

RANCANG BANGUN SISTEM *INFORMASI*
ABSENSI FINGER PRINT* KARYAWAN BERBASIS *DESKTOP
MENGGUNAKAN *JAVA 2 STANDART EDITION (J2SE)* DAN
***MARIADB 10.1.9* PADA PT COCA COLA DISTRIBUTION**
INDONESIA

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Penyelesaian
Program Sarjana Terapan Program Studi Sistem Informasi Industri Otomotif
Pada Politeknik STMI Jakarta

OLEH

AKBAR

1412002



POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2018

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI *ABSENSI FINGERPRINT KARYAWAN* BERBASIS *DEKSTOP* MENGGUNAKAN *JAVA 2 STANDARD EDITION (J2SE)* DAN *MARIADB 10.1.9* PADA PT COCA COLA DISTRIBUTION INDONESIA.

Disusun Oleh :
Nama : Akbar
Nim : 1412002
Program Studi : Sistem Informasi

Telah Diuji Oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem Informasi Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia pada hari.

Jakarta, 19 September 2018

Dosen Pembimbing

Ketua Penguji

Drs. Jacob Saragih, MM
NIP. 195404281986031002

Lucky Heriyanto, ST, MTI
NIP. 197811212009012003

Dosen Penguji

Dosen Penguji

Noveriza Yuliasari, S.Si, MT
NIP. 197811212009012003

Ulil Hamida, ST, MT
NIP. 198103272005022001

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI *ABSENSI FINGERPRINT KARYAWAN* BERBASIS *DEKSTOP* MENGGUNAKAN *JAVA 2 STANDARD EDITION (J2SE)* DAN *MARIADB 10.1.9* PADA PT COCA COLA DISTRIBUTION INDONESIA.

Disusun Oleh :
Nama : Akbar
Nim : 1412002
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 2 Maret 2018
Tanggal Sidang : 10 September 2018
Tanggal Lulus : 10 September 2018

Jakarta, 19 September 2018

Menyetujui

Dosen Pembimbing

Drs. Jacob Saragih, MM
NIP. 195404281986031002

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI *ABSENSI FINGERPRINT KARYAWAN* BERBASIS *DEKSTOP* MENGGUNAKAN *JAVA 2 STANDARD EDITION (J2SE)* DAN *MARIADB 10.1.9* PADA PT COCA COLA DISTRIBUTION INDONESIA.

Disusun Oleh :
Nama : Akbar
Nim : 1412002
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 2 Maret 2018
Tanggal Sidang : 10 September 2018
Tanggal Lulus : 10 September 2018

Jakarta, 19 September 2018

Menyetujui

Asisten Dosen Pembimbing

Ahmad Juniar, S.Kom, MT
NIP. 197906052006041002

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Akbar

Nim : 1412002

Program Studi : Sistem Informasi Industri Otomotif

Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul:

“RANCANG BANGUN SISTEM INFORMASI *ABSENSI FINGER PRINT* KARYAWANBERBASIS *DESKTOP* MENGGUNAKAN JAVA NETBEANS 8.0 DAN MARIADB 10.1.9 PADA PT COCA COLA DISTRIBUTION INDONESIA ”.

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survei lapangan, dibantu oleh dosen pembimbing maupun asisten dosen pembimbing, serta buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya Tugas Akhir ini.
- **Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- **Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini.

Jika terbukti saya tidak memenuhi apa yang telah dinyatakan di atas, maka saya bersedia karya Tugas Akhir ini dibatalkan.

Jakarta, September 2018

Yang Membuat Pernyataan

Akbar

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Akbar
Nim : 1412002
Judul TA : Rancang Bangun Sistem Informasi Absensi *Fingerprint* Karyawan Berbasis Desktop Menggunakan Java 2 Standard Edition (J2SE) dan MariaDB 10.1.9 Pada PT Coca Cola Distribution Indonesia
Pembimbing : Drs. Jacob Saragih, MM
Asisten Pembimbing : Ahmad Juniar, S.Kom, MT

Tanggal	Bab	Keterangan	Paraf
22-Mei-17	Bab I, II	Bimbingan Bab I dan II	
06-Jul-17	Bab I, II, III	Revisi Bab I dan Bab II , Bimbingan Bab III	
13-Jul-17	Bab I-IV	Revisi Bab III, Bimbingan Bab IV	
20-Jul-17	Bab IV	Bimbingan Bab IV	
27-Jul-17	Bab IV	Revisi Bab IV	
10-Agu-17	Bab V	Bimbingan Bab V	
25-Agu-17	Bab V, Bab VI	Revisi Bab V dan Bimbingan Bab VI	
1-Sep-17	Bab I – VI	Pemeriksaan Laporan TA Bab I – VI	
30-Okt-17	Pemrograman	Demo Program	

Mengetahui,
Ka Prodi
Sistem Informasi Industri Otomotif

Pembimbing

Drs. Jacob Saragih, MM
NIP : 195404281986031002

Drs. Jacob Saragih, MM
NIP : 195404281986031002

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Akbar
Nim : 1412002
Judul TA : Rancang Bangun Sistem Informasi Absensi *Fingerprint* Karyawan Berbasis Desktop Menggunakan Java 2 Standard Edition (J2SE) dan MariaDB 10.1.9 Pada PT Coca Cola Distribution Indonesia
Pembimbing : Drs. Jacob Saragih, MM
Asisten Pembimbing : Ahmad Juniar, S.Kom, MT

Tanggal	Bab	Keterangan	Paraf
15-Mei-17	Bab I	Bimbingan Bab I	
09-Jun-17	Bab II	Revisi Bab I, Bimbingan Bab II	
29-Agu-17	Bab III	Revisi Bab II, Bimbingan Bab III	
03-Sep-17	Bab IV	Revisi Bab III, Bimbingan Bab IV	
18-Sep-17	Bab IV	Revisi Bab IV	
06-Okt-17	Bab V	Bimbingan Bab V	
12-Okt-17	Bab V	Revisi Bab V	

24-Okt-17	Bab VI	Bab VI	
26-Okt-17	Demo Program	Demo Program	
27-Okt-17	Pemeriksaan Susunan	Pemeriksaan Susunan Laporan Tugas Akhir	

Mengetahui,
Ka Prodi
Pembimbing Sistem Informasi Industri Otomotif

Asisten

Drs. Jacob Saragih, MM
S.Kom, MT NIP : 195404281986031002
NIP : 197906052006041002

Ahmad Juniar,

ABSTRAK

PT Coca Cola Distribtuion Indonesia adalah perusahaan yang bergerak dalam bidang jasa dan manufaktur. Perusahaan tersebut bergerak dalam produksi minuman berkarbonasi dan dsitribusi ke pelanggan ini berlokasi di Cakung, Jakarta Timur. Perusahaan yang dikenal sejak tahun 1927 ini menjadi pioner dalam produksi minuman berkarbonasi di Indonesia. Adapun pemasalahan yang ditemukan pada PT Coca Cola Distribtuion Indonesia adalah dalam hal sistem absensi karyawan. Sistem absensi yang berjalan masih dilakukan secara manual, yaitu dengan melakukan kegiatan absensi menggunakan kartu absensi pada absensi *amano*. Dengan sistem absensi manual yang perusahaan terapkan selama ini, bisa saja terjadi tindak kecurangan yang dilakukan oleh para karyawan yaitu dengan menitipkan kartu absensinya kepada sesama karyawan atau tidak sesuai dengan kedatangannya, sehingga perusahaan tidak dapat mengontrol keterlambatan para karyawan yang datang melebihi waktu yang telah ditetapkan oleh perusahaan. Serta kesulitan dalam merekap data kehadiran karyawan oleh divisi HRD. Untuk mengatasi permasalahan tersebut, maka perlu dilakukan pengembangan sistem informasi absensi yang lebih modern dengan memperhatikan sisi kejujuran dan dapat menghasilkan data yang lebih akurat. Metodologi pengembangan sistem yang digunakan pada penelitian ini adalah *evolutionary prototype*. Analisis perancangan sistem berorientasi objek dengan menggunakan *tools* pemodelan *Unified Modelling Language* (UML). Pemodelan data menggunakan *entity relationship diagram* (ERD) dan kamus data sebagai pendefinisian data yang mengalir dalam sistem secara lengkap. Sistem informasi ini dibangun menggunakan *Java 2 Standard Edition* (J2SE)) sebagai bahasa pemrograman dan MariaDB 10.1.9 sebagai perangkat lunak manajemen basis datanya. Sistem ini diharapkan dapat membantu HRD dalam proses kontrol kehadiran karyawan dan rekapitulasi data kehadiran, pengarsipan data karyawan tanpa kertas dan mengurangi kecurangan penitipan absen antar karyawan.

Kata Kunci : Absensi, *Fingerprint*, J2SE, Netbeans, MariaDB, *evolutionary prototype*, *Unified Modelling Language* (UML), *Entity Relationship Diagram* (ERD), Kamus Data.

KATA PENGANTAR

Puji syukur penulis panjatkan bagi Tuhan Yang Maha Esa, yang atas berkat dan kasih-Nya sehingga penulis mampu menyelesaikan Tugas Akhir dengan judul **“Rancang Bangun Sistem Informasi Absensi *Fingerprint* Karyawan Berbasis Desktop Menggunakan Java 2 Standart Edition dan MariaDB 10.1.9 pada PT Coca Cola Distribution Indonesia”**.

Penulisan Tugas Akhir ini bertujuan untuk memenuhi sebagian syarat penyelesaian program sarjana terapan pada program studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Terselesaikannya 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. Bapak Dr. Mustofa, ST, MT, selaku Direktur Politeknik STMI Jakarta.
2. Bapak Drs. Jacob Saragih, MM, selaku Ketua Program Studi Sistem Informasi Industri Otomotif dan dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
3. Bapak Ahmad Juniar, S.Kom, MT, selaku asisten dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
4. Bapak Zulkarnaen Siregar (Alm) dan Ibu Masturi, selaku Orang Tua penulis yang selalu mendoakan, memberikan motivasi dan pengorbanannya baik segi moril, materi kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik.

5. Kakakku, Aldisa, Astrya, Anggi dan juga adikku Andrian selaku saudara penulis yang selalu memberikan motivasi dan dukungan moril serta doa sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
6. Sdri Ade Kurniawati, Amd. RO yang senantiasa memberikan motivasi serta doa sehingga Tugas Akhir ini dapat diselesaikan dengan baik..
7. Bapak Adhar Novrizal Hamami dan Bapak Teuku Fazlan selaku pembimbing di PT Coca Cola Distribution Indonesia serta seluruh pekerja khususnya divisi *HRD* yang telah membimbing dan memberikan informasi yang berguna kepada penulis dalam penyusunan Tugas Akhir.
8. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
9. Seluruh teman-teman jurusan Sistem Informasi angkatan 2012, khususnya Riyan Alfian, Laura, Aulia Nurul Fitria, Tania dan Riza Anisha yang telah memberikan semangat dan motivasi kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
10. Serta semua pihak yang telah membantu penulis dan yang tidak dapat disebutkan satu per satu.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan ketulusan semua pihak yang telah membantu dan penulis berharap semoga Tugas Akhir ini dapat memberikan manfaat dan informasi yang berguna bagi pembaca. Terima kasih.

Jakarta, 7 September 2018

Penulis,
Akbar

DAFTAR ISI

Halaman

HALAMAN JUDUL	
LEMBAR PENGESAHAN	
LEMBAR PERSETUJUAN DOSEN PEMBIMBING	
LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING	
LEMBAR BIMBINGAN DENGAN DOSEN PEMBIMBING	
LEMBAR BIMBINGAN DENGAN ASISTEN DOSEN PEMBIMBING	
LEMBAR PERNYATAAN KEASLIAN	
ABSTRAK	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xviii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Pokok Permasalahan	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah.....	2
1.5. Manfaat Tugas Akhir	3
1.6. Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1. Rancang Bangun	5
2.2. Konsep Dasar Sistem Informasi.....	5
2.3 Konsep Dasar Absensi dan Komponen Gaji	5
2.3.1 Pengertian Absensi	5

2.3.2	Lembur	6
2.3.2.1	Tarif Upah Lembur.....	7
2.3.2.2	Menghitung Upah Lembur	8
2.3.3	Cuti	8
2.4	Biometrik	9
2.5	Sidik Jari (<i>Fingerprint</i>)	10
2.5.1	Metode Pengenalan Sidik Jari	11
2.5.2	Metode <i>Minutiae</i>	12
2.5.3	<i>Fingerprint Authentication</i>	13
2.5.4	Alur Data Mesin Sidik Jari	14
2.6	Manajemen Sumber Daya Manusia	16
2.6.1	Pengertian Manajemen Sumber Daya Manusia	16
2.6.2	Fungsi Manajemen Sumber Daya Manusia	16
2.6.3	Tujuan Manajemen Sumber Daya Manusia	18
2.7	Analisis Sistem Informasi	20
2.8	Metode Pengembangan Sistem	21
2.8.1	Model <i>Prototype</i>	23
2.8.2	<i>Evolutionary Prototype</i>	25
2.9	<i>Flowmap</i> dan <i>Flowchart</i>	26
2.10	Jenis-jenis <i>Flowchart</i>	26
2.10.1	<i>System Flowchart</i>	28
2.10.2	<i>Document Flowchart</i>	28
2.10.3	<i>Schematic Flowchart</i>	28
2.10.4	<i>Program Flowchart</i>	29
2.10.5	<i>Process Flowchart</i>	30
2.11	<i>Unified Modelling Language (UML)</i>	30
2.11.1	<i>Usecase Diagram</i>	31
2.11.2	<i>Activity Diagram</i>	33
2.11.3	<i>Sequence Diagram</i>	34

2.11.4	<i>Class Diagram</i>	36
2.11.5	<i>Component Diagram</i>	37
2.11.6	<i>Deployment Diagram</i>	39
2.12	Perancangan Database.....	40
2.12.1	Pengertian Database	40
2.12.2	Normalisasi	41
2.12.3	<i>Relational Database Management System</i>	41
2.12.4	<i>Entity Relationship Diagram (ERD)</i>	41
2.13	Kamus Data.....	45
2.14	<i>Hierarchy plus Input-Proses-Output (HIPO)</i>	47
2.14.1	Diagram dalam Paket HIPO	48
2.15	Bahasa Pemrograman.....	51
2.15.1	Netbeans	51
2.15.1.1	Netbeans 8.2	51
2.15.1.2	Kelebihan dan Kekurangan.....	55
2.15.1.3	<i>Java Development Kit (JDK)</i>	57
2.15.1.4	<i>Java Virtual Machine (JVM)</i>	57
2.15.1.5	<i>Java Runtime Environment (JRE)</i>	57
2.15.2	Xampp	57
2.15.3	iReport	58
2.16	MariaDB	59
2.16.1	Keistimewaan MariaDB	59
2.16.2	Tipe Data	61
BAB III	METODOLOGI PENELITIAN	63
3.1.	Metode Penelitian.....	63
3.2.	Identifikasi Masalah	63
3.3.	Pengumpulan Data	63
3.4.	Metode Pengembangan Sistem	65
3.5.	Kerangka Pengembangan Sistem.....	65

BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA	70
4.1.	Sejarah Perusahaan.....	70
4.2.	Profil Perusahaan	71
4.3.	Visi dan Misi Perusahaan	72
4.4.	Struktur Organisasi Perusahaan	72
4.5.	Divisi <i>Business System Manager</i>	76
4.6.	Sistem Absensi Karyawan Berjalan.....	79
4.6.1.	Dokumen Pada Absensi Karyawan Berjalan	81
4.6.2.	Flowmap absensi berjalan	85
4.7.	<i>Use case</i> Diagram yang Berjalan	86
BAB V	ANALISIS DAN PEMBAHASAN	88
5.1.	Analisis Kebutuhan Sistem	88
5.2.	Perancangan Flowmap Sistem Usulan.....	89
5.3.	Analisis dan Perancangan Sistem Usulan	91
5.3.1.	<i>Usecase Diagram</i>	92
5.3.2.	<i>Activity Diagram</i>	96
5.4.	<i>Class Diagram</i>	101
5.5.	<i>Sequence Diagram</i>	101
5.6.	<i>Entity Relationship Diagram</i>	107
5.7.	Kamus Data	108
5.8.	<i>Deployment Diagram</i>	111
5.9.	Perancangan Program	111
5.9.1.	HIPO (<i>Hierarchy plus Input-Process-Output</i>).....	111
5.9.2.	<i>Flowchart</i> Pengoperasian Program	112
5.9.3.	Perancangan <i>Interface</i> Program	114
5.10.	Implementasi Sistem <i>Software</i> dan <i>Hardware</i>	120
BAB VI	KESIMPULAN DAN SARAN	122
6.1.	Kesimpulan	122
6.2.	Saran	122

DAFTAR PUSTAKA	123
-----------------------------	------------

LAMPIRAN A

LAMPIRAN B

DAFTAR GAMBAR

	Halaman
Gambar II.1	Fitur Guratan Sidik Jari..... 12
Gambar II.2	Pola Sidik Jari 13
Gambar II.3	Metode <i>Minutiae</i> 13
Gambar II.4	Hasil Pengolahan Metode <i>Minutiae</i> 14
Gambar II.5	Contoh <i>Fingerprint Image View</i> 15
Gambar II.6	Contoh <i>Fingerprint Minutiae View</i> 16
Gambar II.7	Antarmuka Netbeans 8.2..... 20
Gambar II.8	Antarmuka Pembuatan Project Baru..... 21
Gambar II.9	Antarmuka Pemilihan Lokasi Penyimpanan <i>Source Code</i> .21
Gambar II.10	<i>GUI Builder</i> di Netbeans 8.2..... 22
Gambar II.11	<i>Navigator Windows</i> di Netbeans 8.2..... 22
Gambar II.12	<i>Pallette Windows</i> di Netbeans 8.2..... 23
Gambar II.13	<i>Properties Windows</i> di Netbeans 8.2 23
Gambar II.14	<i>Source Area</i> di Netbeans 8.2..... 24
Gambar II.15	Ilustrasi Model <i>Prototype</i> 32
Gambar II.16	<i>Evolutionary Prototype Model</i> 33
Gambar II.17	Klasifikasi Diagram UML..... 39
Gambar II.18	Contoh <i>Use Case Diagram</i> 40
Gambar II.19	Contoh <i>Activity Diagram</i> 42
Gambar II.20	Contoh <i>Sequence Diagram</i> 43
Gambar II.21	Contoh <i>Class Diagram</i> 45
Gambar II.22	Contoh <i>Component Diagram</i> 46
Gambar II.23	Contoh <i>Deployment Diagram</i> 47
Gambar II.24	<i>Visual Table of Contents</i> 49

Gambar II.25	<i>Overview Diagram</i>	50
Gambar II.26	<i>Detail Diagram</i>	51
Gambar II.27	Contoh Bagan ERD.....	56
Gambar III.1	Kerangka Pengembangan Sistem.....	68
Gambar IV.1	Struktur Organisasi Perusahaan	72
Gambar IV.2	Struktur Organisasi Divisi <i>Business System Manager</i>	77
Gambar IV.3	Kartu Absen Pegawai	80
Gambar IV.4	<i>Form Cuti</i>	81
Gambar IV.5	<i>Form Lembur</i>	82
Gambar IV.6	<i>Flowmap Sistem Berjalan</i>	85
Gambar IV.7	<i>Usecase Proses Berjalan</i>	86
Gambar V.1	<i>Flowmap Sistem Absensi Pegawai Usulan</i>	90
Gambar V.2	<i>Usecase Diagram Sistem Absensi Pegawai Usulan</i>	92
Gambar V.3	<i>Activity Diagram Login Program</i>	96
Gambar V.4	<i>Activity Diagram Manajemen Karyawan</i>	97
Gambar V.5	<i>Activity Diagram Manajemen Lembur</i>	98
Gambar V.6	<i>Activity Diagram Leave</i>	99
Gambar V.7	<i>Activity Diagram Absen</i>	100
Gambar V.8	<i>Activity Diagram Laporan Kehadiran</i>	100
Gambar V.9	<i>Class Diagram Sistem Absensi Karyawan Usulan</i>	101
Gambar V.10	<i>Sequence Diagram Login Admin</i>	102
Gambar V.11	<i>Sequence Diagram Manajemen Karyawan</i>	103
Gambar V.12	<i>Sequence Diagram Proses Data Lembur</i>	104
Gambar V.13	<i>Sequence Diagram Proses Data Leave</i>	105
Gambar V.14	<i>Sequence Diagram Proses Absensi Kehadiran</i>	106
Gambar V.15	<i>Sequence Diagram Proses Cetak Laporan Kehadiran</i>	107
Gambar V.16	<i>Entity Relationship Diagram</i>	107
Gambar V.17	<i>Deployment Diagram Usulan</i>	111
Gambar V.18	<i>HIPO Sistem Absensi Usulan</i>	112

Gambar V.19	<i>Flowchart Pengoperasian Program</i>	113
Gambar V.20	<i>Interface Form Login Admin</i>	114
Gambar V.21	<i>Interface Form Laporan Harian</i>	115
Gambar V.22	<i>Interface Form Laporan Bulanan</i>	116
Gambar V.23	<i>Interface Form Data Karyawan</i>	117
Gambar V.24	<i>Interface Form Data Lembur</i>	118
Gambar V.25	<i>Interface Form Data Leave</i>	119
Gambar V.26	<i>Interface Form Absensi Pegawai</i>	120

DAFTAR TABEL

	Halaman
Tabel II.1	Simbol Bagan Alir Sistem (<i>System Flowchart</i>) 35
Tabel II.2	Simbol Bagan Alir Program (<i>Program Flowchart</i>) 37
Tabel II.3	Simbol Bagan Alir Proses (<i>Process Flowchart</i>) 38
Tabel II.4	Simbol – Simbol <i>Use Case Diagram</i> 40
Tabel II.5	Simbol – Simbol <i>Activity Diagram</i> 41
Tabel II.6	Simbol – Simbol <i>Sequence Diagram</i> 42
Tabel II.7	Simbol – Simbol <i>Class Diagram</i> 44
Tabel II.8	Simbol – Simbol <i>Component Diagram</i> 45
Tabel II.9	Simbol – Simbol <i>Deployment Diagram</i> 47
Tabel II.10	Simbol – Simbol ERD.....53
Tabel II.11	Tipe Data pada My SQL 58
Tabel II.12	Simbol – Simbol Kamus Data.....62
Tabel IV.1	Penghitungan Lembur Menteri Tenaga Kerja dan Transmigrasi 82
Tabel IV.2	Tabel Rekapitulasi Harian yang Berjalan 83
Tabel IV.3	Tabel Rekapitulasi Bulanan yang Berjalan 84
Tabel IV.4	Definisi Aktor Proses Berjalan 87
Tabel IV.5	Definisi <i>Usecase</i> Berjalan.....87
Tabel V.1	Kebutuhan Sistem Absensi Pegawai.....88
Tabel V.2	Definisi <i>Actor</i> 93
Tabel V.3	Definisi <i>Usecase Diagram</i> 93
Tabel V.4	Skenario <i>Usecase</i> Data Karyawan 94
Tabel V.5	Skenario <i>Usecase</i> Data Lembur 94
Tabel V.6	Skenario <i>Usecase</i> Data <i>Leave</i> 95

Tabel V.7	Skenario <i>Usecase</i> Melakukan Absen	95
Tabel V.8	Skenario <i>Usecase</i> Mencetak Laporan Kehadiran	95
Tabel V.9	Tabel <i>Admin</i>	108
Tabel V.10	Tabel Cuti.....	108
Tabel V.11	Tabel Lembur	109
Tabel V.12	Tabel Kehadiran	109
Tabel V.13	Tabel Karyawan	110
Tabel V.14	Tabel <i>Fingerprint</i>	110
Tabel V.15	Tabel Alpa.....	111

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini perusahaan di bidang manufaktur mengalami perkembangan yang pesat. Hal ini dapat dilihat dengan banyaknya perusahaan-perusahaan manufaktur yang semakin berkembang. Dengan pesatnya perkembangan di dunia manufaktur, maka kebutuhan akan sebuah sistem informasi yang baik sangat diperlukan di dalam perusahaan. Salah satu sistem informasi yang dibutuhkan oleh perusahaan yaitu sistem informasi mengenai sumber daya manusia.

Keberadaan sistem informasi di dalam suatu perusahaan sudah menjadi suatu hal yang utama bagi pemenuhan kebutuhan informasi. Sistem informasi merupakan suatu sistem yang dibangun untuk melakukan manajemen terhadap informasi-informasi yang diperlukan oleh perusahaan. Dalam kinerjanya, sistem informasi melakukan pengolahan data untuk mendapatkan informasi yang diinginkan oleh perusahaan.

Pada saat ini, PT Coca Cola Distribution Indonesia Cabang Ujung Menteng masih menggunakan absensi manual (*amano*) dalam melakukan pencatatan kehadiran sedangkan dalam pengelolaan data absensi pegawai masih mengandalkan *software Microsoft Excel*. Divisi HRD mengumpulkan terlebih dahulu kartu absen pegawai lalu melakukan proses rekapitulasi data pegawai untuk membuat laporan kehadiran pegawai bulanan sebelum dilanjutkan ke proses penghitungan gaji pegawai.

Dengan rangkaian proses tersebut, perlu dikembangkan sistem absensi yang lebih handal untuk mempermudah serta memperlancar kinerja divisi HRD dalam merekap daftar kehadiran pegawai tiap bulannya, sekaligus sebagai alat bantu pengawasan bagi pimpinan perusahaan dan HRD dalam hal kedisiplinan pegawai. Dengan demikian perusahaan diharuskan menggunakan sistem informasi absensi yang

baik agar dapat memberikan informasi bagi pihak pengambil keputusan untuk membantu dalam hal pengambilan tindak kedisiplinan bagi pegawainya.

1.2 Pokok Permasalahan

Permasalahan yang terjadi pada PT Coca Cola Distribution Indonesia adalah sebagai berikut:

1. Proses yang berjalan saat ini masih dilakukan secara manual mulai dari proses kontrol kehadiran selama sebulan sampai rekapitulasi data kehadiran pegawai melalui kartu absensi.
2. Pada proses yang berjalan terdapat kesulitan dalam pengarsipan data kehadiran pegawai karena masih menggunakan kertas absensi.
3. Pada proses yang berjalan ditemukan rentan terjadi kecurangan berupa penitipan absen antar pegawai agar tidak terhitung terlambat masuk kerja.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Merancang bangun sistem informasi untuk proses kontrol dan rekapitulasi data kehadiran pegawai menggunakan sistem yang terkomputerisasi.
2. Merancang bangun sistem informasi bagi divisi HRD dalam proses pengarsipan data absensi pegawai tanpa kertas.
3. Merancang bangun sistem yang dapat mengurangi kecurangan penitipan absen antar pegawai karena sistem membutuhkan sidik jari ketika melakukan absensi.

1.4 Batasan Masalah

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

1. Sistem ini dibuat dengan hak akses untuk divisi HRD.

2. Sistem hanya melakukan proses absensi, penyajian data laporan kehadiran harian dan bulanan, penambahan data pegawai, penambahan data cuti dan penambahan data lembur.
3. Sistem tidak mengolah penghitungan gaji pegawai, namun dapat memberikan informasi bagi pengambil keputusan untuk membantu dalam hal pengambilan tindak kedisiplinan bagi pegawai.

