

BAB II

TINJAUAN PUSTAKA

II.1. Definisi Sistem

Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*), sehingga menghasilkan keluaran (*output*) (Kusrini, 2007:11).

II.2. Sistem Informasi

Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan oleh pihak luar.

Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi:

1. Sistem Pemrosesan Transaksi (*Transaction Processing System* atau TPS).
2. Sistem Informasi Manajemen (*Management Information System* atau MIS).
3. Sistem otomasi perkantoran (*Office Automation System/OAS*).
4. Sistem pendukung keputusan (*Decision Support System* atau DSS).
5. Sistem Informasi Eksekutif (*Executive Information System* atau EIS).
6. Sistem pendukung kelompok (*Group Support System* atau GSS)
7. Sistem pendukung cerdas (*Intelligent Support System* atau ISS).

Mengingat bahwa EIS, DSS, dan MIS digunakan untuk mendukung manajemen, maka ketiga sistem tersebut sering disebut sistem pendukung manajemen (*management support system* atau MSS) (kusrini, M.Kom, 2007:15).

II.2.1. Sistem Pemrosesan Transaksi (*Transactional Processing System* atau TPS)

TPS merupakan sistem informasi yang biasanya diimplementasikan pertama kali di suatu organisasi sebelum mengimplementasikan sistem informasi lainnya. TPS memfokuskan pada data transaksi. Sesuai namanya, sistem informasi tersebut digunakan untuk menghimpun, menyimpan, dan memproses data transaksi, dan kadangkala mengendalikan keputusan yang merupakan bagian dari transaksi.

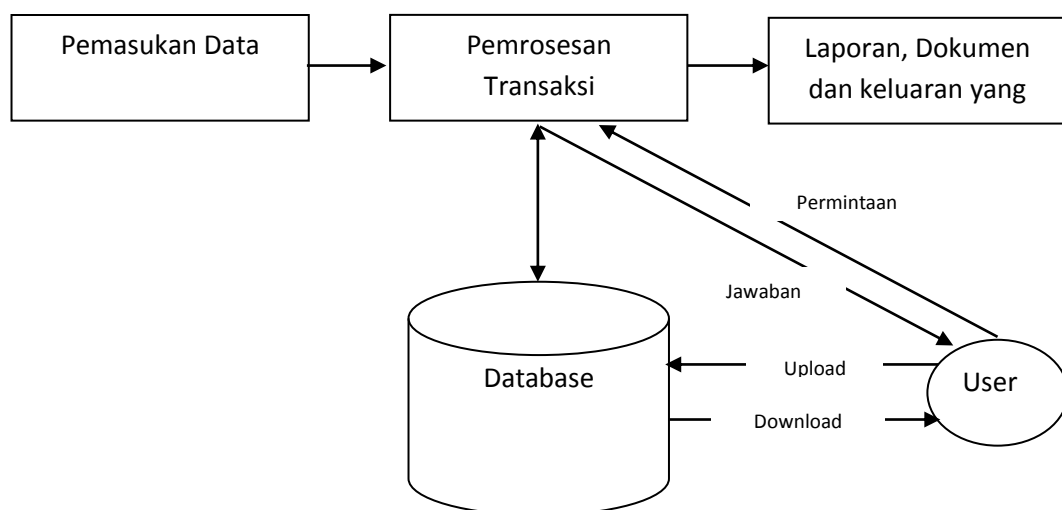
Contoh dari TPS adalah sistem informasi untuk mencatat transaksi penjualan, pemesanan tiket pesawat, dan lain-lain.

Contoh pengendalian keputusan dalam TPS adalah sistem pemrosesan transaksi yang sekaligus bisa memvalidasi keabsahan kartu kredit atau mencari rute pesawat terbang yang terbaik sesuai kebutuhan pelanggan.

Karakteristik sistem informasi yang termasuk TPS adalah sebagai berikut (Turban, McLean, dan Wetherbe, 1999).

1. Jumlah data yang diproses sangat besar.
2. Sumber data umumnya internal dan keluaran dimaksudkan terutama untuk pihak internal (meskipun bisa juga diperuntukkan bagi mitra kerja).

3. Pemrosesan informasi dilakukan secara teratur: harian, mingguan, dan sebagainya.
4. Kapasitas penyimpanan (basis data) besar.
5. Kecepatan pemrosesan yang diperlukan tinggi karena volume yang besar.
6. Umumnya memantau dan mengumpulkan data masa lalu.
7. Masukan dan keluaran terstruktur. Mengingat bahwa data yang diproses cukup stabil, maka data diformat dalam suatu standar.
8. Level kerincian yang tinggi mudah terlihat terutama pada masukan, tetapi seringkali juga pada keluaran.
9. Komputasi tidak rumit (menggunakan matematika sederhana atau operasi statistik)
10. Memerlukan keandalan yang tinggi.
11. Pemrosesan terhadap permintaan merupakan suatu keharusan. Pemakai bisa melakukan permintaan atas basis data.



