



E-Mail Remainder
Untuk Informasi *Update* Data Absen PT Higinia
Alhadin

Laporan Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh :

Bagus Susetyo
41505110028

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2009

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini,

N a m a : Bagus Susetyo
N.I.M : 41505110028
Judul Skripsi : E-MAIL REMAINDER UNTUK KONFIRMASI
 UPDATE DATA ABSEN PT.HIGINA ALHADIN

Dengan ini menyatakan bahwa hasil penulisan Skripsi yang telah saya buat ini merupakan hasil karya sendiri dan benar keasliannya. Apabila ternyata di kemudian hari penulisan Skripsi ini merupakan hasil plagiat atau penjiplakan terhadap karya orang lain, maka saya bersedia mempertanggungjawabkan sekaligus bersedia menerima sanksi berdasarkan aturan tata tertib di Universitas Mercu Buana.

Demikian, pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Jakarta, 14 Januari 2009

[Bagus Susetyo]

LEMBAR PERSETUJUAN

NIM : 41505110028
Nama : Bagus Susetyo
Judul Skripsi : E-MAIL REMAINDER UNTUK
KONFIRMASI UPDATE DATA ABSEN PT
HIGINA ALHADIN

SKRIPSI INI TELAH DIPERIKSA DAN DISETUJUI
JAKARTA, 14 JANUARI 2009

(Ahmad Kodar, Drs., MT)

Pembimbing

Devi Fitriannah, S.Kom., MTI

Koor. Tugas Akhir Teknik Informatika

Abdusy Syarif, ST., MT

Kaprodi Teknik Informatika

KATA PENGANTAR

Puji dan syukur dipanjatkan ke hadirat Tuhan Yang maha Esa karena dengan rahmat-Nya laporan tugas akhir yang berjudul ***E-mail Remainder Untuk Konfirmasi Update Data Absen PT Higina Alhadin*** dapat terselesaikan. Penulisan laporan ini merupakan salah satu tugas akhir yang wajib dilaksanakan sebagai persyaratan untuk menyelesaikan jenjang pendidikan pada program sarjana (S 1) Universitas PKSM Mercu buana.

Ada banyak pihak yang telah membantu dalam semua proses kajian sampai laporan ini tersusun dan atas bantuan tersebut ucapan terima kasih patut disampaikan. Ucapan terima kasih secara khusus disampaikan kepada:

1. Bapak Achmad Khodar, selaku dosen pembimbing akademik dan teknik tugas akhir di PKSM Mercubuana yang telah menyediakan banyak waktu, masukan dan saran selama melakukan tugas akhir,
2. Bapak Toni Stiawan selaku pembimbing lapangan di PT Higina Alhadin yang telah memberikan pengarahan selama melakukan tugas akhir,
3. Team IT PT Higina Alhadin yang telah meluangkan waktu dan tempat untuk melakukan riset ini,
4. koordinator praktik kerja dan tugas akhir,
5. seluruh staf pengajar PKSM Mercubuana,
6. seluruh karyawan PT Higina Alhadin,

7. Orang tua tercinta yang selalu memberikan dukungan moral dan materi,
8. Evelline Hana Andrya atas dukungan, support, bantuan yang diberikan
9. seluruh rekan-rekan mahasiswa PKSM Mercubuana program studi Teknik Informatika angkatan VII dan semua pihak yang telah memberikan bantuan dalam penyusunan laporan tugas akhir ini yang tidak dapat disebutkan satu persatu.

Akhir kata, semoga laporan tugas akhir ini dapat bermanfaat bagi pembaca dan dapat memberikan sumbangan positif bagi kemajuan institusi PKSM Mercuru Buana.

Jakarta, Januari 2009

Penyusun

ABSTRACTION

This reminder e-mail for attendance data update confirmation is built to help the user, in this case are the employees of PT. Higina Alhadin, to use the Human Resources Information System application.

The module is made for help the operational activity of Human Resources Department to check the employee's attendance every month. With this reminder *e-mail*, employees do not need to login into the Human Resources Information System application. We can imagine if all the employees login into the application in the same time, it would run very slowly. We hope with this reminder e-mail, the work of the server can be reduced and the process for the employees attendance data can run faster.

Keywords: E-mail, Attendance, PHP, MySQL.

xiv+99 pages; 10 table; 1 attachment; Bibliography: 7 (1992-2002)

ABSTRAKSI

E-mail remainder untuk konfirmasi *update* data absen ini dibuat untuk membantu *user* dalam hal ini karyawan dalam menggunakan aplikasi *human resources information system* di PT Higinia Alhadin.

Modul yang dikembangkan adalah untuk membantu kegiatan operasional bagian *human resources department* dalam melakukan pengecekan absent karyawan setiap bulan. Dikatakan membantu karena dengan adanya *e-mail remainder* ini karyawan tidak perlu lagi melakukan login lagi ke aplikasi *human resources information system* jika absen mereka tidak ada yang bermasalah. Bisa dibayangkan jika semua karyawan dalam waktu yang bersamaan melakukan *login* ke aplikasi *human resources information system*, tentunya akan sangat lambat sekali. Dengan demikian diharapkan adanya *e-mail remainder* ini maka kinerja server dapat diringankan sehingga waktu untuk memproses data absen karyawan dapat lebih cepat.

Kata Kunci : *E-mail*, Absensi, PHP, MySQL.

xiv+99 halaman; 10 tabel; 1 lampiran; Daftar Pustaka: 7 (1992-2002)

DAFTAR ISI

LEMBAR PERNYATAAN	ii
LEMBAR PERSETUJUAN	iii
KATA PENGANTAR.....	iv
ABSTRACTION	vi
ABSTRAKSI.....	vii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN.....	1
1.1. LATAR BELAKANG MASALAH.....	1
1.2. TUJUAN PENULISAN	3
1.3. BATASAN DAN RUANG LINGKUP MASALAH.....	4
1.4. METODE PENELITIAN	6
1.5. SISTEMATIKA PENULISAN	7
BAB II LANDASAN TEORI	9
2.1. TINJAUAN PUSTAKA.....	9
2.2. DASAR TEORI.....	10
2.2.1 Rekayasa Perangkat Lunak	10
2.2.2 Sistem Basis Data.....	13
2.2.2.1 Mengenal DBMS	14
2.2.2.2 Bahasa Basis Data.....	15

2.2.2.2.1. Data Definition Language (DDL).....	15
2.2.2.2.2. Data Manipulation Language (DML)	15
2.2.2.3 Komponen Dasar dalam Basis Data.....	15
2.2.3 Surat Elektronik	20
2.2.3.1 <i>Mail Server</i>	21
2.2.3.1.1 Protokol SMTP	21
2.2.4 Flowchart	22
2.2.5 PHP	26
2.2.5.1 Script PHP	27
2.2.5.2 Kelebihan PHP	28
2.2.6 MySql.....	31
2.2.7 Diagram Arus Data	33
BAB III ANALISA DAN PERANCANGAN SISTEM.....	36
3.1. DESKRIPSI UMUM.....	36
3.1.1 Visi dan Misi	38
3.1.2 Struktur Organisasi	39
3.2. ANALISA SISTEM ABSENSI.....	43
3.2.1 Bahan Penelitian.....	44
3.2.2 Alat Penelitian.....	45
3.2.3 Metode Pengumpulan Data.....	46
3.3. ANALISA PERMASALAHAN.....	46
3.4. DESIGN SISTEM	48
3.5. SPESIFIKASI KEBUTUHAN.....	49
3.5.1 Input	49

3.5.2	Output.....	50
3.6.	DESKRIPSI FUNGSIONAL	50
3.6.1	<i>Context Diagram</i>	50
3.6.1.1	DFD Level 1.....	51
3.6.1.2	DFD Level 2.....	53
3.6.1.2.1.	DFD LEVEL 2 Proses Pendataan.....	53
3.6.1.2.2.	DFD Level 2 Proses Absen.....	54
3.6.2	<i>E-R Diagram</i>	58
3.6.3	Physical Data Model	59
3.7.	Perancangan.....	60
3.7.1	Perancangan <i>Database</i>	60
3.7.2	Perancangan Antarmuka	64
BAB IV	IMPLEMENTASI DAN PENGUJIAN	65
4.1.	LINGKUNGAN YANG DIBUTUHKAN.....	65
4.1.1	<i>Hardware</i>	65
4.1.1.1	<i>Server</i>	65
4.1.1.2	<i>Client</i>	65
4.1.2	<i>Software</i>	66
4.1.2.1	<i>Server</i>	66
4.1.2.2	<i>Client</i>	66
4.2.	PROSES	67
4.2.1	Pengambilan Data Absen dari mesin absen	67
4.2.2	Pengiriman <i>E-mail Remainder</i>	69

BAB V KESIMPULAN DAN SARAN	71
5.1. Kesimpulan.....	71
5.2. Saran.....	71
DAFTAR PUSTAKA	73
LAMPIRAN A.....	74
LISTING PROGRAM.....	74

DAFTAR GAMBAR

Gambar 2.2.1 Relasi <i>One to One</i>	17
Gambar 2.2.2 Relasi <i>One to Many</i>	18
Gambar 2.2.3 Relasi <i>Many to Many</i>	18
Gambar 2.2.4 <i>Entity</i>	19
Gambar 2.2.5 <i>Relationship</i>	19
Gambar 2.2.6 Relasi <i>Mandatory 1 cardinality</i>	19
Gambar 2.2.7 Relasi <i>Mandatory Many Cardinality</i>	19
Gambar 2.2.8 Relasi <i>Optional Zero-Many Cardinality</i>	20
Gambar 2.2.9 Simbol Proses	34
Gambar 2.2.10 Simbol Arus Data.....	34
Gambar 2.2.11 Simbol Simpanan Data.....	34
Gambar 2.2.12 Simbol <i>External Entity</i>	35
Gambar 3.1.1 Struktur Organisasi PT Higina Alhadin	39
Gambar 3.2.1 Alur proses absensi karyawan PT Higina Alhadin	43
Gambar 3.6.1 Context Diagram <i>Untuk Human Resources Information System</i>	51
Gambar 3.6.2 DFD Level 1 <i>Untuk Human Resources Information System</i>	53
Gambar 3.6.3 <i>DFD Level 2 Untuk Human Resources Information System</i>	56
Gambar 3.6.4 DFD Level 2 <i>Human Resources Information System</i>	57
Gambar 3.6.5 Entity Relational Diagram <i>Untuk Sistem Email Remainder Untuk Konfirmasi Update Data Absen</i>	58
Gambar 3.6.6 Physical Data Model <i>Untuk Sistem Email Remainder Untuk Konfirmasi Update Data Absen</i>	59

Gambar 3.7.1 Perancangan File yang akan di Kirim Melalui E-mail Karyawan 64

Gambar 4.2.1 Isi File *.sdf 68

Gambar 4.2.2 Detail Isi File *.sdf 68

Gambar 4.2.3 *E-mail remainder* yang dikirim ke masing-masing karyawan setiap bulannya..... 70

DAFTAR TABEL

Tabel 2.2-1 <i>Flowchart</i>	23
Tabel 3.3-1 Tabel Data Jumlah Bandwidth Tanggal Jatuh Tempo Penghitungan Absensi	46
Tabel 3.4-1 Alur Proses <i>E-mail Remainder</i> Untuk <i>Update</i> Data Absen P.T. Higinia Alhadin	48
Tabel 3.7-1 Rancangan tabel company	60
Tabel 3.7-2 Rancangan tabel division	60
Tabel 3.7-3 Rancangan tabel user	61
Tabel 3.7-4 Rancangan tabel absensi	62
Tabel 3.7-5 Rancangan tabel mutation	62
Tabel 3.7-6 Rancangan tabel tanggal_merah	63
Tabel 3.7-7 Rancangan tabel <i>work_timezone</i>	63

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG MASALAH

Perkembangan teknologi terutama teknologi informasi sangatlah membantu kita. Dari sekian banyak, salah satu yang banyak digunakan adalah sistem absensi dengan menggunakan sidik jari (*finger print*). Penggunaan absensi dengan menggunakan sidik jari ini banyak digunakan oleh perusahaan karena tingkat keamanan dari mesin *scan* sidik jari sangatlah sensitif. Kelamahan yang sering dimanfaatkan oleh karyawan, mereka biasanya bisa menitipkan absen mereka ke orang lain. Dengan sensitifitas yang tinggi ini sangat tidak mungkin apabila satu seorang karyawan menitipkan absen mereka ke karyawan lain.

Jika dijabarkan secara singkat, cara kerja mesin absensi sidik jari ini biasanya terhubung dengan sebuah komputer. Mesin akan merekam jam kedatangan dan kepulangan dari masing-masing karyawan dan disimpan sementara didalam memori yang terdapat di mesin tersebut. Kemudian secara berkala komputer yang terhubung ke mesin tersebut akan mengambil data yang tersimpan di memori sementara tadi untuk selanjutnya disimpan ke dalam bentuk database yang berformat *.sdf.

Database ini kemudian akan diolah dalam sebuah aplikasi berbasis web. Masing-masing karyawan dapat melakukan login untuk melihat detail absen mereka selama satu bulan. Tujuan pengecekan ini adalah melihat apakah ada absen mereka yang kosong, atau tidak tercatat. Jika tidak tercatat resikonya adalah tunjangan makan dan transport dari karyawan akan berkurang.

Kemudian meskipun sudah tersedianya fasilitas web untuk melakukan pengecekan, tetapi masih saja timbul masalah-masalah lain misalnya :

1. Tidak semua karyawan secara rutin melakukan pengecekan, sehingga pada saat pembagian *report* absensi bulanan masih ada absensi mereka yang kosong atau tanpa keterangan.
2. Jika semua karyawan pada tanggal jatuh tempo pengecekan, mereka meng-akses database secara bersamaan maka akan memperlambat kinerja dari *server*.
3. Karena proses memakan waktu yang cukup lama maka proses perhitungan yang dilakukan *HRD (Human Resource Department)* akan terhambat pula. Dengan begitu pendistribusian uang makan transport menjadi terhambat pula.
4. Pihak *HRD (Human Resource Department)* pada tanggal jatuh tempo harus melakukan pemberitahuan kepada masing-masing penanggung

jawab divisi untuk memberitahukan kepada masing-masing karyawannya agar melakukan pengecekan.

5. Tidak jarang terjadi pula komplain dari karyawan pada saat *report* sudah selesai, sehingga harus mengoreksi dan membuat *report* tersebut.

Karena masalah-masalah yang muncul tersebut, maka perlu dibuat sebuah *warning* atau tanda pengingat yang berfungsi memberikan informasi peringatan kepada karyawan agar mereka melakukan pengecekan. Tanda pengingat ini berupa *e-mail (surat elektronik) remainder* yang isinya mencantumkan detail absen karyawan selama satu bulan. Sehingga jika ada absen yang kosong dapat segera diperbaiki.

1.2. TUJUAN PENULISAN

Maksud dan tujuan yang ingin dicapai dalam pembuatan *e-mail remainder* ini dibagi menjadi dua kelompok yaitu :

1. Bagi Mahasiswa
 - a. Untuk memenuhi persyaratan dalam rangka menyelesaikan program studi Teknik Informatika Universitas Mercu Buana.
 - b. Mengembangkan diri dan membuka wawasan pengetahuan baru sesuai dengan bidang yang saat ini ditekuni, yaitu bagaimana membangun sebuah sistem yang reliabilitas.

2. Bagi Perusahaan

- a. Mempersingkat waktu pengecekan yang harus dilakukan *human resources department*. Karena biasanya untuk melakukan pengecekan ini akan memakan waktu tiga hari, diharapkan dengan adanya pengiriman *e-mail remainder* ini waktu penghitungan menjadi tepat setiap tanggal lima belas perbulannya.
- b. Secara teknis *e-mail remainder* ini akan mengirimkan *e-mail* otomatis yang berisi informasi data absen karyawan selama satu bulan setiap tanggal tiga belas. Dengan demikian karyawan akan memperoleh informasi lebih awal. Memberikan kemudahan bagi masing-masing karyawan dalam mengelola dan melakukan cek silang data absen mereka.
- c. Berdasarkan data, *traffic* peng-aksesan aplikasi *human resources information system* setiap tanggal empat belas dan lima belas setiap bulannya selalu meningkat. Hal ini dapat dilihat dari badwidth yang rata-rata dihabiskan selama dua hari tersebut maximal sebesar 672.77 MB. Diharapkan dengan adanya *e-mail remainder* ini dapat membantu meringankan kinerja server.

1.3. BATASAN DAN RUANG LINGKUP MASALAH

Pada tugas akhir ini, batasan dan ruang lingkup masalahnya adalah sebagai berikut :

Contoh kasus yang diambil adalah PT Higina Alhadin, salah satu perusahaan yang menggunakan absen sidik jari untuk melakukan proses perekaman jam kedatangan dan kepulangan masing-masing karyawan. Untuk saat ini pihak perusahaan sudah memiliki aplikasi berbasis web yang dapat digunakan untuk melihat data absen.

Tetapi penulis tidak akan membahas secara men-detail aplikasi web yang sudah tersedia. Dalam hal ini hanya akan dibahas proses pembuatan *e-mail remainder* ini.

Untuk membangun sistem *e-mail remainder* ini penulis menggunakan bahasa pemrograman *PHP: Hypertext Preprocessor* versi 5 dengan database *MySql* dan sebagai *reportnya* adalah pengiriman *e-mail* secara otomatis ke setiap karyawan. *E-mail* tersebut menyisipkan data absen selama tenggang waktu satu periode (satu bulan) terhitung dari tanggal enam belas sampai dengan lima belas bulan berikutnya.

