**PROPOSAL**

**SISTEM PENDUKUNG KEPUTUSAN**

**PENILAIAN DAN PEMBERIAN BONUS SALESMAN**

**PADA PT. MATAKAR KENDARI**

**DENGAN MENGGUNAKAN METODE *PROFILE MATCHING***

**Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Derajat Sarjana Teknik**



**ADIN SETIAWAN NIM. E1E1 15 100**

# JURUSAN TEKNIK INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS HALU OLEO

**2017**

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang Masalah

Dalam setiap perusahaan atau badan usaha akan memberikan gaji sebagai kompensasi dari kerja seorang salesmannya, disamping pemberian gaji pokok untuk memacu kinerja dan produktivitas kerja salesmannya. Pembayaran upah salesman dengan sistem bonus yang diterima oleh salesman disamping gaji pokok mereka berfungsi untuk merancang pekerja agar bekerja dengan lebih baik sehingga mencapai kinerja yang diharapkan oleh perusahaan, dikarenakan seorang salesman yang menerima bonus tersebut harus memenuhi beberapa kriteria tertentu yang sesuai dengan yang telah ditentukan oleh masing-masing instansi atau perusahaan. Demikian pula besaran jumlah bonus yang diterima oleh masingmasing salesman akan berbeda tergantung dengan kinerja dan beberapa faktor yang lain.

Bagi setiap perusahaan yang telah menggunakan sistem informasi berbasis komputer dalam kegiatan usahanya maka memerlukan sistem pendukung keputusan untuk menentukan karyawan manakah yang memiliki prioritas untuk mendapatkan gaji berdasarkan dengan kinerja dan produktivitasnya serta dapat menentukkan besarnya bonus yang pantas untuk diterima salesman tersebut. Sistem pendukung keputusan ini juga dapat berguna untuk memonitor kinerja salesman dari waktu ke waktu. Sistem ini juga dapat digunakan sebagai acuan dalam penentuan langkah selanjutnya bagi salesman yang berprestasi maupun tidak.

PT Matakar Kendari adalah salah satu perusahaan yang belum menggunakan proses pemberian bonus dengan menggunakan sistem komputer. PT. Matakar Kendari merupakan suatu perusahaan yang bergerak di bidang multidistributor, mendistribusikan berbagai macam produk kosmetik, makan, minuman dan obat-obatan.

Penelitian ini membahas tentang sistem pendukung keputusan pemberian bonus pada salesman untuk setiap periodenya. Sistem penentuan pemberian bonus

salesman pada PT. Matakar Kendari sampai saat ini masih dilakukan secara manual. Dengan mengecek satu persatu kriteria yang dijadikan pedoman dasar dalam pengambilan keputusan siapakah pegawai yang memperoleh bonus pada periode kali ini, membuat banyaknya waktu yang tersita dalam memilih, menimbang hingga memutuskan siapa pegawai yang layak diberikan bonus pada periode kali ini.

Sistem ini tidak dimaksudkan untuk menggantikan fungsi seorang manajer dalam mengambil keputusan tetapi hanya untuk membantu manajer dalam mengambil sebuah keputusan secara lebih cepat dan tepat, sesuai dengan kriteria yang diinginkan atau setidaknya mendekati kriteria yang diinginkan. Alternatifalternatif pilihan yang diharapkan dapat memberikan daftar referensi kepada pembuat keputusan sebelum benar-benar mengambil suatu keputusan akhir. Sistem pendukung keputusan ini menggunakan metode *Profile* *Matching* (Pencocokkan profil) yang merupakan suatu metode yang dapat membandingkan antara kompetensi yang dimiliki salesman dengan kompetensi yang ditentukan oleh perusahaan sehingga diperoleh perbedaan kompetensinya atau disebut gap. Metode ini dilakukan dengan menentuan nilai bobot untuk setiap kriteria, kemudian dilanjutkan dengan proses perangkingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif yang ada. Sehingga dapat mengurangi subjektifitas dalam pengambilan keputusan.

# Berdasarkan latar belakang yang telah dikemukakan, maka penulis mengambil judul tugas akhir “SISTEM PENDUKUNG KEPUTUSAN PENILAIAN DAN PEMBERIAN BONUS SALESMAN PADA PT. MATAKAR KENDARI DENGAN MENGGUNAKAN METODE *PROFILE*

***MATCHING”.***

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut dapat dirumuskan permasalahan yang akan diselesaikan yaitu bagaimana merancang dan mengimplementasikan sistem pendukung keputusan menggunakan metode *Profile Matching* pada penilaian dan pemberian bonus salesman di PT. Matakar Kendari.

## 1.3 Batasan Masalah

Permasalahan yang akan dibahas didalam judul ini mencakup beberapa hal yang dibatasi dalam ruang lingkup.

1. Data yang diolah adalah data angket untuk penilaian dan pemberian bonus salesman di PT. Matakar Kendari
2. Kriteria penilaian berupa Posisi/Jabatan, Target dan Perilaku.
3. Sistem Pendukung Keputusan ini hanya membahas salesman yang layak untuk mendapatkan bonus tersebut.
4. Metode yang digunakan adalah Metode *Profile Matching*.

## 1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam tugas akhir ini yaitu untuk membangun sebuah sistem pendukung keputusan untuk membantu pimpinan dalam pemberian bonus salesman dengan mencocokkan kriteria – kriteria dan bobot menggunakan metode *Profile Matching.*

## 1.5 Manfaat Penelitian

Adapun manfaat dari penulisan tugas akhir ini yaitu mempermudah pengolahan data dalam sistem pemberian bonus dan membantu dalam pengambilan keputusan untuk pemberian bonus salesman PT. Matakar Kendari.

## 1.6 Sistematika Penulisan

Penulisan laporan tugas akhir ini disusun secara bertahap sehingga pembaca dapat lebih mudah memahami isi laporan ini, laporan ini disusun dalam 6 bab yang menerapkan secara rinci hasil dari laporan tugas akhir, isi dari laporan ini adalah :

# BAB I PENDAHULUAN

Bagian ini berisi pendahuluan yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

|  |  |
| --- | --- |
| **BAB II** | **LANDASAN TEORI**  Pada bab ini menjelaskan tentang definisi sistem, alat bantu (*tools*), yang digunakan dalam pemodelan sistem pada tahap analisis, perancangan, landasan teori perangkat keras, landasan teori perangkat lunak, serta teori-teori lain yang mendukung topik Tugas Akhir ini. |
| **BAB III** | **METODOLOGI PENELITIAN**  Bagian ini memuat prosedur dan pengumpulan data, prosedur pengembangan perangkat lunak serta waktu dan tempat penelitian. serta metode pendekatan yang dilakukan dalam penyusunan Tugas Akhir. |
| **BAB IV** | **ANALISIS DAN PERANCANGAN**  Bab ini berisi perancangan sistem yang bertujuan untuk memenuhi kebutuhan pemakai sistem serta memberikan gambaran yang jelas dari sistem yang akan dibuat. |
| **BAB V** | **IMPLEMENTASI DAN PEMBAHASAN**  Bab ini membahas tentang inplementasi dari aplikasi yang dibuat dan akan dilakukan pembuatan aplikasi yang akan dibangun dengan komponen-komponen yang telah ada yang disesuaikan dengan permasalahan dan batasannya yang telah dijabarkan pada bab pertama. Bab ini juga berisi implementasi dari perancangan sistem yang dibuat, pengkodean hasil perancangan untuk menghasilkan *software* sesuai dengan *requiremen*t yang telah didefinisikan sebagai kebutuhan pengguna sistem. |
| **BAB VI** | **PENUTUP** |

Bagian ini memuat kesimpulan dari penelitian yang telah dilakukan serta saran untuk penelitian selanjutnya.

## 1.7 TINJAUAN PUSTAKA

Penelitian dalam mengembangkan sistem pendukung keputusan menggunakan Metode *profile matching* telah banyak dilakukan. Objek dan faktor penilaian yang digunakan sangat beragam, mulai dari penentuan penerima beasiswa, penetuan kenaikan jabatan karyawan sampai penentuan lokasi penempatan bidan PTT. Penelitian yang dilakukan (Darmawan, 2012), untuk mengembangkan sistem pendukung keputusan pemilihan beasiswa bagi mahasiswa STMIK Widya Pratama dengan metode *profile matching.* Sedangkan penelitian (Sherly, 2013) menggunakan tiga aspek (kapasitas intelektual, sikap kerja dan perilaku) dalam sistem pendukung keputusan pemberian bonus karyawan PT. Sanghyang seri persero. Berbeda dengan penelitian (Iqbal & Hartati, 2011) yang menggunakan empat kriteria (jarak, evaluasi diri, pengalaman kerja dan uji kompetensi) untuk penempatan bidan PTT (pegawai tidak tetap) pada Kabupaten Bireuen. Penelitian ini juga menggunakan *profile matching* dalam proses perhitungan yang diterapkan dalam sebuah sistem pendukung keputusan.

