BAB II

LANDASAN TEORI

2.1 *Outsourcing*

2.1.1 Definisi *Outsourcing*

Dalam pengertian umum, istilah *outsourcing* diartikan sebagai *contract* (*work out*). Menurut definisi Maurice Greaver, *outsourcing* dipandang sebagai tindakan mengalihkan beberapa aktivitas perusahaan dan hak pengambilan keputusaannya kepada pihak lain (*outside provider*), dimana tindakan ini terkait dalam suatu kontrak kerja sama.

Dapat juga dikatakan *outsourcing* sebagai penyerahan kegiatan perusahaan baik sebagian ataupun secara menyeluruh kepada pihak lain yang tertuang dalam kontrak perjanjian. Ada tiga unsur penting dalam *outsourcing*, yaitu:

1. Terdapat pemindahaan fungsi pengawasan.
2. Ada pendelegasian tanggung jawab/tugas suatu perusahaan.
3. Dititik beratkan hasil/*output* yang ingin dicapai oleh perusahaan.

Dari beberapa definisi yang dikemukakan diatas, terdapat persamaan dalam memandang *outsourcing*, yaitu adanya penyerahan sebagai kegiataan perusahaan pada pihak lain, yang diharapakan memberikan hasil berupa peningkatan kinerja agar dapat lebih kompetitif dalam mengahadapi

perkembangan ekonomi dan teknologi global. Secara umum pengertian *outsourcing* adalah:

1. Penyerahan tanggung jawab kegiatan perusahaan kepada pihak ketiga sebagai pengawas pelayanan yang telah disepakati.
2. Penyerahaan kegiatan, tugas atau pun pelayanan pada pihak lain, dengan tujuan untuk mendapatkan tenaga ahli serta meningkatkan efisiensi dan efektivitas perusahaan.

2.1.2 Pekerjaan yang Bisa Di-*Outsource*

Pada dasarnya, pekerjaan yang bisa di-outsource adalah pekerjaan penunjang (*non core*) dan bukan pekerjaan utama (*core*). “Pekerja/buruh dari perusahaan penyedia jasa pekerja/buruh tidak boleh digunakan oleh pemberi kerja untuk melaksanakan kegiatan pokok atau kegiatan yang berhubungan langsung dengan proses produksi, kecuali untuk kegiatan jasa penunjang atau kagiatan yang tidak berhubungan langsung dengan proses produksi.”

2.1.3 Manfaat *Outsourcing*

Outsourcing memiliki beberapa manfaat bagi beberapa pihak. Mulai dari pihak pemerintah,pihak masyarakat dan pekerja, dan juga pihak perusahaan tentunya. Berikut beberapa manfaat dari ketiga pihak tersebut :

A. Manfaat *outsourcing* bagi pemerintah

1. Dapat membantu mengembangkan dan mendorong pertumbuhan ekonomi masyarakat dan perekonomian nasional.

2. Sebagai pembinaan dan pengembangan kegiatan koperasi dan usaha kecil karena perusahaan sebagaian pelaku *outsource* berbentuk koperasi maupun usaha kecil.
3. Mengurangi beban pemerintah dalam penyediaan fasilitas umum seperti transportasi, listrik, air dan pelaksanaan ketertiban umum karena kegiatan tersebut bisa dilakukan oleh perusahaan *outsource*.

B. Manfaat bagi masyarakat dan pekerja

1. *Outsourcing* akan mempercepat pertumbuhan industri yang pada gilirannya akan mendorong ekonomi penunjang dilingkungan masyarakat seperti adanya pasar, warung makan sarana transportasi dan sebagainya.
2. Mengembangkan infrastruktur sosial masyarakat,budaya kerja, disiplin dan peningkatan kemampuan ekonomi..
3. Mengurangi pengangguran dan mencegah terjadinya urbanisasi.
4. Meningkatkan kemampuan budaya perusahaan budaya perusahaan dilingkungan masyarakat.
5. Bagi *fresh graduate outsourcing* bisa menjadi jembatan untuk karir sebelumnya.