Proses pengiriman *e-mail remainder* ini dibagi menjadi tiga proses, pertama pengambilan database dari mesin absensi kemudian disimpan didalam database, kedua dengan menggunakan *script PHP* diurutkan berdasarkan nik (nomor induk karyawan) jam kedatangan dan jam kepulangan untuk masing-masing karyawan. Proses ketiga adalah dengan menggunakan *script PHP* pula secara otomatis *ter-scheduler* mengirimkan ke alamat *e-mail* masing-masing karyawan.

1.4. METODE PENELITIAN

Metode penelitian yang digunakan pada tugas akhir ini adalah dengan menggunakan metode gabungan yakni gabungan antara metode studi pustaka dengan studi lapangan. Proses studi pustaka dilakukan guna mendukung proses studi lapangan.

Tahap-tahap yang dilakukan dalam proses ini melalui beberapa tahap :

1. Analisis

Proses pengumpulan kebutuhan, diintensifkan dan difokuskan khususnya perangkat lunak untuk memahami sifat program yang dibangun. Perencanaan perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja dan antar muka (*interface*) yang diperlukan. Kebutuhan, baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi oleh pengguna.

2. Desain

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; detail (algoritma) prosedural, struktur data, arsitektur perangkat lunak, representasi interface.

3. Generasi Kode

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

4. Pengujian

Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi memberikan hasil aktual yang sesuai dengan hasil yang di butuhkan.

5. Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah di selesaikan kepada pengguna (perkecualian yang mungkin adalah perangkat lunak yang sudah dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak disesuaikan untuk mengakomodasi perubahan-perubahan didalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pengguna membutuhkan perkembangan fungsional untuk kerja.

1.5. SISTEMATIKA PENULISAN

Proyek tugas akhir ini dituangkan ke dalam lima buah bab, yang masing-masing akan dijabarkan sebagai berikut :

BAB I

Menjelaskan mengenai latar belakang, rumusan masalah yang dianalisis, batasan dari masalah yang timbul, tujuan penulisan, pemecahan masalah, metode penelitian dan sistematika penulisan

BAB II

Berisi teori – teori yang menjadi referensi untuk mendukung tugas akhir ini.

BAB III

Berisi tentang deskripsi umum perusahaan, teknologi yang digunakan (*Software* dan *Hardware*) dan hasil analisis terhadap perangkat lunak yang dibuat sehingga bisa ditentukan kebutuhan apa saja yang harus dipenuhi serta berisi tentang Perancangan dari sistem yang dibuat meliputi perancangan aplikasi, perancangan basis data serta perancangan user *interface*.

BAB IV

Berisi tentang deskripsi implementasi dari analisis dan *design* serta pengujian terhadap perangkat lunak.

BAB V

Berisi kesimpulan dan saran saran dari tugas akhir ini.

BAB II

LANDASAN TEORI

2.1. TINJAUAN PUSTAKA

Proses absensi akan dilakukan melalui pengontrolan data pegawai yang keluar ataupun masuk. Jika pegawai tidak masuk melebihi target yang ditentukan, maka karyawan akan menerima konsekuensi yaitu gaji di potong. Dari sistem absensi juga akan dibuat penilaian kinerja dari para pegawai. Absensi pegawai sangat dipengaruhi oleh faktor-faktor berikut :

- Jam kerja dilakukan selama 8 jam. Jika melebihi, maka harus dikategorikan lembur.
- Pada hari sabtu jam kerja dilakukan sampai jam 12 siang. Lebih dari itu dikatakan lembur.
- Ada waktu istirahat.
- Ada waktu untuk beribadah.
- Pada hari libur nasional semua perusahaan diharapkan mengikuti.
- Jika ada kegiatan lain maka dikatakan lembur.
- Untuk semua pegawai mempunyai jatah juti 12 hari selama 1 tahun.
- Untuk pegawai yang akan dan telah melahirkan dapat diberikan cuti.

Selain perihal diatas, perusahaan berhak membuat persepakatan kerja bersama sesuai dengan hasil musyawarah bersama dengan pegawai dan disahkan oleh pemerintah (Ditjen Pajak, 2006).

2.2. DASAR TEORI

2.2.1 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak menurut Pressman (2005) adalah sebuah disiplin yang menggabungkan proses, metode-metode dan alat bantu untuk mengembangkan perangkat lunak komputer. Perangkat lunak saat ini telah menjadi kekuatan yang menentukan. Perangkat lunak menjadi mesin yang mengendalikan pengambilan keputusan di dalam dunia bisnis, berfungsi sebagai dasar dari semua bentuk pelayanan serta penelitian keilmuan modern. Perangkat lunak dilekatkan dengan segala bentuk sistem, anantara lain: transportasi, medis, telekomunikasi, militer, proses industri, hiburan, perkantoran, dan lain-lain. Program-program perangkat lunak sudah tersebar luas, dan masyarakat memandangnya sebagai kemajuan teknologi dalam kehidupan.

Tiga fase dalam rekayasa perangkat lunak tanpa mempedulikan area aplikasi, ukuran proyek, atau kompleksitasnya (Pressman, 2002), yaitu:

1. Fase definisi (*Definition phase*): pada definisi ini pengembang perangkat lunak harus mengidentifikasi informasi apa yang akan diproses, fungsi

dan unjuk kerja apa yang akan dibangun, batasan desain apa yang akan ada, dan kriteria validasi apa yang akan dibentuk untuk mendefinisikan sistem yang sukses. Kebutuhan (*requirement*) kunci dari sistem perangkat lunak yang didefinisikan. Ada tiga tugas utama dalam fase ini: sistem atau rekayasa informasi, perancangan proyek perangkat lunak, serta analisa kebutuhan.

2. Fase pengembangan (*development phase*) selama pengembangan perangkat lunak, harus didefinisikan bagaimana data dikonstruksikan, bagaimana fungsi-fungsi diimplementasikan sebagai arsitektur perangkat lunak, bagaimana desain prosedur akan diimplementasikan, bagaimana *interface* ditandai, bagaimana rancangan akan diterjemahkan kedalam bahasa pemrograman serta bagaimana pengujian akan dilakukan. Ada tiga tugas utama di dalam fase ini: rancangan perangkat lunak, pemunculan kode, dan pengujian perangkat lunak.
3. Fase pemeliharaan (*maintenance phase*) berfokus pada perubahan koreksi kesalahan, penyesuaian yang dibutuhkan ketika lingkungan perangkat lunak berkembang, serta perubahan dengan perkembangan yang disebabkan oleh perubahan pelanggan. Perkembangan perangkat lunak yang sistematis dan sekuensial melingkupi aktivitas-aktivitas sebagai berikut:

a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Perakayasa perangkat lunak (analisis) harus memahami domain informasi, tingkah laku, unjuk kerja, dan antarmuka (*interface*) yang diperlukan untuk memahami sifat program yang dibangun.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menerjemahkan syarat/kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

c. Pengkodean

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Desain dilakukan dengan cara yang lengkap, maka pembuatan kode dapat diselesaikan secara mekanis.

d. Pengujian/*Testing*

Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

2.2.2 Sistem Basis Data

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau melaksanakan sasaran tertentu (Jogiyanto, H.M., 1999). Dari definisi tersebut dapat disimpulkan bahwa sistem adalah kumpulan elemen-elemen yang saling berhubungan dan bertanggung jawab memproses masukan (*input*), sehingga menghasilkan keluaran (*output*)

Basis data atau *database* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya yang tersimpan diperangkat keras komputer dan diperlukan perangkat lunak tertentu yang bisa memanipulasi basis data tersebut (Fathansyah, 1999), seperti Microsoft Access, MySQL, ORACLE dan sebagainya. Pokok dari perancangan basis data adalah bagaimana merancang struktur logikal dan fisikal dari satu atau lebih basis data untuk memenuhi kebutuhan informasi yang diperlukan oleh pemakai sesuai dengan aplikasi-aplikasi yang telah ditentukan. Ada beberapa

tujuan perancangan basis data yang perlu diperhatikan (Martina, I., 2001), yaitu:

1. Memenuhi kebutuhan informasi sesuai dengan yang diperlukan oleh pemakai untuk aplikasi tertentu.
2. Mempermudah pemahaman terhadap struktur informasi yang tersedia dalam basis data.
3. Memberikan keterangan tentang persyaratan pemrosesan dan kemampuan sistem, seperti lama pengaksesan data, kapasitas memori yang harus ada, dan sebagainya.

2.2.2.1. Mengenal DBMS

DBMS merupakan *software* yang tugas utamanya adalah membantu dalam mewujudkan sistem basis data yang sempurna. Pekerjaan ini bukanlah suatu pekerjaan yang mudah. Perancang harus dapat mengatur basis data tersebut sehingga dapat tersimpan di dalam media penyimpanan dengan baik tanpa menimbulkan kekacauan, dan dapat dipakai oleh banyak pemakai sesuai dengan kepentingan masing-masing. Jangan sampai ada yang ditonjolkan dan ada yang tidak dihiraukan dari masing-masing kepentingan pemakai tersebut.

Nama-nama *software* seperti dBase, Foxpro, dan Clipper merupakan nama yang sudah cukup dikenal dalam dunia *software*. Semua ini merupakan contoh dari PC DBMS, yaitu DBMS yang digunakan pada

komputer jenis PC. Dalam skala yang lebih besar dikenal dengan DBMS yang bernama Sybase, DB2, Informix, Ingres, Progress, Interbase, Oracle, SQL Server dan masih banyak lagi.

2.2.2.2. Bahasa Basis Data

Supaya pengguna dapat mengoperasikan DBMS, maka perlu menggunakan bahasa yang dapat dimengerti. Terdapat 2 bahasa basis data, yaitu (Fathansyah, 1999):

2.2.2.2.1. Data Definition Language (DDL)

Bahasa ini berguna dalam pembuatan desain basis data, misalnya, pembuatan tabel, pembuatan *index*, pengubahan tabel, penyimpanan tabel dan sebagainya. Apabila *Data Definition Language* (DDL) dikompilasi akan berupa kumpulan tabel yang disimpan dalam *file* khusus yang disebut Kamus Data.

2.2.2.2.2. Data Manipulation Language (DML)

Bahasa ini digunakan untuk melakukan manipulasi pada data. Operasi yang dilakukan meliputi: penyimpanan data baru, penghapusan data, pengubahan data.

2.2.2.3. Komponen Dasar dalam Basis Data

Entity Relationship Diagram mempunyai beberapa komponen. Berikut merupakan komponen dasar ERD:

a. Entity

Entity menyatakan objek atau kejadian. PELANGGAN, PEGAWAI, DEPARTMENT merupakan contoh *entity*. Pada model relasional, *entity* akan menjadi tabel.

b. Attribute

Attribute adalah *item* data yang menjadi bagian dari suatu *entity*. Istilah lain *attribute* adalah properti. Nama pegawai, NIP adalah contoh *attribute* yang terdapat pada *entity* PEGAWAI. (Kadir, Abdul,1999:46).

c. Identifier

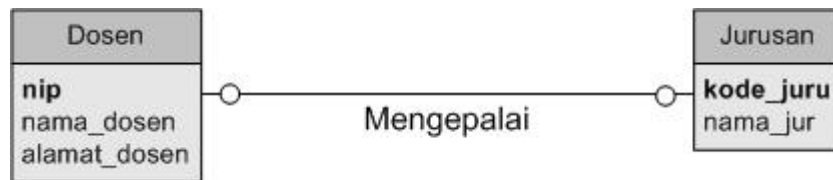
Identifier merupakan kunci kandidat (*candidate key*) yang dipilih sebagai karakteristik unik dari tipe *entity* atau identifikasi dari sebuah *entity* yang sifatnya unik. Ada 2 macam *identifier* yaitu *primary key* dan *foreign key*. *Primary key* merupakan atribut *identifier* yang memiliki karakteristik unik yang membedakan satu *record* dengan *record* lainnya dalam sebuah *entity*. Sedangkan *foreign key* adalah atribut dalam suatu *entity* atau tabel yang merupakan *primary key* di *entity* atau tabel asalnya.

d. Relationship

Untuk menghubungkan antara *entity* yang satu dengan *entity* yang lain diperlukan penghubung, dan macam hubungan yang terjadi ada tiga macam yaitu:

1. One to One

Hubungan dimana satu anggota *entity* mempunyai hubungan dengan satu anggota *entity* lain yang berbeda. Hubungan ini dibedakan menjadi dua yaitu *obligatory*, dimana bila anggota dari satu *entity* harus berpartisipasi atau mempunyai hubungan dengan *entity* yang lain, dan *non obligatory*, dimana bila semua anggota *entity* tidak harus mempunyai hubungan dengan anggota *entity* yang lain. (Fathansyah, 1999) Relasi *one to one* dapat dilihat pada Gambar 2.2.1



Gambar 2.2.1 Relasi One to One

2. One to Many

Hubungan dimana satu anggota *entity* mempunyai hubungan dengan beberapa anggota *entity* lain yang berbeda. (Fathansyah, 1999:71) Relasi *one to many* dapat dilihat pada Gambar 2.2.2



Gambar 2.2.2 Relasi *One to Many*

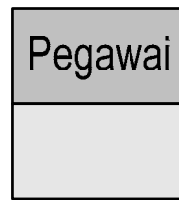
3. *Many to Many*

Hubungan dimana beberapa *entity* mempunyai hubungan dengan beberapa anggota *entity* yang lain pada *entity* yang berbeda. Jadi kedua belah pihak dapat memiliki hubungan dengan lebih dari satu anggota *entity* yang lain. Hubungan ini juga dibedakan menjadi hubungan *obligatory* dan *non obligatory*. (Fathansyah, 1999:72) Relasi *many to many* dapat dilihat pada Gambar 2.2.3

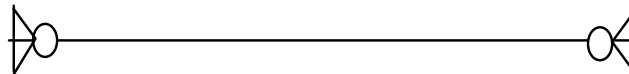


Gambar 2.2.3 Relasi *Many to Many*

Gambar 2.2.4 dan 2.2.5 di bawah ini adalah simbol yang digunakan dalam *ER-modeling*



Gambar 2.2.4 Entity



Gambar 2.2.5 Relationship

Relationship Cardinality

1. Mandatory 1 cardinality

Harus ada satu dan hanya satu. Relasi ini dapat dilihat pada Gambar 2.2.5



Gambar 2.2.6 Relasi Mandatory 1 cardinality

2. Mandatory many (M) cardinality

Harus ada satu atau banyak. Relasi ini dapat dilihat pada Gambar 2.2.7



Gambar 2.2.7 Relasi Mandatory Many Cardinality

3. *Optional zero-many cardinality*

Boleh ada satu atau lebih atau tidak sama sekali. Relasi ini dapat dilihat pada Gambar 2.2.8



Gambar 2.2.8 Relasi *Optional Zero-Many Cardinality*

2.2.3 Surat Elektronik

Surat elektronik (disingkat ratel atau surel atau surat-e) atau pos elektronik (disingkat pos-el) atau nama umumnya dalam bahasa Inggris "*e-mail* atau *email*" (ejaan Indonesia: imel) adalah sarana kirim mengirim surat melalui jalur internet.

Dengan surat biasa umumnya pengirim perlu membayar per pengiriman (dengan membeli perangko), tetapi surat elektronik umumnya biaya yang dikeluarkan adalah biaya untuk membayar sambungan internet. Tapi ada perkecualian misalnya surat elektronik ke telepon genggam, kadang pembayarannya ditagih per pengiriman.

www.id.wikipedia.org

2.2.3.1. Mail Server

Mail server adalah *daemon* yang bekerja menampung dan mendistribusikan *email* dalam jaringan. Protokol yang umum digunakannya antara lain adalah protokol *SMTP*, *POP3* dan *IMAP*. *SMTP* (*Simple Mail Transfer Protocol*) digunakan sebagai standar untuk menampung dan mendistribusikan *email*, sedangkan *POP3* (*Post Office Protocol v3*) dan *IMAP* (*Internet Mail Application Protocol*) digunakan agar *user* dapat mengambil dan membaca *email* secara *remote*, yaitu tidak perlu *login* ke dalam sistem *shell* mesin *mail server*, cukup menghubungi *port* tertentu dengan *mail client* yang mengimplementasikan *protokol POP3* dan/atau *IMAP*.

