dalam Yakub (2012) mengemukakan karakteristik SPK sebagai berikut:

1. SPK ditujukan untuk membantu keputusan-keputusan yang kurang terstruktur dan umumnya dihadapi oleh para manajer yang berada di tingkat puncak.
2. SPK merupakan gabungan antara kumpulan model kualitatif dan kumpulan data.
3. SPK memiliki fasilitas interaktif yang dapat mempermudah hubungan antara manusia dengan komputer.
4. SPK bersifat luwes dan dapat menyesuaikan dengan perubahan-perubahan yang terjadi.

2.6.2 Tujuan Sistem Pendukung Keputusan

Sistem pendukung keputusan pada hakekatnya memiliki beberapa tujuan yaitu:

1. Membantu manajer dalam pengambilan keputusan atas masalah semi-terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukan untuk menggantikan fungsi manajer.
3. Meningkatkan efektifitas keputusan yang diambil manajer lebih daripada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat, misalnya: semakin banyak data yang diakses, makin banyak juga alternatif yang bisa dievaluasi.
6. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. Menurut Simon, otak manusia memiliki kemampuan yang terbatas untuk memproses dan menyimpan informasi. Orang-orang kadang sulit mengingat dan menggunakan sebuah informasi dengan cara yang bebas dari kesalahan.

2.7 *System Development Life Cycle (SDLC)*

Menurut Dennis (2012) System Development Life Cycle (SDLC) memiliki 4 perangkat fase dasar yaitu planning, analysis, design, dan implementation. Beberapa proyek dalam membangun sebuah sistem informasi mungkin dapat berbeda-beda sesuai dengan caranya masing-masing, tetapi hampir semua proyek memiliki elemen dari 4 fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah-langkah yang menghasilkan deliverable / hasil kegiatan contohnya beberapa dokumen spesifik dan file yang menjelaskan pemahaman tentang proyek.

Dalam beberapa proyek, tahapan dan langkah-langkah SDLC diproses dalam suatu alur tahapan, dimulai dari awal hingga akhir. Dalam proyek lain, tim proyek terus berpindah dari fase ke fase tersebut secara teratur, bertahap, iteratif, atau dalam bentuk pola lainnya. Dalam SDLC, dijelaskan fase-fase, tindakan, serta beberapa teknik yang digunakan untuk menyempurnakan langkah-langkah yang dijelaskan secara umum. Ada dua hal penting untuk dipahami tentang SDLC:

1. Mendapatkan dan memahami pengertian umum dari tahapan dan langkah-langkah yang digunakan dalam dalam proyek Sistem Informasi serta beberapa teknik yang menghasilkan dokumen yang pasti.
2. Kedua hal penting untuk dipahami bahwa SDLC merupakan proses penyempurnaan yang dilakukan secara bertahap.

Dokumen yang dihasilkan dalam tahap analisis, memberikan ide umum dari suatu bagian dari sistem baru. Dokumen dari deliverable ini digunakan sebagai input pada tahap design, yang kemudian disempurnakan untuk menghasilkan dokumen yang menjelaskan secara detail dari sistem yang dibuat. Dokumen ini, akan digunakan dalam tahap implementasi untuk menghasilkan sistem yang sebenarnya. Berikut dibawah ini merupakan penjabaran dari setiap fase dalam SDLC:

1. Planning (Perencanaan)

Tahap planning / perencanaan adalah proses dasar yang menjelaskan mengapa sebuah sistem informasi harus dibangun dan menentukan bagaimana sebuah tim proyek akan membangunnya. Terdapat dua langkah dalam planning (perencanaan) yaitu:

- a. Inisiasi proyek, pada tahap ini nilai bisnis suatu sistem terhadap organisasi diidentifikasi: Apakah sistem tersebut akan menurunkan biaya atau meningkatkan pendapatan? Sebagian besar gagasan untuk sistem baru berasal dari luar area Departemen IT (misalnya dari Departemen Pemasaran, Departemen Akuntansi) dalam suatu bentuk system request (permintaan sistem). Permintaan sistem menyajikan ringkasan singkat tentang kebutuhan sistem dalam suatu bisnis dan menjelaskan bagaimana suatu sistem akan menciptakan nilai bisnis. Departemen

IT bekerja sama dengan departemen lain yang menghasilkan permintaan (disebut sponsor proyek) untuk melakukan analisis kelayakan sistem informasi. System request (permintaan sistem) dan analisis kelayakan disajikan dan dipresentasikan kepada pihak komite persetujuan Information System (komite pengarah), yang menentukan apakah proyek tersebut harus diambil atau tidak.

b. Setelah proyek disetujui, proyek memasuki manajemen proyek. Selama manajemen proyek, manajer proyek membuat sebuah rencana kerja, membentuk anggota staff proyek, dan anggota teknisi lainnya di tempat, untuk membantu tim proyek mengendalikan dan mengarahkan proyek pada keseluruhan tahap SDLC.

2. Analysis (Analisis)

Analisis menurut Muslihudin dan Oktafianto (2016) merupakan tahapan awal dalam pengembangan sistem dan merupakan tahap fundamental yang sangat menentukan kualitas sistem yang akan dikembangkan. Tahap analisis dalam SDLC menjelaskan siapa yang akan memakai sistem, apa yang sistem kerjakan serta kapan dan dimana sistem akan digunakan. Selama pada tahap ini, tim proyek menyelidiki sistem yang ada saat ini, mengidentifikasi peluang untuk melakukan perbaikan, dan mengembangkan sistem baru. Tahap ini memiliki tiga langkah:

a. Langkah yang pertama adalah Strategi Analisis. Langkah ini dikembangkan untuk mengarahkan usaha dari tim proyek untuk menganalisis sistem yang telah ada (disebut sebagai as-is system) beserta masalah-masalah yang ada dan untuk merancang suatu sistem yang baru (disebut sebagai to-be system).

b. Langkah selanjutnya adalah mengumpulkan kebutuhan-kebutuhan sistem (melalui wawancara atau kuesioner). Pada tahap ini terdapat juga masukan analisis informasi dari sponsor proyek dan pihak lainnya yang mengarah pada pengembangan konsep untuk sistem baru. Konsep sistem ini kemudian digunakan sebagai dasar untuk mengembangkan seperangkat model analisis bisnis, yang menggambarkan bagaimana bisnis akan beroperasi jika sistem baru dikembangkan.

c. Kemudian Analisis, konsep sistem, dan model digabungkan menjadi dokumen yang disebut proposal sistem, yang dipresentasikan ke sponsor proyek dan

pengambil keputusan utama lainnya (misalnya, anggota komite persetujuan) yang memutuskan apakah proyek tetap terus dilanjutkan.

3. Design (Perancangan)

Tahap design atau perancangan memutuskan bagaimana sistem akan beroperasi, mulai dari hal yang berkaitan dengan perangkat keras (hardware), perangkat lunak (software) dan infrastruktur jaringan seperti:

- a. Antarmuka pengguna (interface),
- b. Formulir,
- c. Laporan,
- d. Program spesifik: database, dan file-file lainnya yang akan dibutuhkan.

Meskipun sebagian besar keputusan strategis mengenai sistem dibuat dalam pengembangan konsep sistem selama tahap analisis, langkah-langkah dalam tahap desain menentukan secara tepat bagaimana sistem akan beroperasi. Fase desain ini memiliki empat langkah:

- a. Strategi perancangan (Design Strategy) adalah hal yang pertama kali yang dikembangkan. Strategi perancangan menjelaskan apakah sistem akan dikembangkan oleh programmer perusahaan sendiri, atau akan di-outsourcing ke perusahaan lain (biasanya perusahaan konsultan), atau apakah perusahaan akan membeli paket perangkat lunak yang sudah ada.
- b. Pengembangan desain arsitektur (architecture design) yang mengarah pada dasar untuk sistem, yang menggambarkan perangkat keras, perangkat lunak, dan infrastruktur jaringan yang akan digunakan. Dalam banyak kasus, sistem akan menambah atau mengubah infrastruktur yang sudah ada dalam suatu organisasi. Desain antarmuka menentukan bagaimana pengguna akan berpindah ke sistem yang baru (misalnya, metode navigasi seperti menu dan tombol di layar) serta laporan yang akan digunakan oleh sistem.
- c. Pengembangan database dan spesifikasi file. Pengembangan ini menentukan dengan tepat data apa yang akan disimpan dan di mana data tersebut akan disimpan.

d. Tim analis mengembangkan desain program (Program Design), yang mendefinisikan program yang perlu dikembangkan dan apa yang akan dilakukan masing-masing program.

Kumpulan dari hasil kegiatan ini seperti desain arsitektur, desain antarmuka, spesifikasi database dan file, dan perancangan program merupakan spesifikasi sistem yang diserahkan ke tim pemrograman untuk diimplementasi. Pada akhir tahap desain, analisis kelayakan dan rencana proyek diperiksa ulang dan diperbaiki, dan keputusan lain dibuat oleh sponsor proyek dan komite persetujuan mengenai apakah proyek dihentikan atau dilanjutkan.

4. Implementation (Implementasi)

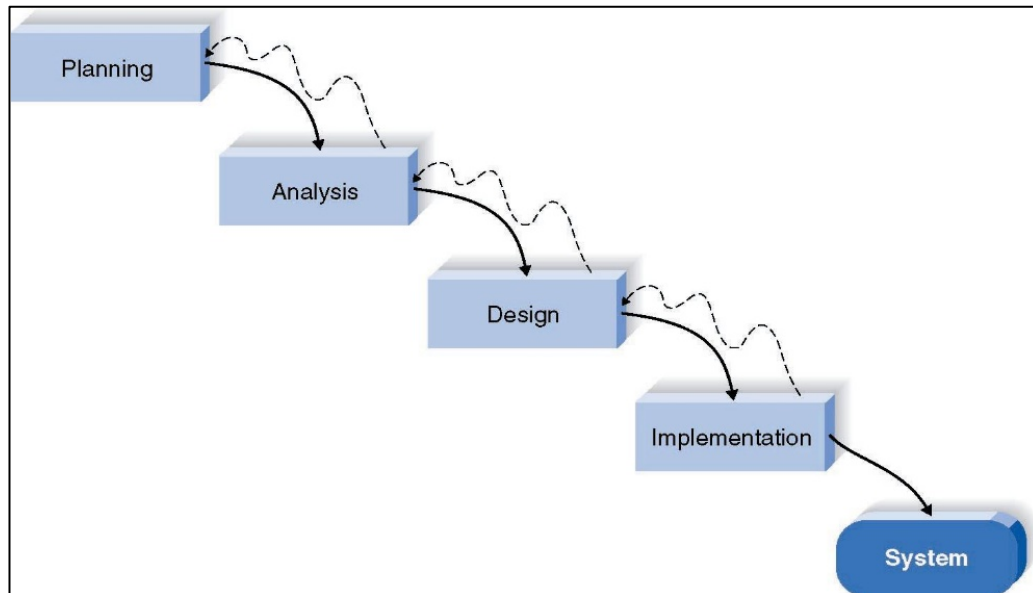
Tahap akhir dalam SDLC adalah tahap implementasi, di mana sistem benar-benar dibangun (atau dibeli, dalam kasus paket desain software). Fase ini sangat penting, karena kebanyakan sistem merupakan bagian terlama dan paling mahal dari proses pengembangan. Fase implementasi ini memiliki tiga langkah:

- a. Langkah awal adalah Konstruksi Sistem (System Construction). Sistem ini dibangun dan diuji untuk memastikan kinerja dari sistem, bekerja sebagaimana yang telah dirancang. Sebagian besar organisasi memberi lebih banyak waktu dan perhatian untuk menguji daripada mengubah program di tempat penerapan pertama.
- b. Instalasi Sistem, instalasi adalah proses dimana sistem lama dimatikan dan yang baru dinyalakan atau dihidupkan.
- c. Tim analis menetapkan sebuah rencana dukungan (support plan) untuk sistem.

Rencana ini biasanya mencakup tinjauan pasca-pelaksanaan formal atau informal serta cara sistematis untuk mengidentifikasi perubahan besar dan kecil yang diperlukan untuk sistem tersebut.

2.8 Model Waterfall

Metode Waterfall adalah sebuah metode pengembangan sistem dimana antar satu fase ke fase yang lain dilakukan secara berurutan. Terdapat 5 langkah tahapan dalam metode waterfall adalah sebagai berikut:



Gambar II.1 Metode *Waterfall*
(Sumber: Pressman, 2015)

Adapun penjelasan dari tahapan-tahapan Metode *Waterfall* menurut Dennis (2012) tersebut sebagai berikut:

1. Perencanaan (*Planning*)

Dalam tahapan ini, menjelaskan dan mengargumentasikan untuk melanjutkan proyek yang telah dipilih, rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Pada tahap ini ditentukan secara detail rencana kerja yang harus dikerjakan, durasi yang diperlukan masing-masing tahap, sumber daya manusia, perangkat lunak, dokumentasi, perangkat keras, maupun *financial diestimasi*. Pembuatan perencanaan ini bukan langkah mudah karena untuk mengestimasi beban kerja dan durasi dari masing-masing tahap dibutuhkan pengalaman cukup banyak. Kesalahan pada tahap ini akan mengakibatkan keuntungan yang diperoleh tidak maksimal, bahkan bisa rugi. Pada tahapan ini peran manajemen sistem informasi berpengalaman yang dibutuhkan.

2. Analisis (*Analysis*)

Tahap kedua, adalah tahap analisis, yaitu tahap dimana kita berusaha mengenali segenap permasalahan yang muncul pada pengguna dengan mendekomposisi dan merealisasikan komponen-komponen sistem. Tujuan utama dari tahap analisis adalah untuk memahami dan mendokumentasikan kebutuhan bisnis dan persyaratan proses dari sistem baru. Menganalisa kebutuhan sebagai bahan dalam membuat spesifikasi di tahapan selanjutnya.

3. Perancangan (*Design*)

Tahap perancangan (*design*) dimana kita mencoba mencari solusi permasalahan yang didapat dari tahap analisis. Tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang nyata untuk diimplementasikan. Jika pada tahapan analisis (*form requirement to specification*), maka tahapan desain adalah (*form specification to implementation*). Jadi, bagaimana pembuatan spesifikasi yang detail untuk bisa diimplementasikan.

4. Implementasi (*Implementation*)

Tahap implementasi, dimana kita mengimplementasikan perancangan sistem ke situasi nyata atau desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Disini kita mulai berurusan dengan pemilihan perangkat keras dan penyusunan perangkat lunak aplikasi (pengkodean/*coding*).

5. Sistem

Pada tahapan sistem dilakukan pengujian (*testing*) dan pemeliharaan, yang dapat digunakan untuk menentukan apakah sistem/perangkat lunak yang kita buat sudah sesuai dengan kebutuhan pengguna atau belum. Jika belum, proses selanjutnya adalah bersifat *interactive*, yaitu kembali ke tahap sebelumnya. Tahap pemeliharaan dan perawatan dimana kita mulai melakukan pengoperasian sistem dan jika diperlukan melakukan perbaikan-perbaikan kecil. Kemudian jika waktu pengguna sistem habis, maka kita akan masuk lagi pada tahap perancangan (*design*).

2.9 Kelebihan Model *Waterfall*

Kelebihan menggunakan metode air terjun (*waterfall*) adalah metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model fase *one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan.

2.10 Definisi Absensi

Edi Purwanto (2010) absensi adalah suatu pendataan kehadiran, bagian dari pelaporan aktifitas suatu institusi, atau komponen institusi itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan.

2.10.1 Jenis-Jenis Absensi

Kita mengenal beberapa jenis absensi. Yang membedakan jenis-jenis absensi tersebut adalah cara penggunaannya, dan tingkat daya gunanya secara umum jenis-jenis absensi dapat di kelompokkan menjadi dua, yaitu:

1. Absensi manual adalah cara pengentrian kehadiran dengan cara menggunakan pena (tanda tangan) ataupun kartu absensi.
2. Absensi non manual (dengan menggunakan alat) Absensi non manual adalah suatu cara pengentrian kehadiran dengan menggunakan system terkomputerisasi, bisa menggunakan kartu dengan barcode, finger print ataupun dengan mengentrikan nip dan sebagainya.

2.10.2 Pengelolaan Absensi

Pengelolaan absensi dengan memanfaatkan kelebihan teknologi informasi adalah:

1. Absensi dapat dilakukan dengan lebih mudah dan cepat.
2. Informasi hasil pencarian yang disajikan lebih lengkap.

3. Link antar bagian divisi (Pencarian data kehadiran dari satu divisi ke divisi yang lain)
4. Mempermudah dalam melakukan Entry dan Update data.
5. Memudahkan pembuatan laporan dan rekapitulasi.
6. Terdapat fasilitas informasi Dengan demikian sistem yang terkomputerisasi akan mempermudah kerja bagian kepegawaian, serta meningkatkan kualitas pelayanan dan kinerja dari Perusahaan.

2.11 Pengertian Presensi

Daftar presensi atau biasa disebut dengan daftar hadir atau kehadiran adalah sebuah lembaran tertentu yang berisi kolom nama, alamat, jabatan dan tanda tangan. Menurut Bastian (2007), Presensi karyawan adalah suatu kegiatan mendokumentasikan kehadiran karyawan di perusahaan, setiap hari kerja pegawai diharuskan melakukan presensi pada waktu datang dan pulang, dalam satu periode waktu.

2.12 Definisi Jam Lembur

Nurachmad (2009), Lembur digambarkan sebagai pekerjaan dengan tambahan waktu kerja regular. Menurut undang-undang kebutuhan, waktu kerja regular biasanya delapan jam sehari, dan 40 jam dalam seminggu. Waktu ini bervariasi di setiap negara-negara industri. Informasi ini dapat digunakan untuk mengkalkulasi tenaga kerja yang efisiensi dan lainnya, untuk mengkalkulasi tergantung pada pembayaran tambahan di luar tingkat tarif yang sesuai pada hukum. Berdasarkan KEP.102/MEN/VI/2004 tentang waktu lembur dan upah lembur, waktu lembur kerja adalah waktu kerja yang melebihi 7 jam sehari dan 40 jam 1 minggu untuk 6 hari kerja dalam 1 minggu atau 8 jam sehari, dan 40 jam 1 minggu untuk 5 hari kerja dalam 1 minggu atau upah yang diterima pada hari istirahat mingguan.

2.13 Definisi Gaji

Gaji adalah pembayaran atas penyerahan jasa yang dilakukan oleh para karyawan yang mempunyai jenjang jabatan manager, dan dibayarkan secara tetap per bulan. Sedangkan upah merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan pelaksana (buruh) umumnya dibayarkan berdasarkan hari kerja, jam kerja, atau jumlah satuan produk yang dihasilkan oleh karyawan.

2.14 Diagram Bagan Alir (*Flowmap*)

Febriani (2014), *flowmap* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowmap* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian .

Menurut Jogiyanto (2005) bagan alir digunakan untuk alat bantu komunikasi dan dokumentasi. Pada waktu menggambar suatu diagram alir, analis sistem dapat mengikuti ketentuan- ketentuan sebagai berikut:

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol-simbol bagan alir yang standar.

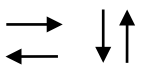


Simbol-simbol *flowmap* yang digunakan merupakan simbol-simbol *flowmap* standar yang dikeluarkan oleh ANSI dan ISO. Berikut ini merupakan simbol-simbol standar menurut Febriani (2014) yang digunakan untuk

menggambarkan diagram alir sesuai kegunaan simbol:

1. *Flow Direction Symbols*

Simbol yang dipakai untuk menghubungkan antara simbol yang satu dengan simbol lainnya atau disebut juga *connecting line*. Tabel II.1 menjelaskan tentang simbol-simbol penghubung dalam menggambarkan diagram alir.

Tabel II.1 Simbol *Flow Direction*

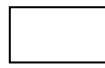

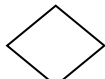
Simbol	Nama	Keterangan
	<i>Alur/Flow</i>	Penghubung antara prosedur atau proses.
	<i>Connector</i>	Simbol keluar atau masuk prosedur atau proses dalam lembar atau halaman yang sama.
	<i>Off-line Connector</i>	Simbol keluar atau masuk prosedur atau proses dalam lembar atau halaman lain.

(Sumber: Febriani, 2014)



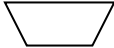
2. Simbol Proses

Simbol proses adalah simbol yang menunjukkan jenis operasi pengolahan dalam suatu prosedur. Tabel II.2 dan Tabel II.3 menjelaskan tentang simbol-simbol proses dalam menggambarkan diagram alir.

Tabel II.2 Simbol Proses

Simbol	Nama	Keterangan
	<i>Process</i>	Simbol yang menunjukkan pengolahan yang dilakukan komputer.
	<i>Predefined Process (Sub Proses)</i>	Permulaan sub proses.
	<i>Decision</i>	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau aksi.

Tabel II.2 Simbol Proses (Lanjutan)

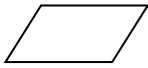
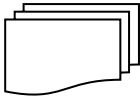

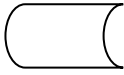
Simbol	Nama	Keterangan
	<i>Predefined Process</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i> .
	Terminal	Simbol untuk permulaan atau akhir dari suatu program.
	<i>Manual Process</i>	Simbol untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer.