C. Manfaat bagi perusahaan

1. Meningkatkan fokus perusahaan inti.
2. Penghematan dana kapital.
3. Efisiensi biaya operasional.
4. Memperoleh SDM yang lebih propesional.

2.2 Sistem Pendukung Keputusan (SPK)

Sistem pendukung keputusan (SPK) adalah bagian dari sistem informasi berbasis komputer yang termasuk sistem berbasis pengetahuan dalam suatu organisasi atau manajemen pengetahuan yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi terstruktur yang spesifik.

Dengan pengertian diatas dapat dijelaskan bahwa SPK bukan merupakan alat pengambilan keputusan melainkan merupakan sistem yang membantu pengambilan keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Sehingga sistem ini tidak dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan.

Menurut (Azhar, 1995) dari pengertian SPK maka dapat ditentukan karakteristik antara lain :

1. Mendukung proses pengambilan keputusan, menitik beratkan pada *management by perception.*
2. Adanya *interface* manusia atau mesin di mana manusia atau *user* tetap memegang kontrol proses pengambilan keputusan
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur dan tak terstruktur

4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan
5. Memiliki subsistem-subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan item
6. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan informasi seluruh tingkatan manajemen

Dalam sistem pendukung keputusan terdapat tiga keputusan tingkatan perangkat keras maupun lunak. Masing-Masing tingkatan berdasar pada kemampuan dan perbedaan tingkat teknik, lingkungan dan tugas yang akan dikerjakan. Ketiga tingkatan tersebut adalah :

1. Sistem Pendukung Keputusan (*Specific DSS*)
2. Pembangkit Sistem Pendukung Keputusan (*DSS Generator*)
3. Peralatan Sistem Pendukung Keputusan

Dalam sistem pendukung keputusan terdapat tiga jenis keputusan, yaitu :

1. Keputusan Terstruktur

   Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Informasi yang dibutuhkan spesifik, terjadwal, sempit, interaktif, *real time*, *internal*, dan *detail*. Prosedur yang dilakukan untuk pengambilan keputusan sangat jelas. Keputusan ini terutama dilakukan pada manajemen tingkat bawah. Contoh: Keputusan pemesanan barang dan keputusan penagihan piutang, menentukan

kelayakan lembur, mengisi persediaan, dan menawarkan kredit pada pelanggan.

2. Keputusan Semiterstruktur

   Keputusan semiterstruktur adalah keputusan yang mempunyai sifat yakni sebagian keputusan dapat ditangani oleh komputer dan yang lain tetap harus dilakukan oleh pengambil keputusan. Informasi yang dibutuhkan harus spesifik, interaktif, internal, *real time*, dan terjadwal. Contoh: Pengevaluasian kredit, penjadwalan produksi dan pengendalian ketersediaan, merancang rencana pemasaran, dan mengembangkan anggaran departemen.

3. Keputusan Tidak Terstruktur

   Keputusan tak terstruktur adalah keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Keputusan ini menuntut pengalaman dan berbagai sumber yang bersifat eksternal. Keputusan ini umumnya terjadi pada manajemen tingkat atas. Informasi yang dibutuhkan umum, luas, internal, dan eksternal. Contoh: Pengembangan teknologi baru, keputusan untuk bergabung dengan perusahaan lain, perekrutan eksekutif.