2.2.3.1.1. Protokol SMTP

Protokol ini merupakan protokol yang sekarang banyak dipakai oleh *mail transfer agent* (*MTA*) di internet. Mekanisme protokol ini adalah meneruskan secara langsung dari *server email* pengirim (*sender*) ke *server email* penerima (*recipient*) dengan syarat kedua *server* tersebut terhubung dan menggunakan metode pengiriman (*transport*) yang sama. Proses pengiriman tersebut akan melalui sebuah *relay* atau lebih, jika kedua *server* tersebut tidak menggunakan metode *transport* yang sama. Protokol ini juga mendukung *email* yang




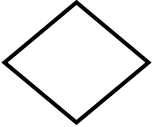
mempunyai format *MIME* (*Multipurpose Internet Mail Extension*), dimana *user* bisa mengirimkan *email* dengan menyertakan (*attachment*) sebuah file ataupun gambar. Untuk proses pengiriman *email* melalui protocol *SMTP*, Pertama sekali kita menulis *email* tersebut menggunakan *outlook*, *eudora*, *kmail*, *mutt*, *pine* atau *software* yang lain yang kemudian kita sebut sebagai *MUA* (*Mail User Agent*), kemudian *MUA* meneruskan ke *SMTP server* yang membuka koneksi pada *port 25* atau kita sebut *MTA* (*Mail Transfer Agent*). *MTA* kemudian mencari *server* tujuan dari alamat *email* tersebut dengan menggunakan hasil dari pencarian (*lookup*) database *DNS* (*Domain Name System*). Dari hasil tersebut *MTA* akan berkomunikasi dengan *MTA* tujuan menggunakan protokol *SMTP*. Dari komunikasi tersebut bisa ditentukan *email* tujuan untuk disimpan didalam *storage* pada *remote MTA* untuk kemudian diambil sewaktu-waktu oleh penerima dengan menggunakan protokol *POP3* (*Post Office Protocol*) ataupun *IMAP* (*Internet Message Access Protocol*).

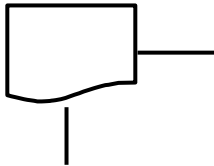
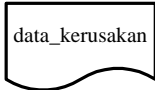
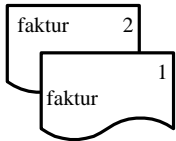
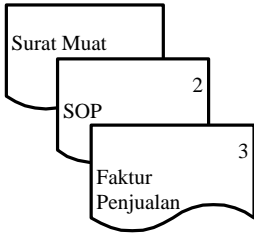
2.2.4 Flowchart

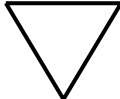
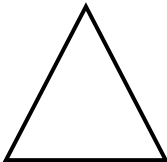

Flowchart secara grafis menyajikan sebuah proses atau sistem dengan menggunakan kotak dan garis yang saling berhubungan. Diagram




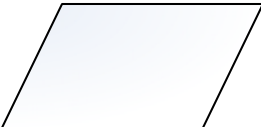
ini cukup sederhana tetapi merupakan alat yang sangat berguna untuk mencoba memahami sebuah proses atau menjelaskan sebuah proses. Pada tabel berikut dapat dilihat contoh simbol-simbol dasar untuk sistem.

Tabel 2.2-1 Flowchart

GAMBAR	KETERANGAN
	<p><i>Simbol Proses Komputer (Computer Process)</i> digunakan untuk menggambarkan proses pengolahan data dengan komputer.</p>
	<p><i>Simbol Proses Manual (Manual Process)</i> digunakan untuk menggambarkan proses yang masih menggunakan operasi <i>manual</i>.</p>
	<p><i>Simbol Persiapan (Preparation)</i> digunakan untuk mencantumkan dokumen atau proses yang dianggap sebagai persiapan yang dilakukan atau disiapkan sebelum masuk ke dalam alur proses dari sistem yang di buat.</p>
	<p><i>Simbol Keputusan (Decision)</i> digunakan untuk menggambarkan keputusan yang harus dibuat</p>

	dalam proses pengolahan data.
	<p><i>Simbol Dokumen (Document)</i> digunakan untuk menggambarkan semua jenis dokumen, yang merupakan formulir yang digunakan untuk merekam data terjadinya suatu transaksi.</p>
	<p><i>Simbol Dokumen Lisan</i> digunakan untuk memberitahukan informasi secara lisan.</p>
	<p><i>Simbol Dokumen dan Tembusannya</i> digunakan untuk menggambarkan dokumen asli dan tembusannya. Nomor lembar dokumen dicantumkan di bagian sudut kanan atas simbol.</p>
	<p><i>Simbol Berbagai Dokumen (Multiple Document)</i> digunakan untuk menggambarkan berbagai jenis dokumen yang digabungkan bersama di dalam satu paket. Nama dokumen dituliskan pada masing-masing simbol, sedangkan nomor lembar dokumen dicantumkan di sudut kanan atas simbol dokumen yang bersangkutan.</p>

	<p><i>Simbol Arsip Sementara (Temporary File)</i></p> <p>digunakan untuk menunjukkan tempat penyimpanan dokumen yang dokumennya akan diambil kembali dari arsip tersebut di masa yang akan datang untuk keperluan yang lebih lanjut terhadap dokumen tersebut. Untuk menunjukkan urutan pengarsipan dokumen digunakan simbol :</p> <p>T = kronologis, menurut tanggal.</p>
	<p><i>Simbol Arsip Permanen (Permanent File)</i></p> <p>digunakan untuk menggambarkan arsip permanen yang merupakan tempat penyimpanan dokumen yang tidak akan diproses lagi dalam sistem yang bersangkutan.</p>
	<p><i>Simbol Penghubung pada halaman yang sama (On Page Connector)</i> memungkinkan aliran dokumen berhenti di suatu lokasi pada halaman tertentu dan kembali berjalan di lokasi lain pada halaman yang sama.</p>

	<p><i>Simbol Penghubung pada halaman yang berbeda (Off Page Connector)</i> digunakan untuk menggambarkan proses yang masih berlanjut pada halaman lain, biasanya menggunakan huruf (A,B,C,D,...).</p>
	<p><i>Simbol Garis Alir (Flowline)</i> digunakan untuk menggambarkan arah proses pengolahan data</p>
	<p><i>Simbol Barang</i> digunakan untuk menggambarkan barang</p>
	<p><i>Simbol Data</i> digunakan untuk menggambarkan data</p>

Sumber : Stern, 1995

2.2.5 PHP

PHP yang merupakan singkatan dari *Personal Home Page* adalah suatu bahasa pemrograman berbentuk skrip yang bersifat *server-side* dan didesain secara khusus untuk web. Sifat *server-side* di sini berarti

pengerjaan skrip akan dilakukan di *server*, baru kemudian hasilnya dikirimkan ke *browser*. Skrip ini akan diimplementasikan pada halaman HTML yang akan dieksekusi setiap kali halaman tersebut diakses/dikunjungi.

2.2.5.1. Script PHP

Ada dua cara yang sering digunakan dalam menuliskan *script* atau program PHP yaitu *embedded script* dan *non-embedded script*.

a. *Embedded script*

Cara ini dilakukan dengan meletakkan *script* PHP di antara *tag-tag* HTML. Berikut contoh *embedded script*:

```
<html>
<head>
<title>embedded script</title>
</head>
<body>
<?
Echo "Ini adalah contoh embedded script";
?>
</body>
</html>
```

b. *Non-embedded script*

Cara ini merupakan pembuatan program murni PHP, di mana *tag-tag* HTML yang diletakkan di dalamnya. Akan tetapi cara ini tidak banyak diketahui atau tidak populer di kalangan *programmer*. Contoh:

```
<?
echo "<HTML>";
echo "<HEAD>";
echo "<TITLE> Non-embedded script </TITLE>";
echo "</HEAD>";
echo "<BODY>";
echo "Ini adalah contoh non-embedded script";
echo "</BODY>";
echo "</HTML>";
?>
<html>
<head>
<title>embedded script</title>
</head>
<body>
<?
Echo "Ini adalah contoh embedded script";
?>
</body>
</html>
```

2.2.5.2. Kelebihan PHP

Ada beberapa kelebihan dari PHP antara lain:

1. *Server side*, artinya tidak diperlukan adanya kompatibilitas *browser* atau harus menggunakan *browser* tertentu. *Server* yang akan mengerjakan skrip tersebut dan hasil yang dikirimkan kembali ke *browser* biasanya dalam bentuk teks ataupun gambar sehingga dapat dikenali oleh *browser* apapun.

2. Skrip asli PHP tidak dapat dilihat oleh *browser* sehingga keamanan lebih terjamin.
3. *Open source*, sehingga *source code* dari PHP dapat digunakan, diganti atau diubah tanpa terkena biaya.
4. *Multi platform*. Selain dapat dijalankan pada *platform* LINUX, PHP juga dapat dijalankan pada *platform* windows dengan menggunakan *software* PHP untuk Windows, dengan *web server* IIS pada Windows 2000/XP ataupun PWS pada Windows 98.
5. Koneksi ke basis data. PHP mendukung banyak sekali basis data dalam mode *native* dan melalui ODBC. Sistem basis data yang telah didukung oleh PHP hingga saat ini adalah: Oracle, Sybase, mSQL, MySQL, Solid, Generic, ODBC, PostgreSQL, Adabas D, FilePro, Velocis, dBase, Unix dbm dan semua basis data dengan *interface* ODBC.
6. HTTP *Cookie*. PHP secara transparan mendukung HTTP *cookie*. *Cookie* adalah mekanisme untuk menyimpan data pada *browser* sehingga administrator dapat melacak atau mengidentifikasi pemakai.
Administrator
dapat menset *cookies* menggunakan fungsi *setcookie()*. *Cookie* adalah bagian dari HTTP *header*, sehingga fungsi *setcookie()* harus dipanggil sebelum ada output yang dikirimkan ke *browser*. Hal ini sama dengan saat menggunakan fungsi *header()*. Setiap *cookie* yang dikirimkan ke

administrator dari klien akan secara otomatis dianggap sebagai variabel PHP seperti halnya variable dari metode GET dan POST. Jika hendak mengisi banyak data untuk satu *cookie*, hanya dengan menambahkan tanda [] pada nama *cookie*.

7. *File Upload*. PHP mampu menerima *file* yang di-*upload* dari setiap *browser* yang RFC-1867 *compliant*. Kemampuan *upload* ini memungkinkan pemakai untuk meng-*upload file* teks atau *binary*. Dengan fungsi-fungsi *authentication* dan logik PHP, dimiliki kontrol untuk menentukan siapa yang boleh meng-*upload* dan apa yang harus dikerjakan terhadap *file* tersebut setelah *file* itu selesai di-*upload*.

8. Pengaksesan *File* Gambar dengan PHP. PHP tidak terbatas hanya untuk menghasilkan *output* halaman HTML. PHP juga dapat digunakan untuk menghasilkan *file* gambar dengan format antara lain: GIF, JPEG, PNG dan WBMP. Bahkan agar lebih elegan gambar dapat bersifat *stream* dari *server* ke *browser*, tetapi PHP terlebih dahulu harus di-*compile* dengan GD *library* yang mengandung fungsi-fungsi manipulasi gambar.

9. *Error Handling*. Seluruh PHP *expression* dapat dipanggil dengan prefix “@”, yang akan mematikan *error reporting* untuk *expression* itu. Jika terjadi *error* pada *expression* itu dan *feature track errors* diaktifkan. Untuk mendapatkan pesan error dapat menggunakan variabel global \$php_errormsg.

2.2.6 MySql

MySQL adalah sebuah aplikasi *Relational Database Management Server* RDBMS yang sangat cepat dan kokoh. MySQL merupakan sebuah *server* basis data yang banyak digunakan di *internet* karena keandalannya dan juga karena sifatnya yang *shareware*. Bahasa yang digunakan dalam MySQL sama dengan bahasa yang dipakai oleh *server* basis data lainnya, yaitu bahasa SQL. SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses *server* basis data. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dalam konteks bahasa SQL, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan struktur dua dimensi yang terdiri atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column*). Baris pada tabel sering disebut sebagai *attributes* atau *field*. Keseluruhan tabel itu dihimpun dalam satu kesatuan yang disebut basis data. Ketika membandingkan MySQL dengan sistem basis data yang lain, '*performance*' (dayaguna), '*support*', keistimewaan, kebebasan dan pembatasan dalam penggunaan dan harga adalah faktor-faktor yang perlu dipikirkan. Sebagai pertimbangan, MySQL mempunyai banyak keistimewaan yang ditawarkan, yaitu:

- a. *Speed* (kecepatan). MySQL bekerja dengan cepat. Para pengembang berpendapat bahwa MySQL adalah basis data yang tercepat yang pernah didapat. Pendapat ini dapat dibuktikan melalui berbagai macam pengujian

yang dapat dilihat pada tabel perbandingan yang terdapat pada *web site*

<http://www.mysql.com/benchmark.html>.

b. *Ease of use* (kemudahan dalam penggunaan). MySQL adalah sistem basis data yang sederhana dengan performa tinggi dan sangat mudah untuk di-*setup* dan sangat mudah dalam pengaturan, dibandingkan dengan sistem basis data yang lebih besar.

c. *Cost* (biaya). MySQL adalah gratis untuk semua pengguna.

d. *Query language support* (mendukung bahasa query). MySQL memahami SQL (*Structured Query Language*), bahasa umum yang digunakan pada hampir semua basis data modern. MySQL juga dapat diakses menggunakan aplikasi yang mendukung ODBC (*Open Database Connectivity*), yaitu sebuah protokol untuk komunikasi basis data yang dikembangkan oleh Microsoft.

e. *Capability* (kemampuan). Banyak klien dapat melakukan koneksi ke *server* pada saat bersamaan. Klien dapat menggunakan beberapa basis data secara bergantian. Terdapat beberapa macam *interface* yang dapat digunakan untuk melakukan koneksi ke basis data dan melakukan *queries* serta menampilkan hasil. Sebagai tambahan bermacam-macam *programming interface* yang tersedia adalah C, perl, Java, PHP, dan Python.

f. *Connectivity and Security* (konektifitas dan keamanan). MySQL terhubung dengan jaringan secara keseluruhan dan basis data dapat diakses dari manapun melalui *internet*, sehingga dapat membagi data dimana saja. Dan MySQL

memiliki kontrol akses (*privilege*) sehingga siapa saja yang tidak diperbolehkan melihat data tidak dapat melihat data.

g. *Portability*. MySQL dapat berjalan di banyak varian dari UNIX dan juga dapat berjalan pada sistem selain sistem UNIX., seperti Windows dan OS/2. MySQL dapat berjalan pada PC rumahan (*home PC*) sampai dengan PC *server* dengan teknologi tinggi.

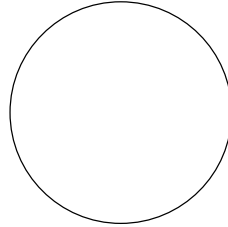
h. *Open distribution* (distribusi yang terbuka). MySQL mudah untuk didapatkan, hanya dengan menggunakan *web browser* sudah dapat mendapatkan *source* dari MySQL.

2.2.7 Diagram Arus Data

DFD adalah representasi dari sebuah sistem secara grafis yang digambarkan dengan sejumlah simbol tertentu untuk menunjukkan perpindahan data dalam proses–proses suatu sistem. Dalam hal ini *DFD* menunjukkan perpindahan dan perubahan data dalam suatu sistem. Meskipun diberi nama *DFD*, namun penekanan pada *DFD* lebih pada prosesnya, bahkan *DFD* merupakan salah satu alat pemodelan proses dari sistem yang paling sering digunakan. Simbol-simbol yang digunakan dalam *DFD* hanya terdiri dari empat macam yaitu proses, arus data, simpanan data, dan *entity* (*external entity*). Berikut uraian mengenai empat macam simbol yang ada:

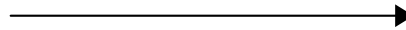
a. Proses adalah simbol yang mengubah suatu data dari suatu bentuk menjadi bentuk yang lain. Atau dengan kata lain, proses menerima *input* data dan

mengeluarkan *output* data lain yang telah diproses. Simbol dari proses dalam DFD ada dua macam:



Gambar 2.2.9 Simbol Proses

b. Arus Data atau *Data Flow* adalah aliran yang menunjukkan perpindahan data dari satu bagian ke bagian yang lain dalam sistem. *Data flow* dalam DFD disimbolkan dengan tanda panah dan diberi nama atau keterangan di sampingnya yang menunjukkan data apa yang mengalir. Contoh:



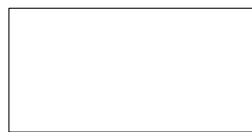
Gambar 2.2.10 Simbol Arus Data

c. Simpanan Data adalah tempat penyimpanan data dalam suatu sistem, baik secara manual maupun secara elektronik. Simpanan data digunakan jika suatu proses perlu menggunakan data tersebut lagi. Simbol dari simpanan data dalam DFD ada dua macam:



Gambar 2.2.11 Simbol Simpanan Data

d.Entity adalah seseorang, sekelompok orang, sebuah departemen di dalam maupun di luar organisasi, atau sebuah sistem yang lain yang memberikan *input* untuk sistem yang ada atau menerima *output* dari sistem yang ada. *External entity* juga disebut *terminator*, karena merupakan batas dari sebuah sistem. Dalam *DFD entity* disimbolkan dengan sebuah kotak persegi panjang sebagai berikut:



Gambar 2.2.12 Simbol *External Entity*

Dalam membangun *DFD*, dilakukan secara bertingkat, yaitu mulai dari *Context Diagram* atau *DFD level 0*, *DFD level 1*, *DFD level 2*, dan seterusnya tergantung kebutuhan. *Context Diagram* adalah *DFD* yang menunjukkan batas-batas dari sebuah sistem informasi, yaitu *top-level view* dari sebuah sistem. Untuk menggambar *Context Diagram*, hanya dibuat satu proses saja yang menggambarkan keseluruhan proses dari sistem tersebut dan beberapa *entity* disekelilingnya yang berhubungan. Dari *DFD level 0*, kadang-kadang ada proses yang dapat dijabarkan lebih detail lagi. Jadi dari sebuah *DFD level 0* dapat dibuat sejumlah *DFD level 1*, demikian pula dari setiap *DFD level 1* tersebut jika masih dapat dijabarkan lagi menjadi *DFD level 2*, 3, dan seterusnya. Sebuah *DFD* minimal digambarkan sampai level 0 saja, namun dalam kenyataan sebuah *DFD* umumnya pasti terdiri dari beberapa level.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1. DESKRIPSI UMUM

Penelitian ini dilaksanakan di P.T. Higina Alhadin, Jln. Lombok No73, Jakarta Pusat. MRA (Mugi Rekso Abadi) group merupakan salah satu perusahaan yang menaungi beberapa perusahaan, salah satu divisi MRA group adalah MRA *printed media*. MRA *Printed Media* ini sendiri merupakan gabungan dari beberapa perusahaan yang memproduksi beberapa majalah dengan membeli ijin (*lisence*) terbit dari majalah aslinya yang berasal dari luar negri untuk diterbitkan di Indonesia. Walaupun tidak semua majalah ini *lisence*, namun 90% dari majalah yang ada di bawah MRA *printed media* ini merupakan *lisence* dari luar.