(Sumber: Febriani, 2014)

3. Simbol *Input* dan *Output*

Simbol yang dipakai untuk menyatakan jenis peralatan yang digunakan sebagai media *input* atau *output*. Tabel II.3 menjelaskan tentang simbol-simbol *input* dan *output* dalam menggambarkan diagram alir.

Tabel II.3 Simbol *Input* dan *Output*

Simbol	Nama	Keterangan
	<i>Input-Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
	Dokumen Rangkap	Menggambarkan dokumen asli dan tembusannya.
	<i>Document</i>	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak dikertas.
	<i>Disk and On-line Storage</i>	Simbol untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> di simpan ke <i>disk</i> .

(Sumber: Febriani, 2014)

2.15 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Tujuan dari UML adalah untuk menyediakan kosa kata yang umum dari istilah-istilah berbasis objek dan teknik yang cukup banyak untuk memodelkan proyek pengembangan sistem dari analisis ke desain.

Menurut Dennis (2010), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu, *Structure diagram* dan *Behavior diagram*. *Structure diagram* biasanya digunakan untuk mepresentasikan data dan hubungan statik yang ada di dalam sebuah sistem informasi. Sedangkan *Behavior diagram* menyediakan para analis dengan sebuah gambaran hubungan yang dinamis antara instansi atau objek yang merepresentasikan sistem informasi bisnis. Berikut adalah tabel dari jenis diagram yang ada dan fungsi-fungsinya:

Tabel II.4 Jenis Diagram Pada UML dan Fungsi-fungsinya

Nama Diagram	Fungsi
<i>Class Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan <i>class</i> di dalam sistem
<i>Object Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan objek di dalam sistem
<i>Package Diagram</i>	Mengelompokkan elemen UML lainnya menjadi satu untuk membentuk level konstruksi yang lebih tinggi
<i>Deployment Diagram</i>	Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk menunjukkan komponen <i>software</i> sedang dijadikan arsitektur fisik suatu sistem
<i>Component Diagram</i>	Menggambarkan hubungan fisik diantara komponen-komponen <i>software</i>

Tabel II.4 Jenis Diagram Pada UML dan Fungsi-fungsinya (Lanjutan)




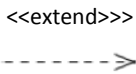
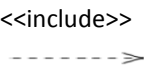


Nama Diagram	Fungsi
<i>Composite Diagram</i>	Menggambarkan struktur internal dari suatu <i>class</i> dan hubungan diantara bagian-bagian dari suatu <i>class</i>
<i>Activity Diagram</i>	Menggambarkan proses bisnis masing-masing <i>class</i>
<i>Sequence Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam aktifitas berdasarkan urutan waktu
<i>Communicaiton Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu objek
<i>Communicaiton Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu objek
<i>Interaction Overview Diagram</i>	Menggambarkan suatu ikhtisar alur dari kontrol suatu proses
<i>Timing Diagram</i>	Menggambarkan interaksi yang terjadi diantara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu
<i>Behavioral State Machine Diagram</i>	Memeriksa kebiasaan dari suatu <i>class</i>
<i>Protocol State Machine Diagram</i>	Menggambarkan ketergantungan diantara perbedaan-perbedaan <i>interface</i> dari suatu <i>class</i>
<i>Use Case Diagram</i>	Menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi diantara sistem dan lingkunganya

(Sumber: Dennis, 2010)

2.15.1 Use Case Diagram

Diagram *use case* dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam diagram *use case* ini adalah apa yang dapat diperbuat oleh sistem, dan bukan bagaimana sistem melakukannya. Diagram *use case* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut. Simbol-simbol yang terdapat didalam *use case diagram* dijelaskan pada tabel II.5 dibawah ini:

Tabel II.5 Simbol-simbol *Use case Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Use case</i>	Merepresentasikan bagian utama dari sistem secara fungsional.
2		<i>Actor</i>	Seseorang atau sistem yang mendapatkkn keuntungan dari sistem
3		<i>Association</i>	Menghubungkan suatu <i>actor</i> dengan <i>use case</i> .
4		<i>Extend</i>	Merepresentasikan eksistensi dari <i>use case</i> untuk menyertakan perilaku <i>optional</i> .
5		<i>Include</i>	Mewakili dimasukkannya satu <i>use case</i> ke <i>use case</i> yang lain.
6		<i>Subject</i>	Menyertakan nama subjek didalam maupun diatas
7		<i>Generalization</i>	Merepresentasikan <i>use case</i> khusus ke yang lebih umum.

(Sumber: Dennis, 2015)

2.15.2 *Use Case Description*

Menurut Dennis (2010), *Use Case Description* merupakan deskripsi yang mencatat mengenai detail pemrosesan dari suatu *use case*. *Use Case Description* memiliki urutan yang lengkap dari tahapan-tahapan untuk menyelesaikan suatu proses bisnis, elemen yang dimiliki *Use Case Description* adalah sebagai berikut:

1. *Basic Information*

a. Nama

Nama digunakan untuk menjelaskan *use case* dan membedakan *use case* satu dengan yang lainnya. Nama yang digunakan harus sederhana namun mampu menjelaskan kondisi yang memungkinkan.

b. *Description*

Description berarti uraian singkat untuk menyampaikan tujuan dari *use case*.

c. Aktor

Aktor merujuk kepada orang, sistem, perangkat lunak ataupun perangkat keras yang berinteraksi langsung dengan sistem untuk mencapai tujuan.

2. *Normal Courses*

Bagian utama dari *use case* berikutnya adalah *normal courses*. Pada *normal courses* menjelaskan mengenai langkah-langkah yang dilakukan untuk mengeksekusi respon terhadap masukan dan keluaran.

3. *Alternative Courses*

Untuk beberapa *use case* memiliki jalur alternatif pada setiap kasusnya. Pada penggambaran jalur alternatif ini terdapat cabang yang juga akan menghasilkan kesimpulan dari *use case* tersebut.

2.15.3 *Activity Diagram*

Menurut Dennis (2010) *Activity diagram* digunakan untuk memodelkan perilaku dalam proses bisnis yang independen terhadap objek. Dalam banyak hal, *activity diagram* dapat dipandang sebagai data flow diagram canggih yang digunakan pada konjungsi dengan analisis terstruktur. Walaupun begitu, *activity*



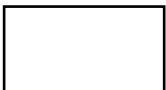
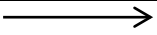
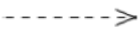

diagram tidak seperti *data flow diagram*, *activity diagram* menggunakan notasi yang mengalamatkan pemodelan paralel.

Yang perlu diperhatikan disini adalah bahwa diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan actor, jadi aktifitas yang dapat dilakukan oleh sistem. Diagram aktifitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:


1. Rancangan proses bisnis dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktifitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktifitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang akan ditampilkan pada perangkat lunak

Berikut adalah simbol-simbol yang ada pada Activity Diagram:

Tabel II.6 Simbol-simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Action</i>	Merupakan notasi yang simpel, dan tidak untuk dianalisis perilakunya. Dinamakan sesuai aksi yg dilakukan.
2		<i>Activity</i>	Digunakan untuk mewakili satu set aksi yang dilakukan.
3		<i>Object Node</i>	Digunakan untuk mewakili objek yang terhubung ke satu set arus objek.
4		<i>Control Flow</i>	Menampilkan urutan eksekusi.
5		<i>Object Flow</i>	Menunjukkan aliran dari sebuah objek dari satu aktifitas ke aktifitas lainnya.
6		<i>Initial Node</i>	Menggambarkan permulaan dari suatu set aktifitas atau aksi.

Tabel II.6 Simbol-simbol *Activity Diagram* (Lanjutan)

No	Simbol	Nama	Deskripsi
7		<i>Final-activity Node</i>	Menggambarkan akhir dari setiap aktifitas.



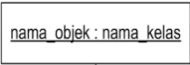

(Sumber: Dennis, 2015)

2.15.4 *Sequence Diagram*

Sequence diagram adalah salah satu dari dua jenis diagram interaksi. *Sequence diagram* menggambarkan benda-benda yang berpartisipasi dalam kasus penggunaan dan pesan yang melewati antara mereka dari waktu ke waktu untuk satu *use case*. *Sequence Diagram* adalah model dinamis yang menunjukkan urutan eksplisit pesan yang lewat di antara objek dalam interaksi didefinisikan.

Karena urutan diagram menekankan pemesanan berbasis waktu kegiatan yang terjadi di antara set benda, diagram tersebut sangat membantu untuk memahami spesifikasi *real-time* dan kompleks menggunakan kasus (Dennis, 2010). Berikut simbol yang ada pada *Sequence Diagram*:

Tabel II.7 Simbol-simbol *Sequence Diagram*


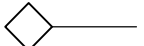
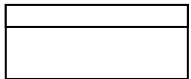
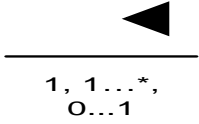
No	Simbol	Nama	Deskripsi
1.		Aktor	Orang atau sistem yang berasal dari manfaat dan eksternal ke sistem yang berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan
2.		Garis Hidup/ <i>lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan yang ditempatkan diatas diagram.
4.		<i>Execution Occurrence</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.

(Sumber: Dennis, 2011)

2.15.5 Class Diagram

Menurut Dennis (2011) sebuah diagram kelas adalah model statis yang menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. Sebuah *class* merupakan deskripsi dari sekumpulan objek yang memiliki properti (*attribute*), operasi (*method*), relasi (*association*), dan tingkah laku (*behavior*) yang sama. Sebuah *class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda /fungsi). Berikut simbol-simbol yang ada pada diagram kelas:

Tabel II.8 Simbol-simbol *Class Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Generalization</i>	Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus.
2		<i>Aggregation</i>	Menggambarkan suatu <i>class</i> terdiri dari <i>class</i> lain atau suatu <i>class</i> adalah bagian dari <i>class</i> lain.
3		<i>Class</i>	Kelas pada struktur sistem
4		<i>Association</i>	Asosiasi yang menghubungkan <i>class</i> dengan <i>class Multiplicity</i> .


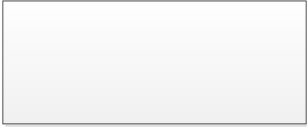

(Sumber: Alan Dennis, 2011)

2.15.6 Deployment

Deployment diagram adalah salah satu jenis alat atau bahasa (UML) yang digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis Object Oriented yang akan dibangun. Hal inilah yang mempermudah user dalam

pemakaian sistem yang telah dibuat dan diagram tersebut merupakan diagram yang statis. Misalnya untuk mendeskripsikan sebuah situs web, deployment diagram menunjukkan komponen perangkat keras (“node”) apa yang digunakan (misalnya, web server, server aplikasi, dan database server), komponen perangkat lunak (“artefak”) apa yang berjalan pada setiap node (misalnya, aplikasi web, database), dan bagaimana bagian-bagian yang berbeda terhubung (misalnya JDBC, REST, RMI). Tujuan atau fungsi dari deployment diagram yaitu untuk menggambarkan/memvisualisasikan secara umum proses yang terjadi pada suatu sistem/software. Berikut simbol-simbol yang ada pada *Deployment* diagram :

Tabel II.9 Simbol-simbol *Deployment* Diagram

NO	Simbol	Nama	Deskripsi
1		Package	Package merupakan sebuah bungkus dari satu atau lebih node.
2		Node	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak (<i>software</i>).
3		Link	Relasi antar package ke package, atau node ke node.

(Sumber: Dennis, 2011)

2.16 *Entity Relationship Diagram (ERD)*

Pengertian dari ERD (Entity Relationship Diagram) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

Pada dasarnya ada tiga komponen yang digunakan, yaitu :

1. Entitas

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.



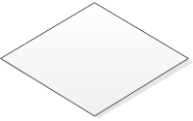
2. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain.

3. Hubungan/Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Tabel II.10 Simbol-simbol *ERD*

SIMBOL	NAMA	DESKRIPSI
	Entitas	Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain.
	Atribut	Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut.
	Relasi	Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

(Sumber: Dennis, 2019)

2.17 XAMPP

Menurut Sutaji (2012), XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai *web server* pada komputer. XAMPP juga dapat disebut sebagai sebuah *CPanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses internet.

Fungsi lainnya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman Java SE. Nama XAMPP merupakan singkatan dari X, Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

Menurut Sutaji (2012) MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang tabel.

2.18 MySQL

My Structured Query Language (MySQL) adalah *Relational Database Management System* (RDBMS) yang sangat cepat dan kuat. Sebuah basis data memungkinkan user untuk melakukan penyimpanan yang efisien, pencarian dan pengurutan data. MySQL *server* memiliki kemampuan melakukan kontrol akses terhadap data untuk memastikan bahwa setiap *user* dapat bekerja dengan sesuai, menyediakan akses yang cepat, dan meyakinkan bahwa hanya *user* yang mempunyai hak akses yang dapat mengaksesnya. Oleh karena itu, MySQL merupakan *database server* yang yang dapat digunakan banyak *user* dan banyak pekerjaan.

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structure Query Language*) Sutaji (2012).

2.19 Tipe data MySQL

Menurut Sutaji (2012) MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang *table*. Jenis data yang tersedia pada MySQL dapat dilihat pada Tabel II.11 berikut:

Tabel II.11 Tipe Data MySQL

Tipe Data	Keterangan
<i>CHAR</i>	Sebuah <i>string</i> dengan panjang tetap. Sisa Jumlah karakter yang belum terisi akan diisi dengan spasi, akan tetapi spasi ini dibuang jika data dipanggil. Jangkauan nilai M adalah 1-255 karakter.
<i>VARCHAR</i>	<i>String</i> dengan panjang berupa variabel. M bisa mencapai 65535.
<i>DATE</i>	Data berupa tanggal. Format tanggal dalam bentuk ‘YYYY-MM-DD’.
<i>TIME</i>	Data berupa waktu. Format waktu dalam bentuk ‘HH:MM:SS’.
<i>TINYINT</i>	Bilangan antara -128 sampai dengan +127.
<i>SMALLINT</i>	Bilangan antara -32768 sampai dengan +32767.
<i>MEDIUMINT</i>	Bilangan antara -8388608 sampai dengan +8388607.
<i>INT</i>	Bilangan antara -2147683648 sampai dengan +2147683647
<i>FLOAT</i>	Bilangan <i>floating point</i> yang kecil (presisi tunggal). Jangkauan nilai yang diizinkan adalah -3.402823466E+38 s/d -1.175494351e-38,0 dan 1.175494351E-38 s/d 3.402823466E+38.
<i>DOUBLE</i>	Bilangan <i>floating point</i> dengan ukuran normal (presisi ganda). Jangkauan nilai yang diizinkan adalah -1.7976931348623157E+308 s/d -2.225073858507201E-308,0 dan 2.225073858507201E-308.

Tabel II.11 Tipe Data MySQL (Lanjutan)

Tipe Data	Keterangan
<i>ENUM</i>	Sebuah <i>enumeration</i> . Sebuah obyek string yang hanya boleh memiliki satu nilai, yang terambil dari 'value1','value2',...,NULL atau nilai spesial "" <i>error</i> . Sebuah <i>enum</i> dapat menampung 65535 pilihan nilai.

(Sumber: Sutaji, 2012)

2.20 Blackbox Testing

Black-Box Testing adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi tanpa mengintip ke dalam struktur internal atau cara kerjanya. Metode pengujian ini dapat diterapkan secara virtual untuk setiap tingkat pengujian perangkat lunak (unit, integrasi, sistem, dan penerimaan). Kadang-kadang disebut sebagai pengujian berbasis spesifikasi (Ron Patton, 2005). Contoh kegiatan yang dilakukan pada *black-box testing* yaitu:

1. Membuat *test case* untuk menguji fungsi-fungsi yang ada pada aplikasi
2. Membuat *test case* untuk menguji kesesuaian alur kerja suatu fungsi di aplikasi dengan *requirement* yang dibutuhkan *user* untuk fungsi tersebut.
3. Mencari *bugs/error* dari tampilan (*Interface*) aplikasi.

2.21 Java Standard Edition

Java Standard Edition atau J2SE banyak digunakan sebagai platform untuk pemrograman dalam bahasa *Java*. Platform ini digunakan untuk mendeploy sebuah aplikasi. *Java SE* terdiri dari virtual machine yang digunakan untuk menjalankan program *java* bersama-sama dengan library atau paket. *Java Standard Edition* merupakan bahasa pemrograman Java untuk aplikasi desktop yang merupakan *object-oriented programming*. Pada JSE atau J2SE, terdiri dari dua buah produk yang dikeluarkan untuk membantu dalam membuat aplikasi tanpa tergantung dari *platform* yang digunakan, yaitu:

1. *Java SE Runtime Environment (JRE)*

Java Runtime Environment (JRE) menyediakan perpustakaan, *Java Virtual Machine* (JVM), dan komponen lain untuk menjalankan *applet* dan aplikasi yang ditulis dengan bahasa pemrograman Java.

2. *Java Development Kit* (JDK)

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk manajemen dan membangun berbagai aplikasi *Java*. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan *compiler* dan *debugger* yang diperlukan untuk mengembangkan *applet* dan aplikasi.

2.22 Mesin Absensi Kartu

Mesin absensi kartu manual terdiri dari celah atau lubang untuk memasukkan kartu absen karyawan serta jam analog atau digital yang menunjukkan waktu saat ini. Pegawai yang ingin mengisi kartu jam hadir harus memasukkan kartu jam hadir ke mesin absen, dan secara otomatis kartu jam hadir akan mencetak jam. hadir atau pulang karyawan sesuai dengan jam kerja.

2.23 Pengertian *Time Card*

Time Card atau kartu jam hadir adalah dokumen yang mencatat jam hadir setiap pegawai. Catatan jam hadir pegawai ini berisikan nama pegawai, jabatan, nik, tanggal bulan tahun, jam masuk, jam keluar dan jam lembur.

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian merupakan proses ataupun cara ilmiah yang digunakan untuk mendapatkan data yang akan digunakan untuk keperluan penelitian. Metodologi penelitian juga sering dikenal sebagai metode ilmiah, metode tersebut digunakan untuk mencari, mengembangkan, serta menguji suatu kebenaran pengetahuan. Metodologi penelitian mempelajari cara-cara melakukan pengamatan dengan pemikiran yang tepat secara terpadu melalui tahapan-tahapan yang disusun secara ilmiah untuk mencari, menyusun, serta menganalisis dan menyimpulkan data-data berdasarkan fakta-fakta secara ilmiah.

3.2 Pengumpulan Data

Terdapat 2 jenis data yang diperlukan dalam melakukan pengumpulan data, yaitu data primer dan data sekunder. Sumber dari data-data ini diperoleh dari tempat penulis melakukan praktek kerja lapangan yang dilakukan di PT. Korea Nano Tech Indonesia. Adapun jenis data yang dikumpulkan meliputi:

1. Data Primer

Data primer merupakan data yang diperoleh secara langsung dari perusahaan, dimana pengumpulan data atau informasi dilakukan langsung diperoleh dari objek yang diteliti. Data tersebut data yang digunakan dalam berlangsungnya proses perekrutan karyawan, diantaranya struktur organisasi, SOP, proses bisnis yang sedang berjalan saat ini, dan dokumen-dokumen yang terlibat dalam proses penjadwalan produksi yang sedang berjalan.

2. Data Sekunder

Data sekunder merupakan data yang diperoleh dari data yang tersedia, dan yang sudah ada terlebih dahulu dikumpulkan dan dilaporkan kepada pihak lain, buku-buku, dan kajian ilmiah dari berbagai sumber yang berkaitan

dengan penelitian. Data tersebut adalah data umum perusahaan, profil perusahaan, serta struktur organisasi perusahaan

3.3 Metode Pengumpulan Data

Menurut Sugiyono (2013), teknik pengumpulan data merupakan langkah yang paling strategis dalam melakukan penelitian, karena tujuan utama dari penelitian tersebut adalah mendapatkan data. Penelitian ini menggunakan beberapa teknik pengumpulan data, yaitu:

1. Observasi dan Wawancara

Observasi yaitu melakukan pengamatan situasi langsung pada sistem yang berjalan di Divisi *Human Resources Department* pada PT. Korea Nano tech Indonesia. Metode wawancara adalah metode pengumpulan data melalui tanya jawab secara langsung kepada pihak-pihak yang terkait dengan objek penelitian. Pihak yang diwawancarai adalah bagian umum dan kepala bagian divisi *Human Resources Department*. Hasil dari observasi dan wawancara yang dilakukan menjadi landasan dalam melakukan pengembangan sistem yang akan dibuat.

2. Studi Pustaka

Studi pustaka yaitu metode pengumpulan data dengan melakukan pencarian referensi yang berkaitan dengan pemrograman *Java script* menggunakan *NetBean IDE* dan *database MySQL* dari berbagai referensi, baik itu referensi elektronik yang didapat dari internet maupun referensi dari buku teks. Referensi yang diperoleh, kemudian dikaji sebagai dasar dalam menyelesaikan penelitian.

3.4 Metode Pengembangan Sistem

Menurut Dennis (2011) *System Development Life Cycle* (SDLC) memiliki empat perangkat fase dasar yaitu *planning*, *analysis*, *design*, dan *implementation*. Beberapa cara dalam membangun sebuah sistem mungkin dapat mempunyai perbedaan, tetapi hampir semua proyek memiliki elemen dari empat fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah-langkah yang

menghasilkan *deliverable* atau hasil kegiatan seperti beberapa dokumen spesifik dan file yang menjelaskan pemahaman tentang proyek. Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *waterfall*.

