

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (*Decision Support Sistem*) merupakan suatu istilah yang mengacu pada suatu sistem yang memanfaatkan dukungan computer dalam proses pengambilan keputusan. Untuk memberikan pengertian tersebut, disini akan diuraikan definisi mengenai Sistem Pendukung Keputusan. Yaitu, SPK merupakan suatu sistem yang interaktif, yang membantu pengambil keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah yang sifatnya semi terstruktur maupun yang tidak terstruktur (Rahman, 2011).

SPK atau *Decission Support Sistem* (DSS) adalah merupakan suatu kumpulan sistem yang dapat mendukung proses pengambilan keputusan, yang selanjutnya dapat menunjang pengambilan keputusan dalam memperoleh data dan menguji beberapa alternatif-alternatif solusi yang mengandung konsekuensi-konsekuensi selama proses pemecahan masalah berlangsung atau boleh disebut merupakan aplikasi dari sebuah sistem informasi yang membantu proses pengambilan keputusan.

SPK tidak ditekankan untuk membuat keputusan, tetapi untuk melengkapi mereka yang terlibat dalam pengambilan keputusan dengan sekumpulan kemampuan untuk mengolah informasi yang diperlukan dalam proses pengambilan keputusan dan sistem ini bukan dimaksudkan untuk mengganti pengambilan keputusan dalam membuat suatu keputusan, melainkan mendukung pengambil keputusan.

2.2 Karakteristik sistem pendukung keputusan

Karakteristik sistem pendukung keputusan adalah sebagai berikut:

- a. Sistem Pendukung Keputusan dirancang untuk membantu pengambilan keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun yang tidak terstruktur dengan menambah kebijaksanaan manusia dan informasi komputerisasi.
- b. Dalam proses pengolahannya, SPK mengkombinasikan penggunaan model-model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari/interogasi informasi.
- c. SPK dirancang sedemikian rupa sehingga dapat digunakan/dioperasikan dengan mudah.
- d. SPK dirancang dengan menekankan pada aspek fleksibel serta kemampuan adaptasi yang tinggi.

Manfaat yang dapat diambil dari sistem pendukung keputusan ini :

- a. SPK memperluas kemampuan pengambilan keputusan dalam memproses data/informasi bagi pemakainya.
- b. SPK membantu pengambilan keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak teratur.
- c. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan. Walaupun mungkin saja SPK, tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun ia dapat menjadi *stimulant* bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan (Parwati, 2009).

2.3 Arsitektur SPK

Tiga komponen (arsitektur) dalam SPK, yaitu sebagai berikut:

2.3.1. KOMPONEN DIALOG (User System Interface)

a. Knowledge Base (Bennott)

- Apa yang diketahui user tentang keputusan, bagaimana cara menggunakan DSS;
- Pengetahuan apa yang harus dimiliki user agar dapat berinteraksi dengan sistem dalam berhubungan dengan area masalah dan dalam pembuatan keputusan yang diperlukan
- atau apa yang sudah dikuasai user (mis : manual, help options, dll)

Teknik pelatihan penggunaan DSS, dapat dilakukan dengan “Melatih”

user :

- One on one tutorial = eksekutif senior
- Closes and lectures = user yang memerlukan training berjumlah banyak
- Instruksi terprogram dan instruksi yang dibantu dengan = komputer bila DSS dipakai dalam jangka lama dan digunakan dalam memberikan layanan bagi banyak pemakai
- Manual book
- Command / sequence file □ Berisi instruksi yang diprogramkan sebelumnya

b. Action Language (Bahasa Tindakan)

Apa yang dapat dilakukan user dalam berkomunikasi dengan sistem (mengontrol DSS); atau Opsion yang mengarahkan tindakan sistem tsb.