MRA *printed media* terdiri dari dua belas perusahaan dibawahnya. Masing-masing perusahaan menerbitkan satu majalah, persusahaan tersebut diantaranya PT Higina Alhadin memiliki ijin untuk menerbitkan majalah cosmopolitan Indonesia, PT Media Rahayu Citra memiliki ijin untuk menerbitkan majalah cosmogirl Indonesia, PT Media Insani Abadi memiliki ijin untuk menerbitkan majalah bazaar Indonesia, PT Fajar Harapan Mandiri memiliki ijin untuk menerbitkan majalah *for him magazine* Indonesia, PT Media Talenta Utama memiliki ijin untuk menerbitkan majalah *autocar* Indonesia, PT Media Dian Sejahtera memiliki ijin untuk menerbitkan majalah *spice & fitness Indonesia*, PT Media Mitra Bunda

memiliki ijin untuk menerbitkan majalah *mother & baby* Indonesia, PT Media Griya Harmony memiliki ijin untuk menerbitkan majalah *goodhousekeeping* Indonesia, PT Media Tiara Victory memiliki ijin untuk menerbitkan majalah *Trax*, PT Media Esquirindo memiliki ijin untuk menerbitkan majalah *esquire* Indonesia, dan beberapa majalah lain.

Seperti yang sudah dijelaskan dalam Bab I, penulis akan membahas secara khusus mengenai PT Higinia Alhadin karena PT Higinia Alhadin merupakan PT yang terbesar dan dapat mewakili kegiatan dari semua PT yang lain.

PT Higinia Alhadin ini berdiri sejak tahun 1999. Awal mulanya, majalah dibawah naungan ini adalah majalah kesehatan dengan nama "HIGINA", namun berjalan dengan seiringnya waktu dan melihat perkembangan jaman, maka PT Higinia Alhadin membeli *lisence* dari luar negeri untuk menerbitkan majalah cosmopolitan. Pada zamannya, majalah cosmopolitan tidak banyak pesaingnya sehingga dapat maju dengan sukses.

Majalah cosmopolitan ini di tujukan bagi mereka wanita muda yang dengan berbagai impiannya, serta para wanita yang memiliki ambisi dan aspirasi. Sebagai seorang wanita cosmo mereka akan mendapatkan aspirasi-aspirasi baru dan pengalaman yang menarik serta menantang dalam gaya hidup mereka dan hubungan relasional. Karena hal tersebut mereka dijuluki *FUN FEARLESS FEMALE*. Secara umum para pelanggan majalah ini adalah wanita yang memiliki

umur antara delapan belas sampai tiga puluh empat tahun, baik mereka yang sudah bekerja maupun tidak.

Tiap tahun pendistribusian majalah ini terus meningkat, tahun 2002 jumlah penjualan majalah ini mencapai 100.000 copy hingga pada tahun 2005 mencapai jumlah 139.000 copy.

3.1.1 Visi dan Misi

Visi

Memajukan Indonesia untuk menyambut *millennium* baru dengan ide-ide baru, produk-produk baru dengan penyajian yang berbeda. Sehingga karena hal tersebut dapat menggugah minat para generasi muda untuk peka terhadap perubahan baru (*produk dan service*) yang tidak terkalahkan. Sebagai hasil akhirnya adalah dapat membangun Indonesia kearah yang lebih baik.

Misi

MRA selalu berusaha menjadi yang terbaik dalam bidangnya, menanamkan nilai-nilai kepada pelanggannya, dan berbagi kepada orang-orang yang memanfaatkan layanannya. Grup MRA juga berusaha menggabungkan citra dan apresiasi, sehingga memungkinkan para pelanggannya untuk menerapkan kebiasaan untuk tersebut yang berguna untuk menaungi bisnisnya yang bervariasi.

3.1.2 Struktur Organisasi

Untuk mencapai tujuan yang diinginkan, maka sebuah perusahaan haruslah mengatur pembagian tugas perusahaan sedemikian rupa. Bentuk dari pembagian tugas itu dapat digambarkan ke dalam suatu struktur organisasi atau bagan organisasi, berikut ini sturktur organisasi dari PT Higina Alhadin.



Gambar 3.1.1 Struktur Organisasi PT Higina Alhadin

Dengan melihat struktur organisasi pada gambar 3.1 diatas, maka PT Higina Alhadin ini memiliki delapan departemen yang saling terkait guna mencapai target penjualan. Tidak hanya departemen redaksi saja, namun juga didukung dengan departemen-departemen lain.

1. *Department Redaksi*

Department ini mempunyai fungsi untuk membuat majalah. Terdiri dari *editor in chief, managing editor, editor, reporter*. Reporter mencari berita dan informasi, *event*, kemudian membuat dalam suatu tulisan. Hasil tulisan ini kemudian akan diperiksa atau diedit oleh *editor*. *Managing editor* bertanggung jawab untuk mengolah tulisan ini menjadi tulisan baik dan berdaya guna bagi pembaca. Kemudian, *managing editor* mendiskusikannya dengan *editor in chief* sebelum proses pencetakan.

2. *Advertising & Sales Department*

Department ini bertanggung jawab untuk mencari klien (untuk pemasangan iklan di majalah), sponsor (untuk pelaksanaan event). Klien ataupun sponsor tersebut sangat diperlukan, karena biaya untuk ini sangatlah besar. Sehingga perusahaan akan mendapatkan profit yang lumayan besar. Didalam department ini memiliki beberapa account executive yang menjalankan tanggung jawab tersebut diatas.

3. *Marcomm (Marketing Communication)Department*

Department ini bertugas untuk mengangani pelaksanaan *event-event* yang dibuat oleh majalah. Mereka bertanggung jawab mulai dari ide acara tentunya haruslah dikemas menarik, penyusunan biaya, sampai kepada pelaksanaannya.

4. *Media Relation Department*

Tidak jauh berbeda dengan *department marketing communications*, tetapi mereka hanya sebatas kerjasama *event*. Misalnya sebagai media *partner* maka departemen ini akan menyediakan *stand* untuk penjualan majalah, mereka juga bertanggung jawab terhadap distribusi pembagian hadiah untuk para pemenang.

5. *Marketing Service Department*

Department Marketing Service ini bertanggung jawab atas pengelolaan *database* pelanggan. Mereka juga mengurus jika ada pelanggan yang komplain atau mereka yang ingin berlangganan. Selain itu *departement* ini juga bertanggung jawab atas *branding* majalah.

6. *HR&GA Departement*

Department ini mengurus urusan kedalam atau *intern* karyawan, selain itu mereka juga mengurus proses *recruitment* karyawan baru. *HRD (Human Resources Department)* lebih cenderung mengurus semua hal yang berkaitan dengan karyawan, sedangkan tim *GA (General Affair)* lebih kearah pemenuhan apa saja yang berkenaan dengan sarana penunjang kerja karyawan. Sebagai contoh tanggung jawab *human resources* adalah menghitung jumlah cuti, melakukan penghitungan tunjangan makan transport karyawan, *recruitment*, asuransi. Sedangkan tanggung jawab *general affair* misalnya pemenuhan kebutuhan alat-alat kerja karyawan, bensin untuk kendaraan operasional, maintenance gedung.

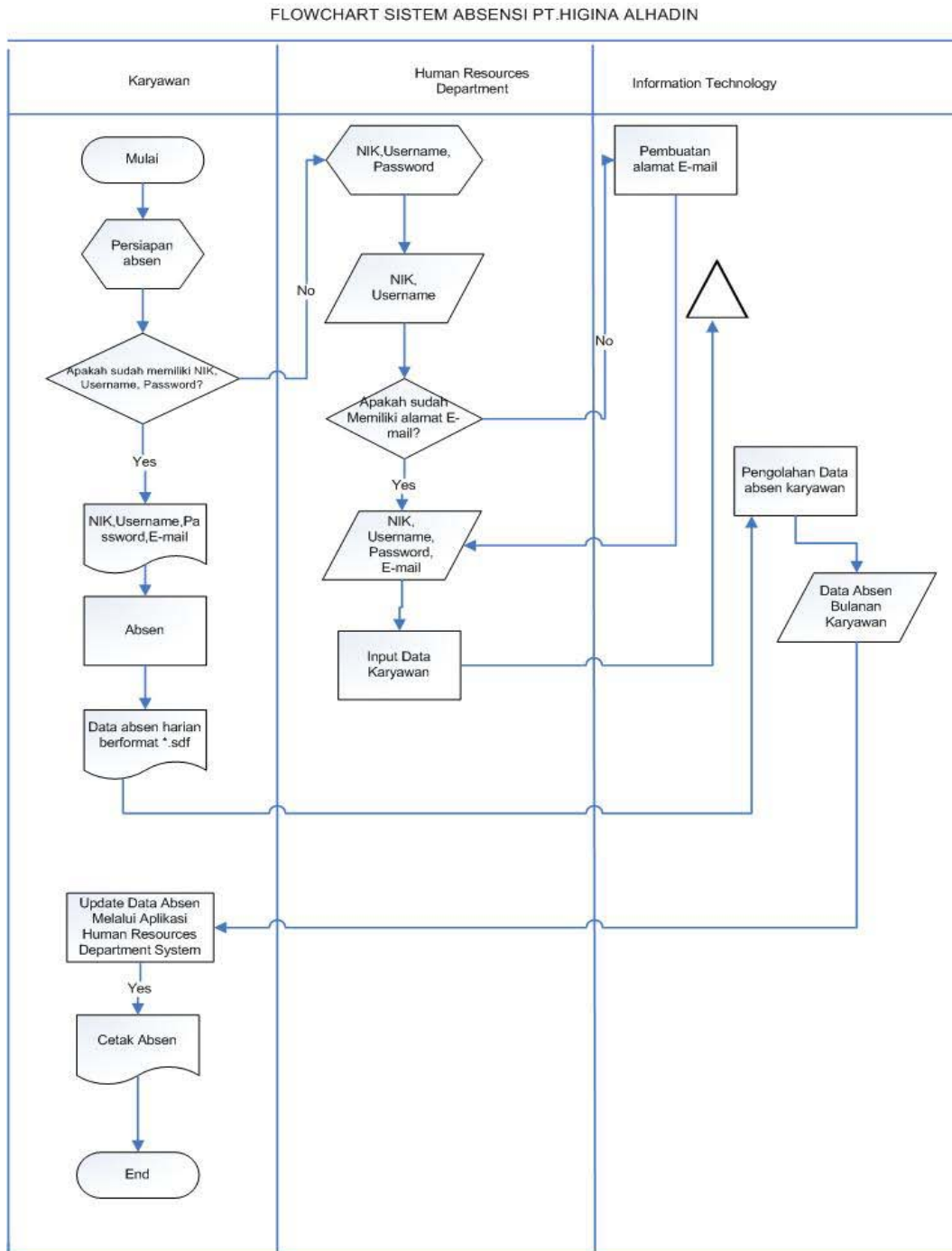
7. *Distribution & Circulation Departement*

Departement ini bertanggung jawab dan mengelola proses distribusi ke agen-agen majalah dan mencatat jumlah penjualan serta mencatat pula jumlah majalah yang tidak laku dipasar.

8. *Finance & Accounting Departement*

Departemen *finance* bertanggung jawab terhadap keuangan yang masuk maupun yang dikeluarkan oleh perusahaan. Seperti digambarkan dalam struktur bahwa tim *finance* dikepalai oleh *financial controller*.

3.2. ANALISA SISTEM ABSENSI



Gambar 3.2.1 Alur proses absensi karyawan PT Higina Alhadin

Secara umum untuk setiap karyawan atau karyawan baru akan mendapatkan nomor induk karyawan yang berjumlah delapan digit yang di generate *HRD (Human Resources Department)*. Nomor induk karyawan ini akan digunakan sebagai identitas unik dari masing-masing karyawan untuk melakukan absen jam kedatangan dan kepulangan. Proses selanjutnya human resources akan meng-informasikan kepada pihak *IT (Information Technologi)*. *IT* akan membuat alamat *e-mai-lnya*, jika sudah di buat maka akan segera di konfirmasi ke *hrd*. Kemudian *hrd* akan memasukkan data tersebut dan data-data lain (nama lengkap, *username*, *password*, sisa cuti) ke dalam *database*. Setelah dimasukkan, karyawan tersebut akan mendapat kan *username* dan *password* untuk *login* ke aplikasi absensi yang berbasis web. Aplikasi ini fungsinya adalah untuk mencatat jumlah absen karyawan selama satu bulan. Karyawan dapat melakukan pengecekan secara rutin dengan melakukan *login* ke dalam aplikasi tersebut. Pengecekan ini dilakukan karena mungkin saja absensi mereka ada yang tidak tercatat didalam aplikasi ini. Jika tidak tercatat maka akan sangat merugikan pihak karyawan, karena akan mengurangi perolehan uang makan dan transport mereka.

3.2.1 Bahan Penelitian

Data-data yang diperlukan dalam penelitian ini adalah:

1. Bagan struktur organisasi P.T. Higina Alhadin
2. File data absensi yang berformat *.sdf
3. Data *e-mail*, nik, nama, perusahaan.

3.2.2 Alat Penelitian

Alat penelitian yang diperlukan terbagi menjadi 2, yaitu Perangkat Lunak, dan Perangkat Keras.

1. Perangkat lunak

Perangkat lunak yang digunakan dalam penelitian ini antara lain:

- a. Adobe Photoshop CS 2, yang digunakan untuk mengolah gambar-gambar yang nantinya digunakan dalam perancangan aplikasi.
- b. MySQL Server 5.0.45, digunakan untuk mengelola *database* yang akan digunakan oleh aplikasi.
- c. Apache HTTPD Server 2.2.6, digunakan untuk mengelola dokumen web yang akan digunakan oleh aplikasi.
- d. PHP 5.2.5, sebagai script interpreter file-file .PHP

2. Perangkat Keras

Penelitian menggunakan komputer dengan spesifikasi pendukung sebagai berikut:

- a. *Processor : Intel® Pentium (R) 4 CPU 3.00GHZ*
- b. *Memori DDR2 2GB PC5300*
- c. *Mainboard Biostar TA-690G*
- d. *Hardisk Seagate SATA 160 GB*
- e. *Monitor, Mouse, Keyboard*

3.2.3 Metode Pengumpulan Data

1. Metode Wawancara.

Dalam metode ini untuk mengumpulkan data dengan cara bertanya langsung dengan pihak-pihak yang terkait, terutama pihak *human resources department* P.T. Higina Alhadin.

2. Metode Observasi

Metode pengumpulan data yang dilakukan dengan cara mengamati secara langsung aktifitas proses absen karyawan dalam P.T. Higina Alhadin.

3. Metode Kepustakaan

Metode kepustakaan dilakukan dengan cara membaca buku, brosur, formulir, literatur, dan mempelajari dokumen-dokumen yang terkait dengan permasalahan yang diteliti.

