

**RANCANG BANGUN SISTEM INFORMASI ABSENSI
MENGUNAKAN PHP 5.6.3 DAN MYSQL 5.6.21 PADA
PT REKADAYA MULTI ADIPRIMA**

TUGAS AKHIR

Diajukan Untuk Memenuhi Sebagian Syarat Penyelesaian Jenjang Diploma
Empat (D-4) Pada Program Studi Sistem Informasi Industri Otomotif Politeknik
STMI Jakarta

**OLEH
FIEKI AMALIA
1313027**



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2017**

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN R.I.

TANDA PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : **Rancang Bangun Sistem Informasi Absensi
Menggunakan PHP 5.6.3 dan MySQL 5.6.21 Pada
PT Rekadaya Multi Adiprima**

Disusun Oleh :

Nama : Fieki Amalia
Nim : 1313027
Program Studi : Sistem Informasi Industri
Tanggal Seminar : Senin, 20 November 2017
Tanggal Sidang : Senin, 27 November 2017
Tanggal Lulus : Senin, 27 November 2017

Menyetujui

Jakarta, 27 November 2017

Dosen Pembimbing

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

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN R.I.**

TANDA PERSETUJUAN ASISTEN DOSEN PEMBIMBING

Judul Tugas Akhir : **Rancang Bangun Sistem Informasi Absensi
Menggunakan PHP 5.6.3 dan MySQL 5.6.21 Pada
PT Rekadaya Multi Adiprima**

Disusun Oleh :

Nama : Fieki Amalia
Nim : 1313027
Program Studi : Sistem Informasi Industri
Tanggal Seminar : Senin, 20 November 2017
Tanggal Sidang : Senin, 27 November 2017
Tanggal Lulus : Senin, 27 November 2017

Menyetujui

Jakarta, 27 November 2017

Asisten Dosen Pembimbing

Noveriza Yuliasari, S.Si, MT
NIP. 197811212009012003

POLETEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI

LEMBAR PENGESAHAN

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
ABSENSI MENGGUNAKAN PHP 5.6.3 DAN MYSQL
5.6.21 PADA PT REKADAYA MULTI ADIPRIMA.

Disusun Oleh :
Nama : Fieki Amalia
NIM : 1313027
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 R.I pada hari Senin tanggal 27 November 2017.

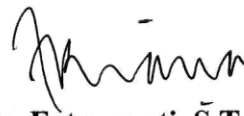
Jakarta, 27 November 2017

Dosen Pembimbing



Fifi L Hadianastuti, S.Kom, M.Kes
NIP. 19731016.200502.2.001

Ketua Penguji



Triana Fatmawati, S.T, MT
NIP. 19800514.200502.2.001

Dosen Penguji



Dedy Trisanto, S.Kom, MMSI
NIP: 19780505.200502.1.002

Dosen Penguji



Ahmad Juniar, S.kom,M.T
NIP. 19790605.200604.1.002

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas limpahan nikmat, rahmat serta karunia yang senantiasa diberikan sehingga penulis mampu menyelesaikan Tugas Akhir dengan judul **“Rancang Bangun Sistem Informasi Absensi Karyawan Menggunakan PHP 5.6.3 dan MySQL 5.6.21 Pada PT Rekadaya Multi Adiprima”**.

Penulisan Tugas Akhir ini bertujuan untuk memenuhi sebagian syarat penyelesaian jenjang Diploma Empat (D4) program studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Terselesainya tugas akhir ini tidak terlepas dari bantuan moril maupun materil, bimbingan, dan petunjuk dari berbagai pihak. Oleh karena itu, tak salah kiranya bila penulis mengungkapkan rasa terima kasih yang sebesar-besarnya dan penghargaan kepada:

1. Kedua Orang tua, adik serta keluarga tercinta yang telah memberikan doa, dukungan, pengorbanan, semangat dan kasih sayang hingga saat ini.
2. Dr. Mustofa, ST, MT, selaku Direktur Politeknik STMI Jakarta.
3. Bapak Drs. Jacob Saragih, MM, selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
4. Ibu Fifi L. Hadianastuti, S.Kom., M.Kes. Selaku dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
5. Ibu Noveriza Yuliasari, MT Selaku Asisten Dosen Pembimbing yang telah membantu memberikan bimbingan serta pengarahan kepada penulis dalam penyelesaian Tugas Akhir.
6. Ibu Rosalina Faried selaku Presiden Direktur PT Rekadaya Multi Adiprima.
7. Bapak Herry, Ibu Merry, Ibu Dita, Bapak Bowo, Bapak Maralih, Bapak Eriza, Bapak Fahmi dan seluruh pegawai di PT Rekadaya Multi Adiprima yang telah membantu memberi arahan serta informasi untuk pengumpulan data.

8. Seluruh jajaran dosen dan *staff* karyawan Politeknik STMI Jakarta yang telah membantu dalam kelancaran pembuatan penulisan ini.
9. Kepada Nur Wahyudi dan Riza Agustin yang sudah menemani, memotivasi, tukar pendapat dan membantu dalam penyelesaian Tugas Akhir ini.
10. Seluruh teman-teman jurusan Sistem Informasi Industri Otomotif angkatan 2013, atas kebersamaan dan motivasinya selama ini.
11. Kepada Dwi, Malinda dan Trisna yang selalu memberi dorongan motivasi, menghibur dan selalu menemani dalam penyelesaian Tugas Akhir ini.
12. Kepada Endah, Hesma, Aliyah dan Helfina atas kebersamaan, dukungan dan motivasinya selama masa perkuliahan di Politeknik STMI Jakarta.
13. Kepada Devi dan Okrivia atas kebersamaan dan motivasinya selama ini.
14. Kepada Aji Setyo yang selalu memberikan do'a, motivasi dan dukungan selama proses penyelesaian Tugas Akhir.
15. Serta semua pihak yang telah membantu penulis dan yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa penulisan dalam penyusunan Tugas Akhir ini masih jauh dari sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang.

Jakarta, 01 November 2017

Penulis,

ABSTRAK

PT Rekadaya Multi Adiprima (PT. RMA) salah satu perusahaan yang bergerak dalam industri manufaktur. Didirikan sejak tahun 1994 telah memproduksi komponen otomotif baik untuk kendaraan roda empat maupun roda dua yang merupakan produk orisinil atau *original equipment manufactured* (OEM). Kini PT. RMA telah memiliki lokasi usaha yang tersebar di Cikeas Nagrak-Kabupaten Bogor sebanyak 9 plant yaitu 1 plant di Sumur kondang – Klari, Kabupaten Karawang, dan 1 plant di Cikarang – Kabupaten Bekasi. Permasalahan pada penelitian ini yaitu dalam penggunaan jam kehadiran karyawan yang tidak dapat dikontrol, karena pencatatan jam kehadiran masih manual yang mengakibatkan terjadinya manipulasi data pada karyawan dan menyebabkan pengolahan data absensi terorganisir, seperti pada saat melakukan rekap data dihitung satu per satu. Hal ini menjadi kendala bagi HRD dalam mengolah data absensi yang akan dijadikan laporan untuk Pimpinan. Waktu untuk mendapatkan laporan absensi karyawan membutuhkan satu sampai dua hari. Oleh karena itu, perlu dibangun sistem informasi absensi merancang *database* agar pelaksanaan absensi karyawan dalam proses lebih baik. Metodologi pengembangan sistem informasi yang digunakan pada penelitian ini adalah *evolutionary prototype*. Kemudian dalam merancang sistem informasi absensi karyawan pemodelan sistem menggunakan *Unified Modeling Language* (UML). Sistem informasi absensi karyawan dibangun menggunakan *PHP 5.6.3* sebagai bahasa pemrograman dan *MYSQL 5.6.21* sebagai basis datanya. Dengan adanya sistem informasi absensi karyawan ini nantinya dapat memudahkan HRD dalam sistem kehadiran dan laporan karyawan terhadap Pimpinan menjadi lebih baik.

Kata Kunci : Sistem Informasi, Absensi Karyawan, *Evolutionary Prototype*, *Unified Modeling Language*, PHP, *MySQL*.

DAFTAR ISI

Halaman

LEMBAR PERSETUJUAN DOSEN PEMBIMBING	
LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING	
LEMBAR BIMBINGAN TUGAS AKHIR	
LEMBAR BIMBINGAN TUGAS AKHIR	
LEMBAR PERNYATAAN KEASLIAN	
ABSTRAK	
KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vii
DAFTAR TABEL	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pokok Permasalahan	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Manfaat Tugas Akhir	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Pengertian Sistem	6
2.1.1 Karakteristik Sistem	6
2.1.2 Klasifikasi Sistem	9
2.2 Pengertian Informasi.....	10
2.2.1 Pengelompokan Informasi	10
2.2.2 Fungsi Informasi	11
2.2.2 Siklus Informasi	11
2.3 Pengertian Sistem Informasi	11
2.3.1 Komponen Sistem Informasi	12
2.4 Konsep Absensi	13
2.4.1 Pengertian Absensi	14

2.4.2 Jenis-Jenis Absensi	15
2.5 Pengertian Lembur	16
2.6 Pengertian Cuti	16
2.7 Pengembangan Sistem	17
2.7.1 Model Prototipe Secara Umum	19
2.7.2 <i>Evolutionary Prototype</i>	19
2.8 Bagan Alir (<i>Flowchart</i>)	21
2.9.1 Jenis-Jenis Flowchart	21
2.9 <i>Unified Modeling Language (UML)</i>	27
2.9.1 <i>Diagram UML</i>	27
2. 9.2 <i>Behavior Diagram</i>	28
2. 9.3 <i>Structure Diagram</i>	35
2.10 Kamus Data	39
2.11 Basis Data	40
2.12 <i>Hierarchy plus Input-Proses-Output (HIPO)</i>	41
2.13 OOP	43
2.14 PHP (Hypertext Preprocessor)	45
2.15 MySQL (My Structured Query Language)	46
2.16 XAMPP	47
BAB III METODOLOGI PENELITIAN	48
3.1 Metodologi Penelitian	48
3.2 Jenis dan Sumber Data	48
3.3 Metode Pengumpulan Data	49
3.4 Metode Pengembangan Sistem	49
3.5 Kerangka Penelitian	50
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA	55
4.1 Latar Belakang Perusahaan	55
4.2 Makna Logo Perusahaan	56
4.3 Profil Perusahaan	56
4.4 Visi, Misi, dan Pilar Lingkungan Perusahaan	57
4.5 Konsep Perusahaan	58

6.2 Saran	131
DAFTAR PUSTAKA	132
LAMPIRAN	

DAFTAR TABEL

	Halaman
Tabel II.1 Bagan Alir Sistem.....	22
Tabel II.2 Bagan Alir Program.....	24
Tabel II.3 Bagan Alir Proses	26
Tabel II.4 Elemen-Elemen Use Case Diagram.....	28
Tabel II.5 Elemen-Elemen Activity Diagram.....	29
Tabel II.6 Elemen-Elemen Sequence Diagram	31
Tabel II.7 Elemen-Elemen <i>communication diagram</i>	33
Tabel II.8 Elemen-Elemen <i>Behavior State Machines</i>	35
Tabel II.9 Elemen-Elemen Class Diagram.....	36
Tabel II.10 Elemen-Elemen <i>Package Diagram</i>	37
Tabel II.11 Elemen-Elemen <i>Deployment Diagram</i>	38
Tabel II.12 Contoh Kamus Data Untuk Tabel Pemasok.....	40
Tabel II.13 Tipe Data Pada MySQL.....	46
Tabel IV.1 Jumlah Karyawan.....	65
Tabel IV.2 Waktu Kerja <i>Shift</i>	65
Tabel IV.3 Form Lembur karyawan	74
Tabel IV.4 Hasil rekap absensi karyawan.....	75
Tabel IV.5 Rekap Lembur Karyawan.....	76
Tabel IV.6 Rekap Cuti Karyawan.....	77
Tabel IV.7 Laporan absensi tahunan PT. RMA	78
Tabel IV.8 Data karyawan PT. RMA	79
Tabel IV.9 Definisi Aktor <i>Use Case Diagram</i> Sistem Informasi Absensi Karyawan.....	83
Tabel IV.10 Definisi <i>Use Case</i> Sistem Informasi Absensi Karyawan	83
Tabel V.1 Identifikasi Kebutuhan Sistem	84
Tabel V.2 Definisi Aktor <i>Use Case Diagram</i> Sistem Usulan	88

Tabel V.3	<i>Use Case Description Login</i>	88
Tabel V.4	<i>Use Case Description Melakukan Absensi</i>	88
Tabel V.5	<i>Use Case Description Mengelola Data Karyawan</i>	89
Tabel V.6	<i>Use Case Description Mengelola Data Jabatan</i>	89
Tabel V.7	<i>Use Case Description Mengelola Data Shift</i>	89
Tabel V.8	<i>Use Case Description Menginput Lembur</i>	90
Tabel V.9	<i>Use Case Description Menginput Cuti</i>	90
Tabel V.10	<i>Use Case Description Merekap Absensi</i>	90
Tabel V.11	<i>Use Case Description Merekap Cuti</i>	91
Tabel V.12	<i>Use Case Description Membuat Laporan Absensi</i>	91
Tabel V.13	<i>Use Case Description Menyetujui Rekap Absensi</i>	91
Tabel V.14	<i>Use Case Description View Rekap Absensi Satu Bulan</i>	92
Tabel V.15	Spesifikasi Tabel Absen.....	115
Tabel V.16	Spesifikasi Tabel karyawan.....	115
Tabel V.17	Spesifikasi Tabel Jabatan	115
Tabel V.18	Spesifikasi Tabel Lembur	116
Tabel V.19	Spesifikasi Tabel Shift.....	116
Tabel V.20	Spesifikasi Tabel <i>Cuti</i>	116

DAFTAR GAMBAR

	Halaman
Gambar II.1 Karakteristik Sistem	8
Gambar II.2 Siklus Informasi	11
Gambar II.3 Komponen Sistem Informasi	13
Gambar II.4 <i>Evolutionary Prototype</i> Model	20
Gambar II.5 Contoh Gambar Bagan Alir Dokumen.....	25
Gambar II.6 <i>Visual Table Of Contents</i>	42
Gambar III.1 Kerangka Penelitian	53
Gambar IV.1 Logo PT Rekadaya Multi Adiprima	56
Gambar IV.2 Struktur Organisasi Perusahaan PT Rekadaya Multi Adiprima	
58	
Gambar IV.3 Struktur Organisasi Departmen HRD	59
Gambar IV.4 <i>Board Assy Deck A</i>	68
Gambar IV.5 <i>Board Assy Deck B</i>	69
Gambar IV.6 <i>Front Door RH (L300)</i>	69
Gambar IV.7 <i>Front Door LH (L300)</i>	70
Gambar IV.8 <i>PAD, S/A FA Door Trim</i>	70
Gambar IV.9 <i>Shield Fender Splash RH/LH</i>	70
Gambar IV.10 Kartu Absensi Harian Karyawan.....	71
Gambar IV.11 Surat Berita Acara untuk Cuti.....	77
Gambar IV.12 Prosedur Absensi dan Penggajian Karyawan.....	89
Gambar IV.13 <i>Use Case Diagram</i> Sistem Informasi Absensi dan	
Penggajian Karyawan	90
Gambar V.1 <i>Flowmap</i> Proses Absensi Karyawan Usulan.....	86
Gambar V.2 <i>Use Case Diagram</i> Sistem Usulan.....	87
Gambar V.3 <i>Activity Diagram</i> Proses Login.....	93
Gambar V.4 <i>Activity Diagram</i> Melakukan untuk absen	94

Gambar V.5	<i>Activity Diagram</i> Mengelola Data Karyawan.....	95
Gambar V.6	<i>Activity Diagram</i> Mengelola Data Jabatan.....	96
Gambar V.7	<i>Activity Diagram</i> Menginput Data Lembur.....	97
Gambar V.8	<i>Activity Diagram</i> Menginput Data Cuti	98
Gambar V.9	<i>Activity Diagram</i> Mengelola <i>Shift</i>	99
Gambar V.10	<i>Activity Diagram</i> Mengelola Data Rekap Absensi	100
Gambar V.11	<i>Activity Diagram</i> Mengelola Data Rekap Lembur	101
Gambar V.12	<i>Activity Diagram</i> View Rekap Disetujui Pimpinan.....	102
Gambar V.13	<i>Sequence Diagram</i> Login	103
Gambar V.14	<i>Sequence Diagram</i> Melakukan Absen	104
Gambar V.15	<i>Sequence Diagram</i> Mengelola Data Karyawan.....	105
Gambar V.16	<i>Sequence Diagram</i> Mengelola Data Jabatan	106
Gambar V.17	<i>Sequence Diagram</i> Mengelola <i>Shift</i>	107
Gambar V.18	<i>Sequence Diagram</i> Mengolah Data Rekap Absensi	108
Gambar V.19	<i>Sequence Diagram</i> Mengolah Data Rekap Lembur	109
Gambar V.20	<i>Sequence Diagram</i> Menginput Cuti.....	110
Gambar V.21	<i>Sequence Diagram</i> Menginput Lembur	111
Gambar V.22	<i>Sequence Diagram</i> View Rekap Absensi Disetujui Pimpinan ..	112
Gambar V.23	<i>Sequence Diagram</i> View Rekap Lembur Disetujui Pimpinan ..	113
Gambar V.24	Class Diagram Sistem Usulan	114
Gambar V.25	Deployment Diagram yang Diusulan.....	118
Gambar V.26	HIPO Sistem Usulan	119
Gambar V.27	Flowchart Sistem Usulan	120
Gambar V.28	Rancangan Form <i>Login</i>	121
Gambar V.29	Rancangan Form <i>Login</i> Absensi.....	122
Gambar V.30	Rancangan Data Karyawan	122
Gambar V.31	Rancangan Form Tambahan Karyawan	123
Gambar V.32	Rancangan Data Jabatan.....	123
Gambar V.33	Rancangan Form Tambahan Jabatan	124
Gambar V.34	Rancangan Data <i>Shift</i>	125
Gambar V.35	Rancangan Form Tambahan <i>Shift</i>	125

Gambar V.36 Rancangan Data Lembur	126
Gambar V.37 Rancangan Form Tambahan Lembur.....	126
Gambar V.38 Rancangan Data Cuti	127
Gambar V.39 Rancangan Data Absensi.....	128
Gambar V.40 Rancangan Data Lembur	128
Gambar V.41 Rancangan Verifikasi Rekap Absensi.....	129
Gambar V.42 Rancangan Verifikasi Rekap Lembur	130

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi merupakan teknologi yang dapat membantu manusia dalam membuat, mengubah, menyimpan, mengkomunikasikan dan menyebarkan informasi. Pertumbuhan teknologi informasi, memberi dampak positif sehingga orang menjadi lebih cepat, tepat waktu dan berinovasi dalam melaksanakan berbagai aktivitas. Pertumbuhan teknologi informasi ditandai dengan banyaknya pengguna komputer, baik untuk kepentingan pribadi, perusahaan atau bisnis. Untuk kebutuhan informasi yang dapat dilakukan adalah dengan membangun sistem informasi.

Sistem informasi dibangun untuk mempermudah penyajian informasi kepada pemakainya, dan banyak dimanfaatkan oleh perusahaan-perusahaan industri baik manufaktur maupun jasa. Selain itu sistem informasi juga digunakan sebagai alat untuk pendukung pengambilan keputusan dalam meningkatkan setiap aspek yang ada disuatu perusahaan baik dalam bidang operasional maupun non-operasional salah satunya yaitu dengan absensi.

Kegiatan absensi karyawan dalam sistem informasi sangat penting untuk memudahkan bagian *Human Resource Development* (HRD) dalam mengolah data. Berkembangnya teknologi informasi yang semakin pesat Sistem absensi secara manual dirasakan kurang mampu memberikan manfaat yang memadai mulai, dari merekap data absensi karyawan. Perusahaan-perusahaan yang dahulunya menggunakan sistem absensi, kini mulai beralih pada sistem informasi yang berbasis komputer.

Pentingnya sistem informasi absensi bagi perusahaan adalah untuk mengurangi kecurangan dalam kegiatan absensi karyawan dan memudahkan bagian HRD dalam mengolah data. Dengan berkembangnya Teknologi Informasi yang semakin pesat, sistem absensi secara manual dirasakan sudah kurang mampu

memberikan manfaat yang memadai mulai dari merekap data absensi karyawan. Perusahaan kini mulai beralih pada sistem yang berbasis komputer.

PT Rekadaya Multi Adiprima (PT. RMA) berdiri pada tahun 1994 sebagai pionir dalam memproduksi *insulator dash panel*, *door trim* dan *pad silencer* untuk kendaraan roda empat, dan produknya banyak digunakan oleh produsen otomotif nasional sebagai komponen *Original Equipment Manufacture* (OEM). PT. RMA merupakan perusahaan yang cukup besar sehingga banyak sumber daya manusia (SDM) yang berkarya sebagai karyawan di PT. RMA.

Sistem informasi absensi karyawan pada PT. RMA masih menggunakan proses manual dengan sistem mesin ceklok, data absensi dicatat di *microsoft excel*. Penggunaan sistem tersebut membutuhkan waktu yang cukup lama selama kurang lebih setengah hari dalam merekap data absensi karena perlu direkap satu persatu setiap harinya. Data absensi dan lembur karyawan memerlukan memori yang cukup banyak karena setiap bulannya data tersebut selalu disalin lalu direkap ulang dan data lembur karyawan dicatat di selembur kertas lalu direkap menggunakan *microsoft excel*. Kesulitan dalam mencari data tersebut memakan waktu yang cukup lama menyebabkan hasil yang diharapkan dari proses absensi kurang cepat.

Untuk mengatasi masalah tersebut, diperlukan sistem yang terkomputerisasi agar lebih mudah dan cepat serta data-data dapat saling terintegrasi dan untuk membantu perusahaan dalam memperbaiki masalah tersebut. Adapun judul Tugas Akhir ini adalah “RANCANG BANGUN SISTEM INFORMASI ABSENSI MENGGUNAKAN PHP 5.6.3 DAN MYSQL 5.6.21 PADA PT REKADAYA MULTI ADIPRIMA”.

1.2 Pokok Permasalahan

Permasalahan yang terjadi pada PT Rekadaya Multi Adiprima di bagian di Departemen HRD (*Human Resource Development*):

1. Proses absensi karyawan menggunakan mesin ceklok sehingga perekapan data absensi diinput satu persatu ke *microsoft excel* dan memerlukan

banyak ruang memori untuk menyimpan data absensi sehingga terjadi penumpukan file.

2. Data-data yang berkaitan dengan absensi belum saling terintegrasi sehingga membutuhkan waktu yang lama saat pencarian data.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Merancang bangun Sistem Informasi yang dapat membantu dalam merekap data absensi sehingga menjadi lebih mudah dalam mengolah data tersebut.
2. Merancang sistem informasi Absensi karyawan yang menggunakan *database* sehingga dalam pengolahan data sistem absensi menjadi saling terintegrasi dengan lebih baik.

1.4 Batasan Masalah

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

1. Tempat penelitian berlokasi di PT Rekadaya Multi Adiprima.
2. Penelitian dilakukan dibagian HRD (*Human Resource Development*) selama 1 bulan mulai dari bulan Agustus 2016 s.d. September 2016.
3. Hanya mengenai sistem informasi absensi karyawan pada PT Rekadaya Multi Adiprima.
4. Penelitian dilakukan berdasarkan absensi, lembur dan cuti karyawan.

1.5 Manfaat Penelitian

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

1. Bagi perusahaan
 - a. Agar bagian personalia lebih mudah dan cepat dalam merekap data.
 - b. Membuat basis data agar data saling berintegrasi.

2. Bagi mahasiswa
 - a. Memberikan kemampuan dalam mengaplikasikan teori secara jelas terhadap masalah yang diamati.
 - b. Memberikan wawasan dan pengalaman kepada mahasiswa dalam menganalisis suatu sistem dan diharapkan dapat memberikan suatu solusi permasalahan.
3. Bagi pihak lain

Hasil penelitian ini diharapkan dapat memberikan sumbangan pemikiran dan sebagai referensi bagi peneliti lain yang ingin melakukan penelitian serupa.

1.6 Sistematika Penulisan

Tugas Akhir ini disusun secara sistematis guna memberikan gambaran yang jelas mengenai isi dan pembahasan yang ada di dalamnya.

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 adalah seputar tentang sistem absensi karyawan.

BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan metode ilmiah dalam mencari, mengembangkan, dan menguji kebenaran tentang suatu pengetahuan. Selain itu dijelaskan pula kerangka pemecahan masalah yang menguraikan tahap-tahap untuk mengatasi permasalahan yang telah diidentifikasi.

BAB IV PENGOLAHAN DATA DAN ANALISIS SISTEM

Pada bab ini akan membahas tentang data yang telah diperoleh berdasarkan penelitian selama melaksanakan Praktik Kerja Lapangan di PT Rekadaya Multi Adiprima seperti tentang profil perusahaan dan bagian HRD (*Human Resource Development*) sebagai objek penelitian, pengolahan data absensi karyawan, dokumen yang terlibat dan laporan yang dihasilkan oleh sistem.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari analisis sistem yang meliputi diagram alir sistem yang berjalan, perancangan basis data, perancangan tampilan layar, perancangan UML, perancangan hierarki menu dan perancangan *interface*.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan secara keseluruhan yang diperoleh dari hasil penelitian serta saran-saran dalam penerapan sistem untuk perusahaan dan pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Kata sistem berasal dari bahasa Yunani, yaitu *systema* yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, bisa diartikan sekelompok elemen yang independen, namun saling berkaitan sebagai satu kesatuan (Rusdiana dan Irfan, 2014).

Sistem adalah kumpulan dari komponen-komponen yang saling berkaitan yang bekerja bersama untuk mencapai beberapa tujuan (Djahir dan Pratita, 2014). Menurut (Sutabri, 2014) sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu (Hutahaeen, 2014).

2.1.1 Karakteristik Sistem

Supaya sistem itu dikatakan sistem yang baik memiliki karakteristik tertentu, yaitu mempunyai komponen, batasan sistem (*boundary*), lingkungan luar sistem, penghubung sistem (*interface*), masukan sistem (*input*), keluaran (*output*), pengolahan (*processing*), sasaran (*objectives*) dan tujuan (Hutahaeen, 2014).

1. Mempunyai komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Mempunyai batasan sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Mempunyai lingkungan luar sistem

Lingkungan luar sistem adalah di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Mempunyai penghubung sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Mempunyai masukan sistem (*input*)

Masukan atau *input* adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem program komputer adalah *maintenance input*, sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Mempunyai keluaran (*output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

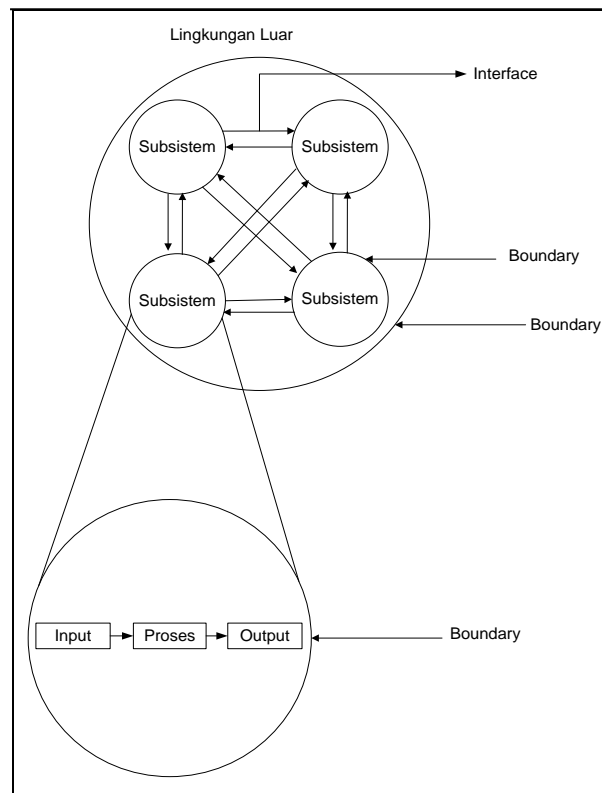
7. Mempunyai pengolahan (*processing*)

Suatu sistem dapat menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Contoh dalam sistem produksi akan mengolah masukan berupa bahan baku menjadi barang jadi.

8. Mempunyai sasaran (*objective*) dan tujuan

Suatu sistem pasti memiliki sasaran (*objective*) atau tujuan (*goal*). Sasaran dari sistem sangat dibutuhkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

Untuk memudahkan dalam membayangkan dan memahami penjelasan mengenai karakteristik sistem yang disampaikan, maka disajikan sebuah bagan sederhana untuk mengilustrasikannya. Gambar II.1 berikut ini menunjukkan bagan sederhana mengenai karakteristik sistem.



Gambar II.1 Karakteristik Sistem
(Sumber: Hutahaean, 2014)

2.1.2 Klasifikasi Sistem

Menurut (Sutabri, 2014), sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang

berbeda untuk setiap kasus yang terjadi dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya:

1. Sistem abstrak dan sistem fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologi, yaitu sistem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, dan sistem persediaan barang.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem tata surya. Sistem buatan manusia adalah sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine system*. Misalnya sistem informasi berbasis komputer (bisnis online dan *e-commerce*).

3. Sistem tertentu (*deterministic*) dan sistem tak tentu (*probabilistic*)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi, sebagai contoh adalah *social network*. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. Contohnya adalah ramalan cuaca.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luar. Contohnya adalah sistem adat masyarakat Baduy. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya. Contohnya adalah teknologi *teleconference* yang digunakan untuk *meeting*, diskusi dan proses belajar mengajar.

2.2 Pengertian Informasi

Informasi atau dalam bahasa Inggrisnya adalah *information*, berasal dari kata *informacion* bahasa Prancis. Kata tersebut diambil dari bahasa Latin, yaitu *informationem* yang artinya konsep, ide, garis besar. Informasi adalah suatu data yang sudah diolah atau diproses sehingga menjadi suatu bentuk yang memiliki arti bagi penerima informasi yang memiliki nilai bermanfaat (Rusdiana dan Irfan, 2014).

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sumber dari informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian serta merupakan suatu kesatuan yang nyata dan merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut melalui suatu model sehingga menghasilkan informasi (Sutabri, 2014).

2.2.1 Pengelompokan Informasi

Informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 bagian, yaitu (Sutabri, 2014):

1. Informasi strategis

Informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.

2. Informasi taktis

Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.

3. Informasi teknis

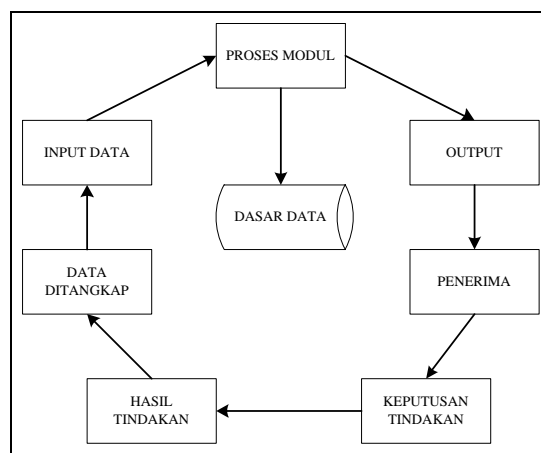
Informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan stok, retur penjualan dan laporan kas harian.

2.2.2 Fungsi Informasi

Menurut (Sutabri, 2014) fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Akan tetapi dalam kebanyakan pengambilan keputusan yang kompleks, informasi hanya dapat menambah kemungkinan kepastian atau mengurangi bermacam-macam pilihan.

2.2.3 Siklus Informasi

Data diolah melalui suatu model informasi. Si penerima akan menerima informasi tersebut untuk membuat suatu keputusan dan melakukan tindakan yang akan mengakibatkan munculnya sejumlah data lagi. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model, dan seterusnya sehingga membentuk suatu siklus. Siklus inilah yang disebut Siklus Informasi (*Information Cycle*) seperti pada gambar II.2.



Gambar II.2 Siklus Informasi
(Sumber: Sutabri, 2014)

2.3 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Sutabri, 2014).

2.3.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok tersebut saling berinteraksi satu dengan yang lain membentuk satu kesatuan untuk mencapai sasaran (Sutabri, 2014).

1. Blok masukan (*input block*)

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

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian sistem secara keseluruhan. Teknologi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

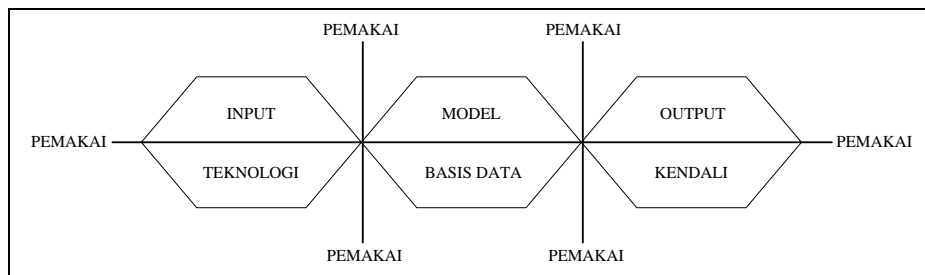
5. Blok basis data (*database block*)

Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan perangkat lunak digunakan untuk memanipulasinya. Data

perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management System*).

6. Blok kendali (*control block*)

Banyak hal dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan pada sistem itu sendiri, sabotase, dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dicegah dan bila terlanjur terjadi maka kesalahan-kesalahan dapat dengan cepat diatasi.



Gambar II.3 Komponen Sistem Informasi
(Sumber: Sutabri, 2014)

2.4 Konsep Absensi

Konsep Absensi merupakan pencatatan jam hadir pada kertas atau kartu jam hadir yang dilakukan oleh setiap pegawai atau pekerja yang nantinya akan direkap untuk mengetahui jumlah kehadiran selama sebulan dan data tersebut digunakan untuk membuat laporan absensi.

2.4.1 Pengertian Absensi

Absensi merupakan dokumen yang mencatat jam hadir setiap karyawan di perusahaan. Catatan jam hadir karyawan ini dapat berupa daftar hadir biasa, dapat pula berbentuk kartu hadir yang diisi dengan mesin pencatat waktu. Pekerjaan mencatat waktu pada dasarnya dapat dipisahkan menjadi dua bagian yaitu pencatatan waktu hadir (*attendance time keeping*) dan pencatatan waktu kerja (*shop time keeping*).

Pencatatan jam hadir pada kartu jam hadir yang dilakukan oleh setiap pegawai/pekerja bisa mempengaruhi gaji bersih yang akan diterima oleh pegawai setiap bulannya. Karena apabila pegawai/pekerja lupa atau tidak mencatatkan jam hadirnya pada kartu jam hadir akan mempengaruhi komponen-komponen yang ada pada gaji.

Pencatatan waktu hadir dimaksudkan untuk mengumpulkan data mengenai jumlah jam hadir karyawan dalam suatu periode pembayaran dan juga mengenai tarif upah untuk pekerjaan yang dilakukan. Pencatatan waktu kerja dimaksudkan untuk mencatat jam kerja sesungguhnya yang digunakan oleh karyawan dalam setiap pekerjaan (*job*) atau departemennya. Catatan waktu kerja ini dapat digunakan untuk mengecek catatan waktu hadir dan juga mendapatkan data produksi yang diperlukan untuk distribusi upah dan gaji dan perhitungan intensif.

Menurut (Simonna, 2009) absensi adalah suatu pendataan kehadiran, bagian dari pelaporan aktifitas suatu institusi, atau komponen institusi itu sendiri yang berisi data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan.

Sering tidak adanya dari tempat kerja merupakan perilaku yang menunjukkan moral yang buruk atau sindrom malas masuk. Namun, ada beberapa perusahaan yang menerapkan kebijakan yaitu memberikan kelonggaran bagi karyawan yang melakukan absensi dikarenakan mengidap penyakit tertentu atau izin tertentu. Namun ada juga perusahaan yang tidak mengizinkan karena karyawan mengidap penyakit yang ringan dan akibatnya, banyak karyawan merasa wajib masuk kerja saat sakit, dan menularkan penyakit ke rekan kerja mereka. Hal ini menyebabkan ketidakhadiran yang lebih besar dan penurunan

produktivitas antara pekerja lain yang mencoba untuk bekerja saat sakit.
Penyebab-Penyebab Karyawan Melakukan Absensi:

1. Menurut Nelson, orang yang sering melakukan absensi dikarenakan ia tidak puas dengan pekerjaan sehingga mereka sering melakukan absensi.
2. Model psikologis yang membahas ini adalah “penarikan model”, menganggap ketidakhadiran disebabkan oleh kondisi kerja yang tidak memuaskan.
3. Konflik antar pegawai yang terjadi di tempat kerja.
4. Bentuk protes karyawan terhadap gaji yang tidak sesuai atau tidak adanya hal-hal yang memotivasi.
5. Tingkat stress yang tinggi.
6. Lingkungan kerja yang sebagian besar mempengaruhi individu.
7. Kesehatan pegawai
8. Kemampuan fisik pegawai.

Pada umumnya instansi atau lembaga selalu memperhatikan pegawainya untuk datang dan pulang tepat waktu, sehingga pekerjaan tidak tertunda. Ketidakhadiran seorang pegawai akan berpengaruh terhadap produktivitas kerja, sehingga instansi atau lembaga tidak bisa mencapai tujuan secara optimal.

Presensi atau kehadiran karyawan dapat diukur melalui:

1. Kehadiran karyawan ditempat kerja.
2. Ketepatan karyawan datang atau pulang.
3. Kehadiran karyawan apabila mendapat undangan untuk mengikuti kegiatan atau acara dalam instansi.