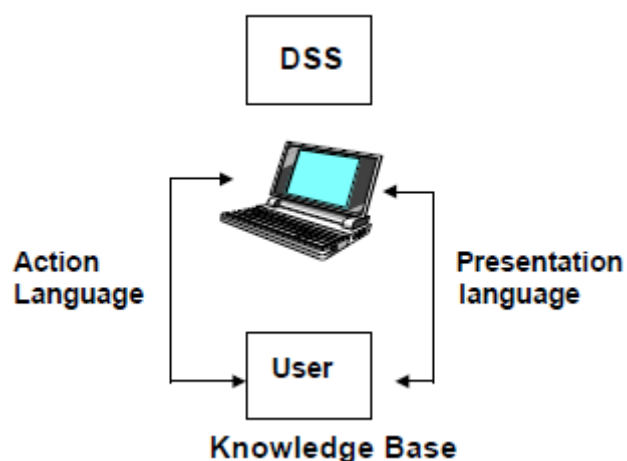
Action language dapat dilakukan dengan :

- Cara tanya jawab
- Menggunakan menu
- Bahasa perintah
- Pendekatan form Input - Output

c. Presentation Language (Bahasa Presentasi)

Presentasi alternatif dari respon sistem tsb; atau apa yang dilihat oleh user (laporan tercetak dari DSS).

Contoh : printer, monitor, grafik, warna, audio output, animation, dll

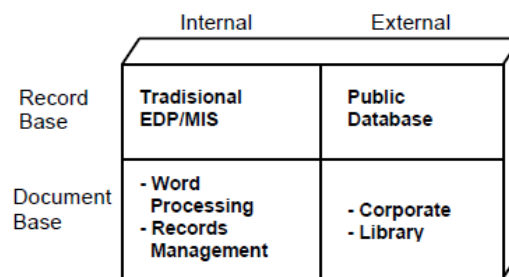


Gambar 2.1 The user system interface.

2.3.2. KOMPONEN DATA

Ada empat Jenis informasi :

- a. Dua jenis informasi yang dikelola secara internal
 - Informasi dari record data (entity)
 - Informasi dari dokumen (konsep, gagasan, laporan, memo)
- b. Dua jenis yang dikelola secara eksternal (dapat dibeli)
 - Informasi yang didasarkan pada catatan / record (Kondisi ekonomi dan keuangan, jadwal penerbangan, kuota harga stock)
 - Informasi yg didasarkan pd dokumen eksternal (Opini mengenai ramalan atau data ekonomi regional)



2.3.3. KOMPONEN MODEL

Model didalam DSS dapat dianggap sebagai model base.

Ada empat jenis-jenis model yaitu sebagai berikut.

- a. Model Strategis
 - Digunakan oleh manajemen puncak untuk :
 - membantu menetapkan tujuan organisasi
 - menetapkan sumber daya untuk meraih tujuan tersebut
 - menetapkan kebijaksanaan untuk mengatur perolehan disposisi sumber daya tsb (misal : perencanaan tujuan

perusahaan, penentuan lokasi, perencanaan dampak lingkungan)

- Data yang dibutuhkan sebagian besar data eksternal dan subyektif
- Cakrawala waktu utk model tsb biasanya diukur dalam tahun (misal : jangka waktu tanggung jawab perencanaan strategis manajemen puncak)
- Model tsb bersifat deterministik dan deskriptif

b. Model Taktis

- Diterapkan oleh manajemen menengah untuk membantu dalam mengalokasi dan mengontrol penggunaan sumber daya organisasi (misal : perencanaan keuangan, perencanaan keperluan pabrik, perencanaan promosi penjualan, penentuan tata letak gedung / pabrik)
- Data yang dibutuhkan sebagian besar data internal dan beberapa data eksternal serta subyektif
- Cakrawala waktu : 1 bulan s/d 2 tahun
- Model ini bersifat : deterministik

c. Model Operasional

- Diterapkan untuk mendukung pembuatan keputusan jangka pendek (misal : harian atau mingguan) yang sering dijumpai pada tingkat organisasi bawah
- Data yang dibutuhkan : data internal
- Sifat deterministik

d. Block & Subroutine bangunan model

- Meliputi : pemrograman linier, analisis rangkaian waktu, analisis regresi, dan Prosedur Sampling Monte Carlo
- Dapat digunakan secara terpisah untuk mendukung keputusan atau digunakan secara bersama untuk merekonstruksi dan memelihara model yang lebih komprehensif

