

**PENGEMBANGAN SISTEM INFORMASI PENGAJIAN
(STUDI KASUS: SMK AL-FIRDAUS JAKARTA)**



Oleh:

SIGIT PRASETIYO

104093003010

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH
JAKARTA
2011 M / 1432 H**

PENGEMBANGAN SISTEM INFORMASI PENGGAJIAN

(STUDI KASUS: SMK AL-FIRDAUS JAKARTA)

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar

Sarjana Komputer

Pada Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

Oleh:

SIGIT PRASETIYO

104093003010

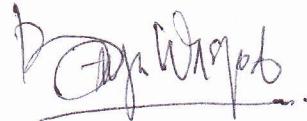
Menyetujui,

Pembimbing I


Nur Aeni Hidayah, MMSI

NIP. 19750818 200501 2 008

Pembimbing II


Bayu Waspodo, MM

NIP. 19740812 200801 1 001

Mengetahui,

Ketua Program Studi Sistem Informasi


Nur Aeni Hidayah, MMSI

NIP. 19750818 200501 2 008

PENGESAHAN UJIAN

Skripsi berjudul "Pengembangan Sistem Informasi Penggajian (Studi Kasus: SMK Al-Firdaus Jakarta)" telah diujikan dan dinyatakan lulus dalam sidang munaqosyah Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta pada hari Jumat, 29 Juli 2011. Skripsi ini telah diterima sebagai salah satu syarat memperoleh gelar Sarjana Strata Satu (S1) Program Studi Sistem Informasi.

Menyetujui,

Penguji I

Nia Kumaladewi, MMSI

NIP. 19750412 200710 2 002

Penguji II

Dr. Syopiansyah Jaya Putra, M.Sis

NIP. 19680117 200112 1 001

Pembimbing I

Nur Aeni Hidayah, MMSI

NIP. 19750818 200501 2 008

Pembimbing II

Bayu Waspodo, MM

NIP. 19740812 200801 1 001

Mengetahui,

Dekan

Fakultas Sains dan Teknologi



Dr. Syopiansyah Jaya Putra, M.Sis

NIP. 19680117 200112 1 001

Ketua

Program Studi Sistem Informasi

Nur Aeni Hidayah, MMSI

NIP. 19750818 200501 2 008

PERNYATAAN

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSIINI BENAR-BENAR HASIL KARYA SENDIRI YANG BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN.

Jakarta, Juli 2011

Sigit Prasetyo

104093003010

ABSTRAK

SIGIT PRASETIYO (104093003010). Pengembangan Sistem Informasi Penggajian (Studi Kasus: SMK Al-Firdaus Jakarta) di bawah bimbingan **NUR AENI HIDAYAH dan BAYU WASPODO**.

SMK Al-Firdaus merupakan sebuah lembaga pendidikan yang bernaung dalam Yayasan Pendidikan Islam Al-Firdaus. Sistem penggajian yang berjalan saat ini telah sesuai prosedur yang berlaku namun prosesnya masih dilakukan secara *manual* yang masih dilakukan pencatatan dalam dokumen. Dengan masih *manual*-nya sistem yang ada maka hal-hal yang menyangkut proses absensi, penggajian, dan laporan-laporan yang dihasilkan hanya disimpan dalam dokumen-dokumen yang dapat memungkinkan kehilangannya data-data. Menurut staf bendahara, proses pengolahan penggajian yang dimulai pada tanggal 1 di bulan berikutnya, pelaksanaannya memerlukan waktu 8 hari yang dirasakan sangat lama dan memerlukan ketelitian. Pembayaran gaji dilaksanakan pada tanggal 10. Menurut para pegawai, waktu pembayaran tersebut dirasakan terlalu lama. Sistem penggajian yang penulis ajukan adalah sistem informasi penggajian berbasis *web* yang dibuat untuk mengatasi dan meminimalisasi masalah di atas. Secara garis besar sistem yang dikembangkan terdiri dari proses pengelolaan data master, absensi, penggajian yang disertai dengan laporan-laporan yang berkaitan dengan penggajian. Metodologi penelitian yang penulis lakukan antara lain metode pengumpulan data berupa studi pustaka, studi lapangan dengan observasi dan wawancara, studi literatur sejenis antara lain dari Mulyadi (2008). Sedangkan metode pengembangan sistem menggunakan *Rapid Application Development* (RAD) yang dimodelkan dengan diagram-diagram *Unified Modelling Language* (UML). Tahapan yang dilakukan dalam RAD adalah *Scope Definition, Analysis, Design, Construction and Testing*. Tools yang digunakan adalah XAMPP 1.7.1 dengan spesifikasi Apache 2.2.11 sebagai *web server*, PHP versi 5.2.9 sebagai bahasa pemrograman dan MySQL versi 5.1.33 sebagai *database*. Dengan diterapkannya sistem ini mampu mempersingkat waktu pengolahan data penggajian, mengatasi masalah *human error* penghitungan absensi dan gaji. Sistem ini juga menghasilkan laporan-laporan penggajian dengan cepat dan akurat yang diharapkan pembayaran gaji dapat dipercepat, dan pegawai dapat memperoleh informasi penghasilannya melalui slip gaji.

Kata Kunci: Sistem Informasi Penggajian, *Rapid Application Development* (RAD), *Unified Modelling Language* (UML).

V Bab + xix Halaman + 135 Halaman + 28 Daftar Pustaka + 51 Gambar + 37

Tabel + 4 Daftar Simbol + 5 Lampiran

Pustaka Acuan: 28 (2000-2008)

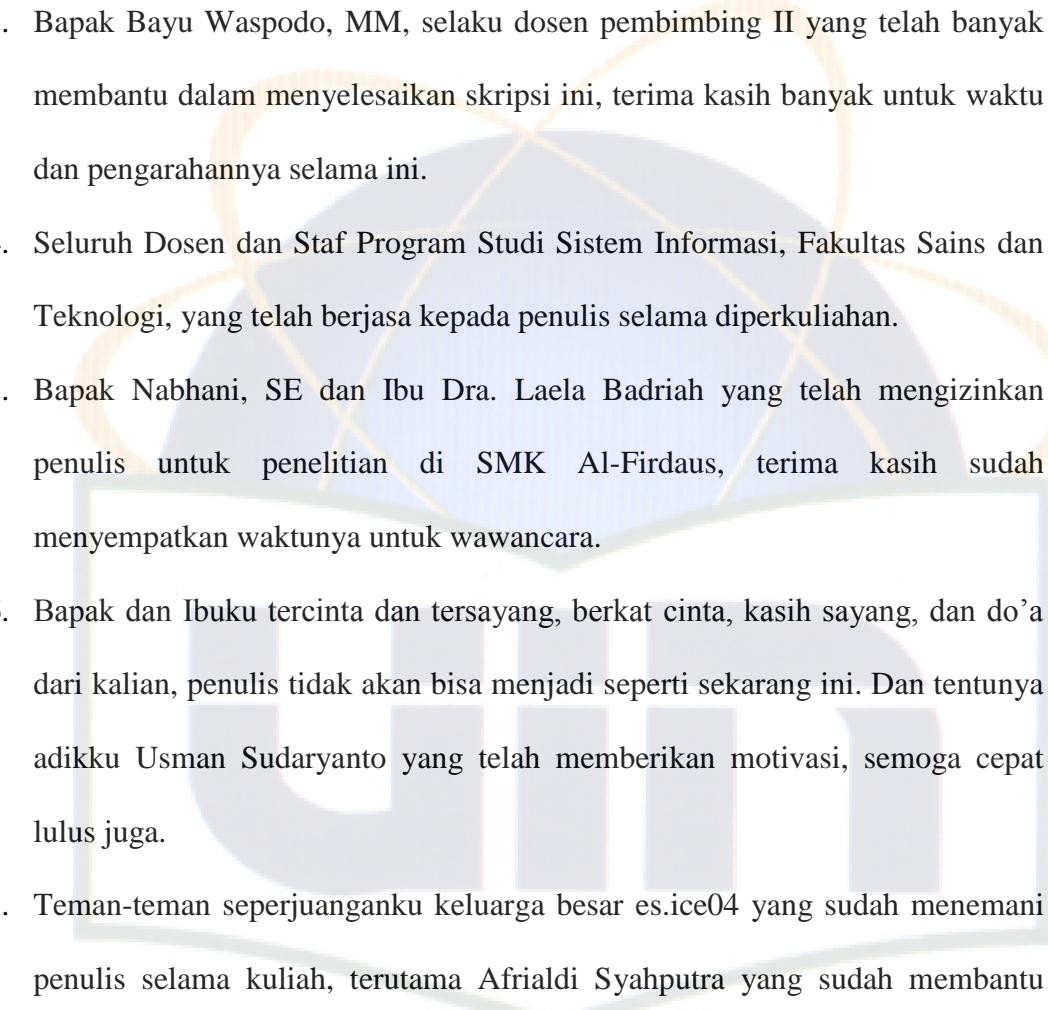
KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kepada Allah SWT atas segala rahmat dan karunia-Nya yang diberikan kepada penulis dalam kehidupan di dunia ini sehingga penulis dapat melaksanakan dan menyelesaikan skripsi dengan lancar. Shalawat serta salam penulis curahkan kepada junjungan, suri tauladan, dan Rasul kita baginda Nabi besar Muhammad SAW yang telah menyampaikan ajaran Islam ke muka bumi ini, dan juga kepada keluarga dan para sahabatnya beserta alim ulama yang senantiasa meneruskan ajaran Islam sebagai penerang jalan kehidupan hingga akhir zaman.

Skripsi ini berjudul “Pengembangan Sistem Informasi Penggajian (Studi Kasus: SMK Al-Firdaus Jakarta)”, yang disusun sebagai salah satu syarat dalam menyelesaikan program Strata Satu (S1) pada Program Studi Sistem Informasi di Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah mendukung atas terselesaiannya laporan ini. Karena tanpa dukungan dari mereka, penulis tidak akan mampu menyelesaikan laporan ini dengan baik. Mereka yang telah mendukung penulis adalah:

1. Bapak Dr. Syopiansyah Jaya Putra, M.Sis, selaku Dekan Fakultas Sains dan Teknologi.
2. Ibu Nur Aeni Hidayah, MMSI, selaku Ketua Program Studi Sistem Informasi sekaligus sebagai dosen pembimbing I yang telah banyak memberikan masukan, bimbingan dan motivasi dalam menyelesaikan skripsi ini.

- 
3. Bapak Bayu Waspodo, MM, selaku dosen pembimbing II yang telah banyak membantu dalam menyelesaikan skripsi ini, terima kasih banyak untuk waktu dan pengarahananya selama ini.
 4. Seluruh Dosen dan Staf Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, yang telah berjasa kepada penulis selama diperkuliahannya.
 5. Bapak Nabhani, SE dan Ibu Dra. Laela Badriah yang telah mengizinkan penulis untuk penelitian di SMK Al-Firdaus, terima kasih sudah menyempatkan waktunya untuk wawancara.
 6. Bapak dan Ibuku tercinta dan tersayang, berkat cinta, kasih sayang, dan do'a dari kalian, penulis tidak akan bisa menjadi seperti sekarang ini. Dan tentunya adikku Usman Sudaryanto yang telah memberikan motivasi, semoga cepat lulus juga.
 7. Teman-teman seperjuanganku keluarga besar es.ice04 yang sudah menemani penulis selama kuliah, terutama Afrialdi Syahputra yang sudah membantu dalam pembuatan aplikasi. Toni, Sahrul, Fahrurrozi, Qoyyimah yang berjuang bersama dalam penyelesaian skripsi. Terima kasih untuk semua nya!

Akhir kata penulis menyadari bahwa dalam penulisan laporan ini masih banyak kesalahan-kesalahan dan kekurangan-kekurangan sehingga kritik membangun sangatlah diperlukan. Semoga skripsi ini dapat bermanfaat baik untuk diri penulis maupun untuk kepentingan universitas.

Jakarta, Juli 2011

Sigit Prasetyo

DAFTAR ISI

Halaman Judul.....	i
Lembar Persetujuan.....	ii
Lembar Pengesahan.....	iii
Lembar Pernyataan.....	iv
Abstrak.....	v
Kata Pengantar.....	vi
Daftar Isi.....	viii
Daftar Gambar.....	xii
Daftar Tabel.....	xiv
Daftar Simbol.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	5
1.6 Metodologi Penelitian.....	6
1.7 Sistematika Penulisan.....	8
BAB II LANDASAN TEORI.....	10
2.1 Konsep Sistem Informasi.....	10
2.1.1 Sistem.....	10
2.1.2 Informasi.....	11
2.1.3 Sistem Informasi.....	13
2.2 Konsep Penggajian.....	15
2.2.1 Gaji atau Upah.....	15
2.2.2 Gaji Pokok.....	15
2.2.3 Jabatan.....	15
2.2.4 Tunjangan.....	16
2.2.5 Sistem Pengupahan atau Penggajian.....	16
2.2.6 Fungsi yang Terkait.....	17

2.2.7 Dokumen yang Digunakan.....	18
2.2.8 Diagram Alir (<i>Flowmap</i>) Sistem Penggajian.....	20
2.3 Metodologi Penelitian.....	21
2.4 Metode Pengumpulan Data.....	23
2.5 Metode Pengembangan Sistem.....	24
2.6 <i>Rapid Application Development (RAD)</i>	25
2.7 <i>Unified Modelling Languange (UML)</i>	27
2.7.1 Definisi UML.....	27
2.7.2 Diagram UML.....	28
2.7.2.1 <i>Use Case Diagram</i>	28
2.7.2.2 <i>Activity Diagram</i>	30
2.7.2.3 <i>Sequence Diagram</i>	30
2.7.2.4 <i>Class Diagram</i>	31
2.7.3 Keunggulan UML.....	31
2.8 Konsep <i>Database</i> (Basis Data).....	32
2.8.1 <i>Database</i> (Basis Data).....	32
2.8.2 <i>Database Management Sistem (DBMS)</i>	34
2.9 Konsep <i>Internet</i>	36
2.9.1 Definisi <i>Internet</i>	36
2.9.2 Sumber Daya <i>Internet</i>	36
2.9.3 <i>HyperText Transfer Protocol (HTTP)</i>	37
2.9.4 Aplikasi <i>Web</i>	38
2.10 Tools Pengembangan Sistem.....	38
2.10.1 <i>Rich Picture</i>	38
2.10.2 Adobe Photoshop.....	39
2.10.3 Macromedia Dreamweaver.....	39
2.10.4 XAMPP.....	40
2.10.5 Apache.....	40
2.10.6 PHP.....	41
2.10.7 MySQL.....	42
2.10.8 <i>Web Browser</i>	42
2.11 Pengujian Perangkat Lunak.....	43

2.12 Literatur Penelitian Sejenis.....	44
BAB III METODOLOGI PENELITIAN.....	48
3.1 Metode Pengumpulan Data.....	48
3.1.1 Studi Pustaka.....	48
3.1.2 Studi Lapangan.....	49
3.1.3 Studi Literatur Sejenis.....	50
3.2 Metodologi Pengembangan Sistem.....	50
3.2.1 Mendefinisikan Lingkup (<i>Scope Definition</i>).....	51
3.2.2 Analisis Sistem (<i>Analysis</i>).....	51
3.2.3 Perancangan Sistem (<i>Design</i>).....	52
3.2.4 Implementasi Sistem (<i>Construction & Testing</i>).....	53
3.3 Kerangka Penelitian.....	54
BAB IV PENGEMBANGAN SISTEM INFORMASI PENGAJIAN.....	56
4.1 Profil Sekolah.....	56
4.1.1 Gambaran Umum.....	56
4.1.2 Logo Sekolah.....	56
4.1.3 Visi dan Misi.....	57
4.1.4 Struktur Organisasi.....	57
4.2 Mendefinisikan Lingkup (<i>Scope Definition</i>).....	58
4.3 Analisis Sistem (<i>Analysis</i>).....	58
4.3.1 Analisa Permasalahan (<i>Problem Analysis</i>).....	58
4.3.1.1 Analisa Sistem Berjalan.....	58
4.3.1.2 Identifikasi Masalah.....	62
4.3.1.3 Sistem Usulan	62
4.3.1.4 Analisa Perbandingan Sistem.....	64
4.3.2 Analisa Persyaratan (<i>Requirement Analysis</i>).....	65
4.3.2.1 <i>Functional Requirement</i>	65
4.3.2.2 <i>Nonfunctional Requirement</i>	66
4.3.3 Analisa Keputusan (<i>Decision Analysis</i>).....	68
4.4 Perancangan Sistem (<i>Design</i>).....	69
4.4.1 <i>Use Case Model Diagram</i>	69
4.4.1.1 Identifikasi Actor.....	69

4.4.1.2 Identifikasi <i>Use Case</i>	70
4.4.1.3 Perancangan <i>Use Case Diagram</i>	72
4.4.1.4 <i>Use Case Narrative</i>	72
4.4.2 <i>Activity Diagram</i>	83
4.4.3 <i>Sequence Diagram</i>	96
4.4.4 <i>Class Diagram</i>	107
4.4.5 Tabel <i>Database</i>	111
4.4.6 Perancangan <i>Interface</i>	116
4.5 Implementasi Sistem (<i>Construction & Testing</i>).....	126
4.5.1 Pemrograman.....	126
4.5.2 Perangkat Sistem.....	126
4.5.4 Pengujian Perangkat Lunak.....	128
BAB V PENUTUP	131
5.1 Simpulan.....	131
5.2 Saran.....	131
DAFTAR PUSTAKA	133
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Siklus Informasi.....	11
Gambar 2.2 Diagram Alir Sistem Penggajian.....	21
Gambar 2.3 Strategi <i>Rapid Application Development</i>	26
Gambar 3.1 Kerangka Penelitian.....	55
Gambar 4.1 Logo Sekolah.....	56
Gambar 4.2 Struktur Organisasi.....	57
Gambar 4.3 <i>Rich Picture</i> Sistem Berjalan.....	61
Gambar 4.4 <i>Rich Picture</i> Sistem Usulan.....	63
Gambar 4.5 <i>Use Case Diagram</i>	72
Gambar 4.6 <i>Activity Diagram Login</i>	83
Gambar 4.7 <i>Activity Diagram Input Data User</i>	84
Gambar 4.8 <i>Activity Diagram Input Data Jabatan</i>	85
Gambar 4.9 <i>Activity Diagram Input Data Jurusan</i>	86
Gambar 4.10 <i>Activity Diagram Input Data Honor</i>	87
Gambar 4.11 <i>Activity Diagram Input Data Pegawai</i>	88
Gambar 4.12 <i>Activity Diagram Melihat Data Pegawai</i>	89
Gambar 4.13 <i>Activity Diagram Input Data Tahun Pelajaran</i>	89
Gambar 4.14 <i>Activity Diagram Input Data Waktu Mengajar</i>	90
Gambar 4.15 <i>Activity Diagram Input Absensi</i>	91
Gambar 4.16 <i>Activity Diagram Melihat Rekap Absensi</i>	92
Gambar 4.17 <i>Activity Diagram Melihat Laporan Penggajian</i>	93
Gambar 4.18 <i>Activity Diagram Mencetak Laporan Penggajian</i>	94
Gambar 4.19 <i>Activity Diagram Mencetak Slip Gaji</i>	95
Gambar 4.20 <i>Sequence Diagram Login</i>	96
Gambar 4.21 <i>Sequence Diagram Input Data User</i>	97
Gambar 4.22 <i>Sequence Diagram Input Data Jabatan</i>	97
Gambar 4.23 <i>Sequence Diagram Input Data Jurusan</i>	98
Gambar 4.24 <i>Sequence Diagram Input Data Honor</i>	99
Gambar 4.25 <i>Sequence Diagram Input Data Pegawai</i>	100
Gambar 4.26 <i>Sequence Diagram Melihat Data Pegawai</i>	101

Gambar 4.27 <i>Sequence Diagram Input</i> Tahun Pelajaran.....	101
Gambar 4.28 <i>Sequence Diagram Input</i> Waktu Mengajar.....	102
Gambar 4.29 <i>Sequence Diagram Input</i> Absensi.....	103
Gambar 4.30 <i>Sequence Diagram</i> Melihat Rekap Absensi.....	104
Gambar 4.31 <i>Sequence Diagram</i> Melihat Laporan Penggajian.....	104
Gambar 4.32 <i>Sequence Diagram</i> Mencetak Laporan Penggajian.....	105
Gambar 4.33 <i>Sequence Diagram</i> Mencetak Slip Gaji.....	106
Gambar 4.34 <i>Class Diagram</i>	110
Gambar 4.35 Rancangan Menu <i>Login</i>	118
Gambar 4.36 Rancangan Halaman Utama Level Kepala Tata Usaha.....	119
Gambar 4.37 Rancangan Halaman Utama Level Guru Piket.....	119
Gambar 4.38 Rancangan Halaman Utama Level Bendahara.....	120
Gambar 4.39 Rancangan Halaman Utama Level Ketua Yayasan.....	121
Gambar 4.40 Rancangan Halaman Utama Level Pegawai.....	121
Gambar 4.41 Rancangan Halaman <i>Input</i> Data Pegawai.....	122
Gambar 4.42 Rancangan Halaman Lihat Data Pegawai.....	123
Gambar 4.43 Rancangan Halaman <i>Input</i> Absensi.....	123
Gambar 4.44 Rancangan Halaman Rekap Absensi Bulanan.....	124
Gambar 4.45 Rancangan Halaman Laporan Penggajian.....	125
Gambar 4.46 Rancangan Halaman Slip Gaji.....	125
Gambar 4.47 Konfigurasi Jaringan.....	128

DAFTAR TABEL

Tabel 2.1 Sumber Daya Pada <i>Internet</i>	36
Tabel 2.2 Literatur Penelitian Sejenis.....	46
Tabel 4.1 Analisa Perbandingan Sistem.....	64
Tabel 4.2 <i>Nonfunctional Requirement</i>	66
Tabel 4.3 Identifikasi <i>Actor</i>	69
Tabel 4.4 Identifikasi <i>Use Case</i>	70
Tabel 4.5 <i>Use Case Narrative Login</i>	73
Tabel 4.6 <i>Use Case Narrative Input Data User</i>	73
Tabel 4.7 <i>Use Case Narrative Input Data Jabatan</i>	74
Tabel 4.8 <i>Use Case Narrative Input Data Jurusan</i>	75
Tabel 4.9 <i>Use Case Narrative Input Data Honor</i>	76
Tabel 4.10 <i>Use Case Narrative Input Data Pegawai</i>	77
Tabel 4.11 <i>Use Case Narrative Melihat Data Pegawai</i>	78
Tabel 4.12 <i>Use Case Narrative Input Data Tahun Pelajaran</i>	78
Tabel 4.13 <i>Use Case Narrative Input Data Waktu Mengajar</i>	79
Tabel 4.14 <i>Use Case Narrative Input Absensi</i>	80
Tabel 4.15 <i>Use Case Narrative Melihat Rekap Absensi</i>	81
Tabel 4.16 <i>Use Case Narrative Melihat Laporan Penggajian</i>	81
Tabel 4.17 <i>Use Case Narrative Mencetak Laporan Penggajian</i>	82
Tabel 4.18 <i>Use Case Narrative Mencetak Slip Gaji</i>	82
Tabel 4.19 Daftar <i>Potential Object</i>	107
Tabel 4.20 Daftar Analisa <i>Potential Object</i>	108
Tabel 4.21 Daftar <i>Object</i>	109
Tabel 4.22 Tabel <i>User</i>	111
Tabel 4.23 Tabel <i>Jabatan</i>	111
Tabel 4.24 Tabel <i>Jurusan</i>	112
Tabel 4.25 Tabel <i>Honor</i>	112
Tabel 4.26 Tabel <i>Pegawai</i>	113
Tabel 4.27 Tabel <i>Tahun Pelajaran</i>	114
Tabel 4.28 Tabel <i>Waktu Mengajar</i>	114

Tabel 4.29 Tabel Piket.....	115
Tabel 4.30 Tabel Absensi.....	115
Tabel 4.31 Tabel Penggajian.....	116
Tabel 4.32 <i>Feature System</i>	116
Tabel 4.33 Daftar <i>Tools</i> Pengembangan Sistem.....	126
Tabel 4.34 Spesifikasi <i>Hardware</i>	127
Tabel 4.35 Pengujian Sistem.....	129



DAFTAR SIMBOL

SIMBOL USE-CASE DIAGRAM

Simbol	Keterangan
	<i>Actor</i>
	<i>Use case</i>
	<i>Association</i>
	<i>Extends</i>
	<i>Uses (includes)</i>
	<i>Inheritance</i>
	<i>System Boundary</i>

(Sumber: Whitten, 2004)

SIMBOL ACTIVITY DIAGRAM

Simbol	Keterangan
	<i>State</i>
	<i>Control Flow</i>
	<i>Initial State</i>
	<i>Final State</i>
	<i>Transition</i>
	<i>Decision</i>
	<i>Swimlane</i>

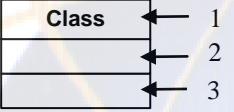
(Sumber: Whitten, 2004)

SIMBOL SEQUENCE DIAGRAM

Simbol	Keterangan
	<i>Object</i>
	<i>Actor</i>
	<i>Lifeline</i>
	<i>Message</i>
	<i>Message (return)</i>
	<i>Message (call)</i>
	<i>Activation</i>

(Sumber: Whitten, 2004)

SIMBOL CLASS DIAGRAM

Simbol	Keterangan
	<i>Class</i> 1. <i>class name</i> 2. <i>attributes</i> 3. <i>behaviors</i>
	<i>Association</i>
	<i>Aggregation</i> <i>(composite)</i>
	<i>Aggregation</i> <i>(shared)</i>
	<i>Generalization</i>

(Sumber: Whitten, 2004)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi Informasi (TI) merupakan salah satu alat bantu penunjang pekerjaan di sebuah organisasi baik profit maupun non profit. Salah satunya adalah sebagai sarana meningkatkan produktivitas kerja para karyawan, sehingga tujuan yang akan dicapai organisasi tersebut dapat tercapai secara optimal. Hal ini merupakan gambaran bahwa pesatnya perkembangan teknologi saat ini, khususnya TI yang telah banyak mempengaruhi cara orang dalam berbisnis. Bukti untuk ini dapat dilihat dengan semakin banyaknya penggunaan komputer yang digunakan untuk menyimpan dan mengolah data sehingga menjadi suatu informasi yang dimanfaatkan oleh berbagai kalangan yang membutuhkannya. Begitu juga dalam proses penggajian pegawai, karena gaji pegawai dapat menentukan kinerja dari pegawai dan perkembangan perusahaan itu sendiri, upah atau gaji adalah hak pekerja/buruh yang diterima dan dinyatakan dalam bentuk uang sebagai imbalan dari pengusaha atas pemberi kerja kepada pekerja/buruh yang ditetapkan yang dibayarkan menurut suatu perjanjian kerja (Adisu, 2008).

SMK Al-Firdaus merupakan sebuah lembaga pendidikan yang bernaung dalam Yayasan Pendidikan Islam Al-Firdaus. SMK Al-Firdaus beralamat di Jalan Peta Utara No. 41 Pegadungan Kalideres Jakarta Barat. Saat ini SMK Al-Firdaus memiliki keseluruhan pegawai 40 orang yang

menyelenggarakan pendidikan dengan program keahlian (jurusan): Administrasi Perkantoran, Akuntansi, dan Pemasaran.

Berdasarkan hasil penelitian yang penulis lakukan, administrasi pengolahan penggajian yang berjalan saat ini pada SMK Al-Firdaus telah sesuai prosedur yang berlaku namun prosesnya masih dilakukan secara *manual*. Diawali dari pencatatan absensi mengajar yang dicatat pada daftar absensi yang masih dalam bentuk kertas, yang kemungkinan dokumennya dapat hilang atau rusak sebelum dibuat rekap absensi. Setiap pada tanggal 1 di bulan berikutnya, bendahara mulai membuat rekap absensi yang dihitung secara *manual* dari daftar absensi dan kemudian memindahkannya ke *Microsoft Excel*. Pada tanggal 4, bendahara harus melengkapi rekap absensi. Penghitungan gaji dimulai setelah pembuatan rekap absensi selesai, yang masih dihitung secara *manual* kemudian memindahkannya ke *Microsoft Excel* untuk dibuatkan laporan penggajian.

Pada tanggal 8, bendahara harus selesai membuat laporan penggajian. Kemudian rekap absensi dan laporan penggajian dicetak dan diserahkan ke ketua yayasan untuk diperiksa dan mendapatkan persetujuan. Setelah disetujui, dilakukanlah pembuatan slip gaji dengan mencatat kembali dari laporan penggajian menggunakan *Microsoft Excel* kemudian dicetak pada amplop. Pembayaran gaji dilakukan setiap tanggal 10, pegawai menerima uang gaji yang berada di dalam amplop. Menurut para pegawai, pembayaran gaji pada tanggal 10 dirasakan terlalu lama.

Proses administrasi penggajian tersebut dirasakan membutuhkan ketelitian dan waktu yang lama dalam pelaksanaannya sehingga pembayaran gaji pun menjadi lama. Masalah semakin besar jika pegawai makin bertambah, tentunya hal tersebut tidak dapat dibiarkan berlarut-larut. Maka untuk mengatasi hal tersebut diperlukan adanya sistem penggajian yang terkomputerisasi dan terintegrasi antara aktivitas absensi dan penggajian sehingga menghasilkan informasi lebih mudah dan cepat daripada menggunakan sistem *manual* yang sedang diterapkan. Sistem informasi yang dapat membantu kinerja guru piket dalam mengisi absensi, membantu bendahara untuk menghindari keterlambatan dan kesalahan dalam pembuatan rekap absensi, penghitungan gaji serta mempermudah pembuatan laporan penggajian dan slip gaji.

Berdasarkan latar belakang permasalahan di atas, penulis tertarik untuk melakukan penelitian yang berkaitan dengan sistem informasi penggajian pada SMK Al-Firdaus. Serta mencoba memberikan solusi alternatif dengan memberikan usulan rancangan sistem informasi penggajian yang penulis susun dalam tugas akhir yang berjudul **“Pengembangan Sistem Informasi Penggajian (Studi Kasus: SMK Al-Firdaus Jakarta)”**.

1.2 Perumusan Masalah

Adapun identifikasi masalah berdasarkan latar belakang yang telah diuraikan sebagai berikut:

1. Adanya keterlambatan dalam proses pengolahan data penggajian.

2. Adanya masalah *human error* dalam penghitungan absensi dan gaji, sehingga akurasi data masih rendah.
3. Pembayaran gaji yang dirasakan terlalu lama.

Berdasarkan identifikasi di atas, maka permasalahan yang akan dibahas dalam penelitian ini adalah “Bagaimana mengembangkan sistem informasi penggajian yang dapat memangkas waktu pengolahan data penggajian, mengurangi masalah *human error* dalam penghitungan absensi dan gaji, dan mengefisiensikan kinerja staf Kepala Tata Usaha dan Bendahara?”.

1.3 Batasan Masalah

Agar masalah tidak terlalu luas namun dapat mencapai hasil yang optimal. Maka penulis akan membatasi ruang lingkup pembahasan sebagai berikut:

1. Pengembangan sistem informasi penggajian yang digunakan oleh pihak internal yaitu Kepala Tata Usaha, Guru Piket, Bendahara, Ketua Yayasan, serta Pegawai.
2. Secara garis besar sistem yang dikembangkan terdiri dari pendataan pegawai, pengisian absensi mengajar, pembuatan rekap absensi mengajar, penghitungan gaji, pembuatan laporan penggajian, dan pembuatan slip gaji.
3. Sistem informasi yang dikembangkan hanya bertanggung jawab terhadap administrasi penggajian, tidak termasuk fungsi keuangan lainnya.

4. Tidak membahas perhitungan pajak penghasilan, karena jumlah penghasilan per bulan pegawainya belum mencapai ketentuan perpajakan yang berlaku.
5. Pembayaran gaji tidak ditransfer melalui bank, namun pembayaran gaji dilakukan secara tunai.
6. Peneliti menggunakan metode pengembangan sistem *Rapid Application Development* (RAD) menurut Whitten 2004 dari tahap *Scope Definition* sampai dengan tahapan *Construction & Testing*.
7. Sistem informasi dirancang dengan menggunakan bahasa pemrograman PHP dan *database MySQL* yang berjalan pada *web server Apache*.

1.4 Tujuan Penelitian

Tujuan dari pelaksanaan penelitian ini adalah:

1. Mempermudah dan mempercepat pengolahan data penggajian.
2. Mengurangi masalah *human error* dalam penghitungan absensi dan gaji, sehingga meningkatkan akurasi data.
3. Mengefisienkan kinerja staf Kepala Tata Usaha dan Bendahara.
4. Memberikan kemudahan bagi pegawai dalam melihat informasi penghasilannya.

1.5 Manfaat Penelitian

Manfaat pembuatan aplikasi dan sistem informasi penggajian ini adalah:

1. Menggunakannya sebagai bahan evaluasi terhadap sistem yang sedang berjalan.
2. Menghasilkan suatu informasi yang cepat dan akurat dalam membantu dan memudahkan proses administrasi penggajian.

Manfaat pembuatan laporan penelitian ini bagi para penggunanya adalah sebagai berikut:

1. Memberikan pemahaman mengenai konsep proses pengembangan sistem informasi penggajian.
2. Menjadi referensi bagi penelitian berikutnya dalam pengembangan sistem informasi penggajian.

1.6 Metodologi Penelitian

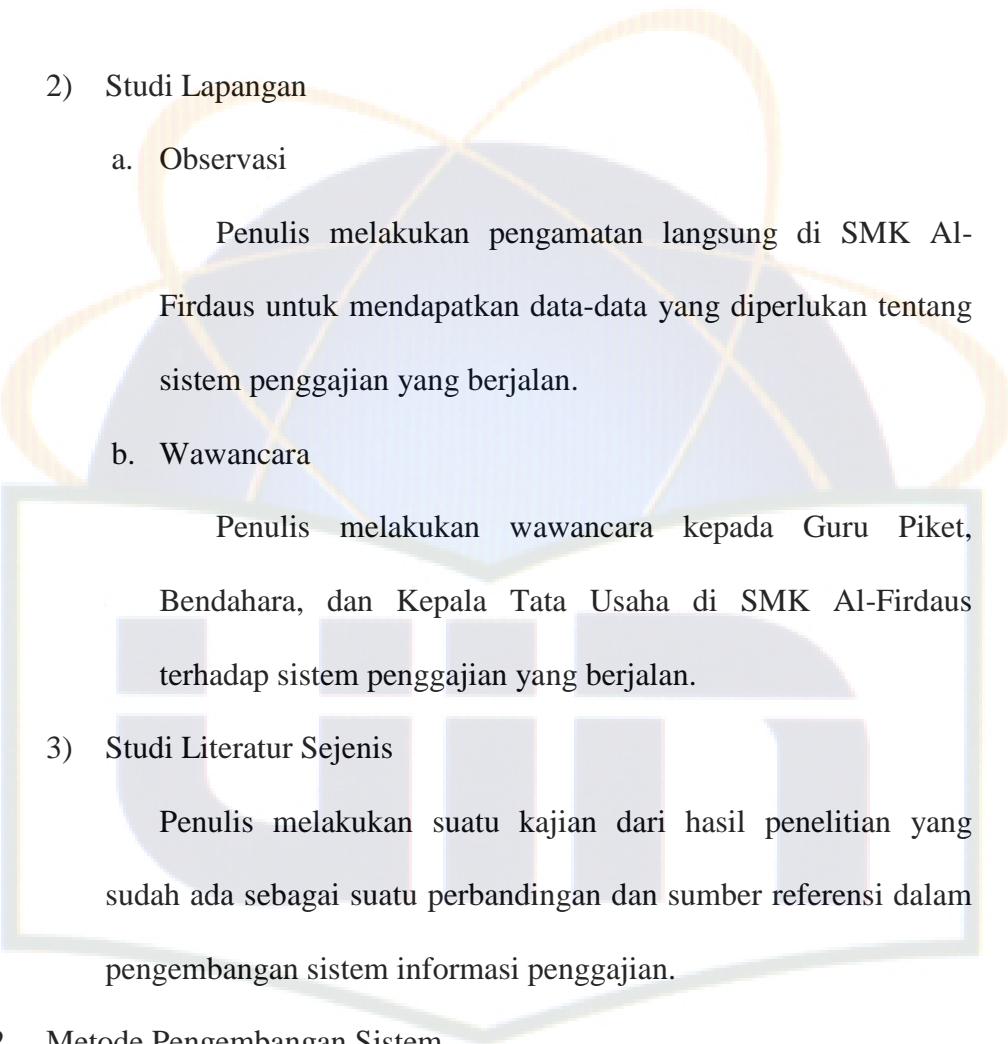
Metode yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1) Studi Pustaka

Penulis membaca dan mempelajari buku-buku, jurnal/artikel dari *website* yang ada kaitannya dengan pengembangan sistem informasi penggajian yang dapat dijadikan bahan acuan penulis dalam pembahasan masalah penelitian ini.



2) Studi Lapangan

a. Observasi

Penulis melakukan pengamatan langsung di SMK Al-Firdaus untuk mendapatkan data-data yang diperlukan tentang sistem pengajian yang berjalan.

b. Wawancara

Penulis melakukan wawancara kepada Guru Paket, Bendahara, dan Kepala Tata Usaha di SMK Al-Firdaus terhadap sistem pengajian yang berjalan.

3) Studi Literatur Sejenis

Penulis melakukan suatu kajian dari hasil penelitian yang sudah ada sebagai suatu perbandingan dan sumber referensi dalam pengembangan sistem informasi pengajian.

2. Metode Pengembangan Sistem

Metodologi pengembangan sistem yang digunakan untuk mengembangkan sistem ini yaitu *Rapid Application Development* (RAD). RAD merupakan sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang ekstensif dalam konstruksi, cepat, berulang dan bertambah serangkaian prototipe bekerja sebuah sistem yang pada akhirnya berkembang ke dalam sistem *final*. RAD dibagi dalam beberapa tahapan, diantaranya: *Scope Definition, Analysis, Design, Construction and Testing, Delivery of a version, dan System Operation and Maintenance* (Whitten, 2004).

Namun penulis membatasi dengan hanya melalui beberapa tahapan, dari *Scope Definition* sampai dengan tahapan *Construction & Testing*, selebihnya diserahkan pada pihak yayasan dalam mengelola aplikasi ini.

Adapun *software* yang digunakan dalam pembuatan sistem ini adalah *EditPlus 2* dan Macromedia Dreamweaver 8 sebagai *software editor*, Adobe Photoshop CS2 untuk mengolah gambar. XAMPP versi 1.7.1. yang mencakup: Apache versi 2.2.11 untuk *web server*, PHP versi 5.2.9 untuk bahasa pemrograman, dan MySQL versi 5.1.33 untuk *database* nya.

1.7 Sistematika Penulisan

Sistematika penulisan skripsi yang merupakan pengembangan sistem informasi dari hasil penelitian meliputi:

BAB I PENDAHULUAN

Bab ini berisikan latar belakang, permasalahan, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menguraikan teori-teori yang terkait dengan konsep dalam penulisan penelitian ini seperti konsep sistem informasi, sistem informasi pengajian, serta teori pendukung lainnya.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metodologi yang digunakan dalam menyusun skripsi ini, yang meliputi metode pengumpulan data dan metode pengembangan sistem, serta kerangka berpikir dalam pelaksanaan penelitian.

BAB IV PENGEMBANGAN SISTEM INFORMASI PENGGAJIAN

Bab ini menjelaskan seluruh tahapan pengembangan sistem informasi penggajian pada SMK Al-Firdaus.

BAB V PENUTUP

Bab ini menyampaikan kesimpulan dari penelitian yang dilakukan dan saran-saran yang diusulkan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih baik.

BAB II

LANDASAN TEORI

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem adalah suatu kesatuan utuh yang terdiri dari beberapa bagian yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu (Wahyono, 2003). Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain (Al Fatta, 2007).

Terdapat dua kelompok pendekatan dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya (Jogiyanto, 2005):

- a. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut ini:

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (FitzGerald, 1981).

- b. Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut:

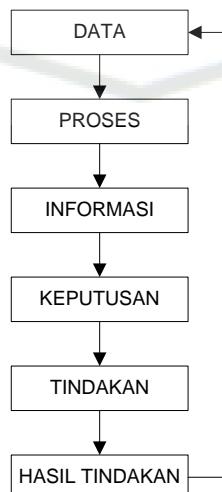
Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

2.1.2 Informasi

Informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan (Wahyono, 2003).

Sedangkan data adalah bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda, dan sebagainya. Data terbentuk dari karakter, dapat berupa alfabet, angka, maupun simbol khusus (Wahyono, 2003).

Pengolahan data menjadi suatu informasi dapat digambarkan sebagai sebuah siklus yang berkesinambungan seperti berikut (Wahyono, 2003):



Gambar 2.1 Siklus Informasi

Secara sederhana dapat dikatakan bahwa data diolah menjadi suatu informasi. Dan pada tahapan selanjutnya, sebuah informasi akan menjadi data untuk terciptanya informasi yang lain.

Kualitas informasi (*quality of information*) sangat dipengaruhi atau ditentukan oleh 3 hal pokok, yaitu *relevancy*, *accuracy* dan *timeliness* (Wahyono, 2003).

a. Relevansi (*relevancy*)

Informasi dikatakan berkualitas jika relevan bagi pemakainya. Informasi akan relevan jika memberikan manfaat bagi pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

b. Akurasi (*accuracy*)

Sebuah informasi dapat dikatakan akurat jika informasi tersebut tidak bias atau menyesatkan, bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksudnya. Ketidakakuratan sebuah informasi dapat terjadi karena sumber informasi (data) mengalami gangguan atau kesengajaan sehingga merusak atau merubah data-data yang asli tersebut.

c. Tepat waktu (*timeliness*)

Informasi yang dihasilkan dari suatu proses pengolahan data, datangnya tidak boleh terlambat. Informasi yang terlambat tidak akan mempunyai nilai yang baik, sehingga kalau digunakan sebagai dasar pengambilan keputusan dapat menimbulkan kesalahan dalam tindakan yang akan diambil.

2.1.3 Sistem Informasi

Sistem informasi adalah suatu cara tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses dan untuk organisasi bisnis dengan cara yang menguntungkan (Wahyono, 2003).

Sistem informasi adalah suatu sistem di dalam satu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005).

Suatu sistem informasi dapat didefinisikan sebagai berikut (Ladjamudin, 2005):

- a. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
- b. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.
- c. Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sebuah sistem informasi yang lengkap memiliki kelengkapan perangkat sistem sebagai berikut (Wahyono, 2003):

1. *Hardware*

Bagian ini merupakan bagian perangkat keras sistem informasi. Sistem informasi modern memiliki perangkat keras seperti komputer, printer, dan teknologi jaringan komputer.

2. *Software*

Bagian ini merupakan bagian perangkat lunak sistem informasi. Sistem informasi modern memiliki perangkat lunak untuk memerintahkan komputer melaksanakan tugas yang harus dilakukannya.

3. Data

Merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.

4. Prosedur

Merupakan bagian yang berisikan dokumentasi prosedur atau proses-proses yang terjadi dalam sistem.

5. Manusia

Manusia merupakan bagian utama dalam suatu sistem informasi. Dalam sistem informasi, manusia dapat bertindak sebagai pembangun, pengelola, maupun sebagai pengguna (*user*) dari sistem tersebut.

2.2 Konsep Penggajian

2.2.1 Gaji atau Upah

Gaji merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan yang dibayarkan secara tetap per bulan. Sedangkan upah adalah pembayaran atas penyerahan jasa yang dilakukan oleh karyawan pelaksana (buruh) yang dibayarkan berdasarkan hari kerja, jam kerja, atau jumlah satuan produk yang dihasilkan oleh karyawan (Mulyadi, 2001).

2.2.2 Gaji Pokok

Gaji pokok adalah gaji dasar yang ditetapkan untuk melaksanakan satu jabatan atau pekerjaan tertentu pada golongan pangkat dan waktu tertentu. Gaji pokok di suatu perusahaan disusun menurut jenjang jabatan dan jenjang kepangkatan (Samsudin, 2006).

2.2.3 Jabatan

Jabatan adalah kelompok pekerjaan atau tugas-tugas dalam unit organisasi yang dalam pelaksanaannya memerlukan syarat-syarat tertentu yang sama atau hampir sama. Syarat-syarat yang diperlukan untuk melakukan pekerjaan pada jabatan tertentu berbeda dengan syarat yang diperlukan untuk melakukan pekerjaan-pekerjaan pada jabatan yang lain (Samsudin, 2006).

2.2.4 Tunjangan

Tunjangan adalah suplemen terhadap upah atau gaji pokok dalam fungsi, yaitu berkaitan dengan pelaksanaan tugas, dalam rangka fungsi sosial, dan sebagai intensif. Tunjangan yang berkaitan dengan pelaksanaan tugas adalah tunjangan jabatan, baik jabatan struktural maupun jabatan fungsional (Samsudin, 2006).

2.2.5 Sistem Pengupahan atau Penggajian

Penggajian adalah suatu penerimaan sebagai imbalan dari pengusaha kepada karyawan untuk suatu pekerjaan atau jasa yang telah atau dilakukan dan dinyatakan atau dinilai dalam bentuk uang yang ditetapkan atas dasar suatu persetujuan atau peraturan perundang-undangan serta dibayarkan atas dasar suatu perjanjian kerja antara pengusaha dengan karyawan termasuk tunjangan, baik untuk karyawan itu sendiri maupun untuk keluarga (Sumarsono, 2003).

Sistem pengupahan atau penggajian merupakan kerangka bagaimana upah diatur dan ditetapkan sistem. Pengupahan di Indonesia pada umumnya didasarkan kepada tiga fungsi upah, yaitu: (a) menjamin kehidupan yang layak bagi pekerja dan keluarganya; (b) mencerminkan imbalan atas hasil kerja seseorang; (c) menyediakan insentif untuk mendorong peningkatan produktivitas kerja (Sumarsono, 2003).

Penghasilan atau imbalan yang diterima seseorang karyawan atau pekerja sehubungan dengan pekerjaannya dapat digolongkan ke dalam

bentuk, yaitu: (a) upah atau gaji dalam bentuk uang; (b) tunjangan dalam bentuk natural; (c) *fringe benefit*; dan (d) kondisi lingkungan kerja (Sumarsono, 2003).

2.2.6 Fungsi yang Terkait

Fungsi yang terkait dalam sistem informasi penggajian adalah sebagai berikut (Mulyadi, 2001):

1. Fungsi Kepegawaian

Fungsi ini bertanggung jawab untuk mencari karyawan baru, menyeleksi calon karyawan, memutuskan penempatan karyawan baru, membuat surat keputusan tarif gaji dan upah karyawan, kenaikan pangkat dan golongan gaji, mutasi karyawan dan pemberhentian karyawan.

2. Fungsi Pencatat Waktu

Fungsi ini bertanggung jawab untuk menyelenggarakan catatan waktu hadir bagi semua karyawan perusahaan. Sistem pengendalian yang baik mensyaratkan fungsi pencatatan waktu hadir karyawan tidak boleh dilaksanakan oleh fungsi operasi atau fungsi pembuat daftar gaji dan upah.

3. Fungsi Pembuat Daftar Gaji dan Upah

Fungsi ini bertanggung jawab untuk membuat daftar gaji dan upah yang berisi penghasilan bruto yang menjadi hak dan berbagai potongan yang menjadi beban setiap karyawan selama jangka waktu pembayaran gaji dan upah. Daftar gaji dan upah

kepada fungsi akuntansi guna pembuatan bukti kas keluar yang dipakai sebagai dasar untuk pembayaran gaji dan upah kepada karyawan.

4. Fungsi Akuntansi

Fungsi ini bertanggung jawab untuk mencatat kewajiban yang timbul dalam hubungannya dengan pembayaran gaji dan upah karyawan (misalnya utang gaji dan upah karyawan, utang pajak, utang dana pensiun).

5. Fungsi Keuangan

Fungsi ini bertanggung jawab untuk mengisi cek guna pembayaran gaji dan upah dan menguangkan cek tersebut ke bank atau melakukan transfer gaji dan upah ke rekening karyawan.

2.2.7 Dokumen yang Digunakan

Dokumen yang digunakan dalam sistem informasi penggajian adalah (Mulyadi, 2001):

1. Dokumen pendukung perubahan gaji dan upah.

Dokumen-dokumen ini umumnya dikeluarkan oleh fungsi kepegawaian berupa surat-surat keputusan yang bersangkutan dengan karyawan, seperti: surat keputusan pengangkatan karyawan baru, kenaikan pangkat, perubahan tarif upah, penurunan pangkat, pemberhentian sementara dari pekerjaan (skorsing), pemindahan, dan lain sebagainya. Tembusan

dokumen-dokumen ini dikirimkan ke fungsi pembuat daftar gaji dan upah untuk kepentingan pembuatan daftar gaji dan upah.

2. Kartu jam hadir

Dokumen ini digunakan oleh fungsi pencatat waktu untuk mencatat waktu hadir setiap karyawan di perusahaan.

3. Kartu jam kerja

Dokumen ini digunakan untuk mencatat waktu yang dikonsumsi oleh tenaga kerja langsung pabrik guna mengerjakan pesanan tertentu. Dokumen ini diisi oleh mandor pabrik dan diserahkan ke fungsi pembuat daftar gaji dan upah untuk kemudian dibandingkan dengan kartu jam hadir, sebelum digunakan untuk distribusi biaya upah langsung kepada setiap jenis produk atau pesanan.

