

APLIKASI HRD dengan SISTEM REMINDER
Studi kasus PT. BJS Group

TUGAS AKHIR

Oleh :

Adi Kurniawan	3310812001
Titin Malia Agustina Wati	3310812037

Disusun untuk memenuhi syarat kelulusan Program Diploma III



POLITEKNIK NEGERI-Batam

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM

BATAM

2012

LEMBAR PENGESAHAN

Batam, Juni 2012

Pembimbing,

Mira Chandra Kirana, ST

NIK. 109064

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310812001

Nama : Adi Kurniawan

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

Aplikasi HRD dengan Sistem Reminder

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 28 Juni 2012

Adi Kurniawan
3310812001

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310812037

Nama : Titin Malia Agustina Wati

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

Aplikasi HRD dengan Sistem Reminder

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 28 Juni 2012

Titin Malia Agustina Wati
3310812037

HALAMAN PERSEMBAHAN

Jika diperlukan

KATA PENGANTAR

Dengan rahmat Allah Yang Maha Kuasa sehingga penyusun dapat menyelesaikan Laporan Tugas Akhir yang berjudul “Aplikasi HRD dengan Sistem Reminder” tepat pada waktunya.

Laporan Tugas Akhir ini dibuat untuk memenuhi kriteria penilaian dari mata kuliah wajib yang harus diselesaikan dan dipertanggung jawabkan sebagaimana mestinya.

Penyusun mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak terkait dalam penyusunan laporan ini, khususnya kepada :

1. Tuhan Yang Maha Esa yang telah memberi kami kemudahan dan kesabaran dalam menyusun laporan ini.
2. Bapak Uuf Brajawidagda, ST. MT, selaku Kaprodi Teknik Informatika.
3. Bapak Ari Wibowo, selaku dosen Pengampu.
4. Ibu Mira Chandra Kirana, selaku dosen pembimbing.
5. Orang Tua, keluarga dan orang-orang terdekat yang selalu memberikan dukungan baik moril maupun materil.

Banyak pengalaman yang kami dapat dari penyusunan laporan Tugas Akhir ini. Banyaknya ilmu baru yang kami dapati dari penyusunan laporan ini. Selain itu juga dalam penyusunan laporan ini masih ada kekurangan dan kesalahan. Untuk itu kami berharap adanya saran dan kritik kepada rekan-rekan pada laporan ini, sehingga laporan ini dapat menjadi sebuah laporan yang baik dan sempurna.

Akhir kata penyusun mengucapkan terima kasih, semoga laporan Tugas Akhir ini dapat menambah pengetahuan dan menjadi laporan yang berguna serta menambah wawasan dalam bidang pemrograman.

Batam, Juni 2012

Penyusun.

ABSTRAK

APLIKASI HRD dengan SISTEM REMINDER

Perkembangan ilmu pengetahuan dan teknologi sekarang ini telah membawa kemajuan yang lebih baik. Banyak sekali manfaat dan kemudahan yang telah dihasilkan dengan adanya perkembangan teknologi, terlebih lagi dengan adanya komputer yang dapat membantu dan mempermudah sistem kerja. Komputer membantu manusia dalam menyelesaikan berbagai pekerjaan, misalnya perhitungan sistematis, pengolahan data statistik dan lain sebagainya.

Pada aplikasi HRD dengan sistem Reminder ini mempunyai dua use case yang mana use case tersebut menggambarkan interaksi antara pengguna dengan aplikasi. Pada use case ini pengguna mempunyai masing-masing hak akses yang berperan sebagaimana fungsinya. Selain itu pada aplikasi ini terdapat diagram kelas yang menggambarkan interaksi antar kelas.

Dengan adanya aplikasi HRD dengan sistem Reminder ini dapat membantu dalam pengelolaan data, yang mana aplikasi ini dapat menampilkan data karyawan, data masa kerja, data gaji serta dapat menampilkan reminder untuk masa kerja yang akan berakhir.

Kata kunci : HRD, Reminder

ABSTRACT

APLIKASI HRD dengan SISTEM REMINDER

Development of science and technology today has brought better progress. There are so many benefits and conveniences that have been generated by the development of technology, especially with computers that can assist and facilitate the work system. Computers assist people in completing various tasks, such as systematic calculations, statistical data processing and so forth. On the application of HRD with Reminder system has two use cases in which use case describes the interaction between users and applications. On this use case each user has access rights to act as its function. In addition to this application there is a class diagram that describes the interactions between classes. With the application of HRD with Reminder systems can assist in the management of data, on which the application can display employee data, data length of service, salary data and can display a reminder for future work will end.

Keywords: HRD, Reminder

DAFTAR ISI

Bab I	Pendahuluan.....	Error! Bookmark not defined.
I.1	Latar Belakang.....	Error! Bookmark not defined.
I.2	Rumusan Masalah.....	Error! Bookmark not defined.
I.3	Batasan Masalah	Error! Bookmark not defined.
I.4	Tujuan.....	Error! Bookmark not defined.
I.5	Sistematika Penulisan	Error! Bookmark not defined.
Bab II	Landasan Teori	Error! Bookmark not defined.
II.1	Human Resources Department (HRD)	Error! Bookmark not defined.
II.1.1	Pengertian Human Resources Department (HRD).....	Error! Bookmark not defined.
II.1.2	Aplikasi HRD Sistem Reminder...Error!	Bookmark not defined.
II.1.3	Reminder	Error! Bookmark not defined.
II.1.4	Penelitian yang Berkaitan	Error! Bookmark not defined.
II.2.1	Java	Error! Bookmark not defined.
II.2.2	Karakteristik dan Kelebihan Java	Error! Bookmark not defined.
II.2.3	NetBeans.....	Error! Bookmark not defined.
II.2.4	Penggunaan Java Swing	Error! Bookmark not defined.
II.2.5	Basis Data	Error! Bookmark not defined.
II.2.6	Alasan Perlunya Basis Data.....	Error! Bookmark not defined.
II.2.7	Oracle.....	Error! Bookmark not defined.
II.2.8	SQL*Plus.....	Error! Bookmark not defined.
II.2.9	Kelompok Pernyataan SQL	Error! Bookmark not defined.
II.2.10	Koneksi Oracle ke Java	Error! Bookmark not defined.
Bab III	Bab-Bab Analisis	Error! Bookmark not defined.
III.1	Deskripsi Umum Sistem.....	Error! Bookmark not defined.
III.2	Fitur Utama Perangkat Lunak.....	Error! Bookmark not defined.
III.2.1	Kebutuhan Fungsional	Error! Bookmark not defined.
III.2.2	Kebutuhan Non Fungsional	Error! Bookmark not defined.

III.3	Use Case	Error! Bookmark not defined.
III.3.1	Diagram Use Case	Error! Bookmark not defined.
III.3.2	Skenario Use Case	Error! Bookmark not defined.
III.3.3	Use Case Mengelola Data Karyawan.....	Error! Bookmark not defined.
III.3.4	Use Case Memeriksa Laporan Data Karyawan.....	Error! Bookmark not defined.
III.3.5	Use Case Mengelola Data Masa Kerja Karyawan	Error! Bookmark not defined.
III.3.6	Use Case Memasukkan Data Gaji Karyawan.....	Error! Bookmark not defined.
III.3.7	Use Case Memeriksa dan Mengelola Data gaji Karyawan....	Error! Bookmark not defined.
III.4	Diagram Analisis Kelas	Error! Bookmark not defined.
Bab IV	Bab-Bab Perancangan.....	Error! Bookmark not defined.
IV.1	Interaction Diagram	Error! Bookmark not defined.
IV.1.1	Interaction Diagram Use Case Mengelola Data Karyawan ..	Error! Bookmark not defined.
IV.1.2	Interaction Diagram Use Case Memeriksa Laporan Data Karyawan.....	Error! Bookmark not defined.
IV.1.3	Interaction Diagram Use Case Mengelola Data Masa Kerja Karyawan.....	Error! Bookmark not defined.
IV.1.4	Interaction Diagram Use Case Memasukkan Data Gaji Karyawan	Error! Bookmark not defined.
IV.1.5	Interaction Diagram Use Case Mengelola dan Memeriksa Data Gaji Karyawan.....	Error! Bookmark not defined.
IV.2	Rancangan Kelas Rinci.....	Error! Bookmark not defined.
IV.2.1	Kelas DataKaryawanConsole	Error! Bookmark not defined.
IV.2.2	Kelas MasaKerjaConsole.....	Error! Bookmark not defined.
IV.2.3	Kelas DataReminderConsole.....	Error! Bookmark not defined.

IV.2.4	Kelas DataGajiConsole.....	Error! Bookmark not defined.
IV.2.5	Kelas DataOrganizer.....	Error! Bookmark not defined.
IV.2.6	Kelas ReportPrinter	Error! Bookmark not defined.
IV.2.7	Kelas Reminder	Error! Bookmark not defined.
IV.2.8	Kelas Koneksi.....	Error! Bookmark not defined.
IV.3	Algoritma.....	Error! Bookmark not defined.
IV.3.1	Kelas DataKaryawanConsole	Error! Bookmark not defined.
IV.3.2	Kelas MasaKerjaConsole.....	Error! Bookmark not defined.
IV.3.3	Kelas DataReminderConsole.....	Error! Bookmark not defined.
IV.3.4	Kelas DataGajiConsole.....	Error! Bookmark not defined.
IV.3.5	Kelas DataOrganizer.....	Error! Bookmark not defined.
IV.3.6	Kelas Report Printer	Error! Bookmark not defined.
IV.3.7	Kelas Reminder	Error! Bookmark not defined.
IV.3.8	Kelas Koneksi.....	Error! Bookmark not defined.
IV.4	Perancangan Tampilan.....	Error! Bookmark not defined.
IV.4.1	Rancangan Tampilan Menu Login	Error! Bookmark not defined.
IV.4.2	Deskripsi	Error! Bookmark not defined.
IV.4.3	Rancangan Tampilan DataKaryawanConsole	Error! Bookmark not defined.
IV.4.4	Deskripsi.....	Error! Bookmark not defined.
IV.4.5	Rancangan Tampilan MasaKerjaConsole	Error! Bookmark not defined.
IV.4.6	Deskripsi.....	Error! Bookmark not defined.
IV.4.7	Rancangan Tampilan DataReminderConsole	Error! Bookmark not defined.
IV.4.8	Deskripsi.....	Error! Bookmark not defined.
IV.4.9	Rancangan Tampilan DataGajiConsole	Error! Bookmark not defined.
IV.4.10	Deskripsi.....	Error! Bookmark not defined.
Bab V	Bab-Bab Hasil dan Pembahasan.....	Error! Bookmark not defined.

V.1 Hasil Program	Error! Bookmark not defined.
V.2 Implementasi Kelas	Error! Bookmark not defined.
V.3 Implementasi Antarmuka.....	Error! Bookmark not defined.
Bab VI Kesimpulan dan Saran	Error! Bookmark not defined.
VI.1 Kesimpulan.....	Error! Bookmark not defined.
VI.2 Saran	Error! Bookmark not defined.

DAFTAR GAMBAR

Gambar 1 Deskripsi Umum Sistem.....	24
Gambar 2 Diagram Use Case.....	26
Gambar 3 Diagram Analisis Kelas.....	29
Gambar 4 Interaction Diagram Use Case Mengelola Data Karyawan.....	30
Gambar 5 Interaction Diagram Use Case Memeriksa Laporan Data Karyawan...	31
Gambar 6 Interaction Diagram Use Case Mengelola Data Masa Kerja.....	32
Gambar 7 Interaction Diagram Use Case Memasukkan Data Gaji Karyawan.....	33
Gambar 8 Interaction Diagram Use Case Mengelola dan Memeriksa Data Gaji..	34
Gambar 9 Layar Menu Login.....	50
Gambar 10 Tampilan Menu Data Karyawan	51
Gambar 11 Tampilan Menu Data Masa Kerja	52
Gambar 12 Tampilan Menu Data Reminder	53
Gambar 13 Tampilan Menu Data Gaji.....	54

DAFTAR TABEL

Tabel 1 Deskripsi Tampilan Menu Login	50
Tabel 2 Deskripsi Tampilan Menu Data Karyawanc	51
Tabel 3 Deskripsi Tampilan Menu Data Masa Kerja.....	52
Tabel 4 Deskripsi Tampilan Menu Data Reminder	53
Tabel 5 Deskripsi Tampilan Menu Data Gaji	55
Tabel 6 Implementasi Kelas.....	60
Tabel 7 Implementasi Antarmuka.....	60

Bab I Pendahuluan

I.1 Latar Belakang

Kantor atau perusahaan merupakan tempat untuk melakukan aktivitas kerja. Untuk mencapai kelancaran aktivitas kerja pada suatu perusahaan atau kantor, diperlukan karyawan yang memiliki kemampuan pada bidang kerja masing-masing. Para pekerja tentunya memiliki identitas pribadi dan data-data pribadi yang perlu dimiliki oleh perusahaan tempat dimana mereka bekerja. Data pribadi tersebut berupa identitas lengkap karyawan, nomor karyawan, nomor jamsostek karyawan dan sebagainya. Setiap karyawan memiliki status yang berbeda, ada yang berstatus sebagai karyawan kontrak dan karyawan tetap (permanent). Karyawan yang berstatus kontrak adalah karyawan yang dipekerjakan oleh perusahaan untuk jangka waktu tertentu yang dituangkan dalam surat perjanjian kerja masa tertentu, sedangkan karyawan yang berstatus tetap (permanent) adalah karyawan yang tidak ada batasan jangka waktu lamanya kerja.

Akan tetapi dalam hal yang tersebut di atas masih ditemukan masalah pada penyimpanan data atau informasi mengenai karyawan, yang menimbulkan kesulitan dalam menangani pengelolaan data dikarenakan data yang masih disimpan pada satu file baik berupa hardcopy maupun softcopy. Selain itu juga terdapat kesulitan dalam mengingat masa kerja karyawan kontrak yang sering terlupakan atau terlambat dikarenakan tidak ada automasi pemberitahuan (reminder) kepada staff yang bertugas memberikan surat pemberitahuan kepada karyawan yang telah habis masa kerja.

Untuk itu data harus dapat disimpan dengan baik serta rapi agar pekerjaan lebih efisien, sehingga jika dibutuhkan data dapat diperoleh dan disajikan dengan cepat. Kemudian dalam masalah pemberitahuan masa kerja karyawan kontrak membutuhkan pengingat untuk membantu dalam mengingat masa kerja karyawan

kontrak tersebut. Sehingga pada saat masa kerja tersebut telah berakhir telah dapat mengetahuinya terlebih dahulu.

I.2 Rumusan Masalah

Adapun rumusan masalahnya sebagai berikut :

1. Kesulitan dalam penyimpanan data yang kapasitasnya besar.
2. Kesulitan dalam mengingat masa kerja karyawan kontrak.
3. Kesulitan dalam memperoleh data yang cepat dan akurat.
4. Kesulitan dalam mengelola data karyawan dalam jumlah banyak.

I.3 Batasan Masalah

Dalam masalah ini memiliki batasan masalah sebagai berikut :

1. Tidak menangani perhitungan lembur (*overtime*) karyawan
2. Tidak menangani bonus tertentu/insentif karyawan

I.4 Tujuan

Adapun tujuan yang didapat dalam masalah ini adalah :

1. Menampilkan pengingat (*reminder*) untuk masa kerja karyawan yang akan berakhir
2. Menyediakan laporan data karyawan dan gaji karyawan
3. Menyediakan laporan masa kontrak kerja karyawan

I.5 Sistematika Penulisan

Untuk memudahkan dalam pemahaman isi Tugas Akhir I ini maka di uraikan penulisannya sebagai berikut :

BAB I PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, ruang lingkup, tujuan dan manfaat serta sistematika penulisan.

