**BAB II**

**LANDASAN TEORI**

# **2.1 Sistem Pendukung Keputusan**

## 2.1.1 Pengertian Sistem Pendukung Keputusan

*Decision Support System* atau Sistem Pendukung Keputusan yang selanjutnya kita singkat dalam skripsi ini menjadi SPK, secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan

baik kemampuan pemecahan masalah maupun kemampuan

pemgkomunikasian 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 (Berawan, 2005).

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

## 2.1.2 Konsep Dasar Sistem Pendukung Keputusan

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

## 2.1.3 Konsep Pengambilan Keputusan

Beberapa definisi keputusan yang dikemukakan para ahli dijalankan sebagai berikut (Hiasan, 2004) : Menurut Ralph C. Navis

1. 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 dukungannya dengan perencanaan. Keputusan dapat pula berupa tindakan terhadap pelaksanaan yang sangat menyimpang dari rencana semula.
2. Menurut James A.F.Stoner

Keputusan adalah pemilihan di antara alternatif-alternatif. Definisi ini mengandung tiga pengertian, yaitu:

* 1. Ada pilihan atas dasar logika atau pertimbangan.
  2. Ada beberapa alternatif yang harus dan dipilih salah satu yang terbaik.
  3. Ada tujuan yang ingin dicapai, dan keputusan itu makin

mendekatkan pada tujuan tertentu.

1. Menurut Prof. Dr. Prajurit Atmosudirjo, SH

Keputusan adalah suatu penaksiran daripada proses pemikiran tentang suatu masalah atau problem untuk menjawab pertanyaan apa yang harus diperluas guna mengatasi masalah tersebut, dengan menjatuhkan pilihan pada suatu alternatif

Beberapa definisi pengambilan keputusan yang dikemukakan para ahli dijalankan sebagai berikut (Hiasan, 2004) :

1. Menurut Georgia R. Terry

Pengambilan keputusan adalah pemilihan alternatif perilaku (kelakuan) tertentu dari dua atau lebih alternatif yang ada.

1. Menurut S.P. Sebagian

Pengambilan keputusan adalah suatu pendekatan yang sistematis terhadap hakikat alternatif yang dihadapi dan mengambil tindakan yang menurut perhitungan merupakan tindakan yang paling tepat.

1. Menurut James A.F. Stoner

Pengambilan keputusan adalah proses yang digunakan untuk memilih suatu tindakan sebagai cara pemecahan masalah.

Dari pengertian-pengertian pengambilan keputusan di atas, dapat disimpulkan bahwa pengambilan keputusan merupakan suatu proses

pemilihan alternatif terbaik dari beberapa alternatif secara sistematis untuk ditindaklanjuti (digunakan) sebagai suatu cara pemecahan masalah.

## 2.1.4 Fase-fase Proses Pengambilan Keputusan

Proses pengambilan keputusan melibatkan tiga tahap utama yaitu Tahap Intelegensi *(Intelligence Phase),* Tahap Perencanaan *(Design Phase)*, Tahap Pilihan *(choice phase)* dan Tahap Implementasi *(Implementation).* Seperti tampak pada gambar tahap-tahap dalam proses pengambilan keputusan berikut : (Turban : 2005)

**Intelligence Khas**

**e**



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** |
|  | Penarikan Keputusan |

**Gambar 2.1 Pengambilan Keputusan / Proses Pemodelan SPK**

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

mengidentifikasi solusi-solusi alternatif dan demikian sebaliknya.

Selanjutnya adalah fase pilihan yang meliputi pilihan terhadap solusi yang diurutkan untuk model (tidak memerlukan masalah yang disajikan). Solusi ni diuji untuk menentukan viabilitasnya. Begitu solusi yang diurutkan tampak masuk akal, maka kita siap untuk masuk kepada fase terakhir yakni fase implementasi keputusan. Hasil implementasi yang berhasil adalah dapat dipecahkannya masalah riel. Sedangkan kegagalan implementasi

mengharuskan kita kembali ke fase sebelumnya.