1.5 Manfaat Tugas Akhir

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

1. Bagi penulis
 - a. Sebagai alat untuk mempraktekkan teori-teori yang telah diperoleh selama perkuliahan, sehingga penulis dapat menambah pengetahuan secara praktis tentang masalah-masalah yang dihadapi oleh perusahaan.
2. Bagi perusahaan
 - a. Hasil penelitian ini agar dapat diimplementasikan di perusahaan untuk membantu kinerja perusahaan, karena dengan adanya sistem informasi absensi karyawan ini dapat digunakan sebagai bahan masukan dan pertimbangan dalam pengambilan keputusan cepat terkait proses kontrol tingkat kehadiran pegawai dan diharapkan dapat memberikan kemudahan pihak HRD dalam proses arsip dan penyajian data laporan kehadiran bulanan.
3. Bagi pihak lain
 - a. Hasil penelitian ini diharapkan dapat memberikan sumbangan pemikiran dan sebagai referensi bagi peneliti lain yang melakukan penelitian serupa.

1.6 Sistematika Penulisan

Agar lebih mempermudah perumusan pada penelitian Tugas Akhir ini, maka penulis menguraikan tahapan-tahapan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang penelitian, permasalahan, batasan masalah, tujuan penelitian, manfaat tugas akhir dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas tentang pengertian informasi, jenis-jenis informasi, metode pengembangan sistem, *Netbeans*, *Mariadb*, *Relational Database Management Sistem (RDBMS)*, *Entity Relationship Diagram (ERD)*, *Flowchart* dan *Unified Modeling Language (UML)*.

BAB III METODOLOGI PENELITIAN

Pada bab ini membahas tentang metode pengumpulan data, serta langkah-langkah yang akan dilakukan dalam perumusan dan pemecahan masalah termasuk metodologi pengembangan sistem yang digunakan.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini menguraikan tentang data yang telah diperoleh mulai dari struktur organisasi dan sejarah perusahaan, data karyawan dan dokumen yang terdapat pada proses berjalan. Pengolahan data sistem yang berjalan menggunakan diagram alir dan *usecase* sistem yang berjalan berdasarkan penelitian selama melaksanakan penelitian di PT Coca Cola Distribution Indonesia.

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis rinci dari pengolahan data, meliputi kebutuhan sistem, diagram alir sistem usulan, pemodelan sistem dengan UML, pemodelan dengan use case diagram, *Entity Relationship Diagram (ERD)*, hierarki program (HIPO), *flowchart* pengoperasian program, perancangan program, dan spesifikasi sistem informasi yang dibutuhkan

sehingga menghasilkan sistem informasi absensi pegawai yang handal dan dapat menyelesaikan permasalahan yang terjadi di perusahaan.

BAB VI KESIMPULAN DAN SARAN

Pada bab penutup ini dikemukakan kesimpulan-kesimpulan dari hasil penelitian dan saran-saran untuk pihak perusahaan dalam berbagai hal yang berhubungan dengan sistem informasi absensi fingerprint karyawan menggunakan *fingerprint*.

BAB II

LANDASAN TEORI

2.1 Rancang Bangun

Rancang bangun adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem (Jogiyanto, 2005).

2.2 Konsep Dasar Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan keputusan. Informasi diperoleh dari sistem informasi (*information systems*) atau disebut juga *processing systems* atau *information processing systems* atau *information-generating systems*. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005).

2.3 Konsep Dasar Absensi dan Komponen Gaji

2.3.1 Pengertian Absensi

Menurut (Erna Simonna, 2009), Absensi adalah suatu pendataan kehadiran, bagian dari pelaporan aktifitas suatu institusi, atau komponen institusi itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah

untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan .

Absensi secara umum adalah suatu pendataan kehadiran, bagian dari laporan aktifitas suatu institusi atau komponen-komponen institusi itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan. Berdasarkan kamus Bahasa Indonesia, absen adalah tidak bekerjanya seorang karyawan pada saat hari kerja, karena sakit, izin, alpa atau cuti. Absensi adalah daftar administrasi ketidakhadiran karyawan, dimana karyawan yang tidak hadir akan tercatat di daftar absensi dan kapan saja bisa diperiksa oleh atasan perusahaan.

Kita mengenal beberapa jenis absensi, yang membedakan jenis-jenis absensi tersebut adalah cara penggunaannya, dan tingkat daya gunanya Secara umum jenis-jenis absensi dapat di kelompokkan menjadi dua, yaitu;

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

2.3.2 Lembur

Kerja lembur atau *overtime* adalah pekerjaan yang dilakukan oleh karyawan, atas dasar perintah atasan, yang melebihi jam kerja biasa pada hari-hari kerja, atau pekerjaan yang dilakukan pada hari istirahat mingguan karyawan atau hari libur resmi. Peraturan ini berlaku untuk semua karyawan golongan I sampai III. Prinsip kerja lembur pada dasarnya bersifat sukarela, kecuali dalam kondisi tertentu pekerjaan harus segera diselesaikan untuk kepentingan perusahaan.

Waktu kerja lembur adalah waktu kerja yang melebihi 7 jam sehari untuk 6 hari kerja dan 40 jam dalam seminggu atau 8 jam sehari untuk 8 hari kerja dan 40 jam dalam seminggu atau waktu kerja pada hari istirahat mingguan dan atau pada hari libur resmi yang ditetapkan Pemerintah (Pasal 1 ayat 1 Peraturan Menteri no.102/MEN/VI/2004). Waktu kerja lembur hanya dapat dilakukan paling banyak 3 jam/hari dan 14 jam dalam 1 minggu diluar istirahat mingguan atau hari libur resmi.

Ketentuan kerja lembur (Pasal 6 Peraturan Menteri no.102/MEN/VI/2004):

- a. Untuk melakukan kerja lembur harus ada perintah tertulis dari pengusaha dan persetujuan tertulis dari pekerja/buruh yang bersangkutan.
- b. Perintah tertulis dan persetujuan tertulis dibuat dalam bentuk daftar pekerja/buruh yang bersedia bekerja lembur yang ditandatangani oleh pekerja/buruh yang bersangkutan dan pengusaha.

2.3.2.1 Tarif Upah Lembur

Perhitungan Upah Lembur didasarkan upah bulanan dengan cara menghitung upah sejam adalah $\frac{1}{173}$ upah sebulan.

1. Tarif upah lembur sebesar $\frac{1}{173} \times$ Gaji Pokok
2. Perhitungan lembur dilakukan pada hari kerja biasa:
 - a. Untuk jam pertama adalah $1,5 \times$ tarif upah lembur.
 - b. Untuk jam-jam berikutnya adalah sebesar $2 \times$ tarif upah lembur.
3. Perhitungan lembur dilakukan pada hari istirahat mingguan atau hari raya resmi:
 - a. Untuk setiap jam dalam batas waktu 7 (tujuh) jam pertama adalah sebesar dua kali tarif upah lembur.
 - b. Untuk jam ke 8 (delapan) sebesar 3 kali tarif upah lembur.
 - c. Untuk jam ke 9 (sembilan) dan seterusnya adalah sebesar empat kali tarif upah lembur.
4. Pekerjaan lembur kurang dari $\frac{1}{2}$ (setengah) jam sehari tidak diperhitungkan dengan upah lembur.

5. Ketentuan upah lembur hanya berlaku untuk karyawan dengan golongan I sampai III atau dinyatakan lain dalam perjanjian kerja.
6. Untuk karyawan shift, bilamana hari tugasnya jatuh pada hari libur resmi (raya), maka jam kerja pada hari tersebut dihitung sebagai kerja lembur, dan perhitungan upah lemburnya mempergunakan perhitungan jam lembur hari raya.

2.3.2.2 Menghitung Upah Lembur

Cara perhitungan upah kerja lembur sebagai berikut :

1. Apabila kerja lembur dilakukan pada hari kerja :
 - a. Untuk jam kerja lembur pertama harus dibayar upah sebesar 1,5 x(satu setengah) kali upah sejam.
 - b. Untuk setiap jam kerja lembur berikutnya harus dibayar upah sebesar 2(dua) kali upah sejam.
2. Apabila kerja lembur dilakukan pada hari istirahat mingguan dan/atau hari libur resmi untuk waktu kerja 6 (enam) hari kerja 40 (empat puluh) jam seminggu maka :
 - a. perhitungan upah kerja lembur untuk 7 (tujuh) jam pertama dibayar 2 (dua) kali upah sejam, dan jam kedelapan dibayar 3 (tiga) kali upah sejam dan jam lembur kesembilan dan kesepuluh dibayar 4 (empat) kali upah sejam.
 - b. apabila hari libur resmi jatuh pada hari kerja terpendek perhitungan upah lembur 5 (lima) jam pertama dibayar 2 (dua) kali upah sejam, jam keenam 3(tiga) kali upah sejam dan jam lembur ketujuh dan kedelapan 4 (empat) kali upah sejam.
3. Apabila kerja lembur dilakukan pada hari istirahat mingguan dan/atau hari libur resmi untuk waktu kerja 5 (lima) hari kerja dan 40 (empat puluh) jam seminggu, maka perhitungan upah kerja lembur untuk 8 (delapan) jam pertama dibayar 2 (dua) kali upah sejam, jam kesembilan dibayar 3(tiga) kali upah sejam dan jam kesepuluh dan kesebelas 4 (empat) kali upah sejam.

2.3.3 Cuti

Pengertian cuti adalah ketidakhadiran sementara atau tertentu karena alasan tertentu yang mendapat keterangan dari pihak-pihak yang terkait. Cuti bertujuan untuk menjamin kesegaran jasmani dan rohani karyawan yang bersangkutan dengan landasan hukum :

1. Pasal 5 UU nomor 8 tahun 1974,
2. Peraturan Pemerintah nomor 24 Tahun 1976,
3. Surat edaran kepala BAKN Nomor 01/SE/1977 tanggal 25 Februari 1977.

2.4 Biometrik

Biometrik mengacu pada identifikasi manusia berdasarkan karakteristik atau ciri-ciri unik yang dimiliki oleh seorang individu tersebut. Dalam ilmu komputer biometrik digunakan sebagai bentuk identifikasi dan akses kontrol. Berbagai aspek fisiologi manusia dapat digunakan untuk otentikasi biometrik. Terdapat beberapa parameter yang dapat digunakan untuk menilai kesesuaian dari setiap sifat yang digunakan untuk otentikasi biometrik. Parameter-parameter tersebut antara lain :

1. *Universality.*
Setiap orang harus memiliki sifat tersebut.
2. *Uniqueness.*
Sifat harus berbeda antara individu satu dengan yang lain dalam sebuah populasi.
3. *Permanence.*
Sifat tersebut harus terukur secara baik, tahan terhadap penuaan dan varian lainnya dari waktu ke waktu.
4. *Collectability.*
Kemudahan akuisisi atau pengukuran terhadap sifat.
5. *Performance.*
Berkaitan dengan akurasi, kecepatan dan ketahanan teknologi yang digunakan.
6. *Acceptability.*

Berkaitan dengan penerimaan oleh masyarakat terhadap teknologi tersebut sehingga mereka bersedia sifat biometrik tersebut digunakan.

7. *Circumvention.*

Berkaitan dengan kemudahan suatu sifat mungkin untuk ditiru atau dipalsukan.

Hingga saat ini belum ditemukan satu sifat biometrik pun yang memenuhi ketujuh parameter diatas. Untuk saat ini terdapat beberapa sifat biometrik yang telah digunakan untuk otentikasi antara lain : fingerprint, hand geometry, iris, retinal scanning, voice, signature (Riomudita, 2013).

2.5 Sidik Jari (*Fingerprint*)

Sidik jari (*fingerprint*) adalah hasil reproduksi tapak jari baik yang sengaja diambil, dicapkan dengan tinta, maupun bekas yang ditinggalkan pada benda karena pernah tersentuh kulit telapak tangan atau kaki. Kulit telapak adalah kulit pada bagian telapak tangan mulai dari pangkal pergelangan sampai ke semua ujung jari, dan kulit bagian dari telapak kaki mulai dari tumit sampai ke ujung jari yang mana pada daerah tersebut terdapat garis halus menonjol yang keluar satu sama lain yang dipisahkan oleh celah atau alur yang membentuk struktur tertentu. Teknologi fingerprint adalah alat untuk memudahkan para pengguna dalam melakukan presensi dan juga menghindari adanya manipulasi data presensi yang sangat mudah dilakukan apabila presensi masih dilakukan secara manual. Sampai saat ini dan untuk masa yang akan datang fingerprint merupakan salah satu cara yang paling aman karena fingerprint tidak bisa dimanipulasi. Fingerprint telah terbukti cukup akurat, aman, mudah dan nyaman untuk dipakai sebagai identifikasi bila dibandingkan dengan system lainnya seperti retina mata atau DNA. Hal ini dapat dilihat pada sifat yang dimiliki oleh sidik jari, antara lain (Suryani. N, 2006):

1. *Perennial nature*, yaitu guratan-guratan pada sidik jari yang melekat pada kulit manusia seumur hidup.

2. *Immutability*, yaitu sidik jari seseorang tidak pernah berubah, kecuali mendapatkan kecelakaan yang serius.

3. *Individuality*, pola sidik jari adalah unik dan berbeda untuk setiap orang.

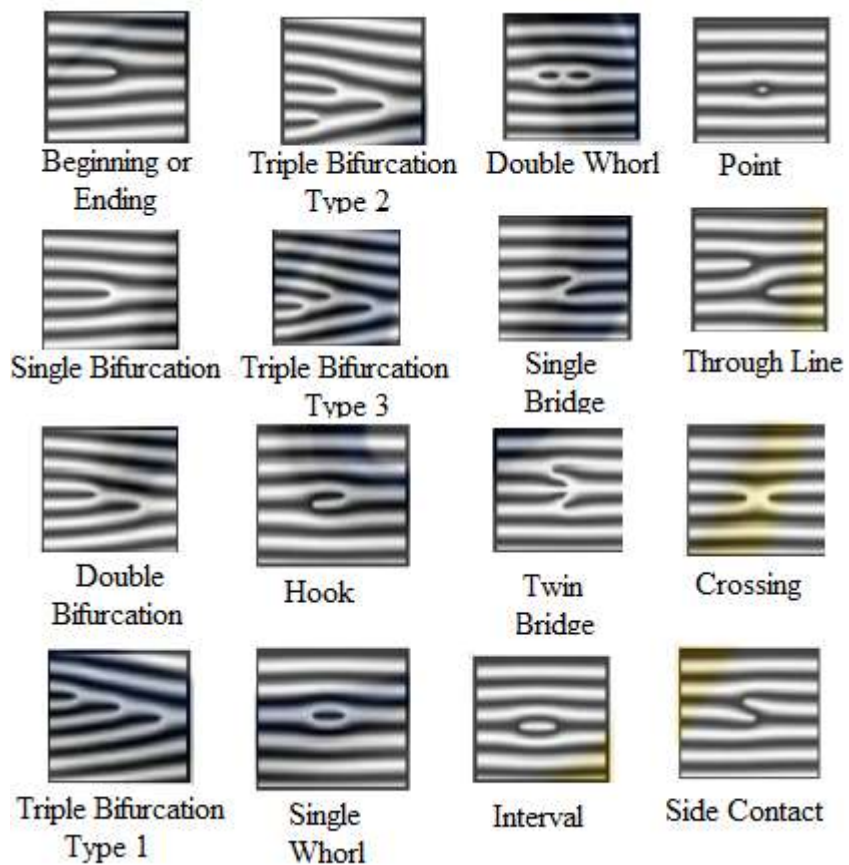
Dari ketiga sifat ini, sidik jari dapat digunakan sebagai sistem identifikasi yang dapat digunakan dalam aplikasi teknologi informasi seperti :

1. *Access System Security*, yaitu akses untuk masuk ke suatu area atau ruangan tertentu yang terbatas.

2. *Authentication System*, yaitu untuk akses data yang sifatnya rahasia dan terbatas.

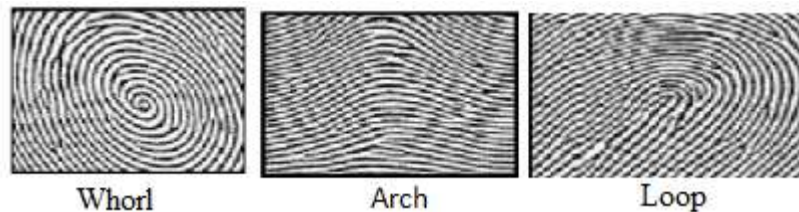
2.5.1 Metode Pengenalan Sidik Jari

Feature sidik jari yang digunakan adalah guratan sidik jari yang dapat diidentifikasi dengan cara menganalisa *fine details* dari guratan-guratan sidik jari yang dinamakan dengan *minutiae*. Beberapa fitur guratan sidik jari dapat dilihat pada gambar dibawah ini (Suryani. N, 2006):



Gambar II.1 Fitur Guratan Sidik Jari
(Sumber: Trisanto dan Juniar, 2014)

Menurut (Francis Galton, 1822-1916) mengatakan bahwa tidak ada dua sidik jari yang sama, artinya setiap sidik jari yang dimiliki oleh seseorang adalah unik. Berdasarkan klasifikasi, pola sidik jari dapat dinyatakan secara umum ke dalam tiga bentuk yaitu :



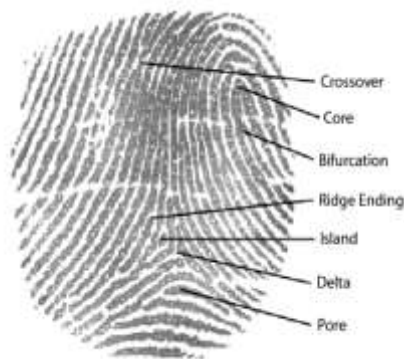
Gambar II.2 Pola Sidik Jari

(Sumber: Trisanto dan Juniar, 2014)

2.5.2 Metode *Minutiae*

Minutiae berasal dari bahasa Inggris yang berarti, barang yang tidak berarti atau rincian yang tidak penting. Seperti artinya, *minutiae* sebenarnya merupakan rincian sidik jari yang tidak penting bagi kita, tetapi bagi sebuah mesin pemindai sidik jari, itu adalah detil yang sangat diperhatikan. Untuk lebih jelasnya, *minutiae* pada sidik jari adalah titik-titik yang mengacu pada:

1. *Crossover* (persilangan dua garis)
2. *Core* (putar-balikan sebuah garis)
3. *Bifurcation* (percabangan sebuah garis)
4. *Ridge ending* (berhentinya sebuah garis)
5. *Island* (sebuah garis yang sangat pendek)
6. *Delta* (pertemuan dari tiga buah garis yang membentuk sudut)
7. *Pore* (percabangan sebuah garis yang langsung diikuti dengan menyatunya kembali percabangan tersebut sehingga membentuk sebuah lingkaran kecil).



Gambar II.3 Metode *Minutiae*

(Sumber: Trisanto dan Juniar, 2014)

Setelah mesin pemindai menyimpan *image* atau gambar yang diambil, mesin lalu melakukan *searching minutiae* atau mencari titik-titik *minutiae*. Lalu mesin

pemindai akan mencari kecocokan pola pada *minutiae-minutiae* yang telah terkumpul tersebut.



Gambar II.4 Hasil Pengolahan Metode *Minutiae*
(Sumber: Trisanto dan Juniar, 2014)

Jika mesin pemindai sidik jari mendapatkan pola sidik jari yang sama, maka proses identifikasi sudah berhasil. Tidak semua *minutiae* harus digunakan, dan pola yang ditemukan tidak harus sama. Maka dapat disimpulkan bahwa posisi jari kita pada saat identifikasi tidak harus sama persis dengan pada saat pertama kali menyimpan data sidik jari pada mesin tersebut (Suryani. N, 2006).

2.5.3 *Fingerprint Authentication*

Otentikasi sidik jari bekerja dalam dua tahap, yaitu:

1. Pengguna terdaftar dengan sistem. Selama proses pendaftaran, satu atau lebih jari-*scan* untuk setiap orang. Disarankan anda mendaftar setidaknya dua jari (lebih dianjurkan) karena dalam hal terjadi kecelakaan atau cedera pada salah satu jari, jari terdaftar lain dapat digunakan untuk mengidentifikasi individu.
2. Ketika seseorang perlu dikonfirmasi (misalnya, untuk membuka pintu atau mengakses komputer), mereka hanya memindai jari mereka pada pembaca sidik jari.

Dalam hal pengembangan aplikasi, ini menurut (Hamid Abdul, 2010) membutuhkan pengembang untuk membangun komponen-komponen berikut:

1. Sebuah aplikasi untuk pengguna sistem untuk mendaftar. Aplikasi ini dapat mengambil beberapa scan setiap jari untuk mendaftarkan diri, untuk memastikan bahwa scan yang diperoleh berkualitas baik.
2. Sebuah layanan/aplikasi yang membaca sidik jari dari pembaca sidik jari dan ekstrak fitur *fingerprint*.

2.5.4 Alur Data Mesin Sidik Jari

Menurut (Perdana M. Budi, 2014) ketika pengembangan sebuah aplikasi sidik jari diaktifkan, aliran data terdiri dari:

1. Gambar sidik jari (*scan*) dari pembaca sidik jari dan membuat data *Record Fingerprint Image Data* dari data mentah. Setiap FID berisi satu atau lebih gambar sidik jari, yang disebut *Fingerprint Image View*. Jika aplikasi mendaftarkan pengguna baru, FID mungkin berisi beberapa FIVs satu untuk setiap jari terdaftar. Sebuah FID khas untuk aplikasi otentikasi hanya berisi satu FIV (salah satu yang sedang digunakan untuk otentikasi pengguna). Setiap FIV (sidik jari) berukuran 140k.
2. Ekstraksi fitur sidik jari menjadi *Fingerprint Minutiae Data Record*. Selama ekstraksi, masing-masing FIV akan dikonversi ke *Fingerprint Minutiae View* pada FMD. Sebuah FMV dalam FMD tidak lebih dari 1.5K.

Alur data yang akan diproses oleh sebuah perangkat komputer di antaranya sebagai berikut :

1. *Fingerprint Image Data*
Catatan berisi gambar sidik jari dari satu atau beberapa jari dari orang yang sama.
2. *Fingerprint Image View*
Bagian dari FID, yang berisi gambar sidik jari dari satu kesan satu jari. Pada kebanyakan kasus, FID hanya berisi satu FIV



Gambar II.5. Contoh Fingerprint Image View

(Sumber: Trisanto dan Juniar, 2014)

3. *Fingerprint Minutiae Data*

Catatan berisi hal-hal kecil sidik jari, dan karakteristik sidik jari opsional lainnya dari satu atau beberapa jari dari orang yang sama. (Fitur jangka waktu yang ditetapkan telah digunakan di masa lalu untuk merujuk FMDs)

4. *Fingerprint Minutiae View*



Gambar II.6. Contoh *Fingerprint Minutiae View*

(Sumber: Trisanto dan Juniar, 2014)

Bagian dari FMD yang berisi karakteristik sidik jari dari satu kesan satu jari. Dalam sebagian besar kasus, catatan minutiae sidik jari hanya berisi satu tampilan.

5. *False Positive Identification Error Rate*

Rasio jumlah usaha setter buka identifikasi yang kembali satu atau lebih calon, dengan jumlah total dari upaya identifikasi setter buka.

6. *False Match Rate*

Rasio jumlah kesalahan upaya perbandingan yang menghasilkan perbedaan skor dibawah ambang batas, dengan jumlah total upaya kesalahan perbandingan.

2.6 Manajemen Sumber Daya Manusia

2.6.1 Pengertian Manajemen Sumber Daya Manusia

Manajemen sumber daya manusia merupakan salah satu hal yang perlu diperhatikan dalam sebuah perusahaan. Menurut Bangun (2012:6) “Manajemen sumber daya manusia dapat didefinisikan sebagai suatu proses perencanaan, pengorganisasian, penyusunan staf, penggerakan dan pengawasan, terhadap pengadaan, pengembangan, pemberian kompensasi, pengintegrasian, pemeliharaan, dan pemisahan tenaga kerja untuk mencapai tujuan suatu organisasi.” Menurut Edy Sutrisno (2012:6) “Manajemen sumber daya manusia merupakan kegiatan perencanaan, pengadaan, pengembangan, pemeliharaan, serta penggunaan sumber daya manusia untuk mencapai tujuan baik secara individu maupun organisasi.” Menurut Sadili Samsudin (2010:22) menyatakan bahwa : “Manajemen Sumber Daya

Manusia merupakan suatu kegiatan pengelolaan yang meliputi pendayagunaan, pengembangan, penilaian, pemberian balas jasa bagi manusia sebagai individu anggota organisasi atau perusahaan bisnis”.

Dari definisi di atas, maka penulis menarik kesimpulan bahwa manajemen sumber daya manusia merupakan proses perencanaan, pengorganisasian, penyusunan staf, dan yang paling penting adalah apakah orang yang kita tempatkan itu tepat guna mencapai tujuan baik secara individu maupun organisasi.

2.6.2 Fungsi-fungsi manajemen sumber daya manusia

Manajemen sumber daya manusia memiliki 5 fungsi menurut (Bangun, 2012:7) fungsi-fungsi operasional dari sumber daya manusia adalah sebagai berikut:

1. Pengadaan sumber daya manusia

Fungsi ini merupakan aktivitas manajemen sumber daya manusia dalam memperoleh tenaga kerja sesuai dengan kebutuhan (jumlah dan mutu) untuk mencapai tujuan organisasi. Penentuan sumber daya manusia yang dibutuhkan disesuaikan dengan tugas-tugas yang tertera pada analisis pekerjaan yang sudah ditentukan sebelumnya. Pengadaan tenaga kerja mencakup analisis pekerjaan, perencanaan sumber daya manusia, dan seleksi serta penempatan sumber daya manusia. Dalam pengadaan sumber daya manusia ada analisis pekerjaan, perencanaan sumber daya manusia, rekrutment, seleksi sumber daya manusia dan penempatan sumber daya manusia.

2. Pengembangan sumber daya manusia

Ini merupakan proses peningkatan pengetahuan dan keterampilan melalui pendidikan dan pelatihan. Pada tahap ini, terdapat dua kegiatan penting sebagai dasar untuk mengembangkan para anggota organisasi, antara lain pendidikan dan pelatihan yang diberikan kepada karyawan. Pengembangan sumber daya manusia dapat dilakukan dengan perencanaan karir (*career planning*), pengembangan karir (*career development*), pengembangan organisasi (*organization development/OD*) dan manajemen dan penilaian kinerja.

3. Pemberian kompensasi

Kompensasi merupakan imbalan yang dibayarkan kepada karyawan atas jasa-jasa yang telah mereka sumbangkan kepada perusahaan. Kompensasi dibagi menjadi 2 yaitu kompensasi finansial dan kompensasi non-finansial. Kompensansi finansial berbentuk berupa uang, baik secara langsung maupun tidak langsung. Kompensasi non-finansial bukan diberikan dalam bentuk uang melainkan seseorang tersebut memperoleh kepuasan dari pekerjaannya dan lingkungan organisasinya.