Menurut Dennis (2011) Metode *waterfall* sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Metode *waterfall* menyediakan pendekatan alur hidup perangkat lunak dengan terurut dari fase awal (*planning*) hingga fase akhir (*implementation*), dan pada setiap fase pihak pengembang dan sponsor akan selalu ada kontak untuk mendapatkan persetujuan sebelum masuk pada tahap selanjutnya.

Tahapan-tahapan dalam *waterfall* adalah sebagai berikut:

- a. Identifikasi kebutuhan pengguna (*planning*)
Pengembang melakukan diskusi dengan *manager* divisi *Product Planning & Inventory Control* tentang kebutuhan sistem yang digunakan.
- b. Analisis kebutuhan perangkat lunak (*Analysis*)
Pengembang melakukan proses pengumpulan kebutuhan yang dilakukan secara intensif dengan pegawai divisi *Product Planning & Inventory Control* tentang kebutuhan sistem yang diinginkan.
- c. Desain (*Design*)
Pengembang membuat sebuah desain program perangkat lunak seperti struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean serta dokumentasinya dari sistem yang telah dijelaskan oleh *Foreman* dan pegawai divisi *Product Planning & Inventory Control*
- d. Pembuatan Kode Program (*Implementaion*)
Pengembang mengimplementasikan perancangan sistem ke situasi nyata atau desain harus diterjemahkan ke dalam suatu bahasa pemrograman yang dapat dibaca oleh mesin.
- e. Sistem (*System*)
Pada tahap ini pengembang melakukan pengujian (*testing*) dan pemeliharaan, yang dapat digunakan untuk menentukan apakah sistem atau perangkat lunak yang sudah dibuat sesuai dengan kebutuhan pengguna atau belum.

3.5 *Literature Review*

Literature review merupakan evaluasi yang terperinci serta kritis tentang penelitian sebelumnya pada suatu topik yang sedang dibicarakan. Termasuk peran penting dalam membuat sebuah karya ilmiah, maupun tulisan karena dapat memberikan ide dan tujuan tentang topik penelitian yang akan dilaksanakan. Pada umumnya, *Literature review* memiliki isi meliputi ulasan, rangkuman, dan pemikiran penulis tentang beberapa pustaka. Adapun beberapa langkah dalam melakukan *literature review* yaitu:

1. Formulasi Permasalahan

Penulis memilih topik yang sesuai, selain itu permasalahan yang diangkat harus sesuai, dalam hal ini sesuai dengan penelitian yang akan dilakukan

2. Mencari Literatur

Literatur harus relevan dengan penelitian yang sedang dilakukan, sehingga dapat membantu untuk mendapatkan gambaran dari suatu topik dan objek penelitian

3. Evaluasi Data

Membandingkan setiap literatur yang ada, lalu mencari kesamaan, maupun perbedaan dengan topik yang akan dibahas.

Berikut merupakan *literature review* dari beberapa jurnal ilmiah yang dapat dilihat pada Tabel III.1

No	Pengarang/ Tahun	Topik Penelitian	Permasalahan	Metode yang digunakan	Kesimpulan Penelitian	Hasil Ulasan
1	Ardianto moenir & Fajar Yulianto, Tahun 2017	Perancangan Sitem Informasi Penggajian berbasis web dengan metode waterfall	Data Penghitungan gaji pegawai menggunakan metode rumus microsoft excel agar mendapatkan hasil yang akurat dalam penghitungan gaji, akan tetapi memakan banyak waktu.	Metode <i>Waterfall</i>	Penerapan metode <i>waterfall</i> membantu dalam pengembangan sistem mulai dari perancangan sampai pemeliharaan sistem, dengan metode ini dapat membantu dalam pembuatan dan implementasi aplikasi sitem terbaru dalam menghitung gaji pegawai.	<p>Persamaan: Menggunakan metode <i>waterfall</i> dapat membantu dalam membuat dan memelihara sistem yang baru dibuat. Selain itu metode <i>waterfall</i> juga memiliki proses yang urut, mulai dar analisa hingga support, setiap proses memiliki spesifikasinya sendiri, sehingga sebuah sistem dapat dikembangkan sesuai dengan apa yang dikehendaki (tepat sasaran), dan juga setiap proses yang ada dalam metode <i>waterfall</i> juga tidak dapat saling tumpang tindih.</p> <p>Perbedaan: Adapun perbedaan penulis jurnal ini menggunakan program <i>Visual Basic 6.0</i>.</p>

Tabel III.1 Literatur Review
(Sumber: Pribadi, 2019)

No	Pengarang/ Tahun	Topik Penelitian	Permasalahan	Metode yang digunakan	Kesimpulan Penelitian	Hasil Ulasan
2	Adian Nurcahyadi & Wenno napitupulu, Tahun 2017	Rancang Bangun Sistem Penggajian menggunakan Visual Studio dengan metode waterfall	Banyaknya pegawai membuat divisi HRD menjadi lebih kerja keras dalam menghitung gaji, hal ini dapat menimbulkan <i>human error</i> dalam penghitungan.	Metode <i>Waterfall</i>	Aplikasi sistem penghitungan gaji memudahkan divisi, hanya dengan memasukan data hadir pegawai dan jam lembur yang sudah ada maka akan menghasilkan output berdasarkan total atau jumlah gaji pegawai tersebut.	<p>Persamaan: Menggunakan metode <i>waterfall</i> memudahkan dalam pengembangan aplikasi sistem. Hal ini disebabkan karena sistem operasi komputer memiliki fungsi yang sangat kompleks, sehingga dalam pengembangannya membutuhkan analisa yang penuh mengenai kebutuhan user akan sistem operasi, hingga fitur-fitur penting yang harus dimiliki oleh sistem operasi tersebut.</p> <p>Perbedaan: Adapun perbedaannya adalah penulis jurnal ini menggunakan database oracle.</p>

Tabel III.1 Literatur Review(Lanjutan)
(Sumber: Pribadi, 2019)

3.6 Kerangka Pemecahan Masalah

Dalam penelitian yang dibahas dalam tugas akhir ini, dilakukan langkah-langkah atau tahapan dalam pembuatan program yang ada sesuai dengan metodologi penelitian yang digunakan dalam pembuatan program tersebut. Penjelasan langkah-langkah atau tahapan dalam pembuatan program tersebut adalah sebagai berikut:

1. Studi Pendahuluan

Pada tahap awal penulis melakukan sebuah studi pendahuluan yaitu dengan Melakukan praktik kerja lapangan di divisi *Human Resources Department* PT. Korea Nao Tech Indonesia selama dua bulan. Penulis juga melakukan studi pustaka dengan membaca buku literatur, *browsing internet*, membaca artikel, jurnal serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan dan semua yang berhubungan dengan judul dan permasalahan tugas akhir yang diambil oleh penulis.

2. Identifikasi Masalah

Pada tahap ini dilakukan cara dalam mengidentifikasi suatu masalah yang terjadi pada divisi *Human Resources Department* yang berfokus pada penghitungan upah pegawai harian. Dalam mengidentifikasi suatu masalah, penulis melakukan beberapa cara seperti:

- a. Melakukan wawancara dan observasi yang melibatkan PT Korea Nano Tech Indonesia dan beberapa karyawan yang bekerja pada Bagian Umum dan Divisi HRD
- b. Melakukan analisis hasil wawancara dan observasi
- c. Merumuskan masalah yang telah ditemukan

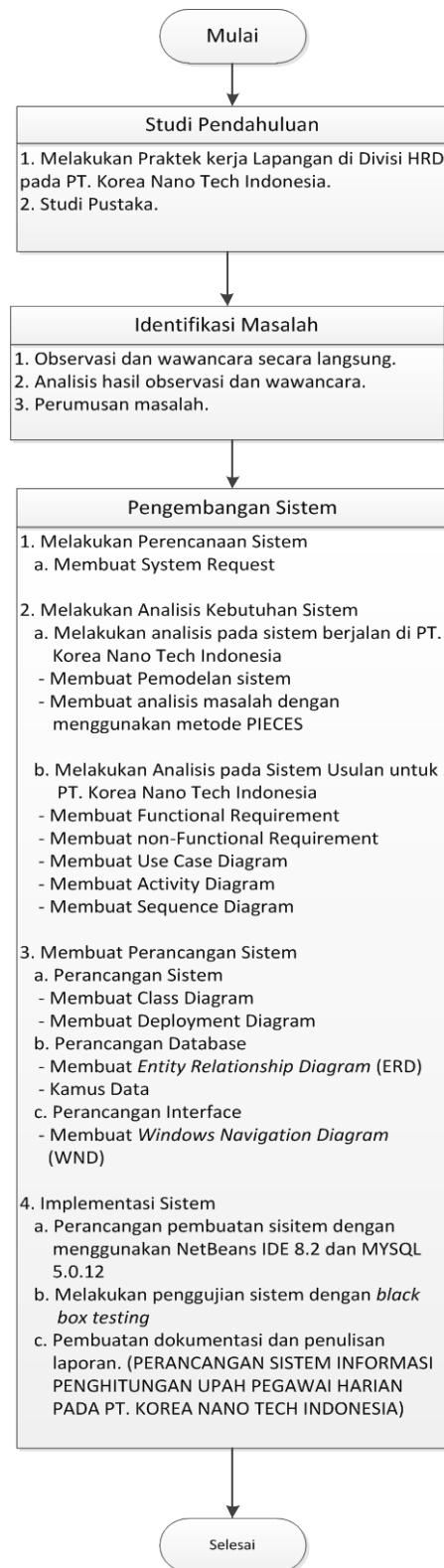
3. Perancangan Sistem

Setelah mengidentifikasi suatu masalah, maka penulis melakukan perancangan sistem sebagai bentuk solusi dari permasalahan tersebut. Terdapat beberapa tahapan dalam merancang sistem yang penulis lakukan seperti:

- a. Membuat *System Request*

- b. Melakukan analisis pada sistem berjalan untuk kebutuhan suatu sistem dengan cara membuat pemodelan sistem dan membuat analisis masalah dengan menggunakan metode analisis PIECES. Kemudian melakukan analisis pada sistem usulan dengan membuat *Functional Requirement* dan *Non-Functional Requirement*. Membuat *use case diagram*, *activity diagram*, dan *sequence diagram*.
- c. Melakukan desain perancangan sistem dengan cara membuat *Class diagram* dan *deploymen diagram*, merancang database dengan membuat *Entity Relationship Diagram* (ERD), *Logical relation structure* beserta kamus data, dan merancang infrastruktur sistem dengan cara membuat *Windows Navigator Diagram* (WND).
- d. Melakukan implementasi sistem dengan cara merancang dan membuat sistem dengan menggunakan *NetBeans 8.2* dan *MYSQL 5.0.12*. Selain itu juga penulis melakukan pengujian sistem dengan metode *blackbox testing* yang bertujuan untuk menguji coba apakah setiap fungsi atau tombol-tombol yang terdapat pada sistem dapat berjalan sesuai yang direncanakan atau diharapkan dan membuat dokumentasi dan penulisan laporan dengan judul “PERANCANGAN SISITEM INFORMASI PENGHITUNGAN UPAH PEGAWAI HARIAN PADA PT. KOREA NANO TECH INDONESIA”

Kerangka penelitian yang telah dibuat untuk laporan Tugas Akhir ini dapat dilihat pada Gambar III.1



Gambar III.1 Kerangka Penelitian
(Sumber: Pribadi, 2019)

BAB V

ANALISIS DAN PEMBAHASAN

5.1 *System Request*

System Request dari sistem informasi penghitungan upah pegawai harian yang diusulkan dapat dilihat pada tabel V.1 berikut:

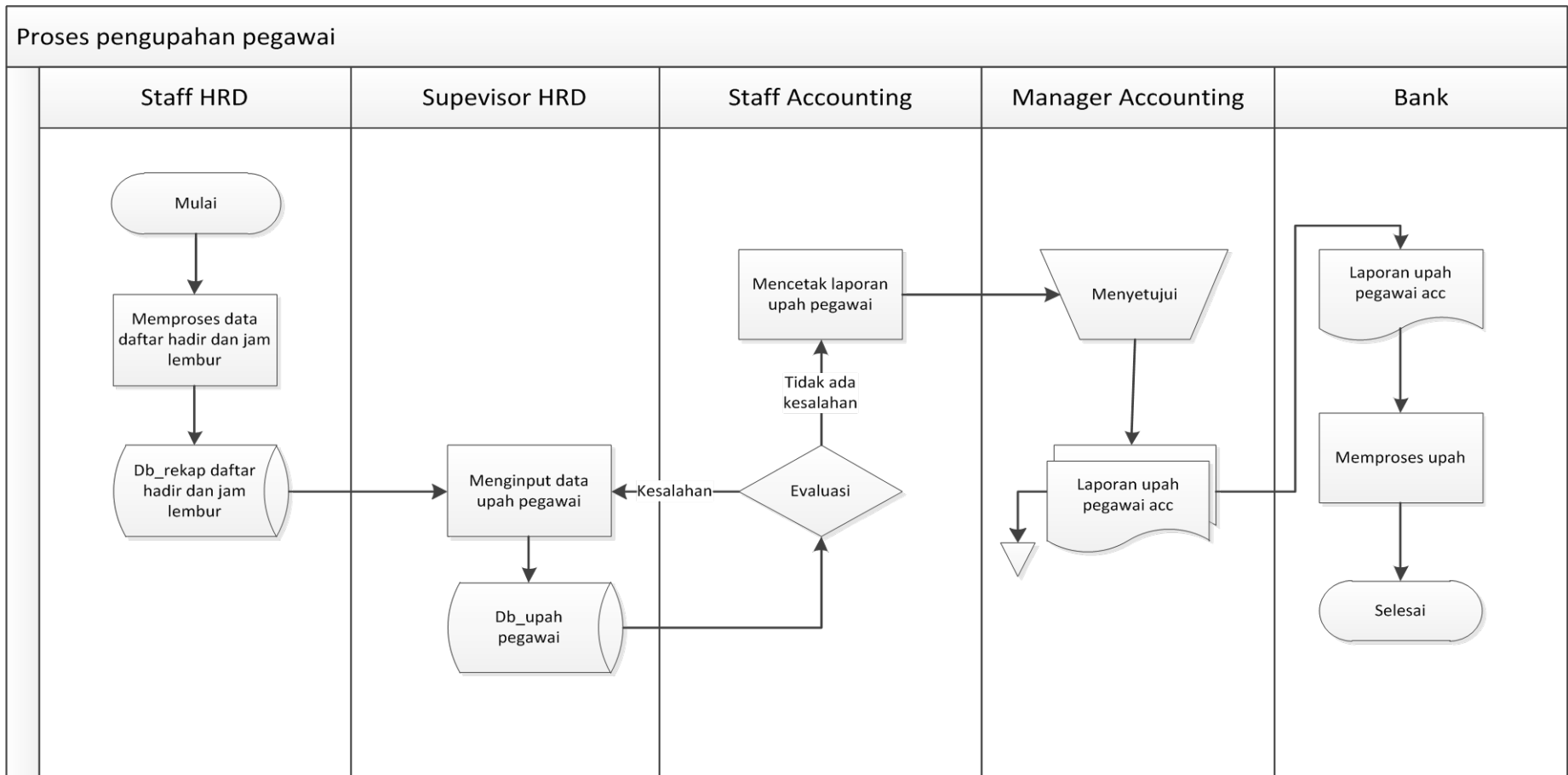
Tabel V.1 *System Request*

No	Project Elemen	Description
1	<i>Project sponsor</i>	<ul style="list-style-type: none"> • Divisi HRD PT Korea Nano Tech Indonesia.
2	<i>Business needs</i>	<ul style="list-style-type: none"> • Membantu <i>user</i> dalam melakukan perhitungan dan dapat optimalisasi waktu dalam proses pengupahan atau penggajian.
3	<i>Business requirement</i>	<ul style="list-style-type: none"> • Data <i>time card</i> pegawai. • Data rekap daftar hadir. • Data rekap Jam Lembur. • Laporan Upah Pegawai.
4	<i>Business value</i>	<ul style="list-style-type: none"> • Meminimalisir penggunaan kertas. • Mempercepat waktu proses penghitungan upah.
5	<i>Special issues or constrains</i>	Divisi HRD membuat aplikasi penghitungan upah atau gaji pegawai dan terintegrasi dengan <i>database</i> dengan demikian pembuatan laporan upah atau gaji pegawai lebih cepat selesai dengan penyimpanan data yang cukup besar kapasitasnya.

(Sumber: Pribadi, 2019)

5.2 **Proses Usulan Pengupahan Pegawai**

Beberapa tahapan untuk dapat melaksanakan perhitungan upah pegawai harian. Adapun prosesnya tertera pada gambar V.1 berikut :



Gambar V.1 Flowmap Usulan Proses Pengupahan Pegawai

(Sumber: Pribadi, 2019)

Beberapa tahapan untuk dapat melakukan proses penggajian pegawai. Adapun proses sebagai berikut :

1. *Staff* HRD melakukan proses data rekap daftar hadir dan jam lembur pegawai.
2. Melakukan input kedalam *database* rekap data daftar hadir dan jam lembur.
3. *Supervisor* HRD mendapatkan hasil rekap daftar hadir dan rekap jam lembur pegawai dari database rekap.
4. *Supervisor* HRD menginput daftar gaji pegawai berdasarkan database rekap daftar hadir dan jam lembur pegawai.
5. Setelah menyimpan ke dalam database *Supervisor* HRD menyerahkan ke pihak *staff Accounting* untuk di evaluasi.
6. *Staff Accounting* mengevaluasi database upah pegawai apabila terjadi kesalahan maka data akan di kembalikan ke *supervisor* HRD dengan menyantumkan catatan kedalam database. Apabila tidak terjadi kesalahan maka data akan langsung dicetak dalam bentuk laporan oleh *staff accounting*.
7. Laporan yang sudah di acc (ditanda tangani) oleh *manager accounting* akan didokumentasikan berdasarkan nama bulan dan tahun.
8. *Manager accounting* memberikan *copy* laporan upah kepada pihak bank.
9. Bank akan memproses gaji pegawai sampai selesai.

5.3 Analisis Kebutuhan Sistem

Kebutuhan-kebutuhan dari sistem informasi penghitungan gaji pegawai yang diusulkan dapat dilihat pada tabel V.2 berikut :

Tabel V.2 Analisis Kebutuhan Sistem

No	Masalah	Kebutuhan User	Solusi	Functional Requirement	Non Functional Requirement
1	Proses perhitungan upah pegawai harian yang relatif	Aplikasi khusus dalam penghitungan upah pegawai.	Membuat sistem aplikasi perhitungan upah yang	Fungsi penghitungan upah berdasarkan daftar hadir dan jam lembur.	Sistem dapat menghitung satu persatu data

Tabel V.2 Analisis Kebutuhan Sistem (Lanjutan)

No	Masalah	Kebutuhan User	Solusi	Functional Requirement	Non Functional Requirement
	memakan banyak waktu yang dapat menimbulkan keterlambatan dalam penyerahan upah.		dengan cepat menampilkan hasil upah setiap pegawai.		pegawai. - Dikelola oleh <i>supervisor</i> divisi HRD. - Menghitung dengan cepat dan akurat.
2	<ul style="list-style-type: none"> - Dalam proses pencatatan upah masih menggunakan banyak kertas. - Penumpukan dokumen dalam lemari arsip. - Tidak terkendalinya dokumen. 	Sistem yang dapat menyimpan data kedalam penyimpanan yang sudah terotomasi.	Membuat sistem yang dapat menyimpan data (<i>database</i>).	Mengelola data upah pegawai.	Dikelola oleh divisi HRD.
3	Sering terjadinya kesalahan dalam		Membuat aplikasi yang dapat menghitung.	Fungsi menampilkan notifikasi/ <i>note</i> kesalahan.	- Sistem dapat mengidentifikasi kesalahan

Tabel V.2 Analisis Kebutuhan Sistem (Lanjutan)

No	Masalah	Kebutuhan User	Solusi	Functional Requirement	Non Functional Requirement
	penghitungan upah pegawai.		upah pegawai dengan memberikan notifikasi/ <i>note</i> kesalahan.		input. - Dikelola oleh <i>supervisor</i> divisi HRD. - Dikelola oleh <i>staff accounting</i> .
4	Ketidaknyamanan divisi HRD dalam penghitungan upah dikarenakan proses terlalu panjang.		Aplikasi sistem yang dapat memudahkan <i>user</i> dalam melakukan proses penghitungan upah, mulai dari merekap daftar hadir hingga menjadi laporan upah.	- Fungsi menampilkan data rekap dari <i>database</i> . - Fungsi menampilkan <i>database</i> upah. - Fungsi pencetakan laporan upah yang terhubung ke dalam <i>database</i> .	- Tampilan menu sistem yang sederhana. - Mencetak laporan yang sudah di input kedalam <i>database</i> . - Dikelola oleh divisi HRD.

