

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manusia merupakan motor penggerak organisasi dalam menciptakan kemampuan organisasi untuk tetap bertahan dan berkembang. Adanya manusia yang bekerja pada organisasi, membuat organisasi atau perusahaan berkewajiban memberi balas jasa yang setimpal terhadap jasa yang diberikan manusia itu. Balas jasa itu berupa gaji, tunjangan, bonus, dan lain-lain.

Pembayaran gaji memerlukan suatu tata cara yang dapat mengatur pengajian dengan baik, karena sistem penggajian merupakan penunjang yang membantu organisasi atau perusahaan dalam memberikan informasi bagi manajemen untuk mengambil keputusan. Perkembangan kemajuan teknologi seiring dengan kemajuan penggunaan komputer yang begitu pesat sangatlah membantu dalam dunia usaha, sehingga penanganan data secara manual sudah mulai ditinggalkan.

Bertolak dalam hal tersebut, penulis mempertimbangkan bahwa sistem penggajian sangat tepat untuk dikomputerisasi mengingat bahwa data yang diolah cukup banyak dan proses pengolahannya dapat dipersingkat sehingga keinginan perusahaan dan pegawai akan kebutuhan informasi dapat dipenuhi tepat pada waktunya.

1.2 Batasan Masalah

Batasan permasalahan yang akan dibahas mengenai perancangan sistem penggajian karyawan pada perusahaan, yaitu hanya terbatas pada masalah-masalah yang berhubungan dengan sistem penggajian yang mekanisme dan metode yang dipergunakan para administrator untuk mengontrol apa yang boleh diakses pengguna, termasuk apa yang boleh dilakukan setelah otentikasi dan otorisasi, dan pemantauannya.

1.3 Tujuan Penulisan

Laporan tugas akhir ini mempunyai tujuan sebagai berikut:

1. Terbentuknya perangkat lunak yang dapat membantu dalam mendukung sistem penggajian pegawai pada sebuah organisasi/perusahaan.
2. Mempermudah kinerja divisi HRD dalam menghitung rician gaji karyawan.

1.4 Metodologi

Dalam melakukan perancangan aplikasi ini metodologi perangkat lunak yang digunakan oleh penulis adalah metode Daur Hidup Pengembangan Sistem, diantaranya:

1. Inisialisasi sistem dan mempelajari kemungkinan
2. Analisa sistem
3. Rancangan sistem
4. Pemrograman
5. Penerapan
6. Evaluasi

Tahapan-tahapan dalam waterfall adalah sebagai berikut:

1. Tahapan Rekayasa Sistem.

Tahapan ini menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak

2. Tahapan Analisis.

Untuk memahami perangkat lunak yang dibangun, perancang perangkat lunak harus memahami domain informasi, tingkah laku, unjuk kerja, dan antarmuka yang diperlukan

3. Tahapan Perancangan.

Proses desain menerjemahkan kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode

4. Tahapan Pemrograman.

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca.

5. Tahapan Pengujian.

Pengujian aplikasi berfokus pada logika di dalam perangkat lunak, untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Tahapan Pemeliharaan.

Karena perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan.

1.5 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini akan disusun menjadi lima bab dengan perinciannya sebagai berikut:

Bab I Pendahuluan

Pada bab ini akan diuraikan latar belakang, batasan permasalahan, tujuan penulisan, metodologi, dan sistematika penulisan.

Bab II Landasan Teori

Dalam bab ini penulis menguraikan teori-teori yang digunakan penulis untuk mendukung permasalahan. Landasan teori ini meliputi konsep daur hidup pengembangan sistem, *unified modeling language* (uml), diagram *use case*, diagram aktifitas, diagram transisi keadaan, konsep basis data, *structured query language*, dan konsep php dan mysql.

Bab III Analisis dan Perancangan

Bab ini berisi mengenai langkah-langkah dalam pembuatan sistem penggajian karyawan dengan bahasa pemrograman PHP.

Bab IV Implementasi dan Pengujian

Dalam bab ini penulis mencoba mengimplementasikan dan menguji fungsi-fungsi program yang penulis buat.

Bab V Penutup

Dalam bab ini penulis mencoba mengambil kesimpulan berdasarkan analisis-*analisis* yang dilakukan. Selain itu penulis juga memberikan saran yang kiranya dapat bermanfaat bagi para *programmer* yang sedang mempelajari pembuatan program sistem penggajian karyawan berbasis web dengan menggunakan bahasa pemrograman PHP.

BAB II

LANDASAN TEORI

2.1 Daur Hidup Pengembangan Sistem

Menurut Kenneth E. Kendall, Julie E. Kendall (1992: 6) Daur hidup pengembangan sistem adalah fase pendekatan untuk analisa dan desain untuk suatu sistem yang dikembangkan melalui daur tertentu dari analisis dan aktifitas pengguna.

Daur hidup pengembangan sistem memiliki enam langkah, yaitu:

1. Inisialisasi sistem dan mempelajari kemungkinan

Inisialisasi sistem dan mempelajari kemungkinan merupakan langkah untuk melakukan dokumentasi atas sistem yang ada saat ini, tujuan dan manfaat pembuatan sistem baru atau pengembangan dari sistem yang telah ada, identifikasi untuk alternatif solusi sistem dan penentuan solusi secara pasti.

2. Analisis sistem

Analisa sistem merupakan langkah untuk membuat sketsa awal perencanaan proyek dan dokumentasi untuk sistem baru.

3. Perancangan sistem

Perancangan sistem merupakan langkah untuk menghasilkan spesifikasi arsip (*file*) basis data, spesifikasi masukan/keluaran, spesifikasi modul program dan rencana proyek yang lebih rinci.

4. Pemrograman

Pemrograman merupakan langkah untuk membuat dan menghasilkan program yang dibutuhkan oleh sistem baru

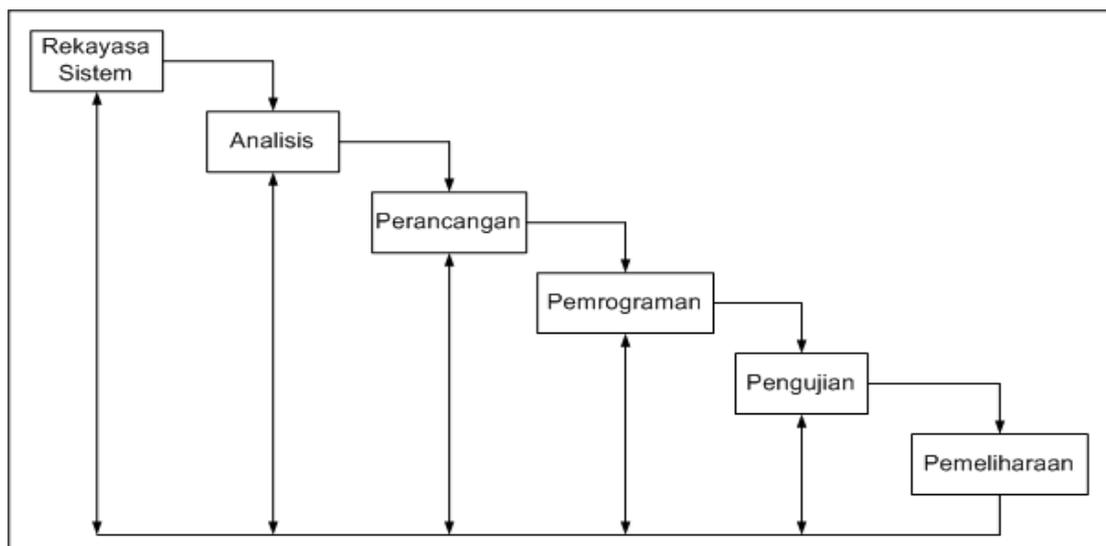
5. Penerapan

Penerapan merupakan langkah dimana pengguna melakukan pengujian terhadap program yang telah dibuat, perubahan *file* basis data, pelatihan pengguna, dan melakukan pengenalan terhadap sistem baru.

6. Evaluasi penerapan

Evaluasi penerapan merupakan langkah untuk melakukan peninjauan ulang apakah sistem ini memenuhi kebutuhan pengguna.

Model Waterfall merupakan salah satu dari model-model yang terdapat pada penerapan Daur Hidup Pengembangan Sistem. Roger.S, Pressman (2003: 37) membagi model Waterfall ke dalam beberapa tahap, yaitu: tahap rekayasa sistem, analisis kebutuhan perangkat lunak, perancangan, pemrograman, pengujian, dan pemeliharaan yang dapat digambarkan pada Gambar 2.1.



Gambar 2.1. Skema model Waterfall

Tahapan-tahapan dalam *waterfall* adalah sebagai berikut:

1. Tahapan Rekayasa Sistem. Tahapan ini menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak.
2. Tahapan Analisis. Pada tahap ini dilakukan pengumpulan perangkat lunak. Untuk memahami perangkat lunak yang dibangun, perekayasa perangkat lunak harus memahami domain informasi, tingkah laku, unjuk kerja, dan antarmuka yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.
3. Tahapan Perancangan. Tahapan perancangan perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut program yang berbeda; struktur data, arsitek perangkat lunak, representasi antarmuka, dan detail prosedural. Proses desain menerjemahkan kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.
4. Tahapan Pemrograman. Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pemrograman melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.
5. Tahapan Pengujian. Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika di dalam perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan di luar fungsi – yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang

dibutuhkan.

6. Tahapan Pemeliharaan. Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk menampung perubahan-perubahan di dalam lingkungan luarnya. Pemeliharaan perangkat lunak menerapkan lagi setiap tahap program sebelumnya dan tidak membuat yang baru lagi.

2.2 Unified Modeling Language (UML)

Unified Modeling Language selanjutnya disebut UML adalah suatu bahasa permodelan standar untuk menulis rancangan perangkat lunak. UML dapat digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi suatu perangkat lunak yang intensif dari suatu sistem. Sebuah pembuat perangkat lunak yang sehat atau normal paling tidak dapat menghasilkan dokumen persyaratan (*requirements*), arsitektur (*architecture*), pemodelan (*design*), kode program (*source code*), perencanaan proyek (*project plants*), pengujian (*tests*), prototipe (*prototypes*), dan penggunaan (*releases*).

UML memiliki tiga unsur penting, yaitu:

- a. Blok-blok bangunan, terdiri dari tiga jenis, yaitu berbagai hal, hubungan, dan diagram,
- b. Aturan yang mengatur bagaimana blok-blok itu dihubungkan,
- c. Mekanisme yang dapat digunakan.

Untuk memahami UML, perlu diketahui tiga karakteristik penting dari UML, yaitu:

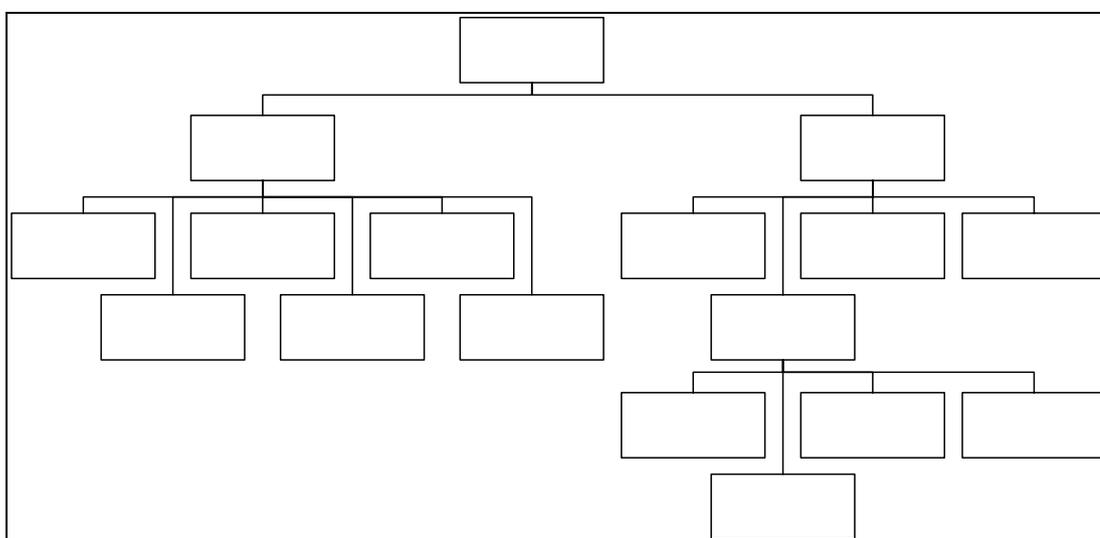
1. Pengendali *use case* (*use case driven*), berarti *use case* digunakan sebagai awalan untuk membuat perilaku, verifikasi dan validasi arsitektur sistem. Selanjutnya *use case* digunakan untuk pengetesan sistem dan sebagai alat komunikasi di antara pihak-pihak yang berkepentingan dengan pembangunan sistem,
2. Arsitektur yang sentris (*architecture centric*), berarti arsitektur sistem digunakan sebagai pegangan utama untuk membuat konsep, mengkonstruksi, mengatur, dan menyusun sistem yang sedang dikembangkan,
3. Iterasi (*iterative*) dan proses penambahan (*incremental process*). *Iterative* berarti proses itu menyangkut pernyataan atau keputusan yang dapat dikerjakan secara berkelanjutan. *Incremental process* adalah satu proses yang melibatkan integrasi terus menerus dari arsitektur sistem untuk menghasilkan pernyataan atau keputusan tersebut diikuti oleh pernyataan atau keputusan berikutnya yang lebih baik dari sebelumnya. *Iterative* dan *incremental process* adalah pengendali resiko (*risk driven*), artinya pernyataan atau keputusan yang baru difokuskan untuk mengatasi atau mengurangi risiko yang paling besar untuk suksesnya sistem yang dibangun.

Penting sekali untuk membedakan antara UML dan diagram-diagram bagian dari sistem. Sebuah diagram adalah sebuah bagian yang mewakili dari suatu sistem model. Model juga mengandung sebuah backplane semantik yang tersimpan, seperti yang sudah tertulis dalam *use case*, yang mengarahkan pada model elemen dan diagram. Ada tiga bagian terkemuka (*prominent*) dari sistem model, yaitu:

1. Model fungsional (*functional model*), menunjukkan fungsi dari masalah suatu sistem dari sudut pandang pengguna, termasuk *use case* diagram.

2. Model objek (*object model*), menunjukkan masalah stuktur dan sub-struktur dari sistem yang menggunakan objek, atribut, operasi dan hubungan, termasuk diagram kelas (*class diagram*).
3. Model dinamis (*dynamic model*), menunjukkan masalah perilaku internal dari sistem, termasuk diagram urutan (*sequence diagram*), diagram aktifitas (*activity diagram*), dan diagram status mesin (*state machine diagram*).

Dalam UML 2.0 ada tiga belas jenis diagram. Untuk memahaminya, diagram tersebut dapat dikategorikan secara hirarki, seperti yang terlihat pada Gambar 2.2.



Gambar 2.2 Hirarki diagram UML (wikipedia.org)

Dari Gambar 2.2 dapat dijelaskan sebagai berikut:

1. Diagram struktur (*structure diagram*) menjelaskan apa yang harus ada pada sistem yang dimodelkan, terdiri atas enam diagram, yaitu:
 - a. Diagram kelas (*class diagram*),
 - b. Diagram komponen (*component diagram*),
 - c. Diagram struktur gabungan (*composite structure diagram*),
 - d. Diagram pengembangan (*deployment diagram*),

Class
Diagram

Component
Diagram

Object
Diagram

Composite

Deployment

- e. Diagram objek (*object diagram*),
 - f. Diagram paket (*package diagram*).
2. Diagram perilaku (*behavior diagram*) menjelaskan apa yang harus terjadi pada sistem yang dimodelkan, terdiri atas tiga diagram, yaitu:
- a. Diagram aktifitas (*activity diagram*),
 - b. Diagram status mesin (*state machine diagram*),
 - c. Diagram *use case* (*use case diagram*).
3. Diagram interaksi (*interaction diagram*), merupakan bagian dari diagram tingkah laku, menunjukkan aliran kontrol dan data diantara proses-proses yang ada pada sistem yang dimodelkan, terdiri atas empat diagram, yaitu:
- a. Diagram komunikasi (*communication diagram*),
 - b. Diagram ikhtisar interaksi (*interaction overview diagram*) pada UML 2.0,
 - c. Diagram urutan (*sequence diagram*),
 - d. Diagram pewaktuan UML (*UML timing diagram*) pada UML 2.0.

2.3 Diagram *use case*

Diagram *use case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem (artikel.webgaul.com). Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya.

Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Diagram *use case* dapat sangat membantu bila kita sedang menyusun persyaratan sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang kasus pengujian untuk semua corak (*feature*) yang ada pada sistem.

Sebuah *use case* dapat meliputi (*include*) fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang ditampung akan dipanggil setiap kali *use case* yang meliputi dieksekusi secara normal. Sebuah *use case* dapat ditampung oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang umum (*common*).

Sebuah *use case* juga dapat meluas (*extend*) *use case* lain dengan tingkah lakunya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Tabel 2.1 menjelaskan empat komponen diagram *use case* yaitu:

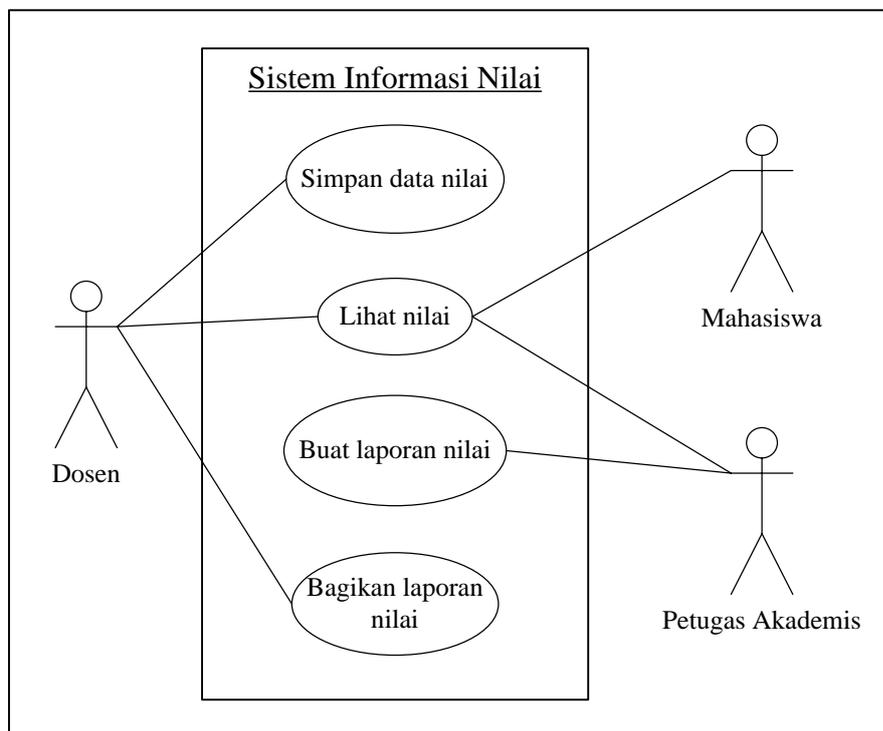
Tabel 2.1 Notasi diagram *use case* (Djajono, 2005: 51-57)

Notasi	Nama	Keterangan
	Aktor	Aktor adalah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Aktor diberi nama sesuai dengan tugasnya ketika menggunakan sistem. Jangan memberi nama aktor dengan nama orang. Aktor dapat berupa sistem lain di luar sistem yang dikembangkan
	Sistem	Sistem berupa ruang lingkup aplikasi yang dikembangkan.
	<i>Use case</i>	<i>Use case</i> adalah kegiatan atau fungsi yang dilakukan oleh aktor, diberi nama sesuai fungsi yang dilakukan oleh aktor.
	Hubungan (<i>relationship</i>)	Hubungan yang dilakukan oleh aktor dengan <i>use case</i> yang dilakukan oleh aktor tersebut.

Langkah-langkah membuat diagram *use case*:

1. Identifikasi semua aktor,
2. Identifikasi semua *use case*,
3. Urutkan prioritas *use case*,
4. Rincilah setiap *use case*,
5. Identifikasi adanya generalisasi tiap *use case*,
6. Identifikasi hubungan *include*,
7. Identifikasi hubungan *extend*,
8. Gambarkan diagram *use case*.

Contoh dari diagram *use case* digambarkan pada Gambar 2.3 berikut:



Gambar 2.3 Contoh diagram *use case* (Djajono, 2005: 54)

Dari Gambar 2.3, terdapat aplikasi Sistem Informasi Nilai yang dipergunakan oleh tiga orang aktor, yaitu Dosen, Mahasiswa, dan Petugas Akademis. Dosen dapat menyimpan data nilai, melihat nilai dan membagikan laporan nilai. Mahasiswa dapat melihat nilai. Petugas Akademis dapat melihat nilai dan membuat laporan nilai.

2.3.1 Diagram Aktifitas

Diagram aktifitas (*activity diagram*) menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir (artikel.webgaul.com). Diagram aktifitas juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Diagram aktifitas merupakan tahapan (*state*) diagram khusus, di mana sebagian besar tahapan adalah aksi dan sebagian besar transisi dipicu (*trigger*) oleh selesainya status sebelumnya (pemrosesan internal atau *internal processing*). Oleh karena itu diagram aktifitas tidak menggambarkan tingkah laku internal (*behaviour internal*) sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktifitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktifitas.

Sama seperti tahapan, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Keputusan (*decision*) digunakan untuk menggambarkan tingkah laku pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* atau transisi menyebar dan *join* atau transisi menyatu) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Diagram aktifitas dapat dibagi menjadi beberapa object garis batas (*swimlane*) untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Beberapa alasan membuat diagram aktifitas adalah sebagai berikut:

1. Menganalisis diagram *use case* lebih rinci,
2. Mengidentifikasi kondisi sebelum dan sesudah suatu *use case* (*pre and post condition*),
3. Menemukan *use case* baru yang tersembunyi.

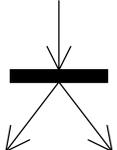
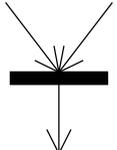
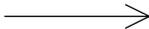
Cara untuk membuat diagram aktifitas adalah:

1. Identifikasi *use case*,

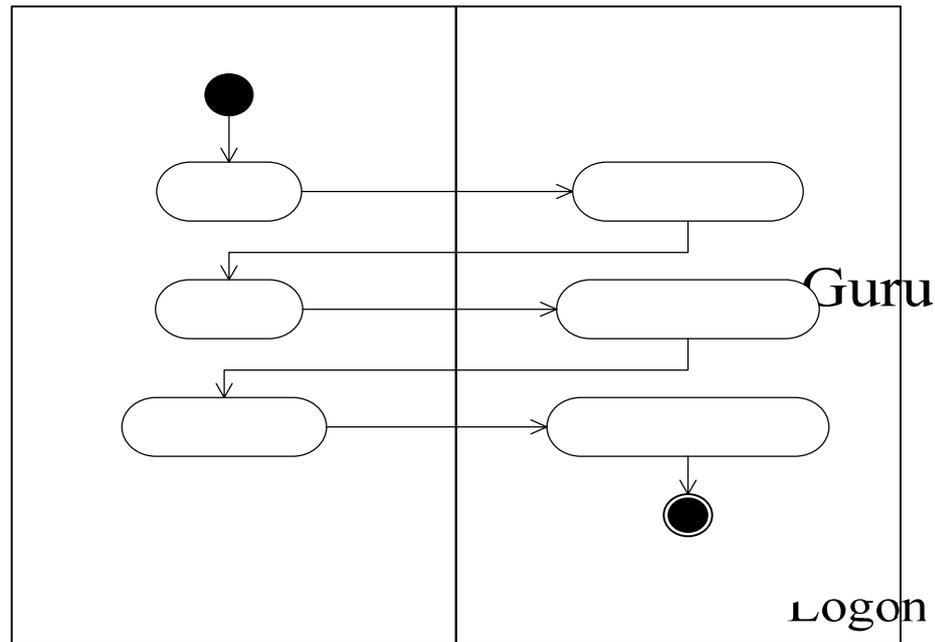
2. Pemodelan awal untuk setiap *use case*,
3. Pemodelan alternatif untuk setiap *use case*,
4. Menambahkan garis *swimlane*,
5. Refine aktivitas level akhir ke dalam diagram aktifitas.

Notasi diagram aktifitas dapat digambarkan pada Tabel 2.9 berikut:

Tabel 2.2 Notasi diagram aktifitas (Djajono, 2005: 85-93)

Notasi	Nama	Keterangan
	Action State	Kegiatan atau proses yang dilakukan.
	Initial state	Terminasi awal proses dilakukan.
	Final state	Terminasi akhir proses yang dilakukan.
	Transition (fork)	Transisi satu proses ke dua atau lebih proses berikutnya.
	Transition (join)	Transisi dua atau lebih proses ke satu proses berikutnya.
	Swimlane	Batasan proses yang dilakukan oleh aktor maupun oleh aplikasi.
	Control Flow	Alur kontrol dari satu proses ke proses lainnya.
	Decision	Percabangan proses pada kondisi tertentu.

Contoh dari diagram aktifitas dapat digambarkan pada Gambar 2.4 berikut:



Gambar 2.4 Contoh diagram aktifitas (Djajono, 2005: 88)

Dari Gambar 2.4, terdapat dua *swimlane* yaitu Guru dan antarmuka situs web.

Guru akan melakukan logon pada antarmuka situs web. Antarmuka situs web akan melakukan validasi pengguna. Setelah validasi, antarmuka situs web akan menampilkan daftar pilihan siswa. Guru akan memilih salah satu siswa dan antarmuka situs web akan menampilkan info siswa. Guru melakukan perubahan pada info siswa dan antarmuka situs web memastikan info pengguna dan aktifitas selesai dengan disimpannya nilai (data) pada basis data.

2.4 Konsep Basis Data

Menurut Fathansyah (2001: 3) basis data dapat didefinisikan dalam sejumlah sudut pandang, seperti:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan yang tidak perlu.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Perancangan basis data diperlukan, agar kita bisa memiliki basis data yang kompak dan efisien. Salah satu caranya dengan membuat Diagram Keterhubungan Entitas.

Diagram Keterhubungan Entitas atau Entity Relationship Diagram (ERD) menurut pendapat Fathansyah (2001: 70) digunakan untuk mewakili seluruh fakta dari dunia nyata yang kita tinjau dengan menggunakan notasi-notasi, lihat Tabel 2.4. Contoh notasi-notasi dalam pembuatan ERD dapat digambarkan, yaitu:

1. Entitas

Entitas adalah objek yang mewakili sesuatu yang nyata. Contoh: Mahasiswa.

2. Atribut

Atribut merupakan Gambaran ciri-ciri entitas. Contoh: nim, nama, alamat.

3. Relasi

Relasi adalah hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Contoh: *nama_mhs*="Edwin Mahendra" dengan *nim*="01501-024" mempunyai relasi dengan sebuah entitas matakuliah dengan *kode_kul*="IF1220".

4. Tipe indikator (*indicator type*)

Tipe indikator terdiri atas:

- a. Tipe indikator objek asosiatif (*indicator type associative object*), berfungsi sebagai objek dan suatu hubungan,

- b. Tipe indikator super (*indicator type supertype*), terdiri dari suatu objek dan satu subkategori atau lebih yang dihubungkan dengan satu hubungan yang tidak bernama.

5. Perbandingan kardinalitas

Perbandingan kardinalitas (*cardinality ratio*) menjelaskan hubungan batasan jumlah keterhubungan suatu entitas dengan entitas lainnya atau banyaknya entitas yang bersesuaian dengan entitas yang lain melalui suatu hubungan. Jenis Perbandingan kardinalitas:

- a. Satu ke satu atau *one to one* (1 : 1), yaitu hubungan satu entitas dengan satu entitas.
- b. Satu ke banyak atau *one to many* (1 : M) dan banyak ke satu atau *many to one* (M : 1), yaitu hubungan satu entitas dengan banyak entitas atau banyak entitas dengan satu entitas.
- c. Banyak ke banyak atau *many to many* (M : N), yaitu hubungan banyak entitas dengan banyak entitas.

6. Derajat hubungan

Derajat hubungan (*relationship degree*) menyatakan jumlah entitas yang berpartisipasi di dalam suatu hubungan. Derajat hubungan terdiri atas:

- a. Derajat satu (*unary degree*) adalah derajat yang memiliki satu hubungan untuk satu entitas,
- b. Derajat dua (*binary degree*) adalah derajat yang memiliki satu hubungan untuk dua entitas,
- c. Derajat tiga (*ternary degree*) adalah derajat yang memiliki satu hubungan untuk tiga atau lebih entitas.

7. Batasan keikutsertaan

Batasan keikutsertaan (*participation constraint*) menjelaskan apakah keberadaan suatu entitas tergantung pada hubungan dengan entitas lain. Terdapat dua macam batasan keikutsertaan, yaitu:

- a. Partisipasi total (*total participation*), yaitu keberadaan suatu entitas tergantung pada hubungannya dengan entitas lain. Di dalam diagram hubungan entitas (*entity relationship diagram*) digambarkan dengan dua garis penghubung antar entitas dan hubungan.
- b. Partisipasi sebagian (*partial participation*), yaitu keberadaan suatu entitas tidak tergantung pada hubungan dengan entitas lain. Di dalam diagram hubungan entitas digambarkan dengan satu garis penghubung.

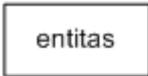
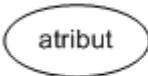
8. Representasi dari set entitas

Set entitas (*entity set*) direpresentasikan dalam bentuk tabel dan nama yang unik. Setiap tabel terdiri dari sejumlah kolom. Masing-masing kolom diberi nama yang unik pula. Set entitas *set* terbagi atas:

- a. Set entitas kuat (*strong entity set*), adalah set entitas yang satu atau lebih atributnya digunakan oleh set entitas lain sebagai kunci (*key*), digambarkan dengan empat persegi panjang.
- b. Set entitas lemah (*weak entity set*), adalah set entitas yang tergantung (*dependent*) terhadap set entitas kuat. Keberadaan entitas tersebut tergantung keberadaan entitas lain. Entitas lain tersebut disebut pemilik identifikasi (*identifying owner*) dan hubungannya disebut hubungan identifikasi (*identifying relationship*). Entitas lemah selalu mempunyai

partisipasi total dengan pemilik identifikasi. Entitas lemah digambarkan dengan persegi panjang bertumpuk.

Tabel 2.4 Notasi-notasi diagram keterhubungan entitas

Notasi	Keterangan
	Menyatakan himpunan entitas
	Mewakilkkan atribut
	Menyatakan himpunan relasi
	Menghubungkan antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut

Selain notasi-notasi di Tabel 2.3, terdapat notasi lain yang berdasarkan derajat relasi minimum dan maksimum yang dapat dilihat pada Tabel 2.5.