4. Pengintegrasian

Integrasi berarti mencocokkan keinginan karyawan dalam kebutuhan organisasi. Oleh karena itu diperlukan perasaan dan sikap karyawan dalam

menetapkan kebijakan organisasi. Dalam pengintegrasian ada juga faktor-faktor seperti motivasi kerja, kepuasan kerja, dan kepemimpinan.

5. **Pemeliharaan sumber daya manusia**

Pemeliharaan karyawan berarti mempertahankan karyawan untuk tetap berada pada organisasi sebagai anggota yang memiliki loyalitas dan kesetiaan yang tinggi. Karyawan yang memiliki loyalitas tinggi terhadap perusahaan akan bertanggung jawab atas pekerjaannya dan biasanya memiliki kinerja yang baik. Pemeliharaan sumber daya manusia berhubungan dengan komunikasi kerja dan ada juga keselamatan dan kesehatan kerja.

2.6.3 Tujuan manajemen sumber daya manusia

Setiap organisasi termasuk perusahaan tentunya memiliki suatu tujuan, manajemen sumber daya manusia pun memiliki tujuan. Menurut Cushway (Dalam Edi Sutrisno 2012:7) mengemukakan tujuan manajemen sumber daya manusia sebagai berikut:

1. Memberi pertimbangan manajemen dalam membuat kebijakan sumber daya manusia untuk memastikan bahwa organisasi memiliki pekerja yang bermotivasi dan berkinerja tinggi, pekerja yang selalu siap mengatasi perubahan dan memenuhi kewajiban pekerjaan secara legal.
2. Mengimplementasikan dan menjaga semua kebijakan dan prosedur sumber daya manusia yang memungkinkan organisasi mencapai tujuannya.
3. Membantu dalam pengembangan keseluruhan organisasi dan strategi, khususnya yang berkaitan dengan implikasi sumber daya manusia..
4. Memberi dukungan dan kondisi yang akan membantu manajer lini mencapai tujuan.
5. Menangani berbagai krisis dan situasi sulit dalam hubungan antar pekerja untuk meyakinkan bahwa mereka tidak menghambat organisasi dalam mencapai tujuannya.
6. Menyediakan media komunikasi antara pekerja dan manajemen organisasi.

7. Bertindak sebagai pemelihara standar organisasional dan nilai dalam manajemen sumber daya manusia (Sadili Samsudin, 2010:30) menyatakan ada empat tujuan manajemen sumber daya manusia adalah:
 - a. Tujuan Sosial manajemen sumber daya manusia adalah agar organisasi atau perusahaan bertanggung jawab secara sosial dan etis terhadap kebutuhan dan tantangan masyarakat dengan meminimalkan dampak negatifnya.
 - b. Tujuan Organisasional adalah sasaran formal yang dibuat untuk membantu organisasi mencapai tujuannya.
 - c. Tujuan Fungsional adalah tujuan untuk mempertahankan kontribusi departemen sumber daya manusia pada tingkat yang sesuai dengan kebutuhan organisasi.
 - d. Tujuan individual adalah tujuan pribadi dari tiap anggota organisasi atau perusahaan yang hendak di capai melalui aktivitasnya melalui organisasi.

2.7 Analisis Sistem Informasi

Analisis sistem informasi merupakan langkah untuk menentukan seberapa jauh sistem yang berjalan telah mencapai sasarannya dan mengetahui kelemahan-kelemahannya sehingga dapat diusulkan perbaikan.

Dalam buku *Analisa Sistem Informasi* oleh Sutabri (2004), berikut ini merupakan pengertian analisis sistem menurut para ahli, yaitu:

1. Menurut Mc Leod (2004), analisis sistem adalah penelitian atas sistem yang telah ada dengan tujuan merancang sistem yang baru.
2. Menurut Laudon (2007), analisis sistem adalah memeriksa sebuah masalah yang ada yang akan diselesaikan oleh perusahaan dengan menggunakan sistem informasi. Analisis sistem mencakup beberapa langkah yang harus dilakukan, yaitu:
 - a. Menentukan masalah.
 - b. Mengidentifikasi penyebab dari masalah tersebut.
 - c. Menentukan pemecahan masalahnya.

- d. Mengidentifikasi kebutuhan informasi yang dibutuhkan untuk memecahkan masalah tersebut.
3. Menurut Bentley dan Whitten (2007), Analisis sistem adalah suatu teknik untuk menyelesaikan suatu masalah yang ada pada suatu sistem dengan cara membagi masalah tersebut ke beberapa bagian dengan maksud agar mudah dicari penyelesaiannya.

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

1. *Identify*

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

2. *Understand*

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

3. *Analyze*

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

4. *Report*

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

2.8 Metode 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, 2003):

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, 2011).

Menurut Rosa dan Shalahuddin (2011), 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*)
Menentukan 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.

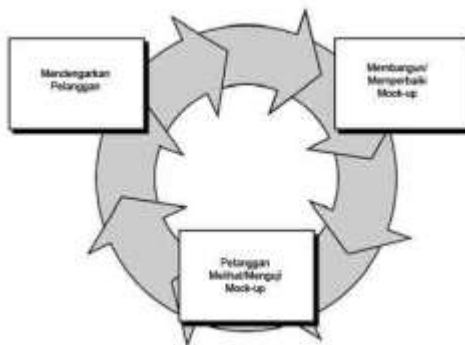
4. Analisis kebutuhan (*requirements analysis*)
Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.
5. Desain (*design*)
Mentransformasikan kebutuhan *detail* menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.
6. Pengembangan (*development*)
Mengkonversi 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*)
Mendeskripsikan 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*.

2.8.1 Model *Prototype*

Model *prototype* dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis yang memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa dan Shalahuddin, 2011).

Tahapan-tahapan pada model *prototype* adalah sebagai berikut (Rosa dan Shalahuddin, 2011):

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



Gambar II.15 Ilustrasi Model *Prototype*

(Sumber: Rosa dan Shalahuddin, 2011)

Mock-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi atau keperluan lain. Sebuah *mock-up* disebut sebagai *prototype* perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada

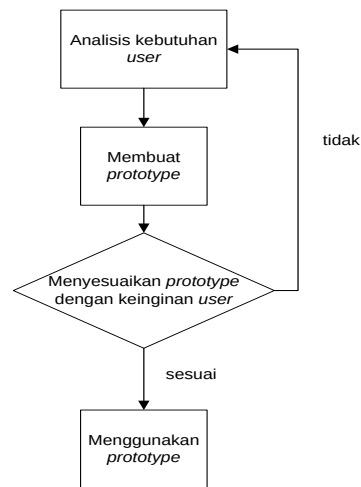
pembuatan *prototype* sampai sesuai dengan keinginan pelanggan atau *user* (Rosa dan Shalahuddin, 2011).

Kelemahan model *prototype* adalah sebagai berikut (Rosa dan Shalahuddin, 2011):

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 *prototype* dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan *prototype* untuk didemonstrasikan. Hal ini dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan iteratif tanpa akhir.

2.8.2 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 II.16 *Evolutionary Prototype Model*

(Sumber: McLeod, 2008)

- a. Analisis 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.

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

2.9 *Flowmap* dan *Flowchart*

Flowmap adalah diagram yang menunjukkan aliran data berupa formulir-formulir ataupun keterangan berupa dokumentasi yang mengalir atau beredar dalam suatu sistem. *Flowchart* adalah bagan yang menunjukkan alir di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Notasi yang digunakan dalam suatu *flowmap* merupakan penggabungan notasi *flowchart* program.

Pada waktu akan menggambar suatu bagan alir, analis sistem atau pemrogram dapat mengikuti pedoman-pedoman sebagai berikut (Jerry, 2003) :

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

2.10 Jenis-Jenis *Flowchart*

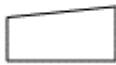
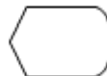
Terdapat beberapa jenis bagan alir (*flowchart*), yaitu sebagai berikut (Jogiyanto, 2005) :

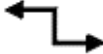
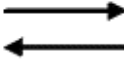
1. Bagan alir sistem (*systems flowchart*)
2. Bagan alir dokumen (*document flowchart*)
3. Bagan alir skematik (*schematic flowchart*)
4. Bagan alir program (*program flowchart*)
5. Bagan alir proses (*process flowchart*)

2.10.1 *Systems Flowchart*

Systems flowchart merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. *Systems flowchart* digambar dengan menggunakan simbol-simbol yang tampak pada Tabel II.1.

Tabel II.1 Simbol Bagan Alir Sistem (*Systems Flowchart*)

Simbol	Keterangan	Simbol	Keterangan
Dokumen 	Menunjukkan dokumen input dan output manual, mekanik atau komputer.	Diskette 	Menunjukkan <i>input/output</i> menggunakan <i>diskette</i>
Kegiatan Manual 	Menunjukkan pekerjaan manual	Drum Magnetik 	Menunjukkan <i>input/output</i> menggunakan drum magnetic
Simpanan Offline N 	File <i>non-komputer</i> yang diarsip menurut angka (<i>Numerical</i>)	Pita Kertas Berlubang 	Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang
Simpanan Offline A 	File <i>non-komputer</i> yang diarsip menurut huruf (<i>Alphabetical</i>)	Keyboard 	Menunjukkan <i>input</i> yang menggunakan <i>online keyboard</i>
Simpanan Offline C 	File <i>non-komputer</i> yang diarsip menurut Tanggal (<i>Cronological</i>)	Display 	Menunjukkan <i>output</i> yang ditampilkan di monitor
Kartu Plong	<i>Input/output</i> yang	Pita Kontrol	Penggunaan pita control

Simbol	Keterangan	Simbol	Keterangan
Kartu Plong 	<i>Input/output</i> yang menggunakan kartu plong (<i>punched card</i>)	Pita Kontrol 	Penggunaan pita control dalam <i>batch control total</i> untuk pencocokan dalam <i>batch processing</i>
Proses 	Menunjukkan kegiatan proses dari operasi program computer	Hubungan Komunikasi 	Menunjukkan proses transmisi data melalui <i>channel</i> komunikasi
Operasi Luar 	Menunjukkan operasi yang dilakukan di luar proses operasi computer	Garis Alir 	Menunjukkan arus dari proses
Pengurutan Offline 	Menunjukkan proses pengurutan data di luar proses computer	Penjelasan 	Menunjukkan penjelasan dari suatu proses
Pita Magnetik 	Menunjukkan <i>input/output</i> menggunakan pita magnetik	Penghubung 	Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain
Hard Disk 	Menunjukkan <i>input/output</i> menggunakan <i>hard disk</i>		

(Sumber: Jogiyanto, 2005)

2.10.2 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. Bagan alir dokumen menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem (*systems flowchart*).

2.10.3 Schematic Flowchart

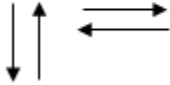
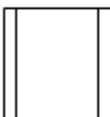
Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem.

Perbedaan bagan alir skematik dari bagan alir sistem adalah selain menggunakan simbol-simbol bagan alir sistem digunakan pula gambar-gambar komputer dan peralatan lain. Maksud dari penggunaan gambar tersebut adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar mudah untuk dipahami, tetapi sulit serta lama pada pengerjaan.

2.10.4 Program Flowchart

Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem. Bagan alir program dibuat dengan menggunakan simbol yang terdapat pada Tabel II.2.

Tabel II.2 Simbol Bagan Alir Program (*Program Flowchart*)

Simbol	Nama	Keterangan
	Input/ output	Simbol <i>input/ output</i> digunakan untuk mewakili data <i>input/ output</i>
	Proses	Simbol proses digunakan untuk mewakili suatu proses
	Garis Alir	Simbol garis alir digunakan untuk menunjukkan arus dari proses
	Penghubung	Simbol penghubung digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau di halaman lain
	Keputusan	Simbol keputusan digunakan untuk suatu penyeleksian kondisi di dalam program
	Proses terdefinisi	Simbol proses terdefinisi digunakan untuk menunjukkan suatu operasi yang rincian ditunjukkan di tempat lain
	Persiapan	Simbol persiapan digunakan untuk memberi nilai awal suatu sasaran
	Titik terminal	Simbol titik terminal digunakan untuk menunjukkan awal dan akhir dari suatu proses

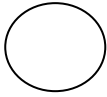
(Sumber: Jogiyanto, 2005)

2.10.5 *Process Flowchart*

Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur. Bagan alir proses memiliki simbol yang terdapat pada Tabel II.3.

Tabel II.3 Simbol Bagan Alir Proses (*Process Flowchart*)

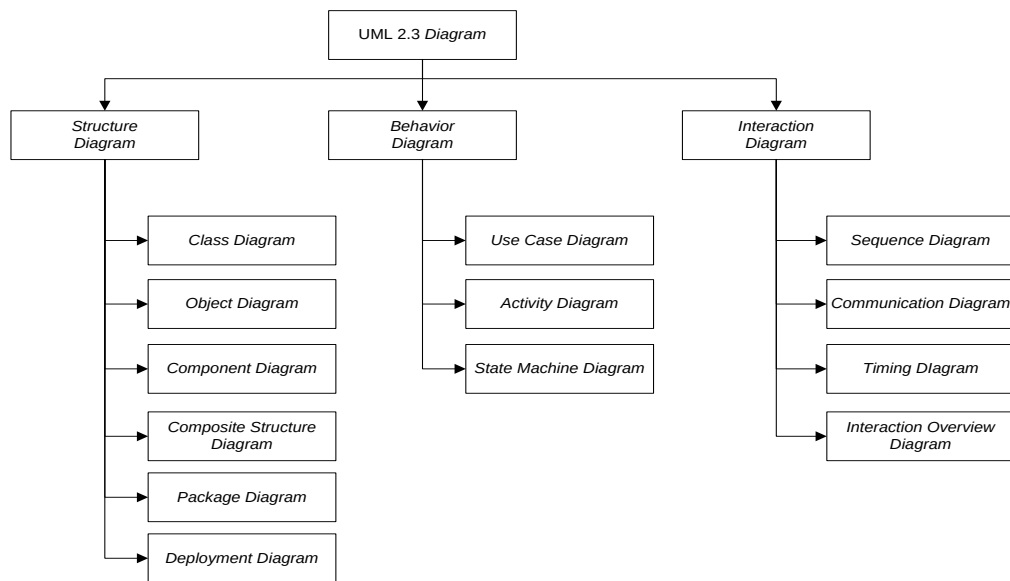
Simbol	Nama	Keterangan
--------	------	------------

	<i>Operation</i>	Menunjukkan suatu operasi
	<i>Movement</i>	Menunjukkan suatu pemindahan
	<i>Storage</i>	Menunjukkan suatu simpanan
	<i>Inspection</i>	Menunjukkan suatu inspeksi
	<i>Delay</i>	Menunjukkan suatu penundaan

(Sumber: Jogyanto, 2005)

2.11 *Unified Modelling Language (UML)*

Unified Modeling Language (UML) adalah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori tersebut dapat dilihat pada gambar berikut ini (Rosa dan Shalahuddin, 2011):



Gambar II.17 Klasifikasi Diagram UML

(Sumber: Rosa dan Shalahuddin, 2011)

Berikut ini penjelasan singkat dari pembagian kategori tersebut (Rosa dan Shalahuddin, 2011):

1. *Structure diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan proses sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsystem pada suatu sistem.

2.11.1 Use Case Diagram

Usecase diagram merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *usecase* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem

informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *usecase* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan *usecase*.

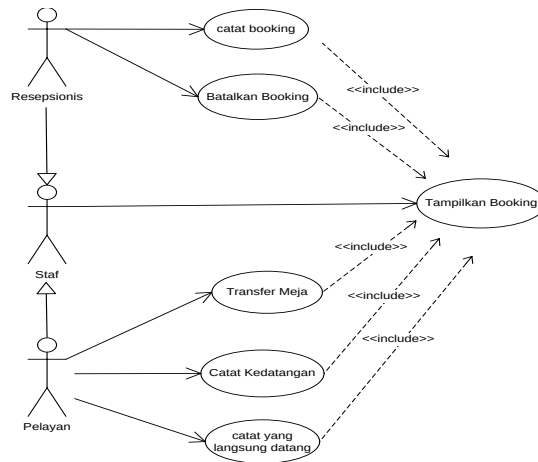
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. (Rosa dan Shalahuddin, 2011).

Tabel II.4 Simbol-simbol *Use Case Diagram*

Simbol	Nama	Deskripsi
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
	<i>Generalization</i>	Disebut juga inheritance (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
	<i>Include</i>	Relasi use case tambahan ke sebuah use-case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau syarat dijalankan use case ini.
	<i>Extend</i>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan.
	<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama use case.

(Sumber: Rosa dan Shalahuddin, 2011)

Contoh *Use Case Diagram* dapat dilihat pada gambar berikut ini:



Gambar II.18 Contoh *Use Case Diagram*

(Sumber: Rosa dan Shalahuddin, 2011)

2.11.2 *Activity Diagram*

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2011).

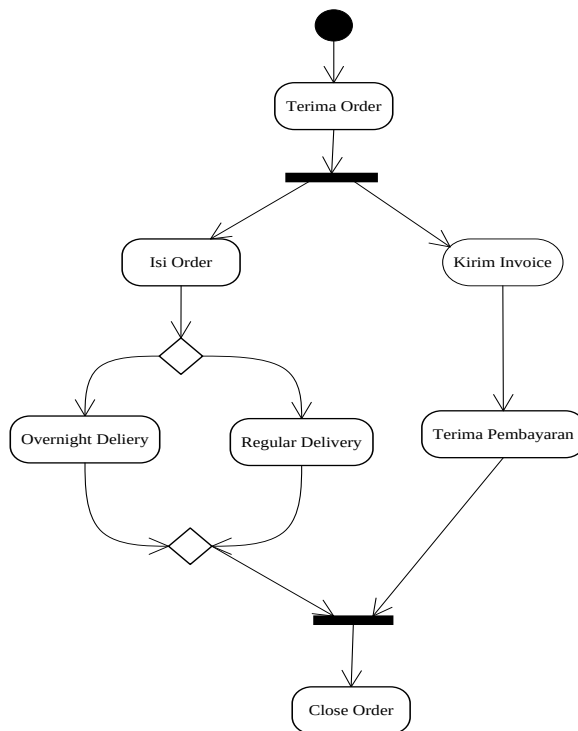
Tabel II.5 Simbol-simbol *Activity Diagram*

Simbol	Nama	Deskripsi
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah aktivitas awal
	Aktivitas	Aktivitas yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja
	Percabangan/ <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

		
	Penggabungan/ <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Simbol	Nama	Deskripsi
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(Sumber: Rosa dan Shalahuddin, 2011)

Contoh *Activity Diagram* dapat dilihat pada gambar berikut ini:



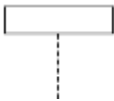
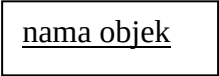
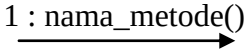
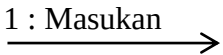
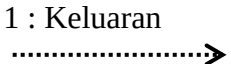
Gambar II.19 Contoh *Activity Diagram*

(Sumber: Rosa Shalahuddin, 2011)

2.11.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeksripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Banyaknya *sequence* diagram yang harus digambar adalah minimal sebanyak pendefinisian *usecase*.

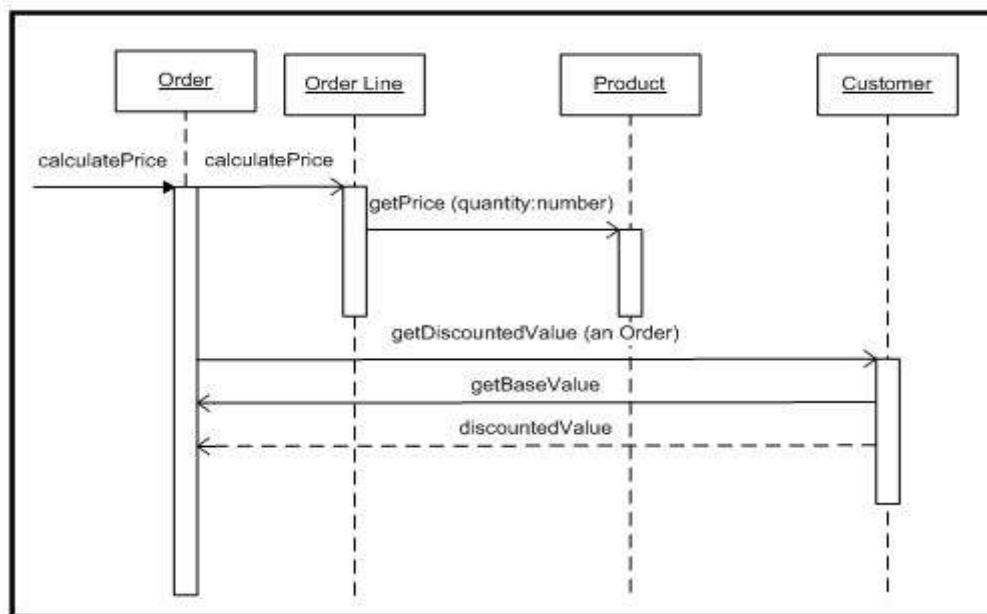
Tabel II.6 Simbol-simbol *Sequence Diagram*

Simbol	Nama	Deskripsi
	Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi dibuat.
	<i>Life Line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	Objek	Menyatakan objek yang berinteraksi pesan.
	waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
	Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri.
	Pesan tipe <i>send</i>	Menyatakan suatu objek mengirimkan data ke objek lain.
	Pesan tipe <i>return</i>	Menyatakan suatu objek telah menjalankan suatu operasi atau

		metode menghasilkan suatu kembalian ke objek tertentu.
--	--	---

(Sumber: Rosa dan Shalahuddin, 2011)

Contoh *Sequence Diagram* dapat dilihat pada gambar berikut ini:



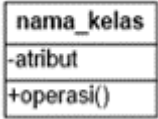
Gambar II.20 Contoh *Sequence Diagram*

(Sumber: Rosa Shalahuddin, 2011)

2.11.4 *Class Diagram*

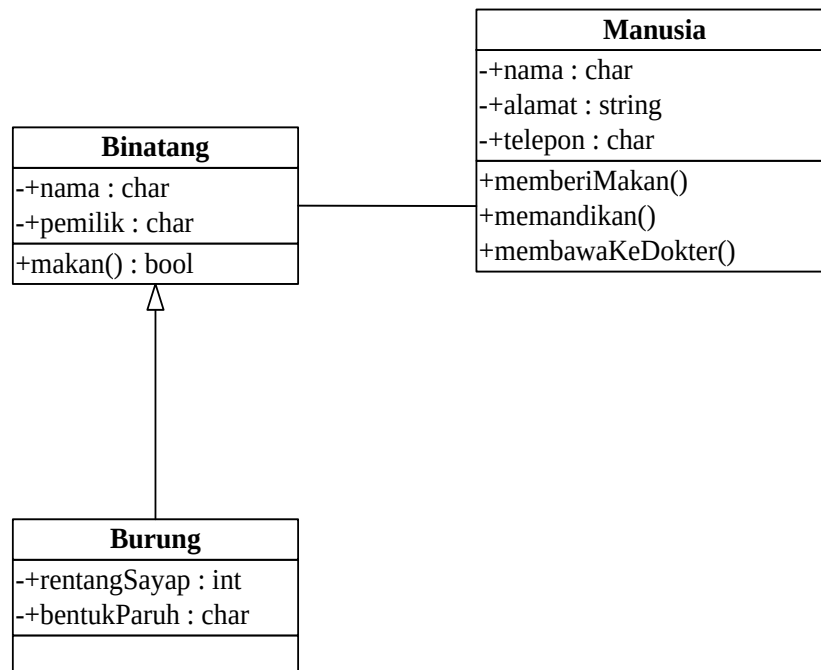
Class diagram merupakan diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa dan Shalahuddin, 2011).

Tabel II.7 Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berbasis objek
3.	Asosiasi/association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
No	Simbol	Deskripsi
4.	Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6.	Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas
7.	Agregasi/aggregation 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

(Sumber: Rosa dan Shalahuddin, 2011)

Contoh *Class Diagram* dapat dilihat pada gambar berikut ini:



Gambar II.21 Contoh *Class Diagram*

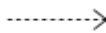
(Sumber: Rosa Shalahuddin, 2011)

2.11.5 *Component Diagram*

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Berikut adalah simbol yang ada pada diagram komponen:

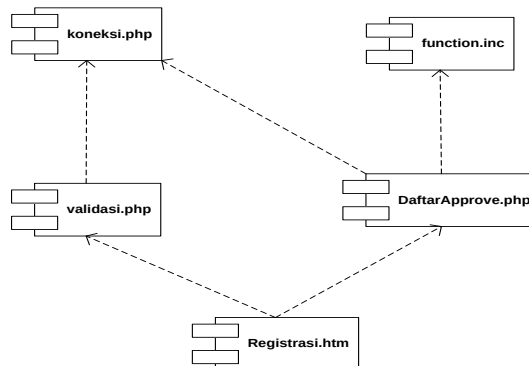
Tabel II.8 Simbol-simbol *Component Diagram*

Simbol	Nama	Keterangan
	<i>Package</i>	Merupakan sebuah kumpulan dari satu atau lebih komponen
	<i>Link</i>	Relasi antar objek
	<i>Component</i>	Komponen system

	<i>Dependency</i>	Hubungan suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Interface</i>	Sebagai antarmuka komponen

(Sumber: Rosa dan Shalahuddin, 2011)

Contoh *Component Diagram* dapat dilihat pada gambar berikut ini:



Gambar II.22 Contoh *Component Diagram*

(Sumber: Rosa Shalahuddin, 2011)

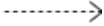
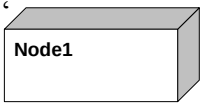
2.11.6 *Deployment Diagram*

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. *Deployment diagram* juga dapat digunakan untuk memodelkan hal-hal berikut (Rosa dan Shalahuddin, 2014):

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node* dan *hardware*.
2. Sistem *client/server*.
3. Sistem terdistribusi murni.

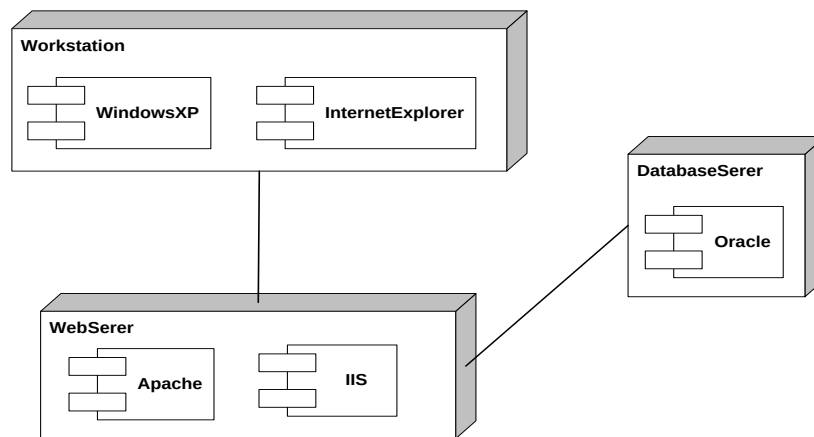
4. Rekayasa ulang aplikasi.