Gambar II.1. Model Konseptual TPS
(Sumber : Kusriani, M. Kom, 2007)

Dalam hal pemrosesan yang dilakukan, TPS memiliki 2 cara, yaitu:

1. Pemrosesan *batch*

Transaksi ditumpuk dahulu dan kemudian diproses belakangan pada waktu tertentu, misalnya pada waktu sore hari atau malam hari. Kelemahan pemrosesan *batch* adalah pembautan basis data yang tidak pernah dilakukan dalam keadaan terkini, karena seringkali terdapat data transaksi yang terlambat untuk dimasukkan ke dalam basis data.

2. Pemrosesan *online*

Tidak ada penundaan pemrosesan. Setiap transaksi yang sudah terjadi segera dibukukan. Dengan demikian, data selalu ada dalam keadaan mutakhir.

II.2.2. Sistem Informasi Manajemen (SIM)

SIM adalah sistem informasi yang digunakan untuk menyajikan informasi agar mendukung operasi, manajemen, dan pengambilan keputusan dalam sebuah organisasi.

SIM menghasilkan informasi untuk memantau kinerja, memelihara koordinasi, dan menyediakan informasi untuk operasi organisasi. Pada umumnya, SIM mengambil data dari sistem pemrosesan transaksi.

SIM sering disebut sistem peringatan manajemen (*management alerting system*), karena sistem itu memberikan peringatan kepada pemakai (umumnya manajemen) atas munculnya masalah maupun peluang. Selain itu, SIM disebut juga sebagai sistem pelaporan manajemen atau *management reporting system*.

Karakteristik dari sistem informasi yang termasuk SIM adalah:

1. Beroperasi pada tugas-tugas yang terstruktur, yakni pada lingkungan yang telah mendefinisikan hal-hal berikut secara tegas dan jelas: prosedur operasi, aturan pengambilan keputusan, dan arus informasi.
2. Meningkatkan efisiensi dengan mengurangi biaya.
3. Menyediakan laporan dan kemudahan akses yang berguna dalam pengambilan keputusan, tetapi tidak secara langsung (manajer menggunakan laporan dan informasi dan membuat kesimpulan-kesimpulan sendiri untuk mengambil keputusan) (kusrini, M.Kom, 2007:79).

II.2.3. Sistem Pendukung Keputusan/*Decision Support System* (DSS)

Keputusan merupakan kegiatan memilih suatu strategi atau tindakan dalam pemecahan masalah tersebut. Tujuan dari keputusan adalah untuk mencapai target tertentu harus dilakui.

Sistem penunjang keputusan merupakan salah satu produk perangkat lunak yang dikembangkan secara khusus untuk membantu dalam proses pengambilan keputusan. Sesuai dengan namanya tujuan dari sistem ini adalah sebagai “*Information Sources*” atau “*Second Opinion*” yang dapat digunakan sebagai bahan pertimbangan dalam pengambilan keputusan atau kebijakan tertentu, merupakan satu model yang fleksibel yang memungkinkan pribadi-pribadi atau kelompok-kelompok untuk membentuk gagasan-gagasan dan membatasi masalah dengan membuat asumsi mereka sendiri dan menghasilkan pemecahan yang diinginkan. (Arfyanti Ita, 2012:17).

Sistem pendukung keputusan (SPK) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan memanipulasikan data. Sistem ini digunakan untuk pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak dapat terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter, 2002:34).

Sistem penunjang keputusan / *Decision Support System* (DSS) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter, 2002:17).

Tujuan dari DSS adalah (Turban, 2005:10)

1. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil manajer lebih dari pada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktivitas. Membangun satu kelompok pengambil keputusan, terutama para pakar, bisa sangat mahal. Pendukung

terkomputerisasi bisa mengurangi ukuran kelompok dan kemungkinan para anggotanya untuk berada di berbagai lokasi yang berbeda-beda.

6. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat. Sebagai contoh, semakin banyak data yang diakses, makin banyak juga alternatif yang bisa dievaluasi.
7. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan. Tekanan persaingan menyebabkan tugas pengambilan keputusan menjadi sulit. Persaingan didasarkan tidak hanya pada harga, tetapi juga pada kualitas, kecepatan, kustomasi produk, dan dukungan pelanggan.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. Menurut Simon (1977), otak manusia memiliki kemampuan yang terbatas untuk memproses dan menyimpan informasi.

Keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari kesetrukturannya yang bisa dibagi menjadi :

1. *Keputusan terstruktur (structured decision)*

Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Prosedur pengambilan keputusan sangatlah jelas. Keputusan tersebut terutama dilakukan pada manajemen tingkat bawah. Misalnya , keputusan pemesanan produk.