BAB II LANDASAN TEORI

Berisi dasar-dasar teori dari suatu penelitian. Dan pemberian contoh dan acuan mulai dari contoh gambar, contoh tabel, serta contoh dalam pemberian rumus dan pengacuannya.

BAB III BAB ANALISIS DAN PERANCANGAN

Berisi tentang deskripsi umum sistem, fitur utama sistem, kebutuhan fungsional & kebutuhan non fungsional dan perancangan diagram use case dan dasar-dasar dari perancangan aplikasi, mulai dari pembuatan form sampai dengan menjadi sebuah aplikasi yang dapat digunakan.

BAB V HASIL DAN PEMBAHASAN

Berisi tentang realisasi, pengujian, dan pembahasan dari program aplikasi tersebut.

BAB VI KESIMPULAN DAN SARAN

Berisi kesimpulan dan saran dari perancangan aplika

Bab II Landasan Teori

II.1 Human Resources Department (HRD)

II.1.1 Pengertian Human Resources Department (HRD)

Karyawan merupakan orang yang bekerja pada suatu lembaga (kantor, perusahaan, dsb) dengan mendapatkan gaji atau upah. Terdapat tiga definisi karyawan :

1. Karyawan kontrak, adalah karyawan yang dipekerjakan oleh perusahaan untuk jangka waktu tertentu saja, waktunya terbatas maksimal hanya tiga tahun saja. Hubungan kerja antara perusahaan dan karyawan kontrak dituangkan dalam "Perjanjian Kerja Untuk Waktu Tertentu".
2. Karyawan tetap, adalah karyawan yang tidak ada batasan jangka waktu lamanya bekerja. Jika terjadi pemutusan hubungan kerja bukan karena pelanggaran berat atau karyawan mengundurkan diri maka karyawan tetap mendapatkan uang pesangon, uang penghargaan masa kerja (bagi karyawan yang bekerja minimal tiga tahun) dan uang penggantian hak sesuai UU (Undang–Undang) yang berlaku.
3. Outsourcing, adalah penyerahan sebagian pelaksanaan pekerjaan kepada perusahaan lain. Outsourcing diatur dalam UU 13/2003 dan Kepmenakertrans 220/MEN/X/2004 tentang syarat-syarat penyerahan pelaksanaan pekerjaan kepada perusahaan lain.

Dalam masalah ini karyawan memiliki data-data pribadi yang harus disimpan dan diamankan. Selain itu bagi karyawan kontrak dibutuhkan pengingat untuk mengingatkan masa kerja karyawan yang bersangkutan.

II.1.2 Aplikasi HRD Sistem Reminder

Aplikasi HRD sistem reminder merupakan aplikasi untuk pengelolaan data. Seperti identitas pribadi dan data-data pribadi serta gaji pokok karyawan, data pribadi tersebut berupa identitas lengkap karyawan, nomor karyawan, nomor jamsostek karyawan dan sebagainya beserta gaji pokok karyawan yang diikuti dengan pemotongan pajak dan jamsostek. Rincian perhitungan pajak dan jamsostek terdapat pada persamaan (1) berdasarkan UU PPh No.36/2008 dan UU JHT No.3/1992. Aplikasi HRD sistem reminder merupakan sistem sumber daya manusia atau HRD (Human Resource Department). Manajemen sumber daya manusia adalah suatu proses menangani berbagai masalah pada ruang lingkup karyawan, pegawai, buruh, manajer dan tenaga kerja lainnya untuk dapat menunjang aktifitas organisasi atau perusahaan demi mencapai tujuan yang telah ditentukan. Bagian atau unit biasanya ditangani oleh HRD.

Menurut A.F. Stoner manajemen sumber daya manusia adalah suatu prosedur yang berkelanjutan yang bertujuan untuk memasok suatu organisasi atau perusahaan dengan orang-orang yang tepat untuk ditempatkan pada posisi dan jabatan yang tepat pada saat organisasi memerlukannya.

Peran dan fungsi HRD :

1. Melakukan persiapan dan seleksi tenaga kerja
 - Persiapan, dalam proses ini dilakukan perencanaan kebutuhan akan sumber daya manusia dengan menentukan berbagai pekerjaan yang mungkin timbul.
 - Rekrutmen tenaga kerja, adalah suatu proses untuk mencari calon atau kandidat pegawai, karyawan, buruh atau tenaga kerja baru untuk memenuhi kebutuhan sumber daya manusia organisasi atau perusahaan.
 - Seleksi tenaga kerja, adalah suatu proses menemukan tenaga kerja yang tepat dari sekian banyak kandidat atau calon yang ada.
2. Pengembangan dan evaluasi karyawan

3. Memberikan kompensasi dan proteksi pada pegawai

$$G = Gp - P - J$$

.....Persamaan

(1)

Keterangan :

G = Hitung gaji

Gp = Gaji pokok

P = PPh-21 (pajak)

J = JHT (jamsostek)

Ket :

Potongan JHT = 2%

Potongan PPh-21

- Penghasilan 0 sampai 50 Juta = 5%
- Penghasilan \geq 50 sampai 250 Juta = 15%
- Penghasilan \geq 250 sampai 500 Juta = 25%
- Penghasilan Di atas 500 Juta = 30%

Semua data yang berhubungan dengan karyawan tersebut dapat disajikan secara lengkap melalui sistem informasi ini. Selain itu E-employee dapat berfungsi sebagai pengingat atau reminder kontrak kerja karyawan yang akan berakhir.

Sehingga E-employee membantu dalam menyimpan dan mengamankan data-data karyawan yang ada dan berguna sebagai reminder.

II.1.3 Reminder

Reminder/Pengingat adalah jenis perangkat lunak manajemen waktu komputer yang dirancang untuk mengingatkan pengguna tentang peristiwa penting yang mereka masukan ke program. Kebanyakan program menyediakan kalender, melihat daftar acara, serta teknik mengingatkan. Kebanyakan teknik mengingatkan umum adalah kotak dialog pop-up dan alarm pendengaran. Seperti Acara ulang tahun, liburan, ulang tahun, acara satu-waktu tertentu atau seperti pengingat masa kerja karyawan kontrak. Sebagai perangkat lunak secara bertahap membuat transisi ke layanan web, sejumlah layanan pengingat online telah muncul. Pengguna berlangganan layanan tersebut biasanya menerima pengingat mereka melalui email atau sms dan menampilkan pengingat di desktop komputer.

II.1.4 Penelitian yang Berkaitan

Beberapa penelitian yang berhubungan dengan penilaian terhadap penggunaan pengingat (*reminder*) masa kerja karyawan telah banyak dilakukan oleh peneliti dalam kurun waktu yang cukup panjang. Seperti yang telah dilakukan oleh Yohanes Suryawan 2010, dengan judul “Aplikasi Berbasis Web Pengingat Kontrak Karyawan“ dengan menggunakan ASP.NET(C#) dan SQL Server 2000.

Penelitian E-Employee mengangkat dari judul di atas, akan tetapi E-Employee memiliki implementasi yang tidak terdapat pada judul “Aplikasi Pengingat Kontrak Karyawan berbasis Web” yaitu :

Perbedaan	Penelitian sebelumnya (Pengingat Kontrak Karyawan berbasis Web)	Penelitian sekarang (Aplikasi HRD dengan Sistem Reminder)
Bahasa pemrograman	ASP.NET(C#)	Java
BasisData	SQL Server 2000	Oracle
Penggajian	Belum ada	Ada
Berbasis	Web (online)	Tidak Online
Pengingat (Reminder)	Melalui email	Tampil pada Menu Aplikasi

II.2 Aplikasi Pendukung

II.2.1 Java

Bahasa pemrograman yang digunakan adalah bahasa pemrograman Java. Java memiliki fitur-fitur yang mudah digunakan dan user friendly dibandingkan dengan bahasa pemrograman lain. Java adalah bahasa pemrograman berorientasi objek (OOP) murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya. Dinamakan OOP karena tujuan utamanya untuk membentuk objek, yang dapat berinteraksi dengan objek lainnya untuk memecahkan masalah. OOP berbeda dengan prosedural programming. Prosedural programming menekankan urutan langkah-langkah untuk memecahkan masalah.

OOP menekankan pada pembentukan dan interaksi objek-objek. Multithreading mampu melakukan beberapa pekerjaan dalam waktu yang sama misalnya meng-query DB sambil menampilkan user interface.

II.2.2 Karakteristik dan Kelebihan Java

Java memiliki beberapa karakteristik yang menjadikan Java memiliki kelebihan dibandingkan bahasa pemrograman lain. Adapun karakteristik-karakteristik tersebut :

- Sederhana
Java adalah suatu bahasa sederhana. Para perancang Java berusaha menciptakan suatu bahasa yang mudah dan cepat dipelajari, sehingga bahasa Java dibuat dengan ukuran relatif kecil. Sintaks Java sama dengan bahasa pemrograman C++, namun dengan memperbaiki beberapa kekurangan dari C++, seperti mengurangi kompleksitas beberapa fitur, menambahkan fungsi, serta menghilangkan beberapa hal yang menyebabkan ketidakstabilan sistem pada C++.
- Berorientasi objek
OOP (Object Oriented Programming - Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di Java adalah objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut Object. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan Java sebagai salah satu bahasa pemrograman termudah, bahkan untuk fungsi-fungsi yang advance seperti komunikasi antara komputer sekalipun.
- Dapat didistribusikan dengan mudah
Java memiliki library rutin yang lengkap untuk dirangkai pada protokol TCP/IP, seperti HTTP dan FTP dengan mudah. Kemampuan networking Java lebih kuat dan mudah dipakai. Java memudahkan tugas-tugas

pemrograman jaringan yang sulit, seperti membuka dan mengakses sebuah soket koneksi. Java bahkan memudahkan pembuatan CGI (Common Gateway Interface).

- Aman

Java dimaksudkan untuk digunakan pada pemrograman jaringan dan sistem terdistribusi. Permasalahan ditujukan pada keamanan. Java memungkinkan membuat suatu program yang bebas virus dan sistem yang bebas dari kerusakan, karena Java membuat suatu sistem yang mekanisme keamanannya benar-benar kuat. Suatu class Java yang datang dari sumber yang tidak dipercaya (untrusted source) memiliki akses sumber daya yang terbatas.

- Portabel

Spesifikasi Java tidak terdapat aspek yang bergantung pada lingkungan implementasi. Ukuran tipe data primitif telah ditentukan sejak awal. Misal, "int" selalu berarti sebuah integer 32 bit dua komplement bertanda dan "float" selalu berarti sebuah bilangan floating point 32 bit (mengikuti spesifikasi IEEE 754). Penentuan ukuran tipe data ini akan membuat keseragaman program pada setiap platform, tidak ada tipe data yang tidak kompatibel untuk semua arsitektur perangkat keras dan perangkat lunak.

- Multiplatform

Java dirancang untuk mendukung aplikasi yang dapat beroperasi di lingkungan jaringan yang berbeda. Untuk mengakomodasi hal tersebut, compiler java membangkitkan kode byte, sebuah format yang tidak tergantung pada arsitektur tertentu yang didesain untuk mengirimkan kode ke banyak platform perangkat keras dan perangkat lunak secara efisien. Dengan slogan "Write Once, Run Anywhere", Java dapat dijalankan pada berbagai platform seperti Linux, Unix, Windows, Solaris maupun Mac.

- Multithread

Multithreading adalah kemampuan suatu program komputer melakukan beberapa pekerjaan sekaligus, misalnya mencetak file sambil browsing internet. Multithreading dalam Java sangat mudah dikelola. Thread dalam Java juga memiliki kemampuan untuk memanfaatkan kelebihan multi prosesor jika sistem operasi yang digunakan mendukung multi prosesor.

- Interpreter

Interpreter Java dapat mengeksekusi kode byte Java secara langsung pada setiap mesin yang terdapat interpreter dan setiap run-time Java. Pada sistem Java, tahap link program adalah sederhana, bertahap dan ringan. Hal ini membuat siklus perkembangan menjadi sangat cepat.

- Kuat

Suatu program yang dibuat dengan Java dapat dipercaya dalam berbagai hal, karena Java banyak menekankan pada pengecekan awal untuk menghindari kemungkinan terjadi masalah, pengecekan pada saat run time dan mengurangi kemungkinan timbulnya kesalahan (error).

II.2.3 NetBeans

Bahasa pemrograman Java yang diimplementasikan adalah Netbeans.

Netbeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas. Saat ini terdapat dua produk yaitu NetBeans IDE dan NetBeans Platform.

NetBeans IDE adalah sebuah lingkungan pengembangan untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java, namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas NetBeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan.

NetBeans Platform adalah sebuah pondasi yang modular dan dapat diperluas dan dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi desktop yang besar. Mitra ISV menyediakan plug-in bernilai tambah yang dapat dengan

mudah diintegrasikan ke dalam Platform dan dapat juga digunakan untuk membuat kaskas dan solusi sendiri. Kedua produk tersebut adalah kode terbuka (open source) dan bebas (free) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi Common Development and Distribution License (CDDL). Saat ini NetBeans mengalami perkembangan yang sangat cepat, hingga sekarang NetBeans telah berkembang sampai versi 7. Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak modular yang disebut dengan modul. Sebuah modul adalah suatu arsip Java (Java Archive) yang memuat kelas-kelas Java untuk berinteraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen. Aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan powerful.

II.2.4 Penggunaan Java Swing

Java menyediakan API untuk membuat aplikasi berbasis grafis atau lebih dikenal dengan Graphical User Interface (GUI). Ada dua package untuk membuat GUI yaitu AWT (Abstract Windowing Toolkit) dan Swing. Pemilihan Java Swing dikarenakan Swing lebih portabel dengan AWT. Swing bersifat lightweight atau multiplatform. Swing dikembangkan murni dengan Java, sehingga dapat berdiri sendiri. Selain memiliki komponen yang lebih banyak dibandingkan AWT, Swing juga memiliki fasilitas look and feel yaitu metal, windows, dan CDE/motif. Dengan look and feel tampilan GUI yang dibuat dapat dilihat lebih menarik. Swing memiliki beberapa komponen untuk aplikasi GUI yaitu button, combo box, check box, label, list, scrollbar, textfield dan sebagainya. Setiap komponen ini memiliki fungsi dan cara pembuatan yang berbeda. Semua komponen pada Swing diawali dengan huruf "J" misalnya JButton, JTextArea.

Pada dasarnya, ide di belakang program Hello World adalah belajar bagaimana membuat program java, kompilasi dan menjalankannya. Untuk membuat kode Java dapat menggunakan editor atau menggunakan IDE seperti Eclipse. Di bawah ini adalah contoh kode Java Hello World.

```
import javax.swing.JFrame;
import javax.swing.JLabel;

// Impor laporan
// Periksa apakah jendela ditutup secara otomatis. Jika tidak menambahkan kode yang sesuai
HelloWorldFrame public class extends JFrame {

    public static void main (String args []) {
        baru HelloWorldFrame ();
    }
    HelloWorldFrame () {
        JLabel jlbHelloWorld = JLabel baru ("Hello World");
        add (jlbHelloWorld);
        this.setSize (100, 100);
        // Pack ();
        setVisible (true);
    }
}
```

- Label adalah komponen yang digunakan untuk menampilkan tulisan atau teks yang hanya dapat dibaca dan tidak dapat dimodifikasi. Pada komponen ini dapat mengatur jenis teks, warna teks, warna latar belakang teks, dan bentuk teks lainnya.
- TextArea adalah komponen yang digunakan untuk menulis teks yang berukuran besar dan dapat diedit. Pada komponen ini, selain dapat mengatur isi teks, dapat juga untuk mengatur jenis teks, warna tombol, warna teks dan keaktifan TextArea tersebut.
- Button adalah tombol dengan label teks dan akan merespon event ketika pengguna menekan atau mengklik tombol tersebut. Pada komponen ini selain dapat mengatur label teks, dapat juga untuk mengatur jenis teks, warna tombol, warna teks dan keaktifan tombol tersebut.

