



ITS
Institut
Teknologi
Sepuluh Nopember

PROYEK AKHIR

RANCANG BANGUN HUMAN RESOURCE INFORMATION SYSTEM (HRIS) SEBAGAI APLIKASI PENILAIAN KINERJA KARYAWAN PERUSAHAAN ASURANSI

Oleh :

LUQMAN PUTRA WIBOWO
NRP. 7208.030.002

Dosen Pembimbing :

Mike Yuliana, ST.MT.
NIP 197811232002122009

Dosen Pembimbing 2 :

Ir.Nanang Syahroni, M.Kom
NIP 196511091991031006

**JURUSAN TEKNIK TELEKOMUNIKASI
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2011**

**RANCANG BANGUN HUMAN RESOURCE INFORMATION
SYSTEM (HRIS) SEBAGAI APLIKASI PENILAIAN KINERJA
KARYAWAN PERUSAHAAN ASURANSI**

Oleh :
LUQMAN PUTRA WIBOWO
NRP. 7208.030.002

Dosen Pembimbing :
Mike Yuliana, ST.MT.
NIP 197811232002122009

Dosen Pembimbing 2 :
Ir.Nanang Syahroni, M.Kom
NIP 196511091991031006

**JURUSAN TEKNIK TELEKOMUNIKASI
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
JANUARI 2011**

RANCANG BANGUN HUMAN RESOURCE INFORMATION SYSTEM (HRIS) SEBAGAI APLIKASI PENILAIAN KINERJA KARYAWAN PERUSAHAAN ASURANSI

Oleh :
LUQMAN PUTRA WIBOWO
7208.030.002

**Proyek Akhir ini Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Ahli Madya (A.Md.)**

Di

**Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember Surabaya**

Disetujui Oleh:

Tim Penguji Proyek Akhir

Dosen Pembimbing

1.

1.

Aries Prastiarso
NIP. 196611171991031004

2.

Mike Yuliana, ST.MT.
NIP. 197811232002122009

2.

Achmad Subhan, KH, ST
NIP. 197701202008011010

3.

Ir.Nanang Syahroni, M.Kom
NIP. 196511091991031006

Amang Sudarsono, ST
NIP. 197409202002121001

**Mengetahui,
Ketua Jurusan Teknik Telekomunikasi**

Arifin ST,MT
NIP. 196005031988031004

iii

ABSTRAK

Seiring dengan perkembangan teknologi maka dibutuhkan kepraktisan dalam segala hal termasuk penerapan pada sistem HRD (*Human Resource Development*). Sistem pendukung keputusan pada sebuah perusahaan dalam studi kasus ini yaitu perusahaan asuransi ditujukan untuk menciptakan suatu aplikasi yang berguna untuk memonitor kinerja karyawan

Pada tugas akhir ini, dibuat sebuah aplikasi Penilaian Kinerja Karyawan Asuransi dengan metode AHP berbasis JAVA untuk mempermudah HRD dan pimpinan perusahaan dalam menilai kinerja karyawan.

Dari hasil pengujian terlihat bahwa sistem telah berjalan dengan baik, waktu eksekusi Program pun hanya berlangsung singkat, terlihat dari uji lama waktu eksekusi login system hanya 5,9ms, dan lama waktu yang dibutuhkan untuk memproses metode AHP hingga dihasilkan nilai kinerja karyawan adalah 1091,8ms. Begitu pula dengan hasil uji kepuasan sistem menggunakan metode kuisioner dimana para responden nya adalah Kepala bagian dan Asisten Manajer, 80% responden menyatakan bahwa tampilan dari sistem menarik, 70% dari Responden menyatakan lebih menyukai aplikasi Penilaian Kinerja Karyawan Berbasis Java ini dibandingkan cara manual dan 80% responden menyatakan bahwa aplikasi ini bermanfaat dalam mengakses data serta memiliki menu akses yang dianggap cukup, sehingga dapat digunakan untuk membantu dalam menilai kinerja karyawan.

Kata kunci : Sistem, Sistem Pendukung Keputusan, penilaian kinerja karyawan dan metode *Analytical Hierarchy Process* (AHP)

ABSTRACT

Along with the technological developments, The implementation of the system of HRD (Human Resource Development) is practicality needed in all things. Decision support systems on an enterprise in this case study is an insurance company intended to create an application that allows you to monitor employee performance

In this thesis, an Insurance Employee Performance Appraisal with AHP method based on JAVA was made to facilitate the Corporate leader in assessing employee performance

From the test results, it shows that the system has been running well, so are the test results using the method quisioner system satisfaction, 80% of respondents stated that the appearance of the system is attractive , 70% of respondents stated preference applications Java-Based Employee Performance Appraisal compared to the manual way and 80% of respondents stated that this application is useful in accessing the data and have access to menus that are considered adequate, so it can be used to assist in assessing performance.

Keywords: System, Decision Support System, employee performance appraisal and methods Analytical Hierarchy Process (AHP)

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.



Dengan mengucapkan puji syukur kepada Allah, atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan proyek akhir ini dengan berjudul :

RANCANG BANGUN HUMAN RESOURCE INFORMATION SYSTEM (HRIS) SEBAGAI APLIKASI PENILAIAN KINERJA KARYAWAN PERUSAHAAN ASURANSI

Proyek Akhir ini digunakan sebagai salah satu syarat akademis untuk memperoleh gelar Ahli Madya (A.Md) di Politeknik Elektronika Negeri suarabaya ITS.

Penyusun mengucapkan terima kasih kepada seluruh pihak yang telah membantu hingga penyusunan buku proyek akhir ini selesai. Penyusun berharap semoga buku ini dapat membawa manfaat khususnya bagi penyusun sendiri dan umumnya bagi pembaca sekalian.

Penyusun menyadari bahwa masih banyak kekurangan dalam penyusunan buku proyek akhir ini. Oleh karena itu, besar harapan penyusun untuk menerima saran dan kritik dari pembaca.

Wassalamu'alaikum Wr. Wb.

Surabaya, Januarii 2011

Penulis

UCAPAN TERIMA KASIH

Dengan penuh rasa syukur kehadiran Allah S.W.T dan tanpa menghilangkan rasa hormat yang mendalam, saya selaku penyusun dan penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu penulis untuk menyelesaikan proyek akhir ini, penulis mengucapkan terima kasih kepada :

1. Kedua orang tua yang tercinta, Terutama Ibu dan adik-adik saya (Adit dan Ayik) tersayang yang telah memberi dukungan kepada saya baik moral maupun materi.
2. Sahabat-sahabat saya terutama Thaufan untuk bimbingannya dan Qindi untuk tumpangan tempatnya
3. Bapak Ir.Dadet Pramadihanto, M.Eng.PhD selaku Direktur Politeknik Elektronika Negeri Surabaya.
4. Bapak Arifin, ST.MT selaku Ketua Jurusan Teknik Telekomunikasi .
5. Ibu Mike Yuliana, dan Pak Nanang Syahrani, selaku dosen pembimbing. Terima kasih atas bimbingan dan bantuannya.
6. Seluruh Dosen Pengajar, terima kasih atas ilmu yang bermanfaat.
7. Seluruh teman-teman Telkom A D3 Angkatan 2008, terima kasih atas semuanya dan moment indah kebersamaan yang tidak akan pernah terlupakan. Semoga tali silaturahmi kita tidak putus dan tetap kompak.
8. Teman-teman jurusan Elektro Industri, Elektronika dan Teknologi Informasi yang selalu ada sebagai tempat sharing ilmu dalam pembuatan software, serta teman saat melepas rasa penat.
9. Dan seluruh pihak yang namanya tidak mungkin saya sebutkan satu per satu di sini. Terima kasih telah membantu saya dalam menyelesaikan studi dan proyek akhir ini.

Segala ucapan terima kasih dari penulis tentunya belum cukup, Semoga Allah SWT membalas segala kebaikan-kebaikan yang diberikan tersebut.

“Dan sebaik-baiknya insan adalah mereka yang selalu bermanfaat bagi orang lain.”

DAFTAR ISI

Lembar Pengesahan	iii
Abstrak	iv
Abstract	v
Kata Pengantar	vi
Ucapan Terima Kasih	vii
Daftar Isi	viii
Daftar Tabel	x
Daftar Gambar	xi
BAB I	
1.1 LATAR BELAKANG	1
1.2 TUJUAN	1
1.3 RUMUSAN MASALAH	2
1.4 BATASAN MASALAH	2
1.5 METODOLOGI	2
1.5.1 Studi Literatur	2
1.5.2. Perancangan Sistem	3
1.5.3 Perancangan Software	3
1.5.4 Integrasi dan Pengujian Sistem	3
1.6 SISTEMATIKA PENULISAN	3
BAB II	
2.1 Penelitian Yang Pernah Dilakukan	5
2.2 Perancangan Sistem	5
2.2.1 Diagram Alur	5
2.2.2 AHP (Analytic Hierarchy Process)	9
2.2.3 Koneksi <i>JAVA</i> dengan <i>MySQL</i>	18
BAB III	
3.1 Pembuatan Sistem	5
3.1.1 Pembuatan Database <i>MySQL</i>	21
3.1.2 Pembuatan Koneksi <i>JAVA-MySQL</i>	24
3.1.3 Pembuatan Koneksi Server ke Client	25
3.1.4 Pembuatan metode Aplik asi <i>HRIS</i>	27
3.1.5 Pembuatan Aplikasi <i>HRIS</i> Penilaian Kinerja Karyawan	27
3.1.6 Menu Utama aplikasi Penilaian Kinerja Karyawan	33
3.2 Analisa	40
3.2.1 Tujuan Analisa	40
3.2.2 Analisa Keberhasilan Sistem	41

BAB IV

4.1 Kesimpulan.....	59
4.2 Saran.....	59
DAFTAR PUSTAKA.....	61
LAMPIRAN.....	63

DAFTAR TABEL

Tabel 2. 1	Tabel Parameter AHP	11
Tabel 2. 2	Tabel Perbandingan penilaian antar elemen kriteria	13
Tabel 2. 3	Tabel Perbandingan antar Kriteria	13
Tabel 2. 4	Tabel Perbandingan 2 kriteria	14
Tabel 2. 5	Tabel Perbandingan elemen kriteria.....	15
Tabel 2. 6	Tabel Nilai Perbandingan kriteria	16
Tabel 3. 1	Tabel Bobot Kriteria	28
Tabel 3. 2	Database Bobot	32
Tabel 3. 3	Pengujian Login	41
Tabel 3. 4	Pengujian Hak Admin Dan User Biasa.....	42
Tabel 3. 5	Pengukuran lama waktu eksekusi login	43
Tabel 3. 6	Tabel Pengukuran lama waktu eksekusi login	43
Tabel 3. 7	Pengujian Algoritma AHP.....	44
Tabel 3. 8	Tampilan Aplikasi Penilaian Kinerja Karyawan.....	48
Tabel 3. 9	Apakah aplikasi ini membantu Pengaksesan Informasi.....	48
Tabel 3. 10	Manfaat Aplikasi Penilaian Kinerja Karyawan.....	49
Tabel 3. 11	Menu Pada Aplikasi Penilaian Kinerja Karyawan.....	49

DAFTAR GAMBAR

Gambar 2. 1 Skema Perancangan Sistem	6
Gambar 2. 2 Struktur Organisasi Perusahaan Asuransi	7
Gambar 2. 3 Menu Sistem	8
Gambar 2. 4 Flowchart Sistem HRIS dengan AHP	17
Gambar 3. 1 Tabel Karyawan.....	21
Gambar 3. 2 Tabel Kriteria.....	22
Gambar 3. 3 Tabel Nilai Sub Kriteria	23
Gambar 3. 4 Tabel Sub Kriteria	24
Gambar 3. 5 letak file mysql-connector-java-5.1.15-bin.jar	24
Gambar 3. 6 Setting IP Server.....	26
Gambar 3. 7 Setting IP Client	26
Gambar 3. 8 Add Data Source Name	27
Gambar 3. 9 Layar Login	32
Gambar 3. 10 Pesan Error Login	32
Gambar 3. 11 Rancangan Menu Utama	35
Gambar 3. 12 Tampilan Daftar Karyawan	36
Gambar 3. 13 Menu Masukan Nilai	36
Gambar 3. 14 Input Data Pegawai.....	37
Gambar 3. 15 Input Nilai Subkriteria	37
Gambar 3. 16 Lihat Hasil Kinerja	38
Gambar 3. 17 Pesan Nilai Ter-update	39
Gambar 3. 18 Lihat Niali Karyawan	40
Gambar 3. 19 Niali Subkriteria Karyawan NIP 1001	44
Gambar 3. 20 Hasil Perhitungan Metode AHP per Kriteria.....	46
Gambar 3. 21 Nilai Rata-rata dari 6 Kriteria.....	46
Gambar 3. 22 Uji Menu pengurutan Nilai Karyawan	47
Gambar 3. 23 Uji Menu pengurutan Kriteria Kemampuan Kerja.....	47
Gambar 3. 24 Uji Menu pengurutan Kriteria Loyalitas	48
Gambar 3. 25 Uji Menu pengurutan Kriteria Disiplin	48
Gambar 3. 26 Uji Menu pengurutan Kriteria Hubungan Kerja.....	49
Gambar 3. 27 Uji Menu pengurutan Kriteria Prestasi.....	49
Gambar 3. 28 Uji Menu pengurutan Kriteria Kepemimpinan.....	50
Gambar 3. 29 Uji Pengurutan Ranking Karyawan.....	50
Gambar 3. 30 Grafik Pengurutan Ranking Kemampuan Kerja.....	51
Gambar 3. 31 Grafik Pengurutan Ranking Loyalitas	51
Gambar 3. 32 Grafik Pengurutan Ranking Kedisiplinan	52
Gambar 3. 33 Grafik Pengurutan Ranking Hubungan Kerja	52
Gambar 3. 34 Grafik Pengurutan Ranking Prestasi	53

Gambar 3. 35 Grafik Pengurutan Ranking Kepemimpinan	53
Gambar 3. 36 Grafik Survei tampilan aplikasi.....	54
Gambar 3. 37 Grafik Survei Pengaksesan Informasi yang disukai	55
Gambar 3. 38 Grafik Survei manfaat aplikasi	55
Gambar 3. 39 Grafik Survei Kelengkapan aplikasi.....	56

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Seiring dengan perkembangan teknologi maka dibutuhkan kepraktisan dalam segala hal termasuk penerapan pada sistem HRD (Human Resource Development). Banyaknya kecurangan yang dilakukan oleh karyawan sebuah perusahaan dalam studi kasus ini adalah perusahaan asuransi mendorong munculnya suatu aplikasi untuk memonitor kinerja karyawan baik dalam hal kehadiran maupun prestasi.

Ada beberapa jenis penilaian kinerja karyawan seperti system tradisional, penilaian diri, penilaian oleh atasan, dan penilaian 360 derajat (umpan balik). Penilaian kinerja karyawan umumnya dilakukan secara formal atau terstruktur. Apabila dilakukan secara informal, manajer dapat bertemu dengan para anggota tim untuk mendiskusikan kinerja karyawan dalam periode tertentu dalam suasana rileks dan tidak kaku. Baik secara formal maupun informal, perlu ditelaah beragam faktor yang berpengaruh terhadap kinerja karyawan.

Pada tugas akhir ini akan dibuat suatu aplikasi sistem penilaian kinerja karyawan perusahaan asuransi berbasis Java karena Java merupakan Bahasa pemrograman berbasis objek dengan metode AHP, sebuah metode untuk proses pengambilan keputusan secara efektif atas permasalahan yang kompleks dan menyederhanakannya. Sistem penilaian ini akan melakukan pengambilan keputusan yang komprehensif dengan memperhitungkan hal – hal yang bersifat kualitatif & kuantitatif

1.2 TUJUAN

Tujuan proyek akhir ini adalah sebagai berikut :

Tujuan dari proyek akhir ini adalah Pembuatan aplikasi penilaian kinerja karyawan yang dapat mengolah penilaian kinerja dan tugas tugas karyawan agar dapat didokumentasikan dengan baik dan dijadikan bahan pertimbangan oleh dewan direksi atau manajer dalam hal memberikan penilaian kinerja karyawan

1.3 RUMUSAN MASALAH

Berdasarkan uraian tersebut di atas, dalam pengerjaan proyek akhir ini muncul beberapa permasalahan diantaranya adalah :

- 1) Bagaimana menentukan parameter parameter yang akan digunakan untuk penilaian kinerja karyawan
- 2) Bagaimana melakukan pembobotan perspektif untuk mengetahui bobot masing masing parameter
- 3) Bagaimana mengimplementasikan bahasa pemrograman Java yang terkoneksi dengan database *MySQL* untuk membuat sistem penilaian kinerja karyawan

1.4 BATASAN MASALAH

Batasan masalah yang harus diselesaikan pada proyek akhir ini adalah :

- 1) Bahasa pemrograman yang digunakan dalam tugas akhir ini adalah bahasa pemrograman Java.
- 2) Database yang digunakan adalah *MySQL*
- 3) Studi kasus yang dilakukan pada penelitian ini adalah penilaian kinerja karyawan yang dilakukan pada perusahaan asuransi
- 4) Parameter kinerja yang digunakan adalah :
 - Kemampuan Kerja
 - Loyalitas
 - Disiplin
 - Prestasi
 - Hubungan kerja
 - Kepemimpinan

1.5 METODOLOGI

Untuk menyelesaikan proyek akhir ini, dilakukan langkah-langkah sebagai berikut :

1.5.1 Studi literatur

Pada tahap studi literatur dilakukan pemahaman teori dari buku jurnal, paper, ulasan dari media internet mengenai Asuransi termasuk melakukan studi kasus di salah satu perusahaan asuransi di surabaya, penentuan Parameter Kinerja Karyawan, Metode AHP, database *MySQL*, dan pemrograman *JAVA*.

1.5.2 Perancangan sistem

Untuk membuat aplikasi Penilaian Kinerja Karyawan, pertama-tama dilakukan survei di perusahaan asuransi untuk mengetahui parameter penilaian kinerja karyawan, lalu data-data tersebut diolah dengan metode AHP dan diimplementasikan ke dalam pemrograman JAVA untuk menciptakan aplikasi HRIS penilaian kinerja karyawan.