2. *Keputusan semiterstruktur (semistructured decision)*

Keputusan semistruktur adalah keputusan yang memiliki dua sifat. Sebagian keputusan bisa ditangani oleh komputer dan yang lain tetap harus dilakukan

oleh pengambil keputusan. Produser dalam pengambilan keputusan tersebut secara garis besar sudah ada, tetapi ada beberapa hal yang masih memerlukan kebijakan dari pengambil keputusan. Biasanya, keputusan semacam ini diambil oleh manajer level menengah dalam suatu organisasi. Contoh keputusan jenis ini adalah pengevaluasian kredit, penjadwalan produksi, dan pengendalian sediaan.

3. Keputusan tak terstruktur (*unstructured decision*)

Keputusan tak terstruktur adalah keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Keputusan tersebut menuntut pengalaman dan berbagai sumber yang bersifat eksternal.

II.2.4.1 Arsitektur Sistem Pendukung keputusan

Aplikasi sistem pendukung keputusan bisa terdiri dari beberapa subsistem, yaitu:

1. Subsistem manajemen data

Subsistem manajemen data memasukkan satu database yang berisi data yang relevan untuk suatu situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen database (DBMS/Data Base Management System). Subsistem manajemen data bisa diinterkoneksi dengan data warehouse perusahaan, suatu repositori untuk data perusahaan yang relevan dengan pengambilan keputusan.

2. Subsistem manajemen model

Merupakan paket perangkat lunak yang memasukkan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lain yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat. Bahasa-bahasa pemodelan untuk membangun model-model kustom juga dimasukkan. Perangkat lunak itu sering disebut sistem manajemen basis model (MBMS). Komponen tersebut bisa dikoneksikan ke penyimpanan korporat atau eksternal yang ada pada model.

3. Subsistem antarmuka pengguna

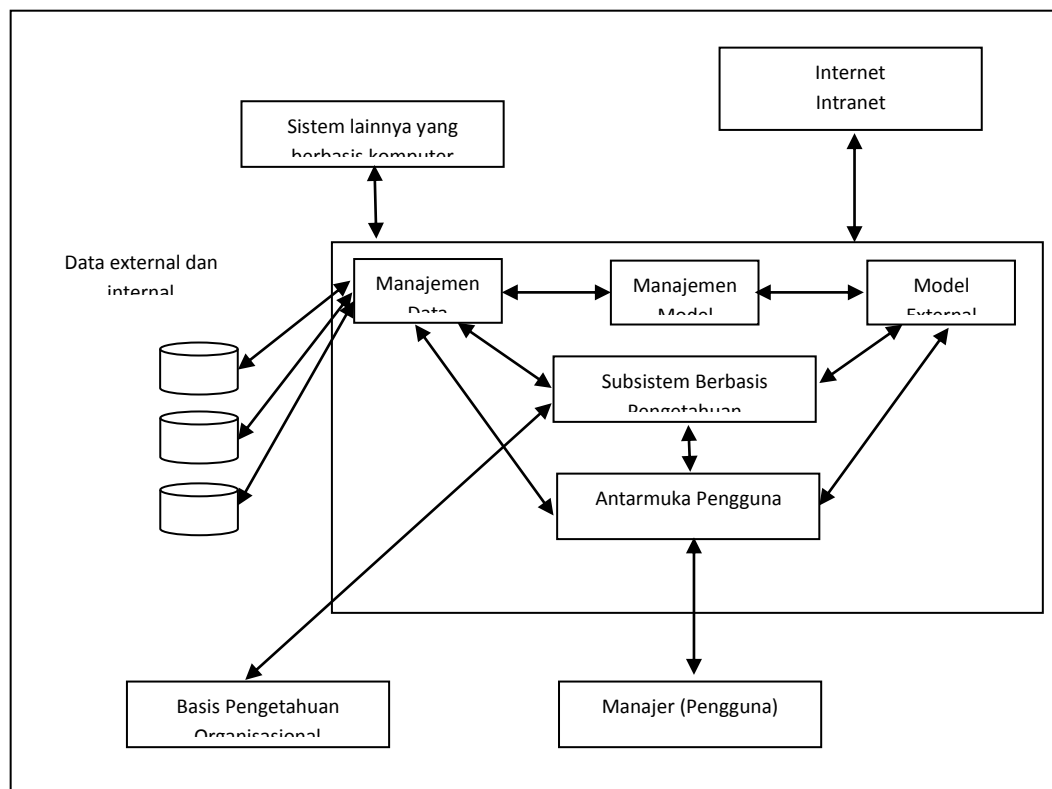
Pengguna berkomunikasi dengan memerintahkan sistem pendukung keputusan melalui subsistem tersebut. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti mengaskan bahwa beberapa kontribusi unik dari sistem pendukung keputusan berasal dari interaksi yang intensif antar komputer dan pembuat keputusan.