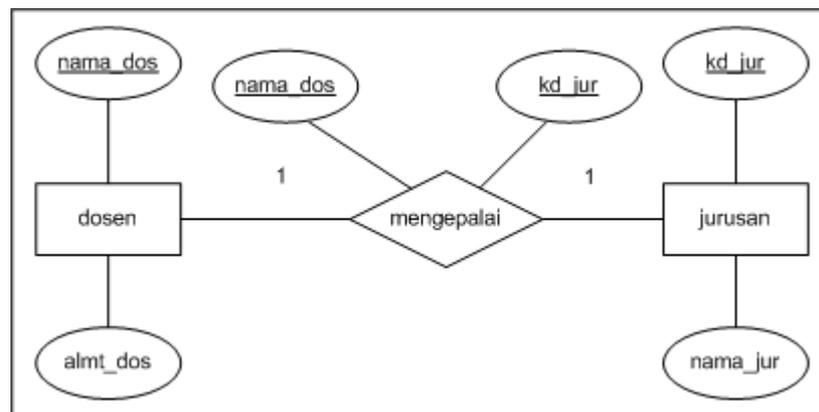
Tabel 2.5 Notasi-notasi lain diagram keterhubungan entitas

Notasi	Derajat Relasi Minimum-Maksimum
 atau 	(0, N)
 atau 	(1, N)
 atau 	(1, 1)
 atau 	(0, 1)

Selain menggunakan notasi, keterhubungan antar entitas juga menggunakan kardinalitas relasi seperti:

a. Relasi satu-ke-satu

Relasi ini menyatakan bahwa setiap entitas mempunyai hubungan paling banyak satu. Pada Gambar 2.6, himpunan entitas dosen dan himpunan entitas jurusan masing-masing memiliki 2 buah atribut. Pada himpunan relasi mengepalai terdapat 2 atribut yang secara bersama-sama berfungsi juga sebagai atribut kunci pada himpunan relasi tersebut.



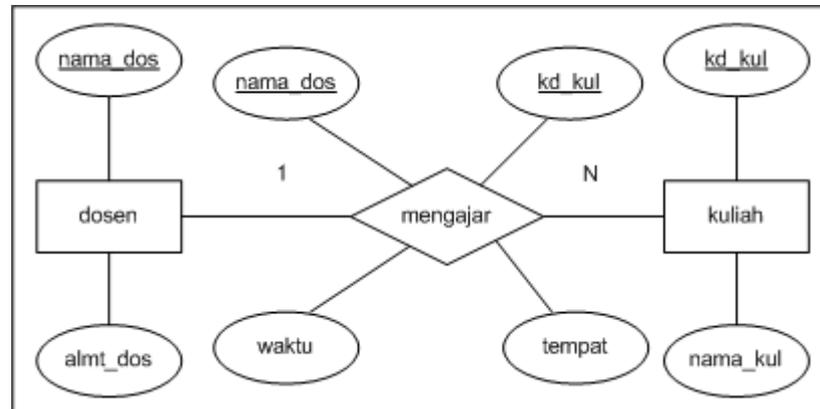
Gambar 2.6 Contoh ERD untuk satu ke satu

Karena kedua atribut tersebut berasal dari atribut kunci masing-masing himpunan entitas yang dihubungkannya, maka keduanya digolongkan sebagai *foreign-key*. Adanya kedua atribut kunci tersebut akan menunjukkan dosen mana yang mengepalai suatu jurusan, atau jurusan mana yang dikepalai seorang dosen.

b. Relasi satu-ke-banyak

Relasi ini menyatakan bahwa satu entitas mempunyai hubungan dengan banyak entitas. Pada Gambar 2.7 terdapat himpunan entitas dosen, himpunan entitas kuliah, dan himpunan entitas mengajar. Pada relasi ini setiap dosen dapat mengajar lebih dari satu mata kuliah, sedang setiap mata kuliah diajar hanya oleh paling banyak satu orang dosen. *Foreign-key* dari himpunan relasi mengajar

adalah nama_dos dan kd_kul, yang masing-masing berasal dari himpunan entitas dosen dan himpunan entitas kuliah.

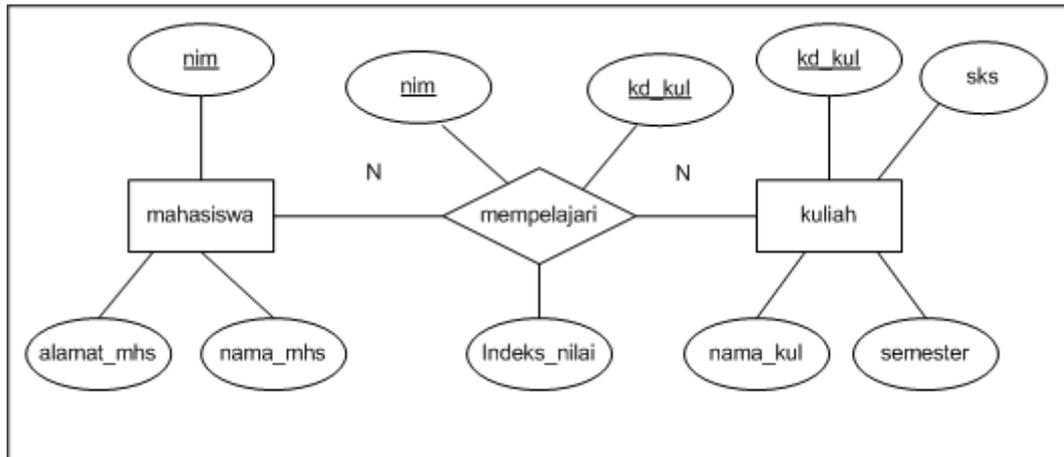


Gambar 2.7 Contoh ERD untuk satu ke banyak

Pada himpunan relasi mengajar terdapat 2 atribut tambahan yang tidak berasal dari salah satu himpunan entitas yang dihubungkan. Hal ini memang dimungkinkan dan bahkan umum terjadi. Dengan adanya keempat atribut tersebut pada himpunan relasi mengajar, maka dapat kita ketahui jadwal pelaksanaan setiap mata kuliah beserta dosen yang mengajarkannya.

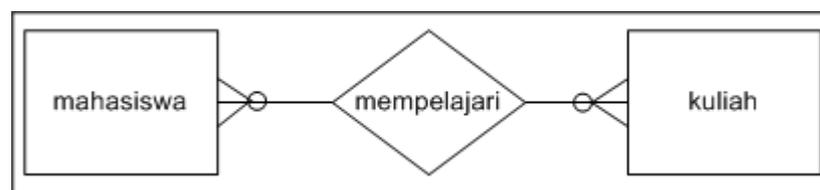
c. Relasi banyak-ke-banyak

Relasi ini menyatakan bahwa banyak entitas dapat berhubungan dengan banyak entitas. Pada Gambar 2.8 terdapat himpunan entitas mahasiswa, himpunan entitas kuliah, dan himpunan relasi mempelajari.



Gambar 2.8 Contoh ERD untuk banyak ke banyak

Pada relasi ini, setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah. Demikian juga sebaliknya, setiap mata kuliah dapat dipelajari oleh lebih dari satu orang mahasiswa. Keberadaan himpunan relasi mempelajari akan memiliki dua fungsi yaitu untuk menunjukkan mata kuliah manasaja yang diambil oleh seorang mahasiswa dan indeks nilai yang diperoleh seorang mahasiswa untuk mata kuliah tertentu. Gambar 2.9 merupakan ilustrasi dengan notasi lain berdasarkan derajat relasi minimum. Seorang mahasiswa dapat mempelajari banyak mata kuliah sekaligus juga tidak mempelajari kuliah satupun.



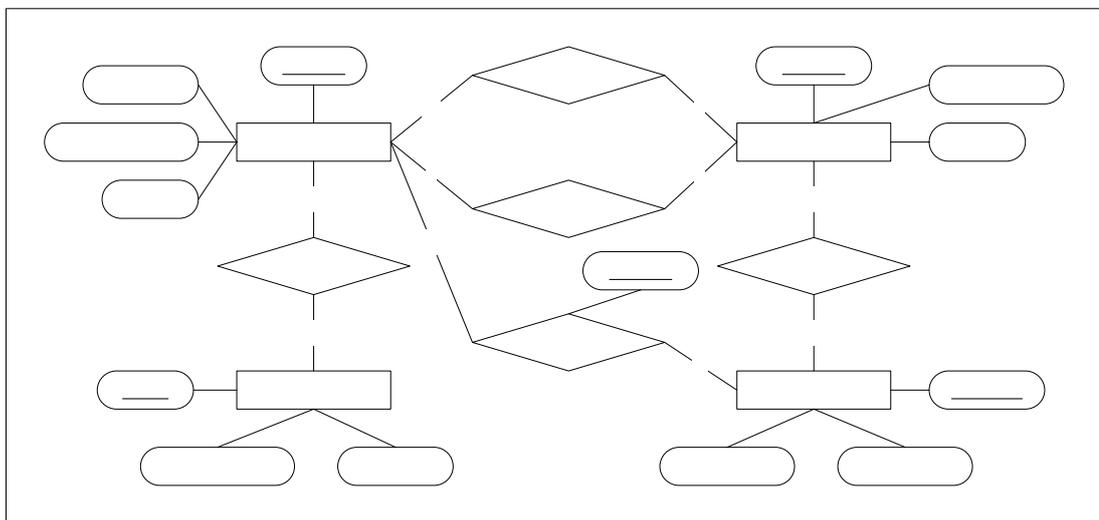
Gambar 2.9 Contoh ERD dengan notasi lain

Derajat relasi minimum untuk mahasiswa adalah 0,N. Setiap mata kuliah dapat diikuti oleh banyak mahasiswa, tapi bisa saja mata kuliah tidak diikuti oleh satupun mahasiswa. Derajat relasi minimum untuk kuliah adalah 0,N.

Langkah-langkah membuat diagram hubungan entitas:

1. Tentukan entitas-entitas yang diperlukan,
2. Tentukan hubungan antar entitas-entitas,
3. Tentukan rasio kepentingan dan *participation constraint*,
4. Tentukan atribut-atribut yang diperlukan dari tiap entitas,
5. Tentukan kunci di antara atribut-atribut,
6. Hindari penamaan entitas, hubungan, dan atribut yang sama.

Contoh dari diagram hubungan entitas digambarkan pada Gambar 2.10 berikut.



Gambar 2.10 Contoh diagram hubungan entitas (Marlinda, 2004: 24)

Dari Gambar 2.10 terdapat empat buah entitas yaitu Pegawai, Departemen, Proyek, dan Tanggungan. Entitas Pegawai terdiri atas empat buah atribut yaitu no_peg, nama_peg, jenis_kelamin, dan gaji. *Primary key* dari entitas Pegawai adalah no_peg. Entitas Departemen memiliki tiga atribut yaitu no_dep, nama_dep, dan lokasi. *Primary key* dari entitas Departemen adalah no_dep. Entitas Proyek terdiri atas atribut no_pryk, nama_pryk, dan lokasi_pryk. *Primary key* entitas Proyek adalah no_pryk. Entitas keempat adalah entitas Tanggungan yang terdiri dari atribut nama,

gaji

1

Mempunyai

M

1

M

`jenis_kelamin`, dan hubungan dengan nama sebagai *primary key* entitas ini.

Hubungan dari empat buah entitas tersebut adalah sebagai berikut:

1. Banyak pegawai menjadi anggota satu departemen (M : 1),
2. Satu pegawai memimipin satu departemen (1 : 1),
3. Banyak pegawai bekerja pada banyak proyek (M : N),
4. Satu departemen membawahi banyak proyek (1 : M), dan
5. Satu pegawai mempunyai banyak tanggungan (1 : M).

2.5 Structured Query Language

Menurut Fathansyah (2001: 195) Structured Query Language (SQL) dapat diartikan sebagai bahasa standar yang digunakan untuk berinteraksi dengan basis data relasional. SQL menawarkan beberapa alat untuk berinteraksi dengan basis data, yaitu:

1. Pendefinisian data terstruktur: SQL dapat menentukan konstruksi yang bermacam-macam yang digunakan basis data untuk menyimpan data.
 - a. Untuk membuat tabel menggunakan **create table**.
 Contoh : `create table mahasiswa (nim char(6), nama_mhs varchar(60), alamat_mhs varchar(60) tgl_lahir date).`
 - b. Untuk menghapus tabel menggunakan **drop table**.
 Contoh: `drop table mahasiswa.`
 - c. Untuk menambahkan atribut baru menggunakan **alter table**.
 Contoh: `alter table mahasiswa add ip numeric(5,2).`
2. Pencarian data: SQL dapat mendapatkan data dalam basis data dan ditampilkan dalam bentuk yang dapat dibaca.

Contoh: **select** nim, nama_mhs **from** mahasiswa.

3. Manipulasi data: SQL dapat memasukkan, memperbaharui, dan menghapus data.

a. Memasukkan data.

Contoh: **insert into** mahasiswa(nim, nama_mhs, alamat_mhs, tgl_lahir) **values** ('01501030', 'fery setiawan', 'Jl. Kp.Kelapa No. 63 Tangerang', '26/03/1983').

b. Memperbaharui data

Contoh: **update** mahasiswa **set** alamat='Jl. Beringin 3 No.63 Tangerang' **where** nim='01501030'.

c. Menghapus data

Contoh: **delete from** mahasiswa **where** nim='01501030'.

4. Pengendalian akses data: SQL memungkinkan untuk mengkoordinir kelebihan pengendalian pengguna-khusus yang menampilkan, memasukkan, memperbaharui, dan menghapus data.

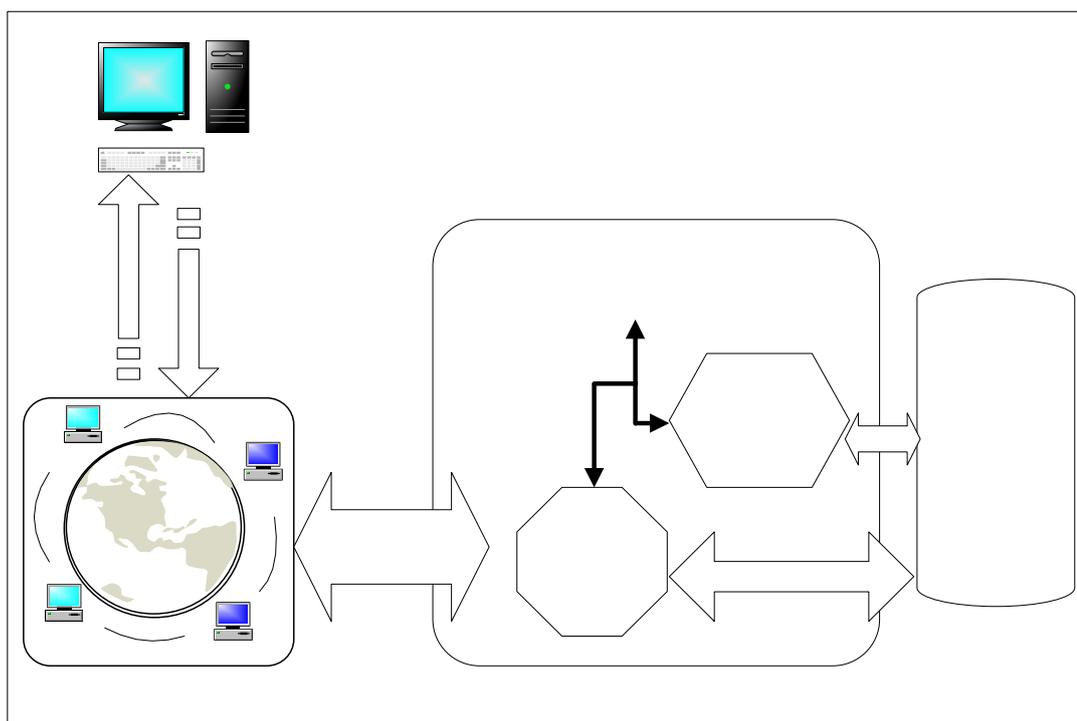
5. Kesatuan data: SQL mencegah korupsi data yang akan menyebabkan masalah seperti kegagalan sistem.

2.6 Pengertian Personal Home Page (PHP)

PHP dibuat pada tahun 1995, ketika Rasmus Lerdorf mengembangkan skrip Perl/CGI untuk mengetahui banyaknya pengunjung yang membaca ringkasan *online*-nya. PHP 2.0, atau PHP-FI (Personal Home Page-Form Interpreter) terbit dimana dapat merubah masukan data dalam HTML menjadi peubah-peubah simbol yang mengijinkan untuk dipakai ke sistem lain. Penulisan ulang mesin parsing, melahirkan PHP 3.0 pada tahun 1997 dan pada 22 Mei tahun 2000 diterbitkan PHP 4.0. Menurut

Jason Gilmore, William (2001, 5) PHP mempunyai enam karakteristik yang penting, yaitu: dikenal, sederhana, efisien, keamanan, fleksibel, dan gratis.

Skema dalam pembuatan aplikasi berbasis web antara HTML, PHP, dan basis data terdiri dari tiga tingkatan (*tier*) lihat Gambar 2.14. Pada Gambar 2.14, terdapat 2 komunikasi, satu dirancang untuk dapat diakses oleh manusia melalui pembaca jaringan (*web browser*), dan lainnya untuk pertukaran data aplikasi-ke-aplikasi. Antarmuka yang mengandung bentuk HTML pada komputer *client* memungkinkan pengguna untuk memasuki SQL query. Bahasa PHP digunakan untuk membangun pernyataan SQL dan mengaitkan basis data; data lalu dikeluarkan dari basis data, dan pengguna disajikan oleh tampilan struktur tabel.



Gambar 2.11 Konsep kerja PHP dan Mysql

Menurut Castagnetto dkk (1999: Bab 26: 3) tingkatan-tingkatan pada aplikasi berbasis web dijelaskan seperti di bawah ini:

Client

1. Tingkatan Pertama

Tingkatan ini terdiri dari antarmuka, diterapkan melalui bentuk HTML dengan dan tanpa dukungan dari JavaScript. Hasilnya dikembalikan dan ditampilkan sebagai tabel.

2. Tingkatan Kedua

Tingkatan Kedua terdiri dari penyedia jaringan (*web server*) Apache. Modul PHP mendukung untuk beberapa basis data belakang (*backend*) seperti mSQL dan MySQL, sebaik dengan pembangkitan gambar GIF, dan pengelolaan data XML.

3. Tingkatan Ketiga

Tingkatan ketiga terdiri dari mesin basis data itu sendiri. Di sini, digunakan basis data MySQL.

2.7 Pengujian Perangkat Lunak

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

a. Pengujian Black-Box

Pengujian Black-Box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian Black-Box memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program (Pressman, 2002: 551). Pengujian

Black-Box bukan merupakan alternatif dari teknik White-Box, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkapkan kelas kesalahan daripada metode White-Box.

Pengujian Black-Box berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan antarmuka,
3. Kesalahan dalam struktur data atau akses basis data eksternal,
4. Kesalahan kinerja,
5. Inisiasi dan kesalahan terminasi.

Pengujian Black-Box cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian Black-Box memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi.

Dengan mengaplikasikan teknik Black-Box, maka kita menarik serangkaian pengujian kasus yang memenuhi kriteria berikut ini:

1. Pengujian kasus yang mengurangi, dengan harga lebih dari satu, jumlah kasus pengujian tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggungjawabkan, dan
2. Pengujian kasus yang memberi tahu kita sesuatu mengenai kehadiran atau ketidakhadiran kelas kesalahan, daripada memberi tahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

Masalah yang dihadapi baru timbul apabila proses perhitungan gaji tersebut harus dilakukan dalam jumlah yang cukup besar, serta ditambah lagi dengan adanya keaneka-ragaman peraturan perhitungan gaji yang berlaku bagi setiap pegawai. Dalam proses kegiatan perusahaan sehari-hari, ada beberapa masalah yang dihadapi dalam pengolahan gaji yang digunakan pada saat ini, antara lain:

1. Semua proses penggajian masuk dikerjakan secara manual, akan dibutuhkan ketelitian dalam pengolahan data penggajian sangat terbatas, dan kemungkinan terjadinya kesalahan lebih besar.
2. Diperlukan waktu yang relatif lama untuk mendapatkan laporan dari proses penggajian yang dibutuhkan, sehingga hal ini akan menghambat jalannya operasi perusahaan dan kebutuhan data yang seharusnya tersedia tepat waktu.
3. Kejenuhan dalam melakukan perhitungan yang diakibatkan oleh banyaknya data gaji pegawai. Pelaksanaan perhitungan gaji kurang sistematis sehingga menimbulkan kesulitan dalam membuat laporan mengenai gaji pegawai.
4. Banyaknya permintaan rincian gaji yang akan didapatkan karyawan untuk keperluan manajemen dan karyawan itu sendiri.

Oleh karena masalah tersebut di atas, diadakanlah proses komputerisasi dalam pengolahan gaji.

Pada aplikasi sistem penggajian karyawan ini juga terdapat permasalahan terhadap keamanan masuk (*login*) data *username* dan *password*. Nilai data-data berupa *username* dan *password* baik di tingkat karyawan dan administrator manajemen disimpan menggunakan fungsi *session()* di penyedia (*server*). Jika nilai data tersebut dapat dibaca maka akan mudah untuk mengakses dengan *username* dan *password* pemilih lain. Untuk itu penulis membuat enkripsi pada nilai data yang disimpan menggunakan fungsi *session()*.

Dalam melakukan pengujian harus ada kebutuhan sistem yang didasarkan pada sudut pandang pemakai, yaitu:

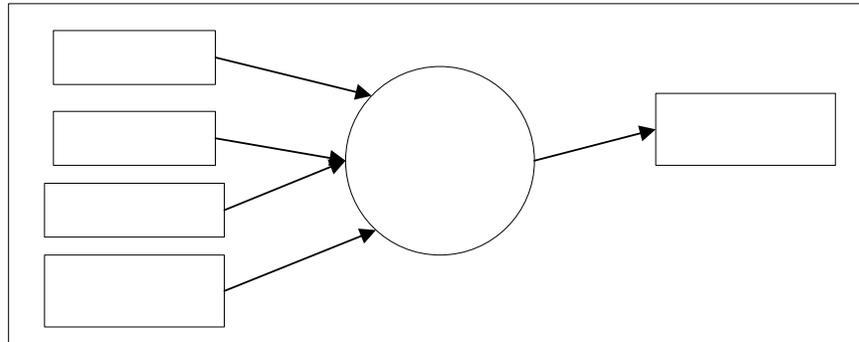
1. Antarmuka (tampilan) yang jelas dan mudah digunakan.
2. Nama hubungan (*link*) sesuai dengan proses yang dikerjakan.
3. Jumlah dari gaji beserta rinciannya dapat dilihat langsung oleh karyawan.

Tampilan setiap objek pada setiap halaman harus sesuai dengan fungsi halamannya.

3.2 Pendefinisian Kebutuhan

Secara umum Aplikasi Penggajian Karyawan berbasis WEB yang akan dikembangkan dipergunakan oleh departemen HRD (*Human Resources Development*) pada perusahaan. Aplikasi ini ini membutuhkan masukan berupa data karyawan, data elemen gaji karyawan, dan data hasil perhitungan elemen gaji yang dimiliki oleh tiap-tiap jabatan dalam perusahaan. Masukan ini akan diproses oleh aplikasi sehingga menjadi data perincian gaji yang diterima oleh karyawan dalam satu bulan. Hasil proses akhirnya akan menghasilkan laporan rincian gaji karyawan

mulai dari gaji pokok, absen dan uang lembur beserta poongan gaji yang diterima oleh karyawan. Laporan ini akan digunakan oleh departemen HRD sebagai sarana perhitungan gaji yang lebih cepat dan efisien dan karyawan sebagai perincian gaji yang diterimanya. Ilustrasi masukan, proses, dan keluaran aplikasi secara umum digambarkan pada Gambar 3.1 berikut.



Gambar 3.1 Masukan, proses, dan keluaran aplikasi secara umum

3.3 Rekayasa Sistem

Sistem Penggajian merupakan fungsi penting yang menjadi tanggungjawab Manajemen Sumber Daya Manusia. Gaji merupakan **Data Karyawan** tambahan penghasilan finansial yang diterima oleh para pegawai sebagai ganti kontribusi mereka terhadap organisasi. Sistem Penggajian adalah proses yang menentukan tingkat penggajian pegawai, memantau atau mengawasi, mengembangkan serta mengendalikan gaji pegawai. Modul Penggajian yang dibangun dalam penelitian ini mengacu pada **Elemen Gaji** aturan Sistem Penggajian Pegawai yang berlaku di Indonesia, dimana struktur gaji terdiri dari beberapa komponen, yaitu **Data nominal gaji**

1. Gaji pokok.
2. Tunjangan jabatan fungsional atau jabatan struktural,

**Data hasil
perhitungan
elemen gaji**

3. Uang absen dan uang lembur,
4. Bonus,
5. Pajak penghasilan (pph), dan
6. Potongan lain-lain (seperti jamsostek, potongan pinjaman karyawan (bila ada), dan lainnya) dan pembulatan gaji ke kelipatan seratus.

Nilai gaji pokok ditentukan oleh golongan jabatan karyawan pada perusahaan, dan status kepegawaian. Selain gaji pokok, para karyawan juga mendapatkan intensif berupa tambahan pendapatan yang berasal dari tunjangan jabatan, uang absen, uang lembur dan bonus yang diberikan perusahaan. Nilai tunjangan jabatan adalah sepuluh persen dari gaji pokok dan uang absen dan lembur adalah berdasarkan ketetapan perusahaan. Selain gaji yang didapat oleh karyawan ada juga kewajiban yang harus di penuhi oleh karyawan yang berupa potongan gaji. Komponen dalam potongan gaji adalah PPh dengan perhitungan sebagai berikut:

$$\text{PPh} = (\text{penghasilan} - (5\% \times \text{penghasilan}) - (4,75\% \times (\text{gaji pokok} + \text{tunjangan jabatan})) - \text{PTKP}) \times 10\%$$

dimana nilai penghasilan dari formula di atas adalah penjumlahan dari gaji pokok, tunjangan jabatan, uang absen dan uang lembur.

Selain informasi gaji, keluaran dari modul penggajian ini adalah laporan bulanan penggajian yang digunakan untuk pengajuan gaji pegawai kepada divisi HRD. Laporan bulanan ini terdiri dari data karyawan, absen karyawan dalam satu bulan dan jam lembur yang diambil karyawan dalam satu bulan jumlah gaji kotor karyawan (gaji sebelum pemotongan), dan jumlah gaji yang diterima karyawan dalam satu bulan. Selain laporan bulanan untuk divisi HRD, modul penggajian yang dibuat juga harus dapat membuat daftar rincian gaji per pegawai untuk keperluan manajemen perusahaan.

4. Antarmuka sistem, terdiri atas struktur menu dan antarmuka aplikasi

3.4.1 Basis Data

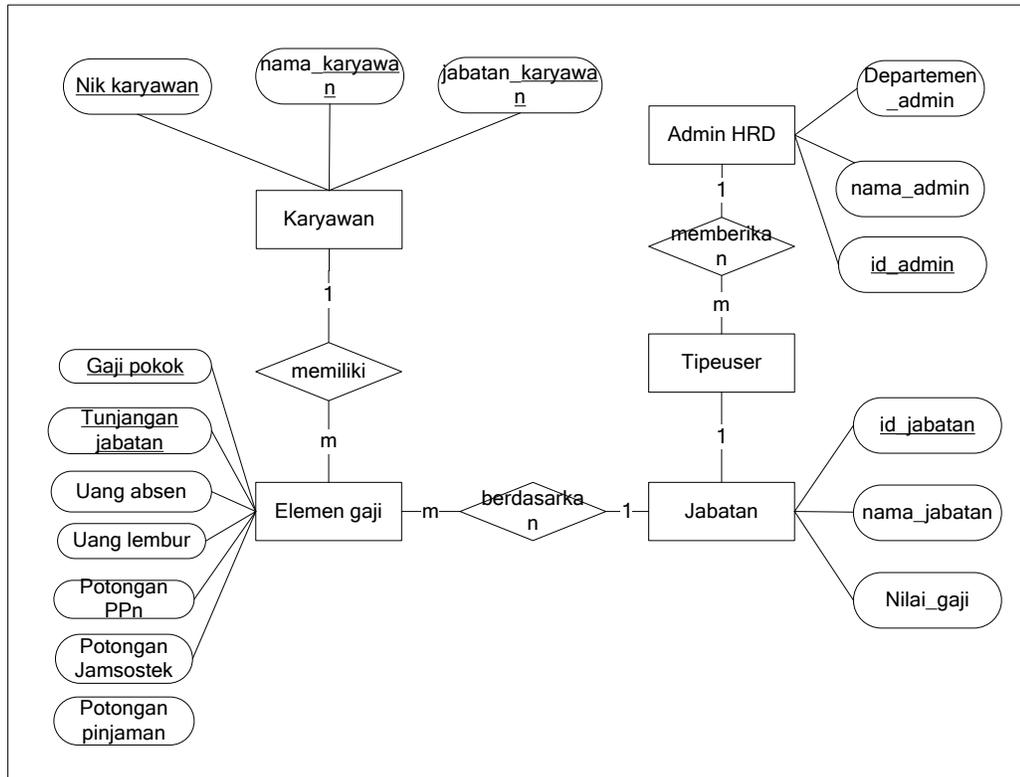
Analisis pada segi basis data aplikasi yang dikembangkan akan ditinjau dari diagram hubungan entitas dan struktur basis data.

3.4.1.1 Diagram Hubungan Entitas

Aplikasi yang dikembangkan terdiri atas empat tabel yang saling berhubungan, yaitu tabel login, tipeuser, report, dan produk. Empat tabel ini memiliki hubungan kardinalitas sebagai berikut:

1. Satu karyawan memiliki banyak elemen gaji,
2. Satu admin HRD memberikan banyak tipe user, dan
3. Satu tipe user memiliki satu jabatan.

Diagram hubungan entitas dari aplikasi digambarkan pada Gambar 3.3.



Gambar 3.3 Diagram hubungan entitas aplikasi

3.4.1.2 Struktur Basis Data

Tujuh tabel entitas pada basis data aplikasi strukturnya dapat dijelaskan sebagai berikut:

1. Tabel data administrator

Pada tabel *administrator*, penulis membuat 8 field, yaitu: field nik, field nama, field departemen, field jabatan, field *password*, field *tipeuser* dan field foto. Field nik mempunyai panjang 4 karakter dan merupakan *primary key*. Hal ini didasarkan pada jumlah perkiraan nomor identitas karyawan yang akan diberikan oleh perusahaan. Field nama mempunyai panjang 20 karakter, karakter nama yang menjadi dasar penulis adalah Edwin Mahendra yang mempunyai panjang

14 karakter sedangkan 6 karakter sisa merupakan pengalokasian karakter tambahan bila ternyata ada yang lebih dari 14 karakter. Field departemen mempunyai panjang karakter 15. Panjang 15 karakter ini diambil berdasarkan nama jenis departemen dan jabatan di perusahaan yang penulis gunakan untuk membuat aplikasi. Field *password* mempunyai panjang 8 karakter yang berdasarkan oleh kata yang dimengerti oleh penulis. field *tipesuer* mempunyai panjang 10 karakter yang berdasarkan jenis pengguna aplikasi yang ada pada perusahaan dan pada field foto menggunakan 16 yang mempunyai bentuk `../foto/nama_gambar`. Nama gambar akan dialokasikan sebanyak 8 karakter.