“Sistem Pendukung Keputusan Pemberian Reward Karyawan Menggunakan Metode *Profile Matching* Pada Rumah Makan Pondok Alam”. Oleh Lucky Priyo Prayogo Universitas Dian Nuswantoro. Dalam jurnal ini terdapat penelitian menggunakan Metode *Profile Matching* karena metode ini merupakan salah satu metode penyelesaian multi kriteria dimana dalam penentuan pemberian reward karyawan banyak kriteria yang harus dipertimbangkan. Metode pengembangan pada sistem ini menggunakan metode *waterfall*. Bahasa pemograman yang digunakan adalah Microsoft Visual Basic 6.0 dan MySQL sebagai *database server*.

# BAB II

# LANDASAN TEORI

## 2.1 Sistem Pendukung Keputusan

Sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Kusrini, 2007).

*Decision Support System*atau Sistem Pendukung Keputusan yang selanjutnya menjadi SPK, secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan komunikasi untuk masalah semi-terstruktur. Secara khusus, SPK didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manajer maupun sekelompok manajer dalam memecahkan masalah semi-terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu (Hermawan, 2005).

Pembuatan keputusan merupakan fungsi utama seorang manajer atau *administrator*. Kegiatan pembuatan keputusan meliputi identifikasi masalah, pencarian alternatif penyelesaian masalah, evaluasi dari alternatif-alternatif tersebut dan pemilihan alternatif keputusan yang terbaik. Kemampuan seorang manajer dalam membuat keputusan dapat ditingkatkan apabila manajer mengetahui dan menguasai teori dan teknik pembuatan keputusan. Dengan peningkatan kemampuan manajer dalam pembuatan keputusan diharapkan dapat meningkatkan kualitas keputusan yang dibuatnya, dan hal ini tentu akan meningkatkan efisiensi kerja manajer yang bersangkutan.

### 2.1.1 Konsep Dasar Sistem Pendukung Keputusan

Pada awalnya Turban and Aronson (1998), mendefinisikan sistem penunjang keputusan (*Decision Support Systems* – DSS) sebagai sistem yang digunakan untuk mendukung dan membantu pihak manajemen melakukan

6

pengambilan keputusan pada kondisi semi terstruktur dan tidak terstruktur. Pada dasarnya konsep DSS hanyalah sebatas pada kegiatan membantu para manajer melakukan penilaian serta menggantikan posisi dan peran manajer.

Konsep DSS pertama kali diperkenalkan pada awal tahun 1970-an oleh

Michael Scott Morton, yang selanjutnya dikenal dengan istilah “*Management Decision System”*. Konsep DSS merupakan sebuah sistem interaktif berbasis komputer yang membantu pembuatan keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang bersifat tidak terstruktur dan semi terstruktur. DSS dirancang untuk menunjang seluruh tahapan pembuatan keputusan, yang dimulai dari tahapan mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pembuatan keputusan sampai pada kegiatan mengevaluasi pemilihan alternatif.

### 2.1.2 Konsep Pengambilan Keputusan

Definisi keputusan yang dikemukakan para ahli dijelaskan sebagai berikut (Hasan, 2004) :

Menurut Ralph C. Davis

Keputusan adalah hasil pemecahan masalah yang dihadapinya dengan tegas. Suatu keputusan merupakan jawaban yang pasti terhadap suatu pertanyaan. Keputusan harus dapat menjawab pertanyaan tentang apa yang dibicarakan dalam hubungannya dengan perencanaan. Keputusan dapat pula berupa tindakan terhadap pelaksanaan yang sangat menyimpang dari rencana semula.

### 2.1.3 Fase-fase Proses Pengambilan Keputusan

Proses pengambilan keputusan melibatkan empat tahap utama yaitu 1.

Tahap Intelegensi *(Intelligence Phase),* 2. Tahap Perencanaan *(Design Phase)*, 3. Tahap Pilihan *(choice phase)* dan 4. Tahap Implementasi *(Implementation).* Seperti tampak pada gambar tahap-tahap dalam proses pengambilan keputusan berikut (Turban, 2005) :

***Intelegence Phase***



Sasaran Organisasional



Pengumpulan Data



Identifikasi Masalah



Klasifikasi Masalah



Pernyataan Masalah

***Design Phase***



Formulasi Model



Menentukan Kriteria Pemilihan

***Choice Phase***



Solusi Untuk Model



Pemilihan Alternatif Terbaik

***Implementation Phase***

Penariakan Keputusan

# Gambar 2.1 Pengambilan Keputusan / Proses Pemodelan SPK

Proses pengambilan keputusan dimulai dari fase inteligensi kemudian realitas diuji dan masalah diidentifikasi dan ditentukan. Kepemilikan masalah juga ditetapkan. Selanjutnya pada fase desain akan dikonstruksi sebuah model yang merepresentasikan sistem. Hal ini dilakukan dengan membuat asumsi-asumsi yang menyederhanakan realitas dan menuliskan hubungan di antara semua variabel. Model ini kemudian divalidasi dan ditentukanlah kriteria dengan menggunakan prinsip memilih untuk mengevaluasi alternatif tindakan yang telah diidentifikasi. Proses pengembangan model sering mengidentifikasi solusi-solusi alternatif dan demikian sebaliknya.

Selanjutnya adalah fase pilihan yang meliputi pilihan terhadap solusi yang diusulkan untuk model (tidak memerlukan masalah yang disajikan). Solusi ini diuji untuk menentukan viabilitasnya. Begitu solusi yang diusulkan tampak masuk akal, maka siap untuk masuk kepada fase terakhir yakni fase implementasi keputusan. Hasil implementasi yang berhasil adalah dapat dipecahkannya masalah riil. Sedangkan kegagalan implementasi mengharuskan kembali ke fase sebelumnya. (Turban, 2005).

## 2.1.4 *Intelligence Phase*

Tahap intelegensi (*Intelligence Phase*) merupakan tahap pendefinisian masalah serta identifikasi informasi yang dibutuhkan yang berkaitan dengan persoalan yang dihadapi serta keputusan yang akan diambil, tentunya persoalan yang dihadapi harus dirumuskan terlebih dahulu secara jelas. Tahap intelegensi terdiri atas :

***2.1.4.1*** Identifikasi Masalah (Peluang)

Fase inteligensi dimulai dengan identifikasi terhadap tujuan dan sasaran organisasional yang berkaitan dengan isu yang diperhatikan (misal manajemen inventori, seleksi kerja), dan determinasi apakah tujuan tersebut telah terpenuhi. Masalah terjadi karena ketidakpuasan terhadap status *quo.* Ketidakpuasan merupakan hasil dari perbedaaan antara apa yang di inginkan (harapkan) dan apa yang terjadi. Pada fase pertama ini, seseorang berusaha menentukan apakah ada suatu masalah, mengidentifikasi gejala-gejalanya, menentukan keluasannya, dan mendefinisikan-nya secara eksplisit.

Eksistensi masalah dapat ditentukan dengan memonitor dan menganalisis tingkat produktivitas organisasi. Ukuran produktivitas dan konstruksi sebuah model didasarkan pada data riil. Menentukan apakah masalah benar-benar ada, di mana masalah tersebut, dan seberapa signifikan, dapat dilakukan setelah investigasi awal selesai dilakukan. Poin kunci adalah apakah sistem informasi melaporkan masalah atau hanya melaporkan gejala-gejala dari sebuah masalah.

***2.1.4.2*** Klasifikasi Masalah

Klasifikasi masalah adalah konseptualisasi terhadap suatu masalah dalam rangka menempatkannya dalam suatu kategori yang dapat didefinisikan, barangkali mengarah kepada suatu pendekatan solusi standar. Pendekatan yang penting mengklasifikasikan masalah-masalah sesuai tingkat strukturisasi pada masalah tersebut.

***2.1.4.3*** Kepemilikan Masalah

Menentukan kepemilikan masalah merupakan hal penting pada fase inteligensi. Sebuah masalah ada di dalam sebuah organisasi hanya jika seseorang atau beberapa kelompok mengambil tanggung jawab untuk mengatasinya dan jika organisasi punya kemampuan untuk memecahkannya. Ketika kepemilikan masalah tidak ditentukan, maka seseorang tidak melakukan tugasnya atau masalah akan diidentifikasi sebagai masalah orang lain.

## 2.1.5 *Design phase*

Fase desain meliputi penemuan atau mengembangkan dan menganalisis tindakan yang mungkin untuk dilakukan. Hal ini meliputi pemahaman terhadap masalah dan menguji solusi yang layak. Tahap desain terdiri atas :

2.1.5.1 Memilih Sebuah Prinsip Pilihan

Prinsip pilihan adalah sebuah kriteria yang menggambarkan akseptabilitas dari sebuah solusi (kemampuan untuk data diterima). Pada sebuah model, prinsip tersebut adalah sebuah variabel hasil. Memilih sebuah prinsip pilihan bukanlah bagian dari fase pilihan, namun melibatkan bagaimana kita membangun sasaran pengambilan keputusan dan bagaimana sasaran tersebut disatukan ke dalam suatu model.