II.2.5 Basis Data

Untuk menyimpan data-data karyawan diperlukan sebuah database yang dapat menyimpan data dalam jumlah besar.

Basis data atau database adalah kumpulan item data yang saling berhubungan satu dengan yang lainnya yang dapat diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware komputer dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu. Ada pula yang mendefinisikan basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

II.2.6 Alasan Perlunya Basis Data

- Basis data merupakan salah satu komponen penting dalam sistem informasi karena merupakan dasar dalam menyediakan informasi.
- Basis data menentukan kualitas informasi yang akurat, tepat pada waktunya dan relevan. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
- Basis data mengurangi duplikasi data (data redundancy).
- Dengan mengaplikasikan basis data hubungan data dapat ditingkatkan.
- Basis data dapat mengurangi pemborosan tempat simpanan luar.

Bahasa basis data terdiri atas :

- Data Definition Language (DDL), merujuk pada kumpulan perintah yang dapat digunakan untuk mendefinisikan objek-objek basis data, seperti membuat sebuah tabel basis data atau indeks primer atau sekunder.
- Data Manipulation Language (DML), mengacu pada kumpulan perintah yang dapat digunakan untuk melakukan manipulasi data, seperti penyimpanan data ke suatu table, kemudian mengubahnya dan menghapusnya atau hanya sekedar menampilkannya kembali.

Pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan objektif seperti berikut:

- Kecepatan dan kemudahan (Speed)
- Efisiensi ruang penyimpana (Space)
- Keakuratan (Accuracy)
- Keakuratan (Avaibility)
- Kelengkapan (Completeness)
- Keamanan (Security)
- Kebersamaan pemakaian (Sharability).

SQL (Structured Query Language) adalah bahasa yang digunakan untuk berkomunikasi dengan database. Menurut ANSI (American National Standards Institute), bahasa ini merupakan standar untuk relational database management systems (RDBMS). Pernyataan-pernyataan SQL digunakan untuk melakukan beberapa tugas seperti : update data pada database atau menampilkan data dari database. Hampir semua software database mengimplementasikan bahasa SQL sebagai komponen utama dari produknya, salah satunya MySQL.

II.2.7 Oracle

Database yang digunakan adalah Oracle. Oracle Merupakan RDBMS (Relational DataBase Management System) yang paling banyak digunakan oleh perusahaan-perusahaan di dunia. Database Oracle adalah produksi dari Oracle Cooperation, sebuah perusahaan komputer raksasa yang bermarkas di Redwood City, California. Pengembangan Oracle dimulai pada tahun 1977 ketika Larry Ellison melihat ada peluang bagus yang dimanfaatkan perusahaan-perusahaan software kala itu.

Pada tahun 1979 versi pertama dijual kepada umum. Versi pertama meyertakan interface SQL untuk berinteraksi dengan database. Tahun 1983, mereka mengubah nama perusahaan menjadi Oracle Corporation . Pada tahun itu juga, Oracle Corp. meluncurkan versi ketiga. Tahun 1984 versi keempat diluncurkan, Oracle telah mendukung beberapa sistem operasi yang ada waktu itu. Pengembangan terus dilakukan sesuai dengan perkembangan teknologi komputer. Tahun 1998, Oracle mulai mengadopsi konsep Orientasi Objek(object oriented). Konsep orientasi

objek pada database sedikit berbeda dengan konsep yang dikenal di lingkungan internet sehingga mereka membubuhkan huruf 'i' yang merupakan huruf awal "internet". Sampai saat ini Oracle telah mengeluarkan versi terbarunya, yakni Oracle 11g. Keunggulan Oracle dibanding dengan yang lain :

1. Scalability, memiliki kemampuan menangani banyak user yang melakukan koneksi secara bersamaan tanpa berkurangnya performance.
2. Reliability, memiliki kemampuan untuk melindungi data dari kerusakan jika terjadi kegagalan fungsi pada sistem seperti disk failure.
3. Serviceability, memiliki kemampuan untuk mendeteksi masalah, kecepatan dalam mengoreksi kesalahan, dan kemampuan melakukan konfigurasi ulang struktur data.
4. Stability, memiliki kemampuan untuk crash karena beban load yang tinggi. Hal ini berkaitan dengan scalability.
5. Availability, yaitu kemampuan dalam penanganan crash atau failure agar service dapat tetap yaitu kemampuan dalam penanganan crash atau failure agar service dapat tetap berjalan.
6. Multiplatform, dapat digunakan pada banyak sistem operasi seperti seperti windows, unix, linux, dan solaris.
7. Mendukung data yang sangat besar, Oracle dapat menampung data hampir 512 pet byte (1 pet byte = 1.000.000 gigabyte).
8. Sistem keamanan yang cukup handal
9. Dapat menampung hampir semua tipe data seperti data teks, image, sound, video, dan time series.

Oracle menggunakan SQL (Structured query language) sebagai bahasa perantara antara user dan database. Kemudian Oracle Corp mengembangkan sebuah bahasa prosedural yang mereka sebut sebagai PL/SQL. Dengan menggunakan kedua bahasa ini, seorang user dapat mengoptimalkan penggunaan DataBase Oracle.

SQL merupakan bahasa yang digunakan oleh banyak aplikasi atau tool untuk berinteraksi dengan server DataBase. SQL adalah bahasa fungsional yang tidak mengenal iterasi dan tidak bersifat prosedural. SQL menggunakan perintah-perintah dengan kata-kata sederhana dan mirip dengan bahasa manusia sehari-hari, tentu saja dalam bahasa inggris. Pada SQL, user hanya menentukan apa hasil akhir

yang diinginkan. Sedangkan PL/SQL adalah bahasa prosedural yang dikembangkan sendiri oleh Oracle Corp untuk mengoptimalkan pembuatan aplikasi DataBase. Kata PL pada PL/SQL adalah singkatan dari "Procedural Language". Pada PL/SQL ditambahkan beberapa hal yang dikenal pada dunia pemrograman, seperti variabel, loop, pemrosesan berdasarkan kondisi, operator cursor, modularisasi, dan banyak lagi yang lainnya. Semua tambahan itu bertujuan untuk meningkatkan kinerja operasi-operasi SQL pada DataBase sehingga manfaat dan keandalannya menjadi lebih maksimal.

Keuntungan dari PL/SQL

- Mudah dipelajari dan digunakan
- Dapat disimpan di DataBase Oracle
- Terintegrasi dengan baik pada DataBase Oracle
- Kemampuan yang bagus untuk menangani blok data yang besar
- Terdapat banyak package yang disediakan Oracle untuk meningkatkan performancenya
- Mendukung advanced data tipe
- Mengenal pengelompokan prosedur-prosedur dan fungsi ke dalam satu kesatuan package.

Oracle juga menyediakan tool bagi-bagi pengguna untuk berkomunikasi dengan DataBase, diantaranya :

1. SQL*Plus
2. TOAD
3. TORA
4. HORA
5. SQL Workshop
6. APEX

II.2.8 SQL*Plus

SQL Plus merupakan salah satu aplikasi yang bisa kita gunakan untuk melakukan administrasi DataBase. Namun ada syarat yang musti dimiliki yaitu kemampuan untuk menggunakan query SQL. Berikut penulis jelaskan beberapa penggunaan query SQL. SQL (Structured Query Language) merupakan bahasa query yang

digunakan untuk mengakses database relasional. SQL sekarang sudah menjadi bahasa database standard dan hampir semua sistem database memahaminya. SQL terdiri dari berbagai jenis statemen. Semuanya didesain agar dia memungkinkan untuk dapat secara interaktif berhubungan dengan database. Penggunaan SQL pada DBMS (Database Management System) sudah cukup luas. SQL dapat dipakai oleh berbagai kalangan, misalnya DBA (Database Administrator), progammer ataupun pengguna. Hal ini disebabkan karena :

- SQL sebagai bahasa administrasi database
Dalam hal ini SQL dipakai oleh DBA untuk menciptakan serta mengendalikan pengaksesan database.
- SQL sebagai bahasa query interaktif
Pengguna dapat memberikan perintah-perintah untuk mengakses database yang sesuai dengan kebutuhannya.
- SQL sebagai bahasa pemrograman database
Pemrogram dapat menggunakan perintah-perintah SQL dalam program aplikasi yang dibuat.
- SQL sebagai bahasa klien/server
SQL juga dipakai sebagai untuk mengimplementasikan sistem klien/server. Sebuah klien dapat menjalankan suatu aplikasi yang mengakses database. Dalam hal ini system operasi antara server dan klien bisa berbeda.

Di samping hal tersebut diatas SQL juga diterapkan pada internet atau intranet untuk mengakses database melalui halaman-halaman web untuk mendukung konsep web dinamis.

II.2.9 Kelompok Pernyataan SQL

Pernyataan SQL dapat dikelompokkan menjadi 5 kelompok DDL, DML, DCL, pengendali transaksi dan pengendali programatik.

1. DDL (Data Definition Language)

DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut database, table, atribut (kolom), batasan-batasan terhadap suatu atribut serta hubungan antar table, yang termasuk kelompok DDL ini adalah :

- CREATE, untuk menciptakan table ataupun indeks. Create digunakan untuk membuat suatu object database. Format untuk membuat tiap object berbeda.

Contoh format untuk membuat tabel :

```
CREATE TABLE "NAMA_TABEL"(  
  "NOMOR" NUMBER(2) NOT NULL,  
  "AGAMA" VARCHAR2(20),  
  CONSTRAINT "Agama_Nomor_1" PRIMARY KEY("NOMOR") USING  
  INDEX);
```

Contoh format untuk membuat user dan hak user :

```
CREATE USER "NAMA_USER" PROFILE "DEFAULT" IDENTIFIED BY  
"PASSWORD" DEFAULT  
TABLESPACE "NAMA_TABLESPACE" TEMPORARY TABLESPACE  
"NAMA_TABLESPACE" ACCOUNT  
UNLOCK;  
GRANT UNLIMITED TABLESPACE TO "NAMA_USER";  
GRANT "CONNECT" TO "NAMA_USER";  
GRANT "RESOURCE" TO "NAMA_USER";
```

- ALTER, untuk mengubah struktur tabel. Alter merupakan perintah yang digunakan untuk mengubah suatu object database, yang perlu diperhatikan Oracle tidak banyak memberikan kemudahan untuk mengubah suatu object. Cukup banyak batasan yang ada, misalkan tidak dapat mengganti nama suatu object. Tidak dapat mengubah suatu jenis data dalam suatu tabel yang sudah ada isinya dan masih banyak lagi. Mengubah Nama Tablespace :

```
ALTER DATABASE  
RENAME FILE 'C:\ORACLE\ORADATA\EIS\NAMA1.ORA' TO  
'C:\ORACLE\ORADATA\EIS\NAMA_BARU.DBF'
```

Mengubah Nama Direktori Tablespace :

```
ALTER DATABASE  
RENAME FILE 'C:\ORACLE\ORADATA\EIS\NAMA1.ORA' TO  
'C:\DIR_BARU\NAMA1.ORA'
```

- DROP, untuk menghapus table ataupun indeks. Digunakan untuk menghapus suatu object database tertentu. Karena hanya untuk menghapus, maka perintahnya hampir sama semua.

Format umumnya DROP OBJECT NAMA_OBJECT; Contoh :

```
Drop tablespace NAMA_TABLESPACE;  
Drop user NAMA_USER;  
Drop table NAMA_TABLE;  
Drop index NAMA_INDEX;  
Drop view NAMA_VIEW;  
Drop sequence NAMA_SEQUENCE;  
Drop function NAMA_FUNCTION;  
Drop procedure NAMA_PROCEDURE;
```

2. DML (Data Manipulation Language)

Adalah kelompok perintah yang berfungsi untuk memanipulasi data, misalnya untuk pengambilan, penyisipan perubahan dan penghapusan data, yang termasuk DML adalah :

- SELECT, memilih data. Salah satu syntax yang paling penting dalam SQL adalah select. Select berfungsi untuk menampilkan data-data pada tabel-tabel yang terdapat dalam database. Select secara sederhana dapat dituliskan sebagai berikut :

```
SELECT  
[tb1].[nama_kolom1],  
[tb2].[nama_kolom2],...  
FROM  
[nama_tabel1][tb1],  
[nama_tabel2][tb2]  
WHERE  
[kondisi_join]  
[operator_boolean][kondisi_select]  
ORDERBY[nama_kolom_order]  
GROUP BY [nama_kolom_group];
```

- INSERT, menambah data. Insert digunakan untuk memasukkan data ke sebuah tabel dalam user (schema) tertentu. Syntax insert dapat dituliskan sebagai berikut :

```
INSERT INTO [nama_user].[nama_table]
([nama_kolom1], [nama_kolom2], . . .)
VALUES
([nilai1], [nilai2], . . .);
```

- DELETE, menghapus data. Delete digunakan untuk menghapus data pada sebuah tabel dalam user (schema) tertentu berdasarkan kondisi tertentu.

Syntax delete dapat dituliskan sebagai berikut :

```
DELETE [nama_user].[nama_table]WHERE
[kondisi_delete];
```

- UPDATE, mengubah data. Update digunakan untuk mengubah data pada sebuah tabel dalam user (schema) tertentu berdasarkan kondisi tertentu. Syntax update dapat dituliskan sebagai berikut :

```
UPDATE [nama_user].[nama_table]SET
[nama_kolom1] = [nilai1],
[nama_kolom2] = [nilai2],
[nama_kolom3] = [nilai3],
. . .
WHERE
[kondisi_update];
```

3. DCL (Data Control Language)

Berisi perintah-perintah untuk mngendalikan pengaksesan data, yang termasuk DCL adalah :

- GRANT memberikan kendali pada pengaksesan data.

- REVOKE mencabut kemampuan pengaksesan data
- LOCK TABLE mengunci table

4. Pengendali transaksi

Adalah perintah-perintah yang berfungsi untuk mengendalikan pengeksekusian transaksi, yang termasuk kelompok ini adalah :

- COMMIT menyetujui rangkaian perintah yang berhubungan erat yang telah berhasil dilakukan
- ROLLBACK membatalkan transaksi yang dilakukan karena adanya kesalahan atau kegagalan pada salah satu rangkaian perintah.

II.2.10 Koneksi Oracle ke Java

Untuk menghubungkan java dengan database oracle dibutuhkan sebuah driver yaitu jdbc, yang secara default oracle telah menyediakan jdbc di dalam folder hasil instalasinya, lebih tepatnya di oracle/ora92/jdbc/lib/classes12.jar.