Tabel 3.1 Tabel administrator

Atribut	Tipe	Panjang	Keterangan
Nik	Varchar	4	Nomor identitas pengguna
Nama	Varchar	20	Nama pengguna
Departemen	Varchar	15	Departemen pengguna
Jabatan	Varchar	15	Jabatan pengguna dalam perusahaan
Password	Varchar	8	Kata kunci pengguna
Tipeuser	Varchar	10	jenis user dalam akses pada aplikasi
Foto	Varchar	16	Foto identitas pengguna

2. Tabel *tipeuser*

Pada tabel *tipeuser*, penulis membuat dua field, yaitu: field *id* dan field *tipeuser*. Field *id* mempunyai panjang karakter 4 dan field *tipeuser* mempunyai panjang 10 karakter. Tabel ini berfungsi untuk membedakan jenis pengguna dan hak akses yang dimilikinya.

Tabel 3.2 Tabel *tipeuser*

Atribut	Tipe	Panjang	Keterangan
Id	Varchar	4	Nomor identitas pengguna
Tipeuser	Varchar	10	Tipe pengguna

3. Tabel elemen gaji

Pada tabel elemen gaji, penulis membuat enam field, yaitu: field id, field jabatan, field gapok, field tjng_jbtn, field absen dan field lbrperjam. Field id mempunyai karakter sebanyak 4 karakter, berdasarkan perkiraan jumlah karyawan dalam perusahaan. Field jabatan mempunyai panjang 15 karakter. Field gapok adalah jumlah nilai gaji pokok karyawan berdasarkan jabatannya. Field tjng_jbtn adalah field yang berisi nominal jumlah tunjangan jabatan yang diterima karyawan, field ini menggunakan tipe float. Field absen adalah field yang berisi nominal jumlah uang absen yang diterima karyawan, field ini menggunakan tipe float dan field lbrperjam adalah field yang berisi nominal jumlah uang lembur yang diterima karyawan, field ini menggunakan tipe integer dengan panjang 5 karakter.

Tabel 3.3 Tabel elemen gaji

Atribut	Tipe	Panjang	Keterangan
Id	Varchar	4	Nomor identitas pengguna
Jabatan	Varchar	15	Nama pengguna
Gapok	Integer	11	Nilai gaji pokok
Tjng_Jbtn	Float		Tunjangan berdasarkan jabatan karyawan
Absen	Float		Jumlah absen yang diambil karyawan
Lbrperjam	Integer	5	Jumlah Jam lembur yang diambil karyawan

4. Tabel gaji

Tabel gaji dibuat oleh penulis sebagai koneksi untuk tampilan hasil perhitungan gaji bagi karyawan. Pada tabel gaji, penulis membuat enam field, yaitu: field nik, field nama, field gapok, field tjng_jbtn, field lbrperjam dan field absen. Field nik mempunyai karakter sebanyak 4 karakter, berdasarkan Hal ini didasarkan pada perkiraan jumlah nomor identitas karyawan yang akan diberikan oleh perusahaan. Field nama mempunyai panjang 20 karakter. Field gapok adalah jumlah nilai gaji pokok karyawan berdasarkan jabatannya. Field tjng_jbtn adalah field yang berisi nominal jumlah tunjangan jabatan yang diterima karyawan, field ini menggunakan tipe float. field lbrperjam adalah field yang berisi nominal jumlah uang lembur yang diterima karyawan, field ini menggunakan tipe float dan Field absen adalah field yang berisi nominal jumlah uang absen yang diterima karyawan, field ini menggunakan tipe float.

Tabel 3.4 Tabel gaji

Atribut	Tipe	Panjang	Keterangan
Nik	Varchar	4	Nomor identitas karyawan
Nama	Varchar	20	Nama pengguna
Gapok	Integer	15	Nilai gaji pokok
Tjng_Jbtn	Float		Tunjangan berdasarkan jabatan karyawan
Lbrperjam	Float		Jumlah Jam lembur yang diambil karyawan
Absen	Float		Jumlah absen yang diambil karyawan

5. Tabel data hadir.

Tabel data hadir dibuat penulis untuk melaksanakan penjumlahan absen karyawan. Pada tabel hadir, penulis membuat tiga field, yaitu: field nik, field nama dan field hadir. Field nik akan menjadi *primary key* yang menjadi dasar

ketika terjadi pemanggilan salah satu data absen karyawan. Panjang field nik adalah 4 karakter dimana perkiraan jumlah nomor identitas karyawan yang akan diberikan oleh perusahaan. Field nama mempunyai panjang 20 karakter dan field hadir mempunyai panjang 3 karakter didasarkan jumlah hari dalam satu bulan.

Tabel 3.5 Tabel data hadir

Atribut	Tipe	Panjang	Keterangan
Nik	Varchar	4	Nomor identitas pengguna
Nama	Varchar	20	Nama pengguna
Hadir	Integer	3	Jumlah absen karyawan

6. Tabel data jabatan.

Pada tabel jabatan, penulis membuat dua field, yaitu: field id dan jabatan field. Panjang field id adalah 4 karakter dimana perkiraan jumlah jenis jabatan dalam perusahaan. Field jabatan mempunyai panjang 15 karakter berdasarkan jenis jabatan yang penulis buat dalam aplikasi.

Tabel 3.6 Tabel data jabatan

Atribut	Tipe	Panjang	Keterangan
Id	char	4	Nomor identitas pengguna
Jabatan	Varchar	15	Jabatan pengguna

7. Tabel data lembur.

Tabel data lembur dibuat penulis untuk melaksanakan penjumlahan lembur karyawan. Pada tabel lembur, penulis membuat tiga field, yaitu: field nik, field nama dan field lembur. Field nik akan menjadi *primary key* yang menjadi dasar ketika terjadi pemanggilan salah satu data lembur karyawan. Panjang field nik adalah 4 karakter dimana perkiraan jumlah nomor identitas karyawan yang akan

diberikan oleh perusahaan. Field nama mempunyai panjang 20 karakter dan field lembur mempunyai panjang 3 karakter didasarkan perkiraan jumlah jam lembur dalam satu bulan.

Tabel 3.7 Tabel data lembur

Atribut	Tipe	Panjang	Keterangan
Nik	Varchar	4	Nomor identitas pengguna
Nama	Varchar	20	Nama pengguna
Lembur	Integer	3	Jumlah lembur karyawan

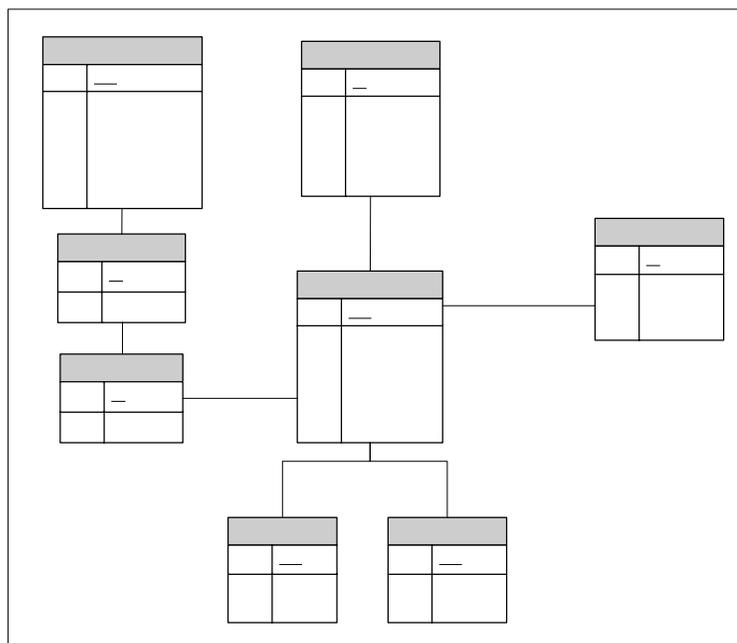
8. Tabel data potongan gaji.

Tabel potongan gaji dibuat penulis untuk melaksanakan penjumlahan potongan gaji untuk menghasilkan jumlah gaji bersih yang diterima karyawan. Tabel potongan gaji, penulis membuat empat field, yaitu: field id, field ptg_ppn, ptg_jms dan ptg_pjm. Field id akan menjadi *primary key* yang menjadi dasar ketika terjadi pemanggilan salah satu data potongan gaji karyawan. Panjang field id adalah 4 karakter dimana perkiraan jumlah nomor identitas karyawan yang akan diberikan oleh perusahaan. Field ptg_ppn adalah data dari pajak penfapatan karyawan yang mempunyai panjang 6 karakter, field ptg_jms merupakan nilai potongan jamsostek karyawan yang mempunyai panjang 6 karakter dan field ptg_pjm memiliki panjang 7 karakter yang merupakan jumlah potongan pinjaman karyawan.

Tabel 3.8 Tabel potongan gaji

Atribut	Tipe	Panjang	Keterangan
Id	Varchar	4	Nomor identitas pengguna
Ptg_ppn	Integer	6	Nilai potongan pajak penghasilan karyawan
Ptg_jms	Integer	6	Nilai potongan jamsostek karyawan
Ptg_pjm	Integer	7	Nilai potongan pinjaman kayawan

Struktur dan hubungan antar delapan entitas dapat digambarkan pada Gambar 3.4.



Gambar 3.4 Struktur entitas tabel basis data aplikasi

3.4.2 Diagram Alir Data (DAD)

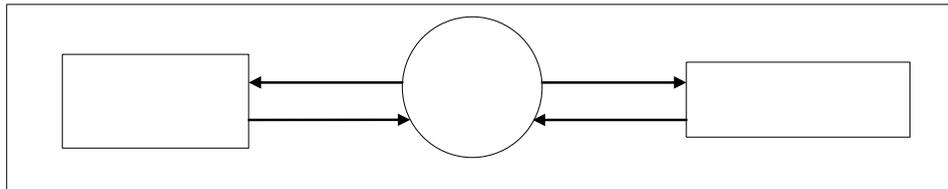
Alir data yang terjadi pada aplikasi penggajian karyawan dapat digambarkan pada diagram alir data dibawah ini.

Administrator
 PK nik
 nama
 departemen
 jabatan
 password
 tipeuser
 foto
 tipeuser
 PK id
 tipeuser
 Jabatan
 PK id

PK
 PK
 PK

1. Diagram konteks

Hubungan antara aplikasi proteksi keamanan dengan lingkungan luar digambarkan pada Gambar 3.5.



Gambar 3.5 Diagram konteks

Pada Gambar 3.5 terdapat proses penentuan hak akses, entitas pengguna, dan entitas posisi dalam perusahaan. Entitas pengguna dan entitas posisi dalam perusahaan memberikan masukan berupa perintah untuk menjadi masukan bagi proses penentuan hak akses. Proses hak akses mempunyai keluaran berupa tampilan informasi berdasarkan entitas pengguna dan entitas posisi dalam perusahaan.

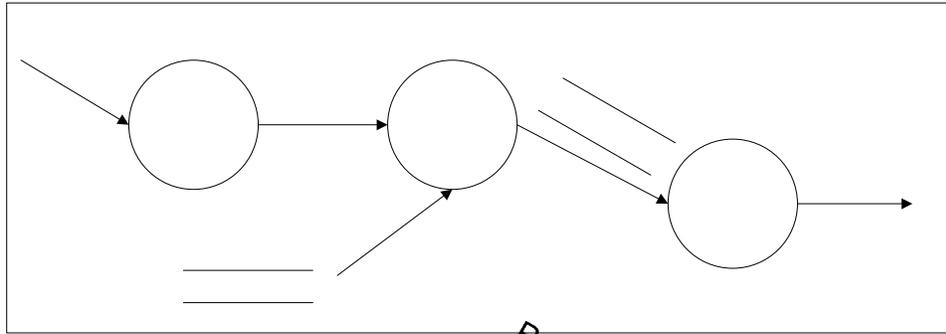
Perintah

Pegguna

2. Diagram tingkat satu

Tampilan Informa

Pada tingkat satu ini penulis membahas tingkat satu untuk karyawan dan posisi karyawan pada perusahaan. Pada Gambar 3.5, data perintah mengalir ke dalam proses interaksi dengan karyawan dan menghasilkan 1 jalur aksi, yaitu masuk ke tampilan menu login yang akan menentukan tipe user pengguna aplikasi (aplikasi terdiri dari dua tipe user yaitu karyawan dan administrator HRD). Aliran informasi nama dan password akan menuju ke proses baca nama dan password dan akan mengeluarkan identifikasi pesan sah yang akan menjadi masukan bagi proses tampilkan pesan dan status. Proses pesan dan status akan menghasilkan aliran data tampilan informasi.



Gambar 3.6 DAD Tingkat 1 karyawan

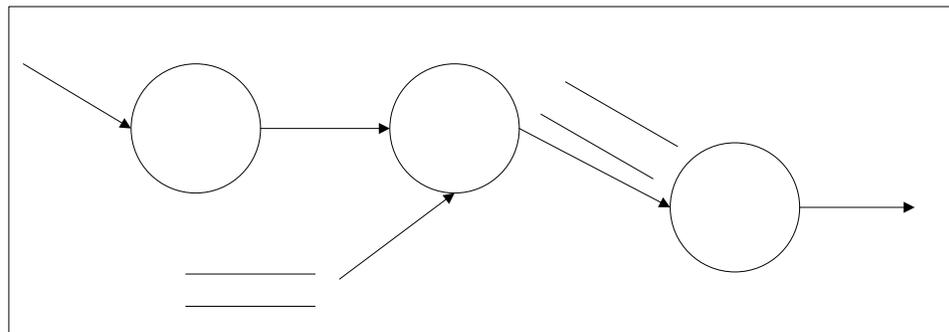
Data perintah akan menjadi masukan untuk proses dengan administrator yaitu masuk ke tampilan menu login yang akan menentukan tipe user pengguna aplikasi (aplikasi terdiri dari dua tipe user yaitu karyawan dan administrator HRD). Aliran informasi nama dan password akan menuju ke proses baca nama dan password dan akan mengeluarkan identifikasi pesan yang akan menjadi masukan bagi proses tampilkan pesan dan status. Proses pesan dan status akan menghasilkan aliran data tampilan informasi.

Interaksi dengan pengguna

Logir karyawan

Nama, password

File login karyawan

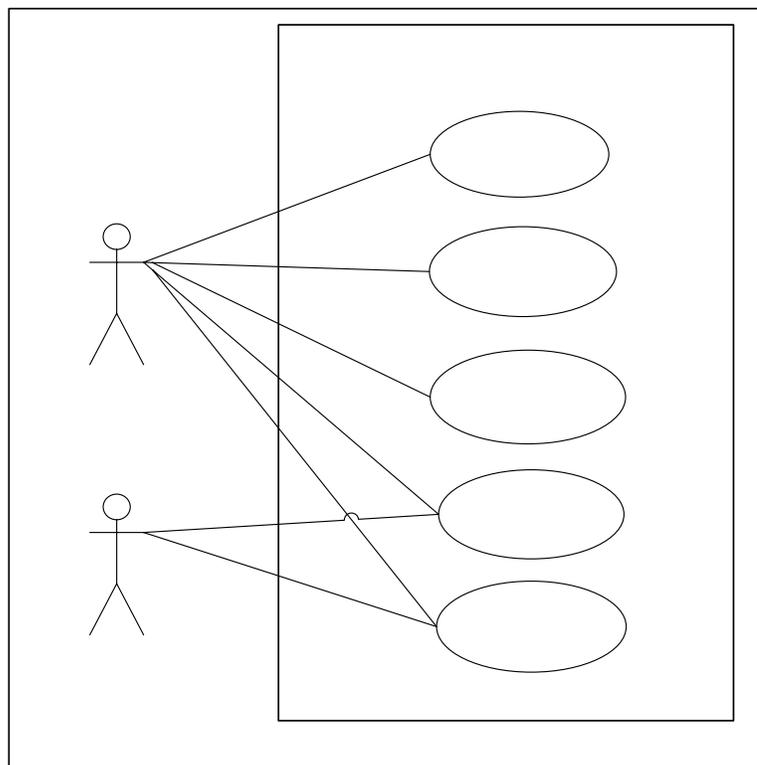


Gambar 3.7 DAD Tingkat 1 Administrator

Keluaran dari proses nama dan password adalah pesan yang berisi valid atau tidaknya *username* dan *password* pengguna.

3.5 Diagram *Use Case*

Aplikasi ini dikembangkan untuk dipergunakan oleh divisi HRD dalam mennetukan gaji karyawan dalam satu bulan. Gaji karyawan ditentukan oleh jabatan karyawan dalam perusahaan. Perhitungan gaji karyawan itu sendiri dilakukan oleh administrator yang berada pada departemen HRD (Human Resources Development) yang ditunjuk oleh perusahaan. Pemasukkan data gaji dilakukan setiap satu bulan sekali. Setelah dilakukan pemasukan data oleh adminstrator oleh admin HRD maka karyawan dapat melihat rincian gaji yang diberikan oleh perusahaan secara online pada jaringan komputer yang terdapat pada perusahaan. diagram *use case* dapat dijelaskan pada Gambar 3.8.



Gambar 3.8 Diagram *use case* aplikasi

Dari diagram *use case* aplikasi pada Gambar 3.7 dapat dijelaskan lagi dari ilustrasi skenario per *use case* sebagai berikut:

1. Nama *use case*: Update data karyawan

Aktor: Administartor HRD

Deskripsi: Kegiatan untuk memasukkan data yang dibutuhkan pada basis data aplikasi, terdiri atas tiga jenis proses memasukkan data, yaitu :

- a. Update jabatan,
- b. Update *tipe user*,
- c. Input *new user*,

Exception (pengecualian): pada jenis masukan update jabatan dan update *tipe user* data tidak akan bisa dirubah bila data karyawan belum ada pada basis data.

Skenario:

Skenario untuk proses input data update jabatan dan update *tipe user* hampir sama, sedang kan untuk proses input new user berbeda yaitu:

1. Pada proses input data update jabatan dan update *tipe user* Admin HRD memilih masukkan dan sub menu update jabatan atau *tipe user* lalu memilih data pegawai yang sudah ada dan memberika jabatan atau tipe user baru.
2. Aplikasi menampilkan antarmuka masukan data sesuai submenu yang dipilih,
3. Admin HRD melakukan proses masukan data,
 - a. Pada proses update jabatan data yang dimasukkan adalah id dan jabatan
 - b. Pada proses update *tipeuser* data yang dimasukkan adalah id dan *tipe user*
 - c. Pada proses input new user data yang dimasukkan adalah nik, nama, departemen, jabatan, password, tipeuser dan foto.
4. Pengguna menekan tombol simpan untuk menyimpan data pada basis data,

5. Aplikasi menyimpan data yang dimasukkan pada basis data.
6. menampilkan antar muka update data karyawan.

Alternatif: Nomor 5, jika input data gagal, maka data karyawan pada tampilan update data karyawan tidak berubah.

Precondition (kondisi awal): Aplikasi akan menampilkan data karyawan yang ada pada basis data.

Postcondition (kondisi akhir): Data yang dimasukkan telah tersimpan pada basis data dan data karyawan telah berubah.

2. Nama *use case*: Update elemen gaji

Aktor: Admin HRD

Deskripsi: Kegiatan merubah data yang terdapat pada basis data, jika pada data tersebut terdapat kesalahan atau pembaharuan data. Terdapat enam jenis proses pada update elemen gaji , yaitu:

- a. input id
- b. input jabatan,
- c. input gaji pokok,
- d. input tunjangan jabatan,
- e. input absen,
- f. input lembur perjam,

Exception: Perubahan tidak dapat dilakukan jika tidak terdapat data jabatan pada basis data aplikasi.

Skenario:

Skenario untuk empat jenis proses edit data adalah sama, yaitu:

1. Pengguna memilih submenu Update elemen gaji

2. Aplikasi menampilkan antarmuka rubah data sesuai submenu yang dipilih,
3. Pengguna melakukan proses perubahan data,
4. Pengguna menekan tombol simpan untuk menyimpan perubahan data pada basis data,
5. Aplikasi menyimpan perubahan data pada basis data,

Alternatif: Nomor 5, jika edit data gagal, aplikasi menampilkan kotak dialog rubah data gagal, pengguna akan kembali ke update elemen gaji.

Precondition: Aplikasi akan menampilkan edit administrator.

Postcondition: Data yang dirubah telah tersimpan pada basis data.

3. Nama *use case*: Input gaji bulanan.

Aktor: Admin HRD.

Deskripsi: Kegiatan melakukan proses input gaji bulanan pada aplikasi. Terdapat tiga jenis proses yaitu:

- a. Melihat jumlah nilai gaji pokok, tunjangan jabatan, absen dan lembur karyawan tersebut yang ditetapkan oleh perusahaan sesuai dengan jabatannya.
- b. Proses untuk menghitung jumlah hari absen dengan nilai absen yang ditetapkan oleh perusahaan.
- c. Proses untuk menghitung jumlah jam lembur dengan nilai lembur yang ditetapkan oleh perusahaan.

Exception: Aplikasi tidak bisa melakukan proses data jika data karyawan tidak ada pada basis data.

Skenario:

1. Aplikasi akan menampilkan halaman input gaji yang berisi nik, nama, jabatan input absen dan input lembur.

2. Administrator memilih menu proses yang akan di pakai,
3. Ketika administrator menekan nama karyawan pada tabel input gaji maka sistem akan menampilkan data nik, nama, nilai gaji pokok, nilai tunjangan jabatan, nilai absen perhari dan nilai lembur perjam karyawan yang dipilih oleh administrator.
4. Ketika administrator menekan absen pada tabel input gaji maka sistem akan menampilkan tabel pengisian uang absen bulanan karyawan yang terdiri dari nama, nilai absen dan perkalian jumlah hari karyawan memasukkan absen sehingga mendapatkan total nilai uang absen yang diterima karyawan dalam satu bulan.
5. Ketika administrator menekan lembur pada tabel input gaji maka sistem akan menampilkan tabel pengisian uang lembur bulanan karyawan yang terdiri dari nama, nilai lembur dan perkalian jumlah jam karyawan melaksanakan lembur sehingga mendapatkan total nilai uang lembur yang diterima karyawan dalam satu bulan.
6. Pengguna menekan tombol simpan untuk menyimpan perubahan data pada basis data,
7. Aplikasi menyimpan perubahan data pada basis data,

Alternatif: Nomor 4 dan 5, jika administrator tidak mengisi jumlah hari absen dan jumlah jam lembur karyawan maka sistem akan menganggap bahwa karyawan tidak mengambil absen dan lembur.

Precondition: Aplikasi akan menampilkan antarmuka input gaji.

Postcondition: data karyawan pada halaman lihat gaji karyawan akan berubah.

4. Nama *use case*: lihat gaji karyawan

Aktor: Admin HRD

Deskripsi: Kegiatan menampilkan data gaji karyawan pada halaman lihat gaji karyawan untuk administrator agar bisa diperbaiki apabila terjadi kesalahan input data.

Exception: data pada tabel lihat gaji tidak akan tampil jika administrator belum mengisi data pada menu input gaji.

Skenario:

1. Administrator memilih menu melihat gaji karyawan dengan login sebagai administrator,
2. Aplikasi menampilkan tampilan rincian gaji karyawan yang berupa no, nik, nama, gaji pokok, tunjangan jabatan, absen perhari (uang absen yang diterima karyawan per satu kali absen), absen [hari] (jumlah hari karyawan melakukan absen), total absen (uang absen dikali jumlah hari absen), lembur perjam (uang lembur yang diterima karyawan per satu jam lembur), lembur [jam] (jumlah jam karyawan melakukan lembur), total lembur (uang lembur dikali jumlah jam lembur), total gaji bruto (gapok+tunjangan jabatan+total absen+total lembur), PPn (pajak penghasilan), jamsostek (jaminan sosial tenaga kerja), pinjaman karyawan, total gaji (gaji bersih yang diterima karyawan setelah potongan), hapus dan edit.
3. Administrator menekan tombol hapus untuk menghapus data karyawan yang ada pada tabel.
4. Administrator menekan tombol edit untuk merubah data karyawan yang ada pada tabel.

Alternatif: tidak ada.

Precondition: tabel lihat gaji karyawan.

Postcondition: data pada tabel telah berubah.

5. Nama *use case*: lihat gaji

Aktor: karyawan

Deskripsi: Kegiatan menampilkan data gaji pada halaman lihat gaji untuk karyawan .

Exception: data pada tabel lihat gaji tidak akan tampil jika administrator belum mengisi data pada menu input gaji.

Skenario:

1. Administrator memilih menu memilih menu lihat gaji dengan login menggunakan *tipe user* karyawan,
2. Aplikasi menampilkan tampilan rincian gaji karyawan yang berupa no, nik, nama, gaji pokok, tunjangan jabatan, absen perhari (keterangan pada tabel lihat gaji karyawan), absen [hari] (keterangan pada tabel lihat gaji karyawan), total absen (keterangan pada tabel lihat gaji karyawan), lembur perjam (keterangan pada tabel lihat gaji karyawan), lembur [jam] (keterangan pada tabel lihat gaji karyawan), total lembur (keterangan pada tabel lihat gaji karyawan), total gaji bruto (gapok+tunjangan jabatan+total absen+total lembur), PPn (pajak penghasilan), jamsostek (jaminan sosial tenaga kerja), pinjaman karyawan, total gaji (gaji bersih yang diterima karyawan setelah potongan) dan foto karyawan.

Alternatif: tidak ada.

Precondition: tampilan layar menu login karyawan.

Postcondition: tabel lihat gaji.

6. Nama *use case*: Merubah password pengguna

Aktor: karyawan

Deskripsi: Kegiatan untuk merubah data nama pengguna dan data kata kunci pada basis data aplikasi yang digunakan pada proses login.

Exception: Tidak ada.

Skenario:

1. Pengguna memilih menu rubah password,
2. Aplikasi menampilkan antarmuka rubah password yang menampilkan data nama pengguna dan data kata kunci yang terdapat pada dari basis data,
3. Pengguna memilih akan mengubah data nama pengguna dan data kata kunci.
4. Jika memilih menu rubah nama pengguna, aplikasi akan menampilkan antarmuka rubah nama pengguna, barulah pengguna melakukan perubahan data nama pengguna. Kemudian aplikasi menampilkan kotak dialog rubah data nama pengguna berhasil.
5. Jika memilih menu rubah kata kunci, aplikasi akan menampilkan antarmuka rubah kata kunci dan pengguna dapat melakukan perubahan data kata kunci. Kemudian aplikasi menampilkan kotak dialog rubah data kata kunci berhasil.

Alternatif: Tidak ada.

Pre condition: Aplikasi menampilkan antarmuka rubah pengguna.

Post condition: Data nama pengguna dan data kata kunci teredit pada basis data aplikasi.

3.6 Proses

Analisis pada segi proses yang berjalan pada aplikasi yang dikembangkan akan ditinjau dari diagram aktifitas.

3.6.1 Diagram Aktifitas

Diagram aktifitas menjelaskan tentang berbagai aktifitas yang berlangsung selama pengguna berinteraksi dengan aplikasi. Diagram aktifitas menjelaskan diagram *use case* secara lebih rinci. Pada diagram *use case* terdapat enam proses. Proses-proses tersebut akan dijelaskan secara lebih rinci sebagai berikut:

1. Login

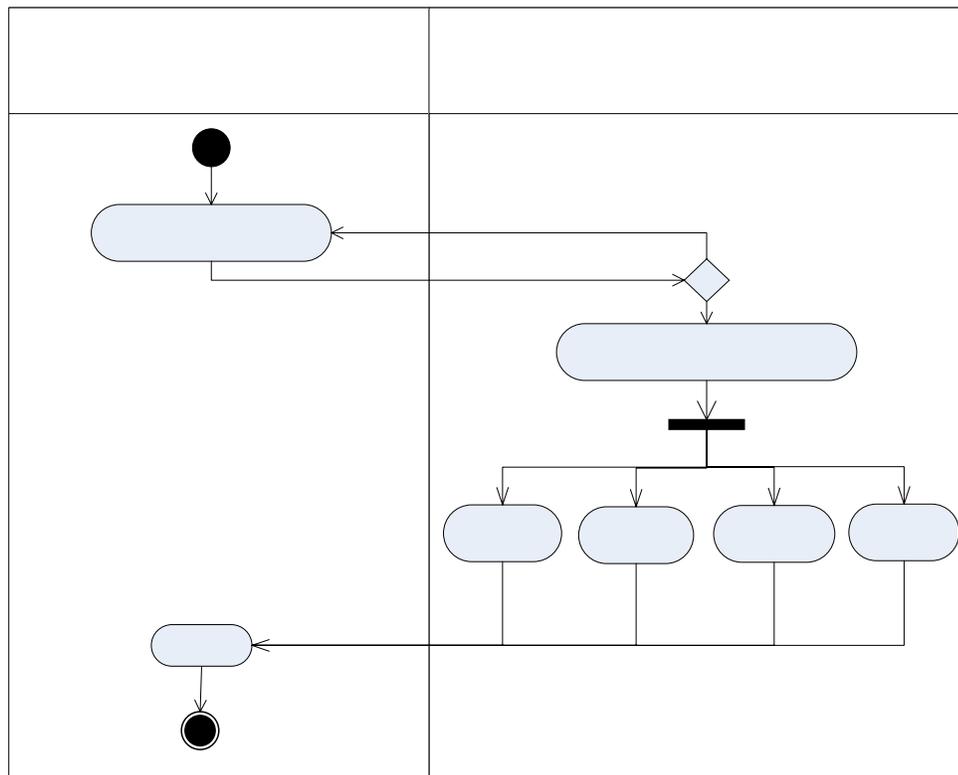
Login adalah sebuah proses awal yang dilakukan oleh pengguna (dalam hal ini adalah Admin HRD) untuk menggunakan aplikasi sistem penggajian karyawan. Proses ini berfungsi untuk membatasi akses pada aplikasi, sehingga hanya pengguna yang berwenang saja yang dapat mempergunakannya. Pengguna akan memasukkan data berupa nama pengguna dan kata kunci. Pengguna selanjutnya akan menekan tombol Login. Tombol ini akan membuat aplikasi melakukan validasi nama pengguna dan kata kunci yang dimasukkan apakah sesuai dengan data nama pengguna dan data kata kunci yang ada pada basis data aplikasi. Jika data nama pengguna dan data kata kunci sesuai, maka aplikasi akan menampilkan antarmuka atau layar menu utama administrator.