3.3. ANALISA PERMASALAHAN

Day	Number of visits	Pages	Hits	Bandwidth
14-Jan-08	129	3744	13063	264.85 MB
15-Jan-08	243	11225	27283	672.77 MB
14-Feb-08	153	4764	10065	240.06 MB
15-Feb-08	200	2860	5762	127.49 MB
14-Mar-08	67	1018	6658	137.52 MB
15-Mar-08	3	13	19	1.34 MB

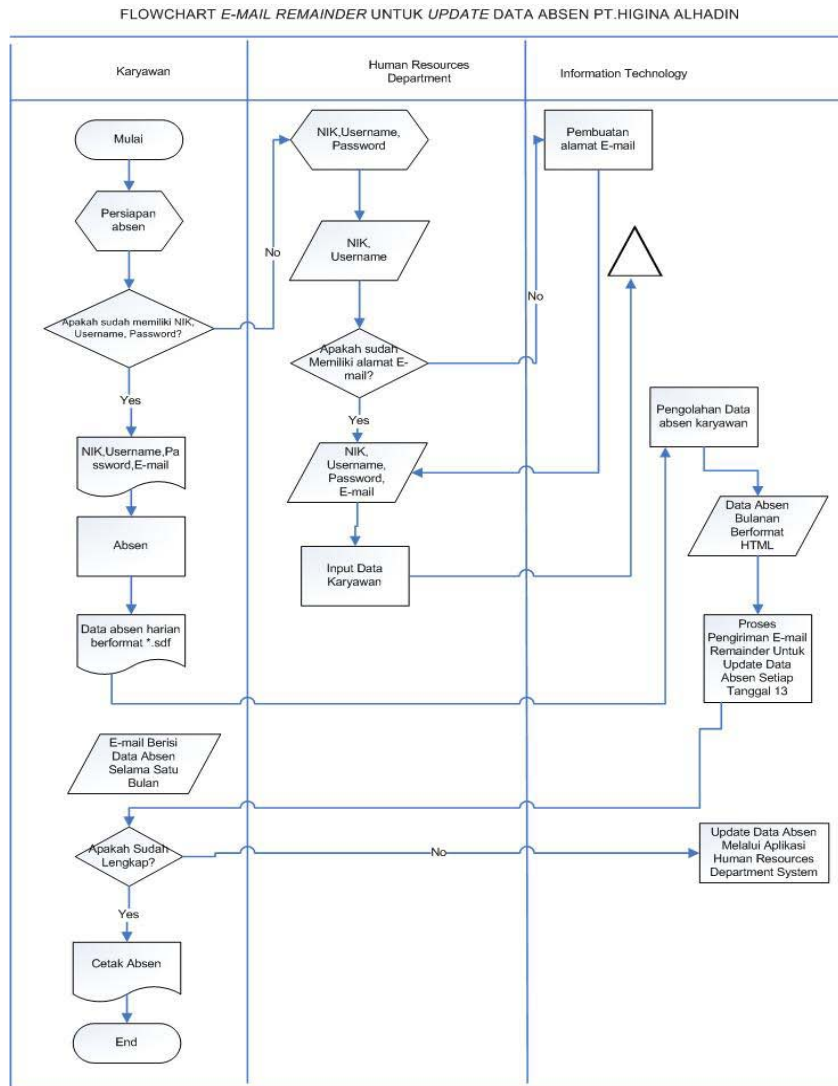
Tabel 3.3-1 Tabel Data Jumlah Bandwidth Tanggal Jatuh Tempo Penghitungan Absensi

Dari melihat Tabel 3.3-1, tabel jumlah bandwidth mendekati tanggal jatuh tempo, dan tingginya complain dari user yang tercatat di IT *department* jika mendekati tanggal jatuh tempo penghitungan absensi karyawan maka dapat disimpulkan masalah-masalah yang sering terjadi :

1. Jika semua karyawan pada tanggal jatuh tempo pengecekan, mereka meng-akses database secara bersamaan maka akan memperlambat kinerja dari *server* (dapat dilihat dari tabel 3.1).
2. Karena diakses oleh banyak karyawan maka akan memakan waktu yang cukup lama sehingga berakibat perhitungan yang dilakukan *HRD (Human Resource Department)* akan terhambat pula.
3. Pihak *HRD (Human Resource Department)* pada tanggal jatuh tempo harus melakukan pemberitahuan kepada masing-masing sekretaris redaksi untuk memberitahukan kepada masing-masing karyawannya agar melakukan pengecekan.
4. Tingginya angka komplain karyawan ke pihak IT dan HRD setiap mendekati tanggal jatuh tempo.

3.4. DESIGN SISTEM

Tabel 3.4-1 Alur Proses *E-mail Remainder* Untuk *Update* Data Absen P.T. Higina Alhadin



Sistem *email remainder* untuk *update* data absen yang akan dibangun bertujuan untuk menambah sistem yang selama ini digunakan sehingga dapat mempermudah, mempercepat dan meningkatkan akurasi proses-proses pengelolaan absensi, pemantauan pegawai, dan penghitungan tunjangan makan dan transport diantaranya:

- a) Proses pengecekan jam kerja kehadiran pegawai selama satu bulan
- b) Proses koneksi ke server aplikasi dapat diminimalisir.
- c) Proses pembuatan laporan statistik kehadiran pegawai baik harian, mingguan, bulanan maupun tahunan.
- d) Proses penghitungan uang makan dan transport.
- e) Proses approval dari masing-masing departemen head.

Otomatisasi ini sepertinya sederhana, tapi sangat membantu. Diharapkan dengan penambahan ini, sistem absensi dan pemantauan pegawai akan lebih mudah, cepat dan akurat. Hal ini dimungkinkan karena seluruh data dan informasi akan dengan cepat didapatkan.

3.5. SPESIFIKASI KEBUTUHAN

3.5.1 Input

Sistem yang dibangun membutuhkan sejumlah input data, yang dapat dijabarkan sebagai berikut:

1. Data Absen Karyawan

Data absen mentah absen karyawan yang diambil langsung dari mesin absensi berformat) *.sdf berisi tanggal, bulan, tahun, jam, menit, detik, nik dari karyawan.

2. Data Lengkap Karyawan

Daftar nik, nama, alamat *e-mail*, perusahaan, jabatan.

3.5.2 Output

1. Sajian Informasi *e-mail remainder*

Sajian informasi mengenai data absen selama satu bulan. Bila data sudah benar maka tidak perlu melakukan update ke aplikasi absensi.

3.6. DESKRIPSI FUNGSIONAL

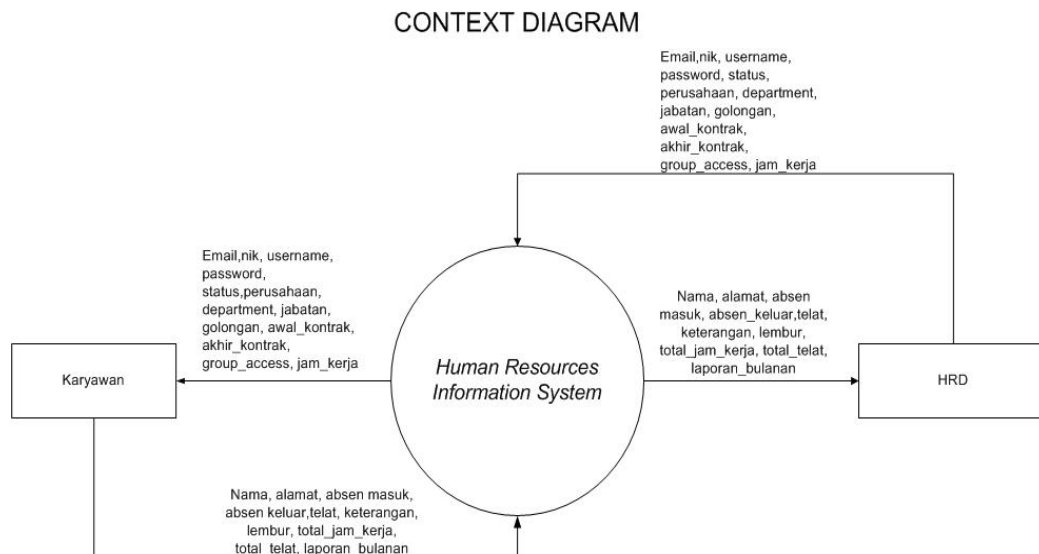
3.6.1 *Context Diagram*

E-mail remainder ini merupakan salah satu bagian dari *human resources information system*. Sehingga *context diagram*, *data flow diagram* dari sistem *e-mail remainder* untuk *update* data absen ini akan di gabungkan dalam *human resources information system*. Ini terdiri dari dua *entity*, yaitu karyawan dan *hrd (human resources department)*. Berikut ini aliran data dari masing-masing *entity* :

1. Data yang masuk dari karyawan ke proses *email remainder* untuk konformasi *update* data absen di PT Higinia Alhadin adalah nama, alamat, absen masuk, absen keluar, telat, keterangan, lembur, total_jam_kerja, total_telat, laporan_bulanan.
2. Data nama, alamat, absen masuk, absen keluar, telat, keterangan, lembur, total_jam_kerja, total_telat, laporan_bulanan yang masuk ke proses *email remainder* tersebut kemudian diterima oleh *entity hrd (human resources department)*.

3. Data yang masuk dari entity hrd (*human resources department*) ke proses email remainder untuk *update* data absen adalah email, nik, *username*, *password*, *status*, perusahaan, *department*, jabatan, golongan, awal_kontrak, akhir_kontrak, group_access, jam_kerja.
4. Data tersebut kemudian diberikan ke karyawan sebagai informasi karyawan dan disimpan didalam *database*.

Penjelasan lebih lanjut mengenai *contex diagram*, dapat dilihat dalam gambar 3.6.1 di bawah ini.



Gambar 3.6.1 Context Diagram Untuk Human Resources Information System

3.6.1.1. DFD Level 1

DFD Level 1 pada proses email *remainder* untuk *update* data absen ini terbagi menjadi tiga proses utama yaitu pendataan, absen, dan cetak.

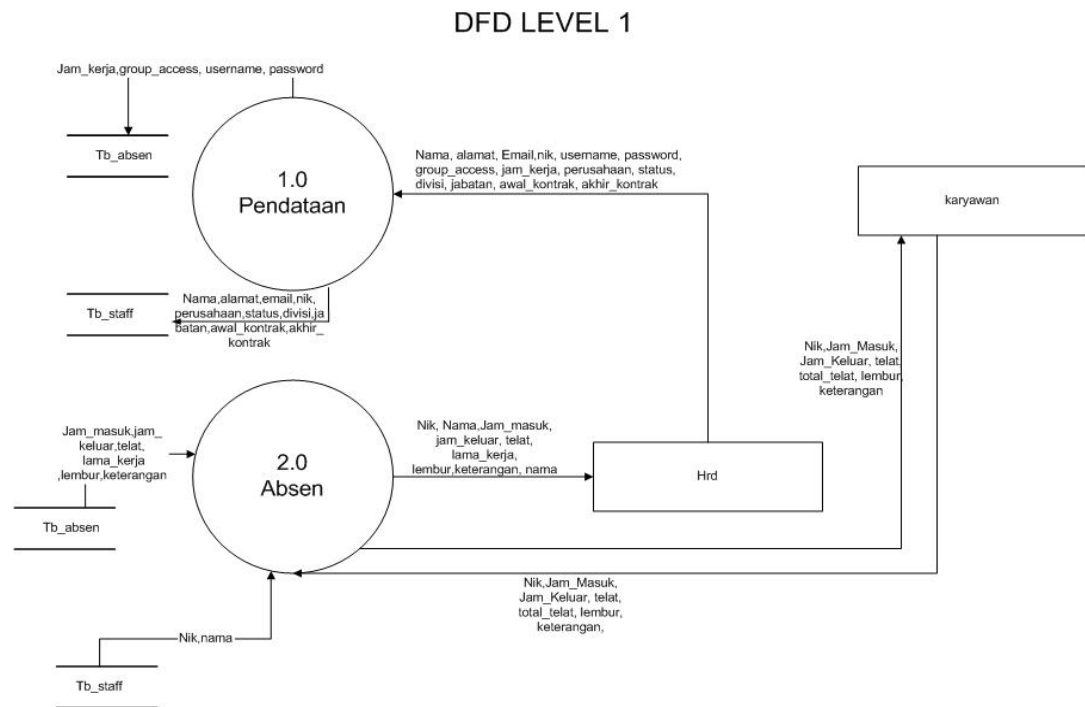
1. Proses pendataan

Data yang masuk dalam proses ini merupakan *inputan* dari *entity human resources department*. Data ini akan diproses dan disimpan kedalam data *store* staff dan absen.

2. Proses absen.

Data yang masuk ke proses absen ini merupakan *inputan* dari *entity karyawan*. Karyawan melakukan proses absen setiap harinya dan akan diproses dan simpan kedalam data *store* absen. Data yang keluar dari proses absen ini *entity human resource department* dapat melihat data absen bulanan masing-masing karyawan.

Penjelasan lebih lanjut mengenai DFD *Level 1*, dapat dilihat dalam gambar 3.6.2



Gambar 3.6.2 DFD Level 1 Untuk Human Resources Information System

3.6.1.2. DFD Level 2

3.6.1.2.1. DFD LEVEL 2 Proses Pendataan

1. Tambah Data Karyawan

Proses tambah data karyawan ini *entity human resources department* melakukan penginputan data karyawan kedalam proses tambah data karyawan untuk di simpan kedalam data *store staff*.

2. Ubah Data Karyawan

Data yang masuk ke dalam proses ubah data karyawan ini *entity human resources department* mencari data karyawan yang akan diubah kemudian data yang sudah diubah tersebut akan disimpan kembali kedalam data store staff.

3. Hapus Data Karyawan

Data yang masuk kedalam proses hapus data karyawan *entity human resources department* memasukkan karyawan yang akan dihapus, kemudian data tersebut akan dihapus dari data store staff.

3.6.1.2.2. DFD Level 2 Proses Absen

1. Kirim E-mail

Data yang masuk kedalam proses kirim e-mail ini diperoleh dari database absen. Kemudian data tersebut akan dikirimkan ke karyawann setiap bulannya terschedule setiap tanggal 13 setiap bulannya.

2. Ubah Absen

Data yang masuk kedalam proses ubah absen diperoleh dari *entity karyawan dan human resorces department*. Entity karyawan memasukkan data keterangan dan tanggal jika absensi mereka tidak terdeteksi atau kosong. Sedangkan *entity*

human resources department akan memasukkan data yang diubah sesuai dengan keterangan dari *entity karyawan*. Data tersebut akan di simpan kedalam data *store* absen.

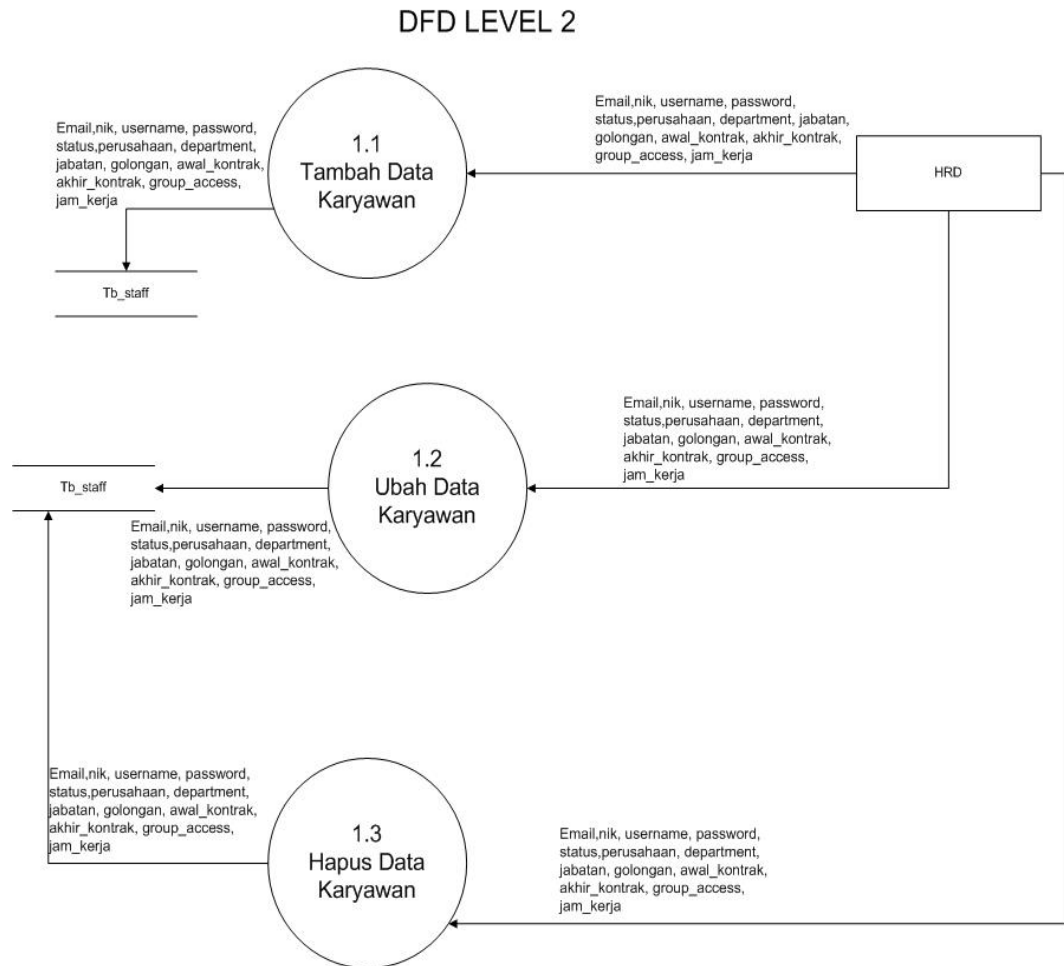
3. Lihat Absen

Proses lihat absen ini memperoleh data dari *entity karyawan* dan *human resources department*. *Entity* karyawan memasukkan data nik, nama, bulan, tahun yang akan dicari kemudian proses lihat absen mengakses data store absen sesuai dengan data masukan dan kemudian akan menampilkan detail absen yang dicari. Hal yang sama akan dilakukan jika *entity human resources department* ingin mencari data absen.