Tabel II.9 Simbol-simbol *Deployment Diagram*

Simbol	Nama	Keterangan
	<i>Package</i>	Merupakan sebuah kumpulan dari satu atau lebih komponen
	<i>Link</i>	Relasi antar objek
	<i>Dependency</i>	Hubungan pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Node</i>	Perangkat keras dan perangkat lunak

(Sumber: Rosa dan Shalahuddin, 2011)

Contoh *Deployment Diagram* dapat dilihat pada gambar berikut ini:



Gambar II.23 Contoh *Deployment Diagram*

(Sumber: Rosa Shalahuddin, 2011)

2.12 Perancangan Database

2.12.1 Pengertian Database

Kumpulan file-file yang saling berhubungan atau berelasi sehingga membantu suatu basis data. Basis data adalah sekumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa adanya suatu kerangkapan data, sehingga mudah untuk digunakan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya, data disimpan sedemikian rupa sehingga apabila ada penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

Dari pengertian tersebut dapat disimpulkan bahwa basis data mempunyai beberapa kriteria yang penting yaitu :

- Bersifat data oriented dan bukan program oriented.
- Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
- Dapat berkembang dengan mudah, baik volume maupun strukturnya.
- Dapat memenuhi sistem-sistem baru secara mudah.

- e. Dapat digunakan dengan cara yang berbeda-beda.

2.12.2 Normalisasi

Proses pengelompokan data ke dalam bentuk tabel atau relasi atau file untuk menyatakan entitas dan hubungan mereka sehingga terwujud satu bentuk database yang mudah untuk dimodifikasi.

Pada proses normalisasi selalu diuji beberapa kondisi. Apakah ada kesulitan selama proses *delete*, *insert* atau *update* dalam basis data yang disebut penyimpangan-penyimpangan atau dikenal dengan *Anomaly*.

2.12.3 Relational Database Management Systems

Relationship Database Management System (RDBMS) adalah sebuah DBMS yang mendukung adanya hubungan relationship antara tabel. MySQL adalah salah satu RDBMS yang paling populer digunakan di seluruh dunia karna selain gratis MySQL juga stabil dan memiliki fitur yang lumayan lengkap.

Suatu database terdiri dari banyak tabel dan tabel ini terdiri dari banyak field yang merupakan kolomnya. Isi tiap baris dari tabel inilah merupakan data. Sedangkan untuk membuat data saling terhubung dan terintegrasi, maka satu tabel dengan tabel yang lainnya harus memiliki hubungan antara satu dengan yang lainnya. Setiap tabel mempunyai sebuah primary key, lalu primary key ini dihubungkan dengan tabel kedua dan menjadi foreign key di tabel kedua ini.

Dengan konsep seperti inilah data akan saling terhubung dan disebut relational, karna data dalam tabel yang satu terhubung dengan tabel yang lainnya.

2.12.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang menggambarkan keterkaitan antar tabel beserta dengan *field-field* di dalamnya pada suatu database sistem. Sebuah database memuat minimal sebuah tabel dengan sebuah atau beberapa buah *field* (kolom) di dalamnya.

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada tiga macam komponen yang digunakan yaitu:

1. Entitas

Adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh pelanggan, pekerja dan lain-lain.

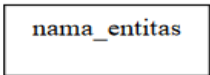
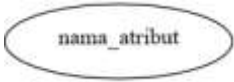
2. Atribut

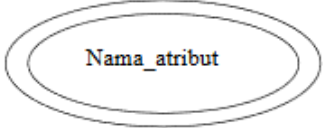
Berfungsi mendeskripsikan karakter entiti. Misalnya atribut nama pekerja dari entiti pekerja. Setiap entiti bisa terdapat lebih dari satu atribut.

3. Hubungan atau (Relationship)

Sebagaimana halnya entiti maka dalam hubunganpun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isi dari hubungan itu sendiri. Misalnya dalam kasus hubungan antara entiti siswa dan entiti mata_kuliah adalah mengikuti, sedangkan isi hubungannya dapat berupa nilai_ujian. Relationship disimbolkan dalam bentuk intan /diamonds. Simbol-simbol yang digunakan dalam Entity Relationship Diagram dapat dilihat pada tabel II.10.

Tabel II.10 Simbol ERD

No	Simbol	Keterangan
1	Entitas/ <i>entity</i> 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
2	Atribut 	Field atau kolom data yang butuh disimpan dalam suatu entitas
3	Atribut kunci primer	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record

		yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
No	Simbol	Keterangan
4	Atribut multivalued/multivalve 	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5	Relasi 	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.
6	Asosiasi 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antar entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dan entitas B.

(Sumber : Rosa Salahudin, 2008:75)

Namun, database lebih sering memiliki lebih dari satu tabel (dengan beberapa *field* di dalamnya). Setiap tabel pada umumnya memiliki keterkaitan hubungan. Keterkaitan antar tabel ini biasa disebut dengan relasi. Terdapat tiga buah jenis relasi antar tabel di dalam bagan ERD. Ketiga relasi tersebut yaitu:

a. *One to One* (Satu ke Satu)

Relasi ini menggambarkan hubungan satu *field* pada tabel pertama ke satu *field* pada tabel kedua. Relasi ini paling sederhana. Sebagai contoh, pada sistem informasi perpustakaan terdapat tabel buku (dengan *field* Kode_Buku, Kode_Kategori, Kode_Penulis, Nama_Penulis, Judul, Penerbit) dan tabel Kategori (Kode_Kategori, Nama_Kategori, Alamat). *Field* Kode_Kategori memiliki keterkaitan (relasi) satu ke satu pada tabel Buku dan tabel Kategori.

b. *One to Many* (Satu ke Banyak)

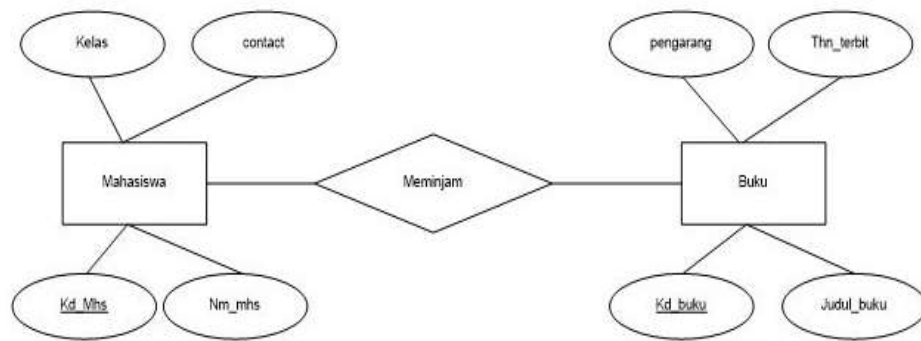
Relasi ini menggambarkan hubungan satu *field* pada tabel pertama ke dua atau beberapa buah *field* di tabel kedua.

c. *Many to Many* (Banyak ke Banyak)

Sebagai contoh, sebuah sistem informasi sekolah memiliki pengguna guru dan siswa di dalamnya. Sistem informasi ini memiliki sebuah database bernama sisfo sekolah dengan tiga buah tabel didalamnya. Ketiga tabel tersebut adalah tabel Guru (memuat *field* Kode_Mata_Pelajaran, Nama_Mata_Pelajaran) dan tabel Mengejar (memuat *field* NIP, Kode_Mata_Pelajaran, Kelas).

Tabel Mengajar terbentuk dari relasi antara tabel Guru dan tabel Mata Pelajaran. Relasi dalam hal ini berupa relasi N ke N, yang artinya satuan atau lebih *field* pada tabel pertama dapat dihubungkan ke satu atau lebih ke *field* pada tabel kedua dan membentuk tabel ketiga. Pada contoh ini satu guru (pada tabel Guru) dapat mengajar lebih dari satu mata pelajaran (pada tabel Mata Pelajaran) dan satu mata pelajaran dapat memiliki lebih dari satu guru pengajar, untuk kemudian membentuk tabel Mengajar.

Berikut ditampilkan bagan ERD untuk ketiga tabel di atas (berserta dengan *field* masing-masing):



Gambar II.27 Contoh Bagan ERD

(Sumber: Sutanta, 2011)

2.13 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.

Kamus data mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk keperluan tersebut maka harus memuat hal-hal sebagai berikut :

1. Alias atau Nama lain
Alias perlu ditulis karena data yang sama mempunyai nama yang beda untuk orang atau departemen lainnya.
2. Bentuk Data
3. Bentuk dari data ini perlu dicatat di kamus data karena dapat digunakan untuk mengelompokkan kamus data kedalam kegunaannya sewaktu perancangan sistem. Bentuk data yang mengalir dapat berupa:

- a. Dokumen dasar atau formulir
- b. Dokumen hasil cetakan komputer
- c. Laporan tercetak
- d. Lampiran dilayar monitor
- e. Variabel
- f. Farameter
- g. Field

4. Arus Data

Arus data menunjukkan dari mana data mengalir dan kemana data akan menuju. Keterangan arus data ini perlu di catat di kamus data supaya memudahkan mencari arus data ini DAD.

5. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang di catat di kamus data sehingga bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

6. Periode

Periode ini menunjukkan kapan terjadinya arus data ini, periode perlu dicatat di kamus data karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke sistem, kapan proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

7. Volume

Volume dicatat di kamus data meliputi tentang volume rata – rata dan volume dari arus data. Volume rata-rata menunjukkan layaknya rata – rata arus data yang mengalir dalam satu periode tertentu dan volume puncak menunjukkan volume yang terbanyak. Volume ini digunakan untuk mengidentifikasi besarnya simpanan luar yang akan digunakan, kapasitas, dan jumlah dari alat input, alat pemroses, dan alat output.

8. Struktur Data

Struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari item-item data apa saja.

Simbol-simbol yang di gunakan dalam kamus data adalah :

Tabel II.12 Simbol-simbol kamus data

Simbol	Keterangan
=	Terdiri dari
+	Dan
()	Optimal
{}	Perulangan
[]	Pilih salah satu dari beberapa alternatif
@	Kunci record
*	Komentar /pendapat
!	Garis pemisah

(Sumber: Rosa Shalahuddin, 2013)

2.14 *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.

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

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-statemen program yang digunakan untuk melaksanakan fungsi tersebut.

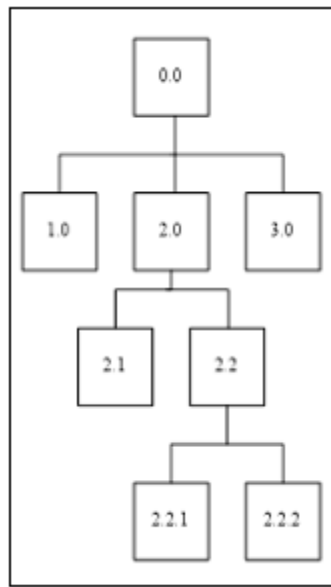
3. Untuk menyediakan penjelasan yang jelas dari *input* yang harus digunakan dan *output* yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari diagram-diagram HIPO.
4. Untuk menyediakan *output* yang tepat dan sesuai dengan kebutuhan-kebutuhan pemakai.

2.14.1 Diagram dalam Paket HIPO

Menurut (Jogiyanto, 2005) HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program, fungsi-fungsi dari sistem digambarkan oleh HIPO dalam tiga tingkatan. Untuk masing-masing tingkatan digambarkan dalam bentuk diagram tersendiri, dengan demikian HIPO menggunakan tiga macam diagram untuk masing-masing tingkatannya, yaitu sebagai berikut:

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 sebagai berikut:



Gambar II.24 *Visual Table Of Contents*

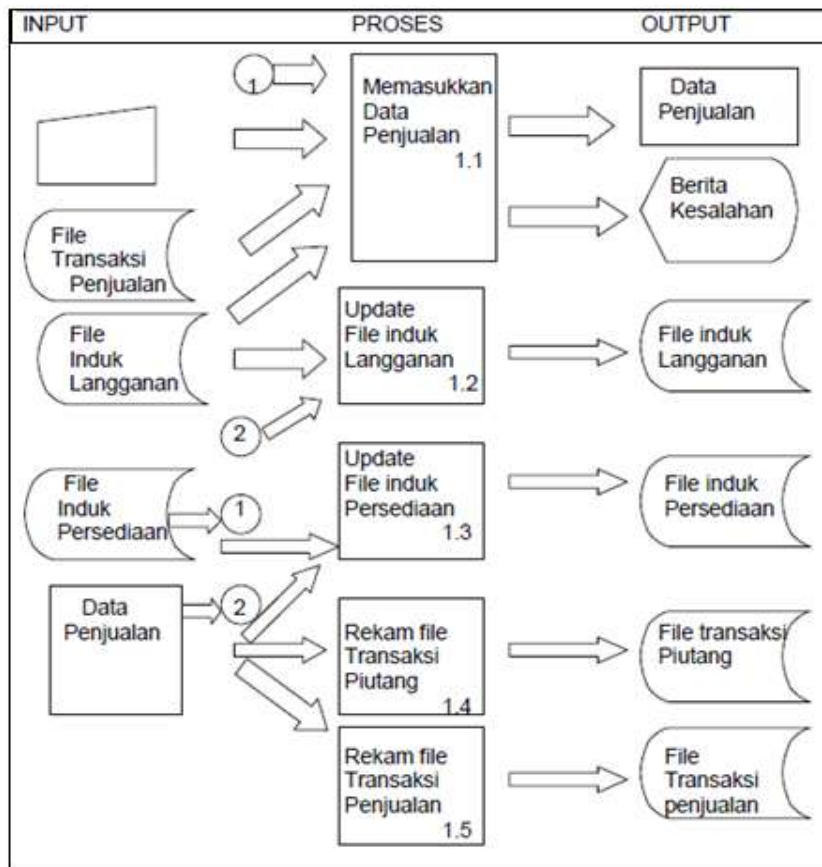
(Sumber: Jogiyanto, 2005)

Penjelasan gambar II.25 Visual Table Of Contents

- Gambar II.25 menunjukkan ada 7 buah fungsi didalam sistem
- Fungsi dengan nomor 1.0, 2.0, 3.0 merupakan tingkatan yang tertinggi
- Fungsi 2.1 dan 2.2 merupakan fungsi dibawah 2.0
- Fungsi 2.2.1 dan 2.2.2 merupakan fungsi dibawah 2.2

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.

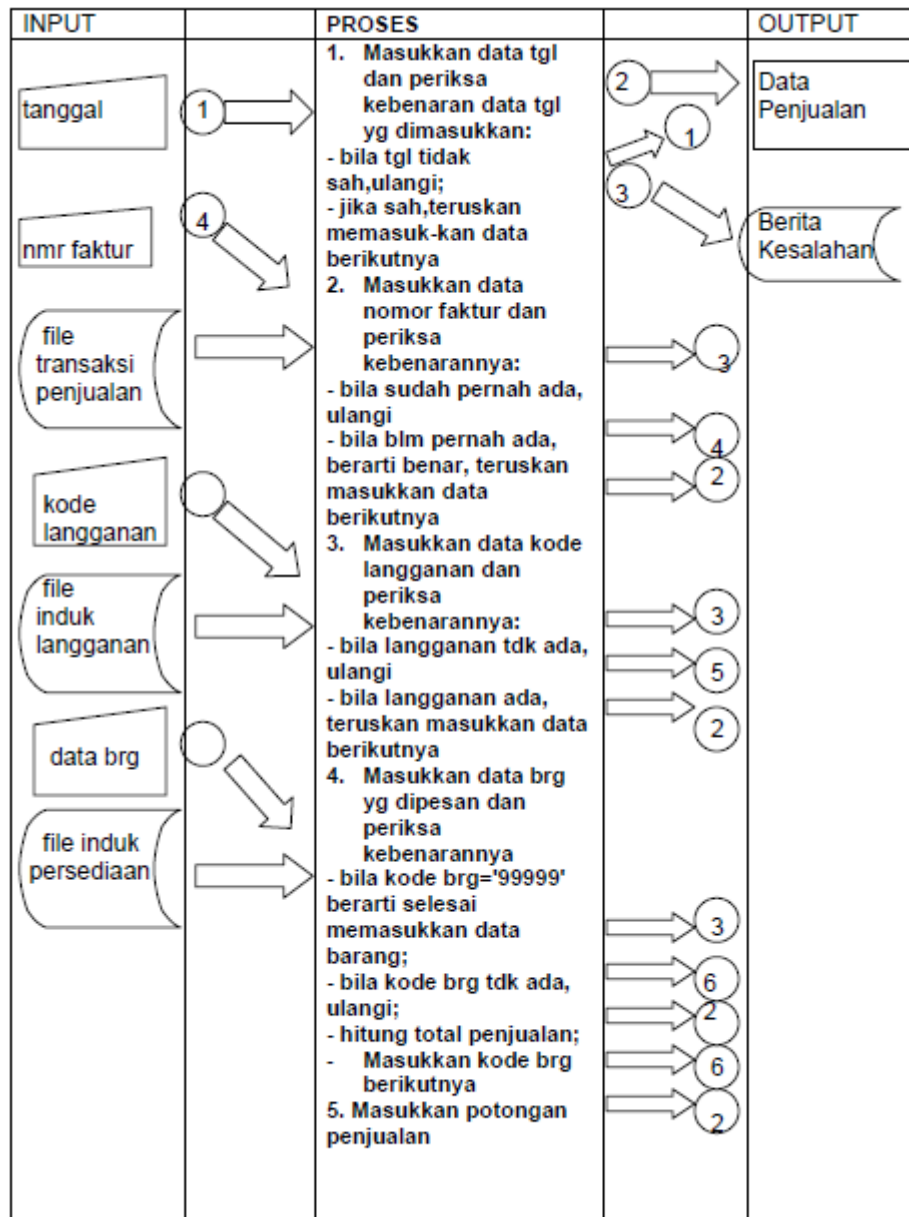


Gambar II.25 Overview Diagram

(Sumber: Jogiyanto, 2005)

3. Detail Diagram

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



Gambar II.26 Detail Diagram

(Sumber: Jogiyanto, 2005)

2.15 Bahasa Pemrograman

2.15.1 Netbeans

NetBeans adalah suatu perangkat lunak yang dituliskan dalam bahasa pemrograman Java. NetBeans pun memperkenalkan suatu pengembangan aplikasi dilakukan dengan dimulai dari sesetel pembentukan kesatuan perangkat lunak modular yang dinamai modules. Semula, aplikasi NetBeans IDE ini diperuntukkan demi suatu pengembangan dalam Java. Namun, aplikasi ini juga mendukung program-program pembuatan bahasa lain secara khusus seperti PHP, C/C++ dan HTML5. NetBeans adalah lintas platform serta penerapannya dijalankan pada Microsoft Windows, Mac OS X, Linux, Solaris dan serambi-serambi lain yang mendukung suatu JVM yang sepadan.

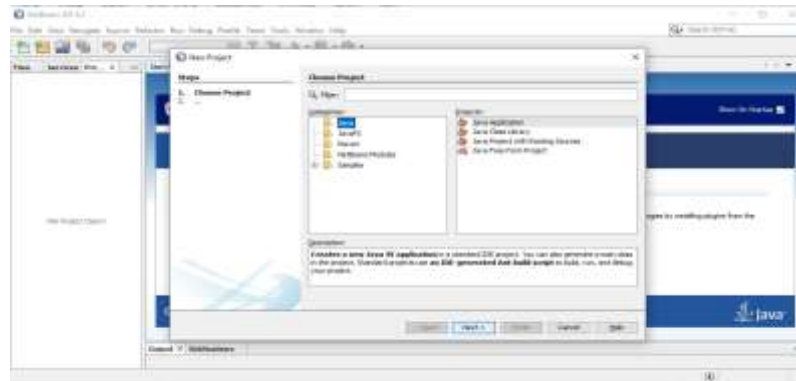
2.15.1.1 Netbeans 8.2

Untuk menjalankan Netbeans IDE 8.2 yaitu dengan cara klik Start > Programs > Netbeans IDE 8.2. Di dalam Netbeans 8.2 terdapat beberapa komponen dasar seperti *GUI Builder*, *Navigator Windows*, *Pallette Windows*, *Properties Windows* dan *Source Area*. Antarmuka Netbeans IDE 8.2 dapat dilihat pada gambar II.7 sebagai berikut:



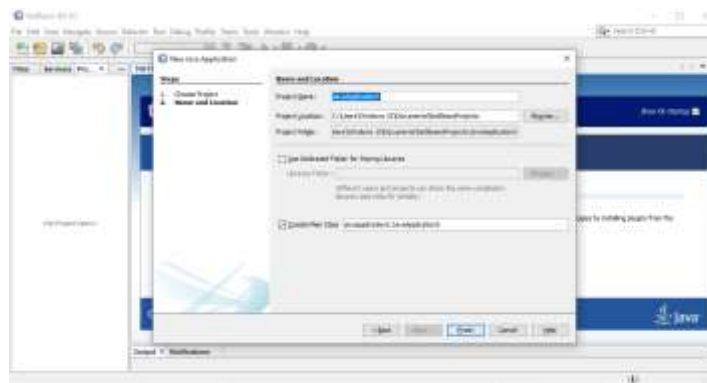
Gambar II.7 Antarmuka Netbeans 8.2

Tahap selanjutnya adalah membuat *project* baru yang tampilannya bisa dilihat di Gambar II.8 sebagai berikut:



Gambar II.8 Antarmuka Pembuatan *Project* Baru

Lalu berikan nama *project* dan pilih lokasi penyimpanan *project* baru tersebut yang tampilannya bisa dilihat di Gambar II.9 sebagai berikut:

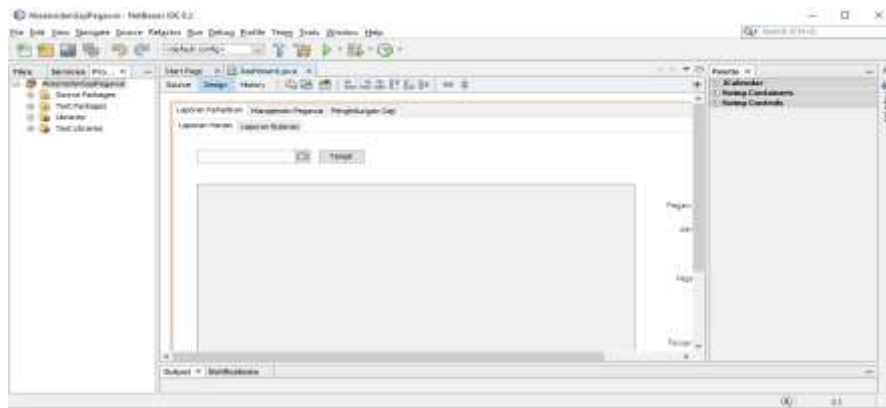


Gambar II.9Antarmuka Pemilihan Lokasi Penyimpanan *Source Code*

Lingkungan kerja pada Netbeans 8.2 umumnya memiliki lingkungan sebagai berikut:

1. *GUI Builder*

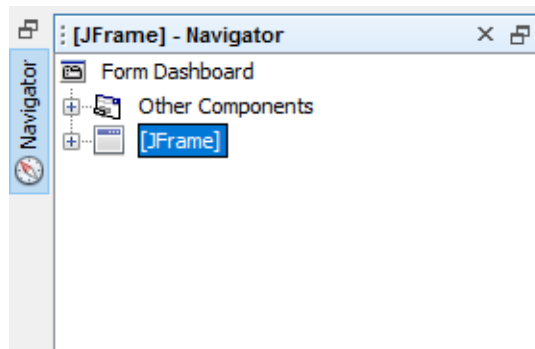
GUI Builder merupakan jendela utama yang didalamnya terdapat komponen untuk merancang GUI.



Gambar II.10 GUI Builder di Netbeans 8.2

2. Navigator Windows

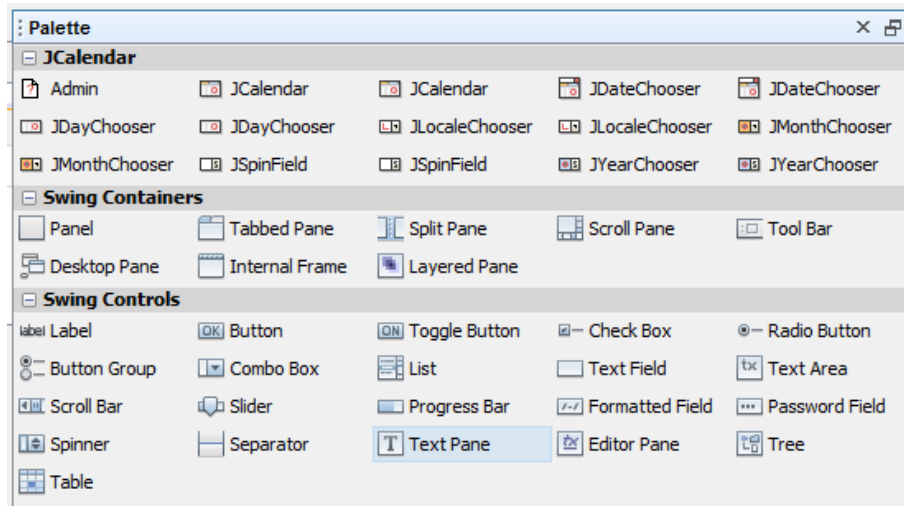
Navigator Windows merupakan jendela yang menampilkan pohon pewarisan dari semua komponen *form* yang dibuka seperti button, label, menu, timer dan sebagainya.