4. Subsistem manajemen berbasis pengetahuan

Subsistem tersebut mendukung semua subsistem lain atau bertindak langsung sebagai suatu komponen independen dan bersifat opsional.

Selain memberikan intelegensi untuk memperbesar pengetahuan si pengambil keputusan, subsistem tersebut bisa diinterkoneksikan dengan repositori pengetahuan perusahaan (bagian dari sistem manajemen pengetahuan), yang kadang-kadang disebut basis pengetahuan organisasional.

Berdasarkan definisi, sistem pendukung keputusan harus mencakup tiga komponen utama dari DBMS, MBMS, dan antarmuka pengguna. Subsistem manajemen berbasis pengetahuan adalah opsional, tetapi bisa memberikan banyak manfaat, karena memberikan intelegensi bagi ketiga komponen utama tersebut. Seperti pada semua sistem informasi manajemen, pengguna bisa dianggap sebagai komponen sistem pendukung keputusan. Komponen-komponen tersebut membentuk sistem aplikasi sistem pendukung keputusan yang bisa dikoneksikan ke intranet perusahaan, ekstranet, atau internet. Arsitektur dari sistem pendukung keputusan ditunjukkan dalam gambar II.2. berikut.



Gambar II.2. Arsitektur DSS
(Sumber, Kusrini, M. Kom, 2007)

II.3. Multifactor Evaluation Process (MFEP)

Metode *Multifactor Evaluation Process (MFEP)* adalah kuantitatif yang menggunakan. 'weighting system'¹. Dalam pengambilan keputusan multifaktor, pengambil keputusan secara subyektif dan intuitif menimbang berbagai faktor yang mempunyai pengaruh penting terhadap alternatif pilihan mereka. Untuk keputusan yang berpengaruh secara strategis, lebih dianjurkan menggunakan sebuah pendekatan kuantitatif seperti MFEP. Dalam MFEP pertama-tama seluruh kriteria yang menjadi faktor penting dalam melakukan pertimbangan diberikan pembobotan (weighting) yang sesuai. Langkah yang sama juga dilakukan terhadap alternatif-alternatif yang akan dipilih, yang kemudian dapat dievaluasi berkaitan dengan faktor-faktor pertimbangan tersebut. Metode MFEP menentukan bahwa alternatif dengan nilai tertinggi adalah solusi terbaik berdasarkan kriteria yang telah dipilih. (Ahmad Khaidir;4:2014)

II.3.1. Langkah Penyelesaian Multifactor Evaluation Process (MFEP) sebagai berikut :

1. Menentukan faktor dan bobot faktor dimana total pembobotan harus sama dengan 1.
2. Mengisikan nilai untuk setiap faktor yang mempengaruhi dalam pengambilan keputusan dari data-data yang akan diproses., nilai yang di masukkan merupakan nilai objektif, yaitu faktor evaluation antara 0-1.

3. Proses perhitungan weight evaluation yang merupakan proses perhitungan bobot antara factor weight dan faktor evaluation serta penjumlahan seluruh hasil weight evaluation untuk memperoleh total hasil evaluasi.

Rumus metode ini adalah :

$$TWE = \sum FW \times FE \quad (1)$$

Keterangan:

TWE = *Total Weight Evaluation* (perkalian bobot faktor dan faktor evaluasi)