1.5.3 Perancangan software

Software yang digunakan untuk pembuatan aplikasi HRIS ini antara lain Netbeans, JAVA , MySQL-FRONT untuk MySQL

1.5.4 Integrasi dan Pengujian Sistem

Setelah penghimpunan data dan perancangan sistem maupun software terlaksana, maka pengujian sistem HRIS dimulai dari :

- a. Kebenaran dari algoritma AHP yang digunakan, untuk mengetahui apakah hasil yang didapat dengan metode AHP ini sesuai dengan hasil yang sebenarnya,
- b. Pengujian kepuasan responden dengan kuisioner, diberikan kepada staff HRD dan direktur/kepala cabang, apakah aplikasi ini bermanfaat bagi mereka dalam menilai kinerja karyawan.

1.6 SISTEMATIKA PENULISAN

Sistematika pembahasan dari Proyek Akhir ini direncanakan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang pendahuluan yang terdiri dari latar belakang, perumusan masalah, batasan masalah, tujuan dan sasaran, metodologi, serta sistematika pembahasan dari Proyek Akhir ini.

BAB II PERANCANGAN SISTEM DAN TEORI PENUNJANG

Bab ini membahas mengenai teori-teori yang berkaitan dengan penyelesaian Proyek Akhir, yang didapatkan dari

berbagai macam buku serta sumber-sumber terkait lainnya yang berhubungan dengan pembuatan Proyek Akhir ini.

BAB III PEMBUATAN SISTEM DAN ANALISA

Bab ini membahas mengenai pembuatan sistem, meliputi metode, pengolahan data, dan analisa hasil metode AHP.

BAB IV PENUTUP

Bab ini berisi kesimpulan dari uji coba dan analisa yang telah di dapat, dan saran untuk pengembangan, dan penyempurnaan terhadap aplikasi yang telah dibuat.

BAB II

PERANCANGAN SISTEM DAN TEORI PENUNJANG

2.1 Penelitian Yang Pernah Dilakukan

Sistem Pendukung Penilaian Kinerja Karyawan oleh Bambang Eka Pamungkas [1] yang dilakukan di PT.Multiterminal Indonesia pada tahun 2009, yaitu program yang dibuat untuk membantu proses penilaian karyawan dengan metode AHP menggunakan VisualBasic sebagai dasar pemrogramannya, parameter yang digunakan adalah Kedisiplinan, Pengalaman Kerja, prestasi, dan perilaku.

Johannes Sinaga, “ Penerapan *Analitycal Hierarchy Process* (AHP) Dalam Pemilihan Perusahaan Badan Usaha Milik Negara (BUMN) Sebagai Tempat Kerja Mahasiswa Universitas Sumatera Utara ” menggunakan metode AHP untuk menentukan urutan prioritas perusahaan BUMN (PERSERO) yang akan dipilih mahasiswa USU sebagai tempat bekerja dan mengarahkan mahasiswa untuk menentukan di perusahaan BUMN manakah peluang mereka lebih besar untuk mendapatkan pekerjaan, namun masih secara manual dengan parameter Gaji, Jenjang Karir, Fasilitas, dan Penempatan.

Pada tugas akhir ini akan dibuat aplikasi penilaian kinerja karyawan di perusahaan Asuransi berbasis JAVA dan menggunakan *MySQL* sebagai *database*-nya. Aplikasi yang dibuat menggunakan beberapa parameter untuk penilaian yang meliputi Kemampuan kerja, Loyalitas, Disiplin, Prestasi, Hubungan Kerja, dan kepemimpinan.

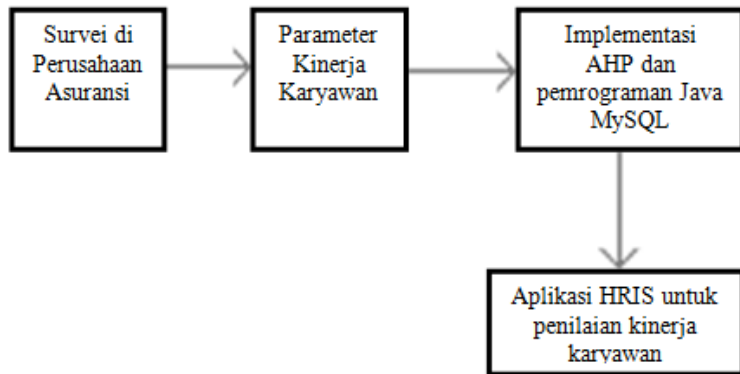
2.2 Perancangan Sistem

2.2.1 Diagram Alur

Untuk membuat aplikasi Penilaian Kinerja Karyawan di butuhkan langkah-langkah sebagaimana tertera pada diagram alur pada gambar 2.1 dibawah ini :

1. Survei di Perusahaan Asuransi
 2. Penentuan Parameter Kinerja Karyawan
 3. Implementasi Metode AHP dan Pemrograman Java
 4. Pembuatan HRIS
-

Pada Gambar 2.1 dijelaskan proses pembuatan aplikasi penilaian kinerja, pertama-tama dilakukan survei di perusahaan asuransi untuk mengetahui parameter penilaian kinerja karyawan, lalu data-data tersebut diolah dengan metode AHP dan diimplementasikan ke dalam pemrograman JAVA untuk menciptakan aplikasi HRIS penilaian kinerja karyawan.



Gambar 2.1 Skema perancangan sistem

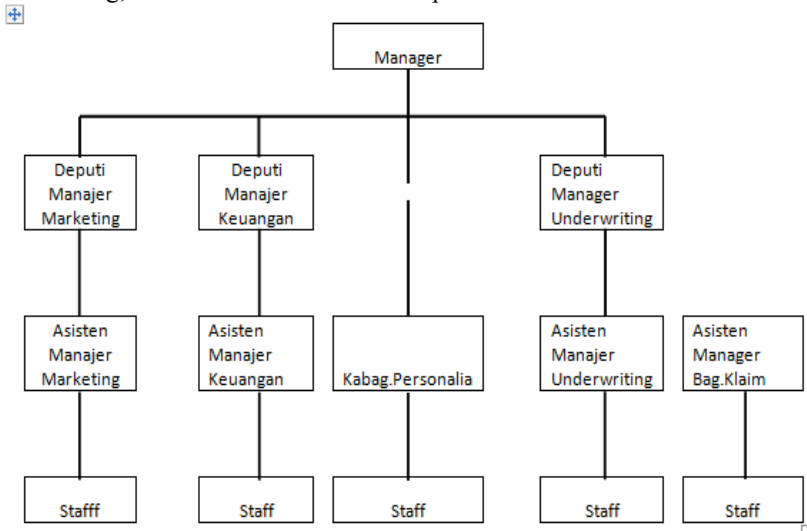
Perusahaan Asuransi

A. Struktur Perusahaan

Perusahaan Asuransi adalah perusahaan yang memiliki program perlindungan nilai ekonomi bertanggung untuk nasabahnya, jika bertanggung mengalami suatu musibah. musibah yang dimaksud adalah termasuk karena : meninggal, sakit parah yang mengakibatkan bertanggung tidak lagi bisa bekerja kembali, kecelakaan, bencana alam, teroris (dilihat case per case) atau jika bertanggung kehilangan pemegang polis sesuai dengan perjanjian atau kesepakatan yang telah dibuat sebelumnya.

Struktur Perusahaan Asuransi biasanya dipimpin oleh seorang Presiden Direktur yang membawahi beberapa direktur seperti direktur finansial, direktur marketing, direktur IT, dan direktur HRD yang

masing-masing membawahi senior manajer dengan tugasnya masing-masing, berikut contoh struktur dari perusahaan asuransi :



Gambar 2.2 Struktur Organisasi dari Perusahaan Asuransi [4]

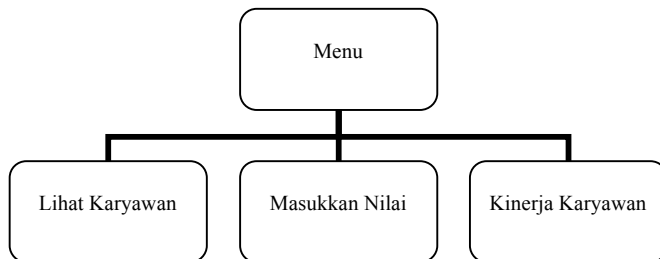
B. Parameter Kinerja

Parameter penilaian kinerja karyawan yang akan digunakan dalam penelitian ini yaitu :

- Kemampuan Kerja: parameter ini menilai kemampuan karyawan dalam menyelesaikan tugas yang dibebankan padanya, kemampuan karyawan mengatasi tekanan kerja, dan ketuntasan bekerja dan hasil dari pekerjaannya
- Loyalitas : Parameter loyalitas yaitu sejauh mana karyawan tersebut setia pada perusahaan, dalam hal ini tidak membocorkan rahasia perusahaan ke pihak luar
- Disiplin : Kedisiplinan merupakan salah satu faktor penting dalam penilaian karyawan terbaik, akan digunakan absensi baik kehadiran/ jam masuk/ absen/ lembur masing-masing karyawan sebagai penilaian untuk parameter ini

- Prestasi : Hasil kerja dari karyawan, kualitas dari hasil tugas yang dibebankan pada karyawan
- Hubungan Kerja : Hubungan antar karyawan dengan pimpinan, karyawan dan karyawan, serta karyawan dengan relasi perusahaan juga memiliki penilaian tersendiri
- Kepemimpinan : Berkaitan dengan kemampuan karyawan tersebut mempengaruhi karyawan lain yang berakibat positif/meningkatkan kinerja karyawan lain, atau kemampuan karyawan berpikir kreatif dalam menyelesaikan permasalahan saat tidak ada pimpinan yang hadir.

1. Menu Sistem



Gambar 2.3 Menu Sistem

Pada Gambar 2.3 Dijelaskan Menu tampilan awal dari sistem yang akan dibuat, terdiri dari “Lihat Karyawan”, yaitu menu untuk melihat daftar karyawan yang ada. Lalu menu “Masukkan Nilai” untuk memasukkan nilai rata-rata dari tiap kriteria, menu ini hanya bisa diakses oleh pihak yang memiliki *privileges* untuk melakukan penilaian. Pada menu “Kinerja Karyawan” merupakan menu dengan *privileges* khusus yang hanya bisa diakses pihak yang berhak melakukan penilaian, dengan memilih menu ini maka seluruh data dari hasil perhitungan dengan metode AHP akan di-*rank*-ing sehingga terlihat hasil kinerja karyawan

2.2.2 AHP (*Analytic Hierarchy Process*)

Metode AHP dikembangkan oleh Thomas L. Saaty, seorang ahli matematika. Metode ini adalah sebuah kerangka untuk mengambil keputusan dengan efektif atas persoalan yang kompleks dengan menyederhanakan dan mempercepat proses pengambilan keputusan dengan memecahkan persoalan tersebut kedalam bagian-bagiannya, menata bagian atau variabel ini dalam suatu susunan hirarki, member nilai numerik pada pertimbangan subjektif tentang pentingnya tiap variabel dan mensintesis berbagai pertimbangan ini untuk menetapkan variabel yang mana yang memiliki prioritas paling tinggi dan bertindak untuk mempengaruhi hasil pada situasi tersebut. Metode AHP ini membantu memecahkan persoalan yang kompleks dengan menstruktur suatu hirarki kriteria, pihak yang berkepentingan, hasil dan dengan menarik berbagai pertimbangan guna mengembangkan bobot atau prioritas. Metode ini juga menggabungkan kekuatan dari perasaan dan logika yang bersangkutan pada berbagai persoalan, lalu mensintesis berbagai pertimbangan yang beragam menjadi hasil yang cocok dengan perkiraan kita secara intuitif sebagaimana yang dipresentasikan pada pertimbangan yang telah dibuat. (Saaty, 1993)[3]

Menyusun Hirarki

Menurut Saaty, ada tiga prinsip dalam memecahkan persoalan dengan AHP, yaitu prinsip menyusun hirarki (*Decomposition*), prinsip menentukan prioritas (*Comparative Judgement*), dan prinsip konsistensi logis (*Logical Consistency*). Hirarki yang dimaksud adalah hirarki dari permasalahan yang akan dipecahkan untuk mempertimbangkan kriteria-kriteria atau komponen-komponen yang mendukung pencapaian tujuan. Dalam proses menentukan tujuan dan hirarki tujuan, perlu diperhatikan apakah kumpulan tujuan beserta kriteria-kriteria yang bersangkutan tepat untuk persoalan yang dihadapi. Dalam memilih kriteria-kriteria pada setiap masalah pengambilan keputusan perlu memperhatikan kriteria-kriteria sebagai berikut:

- a. Lengkap
Kriteria harus lengkap sehingga mencakup semua aspek yang penting, yang digunakan dalam mengambil keputusan untuk pencapaian tujuan.
- b. Operasional
Operasional dalam artian bahwa setiap kriteria ini harus mempunyai arti bagi pengambil keputusan, sehingga benar-benar dapat menghayati terhadap alternatif yang ada, disamping terhadap sarana untuk membantu penjelasan alat untuk berkomunikasi.
- c. Tidak berlebihan
Menghindari adanya kriteria yang pada dasarnya mengandung pengertian yang sama.
- d. Minimum
Dusahakan agar jumlah kriteria seminimal mungkin untuk mempermudah pemahaman terhadap persoalan, serta menyederhanakan persoalan dalam analisis.

Decomposition

Setelah persoalan didefinisikan maka perlu dilakukan *decomposition*, yaitu memecah persoalan yang utuh menjadi unsur-unsurnya. Jika ingin mendapatkan hasil yang akurat, pemecahan juga dilakukan terhadap unsur-unsurnya sehingga didapatkan beberapa tingkatan dari persoalan tadi. Karena alasan ini maka proses analisis ini dinamai hirarki (*Hierarchy*). Pembuatan hirarki tersebut tidak memerlukan pedoman yang pasti berapa banyak hirarki tersebut dibuat, tergantung dari pengambil keputusan-lah yang menentukan dengan memperhatikan keuntungan dan kerugian yang diperoleh jika keadaan tersebut diperinci lebih lanjut. Ada dua jenis hirarki, yaitu hirarki lengkap dan hirarki tidak lengkap. Dalam hirarki lengkap, semua elemen pada semua tingkat memiliki semua elemen yang ada pada tingkat berikutnya. Jika tidak demikian maka dinamakan hirarki tidak lengkap.

Comparatif Judgement

Prinsip ini berarti membuat penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu dalam kaitannya dengan tingkat yang di atasnya. Penilaian ini merupakan inti dari AHP, karena akan berpengaruh terhadap prioritas elemen-elemen. Hasil dari penilaian ini

akan ditempatkan dalam bentuk matriks yang dinamakan matriks *pairwise comparison*. Dalam melakukan penilaian terhadap elemen-elemen yang diperbandingkan terdapat tahapan-tahapan, yakni:

- Elemen mana yang lebih (penting/disukai/berpengaruh/lainnya)
- Berapa kali sering (penting/disukai/berpengaruh/lainnya)

Agar diperoleh skala yang bermanfaat ketika membandingkan dua elemen, perlu dipahami tujuan yang diambil secara umum. Dalam penyusunan skala kepentingan, Saat menggunakan patokan pada tabel berikut.

Tabel 2.1 Parameter AHP

Intensitas Kepentingan nya	Definisi	Penjelasan
1	Kedua elemen sama pentingnya	Dua elemen menyumbang sama besar pada sifat itu
3	Elemen yang satu sedikit lebih penting ketimbang yang lainnya	Pengalaman dan pertimbangan sedikit menyokong satu elemen atas yang lainnya
5	Elemen yang satu esensial atau sangat penting ketimbang elemen yang lainnya	Pengalaman dan pertimbangan dengan kuat menyokong satu elemen atas elemen yang lainnya
7	Satu elemen jelas lebih penting dari elemen yang lainnya	Satu elemen dengan kuat disokong, dan dominannya telah terlihat dalam praktek
9	Satu elemen mutlak lebih penting ketimbang elemen yang lainnya	Bukti yang menyokong elemen yang satu atas yang lain memiliki tingkat penegasan tertinggi yang mungkin menguatkan
2,4,6,8	Nilai-nilai diantara dua pertimbangan yang berdekatan	Kompromi diperlukan antara dua pertimbangan
Kebalikan	Jika untuk aktivitas i mendapat satu angka bila dibandingkan dengan suatu aktivitas j , maka j mempunyai nilai kebalikannya bila dibandingkan dengan aktivitas i .	

Dalam penilaian kepentingan relative dua elemen berlaku aksioma reciprocal, artinya jika elemen i dinilai 3 kali lebih penting dibanding j , maka elemen j harus sama dengan $1/3$ kali pentingnya dibanding elemen i . Disamping itu, perbandingan dua elemen yang sama akan menghasilkan angka 1, artinya sama penting. Dua elemen yang berlainan dapat saja dinilai sama penting. Jika terdapat m elemen, maka akan diperoleh matriks *pairwise comparison* berukuran $m \times n$. Banyaknya penilaian yang diperlukan dalam menyusun matriks ini adalah $n(n-1)/2$ karena matriks *reciprocal* dan elemen-elemen diagonalnya sama dengan 1.

Synthesis of Priority

Dari setiap matriks *pairwise comparison* kemudian dicari nilai *eigen vectornya* untuk mendapatkan *local priority*. Karena matriks-matriks *pairwise comparison* terdapat pada setiaptingkat, maka untuk mendapatkan *global priority* harus dilakukan sintesis antara *local priority*. Pengurutan elemen-elemen menurut kepentingan relatif melalui prosedur sintesis dinamakan *priority setting*.

Logical Consistency

Konsistensi memiliki dua makna, pertama adalah objek-objek yang serupa dapat dikelompokkan sesuai dengan keseragaman dan relevansi. Arti kedua adalah menyangkut tingkat hubungan antara objek-objek yang didasarkan pada kriteria tertentu.