Gambar II.11 Navigator Windows di Netbeans 8.2

3. Pallete Windows

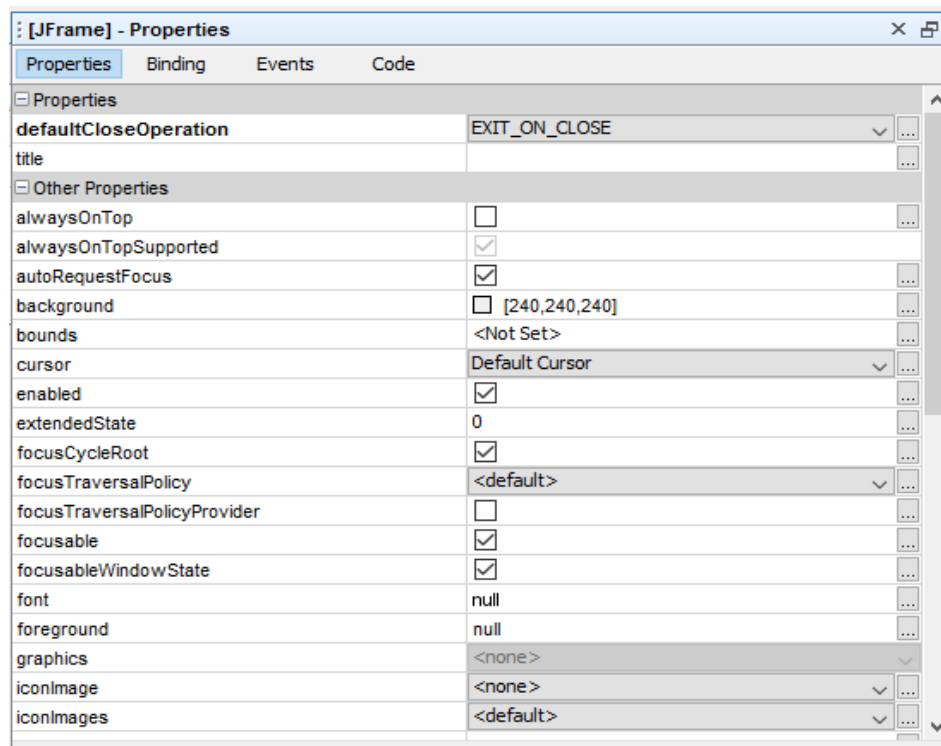
Pallete Windows adalah jendela yang menampilkan daftar semua komponen *swing* yang dapat dimasukkan ke dalam form seperti button, label, menu, timer dan sebagainya



Gambar II.12 *Pallette Windows* di Netbeans 8.2

4. *Properties Windows*

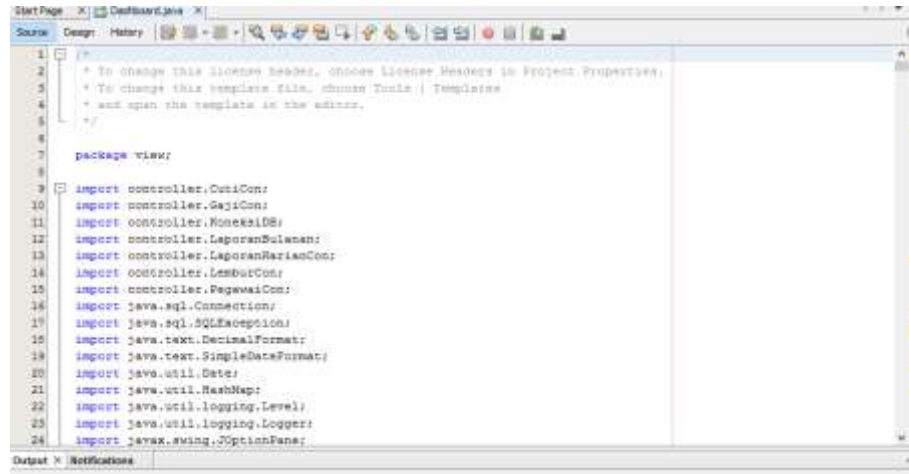
Properties Windows merupakan jendela yang menampilkan komponen dari *project* yang sedang digunakan



Gambar II.13. *Properties Windows* di Netbeans 8.2

5. Source Area

Source Area merupakan jendela yang digunakan untuk menambahkan kode program pada pemrograman JAVA



Gambar II.14 Source Area di Netbeans 8.2

2.15.1.2 Kelebihan dan Kekurangan

Setiap bahasa pemrograman mempunyai kelebihan dan kekurangan masing, berikut adalah kelebihan dan kekurangan Java.

Kelebihan Bahasa Pemrograman Java :

a. *Multiplatform.*

Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis Java dikerjakan di atas *operating system Linux* tetapi dijalankan dengan baik di atas *Microsoft Windows*. *Platform* yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya

adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.

b. *Object Oriented Programming*

Perpustakaan Kelas Yang Lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman Java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

c. Bergaya C++,

Java memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.

d. Pengumpulan sampah otomatis,

Java memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

Kekurangan Bahasa Pemrograman Java

a. Tulis sekali, jalankan di mana saja.

Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

b. Mudah didekompilasi.

Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/*direverse-engineer*.

c. Penggunaan memori yang banyak.

Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkuat dengan mesin komputer berumur lebih dari 4 tahun.

2.15.1.3 Java Development Kit (JDK)

JDK adalah Perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke bytecode yang dapat dimengerti dan dapat dijalankan oleh JRE (Java Runtime Envirotment). JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis java, namun tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun dengan java.

2.15.1.4 Java Virtual Machine (JVM)

JVM adalah inti dari Bahasa Java. JVM berfungsi untuk mengubah bahasa program yang kawan buat menjadi bahasa mesin, Karena bahasa java adalah bahasa pemrograman tingkat tinggi atau mendekati bahasa manusia maka dibutuhkan pengkonversi ke bahasa mesin. JVM mampu menerjemahkan program yang kawan buat ke hampir semua *platform*. “*write once, run everywhere*” itulah slogan dari bahasa pemrograman java karena adanya JVM.

2.15.1.5 Java Runtime Environment (JRE)

JRE adalah perangkat lunak yang digunakan untuk menjalankan aplikasi yang dibangun menggunakan java. Versi JRE harus sama atau lebih tinggi dengan JDK yang digunakan untuk membangun aplikasi java.

2.15.2 Xampp

XAMPP merupakan salah satu paket instalasi Apache, PHP dan MySQL instant yang dapat kita gunakan untuk membantu proses instalasi ketiga produk tersebut.

1. Sejarah Singkat XAMPP

XAMPP merupakan pengembangan dari LAMP (Linux, Apache, MySQL, PHP dan PERL), XAMPP ini merupakan proyek non-profit yang dikembangkan oleh Apache Friends yang didirikan Kai 'Oswald' Seidler dan Kay Vogelsang pada tahun 2002, proyek mereka ini bertujuan untuk mempromosikan penggunaan Apache web server.

2. Pengertian XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (*localhost*). XAMPP merupakan perangkat yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis.

2.15.3 iReport

iReport adalah utilitas (perkakas) pelaporan, yang dikembangkan di lingkungan Java guna membantu *user* dan *developer* (pengembang) mendesain laporan secara visual. Melalui antar muka yang sederhana namun kaya akan fitur,

iReport menyediakan fungsi yang paling penting untuk membuat laporan yang kompleks dengan mudah, sehingga akan menghemat banyak waktu.

iReport menggunakan library JasperReports untuk membuat laporan. JasperReports, dalam hal ini adalah inti dari iReport. JasperReports adalah utilitas pelaporan *open source* yang paling populer didalam teknologi Java, dan iReport adalah *report designer visual* untuk JasperReports.

a. Fitur fitur iReport

iReport dapat terhubung ke database apapun dan memperoleh data dari sumber data yang berbeda. Selain itu dapat mengekspor laporan ke PDF, XHTML, OpenOffice, MS Word, MS Excel, XML, Text, dan lain lain.

b. Antar muka iReport

User Interface (UI) dari iReport sangat *user friendly*. Hal ini memberikan beberapa langkah mudah untuk menghasilkan laporan dengan menggunakan bantuan *Wizard* Laporan.

Fitur antar muka di iReport termasuk: Fasilitas *Drag-and-drop* (tarik dan letakkan) untuk *Elements Report* (elemen laporan). *Window Palette* yang terdiri dari *Report Elements* dan *Tools* Navigasi yang mudah dari *report objects* melalui *Report Inspector*.

2.16 MariaDB

MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB.

MariaDB adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MariaDB, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

2.16.1 Keistimewaan MariaDB

Semua kemampuan MySQL dimiliki pula oleh MariaDB yakni:

1. Portabilitas.
MariaDB dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka.
MariaDB didistribusikan sebagai perangkat lunak sumber terbuka dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user.
MariaDB dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. Performance tuning.
MariaDB memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data.
MariaDB memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan fungsi.
MariaDB memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan.
MariaDB memiliki beberapa lapisan keamanan seperti level subnetmask seperti nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan.
MariaDB mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris.

Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas.

MariaDB dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).

10. Pelokalan bahasa.

MariaDB dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antarmuka.

MariaDB memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. Klien dan peralatan.

MariaDB dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. Struktur tabel.

MariaDB memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.16.2 Tipe Data

Tipe data adalah skema pengkodean rinci yang dikendalikan oleh DBMS untuk mempresentasikan data yang terorganisir. Setiap data memiliki tipe data, beberapa tipe data sebagai berikut :

Tabel II.11 Tipe Data pada MySQL

No	Tipe Data	Keterangan
----	-----------	------------

1.	INT	Merupakan tipe data angka (<i>Integer</i>) yang berukuran normal. Jangkauan nilainya adalah dari -2147483648 hingga 2147483647
2.	BIGINT	Merupakan type data Integer (angka) yang berukuran besar. Jangkauan nilainya adalah dari -9223372036854775808 hingga 9223372036854775807.
3.	SMALLINT	Tipe data Integer berukuran kecil. Jangkauannya -32768 hingga 32767.
4.	MEDIUMINT	Merupakan type data integer tingkat menengah. Jangkauannya adalah dari -8388608 hingga 8388607.
5	TINYINT	Merupakan type data integer yang berukuran sangat kecil. Jangkauannya adalah dari -128 hingga 127
6.	FLOAT	Merupakan type data untuk bilangan floating-point presisi tunggal. Jankauannya adalah dari -3.402823466E+38 hingga -1.175494351 - 38 untuk bilangan negatif. Untuk bilangan positif antara 0, dan 1.175494351-38hingga 3.402823466E+38.

Tabel II.11 Tipe Data pada MySQL (Lanjutan)

No	Tipe Data	Keterangan
7.	DOUBLE	Merupakan type data untuk bilangan berkoma presisi ganda. Jankauannya adalah dari -1.7976931348623157E+308 hingga -2.2250738585072014E-308 untuk bilangan negatif. Untuk bilangan positif antara 0, dan 2.22507385855072014E-308 hingga 1.7976931348623157E+308.
8.	DECIMAL	Bilangan <i>decimal</i> merupakan bilangan <i>floating-point</i> yang bersifat “ <i>unpacked</i> ”. Yang berarti bilangan disimpan sebagai string, menggunakan satu karakter untuk setiap digitnya. Jangkauan DECIMAL sama dengan DOUBLE.
9.	DATE	Merupakan type data tanggal (date). Jika di MySQL berformat ‘YYYY-MM-DD’. Jangkauannya adalah antara nilai ‘1000-01-01’ hingga ‘9999-12-31’.

10.	DATETIME	Kombinasi dari waktu (time) dan tanggal (date). Jika di MySQL berformat 'YYYY-MM-DD HH:MM:SS'. Jangkauannya adalah dari '1000-01-01 00:00:00' hingga '9999-12-31 23:59:59'.
11.	TIMESTAMP	Sebuah timestamp. Jangkauannya adalah dari '1970-01-01 00:00:00' hingga suatu waktu di tahun 2037.
12.	TIME	Tipe data waktu, jangkauannya adalah '-838:59:59' hingga '838:59:59'. Time berformat 'HH:MM:SS'.
13.	YEAR	Angka tahun, dalam format 2 digit atau 4 digit (default 4 digit). Nilai yang mungkin adalah 1901 hingga 2155. '0000' pada format 4 digit, dan 1970-2069 pada format 2 digit (70-69).
14.	CHAR	Merupakan type data string yang memiliki lebar tetap. Nilainya adalah dari 1 hingga 255 karakter. Jika ada sisa, maka sisa tersebut diisi dengan spasi. Dengan arti kata spasi dihitung sebagai karakter.
15.	VARCHAR	Merupakan type data string yang memiliki lebar bervariasi. Nilainya adalah dari 1 hingga 255 karakter. Jika Anda mengset panjang karakternya 10 sedangkan data yang disimpan hanya terdiri dari 5 karakter, maka lebar data tersebut hanya 5 karakter saja. Dengan arti kata spasi tidak dihitung sebagai karakter.
16.	TINYBLOB dan TINYTEXT	Sebuah BLOB (catatan atau gambar) atau TEXT dengan lebar maksimum 255 ($2^8 - 1$) karakter.

Tabel II.11 Tipe Data pada MySQL (Lanjutan)

No	Tipe Data	Keterangan
17.	BLOB dan TEXT	Sebuah BLOB atau TEXT dengan panjang karakter maksimum 65535 ($2^{16}-1$) karakter. BLOB atau TEXT mendukung untuk karakter unik seperti derajat celcius (°C), Gender Male (?) atau Female (?), dan karakter unik lainnya.
18.	MEDIUMBLOB dan MEDIUMTEXT	Sebuah BLOB atau TEXT dengan panjang karakter maksimum 16777215 ($2^{24} - 1$) karakter.
19.	LONGBLOB dan LONGTEXT	Sebuah BLOB atau TEXT dengan panjang karakter maksimum 4294967295 ($2^{32} - 1$) karakter.
20.	ENUM	Type data ini merupakan sebuah enumerasi, yaitu objek string yang hanya dapat memiliki nilai yang dipilih dari daftar nilai 'value1', 'value2', ..., NULL atau nilai special "" error. Sebuah ENUM maksimal dapat memiliki 65535 jenis nilai.
21.	SET	Type data ini merupakan sebuah SET, yaitu objek

		string yang dapat memiliki 0 nilai atau lebih, yang harus dipilih dari daftar nilai 'value1', 'value2', ... Sebuah SET maksimum dapat menampung 64 anggota.
--	--	--

(Sumber : Sutaji, 2012)

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Menurut Sugiyono (2013) metode penelitian merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Metodologi penelitian juga dikenal sebagai metode ilmiah dalam mencari, mengembangkan dan menguji sebuah kebenaran pengetahuan. Metodologi penelitian juga membuat penelitian lebih terarah.

Metodologi penelitian berasal dari bahasa Yunani “*metodos*”, kata ini terdiri dari dua suku kata yaitu “*metha*” yang berarti melalui atau melewati dan “*hodos*” yang berarti jalan atau cara. Metode berarti suatu jalan yang dilalui untuk mencapai tujuan. Metodologi adalah ilmu atau cara yang digunakan untuk memperoleh kebenaran, tergantung dari realitas yang sedang dikaji.

3.2 Identifikasi Masalah

Identifikasi Masalah adalah salah satu proses penelitian yang boleh dikatakan paling penting di antara proses lain. Masalah penelitian akan menentukan kualitas dari penelitian, bahkan juga menentukan apakah sebuah kegiatan bisa disebut penelitian atau tidak. Masalah penelitian secara umum bisa ditemukan lewat studi literatur atau lewat pengamatan lapangan (observasi, survei dan sebagainya).

3.3 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 diperoleh secara langsung maupun tidak langsung dengan menggunakan teknik

pengumpulan data. Teknik pengumpulan data dalam rangka pengumpulan informasi mengenai objek penelitian ini, yaitu :

1. Observasi

Tahap ini merupakan tahap melakukan pengamatan langsung terhadap pengguna sistem dan pengamatan langsung terhadap sistem yang sedang berjalan di PT Coca Cola Distribution Indonesia. Observasi ini dilakukan pada divisi HRD untuk mengetahui alur proses kontrol sistem absensi dan penggajian karyawan yang meliputi proses kegiatan pendaftaran, pengambilan data, hingga proses penghitungan gaji.

2. Wawancara

Pada tahap ini penulis melakukan wawancara dengan para pengguna sistem yang sedang berjalan tersebut untuk memperoleh keterangan mengenai alur proses kegiatan kontrol absensi dan penggajian serta mengenai kebutuhan sistem yang akan dikembangkan selanjutnya. Wawancara ini dilakukan pada divisi HRD di PT Coca Cola Distribution Indonesia

3. Sampling Data

Merupakan sebuah teknik pengumpulan dokumen-dokumen yang diminta secara langsung kepada pihak-pihak yang terkait dengan proses sistem berjalan pada bagian HRD.

4. Studi Pustaka

Tahap ini merupakan tahap awal pengumpulan data dari sumber-sumber seperti buku, jurnal, yang berhubungan dengan masalah yang sedang dianalisis. Studi pustaka yang dilakukan berkaitan dengan analisis dan desain sistem informasi, pemrograman Java, pemrograman MariaDB, Konsep dasar absensi, teori dasar *finger print* dan pembacaan sidik jari, pengembangan perangkat lunak, dan juga beberapa contoh *coding* program melalui media *internet*.

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem adalah menyusun suatu sistem baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah berjalan (Arikunto, 2006).

Pada penelitian ini untuk mengatasi masalah yang ada pada sistem, diputuskan untuk membuat pengembangan sistem. Dalam pengembangan sistem ini digunakan metode *evolutionary prototype*. Alasan digunakannya metode *evolutionary prototype* karena dalam kegiatannya metode ini akan terus menerus diperbaiki sampai semua kriteria sistem yang baru terpenuhi.

Tahapan-tahapan dalam *evolutionary prototype* adalah sebagai berikut:

1. Identifikasi kebutuhan *user*
Pengembang melakukan diskusi dengan bagian Hrd tentang kelemahan sistem absensi saat ini dan kebutuhan *user* terhadap proses-proses pada sistem absensi karyawan yang belum terpenuhi pada sistem saat ini.
2. Membuat *prototype*
Pengembang membuat *prototype* dari sistem yang telah dijelaskan oleh bagian Hrd sebagai calon *user*.
3. Menyesuaikan dan evaluasi *prototype* dengan keinginan *user*
Pengembang menanyakan bagian Hrd tentang *prototype* yang sudah dibuat apakah sesuai atau tidak dengan kebutuhan *user*.
4. Menggunakan *prototype*
Prototype yang sudah dikembangkan sesuai dengan kebutuhan *user*, dan di-install untuk digunakan oleh *user*.

3.5 Kerangka Penelitian

Beberapa tahapan yang penulis lakukan dalam melakukan penelitian ini adalah sebagai berikut:

1. Studi Pendahuluan

Studi Pendahuluan merupakan langkah awal yang dilakukan untuk memulai penelitian. Pendahuluan dilakukan dengan membaca buku literatur, *browsing* internet serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan judul dan permasalahan tugas akhir. Pendahuluan ini dimaksudkan untuk dapat mengetahui suatu gambaran yang jelas untuk mengetahui masalah yang sedang dihadapi. Langkah-langkah yang dilakukan dalam studi yang bertujuan untuk mendapatkan informasi adalah dengan cara observasi dan melakukan wawancara terhadap pihak-pihak terkait yang berada di PT Coca Cola Distribution Indonesia.

2. Identifikasi masalah

Pokok dari permasalahan yang menjadi objek penelitian ini adalah mengenai persoalan sistem kontrol absensi karyawan di PT Coca Cola Distribution Indonesia. Identifikasi dari permasalahan tersebut yaitu Proses yang berjalan masih dilakukan secara manual, kesulitan dalam proses pengarsipan data kehadiran karyawan dan rentan kecurangan penitipan absen antar karyawan.

3. Tujuan Penelitian

Tujuan penelitian dari tugas akhir ini adalah merancang suatu aplikasi absensi karyawan berbasis desktop untuk proses kontrol dan rekapitulasi data kehadiran karyawan, memudahkan divisi HRD dalam proses pengarsipan data absensi karyawan, serta dapat mengurangi kecurangan penitipan absen antar karyawan karena sistem ini membutuhkan sidik jari ketika melakukan absensi.

4. Identifikasi Kebutuhan *User*

Identifikasi kebutuhan *user* merupakan langkah lanjutan dari pengolahan data, dimana semua spesifikasi sistem dan kebutuhannya dituangkan ke dalam sebuah dokumen persyaratan perangkat lunak. Identifikasi kebutuhan user didapatkan melalui dua proses yaitu proses pengumpulan data dan pengolahan data. Pengumpulan data didapatkan melalui gambaran umum tentang proses absensi karyawan, laporan kehadiran karyawan, dan *flowchart* absensi

karyawan. Selain itu pada pengolahan data dapat dilihat dari *usecase diagram* sistem absensi karyawan.

5. Pembuatan *Prototype*

Membuat sebuah prototipe sesuai dengan analisis yang dilakukan. Pada tahap ini akan dirancang aplikasi tentang sistem informasi dan penggajian karyawan. Pembuatan prototipe dijabarkan menjadi 4 tahap. Tahap pertama adalah pemodelan sistem yang dibuat dengan *usecase diagram*, *activity diagram*, dan *sequence diagram*. Tahap selanjutnya adalah pemodelan data yang terdiri dari *class diagram* dan kamus data. Setelah dilakukan pemodelan data maka akan dilanjutkan dengan pemodelan implementasi dengan *deployment diagram*. Setelah ketiga tahap tadi dilakukan maka tahap terakhir adalah perancangan program dengan HIPO (*Hierarchy plus Input-Process-Output*), *flowchart* program dan rancangan program.

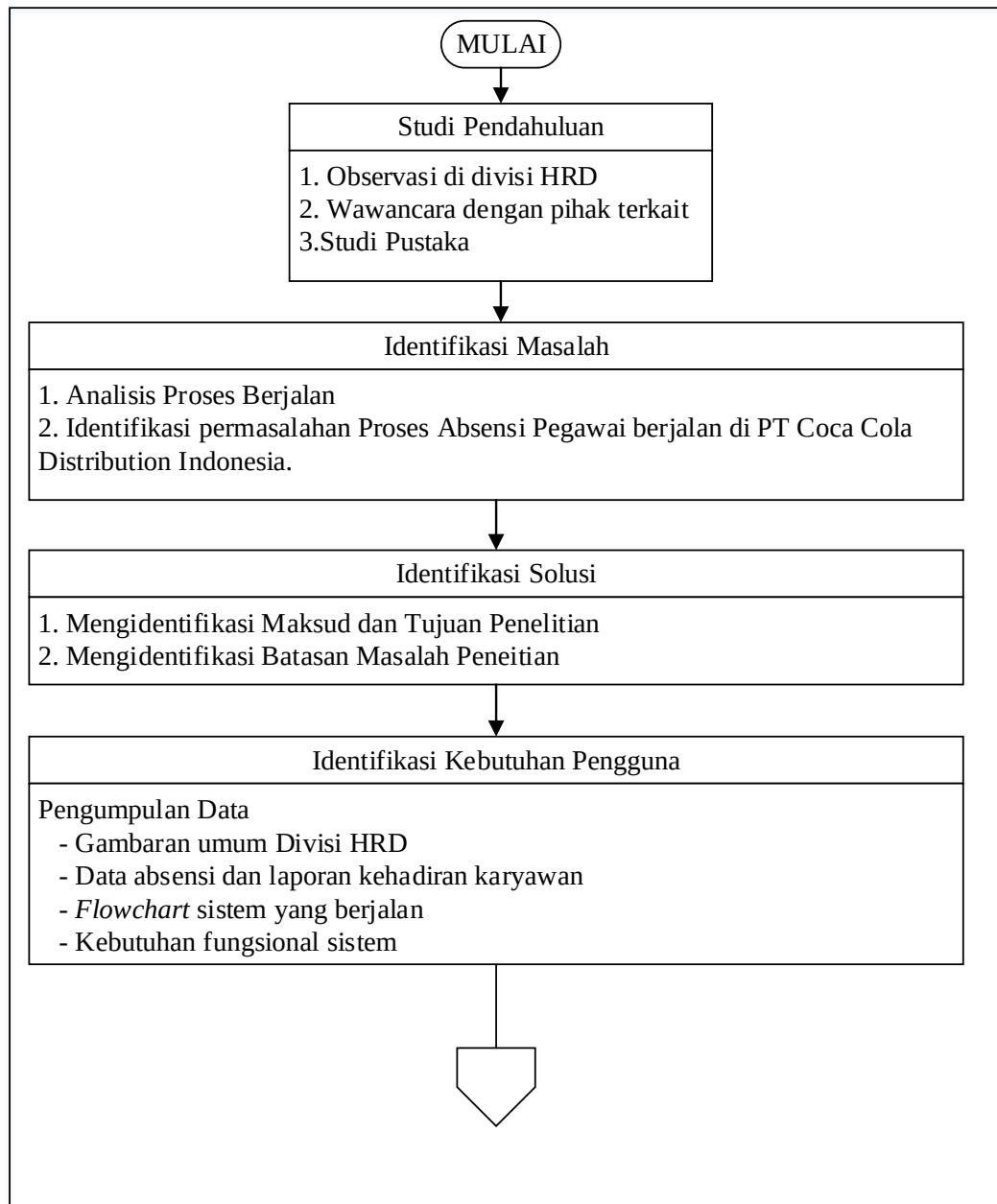
6. Menggunakan *Prototype*

Pada tahap ini, *prototype* dari sistem yang diusulkan akan dilihat apakah sistem usulan tersebut sudah sesuai dengan keinginan pengguna atau belum. Jika belum, maka akan dilakukan pengumpulan dan menganalisis data kembali dan mencari tahu kebutuhan apa yang dibutuhkan oleh pengguna sistem, serta melakukan perancangan sistem ulang. Sedangkan jika sudah sesuai keinginan *user*, maka *prototype* tersebut dapat digunakan. Kemudian akan dibuat spesifikasi berupa *hardware* dan *software* yang menunjang prototipe ini.

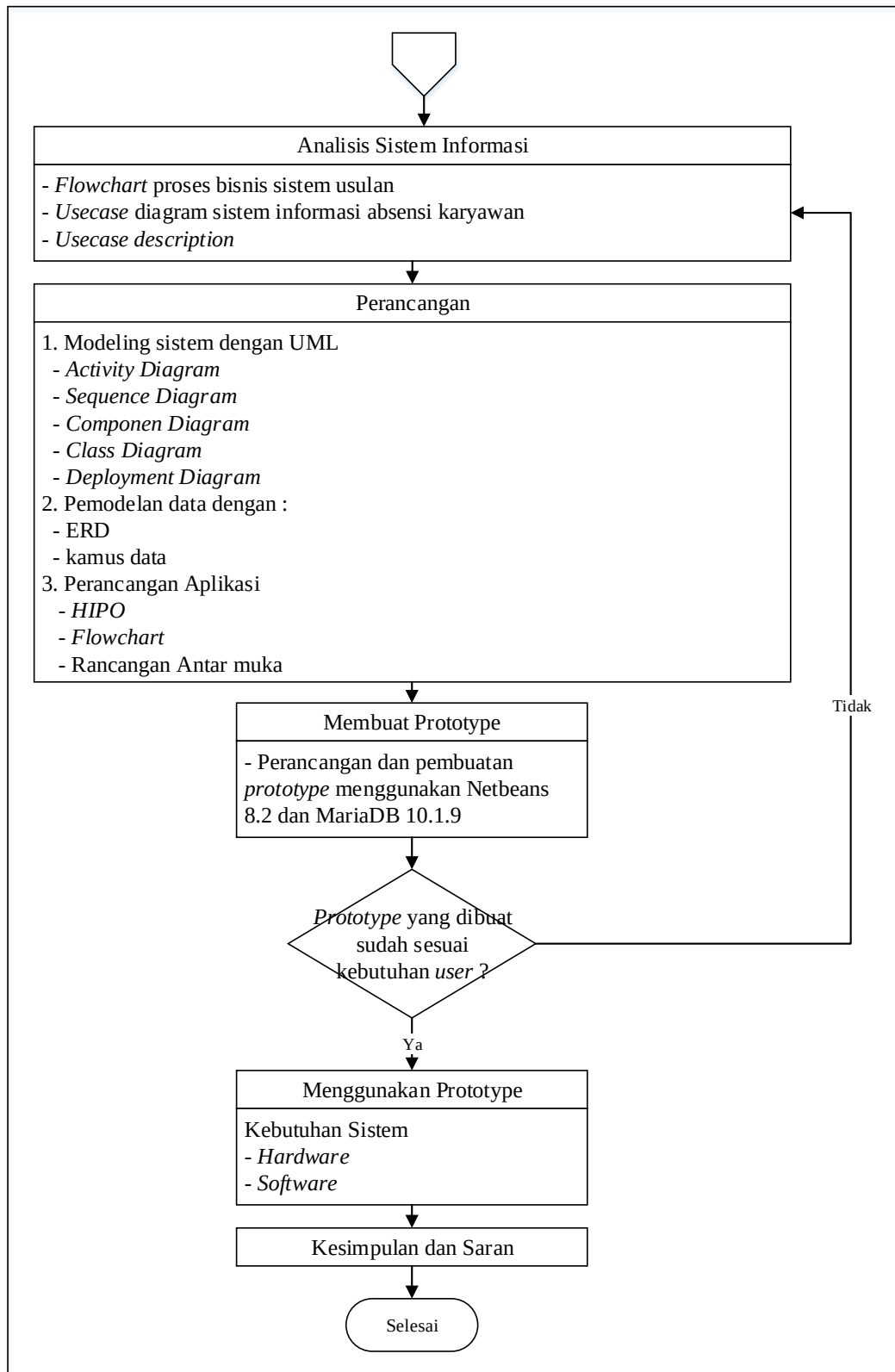
7. Kesimpulan dan Saran

Kesimpulan digunakan untuk membandingkan hasil penelitian atau pengembangan sistem dengan sistem sebelumnya. Sedangkan saran berisi masukan untuk pengembangan sistem selanjutnya.