(Sumber: Pribadi, 2019)

5.3.1 Functional Requirement

Functional requirement dari sistem informasi penghitungan upah pegawai harian:

1. Fungsi penghitungan upah berdasarkan daftar hadir dan jam lembur.
2. Sistem dapat mengelola data upah pegawai.
3. Fungsi menampilkan data rekap dari *database*, menampilkan *database* upah, fungsi pencetakan laporan upah yang terhubung ke dalam *database*.
4. Fungsi menampilkan notifikasi kesalahan apabila terjadi kesalahan input.
5. Sistem dapat menampilkan hasil penghitungan upah mulai dari hasil upah dan hasil upah jam lembur.

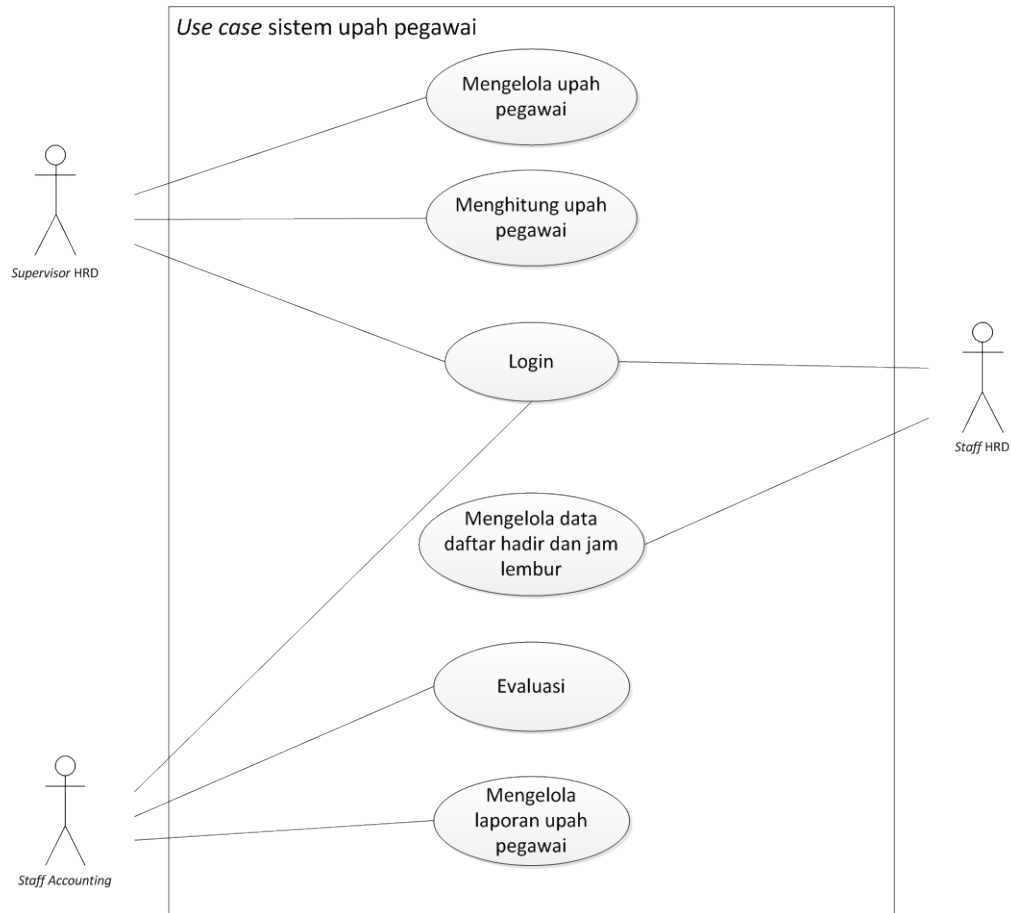
5.3.2 *Non Functional Requirement*

Non functional requirement dari sistem informasi penghitungan upah pegawai harian diusulkan antara lain:

1. Sistem yang dikelola oleh *supervisor* HRD ini hanya dapat menghitung satu persatu data pegawai proses pengelolaan berupa menghitung, menghapus, mengubah dan menambah data.
2. Proses tidak akan bisa berlanjut apabila sistem mengidentifikasi kesalahan input data hal ini mengakibatkan sistem tidak dapat menampilkan hasil (*error*).
3. Sistem dapat mengidentifikasi kesalahan input.
4. Apabila sistem mengidentifikasi kesalahan *username* dan *password* maka sistem tidak akan menampilkan menu utama.
5. Tampilan menu sistem yang sederhana dan dapat dengan mudah dioperasikan dan dimengerti oleh *user*.

5.4 *Use Case Diagram*

Use case diagram merupakan sebuah diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dan apa yang diperbuat oleh sistem. Berikut merupakan *use case diagram* usulan proses upah pegawai berdasarkan analisis kebutuhan sistem yang telah dibuat pada gambar V.2 berikut :



Gambar V.2 Use Case diagram usulan Sistem Informasi upah pegawai

(Sumber: Hasil analisis, 2019)

5.4.1 Use Case Description

Berikut merupakan *use case description* usulan sistem upah pegawai dari *use case* usulan yang telah dibuat pada tabel V.3:

Tabel V.3 Case Description Login

Nama Use Case	Login
Aktor	Supervisor HRD, Staff HRD, Staff Accounting
Deskripsi	Use case ini menggambarkan beberapa aktor dapat masuk kedalam sistem.
Normal Flow of Events	1. Aktor membuka login form. 2. Aktor memasukkan username dan password pada login

Tabel V.3 *Case Description Login* (Lanjutan)

	<p><i>form.</i></p> <p>3. Sistem akan memvalidasi <i>username</i> dan <i>password</i>.</p> <p>4. A. Apabila sesuai, maka aktor akan masuk kedalam menu utama sistem.</p> <p>5. B. Jika salah, maka aktor akan mendapatkan <i>notification</i> pada <i>login form</i> yang menandakan bahwa data tidak sesuai.</p>
<i>Alternate Flow</i>	<p>4.B</p> <pre> graph TD Mulai([Mulai]) --> Input[Memasukan username dan password] Input --> Validasi{Validasi} Validasi -- Salah --> Input Validasi -- Benar --> Menu[Masuk Menu Utama] Menu --> Selesai([Selesai]) </pre>

(Sumber: Hasil Analisis, 2019)

Tabel V.4 *Case Description* Mengelola Data Daftar Hadir dan Jam Lembur

Nama <i>Use Case</i>	Mengelola data daftar hadir dan jam lembur
Aktor	<i>Staff</i> HRD
Deskripsi	<i>Use case</i> ini menggambarkan <i>staff</i> HRD merekap data daftar hadir.

Tabel V.4 *Case Description* Mengelola Data Daftar Hadir dan Jam Lembur
(Lanjutan)

<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Staff</i> HRD mengumpulkan <i>time card</i> setiap pegawai 2. <i>Staff</i> HRD memasukan data daftar hadir dan daftar jam lembur yaitu nama, nik, hari kerja, jam lembur, bulan dan tahun. 3. Data dimasukan kedalam <i>database</i> rekap. 4. Data yang sudah masuk akan diberikan kepada <i>supervisor</i> HRD.
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2019)

Tabel V.5 *Case Description* Mengelola Upah Pegawai

Nama <i>Use Case</i>	Mengelola upah pegawai
Aktor	<i>Supervisor</i> HRD
Deskripsi	<i>Use case</i> ini menggambarkan <i>supervisor</i> HRD mengelola penghitungan upah pegawai.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Masuk kedalam sistem dengan menggunakan <i>username</i> dan <i>password</i>. 2. <i>Supervisor</i> HRD akan masuk ke menu aplikasi penghitungan upah <i>user</i> dadapat menambah, mengubah dan menghapus data. 3. Hasil data akan di tampilkan didalam tabel upah dan akan masuk secara otomatis keladam database upah pegawai. 4. <i>Supervisor</i> HRD dapat melakukan penghapusan data, mengubah, <i>reset</i> dan menambah data.
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2019)

Tabel V.6 *Case Description* Menghitung Upah Pegawai

Nama <i>Use Case</i>	Menghitung upah pegawai
Aktor	<i>Supervisor</i> HRD
Deskripsi	<i>Use case</i> ini menggambarkan <i>supervisor</i> HRD

Tabel V.6 *Case Description* Menghitung Upah Pegawai (Lanjutan)

	melakukan proses penghitungan upah pegawai.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Masuk kedalam sistem dengan menggunakan <i>username</i> dan <i>password</i>. 2. <i>Supervisor</i> HRD akan masuk ke menu aplikasi penghitungan upah dan mulai memproses upah pegawai dengan menarik data dari <i>database</i> rekap dan menghitung hasil upah didalam sistem. 3. Hasil upah pegawai akan di tampilkan didalam tabel upah dan akan masuk secara otomatis kedalam database upah pegawai.
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2019)

Tabel V.7 *Case Description* Mengelola Laporan Upah Pegawai

Nama <i>Use Case</i>	Mengelola laporan upah pegawai
Aktor	<i>Staff Accounting</i>
Deskripsi	<i>Use case</i> ini berfungsi untuk membuat laporan hasil proses penghitungan upah pegawai.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Staff accounting</i> login kedalam sistem 2. Sistem menampilkan menu utama 3. <i>Staff accounting</i> menekan <i>button report</i> upah. 4. Sistem akan menampilkan laporan upah pegawai. 5. <i>Staff accounting</i> memasukan periode laporan yang akan ditampilkan. 6. <i>Staff accounting</i> sudah dapat melakukan pencetakan laporan dengan menggunakan mesin <i>print</i>. 7. <i>Staff accounting</i> melakukan penyimpanan laporan kedalam laptop atau PC.
<i>Alternate Flow</i>	

(Sumber: Hasil Analisis, 2019)

Tabel V.8 *Case Description* Evaluasi

Nama <i>Use Case</i>	Evaluasi
Aktor	<i>Staff accounting</i>
Deskripsi	<i>Use case</i> ini menggambarkan <i>staff accounting</i> melakukan evaluasi terhadap penghitungan upah pegawai
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Staff accounting</i> login kedalam sistem 2. Memilih menu aplikasi evaluasi 3. <i>Staff accounting</i> memanggil <i>database</i> penghitungan upah ke dalam aplikasi 4. Evaluasi dilakukan dengan memeriksa satuper satu data yang masuk kedalam <i>database</i> upah pegawai 5. Apabila terjadi kesalahan makan <i>Staff accounting</i> akan menyertakan <i>note</i> yang berisi tentang kesalahan tersebut untuk diperbaiki oleh <i>supervisor</i> HRD
<i>Alternate Flow</i>	

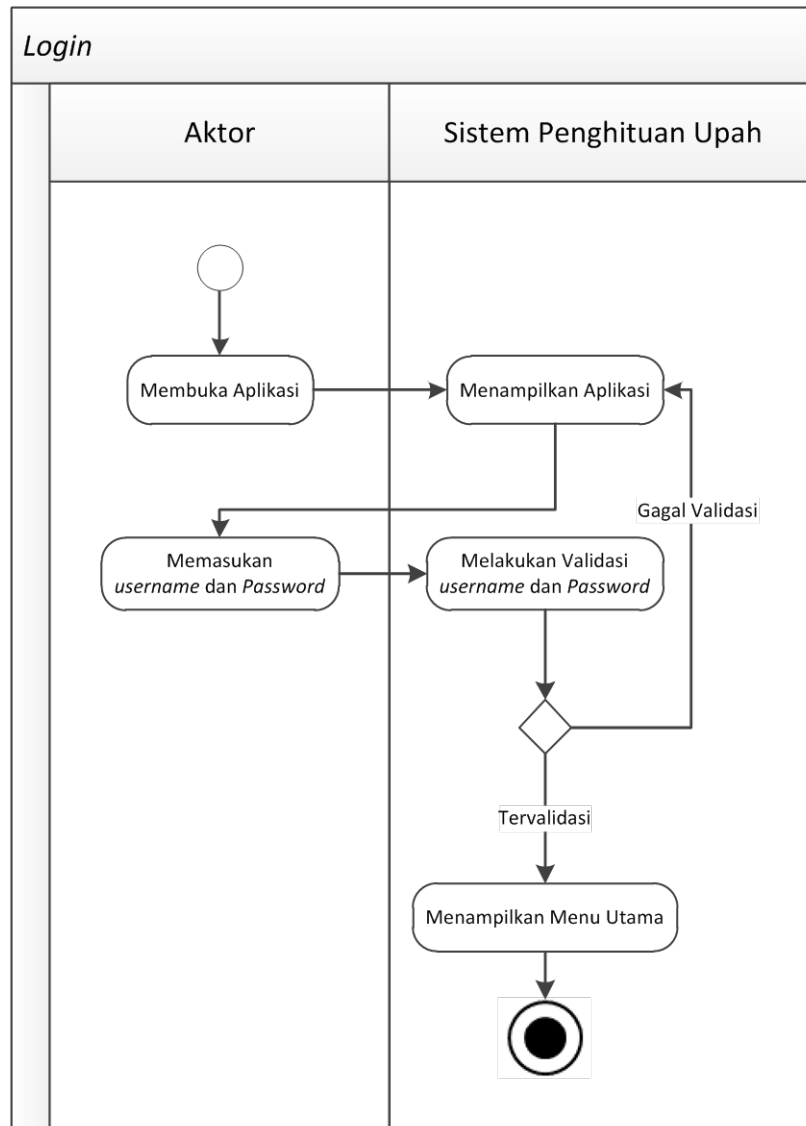
(Sumber: Hasil Analisis, 2019)

5.5 Activity Diagram

Activity diagram merupakan sebuah diagram yang menggambarkan urutan aktivitas proses bisnis pada suatu sistem. Berikut merupakan beberapa *activity diagram* usulan dari sistem informasi penghitungan upah pegawai:

1. Activity Diagram Login

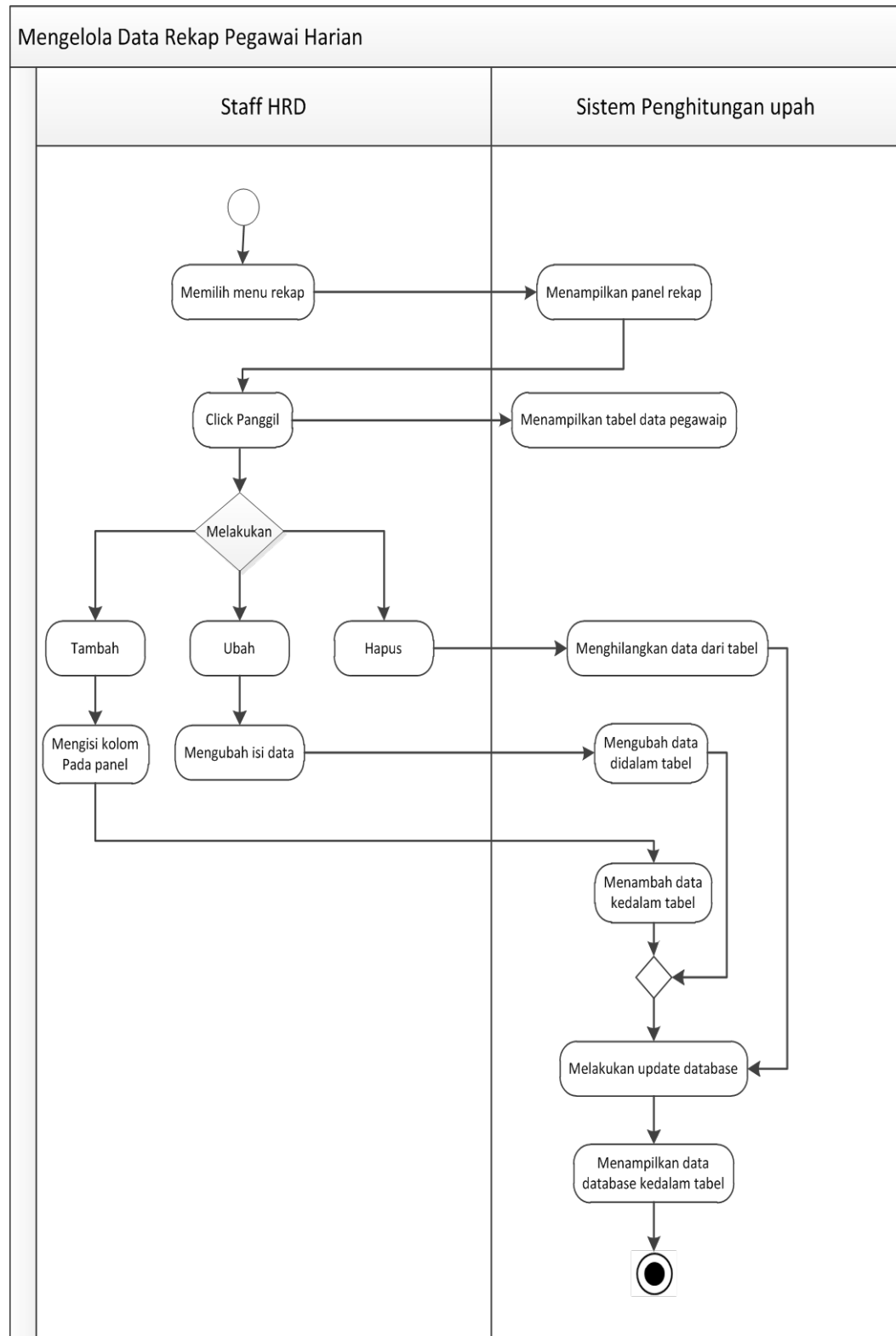
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user*, yaitu beberapa aktor seperti *Supervisor* HRD dan *staff* HRD PT. Korea Nano Tech Indonesia untuk dapat mengakses sistem informasi penghitungan upah pegawai. Berikut merupakan *activity diagram login*:



Gambar V.3 Activity Diagram Login
(Sumber: Hasil Analisis, 2019)

2. Activity Diagram Mengelola Data Daftar Hadir dan Jam Lembur

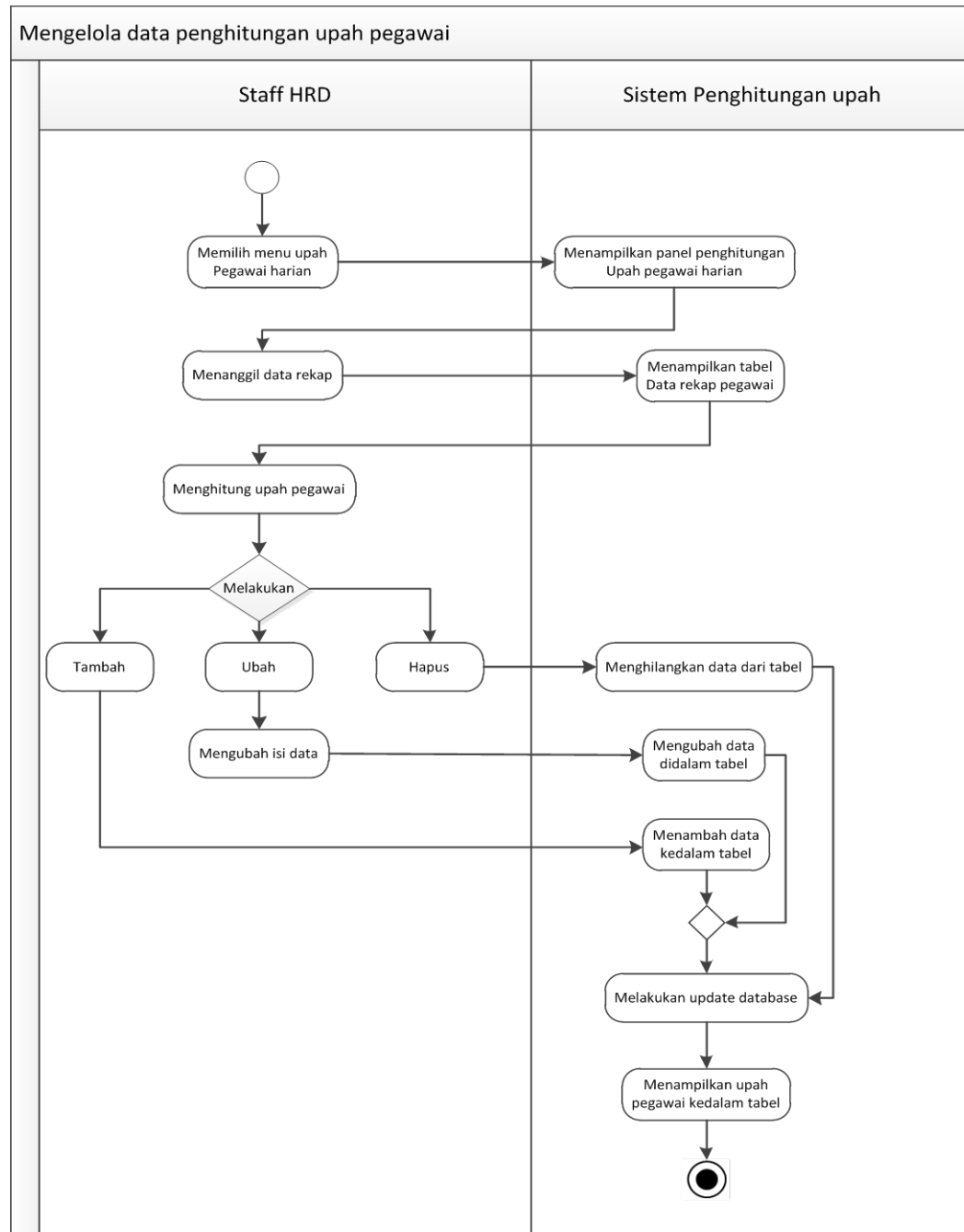
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *staff* HRD pada PT. Korea Nano Tech Indonesia, dalam melakukan proses perekapan data daftar hadir dan jam lembur pegawai harian, yang mana data rekap tersebut sebagai data utama dalam penghitungan upah pegawai harian:



Gambar V.4 Activity Diagram Mengelola Data Daftar Hadir dan Jam Lembur
(Sumber: Hasil Analisis, 2019)

3. Activity Diagram Mengelolah Upah Pegawai

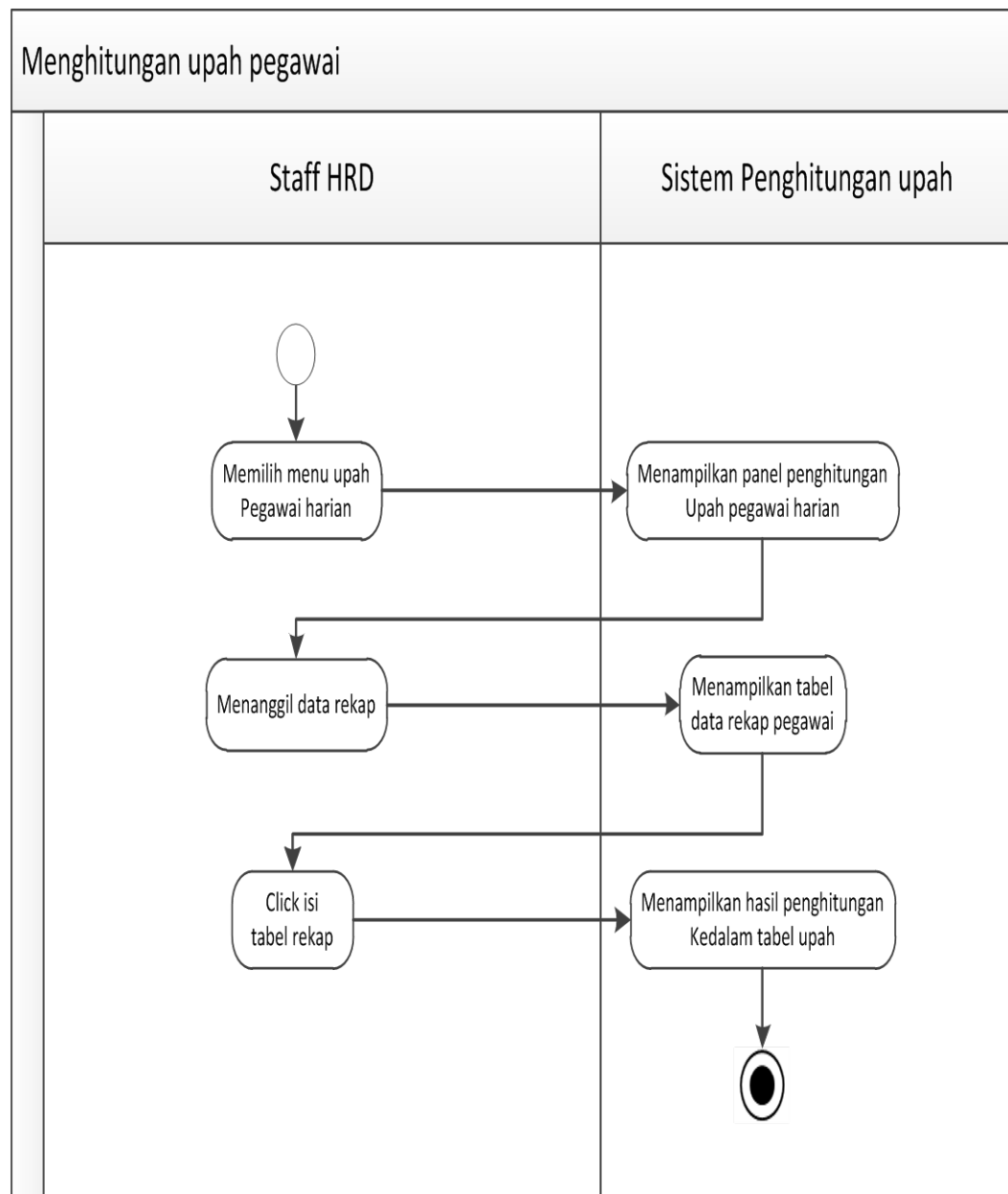
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *supervisor* HRD pada PT. Korea Nano Tech Indonesia, dalam melakukan proses penghitungan upah pegawai harian berdasarkan daftar hadir dan jam lembur pegawai :



Gambar V.5 Activity Diagram Mengelola Data Penghitungan Upah Pegawai
(Sumber: Hasil Analisis, 2019)

4. *Activity Diagram Menghitung Upah pegawai*

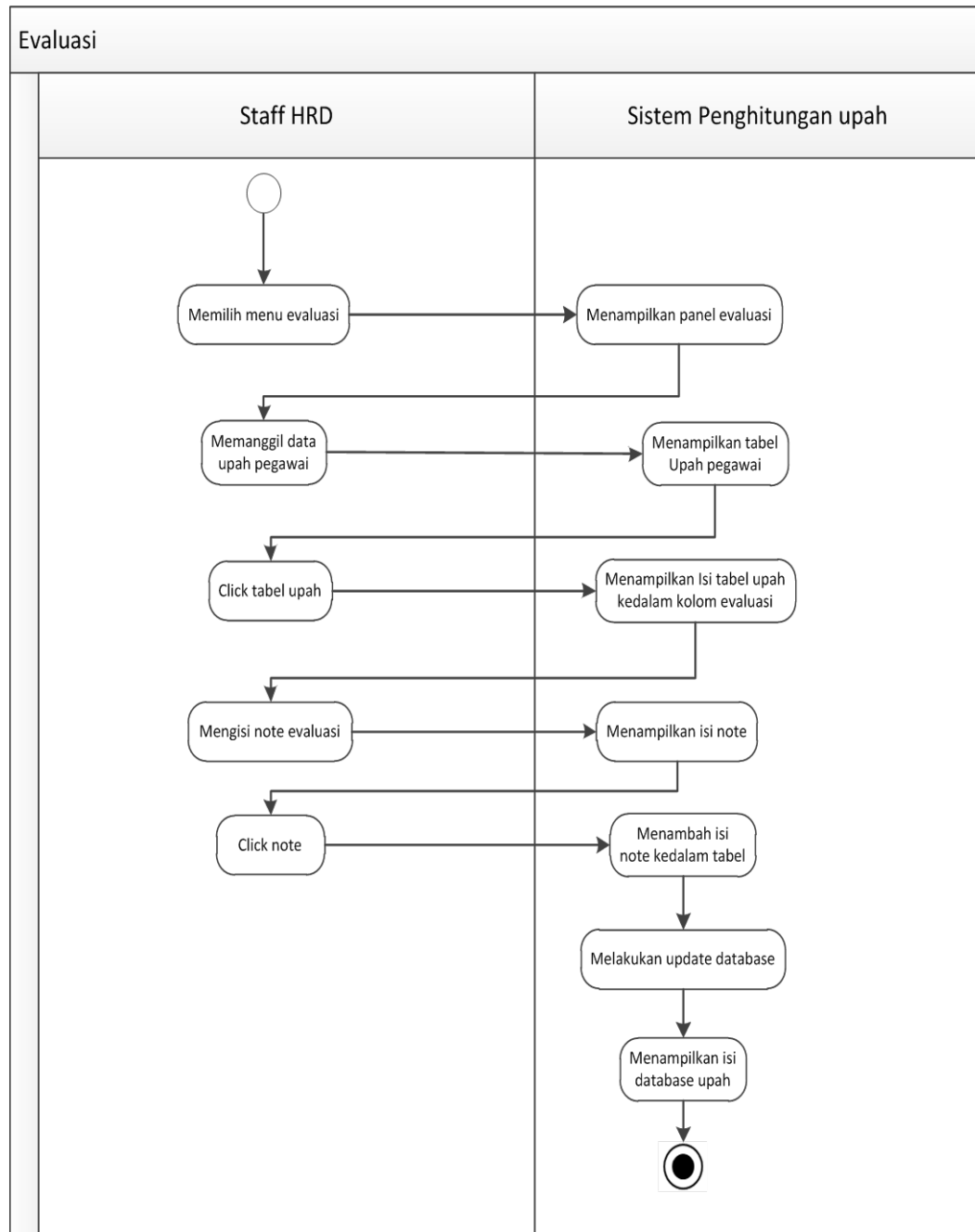
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh supervisor HRD pada PT. Korea Nano Tech Indonesia, dalam melakukan proses penghitungan upah pegawai yang bertujuan untuk menjumlahkan hasil upah kerja harian dan upah kerja jam lembur :



Gambar V.6 *Activity Diagram Menghitung Upah Pegawai*
(Sumber: Hasil Analisis, 2019)

5. Activity Diagram Evaluasi

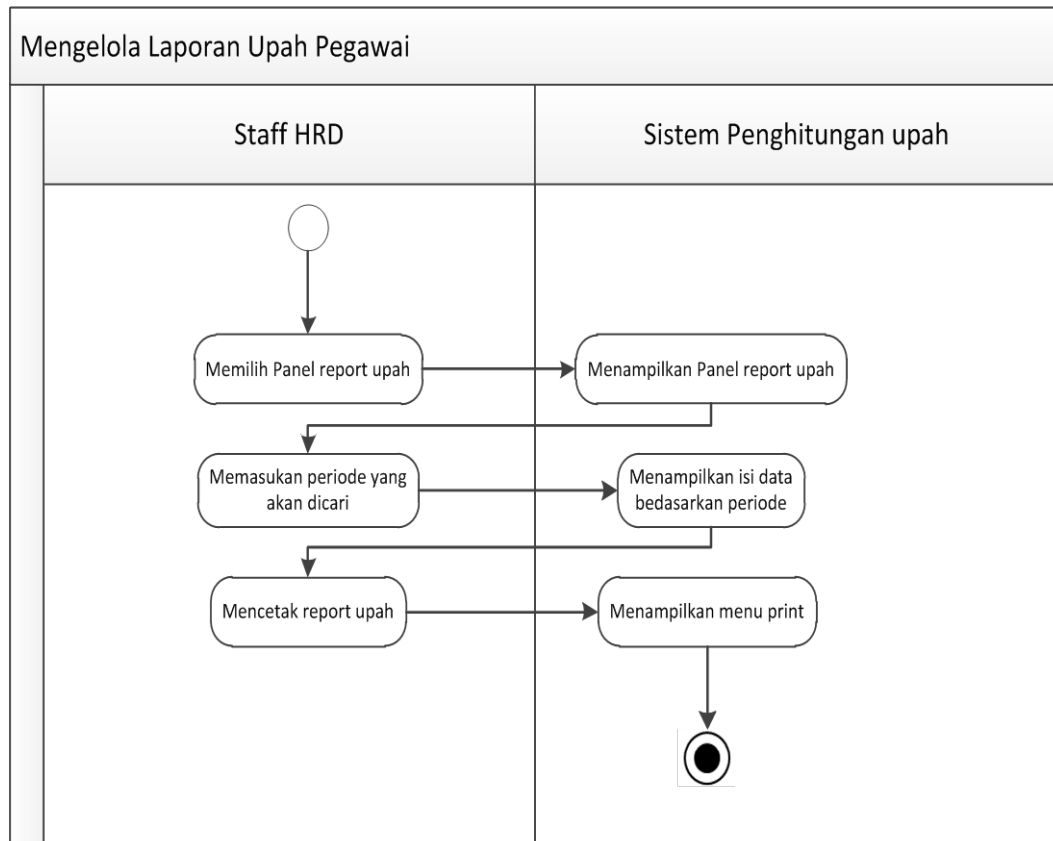
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *staff accounting* pada PT. Korea Nano Tech Indonesia, dalam melakukan proses evaluasi yang bertujuan untuk melakukan perbaikan pada penghitungan upah pegawai harian pada perusahaan :



Gambar V.7 Activity Diagram Evaluasi
(Sumber: Hasil Analisis, 2019)

6. Activity Diagram Mengelola Laporan Upah Pegawai

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *staff accounting* pada PT. Korea Nano Tech Indonesia, dalam melakukan proses evaluasi yang bertujuan untuk melakukan perbaikan pada penghitungan upah pegawai harian pada perusahaan :



Gambar V.8 Activity Mengelola Laporan Upah Pegawai
(Sumber: Hasil Analisis, 2019)

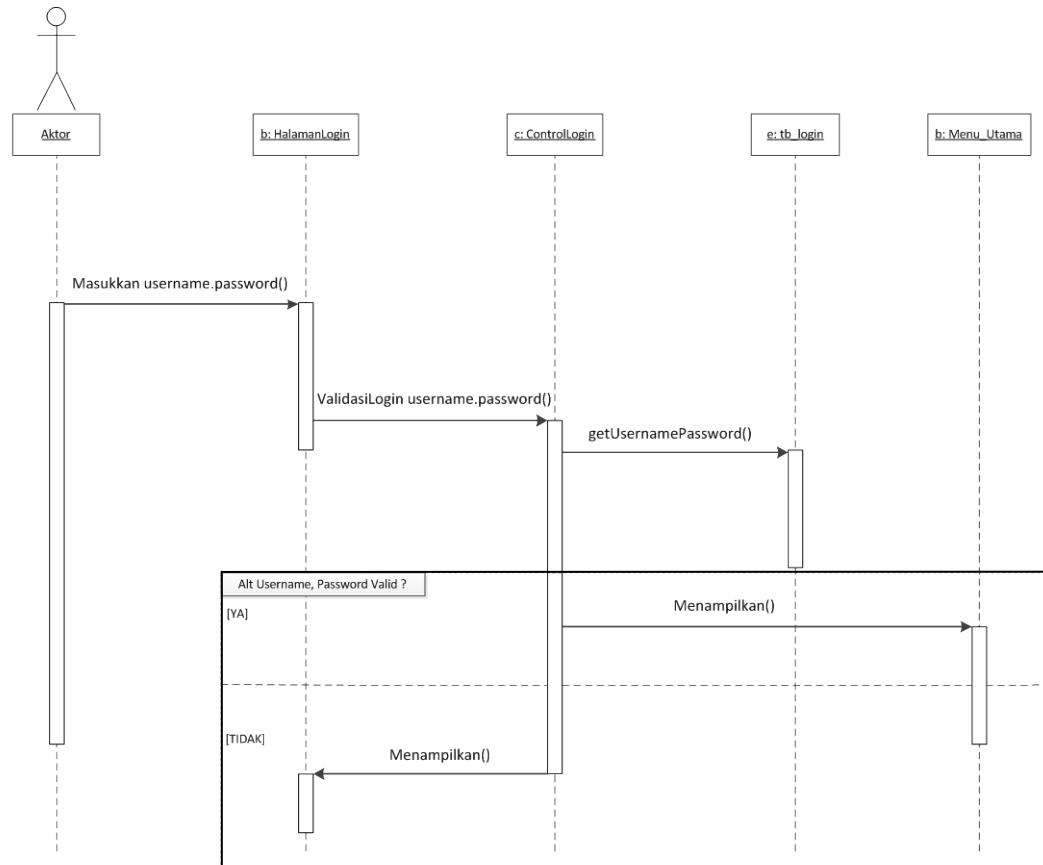
5.6 Sequence Diagram

Sequence diagram merupakan diagram untuk menggambarkan skenario menghasilkan output rangkaian langkah yang dilakukan sebagai respon dari sebuah event dalam menghasilkan *output* tertentu. Berikut beberapa *sequence diagram* usulan dari sistem informasi perhitungan upah pegawai harian :

1. Sequence Diagram Login

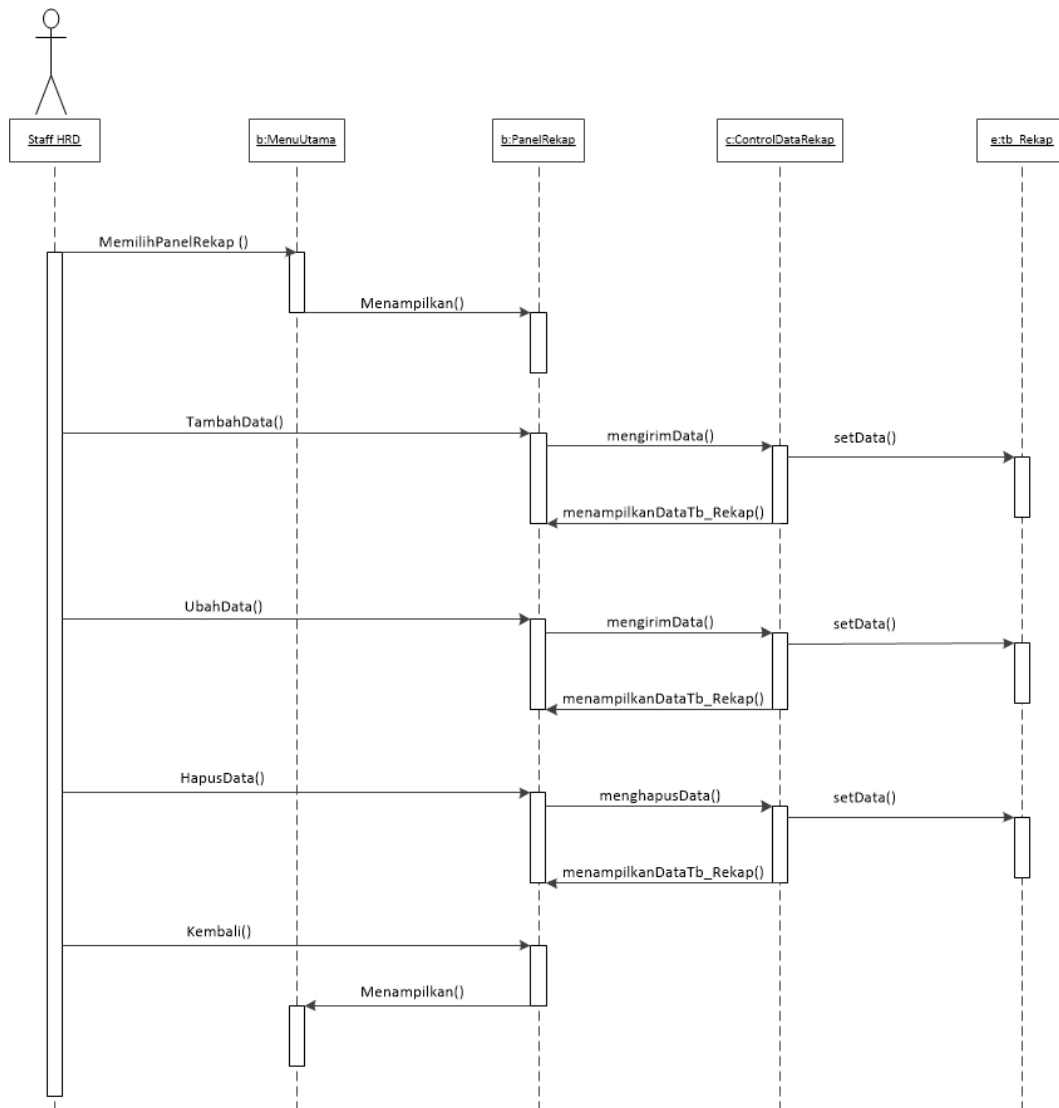
Sequence diagram *Login* merupakan sebuah diagram *Sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses *login*. Didalam

sistem informasi perhitungan upah pegawai harian ini, *login* diperlukan agar seorang pengguna dapat mengakses sistem tersebut. Berikut merupakan *sequence diagram* dari *use case login* :



Gambar V.9 *Sequence Diagram Login*
(Sumber: Hasil Analisis, 2019)