2.4.2 Jenis-jenis Absensi

Jenis-jenis absensi tersebut adalah cara penggunaannya dan tingkat dayagunanya. Secara umum jenis-jenis absensi dapat di kelompokkan menjadi dua, yaitu:

1. Absensi Manual, adalah cara pengentrian kehadiran dengan cara menggunakan pena (tanda tangan).

2. Absensi Non Manual (dengan menggunakan alat), adalah suatu cara pengentrian kehadiran dengan menggunakan sistem terkomputerisasi, bisa menggunakan kartu dengan *barcode, finger print,RFID*.

2.5 Pengertian Lembur

Menurut (Thomas, 2002) Pengertian kerja lembur adalah pekerjaan tambahan yang dilakukan di luar jam kerja yang melebihi 40 jam kerja per minggu atau kerja yang dilakukan untuk menyelesaikan pekerjaan yang tidak mungkin diselesaikan dalam hari kerja normal.

Di Indonesia, ketentuan kerja lembur diatur oleh Menteri Tenaga Kerja dengan dikeluarkannya SK Menteri Tenaga Kerja No. 580/M/BM/BK/1992 pasal 2 dan 3, yang menyebutkan bahwa kerja lembur merupakan waktu dimana seorang pekerja bekerja melebihi dari jadwal waktu yang berlaku, yaitu 7 jam sehari dan 40 jam seminggu.

2.6 Pengertian Cuti

Menurut (Kamus Umum Bahasa Indonesia, 2007) yang dimaksud dengan cuti adalah libur beberapa hari lamanya secara resmi tidak bekerja (untuk beristirahat), hampir sama dalam Kamus Umum Bahasa Indonesia pada (Kamus Besar Bahasa Indonesia, 2007) disebutkan bahwa cuti adalah meninggalkan pekerjaan beberapa waktu untuk beristirahat.

Cuti menurut (Setiyanto, Samopa&Alwin, 2013), cuti merupakan salah satu hak pegawai. Cuti dapat digunakan oleh pegawai untuk tidak masuk kerja dengan alasan tertentu, misalkan refreshing, istirahat, sakit, melahirkan, menunaikan kewajiban agama, dan keperluan lain sesuai dengan ketentuan cuti pada masing-masing organisasi. Menurut peraturan Pemerintah Nomor 24 Tahun 1976, Cuti adalah keadaan tidak masuk kerja yang diijinkan dalam jangka waktu tertentu.

2.7 Pengembangan Sistem

Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada.

Sistem yang lama perlu diperbaiki oleh karena beberapa hal, yaitu (Sutabri, 2014):

1. Munculnya masalah pada sistem yang lama.
2. Untuk meraih kesempatan.
3. Adanya instruksi.

Proses perancangan atau pengembangan sistem informasi, mulai dari konsep sampai dengan implementasinya disebut dengan istilah *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Rosa dan Shalahuddin, 2016).

Menurut (Rosa dan Shalahuddin, 2016) tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut:

1. Inisiasi (*initiation*)
Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.
2. Pengembangan konsep sistem (*system concept development*)
Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem analisis manfaat biaya, manajemen rencana dan pembelajaran kemudahan sistem.
3. Perencanaan (*planning*)
Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

4. Analisis kebutuhan (*requirements analysis*)
Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.
5. Desain (*design*)
Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.
6. Pengembangan (*development*)
Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengkodean, pengkompilasian, memperbaiki dan membersihkan program serta peninjauan pengujian.
7. Integrasi dan pengujian (*integration and test*)
Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user* sehingga menghasilkan laporan analisis pengujian.
8. Implementasi (*implementation*)
Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan luar *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.
9. Operasi dan pemeliharaan (*operations and maintenance*)
Mendesripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.
10. Disposisi (*disposition*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

2.7.1 Model Prototipe Secara Umum

Model prototipe dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis yang memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa dan Shalahuddin, 2013). Tahapan-tahapan pada model prototipe (*prototype model*) adalah sebagai berikut (Rosa dan Shalahuddin, 2013):

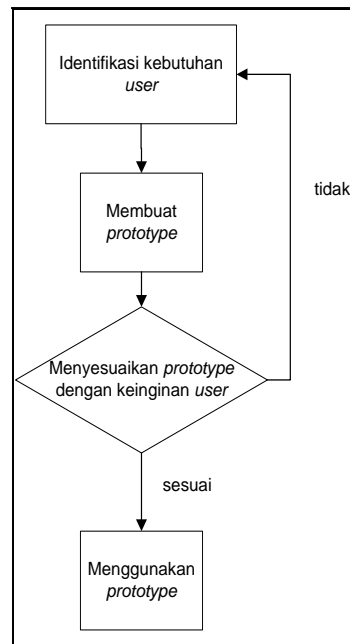
1. Mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat.
2. Membuat prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi.
3. Program prototipe selanjutnya dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*.

Kelemahan model prototipe adalah sebagai berikut (Rosa dan Shalahuddin, 2013):

1. *User* dapat sering mengubah-ubah atau menambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya iterasi ini dapat menyebabkan pengembang banyak mengalah dengan *user* karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.
2. Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan prototipe dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan prototipe untuk didemonstrasikan. Hal ini dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan iteratif tanpa akhir.

2.7.2 Evolutionary Prototype

Evolutionary prototype yaitu *prototype* yang secara terus menerus dikembangkan hingga *prototype* tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sistem. Pada pendekatan evolusioner, suatu *prototype* dibangun berdasarkan pada kebutuhan dan pemahaman secara umum. *Prototype* kemudian diubah dan dievolusikan dari pada dibuang. *Prototype* yang dibuang biasanya digunakan dengan aspek sistem yang dimengerti secara luas dan dibangun atas kekuatan tahapan *evolutionary prototype* (McLeod, 2008). Gambar *evolutionary prototype* dapat dilihat di Gambar II.4:



Gambar II.4 *Evolutionary Prototype Model*

(Sumber: McLeod, 2008)

- a. Identifikasi kebutuhan *user*, pengembang dan *user* atau pemilik sistem melakukan diskusi dimana *user* atau pemilik sistem menjelaskan kepada pengembang tentang kebutuhan sistem yang mereka inginkan.
- b. Membuat *prototype*, pengembang membuat *prototype* dari sistem yang telah dijelaskan oleh *user* atau pemilik sistem.
- c. Menyesuaikan *prototype* dengan keinginan *user* atau pemilik sistem, pengembang menanyakan kepada *user* atau pemilik sistem tentang *prototype* yang sudah dibuat, apakah sesuai atau tidak dengan kebutuhan sistem.

- d. Menggunakan *prototype*, sistem mulai dikembangkan dengan *prototype* yang sudah dibuat.

2.8 Flowchart

Flowchart atau diagram alir merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksi. Sedangkan antara proses digambarkan dengan garis penghubung. *Flowchart* adalah bagan yang menunjukkan alir di dalam program atau prosedur sistem secara logika (Hartono, 2005). Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Notasi yang digunakan dalam suatu *flowmap* merupakan penggabungan notasi *flowchart* program.

1. Pada waktu akan menggambar suatu bagan alir, analisis sistem atau pemrogram dapat mengikuti pedoman-pedoman sebagai berikut:
2. Bagan alir sebaiknya di gambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
3. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
4. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
5. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan, misal:
 - a. "Persiapkan" dokumen
 - b. "Hitung" gaji
6. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
7. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
8. Gunakan simbol-simbol bagan alir yang standar.


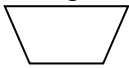
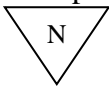
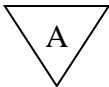
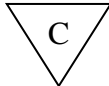

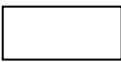
2.8.1 Jenis-Jenis Flowchart

Menurut (Hartono, 2002), jenis-jenis *flowchart* dibagi menjadi 5 jenis, yaitu:





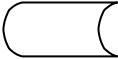




1. Bagan Alir Sistem


Bagan yang memperlihatkan urutan prosedur dan proses dari beberapa *file* dalam media tertentu. Diagram alur sistem ini menggambarkan hubungan antar suatu *file* dengan *file* lain dan media yang dipakai untuk setiap *file*. Simbol-simbol bagan alir sistem dapat dilihat pada tabel II.1.

Tabel II.1 Bagan Alir Sistem

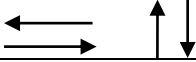

Simbol	Keterangan
Simbol Dokumen 	Menunjukkan dokumen input dan output baik untuk proses manual, mekanik atau computer.
Simbol kegiatan manual 	Menunjukkan pekerjaan manual
Simbol simpanan offline 	File non-komputer yang diarsipkanurut angka (numerical)
	File non-komputer yang diarsipurut huruf (alphabetical)
	File non-komputer yang diarsipurut tanggal (cronological)
Simbol kartu plong 	Menunjukkan input/output yang menggunakan kartu plong (<i>Punchaed card</i>)
Simbol proses 	Menunjukkan kegiatan proses dari operasi program computer

Tabel II.1 Bagan Alir Sistem (Lanjutan)

Simbol	Keterangan
Simbol pengurutan offline 	Menunjukkan proses pengurutan data di luar proses computer
Simbol operasi luar 	Menunjukkan operasi yang dilakukan di luar proses operasi computer
Simbol pita magnetic 	Menunjukkan input/output menggunakan pita magnetic
Simbol hard disk 	Menunjukkan input/output menggunakan hard disk
Simbol diskette 	Menunjukkan input/output menggunakan diskette
Simbol drum magnetic 	Menunjukkan input/output menggunakan drum magnetic
Simbol pita kertas berlubang 	Menunjukkan input/output menggunakan pita kertas berlubang
Simbol keyboard 	Menunjukkan input yang menggunakan on-line keyboard
Simbol display 	Menunjukkan output yang ditampilkan di monitor

Simbol penjelasan 	Menunjukkan penjelasan dari suatu proses
--	--

Tabel II.1 Bagan Alir Sistem (Lanjutan)

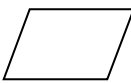

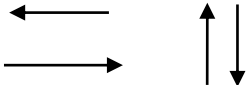
Simbol	Keterangan
Simbol garis alir 	Menunjukkan arus dari proses
Simbol penghubung 	Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain


(Sumber: Hartono (2002))

2. Bagan Alir Program

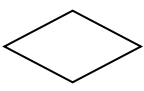

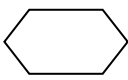
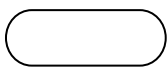
Bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. Diagram alur program ini berguna sebagai langkah awal dalam pembuatan program, dan membuat agar urutan proses pada program menjadi lebih jelas. Simbol-simbol bagan alir program dapat dilihat pada tabel II.2.

Tabel II.2 Bagan Alir Program

Simbol	Keterangan
Simbol input/output 	Simbol input/output digunakan untuk mewakili data input/output
 Simbol proses	Simbol proses digunakan untuk mewakili suatu proses
 Simbol garis alir	Simbol garis alir digunakan untuk menunjukkan arus dari proses

Simbol	Keterangan
Simbol penghubung 	Simbol penghubung digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau di halaman lainnya

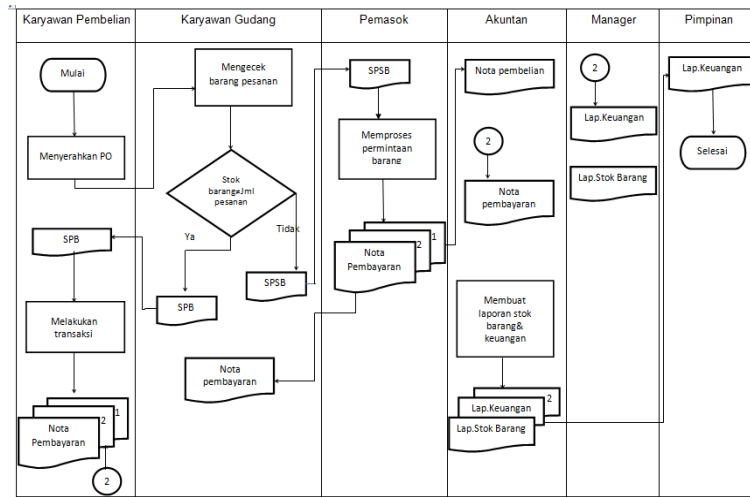
Tabel II.2 Bagan Alir Sistem (Lanjutan)

Simbol	Keterangan
Simbol keputusan 	Simbol keputusan digunakan untuk suatu penyelesaian kondisi di dalam program
Simbol Proses terdefinisi 	Simbol proses terdefinisi digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain
Simbol Persiapan 	Simbol persiapan digunakan untuk memberi nilai awal suatu besaran
Simbol titik terminal 	Simbol titik terminal digunakan untuk menunjukkan awal dan akhir dari suatu proses

Sumber : Hartono (2002)

3. Bagan Alir Dokumen (*Document Flowchart*)

Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusan yang ada. Kegunaan utama adalah untuk menelusuri alur *form* dan laporan sistem dari satu bagian ke bagian lain baik bagaimana alur *form* dan laporan diproses, dicatat dan disimpan.



Gambar II.5 Contoh Gambar Bagan Alir Dokumen






4. Bagan Alir Skematik (*Schematic Flowchart*)

Bagan alir skematik dengan *flowchart* sistem yang menggambarkan suatu sistem atau prosedur. Bagan alir skematik ini bukan hanya menggunakan simbol-simbol *flowchart* standar, tetapi juga menggunakan gambar-gambar komputer, peripheral, *form-form* atau peralatan lain yang digunakan dalam sistem.

5. Bagan Alir Proses (*Process Flowchart*)

Bagan alir proses merupakan teknik penggambaran rekayasa industrial yang memecah dan menganalisis langkah-langkah lanjut dalam suatu prosedur atau sistem. Bagan alir proses memiliki lima simbol khusus.

Tabel II.3 Bagan Alir Proses

Simbol	Nama	Keterangan
	<i>Operation</i>	Menunjukkan suatu operasi
	<i>Movement</i>	Menunjukkan suatu pemindahan
	<i>Storage</i>	Menunjukkan penyimpanan data <i>file</i>
	<i>Inspection</i>	Menunjukkan suatu inspeksi
	<i>Delay</i>	Menunjukkan suatu penundaan

(Sumber: Hartono, 2005)

2.9 *Unified Modeling Language (UML)*

Sampai tahun 1995, konsep objek menjadi populer tetapi dilaksanakan dengan berbagai cara oleh pengembang. Setiap pengembang memiliki metodologi dan notasi sendiri (misalnya, Boach, Coad, Moses, OMT, OOSE, dan SOMA). Kemudian pada tahun 1995, *Rational Software* mengajak tiga pemimpin industry bersama-sama untuk menciptakan sebuah pendekatan tunggal untuk pengembangan sistem berorientasi objek (Dennis et al, 2012). Boach Jacobson dan Rumbaugh bekerja dengan orang lain untuk menciptakan satu set standar teknik diagram yang dikenal sebagai *Unified Modelling Language (UML)*. Tujuan dai UML adalah untuk menyediakan *vocabulary* (kosakata) umum untuk istilah berorientasi objek dan teknik diagram yang cukup kaya untuk model setiap proyek pengembangan sistem dari analisis melalui implementasi. Pada November 1997. *Object Management Group (OMG)* secara resmi menerima UML sebagai standar untuk semua pengembang objek. Selama bertahun-tahun sejak itu, UML telah melalaui beberapa revisi kecil (Dennis et al 2015).

2.9.1 **Diagram UML**

UML mendefinisikan satu set dari empat belas teknologi diagram yang digunakan untuk memodelkan sistem. Dengan yang digunakan dipecah menjadi dua kelompok utama, kelompok pertama untuk pemodelan struktur dari sistem dan yang kedua untuk pemodelan sifat. Diagram pemodelan struktur termasuk *class diagram*, *object diagram*, *package diagram*, *deployment diagram*, *component diagram*, dan *composite structure diagram*. Diagram pemodelan sistem, meliputi *activity diagram*, *sequence diagram*, *communication diagram*, *interaction overview diagram*, *timing diagram*, *behavior state machine, diagram*, *protocol state machine diagram*, dan *use case diagram*. Diagram bergerak dari mendokumentasikan *requirements* untuk menentukan desain. Secara keseluruhan, notasi yang konsisten, integrasi antara teknik diagram, dan penerapan diagram seluruh proses pengembangan, membuat UML menjadi bahasa yang kuat dan fleksibel untuk analis dan pengembang (Dennis et al, 2012).

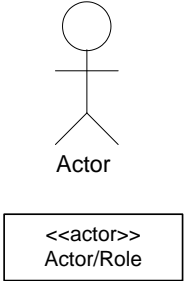

2.9.2 Behavior Diagram

Behavior Diagram menyediakan analisis dengan cara menggambarkan hubungan dinamis antara instansi atau benda yang mewakili sistem informasi bisnis. Diagram ini juga memungkinkan pemodelan perilaku dinamik dari objek individu sepanjang masa aktifnya. Diagram ini mendukung analisis dalam memodelkan *requirements fungsional* agar sistem informasi berkembang (Dennis et al, 2012). Berikut ini akan digambarkan tujuan dasar dari masing-masing *behavior diagram*.

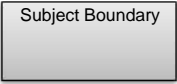

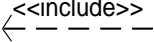
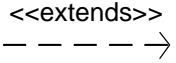
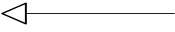
1. Use Case

Use case diagram digunakan untuk memodelkan interaksi antara sistem informasi dan lingkungannya. Lingkungan sistem informasi dapat mencakup *user* dan setiap sistem eksternal yang berinteraksi dengan sistem informasi. Penggunaan utama dari *use case diagram* adalah untuk menyediakan sarana dalam mendokumentasikan dan memahami *requirements* sistem informasi yang sedang berkembang. *Use cases* dan *use case diagram* adalah alat-alat yang paling penting untuk digunakan dalam analisis dan desain sistem berorientasi objek (Dennis et al, 2012). Elemen-elemen dari *use case diagram* terlihat pada Tabel II.4.

Tabel II.4. Elemen-Elemen *Use Case Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Actor</i>	Untuk menggambarkan <i>actor</i> (sistem), yang memperoleh keuntungan dan berada di luar sistem. <i>Actor</i> dapat berasosiasi dengan <i>actor</i> lainnya, dengan menggunakan <i>specialization (superclass association)</i> . <i>Actor</i> ditempatkan di luar <i>subject boundary</i> .
2.		<i>Use case</i>	Untuk mewakili sebuah bagian dari fungsionalitas sistem dan ditempatkan dalam <i>system boundary</i> . Dapat <i>extend</i> dan <i>include use case</i> lain.

Tabel II.4. Elemen-Elemen *Use Case Diagram* (Lanjutan)

No	Elemen	Nama Elemen	Fungsi Elemen
3.		<i>Subject Boundary</i>	Untuk menyatakan lingkup dari subjek.
4.		<i>Association Relationship</i>	Untuk menghubungkan actor dan <i>use case</i> agar dapat saling berinteraksi.
5.		<i>Include Relationship</i>	Untuk menunjukkan <i>inclusion</i> dari sebuah <i>use case</i> dengan <i>use case</i> lainnya. Arah panah dari <i>base use case</i> ke <i>include use case</i> .
6.		<i>Extend Relationship</i>	Untuk menunjukkan <i>extension</i> dari sebuah <i>use case</i> , untuk menambahkan <i>optional behavior</i> . Arah panah dari <i>extension use case</i> ke <i>base use case</i> .
7.		<i>Generalization Relationship</i>	Untuk menunjukkan generalisasi dari <i>use case</i> khusus ke yang lebih umum.




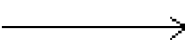
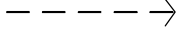


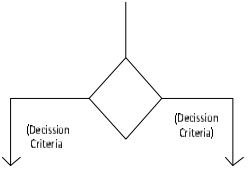
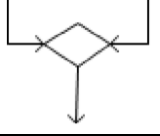

(Sumber: Dennis et al, 2012)

2. Activity Diagram

Activity diagram digunakan untuk memodelkan proses dalam suatu sistem informasi. Diagram ini dapat digunakan untuk memodelkan alur kerja, *use case* individual, atau logika keputusan yang terkandung dalam metode individual. Diagram ini juga menyediakan pendekatan untuk memodelkan proses parallel (Dennis et al, 2012). Elemen-elemen dari *activity diagram* terlihat pada Tabel II.5.


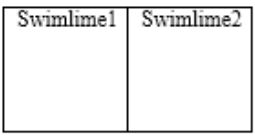
Tabel II.5 Elemen-elemen *Activity Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
----	--------	-------------	---------------

1.		<i>Action</i>	Untuk menggambarkan perilaku yang sederhana dan bersifat <i>non decomposable</i> .
2.		<i>Activity</i>	Untuk mewakili serangkaian aksi (<i>action</i>).
3.		<i>Object Node</i>	Untuk mewakili objek yang terhubung dengan serangkaian <i>object flow</i> .
4.		<i>Control Flow</i>	Untuk mewakili serangkaian pelaksanaan.
5.		<i>Object Flow</i>	Untuk menunjukkan aliran sebuah objek dari sebuah aktivitas (aksi), atau ke sebuah aktivitas (aksi).
6.		<i>Initial Node</i>	Untuk menandakan awal dari serangkaian aksi atau aktivitas.
7.		<i>Final-Flow Node</i>	Untuk menghentikan <i>control flow</i> atau <i>object flow</i> tertentu.
8.		<i>Decision Node</i>	Untuk mewakili suatu kondisi pengujian, yang bertujuan untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> hanya menuju ke satu arah.
9.		<i>Merge Node</i>	Untuk menyatukan kembali <i>decision path</i> yang dibuat dengan menggunakan <i>decision node</i> .
10.		<i>Fork node</i>	Untuk memisahkan perilaku menjadi serangkaian aktivitas yang berjalan secara parallel atau bersamaan.

Tabel II.5 Elemen-elemen *Activity Diagram* (Lanjutan)

No	Elemen	Nama Elemen	Fungsi Elemen
----	--------	-------------	---------------

11.		<i>Join Node</i>	Untuk menyatukan kembali serangkaian aktivitas yang berjalan secara parallel atau bersamaan.
12.		<i>Swimlane</i>	Untuk membagi sebuah <i>activity diagram</i> menjadi beberapa kolom, dengan tujuan menempatkan aktivitas (aksi) tertentu pada individu (objek) yang bertanggung jawab atas terlaksananya aktivitas (aksi) tersebut.

(Sumber: Dennis et al, 2012)

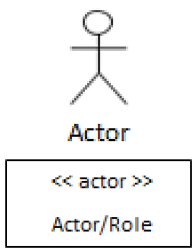
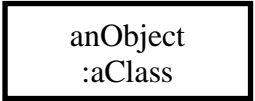


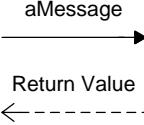

3. *Interaction diagram*

Interaction diagram menggambarkan interaksi antara objek-objek dari suatu sistem informasi berorientasi objek. UML 2.0 menyediakan empat *interaction diagram* yang berbeda-beda, yaitu *sequence diagram*, *communication diagram*, *interaction overview diagram*, dan diagram waktu (Dennis et al, 2012).

- a. *Sequence diagram* memungkinkan analisis untuk menggambarkan interaksi dinamis antara objek-objek dalam suatu sistem informasi. *Sequence diagram* sejauh ini merupakan jenis interaksi yang paling umum digunakan dalam pemodelan berorientasi objek. Diagram ini sangat berguna dalam membantu analisis, memahami spesifikasi *real time* dan *use case* yang rumit. Diagram ini dapat digunakan untuk mendeskripsikan interaksi antara objek-objek, baik secara fisik maupun logis. Hal tersebut menyebabkan diagram ini berguna dalam kegiatan analisis dan juga desain (Dennis et al, 2012). Elemen-elemen dari *sequence diagram* terlihat pada Tabel II.6.

Tabel II.6. Elemen-elemen *Sequence Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
----	--------	-------------	---------------

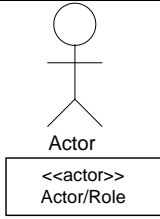
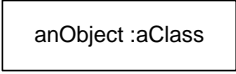

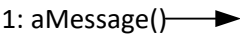
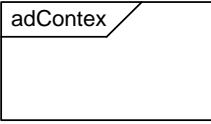
1.		<i>Actor</i>	Untuk menggambarkan <i>actor</i> (sistem), yang memperoleh keuntungan dan berada di luar sistem. Diletakkan di bagian atas diagram. Jika <i>actor</i> bukan manusia, maka menggunakan persegi panjang.
2.		<i>Object</i>	Untuk menggambarkan secara berurutan pengiriman data/atau penerimaan <i>message</i> . Diletakkan di bagian atas diagram.
3.		<i>Lifeline</i>	Untuk menggambarkan keberadaan <i>object</i> pada waktu tertentu.
4.		<i>Execution Occurrence</i>	Untuk menggambarkan periode waktu selama <i>object</i> atau <i>actor</i> melakukan aktivitas dalam pengiriman atau penerimaan pesan.
5.		<i>Message</i>	Untuk menyampaikan informasi dari satu objek ke objek lain. Panggilan operasi diberi label satu pesan dikirim dan panah padat sedangkan yang kembali diberi label suatu nilai yang dikembalikan dan ditampilkan sebagai sebuah panah putus-putus.
6.		<i>Object Destruction</i>	Untuk menggambarkan penghancuran sebuah elemen header setelah selesainya sebuah operasi.
7.		<i>Frame</i>	Untuk menunjukkan konteks dari <i>sequence diagram</i> .

(Sumber: Dennis et al, 2012)

- b. Communication diagram memberi pandangan alternatif dari interaksi diterima yang terjadi di antara objek-objek dalam suatu sistem informasi berorientasi objek. Jika *sequence diagram* menekankan penyusunan berbasis waktu dari suatu kegiatan, *communication*

diagram lebih berfokus pada sekumpulan pesan yang disampaikan dalam sekumpulan objek yang berkolaborasi. Urutan atau penyusunan berdasarkan waktu dari pesan ditunjukkan melalui skema urutan penomoran (Dennis et al, 2012). Elemen-elemen dari *communication diagram* terlihat pada Tabel II.7.

Tabel II.7. Elemen-Elemen *communication diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Actor</i>	Untuk menggambarkan <i>actor</i> (sistem), yang memperoleh keuntungan dan berada di luar sistem.
2.		<i>Object</i>	Untuk menggambarkan pengiriman dan/atau penerimaan <i>message</i> . Diletakkan dibagian atas diagram
3.		<i>Associatioion</i>	Untuk menggambarkan hubungan antara aktor-aktor dan/atau objek-objek. Pesan dikirim melalui <i>association</i> .
4.		<i>Message</i>	Untuk menyampaikan informasi dari satu objek ke objek lainnya, arah ditunjukkan menggunakan sebuah mata panah. Urutan ditunjukkan oleh nomor urut.
5.		<i>Frame</i>	Untuk menunjukkan konteks dari <i>communication diagram</i> .

(Sumber: Dennis et al, 2012)

- c. *interaction overview diagram* membantu analias memahami *use case* yang kompleks. Diagram ini memberikan gambaran kontrol aliran proses. Diagram ini memperluas *activity diagram* melalui penambahan fragmen-fragmen *sequence* dari *sequence diagram*. Akibatnya, fragmen-fragmen *sequence* diperlakukan seolah-olah

fragmen merupakan *activity* dalam *activity diagram*. Keuntungan utama menggunakan diagram ini adalah bahwa anda dapat dengan mudah memodelkan aliran *sequence* alternatif walaupun dalam prakteknya, hal ini bisa dicapai dengan menggunakan *activity diagram* disamping menggunakan *use case diagram* (Dennis et al, 2012).

- d. Timing diagram menggambarkan interaksi antara objek sepanjang suatu poros waktu. Tujuan utama dari diagram ini adalah untuk menunjukkan perubahan keadaan objek dalam menanggapi peristiwa-peristiwa yang terjadi dari waktu ke waktu. Diagram ini cenderung sangat berguna ketika mengembangkan sistem yang *real time* atau *embedded* (Dennis et al, 2005).




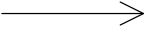

4. State Machines

Dalam UML 2.0, ada dua state machines yang berada yaitu *behavior state machines* dan *protocol state machines*. *Behavior state machines* digunakan untuk menjelaskan perubahan yang dialami objek selama masa aktifnya. *Protocol state machines* menggambarkan spesifikasi sequence dari suatu peristiwa, dapat direspon objek (Dennis et al, 2005).

- a. *Behavior state machines* adalah metode untuk memodelkan keadaan atau sekumpulan nilai yang berbeda. Sebagai contoh, seorang pasien bisa berubah statusnya dari waktu ke waktu, misalnya seorang yang sebelumnya pasien baru bisa berubah menjadi pasien lama. Masing-masing kategori pasien ini benar-benar menunjukkan keadaan yang berbeda dari pasien yang lama. Keadaan yang berbeda tersebut dihubungkan dengan peristiwa yang menyebabkan pasien berubah dari suatu keadaan ke keadaan lainnya (Dennis et al, 2012). Elemen-elemen *behavior state machines* terlihat pada Tabel II.8.

Tabel II.8. Elemen-Elemen *Behavior State Machines*

No	Elemen	Nama Elemen	Fungsi Elemen
----	--------	-------------	---------------

1.		<i>State</i>	Untuk mewakili keadaan dari suatu objek.
2.		<i>Intial State</i>	Untuk menggambarkan keadaan awal dari sebuah objek.
3.		<i>Find State</i>	Untuk menggambarkan penyelesaian suatu keadaan atau aktivitas.
4.	anEvent :	<i>Event</i>	Untuk menunjukkan nama dari tansition, dan merupakan kejadian penting yang memicu perubahan suatu keadaan.
5.		<i>Transition</i>	Untuk menggambarkan perpindahan <i>flow event</i> dari <i>state</i> suatu ke <i>state</i> yang lain.
6.		<i>Frame</i>	Untuk menunjukkan konteks dari <i>behavior state machines</i> .

(Sumber: Dennis et al (2012))

- b. *Protocol state machines* digunakan merancang hubungan antara unsur-unsur *interface* pada suatu kelas. Sebagai contoh, biasanya membuka *file* atau *database* sebelum membuat *query* atau memperbarui *query*. Tidak seperti *behavior state machines*, *protocol state machines* bisa dikaitkan dengan port pada komponen atau *interface* pada kelas (Dennis et al, 2012).

2.9.3 *Structure Diagram*

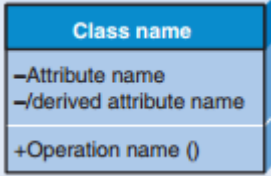
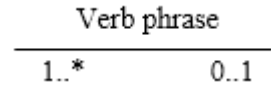
Structure Diagram (diagram struktur) menyediakan cara untuk mewakili data dan hubungan statis yang berada dalam sistem informasi. Dibawah ini, akan digambarkan tujuan dasar masing-masing diagram struktur (Dennis et al, 2012).

1. *Class Diagram*

Tujuan utama *class diagram* adalah untuk menciptakan sebuah *vocabulary* (kosakata) yang digunakan oleh analis dan pengguna. *Class diagram* biasanya merupakan hal-hal, ide-ide atau konsep yang terkandung dalam aplikasi. Misalnya, jika anda sedang membangun sebuah aplikasi

penggajian, *class diagram* mungkin akan berisi kelas yang mewakili hal-hal seperti karyawan, cek, dan pendaftaran gaji. *Class diagram*, juga akan menggambarkan hubungan antara kelas (Dennis et al, 2012). Elemen-elemen dari *class diagram* terlihat pada Tabel II.9.

Tabel II.9. Elemen-Elemen *Class Diagram*

No.	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Class</i>	Untuk menggambarkan <i>class</i> atau kumpulan <i>object-object</i> yang mempunyai <i>attribute</i> , dan <i>operation</i> .
2.	Attribute name /derived attribute name	<i>Attribute</i>	Untuk menggambarkan atribut yang dimiliki suatu <i>class</i> .
3.	Operation name ()	<i>Operation</i>	Untuk menunjukkan suatu tindakan/fungsi yang dapat dilakukan oleh <i>class</i> .
4.		<i>Association</i>	Untuk menggambarkan hubungan antara dua <i>class</i> dan hubungan suatu <i>class</i> itu sendiri.

(Sumber: Dennis et al, 2012)

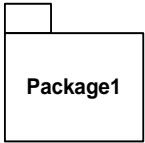
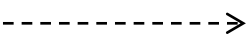
2. *Object diagram*

Object diagram sangat mirip dengan *class diagram*. Perbedaan utama adalah bahwa *object diagram* menggambarkan objek dan hubungan antar objek. Tujuan utama dari *object diagram* adalah memungkinkan analisis untuk mengungkap rincian tambahan dari sebuah kelas. Dalam beberapa kasus, representatif variabel dari sebuah *class diagram* dapat membantu pengguna atau analis dalam menentukan atribut tambahan yang relevan, hubungan, dan atau operasi, atau mungkin menemukan beberapa atribut, hubungan, atau operasi yang salah tempat (Dennis et al, 2012).

3. *Package Diagram*

Package diagram digunakan untuk mengelompokkan elemen diagram UML yang berlainan secara bersama-sama ke dalam tingkat *construct* yang lebih tinggi yaitu berupa sebuah paket (Dennis et al, 2012). Elemen-elemen dari *package diagram* terlihat pada Tabel II.10.

Tabel II.10. Elemen-elemen *Package Diagram*

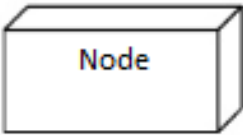
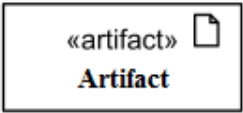
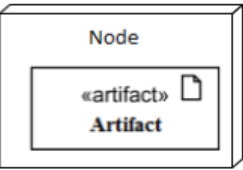
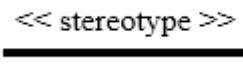
No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Package</i>	Untuk menyederhanakan diagram UML dengan mengelompokkan elemen-elemen terkait ke elemen satu tingkat lebih tinggi.
2.		<i>Dependency Relationship</i>	Untuk menggambarkan ketergantungan antara paket.

(Sumber: Dennis et al, 2012)

4. *Deployment Diagram*

Deployment diagram digunakan untuk mewakili hubungan antara komponen-komponen *hardware* yang digunakan dalam infrastruktur fisik sistem informasi. Misalnya, ketika mengarang suatu sistem informasi terdistribusi yang akan menggunakan jaringan luas, *deployment diagram* dapat digunakan untuk menunjukkan hubungan komunikasi antara node yang berbeda dalam jaringan. *Deployment diagram* juga dapat digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Dalam hal ini, *deployment diagram* juga dapat digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Dalam hal ini, *deployment diagram* mewakili lingkungan pembuatan *software* (Dennis et al, 2012). Elemen-elemen dari *deployment diagram* terlihat pada Tabel II.11.

Tabel II.11 Elemen-Elemen *Deployment Diagram*

No	Elemen	Nama Elemen	Fungsi Elemen
1.		<i>Node</i>	Untuk menggambarkan sumber daya komputasi dalam sebuah sistem (misalnya, computer klien, <i>server</i> , jaringan yang terpisah, atau individu perangkat jaringan).
2.		<i>Artifact</i>	Untuk menggambarkan spesifikasi dari <i>software</i> atau <i>database</i> , misalnya <i>file</i> sumber, tabel <i>database</i> , <i>executable file</i> .
3.		<i>Node with a Deployed Artifact</i>	Untuk menggambarkan <i>artifact</i> yang ditempatkan pada node fisik. Mendukung pemodelan distribusi perangkat lunak melalui jaringan.
4.		<i>Communication Path</i>	Untuk menggambarkan hubungan antara dua node untuk bertukar pesan.

(Sumber: Dennis et al, 2012).

5. *Component Diagram*

Component diagram memungkinkan perancang untuk memodelkan hubungan fisik antara modul-modul fisik dari kode. Diagram ini bila dikombinasikan dengan *deployment diagram* dapat digunakan untuk menggambarkan distribusi fisik dari modul *software* melalui jaringan. Misalnya, ketika merancang sistem *client-server*, hal ini berguna untuk menunjukkan kelas atau paket kelas mana yang akan berada pada node *lient* dan mana yang akan berada di *server*. *Component diagram* juga dapat berguna dalam merancang dan mengembangkan sistem berbasis komponen (Dennis et al, 2012).

6. *Composite Structure Diagram*

UML 2.0 menyediakan diagram baru, ketika struktur internal dari *class* bersifat kompleks maka disediakan *composite structure diagram*. *Composite structure diagram* ini digunakan untuk pemodelan hubungan antara bagian-bagian dari sebuah *class*. Sebagai contoh, ketika pemodelan pendaftaran penggajian, analis mungkin ingin *class* yang mewakili seluruh laporan serta *class* yang mewakili *heade*, *footer*, dan garis-garis detail laporan. Dalam sebuah *class diagram* standar, akan membutuhkan analis untuk memodelkan pendaftaran penggajian menjadi empat *class* terpisah yang memiliki hubungan, kemudian menghubungkan mereka bersama-sama. Sedangkan, jika menggunakan *composite structure diagram* maka akan ada tiga *subclass*: *header*, *footer*, dan garis detail. *Composite structure diagram* juga berguna ketika dilakukan pemodelan struktur internal komponen untuk sistem berbasis komponen. Seringkali, *composite structure diagram* merupakan mekanisme pemodelan yang kurang efektif karena pemodelan juga bisa dilakukan dengan mengkomunikasikan penggunaan paket dan diagram paket (Dennis et al, 2012).

2.10 **Kamus Data**

Menurut Jogiyanto (2005) kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data diharapkan, analisis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

Kamus data dapat berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan di dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Berikut adalah contoh penulisan kamus data:

Spesifikasi tabel pemasok
Nama tabel : Pemasok
Tipe : File *master*

Tabel II.12 Contoh Kamus Data Untuk Tabel Pemasok

No	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	ID pemasok	ID_pemasok	Varchar	10	<i>Primary Key</i>
2	Nama pemasok	Nama_pemasok	Char	40	
3	Alamat pemasok	Alamat	Varchar	100	
4	Nomor telepon	Telepon	Varchar	12	

(Sumber: Jogiyanto, 2005)

2.11 Basis Data (*Database*)

Basis data terdiri dari dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai marka atau gudang, tempat bersarang/berkumpul, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep dan sebagainya yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Menurut Bambang Hariyanto (2004) basis data adalah kumpulan data (elementer) yang secara logik berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. Basisdata adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi. Menurut Fatansyah (2015) sebagai satu kesatuan istilah, basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.12 *Hierarchy plus Input-Proses-Output (HIPO)*

Menurut Jogiyanto (2005) *Hierarchy plus Input-Process-Output (HIPO)* merupakan metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program, akan tetapi sekarang HIPO banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO berbasis pada fungsi, yaitu setiap modul didalam sistem digambarkan oleh fungsi utamanya.