Penggunaan Metode AHP

Setelah menentukan kriteria, langkah selanjutnya yaitu perbandingan antar elemen yaitu kriteria. Dengan Kata lain, penilaian ini dimaksudkan untuk melihat seberapa penting suatu pilihan kriteria dilihat dari kriteria yang ada. Tabel 1 merupakan contoh pembobotan dengan memperhatikan perbandingan antar elemen kriteria yang paling sering digunakan karena dinilai lebih logis (Bourgeois, 2005).[2]

Tabel 2.2 Tabel perbandingan skala penilaian antar elemen kriteria

Hasil penilaian	Kriteria A	Kriteria B
A sangat jauh lebih penting dari B	1,9	0,1
A jauh lebih penting dari B	1,6	0,4
A sedikit lebih penting dari B	1,3	0,7
A dan B sama penting	1,0	1,0
A sedikit kurang penting dari B	0,7	1,3
A jauh kurang penting dari B	0,4	1,6
A sangat jauh kurang penting dari B	0,1	1,9

Sumber : Bourgeois (2005)

Dengan menggunakan penilaian seperti Tabel 1, maka perbandingan antar kriteria akan menghasilkan Tabel 2 di bawah ini. Untuk memudahkan, dalam tabel diasumsikan hanya ada empat kriteria.

Tabel 2.3 Tabel perbandingan antar kriteria

Kriteria	CR_1	CR_2	CR_3	CR_4	jumlah	bobot
CR_1	-	C_{12}	C_{13}	C_{14}	C_1	$B_{C1} = C_1 / C$
CR_2	C_{21}	-	C_{23}	C_{24}	C_2	$B_{C2} = C_2 / C$
CR_3	C_{31}	C_{32}	-	C_{34}	C_3	$B_{C3} = C_3 / C$
CR_4	C_{41}	C_{42}	C_{43}	-	C_4	$B_{C4} = C_4 / C$
					C	

CR_1 adalah Kriteria ke-1. C_{12} berarti perbandingan antara Kriteria 1 dengan Kriteria 2. C_1 adalah jumlah perbandingan Kriteria 1 dengan kriteria lainnya.

Dari tabel 2.3 dapat dirangkum sebagai berikut:

- a) merupakan hasil penilaian/perbandingan antara kriteria i dengan kriteria j $i \neq j$
- b) merupakan penjumlahan nilai yang dimiliki kriteria ke i
- c) c merupakan penjumlahan semua nilai i
- d) Bobot kriteria ke i (w_i) diperoleh dengan membagi nilai c_i dengan c

Dari perhitungan table diatas maka dapat diketahui nilai bobot untuk masing-masing kriteria. Untuk pembobotan sendiri terdapat banyak pilihan. Untuk kasus tertentu semua indikator mempunyai bobot yang sama. Tetapi pada kasus dengan tahap seperti diatas, maka tiap indikator mempunyai bobot yang berbeda sesuai dengan kepentingannya seperti tahap-tahap diatas.

Dibawah ini akan diberikan **contoh** penggunaan metode AHP untuk pemilihan karyawan terbaik dengan parameter Prestasi, Perilaku, Pengalaman, dan Kedisiplinan.

Pada proses ini perhitungan bobot dengan metode AHP diterapkan. Pengujian ini dimulai dari proses perbandingan elemen dan berlanjut ke proses perhitungan nilai bobot. Pada proses pengujian ini diberikan contoh dari tahap perbandingan elemen dan nantinya perhitungan bobot disajikan melalui perhitungan manual dan perhitungan yang dihasilkan oleh perangkat lunak. Pada pengujian perhitungan manual ini terdapat empat kriteria yang akan dilakukan perhitungan nilai bobot, maka dibuat tabel sebagai berikut :

Tabel 2.4 Perbandingan 2 kriteria

KRITERIA A	KRITERIA B
Kemampuan Kerja	Kemampuan Kerja
Loyalitas	Loyalitas
Disiplin	Disiplin
Hubungan Kerja	Hubungan Kerja
Prestasi	Prestasi
Kepemimpinan	Kepemimpinan

Tahap perbandingan elemen dari kriteria-kriteria diatas sebagai bahan pengujian adalah sebagai berikut:

Tabel 2.5 Tabel perbandingan elemen criteria

Perbandingan Kriteria		Opsì Perbandingan
A	B	
Kemampuan Kerja	Loyalitas	A Lebih penting dari B
Kemampuan Kerja	Disiplin	A Lebih penting dari B
Kemampuan Kerja	Prestasi	A Sama penting dengan B
Kemampuan Kerja	Hubungan Kerja	A Lebih penting dari B
Kemampuan Kerja	Kepemimpinan	A Lebih penting dari B
Loyalitas	Disiplin	B Sedikit lebih penting dari A
Loyalitas	Prestasi	B Lebih penting dari A
Loyalitas	Kepemimpinan	B Lebih penting dari A
Loyalitas	Hubungan Kerja	B Jauh lebih penting dari A
Disiplin	Prestasi	B Sedikit lebih penting dari A
Disiplin	Hubungan Kerja	B Sedikit lebih penting dari A
Disiplin	Kepemimpinan	A Sama penting dengan B
Prestasi	Hubungan Kerja	A Jauh lebih penting dari B
Prestasi	Kepemimpinan	A Jauh lebih penting dari B
Hubungan Kerja	Kepemimpinan	B Lebih penting dari A

Dari tabel 2.5 maka akan menghasilkan nilai kriteria dari masing masing perbandingan yaitu:

Tabel 2.6 Tabel nilai perbandingan kriteria

Kriteria	Kemampuan Kerja	Loyalitas	Disiplin	Prestasi	Hubungan Kerja	Kepemimpinan
Kemampuan Kerja	-	1.5	1.4	1	1.6	1.6
Loyalitas	0.5	-	0.8	0.6	0.4	0.5
Disiplin	0.6	1.2	-	0.7	0.8	1
Prestasi	1	1.4	1.3	-	1.4	1.4
Hubungan Kerja	0.4	1.6	1.2	0.6	-	0.7
Kepemimpinan	0.4	1.5	1	0.6	1.3	-

Dari nilai masing-masing kriteria diatas dapat diketahui nilai bobot dari masing-masing kriteria yaitu dengan membagi Jumlah nilai masing-masing kriteria dengan nilai C.

Kemampuan Kerja = $7.1 / 30 = \mathbf{0,2366}$

Loyalitas = $2.8 / 30 = \mathbf{0,093}$

Disiplin = $4.3 / 30 = \mathbf{0,1433}$

Prestas = $6.5 / 30 = \mathbf{0,2166}$

Hubungan Kerja = $4.5 / 30 = \mathbf{0.15}$

Kepemimpinan = $4.8 / 30 = \mathbf{0.16}$

Nilai bobot yang dijumlahkan akan bernilai 1.

$$0.2366 + 0.093 + 0.1433 + 0.2166 + 0.15 + 0.16 = 1$$

Pengujian Proses Perhitungan Nilai Kinerja

Proses perhitungan nilai kinerja atau disebut sebagai sintesa penilaian, diberikan beberapa contoh pengujian perhitungan.

Perhitungan ini digunakan untuk menghasilkan nilai option yang diperoleh dari perkalian nilai bobot dengan nilai option subkriteria.

Perhitungan disesuaikan dengan kebutuhan sistem yaitu dengan mengambil tiga karakter hasil perhitungan dari kiri. Contoh pengujian dari proses ini adalah sebagai berikut.

Nama karyawan : Luqman Putra Wibowo

Kriteria : Kemampuan Kerja

Bobot : 0,2366

Nilai : 9

Hasil : $0,2366 \times 9 = 2,1294$ (2,2)

(Hasil akhir adalah nilai dikalikan dengan hasil pembobotan)

Setelah itu, hasil akhir dari tiap kriteria akan dijumlahkan untuk mendapatkan nilai kinerja karyawan dimana nilai tersebut akan dibandingkan dengan nilai dari karyawan lain sehingga bisa dilihat siapa karyawan dengan nilai kinerja terbaik

Contoh :

Luqman : 8,5 (hasil dari 4 Kriteria)

Budi : 8,3

Fanny : 9.6

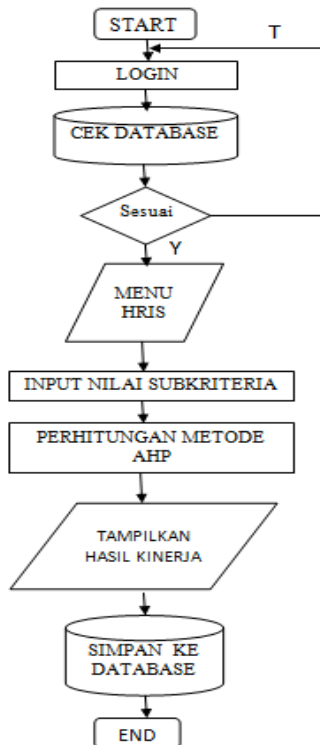
Nilai tersebut akan diurutkan sehingga terlihat nilai yg terbaik :

Fanny : 9.6

Luqman : 8.5

Budi : 8.3

Flowchart Sistem HRIS



Gambar 2.4 Flowchart Sistem HRIS dengan AHP

Pada Gambar 2.4 diatas, dijelaskan flowchart dari integrasi sistem. Dimulai penentuan parameter kinerja yang akan digunakan dengan melakukan survei di perusahaan asuransi, kemudian diambil nilai tiap parameter dari karyawan-karyawan perusahaan yang kemudian diolah dengan metode AHP yang akan diintegrasikan dengan pemrograman JAVA sehingga dapat menghasilkan aplikasi HRIS Kinerja karyawan

2.2.3 Konektivitas JAVA dengan MySQL

1. Membuka koneksi ke server MySQL

mysql_connect()

Digunakan untuk melakukan uji dan koneksi kepada server database MySQL.

Sintaks :

\$conn = mysql_connect ("host","username","password");

\$conn adalah nama variabel penampung status hasil koneksi kepada database.

host adalah nama host atau alamat server database MySQL.

username adalah nama *user* yang telah diberi hak untuk dapat mengakses server database.

password adalah kata sandi untuk *username* untuk dapat masuk ke dalam database.

2. Memilih database yang akan digunakan di server

mysql_select_db()

Digunakan untuk melakukan koneksi kepada database yang dalam server yang berhasil dikoneksi dengan perintah mysql_connect().

Sintaks :

\$pilih = mysql_select_db("namadatabase",\$conn);

\$pilih berisi status koneksi kepada database.

\$conn merupakan koneksi kepada server database yang berhasil.

namadatabase adalah nama database yang akan dikenai proses.

3. Mengambil sebuah query dari sebuah database.

mysql_query()

Digunakan untuk melakukan eksekusi perintah SQL untuk memanipulasi database yang berhasil dilakukan koneksinya menggunakan `mysql_select_db()`.

Sintaks :

\$hasil = mysql_query("SQL Statement");

\$hasil akan berupa record set apabila SQL Statement berupa perintah select.

4. Mengambil record dari database

a. `mysql_fetch_array()`

Digunakan untuk melakukan pemrosesan hasil query yang dilakukan dengan perintah `mysql_query()`, dan memasukkannya ke dalam array asosiatif, array numeris atau keduanya.

Sintaks :

\$row = mysql_fetch_array(\$hasil);

\$row adalah array satu record dari record \$hasil yang diproses nomor record sesuai dengan nomor urut dari proses `mysql_fetch_array` yang sedang dilakukan.

\$hasil adalah record set yang akan diproses.

b. `mysql_fetch_assoc()`

Fungsi ini hampir sama dengan fungsi `mysql_fetch_array()`, hanya saja array yang dihasilkan hanya array asosiatif.

Sintaks :

\$row = mysql_fetch_assoc(\$hasil);

c. `mysql_fetch_row()`

Fungsi ini hampir sama dengan fungsi `mysql_fetch_array()`, hanya saja array yang dihasilkan hanya array numeris.

Sintaks :

\$row = mysql_fetch_row(\$hasil);

d. `mysql_num_rows()`

Fungsi ini digunakan untuk menghitung jumlah record yang ada pada database.

Sintaks :

\$jml = mysql_num_rows(\$hasil);

\$jml akan memiliki nilai sesuai dengan jumlah record yang ada.

[Halaman ini sengaja dikosongkan]

BAB III

PEMBUATAN DAN ANALISA SISTEM

3.1 PEMBUATAN SISTEM

Pada bab ini akan dibahas pembuatan sistem secara keseluruhan yang meliputi :

1. Pembuatan database HRIS menggunakan Java-MySQL
2. Pembuatan Koneksi Java-MySQL
3. Pembuatan koneksi Server-Client Java-MySQL
4. Pembuatan Metode AHP
5. Pembuatan aplikasi Penilaian Kinerja Karyawan dengan pemrograman JAVA

3.1.1 Pembuatan Database dengan MySQL

Untuk Pembuatan database, dibuat tabel-tabel database berikut :

1. Tabel Karyawan

Tabel Karyawan merupakan tabel untuk menyimpan data karyawan perusahaan, juga tipe *user* mereka.

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	NIP	varchar(255)	latin1_swedish_ci		No	None		
☐	nama_karyawan	varchar(255)	latin1_swedish_ci		No	None		
☐	user	varchar(255)	latin1_swedish_ci		No	None		
☐	password	varchar(255)	latin1_swedish_ci		No	None		
☐	Divisi	varchar(255)	latin1_swedish_ci		No	None		
☐	Jabatan	varchar(255)	latin1_swedish_ci		No	None		

		NIP	nama_karyawan	user	password	Divisi	Jabatan
☐		1000	Suyanto Sutedja	admin	pas	Marketing	Deputi Manager
☐		1001	R.Budi Mulyono	bud12	123	Personalia	Deputi Manager
☐		1002	Agnes Fenty B	agn13	agag	Underwriting	Deputi Manager
☐		1003	Dedy Sunardi	ded13	123	Keuangan	Deputi Manager
☐		1004	Valentino Devlin	ded12	123	Keuangan	Staff
☐		1005	Hendriyanto	ded12	123	Marketing	Staff
☐		1006	Danang Sutarto	ded12	123	Personalia	Staff
☐		1007	I nengah Sutarsa	ded12	123	Personalia	Staff
☐		1008	Budi Sinjaya	ded12	123	Marketing	Staff
☐		1009	Veronika Verasanti	ded12	123	Marketing	Kepala Bagian
☐		1010	Christian Chandra	ded12	123	Keuangan	Kepala Bagian
☐		1011	Sudarman	ded12	123	Underwriting	Staff
☐		1012	Bowo Nugroho	ded12	123	Personalia	Kepala Bagian

Gambar 3.1 Database Tabel Karyawan

2. Tabel Kriteria

Tabel yang berisi Kriteria yang digunakan dalam metode AHP, juga berisi Bobot tiap kriteria sesuai dengan perhitungan metode AHP

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	ID_Kriteria	varchar(255)	latin1_swedish_ci		No	None		     
☐	Nama_kriteria	varchar(255)	latin1_swedish_ci		No	None		     
☐	Bobot	varchar(255)	latin1_swedish_ci		No	None		     

Gambar 3.2 Database Tabel Kriteria

3. Tabel Nilai Subkriteria

Tabel Nilai Subkriteria berisi nilai subkriteria dari tiap kriteria yang ada, nilai yang ada di tabel ini akan masuk ke dalam formula perhitungan metode AHP

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	NIP	varchar(4)	latin1_swedish_ci		No	None		     
☐	sk1	varchar(3)	latin1_swedish_ci		No	None		     
☐	sk2	varchar(3)	latin1_swedish_ci		No	None		     
☐	sk3	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk4	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk5	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk6	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk7	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk8	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk9	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk10	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk11	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk12	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk13	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk14	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk15	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk16	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk17	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk18	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk19	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk20	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk21	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk22	varchar(5)	latin1_swedish_ci		No	None		     
☐	sk23	varchar(5)	latin1_swedish_ci		No	None		     
☐	nilaiRataRata	float			No	None		     

NIP	sk1	sk2	sk3	sk4	sk5	sk6	sk7	sk8	sk9	sk10	sk11	sk12	sk13	sk14	sk15	sk16	sk17	sk18	sk19	sk20	sk21	sk22	sk23	nilaiRataRata
1000	80	80	70	70	85	90	80	80	80	80	80	80	80	80	80	80	80	80	80	80	70	75	70	12.8709
1001	80	80	80	80	80	80	80	70	75	70	70	70	70	75	90	90	85	90	90	70	90	80	80	13.0631
1002	90	80	90	80	90	70	70	60	90	80	70	90	90	90	70	70	80	80	80	90	80	90	80	13.5131
1003	80	80	90	80	70	90	70	90	90	80	80	90	85	85	98	75	95	85	90	80	70	90	90	13.9251
1004	90	80	90	80	90	70	70	60	90	80	70	90	90	90	70	70	80	80	80	90	80	90	80	13.5131
1005	90	80	90	80	90	70	70	60	90	80	70	90	90	90	70	70	80	80	80	90	80	90	80	13.5131
1006	90	70	90	80	90	90	90	90	90	70	90	70	90	69	80	80	85	95	90	80	90	60	80	13.5671
1007	90	70	90	80	90	90	90	90	90	70	90	70	90	69	80	80	85	95	90	80	90	60	80	13.5671
1008	90	70	90	80	90	90	90	90	90	70	90	70	90	69	80	80	85	95	90	80	90	60	80	13.5671
1009	90	70	90	80	90	90	90	90	90	70	90	70	90	69	80	80	85	95	90	80	90	60	80	13.5671
1010	90	70	90	80	90	90	90	90	90	70	90	70	90	69	80	80	85	95	90	80	90	60	80	13.5671
1011	90	70	90	80	90	90	90	90	90	70	90	70	90	69	80	80	85	95	90	80	90	60	80	13.5671
1012	90	70	90	80	90	90	90	90	90	70	90	70	90	69	80	80	85	95	90	80	90	60	80	13.5671