2.1.5.2 Mengembangkan Alternatif-alternatif

Bagan signifikan dari proses pembangunan model adalah menghasilkan berbagai alternatif. Pencarian terhadap berbagai alternatif biasanya terjadi setelah kriteria untuk mengevaluasi alternatif dilakukan. Sekuensi ini dapat mengurangi pencarian alternatif dan usaha yang dikeluarkan untuk mengevaluasinya, namun mengidentifikasi alternatif-alternatif potensial kadang-kadang dapat membantu mengidentifikasi kriteria.

2.1.5.3 Mengukur Hasil Akhir

Nilai dari sebuah alternatif dievaluasi dalam hal pencapaian tujuan. Kadang-kadang suatu hasil dinyatakan secara langsung dalam istilah tujuan. Sebagai contoh, laba adalah hasil akhir, maksimalisasi laba adalah suatu tujuan, dan keduanya dinyatakan dalam terminologi dollar. Hasil akhir seperti keputusan pelanggan dapat diukur dengan jumlah keluhan, dengan tingkat loyalitas terhadap sebuah produk, atau dengan rating hasil survei.

## 2.1.6 *Choice Phase*

Pilihan merupakan tindakan pengambilan keputusan yang kritis. Fase pilihan adalah fase di mana dibuat suatu keputusan yang nyata dan diambil suatu komitmen untuk mengikuti suatu tindakan tertentu. Batas antara fase pilihan dan desain sering tidak jelas karena aktivitas tertentu dapat dilakukan selama kedua fase tersebut dan karena orang dapat sering kembali dari aktivitas pilihan ke aktivitas desain. Sebagai contoh, seseorang dapat menghasilkan alternatif baru selagi mengevaluasi alternatif yang ada. Fase pilihan meliputi pencarian, evaluasi, dan rekomendasi terhadap suatu solusi yang tepat untuk model. Sebuah solusi untuk sebuah model adalah sekumpulan nilai spesifik untuk variabel-variabel keputusan dalam suatu alternatif yang telah dipilih.

Memecahkan sebuah model tidak sama halnya dengan memecahkan masalah yang direpresentasikan oleh model. Solusi untuk model menghasilkan sebuah solusi yang direkomendasikan untuk masalah. Masalah dianggap dipecahkan hanya jika solusi yang direkomendasikan sukses diterapkan.

Pemecahan sebuah model pengambilan keputusan melibatkan pencarian terhadap suatu tindakan yang tepat. Pendekatan pencarian melibatkan teknik analitik (memecahkan suatu formula), algoritma (prosedur langkah-demilangkah), heuristik (aturan utama), dan *blind search* (menembak di dalam gelap, idealnya dalam suatu cara yang logis).

Masing-masing alternatif harus dievaluasi. Jika suatu alternatif mempunyai berbagai tujuan, maka semua tujuan harus diuji dan seimbang jika dihadapkan dengan yang lainnya. Analisis sensitivitas digunakan untuk menentukan ketangguhan sembarang alternatif yang diberikan (sedikit perubahan dalam perameter idelanya mendorong ke sedikit atau tidak ada perubahan dalam alternatif yang dipilih).

## 2.1.7 *Implementation phase*

Pada hakikatnya implementasi suatu solusi yang diusulkan untuk suatu masalah adalah inisiasi terhadap hal baru, atau pengenalan terhadap perubahan. Definisi implementasi sedikit rumit karena implementasi merupakan sebuah proses yang panjang dan melibatkan batasa-batasan yang tidak jelas. Implementasi berarti membuat suatu solusi yang direkomendasikan bisa bekerja, tidak memerlukan implementasi suatu sistem komputer.

## 2.1.8 Karakteristik dan Kemampuan Sistem Pendukung Keputusan

Menurut (Turban, 1996), ada beberapa karakteristik dari SPK, di antaranya adalah sebagai berikut:

1. Mendukung seluruh kegiatan organisasi
2. Mendukung beberapa keputusan yang saling berinteraksi
3. Dapat digunakan berulang kali dan bersifat konstan
4. Terdapat dua komponen utama, yaitu data dan model
5. Menggunakan baik data ekternal maupun internal
6. Memiliki kemampuan *what-if analysis* dan *goal seeking analysis*
7. Menggunakan beberapa model kuantitatif

Selain itu, Turban menjelaskan kemampuan yang harus dimiliki oleh sebuah sistem pendukung keputusan, diantaranya adalah sebagai berikut:

1. Menunjang pembuatan keputusan manajemen dalam menangani masalah semi terstruktur dan tidak terstruktur.
2. Membantu manajer pada berbagai tingkatan manajemen, mulai dari manajemen tingkat atas sampai manajemen tingkat bawah.
3. Menunjang pembuatan keputusan secara kelompok dan perorangan.
4. Menunjang pembuatan keputusan yang saling bergantungan dan berurutan.
5. Menunjang tahap-tahap pembuatan keputusan antara lain *intelligence, design, choice* dan *implementation.*
6. Menunjang berbagai bentuk proses pembuatan keputusan dan jenis keputusan.
7. Kemampuan untuk melakukan adaptasi setiap saat dan bersifat fleksibel.
8. Kemudahan melakukan interaksi sistem.
9. Meningkatkan efektivitas dalam pembuatan keputusan daripada efisiensi.
10. Mudah dikembangkan oleh pemakai akhir.
11. Kemampuan pemodelan dan analisis dalam pembuatan keputusan.
12. Kemudahan melakukan pengaksesan berbagai sumber dan format data.

Disamping berbagai kemampuan dan karakteristik seperti yang telah dikemukakan, sistem pendukung keputusan memiliki juga keterbatasan, antara lain:

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan yang sebenarnya.
2. Kemampuan suatu sistem pendukung keputusan terbatas pada pengetahuan dasar serta model dasar yang dimilikinya.
3. Proses-proses yang dapat dilakukan oleh sistem pendukung keputusan biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.
4. Sistem pendukung keputusan tidak memiliki intuisi seperti yang dimiliki oleh manusia. Karena sistem pendukung keputusan hanya suatu kumpulan perangkat keras, perangkat lunak dan sistem operasi yang tidak dilengkapi oleh kemampuan berpikir.
5. Secara implisit, sistem pendukung keputusan berlandaskan pada kemampuan dari sebuah sistem berbasis komputer dan dapat melayani penyelesaian masalah.

## 2.1.9 Keuntungan Sistem Pendukung Keputusan

Beberapa keuntungan penggunaan SPK antara lain adalah sebagai berikut (Surbakti, 2002):

1. Mampu mendukung pencarian solusi dari berbagai permasalahan yang kompleks.
2. Dapat merespon dengan cepat pada situasi yang tidak diharapkan dalam konsisi yang berubah-ubah.
3. Mampu untuk menerapkan berbagai strategi yang berbeda pada konfigurasi berbeda secara cepat dan tepat.
4. Pandangan dan pembelajaran baru.
5. Sebagai fasilitator dalam komunikasi.
6. Meningkatkan kontrol manajemen dan kinerja.
7. Menghemat biaya dan sumber daya manusia (SDM).
8. Menghemat waktu karena keputusan dapat diambil dengan cepat.
9. Meningkatkan efektivitas manajerial, menjadikan manajer dapat bekerja lebih singkat dan dengan sedikit usaha.
10. Meningkatkan produktivitas analisis.

## 2.1.10 Komponen Sistem Pendukung Keputusan

Adapun komponen-komponen dari SPK adalah sebagai berikut :

1. Data *Management*

Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS).

1. Model *Management*

Melibatkan model finansial, statistikal, *management science*, atau berbagai model kualitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang dibutuhkan.

1. *Communication*

*User* dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.

*Other computer*

*-*

*based*

*system*

Data

*Management*

Model

*Management*

*Knowtedge*

*Manager*

Dialog

*Management*

*Manager ( USER )*

Data

:

*external*

&

*Internal*

# Gambar 2.2. Model Konseptual SPK ( Subakti, 2002 )

## 2.2 Metode *Profile Matching*

Metode *Profile Matching* adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat *variable predicator* yang ideal yang harus dimiliki, bukannya tingkat minimal yang harus dipenuhi atau dilewati dalam proses Metode *Profile Matching Modelling* secara garis besar merupakan proses membandingkan antara nilai data *actual* dari suatu *profile* yang akan dinilai dengan nilai profil yang diharapkan, sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap).

Metode *profile matching* atau pencocokan profil adalah metode yang sering sebagai mekanisme dalam pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subjek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati. (Kusrini, 2007).