Menurut (Hartono, 2005) HIPO (*Hierarchy plus Input-Process-Output*) mempunyai sasaran utama sebagai berikut:

1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari sistem.
2. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukkan statemen-statement program yang digunakan untuk melaksanakan fungsi tersebut.
3. Untuk menyediakan penjelasan yang jelas dari *input* yang harus digunakan dan *output* yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari *diagram-diagram* HIPO.
4. Untuk menyediakan *output* yang tepat dan sesuai dengan kebutuhan-kebutuhan pemakai.

Menurut (Jogiyanto, 2005) tujuan HIPO antara lain sebagai berikut (Jogiyanto, 2005):

1. Untuk memberikan struktur yang memungkinkan fungsi suatu sistem dapat dimengerti.
2. Untuk menguraikan fungsi-fungsi yang akan dikerjakan oleh suatu program, bukan untuk mengkhususkan pernyataan program yang dipakai untuk melaksanakan fungsi-fungsi tersebut.
3. Untuk memberikan deskripsi visual dari *input* yang akan dipakai serta *output* yang akan dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkat *diagram*.

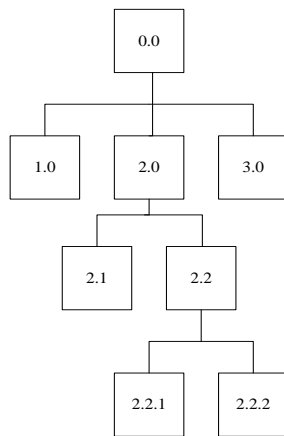
4. Tujuan HIPO yang paling penting adalah untuk menghasilkan *output* yang benar dan dapat memenuhi kebutuhan *user*.

Fungsi-fungsi dari sistem digambarkan oleh HIPO dalam tiga tingkatan yang digambarkan dalam bentuk *diagram* tersendiri. *Diagram* yang digunakan untuk masing-masing tingkatannya, yaitu:

1. *Visual Table Of Contents (VTOC)*

Visual table of contents menggambarkan hubungan fungsi-fungsi di sistem secara berjenjang, *visual table of contents* menggambarkan seluruh program HIPO baik rinci maupun ringkasan yang terstruktur. Pada *diagram* ini nama dan nomor dari program HIPO diidentifikasi.

Struktur paket *diagram* dan hubungan fungsi juga diidentifikasi dalam bentuk hirarki. Keterangan masing-masing fungsi diberikan pada bagian penjelasan yang diikutsertakan dalam *diagram* ini. *Visual table of contents* ini dapat digambarkan pada gambar II.4 sebagai berikut:



Gambar II.4 *Visual Table Of Contents*
(Sumber: Hartono, 2005)

2. *Overview Diagram*

Overview diagram menunjukkan secara garis besar hubungan dari *input*, proses dan *output*. Bagian *input* menunjukkan *item-item* data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *output* berisi

dengan *item-item* data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses.

3. *Detail Diagram*

Detail diagram merupakan *diagram* tingkatan yang paling rendah di *diagram* HIPO. *Diagram* ini berisi elemen-elemen dasar dari paket yang menggambarkan secara rinci kerja dari fungsi.

2.13 *Object Oriented Programming (OOP)*

Pemrograman berorientasi obyek (*object oriented programming* disingkat OOP) memandang aplikasi perangkat lunak sebagai kumpulan obyek yang saling berinteraksi di dalam suatu sistem. Merancang suatu aplikasi dengan teknik OOP dilakukan dengan membagi fungsi-fungsi berdasarkan pembagian tanggung jawab (Azis, 2005). Pada teknik OOP, gambaran cara kerja program diberikan dengan menggunakan tampilan yang lebih mudah dimengerti dan perintah-perintah tidak lagi harus diketikkan secara manual. Beberapa manfaat OOP menurut (Hayder, 2007) adalah sebagai berikut:

1. *Reusability*

Sebuah obyek adalah sebuah entitas yang memiliki kumpulan properti dan metode yang dapat berinteraksi dengan obyek lain. Sebuah obyek mungkin memiliki ketergantungan atas obyek lainnya. Namun obyek biasanya dikembangkan untuk memecahkan serangkaian masalah tertentu. Sehingga, ketika pengembang lain mengalami masalah yang sama pada sistem, mereka dapat menggabungkan *class* sistem sebelumnya untuk proyek mereka dan menggunakannya tanpa mempengaruhi alur kerja yang ada.

2. *Refactoring*

Bila perlu *me-refactor* sistem lama, OOP memberikan manfaat maksimal karena semua obyek adalah entitas kecil dan mengandung sifat dan metode sebagai bagian dari dirinya sendiri. Jadi *refactoring* relatif lebih mudah.

3. *Extensible*

Salah satu fitur inti OOP adalah *extensibility*. Pengembang dapat memperpanjang obyek dan membuat obyek yang baru dengan mempertahankan semua sifat yang diperlukan dan metode dari obyek induk dari yang telah diturunkan, dan kemudian mengekspos fitur baru. Ini disebut "warisan" dan merupakan fitur yang sangat penting dari OOP.

4. *Maintenance*

Kode berorientasi obyek lebih mudah dikelola karena mengikuti konvensi *coding* yang ketat dan ditulis dalam format yang jelas. Misalnya, ketika pengembang memperluas sistem, melakukan *refactors*, atau *debugs*, mereka dapat dengan mudah mengetahui struktur dalam *coding* dan mengelolanya dari waktu ke waktu.

5. *Efficiency*

Konsep pemrograman berorientasi obyek sebenarnya dikembangkan untuk efisiensi yang lebih baik dan kemudahan proses pembangunan. Beberapa pola desain yang dikembangkan untuk menciptakan kode yang lebih baik dan efisien.

Bahasa pemrograman yang mendukung OOP antara lain Visual Foxpro, Java, C++, PHP dan lain-lain. Pada saat PHP dikembangkan, PHP tidak menerapkan fitur OOP itu sendiri. Ketika PHP3 dirilis, fitur OOP yang sangat mendasar diperkenalkan. Kemudian PHP4 dirilis, fitur OOP menjadi lebih matang dengan peningkatan kinerja yang besar. Namun tim PHP menulis ulang mesin inti lagi untuk memperkenalkan model objek yang baru dan dirilis PHP5.

2.14 PHP

Hypertext Preprocessor (PHP) adalah bahasa *server-side scripting* yang digunakan untuk aplikasi web yang dinamis dan interaktif. Sebuah halaman PHP adalah sebuah halaman *HTML* yang memiliki *server-side scripts* yang ditempatkan dalam *server* dan diproses oleh web *server* sebelum dikirim ke *browser* pemakai (Welling dan Thomson, 2003).

Server-side scripts dijalankan ketika *browser* melakukan permintaan *file.php* dari *server*. PHP dipanggil oleh web *server*, dimana proses script perintah yang ada di suatu halaman dieksekusi mulai dari awal sampai akhir di dalam mesin PHP. Setelah *script* PHP tersebut diolah, hasilnya akan ditampilkan kepada *client* melalui web *browser* berupa tampilan *HTML*. Menurut Welling dan (Thomson, 2003), beberapa keunggulan PHP adalah:

1. *High Performance*

PHP sangat efisien. Dengan menggunakan *server* tunggal yang tidak mahal, *user* dapat melakukan banyak pekerjaan setiap harinya.

2. *Database Integration*

PHP mempunyai sambungan ke banyak sistem basis data, antara lain MySQL, PostgreSQL, Oracle, Informix, dan Sysbase databases.

3. *Built-in-Libraries*

PHP dirancang khusus untuk web, dan mempunyai banyak *built-in-function* untuk menampilkan banyak fungsi di dalam web.

4. Harga yang murah

PHP adalah perangkat lunak gratis.

5. Mudah dalam pembelajaran dan penggunaan

Sintaks PHP berdasarkan bahasa pemrograman lainnya, terutama C dan Java.

6. *Portability*

PHP dapat digunakan di banyak sistem operasi yang berbeda.

7. Ketersediaan *Source Code*

Kode PHP dapat langsung diakses dan dimodifikasi secara bebas.

2.15 MySQL (*My Structured Query Language*)

MySQL adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut dengan DBMS (*Database Management System*), sifat dari DBMS ini adalah *open source* (Nugroho, 2007).

MySQL adalah suatu bahasa permintaan yang terstruktur dan telah distandarkan untuk semua program pengakses *database* seperti Oracle, PosgreSQL, SQL Server, dan lain-lain (Kadir, 2008).

Pada MySQL masing-masing tipe data memiliki kegunaan dan keterangan untuk digunakan dalam merancang tabel. Berikut ini beberapa tipe data yang terdapat pada MySQL:

Tabel II.13 Beberapa Tipe Data Pada MySQL

Tipe Data	Ukuran	Keterangan
CHAR	M	Menampung maksimal M karakter (kombinasi huruf, angka, dan simbol-simbol). Jumlah memori yang dibutuhkan selalu M byte. M terbesar adalah 255.
VARCHAR	M	Karakter yang disimpan maksimal M karakter. Jumlah memori yang dibutuhkan tergantung jumlah karakter. M bisa mencapai 65535.
DATE	8 byte	Menyatakan tanggal.
TIME	8 byte	Menyatakan waktu (jam:menit:detik).
TINYINT	1 byte	Bilangan antara -128 sampai dengan +127.
SMALLINT	2 byte	Bilangan antara -32768 sampai dengan +32768.
INT	4 byte	Bilangan antara -2147683648 sampai dengan +2147683647.
FLOAT	4 byte	Bilangan pecahan.
DOUBLE	8 byte	Bilangan pecahan dengan presisi tinggi.

Tabel II.13 Beberapa Tipe Data Pada MySQL (Lanjutan)

Tipe Data	Ukuran	Keterangan
BOOL	1 byte	Untuk menampung nilai true (benar) dan false (salah). Identik dengan TINYINT.
ENUM	-	Menyatakan suatu tipe yang nilainya tertentu (disebutkan dalam pendefinisian).
TEXT	-	Menyimpan teks yang ukurannya sangat panjang.
BLOB	-	Untuk menyimpan data biner (misalnya gambar atau suara).

(Sumber: Kadir, 2008)

2.16 XAMPP

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 dengan internet (Kadir, 2008). Bagian yang terpenting dari XAMPP adalah sebagai berikut (Nugroho, 2008):

1. *Htdoc* adalah *folder* tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.
2. *PhpMyAdmin* merupakan bagian untuk mengelola basis data MySQL yang ada di komputer. Untuk membukanya, buka *browser* lalu ketikkan alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman *phpMyAdmin*.
3. Kontrol Panel yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian adalah tata cara bagaimana suatu penelitian dilaksanakan atau cara yang ditempuh sehubungan dengan penelitian yang dilakukan, dan memilih langkah-langkah sistematis (Hasan, 2002).

Dengan adanya metodologi penelitian, 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. Kegiatan penelitian dikerjakan akan lebih terarah sesuai dengan prosedur yang telah ditetapkan namun tidak menutup kemungkinan akan terjadi variasi urutan penelitian yang tentunya disebabkan oleh permasalahan yang sedang diteliti.

3.2 Jenis dan Sumber Data

Data yang dibutuhkan dalam penelitian ini dikelompokkan menjadi dua jenis, yaitu:

1. Data primer
Data yang diperoleh melalui pengamatan langsung di Departemen *HRD* terhadap sistem yang sedang berjalan dan wawancara dengan pegawai sebagai sumber informasinya. Dalam penelitian ini data tersebut berupa data alur proses absensi pada PT RMA.
2. Data Sekunder
Data sekunder merupakan sumber data penelitian yang diperoleh secara tidak langsung melalui media perantara, kepustakaan, buku-buku, internet dan referensi. Dalam penelitian ini data tersebut berupa data umum perusahaan, profil perusahaan dan struktur organisasi perusahaan serta teori-teori mengenai absensi, pemrograman PHP dan *database* MySQL.

3.3 Metode Pengumpulan Data

Pengumpulan data merupakan sebuah sarana yang menampung dan mengumpulkan data-data dari berbagai sumber dan bentuk yang akan digunakan dalam keperluan menganalisis sistem untuk perancangan sistem usulan.

Sumber data atau informasi penelitian ini berdasarkan kepada jenis data yang diperlukan. Data yang diperoleh dari responden secara langsung yang dikumpulkan melalui *survey* lapangan dengan menggunakan metode pengumpulan data. Metode yang digunakan dalam pengumpulan data penelitian ini, diantaranya:

1. Studi Pustaka

Tahap ini merupakan tahap awal pengumpulan data dari sumber-sumber seperti buku yang berhubungan dengan masalah yang sedang dianalisis. Studi pustaka yang dilakukan berkaitan dengan absensi. Pengembangan sistem, analisis dan desain sistem informasi, pemrograman PHP dan *database* MySQL dari berbagai referensi, baik itu referensi elektronik yang didapat dari internet maupun referensi dari buku teks.

2. Observasi

Tahap ini merupakan tahap melakukan pengamatan langsung pada sistem yang berjalan di Departemen *HRD*. Pengamatan yang dilakukan mengenai alur proses absensi karyawan, dokumen yang terkait dengan sistem yang diteliti, laporan absensi dan penggajia yang digunakan oleh Departemen *HRD*. Hasil dari pengamatan yang dilakukan menjadi landasan dalam melakukan pengembangan sistem yang akan dibuat.

3. Wawancara

Pada tahap ini penulis melakukan tanya jawab secara langsung dengan Departemen *HRD* untuk memperoleh data mengenai alur proses absensi karyawan.

3.4 Metode Pengembangan Sistem

Sistem akan dikembangkan dengan metode *Object Oriented Programming* (OOP). Metode ini menggunakan pemodelan dengan menggunakan *diagram-diagram* yang terdapat dalam *Unified Modeling Language* (UML). Sedangkan metode pengembangan perangkat lunak yang digunakan yaitu dengan metode *Evolutionary Prototype*.

Evolutionary prototype yaitu *prototype* yang secara terus menerus dikembangkan hingga *prototype* tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sistem. Pada pendekatan evolusioner, suatu *prototype* dibangun berdasarkan pada kebutuhan dan pemahaman secara umum. *Prototype* kemudian diubah dan dievolusikan dari pada dibuang. *Prototype* yang dibuang biasanya digunakan dengan aspek sistem yang dimengerti secara luas dan dibangun atas kekuatan tahapan *evolutionary prototype* (McLeod, 2008).

Langkah-langkah dalam pembuatan suatu *evolutionary prototype* adalah sebagai berikut:

1. Mengidentifikasi kebutuhan pengguna. Pengembangan mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.
2. Membuat satu *prototype*. Pengembang merancang *prototype* yang sesuai dengan kebutuhan pengguna.

3. Menentukan apakah *prototype* kepada para pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika ya, langkah 4 akan diambil, jika tidak, *prototype* direvisi dengan mengulang kebutuhan pengguna.
4. Menggunakan *prototype*. Sistem mulai dikembangkan dengan *prototype* yang sudah dibuat.

3.5 Kerangka Penelitian

Dalam penelitian yang dibahas dalam tugas akhir ini, dilakukan langkah-langkah atau tahapan dalam penelitian yang ada pada gambar III.1 sesuai dengan metodologi penelitian yang digunakan dalam penelitian tersebut. Tahapan-tahapan dalam penelitian tersebut adalah sebagai berikut:

1. Studi Kepustakaan
Melakukan studi pendahuluan terlebih dahulu dengan mencari dan menentukan judul penelitian serta mencari latar belakang masalah yang terjadi. Tujuan melakukan studi pendahuluan adalah untuk mendapatkan pengetahuan umum mengenai sistem apa yang sedang diteliti.
2. Pokok Permasalahan
Pada tahap ini dijelaskan pokok permasalahan penelitian yang dilakukan penulis, yaitu sistem absensi karyawan yang digunakan bersifat manual, penyimpanan laporan tidak terorganisir dengan baik serta perhitungan jam kehadiran kurang akurat karena dapat dimanipulasi dan menyebabkan tidak kesesuaian dengan jam kehadiran.
3. Batasan masalah
Pada tahap ini dijelaskan batasan masalah yang dilakukan penulis, yaitu melakukan penelitian dan analisis hanya mengenai proses absensi karyawan pada PT Rekadaya Multi Adiprima.
4. Tujuan penelitian
Pada tahap ini dijelaskan tujuan penelitian yang dilakukan penulis, yaitu untuk merancang proses absensi karyawan menjadi terkomputerisasi, membuat laporan pengolahan data serta menghindari *human error* dalam perhitungan laporan absensi karyawan.
5. Pengumpulan Data
Dalam mengumpulkan data, metode yang digunakan yaitu dengan metode wawancara, dan observasi (pengamatan). Metode wawancara dilakukan dengan kegiatan tanya jawab terhadap karyawan HRD *Department*, pertanyaan yang diajukan yaitu seputar prosedur penerimaan karyawan, proses absensi, dan. Pengamatan dilakukan dengan cara sebagai berikut:
 - Mempelajari dokumen organisasi, seperti profil perusahaan dan bagan struktur organisasi.
 - Mempelajari sistem absensi yang sedang berjalan pada perusahaan, termasuk mempelajari dokumen absensi seperti laporan absensi.

- Survei kegiatan proses absensi karyawan.

6. Merancang suatu *prototype*

Pada tahap ini yang dilakukan adalah merancang *prototype*. Pengamatan dilakukan dengan cara sebagai berikut:

- Pemodelan sistem yang menggunakan UML, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*.

- Pemodelan data menggunakan *Class Diagram* dan kamus data.

- Desain sistem menggunakan *flowchart* dan HIPO.

7. Membuat *Prototype*

Membuat sebuah *prototype* sesuai dengan analisis yang dilakukan. Setelah itu dilakukan perancangan *prototype*, lalu dilanjutkan dengan *prototype* pembuatan aplikasi menggunakan *PHP* dan *MySQL* sebagai basis data.

8. Menentukan apakah *prototype* dapat diterima

Pengembang mendemonstrasikan *prototype* kepada para pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika ya, langkah 3 akan diambil; jika tidak, *prototype* direvisi dengan mengulang kembali langkah 1, 2, dan 3 dengan pemahaman yang lebih baik mengenai kebutuhan pengguna.

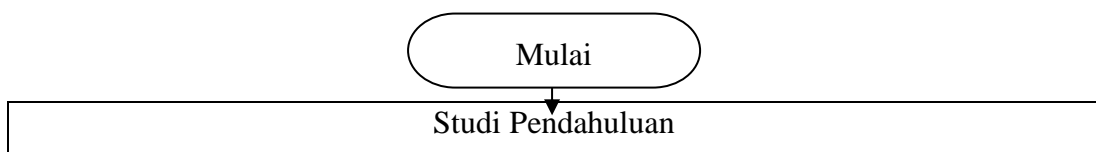
9. Menggunakan *prototype*

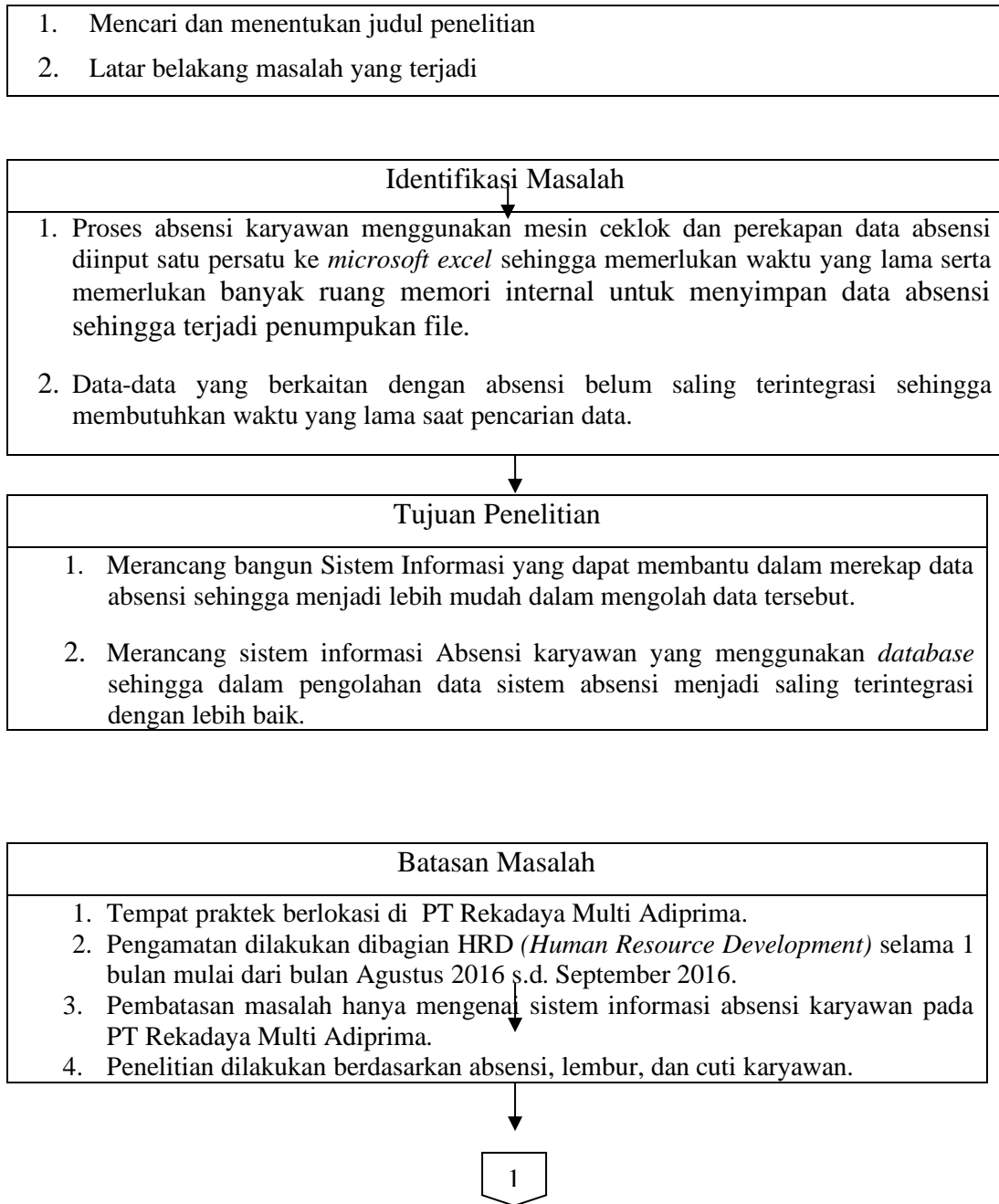
Pada tahap ini yang dilakukan pengumpulan dan menganalisis data kembali dan mencari tahu kebutuhan apa yang dibutuhkan oleh pengguna sistem, serta melakukan perancangan sistem ulang sesuai dengan kebutuhan pengguna.

10. Kesimpulan dan saran

Mengambil kesimpulan dari hasil analisis sistem berjalan, sistem yang diusulkan dan aplikasi yang dirancang serta memberikan saran.

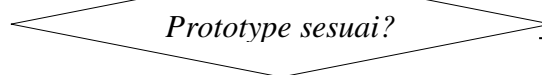
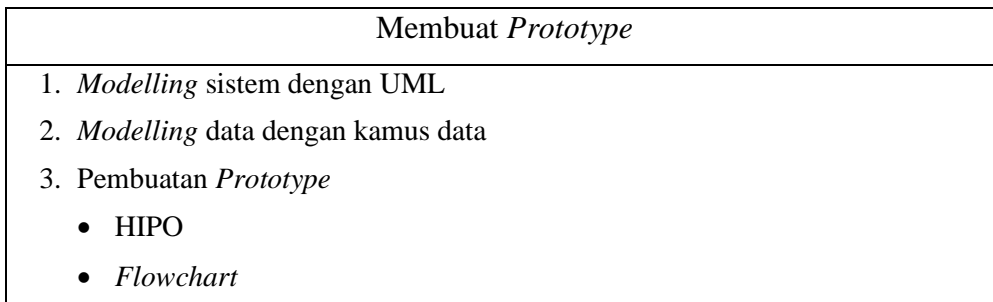
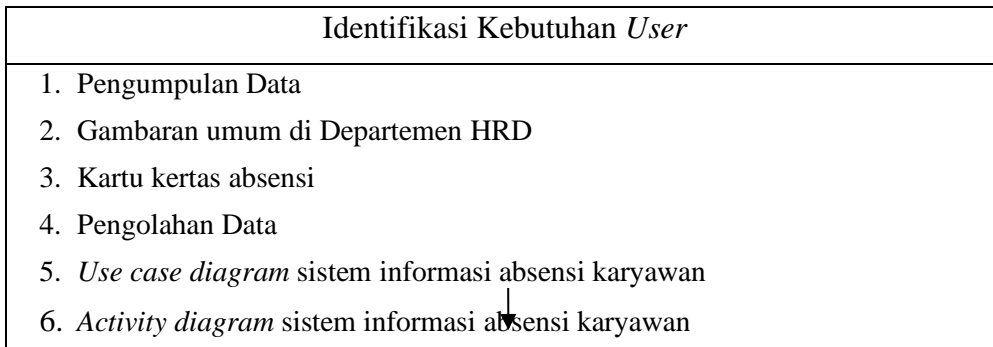
Berikut adalah *flowchart* kerangka penelitian untuk menyelesaikan masalah dalam Tugas Akhir ini:





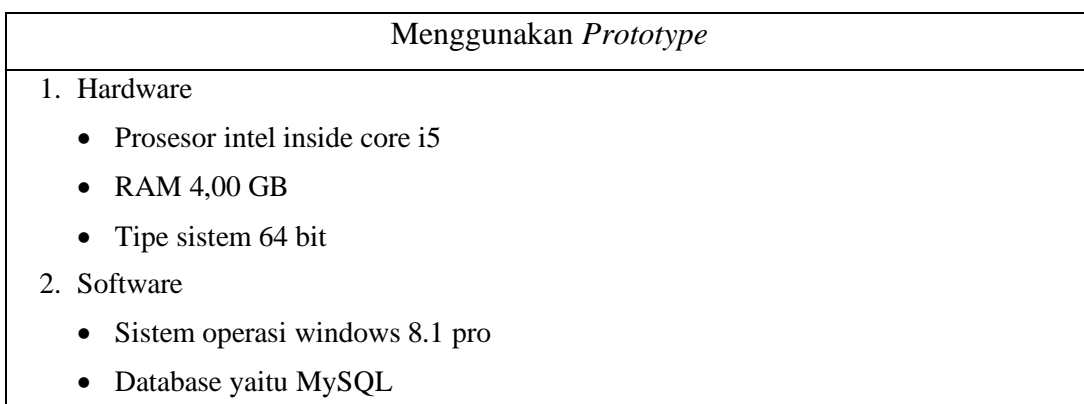
Gambar 3.1 Kerangka Penelitian

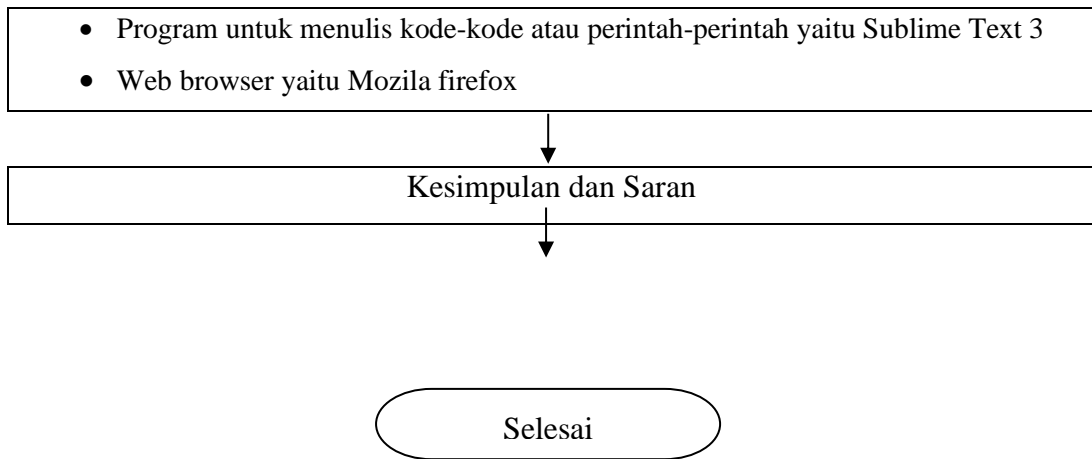
1



YA

TIDAK





Gambar 3.1 Kerangka Penelitian (Lanjutan)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Latar Belakang Perusahaan

PT Rekadaya Multi Adiprima merupakan salah satu perusahaan yang bergerak dalam industri manufaktur. Didirikan sejak tahun 1994 telah memproduksi komponen otomotif baik untuk kendaraan roda empat maupun roda dua yang merupakan produk orisinil atau *original equipment manufactured* (OEM). Kini PT RMA telah memiliki lokasi usaha yang tersebar di Cikeas Nagrak-Kabupaten Bogor sebanyak 9 plant yaitu 1 plant di Sumur kondang – Klari, Kabupaten Karawang, dan 1 plant di Cikarang – Kabupaten Bekasi.

Seiring dengan pertumbuhan bisnis otomotif di Indonesia juga terjadinya peningkatan permintaan di luar negeri (ekspor), berimbas kepada industri komponen otomotif tersebut, PT RMA berfokuskan pada *stamping parts*, *insulator dush panel*, dan *furniture parts* baik untuk kendaraan roda empat maupun roda dua.

Di Indonesia khususnya daerah Jawa Barat merupakan tempat dimana tersebar bagian dari Rekadaya Group. Adapun anggota Rekadaya Group adalah sebagai berikut:

1. PT Rekadaya Multi Adiprima (PT RMA)
2. PT Rekadaya Kreasi Indonesia (PT RKI)
3. PT Rekadaya Prima Indonesia (PT RPI)
4. PT Rekadaya Global Logistik (PT RGL)
5. PT Rekadaya Nonwoven Indonesia (PT RNI)

4.2 Makna Logo Perusahaan

Lingkaran oranye melambangkan PT Rekadaya Multi Adiprima sebagai perusahaan yang memiliki semangat dalam menjalani setiap kegiatan dan selalu mempunyai rasa optimisme. Hal ini berlaku untuk setiap aktor dalam perusahaan. Sedangkan tiga pilar hijau yang berada diatas lingkaran orange melambangkan perusahaan yang berusaha memulai dari bawah dan melewati gelombang kehidupan untuk menuju sukses.



Gambar IV.1 Logo PT Rekadaya Multi Adiprima

4.3 Profil Perusahaan

Nama perusahaan	: PT Rekadaya Multi Adiprima
Alamat kantor pusat	: Alt. Cibubur-Cileungsi, Ciangsana Raya, No.55 Nagrak-Cikeas, Gunung Putri, Bogor – Jawa Barat
Telepon	: 021 – 8232888
Fax	: 021 – 8231774
Email	: rma@rekadaya.com
Website	: www.rekadaya.com
Produksi	: Part Plastik Service Hole, Felt, Insulator Dush Panel
Status	: Perseroan Terbatas
Luas Tanah	: Plant 1 1000m ² Plant 2 1200m ² Plant 3 2000m ² Plant 4 4000m ²

Plant 5 900m²
Plant 6 4000m²
Plant 7 2700m²
Plant 8 1300 m²
Plant 9 3000m²
Cikarang 30000m²
Karawang 15800m²

4.4 Visi, Misi, dan Motto Perusahaan

PT Rekadaya Multi Adiprima perusahaan yang bergerak di bidang industri manufaktur komponen otomotif seperti Part Plastik *Service Hole, Felt, Insulator Dush Panel* serta komponen otomotif lainnya mempunyai visi dan misi sebagai berikut:

Visi:

Menjadi perusahaan manufaktur komponen otomotif dan industri pendukung terbesar di Indonesia.

Misi:

- Mewujudkan perusahaan yang maju berdasarkan Sistem Manajemen berstandar Internasional.
- Mewujudkan produk perusahaan yang bermutu dan berwawasan lingkungan, keselamatan dan kesehatan kerja untuk memenuhi kebutuhan pelanggan dalam hal *Quality, Cost, Delivery, and Moral*.

Motto:

Mutu yang baik dan sistem produksi adalah tanggung jawab kami.

4.5 Konsep Industri Perusahaan

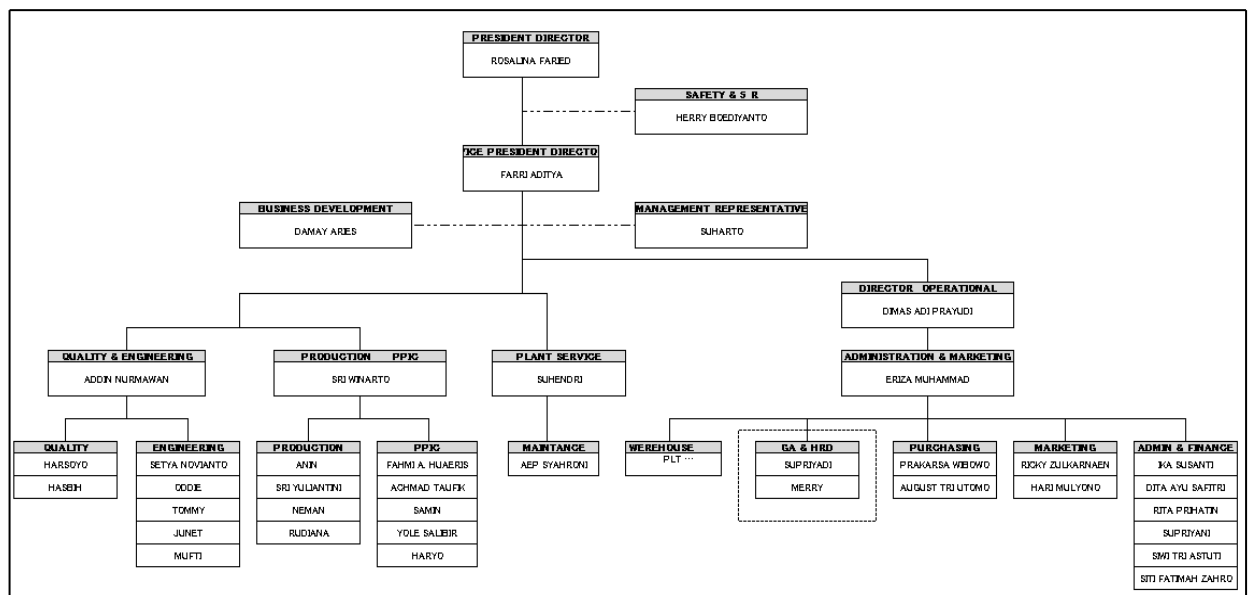
Konsep industri otomotif yang ketat dengan spesifikasi dan karakteristik, menjadikan PT RMA senantiasa menyesuaikan konsep ini

dengan sistem mutu formal ISO 9001 : 2008, ISO 14001 : 2004, OHSAS 18001 : 2007. Seiring dengan kepercayaan Pelanggan, PT RMA terus melakukan pengembangan usaha melalui inovasi teknologi dan penguatan sistem produksi.

4.6 Struktur Organisasi Perusahaan

Untuk menjalankan usahanya, setiap perusahaan memerlukan suatu struktur organisasi yang baik. Struktur organisasi dapat diartikan sebagai susunan dan hubungan antar bagian dan posisi dalam suatu perusahaan. Suatu struktur organisasi menggambarkan pembagian kerja, pelimpahan wewenang, kesatuan perintah dan tanggung jawab yang jelas.