Untuk lebih jelasnya dapat mengikuti langkah-langkah berikut:

1. Buat project pada netbeans
2. Copy folder lib yang berisi file library jdbc di directory
C:\oracle\app\oracle\product\10.2.0\server\jdbc\lib
3. Pada project tersebut, klik kanan – properties
4. Pilih Libraries pada list Properties
5. Add Library
6. Pilih Java DB Driver
7. Add Library
8. Add JAR/Folder
9. Browse file pada lib yang ada pada project
10. Ambil file ojdbc14.jar
11. kemudian open

12. Klik OK

Kemudian membuat beberapa baris untuk mengecek koneksi dengan cara melakukan testing variabel sistem dari Command Prompt. Perintahnya : echo %CLASSPATH% bila pada console tampil path driver maka CLASSPATH sudah berhasil diset. Seperti contoh skrip untuk mencoba koneksi java oracle apakah sudah bejalan atau tidak.

```
import java.sql.*;

class dbAccess {

public static void main (String args []) throws SQLException

{

DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());

//DriverManager.registerDriver (new oracle.jdbc.driver());

Connection conn = DriverManager.getConnection

("jdbc:oracle:thin:@10.4.70.41:1521:mega"."pav"."pav");

// @machineName:port:SID, userid, password

Statement stmt = conn.createStatement();

ResultSet rset = stmt.executeQuery("select * from clogin");

while (rset.next())

System.out.print (rset.getString("id"));

//System.out.print (rset.getString("nama"));

// System.out.println (rset.getString(3));// Print col 1

stmt.close();

}

}
```


Bab III Analisis

III.1 Deskripsi Umum Sistem

Dalam aplikasi HRD dengan Sistem Reminder mempunyai dua user, yaitu HRD Staff dan HRD Manager. HRD Manager mempunyai hak akses untuk dapat mengelola data penggajian karyawan, mengelola data-data karyawan dan mengelola data masa kerja karyawan. HRD Staff mempunyai hak akses untuk dapat mengelola data-data karyawan, memasukkan data gaji karyawan dan mengelola data masa kerja karyawan. Sistem akan menampilkan data yang dimasukkan oleh user, seperti data-data pribadi karyawan, data gaji karyawan dan data masa kerja karyawan. Semua data yang dimasukkan oleh user akan tersimpan kedalam database dan diproses sehingga dapat menghasilkan suatu informasi berupa laporan data-data karyawan, laporan data gaji serta laporan data masa kerja karyawan. Pada gambar 1. merupakan deskripsi umum sistem yang menjelaskan:

1. HRD Staff mengelola data karyawan, data masa kerja dan memasukkan data gaji dan aplikasi menampilkan semua masukan data yang telah dikelola oleh HRD Staff.
2. HRD Manager memeriksa laporan data yang telah dikelola oleh HRD Staff dan ditampilkan oleh aplikasi.



Gambar 1. Deskripsi Umum Sistem

III.2 Fitur Utama Perangkat Lunak

Aplikasi ini mempunyai fitur utama yang terdiri dari kebutuhan fungsional dan kebutuhan non fungsional. Fitur utama menampilkan data-data karyawan, data masa kerja karyawan, data gaji karyawan serta menampilkan reminder untuk masa kerja karyawan yang akan berakhir.

III.2.1 Kebutuhan Fungsional

- F-001 Menyediakan fitur login
- F-002 Menyediakan form untuk mengelola data karyawan.
- F-003 Menampilkan informasi data-data karyawan.
- F-004 Menyediakan form untuk mengelola data masa kerja karyawan.
- F-005 Menampilkan informasi masa kerja karyawan.
- F-006 Menyediakan form untuk mengelola data gaji karyawan.
- F-007 Mampu mengelola perhitungan gaji karyawan dan dapat mencetak slip gaji karyawan.
- F-008 Menampilkan reminder untuk masa kerja karyawan yang akan berakhir.

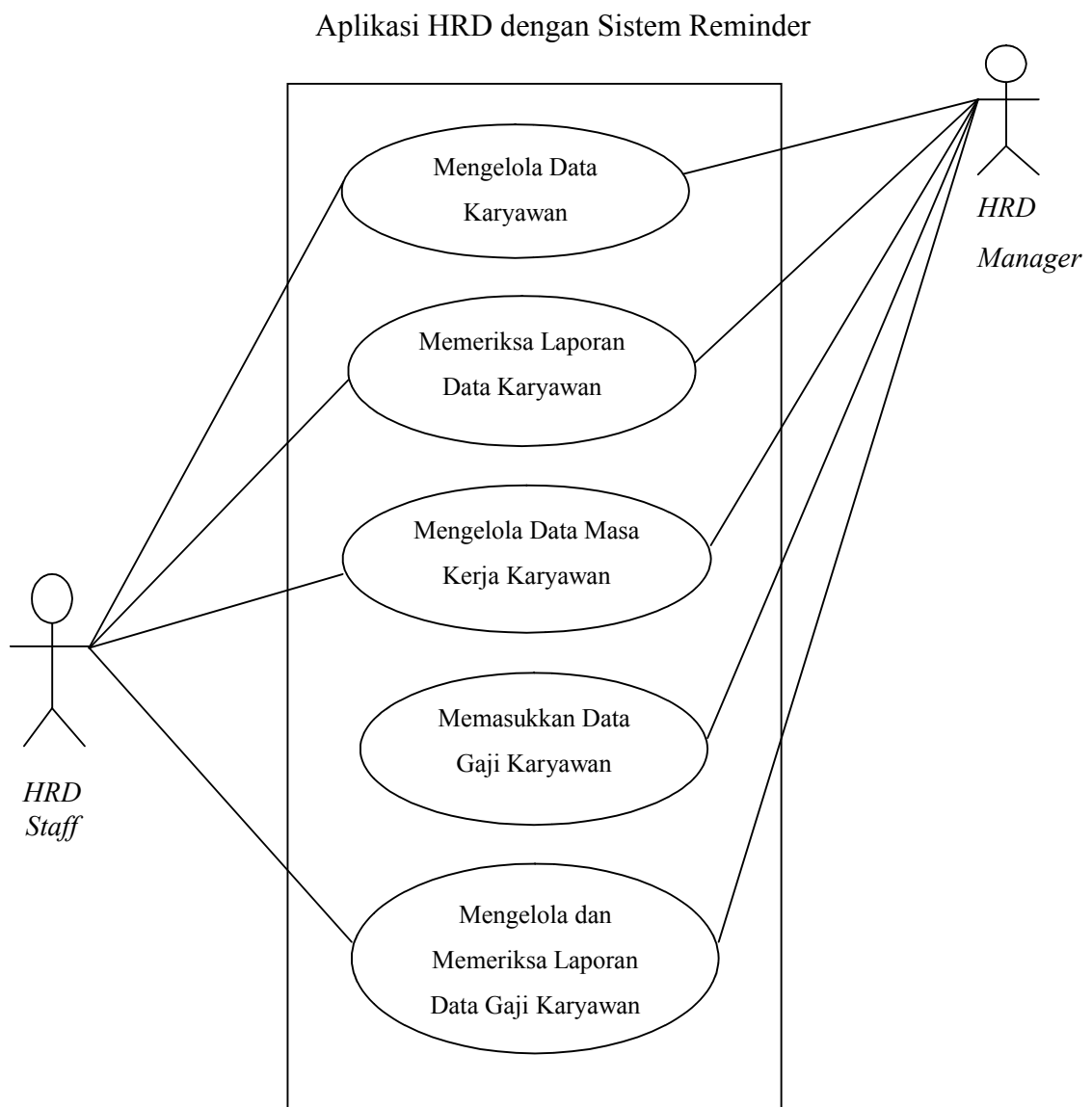
III.2.2 Kebutuhan Non Fungsional

Tidak mempunyai kebutuhan non fungsional

III.3 Use Case

III.3.1 Diagram Use Case

Diagram use case adalah diagram yang menggambarkan interaksi antara pengguna dengan sistem, diagram use case dari aplikasi ini sebagai berikut :



Gambar 2. Diagram Use Case

III.3.2 Skenario Use Case

III.3.3 Use Case Mengelola Data Karyawan

- Kondisi Awal : HRD staff mengumpulkan data karyawan baru yang akan dimasukkan
- Kondisi Akhir : Data yang telah dimasukkan tersimpan kedalam database
- Skenario : Data yang dikumpulkan HRD Staff seperti Emp. Name, Badge ID, Date Join, Dept, Position dan sebagainya. Data dimasukkan kedalam sistem pada menu data karyawan. Data-data tersebut dapat ditambah, diubah dan dihapus. Data-data tersebut tersimpan di dalam database.

III.3.4 Use Case Memeriksa Laporan Data Karyawan

- Kondisi Awal : Data-data karyawan telah dikelola HRD Staff
- Kondisi Akhir : Data telah tersimpan ke dalam database
- Skenario : HRD Manager memeriksa data-data yang telah di tambah, diubah dan dihapus oleh HRD Staff.

III.3.5 Use Case Mengelola Data Masa Kerja Karyawan

- Kondisi Awal : HRD Staff mengumpulkan data-data masa kerja karyawan.
- Kondisi Akhir : Data telah di update dan tersimpan ke dalam database
- Skenario : HRD Staff memasukkan tanggal masuk kerja karyawan, tanggal berakhir kerja karyawan dan merubah data masa kerja karyawan yang diperpanjang masa kerjanya. Dari data tersebut HRD Manager dan HRD Staff dapat melihat lama masa kerja karyawan disertai data lengkap karyawan tersebut.

III.3.6 Use Case Memasukkan Data Gaji Karyawan

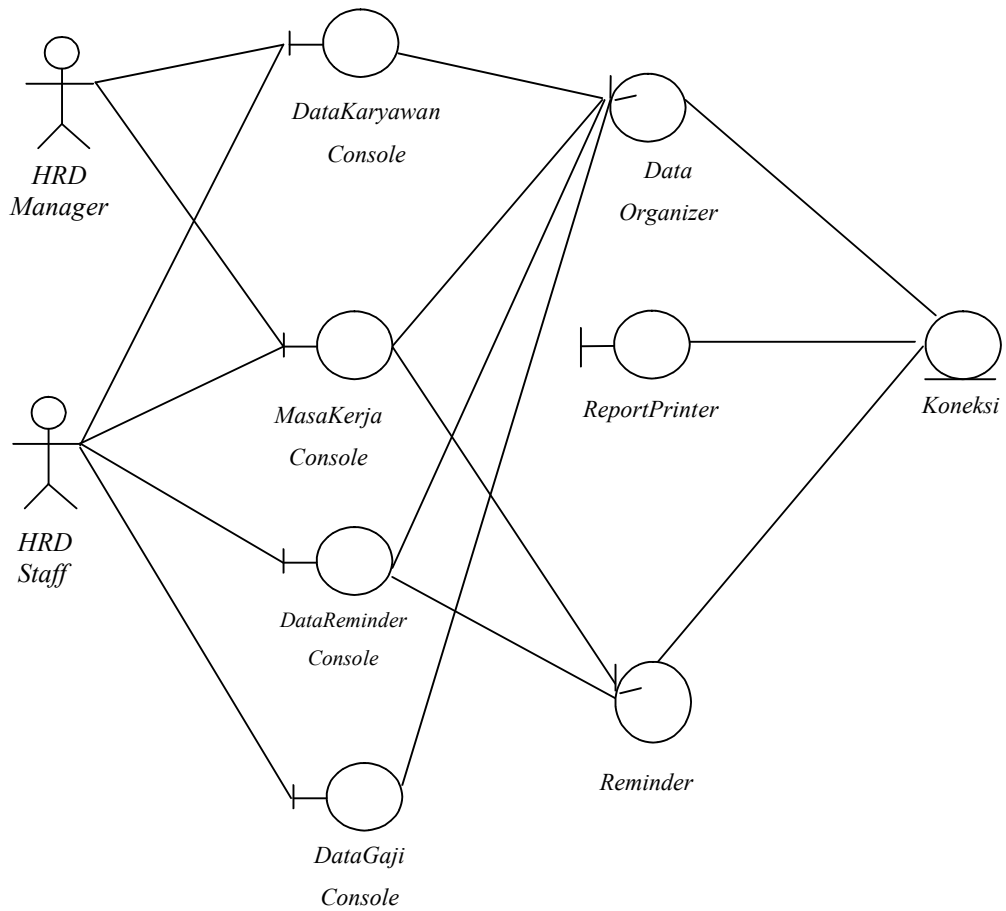
- Kondisi Awal : Data gaji telah disetujui oleh HRD Manager dan diterima oleh HRD Staff.
- Kondisi Akhir : Data telah dimasukkan dan tersimpan ke dalam database
- Skenario : HRD Staff memasukkan data gaji karyawan yang telah distujui oleh HRD Manager berdasarkan dengan posisi serta masa kerja karyawan tersebut. HRD Staff hanya bisa memasukkan data gaji dan tidak dapat menambah dan merubah data tersebut.

III.3.7 Use Case Memeriksa dan Mengelola Data gaji Karyawan

- Kondisi Awal : Data telah dimasukkan oleh HRD Staff dan tersimpan ke dalam database.
- Kondisi Akhir : Data-data gaji karyawan telah tersimpan.
- Skenario : HRD Manager dapat menambah, merubah, menghapus data gaji karyawan yang telah dimasukkan oleh HRD Staff. HRD Manager juga dapat memeriksa laporan data gaji karyawan yang telah dikelola..

III.4 Diagram Analisis Kelas

Analisis kelas merupakan gambaran sistem kerja dari aplikasi. Pada bab ini digambarkan analisis kelas dari aplikasi HRD dengan Sistem Reminder sebagai berikut.



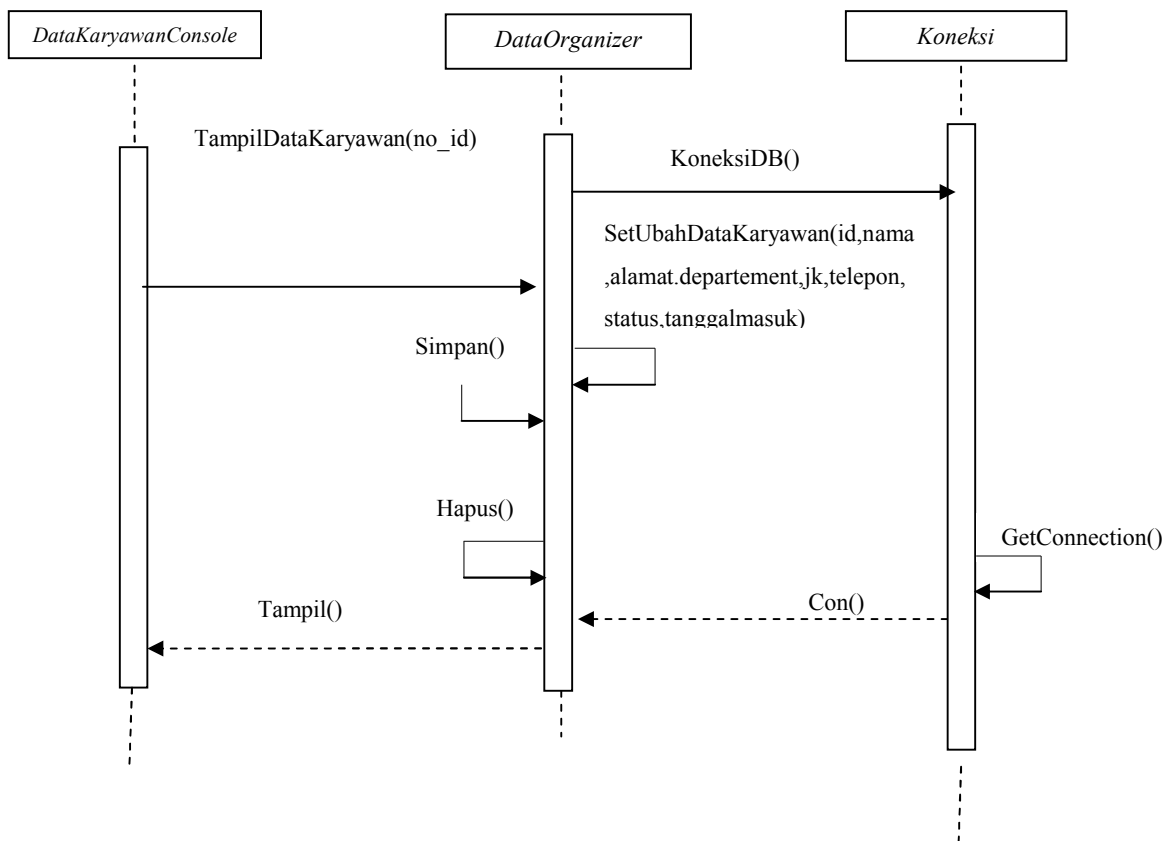
Gambar 3. Analisis Kelas

Bab IV Perancangan

IV.1 Interaction Diagram

Interaction diagram adalah penggambaran interaksi antar kelas dalam satuan waktu. Berikut adalah gambaran dari interaksi antar diagram pada aplikasi HRD dengan Sistem Reminder.