### **1.1.4.1 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 :

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 perbedaan antara apa yang kita diinginkan (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 diajarkan pada data riel. Menentukan apakah masalah benar-benar ada, di mana masalah tersebut, dan seberapa signifikan, dapat dilakukan setelah investasi awal selesai dilakukan. Poin kunci adalah apakah sistem informasi melaporkan masalah atau hanya melaporkan gejala-gejala dari sebuah masalah.

1. 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.

1. 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 identifikasi sebagai masalah orang lain.

### **2.1.4.2 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

1. Memilih Sebuah Prinsip Pilihan

Prinsip pilihan adalah sebuah kriteria yang menggambarkan akseptabilitas dari sebuah solusi (kemampuan untuk data diderita). 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 kita dan bagaimana sasaran tersebut diajukan ke dalam suatu model.

1. Mengembangkan Alternatif-alternatif

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

kriteria.

1. Mengukur Hasil Akhir

Nilai dari sebuah alternatif devaluasi 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 kerjanya 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.

### **2.1.4.3 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 dank arena 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 merepresentasikan oleh model. Solusi untuk model menghasilkan sebuah solusi yang direkomendasikan untuk masalah. Masalah dianggap diserahkan 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-demi-langkah), heuristik (aturan utama), dan *blind search* (menembak di dalam gelap, idealnya dalam suatu cara yang logis).

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

### **2.1.4.4 Implementation phase**

Pada akibatnya implementasi suatu solusi yang diurutkan 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. Pendek kata, implementasi berarti membuat suatu solusi yang direkomendasikan bisa bekerja, tidak memerlukan implementasi suatu sistem komputer.

## 2.1.5 Karakteristik dan Kemampuan Sistem Pendukung Keputusan

Menurut Turban (1996), ada beberapa karakteristik dari SPK, diantaranya 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 eksternal maupun internal

# 6. Memiliki kemampuan what-if analysis dan goal seeking analysis

7. Menggunakan beberapa model kuantitatif

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

1. Menunjang pembuatan keputusan manajemen dalam menangani

masalah semi terstruktur dan tidak terstruktur.

1. Membantu manajer pada berbagai tingkatan manajemen, mulai dari manajemen tingkat atas sampai manajemen tingkat bawah.
2. Menunjang pembuatan keputusan secara kelompok dan perorangan.
3. Menunjang pembuatan keputusan yang saling kebergantungan dan

berurutan.

1. Menunjang tahap-tahap pembuatan keputusan antara lain *intelligence, design, choice* dan *implementation.*
2. Menunjang berbagai bentuk proses pembuatan keputusan dan jenis keputusan.
3. Kemampuan untuk melakukan adaptasi setiap saat dan bersifat

fleksibel.

1. Kemudahan melakukan interaksi sistem.
2. Meningkatkan efektivitas dalam pembuatan keputusan daripada

efisiensi.

1. Mudah dikembangkan oleh pemakai akhir.
2. Kemampuan pemodelan dan analisis dalam pembuatan keputusan.
3. Kemudahan melakukan pengaksesan berbagai sumber dan format data.

Di samping berbagai kemampuan dan karakteristik seperti dikemukakan di atas, sistem pendukung keputusan memiliki juga

keterbatasan, antara lain:

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat memodelkan, 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.

1. Proses-proses yang dapat dilakukan oleh sistem pendukung keputusan biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.
2. Sistem pendukung keputusan tidak memiliki intuisi seperti yang memiliki oleh manusia. Karena sistem pendukung keputusan hanya suatu kumpulan perangkat keras, perangkat lunak dan sistem operasi yang tidak melengkapi oleh kemampuan berpikir
3. Secara implisit, sistem pendukung keputusan berlandaskan pada kemampuan dari sebuah sistem berbasis komputer dan dapat melayani penyelesaian masalah.