Gambar 3.3 Tabel Database Nilai sub Kriteria

4. Tabel Subkriteria

Tabel ini berisi 23 Subkriteria yang akan digunakan dalam formula perhitungan metode AHP

sk1	Ketekunan	k1
sk2	Ketelitian	k1
sk3	Perencanaan Kerja	k1
sk4	Hasil Kerja	k1
sk5	Sikap Pada Pekerjaan	k2
sk6	Motivasi	k2
sk7	Kejujuran	k2
sk8	Tanggung Jawab	k2
sk9	Kehadiran	k3
sk10	Efisiensi Kerja	k3
sk11	Kepatuhan Peraturan	k3
sk12	Kerapihan	k3
sk13	Kualitas Kerja	k4
sk14	Wawasan Pekerjaan	k4
sk14	Wawasan Pekerjaan	k4
sk15	Inisiatif dan Kreatifitas Kerja	k4
sk16	Tugas Belajar	k4
sk17	Kemampuan Kerjasama	k5
sk18	Kemampuan Berkomunikasi	k5
sk19	Kemampuan Menerima Kritik	k5
sk20	kemampuan Memberi saran	k5
sk21	Koordinasi	k6
sk22	Kemampuan Mengambil Keputusan	k6
sk23	Ketegasan Sikap	k6

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	id_subkrit	varchar(4)	latin1_swedish_ci		No	None		      
☐	nama_subkriteria	varchar(255)	latin1_swedish_ci	Swedish, case-insensitive	No	None		      
☐	ID_Kriteria	varchar(3)	latin1_swedish_ci		No	None		      

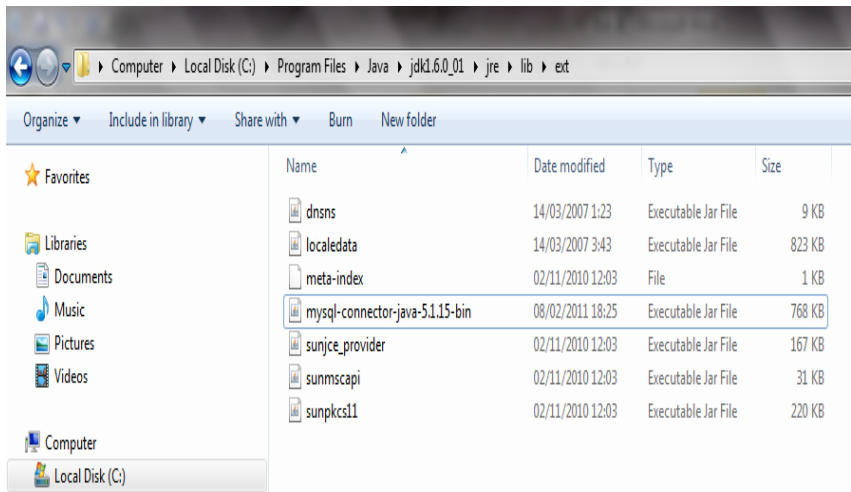
Gambar 3.4 Database Sub Kriteria

3.1.2 Pembuatan Koneksi JAVA – MySQL

Pada pembuatan koneksi JAVA – MySQL ini menggunakan JDBC, *software* yang diperlukan JDBC *driver* dan MySQL.

Pada Netbeans IDE 6.9.1 yang harus dilakukan pada saat melakukan koneksi database dengan bahasa pemrograman java adalah pengaktifkan driver JDBC terlebih dahulu caranya adalah sebagai berikut :

- Download Driver JDBC : *mysql-connector-java-5.1.15-bin.jar*
- Copy file *mysql-connector-java-5.1.15-bin.jar* ke C:\Program Files\Java\jdk1.6.0_01\jre\lib\ext



Gambar 3.5 Letak file *mysql-connector-java-5.1.15-bin.jar*

Fungsi main untuk koneksi ke database adalah sebagai berikut :

Kode program 3.1 Koneksi JAVA – MySQL

```
import java.sql.DriverManager;
import java.sql.ResultSet;

public class coba_database extends javax.swing.JFrame {
    private void cobaActionPerformed(java.awt.event.ActionEvent evt)
    {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("Berhasil");
        } catch (Exception e) {
            System.out.println("Driver not found");
        }
        String URL = "jdbc:mysql://localhost:3306/ncc";
        String user = "root";
        String pwd = "";

        try {
            java.sql.Connection conn= DriverManager.getConnection(URL, user,
            pwd);
            java.sql.Statement stmt = conn.createStatement();

        } catch (Exception e2) {
            System.out.println("Koneksi gagal");
        }
    }
}
```

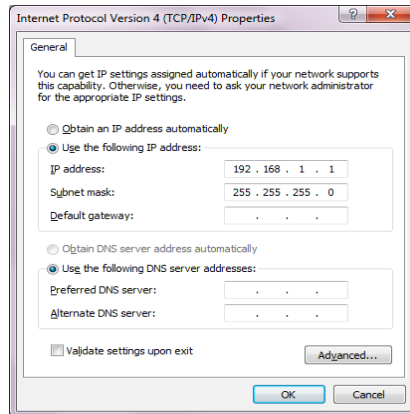
Jika Koneksi Berhasil maka Output yang dihasilkan adalah string Berhasil, Tetapi jika koneksi gagal maka Output nya adalah String Koneksi Gagal.

3.1.3 Pembuatan Koneksi Server ke Client

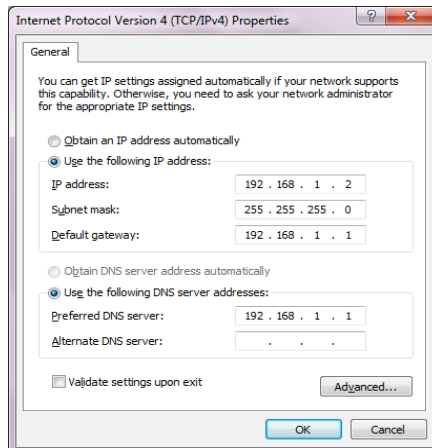
Database dengan menggunakan sistem Client-Server merupakan sesuatu hal digunakan pada sistem antrian ini. Hal ini merupakan suatu sistem dasar yang digunakan dalam merancang database pada sistem antrian rumah sakit yang memakai beberapa komputer didalam suatu jaringan. Sistem ini digunakan pada jaringan lokal atau LAN. Salah satu komputer akan bertugas sebagai database server dan yang lain akan bertugas sebagai client yang memiliki aplikasi. Untuk itu harus terlebih dahulu dilakukan konfigurasi sebagai berikut :

- Hubungkan komputer client dan komputer server menggunakan kabel LAN.

- Setting Internet Protocol sebagai berikut (sebagai contoh) :
IP_Server = 192.168.1.1
IP_Client = 192.168.1.2



Gambar 3.6 Setting IP Server

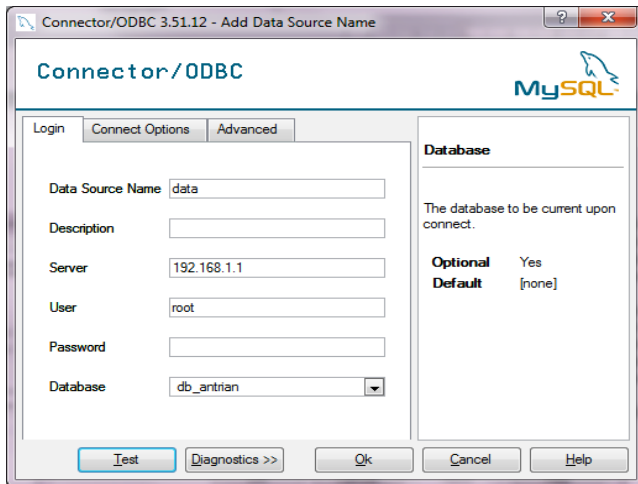


Gambar 3.7 Setting IP Client

- Pada client, Default Gateway dan Preferred DNS Server diisi dengan IP Address dari komputer yang bertindak sebagai server.
- Pastikan setting firewall telah diatur sehingga tidak memblokir koneksi yang telah dibuat.

Agar komputer client mampu membaca database yang ada pada komputer server pastikan database yang digunakan bisa di akses oleh semua nomor IP. Selain itu, install MySQL ODBC Connector dengan konfigurasi sebagai berikut :

- Masuk ke **Control Panel > Administrative Tools > Data Sources (ODBC)**.
- Pada *User DNS*, pilih **Add > MySQL ODBC 3.51 Driver**.



Gambar 3.8 Add Data Source Name

- Pilih tombol Test untuk memastikan koneksi telah berhasil dibuat lalu klik OK.

3.1.4 Pembuatan metode AHP

Kebutuhan proses utama dalam sistem pendukung penilaian kinerja karyawan dengan metode AHP sesuai tabel 2.5 yaitu:

- 1) Penentuan Bobot
Dilakukan perbandingan kriteria untuk mendapatkan nilai bobot masing-masing kriteria berdasarkan dengan metode AHP:

Kemudian dari tabel 2.5 dibuat persentase kepentingan dengan skala 0-2, dengan contoh bila 2 buah kriteria dibandingkan memiliki perbandingan 1,5 dan 0.5 berarti kriteria A Lebih penting dibandingkan kriteria B. Hasil dari perbandingan persentase kriteria dapat dilihat pada tabel 2.6

Dari Tabel 2.6 maka bisa ditentukan bobot dari tiap Kriteria sesuai metode AHP adalah :

Tabel 3.1 Tabel Bobot Kriteria

ID_Kriteria	Nama_kriteria	Bobot
k1	Kemampuan Kerja	0.236
k2	Loyalitas	0.093
k3	Disiplin	0.143
k4	Hubungan Kerja	0.216
k5	Prestasi	0.15
k6	Kepemimpinan	0.16

Setelah didapatkan bobot untuk masing-masing kriteria, maka perhitungan nilai dari subkriteria bisa dilakukan, Berikut Range nilai dari tiap Kriteria :

- Kriteria 1 (Kemampuan Kerja) : sk1-sk4
- Kriteria 2 (Loyalitas) : sk5-sk8
- Kriteria 3 (Disiplin) : sk9-sk12
- Kriteria 4 (Hubungan Kerja) : sk13-sk16
- Kriteria 5 (Prestasi) : sk17-sk20
- Kriteria 6 (Kepemimpinan) : sk21-sk23

Sebagai Contoh, Kriteria 1 (Kemampuan Kerja) dengan range sk1 hingga sk4 = 80,80,80,80 = 320, kemudian diambil rata-ratanya yaitu $320/4 = 80$, setelah itu nilai rata-rata dari Kriteria 1 dikalikan dengan Bobot Metode AHP untuk kriteria tersebut yaitu 0.23. Maka akan didapatkan nilai dari $80 \times 0.23 = 18.4$, begitu juga untuk Kriteria 2 hingga Kriteria 6. Dibawah ini merupakan listing program untuk menghitung nilai karyawan :

Program 1. Rumus perhitungan nilai Rata-rata tiap kriteria

```
for(int a = 1; a <= 23; a++)
{
    System.out.println(k1 + " " + k2+" "+k3+" "+k4+" "+k5+" "+k6);
    if(a > 0 && a <= 4 )
        k1 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 5 && a <= 8)
        k2 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 9 && a <= 12)
        k3 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 13 && a <= 16)
        k4 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 17 && a <= 20)
        k5 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 21 && a <= 23)
        k6 += Float.parseFloat(res.getString("sk"+a));
}
}
k1 /= 4;
k2 /= 4;
k3 /=4;
k4 /=4;
k5 /=4;
k6 /=3;
sql1 = "select * from tabel_kriteria";
res = st1.executeQuery(sql1);
for(int a = 1; a <= 6; a++)
{
    if(res.next())
    {
        switch(a)
        {
            case 1 :
                k1 *= Float.parseFloat(res.getString("Bobot"));
                break;
            case 2 :
                k2 *= Float.parseFloat(res.getString("Bobot"));
                break;
            case 3 :
                k3 *= Float.parseFloat(res.getString("Bobot"));
                break;
            case 4:
                k4 *= Float.parseFloat(res.getString("Bobot"));
                break;
            case 5:
                k5 *= Float.parseFloat(res.getString("Bobot"));
                break;
            case 6:
                k6 *= Float.parseFloat(res.getString("Bobot"));
                break;
```

```

    }
  }
}

```

Pada Program 1, diambil nilai dari tiap kriteria sesuai range kriterianya, misalkan pada kriteria 1 yaitu nilai sk1 hingga sk4 dijumlahkan kemudian dirata-ratakan, rata-rata nilai tersebut lalu dikalikan dengan nilai bobot kriteria 1, begitupun selanjutnya. Kemudian nilai rata-rata tiap kriteria ditampilkan ke dalam textfield pada GUI dengan program berikut :

Program 2. Menampilkan rata-rata kriteria pada textfield

```

kemampuanKerja.setText(String.valueOf(k1));
loyalitas.setText(String.valueOf(k2));
disiplin.setText(String.valueOf(k3));
prestasi.setText(String.valueOf(k4));
hubunganKerja.setText(String.valueOf(k5));
kepemimpinan.setText(String.valueOf(k6));
String ratax = String.valueOf((k1+k2+k3+k4+k5+k6)/6);
rataRata.setText(ratax);

```

3.1.5 Pembuatan aplikasi Penilaian Kinerja Karyawan dengan memakai Pemrograman Java

a) Proses Pengolahan *User*/Pengguna

Proses ini merupakan tahap pengolahan data-data *user* yang menggunakan sistem. Terdapat 2 tingkatan *user* yang berbeda yaitu *admin* dan karyawan. Tiap *User* memiliki kapabilitas berbeda :

1. *Admin*

Mampu mengoptimalkan semua fungsi dalam sistem ini diantaranya melihat data karyawan, mengedit profil, menginputkan nilai, dan melihat hasil kinerja karyawan. Manager Perusahaan dikategorikan *Admin* dalam sistem ini.

2. *Karyawan*

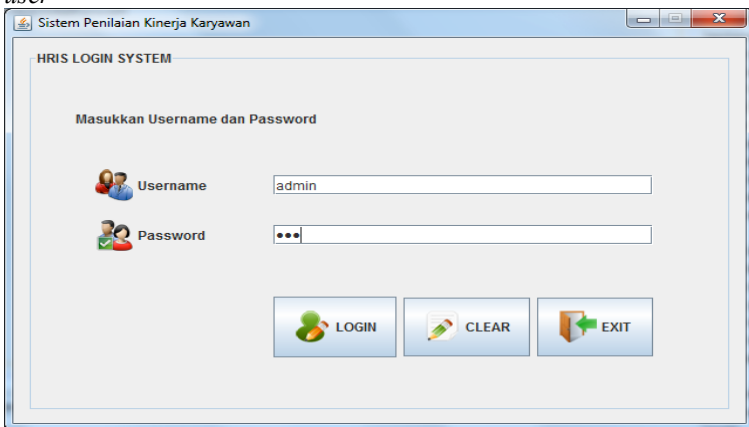
Memiliki kapabilitas terbatas, hanya mampu melihat data karyawan dan melihat kinerja dari karyawan namun tidak mampu melakukan perubahan terhadap data maupun nilai

Program 2. Hak User

```
if(job.equals("admin"))  
    mskanNilai.setEnabled(true);  
else  
    mskanNilai.setEnabled(false);
```

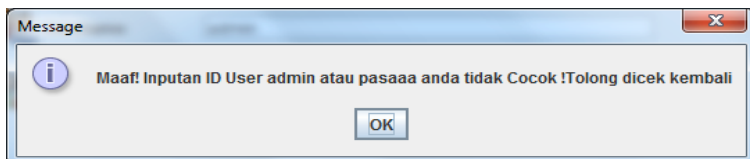
b) Proses Autentikasi User/pengguna (Login)

Proses ini merupakan tahap autentikasi data *user* ketika masuk ke sistem. Proses ini disebut juga proses login. Setelah *user* login maka *user* dapat mengakses menu sesuai dengan hak akses masing-masing *user*



Gambar 3.9 Layar Login

Bila terjadi kesalahan input atau *username* dan password tidak cocok dengan database maka akan muncul peringatan :



Gambar 3.10 Pesan Error Login

Program 3. Deteksi Kesalahan pada Login

```

if(res.next())
{
    if(res.getString("Jabatan").equals("Deputi Manager"))
    {
        HRIS_HOME1 m = new HRIS_HOME1();
        m.job = "admin";
        m.setVisible(true);
        dispose();
        JOptionPane.showMessageDialog(null, "Maaf! Inputan ID User " +
        user.getText() + " atau " + pass.getText() + " anda tidak Cocok/Belum
        Terdaftar !Tolong dicek kembali");
        user.setText("");
        pass.setText("");
    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);
        JOptionPane.showMessageDialog(rootPane, "Anda
        Belum Terhubung dengan Database!");
    }
}

```

c) Proses Pengolahan Nilai dengan metode AHP

Proses ini merupakan tahap pengelolaan nilai kinerja karyawan. Dalam sistem ini terdapat beberapa tahap pengolahan nilai yaitu:

1. Pengolahan kriteria. Pada tahap ini *admin* memasukkan data bobot kriteria ke dalam database karyawan yang nantinya akan digunakan dalam tahap perbandingan elemen. Nilai pembobotan merupakan hasil perbandingan dari kriteria menggunakan metode AHP dan disimpan di database :

Tabel 3.2 Database Bobot

Nama_kriteria	Bobot
Kemampuan Kerja	0.23
Loyalitas	0.093
Disiplin	0.143
Hubungan Kerja	0.216
Prestasi	0.15
Kepemimpinan	0.16

2. Pengolahan sub-kriteria. Merupakan tahap dimana *admin* memasukkan data sub-kriteria penilaian yang digunakan sebagai indikator penilaian kinerja karyawan.
3. Pengolahan nilai kinerja. Merupakan tahap dimana manager memasukkan data penilaian kinerja tiap-tiap karyawan divisinya. Penilaian dilakukan dengan melakukan perkalian antara bobot kriteria dengan nilai option/sub kriteria karyawan yang dinilai. Tahap ini merupakan sintesa/sintesis penilaian dari metode AHP.

d) Proses Pengolahan Nilai Total

Proses ini merupakan tahap pengelolaan nilai yang terakhir. Pada proses ini sistem akan menghitung nilai total penilaian dari nilai tugas dan nilai kinerja karyawan. Kemudian dari hasil nilai total tersebut digunakan sebagai nilai akhir yang akan diurutkan untuk melihat karyawan dengan nilai terbaik

3.1.6 Pembuatan Menu Utama aplikasi Penilaian Kinerja Karyawan dengan Metode AHP memakai Pemrograman Java

Untuk kebutuhan data keluaran menampilkan hasil pengolahan dari data karyawan, dan data penilaian yang sebelumnya telah diolah dengan metode AHP. Output pengolahan sistem secara umum terdiri dari:

- a) Daftar Karyawan
Merupakan keluaran/output sistem dari sistem yang memberikan informasi daftar Karyawan yang ada.
- b) Input Nilai
Merupakan hak bagi *user* yang berstatus *admin* untuk mengedit daftar karyawan, profil, dan nilai dari subkriteria tiap karyawan yang akan digunakan sebagai perhitungan Kinerja karyawan
- c) Nilai kinerja karyawan

Merupakan keluaran/output sistem yang memberikan informasi nilai-nilai kinerja karyawan. Nilai ini berupa score nilai dari hasil perhitungan sintesa penilaian kinerja. Selain itu dapat diketahui score total penilaian kinerja dan peringkat nilai total karyawan.