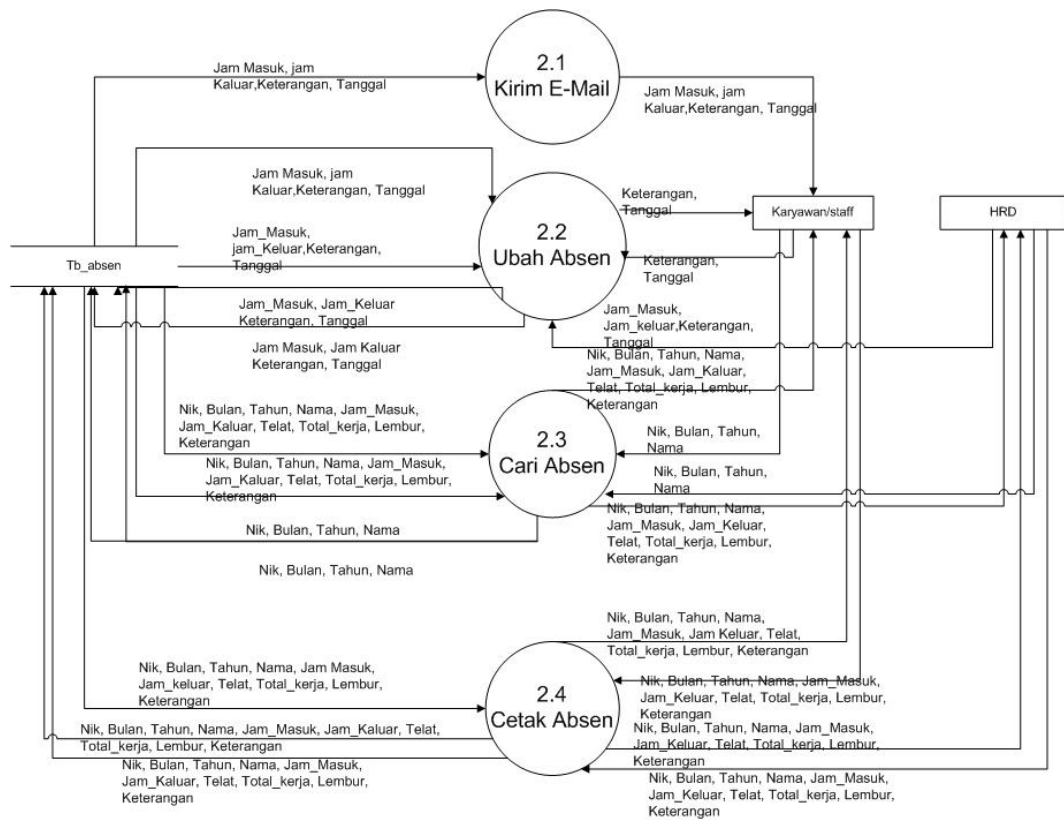
4. Cetak Absen

Proses cetak absen ini dapat dilakukan oleh karyawan maupun *human resources department*. Jika karyawan ingin mencetak data absen mereka, data yang masuk kedalam proses cetak absen adalah detail data dari masing-masing karyawan. Data detail tersebut akan dicari ke dalam data store absen, dan kemudian akan diberikan kembali ke *entity karyawan*. Data yang sama juga akan diterima oleh proses cetak absen jika *entity human resources department* akan melakukan cetak absen.

Penjelasan lebih lanjut mengenai DFD *Level 2*, dapat dilihat dalam gambar 3.6.3 dan 3.6.4 berikut ini.



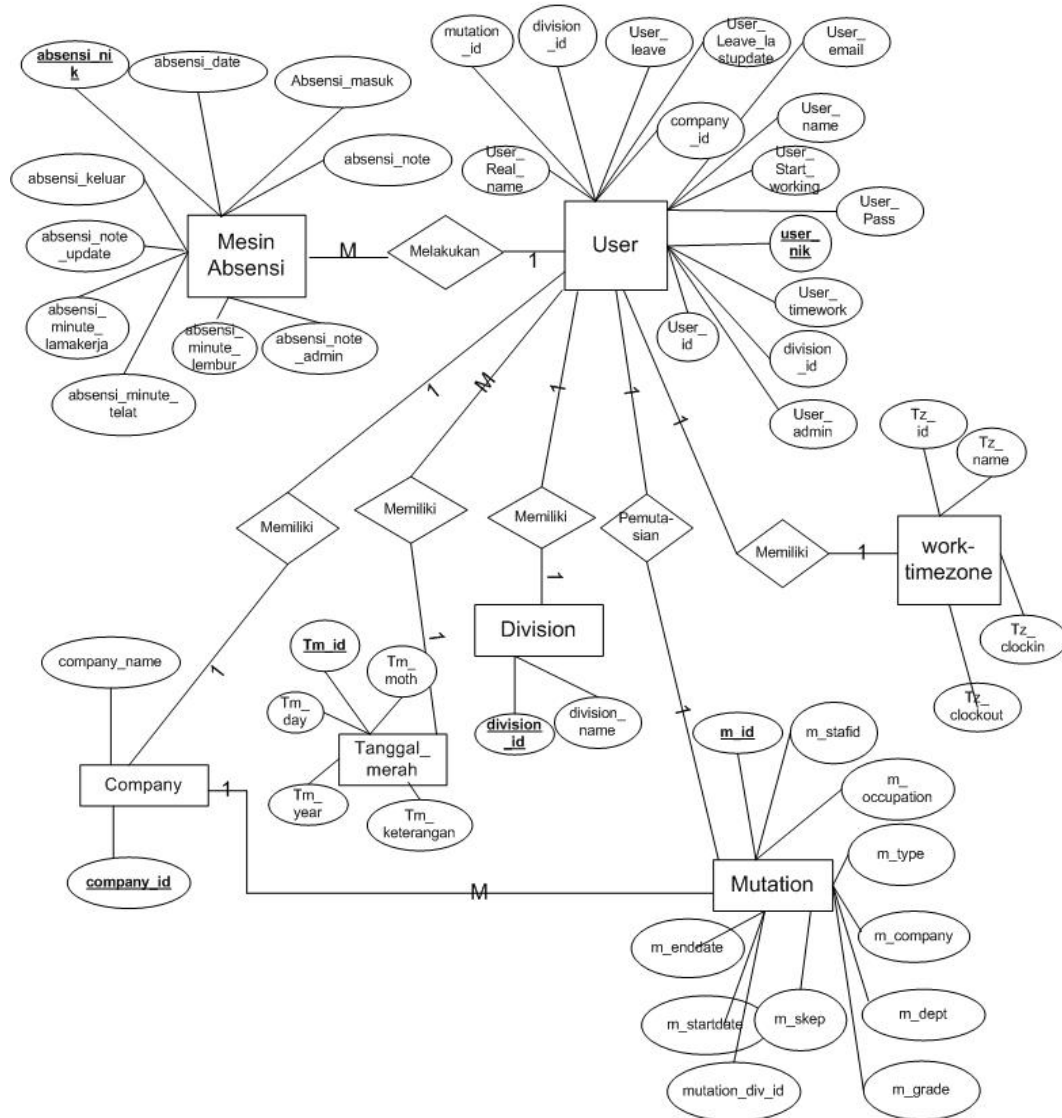
Gambar 3.6.3 DFD Level 2 Untuk Human Resources Information System



Gambar 3.6.4 DFD Level 2 *Human Resources Information System*

3.6.2 E-R Diagram

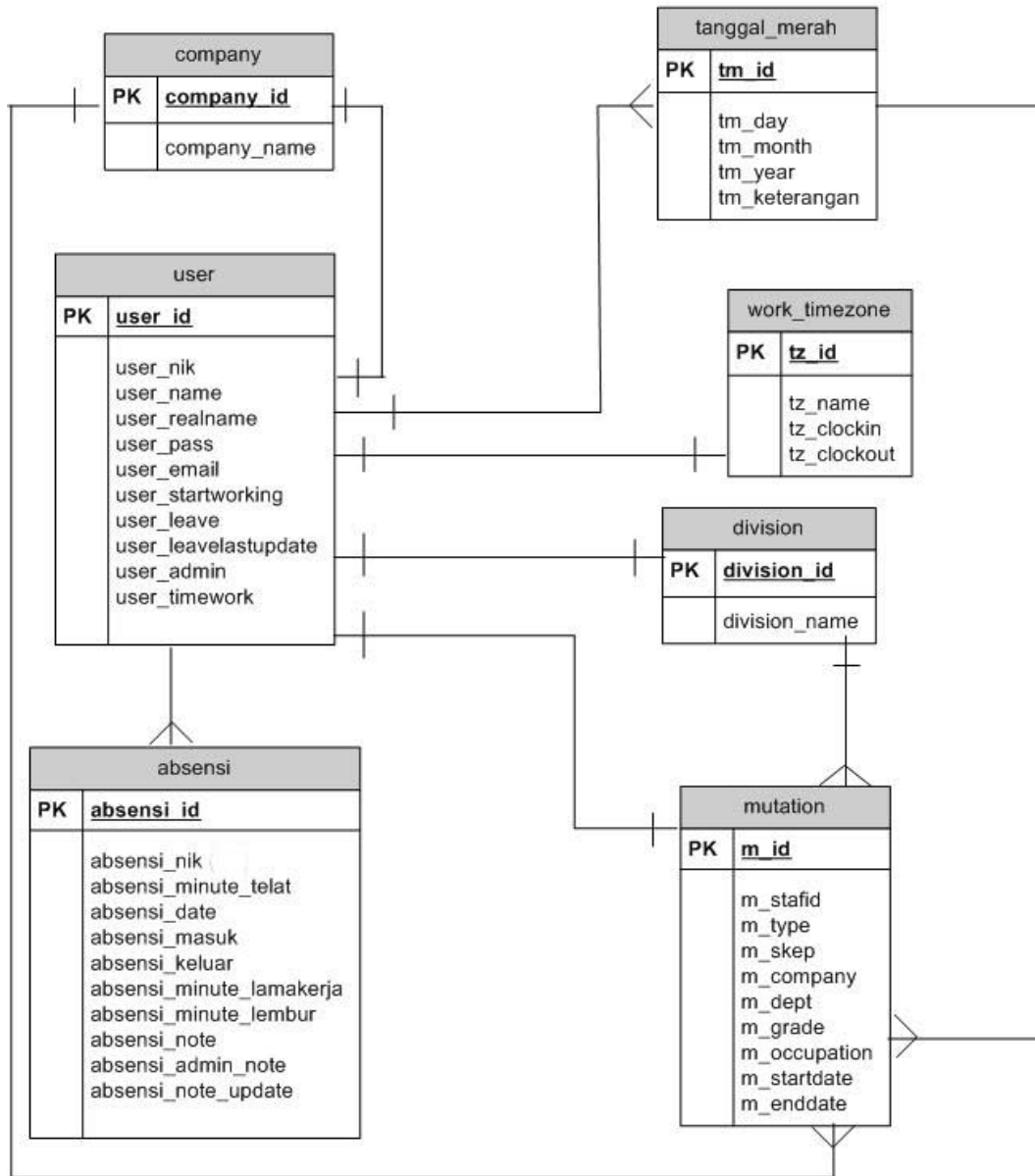
ER-DIAGRAM



Gambar 3.6.5 Entity Relational Diagram Untuk Sistem Email Remainder Untuk Konfirmasi Update Data Absen

3.6.3 Physical Data Model

Physical Data Model



Gambar 3.6.6 Physical Data Model *Untuk* Sistem Email Remainder *Untuk* Konfirmasi Update Data Absen

3.7. Perancangan

3.7.1 Perancangan *Database*

Database dikelola melalui MySQL Server dengan nama database *absensi-smartweb* , dengan format tabel InnoDB.

1. Nama tabel : company

Primary key : company_id

Foreign key : -

Fungsi : Tabel yang menyimpan database nama perusahaan

Tabel 3.7-1 Rancangan tabel company

Nama Field	Tipe Data	Ukuran	Keterangan
company_id	Integer	11	<i>Primary key</i> untuk mengidentifikasi nama perusahaan
company_Name	Varchar	30	Nama perusahaan

2. Nama tabel : division

Primary key : division_id

Foreign key : -

Fungsi : untuk menyimpan data-data *warehouse*

Tabel 3.7-2 Rancangan tabel division

Nama Field	Tipe Data	Ukuran	Keterangan
division_id	Integer	11	<i>Primary key</i>
Division_name	Varchar	30	Nama divisi

3. Nama tabel : user

Primary key : user_id

Foreign key : -

Fungsi : Tabel yang menyimpan data *staff* sebagai *user*nya

Tabel 3.7-3 Rancangan tabel user

Nama Field	Tipe Data	Ukuran	Keterangan
User_id	Integer	11	<i>Primary key</i>
User_nik	Varchar	10	NIK Karyawan
User_name	Varchar	10	User Untuk login
User_realname	Varchar	64	Nama Karyawan
User_pass	Varchar	15	Password
User_email	Varchar	30	Alamat <i>e-mail</i> karyawan
User_startworking	Date	-	Menentukan tanggal mulai bekerja
User_leave	Float	-	Cuti atau ijin
User_leave_lastupdate	Date	-	Cuti atau ijin terakhir kali di update
User_admin	Varchar	2	Memberikan hak akses
User_timework	Date	-	Jam kerja karyawan
User_company	Integer	11	Identifier nama perusahaan

4. Nama tabel : absensi

Primary key : absensi_nik

Foreign key : -

Fungsi : untuk menyimpan data-data transaksi absen karyawan

Tabel 3.7-4 Rancangan tabel absensi

Nama Field	Tipe Data	Ukuran	Keterangan
Absensi_nik	Varchar	8	<i>Primary key</i>
Absensi_date	Date	-	Tanggal Absen karyawan
Absensi_masuk	Varchar	12	Jam Masuk Karyawan
Absensi_keluar	Varchar	15	Jam Keluar Karyawan
Absensi_minute_telat	Integer	11	Total Menit telat karyawan
Absensi_minute_lamakerja	Integer	11	Total Menit lama kerja karyawan
Absensi_minute_lembur	Integer	11	Total Menit lama lembur
Absensi_note	Text	-	Keterangan
Absensi_note_update	Date	-	Keterangan setelah di Update
Absensi_admin_note	Tesxt	-	Keterangan dari admin

5. Nama tabel : mutation

Primary key : m_id

Foreign key : -

Fungsi : Untuk menyimpan data-data mutasi karyawan

Tabel 3.7-5 Rancangan tabel mutation

Nama Field	Tipe Data	Ukuran	Keterangan
M_id	Integer	11	<i>Primary key</i>
M_stafid	Varchar	10	Id karyawan
M_skep	Varchar	64	Keterangan
M_type	Varchar	3	Type Mutasi
M_company	Integer	11	Nama perusahaan tempat mutasi
M_dept	Integer	11	Departemen tempat mutasi
M_grade	Integer	11	Grade
M_occupation	Integer	11	Alasan
M_startdate	Date	-	Tanggal awal mutasi
M_enddate	Date	-	Tanggal akhir mutasi

6. Nama tabel : tanggal_merah

Primary key : tm_id

Foreign key : -

Fungsi : Untuk menyimpan hari libur selama satu tahun

Tabel 3.7-6 Rancangan tabel tanggal_merah

Nama Field	Tipe Data	Ukuran	Keterangan
Tm_id	Integer	11	<i>Primary key</i>
Tm_day	Integer	11	Hari tanggal merah
Tm_month	Integer	11	Bulan tanggal merah
Tm_year	Integer	11	Tahun tanggal merah
Tm_keterangan	Varchar	128	Keterangan

7. Nama tabel : Work_timezone

Primary key : tz_id

Foreign key : -

Fungsi : Untuk menyimpan jam kerja karyawan yang shift

Tabel 3.7-7 Rancangan tabel *work_timezone*

Nama Field	Tipe Data	Ukuran	Keterangan
Tz_id	Integer	11	<i>Primary key</i>

3.7.2 Perancangan Antarmuka

NIK :

Employee :

Company :

Department :

Date	Tanggal	Masuk	Telat	Overtime	Keterangan

Summary

Gambar 3.7.1 Perancangan File yang akan di Kirim Melalui E-mail Karyawan

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. LINGKUNGAN YANG DIBUTUHKAN

4.1.1 *Hardware*

Spesifikasi *hardware* yang dibutuhkan agar sistem berjalan dengan baik adalah:

4.1.1.1. *Server*

1. CPU : minimal Pentium 4 1.8GHz atau AMD Athlon 1.6GHz, disarankan *multi-processor* atau CPU dengan teknologi *dual-core*
2. Memori : minimal 256MB, disarankan 512MB
3. Hardisk : minimal 40GB, disarankan 80GB
4. *VGA Card* : tidak memerlukan pemrosesan grafik yang intensif, maka *VGA card* apapun dapat dipergunakan

4.1.1.2. *Client*

1. CPU : minimal Pentium III 667MHz atau AMD Athlon 650MHz
2. Memori : minimal 128MB, disarankan 256MB
3. Hardisk : tergantung persyaratan yang diminta oleh sistem operasi yang digunakan
4. *VGA Card* : tidak memerlukan pemrosesan grafik yang intensif, maka *VGA card* apapun dapat dipergunakan

4.1.2 *Software*

Spesifikasi *software* yang dibutuhkan agar sistem berjalan dengan baik adalah:

4.1.2.1. *Server*

1. Sistem Operasi yang dibutuhkan adalah sistem operasi kelas *server*, disarankan turunan dari UNIX
2. Apache HTTPD Server minimal versi 2.2.4
3. MySQL Server minimal versi 5.0.1
4. PHP minimal versi 5.2.1, dikonfigurasi sebagai Apache *module*
5. Aplikasi *E-mail Server* apapun yang mendukung protokol POP3 untuk menerima *incoming-mail* dan protokol SMTP untuk mengirimkan (*outgoing*) mail.