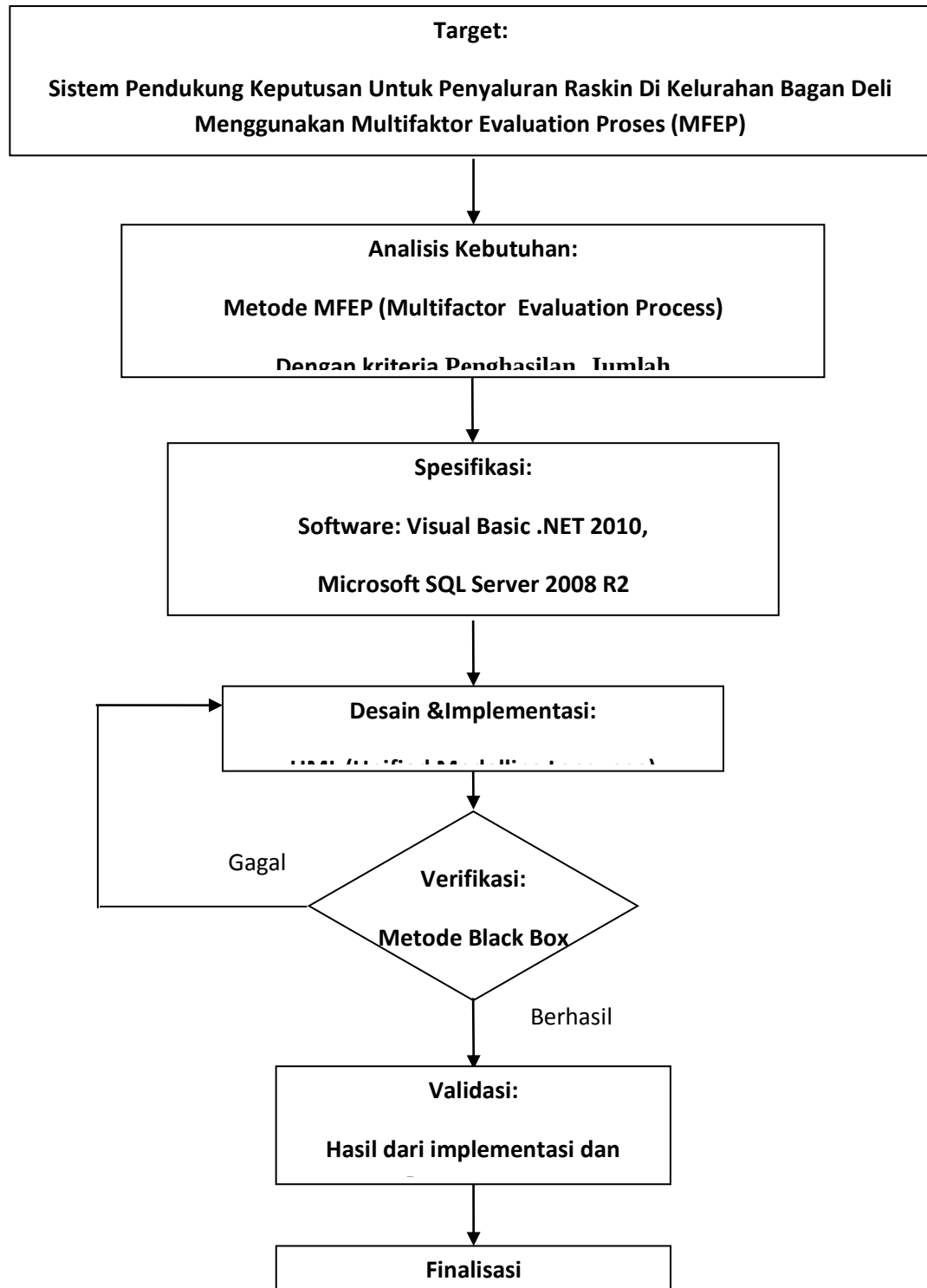
FW = *Factor Weight* (bobot faktor)

FE = *Factor Evaluation* (faktor evaluasi)

dimana:

Bobot faktor merupakan penentuan bobot dari tiap kriteria yang pembobotan berdasarkan kriteria utama atau hanya kriteria pendukung. Sedangkan faktor evaluasi merupakan pembobotan himpunan dari tiap kriteria. Total dari perkalian antara faktor evaluasi dan bobot faktor dijumlahkan dan akan menghasilkan nilai dari alternatif

II.4. Metode Pengembangan Sistem



Gambar II.3. Prosedur Perancangan

II.4.1. Analisis Sistem

Analisis sistem adalah penguraian dari suatu sistem yang utuh kedalam kegiatan-kegiatan komponennya, dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan, yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Langkah-langkah yang dilakukan dalam proses analisis adalah:

1. Identifikasi Masalah

Mengenal masalah merupakan langkah pertama yang dilakukan dalam analisis sistem. Masalah dapat didefinisikan sebagai suatu pertanyaan yang harus dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai. Oleh karena itu, pada tahap analisis sistem, langkah pertama yang harus dilakukan oleh analisis adalah mengidentifikasikan terlebih dahulu masalah-masalah yang terjadi. Adapun permasalahan yang ada pada PT. Nitori Furniture Indonesia adalah sulitnya menentukan kualitas kayu yang akan dijadikan produk furniture. Masalah ini dikarenakan belum adanya sistem untuk penentuan kualitas kayu.

2. Analisis Kelayakan

Studi kelayakan adalah suatu studi yang akan digunakan untuk menentukan apakah pengembangan sistem layak dipakai atau tidak.

Ada empat macam kelayakan yang dapat dipertimbangkan, yaitu:

1. Analisis Kelayakan Teknis

Kelayakan teknis, yaitu dengan menganalisis ketersediaan perangkat keras, perangkat lunak dan instansi untuk melaksanakan proses yang diperlukan.

Perangkat keras yang digunakan untuk membangun sistem yaitu satu unit komputer yang terdiri dari : Intel Core 2 duo 1,5 Ghz, Memory 2 GB, Harddisk 120 GB, Keyboard/Mouse. Perangkat lunak yang digunakan untuk membangun sebuah sistem , yaitu Windows 7, Visual Basic .NET 2010, MySQL Server 5.6, dan Crystal Report 13. Dalam tahap implementasi sudah tersedia tenaga ahli yang dapat mengoperasikan sistem ini.