Berikut adalah Gambar III.1 *flowchart* kerangka penelitian untuk menyelesaikan masalah dalam Tugas Akhir ini:



Gambar III.1 Kerangka Penelitian
Sumber: Pengumpulan dan Pengolahan Data (2017)



Gambar III.1 Kerangka Penelitian
Sumber: Pengumpulan dan Pengolahan Data (2017)

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Berdasarkan hasil pengolahan data dan analisis sistem, PT Coca Cola Distribution Indonesia membutuhkan sebuah sistem informasi absensi yang dapat memberikan informasi kepada divisi HRD dalam kontrol kehadiran karyawan dan memudahkan dalam pengelolaan data karyawan. Dengan cara manual yang sekarang berjalan, mengakibatkan peluang terjadinya kecurangan dalam proses absensi oleh karyawan cukup tinggi serta kesulitan dalam pengarsipan data kehadiran. Hal inilah yang dijadikan sebagai *point of interest* dan dikembangkan untuk menjadi bahasan dalam penelitian ini, dengan merancang sistem informasi absensi *fingerprint* diharapkan mampu membantu menyelesaikan permasalahan-permasalahan yang dihadapi oleh PT Coca Cola Distribution Indonesia yang berkaitan dengan sistem absensinya.

Tabel V.1 Kebutuhan Sistem Absensi Karyawan

Kebutuhan Sistem	
<i>Project Name</i>	Sistem Absensi Karyawan <i>Fingerprint</i>
<i>Business Need</i>	1. Terwujudnya sebuah sistem informasi untuk proses kontrol dan rekapitulasi data kehadiran karyawan menggunakan sistem yang terkomputerisasi. 2. Terwujudnya sebuah sistem informasi bagi divisi HRD dalam proses pengarsipan data absensi karyawan tanpa kertas. 3. Terwujudnya sebuah sistem informasi yang dapat mengurangi kecurangan titip absen antar karyawan karna sistem ini membutuhkan sidik jari.
<i>Business Requirement</i>	1. Memberikan sistem informasi absensi karyawan menggunakan metode sidik jari yang dapat membantu HRD dalam proses kontrol kehadiran karyawan.

Kebutuhan Sistem	
<i>Bussiness Value</i>	1. Memudahkan dalam proses observasi daftar kehadiran serta rekapitulasi kehadiran bulanan menggunakan sistem yang terkomputerisasi. 2. Membantu pengarsipan data kehadiran karyawan melalui sebuah aplikasi 3. Memberikan sebuah sistem informasi absensi karyawan yang dapat mengurangi kecurangan titip absen.
<i>Special Issues or Constrains</i>	-

Sumber: Hasil Analisis Data (2017)

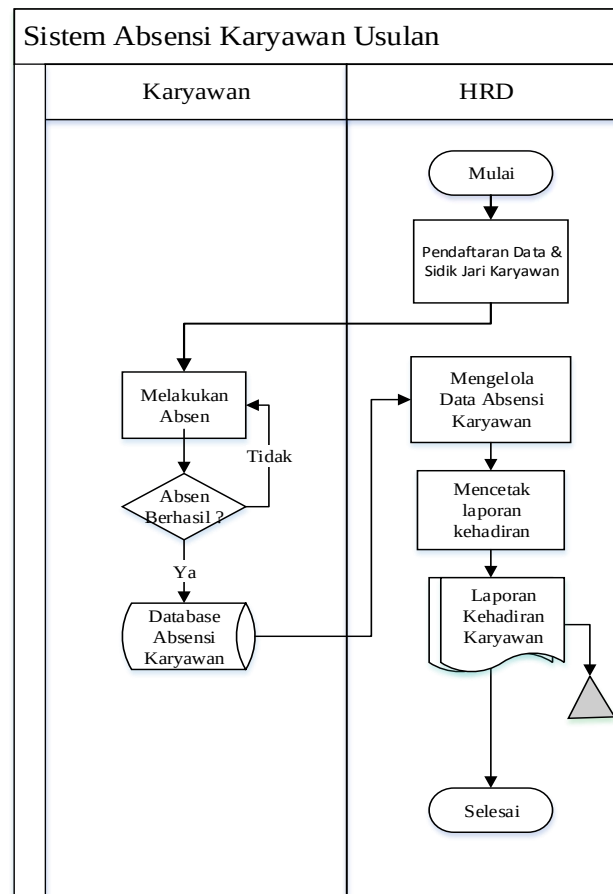
5.2 Perancangan *Flowmap* Sistem Usulan

Prosedur sistem informasi absensi karyawan menggunakan *fingerprint* yang diusulkan melalui tahapan sebagai berikut :

1. Bagian HRD menerima form data diri karyawan,
2. Lalu bagian HRD melakukan pendaftaran ulang data karyawan beserta sidik jari karyawan ke dalam sistem,
3. Setelah data sidik jari terekam kedalam sistem, karyawan melakukan absensi harian menggunakan mesin sidik jari yang telah tersedia,
4. Untuk proses kontrol kehadiran harian, bagian Hrd cukup memilih tanggal kehadiran yang ingin diperiksa,
5. Untuk manajemen lembur, karyawan diharuskan mengisi form lembur yang sudah disetujui oleh atasan lalu diserahkan kepada bagian Hrd untuk proses penambahan data lembur ke dalam sistem,

6. Untuk manajemen cuti, karyawan diharuskan mengisi form cuti yang sudah disetujui oleh atasan lalu diserahkan kepada bagian Hrd untuk proses penambahan data cuti ke dalam sistem,
7. Untuk manajemen sakit, karyawan diharuskan lapor kepada atasan dan menyertakan surat keterangan dokter ke bagian Hrd setelah bisa masuk kerja kembali untuk proses penambahan data sakit ke dalam sistem,
8. Untuk manajemen izin, karyawan diharuskan lapor kepada atasan dan menyertakan surat izin tidak masuk yang sudah disetujui ke bagian Hrd setelah bisa masuk kerja kembali untuk proses penambahan data izin ke dalam sistem,
9. Secara otomatis sistem menghitung jumlah ketidakhadiran karyawan juga jumlah hari kerja efektif yang akan dijadikan sebagai laporan kehadiran bulanan yang bisa digunakan untuk kebutuhan penghitungan gaji karyawan.

Seperti yang terlihat pada Gambar V.1 adalah *Flowmap* Sistem Informasi Absensi Karyawan menggunakan *fingerprint* usulan sebagai berikut:



Gambar V.1 *Flowmap* Sistem Absensi Karyawan usulan
 Sumber: Hasil Analisis Data (2017)

5.3 Analisis dan Perancangan Sistem Usulan

Berdasarkan hasil pengolahan data dan analisis sistem, PT Coca Cola Distribution Indonesia membutuhkan sebuah sistem informasi absensi yang dapat memberikan informasi kepada divisi HRD dalam kontrol kehadiran karyawan dan memudahkan dalam pengelolaan data karyawan. Dengan cara manual yang sekarang berjalan, mengakibatkan peluang terjadinya kecurangan dalam proses absensi oleh karyawan cukup tinggi serta kesulitan dalam pengarsipan data kehadiran. Hal inilah yang dijadikan sebagai *point of interest* dan dikembangkan untuk menjadi bahasan dalam penelitian ini, dengan merancang sistem informasi absensi *fingerprint*

diharapkan mampu membantu menyelesaikan permasalahan-permasalahan yang dihadapi oleh PT Coca Cola Distribution Indonesia yang berkaitan dengan sistem absensinya.

Dalam perancangan sistem informasi absensi *fingerprint* ini metode pengembangan sistem yang digunakan adalah *prototype evolutioner*. Metode pengembangan sistem ini digunakan karena sesuai untuk mengakomodir permasalahan-permasalahan dalam perancangan sistem informasi absensi *fingerprint* ini, dalam perancangan sistem informasi absensi *fingerprint* perubahan serta penambahan kebutuhan pengguna pasti akan terjadi, metode pengembangan sistem ini mampu menerima perubahan yang terjadi khususnya dalam perancangan sistem informasi absensi *fingerprint*. Dalam pengembangan sistem informasi absensi menggunakan *fingerprint*, terdapat 4 tahapan yang digunakan, yaitu :

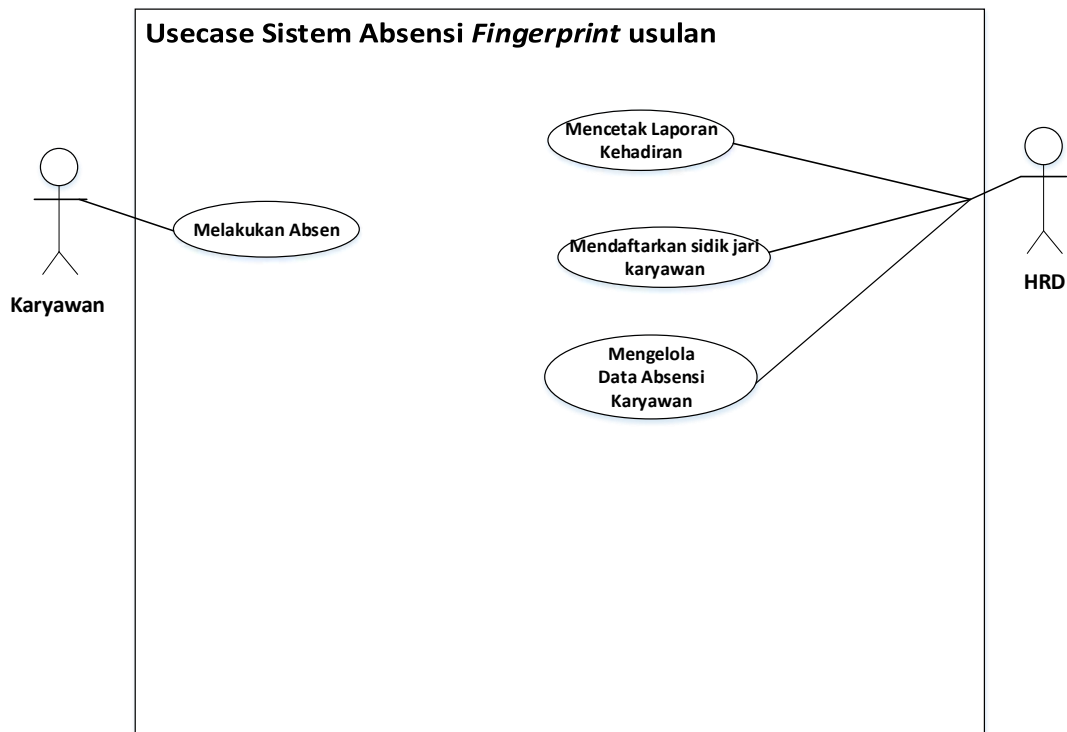
1. Pembuatan model sistem berbasis objek dengan *Unified Modelling Language* (UML) menggunakan *use case diagram*, *activity diagram*, *sequence diagram*, dan *deployment diagram*.
2. Pembuatan model data dengan *class diagram* dan kamus data.
3. Perancangan program yang diusulkan dengan tahapan sebagai berikut: Pembuatan HIPO (*Hierarchy plus Input-Process-Output*) untuk merancang fungsi dari modul-modul sistem, *flowchart* sistem dan perancangan *interface* program menggunakan *Microsoft Office Visio 2010*.
4. Pembuatan aplikasi menggunakan Java Netbeans 8.2 dan Mariadb 10.1.9 sebagai basis 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 *fingerprint* berdasarkan analisis.

5.3.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu

atau lebih *actor* dengan sistem informasi yang dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. *Use case* sistem informasi absensi karyawan di PT. Coca Cola Distribution Indonesia yang diusulkan dapat digambarkan seperti pada Gambar V.2 berikut :



Gambar V.2 *Use case Diagram* Sistem Absensi Karyawan Usulan

Sumber: Hasil Analisis Data (2017)

Penjelasan *use case diagram* sistem saat ini, dapat dijelaskan sebagai berikut:

1. Definisi Aktor

Berikut ini adalah deskripsi pendefinisian aktor pada sistem informasi absensi karyawan

Tabel V.2 Definisi Aktor

No.	Aktor	Deskripsi
1.	Karyawan	Karyawan dapat melakukan absensi masuk kerja dan absensi jam pulang kerja.

2.	HRD	HRD dapat melakukan kegiatan apapun yang ada didalam sistem dalam pengelolaan data karyawan, seperti Tambah data karyawan, Ubah Data, Hapus Data, Simpan Data, <i>manage</i> data lembur dan cuti karyawan serta cetak laporan absensi karyawan.
----	-----	--

Sumber: Hasil Analisis Data (2017)

2. Definisi Use Case

Pendefinisian *use case* pada *use case* sistem usulan, dapat dilihat pada berikut ini.

Tabel V.3 Definisi Usecase Diagram

No.	Use Case	Deskripsi
1.	Login	Proses untuk melakukan <i>login</i> pada aplikasi
2.	Mengelola Data Karyawan	Manajemen data karyawan termasuk input, edit, simpan dan hapus.
3.	Mengelola Data Lembur	Manajemen data lembur termasuk input, edit, simpan dan hapus.
4.	Mengelola Leave	Manajemen data cuti, izin dan sakit termasuk input, edit, simpan dan hapus.
5.	Melakukan Absen	Merupakan proses pendataan kehadiran karyawan.
6.	Mencetak Laporan Kehadiran	Merupakan proses mencetak laporan kehadiran berdasarkan periode (harian atau bulanan).

Sumber: Hasil Analisis Data (2017)

3. Skenario Use Case

Skenario jalannya masing-masing *use case* pada perancangan sistem informasi absensi karyawan usulan dapat dilihat pada tabel berikut:

1. ***Use Case Diagram* Manajemen Karyawan**

Berikut adalah skenario *usecase* Manajemen Karyawan yang terdapat pada Tabel V.4:

Tabel V.4 Skenario *Usecase* Data Karyawan

Nama <i>Use Case</i>	Data Karyawan
<i>Primary Actor</i>	HRD
<i>Use Case Description</i>	Use case ini menggambarkan proses tambah, hapus, ubah dan simpan data karyawan.
<i>Normal Flow of Events</i>	1. <i>User</i> melakukan login 2. Sistem menampilkan menu laporan kehadiran 3. <i>User</i> memilih menu Manajemen. 4. Lalu sistem menampilkan menu Manajemen Karyawan 5. <i>User</i> melakukan pengelolaan Data Karyawan (tambah, hapus, ubah dan simpan) 6. Sistem menampilkan menu Data Karyawan

Sumber: Hasil Analisis Data (2017)

2. **Skenario *Use Case Diagram* Manajemen Lembur**

Berikut adalah skenario *usecase* Manajemen Lembur yang terdapat pada Tabel V.5:

Tabel V.5 Skenario *Usecase* Manajemen Lembur

Nama <i>Use Case</i>	Data Lembur
<i>Primary Actor</i>	HRD
<i>Use Case Description</i>	Use case ini menggambarkan proses tambah, hapus, ubah dan simpan data lembur.
<i>Normal Flow of Events</i>	1. <i>User</i> melakukan login 2. Sistem menampilkan menu laporan kehadiran 3. <i>User</i> memilih menu Manajemen. 4. Lalu sistem menampilkan menu Manajemen Lembur 5. <i>User</i> melakukan pengelolaan Data Lembur (tambah, hapus, ubah dan simpan) 6. Sistem menampilkan menu Lembur

Sumber: Hasil Analisis Data (2017)

3. Skenario *Use Case Diagram* Manajemen Leave

Berikut adalah skenario *usecase* Manajemen Leave yang terdapat pada Tabel V.6:

Tabel V.6 Skenario *Usecase* Data Leave

Nama <i>Use Case</i>	Data Leave
<i>Primary Actor</i>	HRD
<i>Use Case Description</i>	Use case ini menggambarkan proses tambah, hapus, ubah dan simpan data cuti, izin dan sakit.
<i>Normal Flow of Events</i>	1. <i>User</i> melakukan login 2. Sistem menampilkan menu laporan kehadiran 3. <i>User</i> memilih menu Manajemen. 4. Lalu sistem menampilkan menu Leave 5. <i>User</i> melakukan pengelolaan Data Cuti/Izin/Sakit (tambah, hapus, ubah dan simpan) 6. Sistem menampilkan menu Leave

Sumber: Hasil Analisis Data (2017)

4. Skenario *Use Case Diagram* Melakukan Absen

Berikut adalah skenario *use case* Absensi Karyawan yang terdapat pada Tabel V.7:

Tabel V.7 Skenario *Usecase* Melakukan Absen

Nama <i>Use Case</i>	Melakukan Absen
<i>Primary Actor</i>	Karyawan
<i>Use Case Description</i>	Use case ini menggambarkan proses absen masuk dan keluar karyawan.
<i>Normal Flow of Events</i>	1. <i>User</i> menempelkan jari yang terdaftar di mesin <i>finger print</i> 2. Sistem menampilkan verifikasi sidik jari

Sumber: Hasil Analisis Data (2017)

5. Skenario *Use Case Diagram* Laporan Kehadiran

Berikut adalah skenario *usecase* Laporan Kehadiran yang terdapat pada Tabel V.8:

Tabel V.8 Skenario *Usecase* Mencetak Laporan Kehadiran

Nama <i>Use Case</i>	Mencetak Laporan Kehadiran
----------------------	----------------------------

<i>Primary Actor</i>	HRD
<i>Use Case Description</i>	Use case ini menggambarkan proses mencetak laporan kehadiran.
<i>Normal Flow of Events</i>	1. <i>User</i> melakukan <i>login</i> 2. Sistem menampilkan Menu laporan 3. <i>User</i> memilih periode penarikan Data Kehadiran lalu pilih tampil

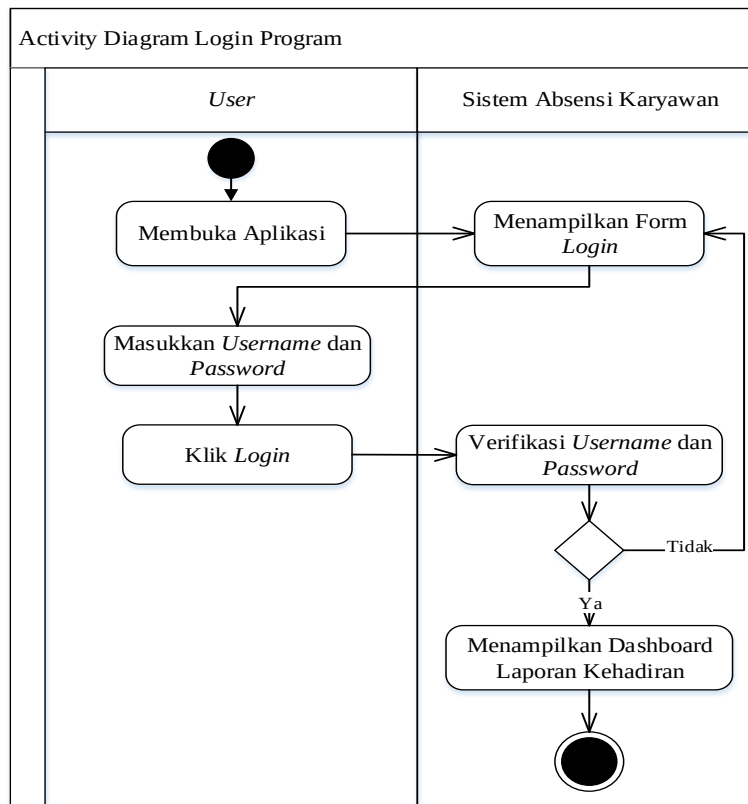
Nama <i>Usecase</i>	Mencetak Laporan Kehadiran
	4. Sistem menampilkan Laporan Kehadiran karyawan 5. <i>User</i> memilih tombol Cetak Laporan 6. Sistem menampilkan Laporan Kehadiran.

Sumber: Hasil Analisis Data (2017)

5.3.2 *Activity Diagram*

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Diagram Aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan *actor*, melainkan aktifitas yang dapat dilakukan oleh sistem. Berikut ini adalah *Activity Diagram* dari Sistem Absensi dan Penggajian PT Coca Cola Distribution Indonesia :

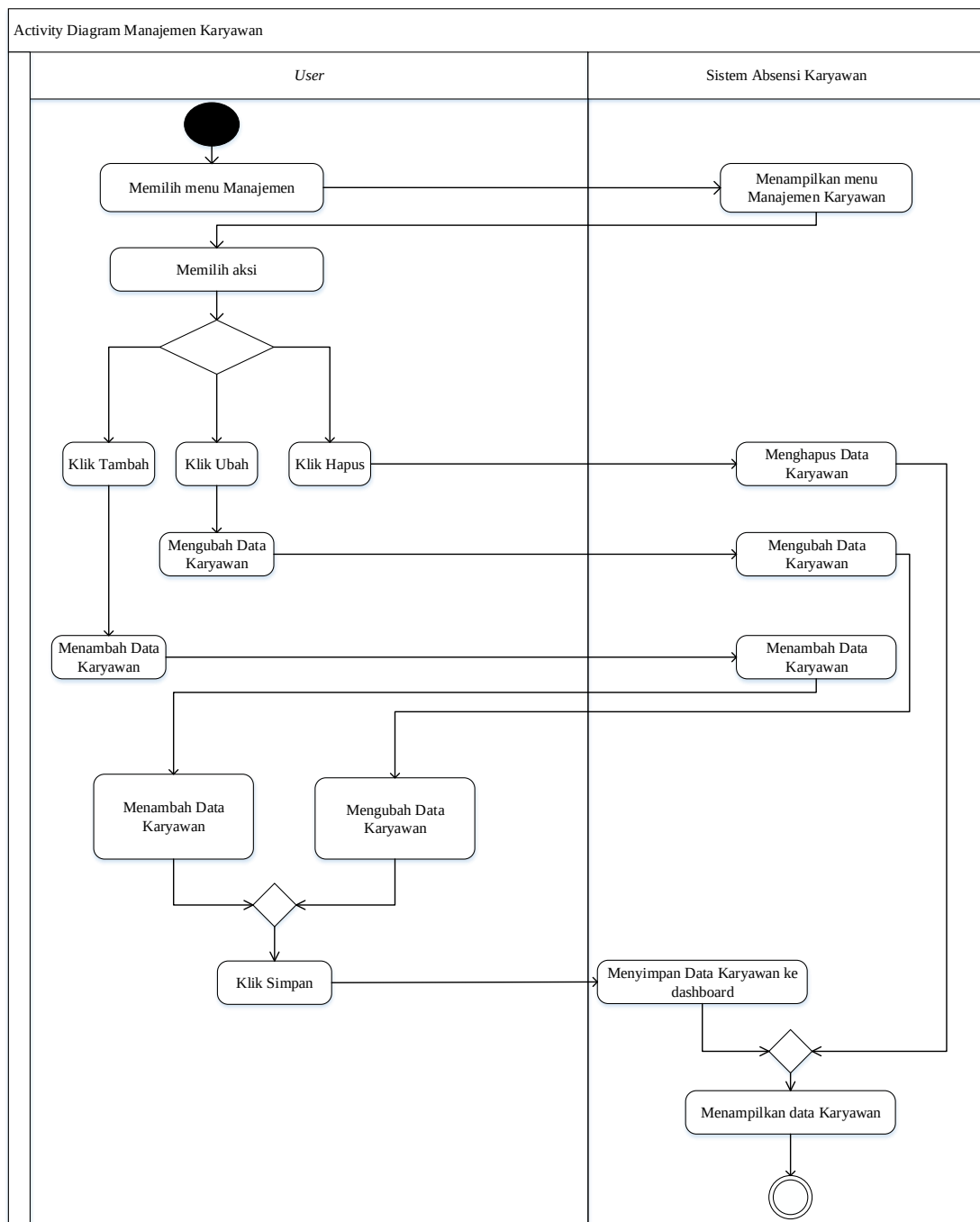
1. *Activity Diagram Login*



Gambar V.3 Activity Diagram Login Program

Sumber: Hasil Analisis Data (2018)