- d) Lihat Nilai
Menu output untuk melihat nilai rata-rata dari semua kriteria tiap Karyawan.
- e) Lihat Kinerja
Menu untuk melihat hasil Kinerja Karyawan berdasarkan metode AHP, disini akan ditampilkan secara urut hasil kinerja karyawan berdasarkan tingkat nilai Kinerjanya

Antar muka pengguna atau lebih dikenal dengan *user interface* adalah bagian penghubung antara sistem dengan pengguna atau *user* dalam hal ini adalah karyawan. Pada bagian ini akan terjadi komunikasi antara keduanya. Program dimulai dari login karyawan. Terdapat beberapa menu yang berbeda yang diperoleh *user* sesuai dengan level masing-masing karyawan setelah melakukan login *user*. Sistem ini berupa pengolahan data *user*, data tugas dan data penilaian kinerja karyawan dengan metode AHP

Antar muka/*user interface* di titik beratkan pada *interface* yang bersifat *user friendly* yang berarti tidak sulit digunakan atau memudahkan pengguna.

Menu Utama adalah form awal yang ditampilkan saat aplikasi sistem pendukung penilaian kinerja dengan metode AHP dijalankan, Pada rancangan ini terdapat 5 Menu, Menu utama yang akan ditampilkan di Halaman utama dari *User Interface* HRIS ini adalah :

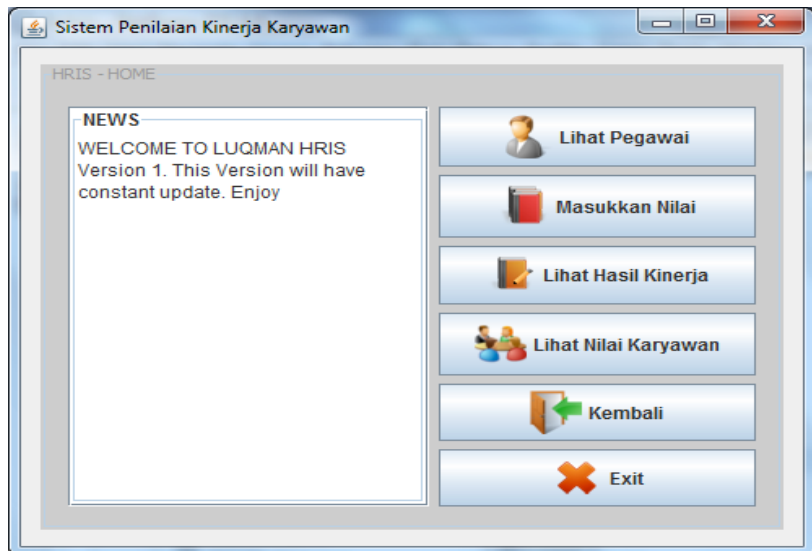
User Interface Aplikasi Penilaian Kinerja Karyawan :

1. Lihat Pegawai : Melihat daftar Karyawan yang ada
2. Masukkan Nilai : Fungsi khusus *admin*/Deputi Manager untuk menginputkan Nilai subkriteria menggunakan metode AHP untuk setiap karyawan
3. Lihat Hasil Kinerja : Melihat Nilai dari Tiap Kriteria penilaian Karyawan berdasar metode AHP sekaligus menginputkan hasil dari Nilai rata-ratanya sehingga bisa diurutkan untuk dilihat nilai yang terbaik

4. Lihat Nilai Karyawan : Melihat Nilai Rata-rata dari seluruh karyawan yang sudah diurutkan
5. Exit : Keluar dari sistem

Menu “Masukkan Nilai” akan terkunci apabila tipe *user* yang login bukan merupakan seorang Deputy Manager/Admin.

Rancangan Menu Utama :

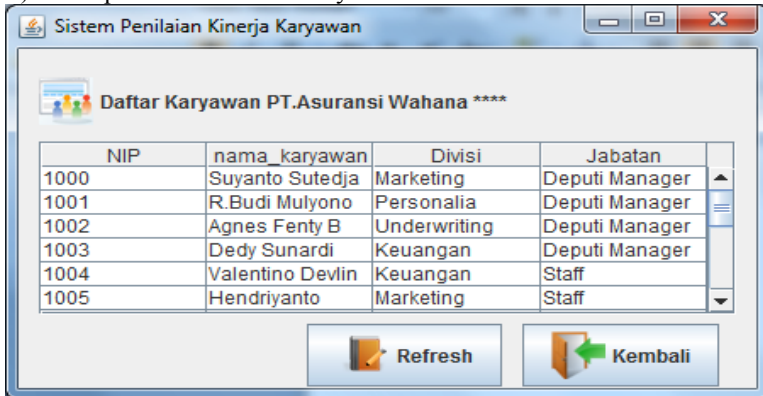


Gambar 3.11 Rancangan Menu Utama

Pada Halaman Menu Utama (gambar 3.1) terdapat 5 Menu utama dari sistem ini, Menu Lihat Pegawai berfungsi untuk melihat daftar karyawan yang ada di perusahaan.

Berikut Penjelasan dari masing-masing menu yang ada di halaman utama:

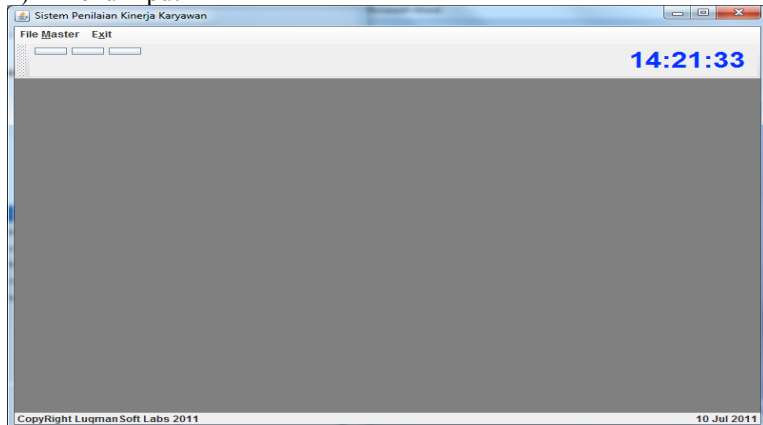
1) Tampilan Menu Lihat Karyawan :



Gambar 3.12 Tampilan Daftar Karyawan

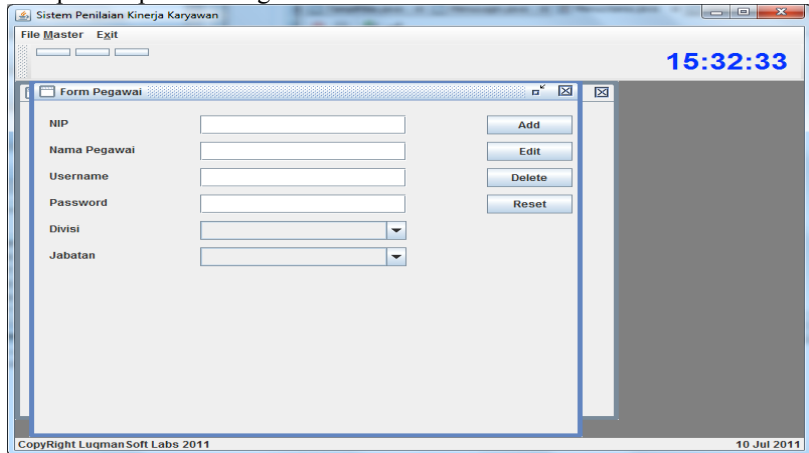
Menu Berikutnya yaitu Menu Masukkan Nilai, merupakan Menu yang hanya bisa diakses oleh *admin*/Deputi Manager, menu ini akan *disabled*/tidak bisa diakses bila *user* bukan termasuk *Admin*/Deputi Manager :

2) Menu Input



Gambar 3.13 Menu Masukkan Nilai

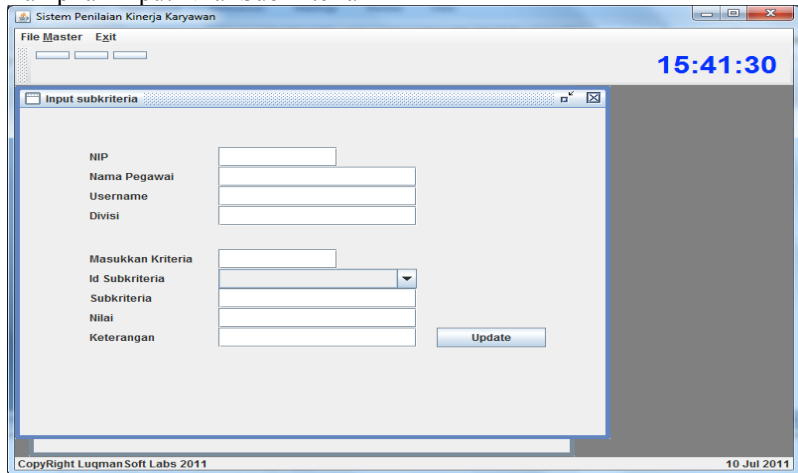
Tampilan Input data Pegawai :



Gambar 3.14 Input Data Pegawai

Pada gambar 3.14 *User*, dalam hal ini *admin/Deputi Manager* bisa memasukkan data karyawan yang terhubung dengan database tabel_karyawan, pada halaman inilah *user* dapat menambah, mengedit, menghapus data di database karyawan.

Tampilan Input Nilai Subkriteria

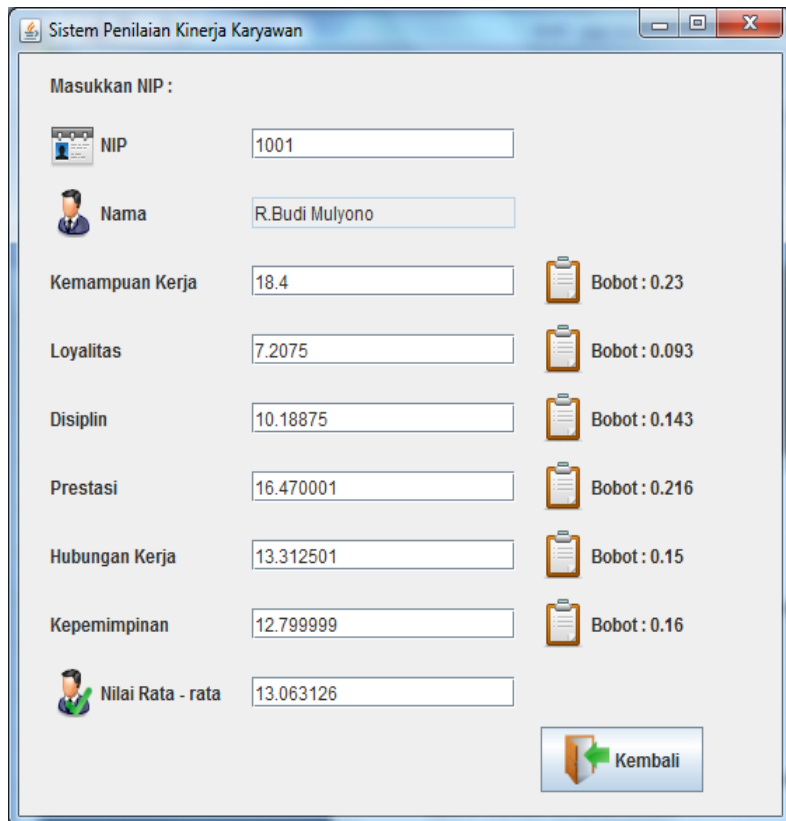


Gambar 3.15 Input nilai Subkriteria

Pada gambar 3.15 *User* dapat mengedit/memasukkan nilai subkriteria tia[karyawan, hanya dengan mengetikkan NIP, maka akan tampil data Nama Karyawan, *Username*, dan Divisi karyawan dengan NIP tersebut, nilai subkriteria yang diubah/dimasukkan akan langsung mengubah nilai subkriteria karyawan dengan NIP yang bersangkutan

3) Tampilan Menu Lihat Hasil Kinerja :

Pada Menu Lihat Hasil Kinerja :



Field	Value	Bobot
NIP	1001	
Nama	R.Budi Mulyono	
Kemampuan Kerja	18.4	0.23
Loyalitas	7.2075	0.093
Disiplin	10.18875	0.143
Prestasi	16.470001	0.216
Hubungan Kerja	13.312501	0.15
Kepemimpinan	12.799999	0.16
Nilai Rata - rata	13.063126	

Gambar 3.16 Lihat Hasil Kinerja



Gambar 3.17 Pesan Nilai Ter-update

Pada Menu Lihat Hasil Kinerja (gambar 3.16) memiliki 2 fungsi yaitu menampilkan data dan mengupdate database, sehingga secara otomatis akan mengupdate database nilai rata-rata.

Pada Menu ini (gambar 3.16) akan ditampilkan nilai rata-rata dari tiap kriteria sesuai perhitungan dengan metode AHP, contoh :

NIP	: 1001
Nama	: R.Budi Mulyono
Kriteria	: Kemampuan Kerja
Bobot	: 0.23 (hasil metode AHP)
Nilai	: 80 (rata-rata dari 4 subkriteria Kemampuan kerja)
Hasil	: $80 \times 0.23 = 18.4$

Dengan cara yang sama maka didapatkan pula hasil untuk kriteria lainnya yaitu :

Loyalitas	: 7,2075
Disiplin	: 10,18875
Prestasi	: 10,470001
Hubungan Kerja	: 13,31250
Kepemimpinan	: 12,79999

Dari 6 nilai rata-rata kriteria diatas, kemudian diambil nilai rata-rata baru untuk menentukan nilai kinerja dari karyawan tersebut, sesuai gambar 3.16 maka didapatkan nilai kinerja untuk karyawan dengan NIP 1001 : R.Budi Mulyono adalah sebesar 13,063.

Nilai 13,063 itu akan diurutkan bersama dengan nilai kinerja karyawan lain untuk dijadikan bahan pertimbangan kinerja karyawan oleh pimpinan perusahaan.

4) Tampilan Menu Lihat Nilai Karyawan :



Gambar 3.18 Lihat Nilai Karyawan

Menu pada gambar 3.18 akan menampilkan daftar karyawan lengkap dengan nilai rata-rata keseluruhan kriteria dengan metode AHP.

3.2 ANALISA

Setelah perencanaan dan pembuatan sistem maka langkah selanjutnya adalah melakukan pengujian terhadap sistem yang telah dibuat. Pengujian dilakukan untuk mengetahui apakah sistem yang direncanakan dan dibangun dapat berjalan sesuai dengan yang diinginkan.

3.2.1 TUJUAN ANALISA

Pengujian merupakan salah satu langkah yang harus dilakukan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil-hasil yang dicapai selama pengujian sistem. Pengujian juga bertujuan untuk mengetahui kelebihan dan kekurangan dari sistem yang telah dibuat. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadi kekurangan dalam sistem

3.2.2 ANALISA KEBERHASILAN SISTEM

1.Keberhasilan Fitur-fitur Pada Aplikasi Penilaian Kinerja Karyawan

Keberhasilan fitur ditinjau dari keberhasilan aplikasi untuk menjalankan fitur-fitur yang telah dibangun.

1. Keberhasilan Sistem

➤ Login

Tabel 3.3 Pengujian Login

Pengujian	Username	Password	Koneksi	
			Sukses	Gagal
1	bud12	123	√	
2	BUD12	123	√	
3	agn13	Agag	√	
4	Agn	AGAG	√	
5	val00	Valen	√	
6	ven13	veny99		√

Tabel 3.2 merupakan hasil pengujian proses login, pada pengujian ke-1 hingga pengujian ke-5 proses login berjalan sukses karena *username* dan *password* cocok dengan database, namun pengujian ke-6 gagal dan *user* tidak bisa masuk ke dalam sistem karena *username/password* salah/belum terdaftar. Pada sistem login ini dibuat tidak *case sensitive* , jadi besar kecil huruf pada saat login tidak berpengaruh.