## 2.1.6 Keuntungan Sistem Pendukung Keputusan

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

1. Mampu mendukung pencarian solusi dari berbagai permasalahan yang kompleks.
2. Dapat merespons dengan cepat pada situasi yang tidak diharapkan dalam kondisi yang berubah-ubah.
3. Mampu untuk menerapkan berbagai strategi yang berbeda pada

konfigurasi berbeda secara cepat dan tepat.

1. Pandangan dan pembelajaran baru.
2. Sebagai fasilitator dalam komunikasi.
3. Meningkatkan kontrol manajemen dan kinerja.
4. Menghemat biaya dan sumber daya manusia (SDM).
5. Menghemat waktu karena keputusan dapat diambil dengan cepat.
6. Meningkatkan efektivitas manajerial, menjadikan manajer dapat bekerja lebih singkat dan dengan sedikit usaha.
7. Meningkatkan produktivitas analisis.

**2.1.7 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 donatur oleh *software* yang disebut *Database Management System* (DBMS).

# 2. Model Management

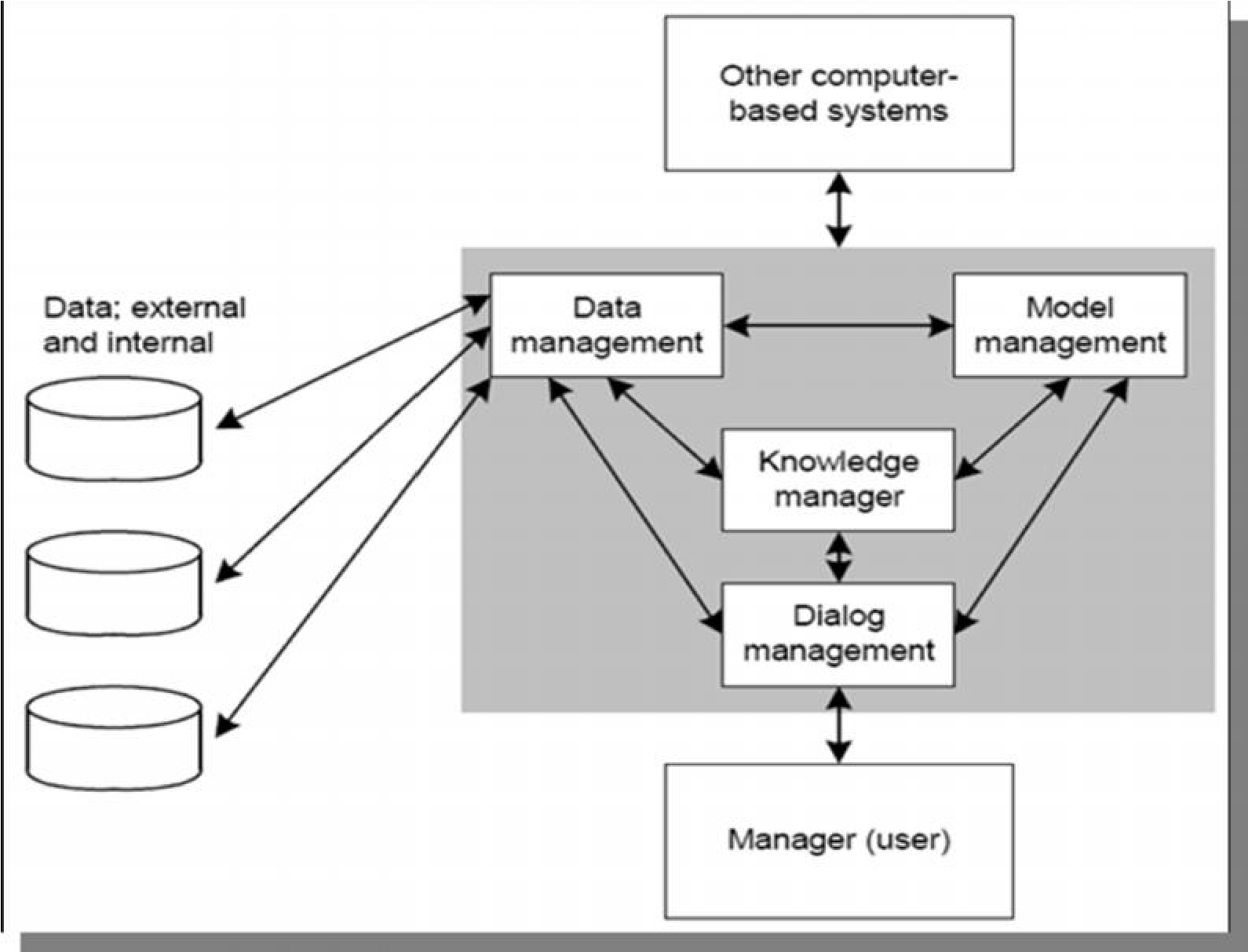
Melibatkan model finansial, statistika, *management science*, atau berbagai model kualitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen software yang dibutuhkan