Pada antarmuka menu utama administrator, pengguna dapat memilih empat menu, yaitu:

- a. Menu update data karyawan, menu untuk memasukkan data karyawan pada basis data.

- b. Menu update elemen gaji, menu untuk memasukkan atau merubah elemen gaji perusahaan yang ada pada basis data.
- c. Menu input gaji bulanan, menu untuk memasukkan gaji bulanan karyawan sehingga menghasilkan keluaran yang diharapkan.
- d. Menu lihat gaji karyawan, menu untuk melihat gaji karyawan yang telah masuk kedalam basis data. Administrator juga dapat menghapus atau mengedit data apabila terdapat kesalahan.

Diagram aktifitas proses login digambarkan pada Gambar 3.9.



Gambar 3.9 Diagram aktifitas proses login
Departemen HRD

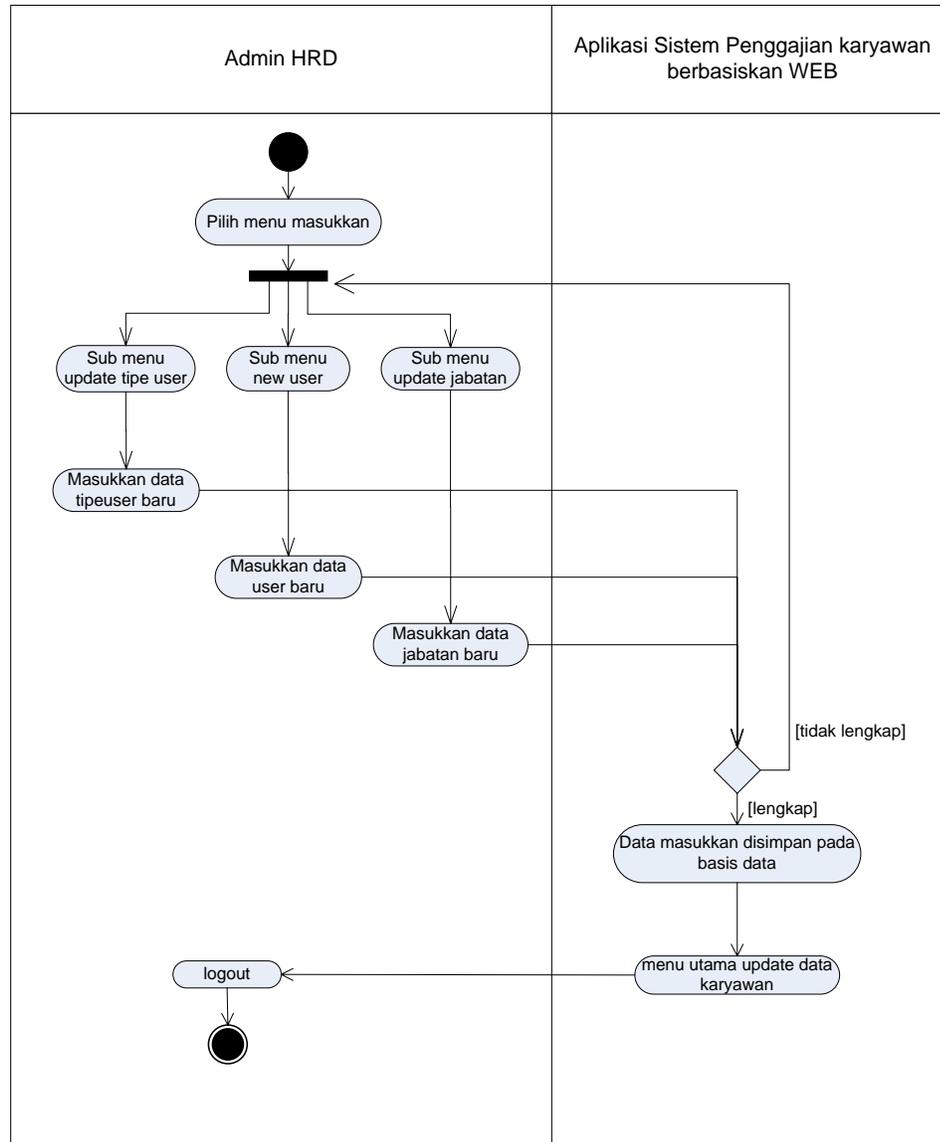
Login(memasukkan data nama dan data kata kunci pengguna

2. Update data karyawan

Administrator yang telah melakukan proses login dapat melakukan proses update data karyawan dengan memilih menu Update data karyawan. Data yang dimasukkan dapat dikelompokkan menjadi tiga kelompok, yaitu:

- a. Update jabatan, terdiri atas id karyawan dan jabatan karyawan.
- b. Update tipe user, terdiri atas id karyawan dan *tipe user* karyawan.
- c. New user, terdiri atas nik, nama, departemen, jabatan, tipeuser dan foto karyawan.

Pada update jabatan dan tipe user, administrator akan memasukkan tipe jabatan dan tipe user baru dalam basis data sedangkan pada *new user* administrator akan mengisi data karyawan baru untuk perusahaan. Setelah data dimasukkan oleh administrator, maka aplikasi akan mengecek apakah seluruh data telah terisi lengkap atau tidak. Jika data yang dimasukkan telah terisi lengkap, maka aplikasi akan menyimpan seluruh data tersebut pada basis data, hal ini menandakan bahwa proses masukkan data telah berhasil dilakukan dan data akan masuk pada tampilan tabel karyawan. Jika terdapat data yang dimasukkan tidak lengkap, maka pengguna akan mengulang proses memasukkan data hasil penilaian dari awal. Diagram aktivfitas dari proses masukkan data digambarkan pada Gambar 3.10.



Gambar 3.10 Diagram aktifitas Update data karyawan

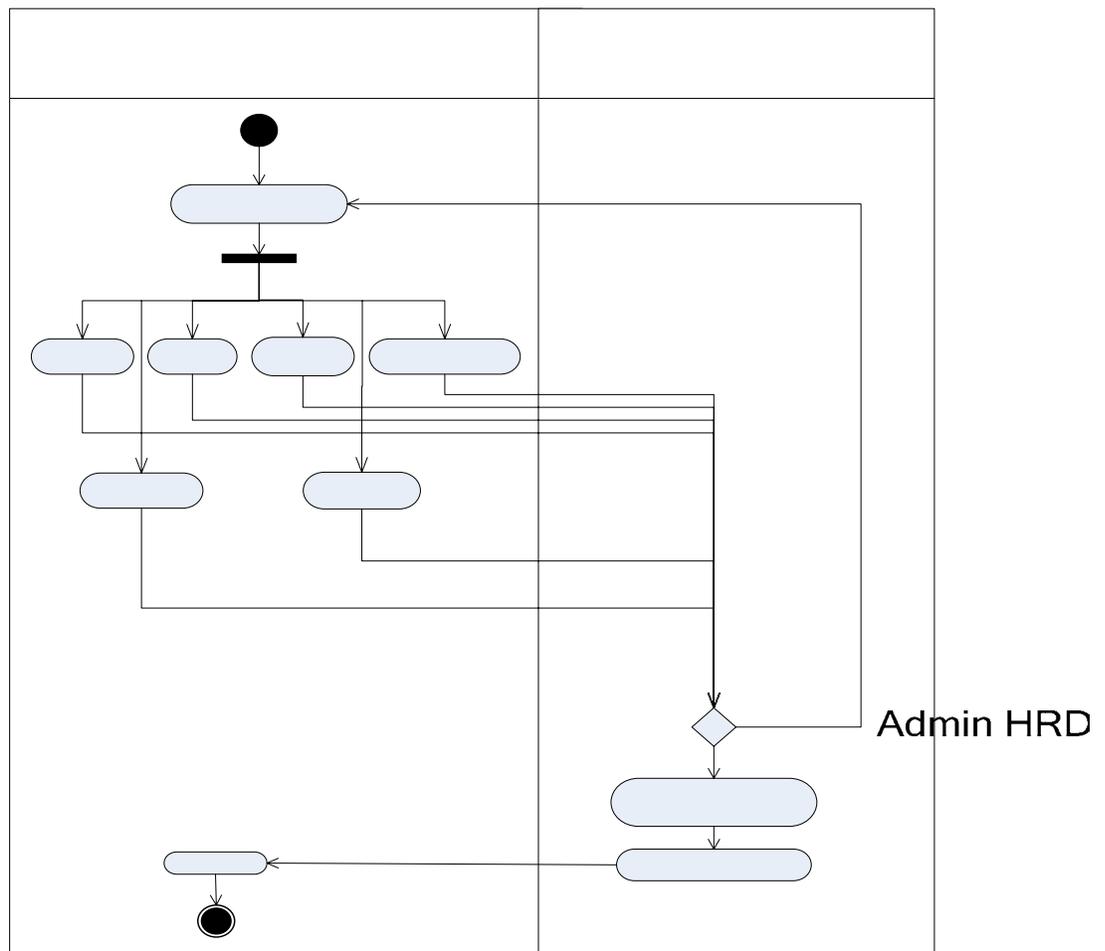
3. Update elemen gaji

Pada menu input elemen gaji, administrator akan diminta untuk mengisi tabel-tabel elemen gaji yang ada pada menu input elemen gaji, yaitu:

- a. Id gaji, berisi kode gaji baru untuk jabatan tertentu,
- b. jabatan, tipe jabatan yang diubah nominal gajinya,
- c. Gaji pokok, nominal gaji pokok yang baru,

- d. Tunjangan jabatan, nominal tunjangan jabatan baru,
- e. Absen per hari, nominal nilai absen per hari
- f. Lembur per jam, nominal nilai lembur perjam.

Data yang telah dimasukkan lalu disimpan ke dalam basis data. Data yang telah masuk lalu di tampilkan pada tabel elemen gaji yang ada dibawah form pengisian elemen gaji.



Gambar 3.11 Diagram aktifitas Update elemen gaji menu masukkan elemen gaji

Masukkan id

Masukkan
jabatar

Masukkan
gaji pokok

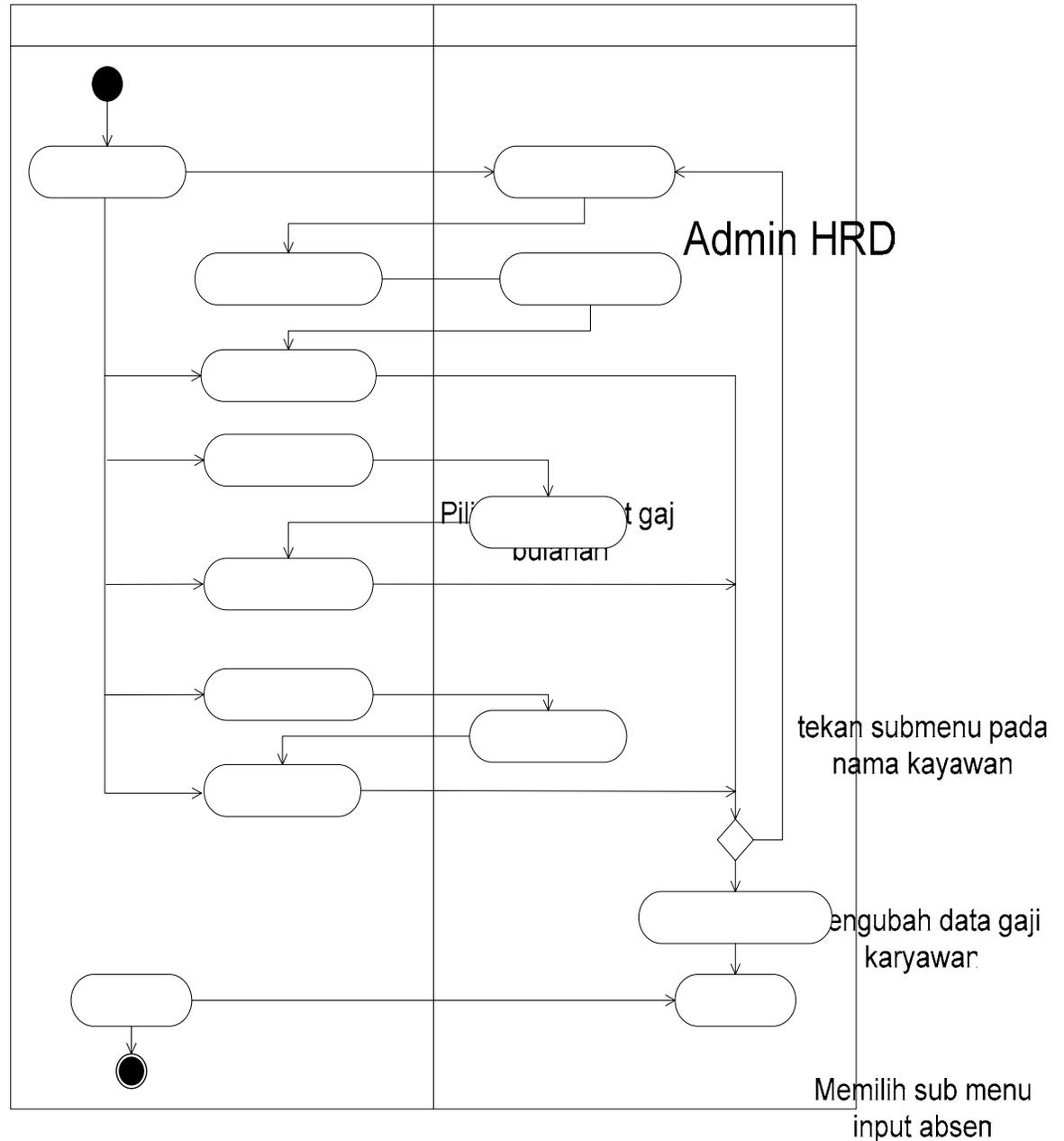
M
tunjar

4. Input gaji

Pada menu input gaji, terdapat tabel data gaji karyawan yang memiliki sub menu langsung pada kolom yang ingin dipilih. Sub menu pada tabel input gaji karyawan adalah:

- a. Data gaji karyawan, administrator dapat langsung melihat data gaji karyawan dengan cara menekan nama karyawan yang dipilih pada tabel dan aplikasi akan menampilkan tabel data gaji karyawan yang bisa di hapus dan diedit oleh administrator.
- b. Input jumlah hari absen karyawan, sub menu ini dapat dipilih dengan menekan kata input absen pada kolom input gaji, dimana administrator dapat mengisi jumlah hari absen yang diambil oleh karyawan.
- c. Input jumlah jam lembur karyawan, sub menu ini dapat dipilih dengan menekan kata input lembur pada kolom input gaji, dimana administrator dapat mengisi jumlah jam lembur yang diambil oleh karyawan.
- d. Gaji pokok, nominal gaji pokok yang baru,
- e. Tunjangan jabatan, nominal tunjangan jabatan baru,
- f. Absen per hari, nominal nilai absen per hari
- g. Lembur per jam, nominal nilai lembur perjam.

Data yang telah dimasukkan lalu disimpan ke dalam basis data. Data yang telah masuk lalu di tampilkan pada tabel gaji bulanan karyawan.



Gambar 3.12 Diagram aktifitas input gaji

5. Diagram aktifitas lihat gaji karyawan

Setelah seluruh data gaji karyawan telah dimasukkan pada basis data dan tidak ada perubahan lagi terhadap data tersebut, maka administrator dapat melihat keseluruhan data gaji karyawan mulai dari gaji bruto karyawan (total gaji

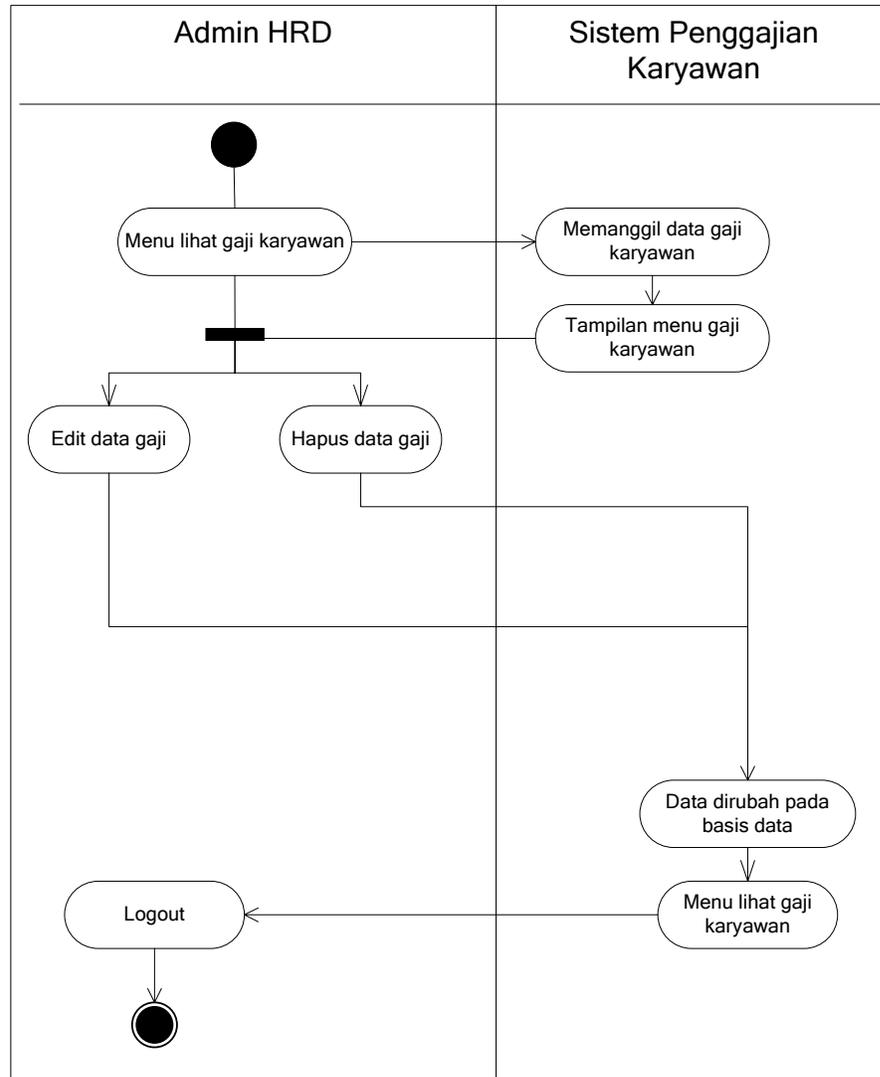
Memilih sub menu
input lembur

sebelum potongan) hingga total gaji bersih karyawan dalam satu bulan. Data yang ada pada tabel di tampilan gaji karyawan adalah :

- a. No (Urut karyawan dalam tabel),
- b. NIK (Nomor Identitas Karyawan),
- c. Nama Karyawan,
- d. Gaji pokok karyawan,
- e. Tunjangan Jabatan, tunjangan yang diberikan oleh perusahaan berdasarkan jabatan karyawan,
- f. Absen /hari, nilai nominal uang yang diterima karyawan saat absen dalam satu hari,
- g. Absen (hari), jumlah hari absen karyawan dalam satu bulan,
- h. Total absen, jumlah nominal uang absen per hari dikali jumlah absen karyawan dalam satu bulan,
- i. Lembur /jam, nilai nominal uang yang diterima karyawan saat lembur dalam satu jam,
- j. Lembur (jam), jumlah jam lembur karyawan dalam satu bulan,
- k. Total lembur, jumlah nominal uang lembur per jam dikali jumlah lembur karyawan dalam satu bulan,
- l. Total gaji bruto, total gaji karyawan sebelum potongan,
- m. PPn (Pajak penghasilan), pajak yang harus dibayar karyawan setiap bulan berdasarkan penghasilan yang diterima oleh karyawan,
- n. Jamsostek (jaminan sosial tenaga kerja), asuransi kesehatan karyawan yang harus dibayar setiap bulan,

- o. Pinjaman karyawan, pemotongan secara langsung pada gaji karyawan apabila karyawan meminjam uang dari perusahaan,
- p. Total potongan, nilai nominal jumlah potongan yang diambil dari gaji karyawan,
- q. Total gaji, jumlah uang gaji bersih yang diterima karyawan setelah mendapatkan potongan,
- r. Hapus, menghapus data gaji karyawan yang sudah tidak diperlukan,
- s. Edit, merubah data gaji karyawan jika terjadi kesalahan pada data.

Setelah melakukan seluruh tahapan input data gaji karyawan, aplikasi akan menyimpan hasil proses input data gaji karyawan pada basis data. Diagram aktifitas proses analisis data hasil penilaian digambarkan pada Gambar 3.13.



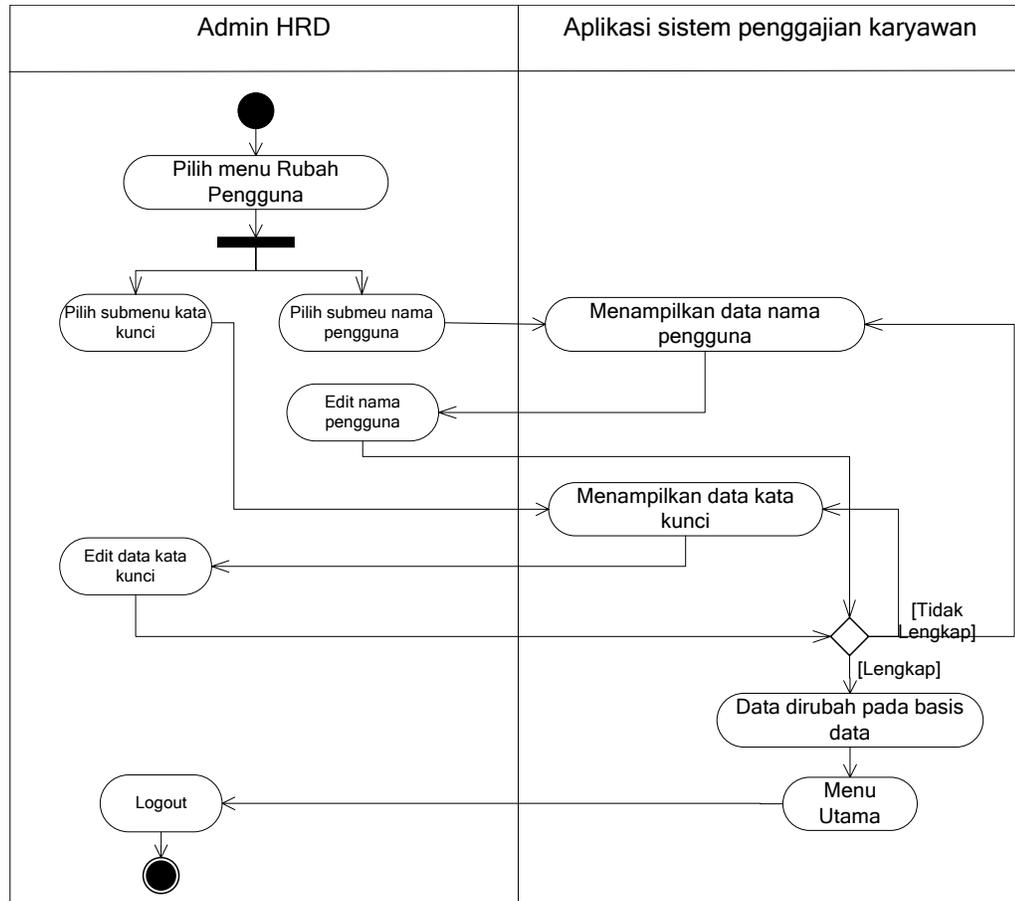
Gambar 3.13 Diagram aktifitas proses lihat gaji karyawan

6. Rubah password

Data nama pengguna dan data kata kunci yang digunakan pengguna pada proses login dapat dirubah melalui proses rubah password. Pengguna memilih menu Rubah Password, aplikasi akan menampilkan data nama pengguna dan data kata kunci yang terdapat pada basis data. Pengguna dapat memilih untuk merubah data nama pengguna atau data kata kunci.

Pengguna dapat memilih untuk merubah data nama pengguna, maka pengguna akan melakukan perubahan terhadap data nama pengguna tersebut dengan mengisi data nama pengguna baru, aplikasi akan mengecek apakah data nama pengguna yang dirubah telah terisi lengkap atau tidak. Jika lengkap, data nama pengguna akan dirubah dari basis data dan pengguna akan kembali ke menu utama. Jika tidak lengkap, pengguna akan mengulangi proses perubahan data nama pengguna.

Selain mengedit data nama pengguna, pengguna juga dapat merubah data kata kunci. Pengguna akan mengisi data kata kunci baru sebanyak dua kali. Aplikasi akan mengecek apakah data kata kunci yang dirubah telah terisi lengkap atau tidak. Jika lengkap, data kata kunci akan dirubah dari basis data dan pengguna akan kembali ke menu utama. Jika tidak lengkap, pengguna akan mengulangi proses perubahan data kata kunci. Diagram aktifitas rubah password digambarkan pada Gambar 3.14.



Gambar 3.14 Diagram aktifitas rubah password

3.7 Antarmuka Sistem

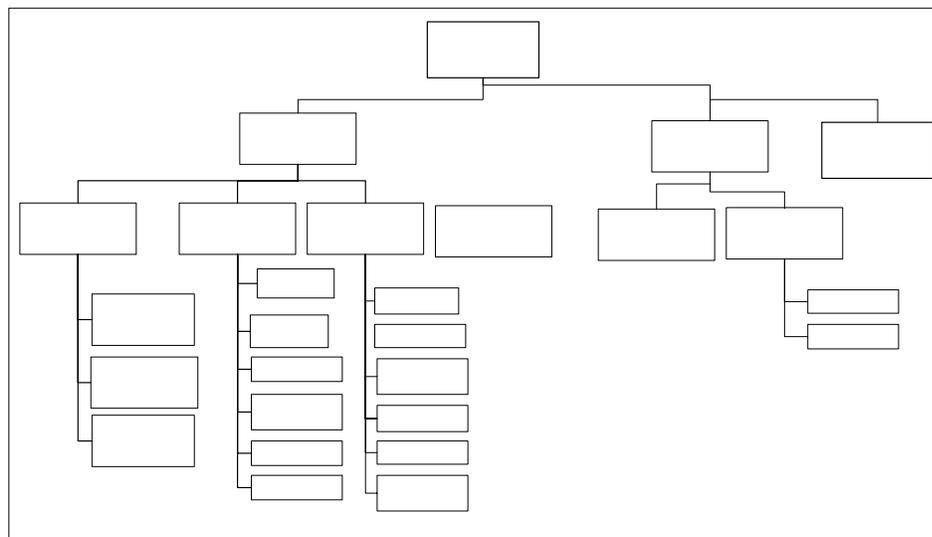
Penulis melakukan perancangan antarmuka aplikasi dengan berpedoman pada prinsip kemudahan pengguna (*user friendly*). Dengan penerapan konsep ini diharapkan pengguna mudah dalam memahami dan menggunakan aplikasi sehingga lebih mempermudah dalam melaksanakan pekerjaannya. Antarmuka sistem pada aplikasi ini dapat dijelaskan melalui struktur menu dan antarmuka.

3.7.1 Struktur Menu

Aplikasi pengukuran kualitas kompetensi manajerial ini terdiri atas dua menu utama, yaitu:

1. Menu utama administrator, terdiri atas empat submenu yaitu:
 - a. update data karyawan, terdiri atas update jabatan, update tipeuser, dan input new user.
 - b. update elemen gaji, terdiri atas update id, jabatan, gaji pokok, tunjangan jabatan, absen, dan lembur.
 - c. input gaji karyawan, terdiri atas input nik, nama, departemen, jabatan, absen dan lembur.
 - d. lihat gaji karyawan.
2. Menu utama karyawan, terdiri atas empat submenu yaitu:
 - a. lihat gaji bulanan, dan
 - b. ubah password, yaitu ubah nama pengguna (*user*) dan password.

Struktur menu aplikasi dapat digambarkan pada Gambar 3.15 berikut.



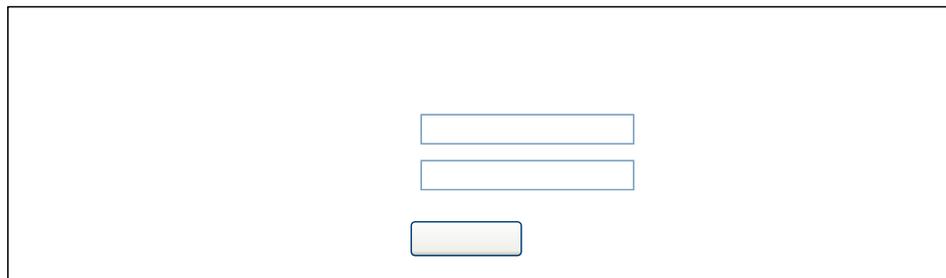
Gambar 3.15 Struktur menu aplikasi

3.7.2 Antarmuka Aplikasi

Antarmuka merupakan suatu bentuk komunikasi dan komponen penghubung antara pengguna dengan aplikasi. Perancangan antarmuka harus selalu berpedoman pada konsep *user friendly*. Rancangan antarmuka yang ada pada aplikasi adalah sebagai berikut:

1. Antarmuka login.

Antarmuka login adalah antarmuka awal dalam menggunakan APKKM. Antarmuka ini berfungsi untuk membatasi akses penggunaan aplikasi, sehingga aplikasi hanya digunakan oleh pengguna yang berwenang, dalam hal ini Administrator HRD dan Karyawan. Antarmuka ini terdiri atas dua kotak masukan (*input box*) yaitu nama pengguna dan kata kunci. Selain itu, juga terdapat tombol login yang berfungsi untuk mengaktifkan prosedur program validasi nama pengguna dan kata kunci pada basis data. Rancangan antarmuka login dapat digambarkan pada Gambar 3.16.

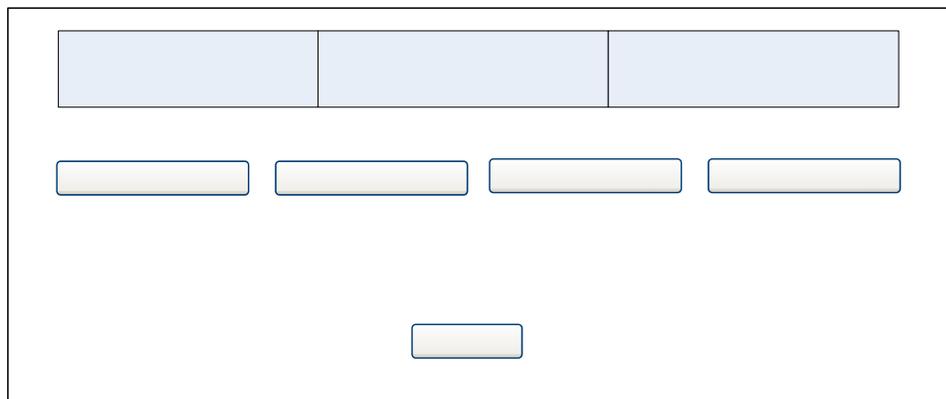
The image shows a simple login interface design within a rectangular frame. It consists of two horizontal input boxes stacked vertically, followed by a single button below them. The input boxes are empty and have a light blue border. The button is rectangular with rounded corners and a light beige background with a thin blue border.