2.4 Beberapa Keterbatasan SPK

- a. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
- b. Kemampuan suatu SPK terbatas pada perbendaharaan pengetahuan yang dimilikinya (pengetahuan dasar serta model dasar).
- c. Proses - proses yang dapat dilakukan SPK biasanya juga tergantung pada perangkat lunak yang digunakan.
- d. SPK tidak memiliki kemampuan intuisi seperti yang dimiliki manusia. Sistem ini dirancang hanyalah untuk membantu pengambilan keputusan dalam melaksanakan tugas (Rahman, 2011).

2.5 Logika *Fuzzy*

a. Pengertian Logika *Fuzzy*

Dalam logika konvensional, nilai kebenaran mempunyai kondisi yang pasti yaitu benar atau salah (true or false), dengan tidak ada kondisi antara. Prinsip ini telah mendominasi pemikiran logika di dunia sampai sekarang. Tentu saja, pemikiran mengenai logika konvensional dengan nilai kebenaran yang pasti yaitu benar atau salah dalam kehidupan yang

nyata sangatlah tidak mungkin. logika *fuzzy* menawarkan suatu logika yang dapat merepresentasikan keadaan dunia nyata (Kusumadewi & Purnomo, 2010).

Teori himpunan logika *fuzzy* di kembangkan oleh Professor Lofti A. Zadeh pada tahun 1965. Ia ber-pendapat bahwa logika benar dan salah dari logika boolelan konvensional tidak dapat mengatasi masalah gradasi yang berada pada dunia nyata. Untuk mengatasi masalah gradasi yang tidak terhingga. Zadeh mengembangkan sebuah himpunan *fuzzy*. tidak seperti logika Boolean, logika *fuzzy* mempunyai nilai yang kontinu. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama.

Berdasarkan hal tersebut diatas Logika *fuzzy* dapat digunakan untuk memodelkan suatu permasalahan yang matematis, dimana konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.

Logika *fuzzy* merupakan generalisasi dari logika klasik (Crisp Set) yang hanya memiliki dua nilai keanggotaan yaitu 0 dan 1. Dalam logika *fuzzy* nilai kebenaran suatu pernyataan berkisar dari sepenuhnya benar sampai dengan sepenuhnya salah.

Fuzzy Logic berhubungan dengan ketidakpastian yang telah menjadi sifat alamiah manusia, mensimulasikan proses pertimbangan normal manusia dengan jalan memungkinkan komputer untuk berperilaku sedikit

lebih seksama dan logis daripada yang dibutuhkan metode computer konvensional.

Pemikiran di balik pendekatan ini adalah pengambilan keputusan tidak sekadar persoalan hitam dan putih atau benar dan salah, namun kerap kali melibatkan area abu-abu, dan hal itu dimungkinkan.

b. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu group yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Pada himpunan tegas (crisp), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $f_A[x]$, memiliki dua kemungkinan, yaitu : Satu (I), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan *fuzzy* $f_A[x] = 0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan *fuzzy* $f_A[x] = 1$ berarti x menjadi anggota penuh pada himpunan A .

Kemiripan antara keanggotaan *fuzzy* dengan probabilitas terkadang menimbulkan kerancuan, karena memiliki nilai pada interval $[0,1]$, namun interpretasi nilainya sangat berbeda. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang.

Himpunan *fuzzy* memiliki dua atribut, yaitu :

- 1) Linguistik, yaitu penamaan suatu group yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : Muda, Parobaya, Tua.
- 2) Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti : 25,40,60.