# 3. Communication

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

# 4. Knowledge Management

Subsistem optional ini dapat mendukung subsistem lain atau bertindak atau bertindak sebagai komponen yang berdiri sendiri.



**Gambar 2.2. Model Konseptual SPK** Sumber: (Iran Berbakti, 2002).

**2.2 Metode *Weighted Product* (WP) *dan Simple Additive Weighting***

# **(SAW)**

## 2.2.1. Metode *Weighted Product* (WP)

Metode *WP* merupakan metode pengambilan keputusan dengan cara perkalian untuk menghubungkan rating atribut, di mana rating setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan. WP adalah salah satu analisis multi-kriteria keputusan (multi-criteria decision analysis / MUDA) yang sangat terkenal. Metode multi-kriteria pengambilan keputusan multi-criteria decision making (MCDM). Metode MUDA, yang diberikan adalah satu set terbatas dari alternatif keputusan yang dijalankan dalam hal sejumlah criteria keputusan. Setiap alternatif keputusan dibandingkan dengan yang lain dengan mengalikan sejumlah rasio, satu untuk setiap kriteria keputusan. Setiap rasio disingkat ke kekuasaan setara dengan berat relatif dari kriteria yang sesuai. *(Basyaib,2006:139)*

**2.2.2 Konsep Perhitungan dengan Metode *Weighted Product* (WP)** 1. Preferensi untuk alternatif Ai diberikan sebagai berikut

(*Basyaib,2006:140*) :

dengan i=1,2,...,m; di mana Σwj = 1

…………….(2.1)



1. wj adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negative untuk atribut biaya.
2. Untuk menentukan nilai maka lebih ke arah benefit ( atribut

keuntungan) jadi pangkatnya bernilai positif.

Metode WP menggunakan perkalian untuk menghubungkan rating atribut, di mana rating setiap atribut harus dipangkatkan dulu dengan bobot yang bersangkutan. Proses ini sama halnya dengan proses normalisasi.

Preferensi untuk alternative Si diberikan sebagai berikut :

…………(2.2)



dengan i = 1,2,…,m.

di mana :

S : *Preferensi* alternatif dianologikan sebagai vektor S

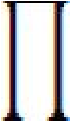
X : Nilai kriteria

W : Bobot kriteria/subkriteria

I : Alternatif J : Kriteria n : Layaknya criteria

di mana ΣWj = 1 .Wj adalah pangkat bernilai positif untuk atribut keuntungan dan bernilai negatif untuk atribut biaya. Preferensi relatif dari setiap alternatif, diberikan sebagai :

…………....(2.3)



di mana :

V : *Preferensi* alternatif dianalogikan sebagai vektor V

X : Nilai Kriteria

W : Bobot kriteria/subkriteria i : Alternatif j : Kriteria n : Layaknya kriteria

\* : Layaknya kriteria yang telah dimulai pada vektor S

Langkah – langkah menggunakan metode WP : [3]

1. Mengalikan seluruh atribut bagi sebuah alternatif dengan bobot sebagai pangkat positif untuk atribut manfaat dan bobot berfungsi sebagai pangkat negatif pada atribut biaya.
2. Hasil perkalian dijumlahkan untuk menghasilkan nilai pada setiap

alternatif.