Gambar 3.16 Rancangan antarmuka login

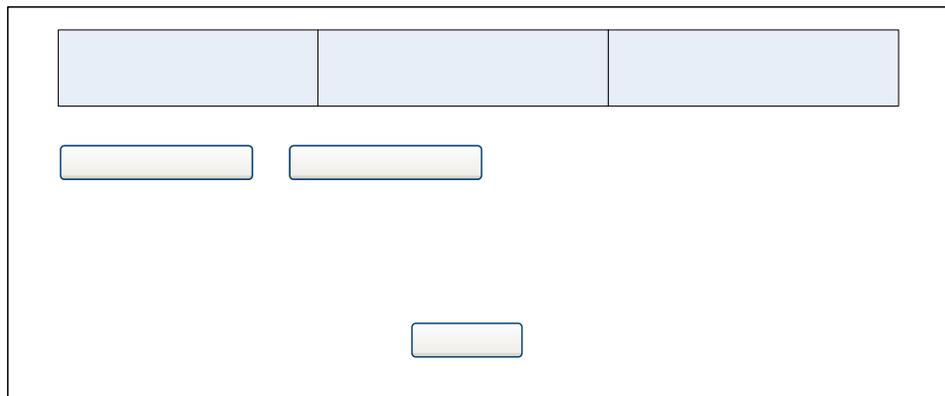
2. Antarmuka menu utama Administrator

Setelah pengguna berhasil melakukan proses login, maka aplikasi akan melalui proses validasi untuk mengenali pengguna menurut tipe user yang ada dalam basis data. Jika pengguna dikenali oleh sistem sebagai administrator HRD maka

aplikasi akan masuk ke antarmuka menu utama administrator. Antarmuka menu utama administrator terdiri atas lima menu utama, yaitu menu Update data karyawan, Update elemen gaji, Input gaji bulanan, Lihat gaji karyawan dan Logout. Jika pengguna ingin mengakhiri penggunaan aplikasi, maka pengguna memilih menu Logout untuk menutup aplikasi. Rancangan antarmuka menu utama administrator digambarkan pada Gambar 3.17 dan menu utama karyawan digambarkan pada Gambar 3.18.



Gambar 3.17 Rancangan antarmuka menu utama administrator



Gambar 3.18 Rancangan antarmuka menu utama karyawan

Pada halaman menu utama karyawan hanya terdapat tiga menu utama yaitu menu lihat gaji, ganti pasword dan logout.

3. Antarmuka menu update data karyawan.

Antarmuka masukan data karyawan adalah antarmuka untuk memasukkan data yang berhubungan dengan karyawan pada basis data. Data yang dimasukkan akan disimpan pada entitas tipeuser, jabatan dan gaji. Tampilan pada antarmuka menu update data karyawan adalah sebagai berikut.

Gambar 3.19 antarmuka menu update data karyawan

Antarmuka proses masukan data karyawan terdiri atas tiga jenis antarmuka yaitu:

- Antarmuka Update jabatan, terdiri atas dua buah kotak **Time** yaitu id jabatan dan nama jabatan. Terdapat tombol Simpan untuk menyimpan data masukan pada entitas jabatan di basis data, tombol batal untuk menghapus isi dari kotak masukan, dan tombol **back** untuk kembali ke antarmuka **Update jabatan** data karyawan. Rancangan antarmuka update jabatan digambarkan pada Gambar 3.20.
- Nik Nama Departemen Jabat

Gambar 3.20 Rancangan antarmuka update jabatan

- b. Antarmuka update tipeuser, terdiri atas dua buah kotak masukan yaitu id tipeuser dan nama tipeuser. Terdapat tombol Simpan untuk menyimpan data masukan pada entitas tipeuser di basis data, tombol batal untuk menghapus isi dari kotak masukan, dan tombol back untuk kembali ke antarmuka menu update data karyawan. Rancangan antarmuka masukan kriteria digambarkan pada Gambar 3.21.

Id
Jabatan

simpa

Id

Gambar 3.21 Rancangan antarmuka update tipeuser

- c. Antarmuka new user, terdiri atas tujuh buah kotak masukan yaitu nik , nama departemen, jabatan, password, tipeuser dan foto. Pada masukkan jabatan dan tipe user penulis menggunakan combo box (kotak pilihan) karena data jabatan dan tipeuser telah ditetapkan oleh administrator (lihat menu update jabatan). Terdapat tombol Simpan untuk menyimpan data masukan pada administrator di basis data, tombol Hapus untuk menghapus isi dari kotak masukan, dan tombol back untuk kembali ke antarmuka menu update data karyawan. Rancangan antarmuka new user digambarkan pada Gambar 3.22.

Id
tipeuser

simpa

Ic

Gambar 3.22 Rancangan antarmuka new user

4. Antarmuka update elemen gaji.

Antarmuka update elemen gaji digunakan untuk melihat dan merubah data gaji pokok, nominal uang absen perhari dan nominal uang lembur perjam untuk tiap jabatan dalam perusahaan yang telah ada pada basis data aplikasi. Setelah administrator melakukan pengisian atau perubahan data baru akan ditampilkan pada tabel dibawah layar tampilan antarmuka update elemen gaji. Antarmuka update elemen gaji digambarkan pada Gambar 3.23.

Nik

Nama

Departemen

Jabatan

Password

Tipeuser

Foto

simpan

Gambar 3.23 Rancangan antarmuka update elemen gaji

a. Antarmuka input gaji bulanan

Antarmuka input gaji bulanan menampilkan seluruh data karyawan yang ada pada basis data. Antarmuka input gaji bulanan mempunyai tampilan utama yang terdiri dari tiga sub menu terdiri dari data gaji pegawai dengan memilih nama karyawan pada tabel menu utama, sub menu input absen yang berfungsi untuk menghitung total jumlah uang absen yang diterima karyawan dalam satu bulan dan sub menu input lembur yang berfungsi untuk menghitung total jumlah uang lembur yang diterima karyawan dalam 1 bulan. Untuk kembali ke menu utama administrator terdapat tombol back.

Rancangan antarmuka input gaji karyawan digambarkan pada Gambar 3.24.

Gaji pokok

Tunjangan jabatan

Absen perhari

Lembur perjam

simpan

Jabatan

Gaji pokok

Tunja

Gambar 3.24 Rancangan antarmuka input gaji karyawan

b. Antarmuka lihat gaji karyawan

Antarmuka lihat gaji karyawan mempunyai dua macam tampilan. Tampilan yang pertama adalah tampilan lihat gaji karyawan untuk administrator menampilkan seluruh data karyawan termasuk perhitungan rinci dari gaji yang ada pada basis data. Tampilan yang ada pada Antarmuka lihat gaji karyawan adalah:

1. Data karyawan, yang terdiri dari nomor, nik, dan nama karyawan.
2. Data gaji bruto (gaji sebelum mendapat potongan) karyawan, yang terdiri dari Gaji pokok tunjangan jabatan, jumlah uang absen sehari, jumlah hari absen karyawan, jumlah total uang absen (jumlah uang absen dikali hari absen karyawan, jumlah uang lembur perjam, jumlah jam lembur karyawan, jumlah total uang lembur karyawan (jumlah nilai uang lembur dikali jam lembur karyawan) dan total gaji bruto.
3. Data potongan gaji karyawan, yang terdiri dari PPn (pajak penghasilan), Jamsostek (jaminan sosial tenaga kerja), pinjaman karyawan dan total potongan yang diterima karyawan.
4. Total gaji bersih karyawan (gaji setelah dikurangi potongan).

Pada tampilan juga terdapat sub menu hapus untuk menghapus data yang ada pada basis data gaji karyawan dan sub menu edit untuk merubah data yang ada pada basis data gaji karyawan. Pada antarmuka lihat gaji karyawan terdapat tombol search dengan satu *combo box* jabatan dan satu kolom pengisian departemen sebagai sarana pencarian data yang berdasarkan jabatan dan departemen juga terdapat satu tombol back untuk kembali ke antarmuka menu utama administrator. Sedangkan untuk tampilan lihat gaji karyawan bagi karyawan memiliki data yang sama tetapi tidak memiliki sub menu hapus edit. tampilan lihat gaji karyawan bagi karyawan juga akan menampilkan foto yang dimasukkan pada basis data perusahaan. Rancangan antarmuka lihat gaji karyawan untuk administrator digambarkan pada Gambar 3.25 dan rancangan antarmuka lihat gaji karyawan untuk karyawan digambarkan pada Gambar 3.26.

				▼							

Gambar 3.25 Rancangan antarmuka lihat gaji karyawan administrator

<div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <div style="width: 30%; height: 20px; background-color: #d9e1f2; border: 1px solid black;"></div> <div style="width: 30%; height: 20px; background-color: #d9e1f2; border: 1px solid black;"></div> <div style="width: 30%; height: 20px; background-color: #d9e1f2; border: 1px solid black;"></div> </div> <div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> <div style="width: 15%; height: 15px; background-color: #d9e1f2; border: 1px solid black;"></div> <div style="width: 70%;"></div> <div style="width: 15%; height: 40px; border: 1px solid black;"></div> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <tr><td colspan="10"></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> <table border="1" style="width: 50%; border-collapse: collapse; margin-bottom: 10px;"> <tr><td colspan="5"></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>																																																																																				
										Nik : Nama : Jabatan : Departemen ;																																																																										

Gambar 3.26 Rancangan antarmuka lihat gaji karyawan (karyawan)

Gaji pokok Tunjangan jabatan Absen sehari Absen

Potongan gaji

PPn Jamsostek Pinjamar Total

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Setelah tahap analisis dan perancangan pada bab III, maka tahap selanjutnya yaitu tahap implementasi. Pada bab ini akan dibahas mengenai implementasi pembuatan basis data, pengkodean dan implementasi perancangan layar.

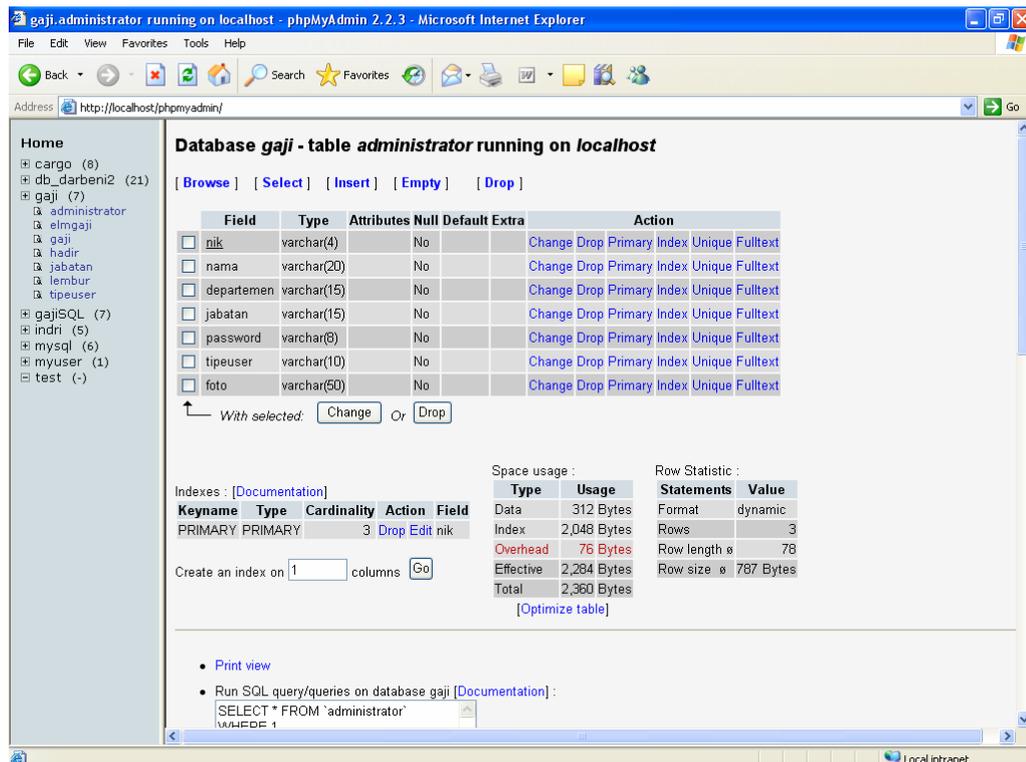
4.1.1 Implementasi Pembuatan Basis Data

Pada implementasi pembuatan basis data ini perangkat lunak yang digunakan yaitu phptriad. Pada implementasi ini akan menampilkan basis data dari tabel-tabel yang ada pada aplikasi penggajian karyawan, seperti: *tb_administrator*, *tb_elm_gaji*, *tb_jabatan*, dan *tb_tipeuser*. Masing-masing tabel tersebut telah dijelaskan pada tahap perancangan pada bab III.

4.1.1.1 Implementasi Pembuatan Basis Data Tb_admin

Tb_administrator terdiri dari tujuh kolom data. Kolom pertama berisi nik karyawan yang mempunyai panjang maksimum 4 karakter, kolom kedua berisi nama yang mempunyai panjang maksimum 20 karakter, kolom ketiga berisi departemen yang departemen mempunyai panjang maksimum 15 karakter, kolom yang keempat

berisi jabatan yang mempunyai panjang maksimum karakter 15, kolom kelima berisi *password* yang mempunyai panjang maksimum 8 karakter, kolom keenam berisi *tipeuser* yang mempunyai panjang maksimum 10 karakter dan kolom ketujuh foto yang mempunyai lebar maksimum 50 karakter. Gambar 4.1 merupakan implementasi basis data *tb_administratator*.

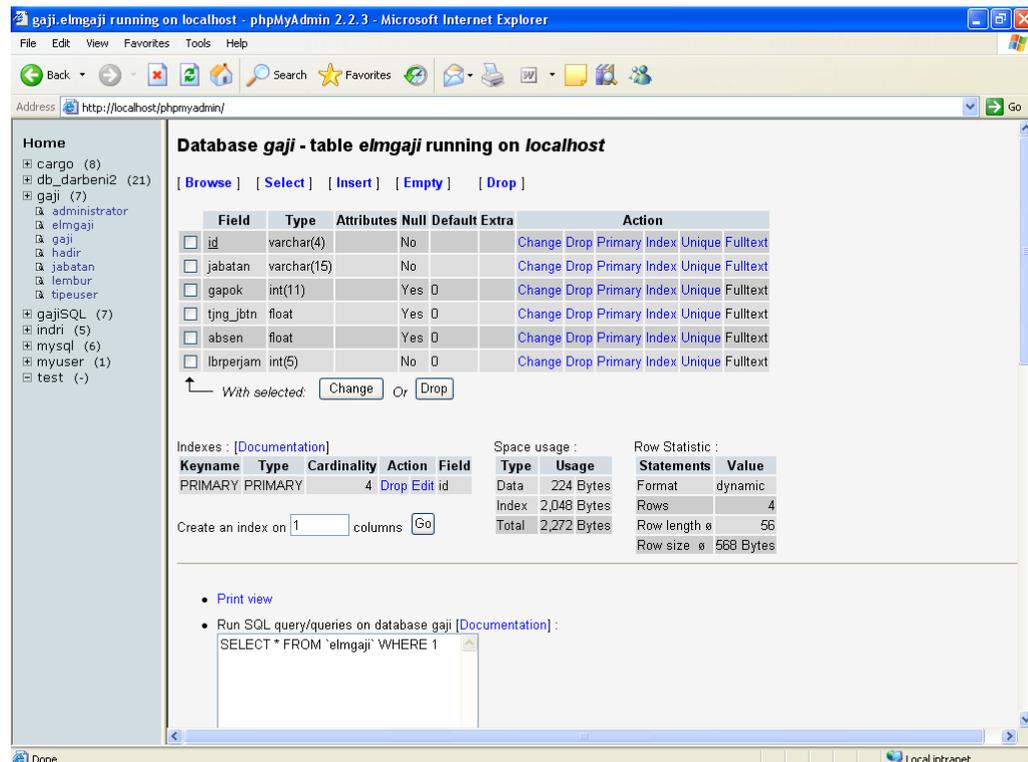


Gambar 4.1 Implementasi Pembuatan Basis Data Tabel Administrator.

4.1.1.2 Implementasi Pembuatan Basis Data *Tb_elm_gaji*

Tb_elemen gaji terdiri dari enam kolom data, kolom pertama berisi nomor id elemen gaji yang mempunyai karakter maksimum sebanyak 4 karakter, kolom yang kedua berisi jabatan yang ada dalam perusahaan yang mempunyai karakter maksimum 15 karakter. Kolom ketiga berisi nilai gaji pokok karyawan kolom ini menggunakan tipe float, kolom keempat adalah kolom absen yang berisi nominal

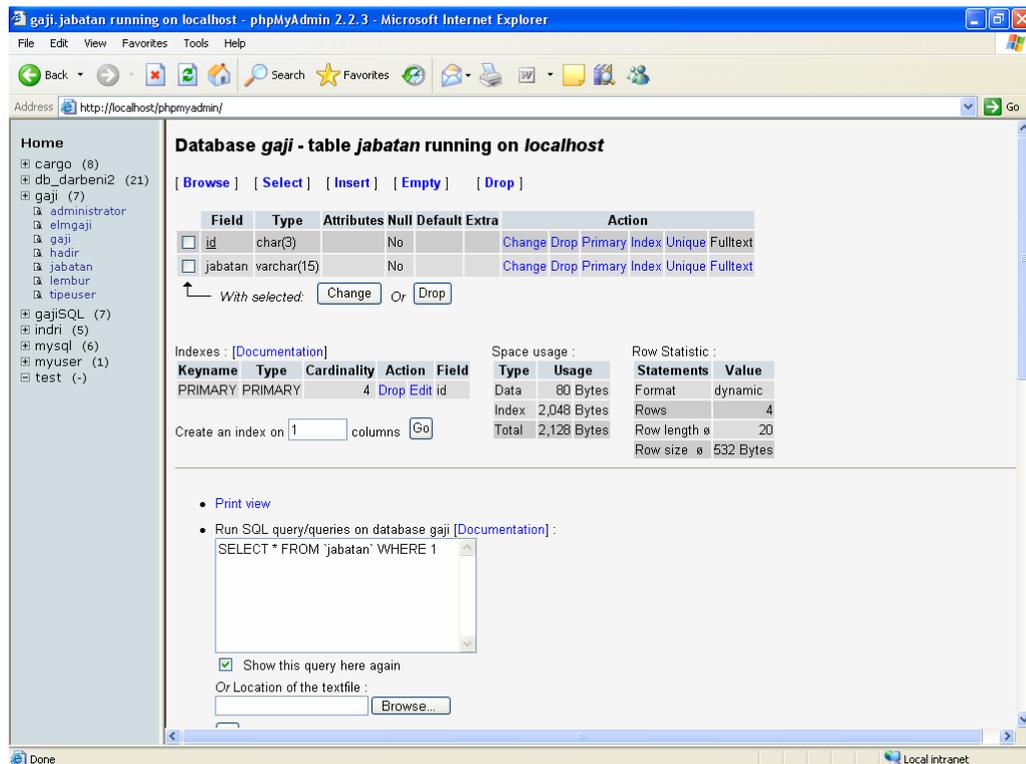
uang absen yang diterima karyawan kolom ini menggunakan tipe float dan kolom ketujuh adalah kolom lbrperjam yang berisi nominal jumlah uang lembur yang diterima karyawan kolom ini menggunakan tipe integer dengan panjang maksimum 5 karakter. Implementasi pembuatan basis data tabel elemen gaji digambarkan pada Gambar 4.2.



Gambar 4.2 Implementasi Pembuatan Basis Data Tabel elemen gaji.

4.1.1.3 Implementasi Pembuatan Basis Data Tb_jabatan

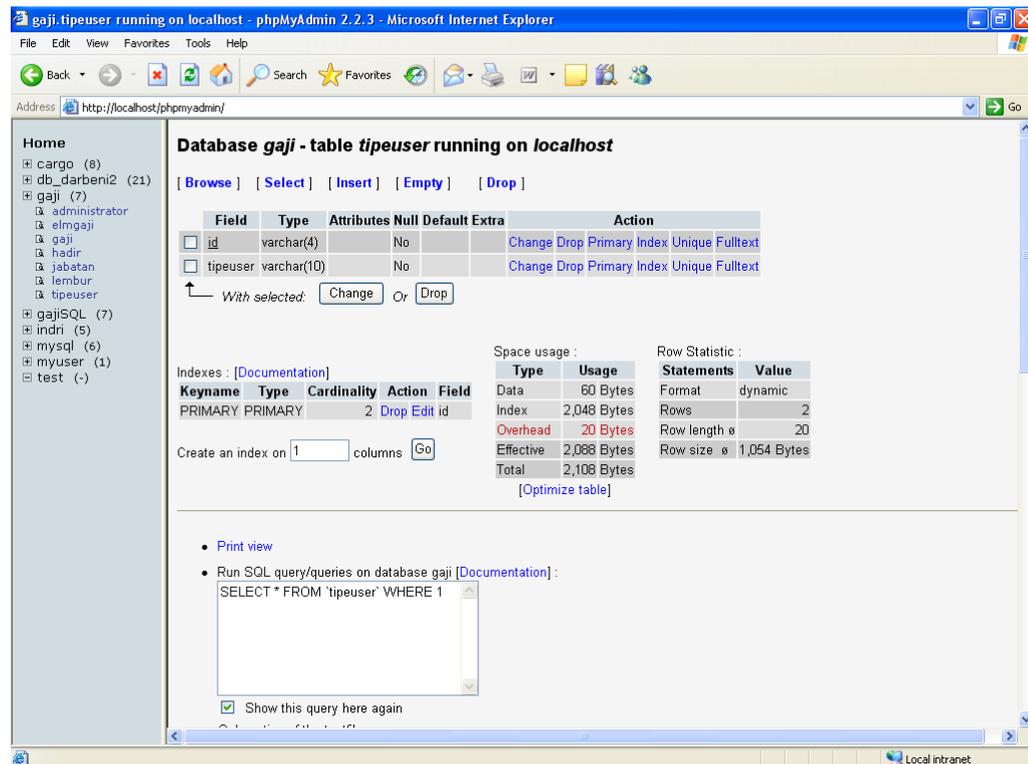
Tb_jabatan terdiri dari dua kolom data, kolom pertama berisi data no id jabatan dengan lebar maksimum 3 karakter dan kolom kedua berisi data dengan lebar maksimum 15 karakter. Gambar 4.3 merupakan implementasi basis data tb_jabatan.



Gambar 4.3 Implementasi Pembuatan Basis Data Tb_jabatan.

4.1.1.4 Implementasi Pembuatan Basis Data Tb_tipeuser

Tb_hasil terdiri dari dua kolom data, kolom pertama berisi data nomor id dengan lebar maksimum 3 karakter dan kolom kedua berisi data *tipeuser* dengan lebar maksimum 10 karakter. Gambar 4.4 merupakan implementasi basis data tb_hasil.



Gambar 4.4 Implementasi Pembuatan Basis Data Tb_tipeuser

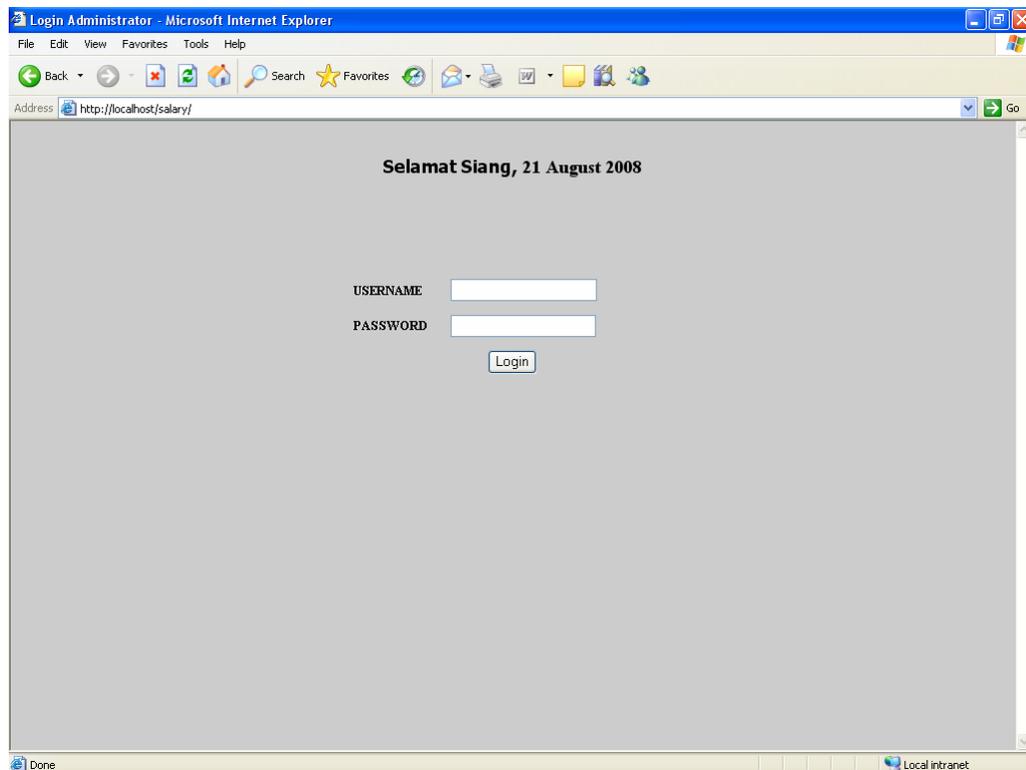
4.1.2 Implementasi Perancangan Layar

Pada implementasi perancangan layar akan menampilkan rancangan layar dan potongan program, seperti: menu login, menu utama administrator, menu utama karyawan, menu tambah *new user*, menu update elemen gaji dan menu lihat data gaji. Masing-masing perancangan layar tersebut telah dijelaskan pada bab III.

4.1.2.1 Implementasi Perancangan Layar Login

Implementasi perancangan layar login peserta dibuat dengan tujuan untuk membatasi siapa saja yang berhak mengakses aplikasi penggajian karyawan ini, sistem akan memberi ijin untuk mengakses aplikasi jika pengguna telah

melaksanakan proses login terlebih dahulu. Gambar 4.5 merupakan implementasi perancangan layar login.



Gambar 4.5 Implementasi Perancangan Layar Login

Kode program 4.1. akan menggambarkan proses yang terjadi setelah peserta melakukan login.

Kode 4.1. Proses login peserta

```

1.  <?
2.  if($Login)
3.  {
4.  if($saman =='admin' && $login =='111')
5.  {
6.      $saman="admin";
7.      $tipe="ADMINISTRATOR";
8.      session_start();
9.      session_register("nama","tipe");
10.     echo "<script>document.location=\\"menu_update_admin.php\\"</script>";
11.     exit();
12. }
13. else
14. {
15.     $nama=$saman;
16.     $psw=$login;
17.     include "KONEKSI.php";
18.     $perintah="select * from ADMINISTRATOR where nama= '$nama' and password = '$login' " ;
19.     $hasil=mysql_query($perintah) or die(mysql_error());
20.     $data=mysql_fetch_array($hasil);
21.     if ($data)
22.     {
23.         $nama="$data[nama]";
24.         $tipe="$data[tipeuser]";
25.         session_start();
26.         session_register('nama','tipe');
27.         echo "<script>document.location=\\"cek2.php\\"</script>";
28.         exit();
29.     }
30.     else
31.     {
32.         echo "<script>alert(\\"Anda Tidak Punya Akses..!\");history.go(-1)</script>";
33.         exit();
34.     }
35. }
36. ?>

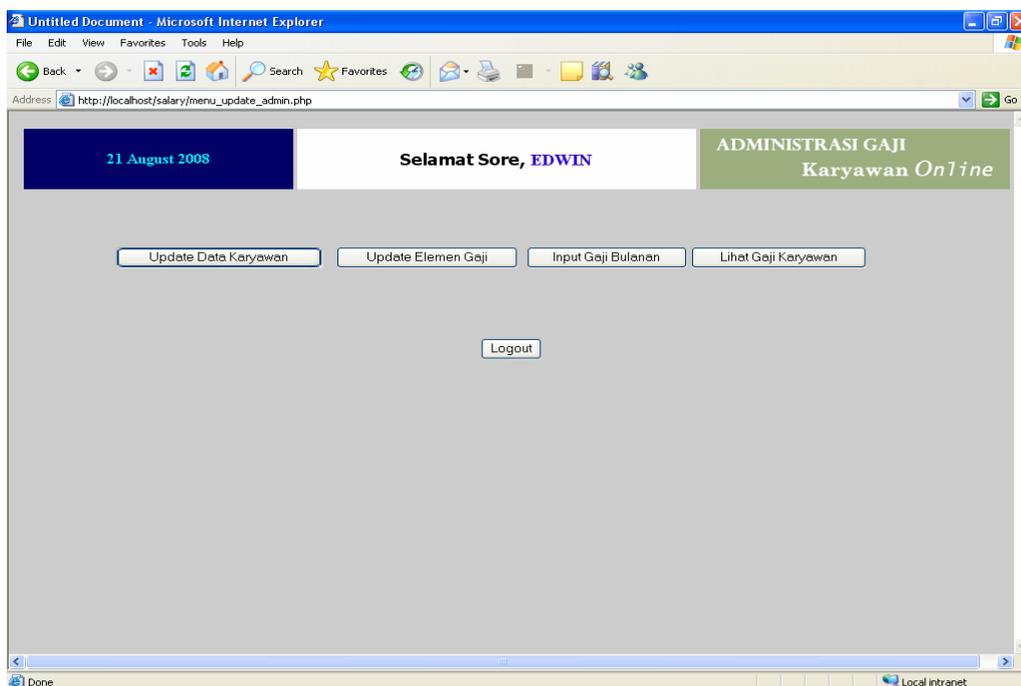
```

Kode program diatas adalah proses validasi dari *username* dan *password* yang dimasukkan oleh pengguna. Kode nomor 2 menjelaskan: Sistem akan ketika pengguna melakukan login maka sistem akan memanggil proses validasi. Kode program nomor 6 sampai nomor 11 menampilkan data *username* dan *password* dari master administrator yang langsung dikenali oleh aplikasi sebagai administrator dan kode program dari nomor 13 sampai nomor 33 menjelaskan apabila pengguna yang login kedalam aplikasi tidak memakai *username* “admin” dan *password* “111” maka

sistem akan melakukan koneksi ke *server* basis data *mysql* untuk mengetahui akses dan tipe user dari pengguna. Jika koneksi tidak berhasil, maka sistem akan menampilkan pesan kesalahan dan pengguna akan diminta kembali ke menu login. Setelah koneksi berjalan baik, sistem akan mengaktifkan basis data, dalam aplikasi sistem penggajian ini *db_gaji* merupakan basis data yang digunakan.