4. Daftar gaji dan upah

Dokumen ini berisi jumlah gaji dan upah bruto setiap karyawan, dikurangi potongan-potongan berupa PPh Pasal 21, utang karyawan, iuran untuk organisasi karyawan, dan lain sebagainya.

5. Rekap daftar gaji dan rekap daftar upah

Dokumen ini merupakan ringkasan gaji dan upah per departemen, yang dibuat berdasarkan daftar gaji dan upah. Dalam perusahaan yang produksinya berdasarkan pesanan, rekap daftar upah dibuat untuk membebankan upah langsung dalam hubungannya dengan produk pesanan yang bersangkutan.

6. Surat pernyataan gaji dan upah

Dokumen ini dibuat oleh fungsi pembuat daftar gaji dan upah bersamaan dengan pembuatan daftar gaji dan upah atau dalam kegiatan yang terpisah dari pembuatan daftar gaji dan upah. Dokumen ini dibuat sebagai catatan bagi setiap karyawan mengenai rincian gaji dan upah yang diterima setiap karyawan beserta berbagai potongan yang menjadi beban setiap karyawan.

7. Amplop gaji dan upah

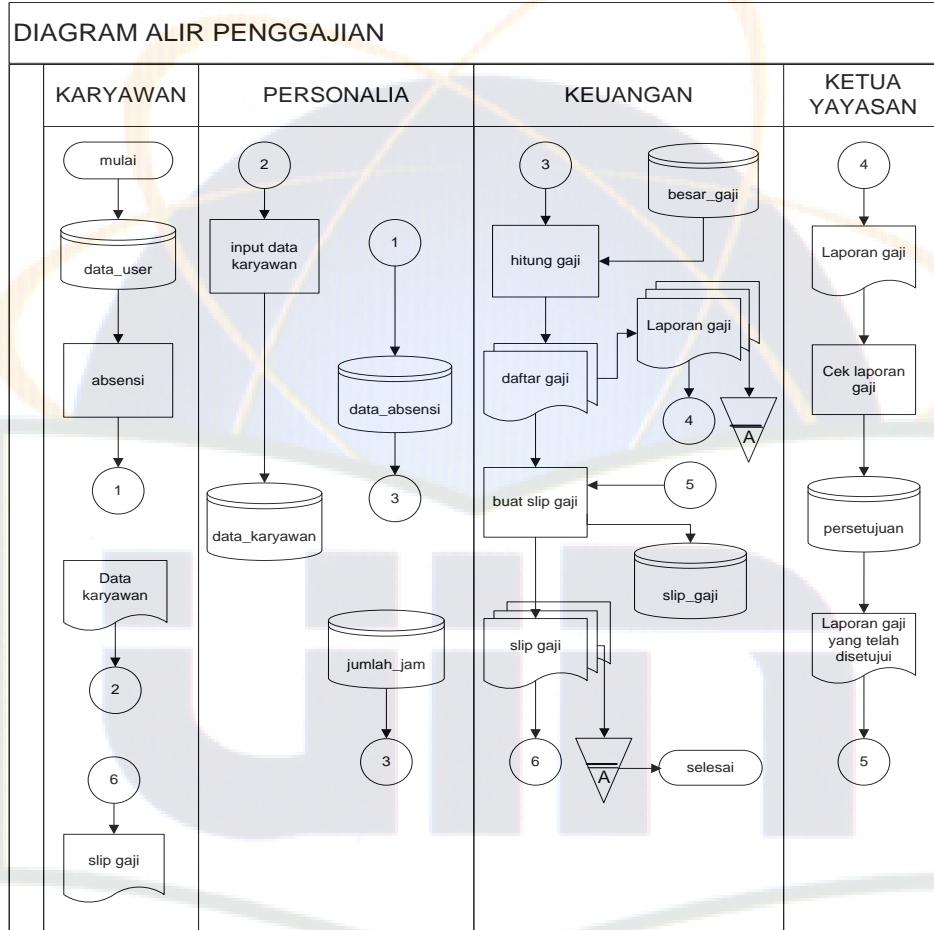
Uang gaji dan upah karyawan diserahkan kepada setiap karyawan dalam amplop gaji dan upah. Di halaman muka amplop gaji dan upah setiap karyawan ini berisi informasi mengenai nama karyawan, nomor identifikasi karyawan dan jumlah gaji bersih yang diterima karyawan dalam bulan tertentu.

8. Bukti kas keluar

Dokumen ini merupakan perintah pengeluaran uang yang dibuat oleh fungsi keuangan, berdasarkan informasi dalam daftar gaji dan upah yang diterima dari fungsi pembuat daftar gaji dan upah.

2.2.8 Diagram Alir (*Flowmap*) Sistem Penggajian

Berikut ini adalah contoh diagram alir (*flowmap*) sistem penggajian dapat dilihat pada gambar 2.2 (Mulyadi, 2008).



Gambar 2.2 Diagram Alir Sistem Penggajian (Mulyadi, 2008)

2.3 Metodologi Penelitian

Metodologi merupakan suatu formula dalam penerapan penelitian dimana dalam melakukan penelitian tersebut terdapat langkah-langkah dan juga hasil penelitian. Sedangkan metodologi penelitian dalam ilmu komputer/sistem informasi/teknologi informasi merupakan langkah-langkah/tahapan perencanaan dengan bantuan beberapa metode, teknik, alat (*tools*) dan dokumentasi dengan tujuan untuk membantu peneliti dalam meminimalkan resiko kegagalan dan menekankan pada proses/sasaran penelitian di bidang CS/IS/IT (Hasibuan, 2007).

Penelitian merupakan sebuah metode untuk menemukan kebenaran yang juga merupakan sebuah pemikiran kritis (*critical thinking*). Penelitian meliputi pemberian definisi dan redefinisi terhadap masalah, memformulasikan hipotesis atau jawaban sementara, membuat kesimpulan dan sekurang-kurangnya mengadakan pengujian yang hati-hati atas semua kesimpulan untuk menentukan apakah ia cocok dengan hipotesis (Nazir, 2005).

Langkah-langkah pokok dalam meneliti kasus adalah sebagai berikut (Nazir, 2005):

1. Rumuskan tujuan penelitian.
2. Tentukan unit-unit studi, sifat-sifat mana yang akan diteliti dan hubungkan apa yang akan dikaji serta proses-proses apa yang akan menuntun peneliti.
3. Tentukan rancangan serta pendekatan dalam memilih unit-unit dan teknik pengumpulan data mana yang digunakan. Sumber-sumber data apa yang tersedia.
4. Kumpulkan data.
5. Organisasikan infomasi serta data yang terkumpul dan analisis untuk membuat interpretasi serta generalisasi.
6. Susun laporan dengan memberikan kesimpulan serta implikasi dari hasil penelitian.

Kegunaan penelitian ialah untuk menyelidiki keadaan dari, alasan untuk, dan konsekuensi terhadap suatu set keadaan khusus. Keadaan tersebut bisa saja dikontrol melalui percobaan (eksperimen) ataupun

berdasarkan observasi tanpa kontrol. Penelitian memegang peranan yang amat penting dalam memberikan fondasi terhadap tindak serta keputusan dalam segala aspek pembangunan. Adalah sangat sulit, bahkan tidak mungkin sama sekali, untuk memperoleh data yang terpercaya yang dapat digunakan dalam perencanaan pembangunan, jika penelitian tidak pernah adakan, serta kenyataan-kenyataan tidak pernah di uji lebih dahulu melalui penelitian (Nazir, 2005).

2.4 Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan untuk mencapai tujuan penelitian (Gulo, 2002). Pada penelitian ini, ada beberapa metode pengumpulan data yang dilakukan oleh penulis adalah sebagai berikut:

1. Studi Pustaka

Studi pustaka adalah penelitian yang membandingkan teori-teori yang berlaku dan dapat dicari pada buku-buku teks ataupun hasil dari penelitian orang lain baik yang sudah dipublikasikan maupun belum merupakan suatu faktor dari keilmiahinan penelitian yang dilakukan (Umar, 2007).

2. Studi Lapangan

a. Observasi

Observasi (pengamatan) adalah metode pengumpulan data di mana peneliti mencatat informasi sebagaimana yang mereka saksikan selama penelitian (Gulo, 2002).

b. Wawancara

Wawancara adalah bentuk komunikasi langsung antara peneliti dengan responden. Komunikasi berlangsung dalam bentuk tanya-jawab dalam hubungan tatap muka, sehingga gerak dan mimik responden merupakan pola media yang melengkapi kata-kata secara verbal (Gulo, 2002).

3. Studi Literatur Sejenis

Studi literatur merupakan salah satu kegiatan penelitian yang mencakup: memilih teori-teori hasil penelitian, mengidentifikasi literatur, dan menganalisa dokumen, serta menerapkan hasil analisa sebagai landasan teori bagi penyelesaian masalah dalam penelitian yang dilakukan. Teori yang tepat disini adalah teori-teori yang bersesuaian dengan ruang lingkup masalah (Subana, 2005).

2.5 Metode Pengembangan Sistem

Pengembangan sistem (*system development*) dapat berarti menyusun suatu sistem baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki atau diganti disebabkan karena beberapa hal (Jogiyanto, 2005).

Pada umumnya proses pengembangan sistem sederhana di organisasi mengikuti pendekatan pemecahan masalah. Pendekatan tersebut biasanya terdiri dari beberapa langkah *problem-solving* yang umum yaitu (Whitten, 2004):

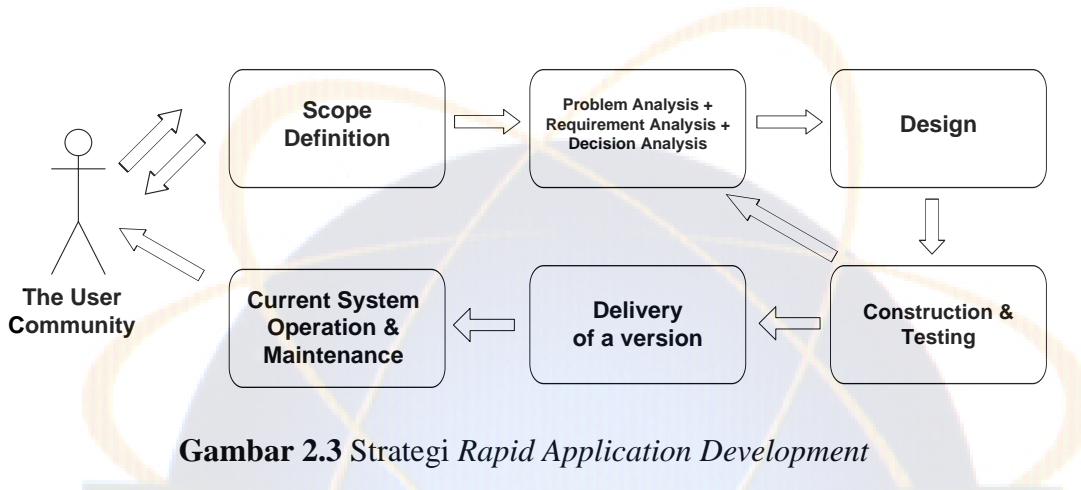
1. Mengidentifikasi masalah.
2. Menganalisis dan memahami masalah.
3. Mengidentifikasi solusi yang diharapkan.
4. Mengidentifikasi solusi alternatif dan memilih tindakan yang terbaik.
5. Mendesain solusi yang dipilih.
6. Mengimplementasikan solusi yang dipilih.
7. Mengevaluasi hasilnya (jika masalah tidak terpecahkan, kembali ke langkah 1 atau 2 seperlunya).

2.6 *Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah suatu pendekatan desain sistem yang menggunakan teknik terstruktur, prototyping, dan JAD (*Joint Application Development*) untuk mengembangkan sistem secara cepat (Whitten, 2004).

RAD adalah sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang ekstensif dalam konstruksi, cepat, berulang dan bertambah serangkaian *prototype* bekerja sebuah sistem yang pada akhirnya berkembang ke dalam sistem *final* (Whitten, 2004).

Secara umum, strategi dari RAD dapat digambarkan melalui gambar 2.3 dibawah ini:



Gambar 2.3 Strategi *Rapid Application Development*

Sebagai respon pada kemajuan ekonomi pada umumnya, *Rapid Application Development* (RAD) / pengembangan aplikasi cepat telah menjadi rute yang populer untuk mengakselerasi pengembangan sistem. Gagasan-gagasan RAD adalah (Whitten, 2004):

1. Lebih aktif melibatkan para pengguna sistem dalam aktivitas analisis, desain, konstruksi.
2. Mengorganisasikan pengembangan sistem ke dalam rangkaian seminar yang intensif dan berfokus dengan para pemilik, pengguna, analis, desainer, pembangun sistem.
3. Mengakselerasi fase-fase analisis dan desain persyaratan melalui pendekatan konstruksi berulang.
4. Memperpendek waktu yang diperlukan sebelum para pengguna mulai melihat sebuah sistem yang bekerja.

Model RAD mempunyai beberapa keunggulan sebagai berikut (Whitten, 2004):

1. Mendorong pengguna aktif dan partisipasi manajemen (berkebalikan dengan reaksi pasif pada model-model sistem yang tidak bekerja). Hal ini meningkatkan antusiasme pengguna akhir proyek.
2. Proyek-proyek memiliki visibilitas dan dukungan lebih tinggi karena keterlibatan pengguna yang ekstensif selama proses.
3. Para pengguna dan manajemen melihat solusi-solusi yang berbasis perangkat lunak dan bekerja lebih cepat dari pada pengembangan model-driven.
4. *Error* dan penghilangan cenderung untuk dideteksi lebih awal dalam prototipe dari pada model sistem.
5. Pengujian dan pelatihan adalah produk tambahan alami dari pendekatan *prototyping* yang mendasar.
6. Pendekatan berulang adalah proses yang lebih “alami” karena perubahan adalah faktor yang diharapkan selama pengembangan.

2.7 Unified Modeling Language (UML)

2.7.1 Definisi UML

UML adalah keluarga notasi grafis yang didukung oleh model-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi obyek (Fowler, 2004)

UML merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan obyek (Whitten, 2004). Jeffrey L.

Whitten menambahkan bahwa UML menawarkan diagram yang dikelompokkan menjadi beberapa perspektif berbeda untuk memodelkan suatu sistem, seperti satu set cetak biru (*blueprint*) yang digunakan untuk membangun sebuah rumah (Whitten, 2004).

2.7.2 Diagram UML

2.7.2.1 *Use Case Diagram*

Diagram yang menggambarkan interaksi antara sistem dengan eksternal sistem dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem (Whitten, 2004). Adapun simbol-simbol dari *use case diagram* terdapat pada daftar simbol.

Dalam *use case diagram* memiliki pemodelan sebagai berikut (Whitten, 2004):

1. *Use case*

Use case merupakan urutan langkah-langkah yang secara tindakan saling terkait (*scenario*), baik otomatis maupun secara *manual*.

2. *Actor* (Pelaku)

Actor merupakan segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi.

3. Relationship (Hubungan)

Pada diagram *use case*, *relationship* digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan *relationship* berbeda-beda tergantung bagaimana garis tersebut digambar dan tipe simbol apa yang digunakan untuk menghubungkan garis tersebut. Berikut ini adalah perbedaan di antara *relationship* yang ada pada sebuah diagram *use case*:

a. *Association*

Association merupakan *relationship* antara *actor* dengan *use case* dimana terjadi interaksi di antara mereka.

b. *Extends*

Extends use case merupakan *use case* yang terdiri dari langkah yang terekstraksi dari *use case* yang lebih kompleks untuk menyederhanakan masalah dan karena itu memperluas fungsinya.

c. *Uses (includes)*

Hubungan *uses* menggambarkan bahwa satu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya.

d. *Depends on*

Terkadang suatu *use case* memiliki ketergantungan pada *use case* yang lainnya yang

bertujuan untuk menentukan urutan dalam pengembangan *use case*. Ketergantungan ini dimodelkan menggunakan *depends on relationship*.

e. *Inheritance*

Hubungan *inheritance* terjadi ketika dua atau lebih *actor* menggunakan *use case* yang sama.

2.7.2.2 *Activity Diagram*

Activity diagram secara grafis digunakan untuk menggambarkan rangkaian aliran aktifitas baik proses bisnis atau *usecase* (Whitten, 2004). Adapun simbol-simbol dari *activity diagram* terdapat pada daftar simbol.

2.7.2.3 *Sequence Diagram*

Sequence diagram secara grafis menggambarkan bagaimana *object* berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah *usecase* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara *object* dan *sequence* (ruang waktu) (Whitten, 2004). Adapun simbol-simbol dari *sequence diagram* terdapat pada daftar simbol.

2.7.2.4 Class Diagram

Class diagram gambar grafis mengenai struktur obyek statis dari suatu sistem, menunjukkan kelas-kelas obyek yang menyusun sebuah sistem dan juga hubungan antara kelas obyek tersebut (Whitten, 2004). Adapun simbol-simbol dari *class diagram* terdapat pada daftar simbol.

2.7.3 Keunggulan UML

Adi Nugroho mengemukakan bahwa secara umum UML diterapkan dalam pengembangan sistem/perangkat lunak berorientasi obyek sebab metodologi UML ini umumnya memiliki keunggulan-keunggulan sebagai berikut (Nugroho, 2005):

- a. **Uniformity.** Dengan metodologi UML, para pengembang cukup menggunakan 1 metodologi dari tahap analisis hingga perancangan. Hal ini tidak bisa dilakukan dalam metodologi pengembangan terstruktur. Dengan perkembangan masa kini ke arah aplikasi GUI (*Graphical User Interface*), UML juga memungkinkan kita merancang komponen antarmuka pengguna (*user interface*) secara integrasi bersama dengan perancangan perangkat lunak sekaligus dengan perancangan basis data.
- b. **Understandability.** Dengan metodologi ini kode yang dihasilkan dapat diorganisasi ke dalam kelas-kelas yang berhubungan dengan masalah sesungguhnya sehingga lebih mudah dipahami siapapun juga.

- c. ***Stability.*** Kode program yang dihasilkan relatif stabil sepanjang waktu sebab sangat mendekati permasalahan sesungguhnya di lapangan.
- d. ***Reusability.*** Dengan metodologi berorientasi obyek, dimungkinkan penggunaan ulang kode, sehingga pada gilirannya akan sangat mempercepat waktu pengembangan perangkat lunak.

2.8 Konsep *Database* (Basis Data)

2.8.1 *Database* (Basis Data)

Database merupakan kumpulan *file* yang saling berhubungan. Akan tetapi, *database* tidak hanya kumpulan *file*. *Record* di dalam tiap *file* harus dapat dihubungkan dengan *record* di dalam *file* lain (Whitten, 2004).

Dalam manajemen *database relational* terdapat komponen utama dalam konsep *database* (Whitten, 2004).

1. *Field* adalah unit terkecil data yang disimpan dalam *database*.

Unit terkecil data yang disimpan dalam *database*:

- a. *Primary key* yaitu *field* yang unik dan mengidentifikasi satu *record*. Contoh: Customer Number dan Order Number.
- b. *Secondary key* yaitu *field* yang mengidentifikasi sebuah *record* atau bagian dari beberapa *record* yang terkait.
- c. *Foreign key* yaitu *field* yang menunjuk beberapa *record* pada *file* lain. Contoh: Order Record berisi *foreign key* Customer Number.

- d. *Descriptive field* yaitu *non-key field*.
2. *Record* adalah kumpulan *field* yang diatur dalam format yang *predetermined* (telah ditentukan).
- a. *Fixed-length record structures*
- Sebagian besar teknologi *database* memaksakan struktur *record fixed length*, dalam artian setiap *instance record* mempunyai *field* yang sama, jumlah *field* yang sama, dan ukuran logika yang sama. Akan tetapi beberapa sistem *database* akan mengkompresi *field-field* dan nilai-nilai yang tidak digunakan untuk menghemat ruang penyimpanan *disk*.
- b. *Variable-length record structures*
- Memperbolehkan *record-record* pada *file* yang sama memiliki *length* yang berbeda.
3. *File* dan Tabel
- File* adalah kumpulan semua kejadian dari struktur *record* yang ditentukan. Tipe-tipe dari *file* yaitu:
- a. *File* induk/master adalah *file* penting dalam sistem dan akan tetap ada selama siklus hidup sistem informasi berputar.
 - b. *File* transaksi adalah *file* yang digunakan untuk merekam data dari suatu transaksi yang terjadi.
 - c. *File* laporan adalah *file* yang berisi sistem informasi yang akan ditampilkan.
 - d. *File* sejarah adalah *file* yang berisi data masa lalu yang sudah tidak aktif lagi.

- e. *File* pelindung adalah salinan dari *file-file* yang masih aktif di *database* pada saat tertentu yang digunakan bila *file database* rusak.
- f. *File* kerja adalah suatu proses program secara sementara karena memori komputer tidak mencukupi.

Sedangkan tabel adalah suatu kesatuan unit dari *row / record* dengan atribut-atributnya (*column*).

2.8.2 Database Management System (DBMS)

DBMS (*Database Management System*) adalah kumpulan/gabungan *database* dengan perangkat lunak aplikasi yang berbasis *database*. DBMS merupakan koleksi terpadu dari *database* dan program-program komputer yang digunakan untuk mengakses dan memelihara *database*. Program-program tersebut menyediakan berbagai fasilitas operasi untuk memasukkan, melacak, dan memodifikasi data kedalam *database*, mendefinisikan data baru, serta mengolah data menjadi informasi yang dibutuhkan (Ladjamudin, 2005).

Tujuan utama DBMS adalah menyediakan lingkungan yang nyaman dan efisien untuk penyimpanan dan pengambilan data dari basisdata. DBMS berperan memberi abstraksi data tingkat tinggi ke pemakai. Sedangkan tujuan lain dari DBMS antara lain (Hariyanto, 2004):

1. Menghindari redudansi dan inkonsistensi data.
2. Menghindari kesulitan pengaksesan data.
3. Menghindari isolasi data.
4. Menghindari terjadinya anomali pengaksesan konkuren.
5. Menghindari masalah-masalah keamanan.
6. Menghindari masalah-masalah integritas.

Keunggulan DBMS (*Database Management System*) antara lain

(Hariyanto, 2004):

1. Pengendalian terhadap redudansi data
2. Konsistensi data
3. Informasi yang lebih banyak yang dapat dibentuk dari data tersimpan yang sama.
4. Pemakaian bersama data.
5. Peningkatan integritas data.
6. Penyeimbangan kebutuhan-kebutuhan sumber daya yang terbatas.
7. Peningkatan pengaksesan dan daya tanggap data.
8. Peningkatan produktivitas.
9. Peningkatan pemeliharaan lewat ketidakbergantungan data.
10. Peningkatan konkurensi.
11. Peningkatan layanan *backup* dan pemulihan data.

2.9 Konsep Internet

2.9.1 Definisi Internet

Interconnected Network yang lebih populer dengan sebutan *Internet* merupakan sekumpulan jaringan yang terhubung satu dengan yang lainnya, dimana jaringan menyediakan sambungan menuju global informasi. *Internet* telah memungkinkan komunikasi antar komputer dengan menggunakan *Transmission Control Protocol / Internet Protocol* (TCP/IP) yang didukung media komunikasi, seperti satelit dan paket radio. Jadi, jarak jangkaunya tidak terbatas. (Oetomo, 2007).

2.9.2 Sumber Daya Internet

Internet memiliki banyak sumber daya seperti pada tabel 2.1 dibawah ini (Kadir, 2003).

Tabel 2.1 Sumber Daya Pada *Internet*

Sumber	Keterangan
<i>E-mail</i>	Digunakan untuk melakukan pertukaran surat elektronis.
<i>User Newsgroup</i>	Forum Diskusi
<i>LISTSERV</i>	Kelompok diskusi dengan menggunakan surat elektronis.
<i>Internet Relay Chat (IRC)</i>	Fasilitas yang memungkinkan pemakai melakukan percakapan dalam bentuk bahasa tertulis secara interaktif.
<i>Telnet</i>	Fasilitas yang memungkinkan koneksi (<i>login</i>) ke suatu sistem komputer.
<i>File Transfer Protocol (FTP)</i>	Sarana untuk melakukan transfer berkas dari komputer lokal ke komputer lain atau sebagainya.

<i>Gopher</i>	Perangkat yang memungkinkan pemakai untuk menemukan informasi yang terdapat pada <i>server gopher</i> melalui menu yang bersifat hirarkis.
<i>Archie</i>	Perangkat yang dapat digunakan untuk melakukan pencarian berkas pada situs FTP.
<i>Very Easy Rodent-Oriented Netwide Index to Computer Archives</i> (Veronica)	Merupakan kemampuan tambahan yang dipakai untuk melakukan pencarian pada situs-situs <i>gopher</i> .
<i>Wide Area Information Servers</i> (WAIS)	Perangkat yang digunakan untuk melakukan pencarian data pada <i>internet</i> yang dilaksanakan dengan menyebutkan nama <i>database</i> dan kata kunci yang dicari.
<i>World Wide Web</i> (WWW)	Sistem yang memungkinkan pengaksesan informasi dalam <i>internet</i> melalui pendekatan <i>hypertext</i> (HTTP).

2.9.3 *HyperText Transfer Protocol* (HTTP)

HyperText Transfer Protocol (HTTP) adalah suatu protokol *internet* yang digunakan oleh WWW. Dengan protokol ini sebuah *web client* (dalam hal ini *browser*) seperti *Internet Explorer* atau *Mozilla Firefox* dapat melakukan pertukaran data *hypermedia*, seperti teks, gambar, suara bahkan video dengan *web server*. HTTP pertama kali dibuat oleh TIM Berners-Lee pada tahun 1990, dengan versi HTTP/0.9. Versi terbaru HTTP adalah HTTP/1.1. (Widodo, 2005).

2.9.4 Aplikasi Web

Aplikasi *web* dapat dibagi menjadi 2 kategori, yaitu *web* statis dan *web* dinamis (Kadir, 2003).

a. *Web* Statis

Web statis adalah aplikasi *web* yang berisi atau menampilkan informasi-informasi yang sifatnya statis (tetap).

Disebut tetap karena pengunjung tidak dapat berinteraksi dengan *web* tersebut. Pada *web* statis pengunjung hanya dapat melihat isi dokumen pada halaman *web*. *Web* statis biasanya menggunakan pemrograman *web* HTML dan tidak memiliki *database*.

b. *Web* Dinamis

Web Dinamis adalah aplikasi *web* yang menampilkan informasi serta dapat berinteraksi dengan pengunjung dengan menggunakan *form* sehingga dapat mengolah informasi yang ditampilkan. *Web* dinamis biasanya menggunakan pemrograman *web* PHP dan memiliki *database* untuk menyimpan informasi, seperti MySQL.

2.10 Tools Pengembangan Sistem

2.10.1 Rich Picture

Rich Picture digunakan pada waktu penyeleksian sistem untuk menyatakan secara keseluruhan dalam memahami proses dari

tahapan proyek pengembangan sistem. *Rich Picture* secara khusus mendeskripsikan masalah sistem dan sekaligus penguasaan aplikasi.

Rich Picture tidak didasari pada acuan catatan khusus tetapi seharusnya memberikan penyesuaian mengenai aspek yang dideskripsikan pada proyek (Mathiassen, 2000).

2.10.2 Adobe Photoshop

Adobe Photoshop adalah merupakan *software* standar profesional untuk pengolahan citra atau gambar (*image*) yang memberikan lingkungan komprehensif bagi para desainer grafis profesional untuk membuat desain grafis yang rumit untuk kebutuhan percetakan, situs *web*, dan media lainnya (Sardi, 2004).

2.10.3 Macromedia Dreamweaver

Macromedia Dreamweaver adalah *software* professional untuk desain, pemrograman, dan manajemen situs *web* dengan tingkat fleksibilitas yang sangat tinggi. Tingkat fleksibilitas yang dimaksud memungkinkan *user* dengan keahlian tingkat lanjut (*advance*) untuk membuat situs *web* dengan cara menulis langsung kode HTML pada *tools* pemrograman yang terintegrasi pada *software* ini atau memungkinkan *user* tingkat awal (*beginner*) membuat situs *web* dengan menggunakan lingkungan *visual editing* pada menu-menu yang sudah disediakan (Sardi, 2004).

2.10.4 XAMPP

Xampp merupakan suatu paket yang terdiri dari Apache, MySQL, PHP, dan Perl yang dapat digunakan untuk membantu proses instalasi produk tersebut (Sukarno, 2006).

XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP merupakan *project non profit* yang dikembangkan oleh Apache Friends yang didirikan Kai 'Oswalad', Seidler dan Kay Vogelgesang pada tahun 2002, *project* mereka ini bertujuan mempromosikan penggunaan *Apache Web Server*. Xampp ditemukan pada musim semi tahun 2002.

Adapun keuntungan dari Xampp antara lain (Sukarno, 2006):

1. Xampp dapat berjalan pada *operating system* Linux dan Windows. Dan kemungkinan juga akan berjalan di Macintosh.
2. Mudah bagi *programmer* apabila ingin berganti versi PHP, cukup melakukan *switch* versi.
3. Mudah dan cepat dalam proses instalasi.

2.10.5 Apache

Web server Apache berbasiskan *Open Source* dan mulai populer di *Internet* sejak tahun 1996 karena *open source*, Apache bebas didistribusikan oleh siapa saja dan kepada siapa saja. *Software* ini dapat diunduh pada situs <http://www.apache.org> dan tersedia untuk berbagai *platform* (Windows, Linux, dan UNIX).

Supaya dokumen-dokumen *web* baik berupa HTML ataupun PHP bisa diakses oleh *browser* maka dokumen-dokumen tersebut perlu diletakkan dalam direktori khusus yang diatur oleh Apache. Oleh karena itu, semua skrip PHP yang digunakan perlu diletakkan di dalamnya (Kadir, 2003).

2.10.6 PHP

PHP pertama awalnya diciptakan oleh Rasmus Lerdorf pada tahun 1994 sebuah aplikasi kecil (berbentuk makro) untuk melengkapi personalnya situs personalnya di *internet*. Itu sebabnya, pada saat itu PHP merupakan singkatan dari *Personal Homepage*. Kemudian, dikembangkan lagi oleh masyarakat *internet* sukarelawan pendukung gerakan Public Licences menjadi sebuah bahasa yang lebih sempurna saat ini. Pada tahun 2001, PHP telah digunakan oleh lebih dari lima juta situs interaktif. Selanjutnya PHP berganti nama menjadi PHP: Hypertext Preprocessor. Informasi lengkap mengenai PHP ini dapat Anda peroleh melalui situsnya di <http://www.php.net>. Seperti halnya dengan MySQL yang berada dibawah bendera *open source*, PHP pun bebas di *download*, diubah dan digunakan oleh siapa saja dengan cuma-cuma.

PHP adalah sebuah bahasa pemrograman yang diperintahnya dilaksanakan pada *server* dan kemudian hasilnya ditampilkan pada komputer klien. PHP juga merupakan HTML *embedded*, yaitu perintah PHP dituliskan bersamaan dengan perintah-perintah HTML.

Dapat dikatakan tanpa HTML, maka PHP dapat digunakan sebagaimana mestinya. HTML (*HyperText Markup Language*) sendiri merupakan bahasa untuk membuat tampilan *web*. Jadi, disini ada sinergi dari dua bahasa yang saling menguatkan, yaitu PHP dan HTML (Arbai, 2004).

2.10.7 MySQL

MySQL merupakan *software* sistem manajemen database (*Database Management System-DBMS*) yang *Open Source* (gratis) yang sangat populer di kalangan pemrogram *web*, sehingga dapat digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolah datanya. Hal ini dikarenakan MySQL dapat digunakan cepat secara kinerja *query*, dan mencukupi untuk kebutuhan *database* perusahaan skala menengah kecil (Sidik, 2005).

2.10.8 Web Browser

Abdul Kadir mendefinisikan *web browser* sebagai perangkat lunak yang berguna untuk mengakses informasi *web* ataupun untuk melakukan transaksi via *web*. Beberapa contoh *browser* yang ada saat ini seperti: Internet Explorer, Mozilla Firefox, Opera, Netscape, Safari. Cara kerja *browser* yaitu dengan membaca dokumen HTML yang diambil dari *web server* melalui ISP sebagai koneksi *internet*, kemudian ditampilkan melalui *browser* pada komputer (Kadir 2003).

Web browser berkomunikasi dengan *web server* lewat jaringan komunikasi menggunakan protokol HTTP. *Browser* mengirim pesan meminta dokumen atau layanan tertentu *web server*. *Web server* kemudian menanggapi dengan mengirim dokumen atau menjalankan layanan tertentu di *server* dan mengirim hasil menggunakan protokol HTTP. Kemudian *browser* akan menerima dokumen (HTML) tanggapan dari *web server* dan menampilkannya dilayar (Hariyanto, 2004). Protokol HTML dirancang untuk dapat menggabung semua protokol *internet* seperti Gopher, Telnet, WAIS, dan sebagainya dalam satu protokol tunggal. Semua layanan protokol lain dikemas sebagai layanan-layanan yang disediakan lewat interaksi *web browser* dan *web server*.

2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara *manual* atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan antara hasil yang diharapkan dengan hasil yang terjadi (Hariyanto, 2004).

Terdapat dua teknik pengujian berdasarkan ketersediaan logika sistem, yaitu *black box testing* dan *white box testing* (Hariyanto, 2004). Namun, dalam membangun perangkat lunak ini penulis hanya menggunakan *black box testing*.

Konsep *black box testing* digunakan untuk mempresentasikan sistem yang cara kerja didalamnya tidak tersedia untuk diinspeksi. Di dalam kotak hitam, item-item yang di uji dianggap "gelap" karena logiknya tidak diketahui hanya apa yang masuk dan apa keluar dari kotak hitam. Pada *black box testing*, kasus–kasus pengujian berdasarkan pada spesifikasi sistem. Rencana pengujian dapat dimulai sedini mungkin di proses pengembangan perangkat lunak. Teknik *black box* juga dapat digunakan untuk pengujian berbasis skenario, dimana isi dalam sistem mungkin tidak tersedia untuk diinspeksi tapi masukan dan keluaran yang didefinisikan dengan *use case* dan informasi analisis yang lain (Hariyanto, 2004).

2.12 Literatur Penelitian Sejenis

Sebagai suatu perbandingan dan sumber referensi dalam pengembangan sistem informasi penggajian, diperlukan suatu acuan terhadap penelitian yang dibuat sebelumnya. Hal ini dilakukan untuk mengetahui kekurangan dan kelebihan dari penelitian yang sama guna merancang dan membangun sistem informasi penggajian yang lebih baik dari penelitian sebelumnya. Adapun hasil penelitian sejenis yang dijadikan referensi dalam studi literatur sejenis adalah sebagai berikut:

1. Jane Deviana, 2007, *Sistem Informasi Penggajian dan Pengupahan PT. Kencana Mas Berjaya (Permasalahan dan Solusinya)*, Universitas Bina Nusantara.

2. Krissonya Surbakti, 2007, *Analisa dan Perancangan Sistem Informasi Penggajian Karyawan PT. Indonesia Raya Audivisi*, Universitas Bina Nusantara.
3. Mulyadi, 2008, *Perancangan Aplikasi Penggajian Karyawan Pada Yayasan Darul Irfan*, UIN Jakarta.
4. Yuli Yanah, 2008, *Pengembangan Sistem Informasi Penggajian Karyawan Pada Yayasan Pendidikan Bidayatul Hidayah (YPBH) Bekasi*, UIN Jakarta.
5. Helmi Setiawan, 2008, *Perancangan Sistem Informasi Penggajian dengan Menggunakan Finger Print Pada PT. Refa Qualitama Multitrade*, UIN Jakarta.
6. Andri Chaidir, 2008, *Sistem Informasi Pengolahan Data Transaksi dan Penggajian Karyawan di CV. Bina Teknik*, UNIKOM Bandung.
7. Demita Budi Radityaningrum, 2008, *Sistem Penggajian Karyawan pada Balai Besar Pengembangan Penangkapan Ikan (BBPPI)* Semarang, Universitas Negeri Semarang
8. Ade Sudrajat, 2009, *Pengembangan Sistem Informasi Penggajian PT. Mustika Abadi Sentosa*, UIN Jakarta.
9. Mursyidah Noviyani, 2009, *Sistem Informasi Penggajian Pegawai Madrasah Aliyah Negeri Binjai*, Universitas Sumatera Utara.
10. Irwan Yulistiawan, 2011, *Rancang Bangun Sistem Informasi Penggajian Karyawan (Studi Kasus: PT. Bank Pembiayaan Rakyat Syariah Harta Insan Karimah)*, UIN Jakarta.

Tabel 2.2 Literatur Penelitian Sejenis

No	Pembanding	1 (Jane Deviana)	2 (Arif Giyarti)	3 (Mulyadi)	4 (Yuli Yanah)	5 (Helmi Setiawan)
1	Metode	Berorientasi obyek	Berorientasi obyek	<i>System Development Life Cycle (SDLC)</i>	<i>System Development Life Cycle (SDLC)</i>	<i>System Development Life Cycle (SDLC)</i>
2	Tools	Visual Basic 6.0, Microsoft Access 2003	Visual Basic 6.0, Microsoft Access 2003	Visual Basic 6.0, Microsoft Access 2003, Microsoft Visio 2003.	Visual Basic 6.0, Microsoft Access 2003, Microsoft Visio 2003, <i>finger print.</i>	Visual Basic 6.0 Microsoft SQL Server 2000
3	Sistem berbasis web (<i>online</i>)	<i>Intranet</i>	<i>Intranet</i>	<i>Intranet</i>	<i>Intranet</i>	<i>Intranet</i>
4	Keluaran (<i>output</i>) yang dihasilkan	Laporan data absensi karyawan Laporan lembur Laporan gaji	Laporan data absensi karyawan Laporan lembur Laporan gaji	Rekap absen Laporan penggajian pegawai Slip gaji	Laporan data pegawai laporan absensi Laporan gaji Slip gaji	Laporan data karyawan Laporan absensi karyawan Laporan gaji Slip gaji
5	Kekurangan (Penelitian ini)	-	-	-	Belum memanfaatkan <i>finger print.</i>	Belum memanfaatkan <i>finger print.</i>
6	Kelebihan (Penelitian ini)	Pegawai sudah dapat melihat slip gaji secara <i>online</i> .	Pegawai sudah dapat melihat slip gaji secara <i>online</i> .	Pegawai sudah dapat melihat slip gaji secara <i>online</i> . Ketua Yayasan sudah dapat mengecek laporan penggajian dari sistem.	Pegawai sudah dapat melihat slip gaji secara <i>online</i> . Ketua Yayasan sudah dapat mengecek laporan penggajian dari sistem.	Pegawai sudah dapat melihat slip gaji secara <i>online</i> .

Tabel 2.2 Literatur Penelitian Sejenis (Lanjutan)

No	Pembanding	6 (Andri Chaidir)	7 (Demita Budi Radityaningrum)	8 (Ade Sudrajat)	9 (Mursyidah Noviyani)	10 (Irwan Yulistiawan)
1	Metode	<i>System Development Life Cycle (SDLC)</i>	Analisis deskriptif kualitatif	<i>System Development Life Cycle (SDLC)</i>	<i>System Development Life Cycle (SDLC)</i>	<i>Rapid Application Development (RAD)</i>
2	Tools	Delphi 7.0, Microsoft Access 2003.	<i>Time Recorder Machine</i>	Visual Basic 6.0, Crystal Report 8.5, Microsoft Access 2003	Visual Basic 6.0, Crystal Report	XAMPP 1.6.2 (Apache, PHP & MySQL)
3	Sistem berbasis web (<i>online</i>)	<i>Intranet</i>	<i>Intranet</i>	<i>Intranet</i>	<i>Stand alone</i>	<i>Intranet</i>
4	Keluaran (<i>output</i>) yang dihasilkan	Laporan data absensi Slip gaji	Laporan pencatat waktu hadir Prosedur pembuat daftar gaji Prosedur pembayaran gaji Laporan pembayaran gaji	Laporan data pegawai Laporan data gaji Slip gaji	Laporan data pegawai Laporan data gaji Slip gaji	Laporan data pegawai Laporan data gaji Slip gaji
5	Kekurangan (Penelitian ini)	-	-	-	-	-
6	Kelebihan (Penelitian ini)	Pegawai sudah dapat melihat slip gaji secara <i>online</i> .	Pegawai sudah dapat melihat slip gaji secara <i>online</i> .	Pegawai sudah dapat melihat slip gaji secara <i>online</i> .	Pegawai sudah dapat melihat slip gaji secara <i>online</i> .	Pegawai sudah dapat melihat slip gaji secara <i>online</i> .

BAB III

METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Dalam menyelesaikan skripsi ini diperlukan data-data serta informasi yang relatif lengkap sebagai bahan yang dapat mendukung kebenaran materi uraian dan pembahasan. Oleh karena itu, sebelum penulisan skripsi ini dilakukan, penulis melakukan riset atau penelitian terlebih dahulu untuk mendapatkan data serta informasi yang terkait.

Adapun metode pengumpulan data yang dilakukan penulis adalah sebagai berikut:

3.1.1 Studi Pustaka

Penulis membaca dan mempelajari buku-buku, artikel/jurnal dari website yang ada kaitannya dengan pengembangan sistem informasi penggajian, antara lain:

1. Al Fatta, Hanif. 2007. *Analisis dan Perancangan Sistem Informasi: untuk Keunggulan Bersaing Perusahaan & Organisasi Modern*. Yogyakarta: Penerbit Andi.
2. Mulyadi. 2001. *Sistem Akuntansi*. Jakarta: Salemba Empat.
3. Whitten, Jeffrey L. 2004. *Metode Desain & Analisis Sistem*. Yogyakarta: Penerbit Andi.

Adapun daftar buku lainnya yang menjadi referensi dapat dilihat dalam daftar pustaka.

3.1.2 Studi Lapangan

1. Observasi

Penulis melakukan observasi pada SMK Al-Firdaus yang dimaksudkan agar dapat mengetahui secara langsung proses penggajian yang dilakukan, meliputi: proses pengelolaan data pegawai, pencatatan absensi, rekap absensi, penghitungan gaji, pembuatan laporan penggajian, pembuatan slip gaji, dan pembayaran gaji. Kegiatan observasi dilakukan pada bulan Januari 2011 di SMK Al-Firdaus yang beralamat di Jl. Peta Utara No. 41 Kalideres, Jakarta Barat 11830.

Adapun hasil observasi terlampir pada lampiran observasi.

2. Wawancara

Wawancara dilakukan dengan cara melakukan berdiskusi dengan pihak yang dapat memberikan informasi mengenai sistem yang berjalan. Penulis melakukan wawancara langsung dengan beberapa pihak seperti Bapak Fahrurrozi selaku salah satu Guru Piket, Ibu Laelatul Badriah selaku Bendahara, dan Bapak Nabhani selaku Kepala Tata Usaha. Wawancara dilakukan beberapa kali yaitu pada tanggal 10 Januari 2011, 17 Januari 2011 di kantor SMK Al-Firdaus.

Adapun daftar pertanyaan terlampir pada lampiran wawancara.

3.1.3 Studi Literatur Sejenis

Penulis mengamati penelitian sebelumnya sebagai acuan dan membandingkan untuk membuat usulan sistem lebih baik. Pengamatan yang penulis lakukan dengan membaca referensi dari beberapa skripsi yang antara lain berjudul *Perancangan Aplikasi Penggajian Karyawan Pada Yayasan Darul Irfan* yang dibuat oleh Mulyadi, dan skripsi yang berjudul *Pengembangan Sistem Informasi Penggajian Karyawan Pada Yayasan Pendidikan Bidayatul Hidayah (YPBH) Bekasi* yang dibuat oleh Yuli Yanah.

Adapun referensi mengenai studi literatur sejenis telah dijelaskan pada akhir BAB II.

3.2 Metode Pengembangan Sistem

Penulis menggunakan metode *Rapid Application Development* (RAD) karena menurut penulis, metode ini merupakan metode yang paling cocok dalam pengembangan dalam pengembangan aplikasi ini karena lebih menekankan pada pembuatan aplikasi/*prototype* dengan melakukan pendekatan kepada *user* atau pengguna sistem dalam mencapai solusi dari permasalahan yang ada.

Pada alur proses RAD, penulis membatasi dengan hanya melalui beberapa tahapan yaitu dari *Scope Definition* sampai dengan tahapan *Construction and Testing, Delivery of a version*, selebihnya diserahkan pada pihak sekolah/yayasan dalam mengelola aplikasi ini.

Adapun dalam tahapan pengembangan sistem ini terdiri dari beberapa aktivitas yang tentunya sesuai dengan tahapan yang sebelumnya telah dijabarkan pada alur proses pengembangan sistem. Tahapan tersebut yaitu:

3.2.1 Mendefinisikan Lingkup (*Scope Definition*)

Pada tahap ini, penulis dan *user* melakukan sejenis pertemuan untuk menggambarkan permasalahan, menentukan ruang lingkup pengembangan sistem, mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan.

3.2.2 Analisis Sistem (*Analysis*)

Pada tahap ini, penulis menjabarkan tentang permasalahan yang terjadi pada sistem yang sedang berjalan dan analisis sistem yang diusulkan untuk memperbaiki sistem yang lama. Penulis menggunakan *rich picture* dalam memvisualisasikan prosedur sistem yang sedang berjalan dan prosedur sistem usulan.

Terdapat tiga tahapan analisis sistem pada alur pengembangan sistem RAD, yaitu:

1. *Problem Analysis*, mempelajari sistem yang ada atau sistem berjalan dengan pemahaman akan masalah-masalah pengembangan sistem.
2. *Requirement Analysis*, mendefinisikan dan memprioritaskan persyaratan-persyaratan bisnis.

3. *Decision Analysis*, menghasilkan arsitektur aplikasi untuk solusi yang disetujui.

3.2.3 Perancangan Sistem (*Design*)

Pada tahap perancangan sistem, *tools* nya menggunakan diagram-diagram UML (*Unified Modelling Language*). Namun tidak semua diagram yang disediakan oleh UML digunakan oleh penulis dalam perancangan sistem ini. Hanya beberapa diagram UML saja yang digunakan oleh penulis, yang menurut penulis dapat mendukung perancangan aplikasi ini. Adapun beberapa diagram berikut ini:

1. *Use Case Diagram*, diagram yang menjelaskan aktivitas apa saja yang dilakukan sistem yang akan dibangun dan siapa saja yang berinteraksi dengan sistem tersebut.
2. *Activity Diagram*, diagram yang menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal sampai bagaimana mereka berakhir.
3. *Sequence Diagram*, diagram yang menjelaskan secara detail urutan proses yang dilakukan oleh sistem untuk mencapai tujuan dari *use case*, interaksi antar *class*, operasi apa saja yang terlibat, urutan antar operasi dan informasi yang diperlukan oleh masing-masing operasi.
4. *Class Diagram*, diagram yang menunjukkan antara *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Selain diagram di atas, digunakan juga diagram lain yang tentunya di luar dari diagram UML, namun perannya cukup penting dalam mendukung perancangan sistem ini, yaitu tabel *database* dan perancangan *interface*.

3.2.4 Implementasi Sistem (*Construction & Testing*)

Setelah melakukan analisis dan perancangan sistem, selanjutnya saatnya sistem untuk diimplementasikan. Pada tahap ini terdapat beberapa aktivitas yang dilakukan. Aktivitas-aktivitas yang dimaksud yaitu:

1. Pemrograman

Pada tahap ini hasil desain dimasukkan ke dalam bentuk bahasa pemrograman yang gunakan agar dapat dijalankan dalam bentuk aplikasi. Adapun *software* yang digunakan dalam pembuatan sistem ini adalah *EditPlus* 2 dan Macromedia Dreamweaver 8 sebagai *software editor*, Adobe Photoshop CS2 untuk mengolah gambar. XAMPP versi 1.7.1. yang mencakup: Apache versi 2.2.11 untuk *web server*, PHP versi 5.2.9 untuk bahasa pemrograman, dan MySQL versi 5.1.33 untuk *database* nya.

2. Perangkat Sistem

Pada tahapan ini melakukan pengaturan perangkat yang dibutuhkan dalam penerapan aplikasi ini agar dapat berjalan sesuai dengan harapkan.

3. Pengujian (*Testing*)

Pada tahap ini dilakukan uji coba terhadap sistem baru agar dapat digunakan tanpa menemukan kendala-kendala apapun. Adapun uji coba yang akan dilakukan menggunakan metode *black-box*.