Berikut adalah beberapa tahapan dan perumusan perhitungan dengan metode *profile matching* (Kusrini, 2007) :

1. Pembobotan

Pada tahap ini, akan ditentukan bobot nilai masing – masing aspek dengan menggunakan bobot nilai yang telah ditentukan bagi masing-masing aspek itu sendiri. Adapun masukan dari proses pembobotan ini adalah selisih dari profil karyawan dan profil jabatan. Dalam penentuan peringkat pada aspek kapasitas intelektual, sikap kerja dan perilaku untuk jabatan yang sama pada setiap gap, diberikan bobot nilai sesuai dengan tabel berikut :

Tabel 2.1 Bobot Nilai GAP

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Selisih Gap** | **Bobot Nilai** | **Keterangan** |
| 1 | 0 | 5 | Kompetensi sesuai dengan yang di butuhkan |
| 2 | 1 | 4,5 | Kompetensi subjek kelebihan 1 tingkat / level |
| 3 | -1 | 4 | Kompetensi subjek kekurangan 1 tingkat / level |
| 4 | 2 | 3,5 | Kompetensi subjek kelebihan 2 tingkat / level |
| 5 | -2 | 3 | Kompetensi subjek kekurangan 2 tingkat / level |
| 6 | 3 | 2,5 | Kompetensi subjek kelebihan 3 tingkat / level |
| 7 | -3 | 2 | Kompetensi subjek kekurangan 3 tingkat / level |

Lanjutan Tabel 2.1 Bobot Nilai GAP

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | 4 | 1.5 | Kompetensi subjek kelebihan 4 Tingkat / level |
| 9 | -4 | 1 | Kompetensi subjek kekurangan 4 tingkat / level |

1. Perhitungan dan Pengelompokan *Core* dan *Secondary Factor*

Setelah menentukan bobot nilai gap untuk ketiga aspek yang dibutuhkan, kemudian tiap aspek dikelompokan lagi menjadi 2 kelompok yaitu *core factor* dan *secondary factor.*

*Core Factor* (Faktor Utama)

*Core factor* (CF) merupakan aspek (kompetensi) yang paling menonjol/paling dibutuhkan oleh suatu jabatan yang diperkirakan dapat menghasilkan kinerja optimal.Untuk menghitung *core factor* digunakan rumus :

∑ NC

NCI =  , , , , , , ( 2.1 )

∑ IC

Keterangan :

*NCI* = Nilai rata – rata (CF) aspek Kualitas

*NC* = Jumlah total nilai (CF) aspek Kualitas

*IC* = Jumlah item (CF)

*Secondary Factor* (Faktor Pendukung)

*Secondary factor* (SF) adalah item – item selain aspek yang ada pada core factor. Untuk menghitung *secondary factor* digunakan rumus

∑ NS

NSI =  , , , , , , ( 2.2 )

∑ IS

Keterangan :

*NSI* = Nilai rata – rata (SF) aspek Kualitas

*NS*  = Jumlah total nilai (SF) aspek Kualitas

*IS*  = Jumlah item (SF)

Rumus diatas adalah rumus untuk menghitung *core factor* dan *secondary factor* dari aspek Kualitas. Rumus tersebut juga digunakan untuk menghitung *core factor* dan *secondary factor* dari aspek Harga dan Pelayanan.

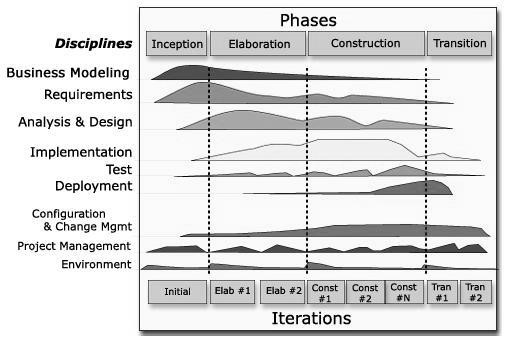
## 2.3 Metode Pengembangan Sistem

*Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perankat lunak.

RUP menggunakan konsep *object oriented*, dengan aktivitas yang berfokus pada pengembangan model dengan menggunakan *Unified Modeling Language* (UML), melalui gambar dibawah dapat dilihat bahwa RUP memiliki, yaitu:

1. Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari *phase* selanjutnya. Setiap *phase* dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *IncePT.ion*, *Elaboration*, *Construction*, dan *Transition*.
2. Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspekaspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan dalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas : *Business*

*Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration* dan *Change Manegement, Project Management, Environtment.*



# Gambar 2.3 Arsitektur *Rational Unified Process* (Ambler, 2005)

Pada penggunaan kedua standar tersebut diatas yang berorientasi objek (*object orinted*) memiliki manfaat yakni:

1. *Improve productivity*

Standar ini dapat memanfaatkan kembali komponen-komponen yang telah tersedia/dibuat sehingga dapat meningkatkan produktivitas.

1. *Deliver high quality system*

Kualitas sistem informasi dapat ditingkatkan sebagai sistem yang dibuat pada komponen-komponen yang telah teruji (*well-tested* dan *well-proven*) sehingga dapat mempercepat *delivery* sistem informasi yang dibuat dengan kualitas yang tinggi.

1. *Lower maintenance cost*

Standar ini dapat membantu untuk menyakinkan dampak perubahan yang terlokalisasi dan masalah dapat dengan mudah terdeteksi sehingga hasilnya biaya pemeliharaan dapat dioptimalkan.

1. *Facilitate reuse*

Standard ini memiliki kemampuan yang mengembangkan komponenkomponen yang dapat digunakan kembali untuk pengembangan aplikasi yang lainnya.

1. *Manage complexity*

Standar ini mudah untuk mengatur dan memonitor semua proses dari semua tahapan yang ada sehingga suatu pengembangan sistem informasi yang amat kompleks dapat dilakukan dengan aman dan sesuai dengan harapan semua manajer proyek IT/IS yakni *deliver good quality software within cost and schedule time and the users accepted*.

## 2.4 Perangkat Perancangan Sistem

Dalam pengembangan perangkat lunak sistem pendukung keputusan ini, digunakan tools *Unified Modelling Language* 2.0 (UML). UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Larman, 2004; Fowler, 2003). Sedangkan konsepsi dasar UML 2.0 yang akan digunakan dalam pengembangan sistem itu sendiri antara lain:

1. *Use Case Diagram* : menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
2. *Class Diagram* : sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).
3. *Statechart Diagram* : menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart* diagram menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart* diagram).
4. *Activity Diagram* : menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.
5. *Sequence Diagram* : menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display,* dan sebagainya) berupa *message* yang digambarkan terhadap waktu*. Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).
6. *Component Diagram* : menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.
7. *Deployment Diagram* : menggambarkan detail bagaimana komponen di*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.

### 2.4.1 *Use Case Diagram*

*Use case diagram* adalah diagram yang menampilkan aktor, *use case* dan hubungan yang terjadi antara aktor dan *use case*. *Use case* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Aktor merupakan sebuah peran yang dimainkan seorang pengguna dalam kaitannya dengan sistem (Munawar, 2005).

## Tabel 2.2 Simbol *Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| NO | GAMBAR | NAMA | KETERANGAN |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Generalization* | Hubungan dimana objek anak  (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 3 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 4 |  | *Association* | Yang menghubungkan antara objek satu dengan objek lainnya. |
| 5 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |

### 2.4.2 *Activity Diagram*

*Activity Diagram* adalah teknik untuk menggambarkan logika prosedural, proses bisnis dan jalur kerja. Dalam beberapa hal, *activity diagram* memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara *activity diagram* dan notasi diagram alir adalah diagram ini mendukung perilaku paralel (Munawar, 2005).

## Tabel 2.3 Simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| NO | GAMBAR | NAMA | KETERANGAN |
| 1 |  | *Actifity* | Memperlihatkan bagaimana masingmasing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | *Action* | *State* dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| 4 |  | *Actifity*  *Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| 5 |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

### 2.4.3 *Sequence Diagram*

*Sequence diagram* menjabarkan perilaku sebuah skenario. Skenario adalah rangkaian langkah-langkah yang menjabarkan sebuah interaksi antara seorang pengguna dengan sebuah sistem. *Sequence diagram* menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek tersebut di dalam *use case* (Munawar, 2005).

## Tabel 2.4 Simbol *Sequence Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| NO | GAMBAR | NAMA | KETERANGAN |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasiinformasi tentang aktivitas yang terjadi. |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasiinformasi tentang aktifitas yang terjadi. |
| 4 |  | *Message* | Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy.* |

### 2.4.4 *Class Diagram*

*Class diagram* adalah diagram yang mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara objekobjek tersebut. *Class diagram* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Munawar, 2005).