Struktur organisasi yang tersusun dengan baik akan memudahkan koordinasi, integrasi, serta meningkatkan efektivitas dan efisiensi suatu perusahaan didalam mencapai tujuannya. Berikut merupakan struktur organisasi PT RMA secara keseluruhan:

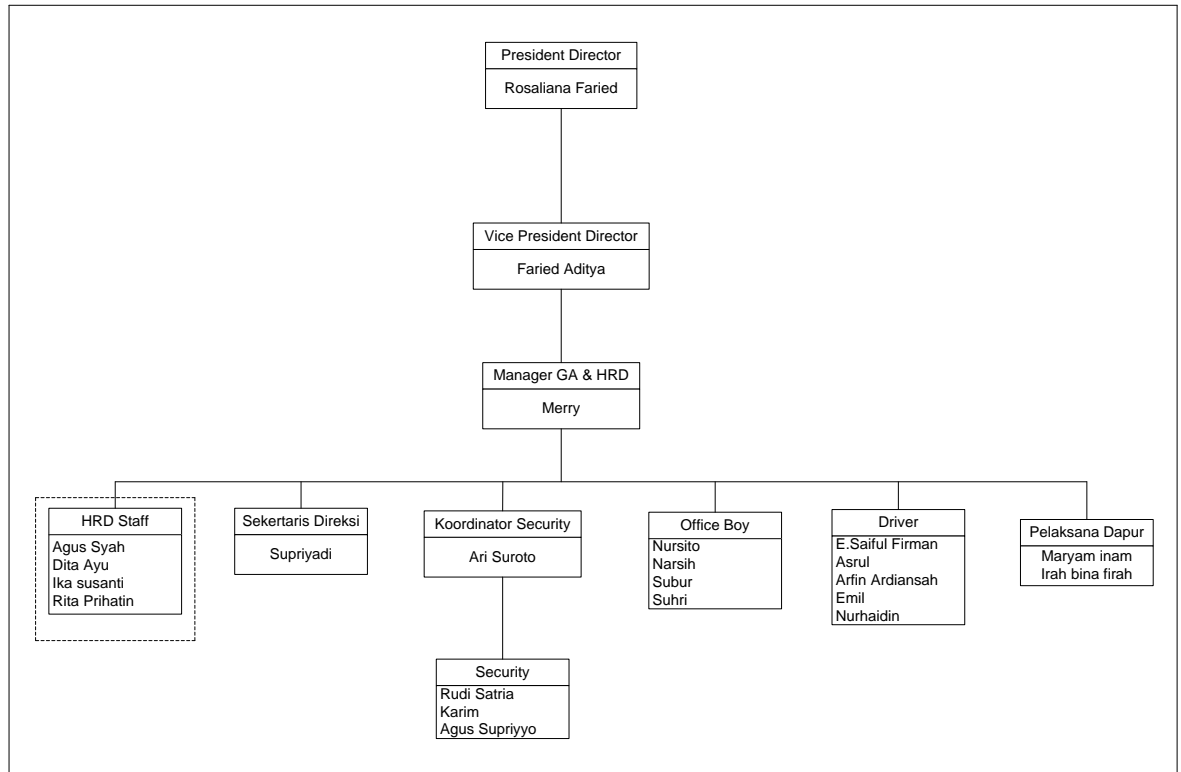


Ket: = Lokasi Penelitian

Gambar IV.2 Struktur Organisasi PT Rekadaya Multi Adiprima

(Sumber: PT Rekadaya Multi Adiprima, 2016)

Pada penelitian tugas akhir ini, mahasiswa memperoleh data-data mengenai hal absensi pada bagian HRD. Berikut adalah Struktur Organisasi Departemen HRD pada PT Rekadaya Multi Adiprima :



Ket: = Lokasi Penelitian

Gambar IV.3 Struktur Organisasi Departmen HRD

(Sumber: PT Rekadaya Multi Adiprima, 2016)

4.7 Tugas dan Wewenang setiap Jabatan di PT Rekadaya Multi Adiprima

Dengan adanya suatu organisasi yang baik, maka memungkinkan terselenggaranya suatu kelancaran dalam proses usaha yang dilakukan oleh perusahaan dikarenakan dalam organisasi tersebut sudah ditetapkan dengan jelas mengenai wewenang, tugas, dan tanggung jawab seseorang. Pada PT RMA dikenal dengan *Job Description*. Berikut ini merupakan *Job Description* dan wewenang kegiatan dari masing-masing jabatan yang ada di PT RMA adalah sebagai berikut.

1. *President Director*

Presiden Direktur bertindak sebagai pimpinan Eksekutif perusahaan dan secara keseluruhan mempunyai tanggung jawab strategi dan manajemen sehari-hari terhadap aktivitas perseroan. Presiden direktur secara mendasar menetapkan arah, tujuan, dan strategi untuk dikerjakan oleh semua unit operasi dan pada akhirnya bertanggung jawab untuk kinerja keuangan perseroan.

2. *Vice President Director*

Vice President Director bertindak sebagai wakil pimpinan perusahaan secara keseluruhan mempunyai tanggung jawab terhadap aktivitas perseroan antar divisi/departemen serta memeriksa laporan-laporan. Secara umum *vice president* ikut serta dalam menjalin kerjasama dengan berbagai perusahaan dan mengawasi kelancaran perusahaan sesuai dengan tujuan untuk kelancaran perusahaan.

3. *Safety & 5 R*

- Melakukan pemeriksaan peralatan kerja, tenaga kerja, kesehatan kerja serta lingkungan kerja
- Meninjau dan mengarahkan karyawan bekerja sesuai kewajiban dan sesuai dengan sistem operasi perusahaan
- Mampu melakukan penanggulangan kecelakaan kerja dan melakukan penyelidikan penyebabnya

4. *Business Development*

- Mengobservasi kinerja *project* yang selama ini telah berjalan pada perusahaan
- Menganalisa potensi untuk melakukan efisiensi dari berbagai lini
- Melihat peluang pengembangan bisnis yang memungkinkan dengan mengacu pada *core business* yang sudah ada

5. *Management Representative*

- Mempromosikan kesadaran tentang persyaratan pelanggan

- Mempersiapkan tinjauan management jadwal pertemuan dan melakukan rapat *management review*
- Membuat ISO atau kesadaran kualitas untuk rekan dengan pelatihan internal

6. *Director Operational*

- Mengelola dan meningkatkan efektivitas dan efisiensi operasi perusahaan
- Mengawasi persediaan, distribusi barang dan tata letak fasilitas operasional
- Meningkatkan sistem operasional, proses dan kebijakan dalam mendukung visi dan misi perusahaan

7. *Quality & Engineering*

- Membuat dan menetapkan prosedur yang berkaitan dengan engineering, seperti:
 - Membuat *design tooling* proses dan *design* alat bantu.
 - Membuat dan memonitor *schedule project project* sesuai dengan waktu yang telah disubmit kepada customer.
 - Mengontrol dan mendistribusikan dokumen-dokumen teknis seperti *Part Drawing, Techical Standart, Engineering Change Instruction, Estimasi Tooling, Mapping Process, Bill of Material*.
- Mengontrol dan memonitor pelaksanaan *preventive maintenance* terhadap *dies, jig jig* produksi.

8. *PPIC (Production Planning & Inventory Control)*

- Menjalankan kebijakan departemen yang tertuang pada sasaran mutu, seperti membuat *loading capacity*, membuat *budget material* dan *request material*, membuat *master plan schedule* dan *daily schedule* serta membuat *monthly schedule delivery*.
- Melakukan kontrol terhadap *inventory raw material*, produk WIP, dan *Finish Goods*.
- Melakukan koordinasi dengan departemen terkait untuk lancarnya pergerakan produksi, seperti dengan departemen :

- Departemen *Purchasing* dalam pengadaan material produksi
- Departemen *Quality* dalam hal kualitas material dari supplier
- Departemen Produksi dalam hal pencapaian hasil produksi

9. *Production*

- Melaksanakan kegiatan produksi sesuai dengan data produksi yang diberikan oleh bagian PPIC.
- Melaksanakan kegiatan produksi dengan menggunakan bahan baku dan mesin secara efisien dan produktif.
- Meneliti, menganalisa dan membuat rekomendasi perbaikan, perubahan atau penambahan sarana yang diperlukan untuk menanggulangi hambatan produksi, dan meningkatkan kinerja secara berkesinambungan.
- Melaksanakan kegiatan produksi setiap produk yang dihasilkan dengan memanfaatkan tenaga kerja yang ada serta memberikan arahan dan training untuk mencapai produktifitas kerja yang tinggi dengan tingkat resiko kecelakaan kerja yang serendah-rendahnya.

10. *Plant Service*

- Menjamin bahwa *break down* mesin atau mesin rusak pada saat produksi sesuai dengan target yang ditetapkan.
- Melakukan pengawasan terhadap pelaksanaan preventive maintenance terhadap mesin-mesin produksi secara berkala atau reguler.
- Melakukan pengawasan terhadap perawatan infrastruktur termasuk *building* dan fasilitasnya.

11. Admin

- Membuat dan menetapkan prosedur yang berkaitan dengan Administrasi seperti surat antara *customer* dan *supplier*
- Mengatur administrasi umum dan membuat jurnal setiap kegiatan
- Membuat surat jalan yang berhubungan dengan pengiriman, menerima dan merekap surat jalan *delivery*, surat tugas keluar, dan pengadaan alat tulis kantor.

12. *Marketing*

- Menjalankan kebijakan terhadap pengembangan Bisnis Perusahaan yang tertuang pada sasaran mutu.
- Menetapkan strategi dalam pencapaian suatu *project* atau *order* seperti meninjau dan menganalisa informasi *project* atau *order* yang diterima, membuat list *control feasibility study* dan membuat *summary* hasil dari *progress marketing*
- Mengkoordinasi, menganalisa, melaksanakan dan memantau tindakan perbaikan atau penanganan terhadap keluhan pelanggan yang diterima.

13. Warehouse

Divisi *Warehouse* terdiri dari tiga bagian yaitu bagian *Inventory*, bagian Logistik dan bagian Gudang Teknik. Tugas & Wewenang ketiga bagian pada divisi *warehouse* dijelaskan sebagai berikut:

- Bagian Inventori bahan baku berfungsi untuk mengatur dan melaksanakan penerimaan, penyimpanan, pengeluaran dan administrasi bahan baku.
- Bagian Inventori barang jadi berfungsi untuk mengatur dan melaksanakan penerimaan, penyimpanan, pengeluaran dan administrasi barang jadi.
- Bagian Gudang teknik berfungsi untuk menyediakan dan menjaga barang teknik.
- Menjaga kondisi bahan baku dan barang jadi terhadap kerusakan karena lingkungan penyimpanan dan kehilangan.

14. HRD (*Human Resources Division*)

Melaksanakan program Sumber Daya Manusia, termasuk pencarian tenaga kerja yang tepat, sistem upah dan tunjangan, pelatihan dan pengembangan karir, dan pengurusan masalah perizinan yang berkaitan dengan operasional usaha perusahaan.

15. Purchasing

- Membuat dan menetapkan prosedur yang berkaitan dengan Purchasing seperti:
 - Melakukan evaluasi terhadap *budget* yang diajukan oleh PPIC

- Melaksanakan pengadaan atau pembelian material berdasarkan *budget* material untuk kebutuhan operasional dan permintaan dari departemen lain yang sudah disetujui atasan.
- Mengkoordinir pelaksanaan audit vendor dan melakukan evaluasi dari hasil audit vendor.

16. *Finance*

- Membuat dan menetapkan prosedur yang berkaitan dengan *Finance*, seperti jurnal pembelian barang, mengontrol laporan posisi aktiva tetap, mengontrol laporan posisi biaya penyusutan aktiva tetap, baik laporan komersil atau laporan pajak.
- Mengendalikan transaksi pembelian, penjualan, dan *project* berdasarkan anggaran atau *budget* yang sudah disetujui (*approval*).

4.8 Fasilitas-Fasilitas Perusahaan

PT Rekadaya Multi Adiprima menyediakan fasilitas-fasilitas untuk menunjang kegiatan karyawan, perusahaan mempunyai tempat ibadah, koperasi dan gudang ATK (alat tulis kantor). Pelatihan dan konseling merupakan fasilitas perusahaan bagi karyawan sehingga masalah-masalah yang membebani karyawan bisa dapat dengan cepat diselesaikan. Training dan evaluasi bagi karyawan dilakukan setiap hari sabtu.

PT Rekadaya Multi Adiprima menyediakan fasilitas-fasilitas bagi semua karyawan. Setiap karyawan mendapatkan fasilitas-fasilitas diantaranya:

a. Tambahan Makan

Untuk mempertahankan kondisi pekerja tetap baik, perusahaan memberikan tambahan makan siang untuk pekerja Shift 1 pukul 11:45 dan makan malam untuk pekerja Shift 2 pukul 19:00.

b. Tunjangan

Tunjangan sosial yang diberikan berupa tunjangan kerja, cuti hamil, tunjangan kesehatan berupa BPJS, tunjangan hari tua, dll.

c. Seragam kerja bagi karyawan training maupun karyawan tetap.

Sistem penggajian di PT RMA adalah bahwa setiap karyawan memperoleh

gaji pokok berdasarkan Upah Minimum Kabupaten (UMK) dan jam lembur dihitung berdasarkan absensi.

4.9 Jumlah Karyawan dan Jam Kerja

Tabel di bawah ini menjelaskan jumlah karyawan secara keseluruhan yang dibedakan menurut tetap atau tidaknya karyawan tersebut dan juga jam kerja karyawan berdasarkan waktu kerja normal.

Tabel IV.1 Jumlah Karyawan

Karyawan	Jumlah
Karyawan Tetap	186 orang
Karyawan Kontrak	124 orang
Total	310 orang

(Sumber: PT Rekadaya Multi Adiprima, 2016)

Tabel IV.2 Waktu Kerja *Shift*

SHIFT	Hari	Waktu Kerja 1	Istirahat	Waktu Kerja 2
1	Senin - Kamis	07.30 – 11.45	11.45 – 12.45	12.45 – 16.00
	Jumat	07.30 – 11.30	11.30 – 13.00	13.00 – 16.30
	Sabtu	08.00 – 12.00	-	-
	Minggu	-	-	-
2	Senin - Kamis	21.00 – 03.00	03.00 – 04.00	04.00 – 06.00
	Jumat	21.00 – 03.00	03.00 – 04.00	04.00 – 06.00
	Sabtu	-	-	-
	Minggu	08.00 – 12.00	-	-

(Sumber: PT Rekadaya Multi Adiprima, 2016)

Pada PT. RMA terdapat perbedaan waktu kerja antara karyawan tetap dan karyawan kontrak pada karyawan kontrak berlaku shift satu dan dua seperti gambar diatas sedangkan, karyawan tetap hanya berlaku shift satu.

4.10 Divisi-Divisi PT Rekadaya Multi Adiprima

Pada PT Rekadaya Multi Adiprima terdapat beberapa divisi yang mempunyai proses produksi berbeda dalam menghasilkan sebuah produk, divisi-divisi yang ada pada PT Rekadaya Multi Adiprima adalah sebagai berikut:

1. *Metal Division*

Lokasi usaha *Plant-1* menempati areal seluas 1000 m² (sertifikat Hak Milik) merupakan bangunan pabrik pertama PT RMA dengan luas lantai (sebagian bertingkat) 1.280 m², dibangun tahun 2000. Lantai bawah dikhususkan untuk memproduksi dengan basis bahan metal/besi, lantai atas untuk kantor. Berikut ini merupakan lokasi dan hasil produksi metal untuk kendaraan motor dan mobil.

2. *Plastic Division*

Plant-2 menempati areal seluas 1.200 m² (sertifikat Hak Milik) merupakan pabrik ke dua PT RMA dengan luas lantai (sebagian bertingkat) 1.050 m², dibangun tahun 2002. Lantai bawah untuk memproduksi *Part Plastic Service Hole*, lantai atas untuk stock bahan baku.

3. *Interior Division*

Plant-3 menempati areal seluas 2.000 m² (sertifikat Hak Milik) merupakan pabrik ke tiga PT.RMA dengan luas lantai (bertingkat penuh) 1.400 m², dibangun tahun 2004. Lantai bawah untuk memproduksi *felt*, ruang makan karyawan, mushola, koperasi karyawan, dan pos keamanan, lantai atas untuk kantor staf dan direksi, ruang rapat, proses komponen interior, bahan baku, dan stok *finish good*.

4. *Non Woven Division*

Plant-4 menempati areal seluas 4.000 m² (sertifikat Hak Milik) merupakan pabrik ke empat PT.RMA dengan luas lantai (sebagian bertingkat) 2.600 m², dibangun tahun 2010. Pada tanggal 19 Maret 2013 diresmikan oleh bapak Gunawan Geniusahardja Presiden Komisaris Astra Mitra Ventura/Direktur ASTRA Internasional, bapak Sudirman MR-Presiden Direktur Astra Daihatsu Motor, bapak Hamdan Dzulkarnaen Direktur ASTRA Honda Motor, ibu Aretta Adrianri Presiden Direktur ASTRA Mitra Ventura. Lantai bawah untuk memproduksi *non woven felt*, ruang visualisasi, ruang makan karyawan, mushola, dan pos keamanan. Lantai atas untuk kantor direksi, ruang rapat *packaging* dan stok *finish good*.

Plant-7 menempati areal seluas 2.700 m² (sertifikat Hak Milik) merupakan pabrik ke tujuh PT.RMA dengan luas lantai (sebagian bertingkat) 1.500 m², dibangun tahun 2011. Lantai bawah untuk memproduksi *felt*, ruang makan karyawan, mushola, kantor, pos keamanan, dan parkir armada *delivery*, lantai atas untuk stok *finish good*.

5. *Printing Division*

Plant-5 menempati areal seluas 600 m² (sertifikat Hak Milik) merupakan pabrik ke lima PT.RMA dengan luas lantai (bertingkat 3 lantai) 900 m², dibangun tahun 2009. Lantai bawah untuk proses produksi *printing* dan *packaging*, lantai 2 untuk ruang produksi *printing*, mushola, lantai 3 untuk bahan baku dan *stock finish good*.

6. *Workshop*

Plant-6 menempati lahan seluas 4000 m² (sertifikat Hak Milik) merupakan pabrik ke enam PT.RMA dengan luas lantai (sebagian bertingkat) 1.400 m², dibangun tahun 2012. Lantai bawah untuk ruang produksi komponen *metal guide cam chain*, gudang, mess karyawan, mushola dan bengkel, lantai atas untuk mess karyawan.

7. *Vacuum Division*

Plant-9 menempati areal seluas 3.000 m² (sertifikat Hak Milik) merupakan pabrik ke-delapan PT.RMA dengan luas lantai 2.800 m², dibangun tahun 2012. Lantai bawah untuk memproduksi *needle*, *vacuum*, *injection*,

gudang, ruang maintenance, lantai 2 untuk kantor direksi dan administrasi, mess direksi, lantai 3 untuk gudang.

4.11 Produk yang Dihasilkan

PT Rekadaya Multi Adiprima menghasilkan berbagai macam jenis komponen otomotif. Plant 9C merupakan tempat di produksinya interior kendaraan roda empat dengan bahan baku hardboard dan MDF. Berikut ini merupakan produk yang dihasilkan di plant 9C :

1. *Board Assy Deck A*

Board Assy Deck A dibuat melalui proses *cutting* MDF, dilapisi dengan *felto*, *carpet D80N black* dan 4 buah *chusion* yang berbeda. *Board Assy Deck A* ini digunakan untuk komponen bawah pada bagasi kendaraan roda empat.



Gambar IV.4 *Board Assy Deck A*

(Sumber: PT Rekadaya Multi Adiprima, 2016)

2. *Board Assy Deck B*

Board Assy Deck B dibuat melalui proses *cutting* MDF, dilapisi dengan *felto*, *carpet D80N black* dan 1 buah *chusion*. *Board Assy Deck A* ini digunakan untuk komponen bawah pada bagasi kendaraan roda empat, dapat dilihat pada gambar IV.5.



Gambar IV.5 *Board Assy Deck B*

(Sumber: PT Rekadaya Multi Adiprima, 2016)

3. *Front Door RH (L300)*

Front Door RH (L300) dibuat melalui proses *cutting* MDF dan dilapisi dengan *PVC Door 290*. *Front Door RH (L300)* ini digunakan untuk komponen pintu depan (*front door*) kendaraan roda empat.



Gambar IV.6 *Front Door RH (L300)*

(Sumber: PT Rekadaya Multi Adiprima, 2016)

4. *Front Door LH (L300)*

Front Door LH (L300) dibuat melalui proses *cutting* MDF dan dilapisi dengan *PVC Door 290*. *Front Door LH (L300)* ini digunakan untuk komponen pintu depan (*front door*) kendaraan roda empat, terlihat pada gambar IV.7.



Gambar IV.7 *Front Door LH (L300)*

(Sumber: PT Rekadaya Multi Adiprima, 2016)

5. *PAD, S/A FA Door Trim*

PAD, S/A FA Door Trim terbuat dari bahan *felto*. Dibuat melalui proses *cutting*, lalu masuk ke proses *punching*. *PAD, S/A FA Door Trim* ini digunakan untuk melapisi komponen pintu kendaraan roda empat.



Gambar IV.8 *PAD, S/A FA Door Trim*

(Sumber: PT Rekadaya Multi Adiprima, 2016)

6. *Shield Fender Splash RH/LH*

Shield Fender Splash RH/LH terbuat dari bahan *plastic Polivinil Clorid (PVC)*. Pembuatan part *Shield Fender Splash RH/LH* menggunakan proses *vacum*. Fungsi dari part tersebut untuk pelindung bagian bawah dekat bagian roda kanan dan kiri pada roda empat, dapat dilihat pada gambar IV.9.



Gambar IV.9 *Shield Fender Splash RH/LH*

(Sumber: PT Rekadaya Multi Adiprima, 2016)

4.12 Jaringan Pemasaran

PT Rekadaya Multi Adiprima mendistribusikan produknya ke berbagai perusahaan besar didalam negeri. Berikut adalah nama-nama perusahaan yang menjadi konsumen PT Rekadaya Multi Adiprima :

- a. PT Astra Daihatsu Motor
- b. PT Honda Prospect Motor
- c. PT Kramayudha Tiga Berlian Motor
- d. PT Hino Motor Manufacturing Indonesia
- e. PT Sugity Creatives
- f. PT Toyota Boshoku Indonesia
- g. PT Inoac Polytekno Indonesia
- h. PT IRC Rubber Indonesia
- i. PT APM Armada Autopart
- j. PT Kotobukia Indo Classic Industries
- k. PT Dasa Windu Agung
- l. PT Nusa Keihin Indonesia
- m. PT Indomitra Sedaya
- n. PT Metindo
- o. PT Indo Karlo Perkasa
- p. PT All Try Industrial
- q. PT Sanko Gosei Technology Indonesia
- r. PT Jaya Victory Cemerlang

- s. PT Echo Advanced Technology Indonesia
- t. PT Usra Tampi Indonesia
- u. PT Howa Textile Indonesia
- v. PT Inti Polymetal
- w. PT 3M Indonesia
- x. PT Dharma Polymetal

4.13 Analisis Sistem Informasi Absensi di PT RMA

Analisis sistem informasi absensi merupakan proses menganalisa permasalahan yang terdapat pada PT RMA mulai dari menganalisis dokumen-dokumen yang terkait dalam proses absensi, serta proses absensi yang sedang berjalan di perusahaan.

4.13.1 Dokumen-Dokumen yang Terkait

Dokumen-dokumen yang terlibat dalam sistem informasi absensi karyawan yang berjalan pada PT. RMA adalah sebagai berikut:

1. Kartu Absensi Karyawan PT. RMA

Berikut adalah kartu absensi harian karyawan yang nantinya akan diolah oleh HRD yang berada di PT RMA :

No. _____ Nama Jawidun

BAGIAN IT-SDH

BULAN 07-2015

SAKIT	IZIN	ALPA	LAMBAT	LAIN ²

Tgl	Pagi		Siang		Lembur	
	Masuk	Keluar	Masuk	Keluar	Masuk	Keluar
16				5.02		
17	7.51					14.43
18	7.22	16.01				
19	7.22					20.01
20	7.22	16.05				
21	7.22	16.04				
22	7.22	16.32				
23	7.50					15.03
24						
25			20.54			
26			20.54	5.01		
27			20.52	5.01		
28			20.53	5.01		
29			20.53	5.00		
30				5.01		
31						

Gambar IV.10 Kartu Absensi Harian Karyawan
(Sumber: PT Rekadaya Multi Adiprima)

Berikut adalah keterangan dari kartu absensi PT RMA :

1. Tgl : Berisi tanggal harian dalam sebulan
2. Pagi : Berisi jam masuk dan keluar karyawan pada shift pertama
3. Siang : Berisi jam masuk dan keluar karyawan pada shift kedua
4. Lembur : Berisi jam masuk dan keluar lembur karyawan
5. Sakit : Berisi jumlah sakit dalam sebulan
6. Izin : Berisi jumlah izin dalam sebulan
7. Alpha : Berisi jumlah alpha dalam sebulan
8. Lambat : Berisi jumlah keterlambatan karyawan dalam sebulan
9. Lain : Berisi jumlah keterangan lain dalam sebulan

2. Form Lembur Karyawan

Form ini dimiliki oleh leader disetiap plan produksi untuk menentukan dan mengetahui karyawan yang akan lembur selanjutnya, form ini akan diserahkan ke bagian HRD untuk direkap kembali. Cara pengisian form yaitu leader yang menentukan siapa saja karyawan yang akan lembur dengan ketetapan waktu yang telah ditentukan, apabila karyawan lembur melebihi dari batas waktu yang telah ditentukan maka tetap dicatat dan setiap karyawan yang lembur harus tanda tangan selanjutnya, leader akan menghitung total waktu lembur setiap karyawan. Karyawan yang dapat melakukan lembur yaitu supervisor, leader produksi dan operator produksi. Berikut adalah form lembur karyawan yang nantinya akan diolah oleh HRD yang berada di PT RMA :

Tabel IV.3 Form Lembur karyawan

PT. REKADAYA MULTI ADIPRIMA Automotive Parts and Manufactures											
RENCANA KERJA LEMBUR											
TANGGAL :											
NO.	NAMA	WAKTU		TOTAL	URAIAN PEKERJAAN		PLANNING PRODUKSI	AKTUAL PRODUKSI	TANDA TANGAN		
		MA BUK	KELUAR		PROSES	NAMA PART			1	2	
1									1	2	
2											
3									3	4	
4											
5									6	8	
6											
7									7	8	
8											
9									9	10	
10											
11									11	12	
12											
13									13	14	
14											
15									15	18	
16											
17									17	18	
18											
					Ks. SECURITY		PRODUKSI	PPIC	HRD	MANAGING DIRECTOR	

(Sumber: PT Rekadaya Multi Adiprima)

BERITA ACARA

Menerangkan bahwa yang bertanda tangan dibawah ini :

Nama :

Jabatan :

Terhitung pada tanggal s/d Tidak bisa mengikuti aktivitas kerja seperti biasa dikarenakan.....

Semoga bapak / ibu HRD dapat memakluminya, Atas perhatiannya saya ucapkan terimakasih.

Disetujui,

Dibuat,

(.....)

(.....)


Gambar IV. 11 Surat Berita Acara untuk Cuti

(Sumber: PT Rekadaya Multi Adiprima)

6. Rekap Cuti Karyawan

Rekap cuti karyawan berfungsi untuk mengetahui berapa banyak cuti yang sudah diambil oleh karyawan dalam setahun.

Tabel IV. 6 Rekap Cuti Karyawan

 PT. REKADAYA MULTI ADIPRIMA Automotive Part Manufacturers				
CUTI KARYAWAN Tahun 2016				
Cuti Service Hole				
No.	Nama	LAMA CUTI	KETERANGAN	SISA CUTI
1	Neny Anina	4	Kel. Meninggal	8
2	Rahmi Winitari	6	istirahat	6
3	Rosid Abidin	7	pulang kampung	5
4	Ali Suryana	3	istirahat	9
5	Mira Julaiha	5	mengurus anak	7
6	Bimo Prasetya	3	istirahat	9
7	Ina Putri	3	urusan keluarga	9
8	Ayu sapitri	2	Mengurus pendaftaran sekolah	10
9	Dodo Suwiryo	4	urusan keluarga	8

(Sumber: PT Rekadaya Multi Adiprima)

7. Laporan Absensi Tahunan Karyawan PT. RMA

Laporan absensi karyawan ini berfungsi untuk mengetahui cuti, sakit dan izin karyawan PT RMA jumlah keseluruhan dalam kurun waktu tahunan di setiap bulannya. Selain rekap absensi karyawan yang dihitung secara manual, rekap absensi karyawan bulanan juga dapat dihitung menggunakan aplikasi *microsoft excel*.

Berikut adalah laporan absensi tahunan karyawan PT RMA:

Tabel IV.7 laporan absensi tahunan PT. RMA

ITEM	2015												TOTAL
	JAN	FEB	MAR	APR	MEI	JUN	JUL	AGS	SEP	OKT	NOV	DES	
IZIN	3	3	0	3	11	3	0	0	1	3	1	0	2
SAKIT	6	6	3	20	0	5	4	5	0	1	5	2	5
ALPHA	0	0	0	0	0	0	0	1	0	0	0	0	0
TOTAL ABSEN	9	9	3	23	11	8	4	6	1	4	6	2	1
TOT. MP	21	21	21	20	20	19	19	15	15	15	15	13	18
TOT. HK	23	22	24	23	22	23	23	23	23	23	23	23	23
M DAYS	483	462	504	460	440	437	437	345	345	345	345	299	409
IZIN	0.01	0.01	0.00	0.01	0.03	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.01
SAKIT	0.01	0.01	0.01	0.04	0.00	0.01	0.01	0.01	0.00	0.00	0.01	0.01	0.01
ALPHA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TOTAL ABSEN	1.86%	1.95%	0.60%	5.00%	2.50%	1.83%	0.92%	1.74%	0.29%	1.16%	1.74%	0.67%	0.15%
TARGET	2.0%	2.0%	2.0%	2.0%	2.0%	2.0%	2.0%	2.0%	2.0%	2.0%	2.0%	2.0%	2.0%
REMARK	X	X	X	X	X	X	o	X	o	X	X	o	

(Sumber: PT Rekadaya Multi Adiprima)

Berikut adalah keterangan data laporan absensi tahunan PT.RMA :

1. ITEM : Berisi Keterangan
 2. IZIN : Berisi jumlah izin karyawan setiap bulannya
 3. SAKIT : Berisi jumlah sakit karyawan setiap bulannya
 4. ALPHA : Berisi jumlah *alpha* karyawan setiap bulannya
 5. TOTAL ABSEN : Berisi total absensi sakit, izin, *alpha* setiap bulannya
 6. TOT.HK : Berisi total harian kerja setiap bulannya
 7. IZIN : Berisi *persentase* jumlah izin karyawan setiap bulannya
 8. SAKIT : Berisi jumlah sakit karyawan setiap bulannya dalam *persentase*
 9. ALPHA : Berisi jumlah *alpha* karyawan setiap bulannya dalam *persentase*
 10. TOTAL ABSEN : Berisi total absensi sakit, izin, *alpha* setiap bulannya dalam *persentase*
- 8. Data Karyawan PT RMA**

Berikut adalah data posisi karyawan di PT RMA :

Tabel IV.8 data karyawan PT. RMA

Tanggal : 28 Juni 2016								
NO	NAMA	TANGGAL MASUK	PDDK	BAG	JBTN	TTL	STTS KWN	ALAMAT
1	Bagus Suherman	21/05/2012	S1	ADM	GD	Jakarta 30 September 1984	K/01	Jl. Raya Hankam Gg. Kampus No.3 Rt. 003 Rw.005 Jatiranggon Jatisampurna Kota Bekasi
2	Dita Ayu	11/11/2010	SMA	ADMIN	FIN	Sukabumi, 13 April 1990	K/01	Komp. TNI AL Graha Jala Yudha Blok E15 RT.007/RW.020 Ciangsana Gunung Putri Bogor
3	Ika Susanti	10/01/2012	S1	ADM	FIN	Jakarta, 14 Februari 1979	K/00	Jl Cibur II Bulak Ringin RT 001/RW.003 Kel. Ciburur Kec. Ciracas Jakarta Timur
4	Ma'ani	02/03/2012	SMK	P4	SJ	Cirebon, 12 Mei 1993	TK	Jl. By Pass Pegagan blok utara RT. 002/RW.003 Kec Palimanan
5	Rita Prihatin	01/02/2012	SMA	ADM	TAX	Jakarta, 19 Januari 1972	K/02	Kp. Nagrak RT.002/RW.004 Ds Nagrak Kec. Gunung Putri Bogor
6	Sis Masitoh	21/11/2012	D3	ADM	SJ	Bogor, 09 Agustus 1989	K/00	Kp. Nagrak RT.004/ RW.003 Kel. Nagrak Kec. Gunung Putri Bogor

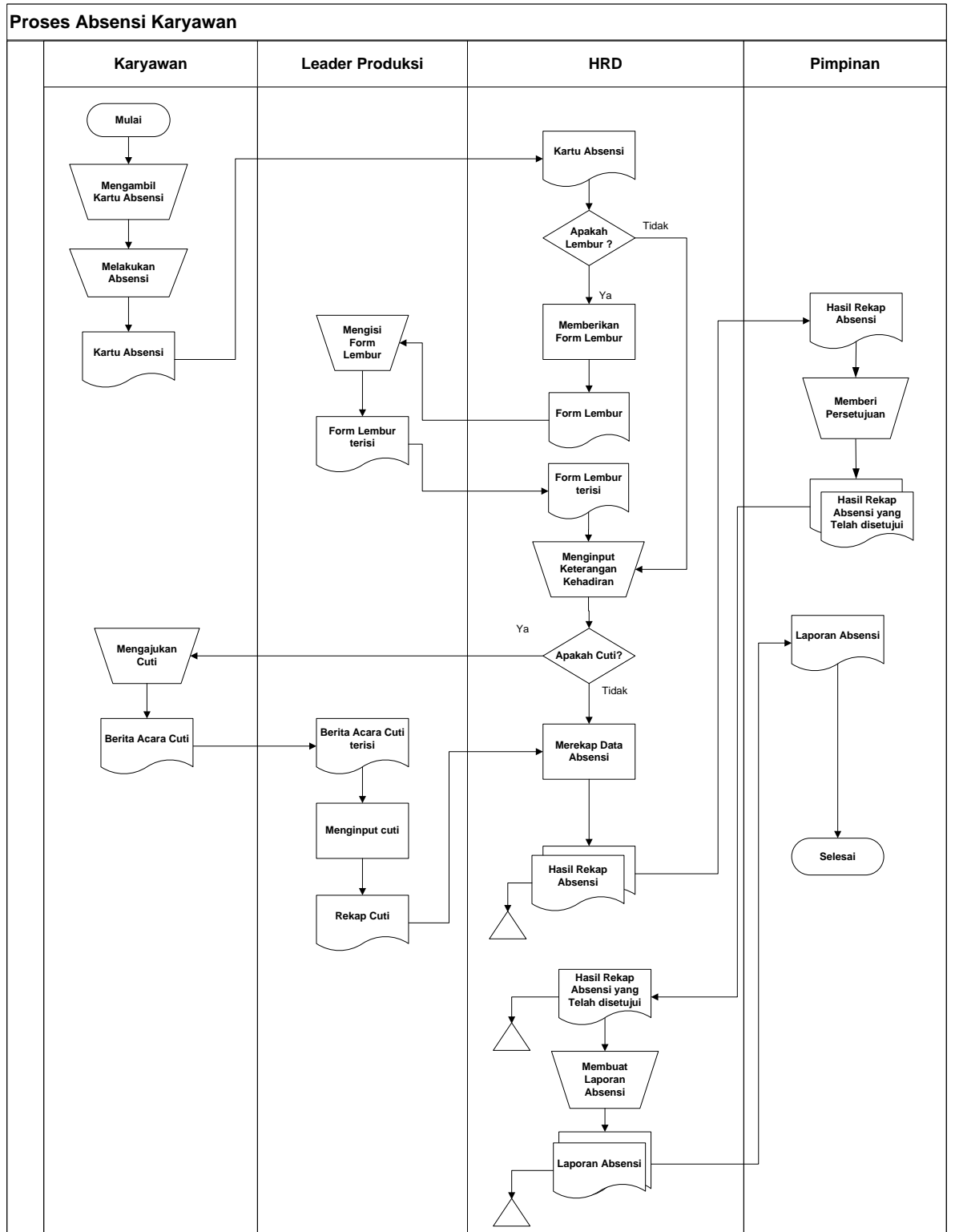
(Sumber: PT. Rekadaya Multi Adiprima)

4.13.2 Proses Absensi yang Sedang Berjalan di PT RMA

Proses absensi karyawan pada PT RMA dapat dilihat pada Gambar IV.13 yang menjelaskan proses absensi karyawan di PT RMA. Berikut adalah alur dari proses absensi karyawan:

1. Mengambil kartu absensi yang berada pada rak.
2. Karyawan melakukan kartu absensi menggunakan mesin ceklok.
3. Mesin ceklok absensi mencetak jam masuk kerja dan jam keluar kerja dari kartu absensi yang telah di masukkan ke dalam mesin tersebut.
4. Karyawan yang hadir dapat melakukan lembur maupun tidak.
5. Leader menentukan karyawan yang akan lembur lalu mengisi form lembur.
6. Bagian HRD mengolah absensi berdasarkan dari kartu absensi tersebut lalu menyeleksi karyawan yang lembur untuk menghitung jam lembur karyawan.
7. Jika karyawan cuti maka karyawan tersebut mengajukan cuti selanjutnya HRD menginputkan cuti karyawan.
8. Lalu setelah data absensi diolah data tersebut direkap oleh HRD.