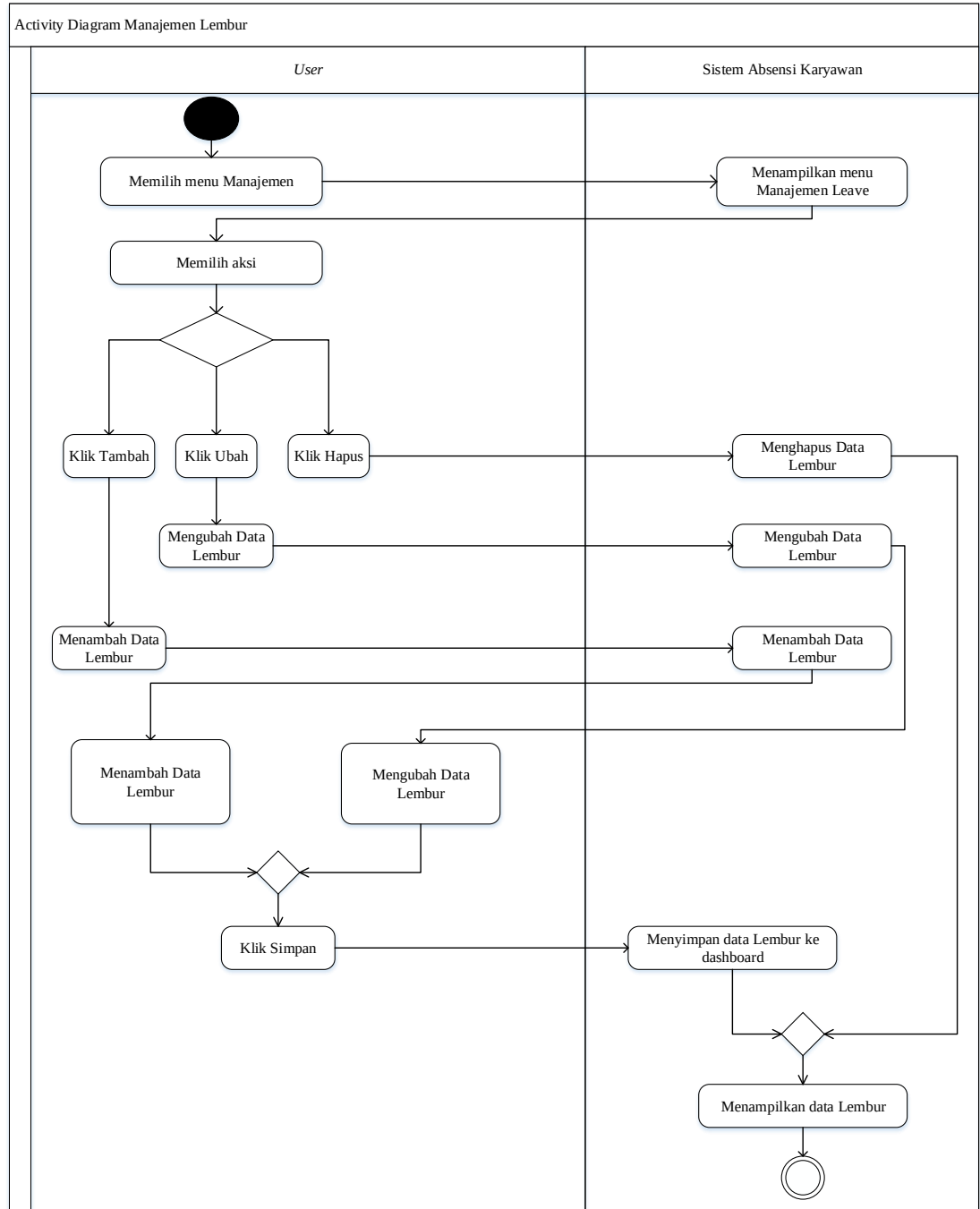
2. Activity Diagram Manajemen Karyawan



Gambar V.4 Activity Diagram Manajemen Karyawan

Sumber: Hasil Analisis Data (2018)

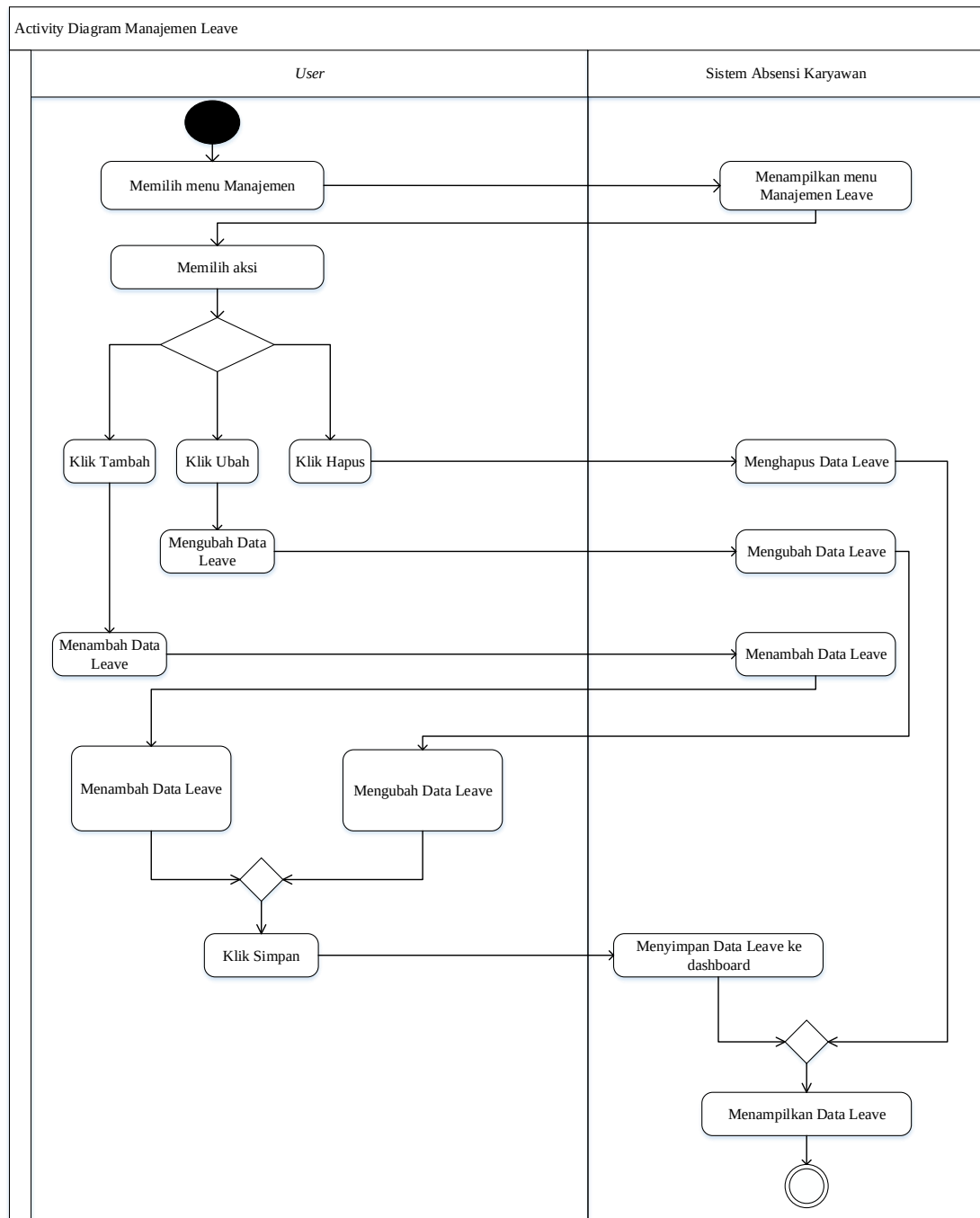
3. Activity Diagram Manajemen Lembur



Gambar V.5 Activity Diagram Manajemen Lembur

Sumber: Hasil Analisis Data (2018)

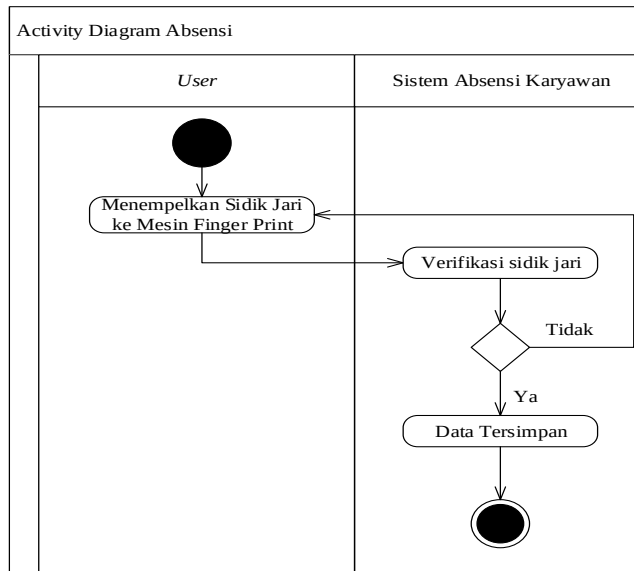
4. *Activity Diagram* Manajemen Leave



Gambar V.6 Activity Diagram Manajemen Leave

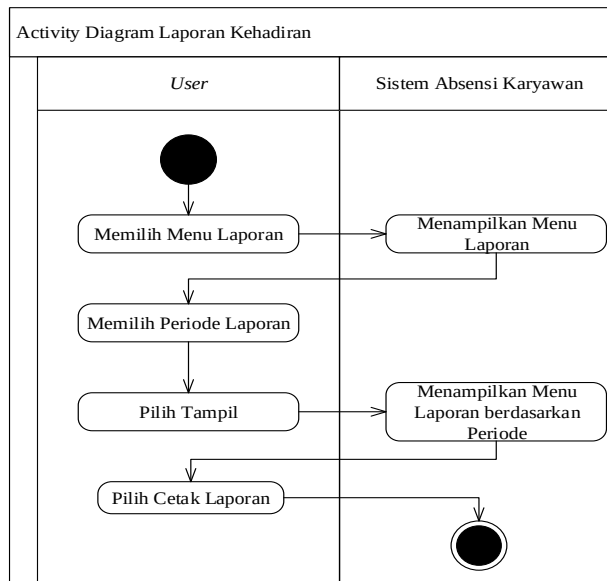
Sumber: Hasil Analisis Data (2018)

5. *Activity Diagram Absen*



Gambar V.7 *Activity Diagram Absen*
Sumber: Hasil Analisis Data (2018)

6. *Activity Diagram Laporan Kehadiran*

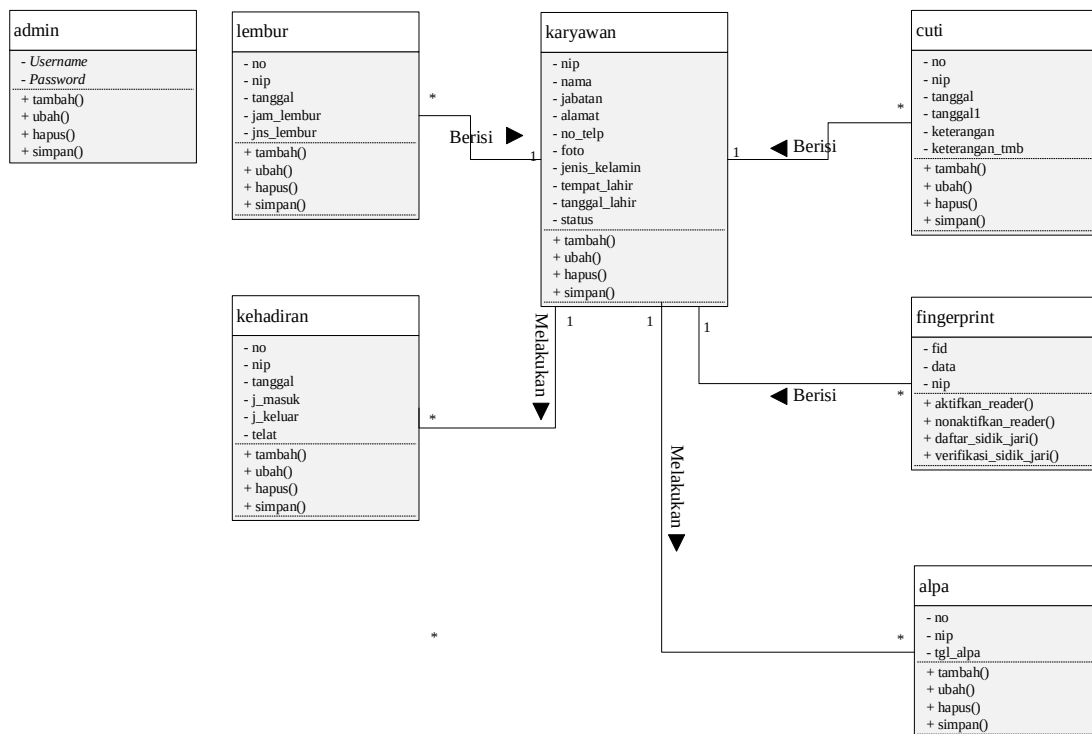


Gambar V.8 *Activity Diagram Laporan Kehadiran*

Sumber: Hasil Analisis Data (2018)

5.4 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 Absensi dan Penggajian yang diusulkan dapat dilihat pada Gambar V.8 sebagai berikut:



Gambar V.9 *Class Diagram* Sistem Informasi Absensi Karyawan Usulan
Sumber: Hasil Analisis Data (2018)

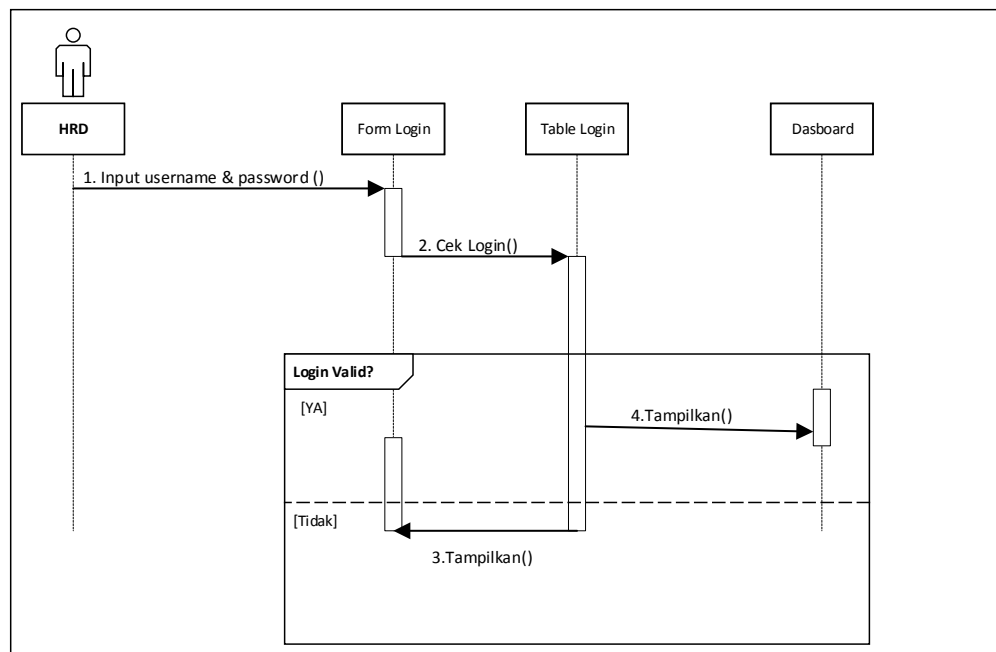
5.5 Sequence Diagram

Sequence Diagram menggambarkan interaksi yang terjadi pada suatu objek *use case diagram* ketika melakukan suatu proses tertentu, dimana 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 Absensi Karyawan :

1. *Sequence Diagram* Login

Sequence diagram login menjelaskan sebuah *sequence diagram* dalam proses login admin. Adapun *sequence diagram* dari mengelola data karyawan dapat dilihat pada Gambar V.10 sebagai berikut:

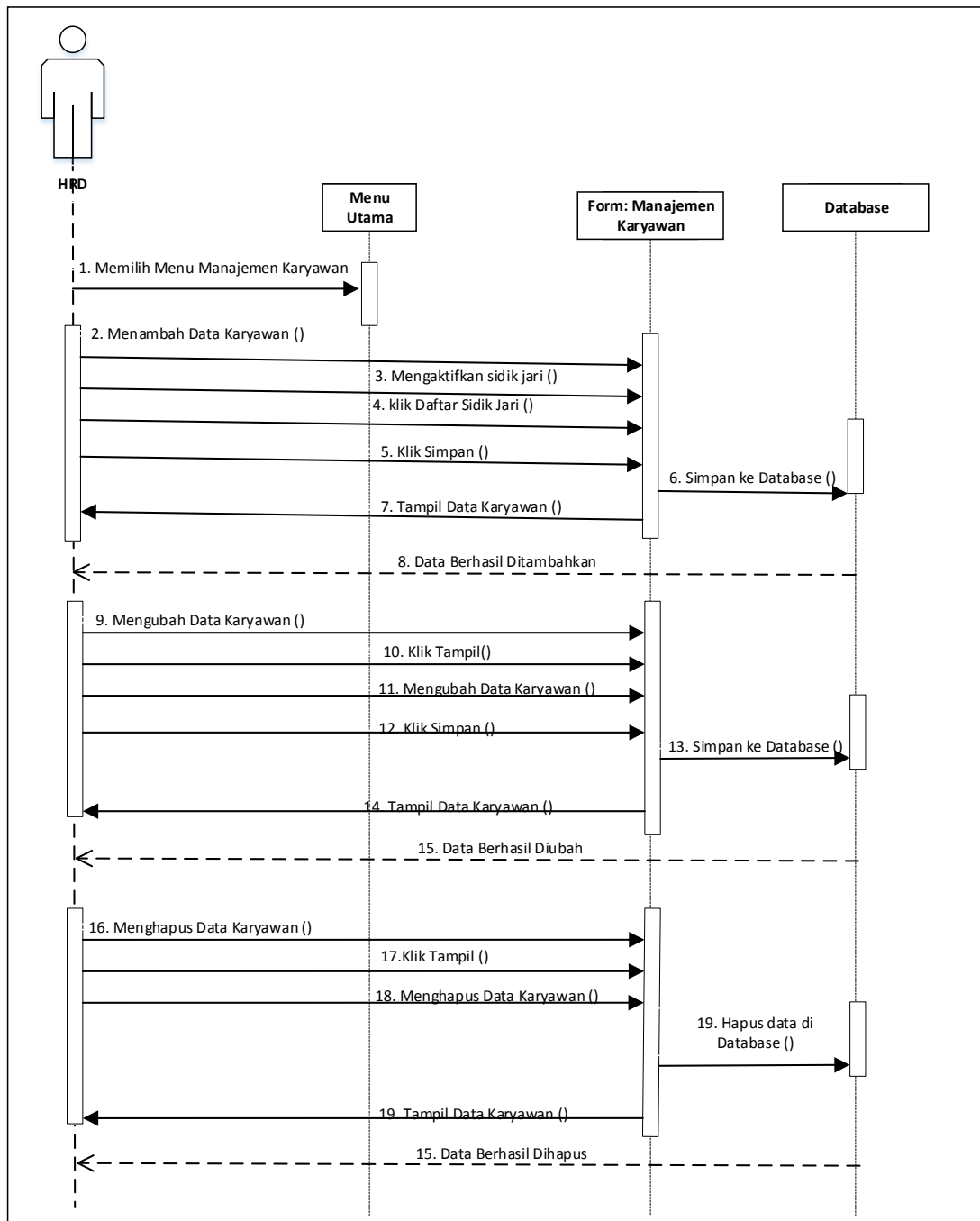


Gambar V.10 *Sequence Diagram* Proses Login Admin

Sumber: Hasil Analisis Data (2018)

2. *Sequence Diagram* Manajemen Karyawan

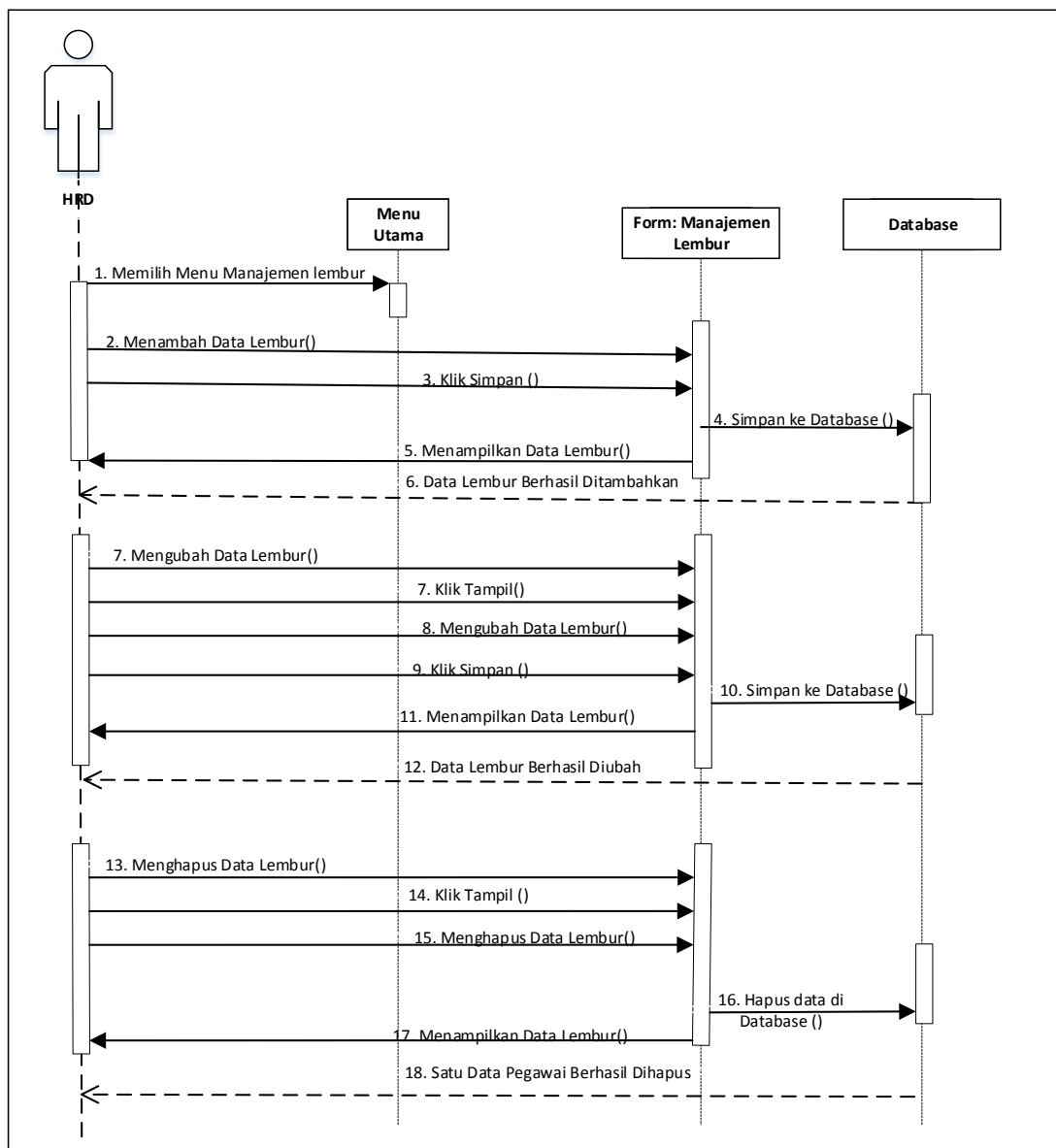
Sequence diagram mengelola data karyawan menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah, menghapus, menyimpan dan mengubah data karyawan. Adapun *sequence diagram* dari mengelola data karyawan dapat dilihat pada Gambar V.11 sebagai berikut:



Gambar V.11 *Sequence Diagram* Proses Manajemen Karyawan
Sumber: Hasil Analisis Data (2018)

3. Sequence Diagram Manajemen Lembur

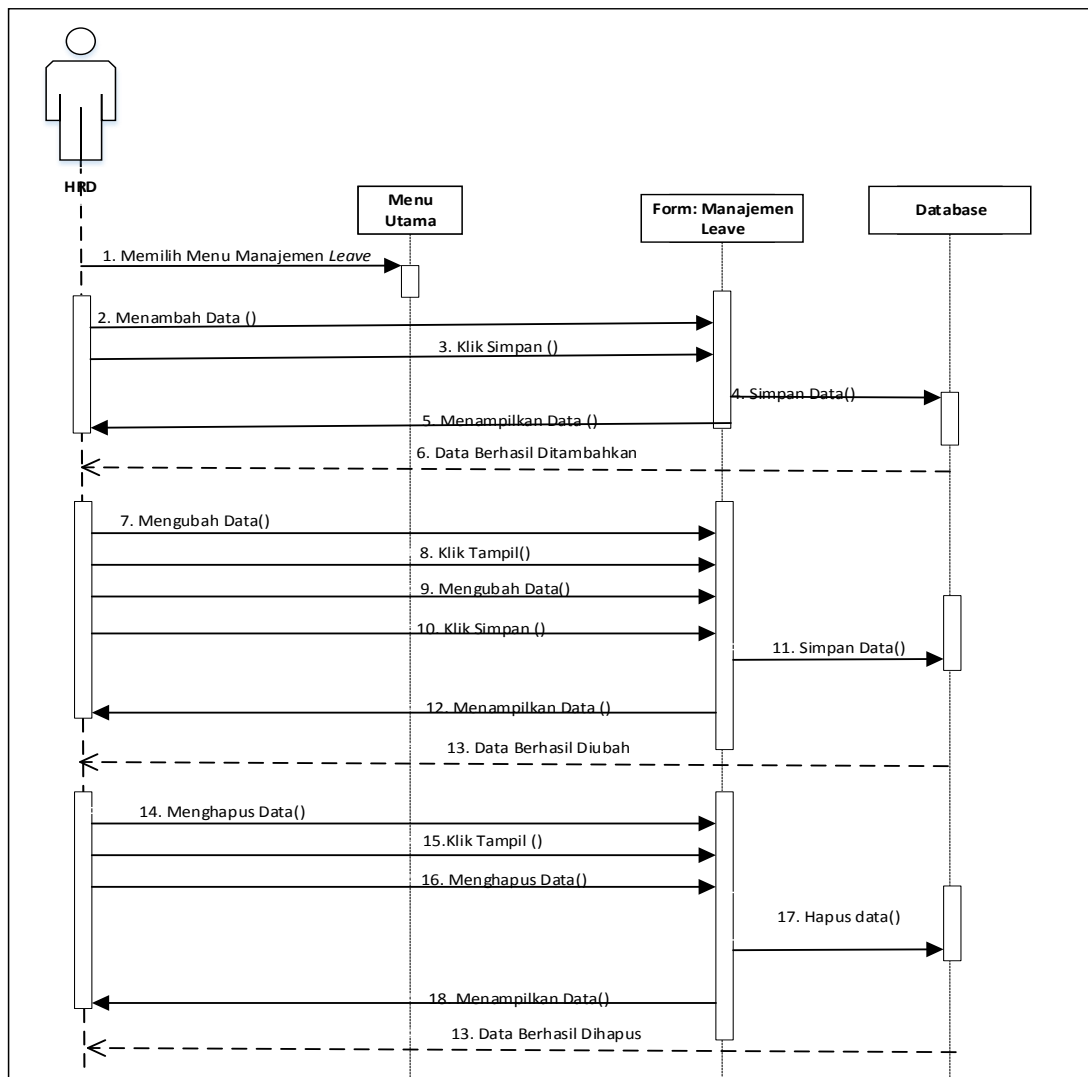
Sequence diagram mengelola data lembur menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah, menghapus, menyimpan dan mengubah data lembur. Adapun *sequence diagram* dari mengelola data lembur dapat dilihat pada Gambar V.12 sebagai berikut:



Gambar V.12 *Sequence Diagram* Proses Data Lembur
Sumber: Hasil Analisis Data (2018)

4. *Sequence Diagram* Manajemen Leave

Sequence diagram mengelola data *leave* menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah, menghapus, menyimpan dan mengubah data cuti, sakit dan izin. Adapun *sequence diagram* dari mengelola data *leave* dapat dilihat pada Gambar V.13 sebagai berikut:

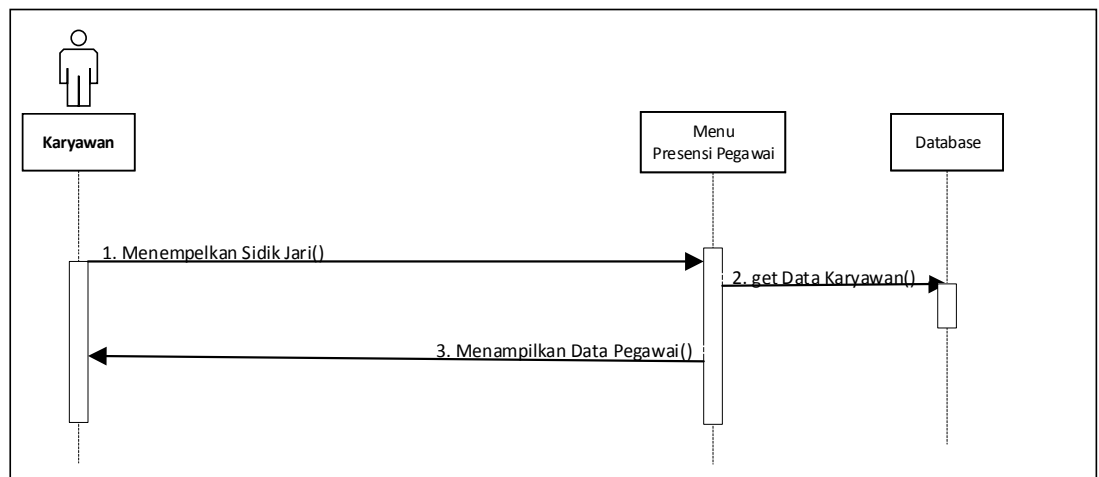


Gambar V.13 *Sequence Diagram* Proses Data Leave

Sumber: Hasil Analisis Data (2018)

5. *Sequence Diagram* Absensi Kehadiran

Sequence diagram absensi kehadiran menjelaskan sebuah *sequence diagram* dalam proses absensi karyawan ke sistem yang dilakukan setiap jam masuk dan jam pulang kerja. Adapun *sequence diagram* dari absensi kehadiran dapat dilihat pada Gambar V.14 sebagai berikut:

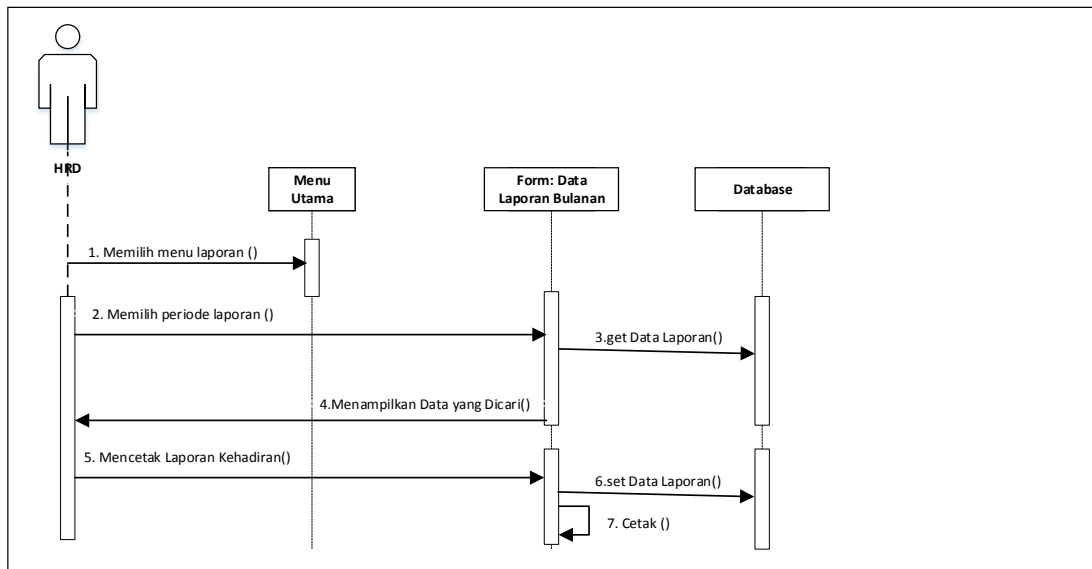


Gambar V.14 *Sequence Diagram* Proses Absensi Kehadiran

Sumber: Hasil Analisis Data (2018)

6. *Sequence Diagram* Cetak Laporan Kehadiran

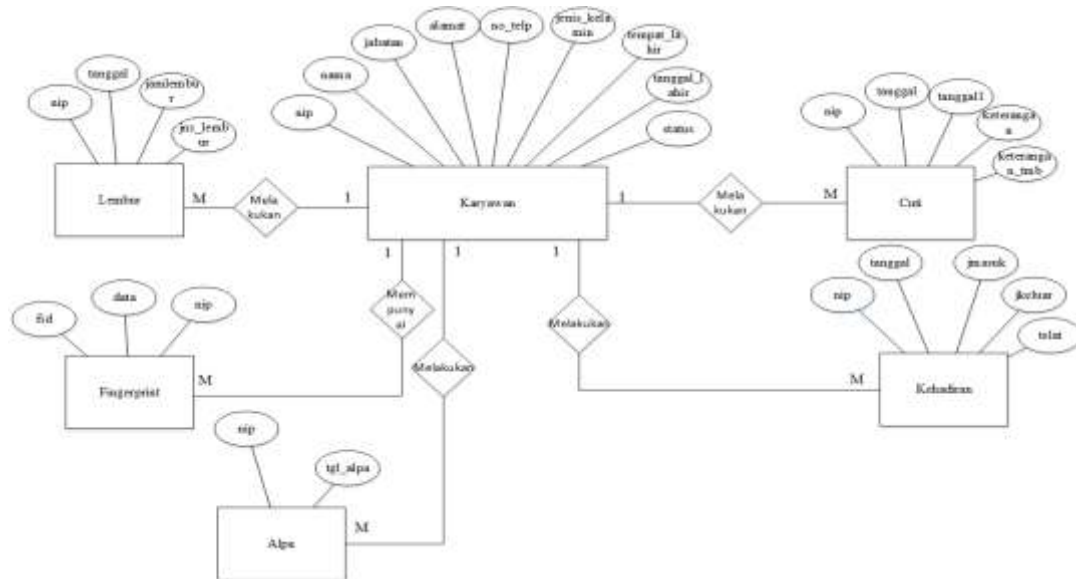
Sequence diagram Cetak Laporan Kehadiran menjelaskan sebuah *sequence diagram* dalam proses cetak laporan kehadiran yang dilakukan oleh HRD. Adapun *sequence diagram* dari cetak laporan kehadiran dapat dilihat pada Gambar V.15 sebagai berikut:



Gambar V.15 *Sequence Diagram* Proses Cetak Laporan Kehadiran
Sumber: Hasil Analisis Data (2018)

5.6 Entity Relationship Diagram

ERD adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional dapat dilihat pada Gambar V.16.



Gambar V.16 *Entity Relationship Diagram*

Sumber: Hasil Analisis Data (2017)

5.7 Kamus Data

Berikut ini adalah tabel-tabel yang digunakan dalam perancangan proses *Unified Modeling Language* (UML).

1. Tabel admin

Nama tabel : admin

Fungsi : Menyimpan data login

Tipe : Data master

Tabel V.9 Tabel Login

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	User Name	username	varchar	20	Primary Key
2	Password	password	varchar	20	Not null

Sumber: Hasil Analisis Data (2017)

2. Tabel Cuti

Nama tabel : cuti

Fungsi : Mengelola data cuti karyawan mulai dari tambah, ubah, hapus, dan simpan.

Tipe : Data Transaksi

Tabel V.10 Tabel Cuti

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	No Cuti	no	Int	11	Primary Key
2	Nip	nip	varchar	10	Foreign Key
3	Tanggal Mulai	tanggal	Date		Not null
4	Tanggal Akhir	tanggal1	Date		Not null
5	Keterangan	keterangan	Varchar	11	Not null
6	Keterangan Detail	keterangan_tmb	Varchar	255	Not null

Sumber: Hasil Analisis Data (2017)

3. Tabel Lembur

Nama tabel : lembur

Fungsi : Mengelola data lembur karyawan mulai dari tambah, ubah, hapus, dan simpan.

Tipe : Data Transaksi

Tabel V.11 Tabel Lembur

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	No Lembur	no	int	11	Primary Key
2	Nip	nip	varchar	10	Foreign Key
3	Tanggal Lembur	tanggal	date		Not null
4	Jam Lembur	jamlembur	int	11	Not null
5	Jenis Lembur	jns_lembur	varchar	20	Not null

Sumber: Hasil Analisis Data (2017)

4. Tabel Kehadiran

Nama tabel : kehadiran

Fungsi : Melakukan absensi kerja

Tipe : Data Absensi

Tabel V.12 Tabel Kehadiran

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	No Kehadiran	no	int	11	Primary Key
2	Nip	nip	varchar	10	Foreign Key
3	Tanggal Kehadiran	tanggal	date		
4	Jam Masuk	jmasuk	time		
5	Jam Keluar	jkeluar	time		

6	Telat	telat	int	11	
---	-------	-------	-----	----	--

Sumber: Hasil Analisis Data (2017)

5. Tabel Karyawan

Nama tabel : karyawan

Fungsi : Mengelola data karyawan mulai dari tambah, hapus, ubah dan Simpan

Tipe : Data Master

Tabel V.13 Tabel Karyawan

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Nip	Nip	varchar	10	Primary Key
2	Nama	Nama	varchar	30	Foreign Key
3	Jabatan	Jabatan	varchar	20	
4	Alamat	Alamat	varchar	255	
5	No Telp	no_telp	varchar	20	
6	Foto	Foto	varchar	40	
7	Jenis Kelamin	jenis_kelamin	varchar	20	
8	Tempat Lahir	tempat_lahir	varchar	40	
9	Tanggal Lahir	tanggal_lahir	date		
10	Status	Status	varchar	20	

6. Tabel Fingerprint

Nama tabel : fingerprint

Fungsi : Mengelola data fingerprint karyawan

Tipe : Data Master

Tabel V.14 Tabel Fingerprint

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Fingerprint	Fid	int	11	Primary Key
2	Data Fingerprint	Data	blob		Foreign Key

3	Nip	Nip	varchar	20	
---	-----	-----	---------	----	--

Sumber: Hasil Analisis Data (2017)

7. Tabel alpa

Nama tabel : alpa

Fungsi : Mengelola data alpa karyawan

Tipe : Data Master

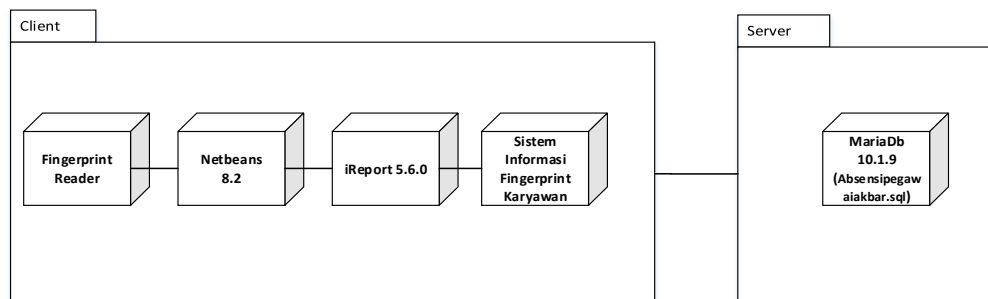
Tabel V.15 Tabel Alpa

No	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	No Alpa	No	Int	11	Primary Key
2	Nip	Nip	varchar	20	Foreign Key
3	Tanggal Alpa	tgl_alpa	date		

Sumber: Hasil Analisis Data (2017)

5.8 Deployment Diagram

Deployment diagram digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. *Deployment diagram* Sistem Absensi yang diusulkan dapat dilihat pada Gambar V.17.



Gambar V.17 *Deployment Diagram* Usulan

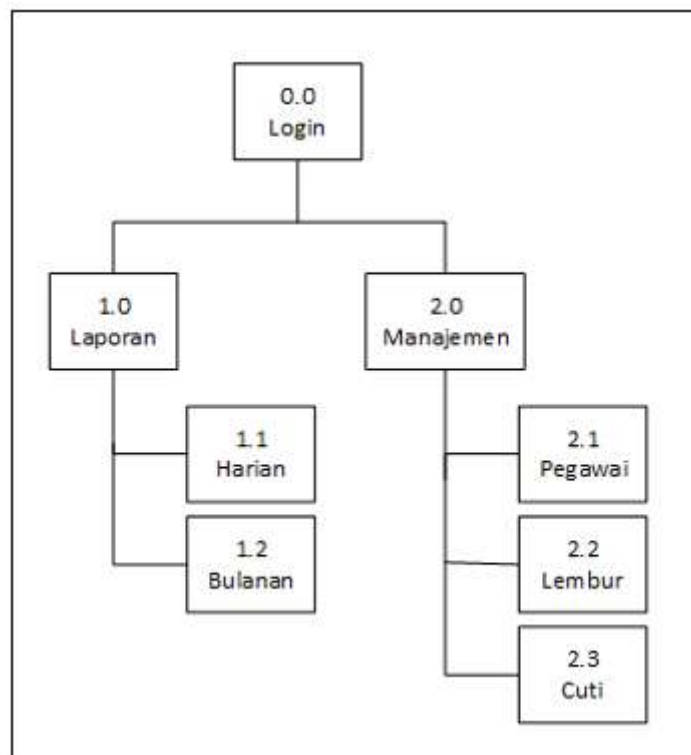
Sumber: Hasil Analisis Data (2017)

5.9 Perancangan Desain Program

Tahap ini merupakan tahap kedua dalam metodologi *prototype evolutioner*, yaitu tahap membuat sebuah prototipe dari program/aplikasi. Dimulai dengan analisis desain usulan meliputi pembuatan struktur menu program, *flowchart* program, dan *interface* program sampai dengan program dapat dijalankan.

5.9.1 HIPO (*Hierarchy plus Input-Process-Output*)

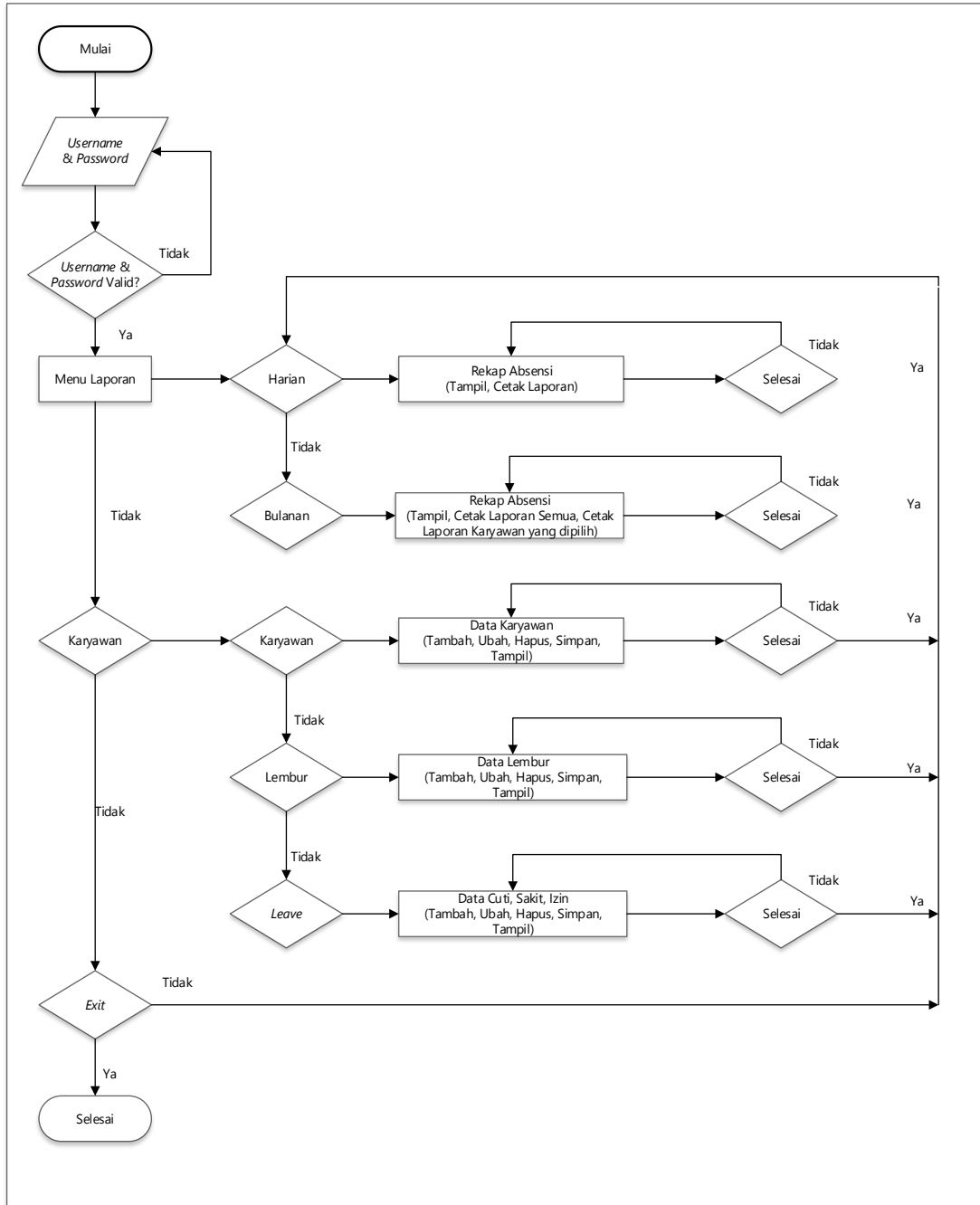
Hierarchy plus Input-Process-Output (HIPO) digunakan untuk mendokumentasikan sebuah struktur yang menggambarkan hubungan antar fungsi dalam program secara hierarkis. Diagram ini memuat semua modul yang ada dalam sistem beserta nama dan nomornya. Perancangan HIPO aplikasi usulan Sistem Absensi dan Penggajian dapat dilihat pada Gambar V.18 sebagai berikut:



Gambar V.18 HIPO Sistem Absensi Usulan
Sumber: Hasil Analisis Data (2017)

5.9.2 *Flowchart* Pengoperasian Program

Flowchart yang digunakan untuk mendokumentasikan aplikasi Sistem Absensi Karyawan ini menggunakan bagan alir logika program (*program logic flowchart*). Bagan alir ini digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika, bukan untuk menggambarkan instruksi-instruksi program komputer secara terinci. Gambar V.19 menunjukkan *program logic flowchart* aplikasi Sistem Absensi Karyawan:



Gambar V.19 *Flowchart* Pengoperasian Program Usulan

Sumber: Hasil Analisis Data (2017)

5.9.3 Perancangan *Interface* Program

Rancangan *interface* dari program Sistem Absensi Karyawan ini adalah sebagai berikut:

1. *Form Login Admin*

Form login adalah *form* yang digunakan untuk masuk ke dalam program aplikasi. Untuk masuk ke dalam aplikasi, Admin harus memasukkan *username* dan *password* yang benar. Rancangan *form login* admin dapat dilihat pada Gambar V.20.

The diagram illustrates the Admin Login Form interface. It is enclosed in a large rectangular border. On the left side, there is a square box containing the text "LOGO" and "PERUSAHAAN" stacked vertically. To the right of this box, there are two stacked rectangular input fields. The top field is labeled "Username" and the bottom field is labeled "Password". Below these input fields, there are two rectangular buttons: "Login" on the left and "Exit" on the right.

Gambar V.20 *Interface Form Login Admin*
Sumber: Hasil Analisis Data (2017)

2. *Form Laporan Harian*

Form ini menampilkan laporan kehadiran periode harian karyawan, admin cukup mengisi periode bulan yang ingin ditampilkan lalu pilih tombol tampil dan pilih Cetak Laporan untuk mengetahui Karyawan yang hadir pada periode hari yang dipilih, seperti yang ditampilkan oleh Gambar V.21.

Laporan	Manajemen
Harian	Bulanan

		Tampil
----------------------	----------------------	--------

NIP	Nama	Jabatan	Jam Masuk	Jam Keluar

Gambar V.21 *Interface Form* Laporan Harian
 Sumber: Hasil Analisis Data (2017)

3. *Form* Laporan Bulanan

Form ini menampilkan laporan kehadiran periode bulanan karyawan, admin cukup mengisi periode bulan yang ingin ditampilkan lalu pilih tombol tampil. Di-*form* ini admin juga dapat melihat rekap kehadiran bulanan keseluruhan, cetak semua laporan kehadiran dan cetak laporan berdasarkan karyawan yang dipilih seperti yang ditampilkan oleh Gambar V.22.

Laporan

Manajemen

Harian

Bulanan

April

2018

Tampil

Tanggal	Nama	Jam Masuk	Jam Keluar

Cetak Laporan Semua

Cetak Laporan Pegawai yang dipilih

Gambar V.22 *Interface Form* Laporan Bulanan
Sumber: Hasil Analisis Data (2017)

4. Form Data Karyawan

Form ini menampilkan proses pengelolaan data karyawan mulai dari tambah, ubah, hapus dan tampil data karyawan seperti yang ditampilkan oleh Gambar V.23.

LaporanManajemen

KaryawanLemburLeave

NIP

Nama

Tanggal

Jenis Lembur

Lembur

Jam

Tambah

Ubah

Hapus

Simpan

Tampil

NIP	Nama	Jabatan	Jenis Lembur	Lembur (Jam)	Foto	id

V.24 *Interface Form Data Lembur*
 Sumber: Hasil Analisis Data (2017)

6. *Form Data Leave*

Form ini menampilkan proses pengelolaan data cuti, sakit dan izin karyawan mulai dari tambah, ubah, hapus dan tampil data karyawan seperti yang ditampilkan oleh Gambar V.25.

LaporanManajemen

KaryawanLemburLeave

NIP

Nama

Dari

Sampai

Jenis Leave

Keterangan

Tambah

Ubah

Hapus

Simpan

Tampil

Dari	Sampai	Nip	Nama	Jabatan	Jenis Leave	Keterangan	Foto	id

Gambar V.25 *Interface Form Data Leave*
 Sumber: Hasil Analisis Data (2017)

7. *Form Absensi Karyawan*

Form ini menampilkan proses absensi karyawan pada saat jam masuk kerja dan jam keluar kerja, karyawan cukup meletakkan jari yang sudah didaftarkan di mesin *fingerprint* dan menunggu hingga data karyawan muncul dilayar monitor seperti yang ditampilkan pada gambar V.26 dibawah ini.

The image shows a web form titled "PRESENSI PEGAWAI". At the top is a header bar with the title. Below the header, on the left, are two stacked input fields labeled "Tanggal" and "Jam". On the right is a box labeled "Logo Perusahaan". In the center-left is a large rectangular area with the text "Silahkan Letakkan Jari Anda" above it. To its right is another large empty rectangular box.

Gambar V.26 *Interface Form* Absensi Karyawan
Sumber: Hasil Analisis Data (2017)

5.10 Implementasi Sistem *Software* dan *Hardware*

Tahap ini adalah tahap pengkodean program menggunakan bahasa pemrograman Java dengan Netbeans sebagai aplikasi editor. Setiap *interface* berisikan kode program agar program dapat dijalankan sesuai dengan fungsinya. Untuk mendukung kebutuhan implementasi sistem diperlukan suatu spesifikasi perangkat lunak (*software*) maupun perangkat keras (*hardware*). Adapun spesifikasinya sebagai berikut:

1. Analisis Kebutuhan *Software*
 - a. Sistem Operasi : *Microsoft Windows 7*
 - b. *Database Server* : *MariaDB 10.1.9*
 - c. Bahasa Pemrograman : *Java Netbeans 8.2*

2. Analisis Kebutuhan *Hardware*

- a. *Processor* : Minimal *Processor Intel Pentium IV*
- b. *RAM* : Minimal RAM 512 MB
- c. *Harddisk* : Minimal *Free space* 8 GB
- d. *Finger Reader* : ZK4000, ZK6000, ZK7000, ZK8000,
ZK4500, ZK7500, ZK8500
- e. *Mouse, Keyboard, Monitor* sebagai peralatan antar muka.
- f. *Printer* sebagai media pencetakan dokumen.

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. Dengan menggunakan sistem informasi absensi karyawan ini, HRD dapat melakukan kontrol kehadiran dan rekapitulasi data kehadiran karyawan melalui sistem informasi absensi fingerprint karyawan tanpa memeriksa kartu absensi karena sistem sudah terkomputerisasi.
2. Dengan menggunakan sistem informasi absensi karyawan ini, HRD dapat melakukan pengarsipan data absen karyawan tanpa menggunakan kertas.
3. Dengan menggunakan sistem informasi absensi karyawan ini dapat menghindarkan kecurangan yang dilakukan oleh karyawan, yaitu berupa titip absen antar karyawan karena membutuhkan sidik jari pada proses absensi.

6.2. Saran

Saran yang diberikan untuk pengembangan Aplikasi Sistem Absensi *Fingerprint* Karyawan selanjutnya adalah sebagai berikut:

1. Untuk menghindari terjadinya manipulasi data oleh *user admin*, maka perusahaan harus menerapkan kebijakan yang tegas dan memberikan sanksi hukuman bagi tindakan-tindakan yang bertujuan untuk manipulasi data absensi oleh *user admin*.
2. Bagian personalia dapat bekerjasama dengan divisi IT dalam pengawasan serta divisi audit bila memang diperlukan untuk memastikan tidak adanya manipulasi data.

DAFTAR PUSTAKA

Sumber : Buku

- A.S Rosa dan Salahuddin M, 2011. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Modula, Bandung.
- Bangun, Wilson. 2012. “Manajemen Sumber Daya Manusia”. Jakarta: Erlangga.
- Connolly, Thomas and Begg, Carolyn. 2010. Database Systems: A Practical Approach to Design, Implementation, and Management, Fifth Edition. Pearson Education, Boston.
- Darmawan, Deni. 2007. *Teknologi Informasi dan Komunikasi*. Bandung: Arum Mandiri Press.
- Dennis, A. et al. 2013. Systems Analysis and Design with UML, 4th Edition. John Wiley and Sons. New York.
- Edy, Sutrisno. 2012. Manajemen Sumber Daya Manusia. Jakarta: Kencana
- Gata, Windu. 2013. Sukses Membangun Aplikasi Penjualan dengan JAVA, Jakarta, Elex Media Komputindo
- Jogiyanto. 2005. Analisis dan Desain Sistem Informasi. Yogyakarta: Penerbit Andi.
- Kristanto, Andri. 2008. Perancangan Sistem Informasi dan Aplikasinya, Gava Media, Yogyakarta
- Raymond, McLeod. 2001. Sistem Informasi Edisi 7 Jilid 2. Prenhallindo. Jakarta.
- Mulyadi, 2001, Sistem Akuntansi, Edisi Ketiga, Cetakan Ketiga, Penerbit Salemba Empat, Jakarta.
- Pressman, Roger. 2005. Software Engineering: A Practitioner's Approach. New York : McGraw-Hill.
- Samsudin, sadili. (2006), Manajemen Sumber Daya Manusia, cetakan ke-1 Bandung : Pustaka Setia
- Sedarmayanti. 2012. Good Governance “Kepemerintahan yang Baik. Bagian Kedua Edisi Revisi. CV Mandar Maju. Bandung
- Soedaryono.2000.Tata Laksana Kantor.Jakarta: Bumi Aksara.

Sumber : Jurnal

Trisnanto, Dedy dan Juniar, Ahmad, 2014. Implementasi Penjadwalan Kuliah dengan PHP dan MySQL berbasis Heuristic Search yang dikombinasikan dengan Teknik Smart Back TR.