4.1.2.2. *Client*

1. Sistem Operasi yang dibutuhkan adalah sistem operasi kelas *consumer*, semisal Windows ME/2000 Workstation/XP, MacOS X, atau Linux *distro* apapun
2. *Web browser* yang mendukung: Internet Explorer 5 dan versi di atasnya, Mozilla/Netscape minimal versi 6, Firefox semua versi, Opera semua versi, Safari/Konqueror semua versi
3. *E-mail client* (Outlook, The Bat, Pegasus, Thunderbird dan sebagainya)

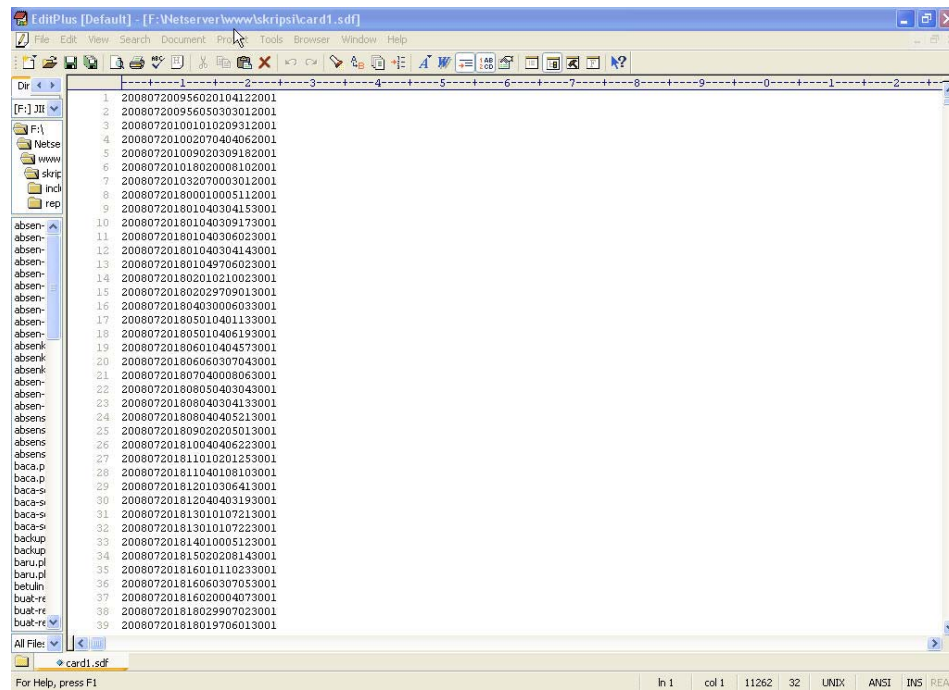
Sistem ini diuji dengan menggunakan metode pengujian *black box*. Metode *black box* adalah metode pengujian yang menguji suatu sistem tanpa harus mengetahui proses internal yang berada pada sistem tersebut

Setelah semua file-file aplikasi telah selesai dirancang dan diujicoba melalui jaringan lokal (*Local Area Network*). Kemudian seluruh tahapan pengujian aplikasi berjalan dengan normal.

4.2. PROSES

4.2.1 Pengambilan Data Absen dari mesin absen

E-mail remainder ini tidak memiliki syarat-syarat khusus. Setiap karyawan baik yang baru atau sudah lama jika mempunyai alamat *e-mail* maka akan mendapatkan *e-mail remainder* ini. Karyawan akan memasukkan nik setiap kali absen dan akan disimpan sementara yang absen setiap harian. *Database* ini berformat *.sdf, *database* ini berisi:



Gambar 4.2.1 Isi File *.sdf

Isi file *.sdf tersebut juga diterjemahkan seperti dibawah ini:



Gambar 4.2.2 Detail Isi File *.sdf

Untuk dapat menetapkan jam masuk dan jam keluar, maka system akan membandingkan jam, menit, detik mana yang lebih kecil kemudian akan di masukkan kedalam jam kedatangan. Sedangkang jam, menit, detik yang lebih

besar sebagai jam kepulangan. Kondisi lain jika dalam satu hari karyawan melakukan absensi tiga kali, maka proses nya akan dilakukan pembandingan juga.

Agar dapat secara otomatis melakukan proses pengambilan dari mesin absensi maka harus dibuat dua buah file *.bat yang akan di-*schedule* harian untuk menjalankan *script* baca.php yang berfungsi membaca *database* *.sdf dari mesin absen dan memindahkan ke dalam *database mysql*).

4.2.2 Pengiriman *E-mail Remainder*

Proses pengiriman e-mail remainder ini dibagi menjadi:

1. Pembuatan File HTML

Pembuatan file HTML dilakukan untuk memudahkan proses pengiriman data absen selama satu bulan kepada masing-masing karyawan. Data absen dari masing-masing karyawan ini akan di generate dalam bentuk HTML dan akan ini akan di ikut sertakan dalam *e-mail* yang akan dikirimkan.

2. Pengiriman E-mail

Jika proses generate file HTML ini sudah selesai maka akan secara otomatis mengirimkan *e-mail* berisi informasi data absen kepada masing-masing karyawan.

Dear **Dedi Nuryadi**,

This is an automated email message to inform you about your daily attendance between 2008-06-17 to 2008-07-13. Should you have any comment or note, please log-in to <https://intranet.skripsi.web.id> with your NIK and PIN.

Thank you for your kind attention.

Regards,
Human Resources Information System (HRIS)
HRD

NIK : 01030950
Employee : Dedi Nuryadi
Company : Cosmopolitan
Department : Redaksi

Date	Tanggal	Masuk	Keluar	Telat	Overtime	Keterangan
Selasa	2008-06-17	12:14	19:01	189		
Rabu	2008-06-18	08:49	18:14			
Sabtu	2008-06-21	-	19:40			
Minggu	2008-06-22	08:59	15:28		240	
Senin	2008-06-23	08:36	20:40		120	
Selasa	2008-06-24	08:53	21:16		120	
Rabu	2008-06-25	09:07	21:26	2		
Sabtu	2008-06-28	09:03	21:23		240	
Minggu	2008-06-29	08:45	20:17		240	
Senin	2008-06-30	09:03	20:36		120	
Selasa	2008-07-01	08:52	20:23		120	
Rabu	2008-07-02	08:54	20:46		120	
Minggu	2008-07-06	08:50	20:43		240	
Senin	2008-07-07	08:43	21:03		120	
Selasa	2008-07-08	08:57	20:18		120	
Rabu	2008-07-09	09:04	18:29			
Sabtu	2008-07-12	09:03	21:09		240	
Minggu	2008-07-13	09:03	18:29		240	
				191		

Summary

Hari kerja yang tercatat mesin : 18
Absensi sempurna : 17

Berikut ini tanggal kerja senin-jum'at yang tidak ada keterangan :

- Senin 2008-06-16
- Kamis 2008-06-19
- Jum'at 2008-06-20
- Kamis 2008-06-26
- Jum'at 2008-06-27
- Kamis 2008-07-03
- Jum'at 2008-07-04
- Kamis 2008-07-10
- Jum'at 2008-07-11
- Senin 2008-07-14
- Selasa 2008-07-15

Gambar 4.2.3 E-mail remainder yang dikirim ke masing-masing karyawan setiap bulannya.

Kedua proses di-*schedule* bulanan (untuk membaca *script* laporan.php yang berfungsi untuk membuat *file html* dan kemudian melakukan proses pengiriman *e-mail* ke masing-masing karyawan).

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan pengujian aplikasi, maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem *E-mail remainder* untuk *update* data absen ini mempermudah transaksi absensi bulanan karyawan di P.T. Higina Alhadin.
2. Jika semua karyawan pada tanggal jatuh tempo pengecekan, mereka meng-akses *database* secara bersamaan maka akan memperlambat kinerja dari *server*. Kekurangan tersebut diatas dapat diatasi, dapat dibuktikan dengan turunnya penggunaan *bandwidth* dan angka *complain* ke IT yang menurun juga.
3. Dengan sistem yang dikembangkan ini, data akan terintegrasi dengan baik sehingga akan menghasilkan informasi yang cepat dan akurat.

5.2. Saran

1. Untuk saat ini informasi yang disajikan hanya *summary* jam absen karyawan untuk hari-hari yang tidak lengkap jam kedatangan dan jam kepulangan. Diharapkan untuk saat mendatang informasi yang diberikan kepada user bisa ditambahkan seperti *siswa cuti*, *total telat*, *total potongan*.

2. *User interface* file ber-format html yang dikirimkan melalui alamat *e-mail* dibuat lebih menarik dan *user friendly*.
3. Masih banyak karyawan yang mengabaikan *e-mail remainder* ini dan langsung melakukan login ke aplikasi *human resources information system*, maka sosialisasi kepada *user* terus dilakukan sehingga *e-mail remainder* ini dapat benar-benar bermanfaat.
4. Saat ini modul ini dijalankan dengan menggunakan sistem operasi windows, untuk kedepannya diharapkan modul ini dapat dijalankan menggunakan sistem operasi *opensource* karena dari estimasi biaya akan lebih murah karena tidak harus membayar *license*.

DAFTAR PUSTAKA

- Kadir, Abdul. (2002). Dasar Pemrograman WEB Dinamis Menggunakan PHP. Yogyakarta: Penerbit ANDI Yogyakarta hal
- Fathansyah, 1999, *Basis Data*, Penerbit Informatika, Bandung
- Jogiyanto, H.M., 1999, *Analisis & Desain: Sistem Informasi Pendekatan Terseruktur Tteori dan Praktek Aplikasi Bisnis*, Andi Offset, Yogyakarta
- Pressman, R.S, 2002, *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*, Penerbit Andi, Yogyakarta
- Federal Information Processing Standard Publications (FIPS PUB), 19 April 1993, *Electronic Data Interchange (EDI)*, Jurnal Internet, National Institute of Standards and Technology (<http://www.itl.nist.gov/fipspubs/>)
- Sidik, Betha, 2002, *Pemrograman Web dengan PHP*, Informatika Bandung, Bandung
- Kadir, Abdul. (1999). Konsep dan Tuntunan Praktis Basis Data. Yogyakarta: Penerbit ANDI

LAMPIRAN A

LISTING PROGRAM

➤ **Baca.Php**

```
<?
$dbhost="localhost";
$dbuser="root";
$dbpass="";
$dbname="absensi-smartweb";
$dbconn=mysql_connect("$dbhost","$dbuser","$dbpass") or die("Gagal
koneksi ke mysql");
$dbopen=mysql_select_db($dbname) or die("Gagal membuka database
$dbname");
$overtime_min=array(
    'daily'=>(2*60),
    'weekend'=>(4*60));
$overtime_max=array(
    'daily'=>(2*60),
    'weekend'=>(4*60));
$time_in=array(
    'hour'=>"09",
    'minute'=>"05"
```



```
$time_out=array(
    'hour'=>"18",
    'minute'=>"00");
$nextday_out=array(
    'hour'=>"03",
    'minute'=>"30");
$weekdays=array("Minggu","Senin","Selasa","Rabu","Kamis","Jum'at","Sabtu");
$fname="CARDRAW.SDF";
$txtfile=`type c:\Netserver\www\skripsi\CARD1.SDF`;
$arr_isifile=explode("\n",$txtfile);
asort($arr_isifile);
$absensi=array()
function is_libur($tgl,$bln) {
    $sql="select tm_day, tm_month from tanggal_merah where
tm_day='$tgl' and tm_month='$bln'";
    $query=mysql_query($sql);
    if (mysql_num_rows($query)>0) {
        return true;
    } else {
        return false;
    }
}
```

```

    }
}

function ambil_jam_masuk($usernik) {

    $sql="

    select

        user_nik,

        user_timework

    from user

    where user_nik='$usernik'

    ";

    $query=mysql_query($sql);

    $userinfo=mysql_fetch_assoc($query);

    $user_timework=ereg_replace("\|",",",ereg_replace("\|$","",ereg_replac
e('^\\|',"", $userinfo['user_timework'])));

    if (!empty($user_timework)) {

    $sql="select tz_clockin from work_timezone where tz_id in
($user_timework)";

        $query=mysql_query($sql);

        $arrHasil=array();

        while ($data=mysql_fetch_assoc($query)) {

```

```
        $arrHasil[]=$data['tz_clockin'];
    }
    return $arrHasil;
} else {
    return false;
}
}
function hitung_lainnya($tgl,$usernik,$masuk="", $keluar="") {
    global $time_in, $time_out, $weekdays, $overtime_min,
    $overtime_max;
    $arr_tgl=explode("-", $tgl);
    $serial_tgl=mktime(0,0,0,$arr_tgl[1],$arr_tgl[2],$arr_tgl[0]);
    ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $masuk, $arr_masuk);
```

```

    $serial_keluar=mktime($sarr_keluar[4],$sarr_keluar[5],0,$sarr_keluar[2],$sarr_k
eluar[3],$sarr_keluar[1]);

    $hari=date("w",$serial_tgl);

    $telat=0;

if (($hari!=0)and($hari!=6)and(!is_libur($sarr_tgl[2],$sarr_tgl[1]))) {

if (empty($masuk)) $telat=4*60;} else {

$SarrTimezone=ambil_jam_masuk($usernik);

    if (is_array($SarrTimezone)) { while (list($k,$v)=each($SarrTimezone)) {

        $sarr_timein=explode(":",$v);

            $telat=($sarr_masuk[4]*60+$sarr_masuk[5])-
($sarr_timein[0]*60+$sarr_timein[1]);

                }

            }

        }

    } else {$telat=0;

}

if ($telat<0) $telat=0;

$lamakerja=0;

if ((!empty($masuk)and(!empty($keluar))) {

    if (($sarr_masuk[1]!=$sarr_keluar[1])or

        ($sarr_masuk[2]!=$sarr_keluar[2])or

        ($sarr_masuk[3]!=$sarr_keluar[3])) {

            $lamakerja=abs(((23*60)+59)-
(($sarr_masuk[4]*60)+$sarr_masuk[5]));

            $lamakerja+=abs(((sarr_keluar[4]*60)+$sarr_keluar[5]));

```

```
        } else  
  
        {  
  
        $lamakerja=abs(((Sarr_keluar[4]*60)+$sarr_keluar[5])-  
((Sarr_masuk[4]*60)+$sarr_masuk[5]));  
  
        }  
  
    } else {  
  
        $lamakerja=0;  
  
    }  
  
    if (($hari=="0")or($hari=="6")) {  
  
        if ($lamakerja>$overtime_max['weekend']) {  
  
            $lembur=$overtime_max['weekend'];  
  
        } else if ($lamakerja<$overtime_min['weekend']) {  
  
            $lembur=0;  
  
        } else  
  
            $lembur=$lamakerja;}  
  
    }
```

```

        } else {

                $lembur=null;

        }

}

$sarr_data=array($stelat,$lamakerja,$lembur);

return $sarr_data;

}

function hitung_lainnya2($masuk="", $keluar="") {

        global $time_in, $weekdays, $overtime_min, $overtime_max;

        ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $masuk, $sarr_masuk);

        $serial_masuk=mktime($sarr_masuk[4], $sarr_masuk[5], 0, $sarr_masuk[2], $sarr_masuk[3], $sarr_masuk[1]);

        ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $keluar, $sarr_keluar);

        $serial_keluar=mktime($sarr_keluar[4], $sarr_keluar[5], 0, $sarr_keluar[2], $sarr_keluar[3], $sarr_keluar[1]);

        $hari=$weekdays[date("w", $serial_masuk)];

        $stelat=null;

```

```

    if
    (($sarr_masuk[1]=="2004")and($sarr_masuk[2]=="07")and($sarr_masuk[3]=="03")and
    ($GLOBAL['current_nik']=="02021004")) {

        echo "<pre>";

        print_r($serial_masuk);

        echo "<pre>";

    }

    if (($sarr_masuk[1]!=$sarr_keluar[1])or
        ($sarr_masuk[2]!=$sarr_keluar[2])or
        ($sarr_masuk[3]!=$sarr_keluar[3])) {

        $lamakerja=abs(((23*60)+59)-(($sarr_masuk[4]*60)+$sarr_masuk[5]));

        $lamakerja+=abs(($sarr_keluar[4]*60)+$sarr_keluar[5]);

    } else {

        $lamakerja=abs(((($sarr_keluar[4]*60)+$sarr_keluar[5])-
    (($sarr_masuk[4]*60)+$sarr_masuk[5]));

    }

    if (($hari=="0")or($hari=="6")) {

        if ($lamakerja>$overtime_max['weekend']) {

            $lembur=$overtime_max['weekend'];

        } else if ($lamakerja<$overtime_min['weekend']) {

            $lembur=null;

        } else {

            $lembur=$lamakerja;

        }

    }

```

```

    } else {
    if
    (((($sarr_keluar[4]*60+$sarr_keluar[5])-
    ($time_out['hour']*60+$time_out['minute']))>$overtime_min['daily'])and
    ($telat<=0)and
    ($lamakerja>0)) {
        $lembur=$overtime_max['daily'];
    } else {
        $lembur=null;
    }
}
    $sarr_data=array($telat,$lamakerja,$lembur);
    return $sarr_data;
}
while (list($idx,$isi)=each($sarr_isifile)) {
    if (trim($isi)!=") {
        ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{4})([0-9]{8})([0-9]{2,4})", $isi, $sarr_hasil);
        list($semua,$thn,$bln,$tgl,$jam,$nik,$status)=$sarr_hasil;
        ereg("([0-9]{2})([0-9]{2})", $jam, $sarr_jam);
        $nextday_out_serial=mktime($nextday_out['hour'],$nextday_out['minute'],0,$bln,($tgl-1),$thn);
    }
}