9. Setelah bagian HRD selesai merekap data absensi dan lembur, data tersebut diserahkan ke pimpinan untuk disetujui.
10. Data rekap absensi karyawan yang telah disetujui di serahkan kembali kebagian HRD untuk diarsipkan.
11. Setiap bulannya bagian HRD membuat laporan absensi bulanan berdasarkan dari data rekap absensi dan lembur karyawan yang telah disetujui.
12. Laporan absensi dan lembur karyawan di simpan sebagai arsip pada HRD.
13. Pada setiap bulannya berdasarkan data rekap absensi dan lembur karyawan yang telah disetujui pimpinan, pada bagian HRD data tersebut diolah menjadi laporan absensi karyawan lalu diberikan kembali pada pimpinan. Berikut merupakan flowmap absensi karyawan PT RMA yang terdapat pada gambar IV.13

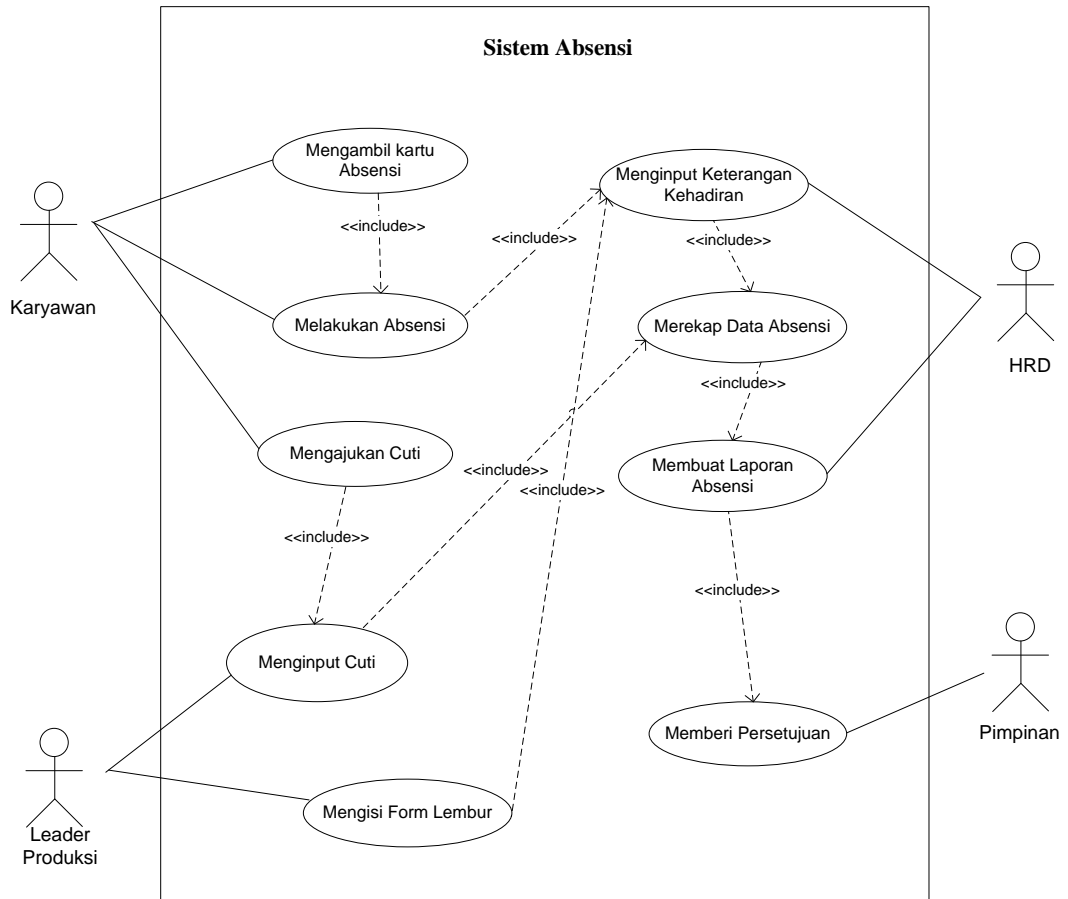


Gambar IV.12 Prosedur Absensi Karyawan

(Sumber: PT Rekadaya Multi Adiprima)

4.13.3 Penggambaran Sistem Absensi Karyawan dengan *Unified Modeling Language (UML)*

Berikut adalah *use case diagram* sistem absensi karyawan pada PT RMA yang sedang berjalan, dapat dilihat pada gambar IV.14.



Gambar IV.13 *Use Case Diagram* Sistem Informasi Absensi Karyawan

(Sumber: Hasil Analisis, 2016)

Penjelasan *use case diagram* sistem informasi absensi karyawan pada PT RMA diatas, dapat dijelaskan sebagai berikut:

1. Definisi Aktor

Pendefinisian aktor pada *use case* sistem informasi absensi karyawan pada PT RMA dapat dilihat pada Tabel IV.10.

Tabel IV.10 Definisi Aktor *Use Case Diagram* Sistem Informasi Absensi Karyawan

No.	Aktor	Deskripsi
1.	Karyawan	Karyawan PT RMA yang status kerjanya sebagai karyawan.
2.	HRD	Bagian yang mengerjakan dan mengolah data absensi karyawan serta pembuatan laporan absensi.
3.	Pimpinan	Orang yang menyetujui rekap absensi.
4.	Leader Produksi	Seseorang yang menentukan karyawan yang akan ditugaskan lembur serta menginputkan cuti.

(Sumber: Hasil Analisis, 2016)

2. Definisi *Use Case*

Pendefinisian *use case* pada *use case diagram* sistem informasi absensi karyawan pada PT RMA dapat dilihat pada tabel IV.11

Tabel IV.11 Definisi *Use Case* Sistem Informasi Absensi Karyawan

No.	<i>Use Case</i>	Deskripsi
1.	Mengambil Kartu Absensi	Karyawan Mengambil kartu absensi pada rak yang telah disediakan
2.	Melakukan absensi	Karyawan melakukan absensi dengan memasukkan kartu absensi setiap masuk dan pulang jam kerja.
3.	Mengajukan cuti	Karyawan mengajukan cuti untuk beberapa hari kedepan.
3.	Mengisi lembur	Leader produksi mengisi form lembur.
4.	Menginput cuti	Leader produksi menginputkan cuti karyawan .
5.	Menginput keterangan kehadiran	Proses pengolahan data absensi menginput keterangan kehadiran karyawan jika izin, sakit dan alpha.
6.	Merekap data absensi	Proses perekapan selama sebulan
7.	Memberi persetujuan	Proses persetujuan data absensi oleh pimpinan perusahaan
8.	Membuat laporan absensi	Proses pembuatan laporan absensi berdasarkan data absensi.

(Sumber: Hasil Analisis, 2016)

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mendapatkan informasi kebutuhan sistem, dengan mengetahui kebutuhan sistem yang diharapkan dapat diusulkan mengenai perbaikan terhadap sistem yang diamati. Berdasarkan hasil pengolahan data, dibutuhkan perancangan aplikasi untuk menunjang kegiatan absensi karyawan pada PT. RMA. Perancangan aplikasi yang memudahkan *user* dalam menggunakan aplikasi ini.

Kebutuhan program aplikasi dapat dijelaskan melalui proses-proses yang dibutuhkan pemakai dan kebutuhan sistem berdasarkan usulan solusi yang dibuat. Identifikasi kebutuhan sistem komputerisasi menjelaskan proses-proses yang dilakukan oleh program aplikasi, dimana proses-proses tersebut dibutuhkan oleh pemakai. Berikut adalah penjelasan identifikasi kebutuhan pemakai aplikasi absensi karyawan:

Tabel V.1 Identifikasi Kebutuhan Sistem

Kebutuhan Sistem	
Nama Proyek	Aplikasi Absensi Karyawan
Proses Sistem Berjalan	1. Terwujudnya sebuah sistem yang dapat membantu proses absensi karyawan.
Data Masukan (<i>input</i>)	1. Data Karyawan 2. Data Jabatan 3. Data <i>Shift</i> 4. Data Cuti
Data Keluaran (<i>output</i>)	1. Laporan absensi 2. Laporan lembur

(Sumber: Hasil Analisis Data, 2017)

Tabel V.1 Identifikasi Kebutuhan Sistem (Lanjutan)

Kebutuhan Sistem

Model Sistem Informasi	1. <i>Input</i> : Data <i>Master</i> Proses : Pengolahan data transaksi <i>Output</i> : Laporan absensi, lembur dan cuti
------------------------	--

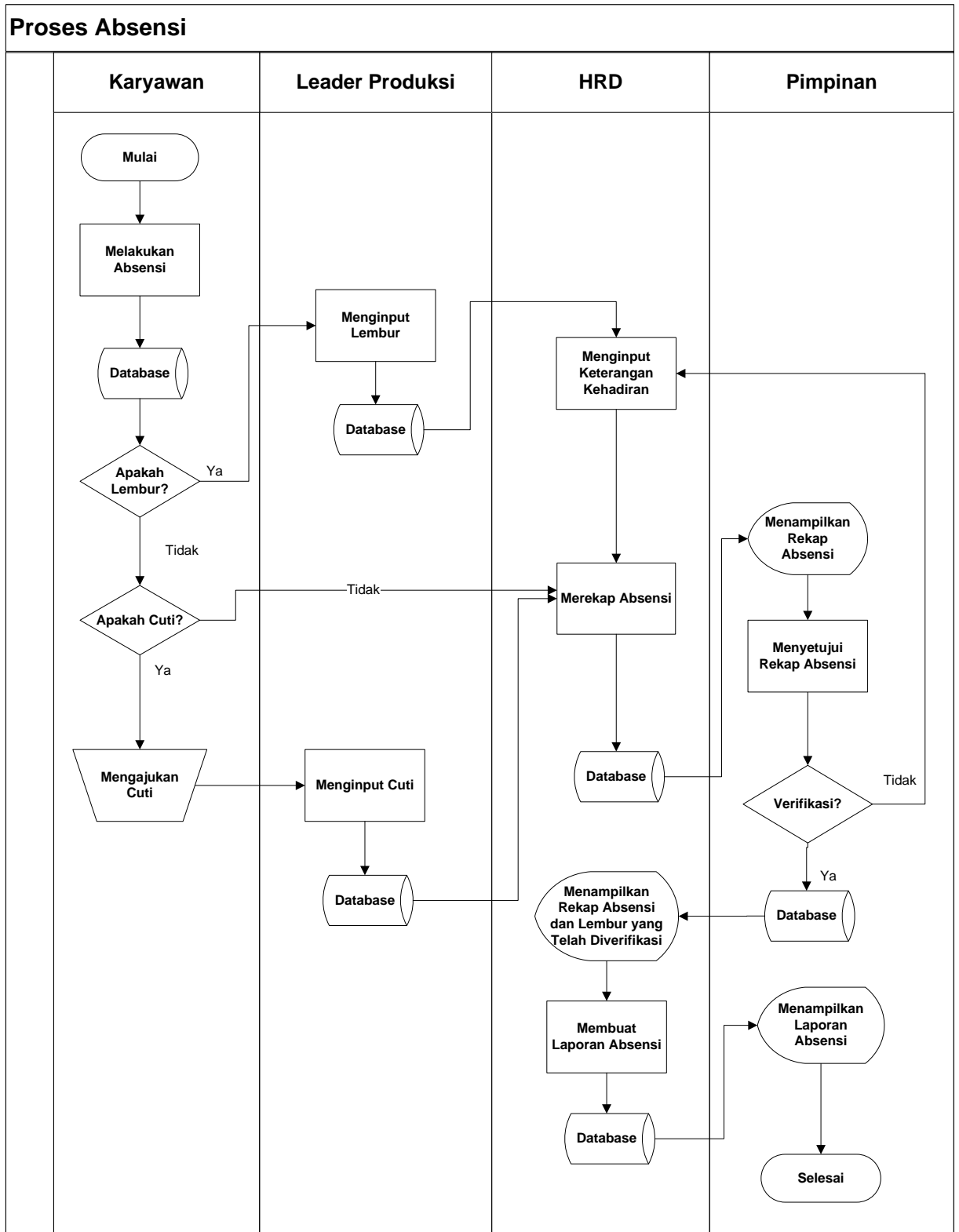
(Sumber: Hasil Analisis Data, 2017)

5.2 Perancangan *Flowmap* Sistem Usulan

Proses absensi yang diusulkan dengan karyawan yang melakukan *input* absensi pada sistem, kemudian sistem akan menyimpan data absensi tersebut ke dalam sebuah basis data absensi dan gaji. Pengolahan data absensi dilakukan oleh HRD (*Human Resource Department*), dimana data absensi dan apabila data perlu di *update* maka HRD akan mengelola data absensi dan gaji tersebut. Hasil akhir dari pengolahan ini dengan dibuatnya suatu laporan absensi yang akan diserahkan kepada *Manager Department*. *Flowmap* Proses Absensi Karyawan Usulan dapat dilihat pada Gambar V.1.

5.3 Analisis Sistem Usulan

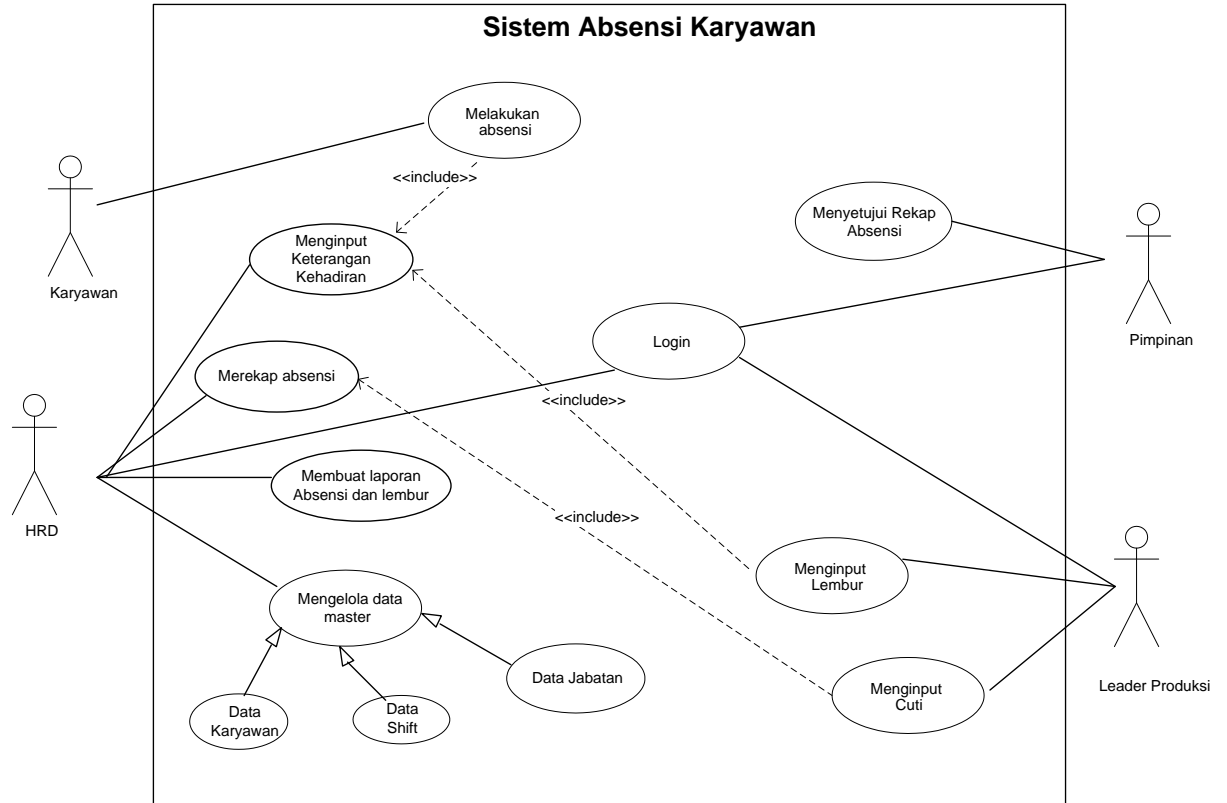
Perancangan aplikasi yang diusulkan menggunakan model pengembangan perangkat lunak *evolutionary prototype*. Tahap ini adalah tahap kedua pada metodologi prototipe evolusioner yaitu membuat prototipe untuk model sistem setelah sebelumnya melakukan tahap mengidentifikasi kebutuhan. Analisis proses sistem informasi absensi karyawan *outsourcing* menggunakan *tools* pemodelan sistem UML (*unified modeling language*) yang meliputi pembuatan *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *deployment diagram* dan juga pembuatan kamus data. Tahapan analisis ini akan memberikan gambaran mengenai aliran informasi dan data pada sistem informasi yang akan dibangun. Selanjutnya akan dilakukan perancangan sistem informasi absensi karyawan berdasarkan hasil analisis.



Gambar V.1 *Flowmap* Proses Absensi Karyawan Usulan
 (Sumber: Hasil Analisis Data, 2017)

5.3.1 Perancangan Use Case Diagram Aplikasi Usulan

Use case diagram digunakan untuk menjelaskan interaksi antara actor dengan sistem informasi yang akan dibuat. Berdasarkan analisis yang telah dilakukan, maka use case diagram sistem informasi absensi karyawan yang diusulkan dapat dilihat pada gambar V.2 sebagai berikut ini:



Gambar V.2 Use Case Diagram Sistem Usulan
(Sumber: Hasil Analisis Data, 2017)

Penjelasan use case diagram perancangan sistem informasi absensi karyawan di atas, dapat dijelaskan sebagai berikut:

1. Definisi Aktor

Pendefinisian aktor pada use case perancangan sistem informasi absensi karyawan usulan dapat dilihat pada tabel V.2.

Tabel V.2 Definisi Aktor Use Case Diagram Sistem Usulan

No.	Aktor	Deskripsi
1.	HRD	karyawan yang bertanggung jawab sebagai <i>admin</i> untuk menjalankan proses absensi secara

- keseluruhan. Dalam hal ini hak akses yang diberikan yaitu melakukan pengelolaan data *master*, transaksi data dan mencetak laporan absensi.
2. Pimpinan Pimpinan menyetujui rekap absensi yang telah diberikan dari HRD.
 3. Karyawan Karyawan melakukan proses absensi sehari-hari selama sebulan yang dicatat jam masuk dan jam pulang kerja.
 4. Leader Produksi Karyawan yang menentukan karyawan yang akan lembur.

(Sumber: Hasil Analisis Data, 2017)

2. Definisi dari *use case diagram* usulan pada PT Rekadaya Multi Adiprima adalah sebagai berikut:

Tabel V.3 Use Case Description Login

No.	Nama Use Case	Mengelola Data Login
1.	Deskripsi Use Case	Sistem ini memungkinkan aktor mengakses sistem aplikasi absensi karyawan.
2.	Actor	HRD
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. Memasukan <i>username</i> 2. Memasukan NIP 3. Melakukan identifikasi

(Sumber: Hasil Analisis Data, 2017)

Tabel V.4 Use Case Description Melakukan Absensi

No.	Nama Use Case	Melakukan Absensi
1.	Deskripsi Use Case	<i>Use case</i> ini menggambarkan proses absensi karyawan
2.	Aktor	Karyawan
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. Karyawan mengaktifkan sistem. 2. Karyawan memasukkan NIP untuk absensi.

Tabel V.5 Use Case Description Mengelola Data Karyawan

No.	Nama Use Case	Mengelola Data Karyawan
1.	Deskripsi Use Case	<i>Use case</i> ini menggambarkan proses pengelolaan data karyawan yang terdiri dari data lembur, absensi karyawan, jadwal <i>shift</i> , data rekap absensi, nama jabatan yaitu menambah data karyawan, mengubah data karyawan, mencari data karyawan dan menghapus data <i>master</i> .
2.	Aktor	HRD

3.	<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> 1. HRD masuk ke tampilan menu utama sistem. 2. HRD memilih menu data <i>master</i>. 3. HRD memilih submenu data karyawan. 4. HRD melakukan proses tambah, ubah, cari dan hapus data karyawan ke <i>database</i>.
----	----------------------------	---

(Sumber: Hasil Analisis Data, 2017)

Tabel V.6 Use Case Description Mengelola Data Jabatan

No.	Nama Use Case	Mengelola Data Jabatan
1.	Deskripsi Use Case	Use case ini menggambarkan proses pengolahan jabatan.
2.	Aktor	HRD
3.	<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> 1. HRD masuk ke tampilan menu utama sistem. 2. HRD memilih menu data <i>master</i>. 3. HRD memilih submenu data Jabatan. 4. HRD melakukan proses tambah, ubah, cari dan hapus data bagian ke <i>database</i>.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.7 Use Case Description Mengelola Data Shift

No.	Nama Use Case	Mengelola Data <i>Shift</i>
1.	Deskripsi Use Case	Use case ini menggambarkan proses pengolahan data <i>shift</i> karyawan.
2.	Aktor	HRD
3.	<i>Normal Flow Events:</i>	<ol style="list-style-type: none"> 1. HRD masuk ke tampilan menu utama sistem. 2. HRD memilih menu data <i>master</i>. 3. HRD memilih submenu data <i>shift</i>. 4. HRD melakukan proses tambah, ubah, cari dan hapus data <i>shift</i> karyawan ke <i>database</i>.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.8 Use Case Description Menginput Lembur

No.	Nama Use Case	Menginput Lembur
1.	Deskripsi Use Case	Use case ini menggambarkan proses penginputan data lembur karyawan.

2.	Aktor	Leader Produksi
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. Leader Produksi masuk ke tampilan menu utama sistem. 2. Leader Produksi memilih menu data <i>master</i>. 3. Leader Produksi memilih submenu data lembur. 4. Leader Produksi melakukan proses tambah, cari dan hapus data lembur ke <i>database</i>.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.9 Use Case Description Menginput Cuti

No.	Nama Use Case	Menginput cuti
1.	Deskripsi Use Case	Use case ini menggambarkan proses penginputan data cuti karyawan.
2.	Aktor	Leader Produksi
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. HRD masuk ke tampilan menu utama sistem. 2. HRD memilih menu data <i>master</i>. 3. HRD memilih submenu data cuti. 4. HRD melakukan proses tambah, cari dan hapus data cuti ke <i>database</i>.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.10 Use Case Description Merekap Absensi

No.	Nama Use Case	Merekap Absensi
1.	Deskripsi Use Case	Use case ini menggambarkan proses perekapan data absensi dan lembur karyawan serta cuti.
2.	Aktor	HRD
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. HRD masuk ke tampilan menu utama sistem. 2. HRD memilih menu data Transaksi. 3. HRD memilih submenu rekap absensi dan rekap lembur. 4. HRD melakukan proses filter bulan tahun pada data absensi dan lembur ke <i>database</i>.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.11 Use Case Description Menginput keterangan kehadiran

No.	Nama Use Case	Menginput keterangan kehadiran
1.	Deskripsi Use Case	Use case ini menggambarkan proses perekapan data cuti karyawan.
2.	Aktor	HRD
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. HRD masuk ke tampilan menu utama sistem. 2. HRD memilih menu data Transaksi. 3. HRD memilih submenu rekap cuti. 4. HRD melakukan proses filter bulan tahun pada data cuti ke <i>database</i>.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.12 Use Case Description Membuat Laporan Absensi

No.	Nama Use Case	Membuat Laporan Absensi dan Lembur
1.	Deskripsi Use Case	Use case ini menggambarkan proses pembuatan laporan absensi karyawan.
2.	Aktor	HRD
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. HRD masuk ke tampilan menu utama sistem. 2. HRD memilih menu Laporan. 3. HRD memilih submenu Absensi. 4. HRD melakukan filter bulan dan tahun lalu cetak laporan absensi dan lembur karyawan ke <i>database</i>.

(Sumber: Hasil Analisis Data, 2017)

Tabel V.13 Use Case Description Menyetujui Rekap Absensi

No.	Nama Use Case	Menyetujui Rekap Absensi
1.	Deskripsi Use Case	Use case ini menggambarkan proses melihat rekap absensi selama satu bulan dan pimpinan melakukan memvalidasi rekap.
2.	Aktor	Pimpinan
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. Pimpinan masuk ke tampilan menu utama sistem. 2. Pimpinan memilih menu data transaksi. 3. Pimpinan memilih submenu rekap absensi dan lembur. 4. Pimpinan melakukan proses persetujuan terhadap

		absensi dan lembur satu bulan.
--	--	--------------------------------

(Sumber: Hasil Analisis Data, 2017)

Tabel V.14 Use Case Description View Rekap Absensi Satu Bulan

No.	Nama Use Case	View Laporan Absensi Satu Bulan
1.	Deskripsi Use Case	Use case ini menggambarkan proses melihat laporan absensi selama satu bulan karyawan.
2.	Aktor	Pimpinan
3.	Normal Flow Events:	<ol style="list-style-type: none"> 1. Pimpinan masuk ke tampilan menu utama sistem. 2. Pimpinan memilih menu data transaksi. 3. Pimpinan memilih submenu absensi. 4. Pimpinan melakukan proses pengecekan laporan absensi karyawan.

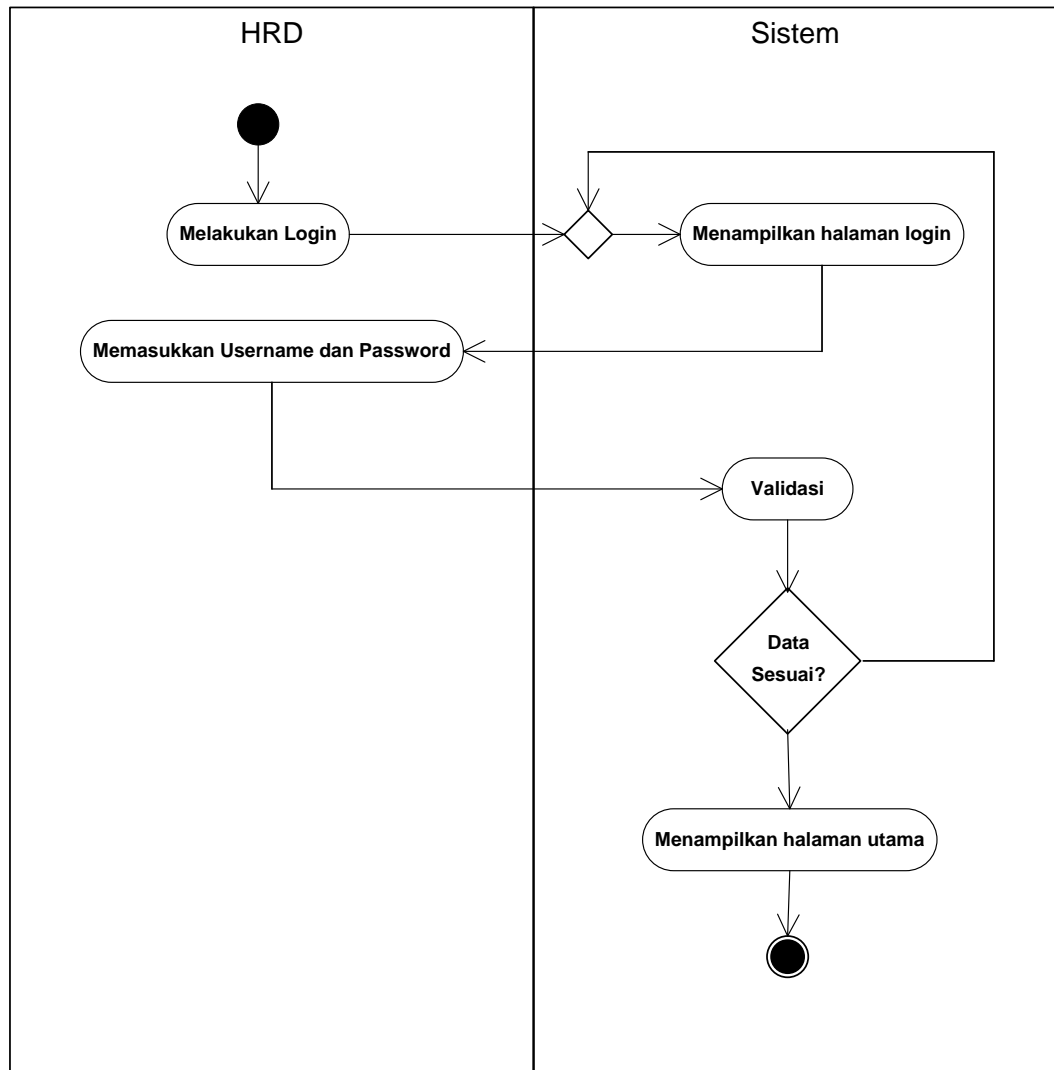
(Sumber: Hasil Analisis Data, 2017)

5.3.2 Activity Diagram

Activity diagram digunakan untuk menggambarkan aliran kerja tiap *use case* pada sistem informasi absensi karyawan. Berikut adalah *activity diagram* tiap *use case*:

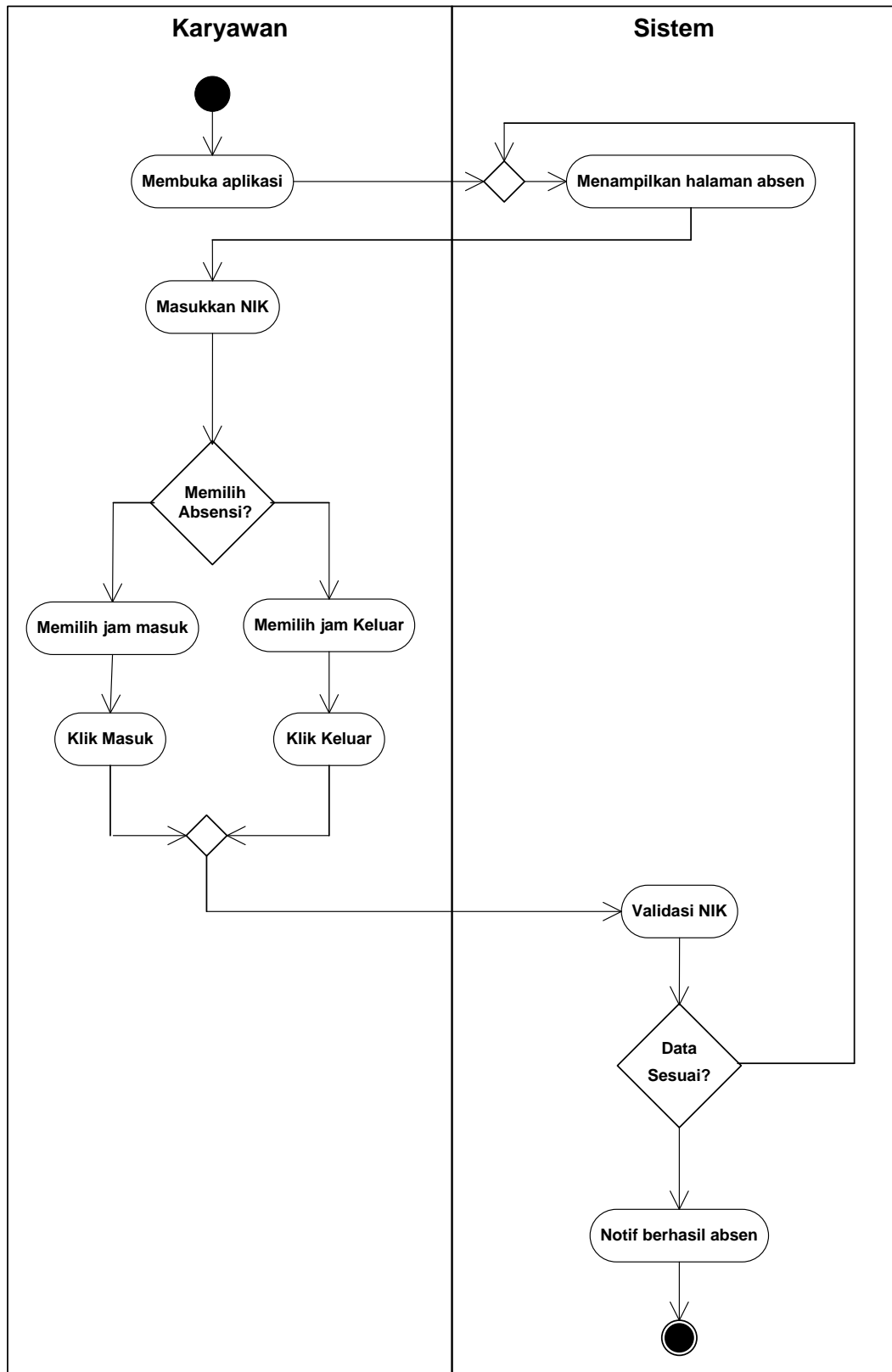
1. *Activity Diagram Proses Login*

Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika melakukan proses *login*. Dimana *user* memasukkan *username* dan *password* untuk dapat masuk ke dalam sistem. Jika *username* dan *password* tidak sesuai maka tidak dapat masuk ke dalam sistem. *Activity diagram* proses *login* dapat dilihat pada Gambar V.3.



Gambar V.3 Activity Diagram Proses Login
(Sumber: Hasil Analisis Data, 2017)

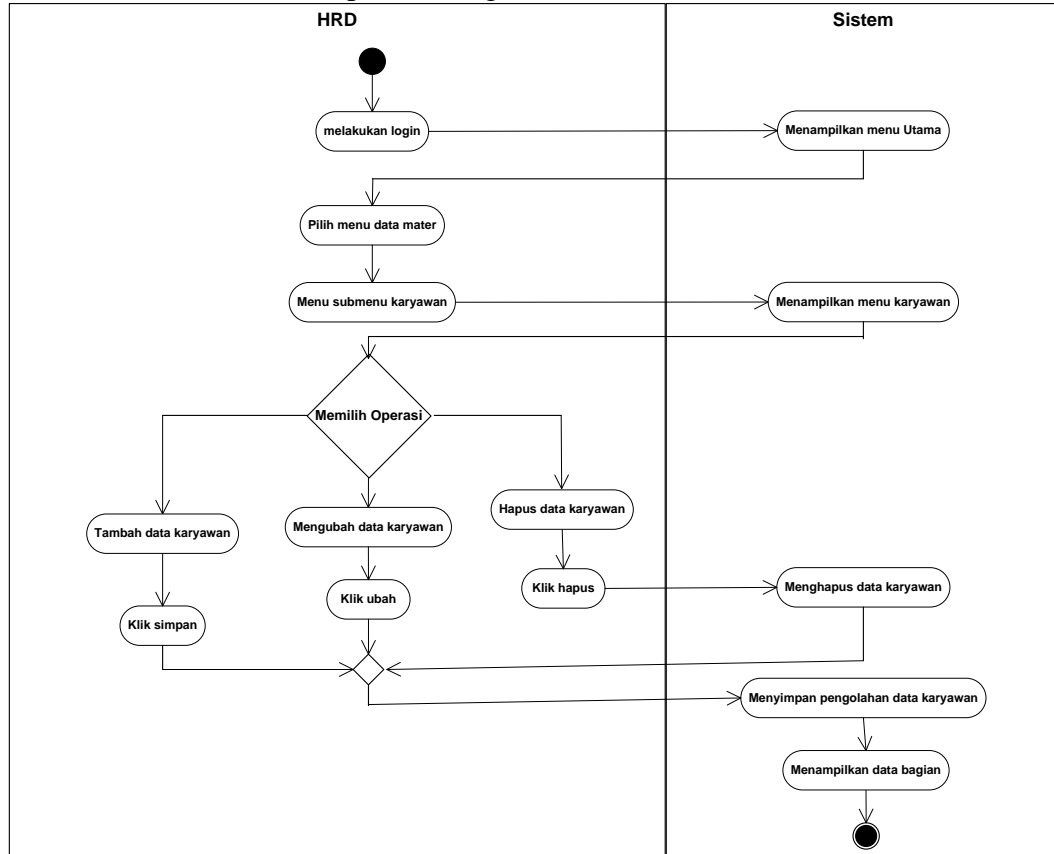
2. *Activity Diagram* Melakukan Absensi
Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu absensi untuk melakukan pencatatan kehadiran. Data absensi online masuk ke *database* yang dikelola oleh HRD. Sistem menampilkan hasil input NIK karyawan. *Activity diagram* absensi dapat dilihat pada gambar V.4.



Gambar V.4 Activity Diagram Melakukan Untuk Absen
(Sumber: Hasil Analisis Data, 2017)

3. *Activity Diagram* Mengelola Data Karyawan

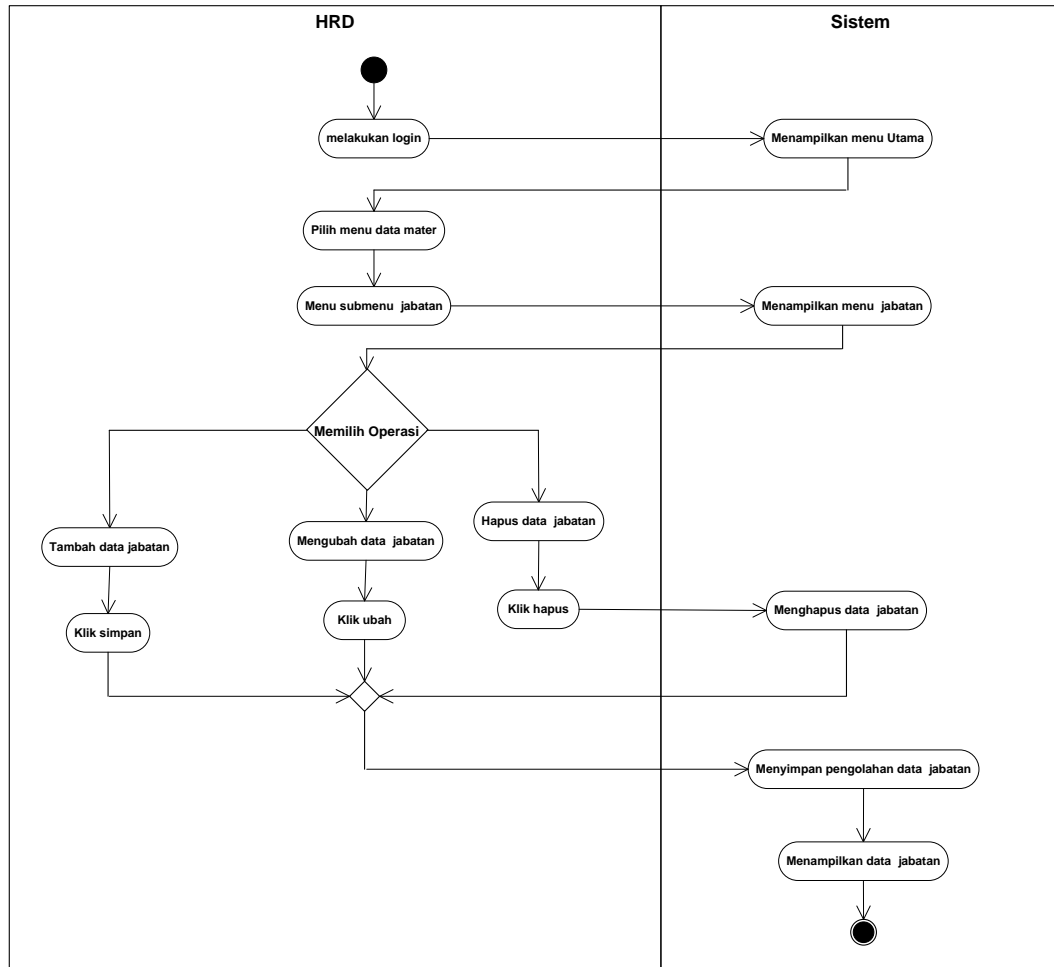
Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu data *master*. Data karyawan dikelola oleh HRD. HRD dapat menggunakan operasi tambah, cari, ubah dan hapus data karyawan masuk ke *database* dan dapat dilihat gambar V.5.