2.3 Logika *Fuzzy*

Menurut Asus Naba, logika *fuzzy* adalah: “Sebuah metodologi berhitung dengan variabel kata-kata (*linguistic variable*) sebagai pengganti berhitung dengan bilangan. Kata-kata digunakan dalam *fuzzy logic* memang

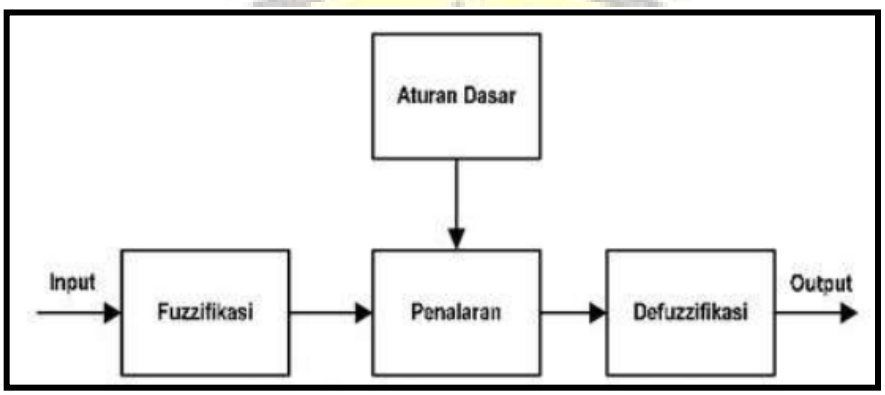
tidak sepresisi bilangan, namun kata-kata jauh lebih dekat dengan intuisi manusia" [NAB-2009]. Mengenai logika *fuzzy* pada dasarnya tidak semua keputusan dijelaskan dengan 0 atau 1, namun ada kondisi diantara keduanya, daerah diantara keduanya inilah yang disebut dengan *fuzzy* atau tersamar. Secara umum ada beberapa konsep sistem logika *fuzzy*, sebagai berikut dibawah ini:

a. Himpunan tegas yang merupakan nilai keanggotaan suatu item dalam suatu himpunan tertentu.
b. Himpunan *fuzzy* yang merupakan suatu himpunana yang digunakan untuk mengatasi kekakuan dari himpunan tegas.
c. Fungsi keanggotaan yang memiliki interval 0 sampai 1
d. Variabel *linguistic* yang merupakan suatu variabel yang memiliki nilai berupa kata-kata yang dinyatakan dalam bahasa alamiah dan bukan angka.
e. Operasi dasar himpunan *fuzzy* merupakan operasi untuk menggabungkan dan atau memodifikasi himpunan *fuzzy*.
f. Aturan (*rule*) *if-then fuzzy* merupakan suatu pernyataan *if-then*, dimana beberapa kata-kata dalam pernyataan tersebut ditentukan oleh fungsi keanggotaan.

Dalam proses pemanfaatan logika *fuzzy*, ada beberapa hal yang harus diperhatikan salah satunya adalah cara mengolah *input* menjadi *output* melalui sistem inferensi *fuzzy*. Metode inferensi *fuzzy* atau cara merumuskan pemetaan dari masukan yang diberikan kepada sebuah keluaran. Proses ini

melibatkan fungsi keanggotaan, operasi logika, serta aturan IF-THEN. Hasil dari proses ini akan menghasilkan sebuah sistem yang disebut dengan FIS (*Fuzzy* InferensiSystem). Dalam logika *fuzzy* tersedia beberapa jenis FIS diantaranya adalah Mamdani, Sugeno, dan Tsukamoto.

2.3.1 Struktur Dasar Logika *Fuzzy*



Gambar 2.1 Blok Diagram Logika *Fuzzy* (Sri Kusuma Dewi, 2010)

Berdasarkan gambar 2.1, dalam system logika *fuzzy* terdapat beberapa tahapan operasional yang meliputi:

a. Fuzzifikasi

   Fuzzifikasi adalah suatu proses pengubahan nilai tegas yang ada ke dalam fungsi keanggotaan.

b. Penalaran (*Inference Machine*)

   Mesin penalaran adalah proses implikasi dalam menalar nilai masukan guna penentuan nilai keluaran sebagai bentuk pengambilan keputusan. Salah satu model penalaran yang banyak dipakai adalah penalaran maxmin. Dalam

penalaran ini, proses pertama yang dilakukan adalah melakukan operasi min sinyal keluaran lapisan fuzzifikasi, yang diteruskan dengan operasi max untuk mencari nilai keluaran yang selanjutnya akan didefuzzifikasikan sebagai bentuk keluaran.