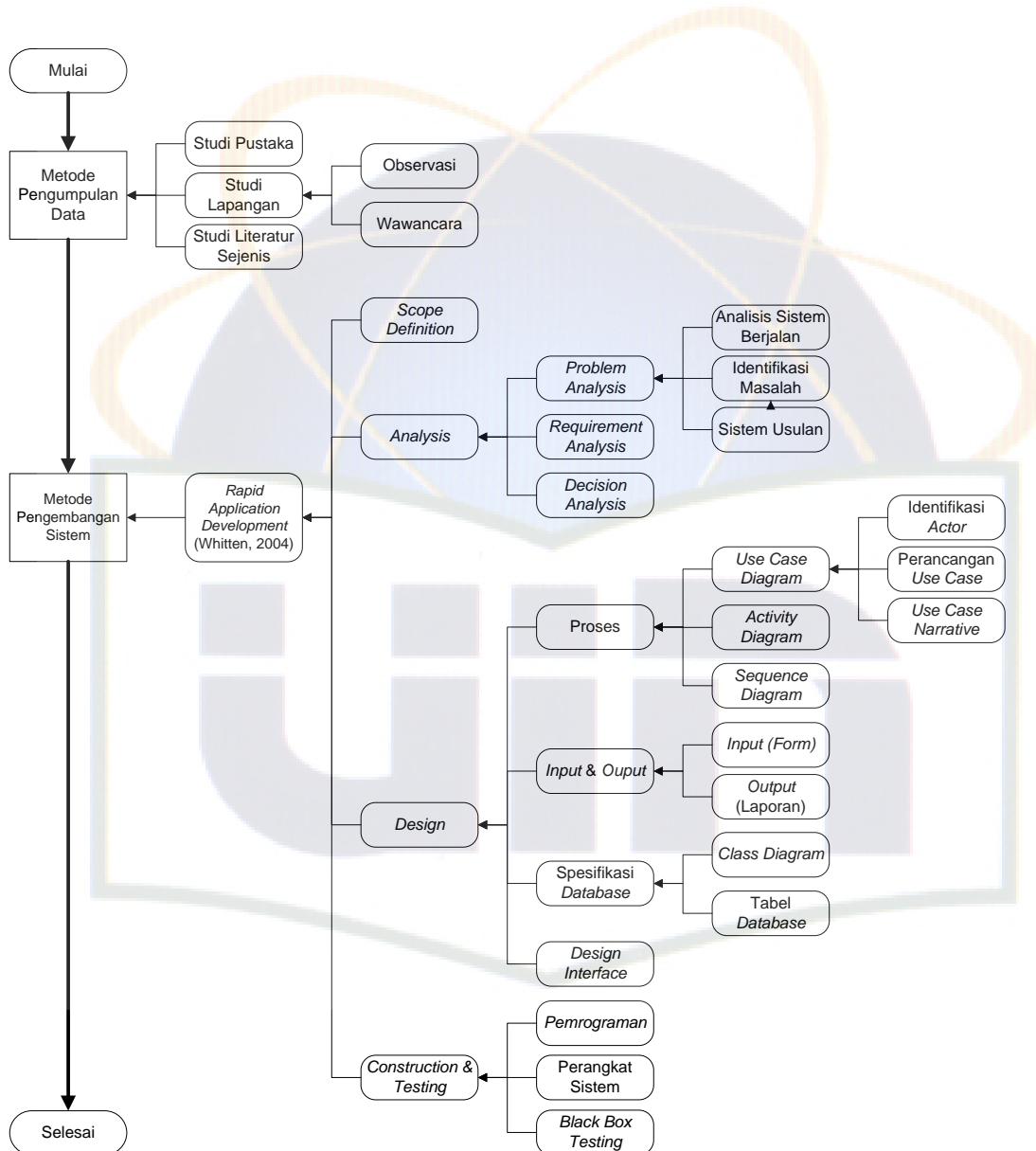
Pada pengujian *black-box testing*, tidak perlu diketahui apa yang sesungguhnya terjadi dalam sistem/perangkat lunak. Karena yang diuji adalah masukan/keluarannya. Dalam pengujian ini kita dapat menggunakan *use case diagram* yang kita kembangkan saat analisis sebagai panduan, apakah keluaran sesuai dengan harapan serta kebutuhan pengguna.

Pengujian secara *black-box*, merupakan pendekatan menguji apakah setiap fungsi di dalam program dapat berjalan dengan benar. Berikut beberapa proses yang dilakukan dalam pengujian ini, yaitu:

1. Fungsi-fungsi yang tidak benar, baik *input* maupun *output*.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database*.

3.3 Kerangka Penelitian

Dalam melakukan penelitian ini, penulis melakukan tahapan-tahapan kegiatan dengan mengikuti rencana kegiatan yang tertuang dalam kerangka penelitian meliputi metode pengumpulan data dan metode pengembangan sistem yang dapat dilihat pada gambar 3.1 berikut:



Gambar 3.1 Kerangka Penelitian

BAB IV

PENGEMBANGAN SISTEM INFORMASI PENGAJIAN

4.1 Profil Sekolah

4.1.1 Gambaran Umum

SMK Al Firdaus berdiri sejak tahun 2006 yang bernaung pada Yayasan Pendidikan Islam Al-Firdaus didirikan oleh bapak H. Asnawi Mursan yang beralamat di Jalan Peta Utara no 41 Kalideres Jakarta Barat, yang disebabkan kebutuhan masyarakat akan pendidikan formal yang berasaskan pendidikan agama semakin bertambah. SMK Al-Firdaus membekali peserta didik dengan kemampuan akademik maupun non akademik untuk dapat berkarir baik sebagai tenaga kerja profesional maupun menjadi seorang *entrepreneur* sesuai bidang yang diminati. Sejak berdirinya sampai saat ini SMK Al Firdaus mengalami perubahan baik dari sarana prasarana, pengajar maupun kurikulumnya.

4.1.2 Logo Sekolah

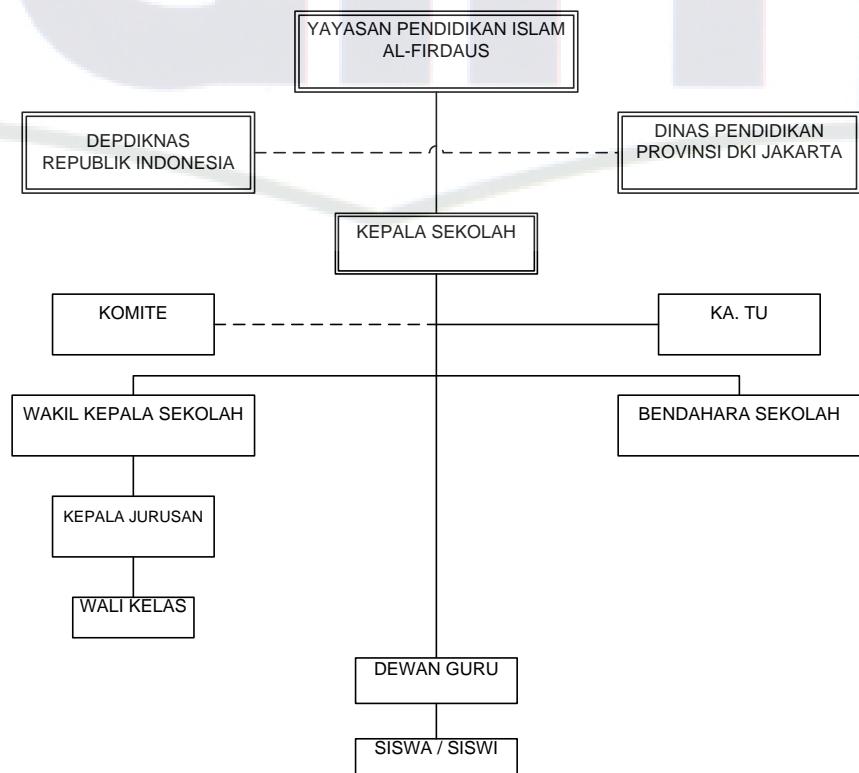


Gambar 4.1 Logo Sekolah

4.1.3 Visi dan Misi

1. Visi: Kompeten dalam prestasi berlandaskan imtaq (iman dan taqwa) dan iptek (ilmu pengetahuan dan teknologi).
2. Misi:
 - a. Menjadikan peserta didik beriman dan berakhhlak mulia.
 - b. Meningkatkan prestasi peserta didik baik akademik maupun non akademik.
 - c. Meningkatkan sikap inovatif, kreatif, dan kompetitif dalam penguasaan iptek.

4.1.4 Struktur Organisasi



Gambar 4.2 Struktur Organisasi (Sumber: SMK Al-Firdaus)

4.2 Mendefinisikan Lingkup (*Scope Definition*)

Untuk lebih memfokuskan penelitian ini, maka penulis akan membatasi permasalahan dan lingkup penelitian pada SMK Al-Firdaus dengan pengembangan sistem informasi pada karena di dalam menangani pengolahan data penggajian butuh pengolahan yang optimal untuk menghasilkan data-data yang akurat di dalam menangani proses penggajian. Perancangan sistem ini membantu pengolahan dan penyimpanan data yang di mulai proses penginputan *file* master data pegawai, mempercepat penghitungan absensi dan gaji agar dapat mendokumentasikan data penggajian dengan rapi dan teratur, guna memudahkan pembacaan hasil laporan penggajian.

4.3 Analisis Sistem (*Analysis*)

4.3.1 Analisa Permasalahan (*Problem Analysis*)

4.3.1.1 Analisa Sistem Berjalan

Berdasarkan hasil observasi dan wawancara yang dilakukan, adapun sistem penggajian yang berjalan adalah sebagai berikut:

1. Waktu Hadir Pegawai

Hari dan Jam Kerja yang diberlakukan dalam sekolah yaitu:

- a. Untuk staf, senin sampai jumat: 06:30 – 14.00
- b. Untuk guru, sesuai waktu mengajar.

Setiap hari staf diharuskan untuk mencatat waktu kehadiran sebanyak 2 (dua) kali yaitu pada saat masuk kerja dan pulang kerja. Namun setiap awal bulan Kepala Tata Usaha mencetak daftar absensi terlebih dahulu. Staf melakukan absensi dengan mengisi daftar absensi. Sedangkan guru mencatat waktu kehadirannya pada saat ingin mengajar dengan mengisi daftar absensi khusus untuk guru yang diawasi oleh guru piker.

2. Gaji Pokok

Setiap guru akan mendapatkan gaji pokok sesuai dengan jumlah waktu mengajar selama satu minggu dihitung untuk periode satu bulan. Gaji pokok sebesar Rp. 20000,- (dua puluh ribu rupiah) per jam mengajar dan bersifat tetap, walaupun sedang libur kegiatan belajar mengajar (KBM) guru akan tetap berhak mendapatkannya.

3. Tunjangan

a. Tunjangan Jabatan

Pegawai yang mendapatkan tunjangan jabatan adalah pegawai yang memiliki jabatan tertentu.

b. Tunjangan Transportasi

Penggantian uang transportasi diberikan sebesar Rp. 5000,- (lima ribu rupiah) per jam mengajar. Tunjangan transportasi hanya diberikan jika guru masuk mengajar. Jadi jika guru tidak masuk mengajar

atau KBM sedang libur maka tidak berhak mendapatkan tunjangan ini. Jumlah kehadiran guru akan diakumulasikan pada akhir bulan dan pembayaran tunjangan ini dilakukan bersamaan dengan pembayaran gaji setiap bulannya.

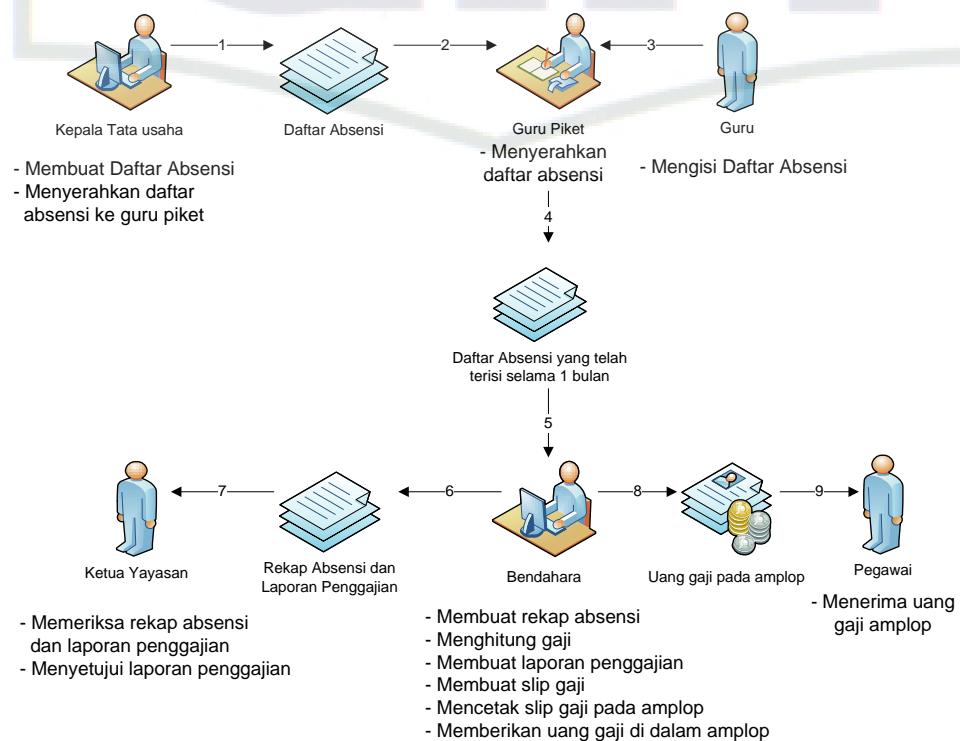
c. Tunjangan Guru Piket

Setiap guru yang mendapatkan tugas menjadi guru piket akan mendapatkan tunjangan sebesar Rp. 20000,- (dua puluh ribu rupiah) per hari. Biasanya yang bertugas sebagai guru piket merupakan guru yang tidak ada waktu mengajar di hari tersebut, tiap hari nya hanya satu guru piket. Pada periode satu minggu KBM terdapat lima guru piket yang berbeda, namun ke lima orang guru piket sudah ditetapkan bertugas untuk periode satu tahun akademik. Jumlah hari guru piket yang bertugas akan diakumulasikan pada akhir bulan dan pembayaran tunjangan ini dilakukan bersamaan dengan pembayaran gaji setiap bulannya.

Setiap pada tanggal 1 di bulan berikutnya, bendahara mulai membuat rekap absensi dihitung secara *manual* dari daftar absensi dan kemudian memindahkannya ke *Microsoft Excel*. Pada tanggal 4, bendahara harus melengkapi rekap absensi. Penghitungan gaji dimulai setelah pembuatan rekap

absensi selesai, dan masih dihitung secara *manual* kemudian memindahkannya ke *Microsoft Excel* untuk dibuatkan laporan penggajian. Setiap tanggal 8, bendahara harus selesai membuat laporan penggajian. Kemudian rekap absensi dan laporan penggajian dicetak dan diserahkan ke ketua yayasan untuk diperiksa dan mendapatkan persetujuan. Setelah disetujui, dilakukanlah pembuatan slip gaji dengan mencatat kembali dari laporan penggajian menggunakan *Microsoft Excel* kemudian di cetak pada amplop. Pembayaran gaji dilakukan setiap tanggal 10 dengan memberikan uang gaji dalam amplop.

Langkah-langkah pada keterangan di atas dapat dilihat pada *rich picture* berikut:



Gambar 4.3 Rich Picture Sistem Berjalan

4.3.1.2 Identifikasi Masalah

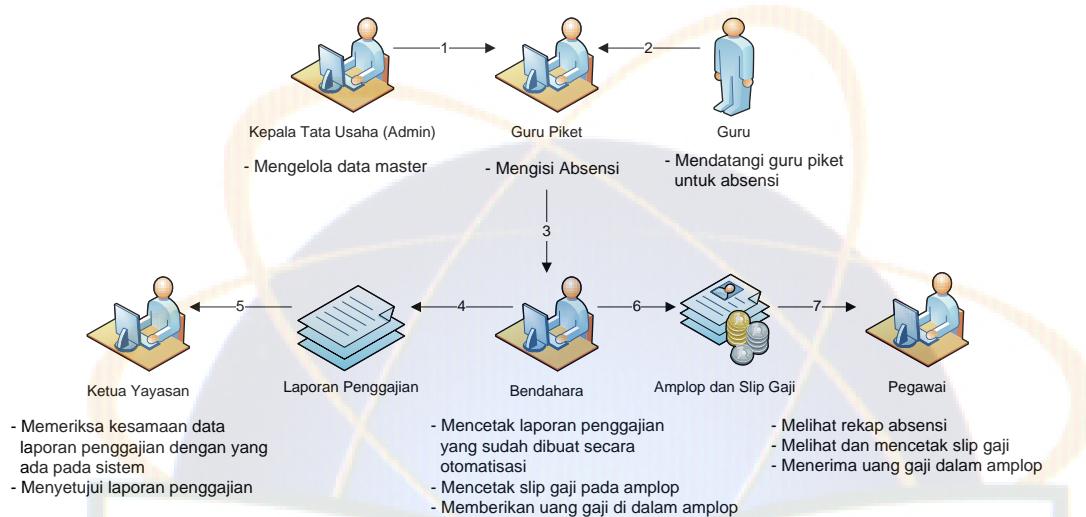
Di dalam mengidentifikasi masalah, peneliti menganalisa sistem yang berjalan. Peneliti menemukan masalah yang terjadi pada pengolahan penggajian yaitu proses penghitungan absensi untuk pembuatan rekap absensi, proses penghitungan, dan pembuatan laporan penggajian dan slip gaji masih bersifat manual yaitu dibuat pada kertas lalu dipindahkan ke *Microsoft Excel*. Sehingga membutuhkan ketelitian yang tinggi untuk melakukan transaksi penggajian dan membutuhkan waktu yang tidak sebentar dalam proses penghitungan absensi dan gaji karena data tersebut yang berhubungan dengan penggajian namun tidak saling terintegrasi.

Untuk memudahkan pihak SMK Al-Firdaus, peneliti membuat program sistem penggajian dan peneliti menggunakan UML untuk merancang sistem ini.

4.3.1.3 Sistem Usulan

Pada sistem usulan ini, penulis memberikan informasi mengenai rancangan sistem yang penulis ajukan untuk mengatasi permasalahan-permasalahan sistem yang telah disebutkan sebelumnya.

Adapun prosedur dari perancangan yang diusulkan adalah sebagai berikut:



Gambar 4.4 Rich Picture Sistem Usulan

Sistem informasi penggajian yang dirancang mendukung tugas dari pegawai dalam sekolah.

1. Kepala Tata Usaha menggunakan sistem ini untuk mengelola *file* data master.
2. Guru Piket menggunakan sistem ini untuk mengisi kehadiran (absensi) bagi guru yang ingin mengajar.
3. Bendahara menggunakan sistem ini untuk melihat dan mencetak laporan penggajian dan slip gaji, sebab penghitungan absensi dan gaji sudah terotomatisasi sistem.
4. Ketua Yayasan menggunakan sistem ini untuk melihat *file* data pegawai, rekap absensi, dan laporan penggajian sebagai sarana dalam memeriksa laporan dan memonitor transaksi penggajian.
5. Pegawai menggunakan sistem ini untuk melihat rekap absensinya, serta melihat dan mencetak slip gajinya.

4.3.1.4 Analisa Perbandingan Sistem

Perbandingan sistem berjalan dengan sistem usulan diklasifikasikan berdasarkan kekurangan dan kelebihan dapat dilihat pada tabel 4.1.

Tabel 4.1 Analisa Perbandingan Sistem

Keterangan	Kekurangan	Kelebihan
Sistem Berjalan	<ul style="list-style-type: none"> - Absensi masih dicatat pada daftar absensi. - Penghitungan absensi masih <i>manual</i>. - Pembuatan rekap absensi menggunakan <i>Microsoft Excel</i>. - Penghitungan gaji masih <i>manual</i>. - Pembuatan laporan penggajian menggunakan <i>Microsoft Excel</i>. - Proses laporan ke ketua yayasan masih <i>manual</i>. - Pembuatan slip gaji menggunakan <i>Microsoft Excel</i>. 	
Sistem Usulan		<ul style="list-style-type: none"> - Absensi sudah terkomputerisasi dan langsung tersimpan ke <i>database</i>. - Penghitungan absensi dan pembuatan rekap absensi sudah terotomatisasi sistem. - Penghitungan gaji dan pembuatan laporan penggajian sudah terotomatisasi sistem. - Proses laporan ke ketua yayasan dapat dicocokkan dengan data yang ada pada sistem. - Pembuatan format slip gaji sudah terotomatisasi. - Pegawai dapat melihat rekap absensi, serta melihat dan mencetak slip gaji.

4.3.2 Analisa Persyaratan (*Requirement Analysis*)

Tahap ini mendefinisikan dan menganalisis persyaratan-persyaratan sistem yang mendukung aktivitas pendokumentasian. Tahap ini bertujuan untuk menentukan apa yang dapat dilakukan oleh sistem dalam membantu proses penghitungan absensi dan gaji menjadi lebih efisien dan efektif.

Requirements yang ada akan dibagi menjadi 2 (dua) bagian.

Bagian yang pertama adalah *Functional Requirement* yaitu aktivitas dan *service* yang harus disediakan oleh sistem yang dikembangkan.

Bagian kedua adalah *Nonfunctional Requirement* yaitu fitur-fitur lain yang diperlukan oleh sistem agar dapat lebih memuaskan. Berikut adalah *requirements* dari sistem informasi penggajian.

4.3.2.1 *Functional Requirement*

Sistem yang dikembangkan harus mempunyai *functional requirements* sebagai berikut:

1. Kepala Tata Usaha menggunakan sistem ini untuk mengelola *file* data master, seperti data *user*, data jabatan, data jurusan, data honor, data pegawai, data tahun pelajaran, serta data waktu mengajar.
2. Guru Piket menggunakan sistem ini untuk mengisi absensi mengajar, dan melihat rekap absensi.

3. Bendahara menggunakan sistem ini untuk melihat rekap absensi, melihat dan mencetak laporan penggajian dan slip gaji.
4. Ketua Yayasan menggunakan sistem ini untuk melihat data pegawai, melihat rekap absensi, dan melihat laporan penggajian sebagai sarana dalam memeriksa laporan dan memonitor transaksi penggajian.
5. Pegawai menggunakan sistem ini untuk melihat rekap absensinya, serta melihat dan mencetak slip gajinya.

4.3.2.2 Nonfunctional Requirement

Berdasarkan hasil perbandingan sistem berjalan dengan sistem usulan. *Nonfunctional requirements* dari sistem yang dikembangkan sebagai berikut:

Tabel 4.2 Nonfunctional Requirement

Jenis Kebutuhan	Penjelasan
Model Tampilan (<i>Performance</i>)	<ul style="list-style-type: none"> a) Mengurangi tingkat kesalahan penghitungan absensi dan gaji. b) Membantu penyusunan rekap absensi dan laporan penggajian. c) Tampilan <i>interface</i> yang menarik dan lebih <i>user friendly</i> sehingga lebih mudah digunakan oleh <i>user</i>.
Model Penyimpanan Data (<i>Information</i>)	<ul style="list-style-type: none"> a) Mencegah terjadinya <i>redundancy</i> data. b) Mencegah hilangnya data-data penggajian yang disebabkan karena

	<p>banyaknya transaksi penggajian yang dilakukan.</p> <p>c) <i>Format</i> penyajian laporan dibuat sehingga lebih mudah dipahami.</p> <p>d) Data terdokumentasi dan terstruktur.</p>
Model Segi Ekonomi (<i>Economic</i>)	<p>a) Mengurangi biaya yang disebabkan penyediaan media penyimpanan yang masih berupa kertas.</p> <p>b) Memperlancar aliran informasi dari setiap <i>user</i>.</p>
Model Pengontrolan Sistem (<i>Control</i>)	<p>a) Mencegah akses dari pengguna yang tidak berwenang.</p>
Model Efisiensi Sistem (<i>Efficiency</i>)	<p>a) Mengefisienkan waktu penghitungan absensi dan gaji.</p> <p>b) Mengefisienkan waktu apa bila laporan segera dibutuhkan.</p>
Model Keakuratan Sistem (<i>Accuracy</i>)	<p>a) Penghitungan terotomatisasi oleh sistem.</p>
Model Keefektifan Sistem (<i>Effectiveness</i>)	<p>a) Kegiatan absensi dan penggajian langsung menggunakan sistem pada komputer.</p>
Model Pelayanan Sistem (<i>Service</i>)	<p>a) Menghasilkan informasi yang akurat untuk bahan evaluasi.</p> <p>b) Memberikan kemudahan dalam penggunaan operasional sistem.</p>

4.3.3 Analisa Keputusan (*Decision Analysis*)

Dari tahapan analisis sebelumnya telah diketahui permasalahan dari sistem berjalan, persyaratan dan kebutuhan yang diinginkan, maka tahapan selanjutnya adalah analisis keputusan yaitu menentukan komponen-komponen dari sistem usulan yang akan dirancang, dibangun, dan diimplementasikan. Berikut merupakan komponen-komponen yang dibutuhkan:

a. Data Master

Proses perekaman data-data yang berhubungan dengan penggajian, yaitu data pegawai, data absensi, data gaji. Seluruh data ini terkait pada proses penghitungan penggajian.

b. Menentukan hak akses (*user*)

Dalam hal ini menentukan hak akses masing-masing pengguna. Setiap pengguna hanya dapat mengakses fitur sesuai masing-masing bagiannya.

Setelah mengetahui komponen-komponen sistem yang diusulkan selanjutnya adalah menentukan jenis perangkat sistem yaitu berupa *tools* atau alat untuk merancang dan mengimplementasikan sistem usulan sehingga menghasilkan arsitektur sistem usulan yang terpenting adalah pemahaman terhadap jenis *tools* yang akan digunakan karena harus sesuai dengan kebutuhan pengguna dan fungsi-fungsi sistem yang terdapat didalamnya.

Di dalam sistem yang lama, proses pengolahan datanya dilakukan secara manual, sedangkan pada sistem informasi yang akan

dikembangkan adalah sistem informasi terkomputerisasi dan konsep pengembangan aplikasi ini menggunakan pemrograman berorientasi obyek. Sistem usulan dirancang dengan menggunakan UML (*Unified Modelling Language*). Sedangkan bahasa pemrograman PHP dan mengimplementasikan sistem *database* menggunakan MySQL.

4.4 Perancangan Sistem (*Design*)

Pada tahapan ini, penulis memberikan informasi mengenai rancangan sistem yang penulis ajukan guna menanggulangi permasalahan-permasalahan sistem yang telah disebutkan sebelumnya.

4.4.1 Use Case Model Diagram

4.4.1.1 Identifikasi Actor

Adapun *actor* yang terlibat dalam sistem informasi penggajian ini dapat diklasifikasikan menjadi 5 kategori, yaitu Kepala Tata Usaha, Guru Piket, Bendahara, Ketua Yayasan, dan Pegawai. Untuk lebih jelas peran-peran *actor* yang ada dalam sistem dapat dilihat pada tabel 4.3 identifikasi *actor*.

Tabel 4.3 Identifikasi *Actor*

<i>Actor</i>	<i>Description</i>
Kepala Tata Usaha	Kepala Tata Usaha bertanggung jawab dalam mengelola data master seperti data pegawai, data gaji.
Guru piket	Guru Piket bertanggung jawab mengisi absensi mengajar.
Bendahara	Bendahara bertanggung jawab dalam pembuatan laporan penggajian dan slip gaji, serta pembayaran gaji.

Ketua Yayasan	Ketua Yayasan bertanggung jawab memeriksa rekap absensi dan laporan penggajian.
Pegawai	Pegawai bertanggung jawab melihat rekap absensinya, serta melihat dan mencetak slip gajinya.

4.4.1.2 Identifikasi Use Case

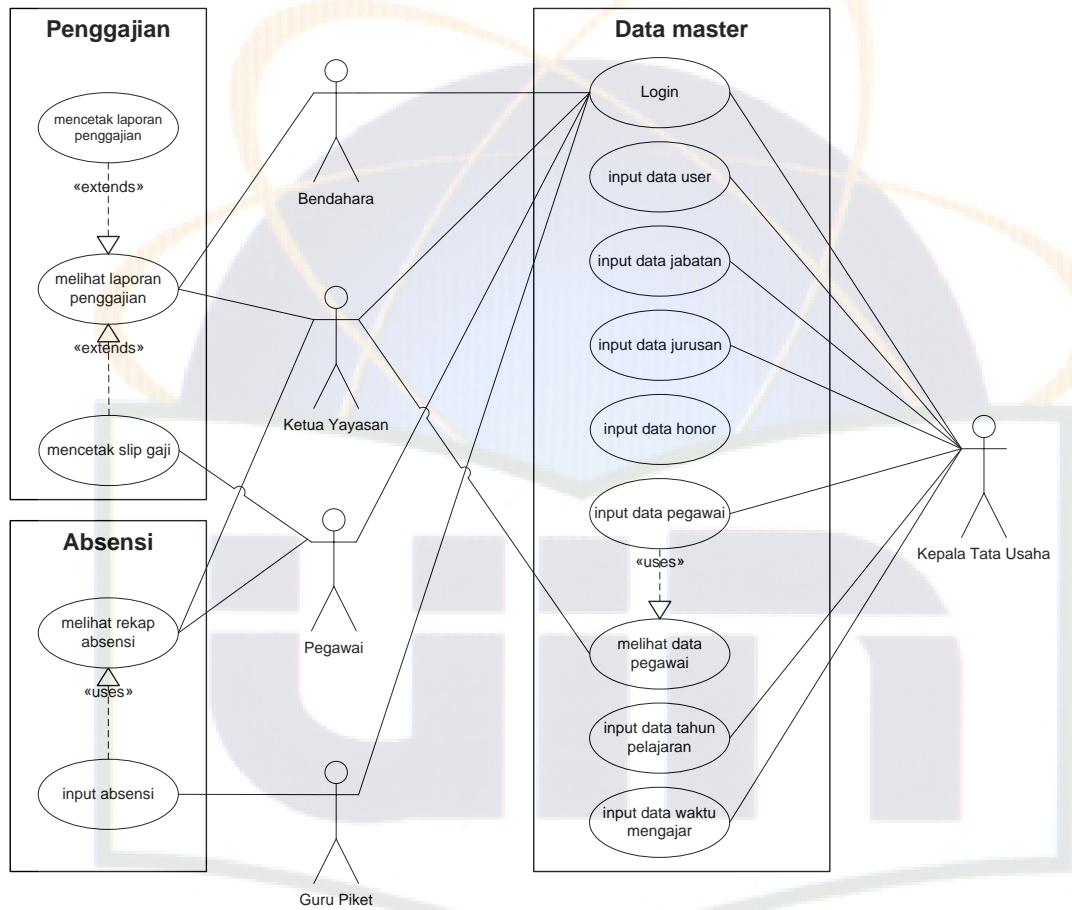
Setelah *actor* teridentifikasi, selanjutnya mengidentifikasi *use case*. Tabel 4.4 menggambarkan secara grafis lingkup dari tiap *use case*.

Tabel 4.4 Identifikasi *Use Case*

No	Use Case Name	Description	Actor
1	Login	<i>Use case</i> menggambarkan kegiatan <i>login</i> ke dalam sistem dengan memasukkan <i>username</i> dan <i>password</i> untuk masuk ke halaman utama.	Kepala Tata Usaha, Guru Piket, Bendahara, Ketua Yayasan, Pegawai
2	<i>Input data user</i>	<i>Use case</i> ini menggambarkan kegiatan <i>menginput</i> data <i>user</i> .	Kepala Tata Usaha
3	<i>Input data jabatan</i>	<i>Use case</i> ini menggambarkan kegiatan <i>menginput</i> data jabatan.	Kepala Tata Usaha
4	<i>Input data jurusan</i>	<i>Use case</i> ini menggambarkan kegiatan <i>menginput</i> data jurusan.	Kepala Tata Usaha
5	<i>Input data pegawai</i>	<i>Use case</i> ini menggambarkan kegiatan pada kegiatan <i>menginput</i> data pegawai.	Kepala Tata Usaha
6	Melihat data pegawai	<i>Use case</i> ini menggambarkan kegiatan melihat data pegawai.	Kepala Tata Usaha, Ketua Yayasan
7	<i>Input data honor</i>	<i>Use case</i> ini menggambarkan kegiatan <i>menginput</i> data honor.	Kepala Tata Usaha
8	<i>Input data tahun</i>	<i>Use case</i> ini menggambarkan	Kepala Tata Usaha

	pelajaran	kegiatan menginput data tahun pelajaran.	
9	<i>Input</i> data waktu mengajar	<i>Use case</i> ini menggambarkan kegiatan menginput data waktu mengajar.	Kepala Tata Usaha
10	Mengisi absensi	<i>Use case</i> ini menggambarkan kegiatan mengisi absensi.	Guru Piket
11	Melihat rekap absensi	<i>Use case</i> ini menggambarkan kegiatan melihat rekap absensi.	Kepala Tata Usaha, Guru Piket, Bendahara, Ketua Yayasan, Pegawai
12	Melihat laporan penggajian	<i>Use case</i> ini menggambarkan kegiatan melihat laporan penggajian.	Bendahara, Ketua yayasan
13	Mencetak laporan penggajian	<i>Use case</i> ini menggambarkan kegiatan mencetak laporan penggajian.	Bendahara
14	Mencetak slip gaji	<i>Use case</i> ini menggambarkan kegiatan mencetak slip gaji.	Bendahara, Pegawai

4.4.1.3 Perancangan Use Case Diagram



Gambar 4.5 Use Case Diagram

4.4.1.4 Use Case Narrative

Use Case Narrative merupakan pemaparan naratif penjelasan tentang kegiatan yang dilakukan oleh *actor* dan respon yang diberikan oleh sistem sesuai dengan yang terjadi pada perangkat lunak sistem informasi penggajian.

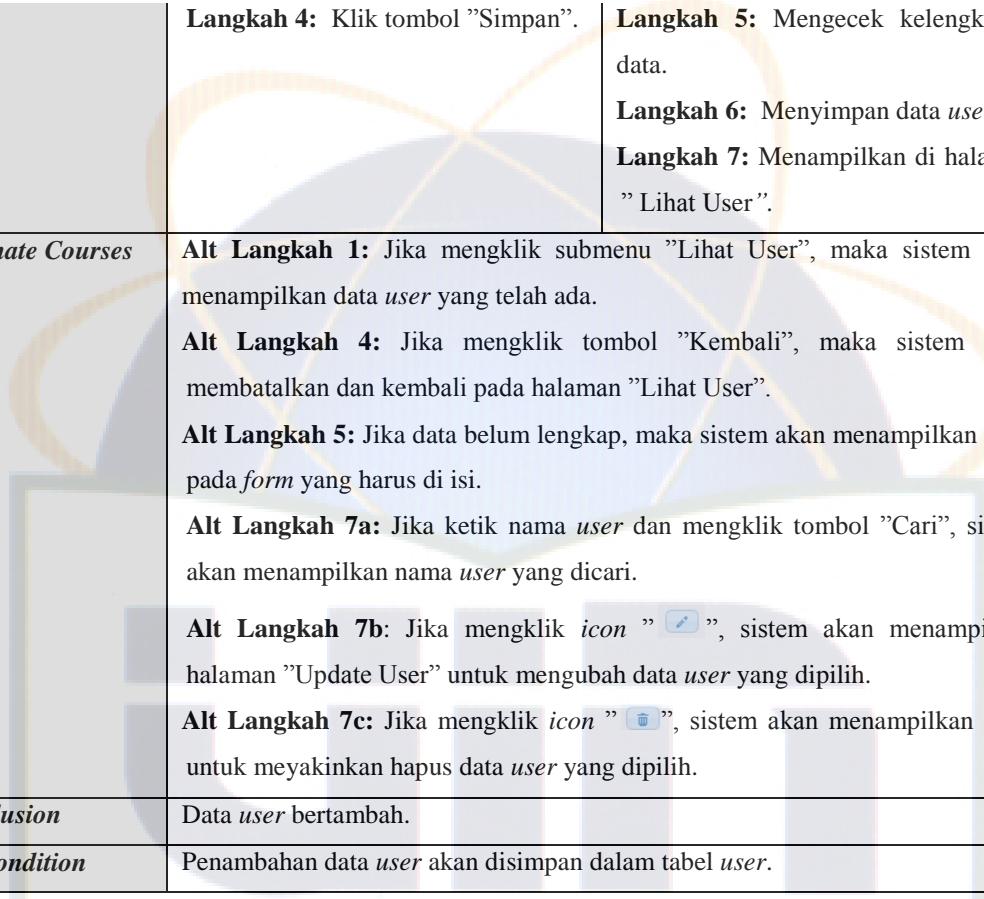
Tabel 4.5 s/d Tabel 4.18 merupakan *use case narrative* dari sistem informasi penggajian.

Tabel 4.5 Use Case Narrative Login

Use Case Name	Login	
Use Case ID	1	
Actor(s)	Ketua Tata Usaha, Guru Piket, Bendahara, Ketua Yayasan, Pegawai	
Description	<i>Use case</i> menggambarkan kegiatan <i>login</i> ke dalam sistem dengan memasukkan <i>username</i> dan <i>password</i> untuk masuk ke halaman utama.	
Precondition	Belum <i>login</i>	
Trigger	<i>Use case</i> ini dilakukan setelah <i>actor</i> membuka <i>browser</i> dan mengetik <i>url</i> http://localhost/penggajian-web-app	
Typical Course of Events	Actor Action Langkah 1: Mengisikan <i>username</i> , <i>password</i> , dan level. Langkah 2: Klik tombol "Login".	System Response Langkah 3: Validasi <i>username</i> , <i>password</i> dan level. Langkah 4: Menampilkan halaman utama <i>authenticated user</i> .
Alternate Courses	Alt Langkah 2: Jika mengklik tombol "Reset", sistem akan membatalkan dan mengkosongkan kembali isian <i>username</i> dan <i>password</i> . Alt Langkah 3: Jika <i>username</i> , <i>password</i> dan level salah, sistem menampilkan pesan kesalahan dan kembali ke halaman <i>login</i> .	
Conclusion	<i>User</i> berhasil masuk ke dalam sistem penggajian.	
Postcondition	<i>User</i> bisa menggunakan sistem.	

Tabel 4.6 Use Case Narrative Input Data User

Use Case Name	<i>Input data user</i>	
Use Case ID	2	
Actor(s)	Kepala Tata Usaha	
Description	<i>Use case</i> ini menggambarkan kegiatan menginput data <i>user</i> .	
Precondition	<i>Login</i> sebagai Kepala Tata Usaha.	
Trigger	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem dan memilih menu data master.	
Typical Course of Events	Actor Action Langkah 1: Memilih submenu "Data User", lalu mengklik submenu "Tambah User". Langkah 3: Mengisikan data <i>user</i> .	System Response Langkah 2: Sistem menampilkan halaman "Tambah User".

	<p>Langkah 4: Klik tombol "Simpan".</p> 	<p>Langkah 5: Mengecek kelengkapan data.</p> <p>Langkah 6: Menyimpan data <i>user</i>.</p> <p>Langkah 7: Menampilkan di halaman "Lihat User".</p>
<i>Alternate Courses</i>	<p>Alt Langkah 1: Jika mengklik submenu "Lihat User", maka sistem akan menampilkan data <i>user</i> yang telah ada.</p> <p>Alt Langkah 4: Jika mengklik tombol "Kembali", maka sistem akan membatalkan dan kembali pada halaman "Lihat User".</p> <p>Alt Langkah 5: Jika data belum lengkap, maka sistem akan menampilkan <i>alert</i> pada <i>form</i> yang harus diisi.</p> <p>Alt Langkah 7a: Jika ketik nama <i>user</i> dan mengklik tombol "Cari", sistem akan menampilkan nama <i>user</i> yang dicari.</p> <p>Alt Langkah 7b: Jika mengklik icon  , sistem akan menampilkan halaman "Update User" untuk mengubah data <i>user</i> yang dipilih.</p> <p>Alt Langkah 7c: Jika mengklik icon  , sistem akan menampilkan <i>alert</i> untuk menyakinkan hapus data <i>user</i> yang dipilih.</p>	
<i>Conclusion</i>	Data <i>user</i> bertambah.	
<i>Postcondition</i>	Penambahan data <i>user</i> akan disimpan dalam tabel <i>user</i> .	

Tabel 4.7 Use Case Narrative Input Data Jabatan

<i>Use Case Name</i>	<i>Input</i> data jabatan	
<i>Use Case ID</i>	3	
<i>Actor(s)</i>	Kepala Tata Usaha	
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan menginput data jabatan.	
<i>Precondition</i>	<i>Login</i> sebagai Kepala Tata Usaha.	
<i>Trigger</i>	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem dan memilih menu data master.	
<i>Typical Course of Events</i>	<p><i>Actor Action</i></p> <p>Langkah 1: Memilih submenu "Data Jabatan", lalu klik submenu "Tambah Jabatan".</p> <p>Langkah 3: Mengisikan data jabatan.</p> <p>Langkah 4: Klik tombol "Simpan".</p>	<p><i>System Response</i></p> <p>Langkah 2: Menampilkan halaman "Tambah Jabatan".</p> <p>Langkah 5: Mengecek kelengkapan data.</p> <p>Langkah 6: Menyimpan data jabatan.</p>

	Langkah 7: Menampilkan di halaman "Lihat Jabatan".
Alternate Courses	<p>Alt Langkah 1: Jika mengklik submenu "Lihat Jabatan", maka sistem menampilkan data jabatan yang telah ada.</p> <p>Alt Langkah 4: Jika mengklik tombol "Kembali", maka sistem akan membatalkan dan kembali pada halaman "Lihat Jabatan".</p> <p>Alt Langkah 5: Jika data belum lengkap, maka sistem akan menampilkan <i>alert</i> pada <i>form</i> yang belum diisi.</p> <p>Alt Langkah 7a: Jika ketik nama jabatan dan mengklik tombol "Cari", sistem akan menampilkan nama jabatan yang dicari.</p> <p>Alt Langkah 7b: Jika mengklik icon  , sistem akan menampilkan halaman "Update Jabatan" untuk ubah data jabatan yang telah ada.</p> <p>Alt Langkah 7c: Jika mengklik icon  , sistem akan menampilkan <i>alert</i> untuk meyakinkan hapus data jabatan yang dipilih.</p>
Conclusion	Data jabatan bertambah.
Postcondition	Penambahan data jabatan akan disimpan dalam tabel jabatan.

Tabel 4.8 Use Case Narrative Input Data Jurusan

Use Case Name	<i>Input</i> data jurusan	
Use Case ID	4	
Actor(s)	Ketua Tata Usaha	
Description	<i>Use case</i> ini menggambarkan kegiatan meng <input data-bbox="632 1275 663 1320"/> data jurusan.	
Precondition	<i>Login</i> sebagai Kepala Tata Usaha.	
Trigger	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem dan memilih menu data master.	
Typical Course of Events	Actor Action <p>Langkah 1: Memilih submenu "Data Jurusan", lalu klik submenu "Tambah Jurusan".</p> <p>Langkah 3: Mengisikan data jurusan</p> <p>Langkah 4: Klik tombol "Simpan".</p>	System Response <p>Langkah 2: Menampilkan halaman "Tambah Jurusan".</p> <p>Langkah 5: Mengecek kelengkapan data.</p> <p>Langkah 6: Menyimpan data jurusan.</p> <p>Langkah 7: Menampilkan di halaman "Lihat Jurusan".</p>

<i>Alternate Courses</i>	<p>Alt Langkah 1: Jika mengklik submenu "Lihat Jurusan", maka sistem akan menampilkan data jurusan yang telah ada.</p> <p>Alt Langkah 4: Jika mengklik tombol "Kembali", maka sistem akan membatalkan dan kembali pada halaman "Lihat Jurusan".</p> <p>Alt Langkah 5: Jika data belum lengkap, maka sistem akan menampilkan <i>alert</i> pada <i>form</i> yang belum diisi.</p> <p>Alt Langkah 7a: Jika ketik nama jurusan dan mengklik tombol "Cari", sistem akan menampilkan nama jurusan yang dicari.</p> <p>Alt Langkah 7b: Jika mengklik icon , sistem akan menampilkan halaman "Update Jurusan" untuk nengubah data jurusan yang dipilih.</p> <p>Alt Langkah 7c: Jika mengklik icon , sistem akan menampilkan <i>alert</i> untuk meyakinkan hapus data jurusan yang dipilih.</p>
<i>Conclusion</i>	Data jurusan bertambah.
<i>Postcondition</i>	Penambahan data jurusan akan disimpan dalam tabel jurusan.

Tabel 4.9 Use Case Narrative Input Data Honor

<i>Use Case Name</i>	<i>Input data honor</i>	
<i>Use Case ID</i>	5	
<i>Actor(s)</i>	Kepala Tata Usaha	
<i>Description</i>	<i>Use case</i> ini menggambarkan kegiatan meng <input data-bbox="555 1167 595 1201"/> data honor.	
<i>Precondition</i>	<i>Login</i> sebagai Kepala Tata Usaha.	
<i>Trigger</i>	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem dan memilih menu data master.	
<i>Typical Course of Events</i>	<i>Actor Action</i>	<i>System Response</i>
	<p>Langkah 1: Memilih submenu "Data Honor".</p> <p>Langkah 3: Mengisikan data honor</p> <p>Langkah 4: Klik tombol "Simpan".</p>	<p>Langkah 2: Menampilkan halaman "Isi Honor".</p> <p>Langkah 5: Mengecek kelengkapan data.</p> <p>Langkah 6: Menyimpan data honor.</p> <p>Langkah 7: Menampilkan halaman "Lihat Honor".</p>
<i>Alternate Courses</i>	<p>Alt Langkah 4: Jika mengklik tombol "Kembali", maka sistem akan membatalkan.</p> <p>Alt Langkah 5: Jika data belum lengkap, maka sistem akan menampilkan <i>alert</i> pada <i>form</i> yang belum diisi.</p>	

	Alt Langkah 7a: Jika mengklik icon "edit", sistem akan menampilkan halaman "Update Honor" untuk mengubah data honor.
Conclusion	Data honor terisi.
Postcondition	Pengisian data honor akan disimpan dalam tabel honor.

Tabel 4.10 Use Case Narrative Input Data Pegawai

Use Case Name	<i>Input data pegawai</i>	
Use Case ID	6	
Actor(s)	Kepala Tata Usaha	
Description	<i>Use case ini menggambarkan kegiatan pada kegiatan menginput data pegawai.</i>	
Precondition	<i>Login sebagai Kepala Tata Usaha.</i>	
Trigger	<i>Use case ini dilakukan setelah dilakukan setelah melakukan login ke dalam sistem, serta data user, data jabatan, data jurusan, dan data honor telah terisi.</i>	
Typical Course of Events	Actor Action Langkah 1: Memilih menu "Data Pegawai", lalu klik submenu "Tambah Pegawai". Langkah 3: Mengisikan data pegawai. Langkah 4: Klik tombol "Simpan".	System Response Langkah 2: Menampilkan halaman "Tambah Pegawai". Langkah 5: Mengecek kelengkapan data. Langkah 6: Menyimpan data pegawai. Langkah 7: Menampilkan di halaman "Lihat Pegawai".
Alternate Courses	Alt Langkah 1: Jika mengklik submenu "Lihat Pegawai", maka sistem akan menampilkan data pegawai yang telah ada. Alt Langkah 4: Jika mengklik tombol "Kembali", maka sistem akan membatalkan dan kembali pada halaman "Lihat Pegawai". Alt Langkah 5: Jika data belum lengkap, maka sistem akan menampilkan <i>alert</i> pada <i>form</i> yang belum di isi. Alt Langkah 7a: Jika ketik id pegawai atau nama pegawai dan mengklik tombol "Cari", sistem akan menampilkan id pegawai atau nama pegawai yang dicari. Alt Langkah 7b: Jika mengklik icon "edit", sistem akan menampilkan halaman "Update Pegawai" untuk mengubah data pegawai yang dipilih.	

	Alt Langkah 7c: Jika mengklik icon ”  ”, sistem akan menampilkan <i>alert</i> untuk meyakinkan hapus data pegawai yang dipilih.
Conclusion	Data pegawai bertambah.
Postcondition	Penambahan data pegawai akan disimpan dalam data pegawai.

Tabel 4.11 Use Case Narrative Melihat Data Pegawai

Use Case Name	Melihat data pegawai	
Use Case ID	7	
Actor(s)	Kepala Tata Usaha, Ketua Yayasan	
Description	<i>Use case</i> ini menggambarkan kegiatan melihat data pegawai.	
Precondition	<i>Login</i> sebagai Kepala Tata Usaha atau Ketua Yayasan.	
Trigger	<i>Use case</i> ini dilakukan setelah dilakukan setelah dilakukan setelah melakukan <i>login</i> ke dalam sistem, serta data pegawai telah terisi.	
Typical Course of Events	Actor Action Langkah 1: Memilih menu ”Data Pegawai”. Langkah 3: Mengisikan <i>form</i> nama pegawai. Langkah 4: Klik tombol ”Cari”	System Response Langkah 2: Menampilkan halaman ”Lihat Pegawai”. Langkah 5: Menampilkan nama pegawai yang dicari.
Alternate Courses	-	
Conclusion	Lihat data pegawai.	
Postcondition	Data pegawai ditampilkan.	

Tabel 4.12 Use Case Narrative Input Data Tahun Pelajaran

Use Case Name	<i>Input</i> data tahun pelajaran	
Use Case ID	8	
Actor(s)	Kepala Tata Usaha	
Description	<i>Use case</i> ini menggambarkan kegiatan meng <input data-bbox="568 1657 600 1693"/> data tahun pelajaran.	
Precondition	<i>Login</i> sebagai Kepala Tata Usaha.	
Trigger	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem	
Typical Course of Events	Actor Action Langkah 1: Memilih menu ”Tahun Pelajaran”, lalu klik submenu ”Tambah Tahun Pelajaran”.	System Response Langkah 2: Menampilkan halaman ”Tambah Tahun Pelajaran”.