Gambar V.5 *Activity Diagram* Mengelola Data Karyawan
(Sumber: Hasil Analisis Data, 2017)

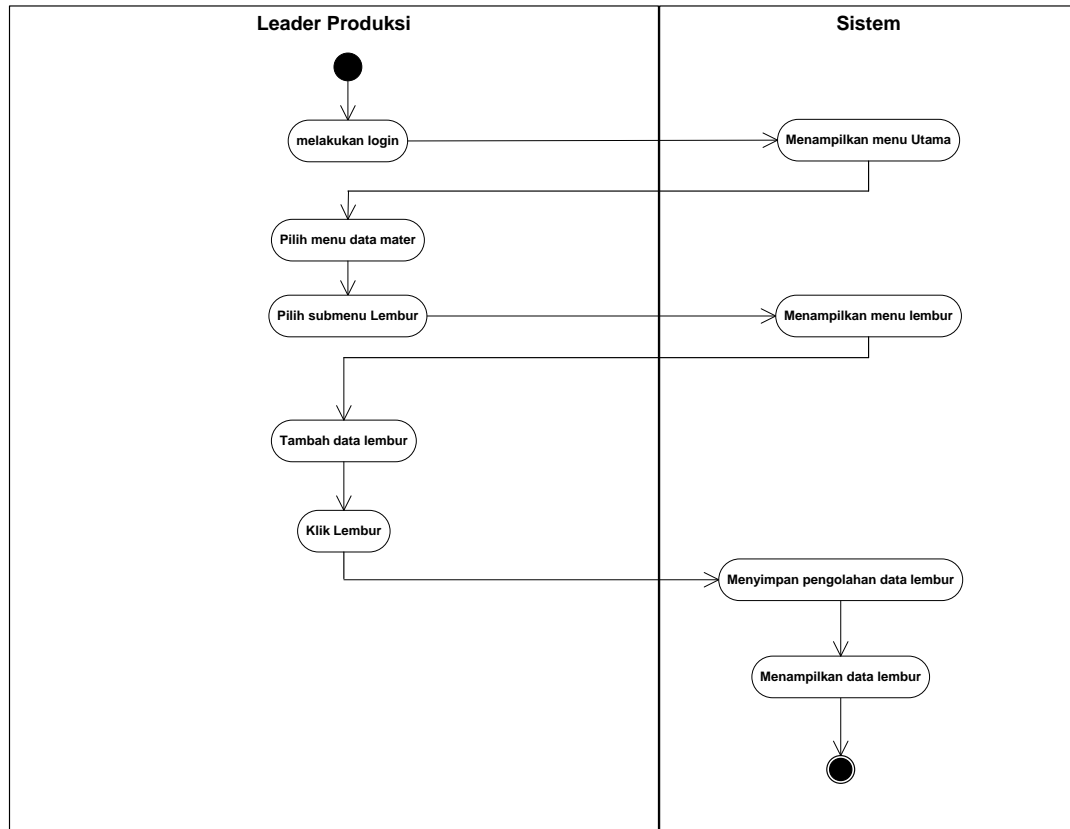
4. *Activity Diagram* Mengelola Data Jabatan

Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu data *master*. Data karyawan dikelola oleh HRD. HRD dapat menggunakan operasi tambah, cari, ubah dan hapus data jabatan masuk ke *database*. *Activity diagram* data *master* dapat dilihat gambar V.6.



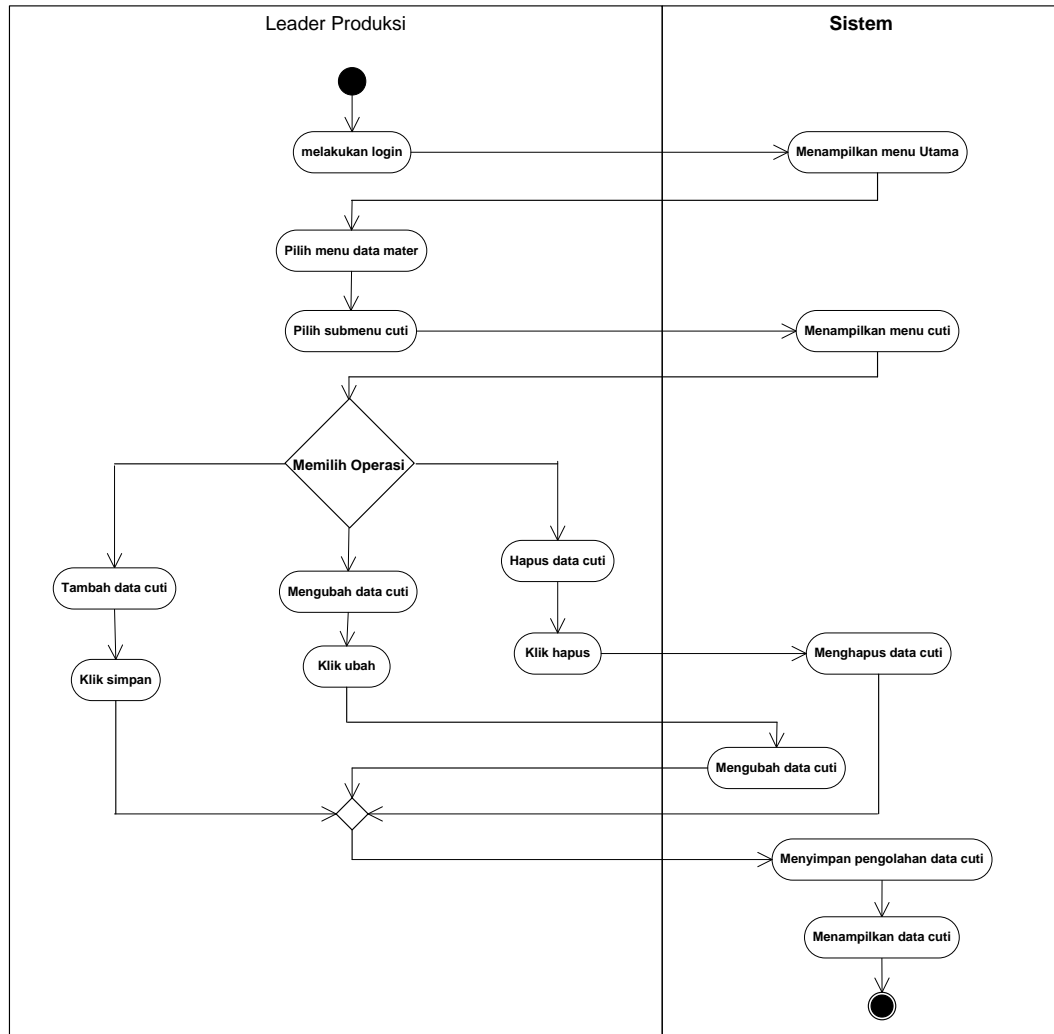
Gambar V.6 Activity Diagram Mengelola Data Jabatan
(Sumber: Hasil Analisis Data, 2017)

5. *Activity Diagram* Menginput Data Lembur
Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu data lembur. Data lembur karyawan dikelola oleh leader produksi. Leader produksi dapat tambah, cari, ubah dan hapus data potongan karyawan. *Activity diagram* menginput lembur karyawan dapat dilihat pada gambar V.7.



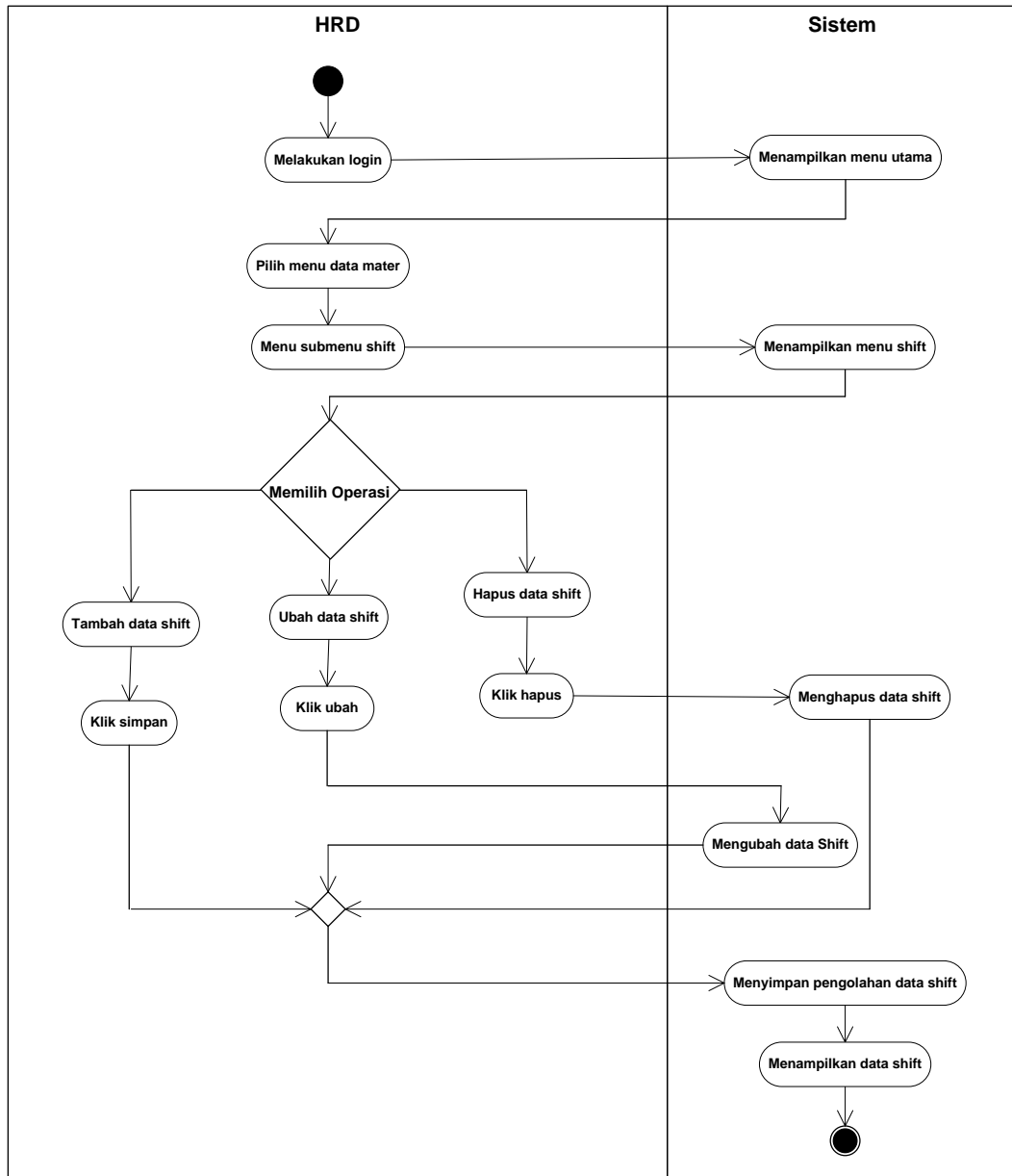
Gambar V.7 *Activity Diagram* Menginput Data Lembur
(Sumber: Hasil Analisis Data, 2017)

6. *Activity Diagram* Menginput Data Cuti
Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu data *master*. Data cuti karyawan dikelola oleh leader produksi. Leader produksi dapat tambah, cari, ubah dan hapus data potongan karyawan. *Activity diagram* menginput cuti karyawan dapat dilihat pada gambar V.8.



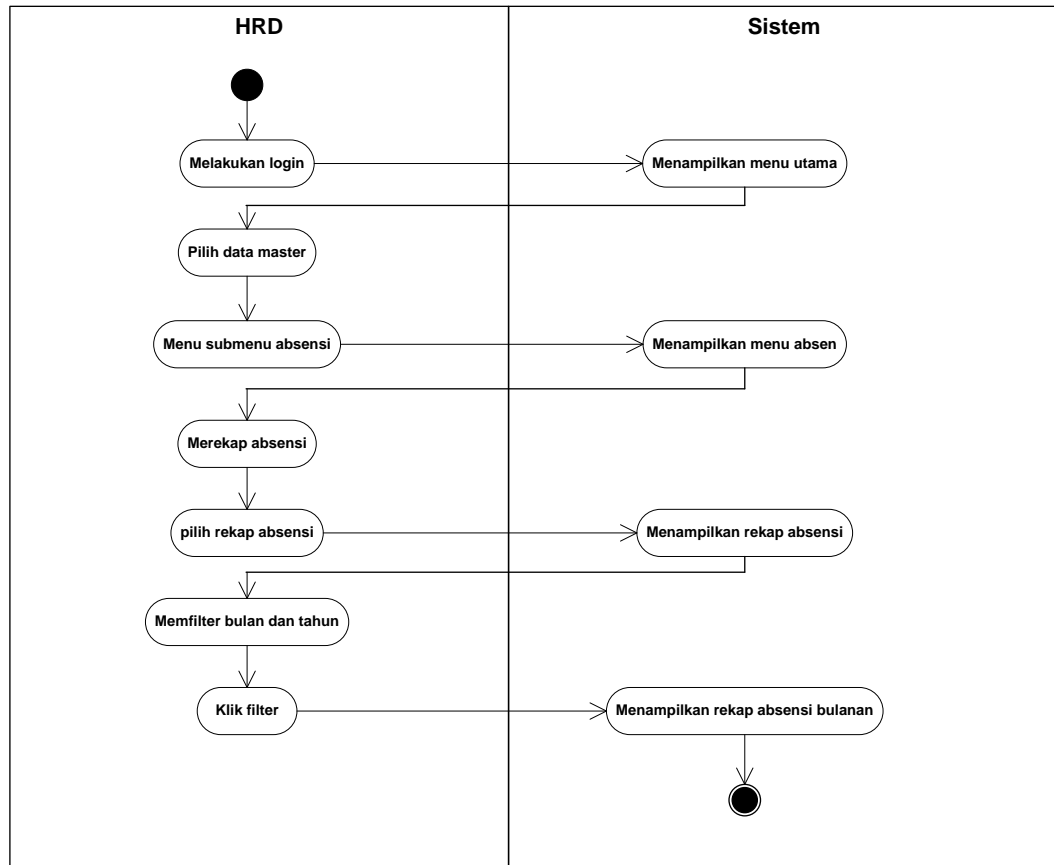
Gambar V.8 Activity Diagram Menginput Data Cuti
(Sumber: Hasil Analisis Data, 2017)

7. *Activity Diagram Mengelola Data Shift*
Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu data *master*. Data *shift* dikelola oleh HRD. HRD dapat menggunakan operasi tambah, cari, ubah dan hapus data *shift*. *Activity diagram* data *shift* dapat dilihat pada gambar V.9.



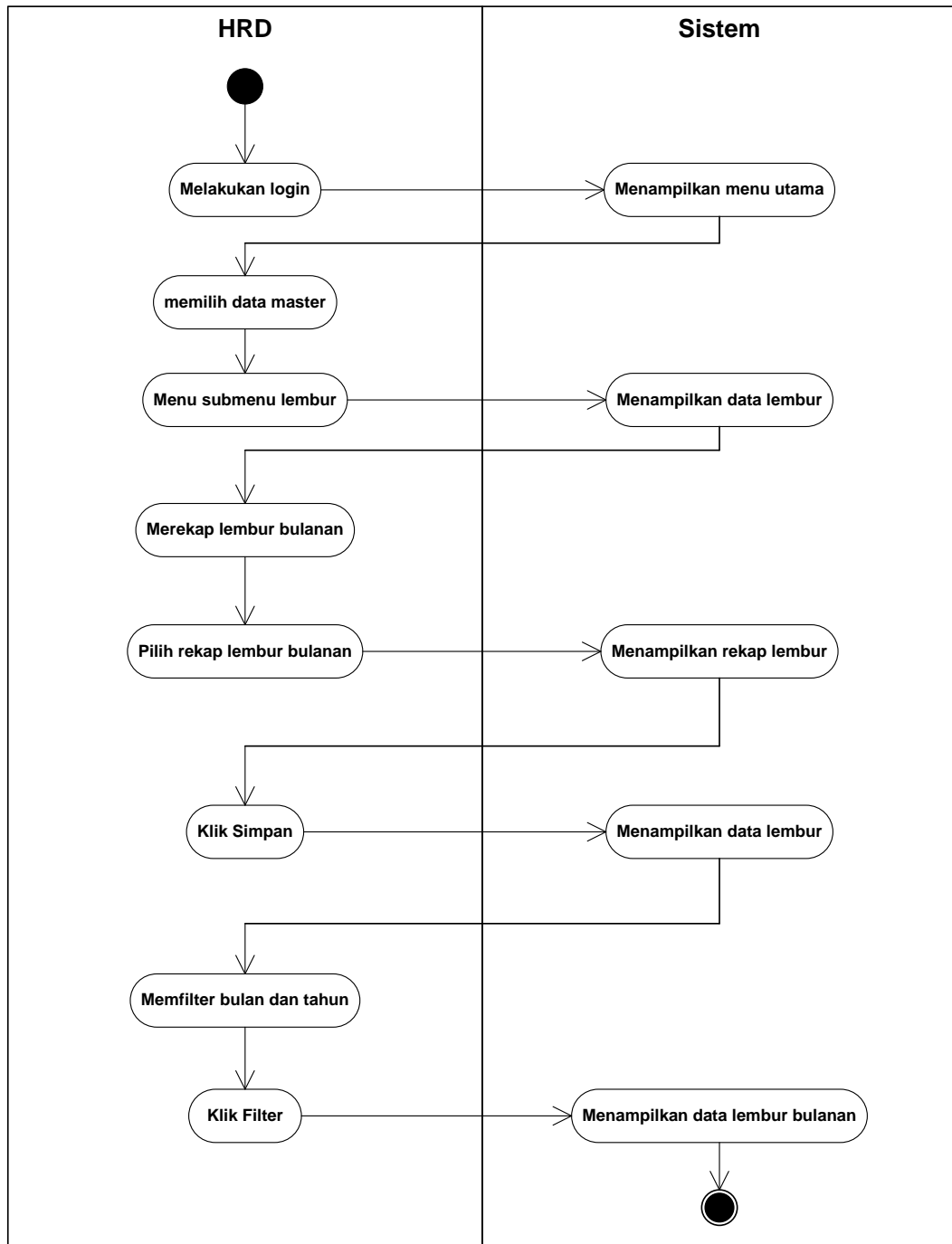
Gambar V.9 Activity Diagram Mengelola Data Shift
(Sumber: Hasil Analisis Data, 2017)

8. *Activity Diagram* Rekap Absensi satu Bulan
Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu data transaksi. Absensi karyawan untuk direkap setiap hari kerja bila karyawan tidak masuk dengan menginput keterangan kehadiran. Berikut adalah *activity diagram* rekap absensi satu bulan dapat dilihat pada gambar V.10.



Gambar V.10 *Activity Diagram* Mengelola Data Rekap Absensi
(Sumber: Hasil Analisis Data, 2017)

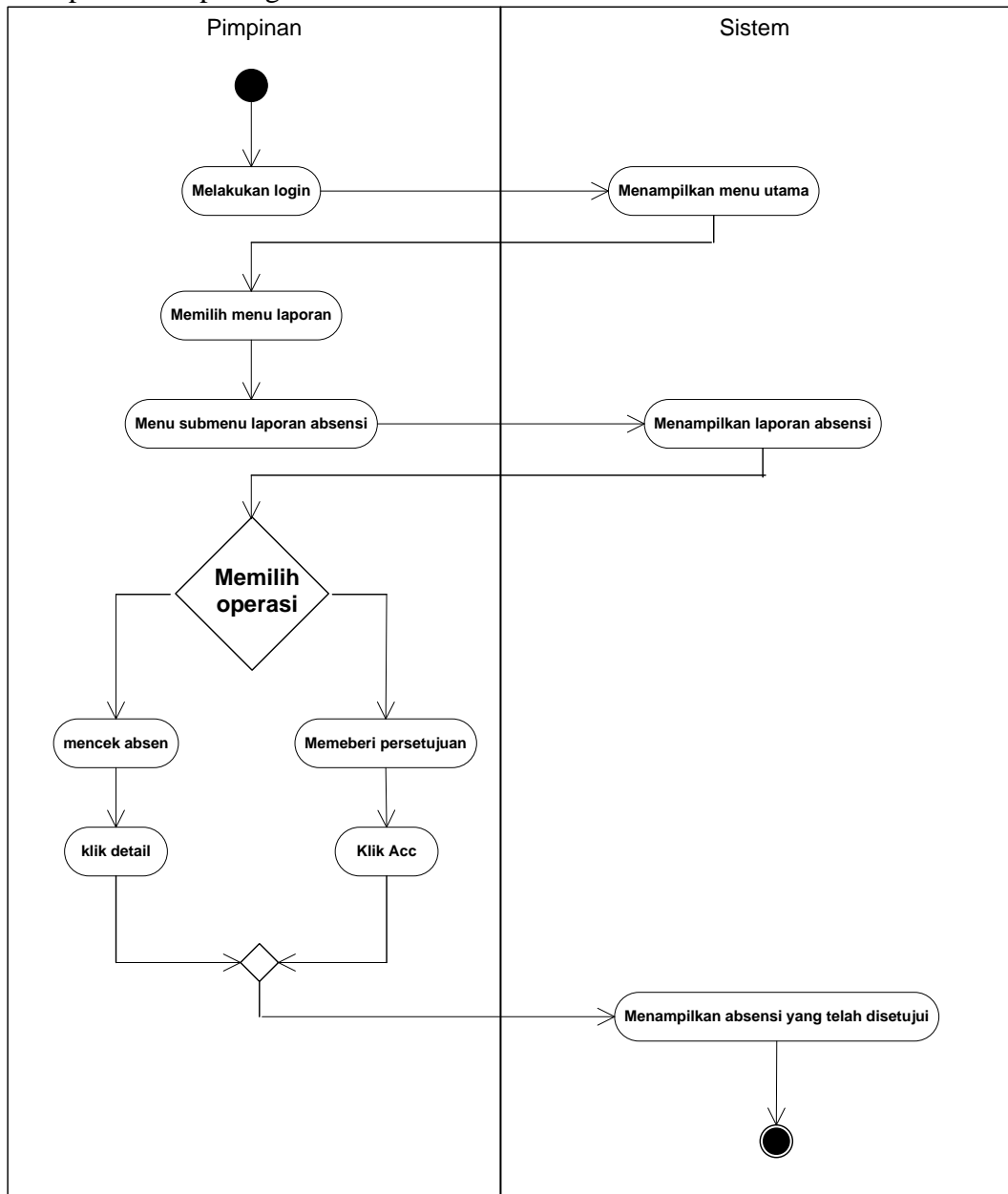
9. *Activity Diagram* Rekap Lembur satu Bulan
Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu data transaksi. Lembur karyawan untuk direkap setiap hari kerja dengan menginput jam lembur dan menghitung total harian jam lembur. Berikut adalah *activity diagram* rekap lembur satu bulan dapat dilihat pada gambar V.11.



Gambar V.11 Activity Diagram Mengelola Data Rekap Lembur
(Sumber: Hasil Analisis Data, 2017)

10. *Activity Diagram View Rekap Disetujui Pimpinan*
Activity diagram berikut ini menjelaskan aktivitas yang dilakukan ketika memilih menu data transaksi. Rekap absensi karyawan yang sudah dikelola oleh HRD lalu, dicek oleh pimpinan untuk disetujui. pimpinan sebagai *user* untuk melihat hasil reka absensi karyawan. Berikut

merupakan *activity diagram view* rekap absensi karyawan yang disetujui dapat dilihat pada gambar V.12.

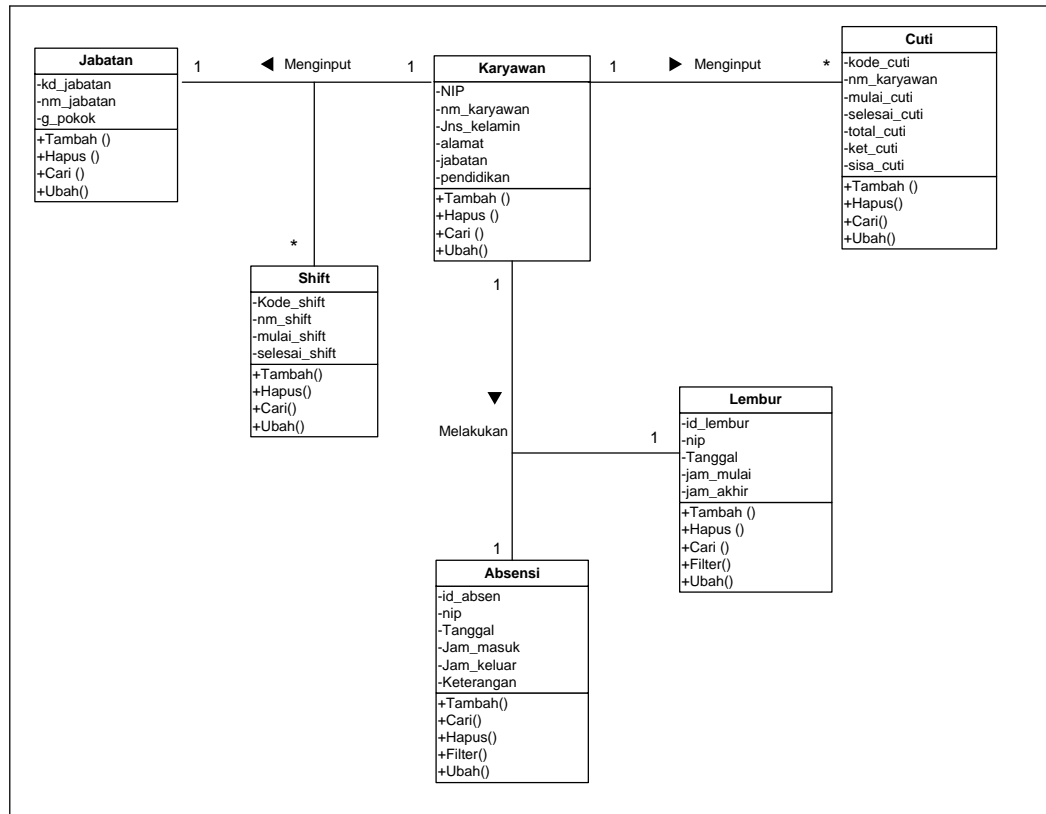


Gambar V.12 *Activity Diagram View* Rekap Absensi dan Lembur Disetujui Pimpinan
(Sumber: Hasil Analisis Data, 2017)

5.3.3 *Class diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem *class diagram* memperlihatkan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas di dalam model desain dari suatu sistem. *Class diagram* sistem informasi

pengendalian kualitas produk yang diusulkan dapat dilihat pada Gambar V.16 menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem *class diagram* memperlihatkan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas di dalam model desain dari suatu sistem. *Class diagram* sistem informasi absensi yang diusulkan dapat dilihat pada Gambar V.13



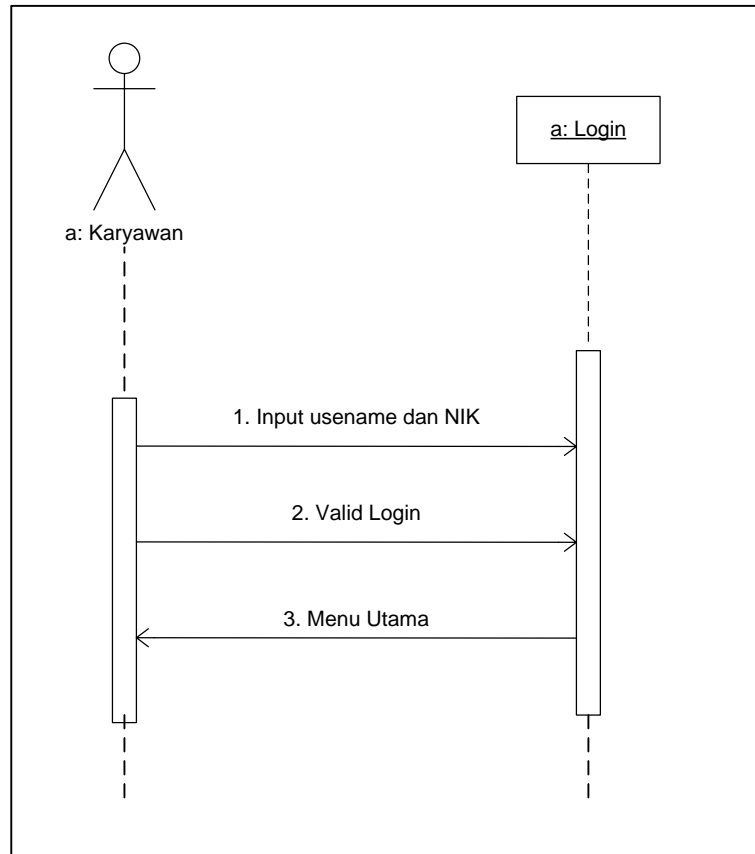
Gambar V.13. *Class Diagram* Absensi Usulan
(Sumber: Hasil Analisis Data, 2017)

5.3.4 *Sequence Diagram*

Sequence diagram menggambarkan interaksi yang terjadi pada suatu objek *use case diagram* ketika melakukan suatu proses tertentu, di mana urutan proses ketika melakukan suatu proses tertentu dapat diketahui dengan melihat gambaran pada *diagram*. Hubungan yang ada pada gambar di bawah ini adalah proses yang dilakukan oleh sistem ketika melakukan proses yang sesuai dengan suatu objek *use case diagram*, berikut adalah *sequence diagram* pada sistem informasi absensi karyawan :

1. *Sequence Diagram* pada *Use Case Login*

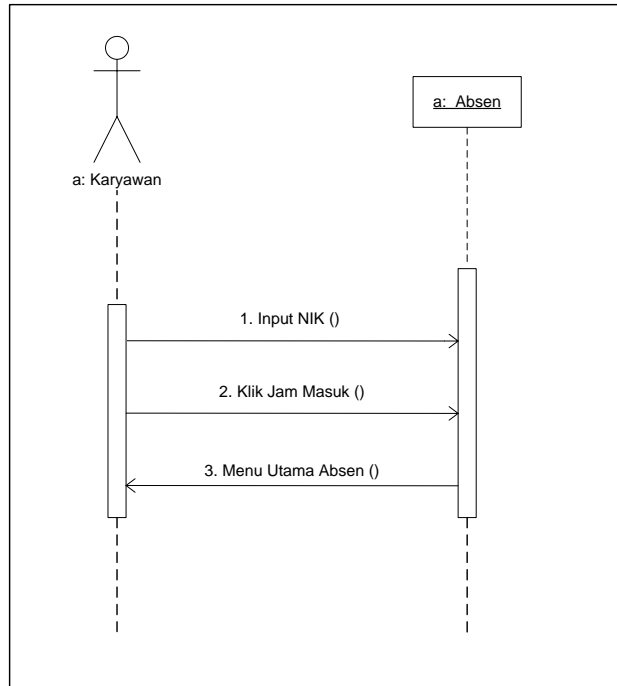
Sequence diagram login menjelaskan sebuah *sequence diagram* dalam proses login. Proses ini dilakukan oleh *user* sebelum masuk ke sistem. Adapun *sequence diagram* dari *use case login* dapat dilihat pada gambar V.14 di bawah ini.



Gambar V.14. *Sequence Diagram Login*
(Sumber: Hasil Analisis Data, 2017)

2. *Sequence Diagram* pada *Use Case* Proses Melakukan Absensi

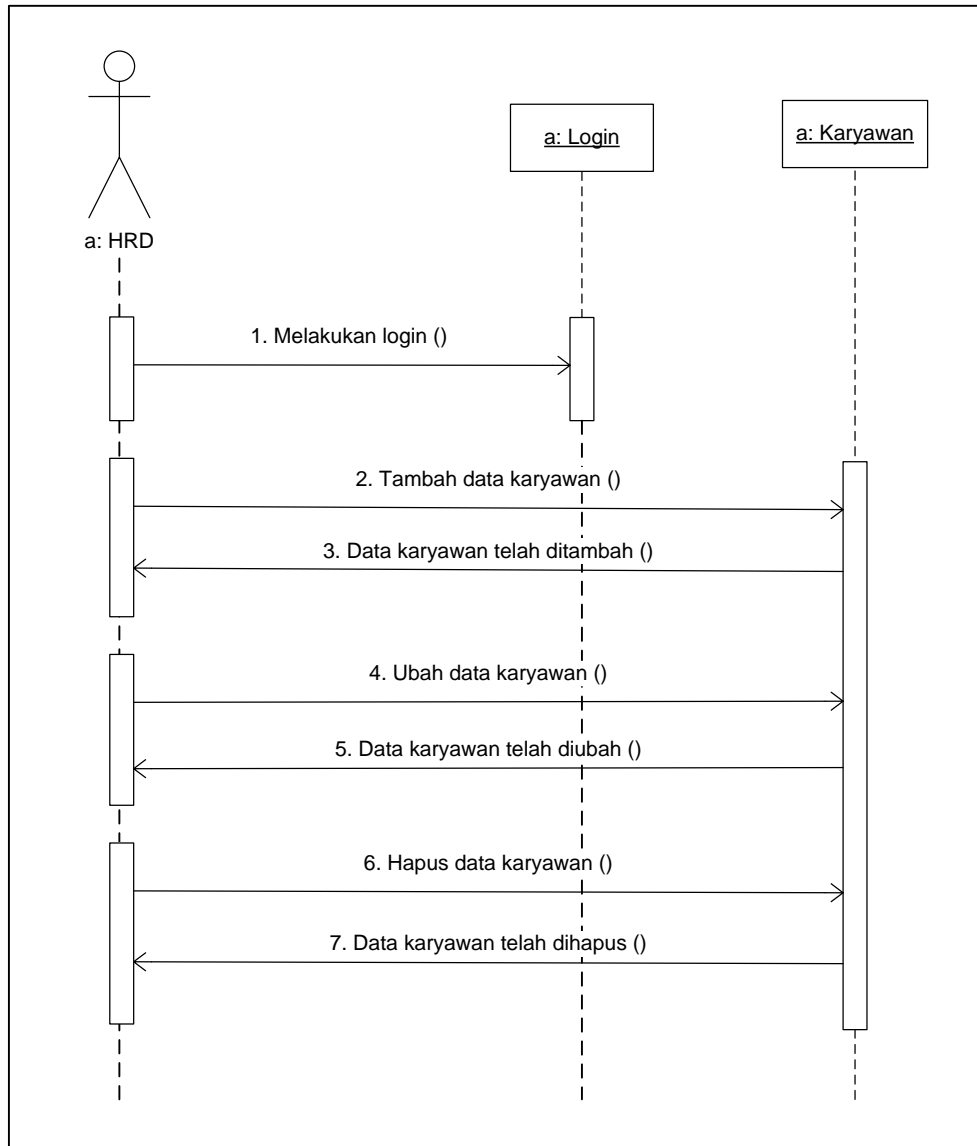
Sequence diagram melakukan absensi menjelaskan sebuah *sequence diagram* dalam proses mengaktifkan sistem untuk melakukan aktivitas absensi yang dilakukan oleh karyawan. Adapun *sequence diagram* dari proses melakukan absensi dapat dilihat pada gambar V.15.



Gambar V.15 *Sequence Diagram* Melakukan Absensi
(Sumber: Hasil Analisis Data, 2017)

3. *Sequence Diagram* pada *Use Case* Mengelola Data Karyawan

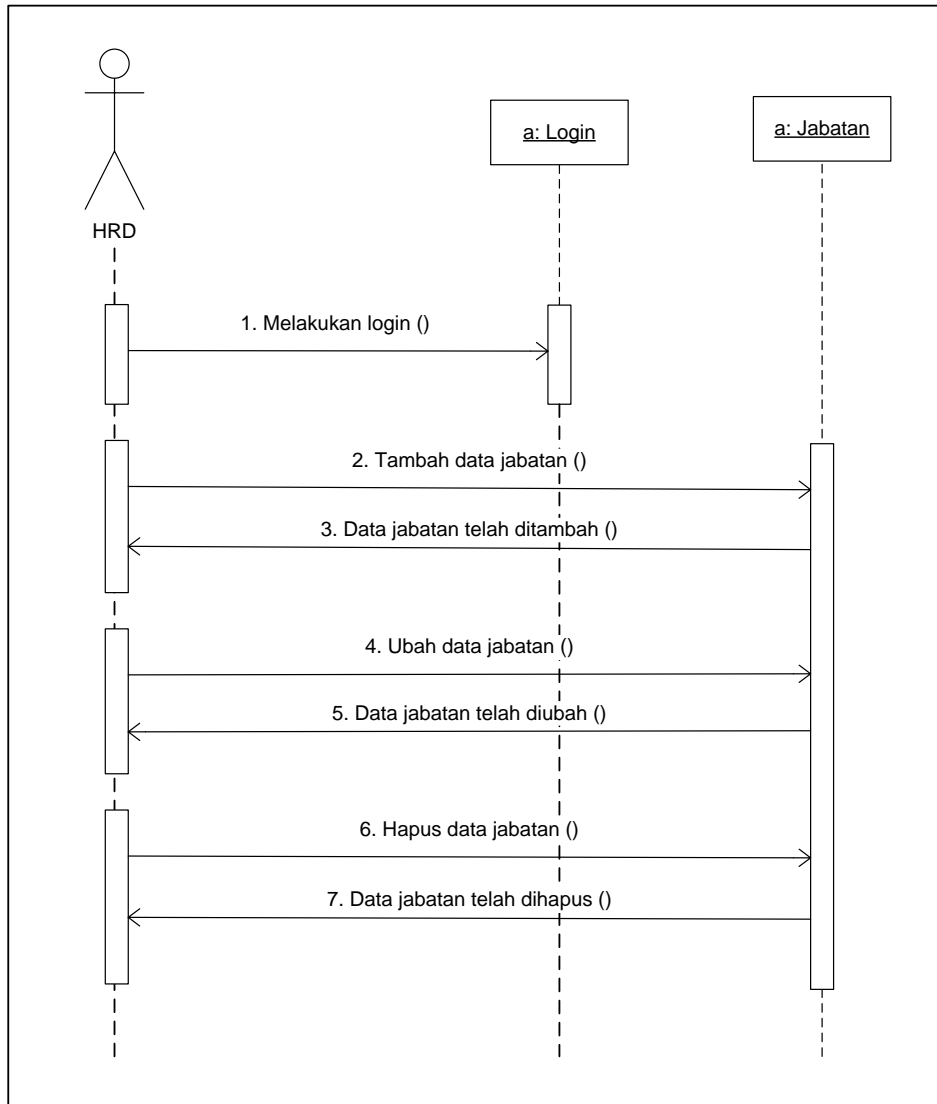
Sequence diagram data karyawan menjelaskan sebuah *sequence diagram* dalam proses tambah, cari, ubah dan hapus data karyawan yang dilakukan oleh HRD. Adapun *sequence diagram* dari *use case* mengelola data dapat dilihat pada gambar V.16.