# Tabel 2.5 Simbol *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| NO | GAMBAR | NAMA | KETERANGAN |
| 1 |  | *Generalization* | Hubungan dimana objek anak  *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary*  *Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor.* |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |

## 2.5 Visual Basic.Net

Visual basic.NET sendiri merupakan bahasa pemrograman yang terdapat dalam Visual Studio .NET untuk mendukung rencana Microsoft ke depan dalam *platform* .NETnya, bisa dikatakan visual basic.NET sebagai bahasa pemrograman yang telah cukup matang menjadikan visual basic .NET menjadi bahasa pemrograman yang akan menjadi pintu gerbang bagi banyak kalangan *developer* dalam *platform* Microsoft yang baru.

Microsoft Visual Basic.NET adalah pengembangan lebih lanjut dari bahasa pemrograman Visual Basic 6.0 yang dikenal dengan kemampuan *Rapid Application Development* dan mengimplementasikan konsep OOP *(Object Oriented Programming)* secara penuh dan merupakan sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET *Framework,* dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para *programmer* dapat membangun aplikasi Windows Forms, Aplikasi web berbasis ASP.NET, dan juga aplikasi *command-line.* Visual Basic.Net merupakan bahasa pemrograman yang berorientasi objek yang memiliki fitur seperti *abstraction, encapsulation, inheritance, polymorphism,* dan juga termasuk penambahan keyword class untuk mendefinisikan kelas-kelas dan *keyword inheritance* untuk objek *inheritance.*

## *2.6 MySQL*

Diagram *MySQL* adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan cepat, *multiuser* serta menggunakan perintah standar *SQL*. *MySQL* memiliki dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. *MySQL* yang biasa digunakan adalah *MySQL* *Free Software* yang berada di bawah lisensi *GNU/GPL* (*General Public License*). Sebagai *database server* yang *free*, artinya *MySQL* dapat secara bebas digunakan untuk kepentingan pribadi atau usaha. Selain sebagai *server*, *MySQL* dapat juga berperan sebagai *client* sehingga sering disebut *database client/server*.

Pada mulanya *MySQL* hanya berjalan di bawah *Platform* Unix, tetapi pada perkembangannya muncul beberapa distro yang mampu berjalan pada beberapa *Platform* yang sifatnya *shareware* dan *corporate* seperti untuk *Linux* dan *Windows*. Di dalam sistemnya, *MySQL* merekam semua data *user* dalam sebuah tabel *user* yang berada pada *database* bernama *MySQL* untuk penanganan hak-hak akses *user*. Sebagai *database server,* *MySQL* mampu menangani beberapa aplikasi berupa program kompiler maupun bahasa *Scripting Server Site* seperti *PHP, Perl, CGI*, dan Java untuk mengakses data yang disimpannya.

Kelebihan yang ditawarkan oleh *MySQL* sebagai *Relational Database Management System* *(RDBMS*) adalah :

1. *MySQL* adalah sebuah *softwareopen source* dengan kapasitas penyimpanan data hingga berukuran *terabytes.*
2. *MySQL* merupakan *database client/server* yang *multiusers,* sehingga sebagai *server* dapat terhubung ke media internet untuk eksplorasinya. Sedangkan sebagai *client* dapat melakukan *query* untuk mengakses *database server.* Selain itu dapat digunakan oleh banyak pengguna sekaligus.
3. *MySQL* didukung oleh *ODBC* (*Open Database Connectivity*), artinya *database*nya dapat diakses aplikasi apa saja seperti *Java*, *Delphi*, dan *Visual Basic*.

## 2.7 Definisi Salesman

Seperti telah diutarakan bahwa pengertian Sales secara sederhana adalah penjualan. Dalam bukunya Sihite (1996:86) menyebutkan bahwa Sales adalah Merchandise *(Something to be sold) plus Service.* Dalam buku yang sama juga dijabarkan mengenai Salesmanship yaitu kecakapan seorang Sales dalam menjual yang meliputi proses dalam penjualan yang dimulai dari langkah pertama sampai dengan terlaksananya suatu penjualan. Jadi pengertian Sales Person atau Salesman di sini adalah individu yang menawarkan suatu produk dalam suatu proses penjualan.

## 2.8 Bonus

Bonus adalah kompensasi tambahan yang diberikan kepada seorang karyawan yang nilainya di atas gaji normalnya. Bonus bisa digunakan sebagai penghargaan terhadap pencapaian tujuan-tujuan spesifik yang ditetapkan oleh perusahaan, atau untuk dedikasinya kepada perusahaan.

Bonus merupakan program yang wajar dalam setiap perusahaan.

Alasannya adalah karena perusahaan percaya terhadap filosofi “memberi imbalan untuk prestasi” di mana bonus terkait erat dengan dua ukuran penting berikut: seberapa bagus kinerja anda berdasarkan ekspektasi manajer dan seberapa bagus kinerja perusahaan berdasarkan apa yang di harapkan.

## *2.9 Flowchart*

*Flowchart* adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* membantu analis dan *programmer* untuk memecahkan masalah kedalam segmensegmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Sistem *flowchart*adalah urutan proses dalam sistem dengan menunjukkan alat media *input, output* serta jenis media penyimpanan dalam proses pengolahan data. Program *flowchart*adalah suatu bagan dengan simbolsimbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

**Tabel 2.6 Lambang *Flowchart***

|  |  |  |
| --- | --- | --- |
| **BAGAN** | **NAMA** | **FUNGSI** |
|  | Terminator | Awal atau akhir program |
|  | *Flow* | Arah aliran program |
|  | *Preparation* | Inisialisasi/pemberian nilai awal |
|  | *Process* | Proses/pengolahan data |
|  | *Input/output* data | Input/output data |
|  | Sub program | Sub program |
|  | *Decision* | Seleksi atau kondisi |
|  | *On page connector* | Penghubung bagianbagian *flowchart* pada halaman yang sama |
|  | *Off page connector* | Penghubung bagianbagian *flowchart* pada halaman yang berbeda |
|  | *Comment* | Tempat komentar tentang suatu proses |

# BAB III

# METODOLOGI PENELITIAN

## 3.1 Tempat dan Waktu Penelitian

### 3.1.1 Tempat Penelitian

Penelitian ini dilakukan pada PT. Matakar Kendari. PT Matakar Kendari merupakan perusahaan Multidistributor yang mendistribusikan berbagai macam produk seperti kosmetik, makanan, minuman, obat-obatan dan lain-lain. Perusahaan ini memperkerjakan lebih dari 40 karyawan, lokasi peruahaan berada di Jl. Supu Yusuf.

### 3.1.2 Waktu Penelitian

Waktu yang diperlukan dalam pengembangan aplikasi sistem pendukung keputusan pemberian bonus salesman akan dilakukan selama 2 bulan mulai dari Desember 2016 sampai dengan Januari 2017. Untuk alur penelitian dapat dijelaskan pada Tabel 3.1 berikut :

# Tabel 3.1 Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | *RUP* |  | Waktu 2016 | | | | | | |
|  | Desember 2016 | | | Januari 2017 | | | |
| I | II | III | IV | I | II | III | IV |
| 1 | *Inception* |  |  |  |  |  |  |  |  |
| 2 | *Elaboration* |  |  |  |  |  |  |  |  |
| 3 | *Construction* |  |  |  |  |  |  |  |  |
| 4 | *Transition* |  |  |  |  |  |  |  |  |

## 3.2 Metodologi Pengumpulan Data

Dalam penelitian ini dilakukan beberapa metode untuk mengumpulkan data yang berguna dalam pembangunan sistem. Metode pengumpulan data tersebut antara lain :

30

1. Studi Literatur

Metode ini dilaksanakan dengan melakukan studi kepustakaan untuk mempelajari metode dan bahan pembelajaran lainnya yang dapat mendukung berjalannya pembangunan sistem dengan baik. Studi literatur ini diperoleh dari buku, artikel, jurnal, *e-book* dan sumber-sumber lainnya yang relevan untuk menunjang penyelesaian penelitian ini.

1. Metode Interview

Metode ini dilakukan dengan cara mengadakan wawancara atau bertanya langsung kepada orang yang ahli dibidangnya untuk mendapatkan informasi yang akan dijadikan bahan untuk pembuatan aplikasi nantinya.

## 3.3 Prosedur Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan dalam pembangunan sistem ini adalah metode *Rational Unified Process* *(RUP)* yang terdiri dari beberapa tahap yaitu :

1. *Inception* ( Permulaan )

Pada fase *inception* terdapat beberapa alur kerja yaitu *business modeling, requirements,* analisis dan desain. *Business modeling* dilakukan analisis mengenai aturan pada sistem yang dikembangkan yaitu algoritma yang akan digunakan pada sistem dalam hal ini proses awal sistem hingga sampai pada hasil pemberian bonus karyawan. Kemudian fase *requirement* yaitu analisis kebutuhan sistem yang akan dikembangkan sehingga analisis dan desain selanjutnya dapat dilakukan. Pada tahap analisis yaitu tahap penjelasan persyaratan dan kebutuhan yang dimengerti oleh *developer* dalam hal ini yaitu diagram-diagram yang akan digunakan nantinya. Desain dilakukan untuk membuat desain awal sistem yang akan dikembangkan sesuai dengan kebutuhan yang telah dianalisis pada tahap sebelumnya.