1. Mencari nilai alternatif dengan melakukan langkah yang sama seperti langkah satu, hanya saja menggunakan nilai tertinggi untuk setiap atribut tertinggi untuk setiap atribut manfaat dan terendah untuk atribut biaya.
2. Membagi nilai *V* bagi setiap alternatif dengan nilai standar (*V*(*A\**)) yang menghasilkan *R*.
3. Ditemukan urutan alternatif terbaik yang akan menjadi keputusan.

## 2.2.3 Metode *Simple Additive Weighting (SAW)*

Metode SAW merupakan metode yang juga dikenal dengan metode penjumlahan berbobot. Konsep dasar metode SAW adalah mencari

penjumlahan berbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn, 1967) (MacCrimmon, 1968).

Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi *Multiple Attribute Decision Making* (MADU). MADU itu sendiri merupakan suatu metode yang

digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu.

Ada beberapa tahapan untuk menyelesaikan suatu kasus menggunakan metode SAW yaitu :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria(Ci), kemudian melakukan normalisasi matriks berdasarkan persamaan yang

disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks menormalisasi R.

1. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks menormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai) sebagai solusi.

Metode SAW *(Simple Additive Weighting)* sering juga dikenal dengan istilah metode penjumlahan berbobot. Konsep dasar metode SAW adalah mencari penjumlahan berbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Formula untuk melakukan normalisasi tersebut adalah sebagai berikut :

**Jika J adalah atribut Keuntungan (*benefit)***

**X**

**ii**

**Max X**

**ii**

**r**

**ij**

**=**

**Min**

**X**

**ii**

**X**

**ii**

…….…….(2.4)

# **Jika J adalah atribut Biaya (cost)**

Keterangan :

Rij : Nilai rating kinerja menormalisasi

Xij : Nilai atribut yang memiliki dari setiap criteria

Max Xij : Nilai terbesar dari setiap criteria

Min Xij : Nilai terkecil dari setiap criteria

Benefit : Jika nilai terbesar adalah terbaik

Cost : Jika nilai terkecil adalah terbaik Di mana rij adalah rating kinerja menormalisasi dari alternatif Ai pada atribut Cj; i=1,2,…,m dan j=1,2,…,n. Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai:

……………. (2.5)



Keterangan :

Vi : Rangking untuk setiap alternative wj : Nilai bobot dari setiap criteria

rij : Nilai rating kinerja menormalisasi. Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.

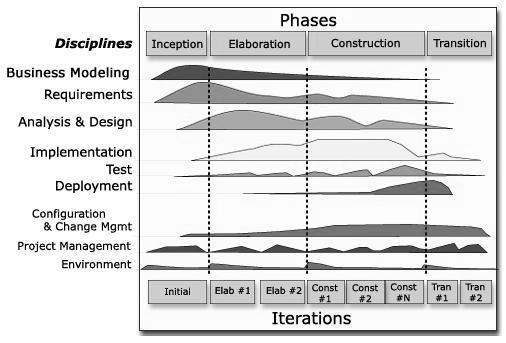
Kelebihan dari model *Simple Additive Weighting (SAW)* dibandingkan dengan model pengambilan keputusan yang lain terletak pada kemampuannya untuk melakukan penilaian secara lebih tepat karena diajarkan pada nilai criteria dan bobot preferensi yang sudah ditentukan, selain itu *SAW* juga dapat menyeleksi alternatif terbaik dari sejumlah alternatif yang ada karena adanya proses perankingan setelah menentukan nilai bobot untuk setiap atribut.