	<p>Langkah 3: Mengisikan data tahun pelajaran</p> <p>Langkah 4: Klik tombol "Simpan".</p>	<p>Langkah 5: Mengecek kelengkapan data.</p> <p>Langkah 6: Menyimpan data tahun pelajaran.</p> <p>Langkah 7: Menampilkan di halaman "Lihat Tahun Pelajaran".</p>
<i>Alternate Courses</i>	<p>Alt Langkah 1: Jika mengklik submenu "Lihat Tahun Pelajaran", maka sistem akan menampilkan data tahun pelajaran.</p> <p>Alt Langkah 4: Jika mengklik tombol "Kembali", maka sistem akan membatalkan dan kembali pada halaman "Lihat Tahun Pelajaran".</p> <p>Alt Langkah 5: Jika data belum lengkap, maka sistem akan menampilkan <i>alert</i> pada <i>form</i> yang belum di isi.</p> <p>Alt Langkah 7a: Jika ketik tahun pelajaran dan mengklik tombol "Cari", sistem akan menampilkan tahun pelajaran yang dicari.</p> <p>Alt Langkah 7b: Jika mengklik icon , sistem akan menampilkan halaman "Update Tahun Pelajaran" untuk mengubah data tahun pelajaran yang dipilih.</p> <p>Alt Langkah 7c: Jika mengklik icon , sistem akan menampilkan <i>alert</i> untuk menyakinkan hapus data tahun pelajaran yang dipilih.</p>	
<i>Conclusion</i>	Data tahun pelajaran bertambah.	
<i>Postcondition</i>	Penambahan data tahun pelajaran akan disimpan dalam data tahun pelajaran.	

Tabel 4.13 Use Case Narrative Input Data Waktu Mengajar

Use Case Name	<i>Input</i> data waktu mengajar	
Use Case ID	9	
Actor(s)	Kepala Tata usaha	
Description	<i>Use case</i> ini menggambarkan kegiatan menginput data waktu mengajar.	
Precondition	<i>Login</i> sebagai Kepala Tata Usaha.	
Trigger	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem, serta data tahun pelajaran dan data pegawai terisi.	
Typical Course of Events	Actor Action	System Response
	<p>Langkah 1: Memilih menu " Waktu Mengajar", lalu klik submenu "Isi Waktu Mengajar".</p> <p>Langkah 3: Mengisikan data waktu</p>	<p>Langkah 2: Menampilkan halaman "Tambah Waktu Mengajar".</p>

	<p>mengajar.</p> <p>Langkah 4: Klik tombol "Simpan".</p>	<p>Langkah 5: Mengecek kelengkapan data.</p> <p>Langkah 6: Menyimpan data waktu mengajar.</p> <p>Langkah 7: Menampilkan di halaman "Lihat Waktu Mengajar".</p>
<i>Alternate Courses</i>	<p>Alt Langkah 1: Jika mengklik submenu "Lihat Waktu Mengajar", maka sistem akan menampilkan data waktu mengajar.</p> <p>Alt Langkah 4: Jika mengklik tombol "Kembali", maka sistem akan membatalkan dan kembali pada halaman "Lihat Waktu Mengajar".</p> <p>Alt Langkah 5: Jika data belum lengkap, maka sistem akan menampilkan <i>alert</i> pada <i>form</i> yang belum di isi.</p> <p>Alt Langkah 7a: Jika ketik tahun pelajaran atau nama pegawai dan mengklik tombol "Cari", sistem akan menampilkan data waktu mengajar yang dicari.</p> <p>Alt Langkah 7b: Jika mengklik icon  , sistem akan menampilkan halaman "Update Waktu Mengajar" untuk mengubah data waktu mengajar yang dipilih.</p> <p>Alt Langkah 7c: Jika mengklik icon  , sistem akan menampilkan <i>alert</i> untuk meyakinkan hapus data waktu mengajar yang dipilih.</p>	
<i>Conclusion</i>	Data waktu mengajar bertambah.	
<i>Postcondition</i>	Penambahan data waktu mengajar akan disimpan dalam data waktu mengajar.	

Tabel 4.14 Use Case Narrative Input Absensi

Use Case Name	Input absensi	
Use Case ID	10	
Actor(s)	Guru Piket	
Description	<i>Use case</i> ini menggambarkan kegiatan mengisi absensi.	
Precondition	<i>Login</i> sebagai Guru Piket.	
Trigger	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem.	
Typical Course of Events	Actor Action <p>Langkah 1: Memilih menu "Isi Absensi", lalu mengklik menu jurusan.</p> <p>Langkah 3: Mengisikan "Sesi" pada guru yang akan mengajar.</p>	System Response <p>Langkah 2: Menampilkan halaman "Input Pegawai" yang menampilkan daftar guru yang mengajar pada jurusan tersebut.</p>

	Langkah 4: Klik tombol "Simpan".	Langkah 5: Menyimpan data absensi. Langkah 6: Menampilkan data absensi.
Alternate Courses	-	
Conclusion	Data Absensi bertambah.	
Postcondition	Penambahan data absensi akan disimpan dalam data absensi.	

Tabel 4.15 Use Case Narrative Melihat Rekap Absensi

Use Case Name	Melihat rekap absensi	
Use Case ID	11	
Actor(s)	Guru Piket, Bendahara, Ketua Yayasan, Pegawai	
Description	<i>Use case</i> ini menggambarkan kegiatan melihat rekap absensi.	
Precondition	<i>Login</i> sebagai Guru Piket, Bendahara, Kepala Tata Usaha, Pegawai	
Trigger	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem	
Typical Course of Events	Actor Action Langkah 1: Memilih menu "Rekap Absensi", lalu mengklik menu harian. Langkah 3: Mengisi form tanggal dan nama pegawai. Langkah 4: Klik tombol "Cari"	System Response Langkah 2: Menampilkan menu cari rekap absensi harian. Langkah 5: Menampilkan rekap absensi harian
Alternate Courses	Alt Langkah 1: Jika mengklik menu "Bulanan", maka sistem akan menampilkan menu cari rekap absensi bulanan.	
Conclusion	Lihat rekap absensi.	
Postcondition	Rekap absensi ditampilkan.	

Tabel 4.16 Use Case Narrative Melihat Laporan Penggajian

Use Case Name	Melihat laporan penggajian	
Use Case ID	12	
Actor(s)	Bendahara, Ketua Yayasan	
Description	<i>Use case</i> ini menggambarkan kegiatan melihat laporan penggajian	
Precondition	<i>Login</i> sebagai Bendahara, Kepala Tata Usaha	
Trigger	<i>Use case</i> ini dilakukan setelah melakukan <i>login</i> ke dalam sistem	
Typical Course of	Actor Action	System Response

Events	<p>Langkah 1: Memilih menu "Laporan Gaji"</p> <p>Langkah 3: Mengisikan <i>form</i> bulan dan tahun</p> <p>Langkah 4: Klik tombol "Cari".</p>	<p>Langkah 2: Menampilkan halaman "Lihat Laporan".</p> <p>Langkah 5: Menampilkan laporan penggajian pada bulan yang telah dipilih.</p>
Alternate Courses	-	
Conclusion	Lihat laporan penggajian.	
Postcondition	Laporan penggajian ditampilkan.	

Tabel 4.17 Use Case Narrative Mencetak Laporan Penggajian

Use Case Name	Mencetak laporan penggajian	
Use Case ID	13	
Actor(s)	Bendahara	
Description	Use case ini menggambarkan kegiatan mencetak laporan penggajian.	
Precondition	<i>Login</i> sebagai Bendahara.	
Trigger	Use case ini dilakukan pada halaman laporan penggajian, serta setelah melakukan <i>login</i> ke dalam sistem	
Typical Course of Events	Actor Action Langkah 1: Memilih menu "Print" pada <i>browser</i> . Langkah 3: Klik "OK".	System Response Langkah 2: Menampilkan menu <i>setting</i> . Langkah 4: Mencetak laporan gaji. Langkah 5 : Menghasilkan <i>print out</i> laporan penggajian.
Alternate Courses	Alt Langkah 3: Jika klik "Cancel", maka sistem akan membatalkan proses mencetak.	
Conclusion	Cetak laporan penggajian.	
Postcondition	Laporan penggajian tercetak dalam bentuk <i>print out</i> (kertas).	

Tabel 4.18 Use Case Narrative Mencetak Slip Gaji

Use Case Name	Mencetak slip gaji
Use Case ID	14
Actor(s)	Bendahara, Pegawai
Description	Use case ini menggambarkan kegiatan mencetak slip gaji.
Precondition	<i>Login</i> sebagai Bendahara atau Pegawai.

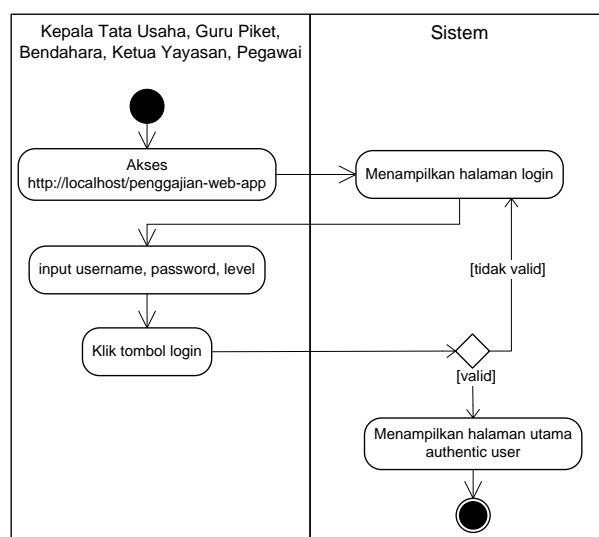
Trigger	Use case ini dilakukan pada halaman laporan penggajian.	
Typical Course of Events	Actor Action	System Response
	<p>Langkah 1: Memilih laporan gaji, lalu icon ”” pada sebelah kanan kolom tiap pegawai.</p> <p>Langkah 3: Memilih menu ”Print” pada browser.</p> <p>Langkah 5: Klik ”OK”.</p>	<p>Langkah 2: Menampilkan <i>format slip gaji</i>.</p> <p>Langkah 4: Menampilkan menu <i>setting</i> sebelum memproses cetak.</p> <p>Langkah 6: Mencetak slip gaji.</p> <p>Langkah 7: Menghasilkan <i>print out</i> slip gaji.</p>
Alternate Courses	Alt Langkah 5: Jika klik ”Cancel”, maka sistem akan membatalkan proses mencetak.	
Conclusion	Cetak slip gaji.	
Postcondition	Slip gaji tercetak.	

4.4.2 Activity Diagram

Berdasarkan dari *narrative use case* maka dapat digambarkan aktivitas-aktivitas yang terjadi atau alur kerja dalam *use case*. Aliran kerja tersebut digambarkan secara grafis dengan *activity diagram*.

Berikut ini adalah *activity diagram* dari masing-masing *use case*:

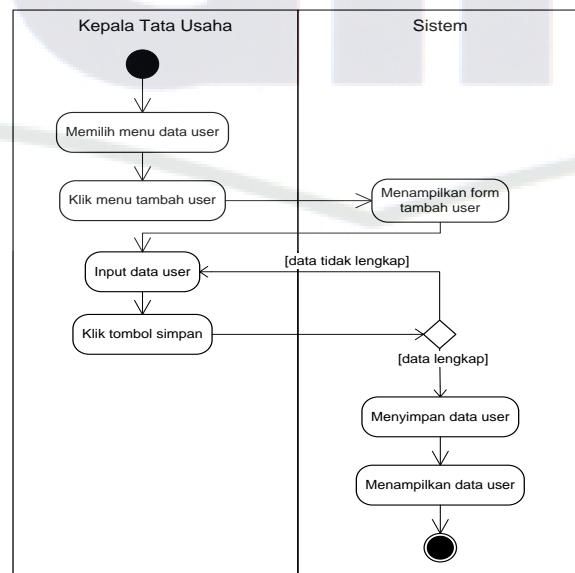
1. Activity Diagram Login



Gambar 4.6 Activity Diagram Login

Aktivitas pada gambar 4.6 dilakukan oleh kepala tata usaha, guru piket, bendahara, ketua yayasan, dan pegawai melakukan pengisian *username*, *password* dan level pada halaman *login*. Jika *username*, *password*, dan level yang dimasukkan salah, maka sistem akan menampilkan peringatan *username*, *password*, dan level yang dimasukkan salah. Jika benar, maka sistem akan menampilkan halaman utama sistem yang sesuai dengan level *user* dan selanjutnya *user* dapat mengakses menu-menu yang disediakan sistem sesuai level masing-masing.

2. Activity Diagram Input Data User

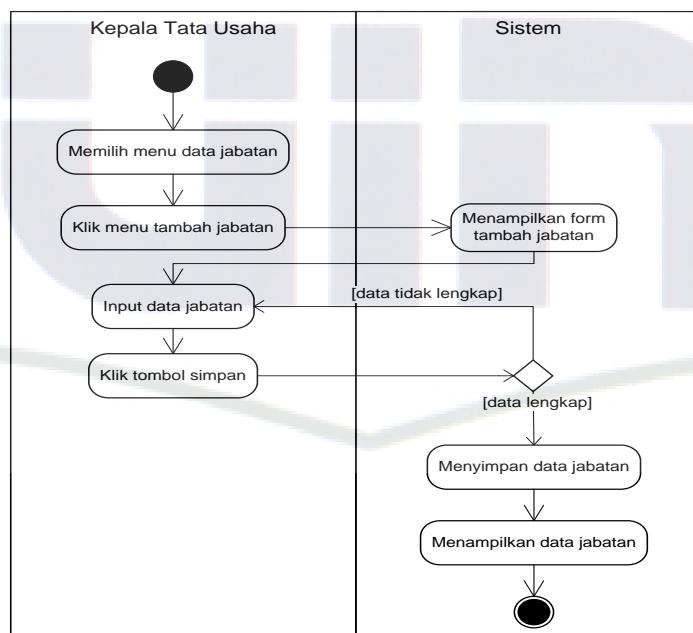


Gambar 4.7 Activity Diagram Input Data User

Aktivitas pada gambar 4.7 digunakan kepala tata usaha untuk mengisi dan menambah data *user*, namun sebelumnya jumlah *user* dan fungsinya sudah ditentukan. Sebelum melakukan penambahan data *user*, kepala tata usaha telah melakukan *login*

untuk mendapatkan hak akses. Setelah itu, kepala tata usaha dapat menggunakannya untuk *menginput* data *user*. Setelah data *diinput* kemudian tata usaha mengklik tombol simpan. Jika data yang *diinput* belum lengkap maka akan diberikan peringatan pada *form* yang belum diisi, tetapi jika data lengkap maka akan tersimpan dalam *database* dan menampilkan data *user*.

3. Activity Diagram Input Data Jabatan

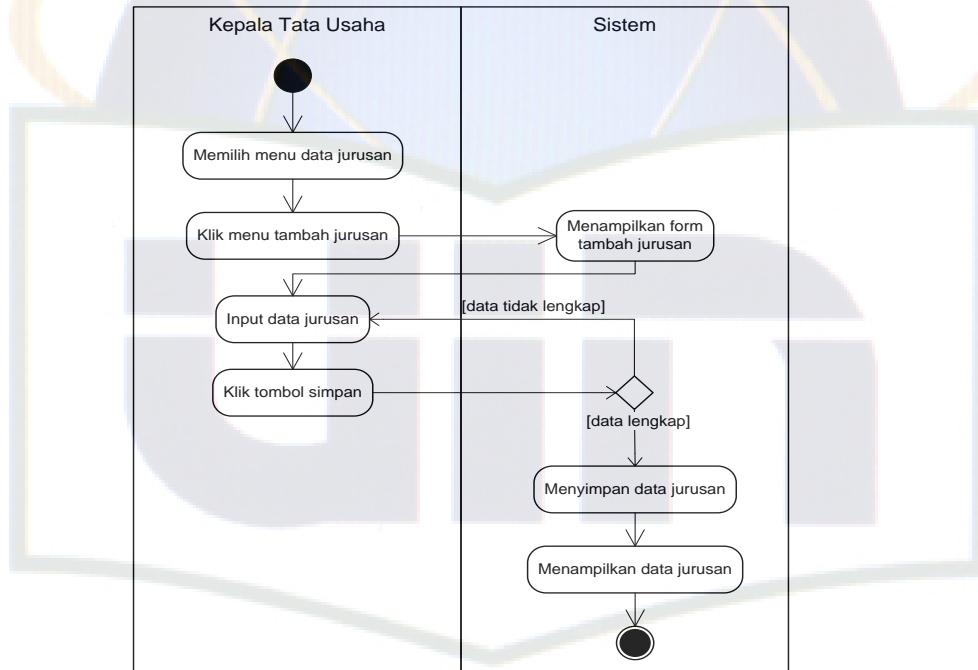


Gambar 4.8 Activity Diagram Input Data Jabatan

Aktivitas pada gambar 4.8 digunakan kepala tata usaha untuk mengisi dan menambah data jabatan. Sebelum melakukan penambahan data jabatan, kepala tata usaha harus melakukan *login* untuk mendapatkan hak akses. Setelah itu, kepala tata usaha dapat menggunakannya untuk *menginput* data jabatan. Setelah data *diinput* kemudian tata usaha mengklik tombol simpan. Jika

data yang *diinput* belum lengkap maka akan diberikan peringatan pada *form* yang belum diisi, tetapi jika data lengkap maka akan tersimpan dalam *database* dan menampilkan data jabatan.

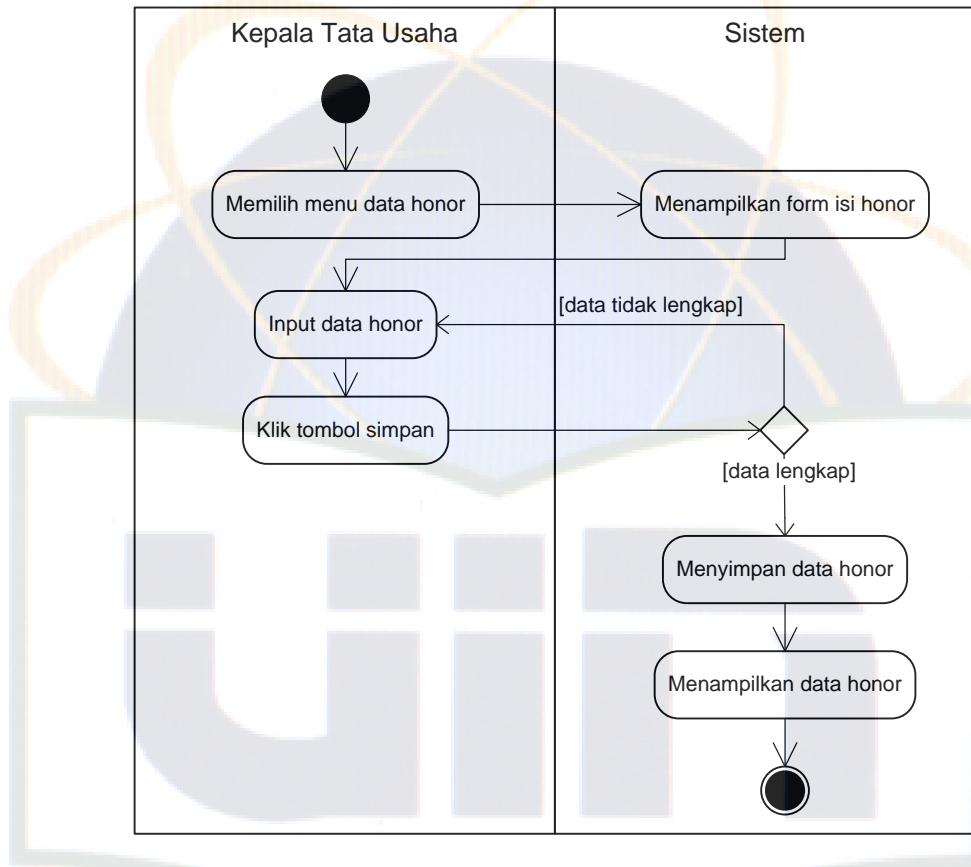
4. Activity Diagram Input Data Jurusan



Gambar 4.9 Activity Diagram Input Data Jurusan

Aktivitas pada gambar 4.9 digunakan kepala tata usaha untuk mengisi dan menambah data jurusan. Sebelum melakukan penambahan data jurusan, kepala tata usaha telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, kepala tata usaha dapat menggunakan untuk *menginput* data jurusan. Setelah data *diinput* kemudian tata usaha mengklik tombol simpan. Jika data yang *diinput* belum lengkap maka akan diberikan peringatan pada *form* yang belum diisi, tetapi jika data lengkap maka akan tersimpan dalam *database* dan menampilkan data jurusan.

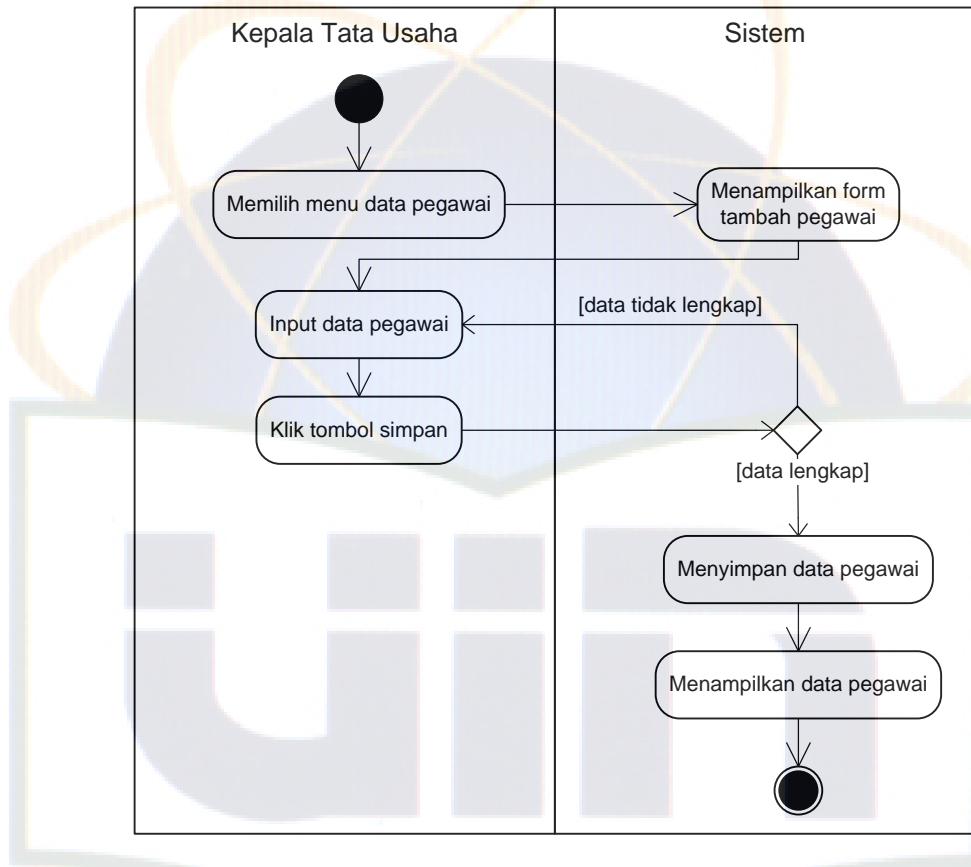
5. Activity Diagram Input Data Honor



Gambar 4.10 Activity Diagram Input Data Honor

Aktivitas gambar 4.10 digunakan kepala tata usaha untuk mengisi dan menambah data honor. Sebelum melakukan penambahan data honor, kepala tata usaha telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, kepala tata usaha dapat menggunakan untuk meng data honor. Setelah data di kemudian tata usaha mengklik tombol simpan. Jika data yang di belum lengkap maka akan diberikan peringatan pada *form* yang belum diisi, tetapi jika data lengkap maka akan tersimpan dalam *database* dan menampilkan data honor.

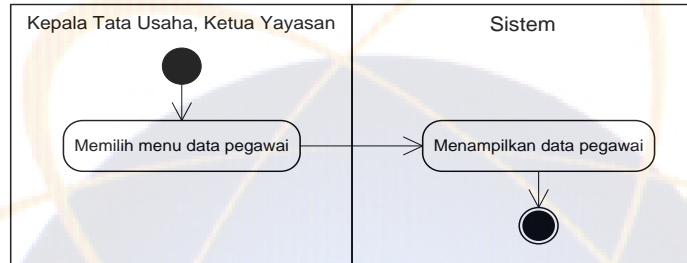
6. Activity Diagram Input Data Pegawai



Gambar 4.11 Activity Diagram Input Data Pegawai

Aktivitas pada gambar 4.11 digunakan kepala tata usaha untuk mengisi dan menambah data pegawai. Sebelum melakukan penambahan data pegawai, kepala tata usaha telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, kepala tata usaha dapat menggunakan untuk meng data pegawai. Setelah data di kemudian tata usaha mengklik tombol simpan. Jika data yang di belum lengkap maka akan diberikan peringatan pada *form* yang belum diisi, tetapi jika data lengkap maka akan tersimpan dalam *database* dan menampilkan data pegawai.

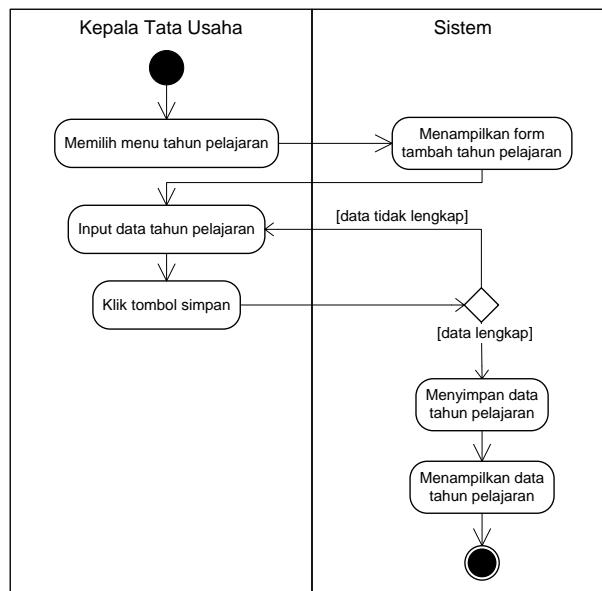
7. Activity Diagram Melihat Data Pegawai



Gambar 4.12 Activity Diagram Melihat Data Pegawai

Aktivitas pada gambar 4.12 digunakan kepala tata usaha dan ketua yayasan untuk melihat data pegawai. Sebelum melihat data pegawai, kepala tata usaha dan ketua yayasan telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, kepala tata usaha dan ketua yayasan dapat menggunakannya untuk melihat data pegawai. Pilih menu lihat pegawai, maka sistem akan menampilkan data pegawai.

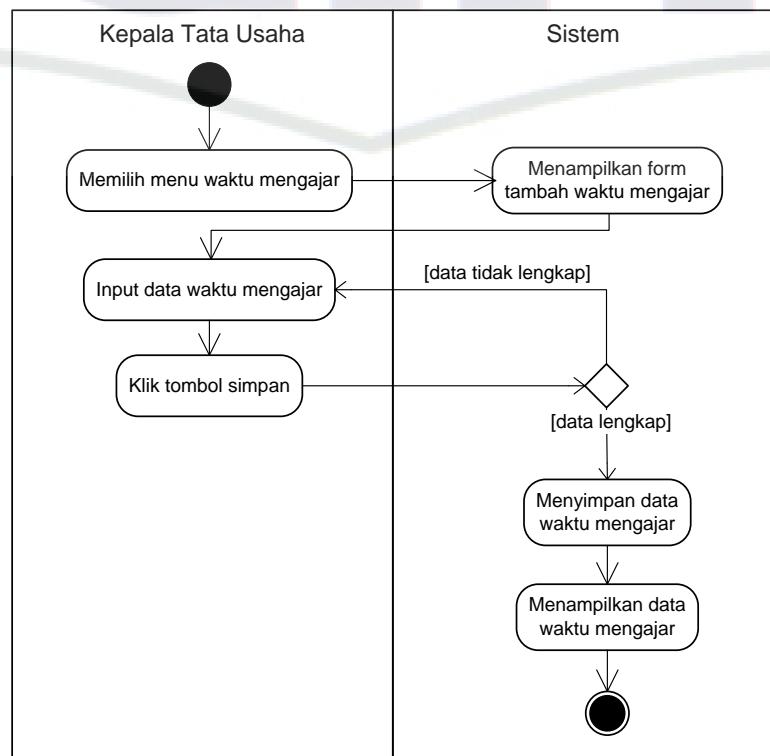
8. Activity Diagram Input Data Tahun Pelajaran



Gambar 4.13 Activity Diagram Input Data Tahun Pelajaran

Aktivitas pada gambar 4.13 digunakan kepala tata usaha untuk mengisi dan menambah data tahun pelajaran. Sebelum melakukan penambahan data tahun pelajaran, kepala tata usaha telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, kepala tata usaha dapat menggunakannya untuk *menginput* data tahun pelajaran. Setelah data *diinput* kemudian tata usaha mengklik tombol simpan. Jika data yang *diinput* belum lengkap maka akan diberikan peringatan pada *form* yang belum diisi, tetapi jika data lengkap maka akan tersimpan dalam *database* dan menampilkan data tahun pelajaran.

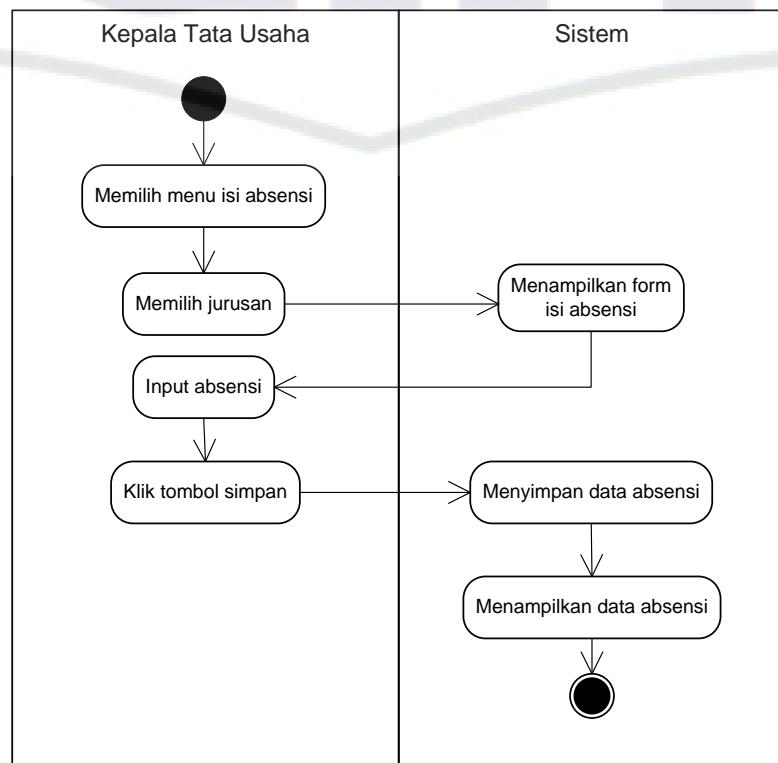
9. Activity Diagram Input Data Waktu Mengajar



Gambar 4.14 Activity Diagram Input Data Waktu Mengajar

Aktivitas pada gambar 4.14 ini digunakan kepala tata usaha untuk mengisi dan menambah data waktu mengajar. Sebelum melakukan penambahan data waktu mengajar, ketua tata usaha telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, kepala tata usaha dapat menggunakan untuk *menginput* data waktu mengajar. Setelah data *diinput* kemudian tata usaha mengklik tombol simpan. Jika data yang *diinput* belum lengkap maka akan diberikan peringatan pada *form* yang belum diisi, tetapi jika data lengkap maka akan tersimpan dalam *database* dan menampilkan data waktu mengajar.

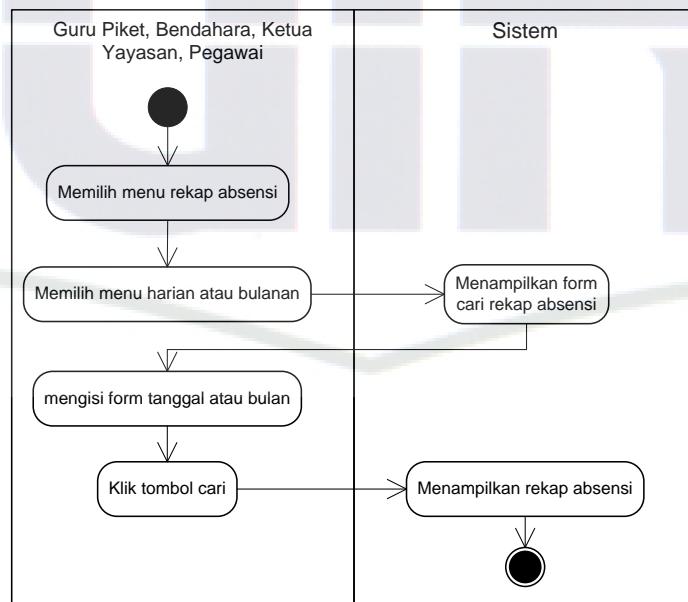
10. Activity Diagram Input Absensi



Gambar 4.15 Activity Diagram Input Absensi

Aktivitas pada gambar 4.15 ini digunakan guru piket untuk mengisi dan menambah absensi. Sebelum melakukan pengisian absensi, guru piket telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, guru piket dapat menggunakan untuk meng absensi. Setelah data di kemudian tata usaha mengklik tombol simpan. Jika data lengkap maka akan tersimpan dalam *database* dan menampilkan data absensi.

11. Activity Diagram Melihat Rekap Absensi

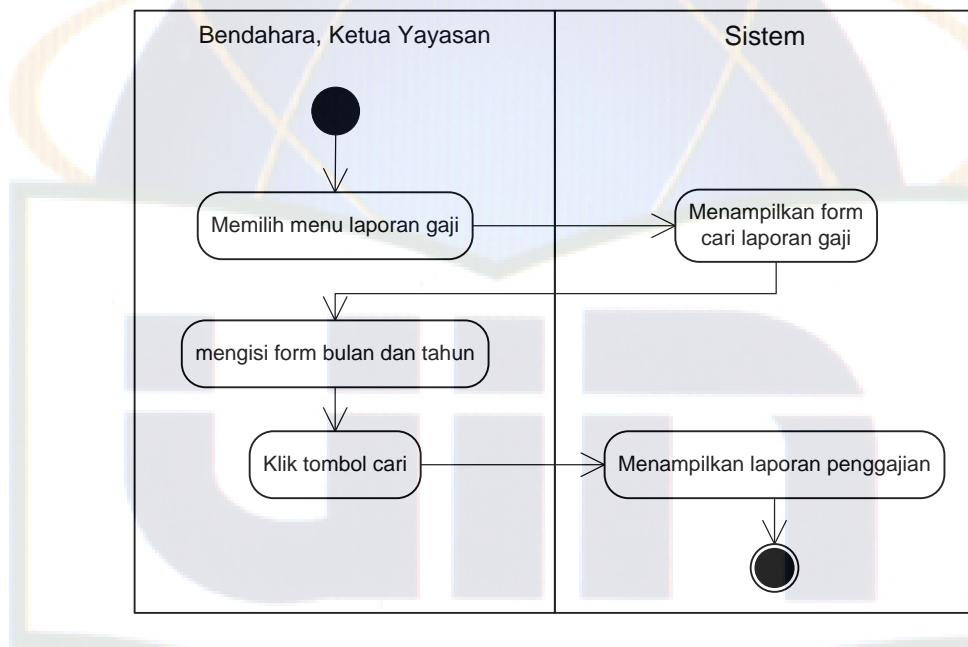


Gambar 4.16 Activity Diagram Melihat Rekap Absensi

Aktivitas pada gambar 4.16 digunakan guru piket, bendahara, ketua yayasan untuk melihat rekap absensi. Sebelum melihat rekap absensi, guru piket, bendahara, dan ketua yayasan telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, guru piket, bendahara, dan ketua yayasan dapat menggunakan untuk melihat rekap absensi. Jika memilih menu harian, maka

akan tampil rekap absensi harian. Jika memilih menu bulanan, maka akan tampil rekap absensi bulanan.

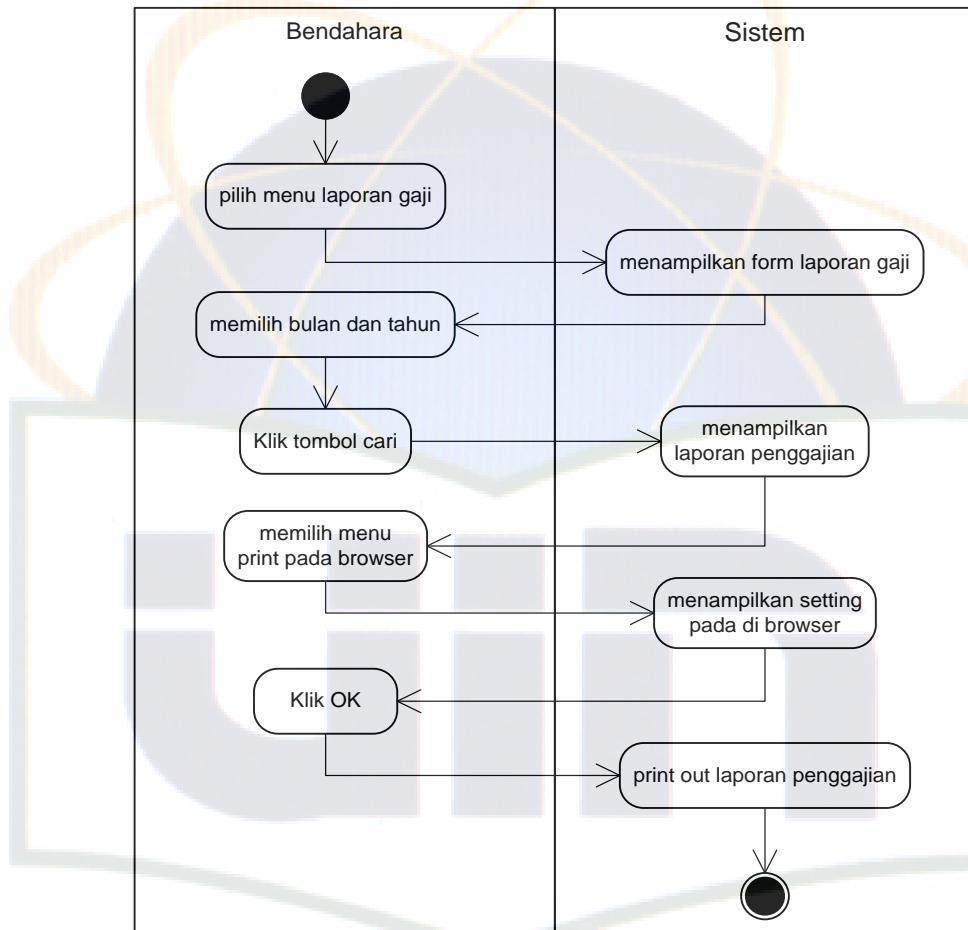
12. Activity Diagram Melihat Laporan Penggajian



Gambar 4.17 Activity Diagram Melihat Laporan Penggajian

Aktivitas pada gambar 4.17 digunakan bendahara, ketua yayasan untuk melihat laporan penggajian. Sebelum melihat laporan penggajian, bendahara dan ketua yayasan telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, bendahara dan ketua yayasan dapat menggunakananya untuk melihat laporan penggajian. Jika memilih bulan dan tahun kemudian klik tombol cari, maka akan menampilkan laporan penggajian pada bulan tersebut.

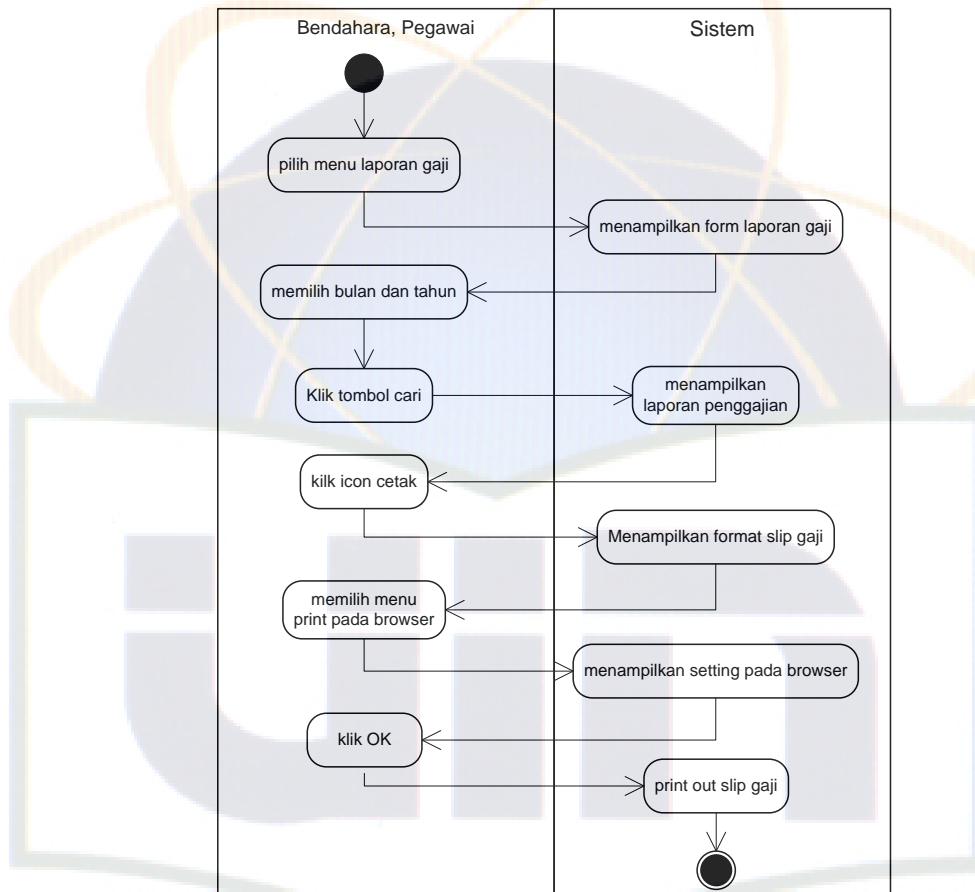
13. Activity Diagram Mencetak Laporan Penggajian



Gambar 4.18 Activity Diagram Mencetak Laporan Penggajian

Aktivitas pada gambar 4.18 ini digunakan bendahara untuk mencetak laporan penggajian. Sebelum melihat laporan penggajian, bendahara telah melakukan *login* untuk mendapatkan hak akses. Setelah itu, bendahara dapat menggunakan untuk mencetak laporan penggajian. Jika memilih bulan dan tahun, maka akan menampilkan laporan penggajian pada bulan tersebut. Bendahara memilih menu *print* di *browser* kemudian klik OK, maka akan menghasilkan *print out* laporan penggajian.

14. Activity Diagram Mencetak Slip Gaji



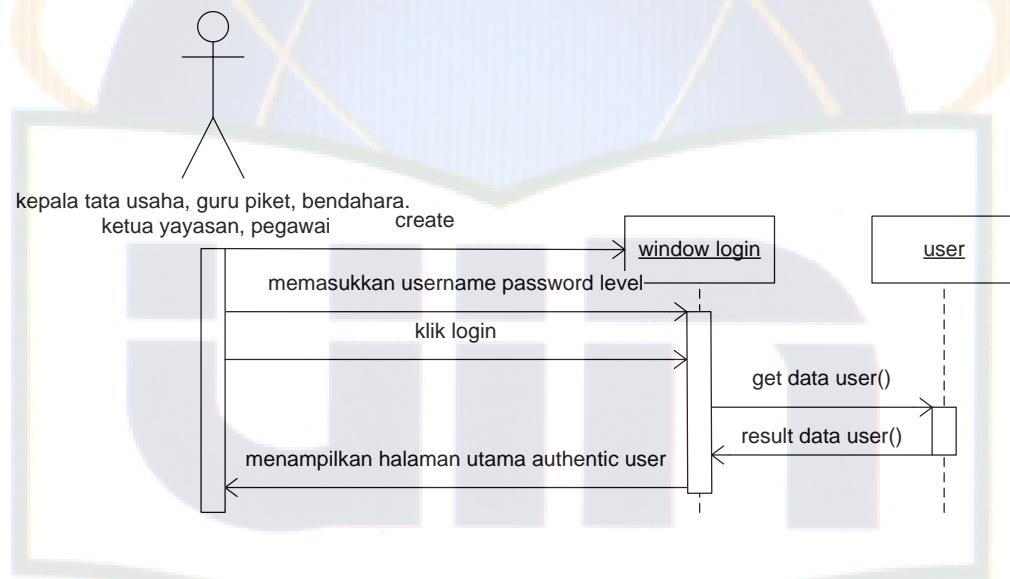
Gambar 4.19 Activity Diagram Mencetak Slip Gaji

Aktivitas pada gambar 4.19 digunakan bendahara dan pegawai untuk mencetak slip gaji. Sebelum melihat slip gaji, bendahara telah melakukan *login* terlebih dahulu untuk mendapatkan hak akses. Setelah itu, bendahara dan dapat menggunakan untuk mencetak slip gaji. Jika memilih bulan dan tahun, maka akan menampilkan laporan penggajian pada bulan tersebut. Bendahara memilih *icon* cetak di setiap nama pegawai, maka akan menampilkan format slip gaji. Kemudian pilih *print* di *browser* kemudian klik *OK*, maka akan menghasilkan *print out* slip gaji.

4.4.3 Sequence Diagram

Interaksi antara *object* yang dibutuhkan untuk menjalankan sebuah *use case*, dalam menangkap interaksi obyek-obyek ini menggunakan *sequence diagram*:

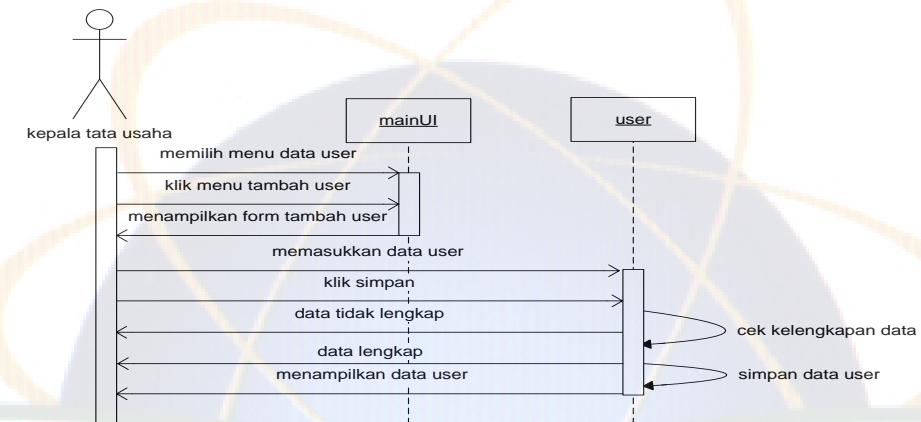
1. Sequence Diagram Login



Gambar 4.20 Sequence Diagram Login

Pada *sequence diagram* gambar 4.20 menjelaskan interaksi antara kepala tata usaha, guru piket, bendahara, ketua yayasan, dan pegawai sebagai *actor* melakukan *login* ke sistem. Untuk melakukan login, *actor* harus mengisi *username*, *password*, dan level. Sistem akan mengecek *username*, *password*, dan level. Jika login tersebut benar, maka akan muncul halaman utama sistem sesuai dengan level yang digunakan.