1. *Elaboration* (Perluasan/Perencanaan)

Pada fase *elaboration,* tahap yang dilakukan pada *inception* yaitu *business modeling* dan *requirement* masih dapat dilakukan untuk melengkapi kekurangan dari fase sebelumnya. Selanjutnya akan dilakukan analisis dan perancangan aplikasi, penulis menggunakan alat bantu *(tools)* yaitu *United Modeling Language* (UML) yang meliputi penentuan *actor,* perancangan *use case diagram,* perancangan *use case diagram,* perancangan *sequence diagram,* perancangan tampilan selanjutnya penulis akan melakukan perancangan yang dilakukan meliputi halaman-halaman yang ada di dalam sistem.

1. *Construction* ( Kontruksi )

Pada Tahap ini penginplementasian rancangan *software* yang telah dibuat sudah mencapai tahap 75%, sehingga sudah bisa dibuat *sequence diagram, class diagram* dan *coding.* Pada fase ini juga analisis dan desain masih dapat dilakukan untuk memperbaiki dan menyesuaikan dengan kebutuhan sistem.

1. *Transition* ( Transisi )

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user.* Tahap ini merupakan tahap dimana *software* sudah dapat digunakan atau dipakai. *Test* melakukan uji coba untuk mnghilangkan kesalahan-kesalahan yang mungkin timbul. Uji coba terdiri dari dua jenis, yaitu uji coba proses yang dilakukan secara otomatis *oleh software* dan uji coba antar muka yang dilakukan oleh *tester*.

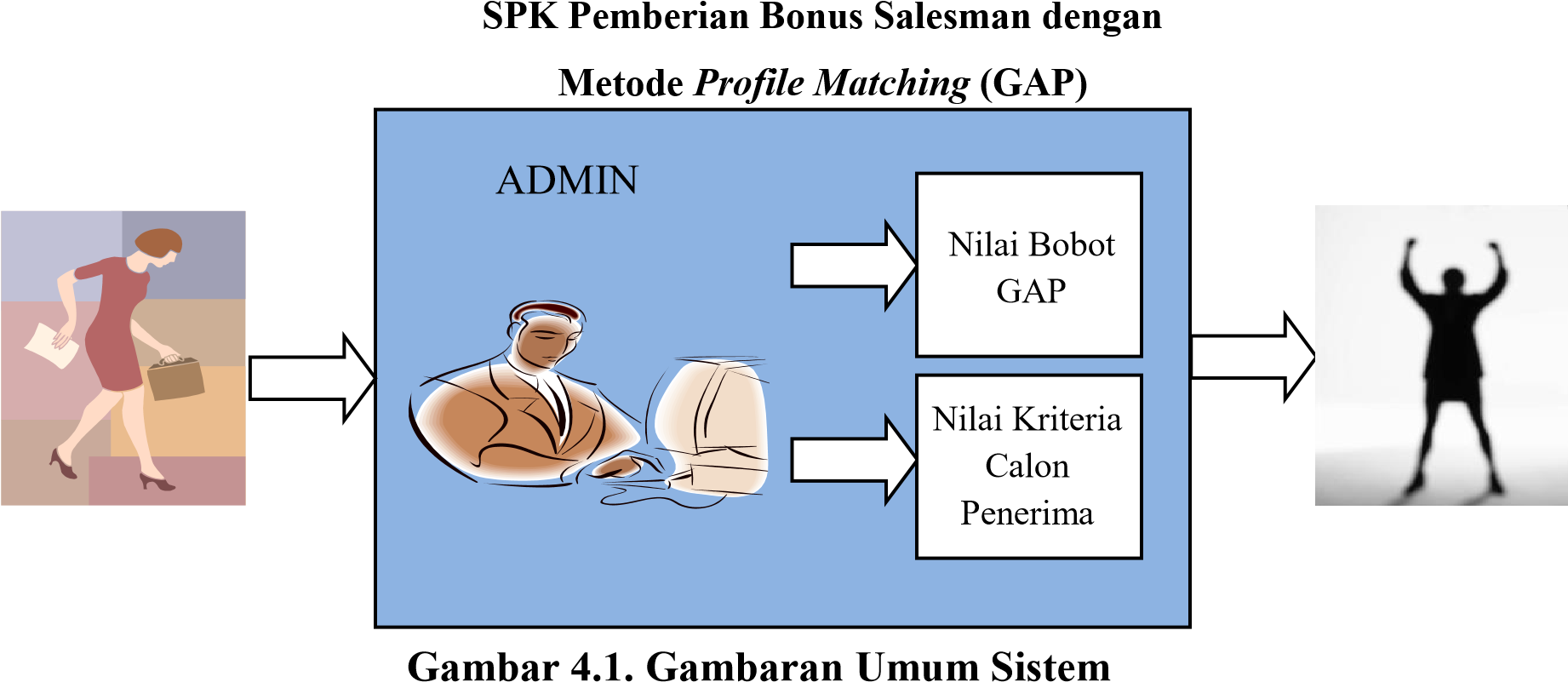
## 3.4 Gambaran Umum Sistem Sebelumnya

Sebelum sistem pendukung keputusan penilaian dan pemberian bonus salesman dibuat, pemberian bonus salesman di PT. Matakar Kendari masih dengan cara manual tanpa adanya bantuan sistem yang terkomputerisasi sehingga pimpinan membutuhkan waktu yang cukup lama untuk menentukan salesman yang menerima bonus di setiap periodenya. Selain itu pimpinan belum mempunyai sistem informasi penilaian kinerja salesman yang dapat berguna untuk memonitor kinerja karyawan dari waktu ke waktu.

## 3.5 Gambaran Umum Sistem Baru

Admin sebelum melakukan proses pemberian bonus salesman, terlebih dahulu memasukkan nilai bobot dan nilai kriteria dari masing-masing calon penerima bonus salesman.

Setelah menginputkan nilai bobot dan nilai kriteria, sistem akan memproses atau menghitung nilai bobot dari masing-masing kriteria. Setelah itu, sistem akan melakukan proses perhitungan alternatif secara otomatis, di mana hasil perhitungan tersebut akan diurutkan dari nilai tertinggi sampai nilai terendah. Adapun gambaran umum sistem dapat dilihat pada Gambar 4.1.



## 3.7 Perancangan Aplikasi

Sistem pendukung keputusan penilaian dan pemberian bonus salesman pada PT Matakar Kendari dengan menggunakan metode *profile matching* (GAP) terdiri dari beberapa tahap yakni perancangan sistem, perancangan *interface* dan perancangan *database* serta perancangan diagram UML.

## 3.8 Analisa Permasalahan

Untuk penilaian dan pemberian bonus salesman pada PT Matakar Kendari dengan menggunakan metode *profile matching* diperlukan kriteria-kriteria, sub kriteria dan bobot untuk melakukan proses perhitungannya. Untuk pemberian nilai bobot tergantung pada pengambil keputusan.

### 3.8.1 Pembobotan Nilai Kriteria *Profile Matching (GAP)*

Dalam metode *Profile Matching* (GAP) terdapat beberapa pembobotan kriteria yang dibutuhkan untuk menentukan penerimaan bonus salesman. Adapun kriteria-kriteriannya adalah sebagai berikut :

# Tabel 4.1 Pembobotan Nilai kriteria *Profil Matching* (GAP)

|  |  |  |  |
| --- | --- | --- | --- |
| No | Kriteria | Pembobotan | |
| Ket | Bobot(W) |
| 1 | Posisi/Jabatan | Sangat Penting Penting  Cukup Penting  Kurang Penting  Tidak Penting | 5  4  3  2  1 |
| 2 | Pencapaian Target | Sangat Penting Penting  Cukup Penting  Kurang Penting  Tidak Penting | 5  4  3  2  1 |

**Lanjutan Tabel 4.1 Pembobotan Nilai kriteria *Profil Matching* (GAP)**

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | Perilaku | Sangat Penting Penting  Cukup Penting  Kurang Penting  Tidak Penting | 5  4  3  2  1 |

## 3.8.2 Pembobotan Nilai Sub Kriteria *Profile Matching* (GAP)

Dalam metode *Profile Matching* (GAP) terdapat beberapa pembobotan sub kriteria dari masing-masin kriteria. Adapun sub kriterianya adalah sebagai berikut:

# Tabel 4.2. Pembobotan Nilai Sub Kriteria *Profile Matching* (GAP)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Kriteria | Sub Kriteria | Pembobotan | |
| Ket | Bobot(W) |
| 1 | Posisi/Jabatan | Kekuasaan  Pengaruh  Kecerdasan | Sangat Penting Penting  Cukup Penting  Kurang Penting  Tidak Penting | 5  4  3  2  1 |
| 2 | Target | Tanggung Jawab  Ketelitian  Pencapaian Target  Perencanaan | Sangat Penting Penting  Cukup Penting  Kurang Penting  Tidak Penting | 5  4  3  2  1 |
| 3 | Perilaku | Disiplin  Kehadiran  Kesabaran  Konsentrasi | Sangat Penting Penting  Cukup Penting  Kurang Penting  Tidak Penting | 5  4  3  2  1 |

## 3.9 Analisis Persyaratan

Analisis persyaratan dibagi menjadi dua bagian yaitu persyaratan fungsional dan persyaratan non-fungsional.