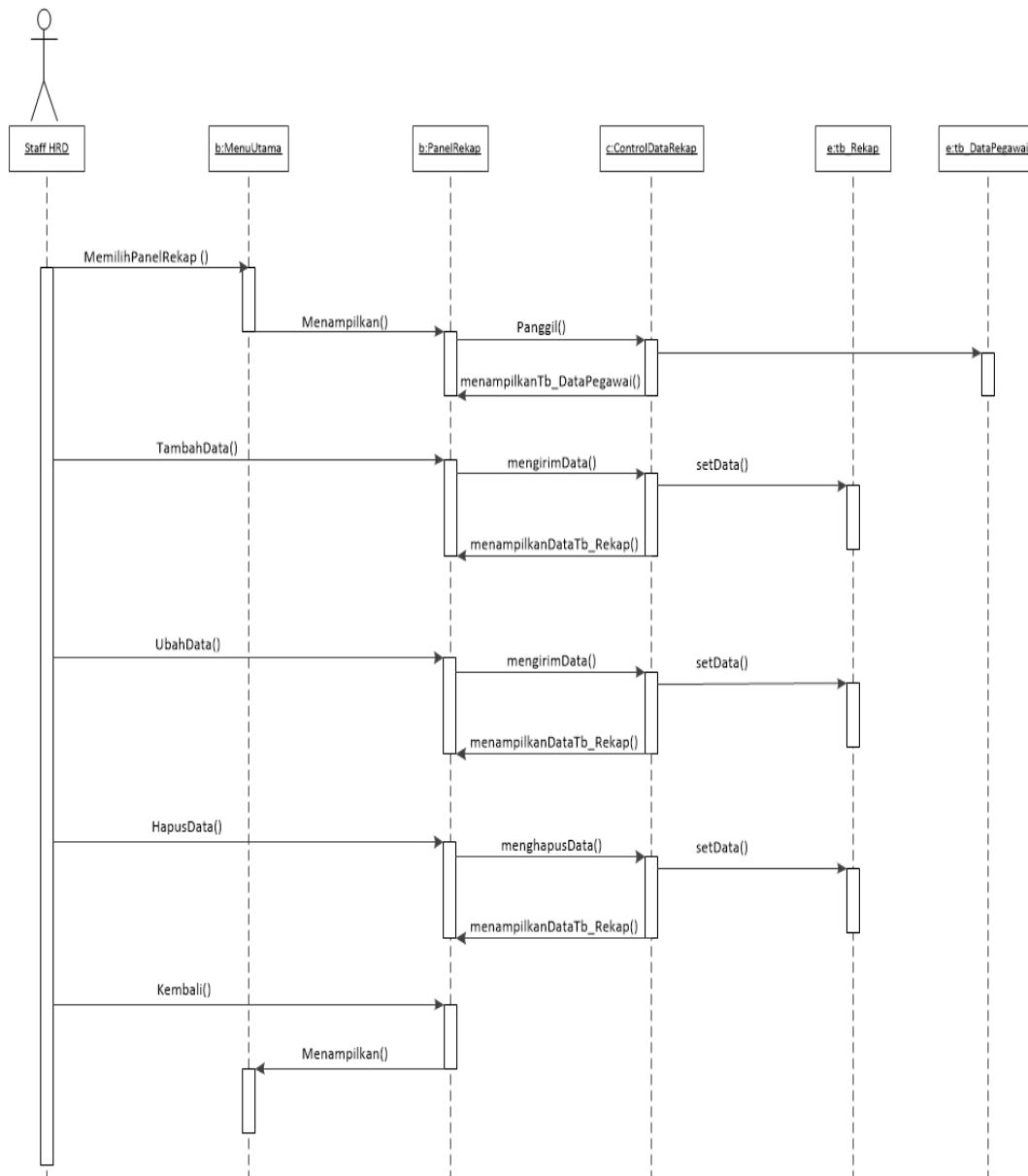
2. *Sequence Diagram Mengelola Data Daftar Hadir dan Jam Lembur*
Sequence diagram mengelola data daftar hadir dan jam lembur merupakan sebuah diagram *Sequence* yang berfungsi untuk melakukan proses perekapan data hadir dan jam lembur yang mana data ini sebagai acuan untuk melakukan proses selanjutnya. Berikut merupakan *sequence diagram* dari *use case* mengelola data daftar hadir dan jam lembur :



Gambar V.10 *Sequence* Mengelola Data Daftar Hadir dan Jam Lembur
(Sumber: Hasil Analisis, 2019)

3. *Sequence* Diagram Mengelola Upah Pegawai

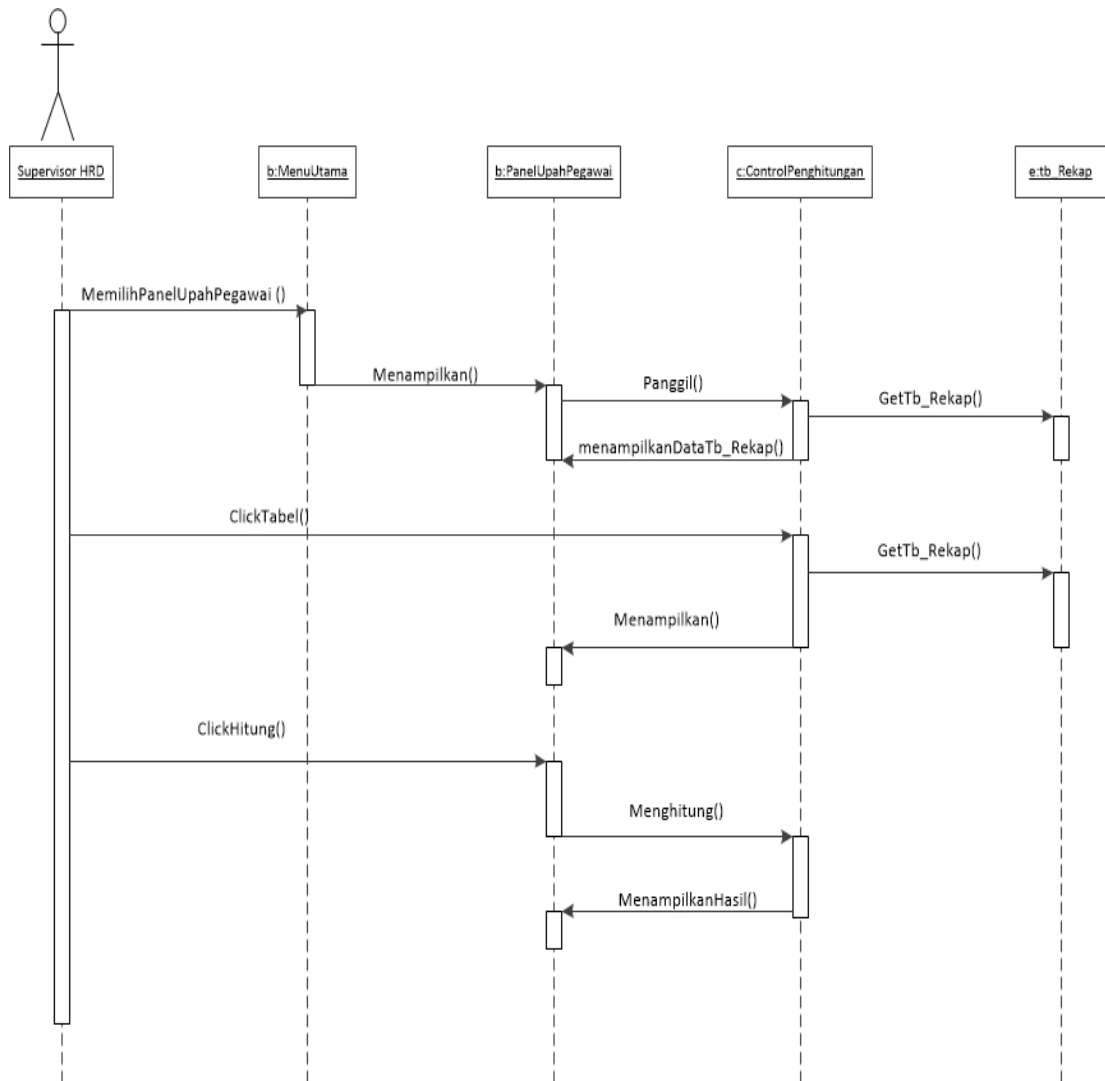
Sequence diagram mengelola upah pegawai merupakan sebuah diagram *Sequence* yang berfungsi untuk melakukan proses pengelolaan upah pegawai yang dilakukan oleh *supervisor* HRD dimana user dapat menambah, mengubah dan menghapus data. Berikut merupakan *sequence* diagram dari *use case* mengelola upah pegawai :



Gambar V.11 *Sequence* Mengelola Upah Pegawai
(Sumber: Hasil Analisis, 2019)

4. *Sequence* Diagram Menghitung Upah Pegawai

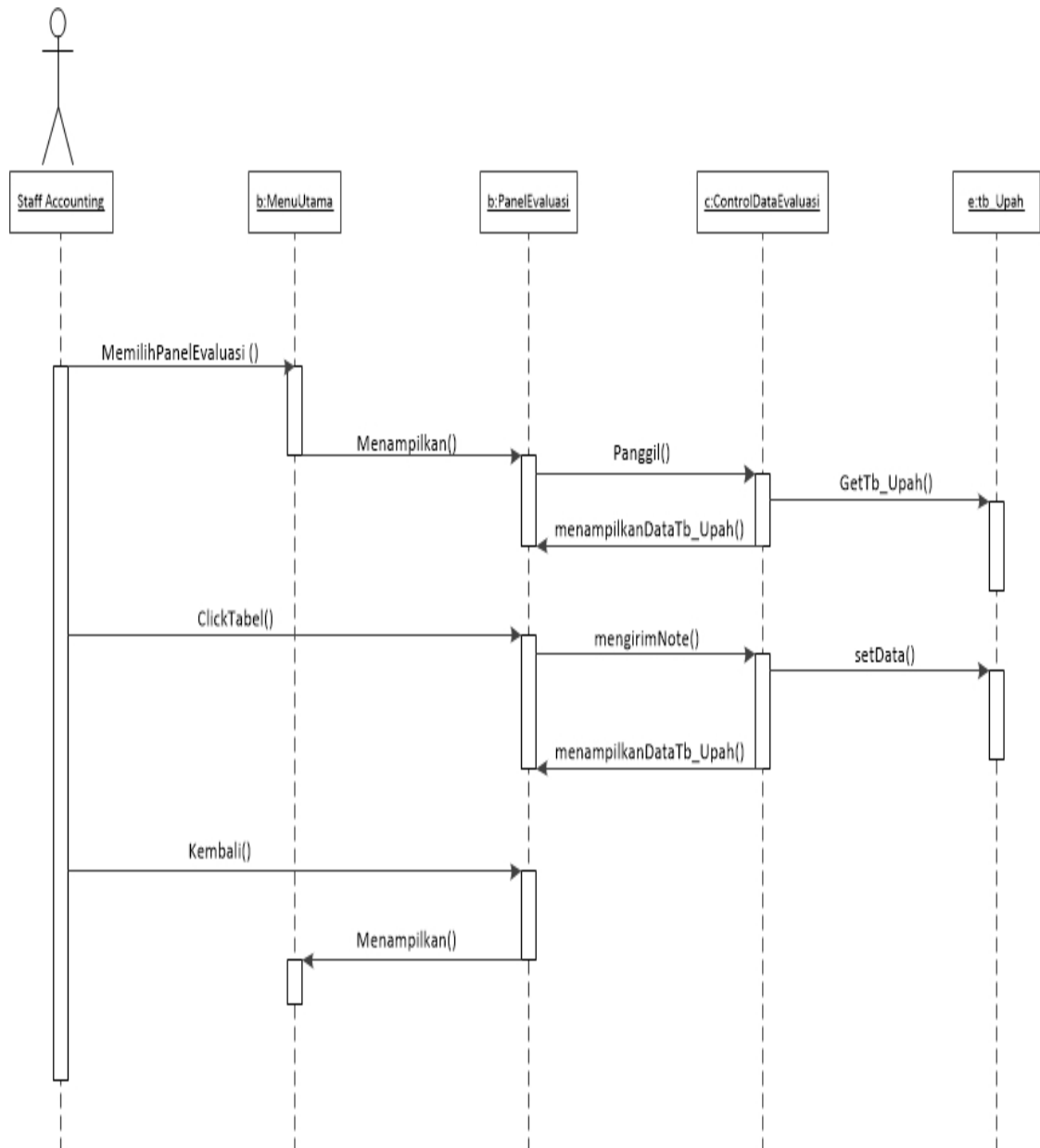
Sequence diagram menghitung upah pegawai merupakan sebuah diagram *Sequence* yang berfungsi untuk melakukan proses penghitungan upah pegawai yang dilakukan berdasarkan data rekap daftar hadir dan jam lembur setiap pegawai. Berikut merupakan *sequence* diagram dari *use case* menghitung upah pegawai :



Gambar V.12 *Sequence* Menghitung Upah Pegawai
(Sumber: Hasil Analisis, 2019)

5. *Sequence* Diagram Evaluasi

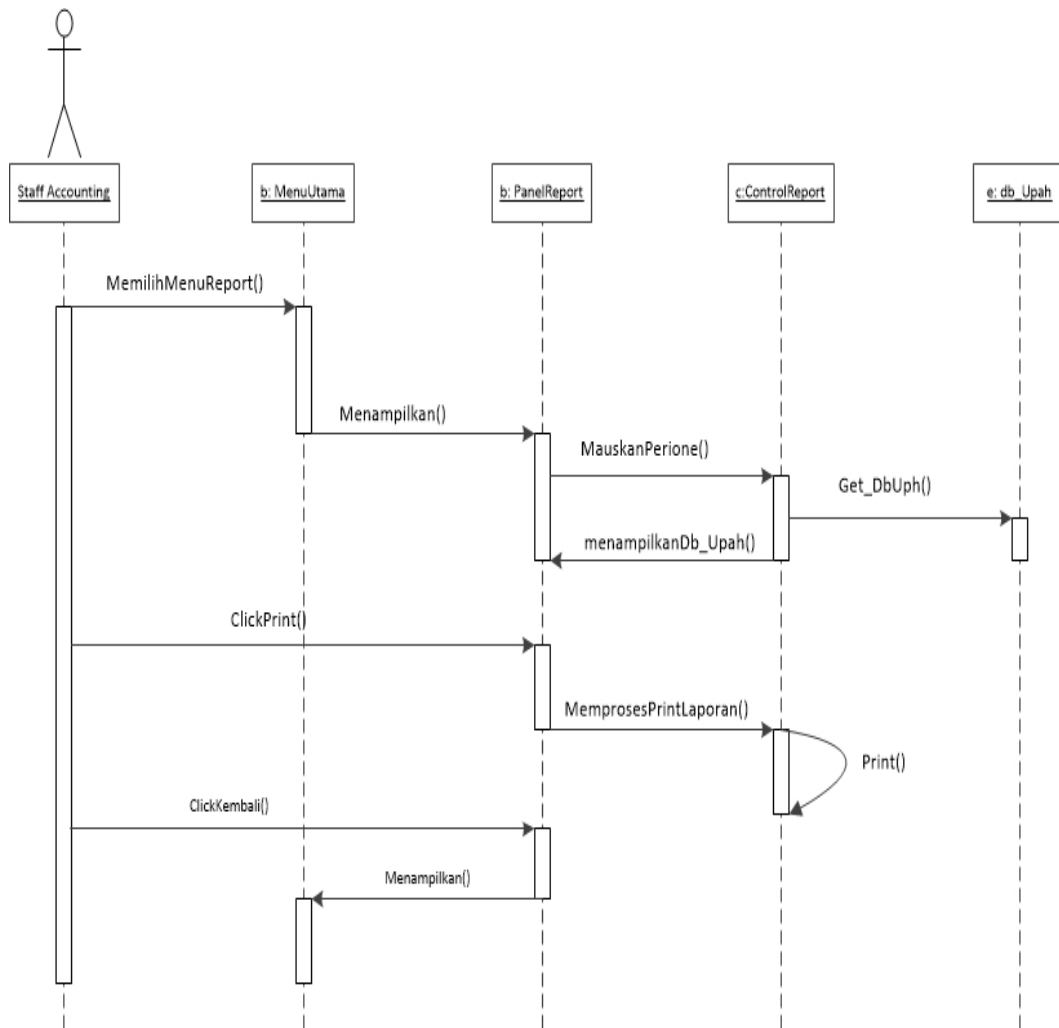
Sequence diagram evaluasi sebuah diagram *Sequence* yang berfungsi untuk melakukan proses pengoreksian data upah pegawai yang sudah diinput sebelumnya kedalam *database* upah pegawai. Berikut merupakan *sequence* diagram dari *use case* evaluasi :



Gambar V.13 *Sequence* Evaluasi
(Sumber: Hasil Analisis, 2019)

6. *Sequence* Diagram Mencetak Laporan Upah Pegawai

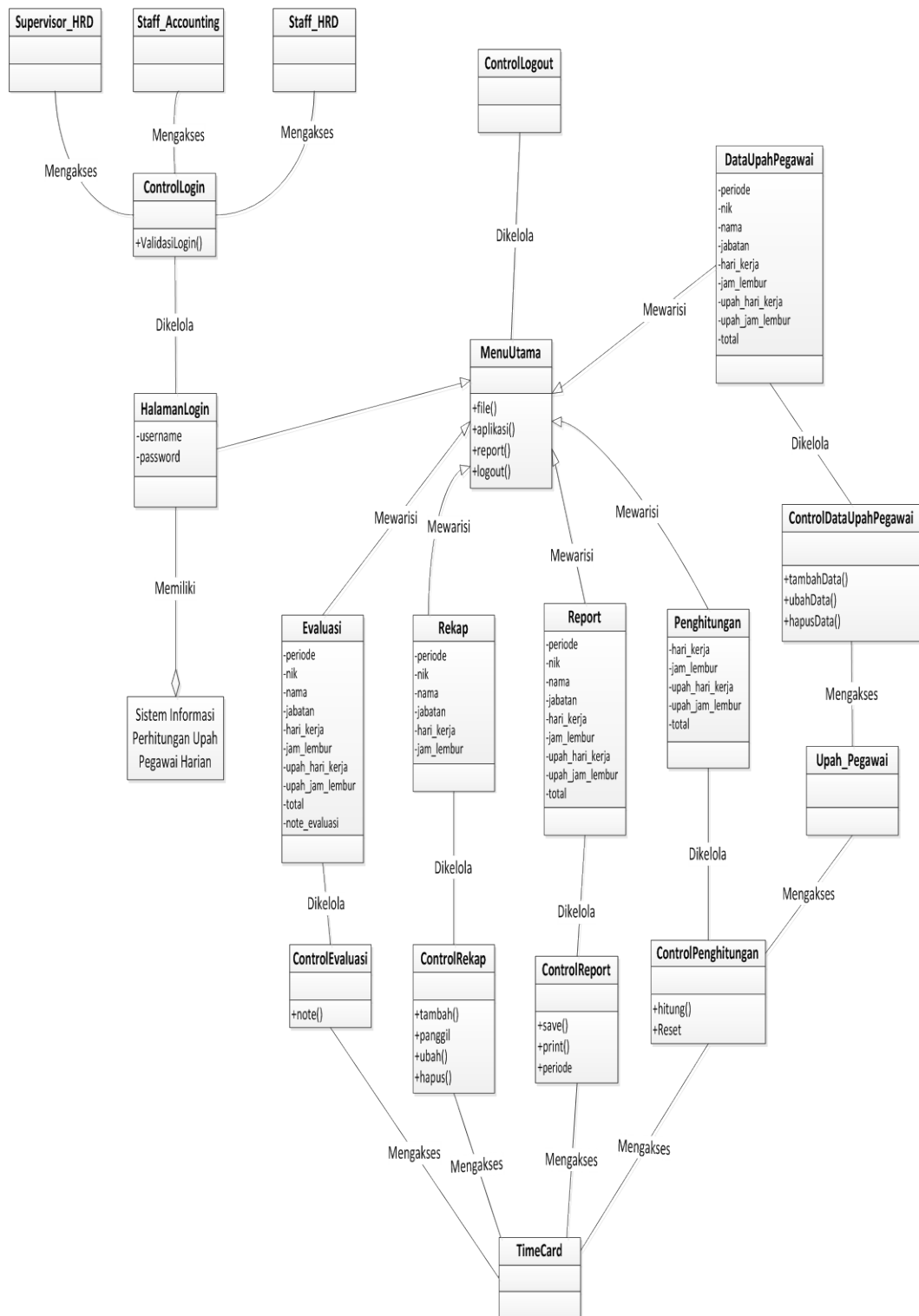
Sequence diagram mencetak laporan upah pegawai sebuah diagram *Sequence* yang berfungsi untuk melakukan proses pencetakan laporan berdasarkan *database* upah pegawai yang sudah di buat sebelumnya. Berikut merupakan *sequence* diagram dari *use case* mencetak laporan upah pegawai :



Gambar V.14 *Sequence* Mengelola Laporan Upah Pegawai
(Sumber: Hasil Analisis, 2019)

5.7 Class Diagram

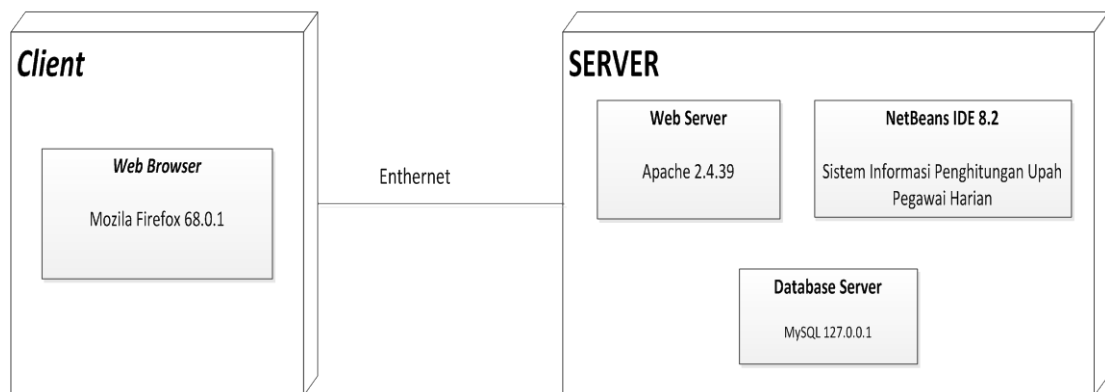
Class Diagram pada usulan sistem informasi penghitungan upah pegawai ini digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat. Untuk membangun sistem, *class* diagram dapat memperlihatkan hubungan antar kelas dan penjelasan detail pada setiap kelas di dalam model desain dari suatu sistem. Berikut merupakan *class* diagram usulan sistem informasi penghitungan upah pegawai :