IV.1.1 Interaction Diagram Use Case Mengelola Data Karyawan

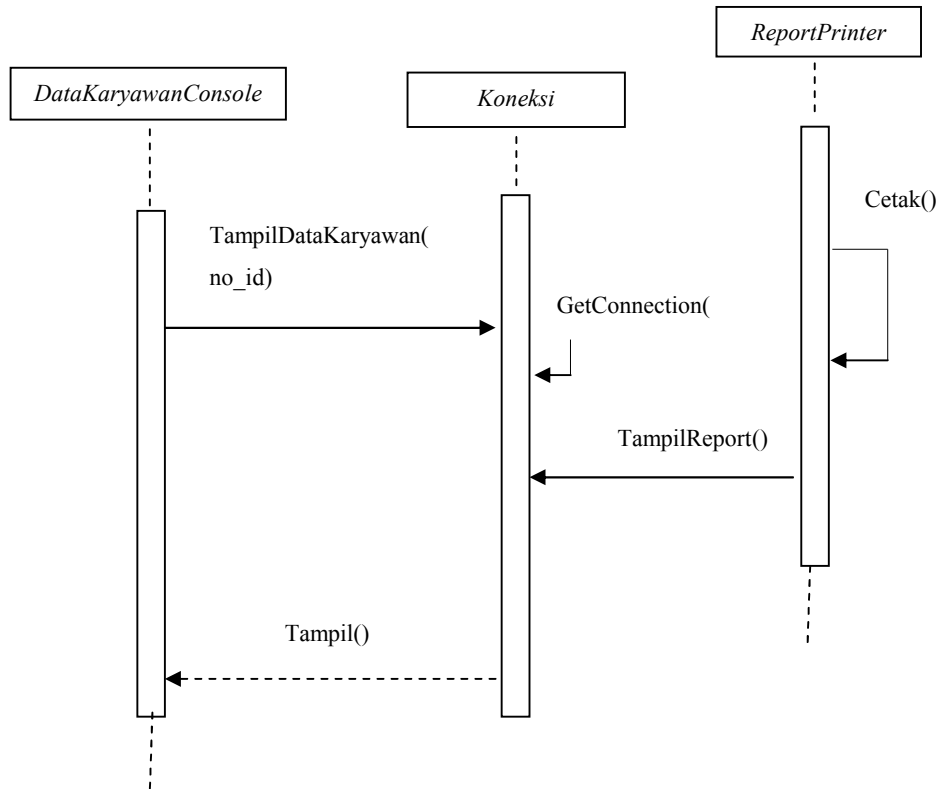


Gambar 4. Interaction Diagram Use Case Mengelola Data Karyawan

Keterangan :

Diagram interaksi di atas menggambarkan proses masukan data yang dimasukkan oleh HRD Staff di *DataKaryawanConsole*, kemudian data diproses pada *DataOrganizer*, data tersebut dapat ditambah, diubah, dihapus dan disimpan atau batal. Data dapat ditampilkan di *DataKaryawanConsole*.

IV.1.2 Interaction Diagram Use Case Memeriksa Laporan Data Karyawan

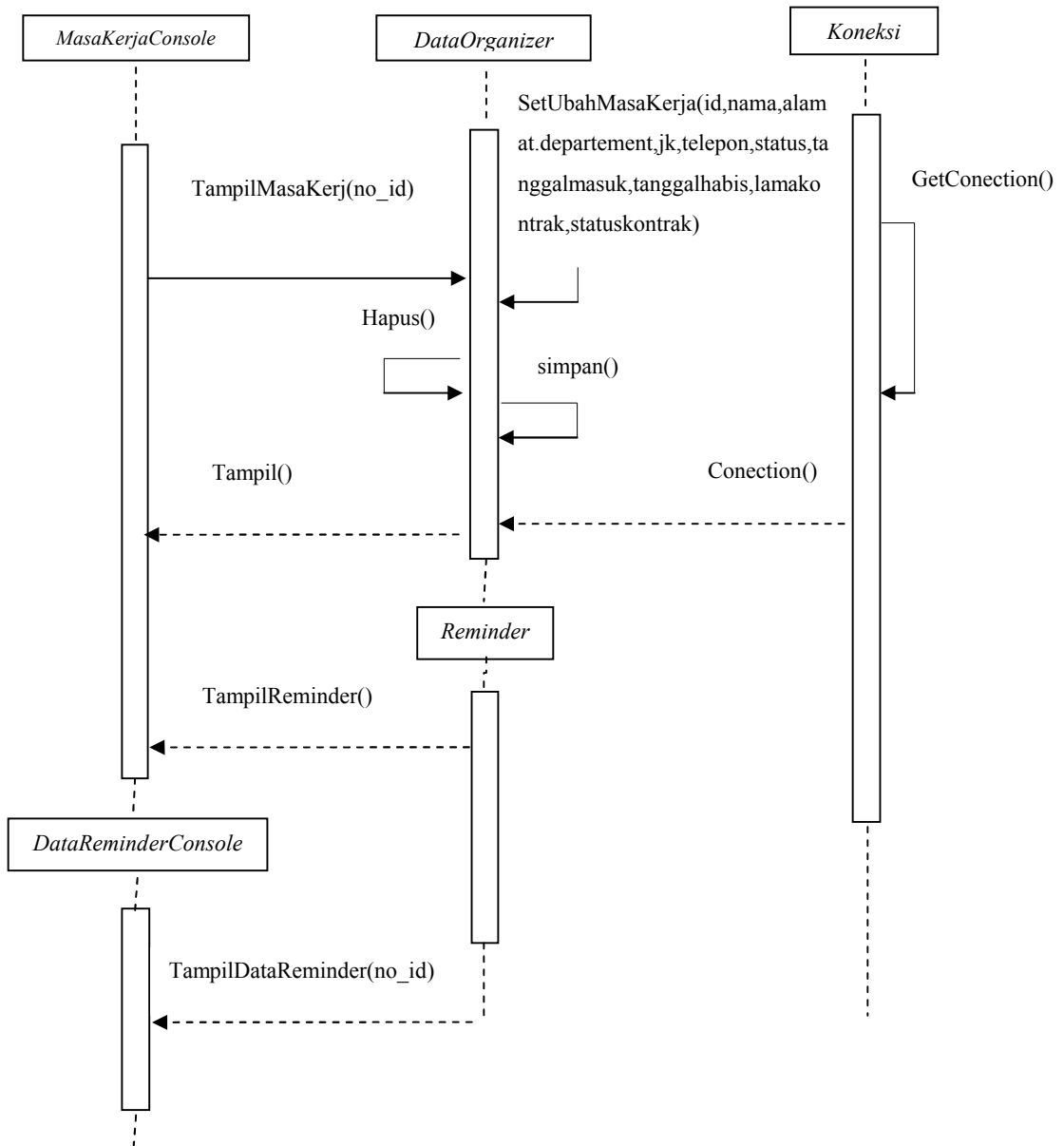


Gambar 5. Interaction Diagram Use Case Memeriksa Laporan Data Karyawan

Keterangan :

HRD Manager memeriksa data yang dimasukkan oleh HRD Staff, data ditampilkan pada *DataKaryawanConsole*. Setelah data-data telah dicek dengan benar maka data disimpan dan data dapat ditampilkan serta dapat dicetak sebagai laporan.

IV.1.3 Interaction Diagram Use Case Mengelola Data Masa Kerja Karyawan

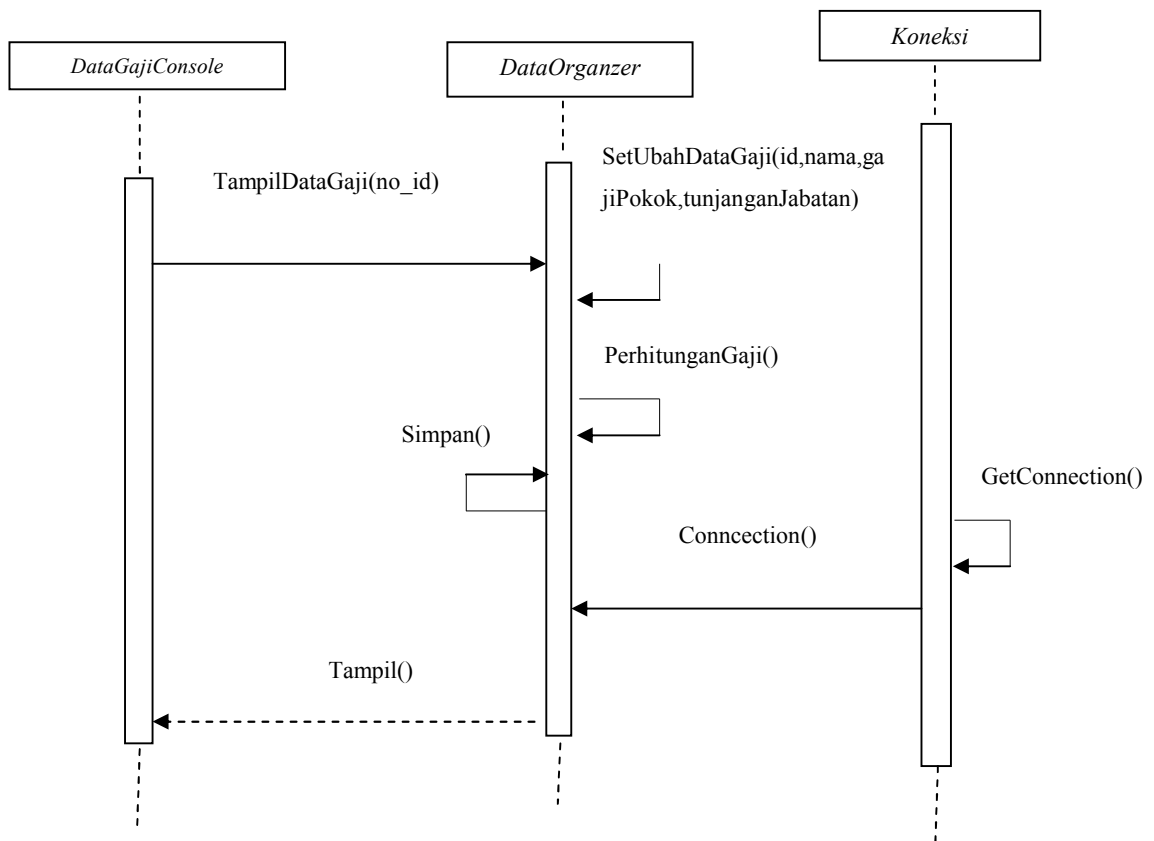


Gambar 6. Interaction Diagram Use Case Mengelola Data Masa Kerja Karyawan

Keterangan :

Data masa kerja dimasukkan oleh HRD Staff pada MasaKerjaConsole dan data diproses di DataOrganizer. Data masa kerja yang akan berakhir dapat diubah atau dihapus jika reminder (peringat) masa kerja yang akan berakhir telah tampil. Akan tampil reminder (peringat) untuk masa kerja yang akan berakhir dan data-data masa kerja dapat ditampilkan.

IV.1.4 Interaction Diagram Use Case Memasukkan Data Gaji Karyawan

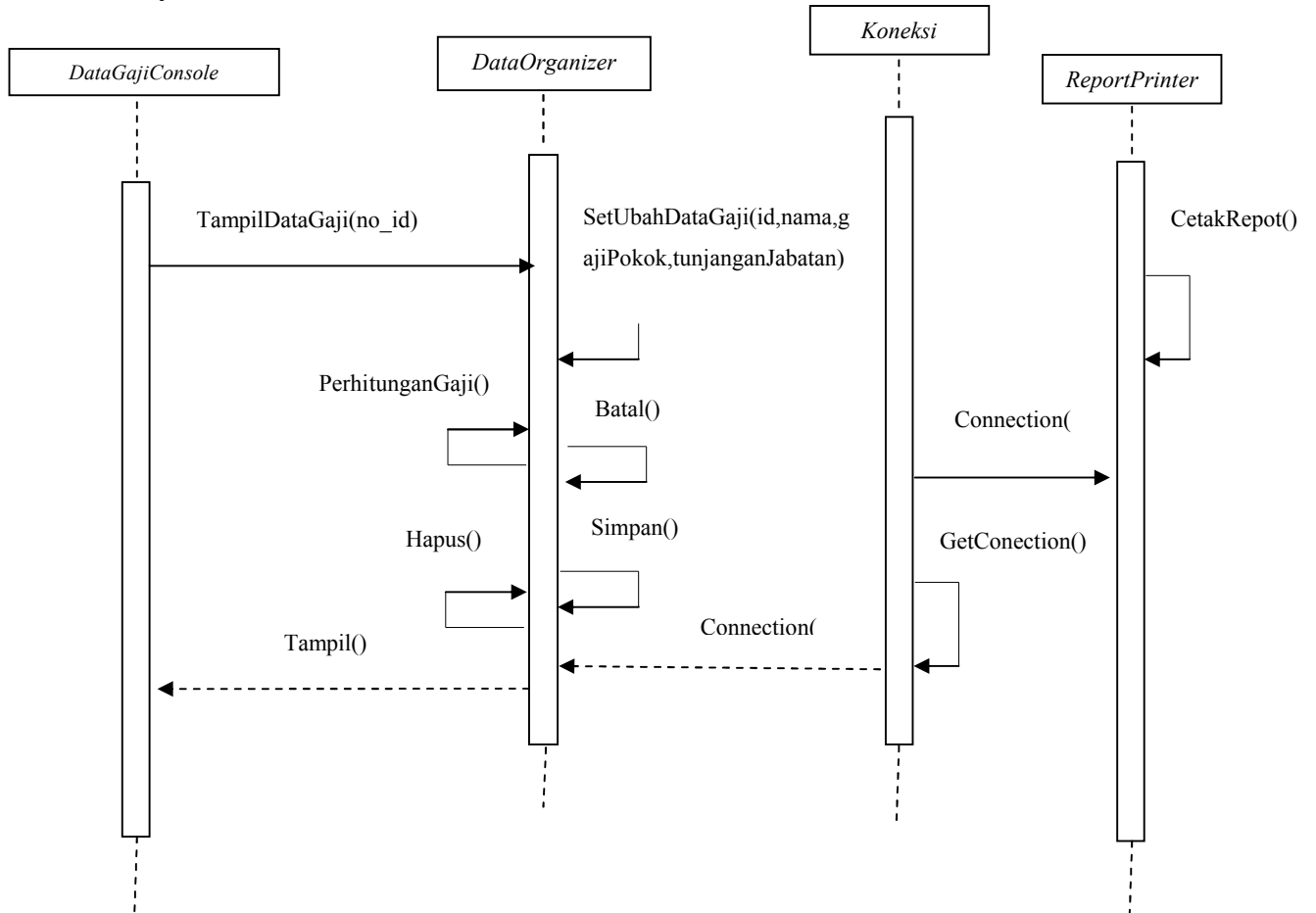


Gambar 7. Interaction Diagram Use Case Memasukkan Data Gaji Karyawan

Keterangan :

HRD Staff memasukkan data gaji di DataGajiConsole, data diproses di DataOrganizer. HRD Staff hanya memasukkan data, data tidak dapat diubah oleh HRD Staff. Data disimpan dan data dapat ditampilkan.