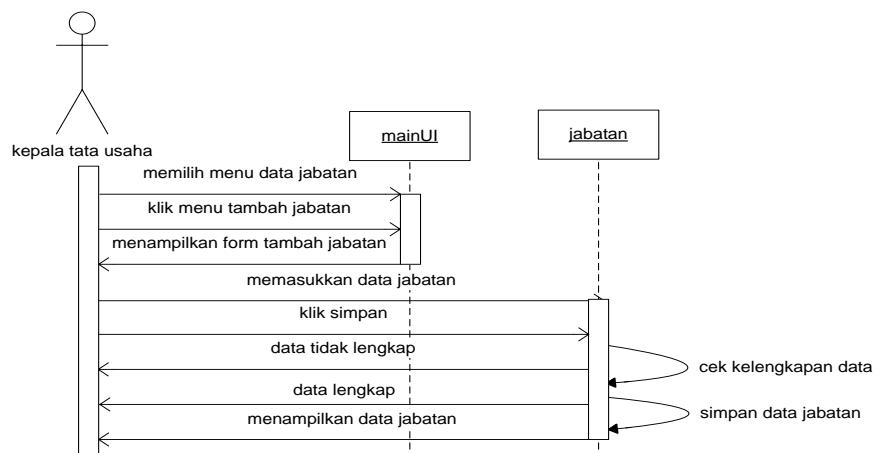
2. Sequence Diagram Input Data User



Gambar 4.21 Sequence Diagram Input Data User

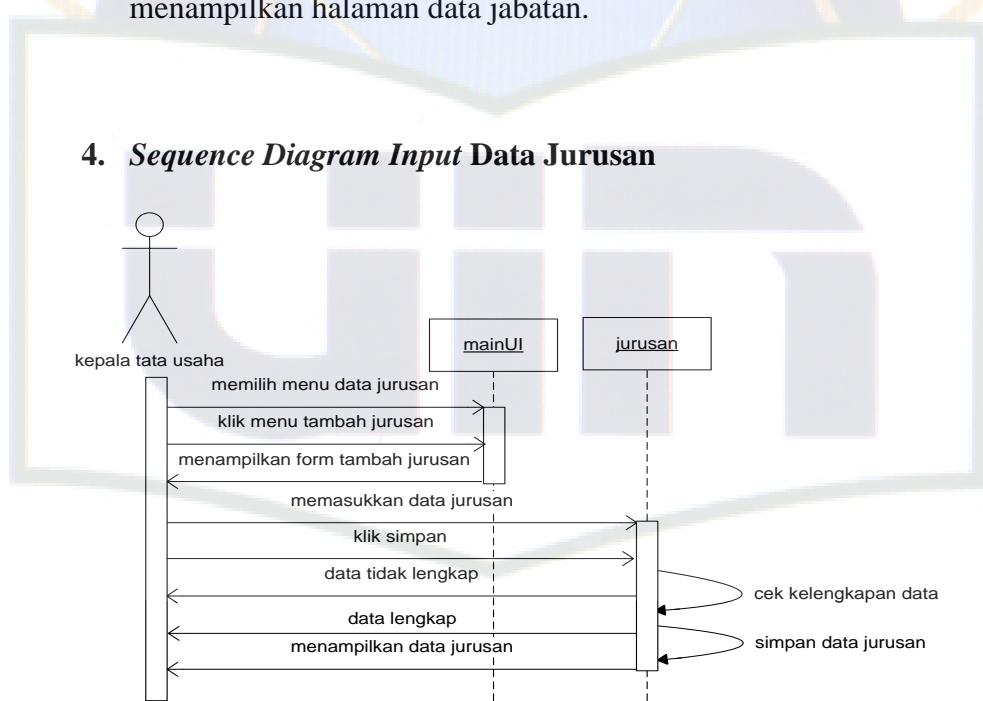
Pada *sequence diagram* gambar 4.21 menjelaskan interaksi antara kepala tata usaha dengan sistem dalam melakukan *input data user*. Dimulai dari kepala tata usaha telah *login* terlebih dahulu. Setelah itu kepala tata usaha akan mengisikan data *user*. Sistem akan mengecek kelengkapan data, jika data sudah lengkap maka data akan disimpan ke *database* dan sistem menampilkan halaman data *user*.

3. Sequence Diagram Input Data Jabatan



Gambar 4.22 Sequence Diagram Input Data Jabatan

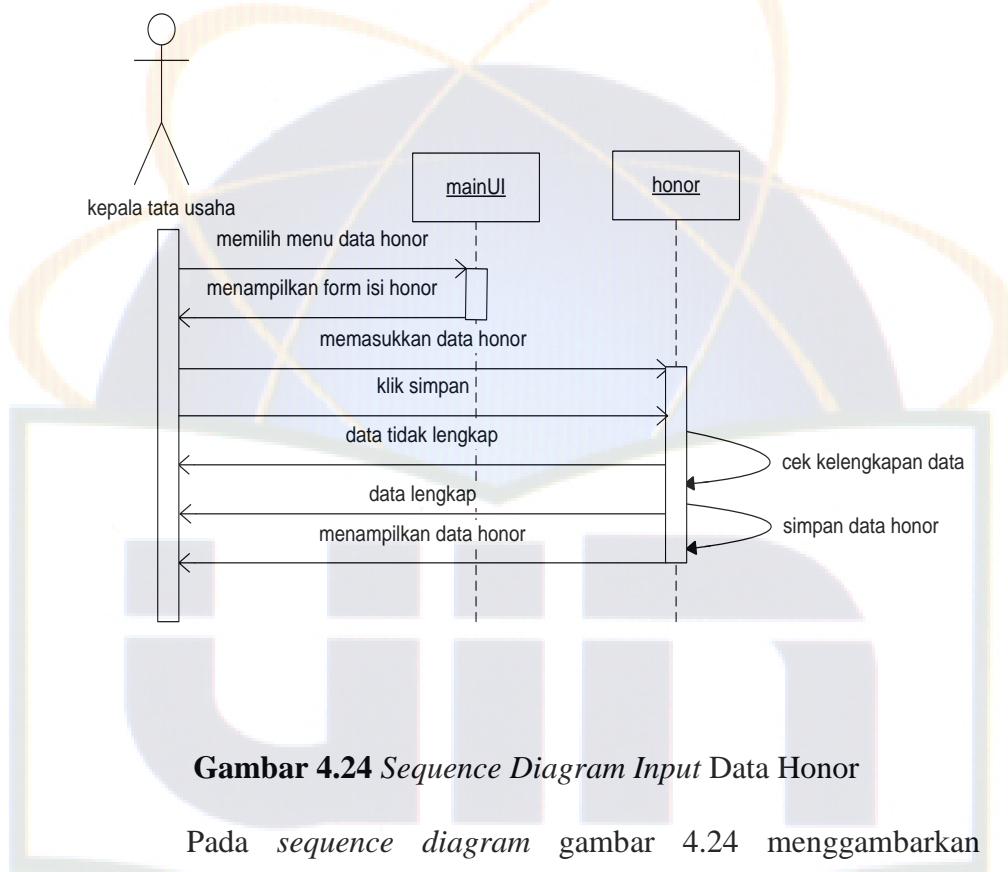
Pada *sequence diagram* gambar 4.22 menggambarkan interaksi antara kepala tata usaha dengan sistem dalam melakukan *input* data jabatan. Dimulai dari kepala tata usaha telah *login* terlebih dahulu. Setelah itu kepala tata usaha akan mengisikan data jabatan. Sistem akan mengecek kelengkapan data, jika data sudah lengkap maka data akan disimpan ke *database* dan sistem menampilkan halaman data jabatan.



Gambar 4.23 Sequence Diagram Input Data Jurusan

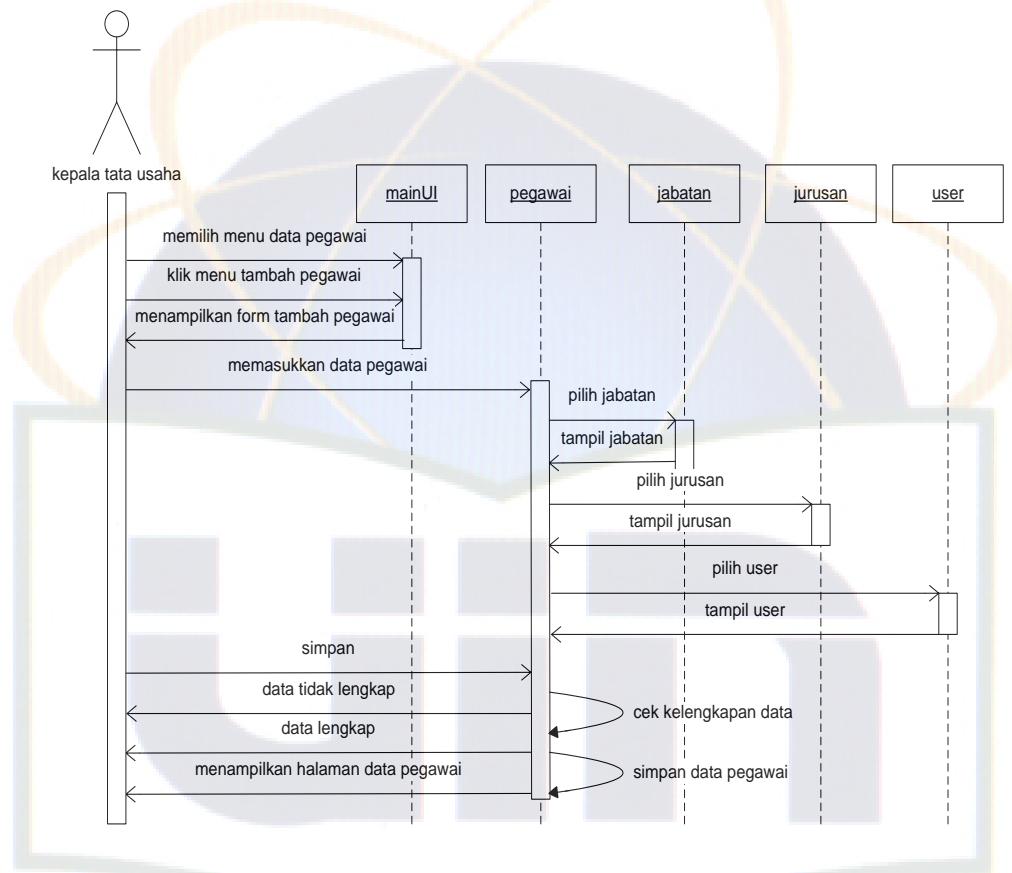
Pada *sequence diagram* gambar 4.23 menggambarkan interaksi antara kepala tata usaha dengan sistem dalam melakukan *input* data jurusan. Dimulai dari kepala tata usaha telah *login* terlebih dahulu. Setelah itu kepala tata usaha akan mengisikan data jurusan. Sistem akan mengecek kelengkapan data, jika data sudah lengkap maka data akan disimpan ke *database* dan sistem menampilkan halaman data jurusan.

5. Sequence Diagram Input Data Honor



Pada *sequence diagram* gambar 4.24 menggambarkan interaksi antara kepala tata usaha dengan sistem dalam melakukan *input* data honor. Dimulai dari kepala tata usaha telah *login* terlebih dahulu. Setelah itu kepala tata usaha akan mengisikan data honor. Sistem akan mengecek kelengkapan data, jika data sudah lengkap maka data akan disimpan ke *database* dan sistem menampilkan halaman data honor.

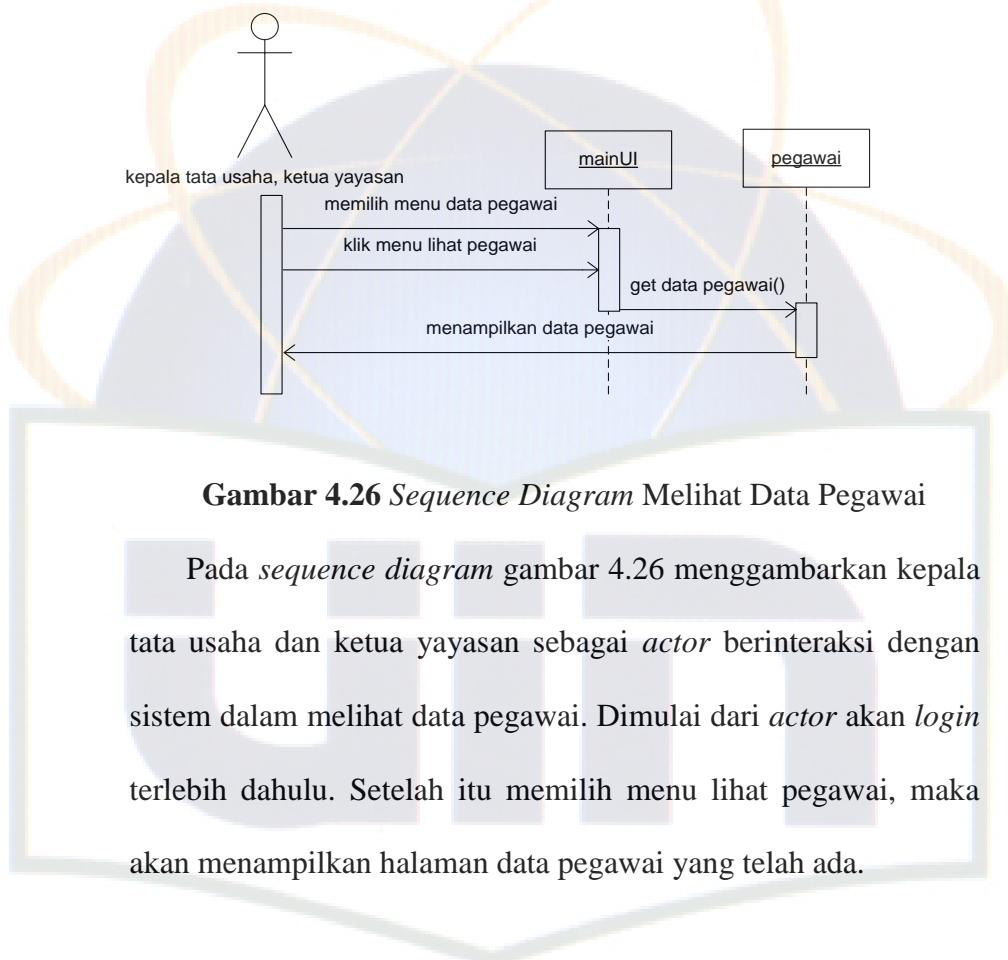
6. Sequence Diagram Input Data Pegawai



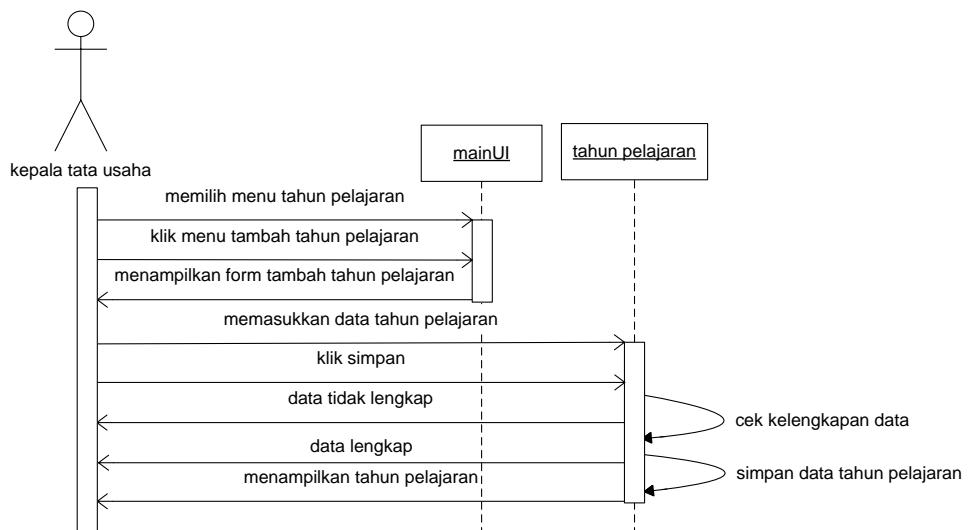
Gambar 4.25 Sequence Diagram Input Data Pegawai

Pada *sequence diagram* gambar 4.25 menggambarkan interaksi antara kepala tata usaha dengan sistem dalam melakukan *input* data pegawai. Dimulai dari kepala tata usaha akan *login* terlebih dahulu. Setelah itu kepala tata usaha akan mengisikan data pegawai. Sistem akan mengecek kelengkapan data, jika data sudah lengkap maka data akan disimpan ke *database* dan sistem menampilkan halaman data pegawai.

7. Sequence Diagram Melihat Data Pegawai



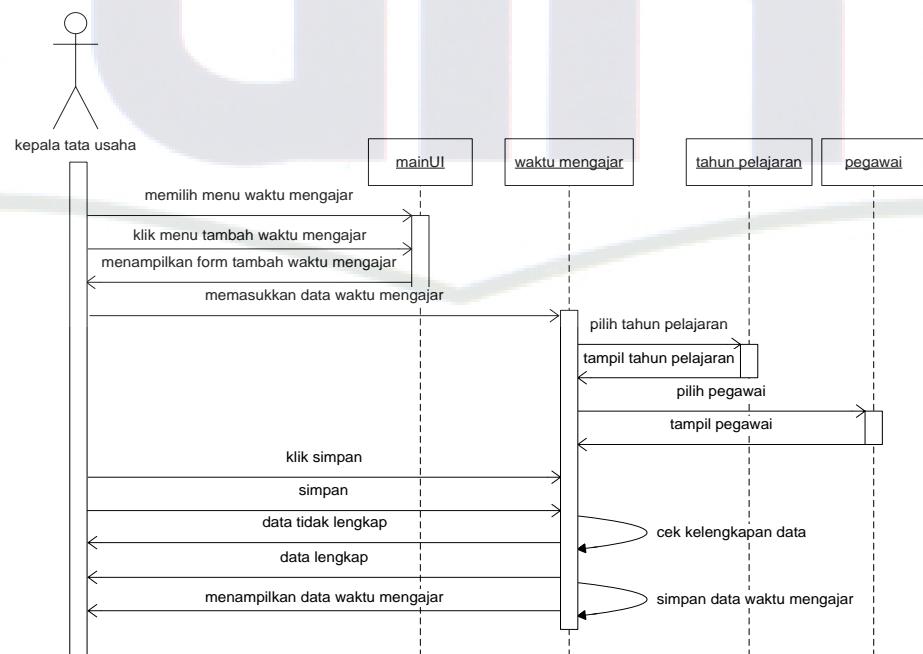
8. Sequence Diagram Input Tahun Pelajaran



Gambar 4.27 Sequence Diagram Input Tahun Pelajaran

Pada *sequence diagram* gambar 4.27 menggambarkan interaksi antara kepala tata usaha dengan sistem dalam melakukan *input* data tahun pelajaran. Dimulai dari kepala tata usaha akan *login* terlebih dahulu. Setelah itu kepala tata usaha akan mengisikan data tahun pelajaran. Sistem akan mengecek kelengkapan data, jika data sudah lengkap maka data akan disimpan ke *database* dan sistem menampilkan halaman data tahun pelajaran.

9. Sequence Diagram Input Waktu Mengajar

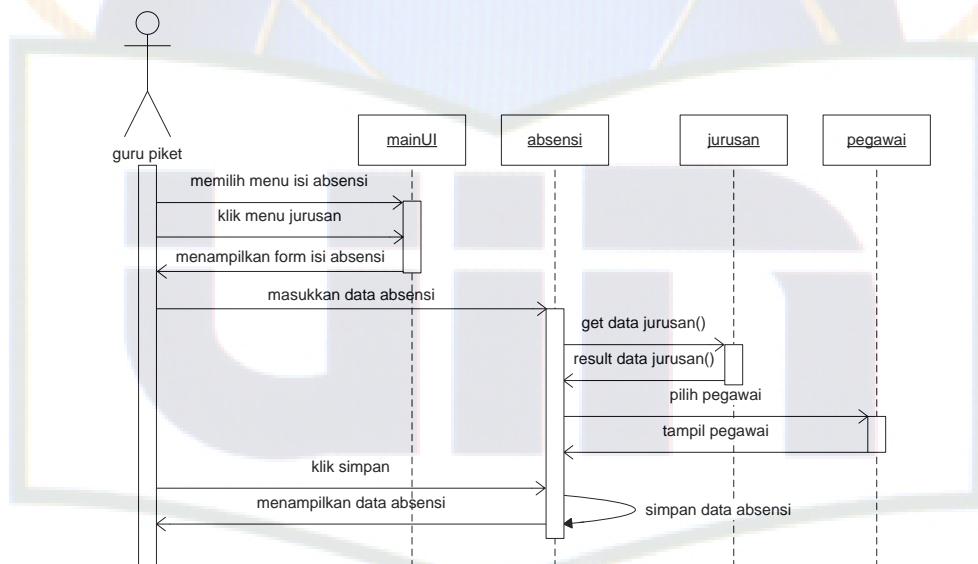


Gambar 4.28 Sequence Diagram Input Waktu Mengajar

Pada *sequence diagram* gambar 4.28 menggambarkan interaksi antara kepala tata usaha dengan sistem dalam melakukan *input* data waktu mengajar. Dimulai dari kepala tata usaha akan *login* terlebih dahulu. Setelah itu kepala tata usaha akan

mengisikan data waktu mengajar. Sistem akan mengecek kelengkapan data, jika data sudah lengkap maka data akan disimpan ke *database* dan sistem menampilkan halaman data waktu mengajar.

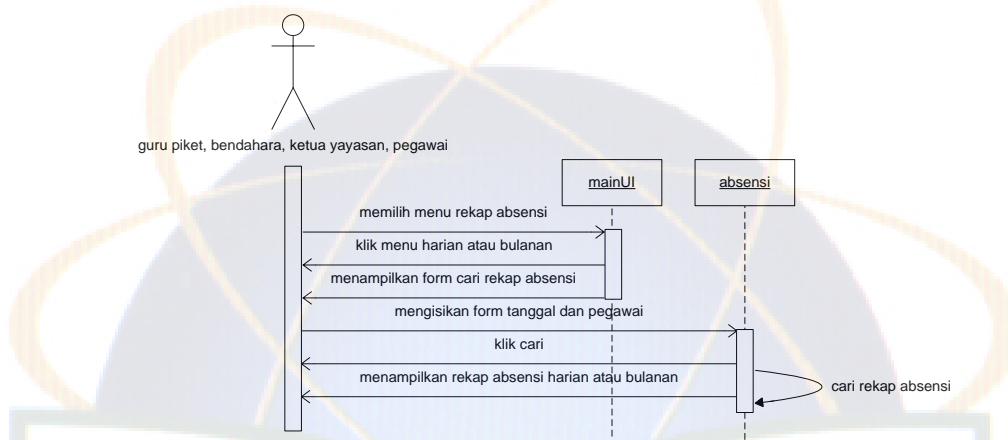
10. Sequence Diagram Input Absensi



Gambar 4.29 Sequence Diagram Input Absensi

Pada *sequence diagram* gambar 4.29 menggambarkan interaksi antara guru piket dengan sistem dalam melakukan *input* absensi. Dimulai dari kepala tata usaha akan *login* terlebih dahulu. Guru piket memilih menu isi absensi lalu memilih jurusannya, maka sistem akan menampilkan *form* absensi. Setelah itu kepala tata usaha akan mengisikan data absensi. Sistem akan mengecek kelengkapan data, jika data sudah lengkap maka data akan disimpan ke *database* dan sistem menampilkan halaman data absensi.

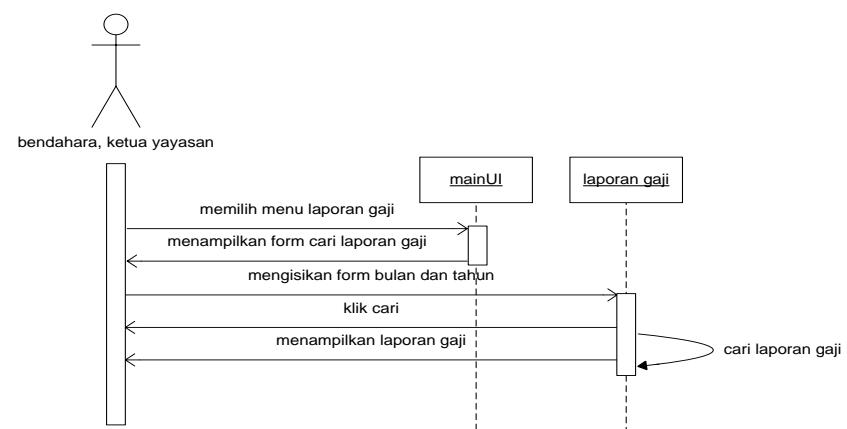
11. Sequence Diagram Melihat Rekap Absensi



Gambar 4.30 Sequence Diagram Melihat Rekap Absensi

Pada *sequence diagram* gambar 4.30 menggambarkan guru piket, bendahara, ketua yayasan sebagai *actor* berinteraksi dengan sistem dalam melihat rekap absensi. Dimulai dari *actor* akan *login* terlebih dahulu. Setelah itu memilih menu rekap absensi harian atau bulanan, lalu memasukkan tanggal atau bulan lalu klik cari, maka akan menampilkan halaman rekap absensi harian ataupun bulanan.

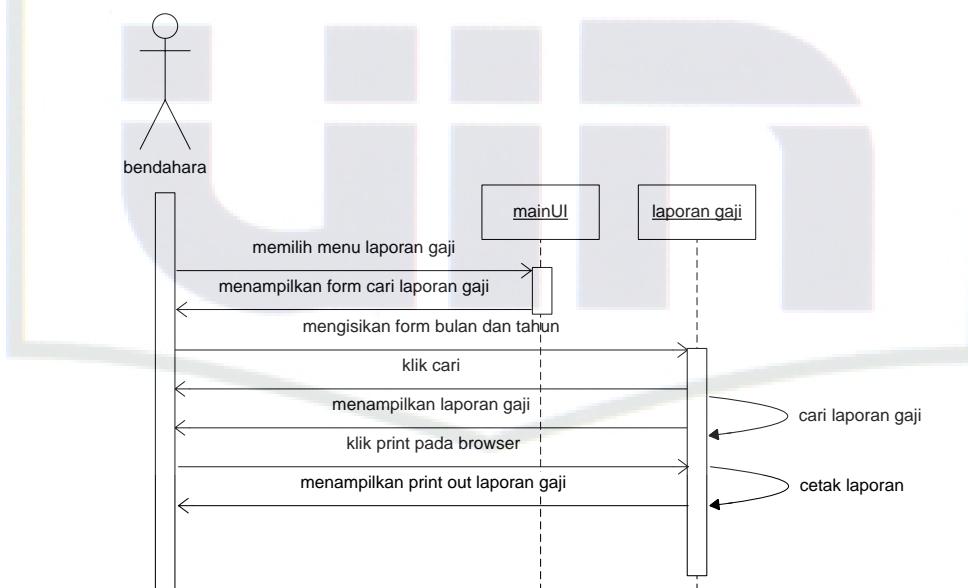
12. Sequence Diagram Melihat Laporan Penggajian



Gambar 4.31 Sequence Diagram Melihat Laporan Penggajian

Pada *sequence diagram* gambar 4.31 menggambarkan bendahara dan ketua yayasan sebagai *actor* berinteraksi dengan sistem dalam melihat laporan penggajian. Dimulai dari *actor* akan *login* terlebih dahulu. Setelah itu memilih menu laporan gaji, lalu pilih bulan dan klik cari. Maka sistem akan menampilkan halaman laporan penggajian sesuai bulan yang dipilih.

13. Sequence Diagram Mencetak Laporan Penggajian

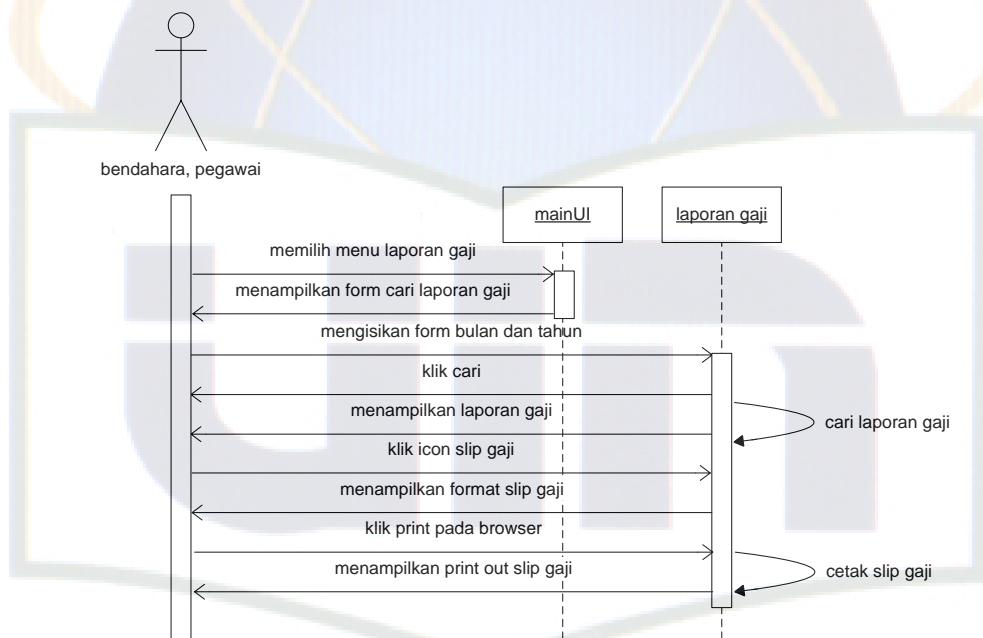


Gambar 4.32 Sequence Diagram Mencetak Laporan Penggajian

Pada *sequence diagram* gambar 4.32 menggambarkan bendahara berinteraksi dengan sistem dalam mencetak laporan penggajian. Dimulai dari bendahara akan *login* terlebih dahulu. Setelah itu memilih menu laporan gaji, lalu pilih bulan dan klik cari. Maka sistem akan menampilkan halaman laporan penggajian sesuai bulan yang dipilih. Setelah itu, bendahara memilih menu

print pada *browser* lalu klik OK dan sistem akan mencetak laporan penggajian yang menghasilkan *print out* laporan penggajian.

14. Sequence Diagram Mencetak Slip Gaji



Gambar 4.33 Sequence Diagram Mencetak Slip Gaji

Pada *sequence diagram* gambar 4.33 menggambarkan bendahara berinteraksi dengan sistem dalam mencetak slip gaji. Dimulai dari *actor* akan *login* terlebih dahulu. Setelah itu memilih menu laporan gaji, lalu pilih bulan dan klik cari. Maka sistem akan menampilkan halaman laporan penggajian sesuai bulan yang dipilih. Setelah itu, bendahara memilih *icon* cetak pada tiap nama pegawai, maka akan tampil format slip gaji. Kemudian memilih menu *print* pada *browser* lalu klik OK dan sistem akan mencetak slip gaji yang menghasilkan *print out* slip gaji.

4.4.4 Class Diagram

Visualisasi dari struktur *object* sistem yang diajukan, digambarkan dalam *class diagram*. Adapun langkah-langkah untuk menentukan *object* yang terlibat adalah sebagai berikut:

- Menemukan *potential object*

Tabel 4.19 Daftar *Potential Object*

<i>user</i>		<i>pegawai</i>
IDUser		IDPegawai
namaUser		namaPegawai
keteranganUser		jenisPegawai
jabatan		tempatLahir
IDJabatan		tanggalLahir
namaJabatan		alamatPegawai
jumlahTunjangan		teleponPegawai
keteranganJabatan		pendidikanTerakhir
jurusan		email
namaJurusan		mulaiTugas
keteranganJurusan		username
honor		password
IDHonor		loginTerakhir
honorPerJam		IP
transportPerJam		statusPegawai
piketPerhari		piket
waktumengajar		IDPiket
IDWaktuMengajar		tanggalPiket
jumlahWaktuMengajar		keteranganPiket
tahunpelajaran		absensi
IDTahunPelajaran		IDAbsensi
namaTahunPelajaran		sesi
keteranganTahunPelajaran		tanggalInput
statusTahunPelajaran		terakhirUpdate
penggajian		terakhirUpdateOleh

b. Memilih *object*

Tabel 4.20 Daftar Analisa Potential Object

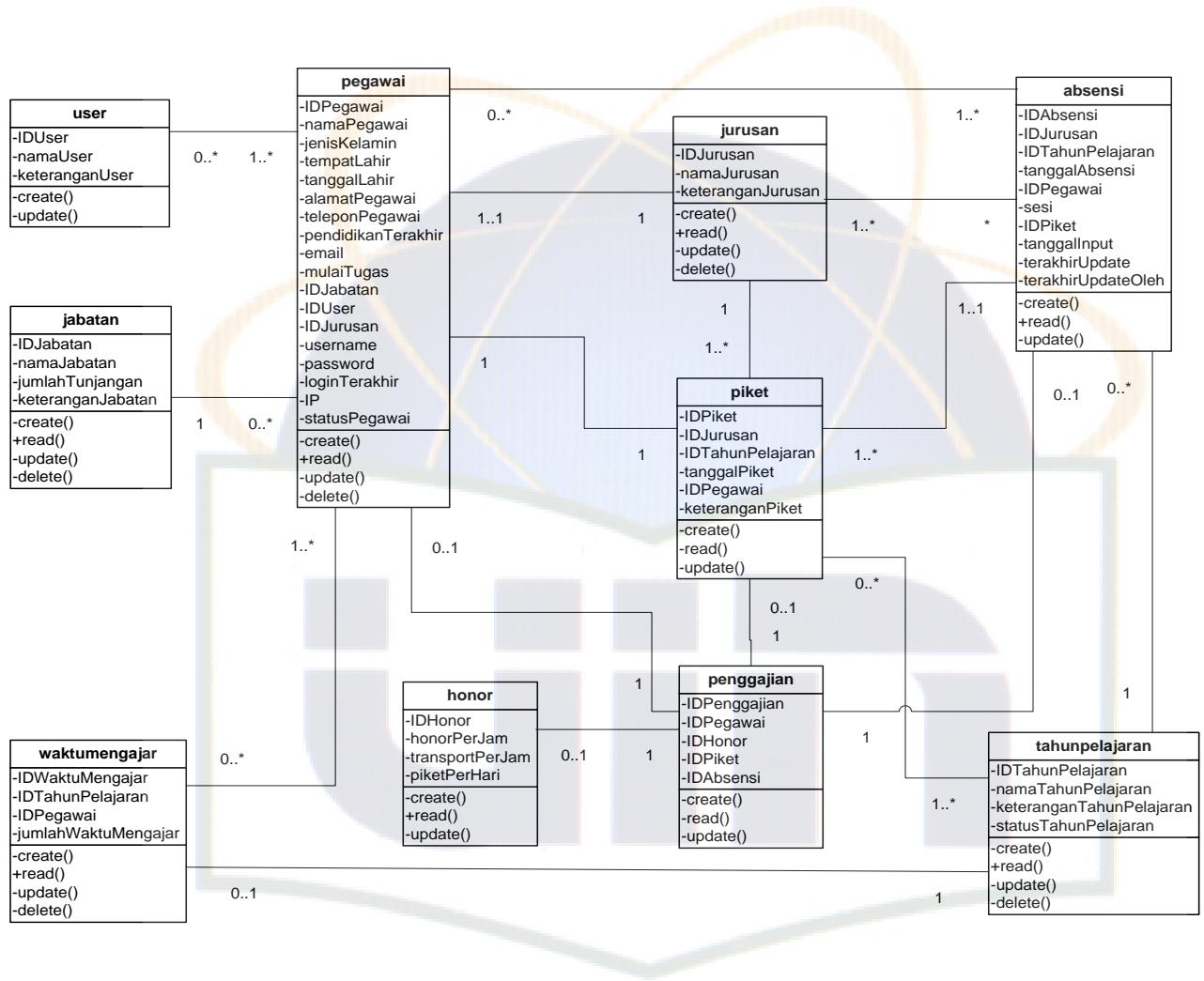
Potential Object	Reason
user	√ Ditulis user
IDUser	X Tidak relevan
namaUser	X Tidak relevan
keteranganUser	X Tidak relevan
jabatan	√ Ditulis jabatan
IDJabatan	X Tidak relevan
namaJabatan	X Tidak relevan
jumlahTunjangan	X Tidak relevan
keteranganJabatan	X Tidak relevan
jurusan	√ Ditulis jurusan
namaJurusan	X Tidak relevan
keteranganJurusan	X Tidak relevan
honor	√ Ditulis honor
IDHonor	X Tidak relevan
honorPerJam	X Tidak relevan
transportPerJam	X Tidak relevan
piketPerhari	X Tidak relevan
waktumengajar	√ Ditulis waktumengajar
IDWaktuMengajar	X Tidak relevan
jumlahWaktuMengajar	X Tidak relevan
tahunpelajaran	√ Ditulis tahunpelajaran
IDTahunPelajaran	X Tidak relevan
namaTahunPelajaran	X Tidak relevan
keteranganTahunPelajaran	X Tidak relevan
statusTahunPelajaran	X Tidak relevan
penggajian	√ Ditulis penggajian
IDPenggajian	X Tidak relevan
pegawai	√ Ditulis pegawai
IDPegawai	X Tidak relevan
namaPegawai	X Tidak relevan
jenisPegawai	X Tidak relevan
tempatLahir	X Tidak relevan
tanggalLahir	X Tidak relevan
alamatPegawai	X Tidak relevan
teleponPegawai	X Tidak relevan

pendidikanTerakhir	X	Tidak relevan
email	X	Tidak relevan
mulaiTugas	X	Tidak relevan
username	X	Tidak relevan
password	X	Tidak relevan
loginTerakhir	X	Tidak relevan
IP	X	Tidak relevan
statusPegawai	X	Tidak relevan
piket	✓	Ditulis piket
IDPiket	X	Tidak relevan
tanggalPiket	X	Tidak relevan
keteranganPiket	X	Tidak relevan
absensi	✓	Ditulis absensi
IDAbsensi	X	Tidak relevan
sesi	X	Tidak relevan
tanggalInput	X	Tidak relevan
terakhirUpdate	X	Tidak relevan
terakhirUpdateOleh	X	Tidak relevan

Dari analisa di atas, didapatkan *object* yang terkait dengan sistem yang diajukan.

Tabel 4.21 Daftar *Object*

user
jabatan
jurusun
honor
waktumengajar
tahunpelajaran
penggajian
pegawai
piket
absensi



Gambar 4.34 Class Diagram

4.4.5 Tabel Database

1. Tabel User

Nama tabel : tbl_user

Type File : Master

Primary Key : IDUser

Foreign Key :-

Tabel 4.22 Tabel User

Nama Field	Tipe Data	Ukuran	Keterangan
IDUser	int	5	Id user
namaUser	varchar	50	Nama user
keteranganUser	varchar	50	Keterangan user

2. Tabel jabatan

Nama tabel : tbl_jabatan

Type File : Master

Primary Key : IDJabatan

Foreign Key :-

Tabel 4.23 Tabel Jabatan

Nama Field	Tipe Data	Ukuran	Keterangan
IDJabatan	int	5	Id jabatan
namaJabatan	varchar	50	Nama jabatan
jumlahTunjangan	int	7	Tunjangan jabatan
keteranganJabatan	varchar	50	Keterangan jabatan

3. Tabel Jurusan

Nama tabel : tbl_jurusan

Type File : Master

Primary Key : IDJurusan

Foreign Key :-

Tabel 4.24 Tabel Jurusan

Nama Field	Tipe Data	Ukuran	Keterangan
IDJurusan	int	5	Id jurusan
namaJurusan	varchar	50	Nama jurusan
keteranganJurusan	varchar	50	Keterangan jurusan

4. Tabel Honor

Nama tabel : tbl_honor

Type File : Master

Primary Key : IDHonor

Foreign Key :-

Tabel 4.25 Tabel Honor

Nama Field	Tipe Data	Ukuran	Keterangan
IDHonor	int	5	Id honor
honorPerJam	int	7	Nominal honor
transportPerJam	int	7	Nominal transport
piketPerHari	int	7	Nominal piket

5. Tabel Pegawai

Nama tabel : *tbl_pegawai*

Type File : *Master*

Primary Key : *IDPegawai*

Foreign Key : *IDUser, IDJabatan, IDJurusan*

Tabel 4.26 Tabel Pegawai

Nama Field	Tipe Data	Ukuran	Keterangan
<i>IDPegawai</i>	<i>int</i>	5	Id pegawai
<i>namaPegawai</i>	<i>varchar</i>	50	Nama pegawai
<i>jenisKelamin</i>	<i>tinyint</i>	1	Jenis kelamin
<i>tempatLahir</i>	<i>varchar</i>	50	Tempat lahir
<i>tanggalLahir</i>	<i>date</i>	0	Tanggal lahir
<i>alamatPegawai</i>	<i>varchar</i>	150	Alamat
<i>teleponPegawai</i>	<i>varchar</i>	30	No telepon
<i>pendidikanTerakhir</i>	<i>varchar</i>	25	Pendidikan terakhir
<i>email</i>	<i>varchar</i>	30	Email pegawai
<i>mulaiTugas</i>	<i>date</i>	0	Mulai bertugas
<i>IDJabatan</i>	<i>int</i>	5	Id jabatan
<i>IDUser</i>	<i>int</i>	5	Id user
<i>IDJurusan</i>	<i>int</i>	5	Id jurusan
<i>Username</i>	<i>varchar</i>	25	Username untuk user
<i>Password</i>	<i>varchar</i>	25	Password untuk user
<i>loginTerakhir</i>	<i>datetime</i>	0	Login terakhir
<i>IP</i>	<i>varchar</i>	50	IP komputer
<i>StatusPegawai</i>	<i>tinyint</i>	1	Status pegawai

6. Tabel Tahun Pelajaran

Nama tabel : tbl_tahunpelajaran

Type File : Master

Primary Key : IDTahunPelajaran

Foreign Key :-

Tabel 4.27 Tabel Tahun Pelajaran

Nama Field	Tipe Data	Ukuran	Keterangan
IDTahunPelajaran	int	5	Id tahun pelajaran
namaTahunPelajaran	varchar	25	Nama tahun pelajaran
keteranganTahunPelajaran	varchar	25	Keterangan tahunpelajaran
statusTahunPelajaran	tinyint	1	Status tahun pelajaran

7. Tabel Waktu Mengajar

Nama tabel : tbl_waktu mengajar

Type File : Master

Primary Key : IDWaktuMengajar

Foreign Key : IDTahunPelajaran, IDPegawai

Tabel 4.28 Tabel Waktu Mengajar

Nama Field	Tipe Data	Ukuran	Keterangan
IDWaktuMengajar	int	5	Id waktu mengajar
IDTahunPelajaran	int	5	Id tahun pelajaran
IDPegawai	int	5	Id pegawai
jumlahWaktuMengajar	int	3	Jumlah waktu mengajar

8. Tabel Piket

Nama tabel : tbl_piket

Type File : Transaction

Primary Key : IDPiket

Foreign Key : IDTahunPelajaran, IDJurusan, IDPegawai

Tabel 4.29 Tabel Piket

Nama Field	Tipe Data	Ukuran	Keterangan
IDPiket	int	5	Id piket
IDJurusan	int	5	Id jurusan
IDTahunPelajaran	int	5	Id tahun pelajaran
tanggalPiket	date	0	Tanggal piket
IDPegawai	int	5	Id pegawai
keteranganPiket	varchar	50	Keterangan piket

9. Tabel Absensi

Nama tabel : tbl_absensi

Type File : Transaction

Primary Key : IDAbsensi

Foreign Key : IDTahunPelajaran, IDJurusan, IDPegawai, IDPiket

Tabel 4.30 Tabel Absensi

Nama Field	Tipe Data	Ukuran	Keterangan
IDAbsensi	Int	5	Id absensi
IDJurusan	Int	5	Id Jurusan
IDTahunPelajaran	Int	5	Id tahun pelajaran
tanggalAbsensi	Date	0	Tanggal absensi
IDPegawai	Int	5	Id pegawai
Sesi	Int	2	Sesi

10. Tabel Penggajian

Nama tabel : *tbl_penggajian*

Type File : *Transaction*

Primary Key : *IDPenggajian*

Foreign Key : *IDPegawai, IDHonor, IDPiket, IDAbsensi*

Tabel 4.31 Tabel Penggajian

Nama Field	Tipe Data	Ukuran	Keterangan
<i>IDPenggajian</i>	<i>Int</i>	5	Id penggajian
<i>IDPegawai</i>	<i>Int</i>	5	Id pegawai
<i>IDHonor</i>	<i>Int</i>	5	Id honor
<i>IDPiket</i>	<i>Int</i>	5	Id piket
<i>IDAbsensi</i>	<i>Int</i>	5	Id absensi

4.4.6 Perancangan Interface

Berikut ini adalah beberapa contoh tampilan rancangan *interface* dari sistem informasi penggajian berdasarkan level pengguna (*user*) berikut dengan *feature* sistem nya:

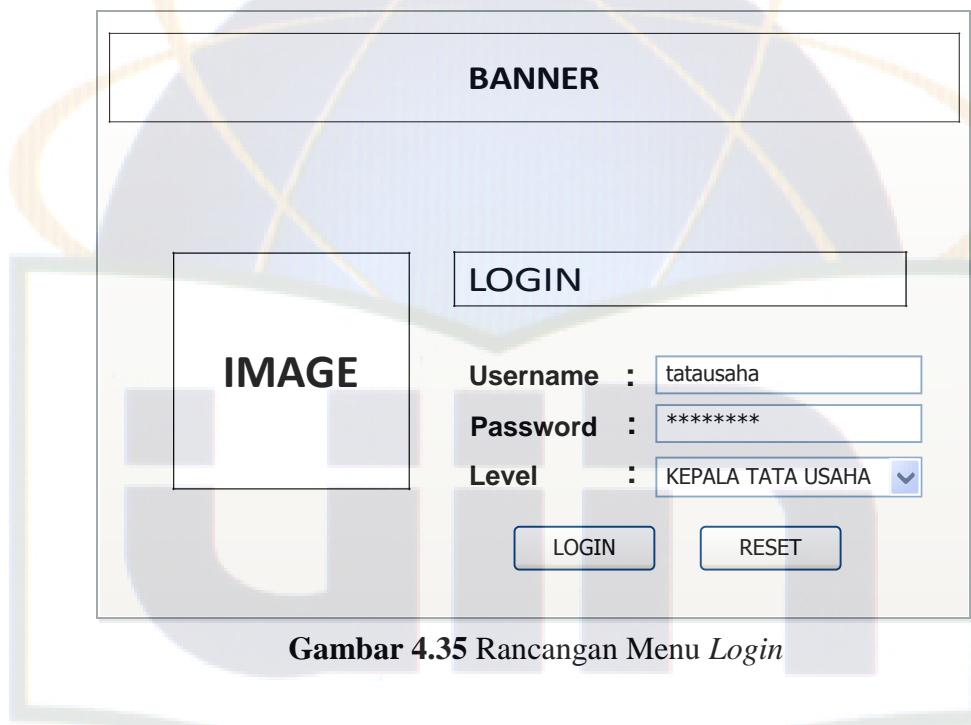
Tabel 4.32 Feature System

User	Feature
Kepala Tata Usaha	<ul style="list-style-type: none"> a. <i>Login</i> b. Mengelola Data <i>User</i> c. Mengelola Data Jabatan d. Mengelola Data Jurusan e. Mengelola Data Honor f. Mengelola Data Pegawai g. Mengelola Tahun Pelajaran h. Mengelola Waktu Mengajar i. Mengubah <i>Password</i>

	j. <i>Logout</i>
Guru Piket	a. <i>Login</i> b. Mengisi absensi c. Melihat Rekap Absensi d. Mengubah password e. <i>Logout</i>
Bendahara	a. <i>Login</i> b. Melihat Rekap Absensi c. Melihat dan mencetak laporan gaji d. Mengubah password e. <i>Logout</i>
Ketua Yayasan	a. <i>Login</i> b. Melihat Data Pegawai c. Melihat Rekap Absensi d. Melihat laporan gaji e. Mengubah password nya f. <i>Logout</i>
Pegawai	a. <i>Login</i> b. Melihat Rekap Absensi nya c. Melihat dan mencetak slip gaji nya d. Mengubah Password e. <i>Logout</i>

1. Rancangan Menu *Login*

Pada gambar 4.35 berikut ini merupakan tampilan untuk *login* agar *user* dapat masuk ke sistem informasi penggajian.



The diagram illustrates the layout of a login screen. At the top is a banner area labeled "BANNER". Below it is a large rectangular input field labeled "IMAGE". To the right of the "IMAGE" field is a smaller rectangular input field labeled "LOGIN". Inside the "LOGIN" field, there are three input boxes: "Username : tatausaha", "Password : *****", and "Level : KEPALA TATA USAHA" with a dropdown arrow. Below these input fields are two blue rectangular buttons labeled "LOGIN" and "RESET".

Gambar 4.35 Rancangan Menu *Login*

2. Rancangan Halaman Utama Level Kepala Tata Usaha

Pada gambar 4.36 berikut ini tampil pada saat kepala tata usaha telah melakukan *login* terlebih dahulu pada halaman *login*. Di dalam tampilan ini terdapat menu-menu yang ditampilkan seperti halaman utama, data master, data pegawai, data tahun pelajaran, data waktu mengajar, profil, ubah *password*, dan *logout*.



Gambar 4.36 Rancangan Halaman Utama Level Kepala Tata Usaha

3. Rancangan Halaman Utama Level Guru Piket

Pada gambar 4.37 berikut ini tampil pada saat guru piket telah melakukan *login* terlebih dahulu pada halaman *login*. Di dalam tampilan ini terdapat menu-menu yang ditampilkan seperti halaman utama, isi absensi, rekap absensi, ubah *password*, *logout*.



Gambar 4.37 Rancangan Halaman Utama Level Guru Piket

4. Rancangan Halaman Utama Bendahara

Pada gambar 4.38 berikut ini tampil pada saat bendahara telah melakukan *login* terlebih dahulu pada halaman *login*. Di dalam tampilan ini terdapat menu-menu yang ditampilkan seperti halaman utama, rekap absensi, laporan gaji, profil, ubah *password*, dan *logout*.



Gambar 4.38 Rancangan Halaman Utama Level Bendahara

5. Rancangan Halaman Utama Level Ketua Yayasan

Pada gambar 4.39 berikut ini tampil pada saat bendahara telah melakukan *login* terlebih dahulu pada halaman *login*. Di dalam tampilan ini terdapat menu-menu yang ditampilkan seperti halaman utama, rekap absensi, laporan gaji, ubah *password*, dan *logout*.