Gambar V.15 *Class Diagram Menghitung Upah Pegawai*
(Sumber: Hasil Analisis, 2019)

5.8 Deployment Diagram

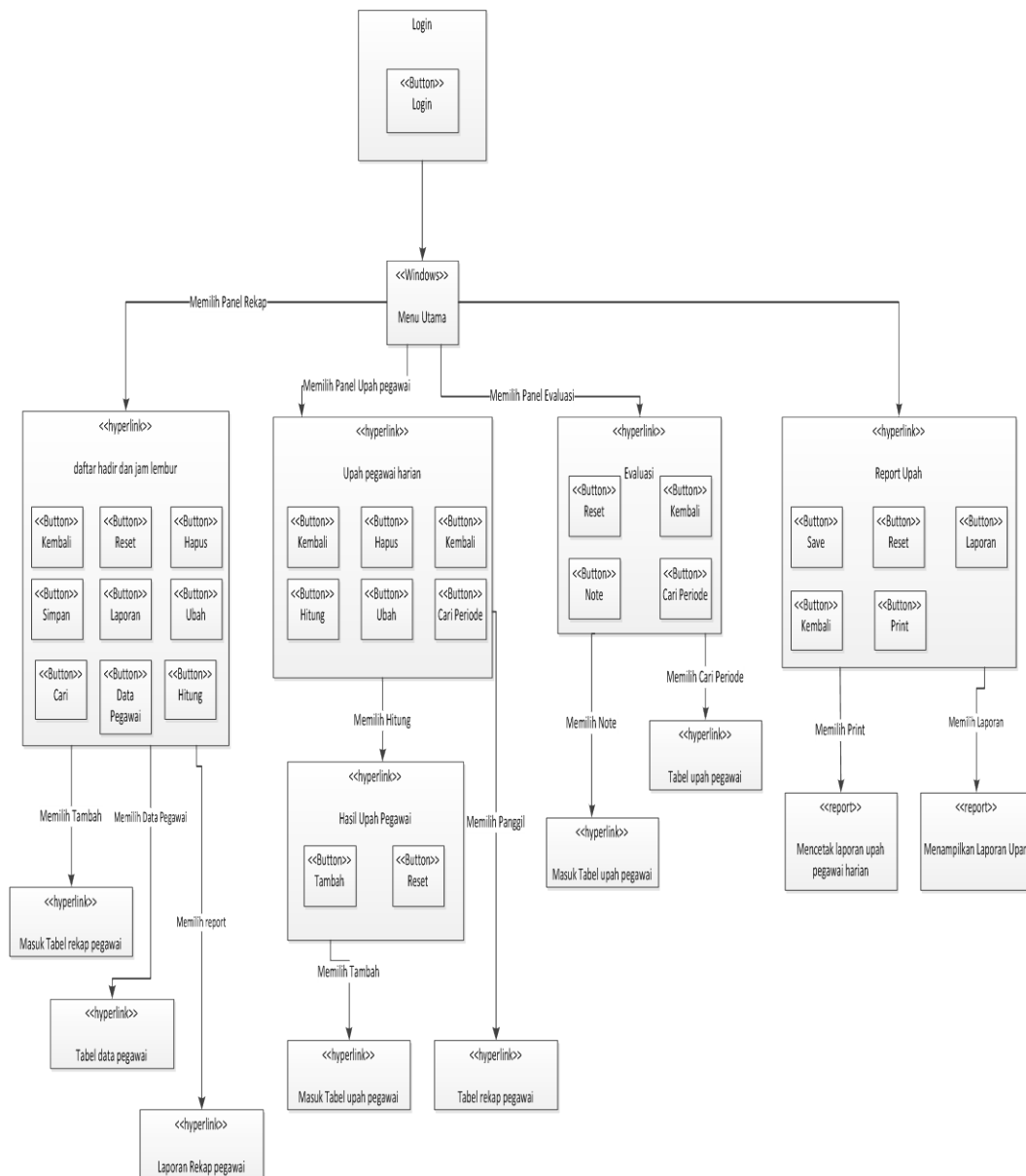
Deployment diagram pada usulan sistem informasi penghitungan upah pegawai harian ini digunakan untuk mewakili komponen-komponen perangkat lunak dan cara perangkat lunak tersebut ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Berikut ini merupakan *deployment* diagram sistem informasi penghitungan upah pegawai harian :



Gambar V.16 *Deployment* Diagram Menghitung Upah Pegawai
(Sumber: Hasil Analisis, 2019)

5.9 Windows Navigation Diagram (WND)

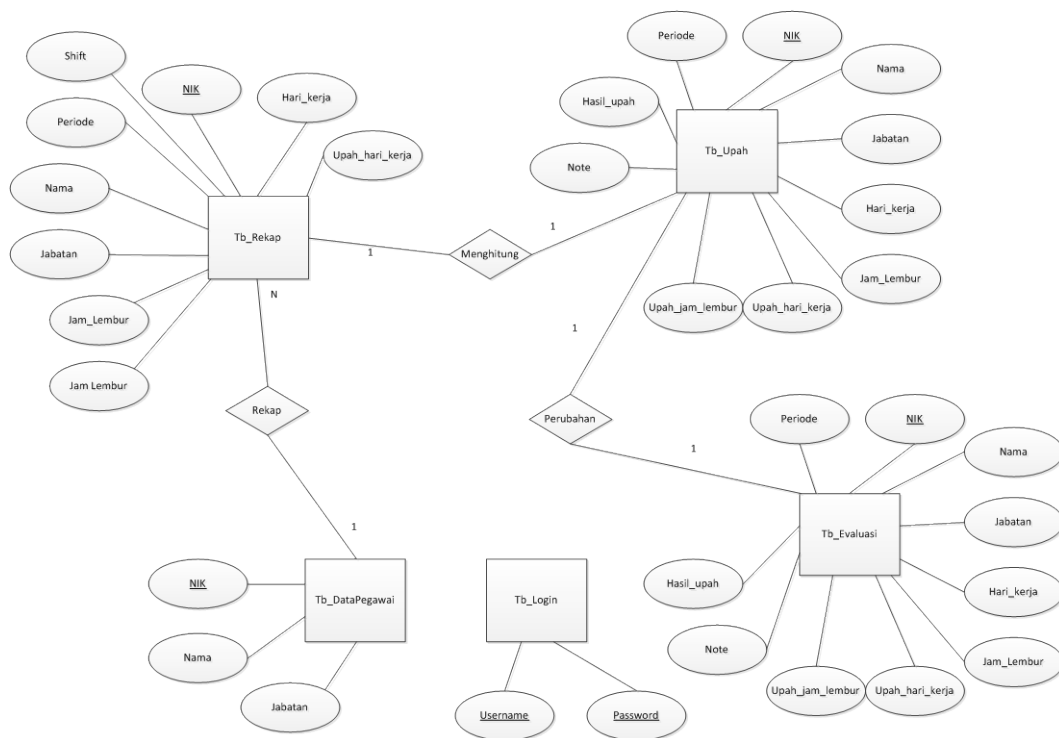
Windows Navigation Diagram pada sistem usulan digunakan untuk menunjukkan bagaimana navigasi dari halaman-halaman yang terdapat di dalam aplikasi, berikut merupakan *Windows Navigation* Diagram sistem informasi penghitungan upah pegawai harian :



Gambar V.17 *Windows Navigation Diagram* Upah Pegawai Harian
(Sumber: Hasil Analisis, 2019)

5.10 *Entity Relationship Diagram (ERD)*

Entity relationship diagram dapat membantu dalam merancang hubungan antara entitas dari sebuah *database*. Diagram ini berisi data-data yang berada di dalam *database* yang dapat membantu dalam mengetahui relasi keterhubungan antar tabel. *Entity relationship diagram* sistem informasi Penghitungan upah pegawai dapat dilihat sebagai berikut:



Gambar V.18 *Entity Relationship* Diagram Upah Pegawai Harian
(Sumber: Hasil Analisis, 2019)

5.10.1 Kamus Data

Kamus data membantu dalam pendefinisian data agar pendefinisian data tersebut dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar penulisan). Berikut ini adalah kamus data yang terdapat dalam aplikasi sistem informasi penghitungan upah pegawai:

1. Spesifikasi Tabel Data Pegawai

Nama Tabel : tb_datapegawai

Fungsi : Untuk menyimpan data pegawai

Tipe : *File data master*

Tabel V.9 Tabel Data Pegawai

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	NIK	nik	int	15	<i>Primary key</i>
2	Nama	nama	<i>Varchar</i>	50	
3	Jabatan	jabatan	<i>Varchar</i>	50	

(Sumber: Hasil Analisis Data, 2019)

2. Spesifikasi Tabel Data Daftar Hadir dan Jam Lembur

Nama Tabel : tb_datarekap

Fungsi : Untuk menyimpan data rekap daftar hadir dan jam lembur

Tipe : *File* data masukan

Tabel V.10 Tabel Data rekap

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	Periode	Bln_Thn	<i>Varchar</i>	15	
2	Waktu	waktu	<i>Varchar</i>	15	<i>Primary key</i>
3	Nik	nik	int	15	
4	Nama	nama	<i>Varchar</i>	50	
5	Jabatan	jabatan	<i>Varchar</i>	50	
6	Shift 1	S1	<i>Varchar</i>	3	
7	Shift 2	S2	<i>Varchar</i>	3	
8	Shift 3	S3	<i>Varchar</i>	3	
9	Lembur Shift 1	L_S1	<i>Varchar</i>	3	
10	Lembur Shift 2	L_S2	<i>Varchar</i>	3	
11	Lembur Shift 3	L_S3	<i>Varchar</i>	3	
12	Hari_Kerja	hk	int	15	
13	Jam_Lembur	jl	int	15	

(Sumber: Hasil Analisis Data, 2019)

3. Spesifikasi Tabel Data Upah Pegawai

Nama Tabel : tb_dataupah

Fungsi : Untuk menyimpan data upah pegawai

Tipe : *File* data masukan

Tabel V.11 Tabel Data upah

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	Bulan/Tahun	Bln_Thn	<i>Varchar</i>	15	
2	Waktu	waktu	<i>Varchar</i>	15	<i>Primary key</i>
3	Nik	nik	int	15	
4	Nama	nama	<i>Varchar</i>	50	
5	Jabatan	jabatan	<i>Varchar</i>	50	
6	Hari Kerja	hk	int	15	
7	Jam Lembur	jl	int	15	
8	Upah Hari Kerja	UHK	int	15	
9	Upah Jam Lembur	UJL	int	15	
10	Total	total	int	20	
11	Evaluasi	eval	<i>Varchar</i>	100	

(Sumber: Hasil Analisis Data, 2019)

5.11 Perancangan *Interface*

Perancangan *interface* merupakan tahapan untuk membuat tampilan atau *design* dari Sistem Informasi Penghitungan Upah Pegawai. Rancangan tampilan yang dibuat meliputi beberapa rancangan *input* dan rancangan *output* (berupa laporan). Rancangan *interface* dalam Sistem Informasi Penghitungan Upah Pegawai tersebut dapat dilihat berikut ini:

1. *Form Login*

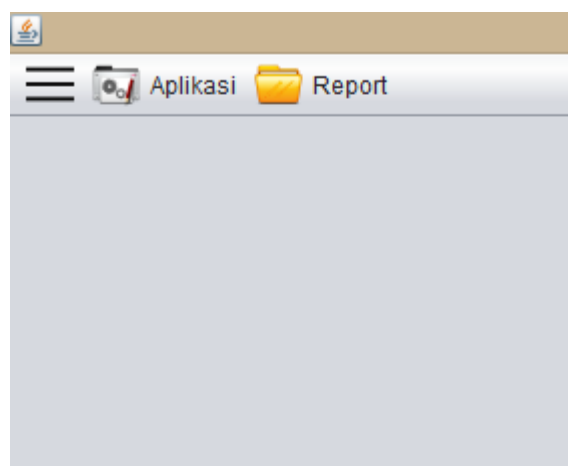
Form Login digunakan untuk membedakan hak akses *user*. Melalui *form* ini, pengguna yang dapat masuk kedalam sistem adalah pengguna yang mengetahui kode *login* dan *password* pengguna yang memiliki wewenang untuk menggunakan sistem informasi penghitungan upah pegawai.

A login form titled "MENU LOGIN" in a stylized font. The form has a light blue background and a dark blue header. It includes two input fields: "USERNAME" and "PASSWORD". Below the input fields are two buttons: "MASUK" (Login) with a key icon and "KELUAR" (Logout) with a door icon. The form is framed by a grey border.

Gambar V.19 Rancangan *Form Login*
(Sumber: Hasil Analisis Data, 2019)

2. Menu Utama

Menu utama ini dirancang untuk memudahkan *user* dalam mengakses aplikasi dengan pilihan-pilihan menu yang sudah disediakan sesuai dengan kebutuhan.



Gambar V.20 Rancangan Menu Utama
(Sumber: Hasil Analisis Data, 2019)

3. Panel Rekap

Panel rekap ini dirancang agar *user* dapat membuat data rekap daftar hadir dan jam lembur pegawai, yang mana panel rekap ini sudah terhubung kedalam dua *database* yaitu *database* rekap dan *database* data pegawai yang sudah di *import* kedalam tabel.

Gambar V.21 Rancangan Panel Rekap
(Sumber: Hasil Analisis Data, 2019)

4. Panel Upah

Panel upah ini dirancang agar *user* dapat menghitung dan mengelola upah pegawai, yang mana panel upah ini sudah terhubung kedalam dua *database* yaitu *database* rekap dan *database* upah pegawai yang sudah di *import* kedalam tabel.

Gambar V.22 Rancangan Panel Upah

(Sumber: Hasil Analisis Data, 2019)

5. Panel Evaluasi

Panel evaluasi ini dirancang agar *user* dapat melakukan evaluasi kesalahan yang ditulis di dalam kolom note. Panel evaluasi ini sudah terhubung kedalam *database* yaitu *database* upah pegawai yang sudah di *import* kedalam tabel.

Gambar V.23 Rancangan Panel Evaluasi

(Sumber: Hasil Analisis Data, 2019)

6. Panel Report upah

Report atau laporan ini dirancang untuk memudahkan *user* agar dapat melakukan cetak laporan dengan mudah tanpa harus merekap ulang dari *database*, selain itu juga dapat menyimpan data kedalam penyimpanan *internal PC* ataupun laptop. Panel ini sudah terhubung kedalam *database* yaitu *database* upah pegawai.

Gambar V.24 Rancangan Panel Report



REPORT UPAH PEGAWAI HARIAN

Laporan

Reset

Kembali

==Pilih==

JasperViewer

54,51%

LAPORAN

UPAH PEGAWAI HARIAN

Periode 9/2019

Nik	Nama	Jabatan	Hari Kerja	Jam Lembur	UHK	U.JL	Total
444	Haffyan	Pegawai Harian	24	30	3600000	360000	3960000
111	Silvester	Pegawai Harian	24	42	3600000	504000	4104000
222	Teguh	Pegawai Harian	24	24	3600000	288000	3888000

Jakarta, 13 September 2019

(Manager Accounting)

Page 1 of 1

Activate Windows
Go to PC settings

(Sumber: Hasil Analisis Data, 2019)

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil sistem informasi yang telah dirancang mengenai sistem informasi penghitungan upah pegawai harian pada PT Korea Nano Tech Indonesia, dapat disimpulkan bahwa:

1. Dengan adanya sistem informasi penghitungan upah pegawai harian, dapat membantu divisi HRD dalam proses penghitungan upah pegawai harian dengan cepat dan akurat.
2. Sistem mempunyai *database* yang sudah terhubung secara otomatis dan menyediakan pengelolaan data mulai dari data master pegawai harian hingga data rekaman upah pegawai harian.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem informasi pengitungan upah pegawai ini selanjutnya yaitu:

1. Mengimplementasikan hasil dari penelitian berupa sistem informasi untuk proses penghitungan upah pegawai harian di divisi HRD pada PT Korea Nano Tech Indonesia.
2. Untuk pengembangan sistem informasi lebih lanjut dapat dikembangkan dengan menyertakan gaji pegawai kontak dan pegawai tetap beserta tunjangannya.

DAFTAR PUSTAKA

- Adian & Wenno. 2017. Rancang Bangun Sistem Penggajian menggunakan Visual Studio dengan metode waterfall. Jakarta: Open Jurnal
- Ardianto & Fajar. 2017. Perancangan Sitem Informasi Penggajian berbasis web dengan metode waterfall. Yogyakarta: Open Jurnal.
- Bastian. 2007. Vol. 2 NO. 2 Desember 2007 : Hubungan Sumber Daya Manusia. Bandung: Remaja Rosdakarya.
- Busro, Muhammad. 2018. Teori-Teori Manajemen Sumber Daya Manusia. Jakarta: Prenadamenida Group.
- Dennis, Alan. 2010. *System Analysis and Design with UML 2.0*. New York: John Wiley and Sons, Inc.
- Dennis, Alan. 2011. *System Analysis and Design with UML 2.0*. New York: John Wiley and Sons, Inc
- Dennis, Alan. 2012 . *Systems Analysis & Design With UML Version 2.0; An. Object-Oriented Approach 4th Edition*. John Wiley & Sons
- Edi, Purwanto. 2010. Manajemen Sumber Daya Manusia. Surabaya: Karunia Penerbit
- Eniyati, Sri. 2011. Perancangan Sistem Pendukung Pengambilan Keputusan untuk Penerimaan Beasiswa dengan Metode SAW (Simple Additive Weighting: Universitas Stikubank, 16/2: 171-176. Semarang
- Febriani. Pengertian Flowchart. <https://www.it-jurnal.com/pengertianflowchart/>. (Tanggal Akses: 27 Maret 2017).
- Herdianti & Widiанти. 2014. Pembangunan Sistem Pendukung Keputusan. Bandung: Remaja Rosdakarya.
- Jogiyanto. 2005. Sistem Teknologi dan Informasi. Edisi II. Yogyakarta: Andi.
- Lucas, Hendry. 2007. Sistem Informasi Manajemen. Boston: Hill
- Muslihudin dan Oktafianto. 2016. Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML. Yogyakarta: Andi.
- Nurchmad, Edhy. 2009. Analisis Sistem Informasi. Yogyakarta: Graha Ilmu.

- Pressma. 2015. Rekayasa Perangkat Lunak Pendekatan Praktisi. Yogyakarta: Andi
- Prihantoro, Agung. 2015. Peningkatan Kinerja Sumber daya Manusia. Pati: Deepublish.
- Ramdhani, Ali. 2008. Sistem Informasi Manajemen. Yogyakarta: Andi.
- Rizky, Soetam. 2011. Konsep Dasar Rekayasa Perangkat Lunak {Software Rengineering). Jakarta: Prestasi Pustaka Publisher.
- Sanusi, Hamid. 2014. Manajemen Sumber Daya Manusia Lanjutan. Yogyakarta : Deepublish.
- Sugiyono. 2013. Observasi dan pengambilan data. Yogyakarta: Graha Ilmu
- Suhariadi, Fendy. 2013. Manajemen Sumber Daya Manusia Dalm Pendekaran Teoritis-Praktis. Surabaya: AUP.
- Sutabri, Tata. 2005. Analisis Sistem Informasi. Yogyakarta: Andi.
- Sutaji, Deni. 2012. *Sistem Inventory Mini Market dengan PHP & query*. Yogyakarta: Lokomedia.
- Whedasmara, Arie. 2009. Jurnal Sistem Informasi (JSI), Vol 1, NO. Universitas Sriwijaya.
- Wijayanto, Dian. 2012. Pengantar Manajemen. Jakarta: PT Gramedia Pustaka Utama.
- Yakub. 2012. Pengantar Sistem Informasi dan sistem pengambil keputusan. Yogyakarta: Graha Ilmu.