IV.1.5 Interaction Diagram Use Case Mengelola dan Memeriksa Data Gaji Karyawan



Gambar 8. Interaction Diagram Use Case Mengelola dan Memeriksa Data Gaji Karyawan

Keterangan :

Diagram interaksi di atas menggambarkan proses data gaji yang dimasukkan oleh HRD Staff dapat dikelola oleh HRD Manager. Data diproses di DataOrganizer, data dapat ditambah, diubah, dihapus dan disimpan. Data dapat ditampilkan dan dapat dicetak sebagai laporan.

IV.2 Rancangan Kelas Rinci

IV.2 .1 Kelas DataKaryawanConsole

DataKaryawanConsole
+ID():string
+Nama():string
+Dept():string
+Alamat():string
+No.Telp():string
+JenisKelamin():string
+tanggalmasuk():string
+TampilDataKaryawan(no_id):void

IV.2 .2 Kelas MasaKerjaConsole

MasaKerjaConsole
+ID():string
+Nama():string
+Dept():string
+Tgl-Masuk():string
+Tgl_Habis():string
+Lama_Kontrak():string
+Status_Kontrak():string
+TampilMasaKerja(no_id):void

IV.2 .3 Kelas DataReminderConsole

DataReminderConsole
+ID():string
+Nama():string
+Dept():string

+Tgl_Masuk():string
+Tgl_Habis():string
+Lama_Kerja():string
+Status_Kontrak():string
+TampilDataReminder(no_id):void

IV.2 .4 Kelas DataGajiConsole

DataGajiConsole
+ID():string
+Nama():string
+Dept():string
+GajiPokok():string
+Tunjangan_Jabatan():string
+TampilDataGaji(no_id): void

IV.2 .5 Kelas DataOrganizer

DataOrganizer
+ID():string
+Nama():string
+Dept():string
+Jenis_Kelamin():string
+Alamat():string
+Status():string
+No_Telp():string
+Tgl_Masuk():string
+Tgl_Habis():string
+Masa_Kerja():string
+Gaji_Pokok():string
+Tunjangan_Jabatan():string

```

+SetUbahMasaKerja(id,nama,dept,tglmasuk,lamakontrak):void
+SetUbahDataKaryawan(id,nama,alamat.departement,jk,telepon,status,tangg
almasuk):void
+SetUbahDataGaji(id,nama,gajiPokok,tunjanganJabatan):void
+pph_jht(gji, jbtn):void
+tampilData(no_id):void
+Simpan_Ubah():void
+Hapus(id,nama,alamat,departement,jk,telepon,status,tanggalmasuk):void
+Simpan(id,nama,alamat.departement,jk,telepon,status,tanggalmasuk):void
+PrintControl():void
+SetTambahDataKaryawanBaru()void

```

IV.2 .6 Kelas ReportPrinter

Kelas Reportprinter
+ID():string
+Nama():string
+Dept():string
+Jenis_Kelamin():string
+Alamat():string
+Status():string
+No_Telp():string
+Tgl_Masuk():string
+Tgl_Habis():string
+Masa_Kerja():string
+Gaji_Pokok():string
+Tunjangan_Jabatan():string
+CetakReport ():void

IV.2 .7 Kelas Reminder

KelasReminder
+ID():string
+Nama():string
+Dept():string
+Tgl_Masuk():string
+Tgl_Habis():string
+Masa_Kerja():string
+TampilReminder():void

IV.2 .8 Kelas Koneksi

KelasKoneksi
-url : String
-usr : String
-pwd : String
+getConnection():Connection

IV.3 Algoritma

IV.3.1 Kelas DataKaryawanConsole

Nama Kelas : DataKaryawanConsole

Nama Operasi : Proses Menampilkan Data Karyawan

Algoritma : (Algo-001)

```
public void TampilDataKaryawan(no_id){
    try {
        String SQL1="SELECT * FROM t_karyawan where id='"+no_id+"' ";
        ResultSet res1=stt1.executeQuery(SQL1);
        while(res1.next()){
            id.setText(no_id);
            nama.setText(res1.getString("nama"));
            al.setText(res1.getString("alamat"));
            dep.setText(res1.getString("departement"));
            jk.setSelectedItem(res1.getString("jk"));
            telp.setText(res1.getString("telepon"));
            stt.setText(res1.getString("status"));
            String[] dat = split(String.valueOf(res1.getDate("tanggalMasuk")),
"-");
```

IV.3.2 Kelas MasaKerjaConsole

Nama Kelas : MasaKerjaConsole

Nama Operasi : Proses Menampilkan Masa Kerja

Algoritma :

(Algo-002)

```
public void TampilMasaKerja(no_id){
}

try {
    String SQL1="SELECT * FROM t_karyawan where id='"+no_id+"' ";
    ResultSet res1=stt1.executeQuery(SQL1);
    while(res1.next()){
        id.setText(no_id);
        nama.setText(res1.getString("nama"));
        al.setText(res1.getString("alamat"));
        dep.setText(res1.getString("departement"));
        jk.setText(res1.getString("jk"));
        telp.setText(res1.getString("telepon"));
        stt.setText(res1.getString("status"));
        join.setText(String.valueOf(res1.getDate("tanggalMasuk")));
        finis.setText(String.valueOf(res1.getDate("tanggalHabis")));
        kon.setText(res1.getString("lamaKontrak"));
        sttsk.setSelectedItem(res1.getString("statuskontrak"));
    }
}
```

IV.3.3 Kelas DataReminderConsole

Nama Kelas : DataReminderConsole

Nama Operasi : Proses Menampilkan Data Reminder

Algoritma :

(Algo-003)

```
public void TampilDataReminder(no_id)
    if(stkst1("Permanen")){
        try{
            String sql="update t_karyawan set
tanggalHabis=to_date(""+tgh1+"','YYYY-MM-DD'),
tanggalTenggang=to_date(""+tgh1+"','YYYY-MM-DD'),lamaKontrak='0',
statusKontrak=""+stkst1+" where id=""+no_id1+""";
            stt2 (sql);
        }
        }else{
            String sql="update t_karyawan set tanggalHabis=to_date(""+tg2+"','YYYY-
MM-DD'), tanggalTenggang=to_date(""+tg3+"','YYYY-MM-
DD'),lamaKontrak=""+lma1+"', statusKontrak=""+stkst1+" where
id=""+no_id1+""";
            stt2(sql);
        }
    }
```

IV.3.4 Kelas DataGajiConsole

Nama Kelas : DataGajiConsole

Nama Operasi : Proses Menampilkan Data Gaji

Algoritma :

(Algo-004)

```
public void TampilDataGaji(no_id){
    String SQL1="SELECT * FROM t_karyawan,t_gaji_pokok where
t_gaji_pokok.ID=no-id ";
    ResultSet res1=stt1.executeQuery(SQL1);
    while(res1.next()){
        data1[0]=res1.getString("ID");
        data1[1]=res1.getString("Nama");
        data1[2]=res1.getString("gaji_pokok");
        data1[3]=res1.getString("tunjangan_jabatan");
        tblModel1.addRow(data1);
        tabel1.setEnabled(true);
    }
}
```

IV.3.5 Kelas DataOrganizer

Nama Kelas : DataOrganizer

Nama Operasi : Proses Pengontrolan Data Karyawan, Data Masa Kerja dan Perhitungan Gaji

Algoritma : (Algo-005)

```
public void setUbahDataKaryawan(String id,String dep,String nama,String jk
,String al,String stt,String telp){
    try{
        String sql="update t_karyawan set
Departement='"+dep+"',nama='"+nama+"',jk='"+jk+"',alamat='"+al+"',status=
 '"+stt+"',telepon='"+telp+"' where id='"+id+"'";
        stt2.executeUpdate(sql);

        JOptionPane.showMessageDialog(null,"Telah Diubah");
    }
}

public void setUbahMasaKerja(id,nama,dept,tglmasuk,lamakontrak)
(String id,String dep,String nama,String jk,String al,String stt,String telp,String
join2,String join1,String join){
    String tgl=join2+"-"+join1+"-"+join;

    try{
        Connection kon2=new KoneksiBD().main();
        Statement stt2=kon2.createStatement();

        String sql="insert into t_karyawan
values('" +id+"','"+nama+"','"+jk+"','"+dep+"','"+al+"','"+telp+"',to_date('" +tg
l+"','YYYY-MM-DD'),to_date('" +tgl+"','YYYY-MM-
DD'),to_date('" +tgl+"','YYYY-MM-DD'),'0','"+stt+"','null')";
```

```

catch(Exception e){
    System.out.println(e.getMessage());
    JOptionPane.showMessageDialog(null, e);
}
}

public void setUbahDataGaji(String id1,String gji,String jam,String pnrmn,String
jbtn,int bln,String pjk,int thn){
    try{
String          sql="insert          into          t_history_gaji
values(""+id1+"",""+gji+"",""+pjk+"",""+jam+"",""+pnrmn+"",to_date(""+tg+"','YY
YY-MM-DD'),'"+jbtn+"");
        stt2.executeUpdate(sql);

        String sql1="insert into t_suport values(""+id1+"",to_date(""+tg+"','YYYY-
MM-DD'))";
        stt2.executeUpdate(sql1);

    }
}

public void pph_jht(String gji,String jbtn){
    int gj_pokok=(gji);
    int jb=(jbtn);
    if(gj_pokok >=0 && gj_pokok <=50000000){
        pph=5*gj_pokok/100;
        jht=2*gj_pokok/100;
        t_gaji=jb+gj_pokok-pph-jht;
    }else if(gj_pokok >=50000001 && gj_pokok <= 250000000){
        pph=15*gj_pokok/100;
        jht=2*gj_pokok/100;
        t_gaji=jb+gj_pokok-pph-jht;
    }
}

```

```

}else if(gj_pokok >=250000001 && gj_pokok <= 500000000){
    pph=25*gj_pokok/100;
    jht=2*gj_pokok/100;
    t_gaji=jb+gj_pokok-pph-jht;
}
public void SimpanUbah(){
    if(user1.getText().equals("staff")){
        simpan_ubah.setVisible(false);
    }else{
        simpan_ubah.setVisible(true);
    }

    Connection kon1=(Connection) new KoneksiBD().main();
    Statement stt1=(Statement) kon1.createStatement();
    String SQL1="SELECT * FROM t_karyawan where id='"+no_id+"' ";
    ResultSet res1=stt1.executeQuery(SQL1);
    while(res1.next()){
        id.setText(no_id);
        nama.setText(res1.getString("nama"));
        al.setText(res1.getString("alamat"));
        dep.setText(res1.getString("departement"));
        jk.setSelectedItem(res1.getString("jk"));
        telp.setText(res1.getString("telepon"));
        stt.setText(res1.getString("status"));
        String[] dat = split(String.valueOf(res1.getDate("tanggalMasuk")), "-");
        join.setSelectedItem(dat[2]);
        join1.setSelectedItem(dat[1]);
        join2.setText(dat[0]);
    }
}

```

```

SetTambahDataKaryawanBaru(ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setTambahData();

    id.setEditable(true);
    al.setEditable(true);
    dep.setEditable(true);
    id.setEditable(true);
    jk.setEnabled(true);
    join.setEnabled(true);
    join1.setEnabled(true);
    join2.setEditable(true);
    tambah.setEnabled(false);
}

Public void Simpan(id,nama,alamat.departement,jk,telepon,status,tanggalmasuk){
    String tgl=join2+"-"+join1+"-"+join;
try{
    String          sql="insert          into          t_karyawan
values(""+id+"",(""+nama+"",(""+jk+"",(""+dep+"",(""+al+"",(""+telp+"",to_date(""+tg
l+"",'YYYY-MM-DD'),to_date(""+tgl+"",'YYYY-MM-
DD'),to_date(""+tgl+"",'YYYY-MM-DD'),'0',(""+stt+"','null')";
    stt2.executeUpdate(sql);
    //OptionPane.showMessageDialog(null,"Telah Diubah");
}

public void Hapus(id,nama,alamat.departement,jk,telepon,status,tanggalmasuk
try {
    String SQL1="SELECT * FROM t_karyawan where id='"+id+"'";
    ResultSet res1=stt1.executeQuery(SQL1);
    while(res1.next()){
        String nm=res1.getString("Nama");

```

```

String jk1=res1.getString("jk");
String depart=res1.getString("Departement");
String alm=res1.getString("alamat");
String telpp=res1.getString("telepon");
String sttm=res1.getString("status");
String sttk=res1.getString("statuskontrak");
String lma=res1.getString("lamakontrak");
String tglm=String.valueOf(res1.getDate("tanggalMasuk"));
String tglh=String.valueOf(res1.getDate("tanggalHabis"));
try{
String      sql1="insert      into      t_hapus_karyawan
values("+id+", "+nm+", "+jk1+", "+depart+", "+alm+", "+telpp+", to_date("
+tglm+", 'YYYY-MM-DD'), to_date(" +tglh+", 'YYYY-MM-
DD'), "+lma+", "+sttm+", "+sttk+", to_date(" +tg+", 'YYYY-MM-DD'))";
      stt2.executeUpdate(sql1);
      String sql="DELETE FROM t_karyawan WHERE ID ="+id+"
      stt2.executeUpdate(sql);

      //JOptionPane.showMessageDialog(null,"Telah Diubah");
}
}

```

IV.3.6 Kelas Report Printer

Nama Kelas : Report_printer

Nama Operasi : Mencetak Report

Algoritma : (Algo-006)

```
//Mencetak Report//  
    DataOrganizer dt=new DataOrganizer();  
    dt.printControl(2);  
}
```

IV.3.7 Kelas Reminder

Nama Kelas : Reminder

Nama Operasi: *TampilReminder*

Algoritma : (Algo-007)

```
//Tampilreminder//  
if(masa tenggang){  
    Reminder rm=new Reminder();  
    rm.setRefresh(user1(),user());  
}else{  
    Reminder rm=new Reminder();  
    rm.setRefresh(user1(),user());  
    (ok());  
}
```

IV.3.8 Kelas Koneksi

Nama Kelas : Koneksi

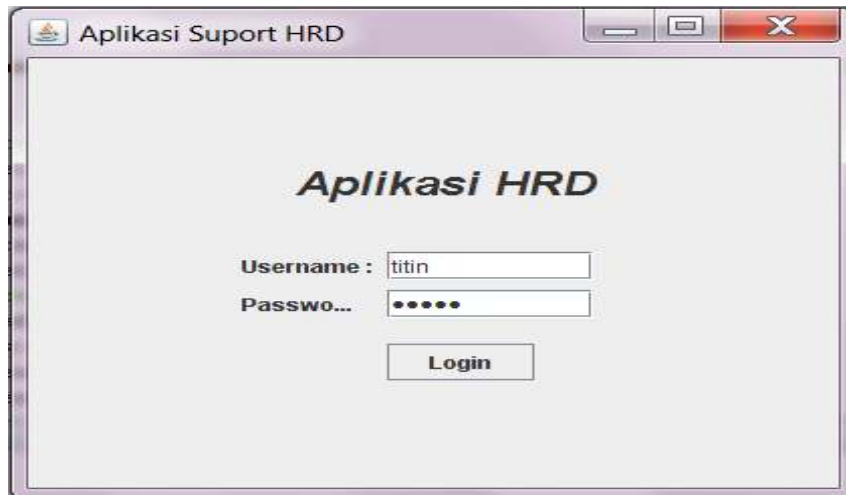
Nama Operasi : *getConnection():Connection*