Gambar 4.39 Rancangan Halaman Utama Level Ketua Yayasan

6. Rancangan Halaman Utama Level Pegawai

Pada gambar 4.40 berikut ini tampil pada saat pegawai telah melakukan *login* terlebih dahulu pada halaman *login*. Di dalam tampilan ini terdapat menu-menu yang ditampilkan seperti halaman utama, laporan gaji, dan *logout*.



Gambar 4.40 Rancangan Halaman Utama Level Pegawai

7. Rancangan Halaman *Input* Data Pegawai

Pada gambar 4.41 berikut ini adalah tampilan dari menu tambah pegawai. Pada tampilan dibawah ini *user* dapat mengisikan data-data pegawai. Menu ini hanya bisa diakses oleh *user* level kepala tata usaha.

BANNER	
Halaman Utama	Data Master
Data Pegawai	Tahun Pelajaran
Waktu Mengajar	Profil
Ubah Password	Logout

Tambah Pegawai

Pegawai

Nama Pegawai *	:	<input type="text"/>
Jenis Kelamin *	:	<input type="text"/>
Tempat Lahir *	:	<input type="text"/>
Tanggal Lahir *	:	<input type="text"/>
Alamat *	:	<input type="text"/>
Telepon Pegawai *	:	<input type="text"/>
Pendidikan Terakhir *	:	<input type="text"/>
Email *	:	<input type="text"/>
Mulai Tugas *	:	<input type="text"/>
Jabatan *	:	<input type="text"/>
Jurusan *	:	<input type="text"/>

Sistem

Level *	:	<input type="text"/>
Username *	:	<input type="text"/>
Password *	:	<input type="text"/>
Status	:	<input checked="" type="radio"/> Aktif <input type="radio"/> Tidak Aktif

Simpan **Kembali**

Gambar 4.41 Rancangan Halaman *Input* Data Pegawai

8. Rancangan Halaman Lihat Data Pegawai

Pada gambar 4.42 berikut ini adalah tampilan dari menu lihat pegawai. Pada tampilan dibawah ini *user* dapat melakukan proses cari, ubah dan hapus data pegawai. Menu ini hanya bisa diakses oleh *user* level kepala tata usaha.

Gambar 4.42 Rancangan Halaman Lihat Data Pegawai

9. Rancangan Halaman Input Absensi

Pada gambar 4.43 berikut ini adalah tampilan dari menu *input absensi*. Pada tampilan dibawah ini *user* dapat mengisi data-data absensi. Menu ini hanya bisa diakses oleh *user* level guru piket.

No.	Nama Pegawai	Sesi							
		1	2	3	4	5	6	7	8
1	Agus Sapardin, S.Pd	<input type="checkbox"/>							
2	Eli Nurhelawati, Dra	<input type="checkbox"/>							
3	Halimudin, S.Pd	<input type="checkbox"/>							
4	Herawati, S.Pd	<input type="checkbox"/>							
5	Lili Ristina, S.Pd	<input type="checkbox"/>							

Gambar 4.43 Rancangan Halaman Input Absensi

10. Rancangan Halaman Rekap Absensi Bulanan

Pada gambar 4.44 berikut ini adalah tampilan dari menu rekap absensi. Pada tampilan dibawah ini *user* dapat melihat rekap absensi pada bulan tersebut. Menu ini dapat diakses oleh *user* level guru piket, bendahara, dan ketua yayasan.

NO.	NAMA PEGAWAI	TANGGAL	JUMLAH HADIR
1	Agus Sapidin, S.Pd	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	1 2 1 4
2	Ahmad, S.Ag		
3	Asnawi Mursan		
4	Atje Holidimjati, B.Sc		
5	Dadang Juarna, S.Pd		

Gambar 4.44 Rancangan Halaman Rekap Absensi Bulanan

11. Rancangan Halaman Laporan Penggajian

Pada gambar 4.45 berikut ini adalah tampilan dari menu laporan gaji. Pada tampilan dibawah ini *user* dapat melihat laporan penggajian pada bulan tersebut. Menu ini dapat diakses oleh *user* level bendahara, dan ketua yayasan.

BANNER

[Halaman Utama](#)
[Rekap Absensi](#)
[Laporan Gaji](#)
[Profil](#)
[Ubah Password](#)
Logout

Lihat Laporan

Laporan Penggajian

Bulan :	<input type="button" value="Juni"/>	Tahun :	<input type="text"/>	<input type="button" value="Cari"/>
---------	-------------------------------------	---------	----------------------	-------------------------------------

LAPORAN PENGGAJIAN BULAN JUNI TAHUN 2011

No.	Nama Pegawai	Jumlah Jam	Honor Pokok	Jumlah Hadir	Honor Transport	Jumlah Piket	Honor Piket	Tunjangan					Jumlah Penerimaan Honor
								Wali Kelas	K3	OSIS	Bendahara	TU	
1	Agus Sapidin, S.Pd			4	20000								20000
2	Ahmed, S.Ag												
3	Asnawi Mursan												
4	Atjep Holidimjati, B.Sc	10	200000										600000
5	Dadang Juarna, S.Pd												800000

Gambar 4.45 Rancangan Halaman Laporan Penggajian

12. Rancangan Halaman Slip Gaji

Pada gambar 4.46 berikut ini adalah tampilan dari menu laporan gaji. Pada tampilan dibawah ini *user* dapat melihat slip gaji pada bulan tersebut. Menu ini dapat diakses oleh *user* level bendahara.

BANNER

[Halaman Utama](#)
[Rekap Absensi](#)
[Laporan Gaji](#)
[Profil](#)
[Ubah Password](#)
Logout

Lihat Laporan

Nama : Atjep Holidimjati, B.Sc
Bulan : Juni 2011

PENERIMAAN

No.	Keterangan	Jam	Honor/Jam	Jumlah
1	Honor	10	20000	200000
2	Transport		5000	
3	Tunjangan:			600000
	* Kepsek * Wakepsek * Pembina OSIS * Kel. TU * Bendahara * K3 * Wali Kelas			
4	Insentif Piket		20000	20000
	Total Penerimaan			800000

Jakarta, Juli 2011
BENDAHARA

(Laelatul Badriah, Dra)

Gambar 4.46 Rancangan Halaman Slip Gaji

4.5 Implementasi Sistem (*Construction & Testing*)

4.5.1 Pemrograman

Design Interface sistem yang telah dimodelkan dan dirancang sebelumnya dieksekusi menjadi sebuah set kode program dengan menggunakan bahasa pemrograman PHP guna mengimplementasikan sistem informasi penggajian.

Sejumlah *tools* digunakan untuk mengembangkan aplikasi sistem informasi penggajian. Pada tabel berikut ditunjukkan daftar *tools* yang digunakan dalam pengembangan aplikasi.

Tabel 4.33 Daftar *Tools* Pengembangan Sistem

No.	Tool	Kegunaan
1	<i>Microsoft Visio 2007</i>	<i>Software</i> perancangan sistem.
2	XAMPP 1.7.1	<i>Web server.</i>
3	PHP 5.2.9	Bahasa pengembangan perangkat lunak.
4	<i>MySQL 5.1.33</i>	<i>Database</i> yang digunakan dalam sistem.
5	<i>PhpMyAdmin</i>	Perangkat untuk mengakses <i>database</i> yang terdapat pada XAMPP Apache.
6	Macromedia Dreamweaver 8	<i>Software editor</i>
7	<i>Editplus 2</i>	Perangkat <i>editor</i> pembuatan <i>program</i> .
8	<i>Navicat Premium 8.2</i>	Perangkat untuk mengakses <i>database</i> .
9	<i>Mozilla Firefox 4.0.1</i>	<i>Web browser.</i>
10	<i>Adobe Photoshop CS 2</i>	<i>Software</i> meng-edit <i>image</i> .

4.5.2 Perangkat Sistem

Dalam pengimplementasian sistem informasi penggajian, penulis mengusulkan penggunaan *hardware*, *software*, dan *network* dengan spesifikasi dan kebutuhan sebagai berikut:

1. *Hardware*

Pada tabel berikut ini dapat dilihat seperti apa spesifikasi *hardware* untuk menjalankan perangkat lunak sistem informasi penggajian.

Tabel 4.34 Spesifikasi *Hardware*

SERVER	
Perangkat	Deskripsi
PC	1. Processor Intel Core 2 Duo 2,8 GHz 2. Memory 2 GB DR2 PC 6400 3. Hard-disk 250GB SATA 7200 rpm
Network	Switch, Modem, LAN Card on board, UTP cable, RJ 45 connector
<i>Input Devices</i>	Keyboard, Mouse
<i>Output Devices</i>	LCD 17 inch, Printer
CLIENT	
Perangkat	Deskripsi
PC	1. Processor Intel Core 2 Duo 2,8 GHz 2. Memory 2 GB DR2 PC 6400 3. Hard-disk 250GB SATA 7200 rpm
Network	LAN Card on board, UTP cable, RJ 45 connector
<i>Input Devices</i>	Keyboard, Mouse
<i>Output Devices</i>	LCD 17 inch

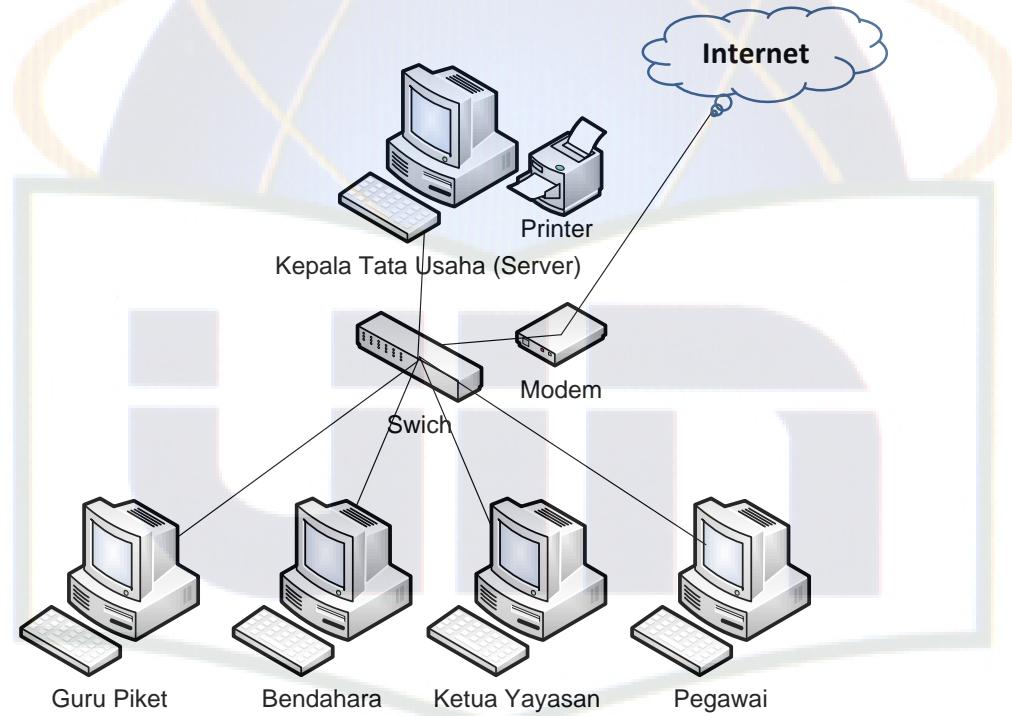
2. *Software*

Berikut ini merupakan *software* yang dibutuhkan dalam menjalankan sistem informasi penggajian:

- a. *Operating System* : Windows XP SP2 or above.
- b. *Webserver* : XAMPP 1.7.1 dengan Apache 2.2.9, PHP 5.2.9, MySQL 5.1.33.
- c. *Browser* : Mozilla Firefox 3.0 (or above).

3. Network

- a. Aristekturn jaringan : *Client-Server*
- b. Jenis Jaringan : *Internet*
- c. *Bandwidth Internet* : 1Mbps



Gambar 4.47 Konfigurasi Jaringan

4.5.3 Pengujian Perangkat Lunak

Pengujian sistem atau aplikasi sistem dilakukan dengan menggunakan metode *black-box testing*. Pengujian ini dimaksudkan untuk mengetahui apakah *output* yang dihasilkan dari pengolahan sistem benar-benar sesuai dengan *output* yang diharapkan oleh pengguna sistem informasi pengajaran.

Tabel 4.35 Pengujian Sistem

No	Rancangan Proses	Hasil yang Diharapkan	Hasil	Keterangan
1	<i>Input address</i> aplikasi http://localhost/penggajian-web-app	Menampilkan halaman <i>form login</i>	OK	
2	Mengisi <i>username</i> , <i>password</i> , level dan klik tombol “Login”	Masuk halaman utama <i>account</i> kepala tata usaha	OK	
3	Pilih Menu Data User →Tambah User	Menampilkan halaman menu tambah user	OK	
4	Isikan data user di <i>form</i> isian data user kemudian pilih tombol “Simpan”	Menampilkan <i>alert</i> verifikasi sukses atau tidak dalam melakukan pengisian	OK	Data tersimpan di <i>database</i> , kemudian data ditampilkan di lihat user.
5	Pilih Menu Data Jabatan →Tambah Jabatan	Menampilkan halaman menu tambah jabatan	OK	
6	Isikan data jabatan di <i>form</i> isian data jabatan kemudian pilih tombol “Simpan”	Menampilkan <i>alert</i> verifikasi sukses atau tidak dalam melakukan pengisian	OK	Data tersimpan di <i>database</i> , kemudian data ditampilkan di lihat jabatan.
7	Pilih Menu Data Jurusan →Tambah Jurusan	Menampilkan halaman menu tambah jurusan	OK	
8	Isikan data jurusan di <i>form</i> isian data jurusan kemudian pilih tombol “Simpan”	Menampilkan <i>alert</i> verifikasi sukses atau tidak dalam melakukan pengisian	OK	Data tersimpan di <i>database</i> , kemudian data ditampilkan di lihat jurusan.
9	Pilih Menu Data Honor	Menampilkan halaman menu input honor	OK	
10	Isikan data honor di <i>form</i> isian data honor kemudian pilih tombol “Simpan”	Menampilkan <i>alert</i> verifikasi sukses atau tidak dalam melakukan pengisian	OK	Data tersimpan di <i>database</i> , kemudian data ditampilkan di update honor.
11	Pilih Menu Data Pegawai →Tambah Pegawai	Menampilkan halaman menu tambah pegawai	OK	
12	Isikan data pegawai di <i>form</i> isian data pegawai kemudian pilih tombol “Simpan”	Menampilkan <i>alert</i> verifikasi sukses atau tidak dalam melakukan pengisian	OK	Data tersimpan di <i>database</i> , kemudian data ditampilkan di lihat pegawai.

13	Pilih Menu Tahun Pelajaran →Tambah Tahun Pelajaran	Menampilkan halaman menu tambah tahun pelajaran	OK	
14	Isi kan data tahun pelajaran di <i>form</i> isian kemudian pilih tombol “Simpan”	Menampilkan <i>alert</i> verifikasi sukses atau tidak dalam melakukan pengisian	OK	Data tersimpan di <i>database</i> , kemudian data ditampilkan di lihat tahun pelajaran.
15	Pilih Menu Waktu Mengajar →Isi Waktu Mengajar	Menampilkan halaman menu tambah waktu mengajar	OK	
16	Isi kan data waktu mengajar di <i>form</i> isian kemudian pilih tombol “Simpan”	Menampilkan <i>alert</i> verifikasi sukses atau tidak dalam melakukan pengisian	OK	Data tersimpan di <i>database</i> , kemudian data ditampilkan di lihat waktu mengajar.
17	Pilih Menu Isi Absensi →Jurusan	Menampilkan halaman input absensi	OK	
18	Isi kan data absensi di <i>form</i> isian kemudian pilih tombol “Simpan”	Menampilkan <i>alert</i> verifikasi sukses atau tidak dalam melakukan pengisian	OK	Data tersimpan di <i>database</i> .
19	Pilih Menu Rekap Absensi → Harian	Menampilkan halaman lihat absensi	OK	
20	Isi <i>form</i> tanggal absensi atau nama pegawai → pilih tombol cari	Menampilkan halaman data rekap absensi harian	OK	
21	Pilih Menu Rekap Absensi → Bulanan	Menampilkan halaman lihat absensi	OK	
22	Isi <i>form</i> bulan atau nama pegawai → pilih tombol cari	Menampilkan halaman data rekap absensi bulanan	OK	
23	Pilih Menu Laporan Gaji	Menampilkan halaman lihat laporan	OK	
24	Isi <i>form</i> tahun pelajaran dan bulan → pilih tombol cari	Menampilkan laporan penggajian sesuai bulan yang dipilih	OK	

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan dari hasil pembahasan pengembangan sistem informasi penggajian yang telah dilakukan, maka dapat diperoleh simpulan sebagai berikut:

1. Sistem informasi penggajian yang diusulkan mampu mempersingkat waktu pengolahan data penggajian dan mengurangi masalah *human error* dalam penghitungan absensi dan gaji.
2. Sistem informasi penggajian ini menghasilkan laporan-laporan penggajian dengan cepat dan akurat, diharapkan pembayaran gaji pun dapat dipercepat.
3. Sistem informasi penggajian ini memberikan informasi penghasilan kepada setiap pegawai, sehingga pegawai dapat melihat dan mencetak penghasilannya melalui slip gaji.
4. Sistem informasi penggajian yang diusulkan terdapat pembatasan hak akses terhadap sistem sehingga dapat menghindari pihak-pihak yang tidak berwenang untuk mengakses data dengan bebas.

5.2 Saran

Berdasarkan simpulan dan analisis yang telah dilakukan, maka terdapat saran-saran sebagai berikut:

1. Penelitian ini dapat dijadikan referensi untuk penelitian selanjutnya di bidang penggajian, khususnya lingkup lembaga pendidikan (sekolah).
2. Sistem informasi penggajian ini berfokus pada penggajian dan tidak membahas penghitungan pajak, untuk penelitian selanjutnya dapat diintegrasikan dengan fungsi keuangan lainnya dan dapat membahas penghitungan pajak.
3. Sistem informasi penggajian ini pembayaran gaji secara tunai, untuk penelitian selanjutnya pembayaran gaji dapat ditransfer melalui bank.

DAFTAR PUSTAKA

- Adisu, Editus. 2008. *Hak karyawan atas gaji & pedoman menghitung : gaji pokok, uang lembur, gaji sundulan, insentif-bonus-THR, pajak atas gaji, iuran jamsostek,/ dana sehat.* Jakarta: Forum Sahabat.
- Al Fatta, Hanif. 2007. *Analisis dan Perancangan Sistem Informasi: untuk Keunggulan Bersaing Perusahaan & Organisasi Modern.* Yogyakarta: Penerbit Andi.
- Arbai. 2004. *Manajemen Database dengan MySQL.* Yogyakarta: Penerbit Andi
- Fowler, Martin. 2004. *UML Distilled Edisi ke-3.* Yogyakarta: Penerbit Andi.
- Gulo, W. 2002. *Metodologi Penelitian.* Jakarta: PT. Grasindo.
- Hariyanto, Bambang. 2004. *Rekayasa Sistem Berorientasi Objek.* Bandung: Informatika.
- Hariyanto, Bambang. 2004. *Sistem Manajemen Basis Data : Pemodelan, Perancangan, dan Terapannya.* Bandung: Informatika.
- Hasibuan, Zainal A. 2007. *Metodologi Penelitian Pada Bidang Ilmu Komputer dan Teknologi Informasi.* Jakarta: Fakultas Ilmu Komputer Universitas Indonesia.
- Jogiyanto, H.M. 2005. *Analisis & Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis.* Yogyakarta: Penerbit Andi.
- Kadir, Abdul. 2003. *Pengenalan Sistem Informasi.* Yogyakarta: Penerbit Andi.
- Ladjamudin, Al-Bahra bin. 2005. *Analisis dan Desain Sistem Informasi.* Yogyakarta: Graha Ilmu.

- Mathiassen L, Munk-Madsen A, Nielsen PA, Stage J. 2000. *Object Oriented Analysis & Design*. Aalborg Denmark: Marko Publishing ApS.
- Mulyadi. 2001. *Sistem Akuntansi*. Jakarta: Salemba Empat.
- Mulyadi. 2008. *Perancangan Aplikasi Penggajian Karyawan Pada Yayasan Darul Irfan*. Jakarta: UIN Jakarta.
- Nazir, Moh. 2005. *Metode Penelitian*. Bogor: Ghalia Indonesia.
- Nugroho, Adi. 2005. *Analisa dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*. Bandung: Informatika.
- Nugroho, Adi. 2005. *Konsep Pengembangan Sistem Basis Data*. Bandung: Informatika.
- Oetomo, Budi Sutedjo Dharma. 2007. *Pengantar Teknologi Internet: Konsep dan Aplikasi*. Yogyakarta: Penerbit Andi.
- Samsudin, Sadili. 2006. *Manajemen Sumber Daya Manusia*. Bandung: Pustaka Setia.
- Sardi, Irawan. 2004. *Manajemen, Desain, dan Pengembangan Situs Web dengan Macromedia Dreamweaver MX dan Adobe Photoshop 7*. Jakarta: Elex Media Komputindo.
- Sidik, Betha. 2005. *MySQL untuk Pengguna, Administrator, dan Pengembang Aplikasi Web*. Bandung: Informatika.
- Subana, H.M, Sudrajat. 2005. *Dasar-Dasar Penelitian Ilmiah*. Bandung: CV. Pustaka Setia.
- Sukarno, Muhammad. 2006. *Membangun Website Dinamis Interaktif dengan PHP-MySQL (Windows & Linux)*. Jakarta: Eska Media

- Sumarsono, Sonny. 2003. *Ekonomi Manajemen Sumber Daya Manusia dan Ketenagakerjaan*. Yogyakarta: Graha Ilmu.
- Umar, Husaein. 2007. *Metode Penelitian*. Jakarta : Raja Grafindo.
- Wahyono, Teguh. 2003. *Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Whitten, Bentley, Dittman. 2004. *Metode Desain & Analisis Sistem*. Yogyakarta: Penerbit Andi.
- Widodo, Priyono Dwi. 2005. *Kamus Istilah Internet dan Komputer*. Jombang: Lintas Media.







YAYASAN AL-FIRDAUS PEGADUNGAN

SMK AL-FIRDAUS

Jl. Peta Utara/Masjid AL-Firdaus Kel. Pegadungan, Kalideres, Jakarta Barat 11830
Telp./FAX. 021-54399864, 5402047 Email : alfirdausSMK@yahoo.com

SURAT KETERANGAN PENELITIAN

Assalamu'alaikum Wr. Wb

Saya yang bertanda tangan dibawah ini,

Nama : Nabhani, SE

Jabatan : Kepala Tata Usaha SMK Al-Firdaus Jakarta

Menerangkan bahwa,

Nama : Sigit Prasetyo

NIM : 104093003010

Mahasiswa Program Studi Sistem Informasi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

Telah mengadakan penelitian, observasi, dan wawancara dalam rangka untuk kepentingan skripsi yang berjudul **Pengembangan Sistem Informasi Penggajian (Studi Kasus: SMK Al-Firdaus Jakarta)**.

Demikian surat keterangan ini dibuat dengan sebenar-benarnya dan dapat digunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Jakarta, 6 Juni 2011

Mengetahui

Kepala/Tata Usaha





Laporan Hasil Observasi Lapangan

Lokasi : SMK Al-Firdaus Jakarta

Waktu Observasi : 3 Januari 2011 – 31 Januari 2011

1. Kegiatan Harian Pegawai SMK Al-Firdaus Jakarta
 - a. Memulai kegiatan belajar mengajar (KBM) pada pukul 06:30 WIB.
 - b. Pegawai staf yang datang mengisikan absensi nya pada daftar absensi staf, dan semua mulai bekerja sesuai *job desk* nya masing-masing.
 - c. Bagi guru yang datang hanya sebagian karena waktu kerja nya menyesuaikan jadwal guru tersebut mengajar.
 - d. Kegiatan kegiatan belajar mengajar (KBM) pada pukul 14:00 WIB, pegawai staf mengisikan kembali jam pulang nya pada daftar absensi staf.
2. Kegiatan Proses Absensi Guru Mengajar
 - a. Bagi guru yang ingin mengajar, guru tersebut mendatangi guru piket untuk mengisikan absensinya pada daftar absensi yang telah tersedia di meja guru piket.
 - b. Setelah kegiatan belajar mengajar (KBM) berakhir, guru piket menandatangi daftar absensi di hari tersebut lalu menyerahkannya ke bagian tata usaha.
3. Kegiatan Proses Pembuatan Rekap Absensi
 - a. Pembuatan rekap absensi dimulai pada tanggal 1 di bulan berikutnya, bendahara menghitung jumlah absensi masing-masing guru dari daftar absensi guru mengajar, hasil penghitungan tersebut dimasukkan pada *form* yang telah tersedia pada Microsoft Excel.
 - b. Biasanya pembuatan rekap absensi selesai dalam 3 hari, namun jika pekerjaan sedang banyak bisa mencapai 4 hari.



4. Kegiatan Proses Pembuatan Laporan Penggajian
 - a. Pembuatan laporan penggajian dimulai setelah pembuatan rekap absensi selesai, karena sebagian sumber datanya berasal dari jumlah absensi guru.
 - b. Pembuatan laporan penggajian dibuat dengan menghitung jumlah gaji tiap pegawai lalu memasukkan data-datanya pada *form* yang telah tersedia pada *Microsoft Excel*.
 - c. Biasanya pembuatan laporan penggajian selesai dalam 4 hari, bahkan bisa sampai 5 hari karena lebih memerlukan konsentrasi dan ketelitian.
5. Kegiatan Proses Penyerahan Laporan
 - a. Jika pembuatan laporan penggajian telah selesai, maka laporan penggajian serta rekap absensi dicetak untuk diserahkan ke ketua yayasan untuk mendapatkan persetujuan pengeluaran kas.
 - b. Jika laporan tersebut telah sesuai, maka akan ketua yayasan memberikan paraf sebagai tanda telah disetujui, lalu difotokopi sebagai arsip yayasan kemudian dokumen tersebut diserahkan kembali kepada bendahara.
6. Kegiatan Proses Pembayaran Gaji
 - a. Setelah laporan penggajian disetujui, bendahara kemudian membuatkan slip gaji tiap pegawai menggunakan *Microsoft Excel*, sumber datanya berasal dari laporan penggajian. Kemudian slip gaji tersebut dicetak pada sebuah amplop.
 - b. Pembayaran gaji biasanya pada tanggal 10, dengan menyerahkan uang gaji di dalam amplop yang telah tercetak slip gajinya sebagai rincian penghasilan.





WAWANCARA

Wawancara I

Narasumber : Bapak Nabhani

Jabatan : Kepala Tata Usaha

Penanya : Sigit Prasetyo

Waktu : 10 Januari 2011

Tema : Prosedur sistem berjalan.

Tujuan : Mengetahui alur kerja sistem penggajian yang sedang berjalan.

Pertanyaan:

1. Bagaimana alur kerja sistem penggajian yang sedang berjalan pada SMK Al-Firdaus?
2. Berapakah nominal dari tiap gaji dan tunjangan yang berlaku?
3. Apakah pengelolaan sistem penggajian yang sedang berjalan sudah terkomputerisasi?
4. Masalah apa saja yang terjadi dengan alur penggajian saat ini?

Hasil Wawancara:

1. Alur sistem penggajian saat ini:
 - a) Di awali dari tahap absensi pegawai. Bagi guru dengan mengisi daftar absensi mengajar. Sedangkan bagi staf itu mengisi absensi staf, namun sifatnya hanya sebagai penilaian kinerja.
 - b) SMK Al-Firdaus melakukan pembayaran gaji dalam periode 1 bulan sekali dengan rincian: honor pokok, tunjangan transportasi, tunjangan piket, dan tunjangan jabatan. Pembayaran gaji biasanya setiap tanggal 10 pada bulan berikutnya, maksudnya jika pembayaran pada tanggal 10 februari maka itu untuk pembayaran kerja di bulan januari.
 - c) Periode penghitungan gaji diawali mulai tanggal 1 sampai dengan tanggal terakhir di bulan tersebut.



2. Nominal dari gaji dan tunjangan:
 - a) Honor pokok itu bagi guru atau staf yang mengajar akan mendapatkan honor sebesar Rp. 20000,- (dua puluh ribu rupiah) per jam mengajar. Namun dihitung waktu jumlah mengajar selama 1 minggu (5 hari KBM) untuk periode 1 bulan dan bersifat tetap, yaitu jika terjadi hari libur nasional ataupun hari libur kegiatan belajar mengajar (KBM) akan tetap berhak mendapatkannya secara utuh.
 - b) Tunjangan transportasi itu bagi guru atau staf yang mengajar akan mendapatkan tunjangan sebesar Rp. 5000 (lima ribu rupiah) per jam hadir mengajar, dihitung berdasarkan dari kehadiran mengajar. Jadi jika seorang guru atau staf tidak masuk mengajar atau KBM sedang libur maka tidak berhak mendapatkan tunjangan ini.
 - c) Tunjangan piket itu bagi guru yang bertugas menjadi guru piket, akan mendapatkan tunjangan sebesar Rp. 20000,- (dua puluh ribu rupiah) per hari. Biasanya yang bertugas sebagai guru piket merupakan guru yang tidak ada waktu mengajar di hari tersebut, tiap hari nya hanya ada satu guru piket. Pada periode satu minggu KBM terdapat lima guru piket yang berbeda, namun ke lima orang guru piket sudah ditetapkan bertugas untuk periode satu tahun akademik.
 - d) Tunjangan jabatan itu bagi staf sesuai jabatannya.
3. Proses pengolahan gaji sudah semi terkomputerisasi, maksudnya pengolahan data masih menggunakan kertas kemudian dipindahkan ke Microsoft Excel.
4. Kendala-kendala yang dialami pada pengolahan penggajian:
 - a) Pada proses absensi, setiap awal bulan harus membuat daftar absensi.
 - b) Proses penghitungan absensi, penghitungan gaji masih secara manual (menggunakan kalkulator) yang membutuhkan ketelitian dan waktu yang lama.
 - c) Menurut kami pembayaran pada tanggal 10 di bulan berikutnya, diarasakan terlalu lama. Namun dikarenakan proses pengolahan gaji yang membutuhkan waktu lama, jadi kami memakluminya.



Wawancara II

Narasumber : Bapak Fahrurrozi
Jabatan : Guru piket
Penanya : Sigit Prasetyo
Waktu : 10 Januari 2011
Tema : Prosedur absensi.
Tujuan : Mengetahui alur kerja absensi.

Pertanyaan:

1. Bagaimana proses absensi?

Hasil Wawancara:

1. Setiap hari KBM dimulai dari pukul 06.30 hingga pukul 13.30. Seorang guru yang ingin mengajar mendatangi guru piket untuk mengisi absensi dengan mencatatkannya di daftar absensi. Di meja guru piket terdapat 3 daftar absensi untuk tiap jurusan karena saat ini SMK Al-Firdaus terdapat 3 jurusan yaitu Akuntansi, Administrasi Perkantoran, dan Pemasaran. Setelah KBM di hari tersebut berakhir, guru piket menandatanganinya sebagai bukti telah bertugas dihari tersebut. Kemudian daftar absensi tersebut diserahkan ke kepala tata usaha.



Wawancara III

Narasumber : Ibu Laelatul Badriah
Jabatan : Bendahara
Penanya : Sigit Prasetyo
Waktu : 17 Januari 2011
Tema : Prosedur pengolahan gaji.
Tujuan : Mengetahui alur kerja pengolahan gaji.

Pertanyaan:

1. Bagaimana prosedur perhitungan gaji?

Hasil Wawancara:

1. Prosedur perhitungan gaji:

Pada akhir bulan bendahara menerima daftar absensi yang telah diisi selama satu bulan, kemudian membuat rekap absensi dengan menghitung jumlah absensi dari tiap guru selama satu bulan yang terdapat di daftar absensi dengan mengisikan pada *form* rekap absensi yang telah dibuat. Setelah rekap absensi selesai, lalu dipindahkan ke Microsoft Excel, pembuatan rekap absensi dibatasi sampai tanggal 4 harus selesai.

Penghitungan gaji dimulai setelah pembuatan rekap absensi selesai, dan masih dihitung secara manual menggunakan kalkulator, kemudian memindahkan ke Microsoft Excel untuk dibuatkan laporan penggajian. Pembuatan laporan dibatasi sampai tanggal 8 harus selesai.

Selanjutnya rekap absensi dan laporan penggajian dicetak, kemudian diserahkan ke Ketua Yayasan untuk diperiksa dan disetujui. Setelah disetujui Bendahara membuat slip gaji untuk setiap pegawai di Microsoft Excel dan dicetak pada amplop. Pembayaran gaji dilaksanakan setiap tanggal 10 dengan menyerahkan slip gaji dan menandatangani kolom paraf di laporan penggajian yang telah disetujui ketua yayasan.





LAMPIRAN

INPUT DAN OUTPUT

INPUT:

1. Form Data User

Nama User *	:	<input type="text"/>
Keterangan User *	:	<input type="text"/>

2. Form Data Jabatan

Nama Jabatan *	:	<input type="text"/>
Jumlah Tunjangan *	:	<input type="text"/>
Keterangan Jabatan *	:	<input type="text"/>

3. Form Data Jurusan

Nama Jurusan *	:	<input type="text"/>
Keterangan Jurusan *	:	<input type="text"/>

4. Form Data Honor

ID Honor *	:	<input type="text" value="1"/>
Honor per Jam *	:	<input type="text" value="20000"/>
Transport per Jam *	:	<input type="text" value="5000"/>
Piket per Hari *	:	<input type="text" value="20000"/>

5. Form Data Pegawai

Nama Pegawai *	:	
Jenis Kelamin *	:	<input type="button" value="▼"/>
Tempat Lahir *	:	
Tanggal Lahir *	:	
Alamat Pegawai *	:	
Telepon Pegawai *	:	
Pendidikan Terakhir *	:	
Email *	:	
Mulai Tugas *	:	
Jabatan *	:	KEPALA SEKOLAH <input type="button" value="▼"/>
Jurusan *	:	<input type="button" value="▼"/>
Level *	:	GURU <input type="button" value="▼"/>
Username *	:	
Password *	:	
Status	:	<input checked="" type="radio"/> Aktif <input type="radio"/> Tidak Aktif

6. Form Data Tahun Pelajaran

Tahun Pelajaran *	:	
Keterangan Tahun Pelajaran *	:	
Status Tahun Pelajaran	:	<input checked="" type="radio"/> Aktif <input type="radio"/> Tidak Aktif

7. Form Waktu mengajar

Tahun Pelajaran *	:	Pilih Tahun Pelajaran <input type="button" value="▼"/>
Nama Pegawai *	:	Pilih Pegawai <input type="button" value="▼"/>
Jumlah Waktu Mengajar *	:	

8. Form Absensi

Tahun Pelajaran : 2011/2012
Jurusan : Akuntansi
Tanggal Absensi : Kamis, 11 Agustus 2011
Guru Piket : Pilih

Nama Pegawai	Sesi			
	1	2	3	4
Darma, Drs	■	■	■	■

OUTPUT:

1. Laporan Rekap Absensi Harian

Tanggal Absensi : Kamis, 11 Agustus 2011
Guru Piket : -

NAMA PEGAWAI	SESI

2. Laporan Rekap Absensi Bulanan

Bulan : Januari
Tahun : 2011

NAMA PEGAWAI	TANGGAL

3. Laporan Penggajian

LAPORAN PENGGAJIAN BULAN AGUSTUS TAHUN 2011

Nama Pegawai	Jumlah Jam	Honor Pokok	Jumlah Hadir	Honor Transport	Jumlah Piket	Honor Piket	Tunjangan						
							Wali Kelas	K3	OSIS	Bendahara	TU	Waka	Kepsek



init.inc.php

```
<?php

date_default_timezone_set("Asia/Jakarta");
//error_reporting(E_ALL);

require_once (dirname(__FILE__) . '/config/config.inc.php');
require_once (dirname(__FILE__) . '/classes/MySQL.php');
require_once (dirname(__FILE__) . '/classes/config.class.php');
require_once (dirname(__FILE__) . '/classes/tlPegawai.class.php');
require_once (dirname(__FILE__) . '/classes/tlMail.class.php');
require_once (dirname(__FILE__) . '/classes/tlUser.class.php');
require_once (dirname(__FILE__) . '/classes/tlJabatan.class.php');
require_once (dirname(__FILE__) . '/classes/tlJurusan.class.php');
require_once (dirname(__FILE__) . '/classes/tlTahunPelajaran.class.php');
require_once (dirname(__FILE__) . '/classes/tlHonor.class.php');
require_once (dirname(__FILE__) . '/classes/tlWaktuMengajar.class.php');
require_once (dirname(__FILE__) . '/classes/tlAbsensi.class.php');
require_once (dirname(__FILE__) . '/classes/tlPiket.class.php');

$objSql = new sql_db_mysql(DB_SERVER, DB_USERNAME,
DB_PASSWORD, DB_DATABASE, true);

$config = new Config($objSql);
$tPegawai = new tPegawai($objSql);
$tUser = new tUser($objSql);
$tJabatan = new tJabatan($objSql);
$tJurusan = new tJurusan($objSql);
$tTahunPelajaran = new tTahunPelajaran($objSql);
$tHonor = new tHonor($objSql);
$tWaktuMengajar = new tWaktuMengajar($objSql);
$tAbsensi = new tAbsensi($objSql);
$tPiket = new tPiket($objSql);

?>
```

init.auth.inc.php

```
<?php

/***
 * init.inc.php
 *
 * @package
 * @subpackage
 *
 * PHP Ver : 5.2.9
 */
?>
<?php require_once(dirname(__FILE__) . "/init.inc.php"); ?>
<?php require_once(dirname(__FILE__) . "/init.check.inc.php"); ?>
```

init.check.inc.php

```
<?php

session_start();
if ($tPegawai->checkSession() === false)
{
// $url = $config->get_url();
// $url = $config->selfURL();
// header('Location:
'.ADMIN_URL.'login/index.php?action=flogin&referer='.urlencode( $url
));
header('Location: '. ADMIN_URL . 'login?action=flogin');
exit();
}

if($_SESSION['LANGUAGE'] != "")
{
    require_once (dirname(__FILE__) . "/language/lang." .
$_SESSION['LANGUAGE'] . ".php");
}
else
{
    require_once (dirname(__FILE__) . "/language/lang.id.php");
}
?>
```

init.noaut.inc.php

```
<?php
/**
```

* init.inc.php

```
*
```

```
* @package
* @subpackage
*
* PHP Ver : 5.2.9
*/
?>
<?php require_once(dirname(__FILE__) . "/init.inc.php"); ?>
<?php require_once (dirname(__FILE__) . "/language/lang.id.php"); ?>
```

inc_footer.php

```
<div class="clear">&nbsp;</div>

<div id="comp_footer">
<span id="ct-footer-left">
<?php echo $GLOBALS['footer_copyrights'] ?>
<b><a href="<?php echo $GLOBALS['footer_url'] ?>"><?php echo
$GLOBALS['footer_name'] ?></a></b><br/>
<?php echo $GLOBALS['footer_fst'] ?><br/>
<?php echo $GLOBALS['footer_uin'] ?><br/>
Email: <a href="mailto:<?php echo $GLOBALS['footer_email'] ?>"><?php echo $GLOBALS['footer_email']
?></a>
</span>

<span id="ct-footer-right">
<span id="ct-footer-right-logo">
<a href="http://www.gudanglontar.com">

</a>
</span>
<br/>
Alpha 1.0 |
<span id="validators">
<a href="http://validator.w3.org/check?uri=referer">XHTML
Valid</a> |
<a href="http://jigsaw.w3.org/css-validator/validator-
uri.html">CSS Valid</a>
</span>
</span>
</div><!-- end-of 'container' -->
</body>
</html>
```

inc_footer_popup.php

```
</div><!-- end-of 'container' -->
</body>
</html>
```

inc_header.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="id">
<head>
<title><?php echo TITLE; ?></title>
<link rel="icon" href="<?php echo HOME_URL;
?>/images/favicon.ico"/>
<link rel="stylesheet" href="<?php echo CSS_URL ?>default.css"
type="text/css" media="screen, print"/>
<link rel="stylesheet" href="<?php echo CSS_URL ?>print.css"
type="text/css" media="print"/>
<!-- JQuery -->
<link type="text/css" href="<?php echo SCRIPT_URL
?>jquery/ui.all.css" rel="Stylesheet" />
<script type="text/javascript" src="<?php echo SCRIPT_URL
?>jquery/jquery-1.4.2.min.js"></script>
```

```

<script type="text/javascript" src="php echo SCRIPT_URL ?&gt;jquery/jquery.ui.all.js"&gt;&lt;/script&gt;

&lt;!-- Verticalmenu --&gt;
&lt;script type="text/javascript" src="<?php echo SCRIPT_URL ?&gt;verticalmenu/jqueryslidemenu.js"&gt;&lt;/script&gt;
&lt;link rel="stylesheet" href="<?php echo SCRIPT_URL ?&gt;verticalmenu/jqueryslidemenu.css" type="text/css" media="print, projection, screen" /&gt;

&lt;!-- Colorbox --&gt;
&lt;script type="text/javascript" src="<?php echo SCRIPT_URL ?&gt;colorbox/jquery.colorbox-min.js"&gt;&lt;/script&gt;
&lt;link rel="stylesheet" href="<?php echo SCRIPT_URL ?&gt;colorbox/colorbox.css" type="text/css" media="print, projection, screen" /&gt;
&lt;script type="text/javascript"&gt;
$(document).ready(function(){
    $(".single").colorbox();
    $(".iframeSingle").colorbox({ width:"800px", height:"350px", iframe:true, overlayClose: false});
    $(".iframeReload").colorbox({ width:"75%", height:"90%", iframe:true, onClosed:function(){location.reload(true);}, overlayClose: false});
});
&lt;/script&gt;

&lt;!-- Tablesorter --&gt;
&lt;link rel="stylesheet" href="<?php echo SCRIPT_URL ?&gt;tablesorter/css/jq.css" type="text/css" media="print, projection, screen" /&gt;
&lt;link rel="stylesheet" href="<?php echo SCRIPT_URL ?&gt;tablesorter/themes/blue/style.css" type="text/css" media="print, projection, screen" /&gt;
&lt;script type="text/javascript" src="<?php echo SCRIPT_URL ?&gt;tablesorter/jquery.tablesorter.min.js"&gt;&lt;/script&gt;
&lt;link rel="stylesheet" href="<?php echo SCRIPT_URL ?&gt;tablesorter/addons/pager/jquery.tablesorter.pager.css" type="text/css" media="print, projection, screen" /&gt;
&lt;script type="text/javascript" src="<?php echo SCRIPT_URL ?&gt;tablesorter/addons/pager/jquery.tablesorter.pager.js"&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="<?php echo SCRIPT_URL ?&gt;tablesorter/chili/chili-1.8b.js"&gt;&lt;/script&gt;

&lt;!-- Validate Engine --&gt;
&lt;script type="text/javascript" src="<?php echo SCRIPT_URL ?&gt;validationengine/js/jquery.validationengine.js"&gt;&lt;/script&gt;
&lt;link rel="stylesheet" href="<?php echo SCRIPT_URL ?&gt;validationengine/css/jquery.validationengine.css" type="text/css" media="screen"/&gt;

&lt;meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/&gt;
&lt;meta http-equiv="Pragma" content="no-cache"/&gt;
&lt;meta http-equiv="Cache-Control" content="no-store"/&gt;
&lt;meta http-equiv="Expires" content="0"/&gt;

&lt;meta name="author" content="Annisa Primasari
nisa@gudanglontar.com annisa_pm@yahoo.com"/&gt;
&lt;meta name="author" content="Afrialdi Syahputra Butar Butar
aldi@gudanglontar.com afrialdi_sbb@yahoo.com"/&gt;

&lt;/head&gt;

&lt;body&gt;
&lt;!-- start-of 'container' --&gt;
&lt;div id="container"&gt;

&lt;div id="banner"&gt;
&lt;img src="<?php echo IMAGES_URL ?&gt;banner.png" alt="<?php echo TITLE ?&gt;"/&gt;&lt;/div&gt;

&lt;div id="topmenu"&gt;
&lt;div id="myslidemenu" class="jqueryslidemenu"&gt;
&lt;ul&gt;
&lt;li&gt;&lt;a href="<?php echo ADMIN_URL; ?&gt;&lt;?php echo $GLOBALS['menu_halamanUtama'] ?&gt;&lt;/a&gt;&lt;/li&gt;

&lt;?php if($_SESSION['LEVEL'] == LEVEL_KETUAYAYASAN) { ?&gt;
&lt;li&gt;&lt;a href="<?php echo ADMIN_URL; ?&gt;pegawai/all/"&gt;&lt;?php echo $GLOBALS['menu_pegawai'] ?&gt;&lt;/a&gt;&lt;/li&gt;
&lt;?php } ?&gt;
</pre

```

```

<?php if($_SESSION['LEVEL'] == LEVEL_KEPALATATAUSAHA) { ?>

<li><a href="#"><?php echo $GLOBALS['menu_master'] ?>
    &nbsp;&nbsp;<a href=<?php echo ADMIN_URL;
?>tahunpelajaran/all/"><?php echo $GLOBALS['menu_tahunpelajaran']
?>
    &nbsp;&nbsp;<img class="fr" src=<?php echo SCRIPT_URL
?>verticalmenu/images/down.gif" style="border: 0pt none ;" alt="" /></a>
    <ul>
        <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>tahunpelajaran/all/"><?php echo
$GLOBALS['menu_tahunpelajaran_view'] ?></a></li>
        <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>tahunpelajaran/add/"><?php echo
$GLOBALS['menu_tahunpelajaran_add'] ?></a></li>
    </ul>
    </li>
    <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>waktumengajar/all/"><?php echo $GLOBALS['menu_waktumengajar']
?>
    &nbsp;&nbsp;<img class="fr" src=<?php echo SCRIPT_URL
?>verticalmenu/images/down.gif" style="border: 0pt none ;" alt="" /></a>
    <ul>
        <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>waktumengajar/all/"><?php echo
$GLOBALS['menu_waktumengajar_view'] ?></a></li>
        <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>waktumengajar/add/"><?php echo
$GLOBALS['menu_waktumengajar_add'] ?></a></li>
    </ul>
    </li>
<!--<li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>gurumengajar/all/"><?php echo $GLOBALS['menu_gurumengajar']
?></a></li>-->
<?php } ?>

<?php if($_SESSION['LEVEL'] == LEVEL_PIKET) { ?>
    <li><a href="#"><?php echo $GLOBALS['menu_absensi_add'] ?>
        &nbsp;&nbsp;<img class="fr" src=<?php echo SCRIPT_URL
?>verticalmenu/images/down.gif" style="border: 0pt none ;" alt="" /></a>
        <ul>
            <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>absensi/add/?IDJurusan=3">Administrasi Perkantoran</a></li>
            <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>absensi/add/?IDJurusan=1">Akuntansi</a></li>
            <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>absensi/add/?IDJurusan=2">Pemasaran</a></li>
        </ul></li>
<?php } ?>

<?php if($_SESSION['LEVEL'] != LEVEL_KEPALATATAUSAHA) { ?>
    <li><a href=<?php echo ADMIN_URL; ?>absensi/all/"><?php echo
$GLOBALS['menu_absensi'] ?>
        &nbsp;&nbsp;<img class="fr" src=<?php echo SCRIPT_URL
?>verticalmenu/images/down.gif" style="border: 0pt none ;" alt="" /></a>
        <ul>
            <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>absensi/all/daily/"><?php echo $GLOBALS['menu_absensi_daily']
?></a></li>
            <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>absensi/all/monthly/"><?php echo
$GLOBALS['menu_absensi_monthly'] ?></a></li>
        </ul></li>
<?php } ?>

<?php if($_SESSION['LEVEL'] == LEVEL_BENDAHARA ||
$_SESSION['LEVEL'] == LEVEL_KETUAYAYASAN) { ?>
    <li style="z-index:100;"><a href=<?php echo ADMIN_URL;
?>report/honor/"><?php echo $GLOBALS['menu_report_honor']
?></a></li>
<?php } ?>
    <li><a href=<?php echo ADMIN_URL;
?>pegawai/edit/?info=profile&IDPegawai=<?php echo
$_SESSION['IDPEGAWAI'];?>"><?php echo
$GLOBALS['menu_profile'] ?></a></li>
    <li><a href=<?php echo ADMIN_URL;
?>pegawai/password/?info=profile&IDPegawai=<?php echo
$_SESSION['IDPEGAWAI'];?>"><?php echo
$GLOBALS['menu_change_password'] ?></a></li>
</ul>
</div>
<div id="topmenu-user">
    <span id="topmenu-log"><b><?php echo
$_SESSION['NAMAPEGAWAI']; ?></b> [<a class="topRight"
href=<?php echo ADMIN_URL; ?>action=logout"><?php echo
$GLOBALS['menu_logout'] ?></a>]</span>
</div>
</div>

```

inc_header_popup.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title><?php echo TITLE; ?></title>
    <link rel="icon" href=<?php echo HOME_URL;
?>images/favicon.ico"/>
    <link rel="stylesheet" href=<?php echo CSS_URL ?>default.css"
type="text/css" media="screen, print"/>
    <!-- JQuery -->
    <link type="text/css" href=<?php echo SCRIPT_URL
?>jquery/ui.all.css" rel="Stylesheet" />
    <script type="text/javascript" src=<?php echo SCRIPT_URL
?>jquery/jquery-1.4.2.min.js"></script>
    <script type="text/javascript" src=<?php echo SCRIPT_URL
?>jquery/jquery.ui.all.js"></script>
    <!-- Tablesorter -->
    <link rel="stylesheet" href=<?php echo SCRIPT_URL
?>tablesorter/css/jq.css" type="text/css" media="print, projection, screen"
/>
    <link rel="stylesheet" href=<?php echo SCRIPT_URL
?>tablesorter/themes/blue/style.css" type="text/css" media="print,
projection, screen" />
    <script type="text/javascript" src=<?php echo SCRIPT_URL
?>tablesorter/jQuery.tablesorter.min.js"></script>
    <link rel="stylesheet" href=<?php echo SCRIPT_URL
?>tablesorter addons/pager/jquery.tablesorter.pager.css" type="text/css"
media="print, projection, screen" />
    <script type="text/javascript" src=<?php echo SCRIPT_URL
?>tablesorter addons/pager/jquery.tablesorter.pager.js"></script>
    <script type="text/javascript" src=<?php echo SCRIPT_URL
?>tablesorter/chili/chili-1.8b.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<meta http-equiv="Pragma" content="no-cache"/>
<meta http-equiv="Cache-Control" content="no-store"/>
<meta http-equiv="Expires" content="0"/>
</head>
<body style="min-width:400px;">
<!-- start-of 'container' -->
<div id="container">