### 3.9.1 Fungsional

Persyaratan Fungsional (*Functional Requirement*) merupakan deskripsi mengenai aktivitas dan layanan yang harus diberikan/disediakan oleh sistem. Persyaratan Fungsional sering dididentifikasikan dalam istilah *input, output,* proses dan data tersimpan yang dibutuhkan untuk memenuhi sasaran peningkatan sistem. Adapun persyaratan fungsional yang harus dimiliki sistem ini yaitu : Sistem harus mampu memberikan solusi terhadap sistem pendukung keputusan dalam merangkingkan calon penerima bonus dengan menggunakan algoritma *profile matching (GAP)*.

### 3.9.2 Non- Fungsional

Persyaratan nonfungsional (*Nonfunctional Requirement*) merupakan deskripsi mengenai fitur, karakteristik, batasan dan lainnya yang menentukan apakah sistem memuaskan atau tidak. Adapun persyaratan nonfungsional yang dimiliki sistem ini yaitu :

1. *User Friendly*

Sistem yang dibangun harus *User Friendly,* artinya sistem ini dapat dengan mudah digunakan dan dimengerti oleh *user* sehingga menjadi salah satu solusi dalam merankingkan calon penerima bonus.

1. *Software* Pendukung

Sistem yang dibangun oleh penulis menggunakan semua *Software* Pendukung bersifat *freeware* sehingga tidak memerlukan izin atau hal-hal yang dapat merugikan pihak lain.

1. Performa

Perankinngan yang dilakukan oleh sistem akan mampu menampilkan calon penerima bonus dengan cara terurut.

## 3.10 Pemodelan Sistem

Pemodelan sistem digunakan untuk merancang bagaimana nantinya sistem akan bekerja. Pemodelan ini menggunakan UML (*Unified Modelling Languange*). Pemodelan sistem yang dirancang penulis bertujuan menggambarkan kondisi dan bagian-bagian yang berperan dalam sistem yang dirancang. Pemodelan sistem dilakukan dengan membuat *use-case diagram, activity diagram* dan *sequence diagram*.

### 3.10.1 *Use-Case Diagram*

*Use Case Diagram* adalah diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Dengan kata lain secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem. Permodelan *use case* mengidentifikasikan dan menggunakan fungsi-fungsi sistem dengan menggunakan alat yang disebut *use case*.

*Use case* menggambarkan bagaimana sistem bekerja dari sudut pandang pengguna bukan dari sudut pandang pembangun sistem. Pada *use case* kita dapat melihat kemampuan sistem secara garis besar. Gambar 4.2 adalah *use case diagram* untuk aplikasi pemberian bonus salesman.

Input Data Salesman Calon Penerima Bonus

Data Nilai Bobot

Data Kriteria

*Profil*

*e*

*Matching (GAP)*

Data Hasil GAP

Administrator

*Logi*

*n*

*Profil*

*e*

*Matching (GAP)*

Data Hasil GAP

Pimpinan

*Logi*

*n*

**Gambar 4.2 *Use Case* Salesman Calon Penerima Bonus**

### 3.10.2 Activity Diagram

*Activity Diagram* adalah sebuah diagram yang dapat digunakan untuk menggambarkan secara grafis aliran proses bisnis, langkah-langkah sebuah *use case* atau logika *behavior* (metode) *object*. Diagram ini berbeda dengan *flowchart* dimana diagram ini menyediakan sebuah mekanisme untuk menggambarkan kegiatan yang tampak secara paralel. Karena itu, diagram ini sangat berguna untuk memodelkan kegiatan yang akan dilakukan saat sebuah operasi dieksekusi dan untuk memodelkan hasil-hasil dari kegiatan-kegiatan ini. *Activity diagram* untuk proses perankingan dengan Algoritma *Profile Matching* dapat dilihat pada gambar

4.3:

*Administrator* *System*

Input Data

kriteria

Hitung GAP

Matriks Ternormalisasi Rating

Kecocokan Alternatif Kriteria

Perengkingan Hasil GAP

Matriks Ternormalisasi Dengan

Nilai Bobot

# Gambar 4.3 *Activity Diagram* Algoritma GAP

## 3.10.3 *Sequence Diagram*

*Sequence Diagram* ialah interaksi menggambarkan interaksi diantara aktor dan sistem dalam skenario *use case.* Tujuan membuat *squencediagram* adalah untuk menggambarkan interaksi antar objek di dalam dan sekitar sistem termasuk pengguna, *display* dan sebagainya yang digambarkan dengan hubungan *massage* dan waktu serta untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

Administrator

Bobot Kriteria

Matriks

GAP

1

()

: Input Kriteria dan Sub Kriteria

2

: Nilai Kriteria Setiap Alternatif

()

3

: Hitung Bobot Kriteria

()

4

()

:

Matriks Normalisasi

5

: Matriks Kriteria

()

6

: Perengkingan Ca

lon Penerima Bonus

## Gambar 4.4 *Sequence Diagram* GAP

Gambar 4.4 menggambarkan *Squence Diagram* yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem dengan menggunakan Metode *profile matching*.

### 3.11 Perancangan Sistem

Setelah dilakukan analisis terhadap sistem, tahap selanjutnya yang dilakukan adalah membuat alur proses atau *flowchart* sistem. *Flowchart* berfungsi untuk memberitahu bagaimana tahap awal mula sistem dijalankan sampai sistem selesai beroperasi. Tahap selanjutnya adalah merancang antarmuka sistem atau *system interface.*

#### 3.11.1 *Flowchart System*

*Start*

Masukan Kriteria

dan Alternatif

Calon

Penerima Bonus

Perhitungan dengan

*Metode Profile*

Matching

Alternatif Penerima

Bonus Salesman

Kriteria dan alternatif

d

ari masing

-

masing

Salesman

Finish

# Gambar 4.5 *Flowchart* Sistem dengan Algoritma *Profile Matching*

Gambar 4.5 menjelaskan tentang alur proses bagaimana sistem berjalan dengan menggunakan Metode *Profile Matching* (GAP).

## 3.11*.2* Perancangan *User Interface*

Rancangan antarmuka pengguna atau *user interface* merupakan penggambaran tampilan yang digunakan secara langsung oleh pengguna. Rancangan *user interface* dibuat sederhana mungkin agar mudah dimengerti dan tidak ada kerumitan dalam mengoperasikannya sehingga dapat mencapai tujuan perangkat lunak yang *user friendly.*

3.11.2.1 Halaman *Login*

Halaman *Login* merupakan halaman yang pertama kali muncul pada saat sistem dijalankan. Halaman *login* ini menerima masukan *userId,* *password.* Kemudian akan dicek oleh sistem valid atau tidaknya masukan tersebut. Jika *userId,* *password* sudah valid, maka *user/Admin* dapat menggunakan aplikasi ini, jika tidak, maka tidak bisa terbuka aplikasi selanjutnya. Tampilan rancangan halaman *login* dapat dilihat pada Gambar 4.6.

|  |  |
| --- | --- |
|  |  |
| |  | | --- | | ***LOGO*** | | *UserID*    :    *Password*    :    *LOGIN*    *CANCEL*    *Level*  : |

# Gambar 4.6 Rancangan Halaman *Login*

3.11.2.2 Halaman Menu Utama

Halaman menu utama terdapat dua menu halaman yaitu halaman untuk

*Administrator* dan halaman untuk *User*

3.11.2.2.1 Menu Utama *Administrator*

Pada halaman menu utama *administrator* ini terdapat menu utama *file* dengan sub menu *setting* nilai bobot, data *user*, *log out*, keluar. Menu utama Master dengan submenu Data salesman, Data Penilaian, Data Pemberian Bonus. Menu utama Proses GAP, menu utama Informasi dengan sub menu info data salesman, info data penilaian salesman, menu utama laporan dengan sub menu laporan hasil penilaian, dan menu utama *About Me.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| File | Master | | | | Proses Gap | | Informasi |  | | Laporan | | About Me |  |
| Setting Nilai Bobot |  |  | Data Salesman | |  | | InfoData Salesman |  | |  | Laporan Hasil Penilaian |  | |
|  |  | |
| Data User |  | Data Penilaian | | | Info Penilaian Salesman |  |  |  |
|  | | | n |  |
| Log Out | Data Pemberia  Bonus | | |
|  |  |
| Keluar Ctrl +X |  |  | |  |  |
|  |  | | |