2. Analisis Kelayakan Operasi

Kelayakan operasi, yaitu menganalisis apakah sistem dapat diimplementasikan. Dalam hal ini sumber daya manusia yang dapat akan mengoperasikan.

3. Analisis Kelayakan Hukum

Kelayakan hukum, yaitu penerapan sistem yang baru tidak menimbulkan masalah karena menyimpang secara hukum.

4. Analisis Kelayakan Jadwal

Kelayakan jadwal digunakan untuk menentukan bahwa pengembangan sistem dapat dilakukan dalam batas waktu yang telah ditetapkan.

VII.II. Desain Sistem

Desain Sistem adalah persiapan rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk yang berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi, menyangkut di dalamnya konfigurasi komponen perangkat lunak dan perangkat keras dari suatu sistem.

1. Desain Model

Desain Model yang digunakan adalah model logika yang mana model logika lebih menjelaskan kepada user bagaimana nantinya fungsi-fungsi sistem informasi secara logika akan bekerja. Logika model digambarkan dengan UML.

2. Desain Database

Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lain, tersimpan disimpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya.

3. Desain Antarmuka (*User Interface*)

Desain antarmuka adalah tahap yang harus dilakukan sebelum mulai membuat suatu aplikasi. Konsep rancangan dalam mendesain control dalam tiap form bertujuan agar aplikasi yang dibuat dapat dengan mudah diakses oleh pengguna.

4. Desain Output

Output adalah keluaran yang dihasilkan oleh sistem informasi baik berupa media keras seperti kertas atau hasil di media lunak seperti tampilan dilayar.

5. Desain Input

Desain input ditentukan dari UML sistem baru yang telah dibuat, selanjutnya ditentukan parameternya, seperti: bentuk dari input, dokumen dasar atau bentuk isian di dialog layar.

6. Desain Kontrol

Desain kontrol bertujuan untuk mencegah atau menjaga terjadinya hal-hal yang tidak diinginkan seperti kesalahan-kesalahan dan kecurangan-kecurangan untuk melindungi sistem informasi dari hal-hal yang merugikan.

Pengendalian dalam sistem informasi yang dilakukan adalah pengendalian aplikasi, yaitu pengendalian yang diterapkan selama proses pengolahan data berlangsung yang meliputi pengendalian masukan, pengolahan dan keluaran.

VII.III. Penulisan Program atau Pengkodean

Menerjemahkan hasil proses perancangan menjadi sebuah bentuk aplikasi komputer yang dimengerti oleh mesin komputer.

VII.IV. Ujicoba Program

Ujicoba software merupakan elemen yang kritis dari SQA(Software Quality Assurance) dan mempresentasikan tinjauan ulang yang menyeluruh terhadap spesifikasi, desain dan pengkodean. Ujicoba mempresentasikan ketidaknormalan yang terjadi pada pengembangan aplikasi. Untuk itu sistem pengujian aplikasi ini akan menggunakan metode *Black Box*.

II.4.5. Implementasi Sistem (implementasi)

Tahap implementasi adalah tahap dimana semua elemen dan aktivitas sistem disatukan dengan langkah-langkah sebagai berikut:

1. Menyiapkan Fasilitas Fisik

Fasilitas-fasilitas fisik yang disiapkan antara lain komputer dan peripheralnya, termasuk keamanan fisik untuk menjaga berlangsungnya peralatan dalam jangka waktu yang lama pada penentuan kualitas kayu pada PT. Nitori Furniture Indonesia dengan metode *Multifactor Evaluation Process* (MFEP).

2. Menyiapkan Pemakai (User)

Pemakai disiapkan dengan terlebih dahulu yaitu dengan memberikan pelatihan secara prosedural maupun tutorial mengenai sistem informasi sesuai fungsi tugasnya. Tujuannya adalah agar para pemakai mengerti dan menguasai operasi sistem dan cara kerja sistem serta apa saja yang diperoleh dari penentuan kualitas kayu pada PT. Nitori Furniture Indonesia dengan metode *Multifactor Evaluation Process* (MFEP).

3. Melakukan Simulasi

Kegiatan simulasi berupa pengujian sistem secara nyata yang melibatkan personil yang sesungguhnya.

II.4.6. Pemeliharaan Sistem (*Maintenance*)

Definisi Pemeliharaan Sistem adalah suatu kombinasi dari berbagai tindakan yang dilakukan untuk menjaga suatu sistem dalam, atau memperbaikinya sampai, suatu kondisi yang bisa diterima. Pada bulan April 1970 didefinisikan sebuah istilah untuk Teknologi Pemeliharaan yang mencakup pengertian yang lebih luas dari pada pengertian Pemeliharaan diatas. Istilah ini adalah Teroteknologi.