```

config.inc.php

```

<?php // $Id: config.inc.php 244 2011-06-26 13:41:08Z annisa.primasari $ ?>
<?php

#
# The ROOT of the Web Application.
# change here to reflect YOUR site configuration.
#
//if (!defined("HOME_URL")) define ("HOME_URL",
"http://demo.gudanglontar.com/");
if (defined("HOME_URL")) define ("HOME_URL",
"http://localhost/penggajian-web-app");

define('TITLE','Penggajian Apps | Sigit Prasetyo ver 1.0');
define('THEMES_NAME', 'default');
define('PAGE_DISPLAY', '10');
define('MAILADDRESS_FROM', 'admin@gudanglontar.com');

#
# Derivative CONSTANTS.
# doesn't necessary need to change if using standard deployment.
#
define ("HOME_DIR", dirname(__FILE__).'/..');

```

```

define ("PUBLIC_DIR",
define ("ADMIN_DIR",
define ("CONFIG_DIR",
define ("INCLUDES_DIR",
define ("COMPONENTS_DIR",
define ("LANGUAGES_DIR",
define ("LIBRARIES_DIR",
define ("THEMES_DIR",
THEMES_NAME . '/');
define ("WEBINF_DIR",
define ("PHOTO_DIR",
define ("PUBLIC_URL"),
HOME_URL . "public/");
if (!defined("ADMIN_URL"))
HOME_URL . "pages/");
if (!defined("CONFIG_URL"))
HOME_URL . "config/");
if (!defined("LIBRARIES_URL"))
("LIBRARIES_URL", HOME_URL . "libs/");
if (!defined("THEMES_URL"))
HOME_URL . "themes/" . THEMES_NAME . "/");
if (!defined("CSS_URL"))
THEMES_URL . "css/");
if (!defined("IMAGES_URL"))
THEMES_URL . "images/");
if (!defined("SCRIPT_URL"))
THEMES_URL . "js/");
if (!defined("PHOTO_URL"))
HOME_URL . "images/photos/");

define ("LOG4PHP_DIR", LIBRARIES_DIR . "Log4Php");
define ("LOG4PHP_CONF", WEBINF_DIR . "log4php.properties");
//log4php.xml
//require_once (LOG4PHP_DIR . '/LoggerManager.php');

$GLOBALS['MAX_SIZE'] = 5242880; //5MB
$GLOBALS['FILE_MIMES'] =
array('image/jpeg','image/jpg','image/gif','image/png');
$GLOBALS['FILE_EXTS'] = array('.jpg', '.png', '.gif', '.jpeg');
$GLOBALS['DELETABLE'] = true;

define('DB_SERVER' , 'localhost');
define('DB_PORT' , '3306');
define('DB_USERNAME' , 'root');
define('DB_PASSWORD' , 'root');
define('DB_DATABASE' , 'db_penggajian');

/*
 * User Level Declaration
 */
define('LEVEL_PUBLIC' , '');
define('LEVEL_PIKET' , 'PIKET');
define('LEVEL_BENDAHARA' , 'BENDAHARA');
define('LEVEL_KEPALATATAUSAHA' , 'KEPALATATAUSAHA');
define('LEVEL_KETUAYAYASAN' , 'KETUAYAYASAN');

/*
 * Table Desclaration
 */
define('TBL_ABSENSI' , 'tbl_absensi');
define('TBL_HONOR' , 'tbl_honor');
define('TBL_JABATAN' , 'tbl_jabatan');
define('TBL_JURUSAN' , 'tbl_jurusan');
define('TBL_PEGAWAI' , 'tbl_pegawai');
define('TBL_PIKET' , 'tbl_piket');
define('TBL_TAHUNPELAJARAN' , 'tbl_tahunpelajaran');
define('TBL_USER' , 'tbl_user');
define('TBL_WAKTUMENGAJAR' , 'tbl_waktumengajar');

?>

config.class.php

<?php

class Config {
    function show_header()
    {
        include_once (THEMES_DIR . "inc_header.php");
        define ("PUBLIC_URL",
        define ("ADMIN_URL",
        define ("CONFIG_URL",
        define ("THEMES_URL",
        define ("CSS_URL",
        define ("IMAGES_URL",
        define ("SCRIPT_URL",
        define ("PHOTO_URL",
        define ("LOG4PHP_DIR", LIBRARIES_DIR . "Log4Php");
        define ("LOG4PHP_CONF", WEBINF_DIR . "log4php.properties");
        //log4php.xml
        //require_once (LOG4PHP_DIR . '/LoggerManager.php');

        public static function hasParameter($parameterName)
        {
            return self::hasGetParameter($parameterName) ||
self::hasPostParameter($parameterName);
        }

        public static function hasGetParameter($parameterName)
        {
            return isset($_GET[$parameterName]) &&
strlen($_GET[$parameterName]) > 0;
        }

        public static function hasPostParameter($parameterName)
        {
            return isset($_POST[$parameterName]) &&
strlen($_POST[$parameterName]) > 0;
        }

        public static function toIndonesianDate( $t )
        {
            $d = date("d ", $t) .
Config::getIndonesianMonth(date("n", $t)) . date(" Y", $t);
return $d;
        }

        public static function toIndonesianFullDate( $t )
        {
            $d = Config::getIndonesianDay(date("N", $t)) . ' . date("d ", $t) .
//          Config::getIndonesianMonth(date("n", $t)) . date(" Y H:i", $t) .
WIB;
            Config::getIndonesianMonth(date("n", $t)) . date(" Y", $t);
return $d;
        }

        public static function toIndonesianTime( $t )
        {
            $d = date("H:i", $t) . ";
return $d;
        }

        public static function toEnglishDate( $t )
        {
            $d = date("d ", $t) .
Config::getEnglishMonth(date("n", $t)) . date(" Y", $t);
return $d;
        }
    }
}

```

```

public static function toEnglishFullDate( $t )
{
    $d = Config::getEnglishDay(date("N", $t)) . ',' . date("d", $t) .
        Config::getEnglishMonth(date("n", $t)) . date(" Y H:i", $t) . '
WIB';
    return $d;
}

public static function toEnglishTime( $t )
{
    $d = date("H:i", $t) . '';
    return $d;
}

public static function getIndonesianMonth($month)
{
    switch( $month )
    {
        case 1: return "Januari";
        case 2: return "Februari";
        case 3: return "Maret";
        case 4: return "April";
        case 5: return "Mei";
        case 6: return "Juni";
        case 7: return "Juli";
        case 8: return "Agustus";
        case 9: return "September";
        case 10: return "Oktober";
        case 11: return "November";
        case 12: return "Desember";
    }
}

public static function getEnglishMonth($month)
{
    switch( $month )
    {
        case 1: return "January";
        case 2: return "February";
        case 3: return "March";
        case 4: return "April";
        case 5: return "May";
        case 6: return "June";
        case 7: return "July";
        case 8: return "August";
        case 9: return "September";
        case 10: return "October";
        case 11: return "November";
        case 12: return "December";
    }
}

public static function getIndonesianDay($day)
{
    switch( $day )
    {
        case 1: return "Senin";
        case 2: return "Selasa";
        case 3: return "Rabu";
        case 4: return "Kamis";
        case 5: return "Jumat";
        case 6: return "Sabtu";
        case 7: return "Minggu";
    }
}

public static function getEnglishDay($day)
{
    switch( $day )
    {
        case 1: return "Monday";
        case 2: return "Tuesday";
        case 3: return "Wednesday";
        case 4: return "Thursday";
        case 5: return "Friday";
        case 6: return "Saturday";
        case 7: return "Sunday";
    }
}

/**
 * @desc Get server current date as valid SQL date format.
 * @return server current date in SQL format
 */
function getCurrentDate()
{
    return date("Y-m-d") . 'T' . date("H:i:s");
}

/**
 * @desc getListMonth
 * @return array
 */
function getListMonth()
{
    $month_array = array(
        "1" => "Januari",
        "2" => "Februari",
        "3" => "Maret",
        "4" => "April",
        "5" => "Mei",
        "6" => "Juni",
        "7" => "Juli",
        "8" => "Agustus",
        "9" => "September",
        "10" => "Oktober",
        "11" => "November",
        "12" => "Desember");
    return $month_array;
}

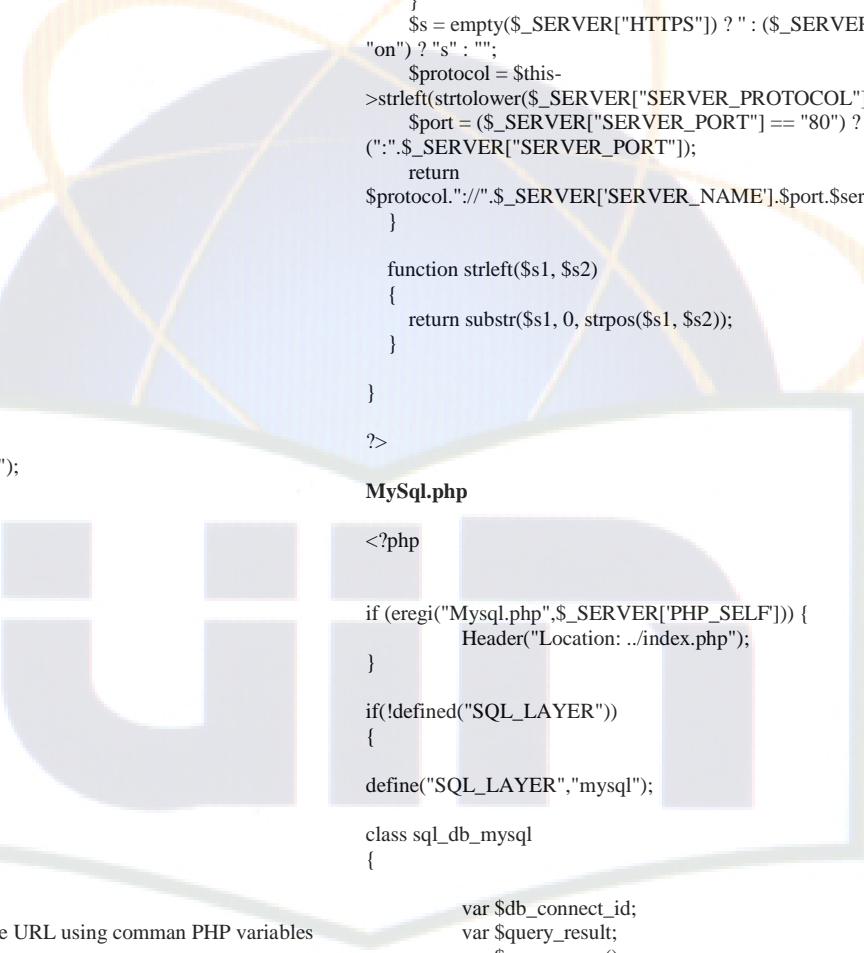
/**
 * @desc getListMonthTA
 * @return array
 */
function getListMonthTA()
{
    $month_array = array(
        "7" => "Juli",
        "8" => "Agustus",
        "9" => "September",
        "10" => "Oktober",
        "11" => "November",
        "12" => "Desember",
        "1" => "Januari",
        "2" => "Februari",
        "3" => "Maret",
        "4" => "April",
        "5" => "Mei",
        "6" => "Juni");
    return $month_array;
}

/**
 * @desc Get User Religion
 * @return array
 */
function getReligion()
{
    $religion_array = array(
        "0" => " ",
        "1" => "Islam",
        "2" => "Kristen katolik",
        "3" => "Kristen Protestan",
        "4" => "Hindu",
        "5" => "Buddha",
        "6" => "Lainnya");
    return $religion_array;
}

/**
 * @desc Get User Sex
 * @return array
 */
function getSex()
{
    $sex_array = array(
        "" => " ",
        "1" => "Laki-laki",
        "2" => "Perempuan");
    return $sex_array;
}

/**
 * @desc getHariKerja
 * @return array
 */
function getHariKerja()
{
    $hariKerja_array = array(
        "0" => "SENIN",

```



```

        "1" => "SELASA",
        "2" => "RABU",
        "3" => "KAMIS",
        "4" => "JUMAT",
        "5" => "SABTU";
    return $hariKerja_array;
}

/**
 * @desc getLanguage
 * @return array
 */
function getLanguage()
{
    $language_array = array(
        "id" => "Bahasa Indonesia",
        "en" => "Bahasa Inggris");
    return $language_array;
}

/**
 * @desc getLevel
 * @return array
 */
function getLevel()
{
    $level_array = array(
        "1" => "PIMPINAN",
        "2" => "ADMINISTRATOR");
    return $level_array;
}

/**
 * @desc getKelas
 * @return array
 */
function getKelas()
{
    $kelas_array = array(
        "0" => "-Pilih Kelas-",
        "1" => "X",
        "2" => "XI",
        "3" => "XII");
    return $kelas_array;
}

/*
Simple function to get current page URL using common PHP variables
Function inputs:
$base if set to true will add the basename to the URL
$www if set to true will add www. to host if not found
$query if set to true will add the query string to the URL
$echo if set to true will echo the URL instead of just returning it
*/
function get_url($base = true, $www = false, $query = true, $echo = false){
    $URL = ""; //open return variable
    $URL .= (($_SERVER['HTTPS'] != "") ? "https://" : "http://"); //get protocol
    $URL .= ((www == true && !preg_match("/^www\./", $_SERVER['HTTP_HOST'])) ? 'www.' : $_SERVER['HTTP_HOST'] : $_SERVER['HTTP_HOST']); //get host
    $path = (($_SERVER['REQUEST_URI'] != "") ? $_SERVER['REQUEST_URI'] : $_SERVER['PHP_SELF']); //tell the function what path variable to use
    $URL .= ((pathinfo($path, PATHINFO_DIRNAME) != '/') ? pathinfo($path, PATHINFO_DIRNAME).'/': pathinfo($path, PATHINFO_DIRNAME)); //set up directory
    $URL .= ((base == true) ? pathinfo($path, PATHINFO_BASENAME) : ""); //add basename
    $URL =
preg_replace("/\?".preg_quote($_SERVER['QUERY_STRING'])."/", "", $URL); //remove query string if found in url
    $URL .= ((query == true && $_SERVER['QUERY_STRING'] != "") ? "?".$_SERVER['QUERY_STRING'] : ""); //add query string

    if($echo == true)
    {
        echo $URL;
    }
    else
    {
        return $URL;
    }
}

function selfURL(){
    if(!isset($_SERVER['REQUEST_URI'])){
        $serverrequiri = $_SERVER['PHP_SELF'];
    }
    else
    {
        $serverrequiri = $_SERVER['REQUEST_URI'];
    }
    $s = empty($_SERVER["HTTPS"]) ? "" : ($_SERVER["HTTPS"] == "on") ? "s" : "";
    $protocol = $this->strleft(strtolower($_SERVER["SERVER_PROTOCOL"]), "/").$s;
    $port = ($_SERVER["SERVER_PORT"] == "80") ? "" :
    ":".$_SERVER["SERVER_PORT"];
    return
$protocol."//".$_SERVER['SERVER_NAME'].$port.$serverrequiri;
}

function strleft($s1, $s2)
{
    return substr($s1, 0, strpos($s1, $s2));
}
?
```

MySql.php

```

<?php

if (ereg("Mysql.php", $_SERVER['PHP_SELF'])) {
    Header("Location: ../index.php");
}

if(!defined("SQL_LAYER"))
{
    define("SQL_LAYER", "mysql");

class sql_db_mysql
{
    var $db_connect_id;
    var $query_result;
    var $row = array();
    var $rowset = array();
    var $num_queries = 0;

    //
    // Constructor
    //

    function sql_db_mysql($sqlserver, $sqluser, $sqlpassword,
    $database, $persistency = false)
    {

        $this->persistency = $persistency;
        $this->user = $sqluser;
        $this->password = $sqlpassword;
        $this->server = $sqlserver;
        $this->dbname = $database;
        if($this->persistency)
        {
            $this->db_connect_id =
            @mysql_pconnect($this->server, $this->user, $this->password);
        }
        else
        {
            $this->db_connect_id =
            @mysql_connect($this->server, $this->user, $this->password);
        }
        if($this->db_connect_id)
        {
            if($database != "")
            {
                $this->dbname = $database;
                $dbselect =
                mysql_select_db($this->dbname);
            }
            if(!$dbselect)
            {
                mysql_close($this->db_connect_id);
            }
        }
    }
}

```

```

    >db_connect_id = $dbselect;
        {
            }
            return $this->db_connect_id;
        }
    }
}

// Other base methods
//
function sql_close()
{
    if($this->db_connect_id)
    {
        if($this->query_result)
        {
            @mysql_free_result($this->query_result);
        }
        $result = @mysql_close($this->db_connect_id);
        return $result;
    }
    else
    {
        return false;
    }
}

// Base query method
//
function sql_query($query = "", $transaction = FALSE)
{
    // Remove any pre-existing queries
    unset($this->query_result);
    if($query != "")
    {
        $this->query_result =
        @mysql_query($query,$this->db_connect_id);

        if (! $this->query_result)
        {
            //echo $query;
            //die($query." Query failed");
            echo "<br/>SQL Error Report : <b>".mysql_errno(). ":" . mysql_error() . "</b>\n";
        }
        //echo $query;
        //echo "<br/>SQL Error Report :
<b>".mysql_errno(). ":" . mysql_error() . "</b>\n";
    }
    if($this->query_result)
    {
        unset($this->row[$this->query_result]);
        unset($this->rowset[$this->query_result]);
        >query_result];
        return $this->query_result;
    }
    else
    {
        return ( $transaction ==
END_TRANSACTION ) ? true : false;
    }
}

function sql_fetchrow($query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if($query_id)
    {
        $this->row[$query_id] =
        @mysql_fetch_array($query_id);
        return $this->row[$query_id];
    }
    else
}
}

$this-
{
    return false;
}

}

function sql_createArray()
{
    $result = array();
    while($row = $this->sql_fetchrow() != false)
    {
        array_push($result,$row);
    }
    return $result;
}

function query($query)
{
    $this->sql_query($query);
    //echo $query;
    $array = $this->sql_createArray();
    //print "<pre>";
    //print_r($array);
    //exit;
    return $array;
}

// Other query methods
//
function sql_numrows($query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if($query_id)
    {
        $result =
        @mysql_num_rows($query_id);
        return $result;
    }
    else
    {
        return false;
    }
}

function sql_affectedrows()
{
    if($this->db_connect_id)
    {
        $result = @mysql_affected_rows($this->db_connect_id);
        return $result;
    }
    else
    {
        return false;
    }
}

function sql_numfields($query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if($query_id)
    {
        $result =
        @mysql_num_fields($query_id);
        return $result;
    }
    else
    {
        return false;
    }
}

function sql_fieldname($offset, $query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if($query_id)
    {
        $result =
        @mysql_field_name($query_id, $offset);
        return $result;
    }
}

```

```

        return $result;
    }
    else
    {
        return false;
    }
}
function sql_fieldtype($offset, $query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if($query_id)
    {
        $result = @mysql_field_type($query_id,
$offset);
        return $result;
    }
    else
    {
        return false;
    }
}

function sql_fetchrowset($query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if($query_id)
    {
        unset($this->rowset[$query_id]);
        unset($this->row[$query_id]);
        while($this->rowset[$query_id] =
@mysql_fetch_array($query_id))
        {
            $result[] = $this-
>rowset[$query_id];
            return $result;
        }
    }
    else
    {
        return false;
    }
}

function sql_fetchfield($field, $rnum = -1, $query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if($query_id)
    {
        if($rnum > -1)
        {
            $result =
@mysql_result($query_id, $rnum, $field);
        }
        else
        {
            if(empty($this-
>row[$query_id]) && empty($this->rowset[$query_id]))
            {
                if($this-
>sql_fetchrow())
                {
                    $result
= $this->row[$query_id][$field];
                }
            }
            else
            {
                if($this-
>rowset[$query_id])
                {
                    $result
= $this->rowset[$query_id][$field];
                }
                else if($this-
>row[$query_id])
                {
                    $result
= $this->row[$query_id][$field];
                }
            }
        }
    }
}

function sql_rowseek($rnum, $query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if($query_id)
    {
        $result = @mysql_data_seek($query_id,
$rnum);
        return $result;
    }
    else
    {
        return false;
    }
}

function sql_nextid()
{
    if($this->db_connect_id)
    {
        $result = @mysql_insert_id($this-
>db_connect_id);
        return $result;
    }
    else
    {
        return false;
    }
}

function sql_freeresult($query_id = 0)
{
    if(!$query_id)
    {
        $query_id = $this->query_result;
    }
    if ($query_id )
    {
        unset($this->row[$query_id]);
        unset($this->rowset[$query_id]);
        @mysql_free_result($query_id);
        return true;
    }
    else
    {
        return false;
    }
}

function sql_error($query_id = 0)
{
    $result["message"] = @mysql_error($this-
>db_connect_id);
    $result["code"] = @mysql_errno($this-
>db_connect_id);
    return $result;
}

class sql_db
{
    // class sql_db
    // if ... define
    ?>
tlAbsensi.class.php
<?php
class tlAbsensi {
    var $db;
    function tlAbsensi($db_object)
    {
        $this->db = $db_object;
    }
}

```

```

    $this->piket = new tlPiket($db_object);
}

/**
* @desc getAllAbsensiDataList in TBL_ABSENSI
* @var
* @param $count, $page, $_GET
* @return Array
*/
function getAllAbsensiDataList($count = 0, $page = 0, $_GET)
{
    if ($this->db)
    {
        $namaPegawai = $_GET['namaPegawai'];
        $namaMataPelajaran = $_GET['namaMataPelajaran'];
        $tanggalAbsensi = $_GET['tanggalAbsensi'];

        $result = 0;
        $query = "SELECT * FROM ".TBL_ABSENSI." a LEFT JOIN
(" . TBL_TAHUNPELAJARAN . " b, " . TBL_PEGAWAI . " c,
    " . TBL_MATAPELAJARAN . " d, " . TBL_JURUSAN . "
e, " . TBL_PIKET . " f)
        ON(a.IDTahunPelajaran=b.IDTahunPelajaran AND
a.IDPegawai=c.IDPegawai
        AND a.IDMataPelajaran=d.IDMataPelajaran AND
a.IDJurusan=e.IDJurusan
        AND a.IDPiket=f.IDPiket) WHERE 1=1 AND
e.IDJurusan=".$_SESSION['IDJURUSAN']."' ";
        if (!empty($_GET))
        {
            if(!empty($namaPegawai) && $namaPegawai != "")
            {
                $query .= " AND c.namaPegawai LIKE '%" .
trim($namaPegawai). "%'";
            }
            if(!empty($namaMataPelajaran) && $namaMataPelajaran != "")
            {
                $query .= " AND d.namaMataPelajaran LIKE '%" .
trim($namaMataPelajaran). "%'";
            }
            if(!empty($tanggalAbsensi) && $tanggalAbsensi != "")
            {
                $query .= " AND a.tanggalAbsensi = '" .
trim($tanggalAbsensi). "'";
            }
        }
        $query .= " ORDER BY c.namaPegawai ASC ";

        if (strlen($page) > 0)
        {
            $query .= " LIMIT ".($page * $count). ", ". $count;
        }
        $result = $this->db->query($query);
        // print "<pre>";
        // print $query . "<br/>";
        // print_r($result);
        // exit;
        return $result;
    }
    else
    {
        echo "Error line 43 tlAbsensi.class.php";
        return;
    }
}

/**
* @desc Gets a total of Absensi available in database.
*
* @param active the active flag
* @return array or null
*/
function getAbsensiCount( $_GET )
{
    if ($this->db)
    {
        $namaAbsensi = $_GET['namaAbsensi'];

        $result = 0;
        $query = "SELECT COUNT(a.IDAbsensi) AS total FROM ".
TBL_ABSENSI . " a
        WHERE 1=1";
        if (!empty($_GET))

```

```

        {
            if(!empty($namaAbsensi) && $namaAbsensi != "")
            {
                $query .= " AND a.namaAbsensi LIKE '%" .
trim($namaAbsensi). "%'";
            }
        }
        $result = $this->db->query($query);
        return $result[0];
    }
    else
    {
        echo "Error line 85 tlAbsensi.class.php";
        return;
    }
}

/**
* @desc getAbsensiById in TBL_ABSENSI at specific IDAbsensi
* @param IDAbsensi from $_GET
* @var IDAbsensi
* @return Array
*/
function getAbsensiById( $IDAbsensi )
{
    if ($this->db)
    {
        $query = "SELECT * FROM ".TBL_ABSENSI." a WHERE
a.IDAbsensi='".$IDAbsensi . "'";
        $result = $this->db->query($query);
        // print "<pre>";
        // print $query;
        // exit;
        return $result;
    }
    else
    {
        echo "Error line 115 tlAbsensi.class.php";
        return;
    }
}

/**
* @desc countAbsensiByIDPegawai in TBL_ABSENSI at specific
IDPegawai
* @param IDPegawai
* @var IDPegawai
* @return Array
*/
function countAbsensiByIDPegawai( $d, $m, $y, $IDPegawai )
{
    if ($this->db)
    {
        $query = "SELECT COUNT(a.IDAbsensi) as jumlahJam FROM ".
TBL_ABSENSI . " a WHERE 1=1
        AND a.IDPegawai='".$IDPegawai . "'";
        if(!empty($d) && $d != "")
        {
            $query .= " AND DAY(a.tanggalAbsensi)='".$d . "'";
        }
        if(!empty($m) && $m != "")
        {
            $query .= " AND MONTH(a.tanggalAbsensi)='".$m . "'";
        }
        if(!empty($y) && $y != "")
        {
            $query .= " AND YEAR(a.tanggalAbsensi)='".$y . "'";
        }
        $result = $this->db->query($query);
        // print "<pre>";
        // print $query;
        // print_r($result);
        // exit;
        return $result;
    }
    else
    {
        echo "Error line 115 tlAbsensi.class.php";
        return;
    }
}

/**
* @desc Save an instance of $_POST to database.
* @param $_POST

```

```

/*
function save( $_POST )
{
    if ($this->db)
    {
        print "<pre>";
        print_r($_POST);
        exit;
    }
    $this->piket->save($_POST);
    $pegawaiPiket = $this->piket-
>getPiketByParameter($_POST['IDJurusan'],$_POST['IDTahunPelajaran'],
$_POST['tanggalPiket']);

    for($ii=0; $ii<count($_POST['IDPegawai']); $ii++)
    {
        $this->remove($_POST['IDJurusan'],
$_POST['IDTahunPelajaran'], $_POST['tanggalAbsensi'],
$_POST['IDPegawai'][$ii]);
        for($jj=0; $jj<8; $jj++)
        {
            if($_POST['sesi'][$ii][$jj] == "") { continue; }
            if ( $this->isExist($_POST['IDJurusan'],
$_POST['IDTahunPelajaran'], $_POST['tanggalAbsensi'],
$_POST['IDPegawai'][$ii], $_POST['sesi'][$ii][$jj]) )
            {
                continue;
            }
            else
            {
                $query = "INSERT INTO " . TBL_ABSENSI . " ";
                $query .= "(IDAbsensi, IDJurusan, IDTahunPelajaran,
tanggalAbsensi, IDPegawai, sesi, kelas, IDPiket, tanggalInput,
terakhirUpdate, terakhirUpdateOleh";
                $query .= ", VALUES (";
                $query .= "", ",";
                $query .= "" . $this->escape( $_POST['IDJurusan'] ) . ", ";
                $query .= "" . $this->escape( $_POST['IDTahunPelajaran']
) . ", ";
                $query .= "" . $this->escape( $_POST['tanggalAbsensi'] );
                $query .= ", ";
                $query .= "" . $this->escape( $_POST['IDPegawai'][$ii] );
                $query .= ", ";
                $query .= "" . $this->escape( $_POST['sesi'][$ii][$jj] );
                $query .= ", ";
                $query .= "" . $this->escape( $pegawaiPiket[0]['IDPegawaiPiket'] );
                $query .= ", ";
                $query .= "NOW(), ";
                $query .= "NOW(), ";
                $query .= "" . $this->escape(
$_POST['terakhirUpdateOleh'] );
                $query .= ", ";
                $query .= ");";
                print "<pre>";
                print $query."<br/>";
                $result = $this->db->query($query);
            }
        }
    }
    // exit;
    return $result;
}

/**
 * @desc Check whether a Absensi with specific parameter exist in
database
 *
 * @param $stanggalAbsensi, $IDPegawai, $sesi
 * @return true if exist false otherwise
 */
function isExist( $IDJurusan, $IDTahunPelajaran, $stanggalAbsensi,
$IDPegawai, $sesi )
{
    $result = 0;
    $query = "SELECT COUNT(a.IDAbsensi) AS total FROM " .
TBL_ABSENSI . " a
WHERE 1=1";

    if(!empty($IDJurusan) && $IDJurusan != "")
    {
        $query .= " AND a.IDJurusan = " . trim($IDJurusan) . "";
    }
    if(!empty($IDTahunPelajaran) && $IDTahunPelajaran != "")
    {
        $query .= " AND a.IDTahunPelajaran = " .
trim($IDTahunPelajaran) . "";
    }
    if(!empty($stanggalAbsensi) && $stanggalAbsensi != "")
    {
        $query .= " AND a.tanggalAbsensi = " .
trim($stanggalAbsensi) . "";
    }
    if(!empty($IDPegawai) && $IDPegawai != "")
    {
        $query .= " AND a.IDPegawai = " . trim($IDPegawai) . "";
    }
    if(!empty($sesi) && $sesi != "")
    {
        $query .= " AND a.sesi = " . trim($sesi) . "";
    }
    if ($this->db)
    {
        $result = $this->db->query($query);
        return $result;
    }
    else
    {
        echo "Error line 223 tlAbsensi.class.php";
        return false;
    }
}

/**
 * @desc Check whether a Absensi with specific parameter exist in
database
 *
 * @param $stanggalAbsensi, $IDPegawai, $sesi
 * @return true if exist false otherwise
 */
function getKelas( $IDJurusan, $IDTahunPelajaran, $stanggalAbsensi,
$IDPegawai, $sesi )
{
    $result = 0;
    $query = "SELECT a.kelas FROM " . TBL_ABSENSI . " a
WHERE 1=1";

    if(!empty($IDJurusan) && $IDJurusan != "")
    {
        $query .= " AND a.IDJurusan = " . trim($IDJurusan) . "";
    }
    if(!empty($IDTahunPelajaran) && $IDTahunPelajaran != "")
    {
        $query .= " AND a.IDTahunPelajaran = " .
trim($IDTahunPelajaran) . "";
    }
}
*/
* Remove Absensi from database.
*
* @param
* @return the number of object removed from database
*/

```

```

function remove( $IDJurusan, $IDTahunPelajaran, $tanggalAbsensi,
$IDPegawai )
{
    $result = 0;
    if ((strlen($IDPegawai) > 0) && $this->db)
    {
        $query = "DELETE FROM ". TBL_ABSENSI . " WHERE
IDJurusan="" . $IDJurusan . " AND IDTahunPelajaran="" .
$IDTahunPelajaran . " AND
        tanggalAbsensi="" . $tanggalAbsensi . " AND
IDPegawai="" . $IDPegawai . """;
        //        print "<pre>";
        //        print $query . "<br/>";
        //        exit;
        $result = $this->db->query($query);
    }
    return $result;
}

/**
 * Escape a form data before adding to database.
 *
 * @param value the String to be escaped
 * @return safe string
 */
function escape( $value )
{
    if (get_magic_quotes_gpc())
    {
        return $value;
    }
    else
    {
        return addslashes($value);
    }
}
?>

tlHonor.class.php

<?php

class tlHonor {
    var $db;

    function tlHonor($db_object)
    {
        $this->db = $db_object;
    }

    /**
     * @desc getAllHonor in TBL_HONOR
     * @var
     * @param
     * @return Array
     */
    function getAllHonor()
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ". TBL_HONOR . " a WHERE
1=1";
            $result = $this->db->query($query);
            //        print "<pre>";
            //        print $query . "<br/>";
            //        print_r($result);
            //        exit;
            return $result;
        }
        else
        {
            echo "Error line 40 tlHonor.class.php";
            return;
        }
    }

    /**
     * @desc getAllHonorDataList in TBL_HONOR
     * @var
     * @param $count, $page, $_GET
     * @return Array
     */
    function getAllHonorDataList($count = 0, $page = 0, $_GET)
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ". TBL_HONOR . " a WHERE
1=1";
            $result = $this->db->query($query);
            //        print "<pre>";
            //        print $query . "<br/>";
            //        print_r($result);
            //        exit;
            return $result;
        }
        else
        {
            echo "Error line 40 tlHonor.class.php";
            return;
        }
    }

    /**
     * @desc getHonorCount in TBL_HONOR
     * @param
     * @return array or null
     */
    function getHonorCount( $_GET )
    {
        if ($this->db)
        {
            $result = 0;
            $query = "SELECT COUNT(a.IDHonor) AS total FROM ".
TBL_HONOR . " a
                WHERE 1=1";
            $result = $this->db->query($query);
            return $result[0];
        }
        else
        {
            echo "Error line 97 tlHonor.class.php";
            return;
        }
    }

    /**
     * @desc getHonorById in TBL_HONOR at specific IDHonor
     * @param IDHonor from $_GET
     * @var IDHonor
     * @return Array
     */
    function getHonorById( $IDHonor )
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ". TBL_HONOR . " a WHERE
a.IDHonor="" . $IDHonor . "";
            $result = $this->db->query($query);
            //        print "<pre>";
            //        print $query;
            //        exit;
            return $result;
        }
        else
        {
            echo "Error line 119 tlHonor.class.php";
            return;
        }
    }

    /**
     * @desc Save an instance of $_POST to database.
     * @param $_POST
     */
    function save( $_POST )
    {
        if ($this->db)
        {
            if ( $this->isExist($_POST['IDHonor']) )
            {
                //        print "<pre>";

```

```

//      print_r($_POST);
//      exit;

//UPDATE TABLE PERSON DATA
$query = "UPDATE ". TBL_HONOR . " SET ";

$query .= "honorPerJam="" . $this->escape(
$_POST['honorPerJam']) . "", ";
$query .= "transportPerJam="" . $this->escape(
$_POST['transportPerJam']) . "", ";
$query .= "piketPerHari="" . $this->escape(
$_POST['piketPerHari']) . "" ";

$query .= "WHERE IDHonor="" . $_POST['IDHonor'] . """;
}

else
{
//      print "<pre>";
//      print_r($_POST);
//      exit;
/*
 * INSERT INTO TBL_HONOR
 */
$query = "INSERT INTO ". TBL_HONOR . " ";
$query .= "(IDHonor, honorPerJam, transportPerJam,
piketPerHari)";
$query .= "VALUES (";
$query .= "", ",";
$query .= "" . $this->escape( $_POST['honorPerJam']) . "", ",";
$query .= "" . $this->escape( $_POST['transportPerJam']) . "", ",";
$query .= "" . $this->escape( $_POST['piketPerHari']) . "" ;
$query .= ")";
}

print "<pre>";
print $query."<br/>";
exit;
$result = $this->db->query($query);
return $result;
}

/**
 * @desc Check whether a Honor with specific IDHonor exist in
database
*/
* @param IDHonor
* @return true if exist false otherwise
*/
function isExist( $IDHonor = "" )
{
if (strlen($IDHonor) <= 0)
{
    return false;
}

$result = 0;
$query = "SELECT COUNT(a.IDHonor) AS total FROM ".
TBL_HONOR . " a
WHERE a.IDHonor="" . $IDHonor . """;

if ($this->db)
{
    $result = $this->db->query($query);
}
if ($result)
{
    $count = 0;
    if (is_array($result))
    {
        $count = $result[0][0];
    }
    return $count > 0;
}
return false;
}

/**
 * Remove a single Honor from database.
*/
* @param IDHonor to remove
* @return the number of object removed from database
*/
function remove($IDHonor = "")
{
$result = 0;
if ((strlen($IDHonor) > 0) && $this->db)
{
    $query = "DELETE FROM ". TBL_HONOR . " WHERE
IDHonor = "" . $IDHonor . """;
    $result = $this->db->query($query);
}
return $result;
}

/*
 * Escape a form data before adding to database.
*
* @param value the String to be escaped
* @return safe string
*/
function escape( $value )
{
if (get_magic_quotes_gpc())
{
    return $value;
}
else
{
    return addslashes($value);
}
}

?>

tlJabatan.class.php

<?php

class tlJabatan {
    var $db;

    function tlJabatan($db_object)
    {
        $this->db = $db_object;
    }

    /**
     * @desc getAllJabatan in TBL_JABATAN
     * @var
     * @param
     * @return Array
     */
    function getAllJabatan()
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ". TBL_JABATAN . " a WHERE
1=1";
            $result = $this->db->query($query);
            //      print "<pre>";
            //      print $query . "<br/>";
            //      print_r($result);
            //      exit;
            return $result;
        }
        else
        {
            echo "Error line 40 tlJabatan.class.php";
            return;
        }
    }

    /**
     * @desc getAllJabatanDataList in TBL_JABATAN
     * @var
     * @param $count, $page, $_GET
     * @return Array
     */
    function getAllJabatanDataList($count = 0, $page = 0, $_GET)
    {
        if ($this->db)
        {
            $namaJabatan = $_GET['namaJabatan'];

            $result = 0;
            $query = "SELECT * FROM ". TBL_JABATAN . " a WHERE
1=1";
            if (!empty($_GET))
            {
                $query .= " AND a.namaJabatan = '$namaJabatan'";
            }
            $result = $this->db->query($query);
        }
        return $result;
    }
}

```

```

        if(!empty($namaJabatan) && $namaJabatan != "")
        {
            $query .= " AND a.namaJabatan LIKE '%" .
trim($namaJabatan) . "%'";
        }
        $query .= " ORDER BY a.IDJabatan ASC ";

        if (strlen($page) > 0)
        {
            $query .= " LIMIT " . ($page * $count) . ", " . $count;
        }
        $result = $this->db->query($query);
        print "<pre>";
        print $query . "<br/>";
        print_r($result);
        exit;
        return $result;
    }
    else
    {
        echo "Error line 43 tJabatan.class.php";
        return;
    }
}

/**
 * @desc Gets a total of Jabatan available in database.
 */
function getJabatanCount( $_GET )
{
    if ($this->db)
    {
        $namaJabatan = $_GET['namaJabatan'];

        $result = 0;
        $query = "SELECT COUNT(a.IDJabatan) AS total FROM " .
TBL_JABATAN . " a
                WHERE 1=1";

        if (!empty($_GET))
        {
            if(!empty($namaJabatan) && $namaJabatan != "")
            {
                $query .= " AND a.namaJabatan LIKE '%" .
trim($namaJabatan) . "%'";
            }
            $result = $this->db->query($query);
            return $result[0];
        }
        else
        {
            echo "Error line 85 tJabatan.class.php";
            return;
        }
    }

    /**
     * @desc getJabatanById in TBL_JABATAN at specific IDJabatan
     * @param IDJabatan from $_GET
     * @var IDJabatan
     * @return Array
     */
    function getJabatanById( $IDJabatan )
    {
        if ($this->db)
        {
            $query = "SELECT * FROM " . TBL_JABATAN . " a WHERE
a.IDJabatan=" . $IDJabatan . "";
            $result = $this->db->query($query);
            // print "<pre>";
            // print $query;
            // exit;
            return $result;
        }
        else
        {
            echo "Error line 115 tJabatan.class.php";
            return;
        }
    }
}

/**
 * @desc Save an instance of $_POST to database.
 * @param $_POST
 */
function save( $_POST )
{
    if ($this->db)
    {
        if ( $this->isExist($_POST['IDJabatan']) )
        {
            // print "<pre>";
            // print_r($_POST);
            // exit;

            //UPDATE TABLE PERSON DATA
            $query = "UPDATE " . TBL_JABATAN . " SET ";

            $query .= "namaJabatan=" . $this->escape(
$_POST['namaJabatan']) . ", ";
            $query .= "jumlahTunjangan=" . $this->escape(
$_POST['jumlahTunjangan']) . ", ";
            $query .= "keteranganJabatan=" . $this->escape(
$_POST['keteranganJabatan']) . " ";

            $query .= "WHERE IDJabatan=" . $_POST['IDJabatan'] . "";
        }
        else
        {
            // print "<pre>";
            // print_r($_POST);
            // exit;
            /*
             * INSERT INTO TBL_JABATAN
             */
            $query = "INSERT INTO " . TBL_JABATAN . " ";
            $query .= "(IDJabatan, namaJabatan, jumlahTunjangan,
keteranganJabatan)";
            $query .= "VALUES (";
            $query .= "" . $this->escape( $_POST['namaJabatan']) . ", ";
            $query .= "" . $this->escape( $_POST['jumlahTunjangan']) . ", ";
            $query .= "" . $this->escape( $_POST['keteranganJabatan']) . " ";
            $query .= ") ";
            // print "<pre>";
            // print $query."<br/>";
            // exit;
            $result = $this->db->query($query);
            return $result;
        }
    }

    /**
     * @desc Check whether a Jabatan with specific IDJabatan exist in
database
     */
    function isExist( $IDJabatan = "" )
    {
        if (strlen($IDJabatan) <= 0)
        {
            return false;
        }

        $result = 0;
        $query = "SELECT COUNT(a.IDJabatan) AS total FROM " .
TBL_JABATAN . " a
                WHERE a.IDJabatan=" . $IDJabatan . "";

        if ($this->db)
        {
            $result = $this->db->query($query);
        }
        if ($result)
        {
            $count = 0;
            if (is_array($result))
            {
                $count = $result[0][0];
            }
        }
    }
}

```

```

        return $count > 0;
    }
    return false;
}

/**
 * Remove a single Jabatan from database.
 *
 * @param IDJabatan to remove
 * @return the number of object removed from database
 */
function remove($IDJabatan = "")
{
    $result = 0;
    if ((strlen($IDJabatan) > 0) && $this->db)
    {
        $query = "DELETE FROM ". TBL_JABATAN . " WHERE
IDJabatan = " . $IDJabatan . "";
        $result = $this->db->query($query);
    }
    return $result;
}

/**
 * Escape a form data before adding to database.
 *
 * @param value the String to be escaped
 * @return safe string
 */
function escape( $value )
{
    if (get_magic_quotes_gpc())
    {
        return $value;
    }
    else
    {
        return addslashes($value);
    }
}
?>

tlJurusan.class.php

<?php

class tlJurusan {
    var $db;

    function tlJurusan($db_object)
    {
        $this->db = $db_object;
    }

    /**
     * @desc getAllJurusan
     * @param
     * @var
     * @return Array
     */
    function getAllJurusan( )
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ". TBL_JURUSAN . " a WHERE
1=1 ORDER BY a.namaJurusan";
            $result = $this->db->query($query);
//            print "<pre>";
//            print $query;
//            exit;
            return $result;
        }
        else
        {
            echo "Error line 40 tlJurusan.class.php";
            return;
        }
    }

    /**
     * @desc getAllJurusanDataList in TBL_JURUSAN
     * @var
     * @param $count, $page, $_GET
     */
    function getAllJurusanDataList($count = 0, $page = 0, $_GET)
    {
        if ($this->db)
        {
            $namaJurusan = $_GET['namaJurusan'];

            $result = 0;
            $query = "SELECT * FROM ". TBL_JURUSAN . " a WHERE
1=1";

            if (!empty($_GET))
            {
                if(!empty($namaJurusan) && $namaJurusan != "")
                {
                    $query .= " AND a.namaJurusan LIKE '%" .
trim($namaJurusan) . "%'";
                }
            }
            $query .= " ORDER BY a.namaJurusan ASC ";

            if (strlen($page) > 0)
            {
                $query .= " LIMIT " . ($page * $count) . ", " . $count;
            }
            $result = $this->db->query($query);
            print "<pre>";
            print $query . "<br/>";
            print_r($result);
            exit;
            return $result;
        }
        else
        {
            echo "Error line 43 tlJurusan.class.php";
            return;
        }
    }

    /**
     * @desc Gets a total of Jurusan available in database.
     *
     * @param active the active flag
     * @return array or null
     */
    function getJurusanCount( $_GET )
    {
        if ($this->db)
        {
            $namaJurusan = $_GET['namaJurusan'];

            $result = 0;
            $query = "SELECT COUNT(a.IDJurusan) AS total FROM ".
TBL_JURUSAN . " a
WHERE 1=1";

            if (!empty($_GET))
            {
                if(!empty($namaJurusan) && $namaJurusan != "")
                {
                    $query .= " AND a.namaJurusan LIKE '%" .
trim($namaJurusan) . "%'";
                }
            }
            $result = $this->db->query($query);
            return $result[0];
        }
        else
        {
            echo "Error line 85 tlJurusan.class.php";
            return;
        }
    }

    /**
     * @desc getJurusanById in TBL_JURUSAN at specific IDJurusan
     * @param IDJurusan from $_GET
     * @var IDJurusan
     * @return Array
     */
    function getJurusanById( $IDJurusan )
    {
        if ($this->db)
        {
    
```

```

    $query = "SELECT * FROM ". TBL_JURUSAN . " a WHERE
a.IDJurusan="" . $IDJurusan . "";
    $result = $this->db->query($query);
//    print "<pre>";
//    print $query;
//    exit;
    return $result;
}
else
{
    echo "Error line 115 tlJurusan.class.php";
    return;
}
}

/**
* @desc Save an instance of $_POST to database.
* @param $_POST
*/
function save( $_POST )
{
    if ($this->db)
    {
        if ( $this->isExist($_POST['IDJurusan']) )
        {
//            print "<pre>";
//            print_r($_POST);
//            exit;

            //UPDATE TABLE PERSON DATA
            $query = "UPDATE ". TBL_JURUSAN . " SET ";

            $query .= "namaJurusan=" . $this->escape(
$_POST['namaJurusan'] ) . ", ";
            $query .= "keteranganJurusan=" . $this->escape(
$_POST['keteranganJurusan'] ) . " ";

            $query .= "WHERE IDJurusan=" . $_POST['IDJurusan'] . "";
        }
        else
        {
//            print "<pre>";
//            print_r($_POST);
//            exit;
/*
            * INSERT INTO TBL_JURUSAN
*/
            $query = "INSERT INTO ". TBL_JURUSAN . " ";
            $query .= "(IDJurusan, namaJurusan, keteranganJurusan) ";
            $query .= "VALUES ";
            $query .= "", ",";
            $query .= "" . $this->escape( $_POST['namaJurusan'] ) . ", ";
            $query .= "" . $this->escape( $_POST['keteranganJurusan'] ) . " ";
            $query .= ") ";
        }
    }
    print "<pre>";
    print $query."<br/>";
//    exit;
    $result = $this->db->query($query);
    return $result;
}

/**
* @desc Check whether a Jurusan with specific IDJurusan exist in
database
*/
* @param IDJurusan
* @return true if exist false otherwise
*/
function isExist( $IDJurusan = )
{
    if (strlen($IDJurusan) <= 0)
    {
        return false;
    }

    $result = 0;
    $query = "SELECT COUNT(a.IDJurusan) AS total FROM ".
TBL_JURUSAN . " a
WHERE a.IDJurusan="" . $IDJurusan . """;
    if ($this->db)
    {
}
}

$result = $this->db->query($query);
if ($result)
{
    $count = 0;
    if (is_array($result))
    {
        $count = $result[0][0];
    }
    return $count > 0;
}
return false;
}

/**
* Remove a single Jurusan from database.
*
* @param IDJurusan to remove
* @return the number of object removed from database
*/
function remove($IDJurusan = "")
{
    $result = 0;
    if ((strlen($IDJurusan) > 0) && $this->db)
    {
        $query = "DELETE FROM ". TBL_JURUSAN . " WHERE
IDJurusan = "" . $IDJurusan . "";
        $result = $this->db->query($query);
    }
    return $result;
}

/**
* Escape a form data before adding to database.
*
* @param value the String to be escaped
* @return safe string
*/
function escape( $value )
{
    if (get_magic_quotes_gpc())
    {
        return $value;
    }
    else
    {
        return addslashes($value);
    }
}
?>

tlPegawai.class.php

<?php

class tlPegawai {
    var $db;
    //var $config;

    function tlPegawai($db_object)
    {
        $this->db = $db_object;

        $this->config = new Config($db_object);
        $this->mail = new tlMail($db_object);
        $this->tahunPelajaran = new tlTahunPelajaran($db_object);
    }

    /**
     * @desc getAllPegawaiDataList in TBL_ADMIN, TBL_ADMIN and
TBL_ROLE
     * @var
     * @param $count, $page, $_GET, $order, $active
     * @return Array
     */
    function getAllPegawaiDataList($count = 0, $page = 0, $_GET, $active
= -1)
    {
        if ($this->db)
        {
            $IDJurusan = $_GET['IDJurusan'];
            $IDPegawai = $_GET['IDPegawai'];
            $namaPegawai = $_GET['namaPegawai'];
        }
    }
}