Algoritma : *(Algo-008)*

```
public class KoneksiBD {  
    private static String ex;  
    public Connection main(){  
        Connection kon = null;  
        try {  
            Class.forName("oracle.jdbc.driver.OracleDriver");    String url =  
            "jdbc:oracle:thin:@localhost:1521:XE";    String username = "hrdalert";  
            String password = "hrdalert";  
            Class.forName("oracle.jdbc.driver.OracleDriver").newInstance();  
            kon = (Connection)  
        }  
    }  
}
```

IV.4 Perancangan Tampilan

IV.4.1 Rancangan Tampilan Menu Login

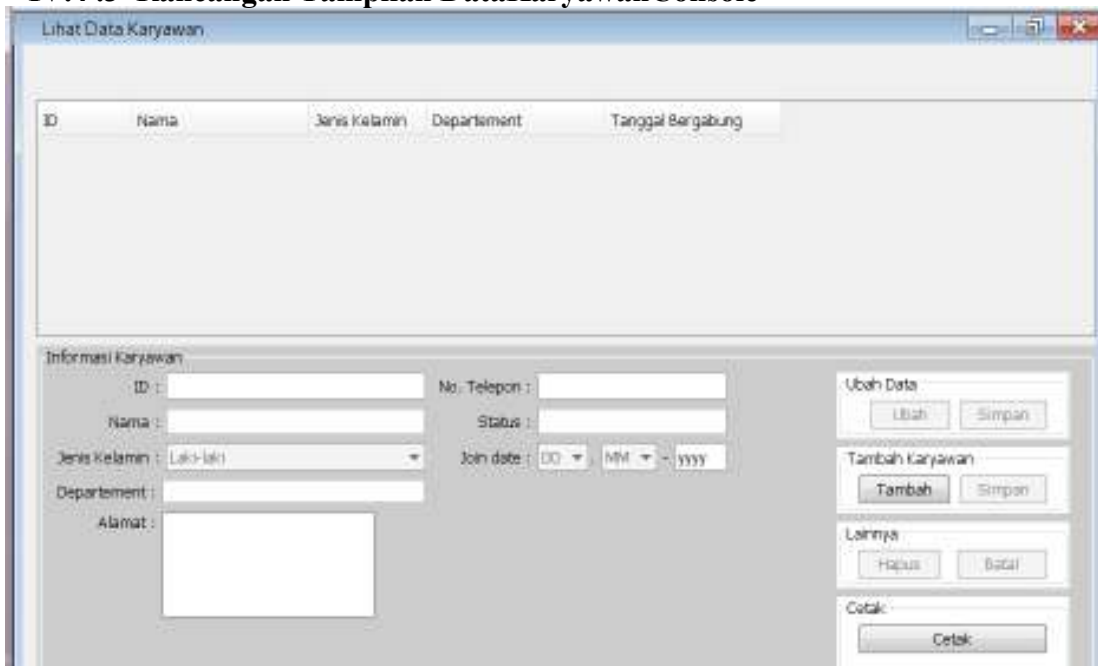


Gambar 9. Layar Menu Logi

IV.4.2 Deskripsi

Id_Objek	Jenis	Nama	Keterangan
jUsername	jLabel	User	Menampilkan Nama Pengguna
jPassword	jLabel	Password	Menampilkan Password Pengguna
jLogin	jButton	Login	Login ke Aplikasi

IV.4.3 Rancangan Tampilan DataKaryawanConsole

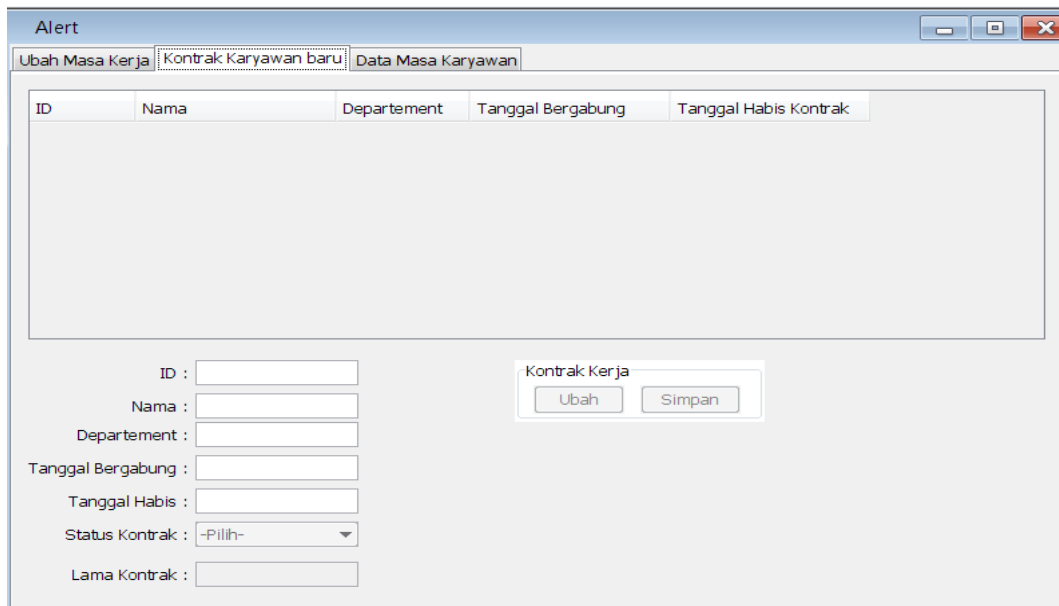


Gambar 10. Menu Data Karyawan

IV.4 .4 Deskripsi

Id_Objek	Jenis	Nama	Keterangan
jTabel	jTable	TblDataKaryawan	Menampilkan Data Karyawan
jID	jLabel	ID	Menampilkan Nomor Karyawan
jNama	jLabel	Nama	Menampilkan Nama Karyawan
jJenisKelamin	jComboBo x	JenisKelamin	Menampilkan Jenis Kelamin
jDepartment	jLabel	Department	Menampilkan Department
jAlamat	jLabel	Alamat	Menampilkan Alamat
jTelepon	jLabel	Telepon	Menampilkan Nomor Telepon
jStatus	jLabel	Status	Menampilkan Status Pribadi Karyawan
jJoinDate	jComboBox	JoinDate	Menampilkan Tgl,Bln,Thn Masuk
jUbah	jButton	Ubah	Merubah Data Karyawan
jSimpan	jButton	Simpan	Menyimpan Data Karyawan
jTambah	jButton	Tambah	Menambah Data Karyawan
jHapus	jButton	Hapus	Menghapus Data Karyawan
jBatal	jButton	Batal	Membatalkan Data Karyawan
jCetak	jButton	Cetak	Mencetak Data Karyawan

IV.4 .5 Rancangan Tampilan MasaKerjaConsole



Gambar 11. Menu Data Masa Kerja

IV.4 .6 Deskripsi

Id_Objek	Jenis	Nama	Keterangan
jTabel	jTable	TblDataMasaKerja	Menampilkan Data MasaKerja
jID	jLabel	ID	Menampilkan Nomor Karyawan
jNama	jLabel	Nama	Menampilkan Nama Karyawan
jTglBergabung	jLabel	TglBergabung	Menampilkan Tanggal Masuk Karyawan Baru
jStatusKontrak	jComboBo x	StatusKontrak	Menampilkan Status Kerja
jLamaKontrak	jLabel	LamaKontrak	Menampilkan Lama Kontrak Kerja Karyawan
jUbah	jButton	Ubah	Merubah Masa Kerja Karyawan
jSimpan	jButton	Simpan	Menyimpan Masa Kerja Karyawan

IV.4.7 Rancangan Tampilan DataReminderConsole

Gambar 12. Menu Data Reminder

IV.4.8 Deskripsi

Id_Objek	Jenis	Nama	Keterangan
jID	jLabel	ID	Menampilkan Nomor Karyawan
jNama	jLabel	Nama	Menampilkan Nama Karyawan
jJenisKelamin	jLabel	JenisKelamin	Menampilkan Jenis Kelamin
jDepartment	jLabel	Department	Menampilkan Department
jAlamat	jLabel	Alamat	Menampilkan Alamat
jTelepon	jLabel	Telepon	Menampilkan Nomor Telepon
jJoinDate	jboBox	JoinDate	Menampilkan Tgl,Bln,Thn Masuk
jFinishDate	jLabel	FinishDate	Menampilkan Tanggal Berkahir Masa Kerja
jStatusKontrak	jComboBox	StatusKontrak	Menampilkan Status Kontrak
jMasaKontrak	jLabel	MasaKontrak	Menampilkan Lama Kontrak
jUbah	jButton	Ubah	Merubah Masa Kerja

Id_Objek	Jenis	Nama	Keterangan
jSimpan	jButton	Simpan	Menyimpan Data Masa Kerja
jCetakMasaTengg	jButton	CetakMasaTengg	Mencetak Masa Tenggang 1 Bulan Sebelum Habis Kontrak

IV.4.9 Rancangan Tampilan DataGajiConsole

Gambar 13. Menu Data Gaji

IV.4.10 Deskripsi

Id_Objek	Jenis	Nama	Keterangan
jTabel	jTable	TblDataKaryawan	Menampilkan Data Karyawan
jTabel	jTable	TblDataGajiKaryawan	Menampilkan Data Gaji Karyawan
jID	jLabel	ID	Menampilkan Nomor Karyawan
jNama	jLabel	Nama	Menampilkan Nama Karyawan

Id_Objek	Jenis	Nama	Keterangan
jGajiPokok	jLabel	Gaji Pokok	Menampilkan Gaji Pokok Karyawan
jTunjangan	jLabel	Tunjangan	Menampilkan Tunjangan Jabatan Karyawan
jSimpan	jButton	Simpan	Menyimpan Data Gaji Karyawan
jSimpanUbah	jButton	SimpanUbah	Menyimpan Data Gaji yang Telah Diubah
jCariID	jLabel	CariID	Menampilkan Data Gaji Karyawan Ketika ID Dimasukkan

Bab V Hasil dan Pembahasan

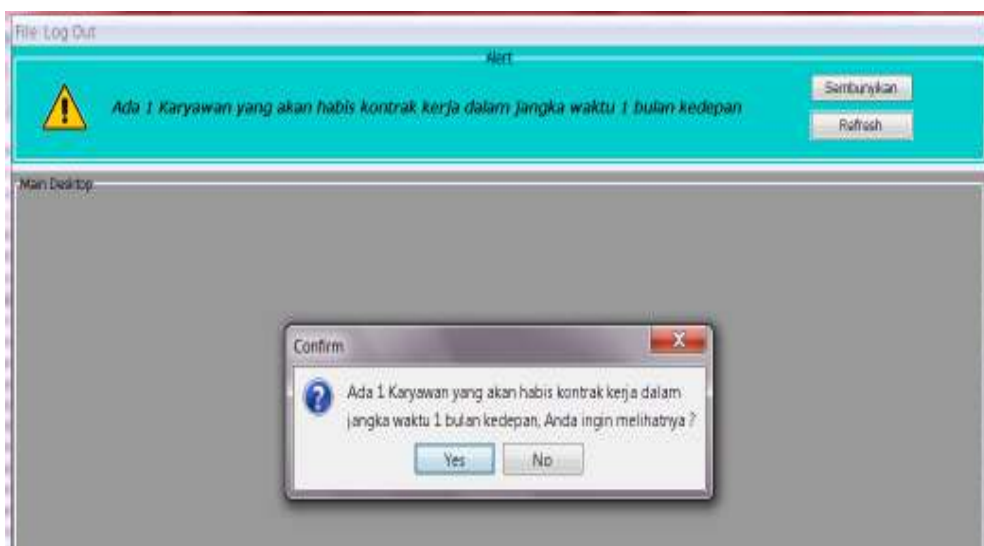
V.1 Hasil Program

Dalam pembuatan aplikasi HRD dengan sistem reminder digunakan perangkat lunak Java NetBeans. Berikut langkah-langkah menjalankan program aplikasi HRD dengan sistem reminder.

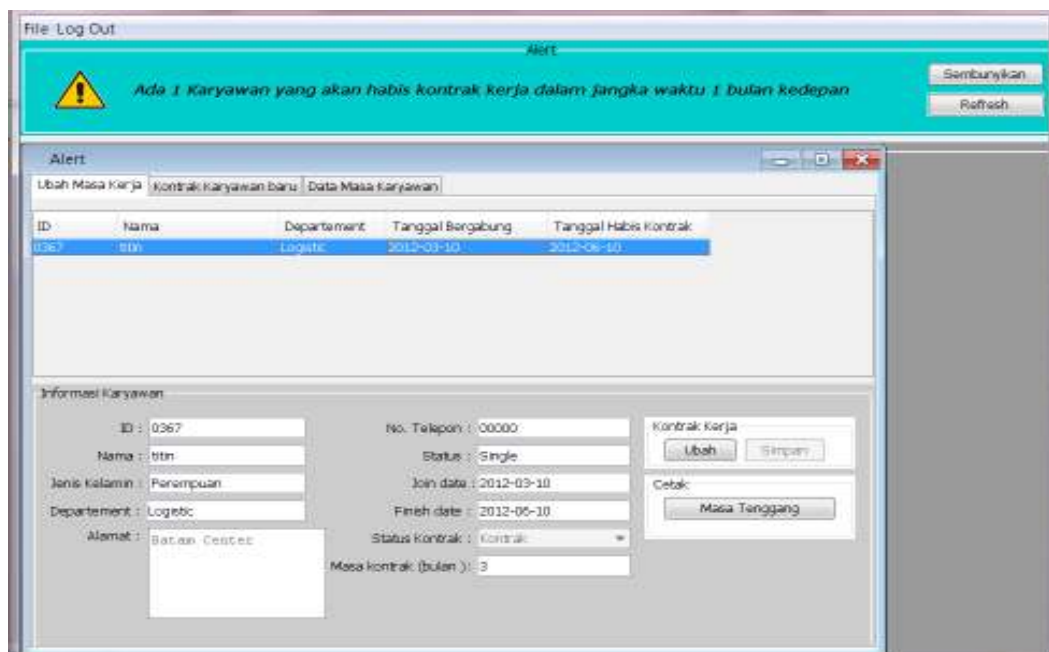
1. Pada menu *login* masukan *username* dan *password* kemudian klik *login*.



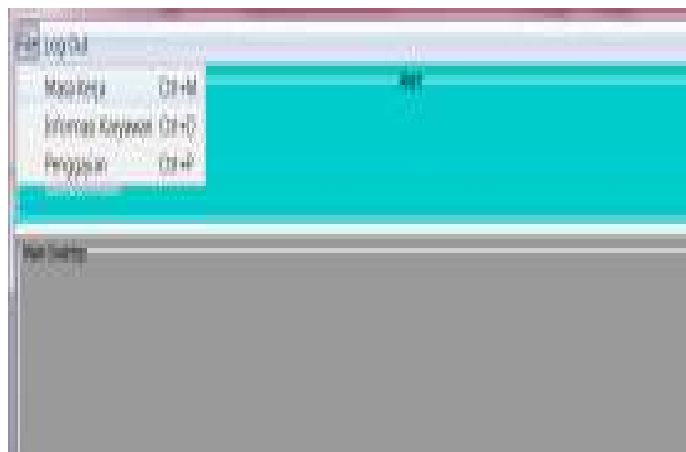
2. Setelah itu muncul menu utama, jika ada karyawan yang akan berakhir masa kerjanya maka akan tampil reminder pada menu utama yang menyatakan bahwa akan ada karyawan yang akan habis kontrak. Reminder muncul satu bulan sebelum kontrak berakhir.