➤ Hak *admin* dan *User* biasa

Pengujian dilakukan dengan melakukan login ke dalam server MySQL yang telah di buat. Ketika *User* yang login adalah *Admin/Deputi Manager*, maka *user* akan dapat mengakses menu “Masukkan Nilai”, bila *user* hanya merupakan *user* biasa/karyawan maka menu “Masukkan Nilai” tidak akan bisa diakses :

Tabel 3.4 Hak *Admin* Dan *User* Biasa

Pengujian	<i>User</i>	Password	Hak Merubah Data		Tampilan
			Bisa	Tidak	
1	bud12	123	√		 Masukkan Nilai
2	<i>admin</i>	pas	√		 Masukkan Nilai
3	agn13	agag	√		 Masukkan Nilai
4	ded13	dedide	√		 Masukkan Nilai
5	val00	valen		√	 Masukkan Nilai

2. Penghitungan Waktu Eksekusi Aplikasi Penilaian Kinerja Karyawan

Untuk mendapatkan lama waktu eksekusi program yang dijalankan dibutuhkan program sebagai berikut :

Kode program Lama Waktu Eksekusi

```

long mulai = System.currentTimeMillis();
.....
Program yang di hitung lama eksekusinya
.....
long akhir = System.currentTimeMillis();
long t_save_db = akhir - mulai;
System.out.println("waktu menyimpan satu data entry ke field database = " + t_save_db +
"milisekon");

```

a) Lama Waktu Login

Untuk mendapatkan waktu eksekusi lama untuk login *user* maka perlu dilakukan pengujian pengukuran waktu eksekusi. Proses pengukuran waktu eksekusi ini dimulai dari running program hingga muncul login sukses dilakukan. Berikut ini adalah hasil pengujian sebanyak 10 kali percobaan yang telah dilakukan :

Tabel 3.5 Pengukuran lama waktu eksekusi login

Login User ke -	Waktu Eksekusi (ms)
1	58
2	61
3	55
4	57
5	67
6	64
7	62
8	53
9	64
10	58
Rata - Rata	59.9

b) Lama Waktu Proses Algoritma

Untuk mendapatkan lama waktu eksekusi algoritma AHP maka perlu dilakukan pengujian pengukuran waktu eksekusi. Proses pengukuran waktu eksekusi ini dimulai dari running program hingga muncul hasil perhitungan metode AHP. Berikut ini adalah hasil pengujian sebanyak 10 kali percobaan yang telah dilakukan :

Tabel 3.6 Pengukuran lama waktu eksekusi metode AHP

Eksekusi ke-	Waktu Eksekusi (ms)
1	1490
2	798
3	1232
4	1159
5	665
6	1374
7	1152
8	1241
9	1117
10	690
Rata - Rata	1091.8

3. Uji Keberhasilan algoritma AHP

Pengujian dilakukan pada menu Lihat Kinerja , dengan memasukkan NIP, akan muncul nama karyawan dan nilai dari tiap Kriteria karyawan tersebut sesuai dengan metode AHP. Berikut ini daftar nilai 5 Karyawan :

NIP	sk1	sk2	sk3	sk4	sk5	sk6	sk7	sk8	sk9	sk10	sk11	sk12	sk13	sk14	sk15	sk16	sk17	sk18	sk19	sk20	sk21	sk22	sk23	nilaiRataRata
1000	80	80	70	70	85	90	80	80	80	80	80	80	80	80	80	80	80	80	80	80	70	75	70	12.9459
1001	80	80	80	80	80	80	80	70	75	70	70	70	70	70	75	90	90	85	90	90	70	90	80	13.1431
1002	90	80	90	80	90	70	70	60	90	80	70	90	90	90	70	70	80	80	80	90	80	90	80	13.5981
1003	80	80	90	80	70	90	70	90	90	80	80	90	85	85	98	75	95	85	90	80	70	90	90	14.0076
1004	90	80	90	80	90	70	70	60	90	80	70	90	90	90	70	70	80	80	80	90	80	90	80	13.5131

Gambar 3.19 Nilai Subkriteria Karyawan NIP 1000-1004

Pada Gambar 3.19 merupakan daftar nilai tiap subkriteria karyawan dengan NIP 1000-1004, pada sistem penilaian ini digunakan parameter Kriteria 1 hingga Kriteria 6.

Berikut Range nilai dari tiap Kriteria :

- Kriteria 1 (Kemampuan Kerja) : sk1-sk4
- Kriteria 2 (Loyalitas) : sk5-sk8
- Kriteria 3 (Disiplin) : sk9-sk12
- Kriteria 4 (Hubungan Kerja) : sk13-sk16
- Kriteria 5 (Prestasi) : sk17-sk20
- Kriteria 6 (Kepemimpinan) : sk21-sk23

Tabel 3.7 Pengujian Algoritma AHP

NIP	Nilai Kriteria						Rata-rata	
	K. Kerja	Loyalitas	Disiplin	Prestasi	H. Kerja	Kepemimpinan	Manual	program
	0.236	0.093	0.143	0.216	0.15	0.16		
1000	17.7	7.788	11.44	17.28	12	11.467	12.945	12.945
1001	18.88	72.075	10.188	16.470	13.312	12.799	13.143	13.143
1002	20.06	6.742	11.797	17.28	12.375	13.333	13.598	13.598
1003	19.47	7.44	12.155	18.522	13.125	13.333	14.007	14.007
1004	20.06	6.742	11.797	17.28	12.375	13.333	13.598	13.598

Pada tabel 3.7 ditunjukkan hasil perhitungan data tabel subkriteria pada gambar 3.19 secara manual dan secara program, nilai rata-rata yang dihasilkan sama dengan perhitungan manual.

Perhitungan Manual :

Sebagai Contoh, karyawan dengan NIP 1000 :

- Kriteria 1 (Kemampuan Kerja) dengan range sk1 hingga sk4 = 80,80,70,70 = 300, kemudian diambil rata-ratanya yaitu $300/4 = 75$, setelah itu nilai rata-rata dari Kriteria 1 dikalikan dengan Bobot Metode AHP untuk kriteria tersebut yaitu 0.23. Maka akan didapatkan nilai dari $75 \times 0.236 = 17,7$
- Kriteria 2 (Loyalitas) dengan range sk5 hingga sk8 = 85,90,80,80 = 335, kemudian diambil rata-ratanya yaitu $335/4 = 83.75$, setelah itu nilai rata-rata dari Kriteria 1 dikalikan dengan Bobot Metode AHP untuk kriteria tersebut yaitu 0.23. Maka akan didapatkan nilai dari $83.75 \times 0.093 = 0.778875$
- Kriteria 3 (Disiplin) dengan range sk9 hingga sk12 = 80,80,80,80 = 320, kemudian diambil rata-ratanya yaitu $320/4 = 80$, setelah itu nilai rata-rata dari Kriteria 1 dikalikan dengan Bobot Metode AHP untuk kriteria tersebut yaitu 0.23. Maka akan didapatkan nilai dari $80 \times 0.143 = 11.44$
- Kriteria 4 (Prestasi) dengan range sk13 hingga sk16 = 80,80,80,80 = 320, kemudian diambil rata-ratanya yaitu $320/4 = 80$, setelah itu nilai rata-rata dari Kriteria 1 dikalikan dengan Bobot Metode AHP untuk kriteria tersebut yaitu 0.23. Maka akan didapatkan nilai dari $80 \times 0.216 = 17.28$
- Kriteria 5 (Hubungan Kerja) dengan range sk17 hingga sk20 = 80,80,80,80 = 320, kemudian diambil rata-ratanya yaitu $320/4 = 80$, setelah itu nilai rata-rata dari Kriteria 1 dikalikan dengan Bobot Metode AHP untuk kriteria tersebut yaitu 0.23. Maka akan didapatkan nilai dari $80 \times 0.15 = 12$
- Kriteria 6 (Kepemimpinan) dengan range sk21 hingga sk23 = 70,75,79 = 215, kemudian diambil rata-ratanya yaitu $215/3 = 75$, setelah itu nilai rata-rata dari Kriteria 1 dikalikan dengan Bobot Metode AHP untuk kriteria tersebut yaitu 0.23. Maka akan didapatkan nilai dari $71.666667 \times 0.236 = 11.466666$

Setelah Kriteria 1 hingga kriteria 6 dilakukan perhitungan maka berikutnya adalah melakukan pengambilan nilai rata-rata dari keenam kriteria yang ada. Nilai inilah yang akan dijadikan dasar pertimbangan kinerja tiap karyawan karena mewakili dari 6 kriteria penilaian yang ada di perusahaan

Masukkan NIP :

	NIP	1000	
	Nama	Suyanto Sutedja	
Kemampuan Kerja	17.7		Bobot : 0.23
Loyalitas	7.78875		Bobot : 0.093
Disiplin	11.440001		Bobot : 0.143
Prestasi	17.28		Bobot : 0.216
Hubungan Kerja	12.0		Bobot : 0.15
Kepemimpinan	11.466666		Bobot : 0.16

Gambar 3.20 Hasil Perhitungan Metode AHP per Kriteria

	Nilai Rata - rata	12.945
---	-------------------	-------------------------------------

Gambar 3.21 Nilai Rata-rata dari 6 Kriteria NIP 1000

Dari perhitungan diatas dapat dicari nilai eror sebagai berikut :

$$\text{Error} = \frac{P.\text{manual} - P.\text{program}}{2} \times 100\%$$

$$= 0\%$$

4. Pengurutan Ranking Karyawan



Gambar 3.22 Uji Menu Pengurutan Karyawan



Gambar 3.23 Uji Menu Pengurutan Kriteria Kemampuan Kerja



Gambar 3.24 Uji Pengurutan Kriteria Loyalitas



Gambar 3.25 Uji Pengurutan Kriteria Disiplin



Gambar 3.26 UjiPengurutan Kriteria Hubungan Kerja



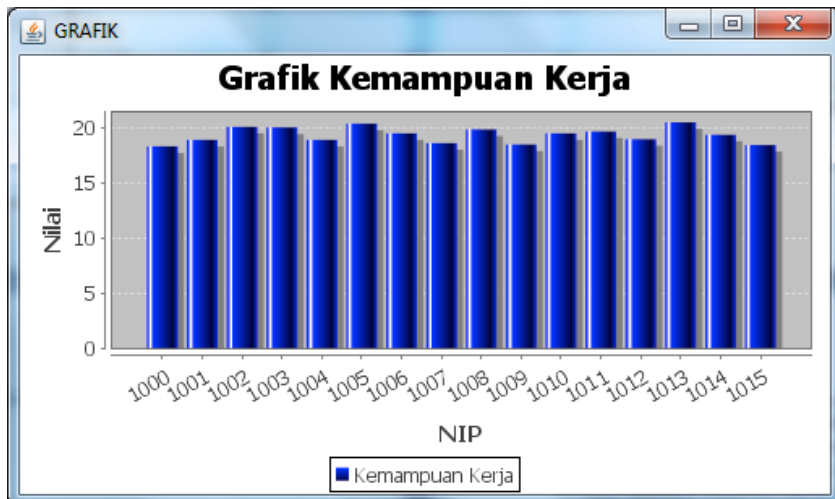
Gambar 3.27 Uji Pengurutan Kriteria Prestasi



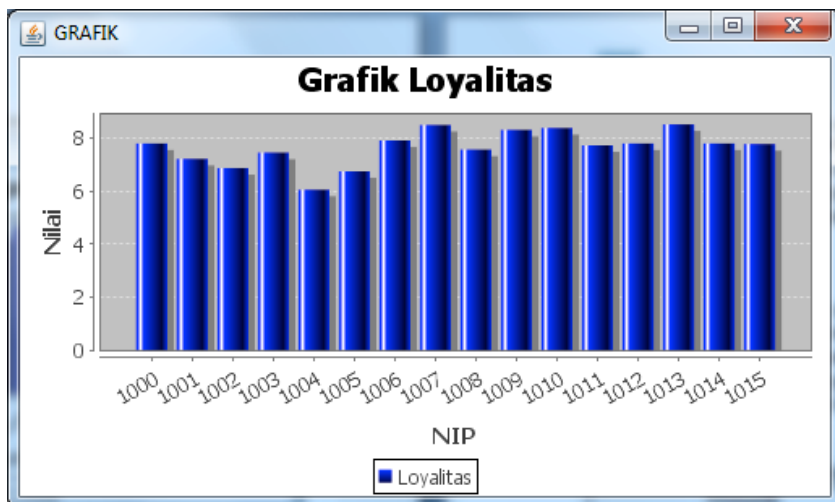
Gambar 3.28 UjiPengurutan Kriteria Kepemimpinan



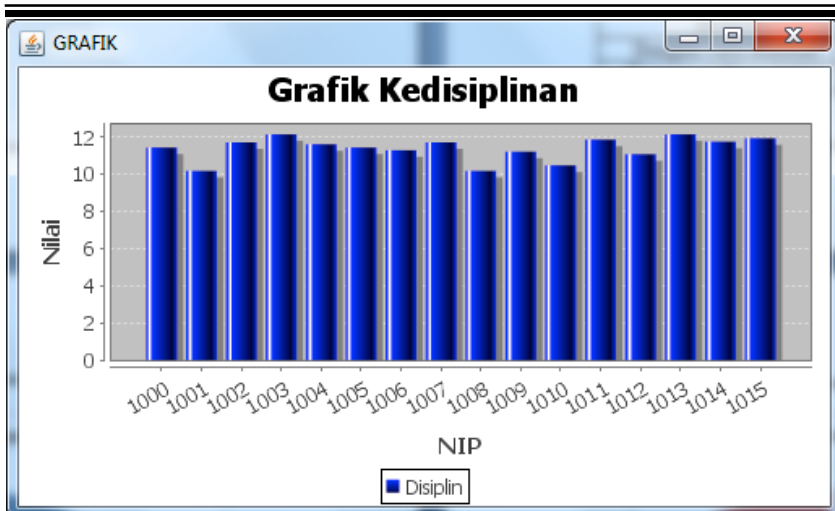
Gambar 3.29 Uji Pengurutan Ranking Karyawan



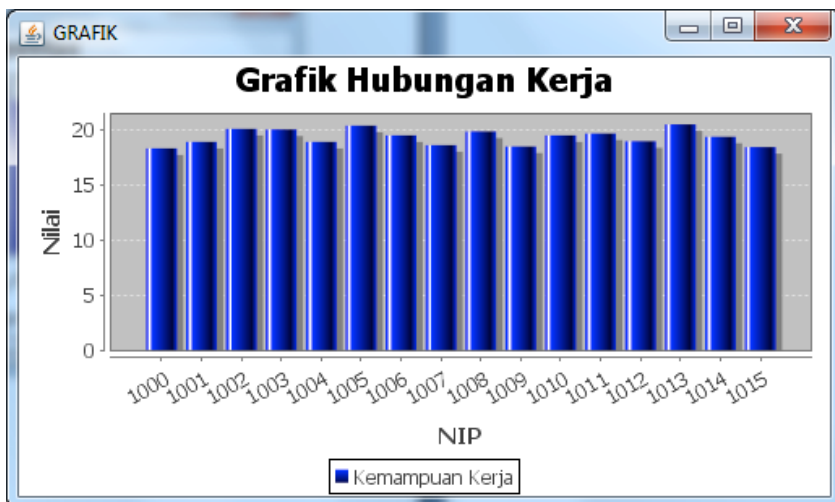
Gambar 3.30 Grafik Pengurutan Ranking Kemampuan Kerja



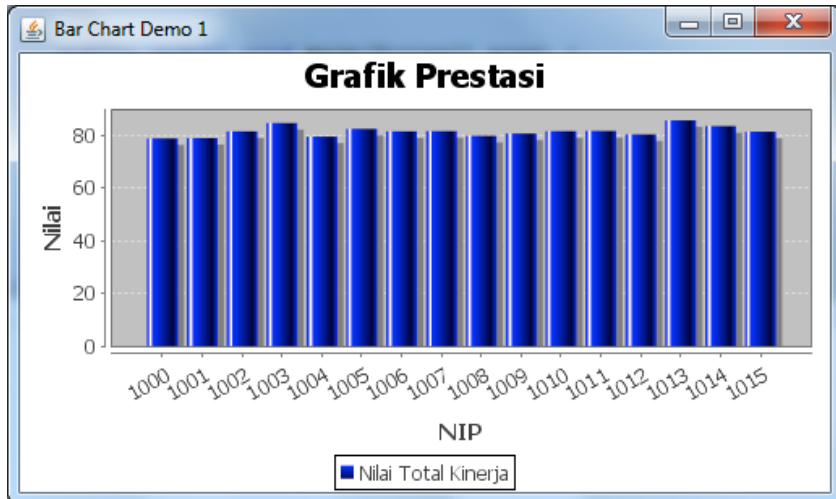
Gambar 3.31 Grafik Pengurutan Ranking Loyalitas



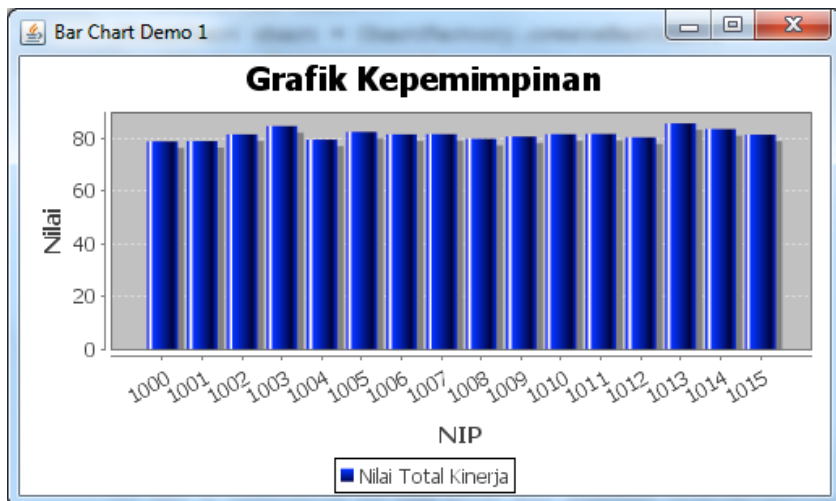
Gambar 3.32 Grafik Pengurutan Ranking Kedisiplinan