# Gambar 4.7 Rancangan Menu Utama Administrator

3.11.2.2.2 Menu Utama Pimpinan

Pada halaman menu utama *administrator* ini terdapat menu utama *file* dengan sub menu *setting* nilai bobot, data *user*, *log out*, keluar. Menu utama Proses GAP, menu utama Informasi dengan sub menu info data salesman, info data penilaian salesman, menu utama laporan dengan sub menu laporan hasil penilaian, dan menu utama *About Me.*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| File | Proses Gap | | | | Informasi | |  | Laporan | | About Me |  |
|  |  |  | | |  | InfoData Salesman |  |  | Laporan Hasil Penilaian |  | |
| Setting Nilai Bobot |  |  |  | |
| Info Penilaian Salesman |  |  |
| Log Out |  |  |
| Keluar Ctrl +X |  |  | |  |  |  |
|  |  | | |

# Gambar 4.8 Rancangan Menu Utama Pimpinan

3.11.3 Halaman Input Data Salesman

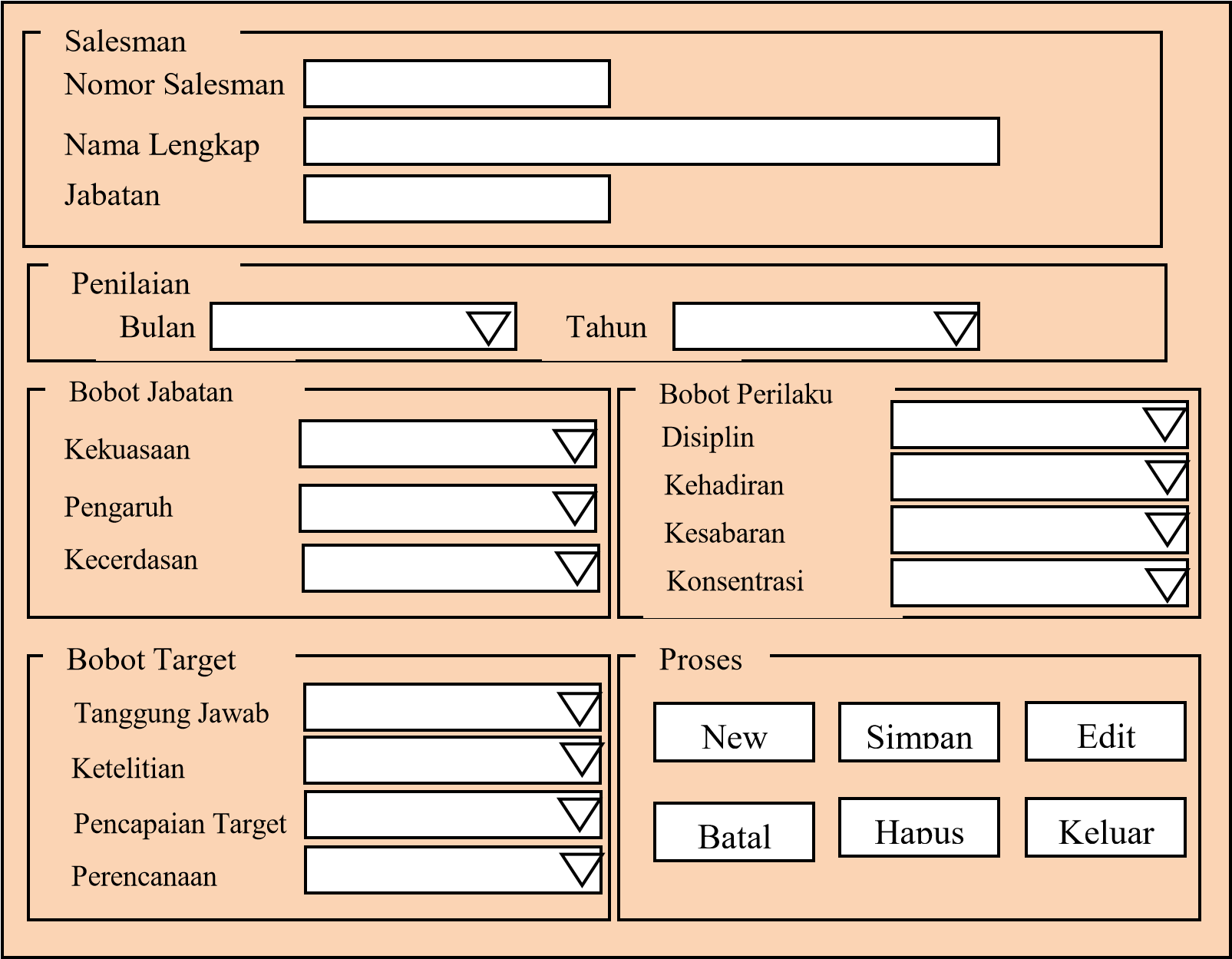
Input data salesman berfungsi untuk memasukkan data-data dari setiap salesman.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  |  |  | | --- | |  |   Nomor Karyawan : Nama Lengkap :   |  | | --- | |  |   TTL :   |  | | --- | |  |   Agama :   |  | | --- | |  |   Jenis Kelamin :   |  | | --- | |  |  |  | | --- | |  |   No. Telp : Alamat :   |  | | --- | |  |     Pend. Terakhir :   |  | | --- | |  |   Jabatan :   |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | New |  | Simpan |  | Edit |  | Batal |  | Hapus |  | Keluar | |   Proses |

# Gambar 4.9 Rancangan Halaman Inpu data salesman

3.11.4 Halaman Input data penilaian

Input data penilaian berfungsi untuk memasukkan data-data penilaian mulai dari bulan dan tahun penilaian selanjutnya menginputkan nilai dari masingmasing bobot kriteria.



# Gambar 4.10 Rancangan Halaman Input Data Penilaian

3.11.5 Halaman Proses Penilaian GAP

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | No | Nik | Jabatan CF | Jabatan SF | Target CF | Tar | |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | | | | | | | | |  | No | Nik | Kekuasaan | Pengaruh | Kecerdasan | Tanggung Jaw |  | |  |  |  |  |  |  | |  | | | | | |   Penilaian Nilai Core dan Secondary Factor  Bulan      Tahun        **Proses**      Data Penilaian Karyawan       |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | | | | | |  |  | | |  |  | Nik | Kekuasaan | Pengaruh | Kecerdasan | No Nik Nama Nilai Jabatan Nilai Target |  | |  |  |  |  |  |  | |  | | | | |  |   Pembobotan Nilai GAP Perengkingan |

**Gambar 4.11 Rancangan Halaman Proses Penilaian GAP**

**Daftar Pustaka**

Alwi Hasan, 2004, *Kamus Besar Bahasa Indonesia.* Jakarta : Departemen Pendidikan Nasional Balai Pustaka.

Darmawan, 2012, Jurnal Sistem Pendukung Keputusan pemilihan beasiswa bagi mahasiswa STMIK Widya Pratama dengan Metode *Profile Matching*. Darmawan : Pekalongan.

Hermawan, J., 2005, *Membangun Decision Suport Sistem*, Jakarta.

Iqbal dan Hartati, 2011, Penempatan Bidan PTT pada Kabupaten Bireuen.

Kusrini. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan.* Yogyakarta:

Andi Offset.

Lucky Priyo Prayogo, 2012, Skripsi Sistem Pendukung Keputusan Pemberian Reward Karyawan Menggunakan Metode *Profile Matching* pada Rumah Makan Pondok Alam. Semarang : Universitas Dian Nuswantoro.

Munawar, 2005, *Pemodelan Visual dengan UML,* Graha Ilmu, Yogyakarta, 17100.

Sherly, 2013, Skripsi Sistem Pendukung Keputusan Pemberian Bonus Karyawan PT. Sanghyang Seri Persero, Yogyakarta.

Subakti, Irfan. 2002. *Sistem Pendukung Keputusan*. Surabaya: Jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember.

Turban, E. dan Aronson, J.E., 1998. *Decision Support Systems and Intelligent Systems* : Prentice-Hall International.

Turban, E., Aronson, J.E. dan Liang, Ting-Peng, 2005, *Decision Support Systems and Intelligent Systems,* Andi : Yogjakarta.

Turban, E., J. E. Aronson, dan T. Liang. (2005:54). *Sistem Pendukung Keputusan dan Sistem Cerdas.* Yogyakarta: Andi Offset.

Turban, Efraim., Ephraim Mc Lean., and James Wetherbe. 1996. *Information Technology for Management*: Improving Quality and Productivity, John Wiley & Son., Inc.