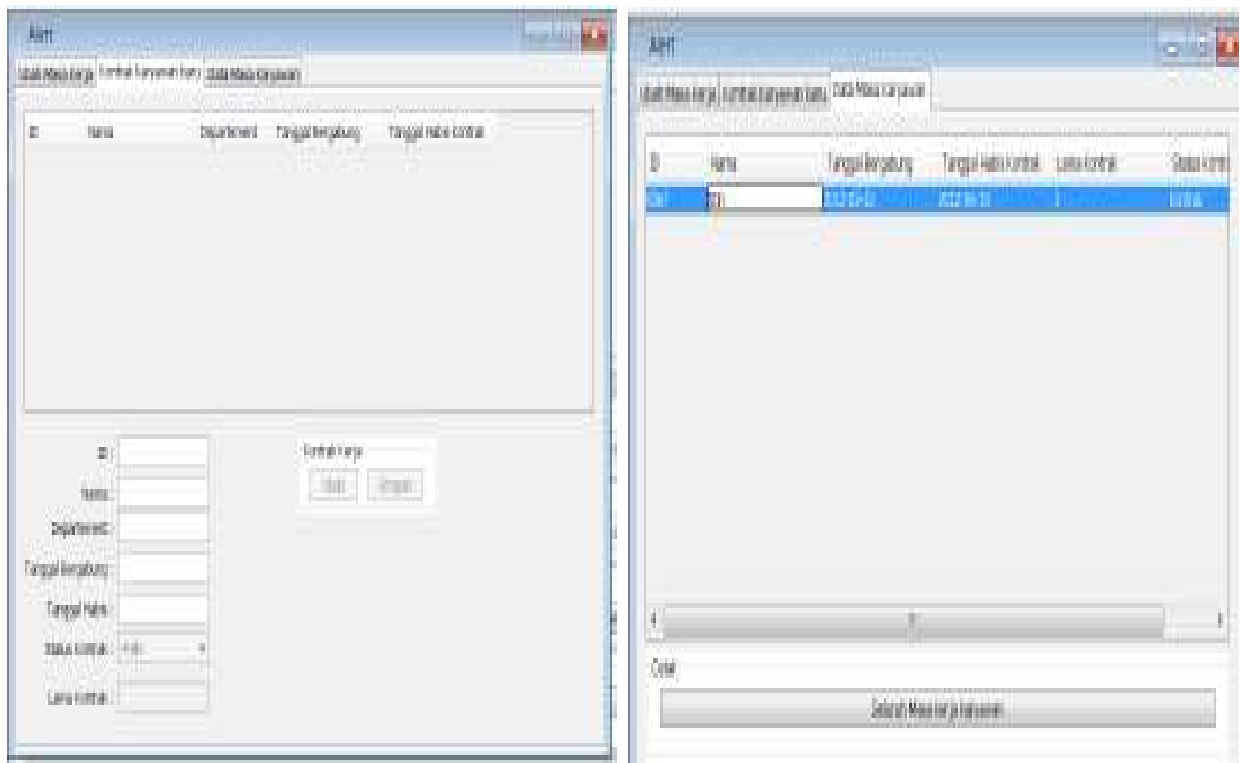
3. Kemudian klik *yes* jika ingin melihat nama-nama yang akan berakhir, jika tidak maka klik *no*. Apabila klik *yes* maka akan tampil data-data karyawan yang akan berakhir dan data kontrak kerja dapat diubah apabila karyawan tersebut akan diperpanjang masa kerjanya. Kemudian tampilan dapat di *close* atau disembunyikan.



4. Pada menu utama ini terdapat menu *file* yang berisi menu Masa Kerja, Informasi Karyawan dan Penggajian juga terdapat menu *log out* untuk keluar aplikasi.



5. Kemudian pada menu Masa Kerja tampil data-data masa kerja karyawan yang telah di isi oleh user HRD Staff. Untuk karyawan baru maka data masa kerja di isi pada menu Kontrak Karyawan Baru dan untuk karyawan yang telah isi masa kerjanya maka data-data masa kerja karyawan dapat dilihat pada menu Data Masa Kerja. Data-data tersebut dapat di ubah, di hapus, di simpan dan di cetak.



6. Pada menu Informasi Karyawan berisikan data-data lengkap karyawan, data-data tersebut dapat di ubah, di simpan, di hapus dan di cetak.

The screenshot shows a window titled "Lihat Data Karyawan". It contains a table with the following data:

ID	Nama	Jenis Kelamin	Departement	Tanggal Bergabung
0367	ttin	Perempuan	Logistic	2012-03-10

Below the table is a form titled "Informasi Karyawan" with the following fields:

- ID :
- Nama :
- Jenis Kelamin :
- Departement :
- Alamat :
- No. Telepon :
- Status :
- Join date :

On the right side, there are several buttons:

- Ubah Data:
- Tambah Karyawan:
- Lainnya:
- Cetak:

7. Pada menu penggajian tampil menu Form Manipulasi Gaji, Form Gaji Bulanan dan Riwayat Gaji dan Kontrak. Pada menu Form manipulasi merupakan menu untuk data-data gaji karyawan, menu Form Gaji Bulanan merupakan data jumlah gaji karyawan pada bulan bersangkutan dan menu Form Riwayat Gaji dan kontrak merupakan menu data-data gaji karyawan yang dapat dilihat data gaji dari awal pertama kerja sampai saat sekarang dan juga terdapat masa kerja karyawan tersebut.

The screenshot shows a window titled "Form Penggajian" with three tabs: "Form Manipulasi Gaji", "Form Gaji Bulanan", and "Riwayat Gaji dan Kontrak". The "Form Manipulasi Gaji" tab is active, showing the following fields:

- Periode :
- ID :
- Nama :
- Gaji Pokok :
- Tunjangan Jabatan :
- Pajak :
- Jamsostek :
- Pemeriksaan :

There is a "Tampil Data" button and a "Data Karyawan" label. Below the form is a "Simpan & Cetak" button. At the bottom, there is a table with the following columns:

ID	Nama	Gaji Pokok	Pajak	Jamsostek	Tunjangan Ja...	pemeriksaan ...	Period
----	------	------------	-------	-----------	-----------------	-----------------	--------

V.2 Implementasi Kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	DataGajiConsole	DataGajiConsole.java	DataGajiConsole.class
2	DataOrganizer	DataOrganizer.java	DataOrganizer.class
3	DataReminderConsole	DataReminderConsole.java	DataReminderConsole.class
4	DataKaryawanConsole	DataKaryawanConsole.java	DataKaryawanConsole.class
5	KoneksiBD	KoneksiBD.java	KoneksiBD.class
6	MasaKerjaConsole	MasaKerjaConsole.java	MasaKerjaConsole.class
7	Reminder	Reminder.java	Reminder.class

V.3 Implementasi Antarmuka

No	Antarmuka	Nama File Fisik	Nama File Executable
1	Layar Menu Utama	menu.java	menu.class
2	Layar Penggajian	DataGaji.java	DataGaji.class
3	Layar Informasi Karyawan	DataKaryawan.java	DataKaryawan.class
4	Layar Masa Kerja	MasaKerja.java	MasaKerja.class
5	Layar Reminder	Reminder	Reminder
6	Layar Data Reminder	Data Reminder	Data reminder

Bab VI Kesimpulan dan Saran

VI.1 Kesimpulan

Kesimpulan dari Tugas Akhir Aplikasi HRD dengan sistem reminder ini, yaitu:

1. Aplikasi ini mampu mengelola data-data karyawan.
2. Aplikasi ini mampu melakukan perhitungan gaji karyawan.
3. Aplikasi ini mampu menampilkan pengingat masa kerja yang akan berakhir.
4. Aplikasi mampu menampilkan data karyawan yang akan berakhir masa kerjanya.
5. Mampu menampilkan dan mencetak laporan

V.2 Saran

Adapun saran dari Tugas Akhir aplikasi HRD dengan sistem reminder, yaitu:

Bagi yang akan mengembangkan Tugas Akhir ini, diharapkan dapat melakukan perhitungan overtime dan dapat menampilkan data jumlah cuti dan jumlah jamsostek karyawan, karena aplikasi HRD dengan sistem reminder ini masih sangat sederhana dan masih banyak pengembangan yang dapat dilakukan.


DAFTAR PUSTAKA




Hanya berisi daftar karya orang lain yang diacu dalam dokumen TA


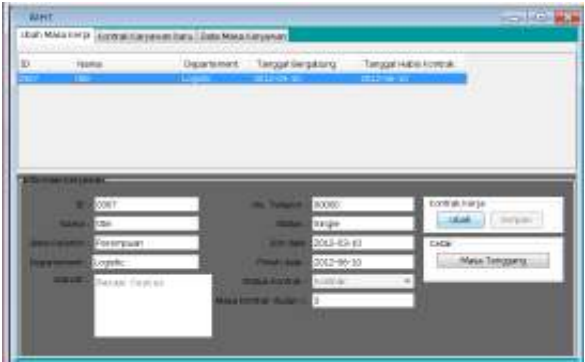

1. Betts, M., *The Almanac: Hot Tech*, ComputerWorld 52, 17 November 2003
2. Berkhin, P., *Survey of Clustering Data Mining Techniques*, Accrue Software, 1045 Forest Knoll Dr., San Jose, 2002
3. <http://Wikipedia.org>
4. <http://google.com>
5. <http://smapluspgri.sch.id/KnowledgeSharing>


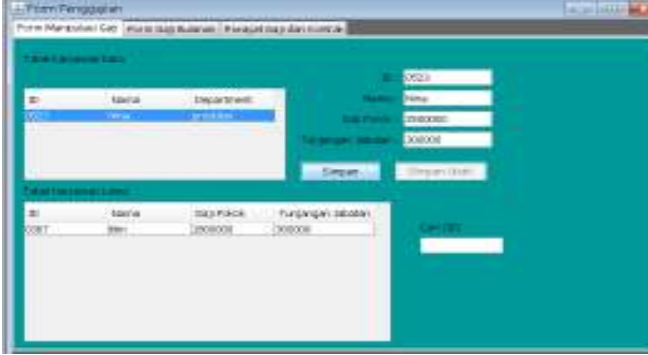
Lampiran


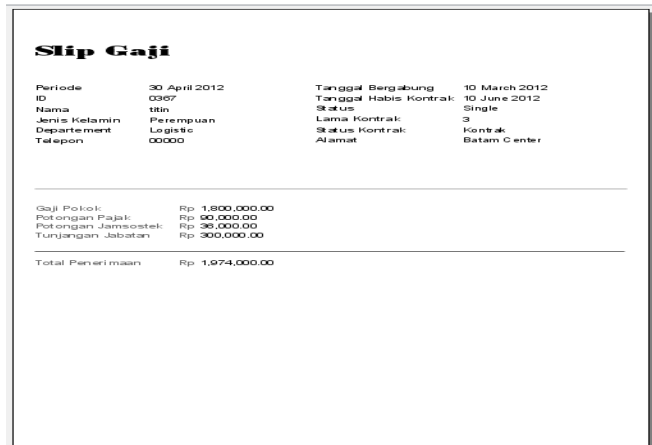
I. Tabel Pengujian

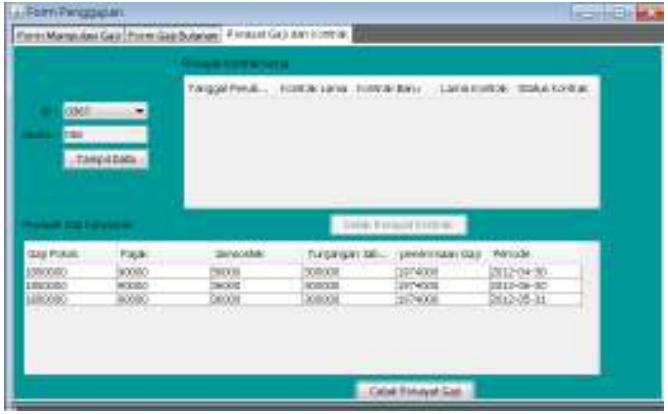

No	User	Prosedur & Kasus uji	Hasil yang diharap	Hasil Test	Ket.
1	User	Menambah dan menyimpan data karyawan	Data berhasil ditambah dan disimpan	  	Diterima


No	User	Prosedur & Kasus uji	Hasil yang diharap	Hasil Test	Ket.
2	User	Mengubah data karyawan	Data berhasil di ubah		Diterima
3	User	Hapus data karyawan	Data karyawan (Adi) berhasil di hapus		Diterima
4	User	Mencetak laporan data-data karyawan	Data-data karyawan berupa laporan berhasil di cetak		Diterima

No	User	Prosedur & Kasus uji	Hasil yang diharap	Hasil Test	Ket.
5	User	Memasukkan data masa kerja karyawan baru.	Menghasilkan informasi masa kerja karyawan baru		Diterima
6	User	Mengubah dan menyimpan data masa kerja karyawan	Data karyawan berhasil diubah kemudian disimpan		Diterima
7	User	Menampilkan seluruh data masa kerja karyawan	Seluruh data masa kerja karyawan berhasil ditampilkan		Diterima

No	User	Prosedur & Kasus uji	Hasil yang diharap	Hasil Test	Ket.
8	User	Mencetak seluruh data masa kerja karyawan	Semua data masa kerja karyawan berhasil dicetak		Diterima
9	User	Memasukkan data gaji karyawan baru beserta tunjangannya	Data gaji dan tunjangan karyawan baru berhasil dimasukkan		Diterima

No	User	Prosedur & Kasus uji	Hasil yang diharap	Hasil Test	Ket.
10	User	Perhitungan total gaji yang diterima karyawan tiap bulan setelah dikurangi pajak dan jamsostek	Data total gaji bulanan karyawan berhasil ditampilkan dan disimpan		Diterima
11	User	Mencetak slip gaji karyawan	Berhasil mencetak slip gaji karyawan		Diterima

No	User	Prosedur & Kasus uji	Hasil yang diharap	Hasil Test	Ket.
12	User	Menampilkan data-data riwayat gaji karyawan dari awal bekerja hingga berakhir masa kerja karyawan tersebut	Data riwayat gaji karyawan berhasil ditampilkan dari awal kerja sampai masa kerja yang telah ditentukan		Diterima
13	User	Mencetak laporan riwayat gaji karyawan	Berhasil mencetak laporan riwayat gaji karyawan		Diterima

No	User	Prosedur & Kasus uji	Hasil yang diharap	Hasil Test	Ket.
14	User	Menampilkan reminder	Berhasil menampilkan reminder untuk masa kerja karyawan yang akan berakhir	 <p>The screenshot shows a software window with a light blue header. A yellow warning triangle icon is on the left. The main text reads: 'Attn : Karyawan yang akan habis masa kerjanya akan segera berakhir & akan berakhir'. Below this text are two buttons: 'Selesai' and 'Tutup'. A smaller dialog box is overlaid in the center, containing the same text and two buttons: 'OK' and 'Batal'.</p>	Diterima