Gambar 3.33 Grafik Pengurutan Ranking Hubungan Kerja



Gambar 3.34 Uji Grafik Pengurutan Ranking Prestasi



Gambar 3.35 Grafik Pengurutan Ranking Kepemimpinan

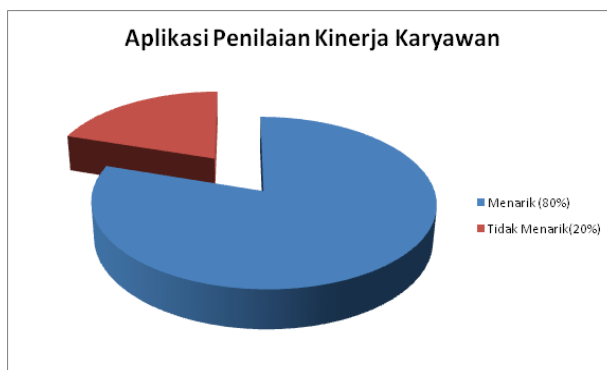
Pada Pengujian Pengurutan Ranking karyawan Terlihat bahwa data nilai rata-rata akhir dari tiap karyawan berhasil diurutkan dari karyawan yang memiliki nilai terkecil hingga nilai terbesar sesuai kriteria masing-masing, sehingga dapat dijadikan bahan pertimbangan bagi Pimpinan untuk mengambil keputusan

5. Survei Responden

Pengujian yang dilakukan yaitu dengan melakukan survey pada 10 responden yaitu Deputy manajer dan kepala bagian/Asisten manajer yang berhak melakukan penilaian untuk menanggapi sistem yang telah dibuat yaitu aplikasi penilaian kinerja karyawan, untuk dibandingkan dengan sistem yang telah ada. Berikut ini hasil *survey* yang telah diambil dari 10 responden:

Tabel 3.8 Tampilan Aplikasi Penilaian Kinerja Karyawan yang Dibuat :

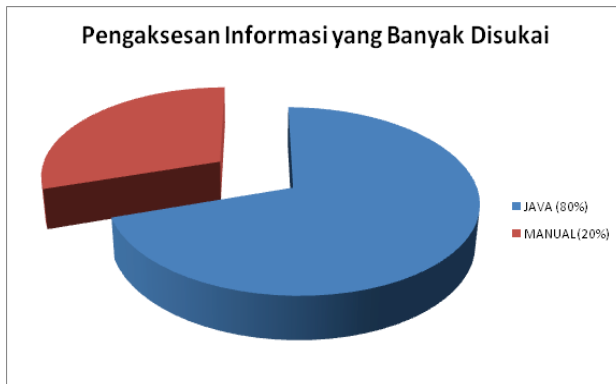
Menarik	Persentase (%)
Ya	80%
Tidak	20%



Gambar 3.36 Grafik survei Tampilan Aplikasi Penilaian Kinerja Karyawan

Tabel 3.9 Apakah aplikasi ini membantu Pengaksesan Informasi database:

Pengaksesan yang disukai	Ya	Tidak
Dengan cara manual	30%	
Dengan aplikasi JAVA		70%



Gambar 3.37 Grafik Pengaksesan Informasi Yang Banyak Disukai

Tabel 3.10 Manfaat Aplikasi Penilaian Kinerja Karyawan Yang Dibuat

Bermanfaat	Persentase
Ya	80%
Tidak	20%



Gambar 3.38 Grafik Manfaat Aplikasi Penilaian Kinerja Karyawan

Tabel 3.11 Menu Yang Terdapat Pada Aplikasi Penilaian Kinerja Karyawan Cukup Dalam Mengakses Data

Kelengkapan menu dalam mengakses data	Persentase
Ya	80%
Tidak	20%



Gambar 3.39 Grafik Kelengkapan Aplikasi Penilaian Kinerja Karyawan

Dari hasil survei yang telah dilakukan dapat disimpulkan bahwa :

- a. Untuk aplikasi Penilaian Kinerja Karyawan di Perusahaan Asuransi yang telah dibuat, 80% responden menyatakan sistem aplikasi ini menarik.
- b. Untuk pengaksesan sistem informasi yang paling banyak disukai adalah sistem inventory dengan JAVA. Hal ini dapat dilihat pada hasil survey yaitu 70% responden lebih menyukai cara pengaksesan dengan JAVA.
- c. Berdasarkan hasil survey , aplikasi Kinerja Karyawan di Perusahaan Asuransi pada perusahaan Asuransi berbasis JAVA memiliki penilaian dengan kategori bermanfaat. Hal ini dapat dilihat hasil persentase sebanyak 80% orang responden menyatakan aplikasi logistik tool ini bermanfaat.
- d. Kelengkapan menu yang terdapat pada aplikasi logistik tool dalam mengakses data dianggap cukup lengkap, terlihat dari hasil survei menyatakan 80% responden menyatakan menu yang terdapat dalam sistem inventory ini cukup dalam mengakses data.

Quisioner:

Nama :

Divisi :

No	PERTANYAAN	KETERANGAN	
		YA	TIDAK
1	Menurut anda, apakah tampilan pada aplikasi HRIS yang telah dibuat menarik ?		
2	Menurut anda,apakah aplikasi HRIS ini dapat membantu anda (<i>admin</i> dan <i>Deputi Manager</i>) mengakses database karyawan?		
3	Apakah aplikasi ini membantu anda dalam pengaksesan dibandingkan cara manual?		
4	Menurut anda, apakah menu - menu dalam aplikasi HRIS tersebut dapat membantu anda dalam menilai kinerja karyawan?		

Saran/Kritik untuk perbaikan aplikasi ini :

.....

.....

.....

.....

.....

.....

[Halaman ini sengaja dikosongkan]

BAB IV

PENUTUP

Dari Hasil perancangan dan pengujian Sistem Penilaian Kinerja Karyawan berbasis metode AHP, mulai dari perancangan sistem, sampai tahap penyelesaian program, maka didapatkan kesimpulan yang berguna bagi pengembangan selanjutnya :

Kesimpulan

1. Dari hasil pengujian terlihat bahwa sistem telah berjalan dengan baik, dan fitur-fitur menu pada aplikasi HRIS ini dapat berjalan dengan semestinya..
2. Dari hasil pengukuran waktu eksekusi :
 - a. Rata-rata waktu login yang dibutuhkan untuk masuk ke dalam sistem adalah 59,9 ms
 - b. Rata- rata waktu proses algoritma AHP hinggaai dihasilkan nilai rata tiap kriteria dan nilai rata-rata kinerja tiap karyawan adalah 1091,8 ms
3. Dari hasil survei responden terlihat bahwa 80% responden menyatakan bahwa aplikasi ini menarik,dimana 70% dari responden menyatakan bahwa mereka lebih menyukai menggunakan system ini disbanding cara manual, dan dari hasil survey terlihat bahwa 80% responden menganggap aplikasi ini bermanfaat dan memiliki menu-menu akses yang cukup

1.1 Saran

Berdasarkan pada pengujian yang telah dilakukan pada perangkat lunak yang dibuat, masih banyak kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar kinerjanya lebih baik, oleh karena itu disarankan:

- a) Dapat merangkum history nilai/ sejarah penilaian kinerja karyawan sehingga dapat melihat nilai-nilai sebelumnya.
-

- b) Adanya sistem *backup* data setiap periode ataupun waktu yang diinginkan secara otomatis, sehingga meminimalisasi kemungkinan hilangnya data secara menyeluruh
 - c) Perangkingan di sisi manager dapat menampilkan urutan dengan nomor peringkat baik pada display penilaian maupun output laporan penilaian kinerja karyawan.
-
-

DAFTAR PUSTAKA

- [1] Bambang Eka Saputra, Sistem Pendukung Penilaian Kinerja Karyawan, Proyek akhir Jurusan Teknik Informatika FTI UII, 2009.
 - [2] Bourgeois R. *Analytical Hierarchy Process: An Overview*. Bogor: UNCAPSA-UNESCAP. 2005
 - [3] Johannes Sinaga, Penerapan *Analitical Hierarchy Processs (AHP)* Dalam Pemilihan Perusahaan Badan Usaha Milik Negara (BUMN) Sebagai Tempat Kerja Mahasiswa Universitas Sumatera Utara , Proyek Akhir Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Sumatera Utara, Medan, 2008.
 - [4] Saaty T.L. 1993. Pengambilan Keputusan bagi Para Pemimpin, Proses Hirarki Analitik untuk Pengambilan Keputusan dalam Situasi yang Kompleks, Pustaka Binama Pressindo, 2008
 - [5] Saaty T.L. Uncertainty and rank order in the analytic hierarchy process. *European Journal of Operational Research*. 1987.
-

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Program 1. Sistem Login

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * MenuLogin.java
 *
 * Created on 12 Mei 11, 19:29:34
 */
package sishris;

import com.mysql.jdbc.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.sql.*;
import java.io.*;

/**
 *
 * @author Absolute1
 */
public class MenuLogin extends javax.swing.JFrame {

    /** Creates new form login */
    private static Connection Con;
    ResultSet RsUser;
    public String username;
    private KoneksiMySQL kmysql;

    public MenuLogin() {
        super("Sistem Penilaian Kinerja Karyawan");
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
```

```

buttonGroup1 = new javax.swing.ButtonGroup();
jPanel1 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
user = new javax.swing.JTextField();
pass = new javax.swing.JPasswordField();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jButton3 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
setResizable(false);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("HRIS      LOGIN
SYSTEM"));

jLabel1.setText("Masukkan Username dan Password");

jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Gambar/pembeli.png"))); // NOI18N
jLabel2.setText("Username");

jLabel3.setIcon(new      javax.swing.ImageIcon(getClass().getResource("/Gambar/daftar
pelanggan.png"))); // NOI18N
jLabel3.setText("Password");

jButton1.setIcon(new      javax.swing.ImageIcon(getClass().getResource("/Gambar/daftar
anggota.png"))); // NOI18N
jButton1.setText("LOGIN");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setIcon(new      javax.swing.ImageIcon(getClass().getResource("/Gambar/report
film.png"))); // NOI18N
jButton2.setText("CLEAR");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jButton3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Gambar/exit.png"))); // NOI18N
jButton3.setText("EXIT");

```

```
jButton3.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton3ActionPerformed(evt);  
    }  
});  
  
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);  
jPanel1.setLayout(jPanel1Layout);  
jPanel1Layout.setHorizontalGroup(  
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(jPanel1Layout.createSequentialGroup()  
            .addGap(36, 36, 36)  
  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,  
false)  
            .addComponent(jLabel1)  
            .addGroup(jPanel1Layout.createSequentialGroup()  
                .addGap(17, 17, 17)  
  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addComponent(jLabel3)  
            .addComponent(jLabel2))  
            .addGap(57, 57, 57)  
  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,  
false)  
            .addComponent(user)  
            .addGroup(jPanel1Layout.createSequentialGroup()  
                .addGap(17, 17, 17)  
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
                    false)  
                    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE, true)  
                    .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true))  
            .addContainerGap(108, false))  
            .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true))  
            .addGap(10, 10, 10))  
        .addGroup(jPanel1Layout.createSequentialGroup()  
            .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
            .addContainerGap(108, false))  
);  
jPanel1Layout.setVerticalGroup(  
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(jPanel1Layout.createSequentialGroup()  
            .addGap(10, 10, 10)  
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
                .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
                .addComponent(user, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
                .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
                .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
                .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, true)  
            .addContainerGap(108, false))  
);
```

```

        .addComponent(user,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel2))
        .addGap(18, 18, 18)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
    )
        .addComponent(pass,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3))
        .addGap(46, 46, 46)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
    )
        .addComponent(jButton2,            javax.swing.GroupLayout.PREFERRED_SIZE,    59,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jButton3,            javax.swing.GroupLayout.PREFERRED_SIZE,    59,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jButton1,            javax.swing.GroupLayout.PREFERRED_SIZE,    61,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(49, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            );

    pack();
} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Connect ObjKoneksi1 = new Connect();

```

```
        Connection con1;

    try {
        String _pass;
        con1 = (Connection) ObjKoneksi1.OpenConnect();
        Statement st1 = con1.createStatement();
        _pass = new String (pass.getText());
        System.out.print(_pass);
        //JOptionPane.showMessageDialog(rootPane, _pass);
        String sql1 = "select * from tabel_karyawan where user = '"+user.getText()+"' and
password = '"+_pass+"'";
        ResultSet res = st1.executeQuery(sql1);

        if(res.next())
        {
            if(res.getString("Jabatan").equals("Deputi Manager"))
            {
                HRIS_HOME1 m = new HRIS_HOME1();
                m.job = "admin";
                m.setVisible(true);
                dispose();
            }
            else
            {
                HRIS_HOME1 m = new HRIS_HOME1();
                m.job = "karyawan";
                m.setVisible(true);
                dispose();
            }
        }
        else

        JOptionPane.showMessageDialog(null, "Maaf! Inputan ID User " + user.getText() + "
atau "+pass.getText()+" anda tidak Cocok !Tolong dicek kembali");
        user.setText("");
        pass.setText("");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
        //Logger.getLogger(TampilNilai.class.getName()).log(Level.SEVERE, null, ex);
        JOptionPane.showMessageDialog(rootPane, "User/pass salah2");
    }

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    user.setText("");
    pass.setText(""); // TODO add your handling code here:
}
```

```

    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);    // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
                new MenuLogin().setVisible(true);

            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPasswordField pass;
    private javax.swing.JTextField user;
    // End of variables declaration
}

```

Program 2. Menu Utama

```

* HRIS_HOME.java
*
* Created on 11 Mei 11, 13:20:35
*/

package sishris;

/**
 *
 * @author Absolute1
 */
public class HRIS_HOME1 extends javax.swing.JFrame {

    /** Creates new form HRIS_HOME */
    public static String job;
    public HRIS_HOME1() {

```

```
super("Sistem Penilaian Kinerja Karyawan");
initComponents();
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    TampilNilai tampil = new TampilNilai();
    tampil.setVisible(true);
    dispose();// TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    LihatKaryawan lihat = new LihatKaryawan();
    lihat.setVisible(true);
    dispose();// TODO add your handling code here:
}

private void mskanNilaiActionPerformed(java.awt.event.ActionEvent evt) {
    MenuUtama menu = new MenuUtama();
    menu.setVisible(true);// TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0); // TODO add your handling code here:
}

private void onLoad(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    if(job.equals("admin"))
        mskanNilai.setEnabled(true);
    else
        mskanNilai.setEnabled(false);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    NilaiUrut tampil = new NilaiUrut();
    tampil.setVisible(true);
    dispose();// TODO add your handling code here:
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    MenuLogin m = new MenuLogin();
    m.setVisible(true);
    dispose();// TODO add your handling code here:
}
```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new HRIS_HOME1().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JDialog jDialog1;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextPane jTextPane1;
private javax.swing.JButton mskanNilai;
// End of variables declaration
}

```

Program 3. Lihat Karyawan

```

/*
 * LihatKaryawan.java
 *
 * Created on 22 Jun 11, 11:41:37
 */
package sishris;
import javax.swing.table.DefaultTableModel;
*
 * @author Absolute1
 */
public class LihatKaryawan extends javax.swing.JFrame {
    private KoneksiMySQL kmsql;
    private ModelTable ModelTable;

    /** Creates new form LihatKaryawan */
    public LihatKaryawan() {
        super("Sistem Penilaian Kinerja Karyawan");
        initComponents();
        kmsql = new KoneksiMySQL();

        ModelTable = new ModelTable ("Select NIP,nama_karyawan,Divisi,Jabatan from
tabel_karyawan");

```

```
jTable1.setModel(new DefaultTableModel(ModelTable.getObj(),
ModelTable.getColNames()));
}
/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    ModelTable = new ModelTable ("Select NIP,nama_karyawan,Divisi,Jabatan from
tabel_karyawan");
    jTable1.setModel(new DefaultTableModel(ModelTable.getObj(),
ModelTable.getColNames())); // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    HRIS_HOME1 m = new HRIS_HOME1();
    m.setVisible(true);
    dispose(); // TODO add your handling code here:
}
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new LihatKaryawan().setVisible(true);
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
}
```

Program 4. Masukkan Nilai

```
*/

package sishris;

import java.awt.*;
import java.awt.event.*;
```

```

import javax.swing.*;
import java.util.*;
import java.text.*;
import javax.swing.Timer;

public class MenuUtama extends JFrame implements ActionListener {
    private JDesktopPane desktop1 = new JDesktopPane();
    /* Menu Pulldown */
    private JMenuBar barMenu1 = new JMenuBar();
    private JMenu mnuFileMaster = new JMenu("File Master"),
    //          mnuInput = new JMenu("Input Nilai"),
    //          mnuKinerjaKaryawan = new JMenu("Report"),
    //          mnuExit = new JMenu("Exit");
    private JMenuItem mnuFMPegawai = new JMenuItem(new ImageIcon ("Gambar/12.jpg")),
    //          mnuInputSub = new JMenuItem("Input Nilai Subkriteria", new ImageIcon
    ("Gambar/12.png")),
    //          mnuReturn1 = new JMenuItem("Return", new ImageIcon ("gambar/")),
    //          mnuInputNilai = new JMenuItem("Input Nilai Karyawan", new ImageIcon
    ("gambar/penempatan.png")),
    //          mnuFTGaji = new JMenuItem("Lihat Nilai Karyawan", new ImageIcon
    ("gambar/gaji.png")),
    //          mnuReturn2 = new JMenuItem("Return", new ImageIcon ("gambar/")),
    //          mnuReportPegawai = new JMenuItem("Report Data Pegawai", new
    ImageIcon ("gambar/rpegawai.png")),
    //          mnuReportGol = new JMenuItem("Report Data Gol", new ImageIcon
    ("gambar/rgol.png")),
    //          mnuReportGaji = new JMenuItem("Report Penggajian", new ImageIcon
    ("gambar/rgaji.png")),
    //          mnuReturn3 = new JMenuItem("Return", new ImageIcon ("gambar/")),
    //          mnuExit1 = new JMenuItem("Exit", new ImageIcon ("gambar/exit.png"));
    /* Menu PopUp */
    private JPopupMenu popMenu1 = new JPopupMenu();
    private JMenuItem mnuPopUp_FMPegawai = new JMenuItem("Data Pegawai", new
    ImageIcon("Gambar/12.png", null)),
    //          mnuPopUp_InputSub = new JMenuItem("Input Nilai Subkriteria", new
    ImageIcon ("gambar/gol.png")),
    //          mnuPopUp_FMReturn = new JMenuItem("Return", new ImageIcon
    ("Gambar/"));
    /* Menu ToolBar */
    private JToolBar toolBar1 = new JToolBar();
    private JPanel ToolbarPanel1 = new JPanel(new FlowLayout(FlowLayout.LEFT)),
    //          ToolbarPanel2 = new JPanel(new FlowLayout(FlowLayout.RIGHT));
    private JButton mnuBtnFMPegawai = new JButton(new
    ImageIcon("gambar/pegawai.png")),
    //          mnuBtnInputSub = new JButton(new ImageIcon("gambar/gol.png")),
    //          mnuBtnFTPenempatan = new JButton(new ImageIcon("gambar/penempatan.png")),
    //          mnuBtnFTGaji = new JButton(new ImageIcon("gambar/gaji.png")),
    //          mnuBtnExit = new JButton(new ImageIcon ("Gambar/Penguins.jpg"));
    private JLabel lblwaktu1 = new JLabel();
    /* Pengaturan Waktu dan Tanggal */