II.5. Basis Data

Basis data adalah kumpulan data yang saling terikat yang diorganisasi untuk memenuhi kebutuhan dan struktur sebuah organisasi serta bisa digunakan oleh lebih dari satu orang dan lebih dari satu aplikasi.

Basis data adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna. (Shahida Shalole, 2013).

Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali. Bahasa basis data

(*database language*) adalah suatu cara untuk berinteraksi atau berkomunikasi antara pemakai dengan basis data yang diatur dalam bahasa khusus yang ditetapkan oleh perusahaan.

II.5.1. Karakteristik Basis Data

Dalam mempelajari sistem basis data, perlu diketahui beberapa karakteristik penting dari basis data, antara lain adalah:

1. Persistent

Persistent berarti bahwa data ditempatkan pada penyimpanan yang stabil seperti magnetic disk, tetapi tidak berarti bahwa data disimpan selamanya. Data yang tidak lagi relevan atau tidak berguna akan dihilangkan.

2. Shared

Shared berarti bahwa basis data dapat memiliki banyak kegunaan dan pengguna. Banyak pengguna dapat menggunakan basis data pada saat yang bersamaan.

3. Interrelated

Interrelated berarti bahwa data yang disimpan sebagai unit yang berbeda dapat dihubungkan untuk menyediakan gambar keseluruhan. (Yanti Efendy : 2012).

II.5.2. Database

Database merupakan suatu tempat untuk menampung data *server*. Dimana data tersebut nantinya akan diproses oleh program oleh *Perogrammer* dalam tahap pembuatan program. (M. Leo Agung : 2011).

Dalam suatu database terdapat 3 bagian bahasa pemrograman database, yaitu;

1. *Data Definition Language* (DDL),
2. *Data Manipulation Language* (DML), dan
3. *Data Control Language* (DCL). (Ivan Arifard Watung, Alicia A.E. Sinsuw, 2014).

II.5.2.1. *Data Definition Language* (DDL)

DDL merupakan singkatan dari *Data Definition Language* yang juga bagian dari *structured query language* (SQL). DDL berfungsi lebih ke dalam memanipulasi struktur dari database. DDL digunakan untuk membuat tabel atau menghapus tabel, membuat *key* atau indeks, membuat relasi Antartabel. (Ivan Arifard Watung, Alicia A.E. Sinsuw, ST.MT, Sary D.E. Paturusi, ST ,M.Eng et,al : 2014).

Data Definition Language (DDL) digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, *user*, dan sebagainya. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data. Secara umum DDL yang digunakan adalah:

1. *CREATE* untuk membuat objek baru.
2. *USE* untuk menggunakan objek.
3. *ALTER* untuk mengubah objek yang sudah ada.
4. *DROP* untuk menghapus objek. (Adelia& Jimmy Setiawan : 2011 ; 115).

II.5.2.2.Data Manipulation Language (DML)

Data Manipulation Language (DML) merupakan sekumpulan sintaks-sintaks bisa digunakan untuk melakukan proses *insert*, *update* atau *delete* ke dalam suatu database. (Ivan Arifard Watung&Alicia A.E. Sinsuw, 2014).

Data Manipulation Language (DML)digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah-perintah yang umum dilakukan adalah:

1. *SELECT* untuk menampilkan data.
2. *INSERT* untuk menambahkan data baru.
3. *UPDATE* untuk mengubah data yang sudah ada.
4. *DELETE* untuk menghapus data. (Adelia& Jimmy Setiawan : 2011 ; 115).

II.5.2.3.Data Control Language (DCL)

Data Control Language (DCL) merupakan kelompok perintah yang berisi untuk mengendalikan pangaksesan data DCL digunakan untuk menangani masalah keamanan dalam database *server*. (Ivan Arifard Watung, 2014).

Data Control Language (DCL) digunakan untuk menangani masalah keamanan dalam database server. (Ivan Arifard Watung& Alicia A.E. Sinsuw, : 2014)

II.6.Normalisasi

Normalisasi adalah suatu teknik untuk mengorganisasikan data ke dalam table-tabel untuk memenuhi kebutuhan pemakai di dalam suatu organisasi. Tujuan dari normalisasi adalah sebagai berikut:

1. Untuk menghilangkan kerangkapan data.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

Adapun proses normalisasi adalah sebagai berikut:

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Tahapan Normalisasi:

Bentuk Tidak Normal



Menghilangkan perulangan group

Bentuk Normal Pertama (1NF)



Menghilangkan ketergantungan sebagian

Bentuk Normal Kedua (2NF)



Menghilangkan ketergantungan transitif

Bentuk Normal Ketiga (3NF)



Menghilangkan anomali-anomali hasil dari ketergantungan fungsional

Bentuk Normal Boyce-Codd (BCNF)



Menghilangkan ketergantungan Multivalue

Bentuk Normal Keempat (4NF)



Menghilangkan anomali-anomali yang tersisa

Bentuk Normal Kelima

II.7. *Microsoft SQL Server 2008*

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah *Transact-SQL* yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh *Microsoft* dan *Sybase*. Umumnya *SQL Server* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar.