4.1.2.2 Implementasi Perancangan Menu Utama Administrator

Implementasi perancangan layar menu utama administrator akan menampilkan empat menu yang dapat dipilih oleh administrator. Menu tersebut adalah : menu update data karyawan, menu update elemen gaji, menu input gaji dan menu lihat gaji karyawan. Gambar 4.6 merupakan implementasi perancangan layar menu utama administrator.



Gambar 4.6 Implementasi Perancangan menu utama administrator.

Kode program 4.2. merupakan gambaran yang terjadi pada saat administrator memasuki menu utama administrator.

Kode 4.2. rancangan layar menu utama administrator

```

1.      <?
2.      session_start();
3.      if(!(session_is_registered("tipe")))
4.      {
5.          echo "<script>alert(\"you're n
6.      exit;
7.      }
8.      ?>
9.      <html><head>
10.     <meta http-equiv="Content-Type" con
11.     <title>Untitled Document</title>
12.     <style type="text/css">
13.     <!--
14.     #Layer1 {
15.         position:absolute;
16.         left:10px;
17.         top:68px;
18.         width:155px;
19.         height:31px;
20.         z-index:1;
21.         background-color: #999999;
22.     }
23.     -->
24.     </style></head>
25.     <body bgcolor="#CCCCCC"><center>
26.     </center>
27.     <table width="882" border="0">
28.     <tr>

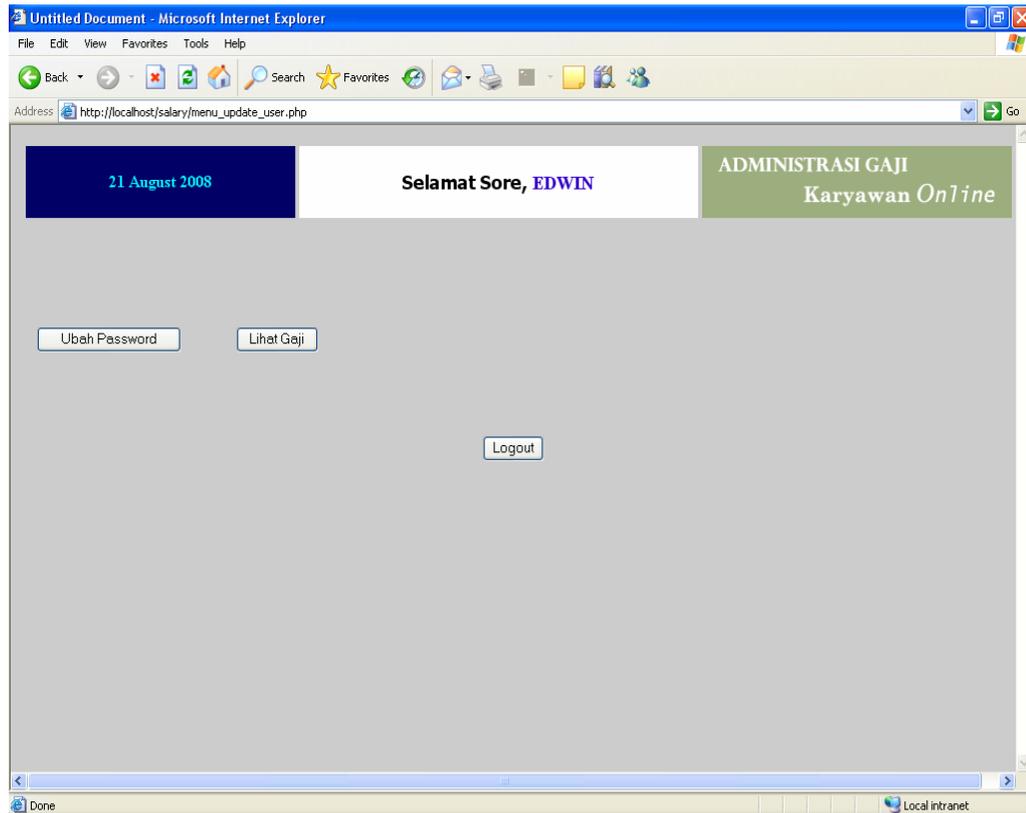
```

Kode program diatas menjelaskan rancangan layar dan pemanggilan yang akan dilakukan oleh sistem ketika administrator memilih salah satu dari menu dilayar menu utama.

Kode program dari nomor 31 sampai nomor 33 akan menjelaskan: jika administrator memilih tombol menu "Update Data Karyawan" maka sistem akan menampilkan layar "table_edit_admin", Kode program dari nomor 36 sampai nomor 38 akan menjelaskan: jika administrator memilih tombol menu "Update elemen gaji" maka sistem akan menampilkan layar "elm_gaji", Kode program dari nomor 42 sampai nomor 44 akan menjelaskan: jika administrator memilih tombol menu "Input gaji bulanan" maka sistem akan menampilkan layar "input_gaji" dan Kode program dari nomor 47 sampai nomor 49 akan menjelaskan: proses pemanggilan layar "view_gaji" jika administrator memilih tombol menu "Lihat gaji karyawan"

4.1.2.3 Implementasi Perancangan layar menu utama karyawan

Implementasi perancangan layar hasil digunakan oleh karyawan dengan pembatasan hak akses sehingga karyawan hanya bisa untuk melihat hasil perincian gaji karyawan yang login kedalam aplikasi dan merubah password karyawan itu sendiri.. Gambar 4.7 merupakan Implementasi perancangan layar menu utama karyawan.



Gambar 4.7 Implemenasi Perancangan layar menu utama karyawan.

Kode program 4.3. merupakan kode implementasi peancangan layar menu utama karyawan.

Kode 4.3. kode layar menu utama karyawan

```

1.  <?
2.  session_start();
3.  if(!(session_is_registered("tipe")))
4.  {
5.      echo "<script>alert('\ you're not licensed to open this Page!\");history.go(-
6.  )</script>";
7.      exit;
8.  }
9.  ?>
10. <html><head><meta http-equiv="Content-Type" content="text/html;" />
11. <title>Untitled Document</title><style type="text/css">
12. <!--
13. #Layer1 {      position:absolute;
14.     left:10px;
15.     top:68px;
16.     width:155px;
17.     height:31px;
18.     z-index:1;
19.     background-color: #999999;
20. }
21. -->
22. </style></head>
23. <body bgcolor="#CCCCCC">
24. <p><center><? include "top.php";?></center></p>
25. <table width="836" border="0">
26. <tr>
27.     <td width="170"><form name="form2" method="post" action="user_admin.php">
28.         <label></label>
29.         <div align="center"><input type="submit" name="Submit2" value="Ubah Password">
30.         </div>
31.     </form></td>
32.     <td width="159"><form name="form2" method="post" action="view_gaji_karyawan">
33.     <label></label>
34.         <div align="center">
35.     <input type="submit" name="Submit4" value="Lihat Gaji">
36.         </div> </form></td>
37.     <td width="433">&nbsp;</td>
38.     <td width="56">&nbsp;</td>
39. </tr>
40. </table>
41. <form name="form3" method="post" action="logout.php">
42.     <label>
43.     <center><input type="submit" name="Submit3" value="Logout"></center>
44. </label></form></body></html>

```

Kode program dari nomor 27 sampai nomor 29 akan menjelaskan: jika karyawan memilih tombol menu "ubah password" maka sistem akan menampilkan layar "user_admin", Kode program dari nomor 32 sampai nomor 35 akan

menjelaskan: proses pemanggilan layar “view_gaji_karyawan” jika administrator memilih tombol menu "Lihat gaji"

4.1.2.4 Implementasi Perancangan Layar Input New User

Implementasi perancangan layar new user digunakan oleh admin untuk memasukkan data pengguna baru dengan memilih tombol menu *new user* pada tampilan utama administrator. Gambar 4.9 merupakan Implementasi perancangan layar *new user*.

The screenshot shows a web browser window titled "EDIT Admin!!!!!! - Microsoft Internet Explorer". The address bar shows "http://localhost/salary/edit_admin.php". The page content includes a date "21 August 2008", a greeting "Selamat Malam, EDWIN", and a header "ADMINISTRASI GAJI Karyawan Online". The main content is a form titled "MASUKKAN DATA PENGGUNA" with the following fields:

NAMA	<input type="text"/>
DEPARTEMEN	<input type="text"/>
JABATAN	STAFF <input type="button" value="v"/>
PASSWORD	111 <input type="text"/>
TIFE USER	KARYAWAN <input type="button" value="v"/>
FOTO	<input type="text"/> <input type="button" value="Browse..."/>

At the bottom of the form are two buttons: "Simpan" and "Batal".

Gambar 4.8 Implementasi Perancangan Layar New User.

Kode program 4.4. merupakan potongan program perancangan *new user*

Kode 4.4. Proses input data new user

```

1.      <?
2.      session_start();
3.      if(!(session_is_registered("tipe")))
4.      {
5.          echo "<script>alert(\`you're not licensed
6.          exit;
7.      }
8.      else
9.      {
10.         if ($tipe!="ADMINISTRATOR")
11.         {
12.             echo "
13.         }
14.     }
15.     ?>
16.     <html>
17.     <title>EDIT Admin!!!!!!</title>
18.     </head>
19.     <body bgcolor="#CCCCCC">
20.     <center>
21.         <? include "top.php";?>&nbsp;
22.     </center>
23.     <?
24.         include "KONEKSI.php";
25.         $cari="SELECT * FROM ad
26.         $hasil=mysql_query($cari);
27.         $data=mysql_fetch_array($h
28.     ?>
29.     <form action=edit_data_admin.php method="post"
30.     <p><font color=#006600 size="5"><b> MASUKK
31.     <?
32.     echo"<input type=hidden name=nik value=$data[ni
33.     echo"<tr>";
34.     echo"<td width=17% bgcolor=#CCCCCC><b>NA
35.     echo"<td width=83%>";

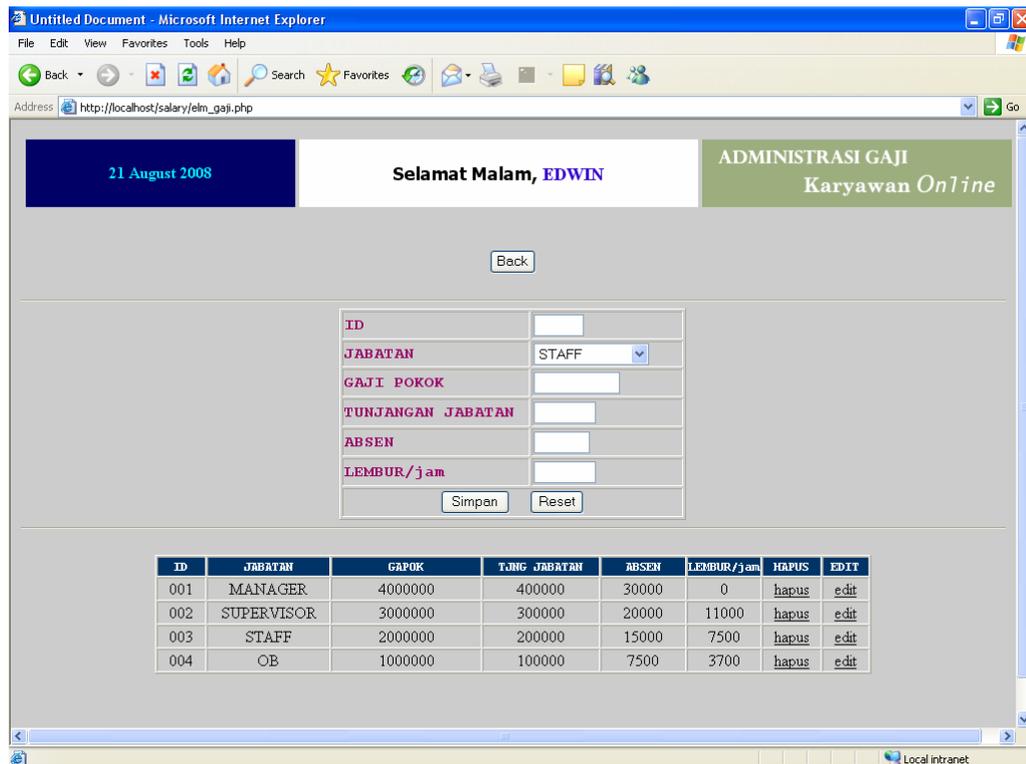
```

Kode program nomor 10 sampa nomor 28 akan menjelaskan: Sistem akan melakukan pemeriksaan terhadap *username* dan *password* yang dimasukan oleh admin apakah sesuai dengan yang ada pada basis data atau tidak.

Kode program nomor 28 sampai nomor 68 akan menggambarkan macam-macam input data yang harus dimasukkan oleh administrator dalam proses penambahan data karyawan dan proses penyimpanan data karyawan kebasis data di jelaskan pada kode program nomor 69 sampi nomor 70.

4.1.2.5 Implementasi Perancangan Layar Menu Update Elemen gaji

Implementasi perancangan layar menu update elemen gaji digunakan oleh admin, setelah pengguna login kedalam aplikasi sebagai administrator maka sistem secara otomatis akan menampilkan menu utama administrator. administrator lalu mdapat memilih menu update elemen gaji pada menu utama administrator. Gambar 4.9 merupakan Implementasi perancangan layar update elemen gaji.



Gambar 4.9 Implementasi Perancangan Layar Update Elemen Gaji

Kode program 4.5. merupakan potongan program implementasi perancangan layar update elemen gaji.

Kode 4.5. Proses halaman update elemen gaji

```

1.     ?>
2.     <html><head><title>Untitled Document</title><st
3.     </style></head>
4.     <body bgcolor="#CCCCCC">
5.     <center><p><? include "top.php";?>
6.     <?
7.     include "KONEKSI.php";
8.     $perintah="SELECT * FROM elmgaji order by id"
9.     $hasil=mysql_query($perintah);
10.    $data=mysql_fetch_array($hasil);
11.    ?>
12.        <form name="form2" method="post" action="cel
13.        <input type="submit" name="Submit3" value="B
14.        <form name="form1" method="post" action="ne
15.        <td width="183"><span class="style31">ID</sp
16.        <td width="101">
17.        <td><span class="style31">JABATAN</span><
18.    <?
19.        include "KONEKSI.php";
20.        $cari="SELECT * FROM jabatan order b
21.        $hasil=mysql_query($cari);
22.    echo"<select name=jbtn>";
23.        while($data=mysql_fetch_array($hasil))
24.    {
25.        echo strtoupper("<option value=$data[jabatan]>$
26.    }
27.        <td><span class="style31">GAJI POKOK </s
28.        <td><input type="text" name="gapok" size="
29.        <td><span class="style31">TUNJANGAN JA
30.        <td><input type="text" name="tjng" size="6"
31.        <td><span class="style31">ABSEN</span></
32.        <td><input type="text" name="absen" size="5
33.        <td><span class="style31">LEMBUR/jam</s
34.        <td><input type="text" name="lmbr" size="6"
35.        <center class="style31"><input type="submit"

```

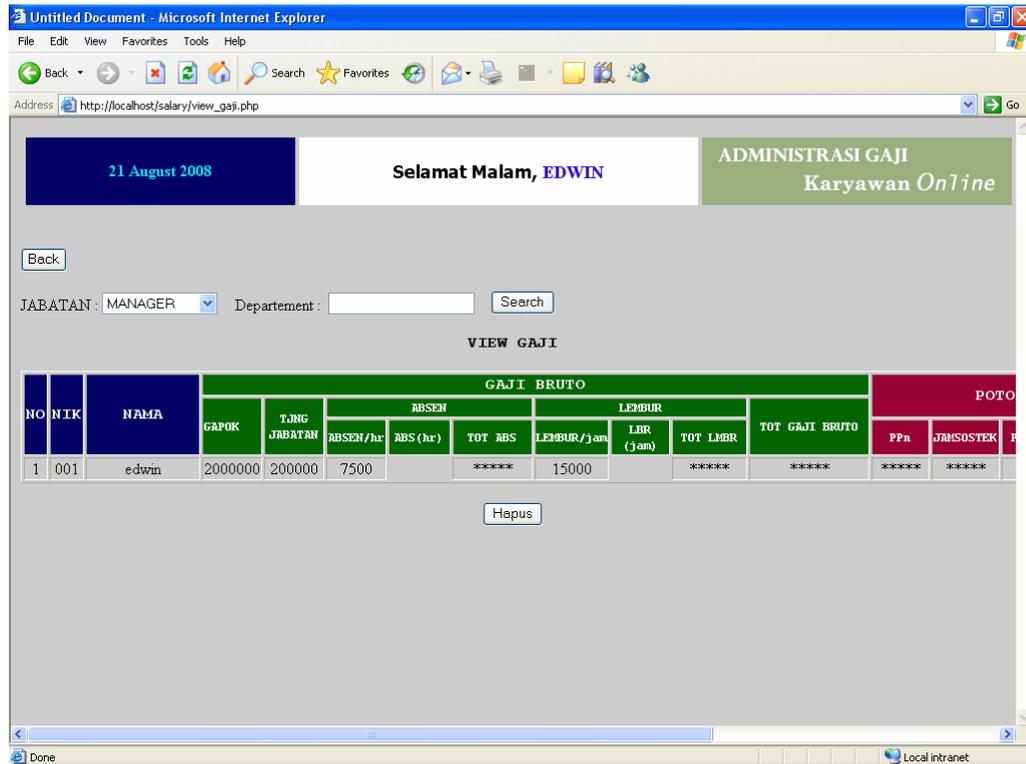
Kode program nomor 7 sampai nomor 10 menjelaskan bahwa sistem akan melakukan koneksi ke *server* basis data *mysql* dengan nama pada *tb_gaji*. Kode program nomor 12 sampai nomor 17 merupakan proses ketika pengguna menekan tombol “back” maka pengguna akan kembali ke layar menu update elemen gaji.

kode program nomor 27 sampai nomor 36 menjelaskan proses pengisian yang harus dilakukan administrator dalam memasukkan data gaji baru untuk karyawan sesuai dengan jabatan dalam perusahaan. Setelah pengisian data selesai maka administrator menyimpan data ke dalam *server* basis data *mysql* dengan menekan tombol simpan atau menekan tombol batal untuk mengosongkan tabel pengisian data.

kode program nomor 37 sampai nomor 70 menjelaskan bahwa sistem akan melakukan koneksi ke *server* basis data *mysql* untuk menampilkan data yang ada pada basis data. Apabila data kosong maka sistem akan menampilkan kalimat DATA MASIH KOSONG pada layar tampilan menu new gaji dan Kode program 47 sampai 55 adalah tampilan jika sistem menemukan data pada *server* basis data *mysql*

4.1.2.6 Implementasi Perancangan Layar Menu Lihat Gaji karyawan (Administrator)

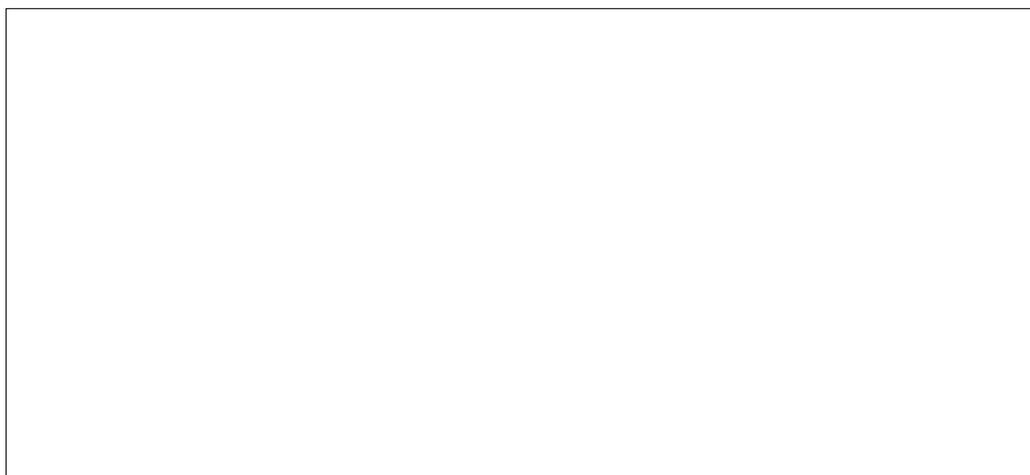
Implementasi perancangan layar menu lihat gaji karyawan dengan login sebagai administrator digunakan oleh administrator untuk memeriksa kembali tampilan rincian gaji karyawan sebelum dilihat oleh karyawan, administrator dapat merubah data apabila terjadi kesalahan dan menghapus data yang sudah tidak terpakai. Gambar 4.10 merupakan Implementasi perancangan layar lihat gaji karyawan untuk administrator..



Gambar 4.10 Implementasi Perancangan Layar Lihat Gaji Karyawan
(Administrator)

Kode program 4.6. merupakan potongan program implementasi perancangan layar lihat gaji karyawan (administrator).

Kode 4.6. Proses halaman lihat gaji karyawan (administrator)



Kode 4.7. (Lanjutan) Proses halaman lihat gaji karyawan (administrator)

```

19.    <?
20.    include "KONEKSI.php";
21.    $perintah="SELECT * FROM gaji order
22.    $hasil=mysql_query($perintah);
23.    $data=mysql_fetch_array($hasil);
24.    if (empty($data))
25.    {
26.        echo("<font size=3 color=#000000><c
27.    }
28.    else
29.    {
30.    ?>
31.    <center><span class="style6 style1"><st
32.    <?
33.    echo "<form action=hapus_data.php metd
34.    ?>
35.    <span class="style6">NO</span></div><
36.    <span class="style6">NIK</span></div>
37.    <span class="style6">NAMA</span></d
38.    <span class="style18">GAJI BRUTO</s
39.    <span class="style18">POTONGAN GA
40.    <span class="style6">TOT GAJI</span>
41.    <span class="style22">HAPUS</span><
42.    <span class="style22">EDIT</span></di
43.    <span class="style6">GAPOK</span></
44.    <span class="style6">TJNG JABATAN
45.    <span class="style6">ABSEN</span></c
46.    <span class="style6">LEMBUR</span><
47.    <span class="style6">TOT GAJI BRUTO
48.    <span class="style6">ABSEN/hr</span>
49.    <span class="style6">ABS(hr)</span></
50.    <span class="style6">TOT ABS </span>
51.    <span class="style6">LEMBUR/jam</sp
52.    <span class="style6">LBR(jam)</span><
53.    <span class="style6">TOT LMBR </span>

```

Kode program nomor 4 dan 5 merupakan proses ketika pengguna menekan tombol “back” maka pengguna akan kembali ke layar menu menu utama administrator.

Kode program nomor 7 sampai nomor 18 menjelaskan proses pencarian data berdasarkan tipe jabatan untuk mempermudah administrator menangani data yang jumlahnya banyak.

Kode program nomor 20 sampai nomor 26 menjelaskan bahwa sistem akan melakukan koneksi ke *server* basis data *mysql* untuk menampilkan data yang ada pada basis data. Apabila data kosong maka sistem akan menampilkan kalimat DATA MASIH KOSONG pada layar tampilan menu lihat gaji karyawan dan Kode program 35 sampai 57 adalah rancangan tampilan tabel rincian gaji karyawan dan 65 sam 83 adalah proses pengambilan data dari *server* basis data *mysql* dalam *tb_gaji* kedalam tampilan menu lihat gaji karyawan.

4.2 Pengujian

Setelah tahap implementasi, tahap selanjutnya adalah tahap pengujian. Pada tahap ini akan dibahas mengenai lingkungan pengujian, skenario pengujian, dokumen pengujian dan hasil pengujian, dan analisis pengujian.

4.2.1 Lingkungan Pengujian

Lingkungan pengujian pada aplikasi ini meliputi perangkat keras dan perangkat lunak, spesifikasi masing-masing lingkungan tersebut adalah sebagai berikut:

1. Spesifikasi Perangkat Lunak, meliputi:

- a. Sistem operasi yang digunakan pada aplikasi sistem penggajian karyawan ini adalah Microsoft Windows XP SP2.
- b. Macromedia Dreamweaver MX 2004 digunakan untuk merancang program aplikasi sistem penggajian karyawan ini.
- c. Adobe Photoshop 7.0, digunakan untuk mendesain tampilan pada aplikasi sistem penggajian karyawan ini.
- d. Phptriad digunakan untuk menjalankan aplikasi psikotes ini
- e. Phpmyadmin digunakan untuk membuat basis data.
- f. Apache digunakan sebagai localhost.
- g. Microsoft Internet Explorer 4.0 digunakan sebagai *browser* internet.

2. Spesifikasi Perangkat Keras, meliputi

- a. Komputer dengan prosesor Intel pentium IV 1,8 GHz.
- b. Memori 256 MB.
- c. VGA G-force 64 MB.
- d. Monitor 15'
- e. Keyboard dan Mouse

4.2.2 Skenario Pengujian

Skenario pengujian meliputi pengujian perangkat lunak terhadap fungsionalitas aplikasi yang dibangun berfungsi untuk melakukan pengujian proses yang terjadi pada perangkat lunak yang dibangun dengan proses yang terjadi didalam sistem dan menguji kelayakan aplikasi dalam sebuah situs yang akan digunakan.

Metode yang digunakan dalam aplikasi psikotes ini adalah *black-box testing*. Metode ini sesuai dengan aplikasi yang akan dibangun, karena metode *black-box testing* melakukan pengujian dengan cara memberikan sejumlah masukan pada program aplikasi yang akan diproses sesuai dengan kebutuhan fungsionalnya untuk menghasilkan keluaran yang diinginkan.

Apabila keluaran proses tidak menghasilkan sesuai yang diinginkan, maka program aplikasi masih terdapat kesalahan-kesalahan yang harus diperbaiki pada program aplikasi tersebut. Apabila keluaran proses menghasilkan kebutuhan fungsionalnya, maka program aplikasi tersebut benar.

Web browser Internet Explorer digunakan untuk melakukan pengujian pada program aplikasi penggajian karyawan, dengan cara mengetikkan alamat URL sebagai berikut: <http://localhost/salary>. *Black-box testing* melakukan pengujian pada program aplikasi aplikasi penggajian karyawan dimulai dari menu login sampai semua menu yang terdapat pada program aplikasi Penggajian Karyawan.

Tabel 4.1 Skenario Pengujian Aplikasi Penggajian Karyawan

No	Deskripsi Fungsional	Kelompok Uji	Kasus Uji	Hasil Yang Diharapkan
1	Login	Data salah	Memasukan data nama pengguna yang berbeda dengan yang ada pada basis data	Menampilkan pesan kesalahan
2	Memasukan data karyawan baru	Normal	Memasukkan data nik, nama, departemen, jabatan, password tipeuser dan foto	Data tersimpan pada basis data dan ditampilkan pada tabel pengguna di antarmuka update data karyawan
3	Memasukan data gaji	Normal	Memasukkan data id, jabatan gaji pokok, tunjangan jabatan, absen dan lembu/jam.	Data tersimpan pada basis data dan ditampilkan pada antarmuka elemen gaji
4	Memasukan data jabatan	Data salah	Hanya memasukkan data id , sedangkan jabatan karyawan dikosongkan	Menampilkan pesan kesalahan
5	Memasukan data tipe user	Normal	Memasukkan data id, dan tipeuser yang belum ada pada bais data.	Data tipeuser telah bertambah dan tersimpan dalam basis data tb_tipeuser
6	Merubah data gaji bulanan	Normal	Memilih sub menu edit pada data gaji bulanan dan merubah nilai gaji pokok.	Data tersimpan dalam basis data dan nilai gaji pokok pada tabel gaji bulanan berubah.
7	Menghapus data gaji bulanan	Normal	Memilih menu hapus pada antarmuka lihat gaji bulanan.	Data karyawan yang dihapus telah hilang dari menu gaji bulanan dan data baru telah tersimpan pada basis data.
8	Logout	Normal	Memilih menu logout pada antarmuka menu utama	Mengakhiri penggunaan aplikasi

4.2.3 Dokumen Hasil Pengujian

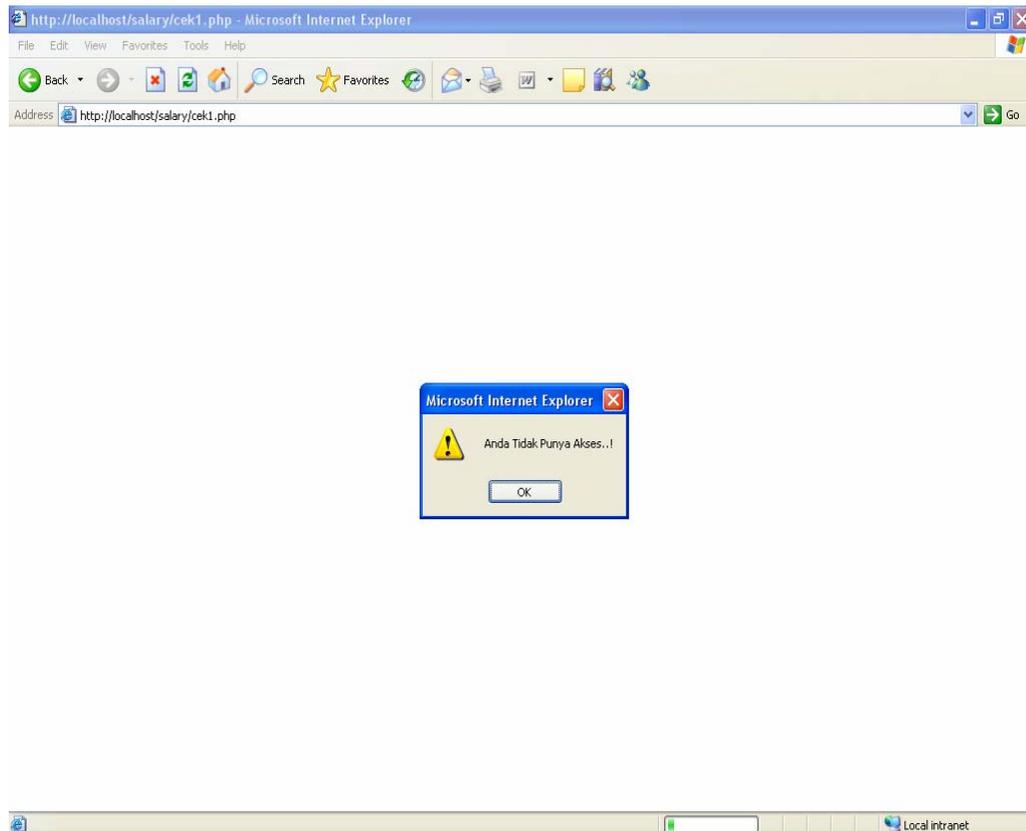
Dokumen hasil pengujian merupakan dokumen yang berisi berdasarkan nomor urut pada skenario pengujian Aplikasi Penggajian Karyawan. Tabel 4.2 adalah tabel dokumen hasil pengujian.