```



```

        if (!is_array($absensi[$thn.$bln.$tgl.'-'. $nik])) {

            $absensi[$thn.$bln.$tgl.'-'. $nik]=array();

        }

        if
(($arr_jam[1]*60+$arr_jam[2])<($nextday_out['hour']*60+$nextday_out['min
ute']))

        {

            $absensi[date("Y",$nextday_out_serial).date("m",$nextday_out_serial
).date("d",$nextday_out_serial).'-'. $nik][].=$thn.$bln.$tgl.$jam;

        } else {

            $absensi[$thn.$bln.$tgl.'-'. $nik][].=$thn.$bln.$tgl.$jam;

        }

    }

}

$baris=0;

echo "Processing ".count($absensi)." data\n";

while (list($idx,$arr_data)=each($absensi)) {

    list($current_date,$current_nik)=split('-', $idx);

    ereg("([0-9]{4})([0-9]{2})([0-9]{2})", $current_date,$arr_date);

    list($full_date,$current_year,$current_month,$current_day)=$arr_date

;

```

```
$jammasuk=null;

$jamkeluar=null;

$telat=null;

$lamakerja=null;

$overtime=null;

$baris++;

$field_list="absensi_nik,
            absensi_date";

$value_list="$current_nik',
            ".$current_year."-".$current_month."-
            ".$current_day."";

if (count($arr_data)==1) {
    ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $arr_data[0], $arr_jammasuk);
    if (($arr_jammasuk[4]*60+$arr_jammasuk[5])>(12*60)) {
```

```
jamkeluar=$arr_data[0];

        $jammasuk=null;

    } else {                                $jammasuk=$arr_data[0];

        $jamkeluar=null;

    }

} else if (count($arr_data)==2) {

    ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $arr_data[0], $arr_jammasuk);

    ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $arr_data[1], $arr_jamkeluar);

    if (($arr_jammasuk[4]*60+$arr_jammasuk[5])>(12*60)) {

        $jammasuk=$arr_data[0];

    }

}
```

```

    } else {

        ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $sarr_data[0], $sarr_jammasuk);

        ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $sarr_data[(count($sarr_data)-1)], $sarr_jamkeluar);

        $serial_jammasuk=mktime($sarr_jammasuk[4], $sarr_jammasuk[5], 0, $sarr_jammasuk[2], $sarr_jammasuk[3], $sarr_jammasuk[1]);

        $serial_jamkeluar=mktime($sarr_jamkeluar[4], $sarr_jamkeluar[5], 0, $sarr_jamkeluar[2], $sarr_jamkeluar[3], $sarr_jamkeluar[1]);

        if (($sarr_jammasuk[4]*60+$sarr_jammasuk[5])>(12*60)) {
            $jamkeluar=$sarr_data[0];
        } else {
            $jammasuk=$sarr_data[0];
        }

        if (($sarr_jamkeluar[4]*60+$sarr_jamkeluar[5])<(12*60)) {
            if ($serial_jammasuk>$serial_jamkeluar) {
                $jammasuk=$sarr_data[(count($sarr_data)-1)];
            }
        } else {
            $jamkeluar=$sarr_data[(count($sarr_data)-1)];
        }

        $sarr_tambahan=hitung_lainnya($sarr_date[1]."-".$sarr_date[2]."-".$sarr_date[3], $current_nik, $jammasuk, $jamkeluar);

        list($telat, $lamakerja, $lembur)=$sarr_tambahan;
    }

```

```
$field_list.=" , absensi_minute_telat"; $value_list.=" , NULL";  
  
    }  
  
    if (!empty($jamkeluar)) {  
        $field_list.=" , absensi_keluar";  
        $value_list.=" , '$jamkeluar'";  
    }  
  
    if (!empty($telat)and($telat>0)) {  
$field_list.=" , absensi_minute_telat";  
        $value_list.=" , '$telat'";  
    }  
  
    if (!empty($lamakerja)) {  
        $field_list.=" , absensi_minute_lamakerja";  
        $value_list.=" , '$lamakerja'";  
    }  
  
    if (!empty($lembur)) {  
        $field_list.=" , absensi_minute_lembur";  
        $value_list.=" , '$lembur'";  
    }  
  
    if ((!empty($jammasuk))or(!empty($jamkeluar))) {  
$sql="  
        INSERT INTO absensi_". $current_year ." (  
            $field_list ) values ( $value_list );
```

```

if (mysql_errno()<>0) {
    $arr_field=explode(" ",$field_list);
    $arr_value=explode(" ",$value_list);
    $sql="
select * from absensi_". $current_year."
        where absensi_nik="". $current_nik."" and
absensi_date="". $current_year."-". $current_month."-". $current_day.""
    ";
    $query=mysql_query($sql);
    $data_lama=mysql_fetch_assoc($query);

$jammasuk_lama=$data_lama['absensi_masuk'];
ereg("([0-9]{4})([0-9]{2})([09]{2})([09]{2})",$jammasuk_lama,$arr_jammasuk_lama);
$idx_masuk=array_search('absensi_masuk',$arr_field);
$jammasuk_baru=$arr_value[$idx_masuk];
ereg("([09]{4})([09]{2})([09]{2})([09]{2})([09]{2})",$jammasuk_baru,$arr_jammasuk_baru);
$serial_masuk_baru=mktime($arr_jammasuk_baru[4],$arr_jammasuk_baru[5],0,
    $arr_jammasuk_baru[2],$arr_jammasuk_baru[3],$arr_jammasuk_baru[1]);
$serial_masuk_lama=mktime($arr_jammasuk_lama[4],$arr_jammasuk_lama[5],0,
    $arr_jammasuk_lama[2],$arr_jammasuk_lama[3],$arr_jammasuk_lama[1]);
if ($idx=array_search('absensi_keluar',$arr_field)) {$sql="update
absensi_". $current_year." set absensi_keluar="". $arr_value[$idx]." where
absensi_nik="". $current_nik."" and absensi_date="". $current_year."-
". $current_month."-". $current_day";mysql_query($sql);}

```

```
If ($idx=array_search('absensi_masuk',$arr_field)) {$sql="update
absensi_". $current_year." set absensi_masuk=".$sarr_value[$idx]." where
absensi_nik=".$current_nik." and absensi_date=".$current_year."-
".$current_month."-".$current_day";
if ((empty($jammasuk_lama))or($serial_masuk_baru<$serial_masuk_lama))
{ mysql_query($sql); }
    }
}
}
$namabaru="CARDRAW-".date("Y-m-d-H-i").".SDF";
?>
```

➤ **Laporan.Php**

```
<?
$dbhost="localhost";
$dbuser="root";
$dbpass="";
$dbname="absensi-smartweb";

include_once("includes/htmlMimeMail5.php");

$report_path='C:\Netserver\www\skripsi\reports';

$dbconn=mysql_connect("$dbhost","$dbuser","$dbpass") or die("Gagal koneksi ke
mysql");

$dbopen=mysql_select_db($dbname) or die("Gagal membuka database $dbname");

$arrHari=array('Minggu','Senin','Selasa','Rabu','Kamis','Jum\'at','Sabtu');

function array_diff_key()
{
    $arrs = func_get_args();
    $result = array_shift($arrs);
    foreach ($arrs as $array) {
        foreach ($result as $key => $v) {
            if (array_key_exists($key, $array)) {
                unset($result[$key]);
            }
        }
    }
    return $result;
}
```



```

}

$sql="SELECT `user_nik` , `user_name` , `user_realname` , `user_email` ,
`company_name` , `div_name` , `m_startdate` , `m_enddate`

FROM `user`

LEFT JOIN mutation ON mutation.m_id = user.user_company

LEFT JOIN company ON company.company_id = mutation.m_company

LEFT JOIN division ON division.div_id = mutation.m_dept

WHERE m_type<>'R'

";

if (!isset($_SERVER['HTTP_HOST'])) {

    $linebreak="\n";

    if ($argc!=3) {

        $namafile=basename(__FILE__);

        echo "\n".$namafile.": \n";

        echo "This command required 2 parameter.\n\n";

        exit;

    }

    $periode_awal=$argv[1];

    $periode_akhir=$argv[2];

} else {

    $linebreak="<br />\r\n";

    $periode_awal=$_REQUEST['periode_awal'];

    $periode_akhir=$_REQUEST['periode_akhir'];

}

```

```

if (!ereg("[0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $periode_awal)) {
    echo "Invalid start date $periode_awal.$linebreak";
    echo "Please use format yyyy-mm-dd$linebreak";
    exit;
}
if (!ereg("[0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $periode_akhir)) {
    echo "Invalid last date $periode_akhir.$linebreak";
    echo "Please use format yyyy-mm-dd$linebreak";
    exit;
}
$pathdir=$report_path.'/'.$periode_awal.'-'. $periode_akhir;
if (!file_exists($pathdir)) {
    @mkdir($pathdir);
} else {
    echo "Saving to old diretory $pathdir".$linebreak;
}
SarrAwal=explode("-", $periode_awal);
SarrAkhir=explode("-", $periode_akhir);
$numberofday_awal=date("t", mktime(0,0,0,$SarrAwal[1],$SarrAwal[2],$SarrAwal[0]));
$numberofday_akhir=date("t", mktime(0,0,0,$SarrAkhir[1],$SarrAkhir[2],$SarrAkhir[0]));
;
SarrDateList=array();
for ($idx=16;$idx<=$numberofday_awal;$idx++) {
    $strDate=$SarrAwal[0].'-'. $SarrAwal[1].'-'. sprintf("%02d", $idx);
}

```

```

        $idx_day=date("w",mktime(0,0,0,$arrAwal[1],$idx,$arrAwal[0]));

        if (($idx_day!=0)and($idx_day!=6))
$arrDateList[$strDate]=$arrHari[$idx_day];
    }

for ($idx=1;$idx<=15;$idx++) {

        $strDate=$arrAkhir[0].'-'.$arrAkhir[1].'-'.sprintf("%02d",$idx);

        $idx_day=date("w",mktime(0,0,0,$arrAkhir[1],$idx,$arrAkhir[0]));

if (($idx_day!=0)and($idx_day!=6)) $arrDateList[$strDate]=$arrHari[$idx_day];
    }

$dbq=mysql_query($sql);

while ($data=mysql_fetch_assoc($dbq)) {

$fname=$data['company_name'].'-'.$data['div_name'].'-'.$data['user_nik'].'-
'.$data['user_name'].".html";

        $strJudul=$data['user_nik'].'-'.$data['user_realname'];

        $strNama=$data['user_realname'];

        $strUser=$data['user_name'];

        $strEmail=$data['user_email'];

        $strCompany=$data['company_name'];

        $strDept=$data['div_name'];

        $awalKerja=$data['m_startdate'];

        $akhirKerja=$data['m_enddate'];

        $nik=$data['user_nik'];

        ob_start();

        echo <<< ENDDATA

```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>$strJudul</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<br>
Dear <strong>$strNama</strong>,
<br>
This is an automated email message to inform you about your daily attendance
between $periode_awal to $periode_akhir.<br>
Should you have any comment or note,Please log-in to https://intranet.skripsi.web.id
with your NIK and PIN. <br><br><br>
Thank you for your kind attention.<br><br>
Regards,<br>
<em>Human Resources Information System (HRIS)</em><br>
<strong>HRD</strong><br><br>
<center>
<table width="44%" border="0">
<tr>
<td width="18%"><div align="left"><strong>NIK </strong></div></td>
<td width="49%"><strong>: $nik</strong></td>
</tr>
```

```

<tr>
  <td><strong>Employee</strong></td>
  <td><strong>: $strNama</strong></td>
</tr>
<tr>
  <td><strong>Company</strong></td>
  <td><strong>: $strCompany</strong></td>
</tr>
<tr>
  <td><strong>Department</strong></td>
  <td><strong>: $strDept</strong></td>
</tr>
</table>
</center>
ENDDATA;
      echo <<< ENDDATA
<table width="97%" border="0" cellpadding="0" cellspacing="2"
bordercolor="#CCCCCC" bgcolor="#666666" style="border:1px; color:black">
  <tr bgcolor="#999999">
    <td width="9%" bgcolor="#CCCCCC"><div align="center">Date</div></td>
    <td width="10%" bgcolor="#CCCCCC"><div align="center">Tanggal</div></td>
    <td width="7%" bgcolor="#CCCCCC"><div align="center">Masuk</div></td>
    <td width="7%" bgcolor="#CCCCCC"><div align="center">Keluar</div></td>
    <td width="8%" bgcolor="#CCCCCC"><div align="center">Telat</div></td>

```

```

<td width="10%" bgcolor="#CCCCCC"><div
align="center">Overtime</div></td>

<td width="49%" bgcolor="#CCCCCC"><div
align="center">Keterangan</div></td>

</tr>

ENDDATA;

$sqlAbsensi="select * from absensi_2008 where absensi_nik='".$data['user_nik']."'
and "."absensi_date>='$periode_awal' and
absensi_date<='$periode_akhir'";$dbqAbsensi=mysql_query($sqlAbsensi);
$intTotalTelat=0; $arrHariKerja=array();

$intHariKerjaYgAbsen=0; $intHariKerjaYgLengkap=0; while
($dataAbsen=mysql_fetch_assoc($dbqAbsensi)) {

$tglKerja=$dataAbsen[absensi_date];

$arrTglKerja=explode('-', $tglKerja);

$serialTglKerja=mktime(0,0,0,$arrTglKerja[1],$arrTglKerja[2],$arrTglKerja[0]);$hariKerja=date("w", $serialTglKerja);

$intTelat=$dataAbsen['absensi_minute_telat'];

$intLembur=$dataAbsen['absensi_minute_lembur'];

$intTotalTelat+=$intTelat;

$absensi_note=$dataAbsen['absensi_note'];

$arrHariKerja[$dataAbsen['absensi_date']]=$arrHari[$hariKerja];

if ((isset($dataAbsen['absensi_masuk']))or(isset($dataAbsen['absensi_keluar']))) {

        $intHariKerjaYgAbsen++;

    }

    if
((isset($dataAbsen['absensi_masuk']))and(isset($dataAbsen['absensi_keluar']))) {

        $intHariKerjaYgLengkap++;
    }
}

```

```

    }

    if (isset($dataAbsen['absensi_masuk'])) {

        $serialMasuk=$dataAbsen['absensi_masuk'];

        ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $serialMasuk, $sarrMasuk);

        $jamMasuk=$sarrMasuk[4].":".$sarrMasuk[5];

    } else $jamMasuk='-';

    if (isset($dataAbsen['absensi_keluar'])) {

        $serialKeluar=$dataAbsen['absensi_keluar']; ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", $serialKeluar, $sarrKeluar);

        $jamKeluar=$sarrKeluar[4].":".$sarrKeluar[5];

    } else $jamKeluar='-';

    echo <<< ENDDATA

<tr bgcolor="#00FFFF">

    <td valign="top"><div align="center">$sarrHari[$hariKerja]&nbsp;  </div></td>

    <td valign="top"><div align="center">$tglKerja&nbsp;  </div></td>

    <td valign="top"><div align="center">$jamMasuk&nbsp;  </div></td>

    <td valign="top"><div align="center">$jamKeluar&nbsp;  </div></td>

    <td valign="top"><div align="center">$sintTelat&nbsp;  </div></td>

    <td valign="top"><div align="center">$sintLembur&nbsp;  </div></td>

    <td valign="top"><div align="left">$absensi_note&nbsp;  </div></td>

</tr>

ENDDATA;

    }

    echo <<< ENDDATA

```

```

<tr>
  <td valign="top" bgcolor="#CCCCCC" colspan="4">&nbsp;&nbsp;&nbsp;&nbsp;</td>
  <td valign="top" bgcolor="#CCCCCC"><div
align="center">${intTotalTelat}&nbsp;&nbsp;&nbsp;</div></td>
  <td valign="top" bgcolor="#CCCCCC" colspan="2">&nbsp;&nbsp;&nbsp;&nbsp;</td>
</tr>
ENDDATA;
    echo <<< ENDDATA
</table><br><br><br>
<strong>Summary</strong><br><br> ENDDATA;
echo "Hari kerja yang tercatat mesin : ".$intHariKerjaYgAbsen."<br />\r\n";
echo "Absensi sempurna : ".$intHariKerjaYgLengkap."<br />\r\n";
$absensi_bolong=array_diff_key($arrDateList,$arrHariKerja);
    if (is_array($absensi_bolong)) {
        echo "<br /> Berikut ini tanggal kerja senin-jum'at yang tidak ada
keterangan :<br />\r\n";
        while(list($tgl_bolong,$hari_bolong)=each($absensi_bolong)) {
            echo "&middot; $hari_bolong $tgl_bolong<br />\r\n";
        }
    }
    echo <<< ENDDATA
</body>
</html>
ENDDATA;
    $strHasil=ob_get_contents();

```



```
ob_get_clean();

$fdata=@fopen($pathdir.'/'.$fname,"w+");

@fwrite($fdata,$strHasil);

@fclose($fdata);

$mail = new htmlMimeMail5();

$mail->setFrom("Sistem Absensi Skripsi <no-replay@skripsi.web.id>");

$mail->setSubject("Notification : Daily Attendance $strNama between
".$periode_awal." to ".$periode_akhir);

$mail->setHeader('Date', date('D, d M y H:i:s O'));

$mail->setHtml($strHasil,$strHasil);

if (!in_array($data['user_nik'],$arrExcludeNIK))
$mail->send(array("bagus@skripsi.web.id"),"smtp");
}

mysql_close();

?>
```