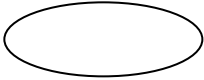
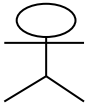
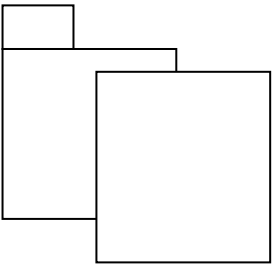
II.8. Visual Basic .NET 2010

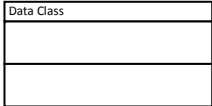
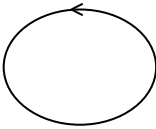
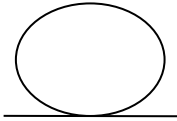
Microsoft Visual Basic .NET 2010 merupakan program aplikasi untuk membentuk program dari bahasa pemrograman klasik, legendaris dan tiada duanya yang paling banyak dipakai oleh jutaan *programmer* dan tercatat sebagai program yang paling disukai oleh mayoritas orang. (Edy Winarno ST & M. Eng. Ali Zaki, 2010).

Visual Basic .NET 2010 terdapat di dalam paket *software Microsoft Visual Studio .NET 2010* yang dirilis pada bulan April 2010, didalam paket tersebut terdapat software lainnya seperti Visual C#, Visual C++, Visual F#, dan *Web Development* (ASP.Net), (XML Web Service), dan Visual Basic tentunya. (Muhammad Sadeli : 2011 ; 2).

II.9. Unified Modelling Language (UML)

Unified Modelling Language (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. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET. (Prastuti Sulistyorini, 2009: 23).

No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Usecase</i>	Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama Use Case. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
2.		<i>Actor</i>	Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki control terhadap Use Case. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
3.		<i>Package</i>	Package antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya

			mengidikasikan aliran data.(Gellysa Urva, Helmi Fauzi Siregar : 2015).
4.		<i>Class</i>	<i>Class</i> merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.(Gellysa Urva, Helmi Fauzi Siregar : 2015).
5.		<i>Control</i>	Control class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.(Gellysa Urva, Helmi Fauzi Siregar : 2015).
6.		<i>Entity</i>	Entity Class, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
7.		<i>Activity</i>	Activites, menggambarkan suatu proses/kegiatan bisnis. (Gellysa Urva, Helmi Fauzi Siregar : 2015).

8.		State	State, pembagian activity diagram untuk menunjukkan siapa melakukan apa. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
----	---	-------	--

Tabel II.1. Daftar Simbol-simbol dalam *Unified Modelling Language* (UML)
(Sumber : Gita Larasati Sumaja : 2013)

UML menyediakan 3 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya, yaitu:

1. *Usecase Diagram* adalah diagram yang harus di buat pertama kali saat pemodelan perangkat lunak berorientasi objek dilakukan. *Usecase* akan menggambarkan apa yang dikerjakan aktor. Yang disebut aktor disini adalah pengguna aplikasi sama seperti pembangunan perangkat lunak terstruktur saat membuat DFD, untuk menggambar diagram use case mengacu pada proses sebelumnya yaitu analisi kebutuhan pada RPL.
2. *Class Diagram* dibuat setelah diagram *usecase* dibuat terlebih dahulu. Diagram ini harus menjelaskan hubungan apa saja yang terjadi antara suatu objek dengan objek lainnya sehingga terbentuk suatu sistem aplikasi. Pembuatan diagram kelas dibagi menjadi 2 bagian, yaitu kelas itu sendiri dan relasi antar-kelas.
3. *Activity Diagram* adalah Representasi secara grafis dari proses dan control flow dan berfungsi untuk memperlihatkan alur dari satu aktivitas ke aktivitas yang lain serta menggambarkan perilaku yang kompleks.

4. *Sequence Diagram* adalah *Sequence* diagram yang dibuat untuk mengetahui alur dari interaksi antar-objek. Isi dari diagram sequence harus sama dengan use case dan diagram kelas. Satu use case tunggal akan digambarkan satu diagram *sequence*-nya.
5. *Diagram konteks* digunakan untuk mengetahui ruang lingkup dan batasan-batasan yang ada dalam perangkat lunak yang sedang dikerjakan. Hal ini didapat pada saat analisi kebutuhan sistem. Batasan yang dimaksudkan adalah mengenai apa saja yang dikerjakan oleh perangkat lunak, siapa yang menggunakan, serta apa yang menjadi *input* dan *output*-nya.