Ada beberapa hal yang perlu diketahui dalam memahami suatu sistem *fuzzy*, yaitu :

- 1) Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu system *fuzzy*. Contoh : umur, temperatur, tingkat pendapatan, dsb.

- 2) Himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.

Contoh :

- a) Variabel kelayakan rumah, terbagi menjadi 3 himpunan *fuzzy*, yaitu : Layak, Cukup Layak, Kurang Layak.
- b) Variabel Tingkat Pendapatan, terbagi menjadi 3 himpunan *fuzzy*, yaitu : Tinggi, Sedang dan Rendah.



Gambar 2.2 Himpunan *Fuzzy* pada variabel tingkat pendapatan

3) Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh :

- a) Semesta pembicaraan untuk variabel tingkat pendapatan: $[0 +\infty]$
- b) Semesta pembicaraan untuk variabel temperatur: $[0 40]$

4) Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

Contoh domain himpunan *fuzzy* :

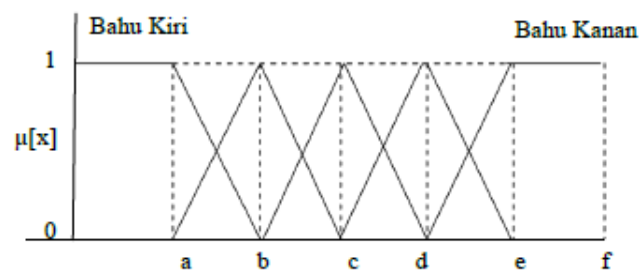
- a) Rendah : $[0 750.000]$
- b) Sedang : $[300.000 2.000.000]$
- c) Tinggi : $[2.000.000 +\infty]$

5) Fungsi keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaan yang

memiliki nilai interval antara 0 dan 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi.

Salah satu representasi fungsi keanggotaan dalam *fuzzy* yang akan dipakai adalah representasi kurva bentuk bahu. Kurva yang bentuknya seperti bahu di sisi paling kanan dan paling kirinya. Himpunan *fuzzy* “bahu”, bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*.



Gambar 2.3 Representasi Kurva Bentuk Bahu.

Ada dua keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.

6) Metode Tsukamoto

Dalam membangun sebuah sistem *fuzzy* dikenal beberapa metode penalaran, antara lain : metode Tsukamoto, metode Mamdani dan metode Sugeno.

Pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-THEN harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan dengan tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

Misal ada 2 variabel input, var-1(x) dan var-2(y) serta 1 variabel output var-3(z), dimana var-1 terbagi atas 2 himpunan yaitu A1 dan A2 dan var-2 terbagi atas himpunan B1 dan B2. Sedangkan var-3 juga terbagi atas 2 himpunan yaitu C1 dan C2. (Kusumadewi, 2003).

Ada dua aturan yang digunakan yaitu:

[R1] IF (x is A1) and (y is B2) THEN (z is C1)

[R2] IF (x is A2) and (y is B1) THEN (z is C2)

Penulis memilih metode tsukamoto karena metode ini setiap konsekuen pada aturan yang berbentuk IF-THEN direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan dengan tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Kemudian hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot. (Kusumadewi & Purnomo, 2010)

Contoh kasus

Berdasarkan sumber dari buku “Sri Kusumadewi dan Hari Purnomo dengan judul APLIKASI LOGIKA FUZZY untuk Pendukung Keputusan pada halaman 32”

Suatu perusahaan makanan kaleng akan memproduksi makanan jenis ABC. Dari data 1 bulan terakhir, permintaan terbesar hingga mencapai 5000 kemasan/hari, dan permintaan terkecil sampai 1000 kemasan/hari. Persediaan barang digudang terbanyak sampai 600 kemasan/hari, dan terkecil pernah sampai 100 kemasan/hari. Dengan segala keterbatasannya, sampai saat ini, perusahaan baru mampu memproduksi barang maksimum 7000 kemasan/hari, serta demi efisiensi mesin dan SDM tiap hari diharapkan perusahaan memproduksi paling tidak 2000 kemasan. Apabila proses produksi perusahaan tersebut menggunakan 4 aturan fuzzy sbb:

[R1] IF Permintaan TURUN And Persediaan BANYAK THEN Produksi Barang BERKURANG;

[R2] IF Permintaan TURUN And Persediaan SEDIKIT THEN Produksi Barang BERKURANG;

[R3] IF Permintaan NAIK And Persediaan BANYAK THEN Produksi Barang BERTAMBAH;

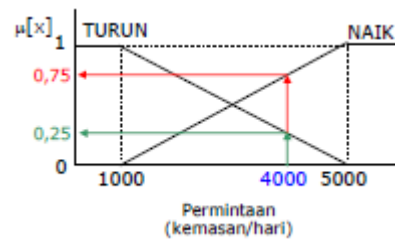
[R4] IF Permintaan NAIK And Persediaan SEDIKIT THEN Produksi Barang BERTAMBAH;

Berapa kemasan makanan jenis ABC yang harus diproduksi, jika jumlah permintaan sebanyak 4000 kemasan, dan persediaan di gudang masih 300 kemasan?

Solusi

Ada 3 variabel fuzzy yang akan dimodelkan, yaitu:

- Permintaan; terdiri-atas 2 himpunan fuzzy, yaitu: NAIK dan TURUN.



$$\mu_{\text{PmtTURUN}}[x] = \begin{cases} 1, & x \leq 1000 \\ \frac{5000 - x}{4000}, & 1000 \leq x \leq 5000 \\ 0, & x \geq 5000 \end{cases}$$

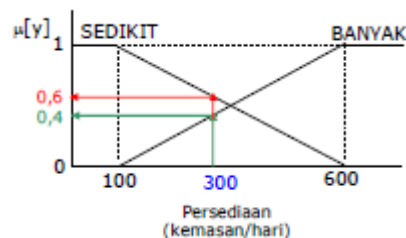
$$\mu_{\text{PmtNAIK}}[x] = \begin{cases} 0, & x \leq 1000 \\ \frac{x - 1000}{4000}, & 1000 \leq x \leq 5000 \\ 1, & x \geq 5000 \end{cases}$$

Kita bisa mencari nilai keanggotaan:

$$\begin{aligned} \mu_{\text{PmtTURUN}}[4000] &= (5000-4000)/4000 \\ &= 0,25 \end{aligned}$$

$$\begin{aligned} \mu_{\text{PmtNAIK}}[4000] &= (4000-1000)/4000 \\ &= 0,75 \end{aligned}$$

- Persediaan; terdiri-atas 2 himpunan fuzzy, yaitu: SEDIKIT dan BANYAK.



$$\mu_{\text{PsdSEDIKIT}}[y] = \begin{cases} 1, & y \leq 100 \\ \frac{600-y}{500}, & 100 \leq y \leq 600 \\ 0, & y \geq 600 \end{cases}$$

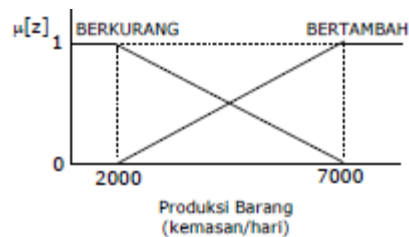
$$\mu_{\text{PsdBANYAK}}[y] = \begin{cases} 0, & y \leq 100 \\ \frac{y-100}{500}, & 100 \leq y \leq 600 \\ 1, & y \geq 600 \end{cases}$$

Kita bisa mencari nilai keanggotaan:

$$\begin{aligned} \mu_{\text{PsdSEDIKIT}}[300] &= (600-300)/500 \\ &= 0,6 \end{aligned}$$

$$\begin{aligned} \mu_{\text{PsdBANYAK}}[300] &= (300-100)/500 \\ &= 0,4 \end{aligned}$$

- Produksi barang; terdiri-atas 2 himpunan fuzzy, yaitu: BERKURANG dan BERTAMBAH.