LAMPIRAN

LAMPIRAN A

KODE PROGRAM

```
package aplikasigaji;
import java.sql.*;
import java.util.HashMap;
import java.util.Map;
import javax.swing.JOptionPane;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;

public class menu_login extends javax.swing.JFrame {
    public menu_login() {
        initComponents();
        this.setExtendedState(menu_login.MAXIMIZED_BOTH);
        jMenuItem1.setVisible(false);
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String user = "admin";
        String passw = "admin";
        if(user.equalsIgnoreCase(nama.getText())                &&
        passw.equalsIgnoreCase(pass.getText())){
            jMenuItem1.setVisible(true);
            this.jPanel2.setVisible(false);
            nama.setText("");
            pass.setText("");}
        else {
            i) javax.swing.JOptionPane.showMessageDialog(null, "Maaf User Atau
                Password Anda Salah");
```

```

        nama.setText("");
        pass.setText("");
        nama.requestFocus();
    } }

    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        int pesan=JOptionPane.showConfirmDialog(null, "Yakin ?", "Konfirmasi",
JOptionPane.YES_NO_OPTION , JOptionPane.QUESTION_MESSAGE);
        if(pesan==JOptionPane.YES_OPTION){
            jMenuItemBar1.setVisible(false);
            this.jPanel2.setVisible(true);
        }
    }

    private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
        int pesan=JOptionPane.showConfirmDialog(null, "Yakin InginKeluar ?", "",
JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE);
        if (pesan==JOptionPane.YES_OPTION);{
            System.exit(0);
        }
    }

    private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
        new rekap().show(); }

    private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
        new upah().show();
    }

    private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            koneksi_report objkoneksi = new koneksi_report ();
            Connection con =objkoneksi.koneksi_report();
            String filename="./src/Report/report1.jrxml";
            String filetoFill="./src/Report/report1.jasper";
            JasperCompileManager.compileReport(filename);
            Map param= new HashMap();

```

```

        JasperFillManager.fillReport(filetoFill, param, con);
        JasperPrint jp=JasperFillManager.fillReport(filetoFill, param,con);
        JasperViewer.viewReport(jp, false);
    }catch(Exception ex){
        System.out.print(ex.toString());
    }}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    new evaluasi().show();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new menu_login().setVisible(true);
        }
    });
}

package aplikasigaji;

import com.mysql.jdbc.Connection;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class rekap extends javax.swing.JFrame {
    public rekap() {
        initComponents();
        load_table();
        kosong();
    }
}

```

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // fungsi hapus data
    try {
        String sql ="delete from rekap where nik='"+txt_nik.getText()+"";
        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.PreparedStatement pst=conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(this, "berhasil di hapus");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    load_table();
    kosong();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String      sql      =      "INSERT      INTO      rekap      VALUES
        ("'+txt_bt.getText()+ "','"+txt_nik.getText()+ "','"+txt_nama.getText()+ "','"+jComb
oBox1.getSelectedItem()+ "','"+txt_hk.getText()+ "','"+txt_jl.getText()+ "')";
        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.PreparedStatement pst=conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "Penyimpanan Data Berhasil");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
    load_table();
    kosong();
}

```

```

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // menampilkan data kedalam form pengisian:
    int baris = jTable1.rowAtPoint(evt.getPoint());
    String bln_thn = jTable1.getValueAt(baris, 1).toString();
    txt_bt.setText(bln_thn);
    String nik = jTable1.getValueAt(baris, 2).toString();
    txt_nik.setText(nik);
    String nama = jTable1.getValueAt(baris, 3).toString();
    txt_nama.setText(nama);
    String jr = jTable1.getValueAt(baris, 4).toString();
    jComboBox1.setSelectedItem(jr);
    String hari_kerja = jTable1.getValueAt(baris, 5).toString();
    txt_hk.setText(hari_kerja);
    String jam_kerja = jTable1.getValueAt(baris, 6).toString();
    txt_jl.setText(jam_kerja);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // Edit Data
    try {
        String sql = "UPDATE rekap SET bln_thn = '"+txt_bt.getText()+"', nik = 
        '"+txt_nik.getText()+"', nama = '"+txt_nama.getText()+"', jabatan = 
        '"+jComboBox1.getSelectedItem()+"', hari_kerja= '"+txt_hk.getText()+"', 
        jam_lembur= '"+txt_jl.getText()+"' WHERE nik = '"+txt_nik.getText()+"'";
        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.PreparedStatement pst=conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "data berhasil di edit");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Perubahan Data 
        Gagal"+e.getMessage());
    }
}

```

```

        load_table();
        kosong(); }
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    kosong();
}
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}
private void panggilActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("No");
    model.addColumn("Nik");
    model.addColumn("Nama");
    model.addColumn("Jabatan");
    //menampilkan data database kedalam tabel
    try {
        int no=1;
        String sql = "select * from datapegawai";
        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.Statement stm=conn.createStatement();
        java.sql.ResultSet res=stm.executeQuery(sql);
        while(res.next()){
            model.addRow(new
Object[]{no++,res.getString(1),res.getString(2),res.getString(3)});
        }
        jTable2.setModel(model);
    } catch (Exception e) {
    } }
private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
    int baris = jTable2.rowAtPoint(evt.getPoint());
    String nik =jTable2.getValueAt(baris, 1).toString();

```

```

        txt_nik.setText(nik);
        String nama = jTable2.getValueAt(baris,2).toString();
        txt_nama.setText(nama);
        String jr = jTable2.getValueAt(baris, 3).toString();
        jComboBox1.setSelectedItem(jr);
    }
    private void load_table() {
        // membuat tampilan model tabel
        DefaultTableModel model = new DefaultTableModel();
        model.addColumn("No");
        model.addColumn("Bln_Thn");
        model.addColumn("Nik");
        model.addColumn("Nama");
        model.addColumn("Jabatan");
        model.addColumn("Hari_Kerja");
        model.addColumn("Jam_Kerja");
        //menampilkan data database kedalam tabel
        try {
            int no=1;
            String sql = "select * from rekap";
            java.sql.Connection conn=(Connection)koneksi.koneksi();
            java.sql.Statement stm=conn.createStatement();
            java.sql.ResultSet res=stm.executeQuery(sql);
            while(res.next()){
                model.addRow(new
Object[] {no++,res.getString(1),res.getString(2),res.getString(3),res.getString(4),re
s.getString(5),res.getString(6)});
            }
            jTable1.setModel(model);
        } catch (Exception e) {
        } }

```

```

private void kosong() {
    txt_nik.setText(null);
    txt_nama.setText(null);
    txt_hk.setText(null);
    txt_jl.setText(null);
    jComboBox1.getSelectedItem();
}
}
package aplikasigaji;

import com.mysql.jdbc.Connection;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class rekap extends javax.swing.JFrame {
    public rekap() {
        initComponents();
        load_table();
        kosong();
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // fungsi hapus data
        try {
            String sql ="delete from rekap where nik='"+txt_nik.getText()+"'";
            java.sql.Connection conn=(Connection)koneksi.koneksi();
            java.sql.PreparedStatement pst=conn.prepareStatement(sql);
            pst.execute();
            JOptionPane.showMessageDialog(this, "berhasil di hapus");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
        load_table();
        kosong(); }

```



```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String sql = "INSERT INTO rekap VALUES
        (" + txt_bt.getText() + "," + txt_nik.getText() + "," + txt_nama.getText() + "," + jComb
        oBox1.getSelectedItem() + "," + txt_hk.getText() + "," + txt_jl.getText() + ")";
        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.PreparedStatement pst=conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "Penyimpanan Data Berhasil");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage()); }
    load_table();
    kosong(); }

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // menampilkan data kedalam form pengisian:
    int baris = jTable1.rowAtPoint(evt.getPoint());
    String bln_thn =jTable1.getValueAt(baris, 1).toString();
    txt_bt.setText(bln_thn);
    String nik =jTable1.getValueAt(baris, 2).toString();
    txt_nik.setText(nik);
    String nama = jTable1.getValueAt(baris,3).toString();
    txt_nama.setText(nama);
    String jr = jTable1.getValueAt(baris, 4).toString();
    jComboBox1.setSelectedItem(jr);
    String hari_kerja=jTable1.getValueAt(baris, 5).toString();
    txt_hk.setText(hari_kerja);
    String jam_kerja = jTable1.getValueAt(baris, 6).toString();
    txt_jl.setText(jam_kerja);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // Edit Data

```

```

try {
    String sql ="UPDATE rekap SET bln_thn = '"+txt_bt.getText()+"", nik =
    '"+txt_nik.getText()+"",  nama  =  '"+txt_nama.getText()+"",  jabatan  =
    '"+jComboBox1.getSelectedItem()+"",  hari_kerja=  '"+txt_hk.getText()+"",
    jam_lembur= '"+txt_jl.getText()+" WHERE nik = '"+txt_nik.getText()+"''";

    java.sql.Connection conn=(Connection)koneksi.koneksi();
    java.sql.PreparedStatement pst=conn.prepareStatement(sql);
    pst.execute();
    JOptionPane.showMessageDialog(null, "data berhasil di edit");
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Perubahan Data
    Gagal"+e.getMessage()); }

load_table();
kosong();}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    kosong();}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();}

private void panggilActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("No");
    model.addColumn("Nik");
    model.addColumn("Nama");
    model.addColumn("Jabatan");
    //menampilkan data database kedalam tabel
    try {
        int no=1;
        String sql = "select * from datapegawai";
        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.Statement stm=conn.createStatement();
        java.sql.ResultSet res=stm.executeQuery(sql);

```

```

        while(res.next()){
            model.addRow(new
Object[] {no++,res.getString(1),res.getString(2),res.getString(3)});
        }
        jTable2.setModel(model);
    } catch (Exception e) {
    } }

private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
    int baris = jTable2.rowAtPoint(evt.getPoint());
    String nik = jTable2.getValueAt(baris, 1).toString();
    txt_nik.setText(nik);
    String nama = jTable2.getValueAt(baris,2).toString();
    txt_nama.setText(nama);
    String jr = jTable2.getValueAt(baris, 3).toString();
    jComboBox1.setSelectedItem(jr);
}

private void load_table() {
    // membuat tampilan model tabel
    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("No");
    model.addColumn("Bln_Thn");
    model.addColumn("Nik");
    model.addColumn("Nama");
    model.addColumn("Jabatan");
    model.addColumn("Hari_Kerja");
    model.addColumn("Jam_Kerja");

    //menampilkan data database kedalam tabel
    try {
        int no=1;
        String sql = "select * from rekap";

```

```

        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.Statement stm=conn.createStatement();
        java.sql.ResultSet res=stm.executeQuery(sql);
        while(res.next()){
            model.addRow(new
Object[] {no++,res.getString(1),res.getString(2),res.getString(3),res.getString(4),re
s.getString(5),res.getString(6)});
        }
        jTable1.setModel(model);
    } catch (Exception e) {

    }
}

private void kosong() {

    txt_nik.setText(null);
    txt_nama.setText(null);
    txt_hk.setText(null);
    txt_jl.setText(null);
    jComboBox1.getSelectedItem();
}
}

package aplikasigaji;

import com.mysql.jdbc.Connection;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class evaluasi extends javax.swing.JFrame {
    public evaluasi() {

```

```

        initComponents();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel model = new DefaultTableModel();
        model.addColumn("No");
        model.addColumn("Bln_Thn");
        model.addColumn("Nik");
        model.addColumn("Nama");
        model.addColumn("Jabatan");
        model.addColumn("Hari Kerja");
        model.addColumn("Jam Lembur");
        model.addColumn("UHK");
        model.addColumn("UJL");
        model.addColumn("TOTAL");
        model.addColumn("Evaluasi");

        //menampilkan data database kedalam tabel
        try {
            int no=1;
            String sql = "select * from upahgaji";
            java.sql.Connection conn=(Connection)koneksi.koneksi();
            java.sql.Statement stm=conn.createStatement();
            java.sql.ResultSet res=stm.executeQuery(sql);
            while(res.next()){
                model.addRow(new
Object[] {no++,res.getString(1),res.getString(2),res.getString(3),res.getString(4),re
s.getString(5),
res.getString(6),res.getString(7),res.getString(8),res.getString(9),res.getString(10)
}); }
            jTable1.setModel(model);

```

```

    } catch (Exception e) {    }}
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // menampilkan data kedalam form pengisian:
    int baris = jTable1.rowAtPoint(evt.getPoint());
    String bln_thn =jTable1.getValueAt(baris, 1).toString();
    txbt.setText(bln_thn);
    String nik =jTable1.getValueAt(baris, 2).toString();
    txnik.setText(nik);
    String nama = jTable1.getValueAt(baris,3).toString();
    txnama.setText(nama);
    String jr = jTable1.getValueAt(baris, 4).toString();
    jComboBox1.setSelectedItem(jr);
    String hari_kerja = jTable1.getValueAt(baris,5).toString();
    txhk.setText(hari_kerja);
    String jam_lembur = jTable1.getValueAt(baris,6).toString();
    txjl.setText(jam_lembur);
    String upah_hari_kerja = jTable1.getValueAt(baris,7).toString();
    txhhk.setText(upah_hari_kerja);
    String upah_jam_lembur = jTable1.getValueAt(baris,8).toString();
    txhjl.setText(upah_jam_lembur);
    String total = jTable1.getValueAt(baris,9).toString();
    txtot.setText(total);
    String evaluasi = jTable1.getValueAt(baris,10).toString();
    txeval.setText(evaluasi);}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // Edit Data
    try {String sql ="UPDATE upahgaji SET bln_thn = '"+txbt.getText()+"", nik
= "'"+txnik.getText()+"",  nama   =   '"+txnama.getText()+"",  jabatan   =
 '"+jComboBox1.getSelectedItem()+"",  UHK=   '"+txhhk.getText()+"",  UJL=
 '"+txhjl.getText()+"", Total= '"+txtot.getText()+"", evaluasi= '"+txeval.getText()+"
WHERE nik = '"+txnik.getText()+"'";

```

```

        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.PreparedStatement pst=conn.prepareStatement(sql);
        pst.execute();
        JOptionPane.showMessageDialog(null, "data berhasil di edit");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Perubahan Data
Gagal"+e.getMessage());}
        load_table();
        kosong();}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    kosong();}

private void load_table() {
    // membuat tampilan model tabel
    DefaultTableModel model = new DefaultTableModel();
    model.addColumn("No");
    model.addColumn("Bln_Thn");
    model.addColumn("Nik");
    model.addColumn("Nama");
    model.addColumn("Jabatan");
    model.addColumn("Hari Kerja");
    model.addColumn("Jam Lembur");
    model.addColumn("UHK");
    model.addColumn("UJL");
    model.addColumn("TOTAL");
    model.addColumn("Evaluasi");
    //menampilkan data database kedalam tabel
    try {
        int no=1;

```

```

        String sql = "select * from upahgaji";
        java.sql.Connection conn=(Connection)koneksi.koneksi();
        java.sql.Statement stm=conn.createStatement();
        java.sql.ResultSet res=stm.executeQuery(sql);
        while(res.next()){
            model.addRow(new
Object[] {no++,res.getString(1),res.getString(2),res.getString(3),res.getString(4),
res.getString(5),res.getString(6),res.getString(7),res.getString(8),res.getString(9),r
es.getString(10)});}
        jTable1.setModel(model);
    } catch (Exception e) {    }}
    private void kosong() {
        txbt.setText(null);
        txnik.setText(null);
        txnama.setText(null);
        txhk.setText(null);
        txjl.setText(null);
        jComboBox1.getSelectedItem();
        txhhk.setText(null);
        txhjl.setText(null);
        txtot.setText(null);
        txeval.setText(null);}}
package aplikasigaji;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class koneksi {
    private static Connection mysqlconfig;
    public static Connection koneksi()throws SQLException{
        try {

```



```

String url="jdbc:mysql://localhost:3306/gaji"; //url database
String user="root"; //user database
String pass=""; //password database
DriverManager.registerDriver(new com.mysql.jdbc.Driver());
mysqlconfig=DriverManager.getConnection(url, user, pass);
} catch (Exception e) {
    System.err.println("koneksi gagal "+e.getMessage()); //perintah
menampilkan error pada koneksi}
    return mysqlconfig;
}
}

```

LAMPIRAN B
JADWAL WAWANCARA

NO	NAMA (Pembimbing KL/Jabatan)	TANGGAL	TEMPAT	BAGIAN	KETERANGAN
1	Elida, Kepala Divisi HRD	10/09/2018	Lobby PT. KNT Indonesia	Divisi HRD	Wawancara HRD secara umum
2	Mika, Staff Divisi HRD	23/10/2018	Lobby PT. KNT Indonesia	Divisi HRD	Wawancara Daftar hadir dan upah gaji
3	Elida, Kepala Divisi HRD	16/02/2019	Lobby PT. KNT Indonesia	Divisi HRD	Wawancara Permasalahan

LAMPIRAN C
JADWAL OBSERVASI

NO	NAMA (Pembimbing KL/Jabatan)	TANGGAL	TEMPAT	BAGIAN	KETERANGAN OBSERVASI
1	Elida, Kepala Divisi HRD	10/10/2018	Ruangan divisi HRD	Divisi HRD	Proses sortir <i>Time Card</i>
2	Mika, Staff Divisi HRD	23/10/2018	Ruangan divisi HRD	Divisi HRD	Proses rekap daftar hadir pegawai
3	Elida, Kepala Divisi HRD	4/11/2018	Ruangan divisi HRD	Divisi HRD	Proses rekap daftar cuti pegawai
4	Elida, Kepala Divisi HRD	23/11/2018	Ruangan divisi HRD	Divisi HRD	Proses penghitungan gaji pegawai

LAMPIRAN C

BLACK BOX TESTING

1. Login

Deskripsi : Menguji fungsi *login* pada tabel *login*

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol <i>Login</i> tanpa memasukkan <i>username</i> dan <i>password</i>	Sistem akan menampilkan maaf user atau password anda salah	Sistem akan menampilkan maaf user atau password anda salah	Valid
2	Memasukkan <i>username</i> dengan benar sedangkan <i>password</i> salah atau sebaliknya, lalu mengklik tombol <i>Login</i>	Sistem akan menampilkan maaf user atau password anda salah	Sistem akan menampilkan maaf user atau password anda salah	Valid
3	Memasukkan <i>username</i> dan <i>password</i> dengan benar lalu mengklik tombol <i>Login</i>	Sistem akan menerima akses <i>login</i> dan menampilkan halaman utama	Sistem menerima akses <i>login</i> dan menampilkan halaman utama	Valid

2. Panel rekap

Deskripsi : Menguji fungsi tambah data pada panel rekap

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol Tambah Data	Sistem akan menampilkan data didalam tabel	Sistem akan menampilkan data didalam tabel	Valid

Deskripsi : Menguji fungsi ubah data pada panel rekap

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol Ubah	Sistem akan menampilkan perubahan pada data	Sistem akan menampilkan perubahan pada data	Valid

Deskripsi : Menguji fungsi hapus pada panel rekap

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol Hapus pada data yang dipilih	Sistem akan menghapus data yang dipilih	Sistem akan menghapus data yang dipilih	Valid

Deskripsi : Menguji fungsi panggil pada panel rekap

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol panggil pada data yang dipilih	Sistem akan menampilkan tabel data pegawai	Sistem akan menampilkan tabel data pegawai	Valid

Deskripsi : Menguji fungsi kembali pada panel rekap

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol kembali pada data yang dipilih	Sistem akan menampilkan menu utama	Sistem akan menampilkan menu utama	Valid

Deskripsi : Menguji fungsi reset pada panel rekap

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol reset pada data yang dipilih	Sistem akan menghilangkan text pada kolom	Sistem akan menghilangkan text pada kolom	Valid

3. Panel upah

Deskripsi : Menguji fungsi tambah data pada panel upah

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol Tambah Data	Sistem akan menampilkan data didalam tabel	Sistem akan menampilkan data didalam tabel	Valid

Deskripsi : Menguji fungsi ubah data pada panel upah

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol Ubah	Sistem akan menampilkan perubahan pada data	Sistem akan menampilkan perubahan pada data	Valid

Deskripsi : Menguji fungsi hapus pada panel upah

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol Hapus pada data yang dipilih	Sistem akan menghapus data yang dipilih	Sistem menghapus data yang dipilih	Valid

Deskripsi : Menguji fungsi panggil pada panel upah

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol panggil pada data yang dipilih	Sistem akan menampilkan tabel rekap	Sistem akan menampilkan tabel rekap	Valid

Deskripsi : Menguji fungsi kembali pada panel upah

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol kembali pada data yang dipilih	Sistem akan menampilkan menu utama	Sistem akan menampilkan menu utama	Valid

Deskripsi : Menguji fungsi reset pada panel upah

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol reset pada data yang dipilih	Sistem akan menghilangkan text pada kolom	Sistem akan menghilangkan text pada kolom	Valid

4. Panel Evaluasi

Deskripsi : Menguji fungsi panggil data pada panel Evaluasi

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol panggil pada data yang dipilih	Sistem akan menampilkan tabel upah	Sistem akan menampilkan tabel upah	Valid

Deskripsi : Menguji fungsi note data pada panel Evaluasi

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol note pada data yang dipilih	Sistem akan menampilkan catatan atau note pada tabel upah	Sistem akan menampilkan catatan atau note pada tabel upah	Valid

Deskripsi : Menguji fungsi kembali pada panel Evaluasi

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol kembali pada data yang dipilih	Sistem akan menampilkan menu utama	Sistem akan menampilkan menu utama	Valid

Deskripsi : Menguji fungsi reset pada panel Evaluasi

Penguji : Herly Frisdiansyah (1315032)

No.	Test Case	Expected Result	Actual Record	Result
1	Mengklik tombol reset pada data yang dipilih	Sistem akan menghilangkan text pada kolom	Sistem akan menghilangkan text pada kolom	Valid

5. Panel Report

Deskripsi : Menguji fungsi print data pada panel Report

Penguji : Herly Frisdiansyah (1315032)

No.	<i>Test Case</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
1	Mengklik tombol print pada data yang dipilih	Sistem akan menampilkan menu print	Sistem akan menampilkan menu print	Valid

Deskripsi : Menguji fungsi save data pada panel Report

Penguji : Herly Frisdiansyah (1315032)

No.	<i>Test Case</i>	<i>Expected Result</i>	<i>Actual Record</i>	<i>Result</i>
1	Mengklik tombol save pada data yang dipilih	Sistem akan menampilkan menu save pada laptop atau PC yang digunakan	Sistem akan menampilkan menu save pada laptop atau PC yang digunakan	Valid