Tabel 4.2 Dokumen Hasil Pengujian Perangkat Lunak

No	Deskripsi Fungsional	Kelompok Uji	Kasus Uji	Hasil Yang Diharapkan	Hasil Uji
1	Halaman login	Data salah	Memasukkan data <i>username</i> dan <i>password</i> yang tidak ada dalam basis data	Menampilkan pesan kesalahan	Diterima
2	Halaman new user	Normal	Memasukan nik=004, nama=otong, departemen=gsd, jabatan OB , <i>password</i> =333, tipe user=karyawan dan foto	Menampilkan data user baru	Diterima
3	Halaman input elemen gaji	Normal	Memasukan data dengan id=004, jabatan=OB, gaji pokok=1000000,tunjangan jabatan=100000,absen=7500 dan lembur/jam=3700	Menampilkan data gaji OB pada tabel gaji di antarmuka elemen gaji	Diterima
4	Halaman update jabatan	Normal	Memasukkan id=004 dengan jabatan=OB	Menampilkan data jabatanbaru	Diterima
5	Halaman update tipe user	Data salah	Tidak Memasukan data tipeuser	Menampilkan pesan kesalahan	Diterima
6	Merubah data gaji bulanan	Normal	Merubah nilai gaji pokok pada data gaji karyawan dengan nama=edwin	Menampilkan data perubahan pada halaman lihat gaji bulanan	Diterima
7	Halaman lihat gaji karyawan	Normal	Menghapus data gaji dengan karyawan nama=otong	Data karyawan dan data gaji karyawan tersebut dapat terhapus dari basis data	Diterima
8	Halaman logout	Normal	Memilih <i>link logout</i> 1	Keluar dari halaman admin	Diterima

4.2.4 Analisis Hasil Pengujian

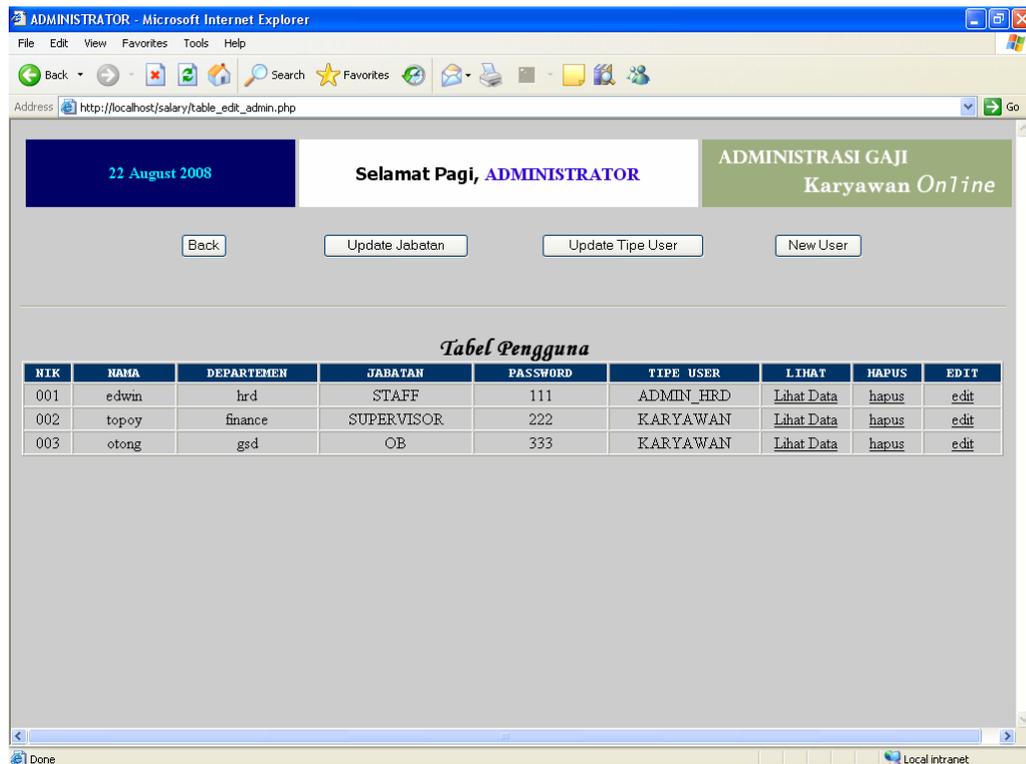
Setelah pengujian perangkat lunak telah dilakukan, selanjutnya melakukan analisis dari hasil pengujian perangkat lunak. Gambar 4.11 merupakan gambar yang menunjukkan hasil pengujian halaman login.



Gambar 4.11 Hasil Pengujian Halaman Register.

Pada halaman login, pengujian dilakukan dengan kelompok uji salah yang berarti kelompok uji memasukan data salah. Pada Gambar 4.11 kelompok uji memasukkan data username dan password yang tidak ada dalam basis data, sistem akan menampilkan pesan kesalahan, dan anda diminta oleh sistem untuk kembali ke halaman register. Pengujian pada halaman register ini menghasilkan rancangan yang sesuai diinginkan.

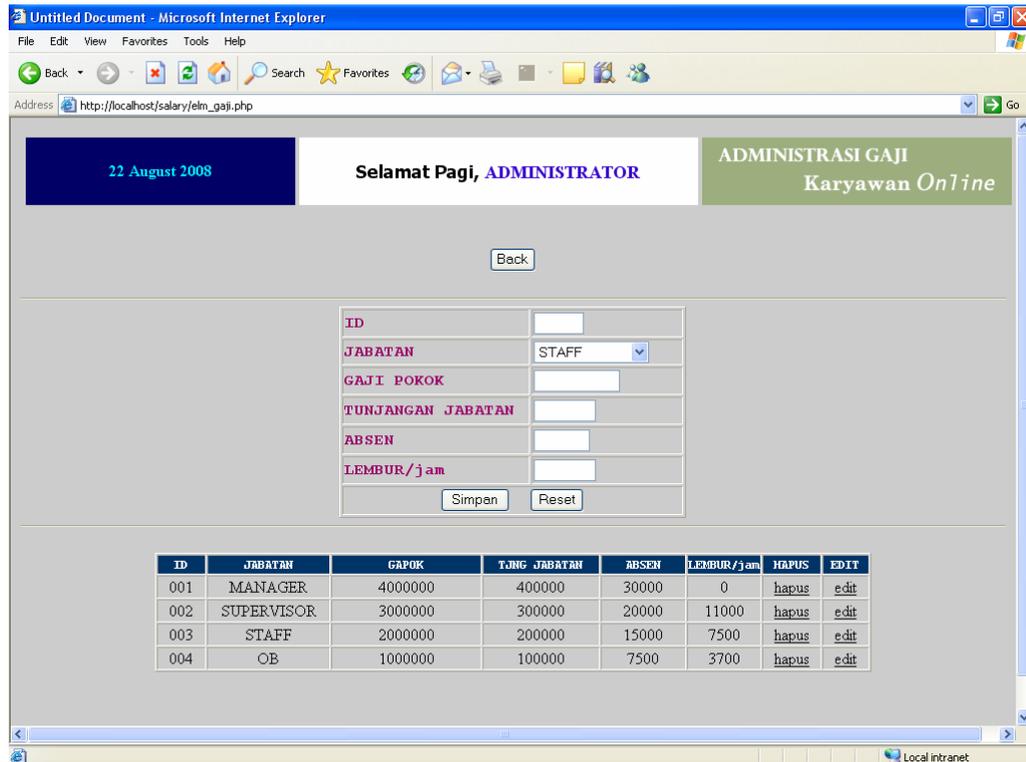
Gambar 4.12 merupakan gambar yang menunjukkan proses yang terjadi ketika administrator memasukkan data pengguna baru.



Gambar 4.12 Hasil Pengujian Halaman Login Peserta.

Pada halaman *new user*, pengujian dilakukan dengan kelompok uji normal yang berarti kelompok uji memasukan data benar. Kelompok uji melakukan penambahan data baru dengan menggunakan data nik= 004, nama=Otong, departemen=gsd, jabatan=OB, *password*=333, tipeuser=karyawan dan foto, sistem akan menyimpan data tersebut dan menampilkanya dilayar halaman tabel edit admin, seperti yang terlihat pada Gambar 4.12.

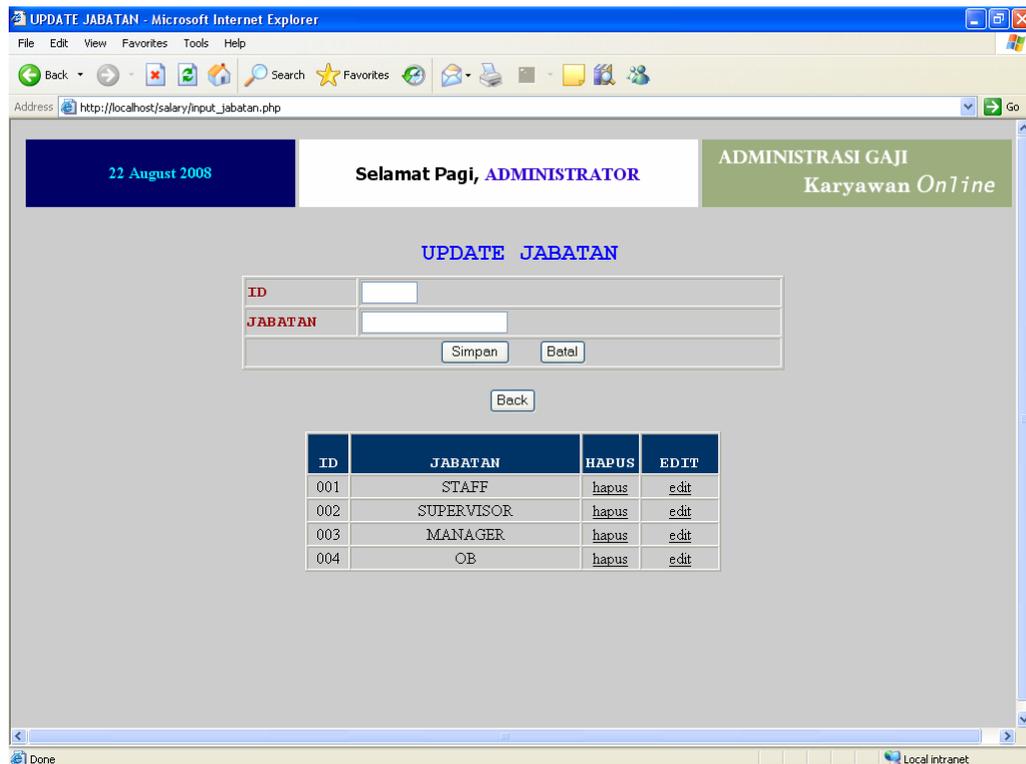
Gambar 4.13 merupakan gambar yang menunjukkan hasil pengujian pada gaji karyawan.



Gambar 4.13 Hasil Pengujian gaji karyawan

Pada halaman gaji karyawan, , pengujian dilakukan dengan kelompok uji normal yang berarti kelompok uji memasukan data benar. Kelompok uji melakukan penambahan data baru dengan menggunakan data id= 004, jabatan=OB, gaji pokok=1000000, tunjangan jabatan=100000, uang absen=7500, dan uang lembur/jam=3700, sistem akan menyimpan data tersebut dan menampilkanya dilayar halaman tabel gaji karyawan, Pada halaman ini terdapat tiga buah tombol. Tombol simpan berguna untuk menyimpan data baru yang dimasukan ke dalam basis data jika data gaji yang dimasukan telah benar, tombol reset berguna untuk mengembalikan kolom ke kondisi awal dan tombol back berfungsi untuk kembali ke halaman sebelumnya.seperti yang terlihat pada Gambar 4.13.

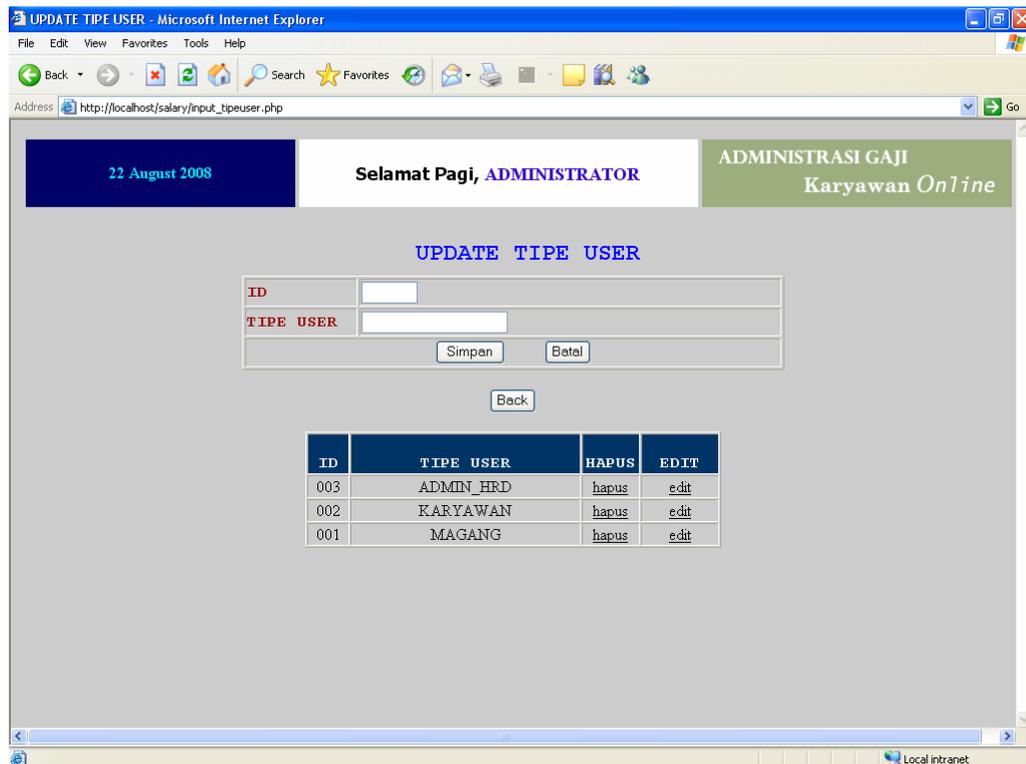
Gambar 4.14 merupakan gambar yang menunjukkan hasil pengujian pada halaman Update jabatan.



Gambar 4.14 Hasil Pengujian Halaman update jabatan.

Pada halaman update jabatan, pengujian dilakukan dengan kelompok uji normal yang berarti kelompok uji memasukkan data benar. Kelompok uji melakukan penambahan data baru dengan menggunakan data id= 004 dan jabatan=OB, sistem akan menyimpan data tersebut dan menampilkannya di tabel layar halaman update jabatan, pada halaman ini terdapat tiga buah tombol. Tombol simpan berguna untuk menyimpan data baru yang dimasukkan ke dalam basis data jika data jabatan yang dimasukkan telah benar, tombol batal berguna untuk mengembalikan kolom ke kondisi awal dan tombol back berfungsi untuk kembali ke halaman sebelumnya. seperti yang terlihat pada Gambar 4.14.

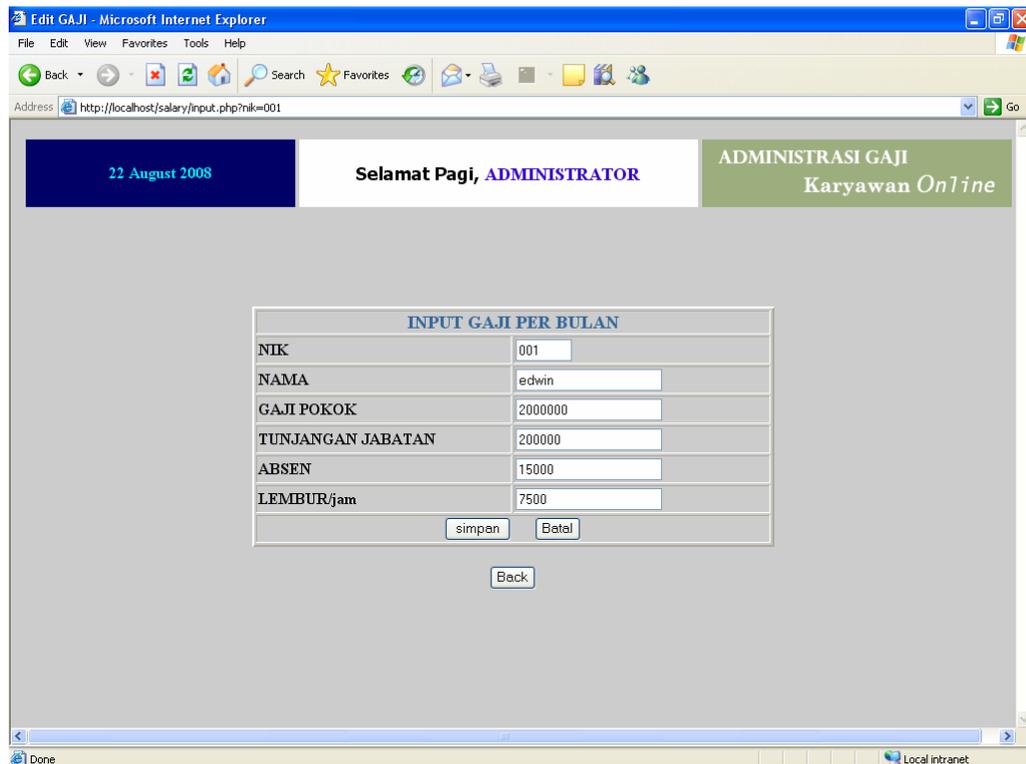
Gambar 4.15 merupakan gambar yang menunjukkan hasil pengujian pada halaman Update tipeuser.



Gambar 4.15 Hasil Pengujian Halaman Update Tipe User

Pada halaman update *tipeuser*, pengujian dilakukan dengan kelompok uji normal yang berarti kelompok uji memasukkan data benar. Kelompok uji melakukan penambahan data baru dengan menggunakan data id= 001 dan *tipeuser*=MAGANG, sistem akan menyimpan data tersebut dan menampilkanya di tabel layar halaman update *tipeuser*, pada halaman ini terdapat tiga buah tombol. Tombol simpan berguna untuk menyimpan data baru yang dimasukan ke dalam basis data jika data tipeuser yang dimasukan telah benar, tombol batal berguna untuk mengembalikan kolom ke kondisi awal dan tombol back berfungsi untuk kembali ke halaman sebelumnya.seperti yang terlihat pada Gambar 4.15.

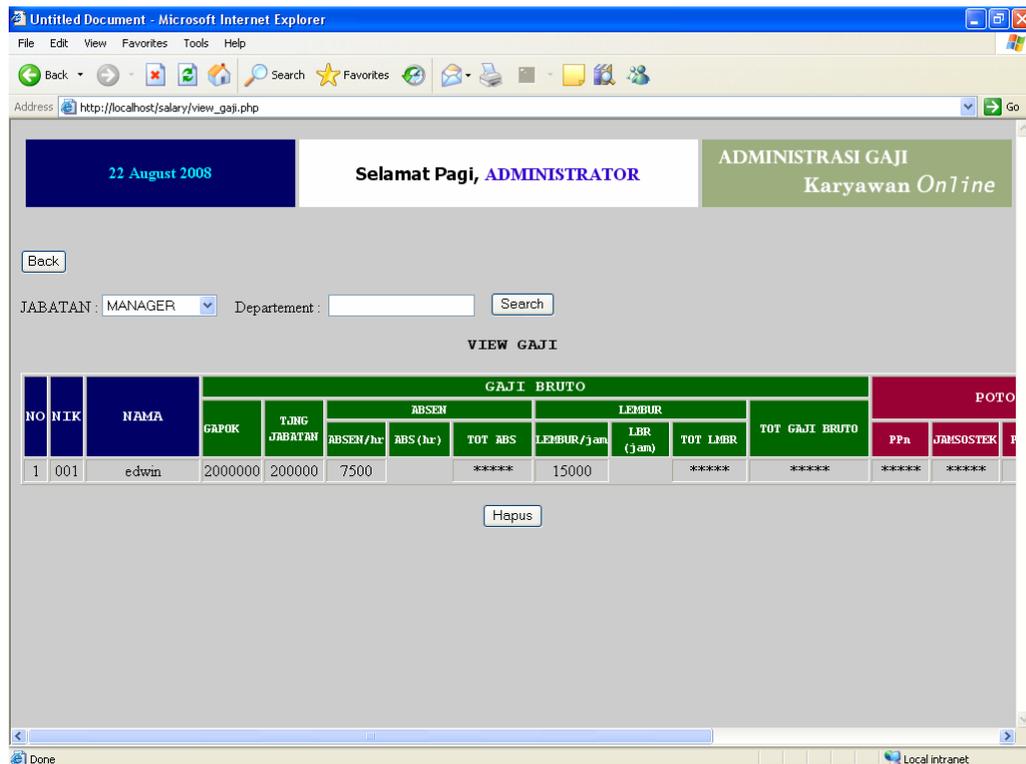
Gambar 4.16 merupakan gambar yang menunjukkan hasil pengujian pada halaman gaji bulanan karyawan.



Gambar 4.16 Hasil Pengujian Halaman Input gaji bulanan

Pada halaman input gaji bulanan, pengujian dilakukan dengan menampilkan data gaji bulanan keseluruhan, data ini akan ditampilkan secara otomatis oleh sistem apabila kita memilih nama pada tabel karyawan di halaman utama input gaji. Dalam hal ini tim penguji memilih nama karyawan “edwin” dan merubah data gaji pokok =2000000 menjadi gaji pokok=1500000. Sistem akan menampilkan perubahan data gaji tersebut pada halaman menu lihat gaji karyawan. Pada halaman ini juga terdapat tombol untuk menyimpan data dan membatalkan proses perubahan data.

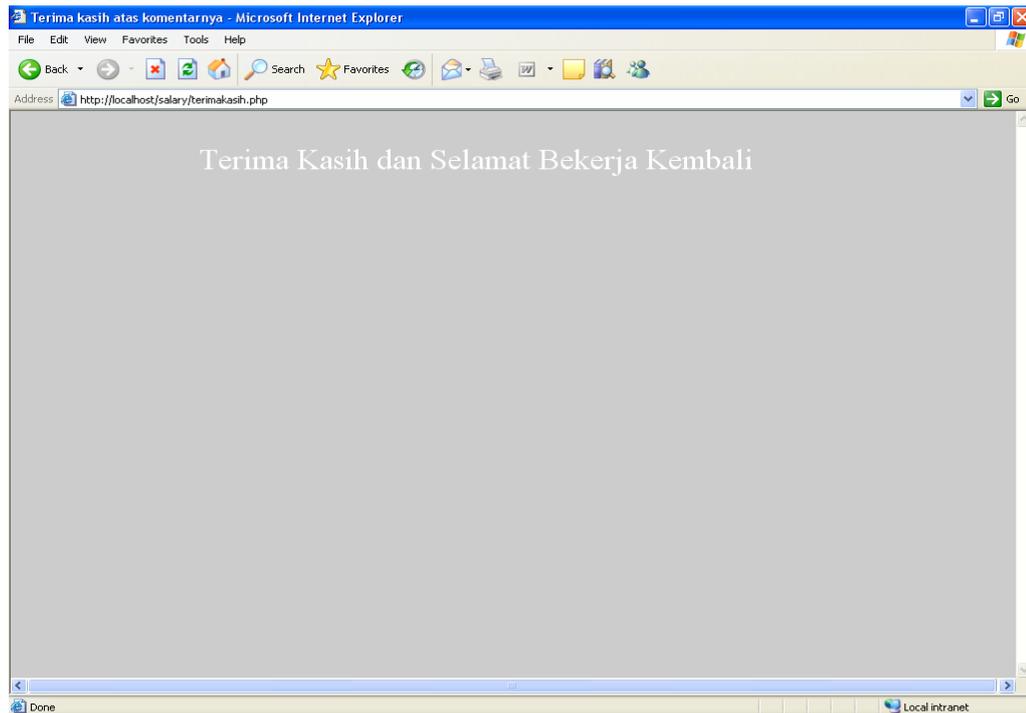
Gambar 4.17 merupakan gambar yang menunjukkan hasil pengujian pada halaman data peserta.



Gambar 4.17 Hasil Pengujian Halaman Lihat Gaji Karyawan

Pada halaman lihat gaji peserta, pengujian dilakukan dengan kelompok uji benar yang berarti dijalankan dengan semestinya. Pada halaman ini digunakan oleh administrator untuk melihat data gaji karyawan dan juga digunakan oleh administrator untuk merubah daata apabila terjadi kesalahan dan menghapus data karyawan yang tidak diperlukan . Untuk menghapus data karyawan, admin memilih tombol hapus hapus, setelah admin memilih tombol hapus sistem akan memproses dengan menghapus data karyawan yang terdapat didalam basis data berdasarkan nik karyawan.

Gambar 4.18 merupakan gambar yang menunjukkan hasil pengujian pada halaman logout



Gambar 4.18 Hasil Pengujian Proses Logout

Untuk menghindari penyalahgunaan, setelah selesai pada halaman admin ini wajib untuk melakukan logout.

Pengujian yang dilakukan dari halaman login hingga halaman logout menghasilkan proses keluaran sesuai dengan yang dirancang. Secara fungsionalitas program, program aplikasi penggajian karyawan berbasis web ini tidak terdapat kesalahan dan hasil program sesuai dengan hasil yang telah diproses didalam sistem. Hasil analisis yang diperoleh menghasilkan rancangan aplikasi penggajian karyawan yang telah dibangun secara keseluruhan berfungsi baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Ada beberapa hal yang dapat disimpulkan dari uraian pada bab I sampai dengan bab IV. Kesimpulan yang dapat diambil antara lain:

1. Setelah dilakukan beberapa kali pengujian, dapat diketahui bahwa aplikasi penggajian karyawan ini dapat membatasi akses dalam pemakaian aplikasi berdasarkan *tipe user* masing-masing pengguna di perusahaan sesuai ketentuan perusahaan.
2. Pengembangan aplikasi penggajian karyawan ini membutuhkan sebuah basis data dan delapan buah tabel yang terdiri dari `tb_administrator`, `tb_elm_gaji`, `tb_gaji`, `tb_jabatan`, `tb_absen`, `tb_lembur`, `tb_potongan` dan `tb_tipeuser`. Semua tabel berguna untuk menyimpan semua data yang diperlukan. Diantara tabel-tabel tersebut terdapat tabel utama, seperti: `tb_administrator`, `tb_elm_gaji` dan `tb_tipeuser`. Tabel utama tersebut mempunyai fungsi utama dalam aplikasi ini, antara lain sebagai penentu tingkatan akses, menentukan nilai gaji karyawan dan menyimpan data karyawan.
3. Aplikasi penggajian yang dibangun telah hampir memenuhi tujuan yang penulis harapkan, yang dalam hal ini perincian nilai gaji seorang karyawan

dalam satu bulan dan penghitungan gaji yang lebih cepat dan efisien bagi perusahaan.

5.2 Saran

Tugas Akhir yang dilakukan oleh penulis masih jauh dari sempurna dikarenakan adanya batasan waktu, untuk itu penulis memberikan saran-saran agar dapat dikembangkan, yaitu:

1. Agar informasi yang ditampilkan lebih menarik, aplikasi ini dapat ditambahkan tampilan gambar atau animasi yang atraktif dan sesuai dengan fungsi dari aplikasi dan penambahan fasilitas-fasilitas online seperti informasi berita dan kegiatan perusahaan.
2. Jangan terlalu banyak menggunakan pengkodean dalam satu halaman, karena dapat memperlambat dalam perbaikan kesalahan. Penulis sarankan untuk menggunakan poses pemanggilan dan *session* dan jangan menggunakan gambar dengan ukuran file yang sangat besar, karena dapat memperlambat proses pemanggilan web tersebut.
3. Aplikasi ini merupakan langkah awal dalam rangka mengembangkan aplikasi pengelola keuangan perusahaan dan data karyawan perusahaan yang lebih efisien, penulis mengharapkan aplikasi ini dapat dikembangkan agar nantinya menjadi lebih sempurna.

DAFTAR PUSTAKA

Fathansyah, Ir. 1999. *Basis Data*. Bandung: Informatika.

Wahana Komputer, Tim. 2003. *Memahami Model Enkripsi dan Security Data*.
Yogyakarta: Andi.

Kurniawan, Yahya. 2002. *Aplikasi Web Database Dengan PHP Dan MySQL*.
Jakarta: Elex Media Komputindo.