Gambar V.16 *Sequence Diagram* Mengelola Data Karyawan
(Sumber: Hasil Analisis Data, 2017)

4. *Sequence Diagram* pada *Use Case* Mengelola Data Jabatan

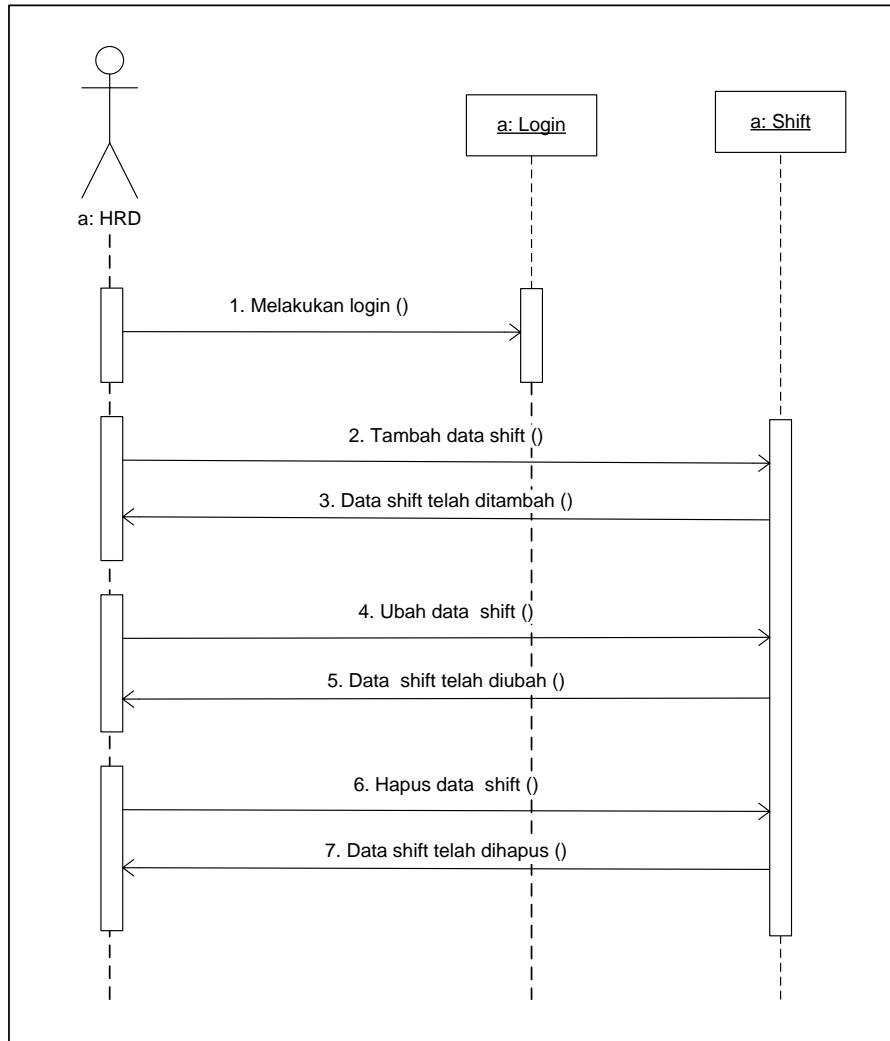
Sequence diagram data jabatan menjelaskan sebuah *sequence diagram* dalam proses tambah, cari, ubah dan hapus data bagian yang dilakukan oleh HRD. Adapun *sequence diagram* dari *use case* mengelola data dapat dilihat pada gambar V.17.



Gambar V.17 *Sequence Diagram* Mengelola Data Jabatan
(Sumber: Hasil Analisis Data, 2017)

5. *Sequence Diagram* pada Use Case Mengelola Data Shift

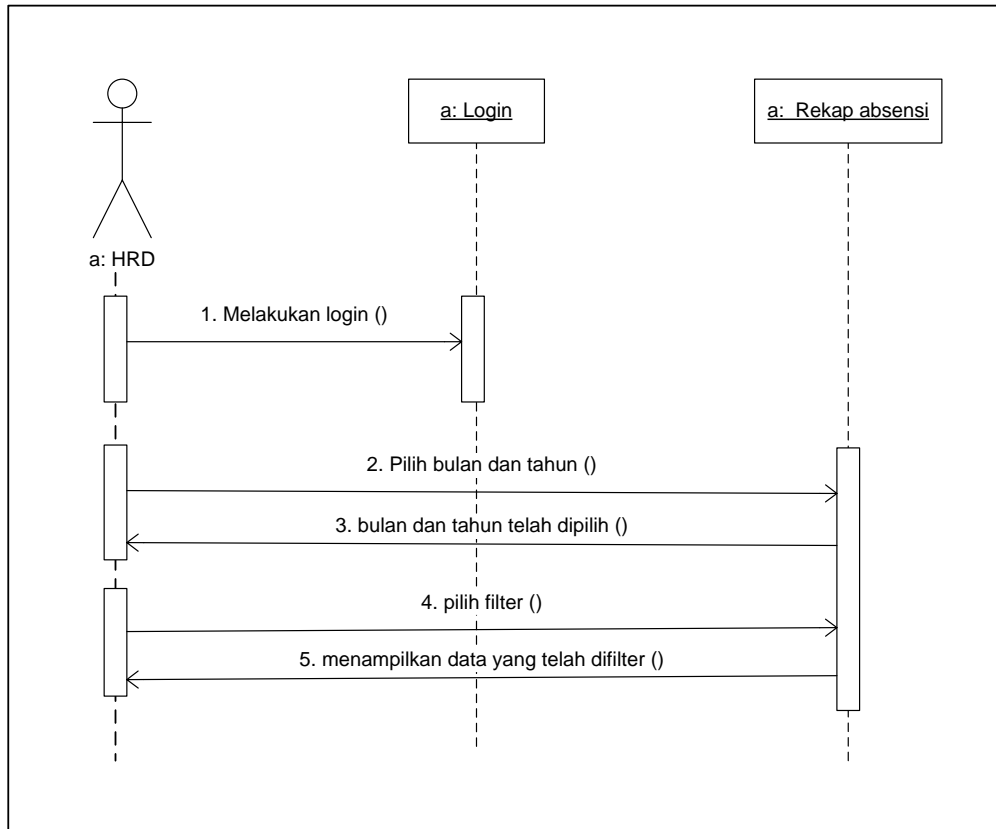
Sequence diagram data master shift menjelaskan sebuah *sequence diagram* dalam proses tambah, cari, ubah dan hapus data master shift yang dilakukan oleh HRD. Adapun *sequence diagram* dari use case mengelola data dapat dilihat pada gambar V.18.



Gambar V.18 *Sequence Diagram* Mengelola Data Shift
(Sumber: Hasil Analisis Data, 2017)

6. *Sequence Diagram* pada Use Case Merekap Absensi satu Bulan

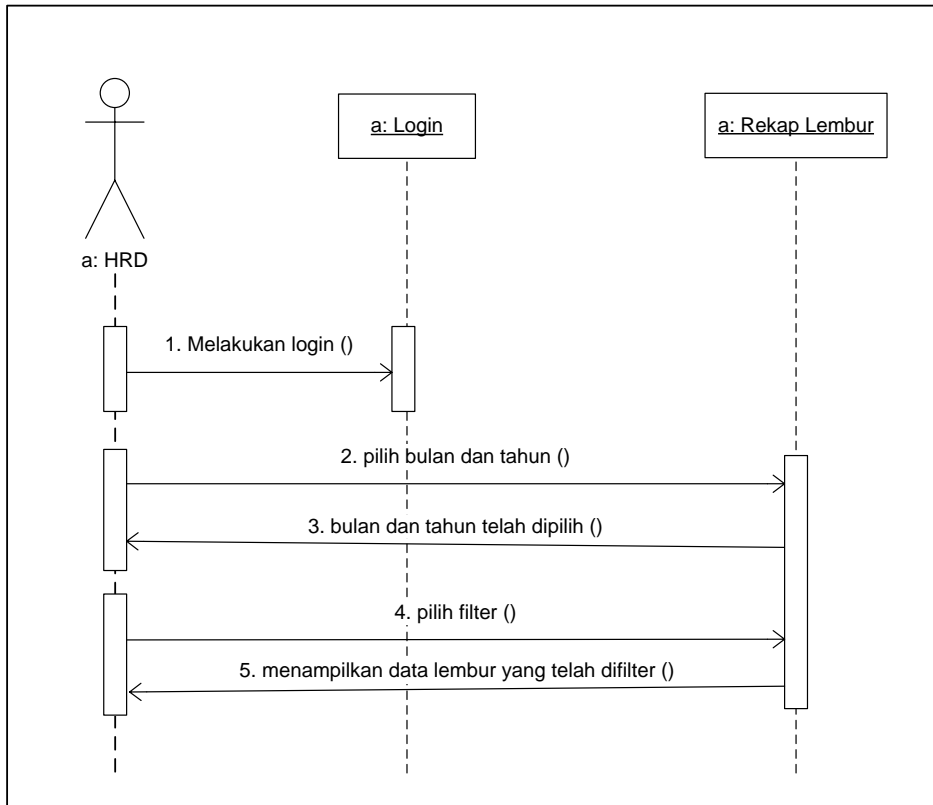
Sequence diagram rekap absensi satu bulan menjelaskan sebuah *sequence diagram* dalam melihat absensi karyawan selama satu bulan yang dilakukan oleh HRD. Adapun *sequence diagram* dari use case rekap absensi satu bulan dapat dilihat pada gambar V.19.



Gambar V.19. *Sequence Diagram* Merekap Data Absensi
(Sumber: Hasil Analisis Data, 2017)

7. *Sequence Diagram* pada *Use Case* Rekap Lembur satu Bulan

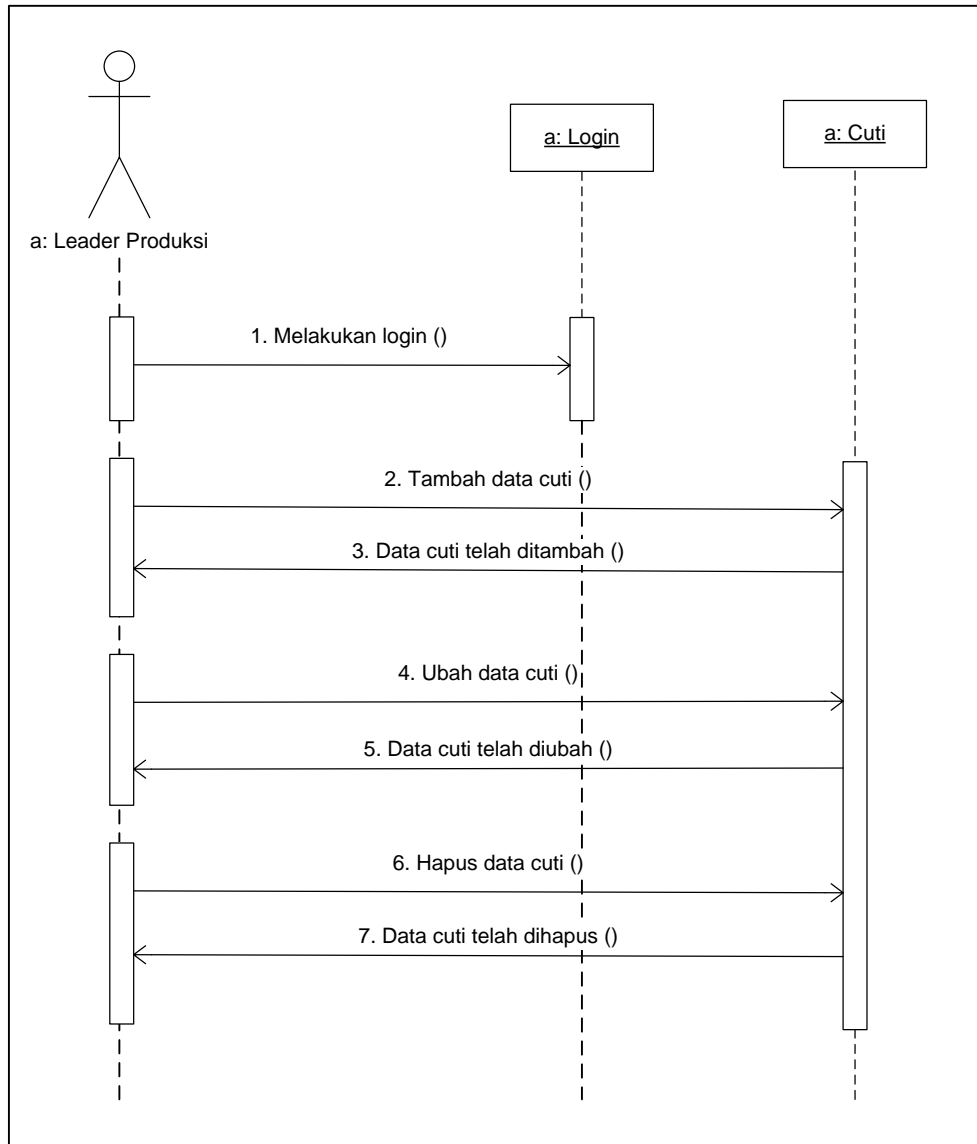
Sequence diagram rekap lembur satu bulan menjelaskan sebuah *sequence diagram* saat perhitungan jam lembur karyawan per-hari sampai sebulan yang dilakukan oleh HRD. Adapun *sequence diagram* dari *use case* rekap absensi satu bulan dapat dilihat pada gambar V.20.



Gambar V.20 *Sequence Diagram* Mengolah Data Rekap Lembur
(Sumber: Hasil Analisis Data, 2017)

8. *Sequence Diagram* pada *Use Case* Menginput Cuti

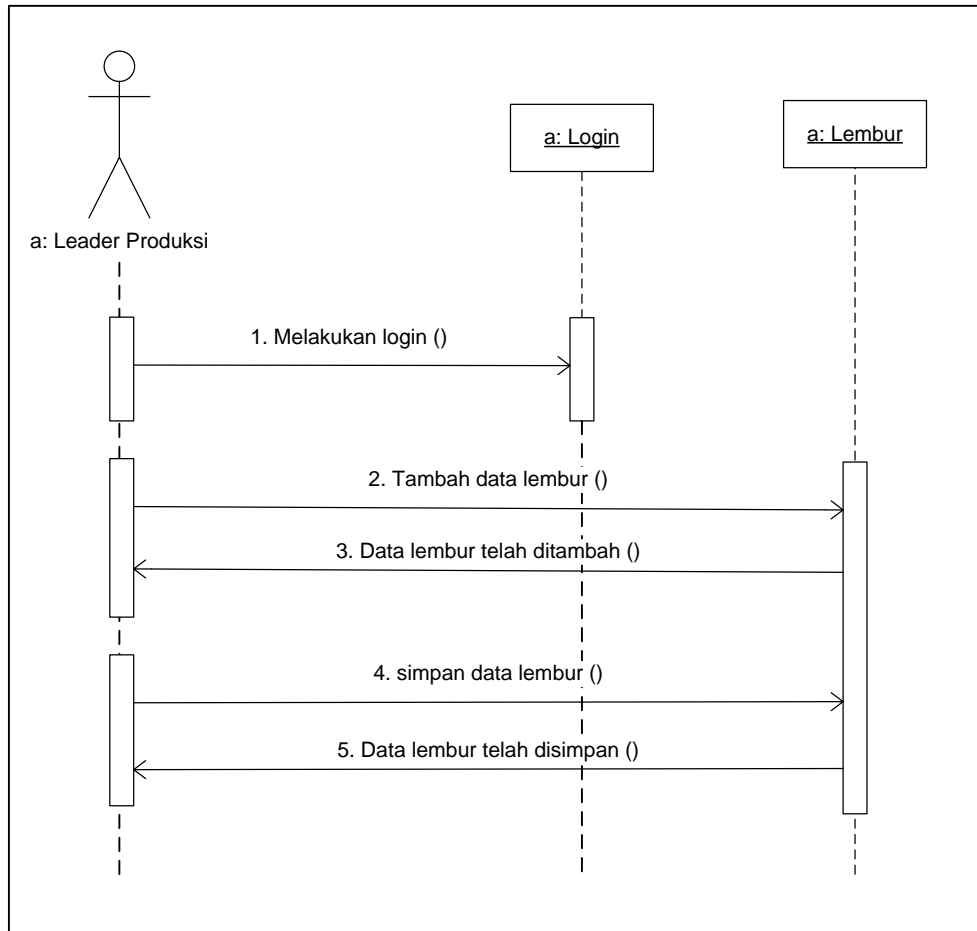
Sequence diagram menginput cuti dikelola oleh leader produksi lalu direkap oleh HRD. Leader menginput karyawan yang akan cuti. Adapun *sequence diagram* dari *use case* menginput cuti dapat dilihat pada gambar V.21.



Gambar V.21. *Sequence Diagram* Menginput cuti
(Sumber: Hasil Analisis Data, 2017)

9. *Sequence Diagram* pada *Use Case* Menginput Lembur

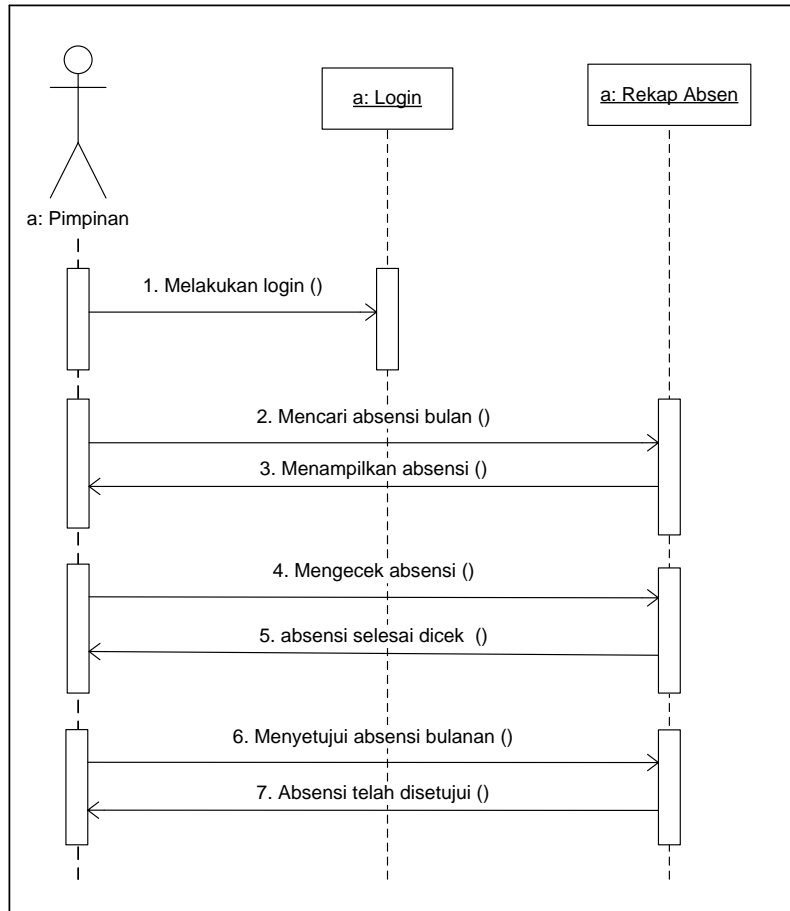
Sequence diagram menginput lembur dikelola oleh leader produksi lalu direkap oleh HRD. Leader menginput karyawan yang akan lembur hari ini dan menentukan jumlah jam lemburnya. Adapun *sequence diagram* dari *use case* menginput cuti dapat dilihat pada gambar V.22.



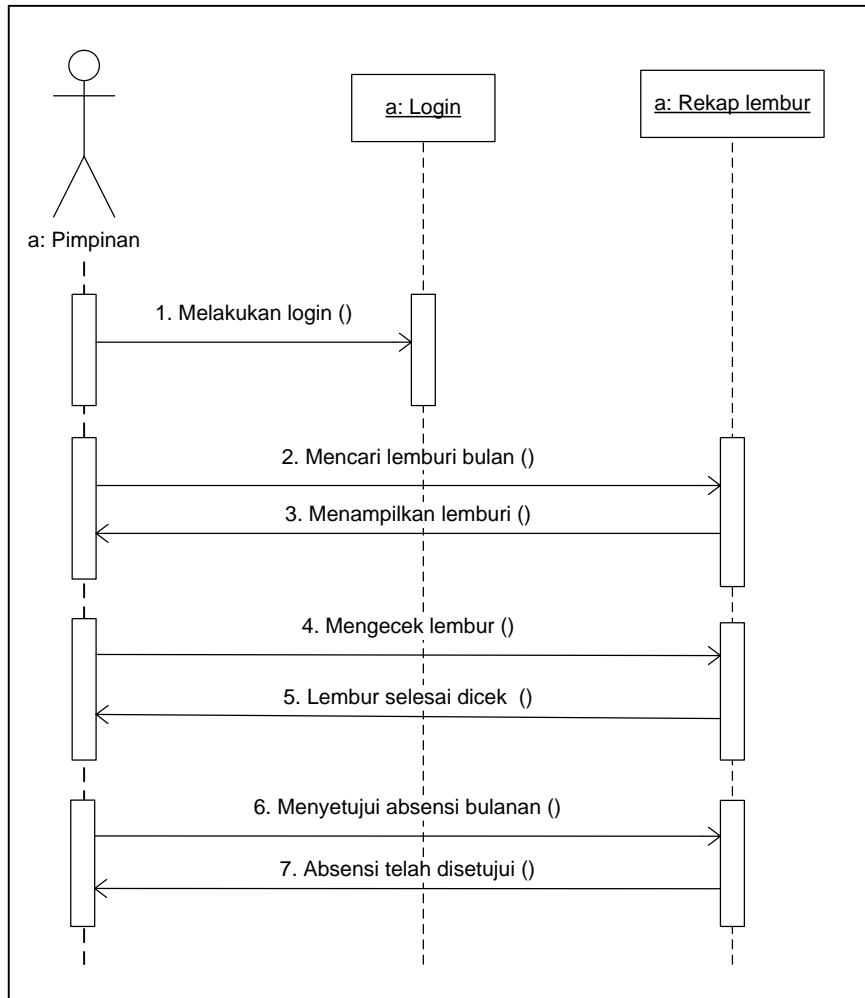
Gambar V.22 *Sequence Diagram* Menginput Lembur
 (Sumber: Hasil Analisis Data, 2017)

10. *Sequence Diagram* pada *Use Case View* Rekap Absensi dan Lembur Disetujui Pimpinan

Sequence diagram view rekap absensi dan lembur karyawan yang sudah dikelola oleh HRD lalu dicek oleh pimpinan untuk disetujui. Pimpinan sebagai *user* untuk melihat hasil laporan absensi karyawan. Adapun *sequence diagram* dari *use case view* laporan absensi yang disetujui pimpinan dapat dilihat pada gambar V.23 dan V.24.



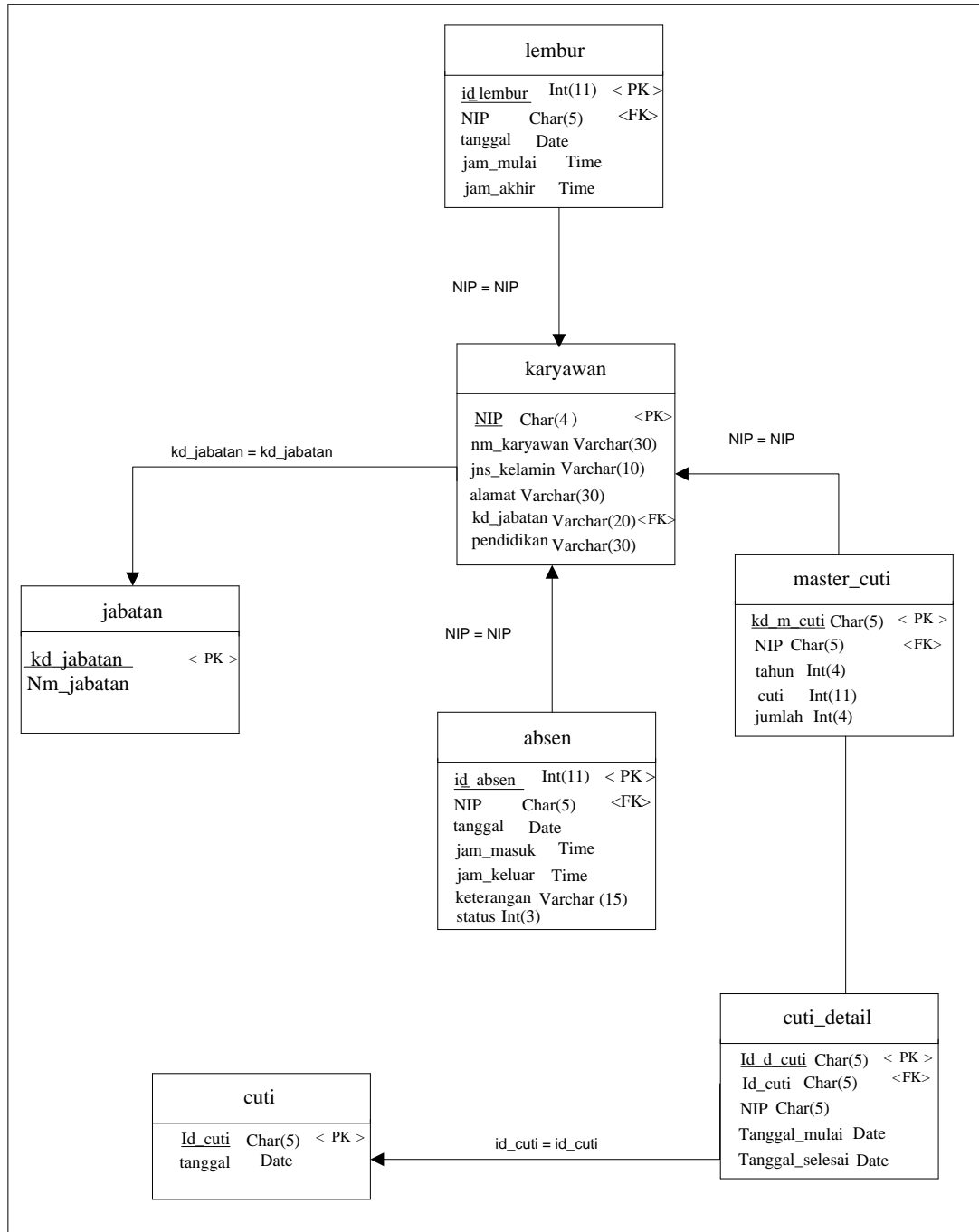
Gambar V.23 *Sequence Diagram View* Rekap Absensi disetujui Pimpinan
 (Sumber: Hasil Analisis Data, 2017)



Gambar V.24 *Sequence Diagram View* Rekap Lembur disetujui Pimpinan
(Sumber: Hasil Analisis Data, 2017)

5.3.4 *Entity Relationship Diagram (ERD)*

ERD merupakan 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. ERD sistem informasi absensi yang diusulkan dapat dilihat pada Gambar V.25.



Gambar V.25 Physical Data Model Usulan
(Sumber: Hasil Analisis Data, 2017)

5.3.5 Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen data *store*. Pada tahap perancangan sistem, kamus data digunakan untuk merancang

input, laporan dan basis data. Berikut kamus data pada aplikasi sistem informasi absensi karyawan.

1. Spesifikasi Tabel absen

Nama Tabel : absen
 Fungsi : Untuk menyimpan data absen karyawan
 Tipe : File Data Absen

Tabel V.15 Spesifikasi Tabel Absen

No.	Elemen	Akronim	Tipe data	Panjang	Keterangan
1.	ID Absen	id_absen	Int	11	<i>Primary key</i>
2.	Nip	Nip	Char	4	<i>Foreign key</i>
3.	Tanggal	Tanggal	Date		
4.	Jam Masuk	jam_masuk	Time		
5.	Jam Pulang	jam_pulang	Time		
6.	Keterangan	Keterangan	Varchar	50	

(Sumber: Hasil Analisis Data, 2017)

2. Spesifikasi Tabel karyawan

Nama Tabel : karyawan
 Fungsi : Untuk menyimpan data karyawan
 Tipe : File Data *Master*

Tabel V.16 Spesifikasi Tabel karyawan

No.	Elemen	Akronim	Tipe data	Panjang	Keterangan
1.	NIP	NIP	Char	4	<i>Primary key</i>
2.	Nama	nm_karyawan	Varchar	30	
3.	Jenis Kelamin	jns_kelamin	Varchar	10	
4.	Alamat	Alamat	Varchar	30	
5.	Jabatan	Jabatan	Vachar	20	
6.	Pendidikan	Pendidikan	Varchar	30	

(Sumber: Hasil Analisis Data, 2017)

3. Spesifikasi Tabel Jabatan

Nama Tabel : jabatan
 Fungsi : Untuk menyimpan data jabatan
 Tipe : File Data *Master*

Tabel V.17 Spesifikasi Tabel jabatan

No.	Elemen	Akronim	Tipe data	Panjang	Keterangan
1.	Kode jabatan	kd_jabatan	Char	4	<i>Primary key</i>
2.	Nama jabatan	nm_jabatan	Varchar	10	

(Sumber: Hasil Analisis Data, 2017)

4. Spesifikasi Tabel Lembur

Nama Tabel : lembur
Fungsi : Untuk menyimpan data lembur karyawan
Tipe : File Data Transaksi

Tabel V.18 Spesifikasi Tabel Lembur

No.	Elemen	Akroim	Tipe Data	Panjang	Keterangan
1.	ID Lembur	id_lembur	Int	11	Primary key
2.	NIP	Nip	Char	5	Foreign key
3.	Tanggal lembur	Tanggal	Date		
4.	Jam mulai	jam_mulai	Time		
5.	Jam akhir	jam_akhir	Time		

(Sumber: Hasil Analisis Data, 2017)

5. Spesifikasi Tabel *Shift*

Nama Tabel : shift
Fungsi : Untuk menyimpan data *shift*
Tipe : File Data *Master*

Tabel V.19 Spesifikasi Tabel *Shift*

No.	Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Kode <i>Shift</i>	kode_shift	Char	5	Primary key
2.	Nama <i>Shift</i>	nm_shift	Varchar	50	
3.	Mulai <i>Shift</i>	mulai_shift	Time		
4.	Selesai <i>Shift</i>	selesai_shift	Time		

(Sumber: Hasil Analisis Data, 2017)

6. Spesifikasi Tabel Cuti

Nama Tabel : cuti
Fungsi : Untuk menyimpan data cuti
Tipe : File Data Transaksi

Tabel V.20 Spesifikasi Tabel cuti

No.	Elemen	Akronim	Tipe Data	Panjang	Keterangan
1.	Kode Cuti	kd_m_cuti	Char	5	Primary key
2.	NIP	NIP	Char	5	
3.	Tahun	Tahun	Int	4	
4.	Jumlah cuti	Jumlah	Int	11	

(Sumber: Hasil Analisis Data, 2017)

5.3.6 *Deployment Diagram* yang Diusulkan

Deployment diagram ini digunakan untuk membuat gambaran susunan fisik sebuah sistem, dan gambaran bagian perangkat lunak mana yang berjalan pada perangkat keras yang mana. Berikut ini adalah penjelasan *deployment diagram* dari sistem informasi absensi karyawan:

1. PHP

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah aplikasi berbasis web dan bias digunakan bersamaan dengan HTML.

2. Database

Database adalah program komputer yang menyediakan layanan data lainnya ke komputer atau program komputer, seperti yang ditetapkan oleh model *client-server*. Istilah ini juga merujuk kepada sebuah komputer yang didedikasikan untuk menjalankan program *server database*.

3. Web Server

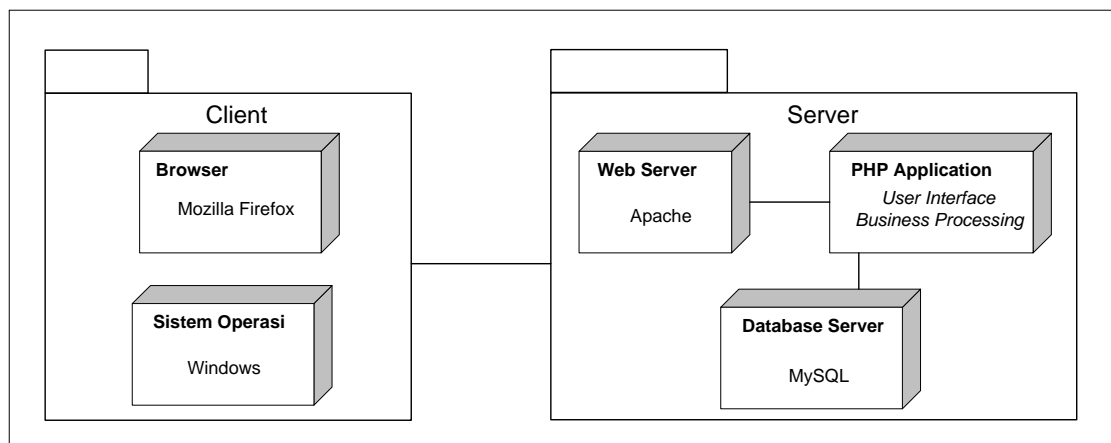
Suatu program sekaligus mesin yang menjalankan program komputer yang mengerti protokol HTTP dan dapat menanggapi permintaan-permintaan dari *web browser*.

4. Web Browser

Web browser memiliki arti sebagai penjelajah *web*. Fungsi *web browser* itu sendiri adalah untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *web server*.

5. Sistem Operasi

Sistem operasi adalah perangkat lunak komputer atau *software* yang bertugas untuk melakukan control dan manajemen perangkat keras dan juga operasi-operasi dasar sistem, termasuk menjalankan *software* aplikasi seperti program-program pengolahan data.



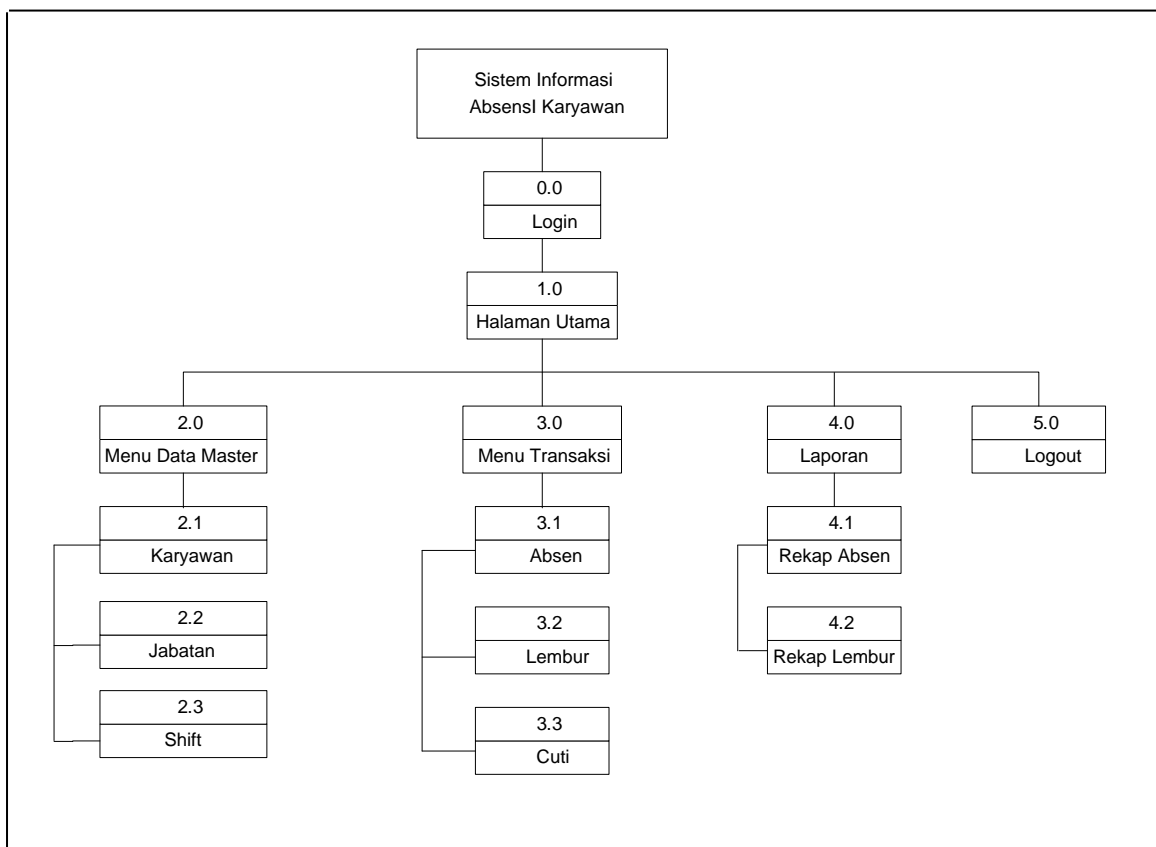
Gambar V.26 *Deployment Diagram* yang Diusulan
(Sumber: Hasil Analisis Data, 2017)

5.4 Analisis Desain Sistem

Tahap ini merupakan tahap kedua dalam metodologi *prototype evolutioner*, yaitu tahap membuat sebuah *prototype* dari aplikasi. Dimulai dengan analisis desain usulan yang meliputi pembuatan HIPO (*Hierarchy Input-Process-Output*), *interface* dan *flowchart* aplikasi persediaan bahan baku.