```

```
private Date tglsekarang = new Date();
private SimpleDateFormat smpdtfmt = new SimpleDateFormat("dd MMM yyyy",
Locale.getDefault());
private String tanggal1 = smpdtfmt.format(tglsekarang) ;
/* Pengaturan Status Bar */
private JPanel statusBar1 = new JPanel();
private JLabel lblExit1          = new JLabel(" " + "CopyRight LuqmanSoft Labs 2011",
JLabel.LEFT),
                                lblTgl1    = new JLabel(" " + tanggal1 + " ", JLabel.RIGHT);

/* Constructor Menu Utama */
public MenuUtama ()
{
/* Judul (Title) dan Icon Frame */
super ("Sistem Penilaian Kinerja Karyawan");
// setIconImage (getToolkit().getImage(""));
/* Pengaturan Ukuran dan Lokasi frame di Layar */
setSize (800, 600);
setLocation((Toolkit.getDefaultToolkit().getScreenSize().width - getWidth()) / 2,
            (Toolkit.getDefaultToolkit().getScreenSize().height - getHeight()) / 2);
/* Memberi Listener Untuk tombol close isinya memanggil method Keluar */
addWindowListener (new WindowAdapter () {
                                public void windowClosing (WindowEvent we) {
                                    Keluar ();
                                }
                            });
/*Mengatur tombol Hot Key (Alt+huruf)*/
/* Hot Key di Menu Utama */
mnuFileMaster.setMnemonic ('M');
mnuExit.setMnemonic ('x');

/* Hot Key di Sub Menu File Master */
mnuFMPegawai.setMnemonic ('P');
mnuInputSub.setMnemonic ('G');
mnuReturn1.setMnemonic ('u');

/* Hot Key di Sub Menu File Transaksi */
mnuInputNilai.setMnemonic ('T');
mnuFTGaji.setMnemonic ('A');
mnuReturn2.setMnemonic ('u');

/*Shortcut di File Master */
mnuFMPegawai.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_P,
Event.CTRL_MASK));
mnuInputSub.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_G,
Event.CTRL_MASK));
mnuReturn1.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_U, Event.CTRL_MASK));

/*Shortcut di File Transaksi */
```

```

mnuInputNilai.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_T,
Event.CTRL_MASK));
mnuFTGaji.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_A, Event.CTRL_MASK));
mnuReturn2.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_U, Event.CTRL_MASK));

/*Shortcut di Report */
mnuReportPegawai.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_P,
Event.CTRL_MASK));
mnuReportGol.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_G,
Event.CTRL_MASK));
mnuReportGaji.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_A,
Event.CTRL_MASK));
mnuReturn3.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_U, Event.CTRL_MASK));

/*Shortcut di Exit */
mnuExit1.setAccelerator (KeyStroke.getKeyStroke(KeyEvent.VK_I, Event.CTRL_MASK));

/* Mengatur ActionListener Pulldown Menu */
mnuFMPegawai.addActionListener (this);
mnuInputSub.addActionListener (this);
mnuReturn1.addActionListener (this);
mnuInputNilai.addActionListener (this);
mnuFTGaji.addActionListener(this);
mnuReturn2.addActionListener (this);
mnuReportPegawai.addActionListener (this);
mnuReportGol.addActionListener (this);
mnuReportGaji.addActionListener(this);
mnuReturn3.addActionListener (this);
mnuExit1.addActionListener (this);

/* Mengatur Sub Menu */
mnuFileMaster.add (mnuFMPegawai);
mnuFileMaster.add (mnuInputSub);
mnuFileMaster.addSeparator ();
mnuFileMaster.add (mnuReturn1);

mnuExit.add (mnuExit1);

/* Mengatur Bar Menu */
setJMenuBar (barMenu1);

/* Menambah Baris/MENU Ke Menu Bar */
barMenu1.add (mnuFileMaster);

barMenu1.add (mnuExit);

/* Mengatur ActionListener PopUp Menu */
mnuPopUp_FMPegawai.addActionListener (this);
mnuPopUp_InputSub.addActionListener (this);
mnuPopUp_FMReturn.addActionListener (this);

```

```
/* Menambah Submenu PopUp Ke Objek PopUp */
popMenu1.add (mnuPopUp_FMPegawai);
popMenu1.add (mnuPopUp_InputSub);
popMenu1.add (mnuPopUp_FMRReturn);

/* Mengatur AddMouseListener pada waktu Klik Kanan PopUp */
addMouseListener (new MouseAdapter () {
public void mousePressed (MouseEvent me) { checkMouseTrigger (me); }
public void mouseReleased (MouseEvent me) { checkMouseTrigger (me); }
private void checkMouseTrigger (MouseEvent me) {
    if (me.isPopupTrigger ())
        popMenu1.show (me.getComponent (), me.getX (), me.getY ());
}
});

/* Mengatur ToolTip Mouse di atas Objek */
mnuBtnFMPegawai.setToolTipText ("Data Pegawai");
mnuBtnInputSub.setToolTipText ("Data Subkriteria");

mnuBtnExit.setToolTipText ("Exit");

/* Mengatur addActionListener sehingga Objek saat di klik mengerti */
mnuBtnFMPegawai.addActionListener (this);
mnuBtnInputSub.addActionListener (this);

mnuBtnExit.addActionListener (this);

/* Meletakkan Objek tombol ke ToolbarPanel1 */
ToolbarPanel1.add (mnuBtnFMPegawai);
ToolbarPanel1.add (mnuBtnInputSub);

ToolbarPanel1.add (mnuBtnExit);

/* Meletakkan Objek tombol ke ToolbarPanel2 */
lblwaktu1.setForeground(Color.blue);
lblwaktu1.setFont(new Font("Arial",Font.BOLD,30));
ToolbarPanel2.add (lblwaktu1);

/* Panel ToolbarPanel1, ToolbarPanel2 diletakkan di Toolbar */
toolBar1.add (ToolbarPanel1);
toolBar1.add (ToolbarPanel2);

/* Mengatur Baris Status */
statusBar1.setLayout (new BorderLayout());
statusBar1.setFont(new Font("Arial",Font.BOLD,10));

/* Meletakkan objek Label ke Baris Status */
statusBar1.add (lblExit1, BorderLayout.WEST);
statusBar1.add (lblTgl1, BorderLayout.EAST);
```

```

/* Mengatur Warna Latar Belakang */
desktop1.setBackground(Color.gray);

/* Menambah Isi Container Dengan Objek toolbar, desktop dan statusbar */
getContentPane().add(toolBar1, BorderLayout.NORTH);
getContentPane().add(desktop1, BorderLayout.CENTER);
getContentPane().add(statusBar1, BorderLayout.SOUTH);

/* Memanggil (menjalankan) Jam */
setJam();

/* Menampilkan Menu Utama Ke Layar */
show();
}

/* Metode Memeriksa Objek yang mendapat listener "addActionlistener" */
public void actionPerformed (ActionEvent ae) {
    Object obj = ae.getSource();
    if (obj == mnuFMPEGawai || obj == mnuPopUp_FMPegawai || obj == mnuBtnFMPEGawai) {
        Pegawai FMPegawai = new Pegawai();
        desktop1.add (FMPegawai);
        FMPegawai.show ();
    }
    if (obj == mnuInputSub || obj == mnuPopUp_InputSub || obj == mnuBtnInputSub) {
        InputSub InputSub = new InputSub();
        desktop1.add (InputSub);
        InputSub.show();
    }
    if (obj == mnuInputNilai || obj == mnuBtnFTPenempatan) {
        InputNilai FMTempat = new InputNilai();
        desktop1.add (FMTempat);
        FMTempat.show();
    }
    if (obj == mnuFTGaji || obj == mnuBtnFTGaji) {
    }
    if (obj == mnuReportPegawai) {
    }

    if (obj == mnuReportGol) {
    }
    if (obj == mnuReportGaji) {
    }
    if (obj==mnuReturn1 || obj==mnuReturn2 || obj==mnuReturn3 || obj==mnuPopUp_FMReturn) {
        new MenuUtama ();
    }
}

```

```
if (obj == mnuExit1 || obj == mnuBtnExit) {
    Keluar ();
}
}

/* Methode Keluar Dari Sistem */
private void Keluar () {
    try {
        int reply = JOptionPane.showConfirmDialog (this,
            "Yakin Mau Keluar ?",
            "Sistem Informasi - Bye", JOptionPane.YES_NO_OPTION,
            JOptionPane.PLAIN_MESSAGE);
        if (reply == JOptionPane.YES_OPTION) {
            setVisible (false);
            dispose();
            System.out.println ("Thank !!!");
            System.out.println ("LuqmanSoft");
            System.out.println ("\n2010");
            setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        }
    }
    catch (Exception e) {
    }
}

/* Methode Setting Waktu Sistem */
public void setJam() {
    ActionListener taskPerformer = new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            String nol_jam = "", nol_menit = "", nol_detik = "";

            Date dateTime = new Date();
            int nilai_jam = dateTime.getHours();
            int nilai_menit = dateTime.getMinutes();
            int nilai_detik = dateTime.getSeconds();

            if (nilai_jam <= 9) nol_jam = "0";
            if (nilai_menit <= 9) nol_menit = "0";
            if (nilai_detik <= 9) nol_detik = "0";

            String jam = nol_jam + Integer.toString(nilai_jam);
            String menit = nol_menit + Integer.toString(nilai_menit);
            String detik = nol_detik + Integer.toString(nilai_detik);

            lblwaktu1.setText(jam + ":" + menit + ":" + detik + " ");
        }
    };
    new Timer(1000, taskPerformer).start();
}
```

 }

Program 5. Lihat Hasil Kinerja

```

/*
 * TampilNilai.java
 *
 * Created on 28 Jun 11, 17:40:02
 */

package sishris;

import java.awt.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;
import java.io.*;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Absolute1
 */
public class TampilNilai extends javax.swing.JFrame {

    /** Creates new form TampilNilai */
    public TampilNilai() {
        super("Sistem Penilaian Kinerja Karyawan");
        initComponents();
    }

    private void jTextField1FocusLost(java.awt.event.FocusEvent evt) {
        // TODO add your handling code here:
        Connect ObjKoneksi1 = new Connect();
        Connection con1;

        try {
            con1 = ObjKoneksi1.OpenConnect();
            Statement st1 = con1.createStatement();
            String sql1 = "select * from tbl_nilaikriteriasub where NIP = "
            + jTextField1.getText()+"";
            ResultSet res = st1.executeQuery(sql1);

            float k1 = 0, k2 = 0, k3 = 0, k4 = 0, k5 = 0, k6 = 0;

            if(res.next())
            {
                /* Jika kode pegawai Ditemukan Di Tabel subkriteria */

```

```
//      nama.setText(res.getString("nama"));
//      }
//      else
//      {
//          JOptionPane.showMessageDialog(this,"Pegawai tidak Tercatat");
//          jTextField1.setText("");

for(int a = 1; a <= 23; a++)
{
    System.out.println(k1 + " " + k2+" "+k3+" "+k4+" "+k5+" "+k6);
    if(a > 0 && a <= 4 )
        k1 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 5 && a <= 8)
        k2 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 9 && a <= 12)
        k3 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 13 && a <= 16)
        k4 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 17 && a <= 20)
        k5 += Float.parseFloat(res.getString("sk"+a));
    else if(a >= 21 && a <= 23)
        k6 += Float.parseFloat(res.getString("sk"+a));
    }
}
k1 /= 4;
k2 /= 4;
k3 /=4;
k4 /=4;
k5 /=4;
k6 /=3;

sql1 = "select * from tabel_kriteria";
res = st1.executeQuery(sql1);
for(int a = 1; a <= 6; a++)
{
    if(res.next())
    {
        switch(a)
        {
            case 1 :
                k1 *= Float.parseFloat(res.getString("Bobot"));
                break;
            case 2 :
                k2 *= Float.parseFloat(res.getString("Bobot"));
                break;
            case 3 :
```

```

        k3 *= Float.parseFloat(res.getString("Bobot"));
        break;
    case 4:
        k4 *= Float.parseFloat(res.getString("Bobot"));
        break;
    case 5:
        k5 *= Float.parseFloat(res.getString("Bobot"));
        break;
    case 6:
        k6 *= Float.parseFloat(res.getString("Bobot"));
        break;
    }
}

sql1 = "select nama_karyawan from tabel_karyawan where NIP =
"+jTextField1.getText()+"";
res = st1.executeQuery(sql1);
if(res.next())
{
    nama.setText(res.getString("nama_karyawan"));
}

kemampuanKerja.setText(String.valueOf(k1));
loyalitas.setText(String.valueOf(k2));
disiplin.setText(String.valueOf(k3));
prestasi.setText(String.valueOf(k4));
hubunganKerja.setText(String.valueOf(k5));
kepemimpinan.setText(String.valueOf(k6));
String ratax = String.valueOf((k1+k2+k3+k4+k5+k6)/6);
rataRata.setText(ratax);

sql1 = "update tbl_nilaikriteriasub set nilaiRataRata = "+ratax+" where NIP
="+jTextField1.getText()+"";
int rows = st1.executeUpdate(sql1);
if (rows == 1 )
    JOptionPane.showMessageDialog(this, "Nilai rata2 Ter-update!!!");

//JOptionPane.showMessageDialog(rootPane, sk1);
} catch (SQLException ex) {
    Logger.getLogger(TampilNilai.class.getName()).log(Level.SEVERE, null, ex);
}

}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    HRIS_HOME1 m = new HRIS_HOME1();

```

```
        m.setVisible(true);
        dispose();    // TODO add your handling code here:
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new TampilNilai().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JTextField disiplin;
    private javax.swing.JTextField hubunganKerja;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField kemampuanKerja;
    private javax.swing.JTextField kepemimpinan;
    private javax.swing.JTextField loyalitas;
    private javax.swing.JTextField nama;
    private javax.swing.JTextField prestasi;
    private javax.swing.JTextField rataRata;
    // End of variables declaration
}
```

Program 6 . Lihat Nilai Karyawan

```

/*
 * NilaiUrut.java
 *
 * Created on 05 Jul 11, 11:03:01
 */

package sishris;

import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Absolute1
 */
public class NilaiUrut extends javax.swing.JFrame {
    private KoneksiMySQL kmsql;
    private ModelTable ModelTable;

    /** Creates new form NilaiUrut */
    public NilaiUrut() {
        super("Sistem Penilaian Kinerja Karyawan");
        initComponents();
        kmsql = new KoneksiMySQL();
        ModelTable = new ModelTable ("SELECT tabel_karyawan.NIP,
tabel_karyawan.nama_karyawan,tabel_karyawan.Divisi, tbl_nilaikriteriasub.nilaiRataRata
FROM tabel_karyawan, tbl_nilaikriteriasub WHERE tabel_karyawan.NIP =
tbl_nilaikriteriasub.NIP ORDER BY nilaiRataRata" );
        jTable1.setModel (new DefaultTableModel(ModelTable.getObj(),
ModelTable.getColNames()));
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

        jScrollPane1.setName("jScrollPane1"); // NOI18N

```

```

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jTable1.setName("jTable1"); // NOI18N
jScrollPane1.setViewportView(jTable1);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Gambar/12.png"))); //
NOI18N
jLabel1.setText("Daftar Nilai Rata-rata Karyawan :");
jLabel1.setName("jLabel1"); // NOI18N

jButton1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Gambar/exit.png"))); // NOI18N
jButton1.setText("Kembali");
jButton1.setName("jButton1"); // NOI18N
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(375, 375, 375)
                    .addComponent(jButton1)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addGap(15, 15, 15)
                            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(layout.createSequentialGroup()
                            .addGap(20, 20, 20)
                            .addComponent(jLabel1))
                    )
                )
            )
        )
    .addContainerGap()
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(375, 375, 375)
                    .addComponent(jButton1)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addGap(15, 15, 15)
                            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
                                javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(layout.createSequentialGroup()
                            .addGap(20, 20, 20)
                            .addComponent(jLabel1))
                    )
                )
            )
        )
    .addContainerGap()
);

```

```

        .addGap(18, 18, 18)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 163,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 9,
Short.MAX_VALUE)
        .addComponent(jButton1)
        .addGap(21, 21, 21))
    );

    pack();
} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
HRIS_HOME1 m = new HRIS_HOME1();
    m.setVisible(true);
    dispose();    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new NilaiUrut().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

```

SEKILAS TENTANG PENULIS



Luqman Putra Wibowo

Luqman, lahir di kota Surabaya, Jawa Timur. Luqman merupakan putra pertama dari tiga bersaudara, dari ibunda **Aty** dan ayahanda **Bowo**. Saat ini bertempat tinggal di Griya Babatan Mukti blok P-6, Wiyung Surabaya.

Contact person :

Luqman_dudut@yahoo.co.id

085645376213

Daftar Riwayat Pendidikan :

1. SD Islam Al-Falah Surabaya 1996-2002
2. SMP Muhammadiyah 5 Surabaya 2002-2005
3. SMA Muhammadiyah 2 Surabaya 2005-2008
4. Politeknik Elektronika Negeri Surabaya-ITS 2008-2011

Selalu asah dan arahkan mata pedang, karena selama di dunia perang ini tak akan ada habisnya