Sutarman, Skom. 2003. *Membangun Aplikasi Web dengan PHP dan MySQL*.
Yogyakarta: Graha Ilmu.

http://en.wikipedia.org/wiki/Unified_Modeling_Language, Mei 2008

LAMPIRAN 1 LISTING PROGRAM

1. Listing program antarmuka login

```
<html><head>
<title>Login Administrator</title>
</head>
<body bgcolor="#CCCCCC">
<form action="cek1.php"
method="post"
enctype="multipart/form-data" >
  <p><br>
  <center>
  <?
$get=getdate();
$bln=$get['month'];
$tgl=$get['mday'];
$th=$get['year'];
$jam=$get['hours'];
if($jam<=11)
{echo "<font size='+1'
face='tahoma'><b> Selamat Pagi,
</b></font>";}
elseif($jam>11 and $jam <=15)
{echo "<font size='+1'
face='tahoma'><b> Selamat Siang,
</b></font>";}
elseif($jam>15 and $jam <=18)
{echo "<font size='+1'
face='tahoma'><b> Selamat Sore,
</b></font>";}
elseif($jam >18)
{echo "<font size='+1'
face='tahoma'><b> Selamat Malam,
</b></font>";}
echo " <font size='+1'><b>$tgl
$bln $th</b></font> ";
?>
</center><div align="left"><font
size="2"><font face="Times,
serif">USERNAME
strong></font></div>      </td>
<input type="text" name="aman">
</tr>
<div align="left"><font
size="2"><font face="Times,
serif">  PASSWORD</font></div>
<input name="login"
type="password"></font></td></tr>
face="Times New Roman, Times,
serif">
<input type="submit" name="Login"
value="Login">
</font></strong></font></p></div>
</body></html>
<?
if($Login)
{
if($aman =='admin' && $login
=='111')
{
$aman="admin";
$tipe="ADMINISTRATOR";
session_start();
session_register("nama","tipe");
echo
"<script>document.location=\"menu_
update_admin.php\"</script>";
exit();
}
else
{
$nama=$aman;
$psw=$login;
include "KONEKSI.php";
$perintah="select * from
ADMINISTRATOR where nama= '$nama'
and password = '$login' ";
$hasil=mysql_query($perintah) or
die(mysql_error());
$data=mysql_fetch_array($hasil);
if ($data)
{
$nama="$data[nama]";
$tipe="$data[tipeuser]";
session_start();
session_register('nama','tipe');
echo
"<script>document.location=\"cek2.
php\"</script>";
exit();
}
else
{echo "<script>alert(\"Anda Tidak
Punya Akses..!\");history.go(-
1)</script>";
exit();}
}
}
?>
```

2. Listing program antarmuka menu utama administrator

```
<?
session_start();
if(!(session_is_registered("tipe")
))
{
echo "<script>alert(\"you're not
licensed to open this
Page!\");history.go(-1)</script>";
exit;
}
?>
```

```

<html><head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-
8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
#Layer1 {
position:absolute;
left:10px;
top:68px;
width:155px;
height:31px;
z-index:1;
background-color: #999999;
}
-->
</style></head>
<body bgcolor="#CCCCCC">
<center><? include "top.php"?>
</center>
<table width="882" border="0">
<tr><td width="195">&nbsp;&nbsp;&nbsp;</td>
<td width="146"><form name="form1"
method="post"
action="table_edit_admin.php">
<label></label>
<input type="submit"
name="Submit2" value="Update Data
Karyawan">
</form></td>
<td width="6">&nbsp;&nbsp;&nbsp;</td>
<td width="131"><form name="form2"
method="post"
action="elm_gaji.php">
<label>
<input type="submit" name="Submit"
value="Update Elemen Gaji">
</label>
</form></td>
<td width="3"> </td>
<td width="132"><form name="form5"
method="post"
action="input_gaji.php">
<label>
<input type="submit"
name="Submit5" value="Input Gaji
Bulanan">
</label> </form></td>
<td width="199"><form name="form4"
method="post"
action="view_gaji.php">
<label></label>
<input type="submit"
name="Submit4" value="Lihat Gaji
Karyawan">
</form></td>
<td width="36">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
<form name="form3" method="post"
action="logout.php">
<label>
<center><input type="submit"
name="Submit3"
value="Logout"></center>
</label></form></body></html>

```

3. Listing program antarmuka update jabatan

```

<?
session_start();
if(!(session_is_registered("tipe")
))
{
    echo "<script>alert(\"Level
Anda Bukan Top
Admin\");history.go(-1)</script>";
    exit;
}
?>
<html><head><title>UPDATE
JABATAN</title>
<script language="JavaScript"
type="text/JavaScript">
<!--
function MM_findObj(n, d) {
//v4.01
    var p,i,x; if(!d) d=document;
if((p=n.indexOf("?"))>0&&parent.fr
ames.length) {
d=parent.frames[n.substring(p+1)].
document; n=n.substring(0,p);}
        if(!(x=d[n])&&d.all) x=d.all[n];
for (i=0;!x&&i<d.forms.length;i++)
x=d.forms[i][n];

for(i=0;!x&&d.layers&&i<d.layers.l
ength;i++)
x=MM_findObj(n,d.layers[i].documen
t);
    if(!x && d.getElementById)
x=d.getElementById(n); return x;
}
function MM_validateForm() {
//v4.0
    var
i,p,q,nm,test,num,min,max,errors='
',args=MM_validateForm.arguments;
    for (i=0; i<(args.length-2);
i+=3) { test=args[i+2];
val=MM_findObj(args[i]);
        if (val) { nm=val.name; if
((val=val.value)!="") {
            if
(test.indexOf('isEmail')!=-1) {
p=val.indexOf('@');

```

```

        if (p<1 || p==(val.length-
1)) errors+='- '+nm+' must contain
an e-mail address.\n';
    } else if (test!='R') { num
= parseFloat(val);
        if (isNaN(val)) errors+='-
'+nm+' must contain a number.\n';
        if
(test.indexOf('inRange') != -1) {
p=test.indexOf(':');
            min=test.substring(8,p);
max=test.substring(p+1);
            if (num<min || max<num)
errors+='- '+nm+' must contain a
number between '+min+' and
'+max+'.\n';
        } } } else if (test.charAt(0)
== 'R') errors += '- '+nm+' is
required.\n'; }
    } if (errors) alert('Nama dan
Password harus diisi');
    document.MM_returnValue =
(errors == '');
}
//-->
</script></head><body
bgcolor="#CCCCCC">
<p><center><? include
"top.php"?></center>
<form name="form1" method="post"
action="new_jabatan.php"
enctype="multipart/form-data">
<table width="55%" align="center"
height="45">
<tr>
<td width="2%"><div
align="center"></div></td>
<td width="85%">
<div align="center"><font
color="#000066" size="5"
face="Courier New, Courier,
monospace">
<strong>
<marquee scrollldelay=200>
<font color="#0000FF">UPDATE
JABATAN</font>
</marquee>
</strong></font></div></td>
<td width="13%">&nbsp;</td>
</tr></table>
<table width="55%" border="1"
cellpadding="1" align="center">
<tr><td width="21%"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">ID</font></strong></td>
<td width="79%">
<input type="text" name="id"
maxlength="4" size="5"></td></tr>
<tr>
<td width="21%"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">JABATAN</font></strong>
</td>
<td>
<input type="text" name="jbtn"
size="20"
maxlength="15"></td></tr>
<tr>
<td colspan="2"><div
align="center"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">
<input name="tambah" type="submit"
onClick="MM_validateForm('user','
','R');return
document.MM_returnValue"
value="Simpan">
<input name="reset" type="reset"
value="Batal">
</font></strong></div></td></tr>
</table></form>
<form name="form2" method="post"
action="table_edit_admin.php">
<label>
<center><input type="submit"
name="Submit"
value="Back"></center>
</label>
</form><form name="form3"
method="post" action="">
<?
include "KONEKSI.php";
$perintah="SELECT * FROM jabatan
order by id";
$hasil=mysql_query($perintah);
$data=mysql_fetch_array($hasil);
if (empty($data))
{
echo("<font size=3
color=#000000><center><b>DATA
MASIH
KOSONG</center></b></font>");
}
else
{
?>
<table width="415" border="1"
align="center" cellpadding="1"
cellspacing="1">
<!--DWLayoutTable-->
<tr>
<td width="40" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">ID</font></strong></fon
t></div></td>
<td width="240" align="center"
valign="center"

```

```

bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">JABATAN</font></strong>
</font></div></td>
<td width="54" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">HAPUS</font></strong></f
ont></div></td>
<td width="77" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">EDIT</font></strong></f
ont></div></td></tr>
<?
include "koneksi.php";

```

```

$scari="SELECT * FROM jabatan order
by id";
$hasil=mysql_query($scari);
while($data=mysql_fetch_array($has
il))
{
echo "<tr>";
echo("<td align=center><font
color=#000000>$data[id]</font></td
>");
echo strtoupper("<td
align=center><font
color=#000000>$data[jabatan]</font
></td>");
echo("<td align=center><a
href=hapus_jbtn.php?id=$data[id]><
font
color=#000000>hapus</font></a></td
>");
echo("<td align=center><a
href=edit_jbtn.php?id=$data[id]><f
ont
color=#000000>edit</font></a></td>
");
echo("</tr>");
}
}
?>
</table></form></body></html>

```

4. Listing program antarmuka update tipeuser

```

<?
session_start();
if(!(session_is_registered("tipe")
))
{echo "<script>alert(\"Level Anda
Bukan Top Admin\");history.go(-
1)</script>";
exit; }
?>
<html><head><title>UPDATE TIPE
USER</title>
<script language="JavaScript"
type="text/JavaScript">
<!--
function MM_findObj(n, d) {
//v4.01
var p,i,x; if(!d) d=document;
if((p=n.indexOf("?"))>0&&parent.fr
ames.length) {
d=parent.frames[n.substring(p+1)].
document; n=n.substring(0,p);}
if(!(x=d[n])&&d.all) x=d.all[n];
for (i=0;!x&&i<d.forms.length;i++)
x=d.forms[i][n];
for(i=0;!x&&d.layers&&i<d.layers.l
ength;i++)
x=MM_findObj(n,d.layers[i].documen
t);
if(!x && d.getElementById)
x=d.getElementById(n); return x; }

```

```

function MM_validateForm() {
//v4.0
var
i,p,q,nm,test,num,min,max,errors='
',args=MM_validateForm.arguments;
for (i=0; i<(args.length-2); i+=3)
{ test=args[i+2];
val=MM_findObj(args[i]);
if (val) { nm=val.name; if
((val=val.value)!="") {
if (test.indexOf('isEmail')!=-1) {
p=val.indexOf('@');
if (p<1 || p==(val.length-1))
errors+='- '+nm+' must contain an
e-mail address.\n';
} else if (test!='R') { num =
parseFloat(val);
if (isNaN(val)) errors+='- '+nm+'
must contain a number.\n';
if (test.indexOf('inRange') != -1)
{ p=test.indexOf(':');
min=test.substring(8,p);
max=test.substring(p+1);
if (num<min || max<num) errors+='-
'+nm+' must contain a number
between '+min+' and '+max+'.\n';
} } } else if (test.charAt(0) ==
'R') errors += '- '+nm+' is
required.\n'; }

```

```

} if (errors) alert('Nama dan
Password harus diisi');
document.MM_returnValue = (errors
== ''); }
//-->
</script></head>
<body bgcolor="#CCCCCC">
<p><center><? include
"top.php"?></center>
<form name="form1" method="post"
action="new_tipeuser.php"
enctype="multipart/form-data">
<table width="55%" align="center"
height="45">
<tr><td width="2%"><div
align="center"></div></td>
<td width="85%">
<div align="center"><font
color="#000066" size="5"
face="Courier New, Courier,
monospace">
<strong><marquee scrolldelay=200>
<font color="#0000FF">UPDATE TIPE
USER</font>
</marquee></strong></font></div></td>
<td
width="13%">&nbsp;</td></tr></tabl
e>
<table width="55%" border="1"
cellpadding="1" align="center">
<tr><td width="21%"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">ID</font></strong></td>
<td width="79%">
<input type="text" name="id"
maxlength="4" size="5"></td></tr>
<tr>
<td width="21%"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">TIPE
USER</font></strong></td>
<td><input type="text" name="tipe"
size="20"
maxlength="15"></td></tr>
<tr>
<td colspan="2"><div
align="center"><strong><font
color="#990000" size="3"
face="Courier New, Courier,
monospace">
<input name="tambah" type="submit"
onClick="MM_validateForm('user','
','R');return
document.MM_returnValue"
value="Simpan">
<input name="reset" type="reset"
value="Batal">
</font></strong></div></td></tr>
</table></form>
<form name="form2" method="post"
action="table_edit_admin.php">
<label>
<center><input type="submit"
name="Submit"
value="Back"></center>
</label></form>
<form name="form3" method="post"
action="">
<?
include "KONEKSI.php";
$perintah="SELECT * FROM TIPEUSER
order by TIPEUSER";
$hasil=mysql_query($perintah);
$data=mysql_fetch_array($hasil);
if (empty($data))
{
echo("<font size=3
color=#000000><center><b>DATA
MASIH
KOSONG</center></b></font>");
}
else
{
?>
<table width="415" border="1"
align="center" cellpadding="1"
cellspacing="1">
<!--DWLayoutTable-->
<tr>
<td width="40" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">ID</font></strong></fon
t></div></td>
<td width="240" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier, monospace">TIPE
USER</font></strong></font></div><
/td>
<td width="54" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">HAPUS</font></strong></
font></div></td>
<td width="77" align="center"
valign="center"
bgcolor="#003366"><font
size="3">&nbsp;</font>
<div align="center"><font
size="3"><strong><font

```

```

color="#FFFFFF" face="Courier New,
Courier,
monospace">EDIT</font></strong></f
ont></div></td>
</tr>
<?
include "koneksi.php";
$cari="SELECT * FROM TIPEUSER
order by TIPEUSER";
$hasil=mysql_query($cari);
while($data=mysql_fetch_array($has
il))
{
echo "<tr>";
echo("<td align=center><font
color=#000000>$data[id]</font></td
>");
echo strtoupper("<td
align=center><font

```

```

color=#000000>$data[tipeuser]</fon
t></td>");
echo("<td align=center><a
href=hapus_tipe.php?id=$data[id]><f
ont
color=#000000>hapus</font></a></td
>");
echo("<td align=center><a
href=edit_tipe.php?id=$data[id]><f
ont
color=#000000>edit</font></a></td>
");
echo("</tr>");
}
}
?>
</table></form></body></html>

```

5. Listing program antarmuka input new user

```

<?
session_start();
if(!(session_is_registered("tipe")
))
{echo "<script>alert(\"Level Anda
Bukan Top Admin\");history.go(-
1)</script>";
exit; }
else
{if ($tipe!="ADMINISTRATOR" &&
$tipe!="HRD")
echo
"<script>document.location=\"edit_
admin.php\"</script>";exit;}
?>
<html><head><title>Tambah Admin
Baru</title>
<SCRIPT LANGUAGE = "JavaScript">
function validate ()
{var kode =
document.form.kode.value;
var ket = document.form.ket.value;
if ((kode == null) | (kode == "")
| (ket == null) | (ket == ""))
{
alert ("Field dengan tanda * harus
diisi !");
return false;
}
else
{return true; }
}
</SCRIPT></head><body
bgcolor="#CCCCCC">
<p><center><? include
"top.php";?></center>
<form name="form1" method="post"
action="NewAdmin.php"
enctype="multipart/form-data">
<table width="55%" align="center"
height="45">

```

```

<tr> <td width="2%"><div
align="center"></div></td>
<td width="85%">
<div align="center"><font
color="#000066" size="5"
face="Courier New, Courier,
monospace">
<strong>
<marquee scrollldelay=200>
<font color="#0000FF">TAMBAH
KARYAWAN </font>
</marquee>
</strong></font></div></td>
<td
width="13%">&nbsp;</td></tr></tabl
e>
<table width="47%" border="1"
cellpadding="1" align="center">
<tr>
<td><strong><font color="#990066"
face="Courier New, Courier,
monospace">NIK</font></strong></td>
<td><label>
<input type="text" name="nik"
maxlength="4" size="5">
</label></td></tr>
<tr>
<td width="21%"><strong><font
color="#990066" size="3"
face="Courier New, Courier,
monospace">NAMA</font></strong></t
d>
<td>
<strong><font color="#990066"
face="Courier New, Courier,
monospace">
<input type="text" name="nama"
maxlength="20" size="21">
</font></strong></td></tr>
<tr>

```

```

<td width="21%"><strong><font
color="#990066" size="3"
face="Courier New, Courier,
monospace">DEPARTEMEN</font></stro
ng></td>
<td>
<strong><font color="#990066"
face="Courier New, Courier,
monospace">
<input type="text" name="dept"
size="16" maxlength="15">
</font></strong></td></tr>
<tr>
<td><strong><font color="#990066"
size="3" face="Courier New,
Courier,
monospace">JABATAN</font></strong>
</td>
<td><strong><font color="#990066"
face="Courier New, Courier,
monospace">
<label></label>
<label>
<?
include "KONEKSI.php";
$cari="SELECT * FROM jabatan order
by id";
$hasil=mysql_query($cari);
echo "<select name=jbbtn>";
while($data=mysql_fetch_array($has
il))
{
echo strtoupper("<option
value=$data[jabatan]>$data[jabatan
]</option>");
}
echo "</select>";
?>
</label>
</font></strong></td></tr>
<tr>
<td width="21%"><strong><font
color="#990066" size="3"
face="Courier New, Courier,
monospace">PASSWORD</font></strong
></td>
<td><strong><font color="#990066"
face="Courier New, Courier,
monospace">
<input type="password" name="pswd"
size="9" maxlength="8">
</font></strong></td></tr>
<tr>
<td><strong><font color="#990066"
size="3" face="Courier New,
Courier,
monospace">TIPE
USER</font></strong></td>
<td><strong><font color="#990066"
face="Courier New, Courier,
monospace">
<?
include "KONEKSI.php";
$cari="SELECT * FROM tipeuser
order by id";
$hasil=mysql_query($cari);
echo "<select name=tipe>";
while($data=mysql_fetch_array($has
il))
{
echo strtoupper("<option
value=$data[tipeuser]>$data[tipeus
er]</option>");
}
echo "</select>";
?>
</font></strong></td></tr>
<tr>
<td><strong><font color="#990066"
face="Courier New, Courier,
monospace">FOTO</font></strong></td>
<td><div align="left">
<label>
<input type="file" name="foto">
</label></div>
<div align="left"></div></td></tr>
<tr>
<td colspan="2"><div
align="center"><strong><font
color="#990066"><font
face="Courier New, Courier,
monospace"><font size="3">
<input name="tambah" type="submit"
onClick="MM_validateForm('user','
','R');return
document.MM_returnValue"
value="Simpan">
<input name="reset" type="reset"
value="Batal">
</font></font></font></strong></di
v></td>
</tr></table></form>
<form name="form2" method="post"
action="table_edit_admin.php">
<label>
<center><input type="submit"
name="Submit"
value="Back"></center>
</label></form></body></html>

```

6. Listing program antarmuka buku tamu

```

<?
session_start();
if (!session_is_registered("nama")
&& !session_is_registered("pass"))
{echo "Anda Tidak Punya akses!!!";
echo "<br>";echo
" <td><b><a href=menu_update_data_ne
w.php>Back</a> </b></td>";
exit;
}
?>
<html><head>

```

```

<title>Buku Tamu</title>
</head>
<body background="newmark1.jpg">
<center><div align="center"><font
color="#CC6600"><strong><font
color="#00FFFF" size="5">BuKu Tamu
</font></strong></font></div>
<table width="100%" height="32"
border="1" align="center"
cellpadding="1"
bordercolor="#999999">
<!--DWLayoutTable-->
<tr bgcolor="#00FFFF">
<td width="163" height="26"
align="left" valign="top">
<div align="center"><font
color="#FF0000"><b>Nama</b></font>
</div></td>
<td width="200" align="left"
valign="top">
<div align="center"><font
color="#FF0000"><b>E-
Mail</b></font></div></td>
<td width="509" align="left"
valign="top"><div
align="center"><font
color="#FF0000"><b>Komentar</b></f
ont></div></td>
<td width="86" align="left"
valign="top">
<div align="center"><font
color="#FF0000"><b>hapus</b></font
></div></td>
</tr>
<?
mysql_connect("localhost","","");
mysql_select_db("indri");
$cari="SELECT * FROM bukutamu
order by right(nama,length(nama)-
1)";
$hasil=mysql_query($cari);
while($data=mysql_fetch_array($has
il))
{echo "<tr>";
echo("<tdalign=left><font
color=#ffffff>$data[nama]</font></
td>");
echo("<tdalign=left><font
color=#ffffff>$data[email]</font><<
/td>");
echo("<tdalign=left><font
color=#ffffff>$data[komentar]</fon
t></td>");
echo("<td align=left><a
href=hapus_bukutamu.php?nama=$data
[nama]><font
color=#ffffff><b><center>hapus</ce
nter></font></a></td>");
echo("</tr>");}
?>
</table></center></body></html>

```

7. Listing program antarmuka update elemen gaji

```

<?
session_start();
if(!(session_is_registered("tipe")
))
{
    echo "<script>alert(\"you're
not licensed to open this
Page!\");history.go(-1)</script>";
    exit;
}
?>
<html><head>
<title>Untitled Document</title>
<style type="text/css">
<!--
.style22 {font-size: 12px; font-
weight: bold; }
.style31 {font-family: "Courier
New", Courier, monospace; font-
size: 16px; font-weight: bold;
color: #990066; }
-->
</style></head>
<body bgcolor="#CCCCCC">
<center>
<p>
<? include "top.php";?>
<?
include "KONEKSI.php";
$perintah="SELECT * FROM elmgaji
order by id";
$hasil=mysql_query($perintah);
$data=mysql_fetch_array($hasil);
?>
<form name="form2" method="post"
action="cek2.php">
<input type="submit"
name="Submit3" value="Back">
</form>
<form name="form1" method="post"
action="new_gaji.php">
<hr>
<table width="346" border="1">
<tr>
<td width="183"><span
class="style31">ID</span></td>
<td width="101">
<input type="text" name="id"
maxlength="3" size="4"> </td>
</tr>
<tr>
<td><span
class="style31">JABATAN</span></td>
<td><?
include "KONEKSI.php";
$cari="SELECT * FROM jabatan order
by id";

```

```

$hasil=mysql_query($cari);
echo"<select name=jbtn>";
while($data=mysql_fetch_array($hasil))
{
echo strtoupper("<option
value=$data[jabatan]>$data[jabatan
]</option>");
}
echo"</select>";
?></td>
</tr>
<tr>
<td><span class="style31">GAJI
POKOK </span></td>
<td><input type="text"
name="gapok" size="10"></td>
</tr>
<tr>
<td><span
class="style31">TUNJANGAN JABATAN
</span></td>
<td><input type="text" name="tjng"
size="6"></td>
</tr>
<tr>
<td><span
class="style31">ABSEN</span></td>
<td><input type="text"
name="absen" size="5"></td>
</tr>
<tr>
<td><span
class="style31">LEMBUR/ jam</span><
/td>
<td><input type="text" name="lmbur"
size="6" maxlength="5"></td>
</tr>
<tr>
<td colspan="2">
<center class="style31"><input
type="submit" name="Submit"
value="Simpan">
<input type="Reset" name="Submit2"
value="Reset">
</center></td></tr>
</table><hr>
<?
include "KONEKSI.php";
$perintah="SELECT * FROM elmgaji
order by jabatan";
$hasil=mysql_query($perintah);
$data=mysql_fetch_array($hasil);
if (empty($data))
{
echo("<font size=3
color=#000000><center><b>DATA
MASIH
KOSONG</center></b></font>");
}
else
{
?>
<table width="715" border="1"
align="center" cellpadding="1"
cellspacing="1">
<!--DWLayoutTable-->
<tr>
<td width="51" align="center"
valign="center" bgcolor="#003366">
<div align="center"
class="style22"><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">ID</font></div></td>
<td width="126" align="center"
valign="center" bgcolor="#003366">
<div align="center"
class="style22"><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">JABATAN</font></div></td>
<td width="171" align="center"
valign="center" bgcolor="#003366">
<div align="center"
class="style22"><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">GAPOK</font></div></td>
<td width="130" align="center"
valign="center" bgcolor="#003366">
<div align="center"
class="style22"><font
color="#FFFFFF" face="Courier New,
Courier, monospace">TJNG
JABATAN</font></div></td>
<td width="92" align="center"
valign="center" bgcolor="#003366">
<div align="center"
class="style22"><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">ABSEN</font></div></td>
<td width="60" align="center"
valign="center" bgcolor="#003366">
<div align="center"
class="style22"><font
color="#FFFFFF" face="Courier New,
Courier, monospace">LEMBUR/ jam</font>
</div></td>
<td width="60" align="center"
valign="center" bgcolor="#003366">
<div align="center"
class="style22"><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">HAPUS</font></div></td>
<td width="47" align="center"
valign="center" bgcolor="#003366">
<div align="center"
class="style22"><font
color="#FFFFFF" face="Courier New,
Courier,
monospace">EDIT</font></div></td>
</tr>

```

```

<?
include "koneksi.php";
$cari="SELECT * FROM elmgaji order
by id asc";
$hasil=mysql_query($cari);
while($data=mysql_fetch_array($has
il))
{
echo "<tr>";
echo("<td align=center><font
color=#000000>$data[id]</font></td
>");
echo strtoupper("<td
align=center><font
color=#000000>$data[jabatan]</font
></td>");
echo("<td align=center><font
color=#000000>$data[gapok]</font><
/td>");
echo("<td align=center><font
color=#000000>$data[tjng_jbtn]</fo
nt></td>");

echo("<td align=center><font
color=#000000>$data[absen]</font><
/td>");
echo("<td align=center><font
color=#000000>$data[lbrperjam]</fo
nt></td>");
echo("<td align=center><a
href=hapus_gaji.php?id=$data[id]><
font
color=#000000>hapus</font></a></td
>");
echo("<td align=center><a
href=edit_gaji.php?id=$data[id]><f
ont
color=#000000>edit</font></a></td>
");
echo("</tr>");
}
}
?>
</table></p></form></center></body
></html>

```

8. Listing program antarmuka view gaji

```

<?
session_start();
if(!(session_is_registered("tipe")
))
{
echo "<script>alert(\"you're not
licensed to open this
Page!\");history.go(-1)</script>";
exit;
}
?>
<html><head><title>Untitled
Document</title>
<style type="text/css">
<!--
.style6 {font-family: "Courier
New", Courier, monospace}
.style16 {color: #FFFFFF; font-
size: 12px; font-weight: bold; }
.style18 {font-family: "Courier
New", Courier, monospace; font-
size: 16px; }
.style19 {font-size: 16}
.style21 {color: #FFFFFF; font-
size: 16; font-weight: bold; }
.style22 {color: #FFFFFF}
.style24 {font-size: 12px; font-
weight: bold; }
-->
</style>
</head>
<body bgcolor="#CCCCCC">
<center>
<p>
<? include "top.php"?></p>

<form name="form1" method="post"
action="cek2.php">
<div align="left"><input
type="submit" name="Submit"
value="Back">
</div></form>
<div align="left">
<form name="form2" method="post"
action="">
<label></label>
JABATAN :
<strong><font color="#990066"
face="Courier New, Courier,
monospace">
<?
include "KONEKSI.php";
$cari="SELECT * FROM jabatan order
by jabatan";
$hasil=mysql_query($cari);
echo"<select name=jbtn>";
while($data=mysql_fetch_array($has
il))
{
echo strtoupper("<option
value=$data[jabatan]>$data[jabatan
]</option>");
}
echo"</select>";
?>
</font></strong>
<label>
Departement : <input type="text"
name="departemen"></label>
<label>

```

```

<input type="submit"
name="Submit2" value="Search">
</label></form></div></center><cen
ter>
<?
include "KONEKSI.php";
$perintah="SELECT * FROM gaji
order by nik";
$hasil=mysql_query($perintah);
$data=mysql_fetch_array($hasil);
if (empty($data))
{
echo "<br>";
echo "<br>";
echo("<font size=3
color=#000000><center><b>DATA
MASIH
KOSONG</center></b></font>");
}
else
{
?>
<center><span class="style6
style1"><strong>VIEW
GAJI</strong></span></center>
</center>
<?
echo "<form action=hapus_data.php
method=post>";
?>
<center>
<table width="1399" border="1">
<tr>
<td width="20" rowspan="3"
bgcolor="#000066"><div
align="center" class="style16
style19"><span
class="style6">NO</span></div></td>
<td width="30" rowspan="3"
bgcolor="#000066"><div
align="center"
class="style21"><span
class="style6">NIK</span></div></t
d>
<td width="114" rowspan="3"><div
bgcolor="#000066"><div
align="center"
class="style21"><span
class="style6">NAMA</span></div></
td>
<td colspan="9"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style18">GAJI
BRUTO</span></div></td>
<td colspan="4" rowspan="2"
bgcolor="#990033"><div
align="center"
class="style16"><span
class="style18">POTONGAN
GAJI</span></div></td>
<td width="110" rowspan="3"
bgcolor="#000000"><div
align="center"
class="style16"><span
class="style6">TOT
GAJI</span></div></td>
<td width="45" rowspan="3"
bgcolor="#000099"><div
align="center"
class="style24"><span
class="style22">HAPUS</span></div>
</td>
<td width="39" rowspan="3"
bgcolor="#000099"><div
align="center"
class="style24"><span
class="style22">EDIT</span></div><
/td>
</tr>
<tr>
<td width="35" rowspan="2"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">GAPOK</span></div><
/td>
<td width="56" rowspan="2"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">TJNG JABATAN
</span></div></td>
<td colspan="3"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">ABSEN</span></div><
/td>
<td colspan="3"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">LEMBUR</span></div>
</td>
<td width="120" rowspan="2"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">TOT GAJI BRUTO
</span></div></td>
</tr>
<tr>
<td width="56"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">ABSEN/hr</span></di
v></td>
<td width="59"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">ABS(hr)</span></div
></td>

```

```

<td width="78"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">TOT ABS
</span></div></td>
<td width="70"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">LEMBUR/ jam</span></div></td>
<td width="56"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">LBR( jam)</span></div></td>
<td width="72"
bgcolor="#006600"><div
align="center"
class="style16"><span
class="style6">TOT LMBR
</span></div></td>
<td width="55"
bgcolor="#990033"><div
align="center"
class="style16"><span
class="style6">PPn</span></div></td>
<td width="64"
bgcolor="#990033"><div
align="center"
class="style16"><span
class="style6">JAMSOSTEK</span></div></td>
<td width="107"
bgcolor="#990033"><div
align="center"
class="style16"><span
class="style6">PJMN KARYAWAN
</span></div></td>
<td width="95"
bgcolor="#990033"><div
align="center"
class="style16"><span
class="style6">TOT POTONGAN
</span></div></td>
</tr>
<?
include "koneksi.php";
$cari="SELECT * FROM gaji order by
nik asc";
$hasil=mysql_query($cari);
$no=1;
while($data=mysql_fetch_array($hasil))
{
echo "<tr>";
echo("<td align=center><font
color=#000000>$no</font></td>");
echo strtoupper("<td
align=center><font
color=#000000>$data[nik]</font></td>");
echo("<td align=center><font
color=#000000>$data[nama]</font></td>");
echo("<td align=center><font
color=#000000>$data[gapok]</font></td>");
echo("<td align=center><font
color=#000000>$data[tjng_jbtn]</font></td>");
echo("<td align=center><font
color=#000000>$data[absen]</font></td>");
$hari=$hr;
echo("<td align=center><font
color=#000000>$hari</font></td>");
echo("<td align=center><font
color=#000000>*****</font></td>");
echo("<td align=center><font
color=#000000>$data[lbrperjam]</font></td>");
echo("<td align=center><font
color=#000000>$jm</font></td>");
echo("<td align=center><font
color=#000000>*****</font></td>");
echo("<td align=center><a
href=hapus_data.php?nik=$data[nik]
><font
color=#000000>hapus</font></a></td>");
echo("<td align=center><a
href=edit_gaji.php?id=$data[id]><font
color=#000000>edit</font></a></td>");
echo("</tr>");
$no++;
}
?>
</table></center><p><center>
<?
echo "<input type=submit
value=Hapus>";
echo "</form>";
?>
</center></p></body></html>

```


DAFTAR RIWAYAT HIDUP

DATA PRIBADI

Nama : Edwin Mahendra Nataatmadja
Tempat Tgl. Lahir : Rangkas bitung, 7 April 1982
Jenis Kelamin : Laki laki
Status Perkawinan : Belum Nikah
Agama : Islam
Kewarganegaraan : Indonesia
Alamat : Jl. H. Gaim Rt 06. Rw 02 No25 f
Petukangan utara, jakarta selatan
Tlp. (021) 586 2125 / 081586417255

PENDIDIKAN

No.	Nama Sekolah	Jurusan	Tahun
1	SD Negeri 05 Pagi Jakarta	-	1988-1994
2	SMP Hang Tuah II Jakarta	-	1994-1997
3	SMUN 63 Jakarta	-	1997-2000
4	Universitas Mercu Buana, Jakarta	-	2001-2008

PENGALAMAN KEGIATAN AKADEMIK

No.	Kegiatan	Institusi	Tahun
1.	Praktikum Laboratorium : <ul style="list-style-type: none">• Komputer• MBEC	Lab. Universitas Mercu Buana (Sertifikat) Mercu Buana English Club	2001 2003

Daftar Riwayat Hidup ini dibuat dengan Sebenar-benarnya.

Jakarta, Agustus 2008

Hormat Saya,

(Edwin Mahendra)