## 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 perangkat lunak.

RUP menggunakan konsep *object oriented*, dengan aktivitas yang berfokus pada pengembangan model dengan menggunakan *Unified Modeling Language* (UML). 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 aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijalankan ke 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 Management, Project Management, Environment.*



## Gambar 2.3 Arsitektur *Rational Unified Process*

Pada penggunaan kedua standard tersebut di atas yang berorientasi obyek (*object oriented*) memiliki manfaat yakni:

### 1. Improve productivity

Standard ini dapat memanfaatkan kembali komponen-komponen yang telah tersedia/dibuat sehingga dapat meningkatkan produktifitas.

### 2. Deliver high quality system

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

### 3. Lower maintenance cost

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

### 4. Facilitate reuse

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

aplikasi yang lainnya.

### 5. Manage complexity

Standard 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 (Laman : 2004, Fowler : 2003). Sedangkan koneksi 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 diterapkan adalah “apa” yang diperluas 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).

1. *Statechart Diagram* : menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diderita. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).
2. *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.

1. *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 antar dimensi vertikal (waktu) dan dimensi horizontal (objekobjek yang terkait).
2. *Component Diagram* : menggambarkan struktur dan hubungan antar komponen peranti lunak, termasuk ketergantungan (*dependency*) di antaranya.
3. *Deployment Diagram* : menggambarkan detail bagaimana komponen

di-deploy dalam infrastruktur sistem, di mana komponen akan terletak

(pada mesin, server atau peranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisika.

##### *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 dimakan seorang pengguna dalam kaitannya dengan sistem (Menawari, 2005).

## Tabel 2.1 Simbol *Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| NO | GAMBAR | NAMA | KETERANGAN |
| 1 |  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri  *(independent)* akan mempengaruhi elemen yang bergantung adanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalizatio* | Hubungan di mana objek anak  (*descendent*) berbagi perilaku dan |
|  |  | *n* | struktur data dari objek yang ada di alasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemenelemennya (sinergi). |

### **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 (Menawari, 2005).

**Tabel 2.2 Simbol *Activity Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| NO | GAMBAR | NAMA | KETERANGAN |
| 1 |  | *Activity* | Memperlihatkan bagaimana masing-masing 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 |  | *Activity*  *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* mengabarkan perilaku sebuah skenario. Skenario adalah rangkaian langkah-langkah yang mengabarkan 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* (Menawari, 2005).

## Tabel 2.3 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 informasi- |
|  |  |  | informasi tentang aktivitas yang terjadi |
| 4 |  | *Message* | Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang di akhir, 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 objek-objek tersebut. *Class diagram* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Menawari, 2005).

**Tabel 2.4 Simbol *Class Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| NO | GAMBAR | NAMA | KETERANGAN |
| 1 |  | *Generalization* | Hubungan di mana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di alasnya 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 di mana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung adanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |

### **2.5 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 gulana *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, Perlu, CGI*, dan Java untuk mengakses data yang disimpannya.

Kelebihan yang diajarkan 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* dukung oleh *ODBC* (*Open Database Connectivity*), artinya *database*nya dapat diakses aplikasi apa saja seperti *Java*, *Delphi*, dan *Visual Basic*.

#### 2.6 Definisi Karyawan

Karyawan merupakan kekayaan utama dalam suatu perusahaan, Karena tanpa adanya keikutsertaan mereka, aktivitas perusahaan tidak akan terlaksana. Beberapa pengertian karyawan menurut para ahli :

1. Menurut Hasibuan (dalam Magelang , 2002), karyawan adalah orang penjual jasa (pikiran atau tenaga) dan mendapat kompensasi yang dasarnya telah ditetapkan terlebih dahulu.
2. Menurut Suri (dalam Magelang, 2002), karyawan adalah penduduk dalam usia kerja (berusia 15-64 tahun) atau jumlahseluruh penduduk dalam suatu Negara yang memproduksibarang dan jasa jika ada permintaan terhadap tenaga mereka, dan jika mereka mau berpartisipasi dalam aktivitas tersebut.

#### *2.7 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 ke dalam segmen-segmen 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 simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

**Tabel 2.5 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 |
|  | *Comment* | berbeda  Tempat komentar  tentang suatu proses |