5.4.1 *Hierarchy plus Input-Process-Output (HIPO) Aplikasi*

HIPO merupakan alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. Dalam hal ini pembuatan sistem informasi absensi karyawan dibuat menggunakan perangkat lunak PHP untuk bahas pemrograman, serta *MySQL My Structured Query Language*) yang berfungsi sebagai perangkat lunak basis data. Berikut adalah struktur menu hirarki program yang digambarkan dalam HIPO (*Hierarchy plus Input-Process-Output*):

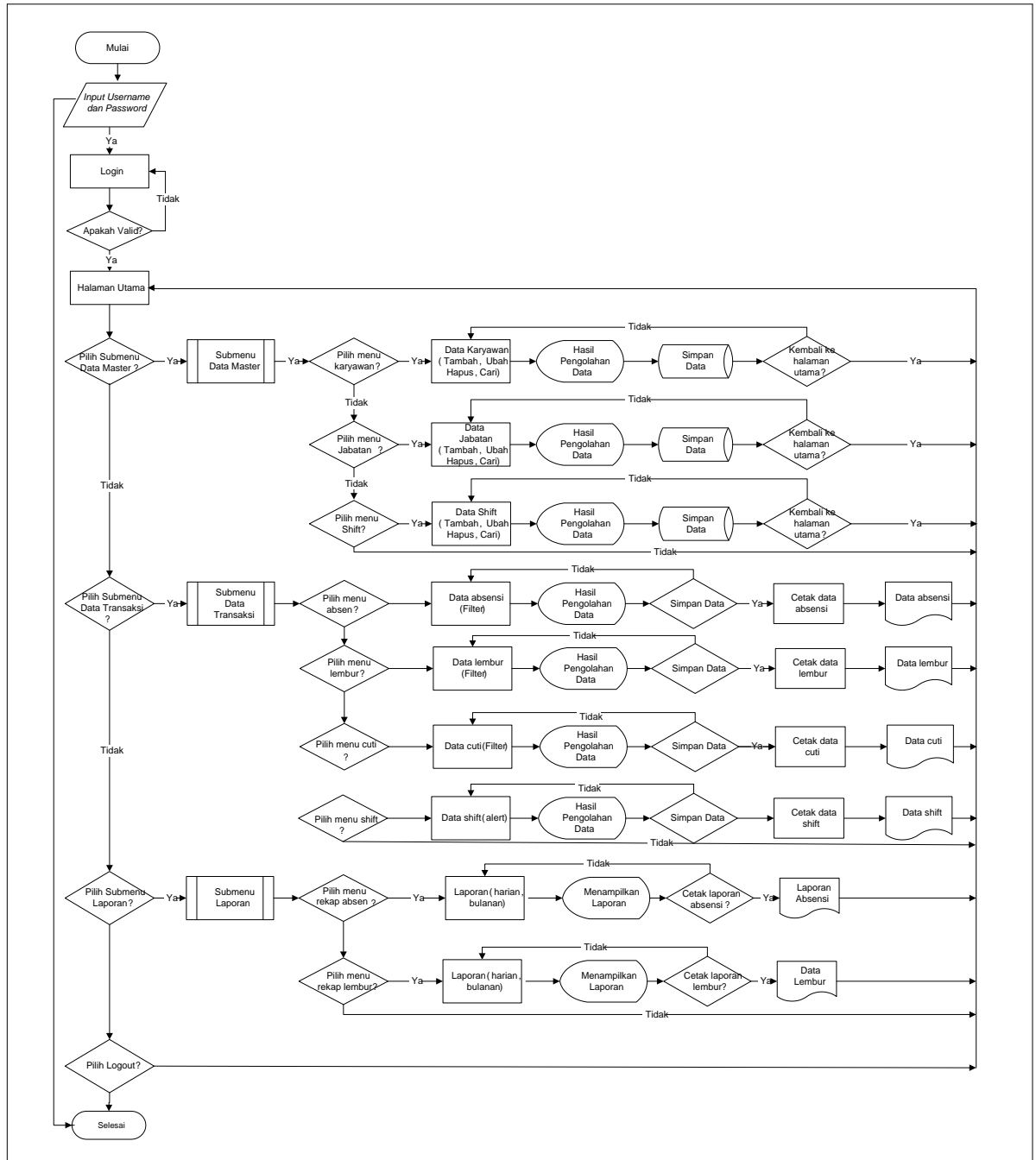


Gambar V.27 HIPO Sistem Usulan

(Sumber: Hasil Analisis Data, 2017)

5.4.2 *Flowchart Aplikasi Usulan*

Flowchart aplikasi sistem informasi absensi karyawan dapat menggambarkan alur logika yang sebenarnya. Bagian ini juga memperjelas urutan prosedur sistem dan spesifikasi proses. Berikut adalah *flowchart* aplikasi sistem informasi absensi karyawan yang diusulkan terdapat pada gambar V.28:



Gambar V.28 Flowchart Sistem Usulan
(Sumber: Hasil Analisis Data, 2017)

5.4.3 Perancangan Interface Aplikasi Usulan

Interface aplikasi yang terdapat pada sistem informasi absensi dan karyawan ini berupa rancangan tampilan yang akan dibuat, mulai dari *form login* hingga tampilan laporan yang akan dihasilkan. Perancangan *interface* aplikasi usulan adalah sebagai berikut:

1. Tampilan *Login*

Form login adalah *form* yang digunakan untuk membedakan hak akses pengguna. *Form login* ini pengguna yang boleh masuk sistem adalah pengguna yang mengetahui *username* dan *password* atau pengguna yang memiliki wewenang untuk menggunakan sistem. Berikut adalah gambar perancangan *form login*:

```
graph TD; Title[LOGIN]; Input1[Masukkan user name]; Input2[Masukkan Password]; Button[Login]; Title --- Input1; Input1 --- Input2; Input2 --- Button;
```

Gambar V.29 Rancangan *Form Login*

(Sumber: Hasil Analisis Data, 2017)

2. Tampilan *login* absensi

Form login absensi adalah *form* yang digunakan untuk masuk ke dalam program aplikasi. *Form login* absensi ini pengguna yang boleh masuk sistem adalah karyawan. Berikut adalah gambar perancangan *form login* absensi:

```
graph TD; Title[Absensi]; Input1[Masukkan NIK]; Button1[Masuk]; Button2[Keluar]; Title --- Input1; Input1 --- Button1; Button1 --- Button2; Button2 --- Footer[Tanggal : DD-MMM-YYYY | Pukul : 00:00 Selamat Pagi];
```

Gambar V.30 Rancangan *Form Login Absensi*
(Sumber: Hasil Analisis Data, 2017)

3. Tampilan Data Karyawan

Menu data karyawan adalah *form* yang digunakan untuk mengelola data karyawan. Rancangan *interface* dari *form* data karyawan dapat dilihat pada gambar V.31 dan gambar V.32. Berikut penjelasan penggunaan pada *form* ini:

- Tombol tambah : untuk menambahkan data karyawan yang baru.
- Tombol ubah : untuk mengubah nama karyawan yang sudah ada.
- Tombol hapus : untuk menghapus data yang sudah ada.
- Tombol batal : untuk membatalkan input data pada *form*.
- Fungsi *search* : untuk mencari data karyawan yang ada pada *database*.

No	NIP	Nama Karyawan	Jenis kelamin	Alamat	Jabatan	Pendidikan	Aksi
							Ubah Hapus

Gambar V.31 Rancangan Data Karyawan
(Sumber: Hasil Analisis Data, 2017)

Form tambah karyawan adalah *form* untuk menambah data karyawan yang terdapat pada gambar V.32. Adapun penjelasan *field-field* dalam *form* tambah karyawan sebagai berikut:

- NIP : Berisi nomor induk karyawan.
- Nama : Berisi nama karyawan.
- Jenis Kelamin : Berisi pilihan jenis kelamin karyawan.
- Alamat : Berisi alamat karyawan.
- Jabatan : Berisi jabatan karyawan.
- Pendidikan : Berisi asal sekolah akhir.

PT RMA		
Beranda	Form Karyawan	
Absensi		
Data Master ▼		
Data Karyawan		
Data Transaksi		
Laporan		
Logout		
	NIP	<input type="text"/>
	Nama Karyawan	<input type="text"/>
	Jenis Kelamin	<input type="text"/>
	Alamat	<input type="text"/>
	Jabatan	<input type="text"/>
	Pendidikan	<input type="text"/>
		<input type="button" value="Simpan"/> <input type="button" value="Batal"/>

Gambar V.32 Rancangan *Form* Tambahan Karyawan
(Sumber: Hasil Analisis Data, 2017)

4. Tampilan Data Jabatan

Menu data bagian adalah *form* yang digunakan untuk mengelola data jabatan. Rancangan *interface* dari *form* jabatan dapat dilihat pada gambar V.33 dan gambar V.34. Berikut penjelasan penggunaan pada *form* ini:

PT Rekadaya Multi Adiprima									
Beranda	Form Jabatan								
Absensi									
Data Master ▼									
Data Jabatan									
Data Transaksi									
Laporan									
Logout									
	<input type="button" value="Tambah"/>								
	Show <input type="text" value="10"/> Entries								
	Search : <input type="text"/>								
	<table border="1"> <thead> <tr> <th>No</th> <th>Kode Jabatan</th> <th>Nama Jabatan</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td>Ubah Hapus</td> </tr> </tbody> </table>	No	Kode Jabatan	Nama Jabatan	Aksi				Ubah Hapus
No	Kode Jabatan	Nama Jabatan	Aksi						
			Ubah Hapus						

Gambar V.33 Rancangan Data Jabatan
(Sumber: Hasil Analisis Data, 2017)

Form tambahan jabatan adalah *form* untuk menambah data jabatan yang terdapat pada gambar V.34. Adapun penjelasan *field-field* dalam *form* tambahan jabatan sebagai berikut:

PT Rekadaya Multi Adiprima		
Beranda	Form Jabatan	
Absensi		
Data Master ▼		
Data Jabatan		
Data Transaksi		
Laporan		
Logout		
	Kode Jabatan	<input type="text"/>
	Nama Jabatan	<input type="text"/>
		<input type="button" value="Simpan"/> <input type="button" value="Batal"/>

Gambar V.34 Rancangan *Form* Tambahan Jabatan

(Sumber: Hasil Analisis Data, 2017)

5. Tampilan Data *Shift*

Menu data *shift* adalah *form* yang digunakan untuk mengelola data *shift*. Rancangan *interface* dari *form shift* dapat dilihat pada gambar V.35 dan gambar V.36. Berikut penjelasan penggunaan pada *form* ini:

- Tombol tambah : untuk menambahkan data potongan yang baru.
- Tombol ubah : untuk mengubah nama potongan yang sudah ada.
- Tombol hapus : untuk menghapus data potongan yang sudah ada.
- Tombol batal : untuk membatalkan input data pada *form*.
- Fungsi *search* : untuk mencari data potongan yang ada pada *database*.

PT RMA						
Beranda	Form Shift					
Absensi						
Data Master ▼						
Data Shift	<input type="button" value="+ Tambah"/>					
Data Transaksi	Show	<input type="text" value="10"/>	Entries	Search :	<input type="text"/>	
Laporan	No	Kode Shift	Nama Shift	Mulai Shift	Selesai Shift	Aksi
Logout						Ubah Hapus

Gambar V.35 Rancangan Data *Shift*

(Sumber: Hasil Analisis Data, 2017)

Form tambahan *shift* adalah *form* untuk menambah data *shift* yang terdapat pada gambar V.36. Adapun penjelasan *field-field* dalam *form* tambahan *shift* sebagai berikut:

- Kode *Shift* : Berisi kode *shift*.
- Nama *Shift* : Berisi nama *shift*.
- Mulai *Shift* : Berisi mulai *shift*.
- Selesai *shift* : Berisi selesai *shift*.

PT RMA	
Beranda	<p>Form Shift</p> <p>Kode Shift <input type="text"/></p> <p>Nama Shift <input type="text"/></p> <p>Mulai Shift <input type="text"/></p> <p>Selesai Shift <input type="text"/></p> <p><input type="button" value="Simpan"/> <input type="button" value="Batal"/></p>
Absensi	
Data Master ▼	
Data Shift	
Data Transaksi	
Laporan	
Logout	

Gambar V.36 Rancangan *Form* Tambahan *Shift*

(Sumber: Hasil Analisis Data, 2017)

6. Tampilan Data Lembur

Menu data lembur adalah *form* yang digunakan untuk mengelola data lembur. Rancangan *interface* dari *form* lembur dapat dilihat pada gambar V.37 dan gambar V.38.

PT Rekadaya Multi Adiprima																			
Beranda	<div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>Absen Hari Ini</p> <p>Show <input type="text" value="10"/> entries Search : <input type="text"/></p> <table border="1"> <thead> <tr> <th>No</th> <th>NIK</th> <th>Nama</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td><input type="button" value="Lembur"/></td> </tr> </tbody> </table> </div> <div style="width: 48%;"> <p>Lembur Hari Ini</p> <p>Show <input type="text" value="10"/> entries Search : <input type="text"/></p> <table border="1"> <thead> <tr> <th>No</th> <th>NIK</th> <th>Masuk</th> <th>Keluar</th> <th>Selisih</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> </div> </div>	No	NIK	Nama	Aksi				<input type="button" value="Lembur"/>	No	NIK	Masuk	Keluar	Selisih					
No		NIK	Nama	Aksi															
				<input type="button" value="Lembur"/>															
No		NIK	Masuk	Keluar	Selisih														
Absensi																			
Data Master ▼																			
Data Tunjangan																			
Data Transaksi																			
Laporan																			
Logout																			

Gambar V.37 Rancangan Data lembur

(Sumber: Hasil Analisis Data, 2017)

Form tambahan lembur adalah *form* untuk menambah data lembur yang terdapat pada gambar V.37.

PT Rekadaya Multi Adiprima	
Beranda	<p>Form Lembur</p> <p>NIK <input type="text"/></p> <p>Nama Karyawan <input type="text"/></p> <p>Jabatan <input type="text"/></p> <p>Jam Masuk <input type="text"/></p> <p>Jam Keluar <input type="text"/></p> <p><input type="button" value="Simpan"/> <input type="button" value="Batal"/></p>
Data Master ▼	
Data Lembur	
Logout	

Gambar V.38 Rancangan Tambah Data lembur

(Sumber: Hasil Analisis Data, 2017)

7. Tampilan Data Cuti

Menu data cuti adalah *form* yang digunakan untuk mengelola data cuti. Rancangan *interface* dari *form* cuti dapat dilihat pada gambar V.39.

PT Rekadaya Multi Adiprima															
Beranda	<p>Form Cuti</p> <p>NIP <input type="text"/></p> <p>Nama <input type="text"/></p> <p>Mulai Cuti <input type="text"/></p> <p>Selesai Cuti <input type="text"/></p> <p>Ket. Cuti <input type="text"/></p> <p><input type="button" value="Simpan"/> <input type="button" value="Batal"/></p> <table border="1"> <thead> <tr> <th>No</th> <th>NIP</th> <th>Nama</th> <th>Mulai Cuti</th> <th>Selesai Cuti</th> <th>Keterangan</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Ubah Hapus</td> </tr> </tbody> </table>	No	NIP	Nama	Mulai Cuti	Selesai Cuti	Keterangan	Aksi							Ubah Hapus
No		NIP	Nama	Mulai Cuti	Selesai Cuti	Keterangan	Aksi								
							Ubah Hapus								
Data Cuti ▼															
Cuti															
Logout															

Gambar V.39 Rancangan Data Cuti

(Sumber: Hasil Analisis Data, 2017)

8. Tampilan Absensi Satu Bulan

Menu data absensi adalah *form* yang digunakan untuk tampilan absen karyawan satu bulan. Rancangan *interface* dari *form* absen dapat dilihat pada gambar V.40. Berikut penjelasan *field-field* dalam *form* transaksi absensi sebagai berikut:

- No : Berisi nomor urutan data absensi.
- NIK : Berisi nomor induk karyawan.
- Nama : Berisi nama karyawan.
- Total : Berisi jumlah masuk karyawan selama satu bulan.

PT RMA							
Beranda	Rekap Absensi Bulanan						
Absensi							
Data Master	Bulan		Tahun				
Data Transaksi ▼	<input type="text"/>		<input type="text"/>		<input type="button" value="Filter"/>		
Rekap Absensi							
Laporan	No	NIK	Nama	Hadir	Sakit	Izin	Alfa
Logout							

Gambar V.40 Rancangan Data Absensi

(Sumber: Hasil Analisis Data, 2017)

9. Tampilan Rekap Lembur

Menu data Rekap lembur adalah *form* yang digunakan untuk tampilan lembur. Rancangan *interface* dari *form* lembur dapat dilihat pada gambar V.41. Berikut penjelasan *field-field* dalam *form* transaksi lembur sebagai berikut:

- NIK : Berisi nomor induk karyawan.
- Nama : Berisi nama karyawan.
- Total : Berisi jumlah jam lembur karyawan.
- Filter : Berisi pencarian lembur karyawan.

PT RMA													
Beranda	<p>Rekap Lembur Bulanan</p> <p>Bulan Tahun</p> <p><input type="text"/> <input type="text"/> <input type="button" value="Filter"/></p> <table border="1"> <thead> <tr> <th>No</th> <th>NIP</th> <th>Nama</th> <th>Total Lembur</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	No	NIP	Nama	Total Lembur								
No		NIP	Nama	Total Lembur									
Absensi													
Data Master													
Data Transaksi ▼													
Rekap Lembur													
Laporan													
Logout													

Gambar V.41 Rancangan Data Lembur

(Sumber: Hasil Analisis Data, 2017)

10. Verifikasi Rekap Absensi

Verifikasi rekap absensi adalah *form* yang digunakan untuk pimpinan untuk memverifikasi data rekap absensi karyawan selama sebulan untuk menyetujui dan dapat melakukan proses selanjutnya yaitu membuat laporan dan mengolah Pimpinan karyawan. Rancangan *interface* dari *form* verifikasi rekap absensi dapat dilihat pada gambar V.42. Berikut penjelasan penggunaan pada *form* ini:

PT RMA																									
Data Transaksi ▼	<p>Rekap Absensi Bulanan</p> <p>Bulan Tahun</p> <p><input type="text"/> <input type="text"/> <input type="button" value="Filter"/></p> <table border="1"> <thead> <tr> <th>No</th> <th>NIK</th> <th>Nama</th> <th>Hadir</th> <th>Sakit</th> <th>Izin</th> <th>Alfa</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> <p><input type="button" value="Verifikasi"/></p>	No	NIK	Nama	Hadir	Sakit	Izin	Alfa	Total																
No		NIK	Nama	Hadir	Sakit	Izin	Alfa	Total																	
Rekap Absensi																									
Laporan																									
Logout																									

Gambar V.42 Rancangan Validasi Rekap Absensi

(Sumber: Hasil Analisis Data, 2017)

11. Verifikasi Rekap Lembur

Verifikasi rekap lembur adalah *form* yang digunakan untuk pimpinan untuk memverifikasi data rekap lembur karyawan selama sebulan untuk menyetujui dan dapat melakukan proses selanjutnya yaitu membuat laporan dan mengolah penggajian karyawan. Rancangan *interface* dari *form* verifikasi rekap lembur dapat dilihat pada gambar V.43. Berikut penjelasan penggunaan pada *form* ini:

PT RMA																
Data Transaksi▼	Rekap Lembur Bulanan															
Rekap Lembur																
Laporan	Bulan		Tahun													
Logout	<input type="text"/>	<input type="text"/>	<input type="button" value="Filter"/>													
	<table border="1"> <thead> <tr> <th>No</th> <th>NIP</th> <th>Nama</th> <th>Total Lembur</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				No	NIP	Nama	Total Lembur								
No	NIP	Nama	Total Lembur													
	<input type="button" value="Verifikasi"/>															

Gambar V.43 Rancangan Validasi Rekap Absensi
(Sumber: Hasil Analisis Data, 2017)

5.4.4 Implementasi Sistem

Untuk dapat menggunakan aplikasi, tentunya diperlukan suatu spesifikasi perangkat keras (*hardware*) ataupun perangkat lunak (*software*) yang mendukung agar aplikasi dapat berjalan dengan baik. Adapun spesifikasinya sebagai berikut:

1. Kebutuhan *Hardware*

- a. *Processor* : Intel Core i5.
- b. *Main Memory* : 4 GB
- c. *Hard Disk* : 389 GB

2. Kebutuhan *Software*

- a. *Sistem Operasi* : *Microsoft Windows 8*
- b. *Web Server* : *Apache*
- c. *Database Server* : *MySQL 5.6.21*
- d. *Design Interface* : *PHP 5.6.3*
- e. *Web Browser* : *Mozilla firefox*

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengumpulan, pengolahan dan analisis data yang telah dilakukan dalam penelitian ini dapat disimpulkan bahwa:

1. Proses absensi karyawan secara manual dapat berkembang dengan adanya aplikasi ini, sehingga proses absensi berjalan dengan terkomputerisasi.
2. Sistem informasi absensi dapat membantu bagian HRD dalam merekap dan mengolah data menjadi mudah.
3. Sistem informasi ini dapat mengelola laporan absensi karyawan dengan baik sehingga mudah dalam pengecekan oleh pimpinan.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem informasi absensi karyawan ini selanjutnya adalah sebagai berikut:

1. Penelitian ini baru sebatas membuat sistem terprogram. Diharapkan adanya pengembangan lebih lanjut dari sistem informasi yang dirancang oleh pihak perusahaan.
2. Sebaiknya dilakukan pemeliharaan aplikasi secara berkala, sehingga aplikasi dapat berjalan dengan baik.
3. Diharapkan bagi pembaca tugas akhir ini dapat membuat sistem informasi absensi karyawan menjadi sistem yang lebih luas dengan memperbaharuihnya.

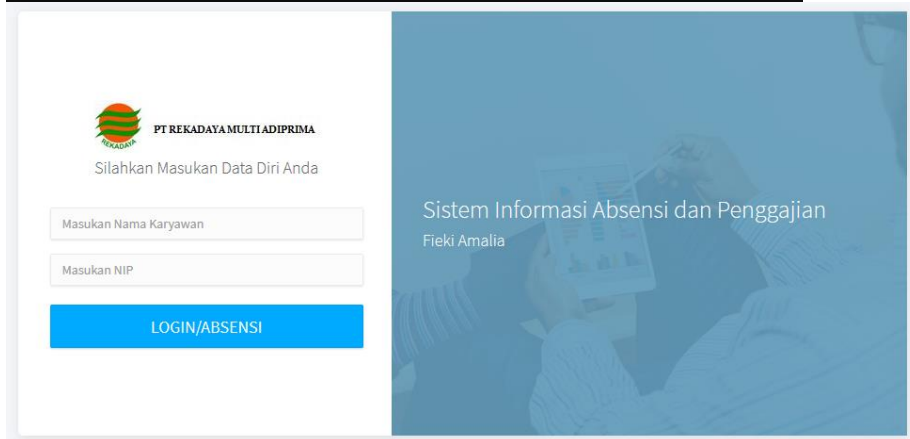
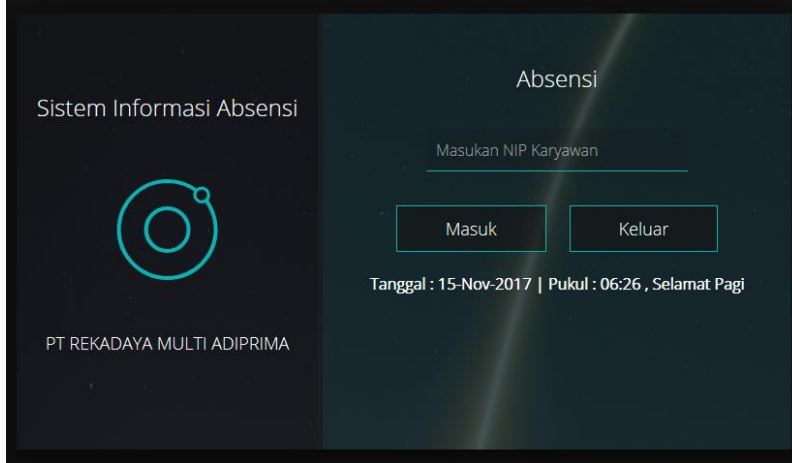
DAFTAR PUSTAKA

- Arsip PT Rekadaya Multi Adiprima.
- Barakbah, Ali Ridho, Karlita, Tita dan Ahsan, Syauqi Ahmad. 2013. Logika dan Algoritma. Surabaya: Politeknik Elektronika Negeri Surabaya.
- Dennis, Alan, Barbara Haley W., Robert M. Roth, 2012. *Systems Analys & Design With UML*, 5th Ed. Wiley
- Departemen Pendidikan Nasional. 2016. Kamus Besar Bahasa Indonesia Pusat Bahasa. Edisi Kelima. Jakarta: PT Gramedia Pustaka Utama.
- Djahir, Yulia dan Pratita, Dewi. 2014. *Bahan Ajar Sistem Informasi Manajemen*. Edisi ke-1. Yogyakarta: Depublish.
- Fathansyah. 2012. Basis Data. Bandung: Informatika.
- Febriani. 2015. *Analisis dan Perancangan Sistem*. Jakarta: Gunadarma.
- Hutahaean, Jeperson. 2014. *Konsep Sistem Informasi*. Cetakan ke-1. Yogyakarta: Depublish.
- Hartono, Jogiyanto, M. 2005. *Analisis dan Desain*. Yogyakarta: Andi.
- Hartono, Jogiyanto, M. 2002. *Analisa dan Desain Sistem Informasi: Pendekatan terstruktur teori dan praktis aplikasi bisnis*. Yogyakarta: Andi.
- Kadir, A. 2008. *Dasar Pemrograman WEB Dinamis Menggunakan PHP*. Yogyakarta: C.V Andi Offset.
- McLeod, M. 2001. *Sistem Informasi Manajemen: Versi Bahasa Indonesia*. Jakarta: Prenhallindo.
- Nugroho, A. 2007. *Konsep Pengembangan Sistem Basis Data*. Bandung: Informatika.
- Panggabean, S., Mutiara. 2004. *Manajemen Sumber Daya Manusia*. Bogor: Ghalia
- Rosa, A.S. dan Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Cetakan Pertama. Bandung: Modula.
- Rusdiana dan Moch. Irfan. 2014. *Sistem Informasi Manajemen. Cetakan I*. Bandung: CV Pustaka Setia.

- Simonna, E. 2009. Definisi Absensi. <http://simonna-erna.blogspot.com>. (Tanggal Akses: 6 Januari 2017).
- Sutabri, Tata. 2014. *Analisis Sistem Informasi*. Yogyakarta: CV Andi Offset.
- Thomas. 2002. *Ekonomi Sumber Daya Manusia*. Jakarta: Raja Grafindo Persada.
- Yakub. 2012. *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.
- . *Pengertian Absensi*. <http://www.kajianpustaka.com//pengertian-dan-jenis-jenis-absensi.html>. (Tanggal Akses: 11 Maret 2017).

LAMPIRAN

Form Absen



Absensi Hari Ini

Q Data Karyawan

Show 10 entries Search:

No	Nik	Nama Karyawan	Tanggal	Keterangan
1	N007	Narto	2017-11-13	Hadir

Showing 1 to 1 of 1 entries Previous 1 Next

Form Mengolah Absensi

NIP	N001
Nama Karyawan	fiki
Jenis Kelamin	Perempuan
Jabatan	Staf IT
Pendidikan	D4 - Sistem Informasi
Absensi	Pilih Status ▼
<input type="button" value="Simpan"/> <input type="button" value="Batal"/>	

Rekap Absensi

Bulan Januari **Tahun** 2017

Show 10 entries Search:

No	Nik	Nama Karyawan	Hadir	Sakit	Izin	Alpha	Total
1	N001	fiki	15	0	0	1	16
2	N002	Wahyudi	13	0	0	2	16
3	N003	Gio	16	0	0	0	16
4	N004	Fayadh	16	0	0	0	16
5	N005	Ari	16	0	0	0	16
6	N006	lili	16	0	0	0	16
7	N007	Narto	13	2	1	0	16

Rekap Lembur

Bulan Januari **Tahun** 2017

Show 10 entries Search:

No	Nik	Nama Karyawan	Total Lembur
1	N007	Narto	4

Showing 1 to 1 of 1 entries Previous 1 Next

Verifikasi Rekap

Bulan Januari **Tahun** 2017 Filter

Show 10 entries Search:

No	Nik	Nama Karyawan	Hadir	Sakit	Izin	Alpha	Total
1	N001	fiki	3	1	0	0	4
2	N002	Wahyudi	1	1	0	0	2
3	N003	Gio	1	0	0	1	2
4	N004	Fayadh	3	0	0	0	3
5	N005	Ari	1	1	0	0	2
6	N006	lili	3	0	1	0	4
7	N007	Narto	5	0	1	0	6

Absen Hari Ini

Show 10 entries Search:

No	Nik	Nama	Lembur
1	N007	Narto	Lembur

Showing 1 to 1 of 1 entries Previous 1 Next

Lembur Hari Ini

Show 10 entries Search:

No	Nik	Masuk	Keluar	Selisih
1	N007	17:00:00	19:00:00	2 Jam

Showing 1 to 1 of 1 entries Previous 1 Next

Data Cuti

+ Tambah

Show 10 entries Search:

No	Kode Cuti	Tanggal Permintaan Cuti	Aksi
No data available in table			

Showing 0 to 0 of 0 entries Previous Next

Source Code

```
<?php
session_start();
if(!isset($_SESSION['nm_karyawan'])&&!isset($_SESSION['NIP'])){\
```

```

        header('location:akses/login.php');}
    else{
        $jabatan=$_SESSION['jabatan'];
        $nm_karyawan=$_SESSION['nm_karyawan'];}
    ?>
<!doctype html>
<html lang="en">

<head>
    <title><?php echo $nm_karyawan; ?> | RMA </title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=0">
    <!-- VENDOR CSS -->
    <link rel="stylesheet"
href="assets/vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="assets/vendor/font-awesome/css/font-
awesome.min.css">
    <link rel="stylesheet" href="assets/vendor/linearicons/style.css">
    <link rel="stylesheet" href="assets/vendor/chartist/css/chartist-
custom.css">
    <!-- MAIN CSS -->
    <link rel="stylesheet" href="assets/css/main.css">
    <!-- FOR DEMO PURPOSES ONLY. You should remove this in your
project -->
    <link rel="stylesheet" href="assets/css/demo.css">

    <link rel="stylesheet" type="text/css"
href="assets/DataTables/media/css/jquery.dataTables.css">
    <link rel="stylesheet" type="text/css"
href="assets/DataTables/media/css/dataTables.bootstrap.css">
    <!-- GOOGLE FONTS -->
    <link
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,70
0" rel="stylesheet">
    <!-- ICONS -->
    <link rel="apple-touch-icon" sizes="5x10" href="assets/img/rb.JPG">
    <link rel="icon" type="image/JPG" sizes="66x66"
href="assets/img/RRR.JPG">--
</head>

<body>
    <!-- WRAPPER -->
    <div id="wrapper">
        <!-- NAVBAR -->

```

```

        <nav class="navbar navbar-default navbar-fixed-top">
            <div class="brand">
                <a href="index.html"></a>
            </div>
            <div class="container-fluid">
                <div class="navbar-btn">
                    <button type="button" class="btn-toggle-
fullwidth"><i class="lnr lnr-arrow-left-circle"></i></button>
                </div>

            </div>
        </nav>
        <!-- END NAVBAR -->
        <!-- LEFT SIDEBAR -->
        <?php include 'menu.php'; ?>
        <!-- END LEFT SIDEBAR -->
        <!-- MAIN -->

        <?php
if(isset($_GET['menu'])){
$menu=$_GET['menu'];
switch($menu){
case('home');include('home.php');break;
case('jabatan');include('master/jabatan/jabatan.php');break;
    case('t_jabatan');include('master/jabatan/tambah.php');break;
    case('e_jabatan');include('master/jabatan/edit.php');break;
case('karyawan');include('master/karyawan/karyawan.php');break;
    case('t_karyawan');include('master/karyawan/tambah.php');break;
    case('e_karyawan');include('master/karyawan/edit.php');break;
case('shift');include('master/shift/shift.php');break;
    case('t_shift');include('master/shift/tambah.php');break;
    case('e_shift');include('master/shift/edit.php');break;
case('a_harian');include('absensi/harian/a_harian.php');break;
    case('a_karyawan');include('absensi/harian/karyawan.php');break;
    case('i_absen');include('absensi/harian/i_absensi.php');break;
    case('a_absen');include('absensi/harian/a_absensi.php');break;
case('r_harian');include('absensi/rekap/harian/a_karyawan.php');break;

case('o_absensi');include('absensi/rekap/harian/o_absensi.php');break;
    case('s_harian');include('absensi/rekap/harian/s_harian.php');break;

case('sd_harian');include('absensi/rekap/harian/r_harian.php');break;

case('ra_bulanan');include('absensi/rekap/bulanan/ra_bulanan.php');break;

```

```

        //case('i_absen');include('absensi/harian/i_absensi.php');break;

        case('l_harian');include('lembur/harian/l_harian.php');break;
        case('i_lembur');include('lembur/harian/f_lembur.php');break;

        case('rr_harian');include('lembur/rekap/bulanan/b_karyawan.php');break;

        case('ss_harian');include('lembur/rekap/bulanan/rr_harian.php');break;

        case('tes_lembur');include('lembur/rekap/bulanan/tes_lembur.php');break;

        case('sdd_harian');include('lembur/rekap/bulanan/rr_harian.php');break;

    case('lem_bulanan');include('lembur/rekap/bulanan/lem_bulanan.php');break;

        case('cuti');include('cuti/d_cuti/cuti.php');break;
        case('t_cuti');include('cuti/d_cuti/tambah.php');break;
        case('td_cuti');include('cuti/d_cuti/tambah_d.php');break;
        case('m_cuti');include('cuti/master/m_cuti.php');break;
        case('t_m_cuti');include('cuti/master/tambah.php');break;

    }
}
?>

<!-- END MAIN -->
<div class="clearfix"></div>
<footer>
    <div class="container-fluid">
        <p class="copyright">&copy; 2017 <a
href="https://www.themeined.com" target="_blank">Theme I Need</a>. All
Rights Reserved.</p>
    </div>
</footer>
</div>
<!-- END WRAPPER -->
<!-- Javascript -->
<script type="text/javascript"
src="assets/DataTables/media/js/jquery.js"></script>
    <script src="assets/vendor/jquery/jquery.min.js"></script>
    <script type="text/javascript"
src="assets/DataTables/media/js/jquery.dataTables.js"></script>

    <script src="assets/vendor/bootstrap/js/bootstrap.min.js"></script>
    <script src="assets/vendor/jquery-
slimscroll/jquery.slimscroll.min.js"></script>

```

```

    <script src="assets/vendor/jquery.easy-pie-
chart/jquery.easypiechart.min.js"></script>
    <script src="assets/vendor/chartist/js/chartist.min.js"></script>
    <script src="assets/scripts/klorofil-common.js"></script>
    <script type="text/javascript">
    $(document).ready(function(){
    $('#provinsi').DataTable();
    });
</script>
<script type="text/javascript">
$(document).ready(function(){
    $('#absen').DataTable();
    });
</script>

<script>
$(function() {
    var data, options;

    // headline charts
    data = {
        labels: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'],
        series: [
            [23, 29, 24, 40, 25, 24, 35],
            [14, 25, 18, 34, 29, 38, 44],
        ]
    };

    options = {
        height: 300,
        showArea: true,
        showLine: false,
        showPoint: false,
        fullWidth: true,
        axisX: {
            showGrid: false
        },
        lineSmooth: false,
    };

    new Chartist.Line('#headline-chart', data, options);

    // visits trend charts
    data = {

```

```

        labels: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
'Sep', 'Oct', 'Nov', 'Dec'],
        series: [{
            name: 'series-real',
            data: [200, 380, 350, 320, 410, 450, 570, 400, 555,
620, 750, 900],
        }, {
            name: 'series-projection',
            data: [240, 350, 360, 380, 400, 450, 480, 523, 555,
600, 700, 800],
        }]
    };

    options = {
        fullWidth: true,
        lineSmooth: false,
        height: "270px",
        low: 0,
        high: 'auto',
        series: {
            'series-projection': {
                showArea: true,
                showPoint: false,
                showLine: false
            },
        },
        axisX: {
            showGrid: false,
        },
        axisY: {
            showGrid: false,
            onlyInteger: true,
            offset: 0,
        },
        chartPadding: {
            left: 20,
            right: 20
        }
    };

    new Chartist.Line('#visits-trends-chart', data, options);

    // visits chart
    data = {

```

```

        labels: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'],
        series: [
            [6384, 6342, 5437, 2764, 3958, 5068, 7654]
        ]
    };

    options = {
        height: 300,
        axisX: {
            showGrid: false
        },
    };

    new Chartist.Bar('#visits-chart', data, options);

    // real-time pie chart
    var sysLoad = $('#system-load').easyPieChart({
        size: 130,
        barColor: function(percent) {
            return "rgb(" + Math.round(200 * percent / 100) + ",
" + Math.round(200 * (1.1 - percent / 100)) + ", 0)";
        },
        trackColor: 'rgba(245, 245, 245, 0.8)',
        scaleColor: false,
        lineWidth: 5,
        lineCap: "square",
        animate: 800
    });

    var updateInterval = 3000; // in milliseconds

    setInterval(function() {
        var randomVal;
        randomVal = getRandomInt(0, 100);

        sysLoad.data('easyPieChart').update(randomVal);
        sysLoad.find('.percent').text(randomVal);
    }, updateInterval);

    function getRandomInt(min, max) {
        return Math.floor(Math.random() * (max - min + 1)) + min;
    }

});
</script>

```

</body>

</html>