$$\mu_{\text{Prg BERKURANG}}[z] = \begin{cases} 1, & z \leq 2000 \\ \frac{7000-z}{5000}, & 2000 \leq z \leq 7000 \\ 0, & z \geq 7000 \end{cases}$$

$$\mu_{\text{Prg BERTAMBAH}}[z] = \begin{cases} 0, & z \leq 2000 \\ \frac{z-2000}{5000}, & 2000 \leq z \leq 7000 \\ 1, & z \geq 7000 \end{cases}$$

Sekarang kita cari nilai z untuk setiap aturan dengan menggunakan fungsi MIN pada aplikasi fungsi implikasinya:

[R1] IF Permintaan TURUN And Persediaan BANYAK THEN
Produksi Barang BERKURANG;

$$\begin{aligned}
\alpha\text{-predikat1} &= \mu_{\text{PmtTURUN}} \cap \mu_{\text{PsdBANYAK}} \\
&= \min(\mu_{\text{PmtTURUN}} [4000], \mu_{\text{PsdBANYAK}} [300]) \\
&= \min(0,25; 0,4) \\
&= 0,25
\end{aligned}$$

Lihat himpunan Produksi Barang BERKURANG,

$$(7000-z)/5000 = 0,25 \text{ ---> } z1 = 5750$$

[R2] IF Permintaan TURUN And Persediaan SEDIKIT THEN
Produksi Barang BERKURANG;

$$\begin{aligned}
\alpha\text{-predikat2} &= \mu_{\text{PmtTURUN}} \cap \mu_{\text{PsdSEDIKIT}} \\
&= \min(\mu_{\text{PmtTURUN}} [4000], \mu_{\text{PsdSEDIKIT}} [300]) \\
&= \min(0,25; 0,6) \\
&= 0,25
\end{aligned}$$

Lihat himpunan Produksi Barang BERKURANG,

$$(7000-z)/5000 = 0,25 \text{ ---> } z2 = 5750$$

[R3] IF Permintaan NAIK And Persediaan BANYAK THEN
Produksi Barang BERTAMBAH;

$$\begin{aligned}
\alpha\text{-predikat3} &= \mu_{\text{PmtNAIK}} \cap \mu_{\text{PsdBANYAK}} \\
&= \min(\mu_{\text{PmtNAIK}} [4000], \mu_{\text{PsdBANYAK}} [300]) \\
&= \min(0,75; 0,4) \\
&= 0,4
\end{aligned}$$

Lihat himpunan Produksi Barang BERTAMBAH,

$$(z-2000)/5000 = 0,4 \text{ ---> } z3 = 4000$$

[R4] IF Permintaan NAIK And Persediaan SEDIKIT THEN
Produksi Barang BERTAMBAH;

$$\begin{aligned}
\alpha\text{-predikat4} &= \mu\text{PmtNAIK} \cap \text{PsdBANYAK} \\
&= \min(\mu\text{PmtNAIK} [4000], \mu\text{PsdSEDIKIT}[300]) \\
&= \min(0,75; 0,6) \\
&= 0,6
\end{aligned}$$

Lihat himpunan Produksi Barang BERTAMBAH,

$$(z-2000)/5000 = 0,6 \rightarrow z = 5000$$

Dari sini kita dapat mencari berapakah nilai z, yaitu:

$$\begin{aligned}
z &= \frac{\alpha\text{pred}_1 * z_1 + \alpha\text{pred}_2 * z_2 + \alpha\text{pred}_3 * z_3 + \alpha\text{pred}_4 * z_4}{\alpha\text{pred}_1 + \alpha\text{pred}_2 + \alpha\text{pred}_3 + \alpha\text{pred}_4} \\
z &= \frac{0,25 * 5750 + 0,25 * 5750 + 0,4 * 4000 + 0,6 * 5000}{0,25 + 0,25 + 0,4 + 0,6} = \frac{7475}{1,5} = 4983
\end{aligned}$$

Jadi jumlah makanan kaleng jenis ABC yang harus diproduksi sebanyak 4983 kemasan.