```

```

$result = 0;
$query = "SELECT * FROM ". TBL_PEGAWAI . " a LEFT
JOIN(" . TBL_JABATAN . " b, " . TBL_USER . " c)
    ON(a.IDJabatan=b.IDJabatan AND a.IDUser=c.IDUser)
    WHERE 1=1";

if ((0 + $active) > -1)
{
    $query .= " AND a.statusPegawai = " . ( 0 + $active) . "";
}

if (!empty($_GET))
{
    if(!empty($IDJurusan) && $IDJurusan != "")
    {
        $query .= " AND a.IDJurusan = " . trim($IDJurusan) . "";
    }
    if(!empty($IDPegawai) && $IDPegawai != "")
    {
        $query .= " AND a.IDPegawai = " . trim($IDPegawai) . "";
    }
    if(!empty($namaPegawai) && $namaPegawai != "")
    {
        $query .= " AND a.namaPegawai LIKE
        '%" . trim($namaPegawai) . "%'";
    }
}
$query .= " ORDER BY a.namaPegawai ASC ";

if (strlen($page) > 0)
{
    $query .= " LIMIT " . ($page * $count) . ", " . $count;
}

$result = $this->db->query($query);
// print "<pre>";
// print $query . "<br/>";
// print_r($result);
// exit;
return $result;
}
else
{
    echo "Error line 48 tlPegawai.class.php";
    return;
}

/***
* @desc Get a total of Pegawai available in database.
*
* @param active the active flag
* @return array or null
*/
function getPegawaiCount( $_GET, $active = -1 )
{
    if ($this->db)
    {
        $IDPegawai = $_GET['IDPegawai'];
        $namaPegawai = $_GET['namaPegawai'];

        $result = 0;
        $query = "SELECT COUNT(a.IDPegawai) AS total FROM ". TBL_PEGAWAI . " a LEFT JOIN(" . TBL_JABATAN . " b, " . TBL_USER . " c)
            ON(a.IDJabatan=b.IDJabatan AND a.IDUser=c.IDUser)
            WHERE 1=1";

        if ((0 + $active) > -1)
        {
            $query .= " AND a.statusPegawai=" . ( 0 + $active) . "";
        }

        if (!empty($_GET))
        {
            if(!empty($IDPegawai) && $IDPegawai != "")
            {
                $query .= " AND a.IDPegawai LIKE '%". trim($IDPegawai) . "%'";
            }
            if(!empty($namaPegawai) && $namaPegawai != "")
            {
                $query .= " AND a.namaPegawai LIKE
                '%" . trim($namaPegawai) . "%'";
            }
        }
    }
}

    }
}
else
{
    echo "Error line 100 tlPegawai.class.php";
    return;
}

/**
* @desc Gets total of Pegawai Active available in database.
*
* @param
* @return array or null
*/
function getActivePegawaiCount()
{
    return $this->getPegawaiCount( $_GET, 1 );
}

/**
* @desc Get total of Pegawai Inactive available in database.
*
* @param
* @return array or null
*/
function getInactivePegawaiCount()
{
    return $this->getPegawaiCount( $_GET, 0 );
}

/**
* @desc Gets All Pegawai Active available in database.
*
* @param
* @return array or null
*/
function getActivePegawaiList($count = 0, $page = 0, $_GET)
{
    return $this->getAllPegawaiDataList($count, $page, $_GET, 1);
}

/**
* @desc Get All Pegawai Inactive available in database.
*
* @param
* @return array or null
*/
function getInactivePegawaiList($count = 0, $page = 0, $_GET)
{
    return $this->getAllPegawaiDataList($count, $page, $_GET, 0);
}

/**
* @desc getPegawaiById in TBL_PEGAWAI at specific IDPegawai
* @param IDPegawai from $_GET
* @var IDPegawai
* @return Array
*/
function getPegawaiById( $IDPegawai )
{
    if ($this->db)
    {
        $query = "SELECT * FROM ". TBL_PEGAWAI . " a LEFT
JOIN(" . TBL_JABATAN . " b, " . TBL_USER . " c)
    ON(a.IDJabatan=b.IDJabatan AND a.IDUser=c.IDUser)
    WHERE a.IDPegawai=" . $IDPegawai . "";

        $result = $this->db->query($query);
//        print "<pre>";
//        print $query;
//        print_r($result);
//        exit;
        return $result;
    }
}
else
{
    echo "Error line 1852 tlPegawai.class.php";
    return;
}

/**
* @desc getPegawaiByEmail in TBL_PEGAWAI at specific email
* @param email from $_POST
*/

```

```

* @var email
* @return Array
*/
function getPegawaiByEmail( $email )
{
    if ($this->db) {
        $query = "SELECT * FROM ". TBL_PEGAWAI . " a LEFT
JOIN( ". TBL_JABATAN . " b, ". TBL_USER . " c)
        ON(a.IDJabatan=b.IDJabatan AND a.IDUser=c.IDUser)
        WHERE a.email LIKE '" . $email . "'";
        $result = $this->db->query($query);
        return $result;
    }
    else
    {
        echo "Error line 210 tlPegawai.class.php";
        return;
    }
}

/**
* @desc getPegawaiByLevel in TBL_PEGAWAI at specific level
* @param level from $_POST
* @var level
* @return Array
*/
function getPegawaiByLevel( $IDLevel )
{
    if ($this->db) {
        $query = "SELECT * FROM ". TBL_PEGAWAI . " a LEFT
JOIN( ". TBL_JABATAN . " b, ". TBL_USER . " c)
        ON(a.IDJabatan=b.IDJabatan AND a.IDUser=c.IDUser)
        WHERE a.IDUser = " . $IDLevel . "";
        $result = $this->db->query($query);
        // print "<pre>";
        // print $query;
        // print_r($result);
        // exit;
        return $result;
    }
    else
    {
        echo "Error line 234 tlPegawai.class.php";
        return;
    }
}

/**
* @desc getPengurus in TBL_PEGAWAI
* @param
* @var
* @return Array
*/
function getPengurus()
{
    if ($this->db) {
        $query = "SELECT * FROM ". TBL_PEGAWAI . " a LEFT
JOIN( ". TBL_JABATAN . " b, ". TBL_USER . " c)
        ON(a.IDJabatan=b.IDJabatan AND a.IDUser=c.IDUser)
        WHERE a.IDJabatan<=5";
        $result = $this->db->query($query);
        // print "<pre>";
        // print $query;
        // print_r($result);
        // exit;
        return $result;
    }
    else
    {
        echo "Error line 265 tlPegawai.class.php";
        return;
    }
}

/**
* @desc Pegawai authentication, logged in TBL_PEGAWAI
* @var PegawaiName, password from $_POST
* @return Session Array
*/
function authenticate($username, $password, $IDUser)
{
    // print "<pre>";
    // echo "username: ".$username."<br/>";
    // echo "Password: ".$password."<br/><br/>";
    // exit;
}

$pass = md5($password);
$query = "SELECT * FROM ". TBL_PEGAWAI . " a LEFT JOIN(
        TBL_JABATAN . " b, ". TBL_USER . " c)
        ON(a.IDJabatan=b.IDJabatan AND a.IDUser=c.IDUser)
        WHERE a.username='" . $username . "' AND a.password=''";
$pass . "
        AND a.IDUser=" . $IDUser . " ";
$result = $this->db->query($query);
// print "<pre>";
// print $query . "<br/>";
// print count($result) . "<br/>";
// print_r($result);
// exit;

$tahunPelajaran = $this->tahunPelajaran-
>getTahunPelajaranActive();

if ( ! $result || $result[0]['statusPegawai'] == 0 )
{
    return false;
}
else
{
    if ($result[0]['username']) {

        session_start();
        session_register( 'IDPEGAWAI' );
        session_register( 'NAMAPEGAWAI' );
        session_register( 'LEVEL' );
        session_register( 'JABATAN' );
        session_register( 'IDTAHUNPELAJARAN' );
        session_register( 'TAHUNPELAJARAN' );
        session_register( 'STATUS' );

        $_SESSION['IDPEGAWAI'] = $result[0]['IDPegawai'];
        $_SESSION['NAMAPEGAWAI'] = $result[0]['namaPegawai'];
        $_SESSION['LEVEL'] = $result[0]['namaUser'];
        $_SESSION['JABATAN'] = $result[0]['namaJabatan'];
        $_SESSION['IDTAHUNPELAJARAN'] =
$tahunPelajaran[0]['IDTahunPelajaran'];
        $_SESSION['TAHUNPELAJARAN'] =
$tahunPelajaran[0]['namaTahunPelajaran'];
        $_SESSION['STATUS'] = $result[0]['statusPegawai'];
        session_write_close();

        $lastIP = $_SERVER['REMOTE_ADDR'];
        $lastLogin = date("Y-m-d H:i:s",time());
        $qrUpdate = "UPDATE ". TBL_PEGAWAI . " a SET a.IP=" .
$lastIP . ", a.loginTerakhir=" . $lastLogin . "
        WHERE a.IDPegawai=" . $_SESSION['IDPEGAWAI'] .
"";

        $resultUpdate = $this->db->query($qrUpdate);

        // print "<pre>";
        // print_r($_SESSION);
        // exit;

        if ( ! $resultUpdate )
        {
            //echo 'Error line 237 tlPegawai.class.php on
'. $qrUpdate . '<br/>';
            return true;
        }
        else
        {
            // $this->db = $_SESSION;
            return true;
        }
        else
        {
            return true;
        }
    }
}

/**
* @desc Pegawai logout, nothing to update just lastLogout
* @param none
* @return bool
*/
function logout()

```

```

{
    //check the session
    if(session_is_registered("IDPEGAWAI"))
    {
        //Close MySQL
        $this->db->sql_close();
        //just destroy
        session_destroy();
        return true;
    }
    else
    {
        //throw error, if no session was registered
        $msg = "NO SESSION WAS REGISTERD";
        session_start();
        session_destroy();
        $url = $this->config->get_url();

        header("location:".ADMIN_URL."login/index.php?action=flogin&referer=".urlencode($url));
        //print $msg;
        return false;
    }
}

/**
 * @desc kick out if reach this file without SESSION
 * @param none
 * @return bool
 */
function checkSession()
{
    if (!isset($_SESSION['NAMAPEGAWAI']) AND
        ($_SESSION['NAMAPEGAWAI'] == null))
    {
        //if sesion registered IDPegawai not found, destroy all session and
        return false;
        session_destroy();
        return false;
    }
    return true;
}

/**
 * @desc Save an instance of $_POST to database.
 * @param $_POST
 */
function save( $_POST )
{
    if ($this->db)
    {
        if ( $this->isExist($_POST['IDPegawai']) )
        {
            //      print "<pre>";
            //      print_r($_POST);
            //      exit;
            $pegawai = $this->getPegawaiById($_POST['IDPegawai']);

            $query = "UPDATE ". TBL_PEGAWAI . " SET ";

            $query .= "namaPegawai="" . $this->escape(
$_POST['namaPegawai']) . "", ";
            $query .= "jenisKelamin="" . $this->escape(
$_POST['jenisKelamin']) . "", ";
            $query .= "tempatLahir="" . $this->escape(
$_POST['tempatLahir']) . "", ";
            $query .= "tanggalLahir="" . $this->escape(
$_POST['tanggalLahir']) . "", ";
            $query .= "alamatPegawai="" . $this->escape(
$_POST['alamatPegawai']) . "", ";
            $query .= "teleponPegawai="" . $this->escape(
$_POST['teleponPegawai']) . "", ";
            $query .= "pendidikanTerakhir="" . $this->escape(
$_POST['pendidikanTerakhir']) . "", ";
            $query .= "email="" . $this->escape( $_POST['email']) . "", ";
            $query .= "mulaiTugas="" . $this->escape(
$_POST['mulaiTugas']) . "", ";
            $query .= "IDJabatan="" . $this->escape( $_POST['IDJabatan']) . "", ";
            $query .= "IDUser="" . $this->escape( $_POST['IDUser']) . "", ";
            $query .= "IDJurusan="" . $this->escape( $_POST['IDJurusan']) . "", ";
            $query .= "statusPegawai="" . $this->escape(
$_POST['statusPegawai']) . "", ";
        }
    }
}

/**
 * @desc Change Pegawai password
 * @param $_POST
 */
function changePassword( $_POST )
{
    if ($this->db)
    {
        if ( $this->isExist($_POST['IDPegawai']) )
        {
            $admin = $this->getPegawaiById( $_POST['IDPegawai'] );
            if($admin[0]['password'] == md5($_POST['passwordLama'])) AND
            $_POST['passwordBaru'] == $_POST['passwordBaruConfirm']
            {
                //UPDATE TABLE SYSTEM admin DATA
                $query = "UPDATE ". TBL_PEGAWAI . " SET password =
" . $this->escape( md5($_POST['passwordBaru']) ) . "
WHERE IDPegawai="" . $_POST['IDPegawai']. """;
            }
        }
    }
}

/**
 * @desc Reset Pegawai password
 */
function sendResetMail()
{
    $query .= "WHERE IDPegawai="" . $_POST['IDPegawai']. """;
    if($admin[0]['password'] == md5($_POST['passwordLama'])) AND
    $_POST['passwordBaru'] == $_POST['passwordBaruConfirm']
    {
        //print "<pre>";
        //print $query . "<br/>";
        //print_r($admin) . "<br/>";
        //exit;
        return true;
    }
}

```

```

/*
* @param $_POST
*/
function forgotPassword( $email )
{
    if ($this->db)
    {
        $admin = $this->getPegawaiByEmail( $email );
        if(is_array($admin) AND $admin[0]['statusPegawai'] == 1)
        {
            $confirmation = md5($admin[0]['email'] . $admin[0]['username']);
            $sendMail = $this->mail->sendPasswordChangeConfirmation($admin[0]['email'],
                $admin[0]['username'], $admin[0]['IDPegawai'], $confirmation);
            return true;
        }
        else
        {
            return false;
        }
    }
}

/**
* @desc Reset Pegawai password
*
* @param $_POST
*/
function resetPassword( $email )
{
    if ($this->db)
    {
        $admin = $this->getPegawaiByEmail( $email );
        if(is_array($admin) AND $admin[0]['statusPegawai'] == 1)
        {
            $passwordBaru = $this->generateRandomString();
            $sendMail = $this->mail->sendForgotPassword($admin[0]['email'], $admin[0]['username'],
                $passwordBaru);
            /*
             * Update Password
             */
            $query = "UPDATE ". TBL_PEGAWAI ." SET password = '". md5($passwordBaru) . "' WHERE IDPegawai='". $admin[0]['IDPegawai'] . "'";
            $result = $this->db->query($query);
            // print "<pre>";
            // print_r($admin) . "<br/>";
            // print $passwordBaru . "<br/>";
            // print $query . "<br/>";
            // exit;
            return true;
        }
        else
        {
            return $result['message'] = "Account anda sudah tidak aktif";
        }
    }
}

/**
* @desc generateRandomString for Pegawai password
*
* @param
* @return String
*/
function generateRandomString($length = 6, $letters =
'1234567890qwertyuiopasdfghjklzxcvbnm')
{
    $s = "";
    $lettersLength = strlen($letters)-1;

    for($i = 0 ; $i < $length ; $i++)
    {
        $s .= $letters[rand(0,$lettersLength)];
    }
    return $s;
}

/**
* @desc Check whether a Pegawai with specific IDPegawai exist in database
*
* @param IDPegawai
*/
function isExist( $IDPegawai = "" )
{
    if (strlen($IDPegawai) <= 0)
    {
        return false;
    }

    $result = 0;
    $query = "SELECT COUNT(a.IDPegawai) AS total FROM ". TBL_PEGAWAI ." a WHERE a.IDPegawai='". $IDPegawai . "'";

    if ($this->db)
    {
        $result = $this->db->query($query);
    }
    if ($result)
    {
        $count = 0;
        if (is_array($result))
        {
            $count = $result[0][0];
        }
        return $count > 0;
    }
    return false;
}

/**
* Remove a single Pegawai from database.
*
* @param IDPegawai to remove
* @return the number of object removed from database
*/
function remove($IDPegawai = "")
{
    $result = 0;
    if ((strlen($IDPegawai) > 0) && $this->db)
    {
        //DELETE FROM TABLE SYSTEM Pegawai
        $query = "DELETE FROM ". TBL_PEGAWAI ." WHERE IDPegawai = ". $IDPegawai . "";
        $result = $this->db->query($query);
    }
    return $result;
}

/**
* Escape a form data before adding to database.
*
* @param value the String to be escaped
* @return safe string
*/
function escape( $value )
{
    if (get_magic_quotes_gpc())
    {
        return $value;
    }
    else
    {
        return addslashes($value);
    }
}

function updatePassword($username)
{
    //UPDATE TABLE SYSTEM PEGAWAI DATA
    $query = "UPDATE ". TBL_PEGAWAI ." SET password = '". $this->escape( md5($username) ) . "' WHERE username='". $username . "'";
    $result = $this->db->query($query);
    // print "<pre>";
    // print $query . "<br/>";
    // exit;
    return true;
}

```

tlPiket.class.php

```
<?php

class tlPiket {
    var $db;

    function tlPiket($db_object)
    {
        $this->db = $db_object;
    }

    /**
     * @desc getAllPiketDataList in TBL_PIKET
     * @var
     * @param $count, $page, $_GET
     * @return Array
     */
    function getAllPiketDataList($count = 0, $page = 0, $_GET)
    {
        if ($this->db)
        {
            $namaPegawai = $_GET['namaPegawai'];

            $result = 0;
            $query = "SELECT * FROM ".TBL_PIKET." a LEFT JOIN(".
TBL_PEGAWAI." b)
                ON(a.IDPegawai=b.IDPegawaiPiket) WHERE 1=1";

            if (!empty($_GET))
            {
                if(!empty($namaPegawai) && $namaPegawai != "")
                {
                    $query .= " AND a.namaPegawai LIKE '%".
trim($namaPegawai). "%'";
                }
            }
            $query .= " ORDER BY a.namaPegawai ASC ";

            if (strlen($page) > 0)
            {
                $query .= " LIMIT ".($page * $count). ", ". $count;
            }
            $result = $this->db->query($query);
            print "<pre>";
            print $query ."<br/>";
            print_r($result);
            exit;
            return $result;
        }
        else
        {
            echo "Error line 43 tlPiket.class.php";
            return;
        }
    }

    /**
     * @desc Gets a total of Piket available in database.
     *
     * @param active the active flag
     * @return array or null
     */
    function getPiketCount( $_GET )
    {
        if ($this->db)
        {
            $namaPiket = $_GET['namaPiket'];

            $result = 0;
            $query = "SELECT * FROM ".TBL_PIKET." a LEFT JOIN(".
TBL_PEGAWAI." b)
                ON(a.IDPegawai=b.IDPegawaiPiket) WHERE 1=1";

            if (!empty($_GET))
            {
                if(!empty($namaPegawai) && $namaPegawai != "")
                {
                    $query .= " AND a.namaPegawai LIKE '%".
trim($namaPegawai). "%'";
                }
            }
            $query .= " ORDER BY a.namaPegawai ASC ";
        }
    }

    if (strlen($page) > 0)
    {
        $query .= " LIMIT ".($page * $count). ", ". $count;
    }
    $result = $this->db->query($query);
    echo "Error line 85 tlPiket.class.php";
    return;
}

/**
 * @desc getPiketById in TBL_PIKET at specific IDPiket
 * @param IDPiket from $_GET
 * @var IDPiket
 * @return Array
 */
function getPiketById( $IDPiket )
{
    if ($this->db)
    {
        $query = "SELECT * FROM ".TBL_PIKET." a WHERE
a.IDPiket='". $IDPiket . "'";
        $result = $this->db->query($query);
        // print "<pre>";
        // print $query;
        // exit;
        return $result;
    }
    else
    {
        echo "Error line 115 tlPiket.class.php";
        return;
    }
}

/**
 * @desc getPiketByParameter in TBL_PIKET at specific Parameter
 * @param Parameter
 * @var Parameter
 * @return Array
 */
function getPiketByParameter( $IDJurusan, $IDTahunPelajaran,
$tanggalPiket )
{
    if ($this->db)
    {
        $query = "SELECT * FROM ".TBL_PIKET." a LEFT JOIN(".
TBL_PEGAWAI." b)
            ON(a.IDPegawaiPiket = b.IDPegawai) WHERE ";
        if($IDJurusan != "")
        {
            $query .= "a.IDJurusan='". $IDJurusan . "' AND ";
        }
        $query .= "a.IDTahunPelajaran='". $IDTahunPelajaran . "' AND
a.tanggalPiket='". $tanggalPiket . "'";
        $result = $this->db->query($query);
        // print "<pre>";
        // print $query;
        // exit;
        return $result;
    }
    else
    {
        echo "Error line 146 tlPiket.class.php";
        return;
    }
}

/**
 * @desc getPiketByParameter2 in TBL_PIKET at specific Parameter2
 * @param Parameter 2
 * @var Parameter
 * @return Array
 */
function getPiketByParameter2( $IDJurusan, $IDTahunPelajaran,
$tanggalPiket, $IDPegawaiPiket )
{
    if ($this->db)
```

```

$query = "SELECT * FROM ". TBL_PIKET . " a LEFT JOIN (" .
TBL_PEGAWAI . " b)
    ON(a.IDPegawaiPiket = b.IDPegawai) WHERE
a.IDJurusan="" . $IDJurusan . ""
    AND a.IDTahunPelajaran="" . $IDTahunPelajaran . "" AND
a.tanggalPiket="" . $tanggalPiket . ""
    AND a.IDPegawaiPiket="" . $IDPegawaiPiket . "";
$result = $this->db->query($query);
//      print "<pre>";
//      print $query;
//      exit;
//      return $result;
}
else
{
    echo "Error line 146 tlPiket.class.php";
    return;
}

/***
* @desc countPiketByIDPegawai in TBL_PIKET at specific IDPegawai
* @param IDPegawai
* @var IDPegawai
* @return Array
*/
function countPiketByIDPegawai( $m, $y, $IDPegawai )
{
    if ($this->db)
    {
        $query = "SELECT COUNT(a.IDPiket) as jumlahPiket FROM " .
TBL_PIKET . " a WHERE 1=1
            AND a.IDPegawaiPiket="" . $IDPegawai . "" AND
MONTH(a.tanggalPiket)="" . $m . "
            AND YEAR(a.tanggalPiket)="" . $y . """;
        $result = $this->db->query($query);
//      print "<pre>";
//      print $query;
//      print_r($result);
//      exit;
        return $result;
    }
    else
    {
        echo "Error line 199 tlAbsensi.class.php";
        return;
    }
}

/***
* @desc Save an instance of $_POST to database.
* @param $_POST
*/
function save( $_POST )
{
    if ($this->db)
    {
        if ( $this->isExist($_POST['IDJurusan'],
$_POST['IDTahunPelajaran'], $_POST['tanggalPiket']) )
        {
//          print "<pre>";
//          print_r($_POST);
//          exit;
            $query = "UPDATE " . TBL_PIKET . " SET ";
            $query .= "IDJurusan="" . $this->escape( $_POST['IDJurusan'] ) .
. "", ";
            $query .= "IDTahunPelajaran="" . $this->escape(
$_POST['IDTahunPelajaran']) . "", ";
            $query .= "tanggalPiket="" . $this->escape(
$_POST['tanggalPiket']) . "", ";
            $query .= "IDPegawaiPiket="" . $this->escape(
$_POST['IDPegawaiPiket']) . "", ";
            $query .= "keteranganPiket="" . $this->escape(
$_POST['keteranganPiket']) . "" ";
            $query .= "WHERE IDJurusan="" . $_POST['IDJurusan'] . "
AND IDTahunPelajaran="" . $_POST['IDTahunPelajaran'] . "
                AND tanggalPiket="" . $_POST['tanggalPiket'] . """;
        }
        else
        {
//          print "<pre>";
//          print_r($_POST);
//          exit;
        }
    }
}

```

```

$query = "INSERT INTO " . TBL_PIKET . " ";
$query .= "(IDPiket, IDJurusan, IDTahunPelajaran,
tanggalPiket, IDPegawaiPiket, keteranganPiket) ";
$query .= "VALUES (";
$query .= "", ", ";
$query .= "" . $this->escape( $_POST['IDJurusan'] ) . "", ";
$query .= "" . $this->escape( $_POST['IDTahunPelajaran'] ) . "", ";
";
$query .= "" . $this->escape( $_POST['tanggalPiket'] ) . "", ";
$query .= "" . $this->escape( $_POST['IDPegawaiPiket'] ) . "", ";
$query .= "" . $this->escape( $_POST['keteranganPiket'] ) . "" ";
$query .= ")";
}
//      print "<pre>";
//      print $query."<br/>";
//      exit;
$result = $this->db->query($query);
return $result;
}

/***
* @desc Check whether a Piket with specific parameter exist in
database
*
*/
function isExist( $IDJurusan, $IDTahunPelajaran, $tanggalPiket )
{
    $result = 0;
    $query = "SELECT COUNT(a.IDPiket) AS total FROM " .
TBL_PIKET . " a
        WHERE a.IDJurusan="" . $IDJurusan . "" AND
a.IDTahunPelajaran="" . $IDTahunPelajaran . "" AND
        a.tanggalPiket="" . $tanggalPiket . """;
    if ($this->db)
    {
        $result = $this->db->query($query);
    }
    if ($result)
    {
        $count = 0;
        if (is_array($result))
        {
            $count = $result[0][0];
        }
        return $count > 0;
    }
    return false;
}

/***
* Remove a single Piket from database.
*
*/
function remove($IDPiket = "")
{
    $result = 0;
    if ((strlen($IDPiket) > 0) && $this->db)
    {
        $query = "DELETE FROM " . TBL_PIKET . " WHERE IDPiket =
" . $IDPiket . "";
        $result = $this->db->query($query);
    }
    return $result;
}

/***
* Escape a form data before adding to database.
*
*/
function escape( $value )
{
    if (get_magic_quotes_gpc())
    {
        return $value;
    }
    else
    {

```

```

        return addslashes($value);
    }
}

?>

tlTahunPelajaran.class.php

<?php

class tlTahunPelajaran {
    var $db;

    function tlTahunPelajaran($db_object)
    {
        $this->db = $db_object;
    }

    /**
     * @desc getAllTahunPelajaran in TBL_TAHUNPELAJARAN
     * @var
     * @param
     * @return Array
     */
    function getAllTahunPelajaran()
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ". TBL_TAHUNPELAJARAN . " a
WHERE 1=1 ORDER BY IDTahunPelajaran ASC";
            $result = $this->db->query($query);
            // print "<pre>";
            // print $query . "<br/>";
            // print_r($result);
            // exit;
            return $result;
        }
        else
        {
            echo "Error line 40 tlTahunPelajaran.class.php";
            return;
        }
    }

    /**
     * @desc getAllTahunPelajaranDataList in TBL_JABATAN
     * @var
     * @param $count, $page, $_GET
     * @return Array
     */
    function getAllTahunPelajaranDataList($count = 0, $page = 0, $_GET,
$status)
    {
        if ($this->db)
        {
            $namaTahunPelajaran = $_GET['namaTahunPelajaran'];

            $result = 0;
            $query = "SELECT * FROM ". TBL_TAHUNPELAJARAN . " a
WHERE 1=1";
            if (!empty($_GET))
            {
                if(!empty($namaTahunPelajaran) && $namaTahunPelajaran != "")
                {
                    $query .= " AND a.namaTahunPelajaran LIKE '%" .
trim($namaTahunPelajaran) . "%'";
                }
            }
            if(!empty($status) && $status > 0)
            {
                $query .= " AND a.statusTahunPelajaran = '1'";
            }
            $query .= " ORDER BY a.IDTahunPelajaran ASC ";

            if (strlen($page) > 0)
            {
                $query .= " LIMIT " . ($page * $count) . ", " . $count;
            }
            $result = $this->db->query($query);
            // print "<pre>";
            // print $query . "<br/>";
            // print_r($result);
        }
    }

    /**
     * @desc getTahunPelajaranCount in TBL_TAHUNPELAJARAN
     * @param
     * @return array or null
     */
    function getTahunPelajaranCount( $_GET )
    {
        if ($this->db)
        {
            $namaTahunPelajaran = $_GET['namaTahunPelajaran'];

            $result = 0;
            $query = "SELECT COUNT(a.IDTahunPelajaran) AS total
FROM ". TBL_TAHUNPELAJARAN . " a
WHERE 1=1";
            if (!empty($_GET))
            {
                if(!empty($namaTahunPelajaran) && $namaTahunPelajaran != "")
                {
                    $query .= " AND a.namaTahunPelajaran LIKE '%" .
trim($namaTahunPelajaran) . "%'";
                }
            }
            $result = $this->db->query($query);
            return $result[0];
        }
        else
        {
            echo "Error line 85 tlTahunPelajaran.class.php";
            return;
        }
    }

    /**
     * @desc getTahunPelajaranById in TBL_TAHUNPELAJARAN at
specific IDTahunPelajaran
     * @param IDTahunPelajaran from $_GET
     * @var IDTahunPelajaran
     * @return Array
     */
    function getTahunPelajaranById( $IDTahunPelajaran )
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ". TBL_TAHUNPELAJARAN . " a
WHERE a.IDTahunPelajaran=" . $IDTahunPelajaran . "";
            $result = $this->db->query($query);
            // print "<pre>";
            // print $query;
            // exit;
            return $result;
        }
        else
        {
            echo "Error line 115 tlTahunPelajaran.class.php";
            return;
        }
    }

    /**
     * @desc getTahunPelajaranActive in TBL_TAHUNPELAJARAN at
specific Jurusan
     * @param $IDJurusan
     * @var $IDJurusan
     * @return Array
     */
    function getTahunPelajaranActiveByJurusan( $IDJurusan )
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ". TBL_TAHUNPELAJARAN . " a
WHERE a.statusTahunPelajaran='1' "
        }
    }
}
// exit;
return $result;
}
else
{
    echo "Error line 43 tlTahunPelajaran.class.php";
    return;
}
/**/
/* @desc Gets a total of TahunPelajaran available in database.
*/
/* @param
 * @return array or null
*/
function getTahunPelajaranCount( $_GET )
{
    if ($this->db)
    {
        $namaTahunPelajaran = $_GET['namaTahunPelajaran'];

        $result = 0;
        $query = "SELECT COUNT(a.IDTahunPelajaran) AS total
FROM ". TBL_TAHUNPELAJARAN . " a
WHERE 1=1";
        if (!empty($_GET))
        {
            if(!empty($namaTahunPelajaran) && $namaTahunPelajaran != "")
            {
                $query .= " AND a.namaTahunPelajaran LIKE '%" .
trim($namaTahunPelajaran) . "%'";
            }
        }
        $result = $this->db->query($query);
        return $result[0];
    }
    else
    {
        echo "Error line 85 tlTahunPelajaran.class.php";
        return;
    }
}

/**
 * @desc getTahunPelajaranById in TBL_TAHUNPELAJARAN at
specific IDTahunPelajaran
 * @param IDTahunPelajaran from $_GET
 * @var IDTahunPelajaran
 * @return Array
 */
function getTahunPelajaranById( $IDTahunPelajaran )
{
    if ($this->db)
    {
        $query = "SELECT * FROM ". TBL_TAHUNPELAJARAN . " a
WHERE a.IDTahunPelajaran=" . $IDTahunPelajaran . "";
        $result = $this->db->query($query);
        // print "<pre>";
        // print $query;
        // exit;
        return $result;
    }
    else
    {
        echo "Error line 115 tlTahunPelajaran.class.php";
        return;
    }
}

/**
 * @desc getTahunPelajaranActive in TBL_TAHUNPELAJARAN at
specific Jurusan
 * @param $IDJurusan
 * @var $IDJurusan
 * @return Array
 */
function getTahunPelajaranActiveByJurusan( $IDJurusan )
{
    if ($this->db)
    {
        $query = "SELECT * FROM ". TBL_TAHUNPELAJARAN . " a
WHERE a.statusTahunPelajaran='1' "
    }
}

```


tlUser.class.php

```
<?php

class tlUser {
    var $db;

    function tlUser($db_object)
    {
        $this->db = $db_object;
    }

    /**
     * @desc getAllUser in TBL_USER
     * @var
     * @param
     * @return Array
     */
    function getAllUser()
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ".TBL_USER." a WHERE 1=1";
            $result = $this->db->query($query);
            // print "<pre>";
            // print $query . "<br/>";
            // print_r($result);
            // exit;
            return $result;
        }
        else
        {
            echo "Error line 40 tlUser.class.php";
            return;
        }
    }

    /**
     * @desc getAllUserDataList in TBL_USER
     * @var
     * @param $count, $page, $_GET
     * @return Array
     */
    function getAllUserDataList($count = 0, $page = 0, $_GET)
    {
        if ($this->db)
        {
            $namaUser = $_GET['namaUser'];

            $result = 0;
            $query = "SELECT * FROM ".TBL_USER." a WHERE 1=1";

            if (!empty($_GET))
            {
                if(!empty($namaUser) && $namaUser != "")
                {
                    $query .= " AND a.namaUser LIKE '%" . trim($namaUser) . "%'";
                }
            }
            $query .= " ORDER BY a.IDUser ASC ";

            if (strlen($page) > 0)
            {
                $query .= " LIMIT " . ($page * $count) . ", " . $count;
            }
            $result = $this->db->query($query);
            // print "<pre>";
            // print $query . "<br/>";
            // print_r($result);
            // exit;
            return $result;
        }
        else
        {
            echo "Error line 43 tlUser.class.php";
            return;
        }
    }

    /**
     * @desc Gets a total of User available in database.
     */
    function getUserCount( $_GET )
    {
        if ($this->db)
        {
            $namaUser = $_GET['namaUser'];

            $result = 0;
            $query = "SELECT COUNT(a.IDUser) AS total FROM ".TBL_USER." a
                      WHERE 1=1";

            if (!empty($_GET))
            {
                if(!empty($namaUser) && $namaUser != "")
                {
                    $query .= " AND a.namaUser LIKE '%" . trim($namaUser) . "%'";
                }
            }
            $result = $this->db->query($query);
            return $result[0];
        }
        else
        {
            echo "Error line 85 tlUser.class.php";
            return;
        }
    }

    /**
     * @desc getUserById in TBL_USER at specific IDUser
     * @param IDUser from $_GET
     * @var IDUser
     * @return Array
     */
    function getUserById( $IDUser )
    {
        if ($this->db)
        {
            $query = "SELECT * FROM ".TBL_USER." a WHERE
a.IDUser=" . $IDUser . "";
            $result = $this->db->query($query);
            // print "<pre>";
            // print $query;
            // exit;
            return $result;
        }
        else
        {
            echo "Error line 115 tlUser.class.php";
            return;
        }
    }

    /**
     * @desc Save an instance of $_POST to database.
     * @param $_POST
     */
    function save( $_POST )
    {
        if ($this->db)
        {
            if ( $this->isExist($_POST['IDUser']) )
            {
                // print "<pre>";
                // print_r($_POST);
                // exit;

                //UPDATE TABLE PERSON DATA
                $query = "UPDATE ".TBL_USER." SET ";
                $query .= "namaUser=" . $this->escape( $_POST['namaUser'] ) . ",";
                $query .= "keteranganUser=" . $this->escape(
$_POST['keteranganUser'] ) . " ";
                $query .= "WHERE IDUser=" . $_POST['IDUser'] . "";

                // print "<pre>";
                // print_r($_POST);
            }
            else
            {
                // print "<pre>";
                // print_r($_POST);
            }
        }
    }
}
```

```

//      exit;
/*
 * INSERT INTO TBL_USER
 */
$query = "INSERT INTO " . TBL_USER . " ";
$query .= "(IDUser, namaUser, keteranganUser) ";
$query .= "VALUES (";
$query .= "", ",";
$query .= "" . $this->escape( $_POST['namaUser'] ) . "", ",";
$query .= "" . $this->escape( $_POST['keteranganUser'] ) . "" ;
$query .= ")";
}
}

// print "<pre>";
// print $query."<br/>";
// exit;
$result = $this->db->query($query);
return $result;
}

/**
* @desc Check whether a User with specific IDUser exist in database
*
* @param IDUser
* @return true if exist false otherwise
*/
function isExist( $IDUser = "" )
{
if (strlen($IDUser) <= 0)
{
    return false;
}

$result = 0;
$query = "SELECT COUNT(a.IDUser) AS total FROM " .
TBL_USER . " a
WHERE a.IDUser="" . $IDUser . """;

if ($this->db)
{
    $result = $this->db->query($query);
}
if ($result)
{
    $count = 0;
    if (is_array($result))
    {
        $count = $result[0][0];
    }
    return $count > 0;
}
return false;
}

/**
* Remove a single User from database.
*
* @param IDUser to remove
* @return the number of object removed from database
*/
function remove($IDUser = "")
{
    $result = 0;
    if ((strlen($IDUser) > 0) && $this->db)
    {
        $query = "DELETE FROM " . TBL_USER . " WHERE IDUser =
"" . $IDUser . """;
        $result = $this->db->query($query);
    }
    return $result;
}

/**
* Escape a form data before adding to database.
*
* @param value the String to be escaped
* @return safe string
*/
function escape( $value )
{
    if (get_magic_quotes_gpc())
    {
        return $value;
    }
    else

```

```

    {
        return addslashes($value);
    }
}

?>

tlWaktuMengajar.class.php

<?php

class tlWaktuMengajar {
    var $db;

    function tlWaktuMengajar($db_object)
    {
        $this->db = $db_object;
    }

    /**
     * @desc getAllWaktuMengajar in TBL_WAKTUMENGAJAR
     * @var
     * @param
     * @return Array
     */
    function getAllWaktuMengajar()
    {
        if ($this->db)
        {
            $query = "SELECT * FROM " . TBL_WAKTUMENGAJAR . " a
WHERE 1=1";
            $result = $this->db->query($query);
            // print "<pre>";
            // print $query . "<br/>";
            // print_r($result);
            // exit;
            return $result;
        }
        else
        {
            echo "Error line 40 tlWaktuMengajar.class.php";
            return;
        }
    }

    /**
     * @desc getAllWaktuMengajarDataList in TBL_WAKTUMENGAJAR
     * @var
     * @param $count, $page, $_GET
     * @return Array
     */
    function getAllWaktuMengajarDataList($count = 0, $page = 0, $_GET)
    {
        if ($this->db)
        {
            $namaTahunPelajaran = $_GET['namaTahunPelajaran'];
            $namaPegawai      = $_GET['namaPegawai'];

            $result = 0;
            $query = "SELECT * FROM " . TBL_WAKTUMENGAJAR . " a
LEFT JOIN (" . TBL_TAHUNPELAJARAN . " b, " . TBL_PEGAWAI . "
c)
ON(a.IDTahunPelajaran=b.IDTahunPelajaran AND
a.IDPegawai=c.IDPegawai) WHERE 1=1 AND
b.statusTahunPelajaran=1";

            if (!empty($_GET))
            {
                if(!empty($namaTahunPelajaran) && $namaTahunPelajaran != "")
                {
                    $query .= " AND b.namaTahunPelajaran = " .
trim($namaTahunPelajaran) . "";
                }
                if(!empty($namaPegawai) && $namaPegawai != "")
                {
                    $query .= " AND c.namaPegawai LIKE '%" .
trim($namaPegawai) . "%'";
                }
            }
            $query .= " ORDER BY a.IDTahunPelajaran DESC,
c.namaPegawai ASC";
        }
        if (strlen($page) > 0)

```

```

        {
            $query .= " LIMIT " . ($page * $count) . ", " . $count;
        }
        $result = $this->db->query($query);
//        print "<pre>";
//        print $query . "<br/>";
//        print_r($result);
//        exit;
        return $result;
    }
else
{
    echo "Error line 43 tlWaktuMengajar.class.php";
    return;
}
}

/**
* @desc Gets a total of WaktuMengajar available in database.
*
* @param active the active flag
* @return array or null
*/
function getWaktuMengajarCount( $_GET )
{
    if ($this->db)
    {
        $namaTahunPelajaran = $_GET['namaTahunPelajaran'];
        $namaPegawai      = $_GET['namaPegawai'];

        $result = 0;
        $query = "SELECT COUNT(a.IDWaktuMengajar) AS total
FROM ". TBL_WAKTUMENGAJAR ." a LEFT JOIN (" .
        TBL_TAHUNPELAJARAN ." b, ". TBL_PEGAWAI ." c)
        ON(a.IDTahunPelajaran=b.IDTahunPelajaran AND
a.IDPegawai=c.IDPegawai) WHERE 1=1 AND
b.statusTahunPelajaran=1";

        if (!empty($_GET))
        {
            if(!empty($namaTahunPelajaran) && $namaTahunPelajaran != "")
            {
                $query .= " AND b.namaTahunPelajaran = " .
                trim($namaTahunPelajaran) . "";
            }
            if(!empty($namaPegawai) && $namaPegawai != "")
            {
                $query .= " AND c.namaPegawai LIKE '%" .
                trim($namaPegawai) . "%'";
            }
        }
        $query .= " ORDER BY a.IDTahunPelajaran DESC,
c.namaPegawai ASC";
        $result = $this->db->query($query);
        return $result[0];
    }
else
{
    echo "Error line 85 tlWaktuMengajar.class.php";
    return;
}
}

/**
* @desc getWaktuMengajarById in TBL_WAKTUMENGAJAR at
specific IDWaktuMengajar
* @param IDWaktuMengajar from $_GET
* @var IDWaktuMengajar
* @return Array
*/
function getWaktuMengajarById( $IDWaktuMengajar )
{
    if ($this->db)
    {
        $query = "SELECT * FROM ". TBL_WAKTUMENGAJAR ." a
LEFT JOIN (" . TBL_TAHUNPELAJARAN ." b, ". TBL_PEGAWAI ." c)
        ON(a.IDTahunPelajaran=b.IDTahunPelajaran AND
a.IDPegawai=c.IDPegawai)
        WHERE a.IDWaktuMengajar='". $IDWaktuMengajar . "'";
        $result = $this->db->query($query);
//        print "<pre>";
//        print $query;
//        exit;
    }
}

return $result;
}
else
{
    echo "Error line 115 tlWaktuMengajar.class.php";
    return;
}
}

/**
* @desc getWaktuMengajarByIdPegawai in
TBL_WAKTUMENGAJAR at specific IDPegawai
* @param IDPegawai from $_GET
* @var IDPegawai
* @return Array
*/
function getWaktuMengajarByIdPegawai( $IDTahunPelajaran,
$IDPegawai )
{
    if ($this->db)
    {
        $query = "SELECT * FROM ". TBL_WAKTUMENGAJAR ." a
LEFT JOIN (" . TBL_TAHUNPELAJARAN ." b, ". TBL_PEGAWAI ." c)
        ON(a.IDTahunPelajaran=b.IDTahunPelajaran AND
a.IDPegawai=c.IDPegawai)
        WHERE a.IDTahunPelajaran='". $IDTahunPelajaran . "'"
        AND a.IDPegawai='". $IDPegawai . "'";
        $result = $this->db->query($query);
//        print "<pre>";
//        print $query;
//        exit;
        return $result;
    }
else
{
    echo "Error line 178 tlWaktuMengajar.class.php";
    return;
}
}

/**
* @desc Save an instance of $_POST to database.
* @param $_POST
*/
function save( $_POST )
{
    if ($this->db)
    {
        if ( $this->isExist($_POST['IDWaktuMengajar']) )
        {
//            print "<pre>";
//            print_r($_POST);
//            exit;

            //UPDATE TABLE PERSON DATA
            $query = "UPDATE ". TBL_WAKTUMENGAJAR ." SET ";
            $query .= "IDTahunPelajaran='". $this->escape(
$_POST['IDTahunPelajaran'] ) . "', ";
            $query .= "IDPegawai='". $this->escape( $_POST['IDPegawai'] ) . "', ";
            $query .= "jumlahWaktuMengajar='". $this->escape(
$_POST['jumlahWaktuMengajar'] ) . "' ";

            $query .= "WHERE IDWaktuMengajar='". $POST['IDWaktuMengajar'] . "'";
        }
        else
        {
//            print "<pre>";
//            print_r($_POST);
//            exit;
/*
            * INSERT INTO TBL_WAKTUMENGAJAR
*/
            $query = "INSERT INTO ". TBL_WAKTUMENGAJAR ." ";
            $query .= "(IDWaktuMengajar, IDTahunPelajaran, IDPegawai,
jumlahWaktuMengajar)";
            $query .= "VALUES (";
            $query .= "''", ",";
            $query .= "''". $this->escape( $_POST['IDTahunPelajaran'] ) . ",";
            $query .= "''". $this->escape( $_POST['IDPegawai'] ) . ",";
            $query .= "''";
        }
    }
}

```

```

        $query .= "" . $this->escape( $_POST['jumlahWaktuMengajar'] )
    ) . '"';
        $query .= " ) ";
    }
}
// print "<pre>";
// print $query."<br/>";
// exit;
$result = $this->db->query($query);
return $result;
}

/**
 * @desc Check whether a WaktuMengajar with specific
IDWaktuMengajar exist in database
*/
* @param IDWaktuMengajar
* @return true if exist false otherwise
*/
function isExist( $IDWaktuMengajar = "" )
{
    if (strlen($IDWaktuMengajar) <= 0)
    {
        return false;
    }

    $result = 0;
    $query = "SELECT COUNT(a.IDWaktuMengajar) AS total
FROM " . TBL_WAKTUMENGAJAR . " a
WHERE a.IDWaktuMengajar="" . $IDWaktuMengajar . """;

    if ($this->db)
    {
        $result = $this->db->query($query);
    }
    if ($result)
    {
        $count = 0;
        if (is_array($result))
        {
            $count = $result[0][0];
        }
        return $count > 0;
    }
    return false;
}

/**
* Remove a single WaktuMengajar from database.
*/
* @param IDWaktuMengajar to remove
* @return the number of object removed from database
*/
function remove($IDWaktuMengajar = "")
{
    $result = 0;
    if ((strlen($IDWaktuMengajar) > 0) && $this->db)
    {
        $query = "DELETE FROM " . TBL_WAKTUMENGAJAR . "
WHERE IDWaktuMengajar = "" . $IDWaktuMengajar . """;
        $result = $this->db->query($query);
    }
    return $result;
}

/**
* Escape a form data before adding to database.
*/
* @param value the String to be escaped
* @return safe string
*/
function escape( $value )
{
    if (get_magic_quotes_gpc())
    {
        return $value;
    }
    else
    {
        return addslashes($value);
    }
}

}
?>

```

index.php

```

<?php // $Id: index.php 194 2011-05-01 11:02:11Z afrialdi.syahputra $ ?>
<?php
//ob_start();
include_once("../init.noauth.inc.php");

if (!isset($_GET['action'])){
    header('location: '.ADMIN_URL.'home.php');
    exit;
}

switch ($_GET['action']){
    case 'login':
        $status = $tlPegawai->authenticate( $_POST['username'],
$_POST['password'], $_POST['level']);
        /*
         * Result Action
         */
        if ($tlPegawai->checkSession() === true ){
            header('location: '.ADMIN_URL.'home.php?info=success');
            exit;
        }
        else{
            header('location:
'.ADMIN_URL.'login/index.php?action='.$status.'&err=invalid');
            exit;
        }
        break;
    case 'logout':
        $logoutResult = $tlPegawai->logout();
        if($logoutResult == TRUE){
            header('location: '.ADMIN_URL.'login/index.php?action=logout');
        }
        break;
}
//ob_end_flush();
?>

```

home.php

```

<?php // $Id: home.php 194 2011-05-01 11:02:11Z afrialdi.syahputra $ ?>
<?php

include_once("../init.auth.inc.php");

$pegawaiData = $tlPegawai->getPegawaiById(
$_SESSION['IDPEGAWAI'] );

$config->show_header();

?>
<div id="content" style="padding:0px 10px;">

<div class="content-detail">

<div class="content-title home" style="padding-left: 20px;">
    <?php echo $str['welcome'];?>, <?php echo
$_SESSION['NAMAPEGAWAI'];?></div>
    <br/>

    <?php echo $str['home_page'];?>

<div class="clear">&nbsp;</div>

<div class="content-title"><?php echo $str['history'];?></div>

    <?php echo $str['username'];?>; <?php echo
$pegawaiData[0]['username']?><br/>
    <?php echo $str['last_login'];?>; <?php echo $config-
>toIndonesianFullDate(strtotime($pegawaiData[0]['loginTerakhir']))?><br
/>
        <?php echo $str['last_IP'];?>; <?php echo $pegawaiData[0]['IP']?>
    </div>
</div>

<?php
$config->show_footer();
?>

```