2.6 Visual Basic.NET

Visual Studio .NET terdiri dari empat edisi yang dimulai dari edisi yang paling dasar sampai edisi paling lengkap. Keempat edisi yang ada di Visual Studio .NET sebagai berikut: (Yuswanto, 2006)

a. Visual Studio .NET Professional

Pada edisi ini selain terdiri dari empat bahasa pemrograman juga dilengkapi dengan layanan Web XML (Extensible Markup Language) membangun aplikasi Web maupun Windows dan membangun aplikasi mobile dengan WML pada telepon seluler berbasis WAP dan HTML untuk PC, Palm dan pager. Pada edisi ini disertakan Microsoft SQL Server 2000 Desktop Engine yang kompatibel dengan database SQL Server.

b. Visual Studio .NET Academic

Pada edisi ini selain terdapat seluruh fitur pada edisi Professional juga terdapat beberapa fitur instruksional yang didesain untuk menyederhanakan pengelolaan mata kuliah. Dengan fitur-fitur yang ada memungkinkan materi kuliah atau tugas-tugas untuk mahasiswa dapat diakses melalui server Web atau situs FTP (File Transfer Protocol).

c. Visual Studio .NET Enterprise Developer

Pada edisi ini selain terdapat seluruh fitur pada edisi professional juga ditambah kemampuan pengembangan aplikasi enterprise, seperti:

- 1) Tim pengembangan enterprise dapat bekerja sama untuk aplikasi Windows dan Web.
- 2) Terdapat tool dan teknologi untuk membangun aplikasi enterprise.
- 3) Mempunyai template proyek enterprise dan Frameworks yang berisi petunjuk arsitektur untuk membuat aplikasi.

d. Visual Studio .NET Enterprise Architect

Pada edisi ini selain terdapat seluruh fitur pada edisi enterprise Developer juga ditambah dengan kemampuan untuk mendesain, menentukan dan mengkomunikasikan arsitektur dan fungsionalitas aplikasi. Beberapa fitur tambahan yang ada, diantaranya:

- 1) Dapat digunakan untuk mendesain layanan Web XML dan aplikasi secara Visual.
- 2) Terdapat software modeling Visio berbasis UML (Unified Modelling language) untuk membuat aplikasi arsitektur bisnis dan desain database.

- 3) Terdapat fasilitas BizTalkServer 2000 Developer Edition untuk proses bisnis.
- 4) Terdapat Enterprise Template dan Frameworks dengan menggunakan template Description Language.

2.7 MySQL

MySQL merupakan database server yang dikembangkan oleh sebuah perusahaan bernama MySQL AB. Perusahaan yang beralokasi di Swedia ini memiliki hak resmi untuk mengembangkan dan mengelola sistem, memberikan dukungan penjualan dan layanan, serta memperkerjakan orang-orang yang memberikan kontribusi pada MySQL.

Sebagaimana diungkapkan oleh pengembang resminya, MySQL dilafalkan dengan ucapan “My Ess Que Ell”, bukan my sequel” atau yang lainnya. Meskipun demikian, pihak mySQL AB juga tidak begitu mempermasalahkan bagaimana kita mengucapkannya (Didik, 2005).

2.8 Diagram Use Case

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 actor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. (Lukman Hakim, 2012).

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun

requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang diinclude akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

2.9 Diagram Activity

Activity diagrams 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. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). (Lukman Hakim, 2012)

Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case

menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

2.10 Diagram Sequence

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). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. (Hakim, 2012).

2.11 Penerimaan Karyawan/Rekrutmen

Rekrutmen adalah proses penarikan, seleksi, penempatan, orientasi, dan induksi untuk mendapatkan karyawan yang efektif dan efisien membantu tercapainya tujuan perusahaan. (Hasibuan 2008 : 28).