c. Aturan Dasar (*Rule Based*)

Aturan dasar (*rule based*) pada control logika *fuzzy* merupakan suatu bentuk aturan relasi "Jika-Maka"atau "if-then" seperti berikut ini:

if x is A then y is B dimana A dan B adalah *linguistic values* yang didefinisikan dalam rentang variabel X dan Y. Pernyataan "x is A" disebut *antecedent* atau premis. Pernyataan "y is B" disebut *consequent* atau kesimpulan.

d. Defuzzifikasi

Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu.

2.3.2 Alasan Digunakannya Logika *Fuzzy*

Ada beberapa alasan mengapa orang menggunakan logika *fuzzy*, antara lain :

a. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.

b. Logika *fuzzy* sangat fleksibel.

c. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.

d. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.

e. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalamanpengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.

f. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.

g. Logika *fuzzy* didasarkan pada bahasa alami.

(http://www.henypratiwi.com)

2.3.3 Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu objek x dalam suatu himpunan A, yang sering ditulis dengan μA[x], memiliki 2 kemungkinan yaitu sebagai berikut:

a. satu (1), yang berarti bahwa suatu objek menjadi anggota dalam suatu himpunan, atau

b. nol (0), yang berarti bahwa suatu objek tidak menjadi anggota dalam suatu himpunan (Kusumadewi et al, 2004).

Misalkan variabel umur dibagi 3 kategori sebagai berikut :

MUDA : umur < 35 tahun

PAROBAYA : 35 ≤ umur ≤ 55 tahun

TUA : umur > 55 tahun

Apabila seseorang berusia 34 tahun, maka ia dikatakan MUDA (μMUDA[34thn]= 1). Apabila seseorang berusia 35 tahun kurang 1 hari, maka ia dikatakan TIDAK MUDA (μMUDA [35thn -1 hr] = 0).

Adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan. Himpunan *fuzzy* digunakan untuk mengantisipasi hal tersebut. Seseorang dapat masuk dalam 2 himpunan yang berbeda, MUDA dan PAROBAYA, PAROBAYA dan TUA, dsb. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat berdasarkan nilai keanggotaannya.

Himpunan *Fuzzy* memiliki 2 atribut yakni sebagai berikut:

a. *Linguistic* adalah penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami. Menurut Wang, suatu variabel *linguistic* adalah sebuah variabel yang memiliki nilai berupa kata-kata dalam bahasa alamiah. Setiap variabel *linguistic* berkaitan dengan sebuah fungsi keanggotaan (Kusumadewi et al, 2004). Seperti : MUDA, PAROBAYA, TUA
b. Numeris adalah suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 35 Hal-hal yang terdapat dalam sistem *fuzzy* yaitu sebagai berikut: 1. Variabel *fuzzy* merupakan variabel yang dibahas dalam suatu sistem *fuzzy* seperti umur, temperatur, permintaan dsb. 2. Himpunan

fuzzy, merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Contoh: variabel umur, terbagi atas 3 himpunan *fuzzy*, yaitu: MUDA, PAROBAYA, TUA

c. Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraaan 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 tidak dibatasi batas atasnya. Contoh: Semesta pembicaraan untuk variabel umur: [0 40]
d. Domain adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam himpunan *fuzzy*. (Sri Kusumadewi, 2010)

2.3.4 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah kurva yang mendefinisikan bagaimana masing-masing titik dalam ruang *input* dipetakan ke dalam nilai keanggotaan (derajat keanggotaan) antara 0 dan 1. Fungsi keanggotaan μ memetakan elemen x dari himpunan semesta X, ke sebuah bilangan μ[x], yang menentukan derajat keanggotaan dari elemen dalam himpunan *fuzzy* A.

$$A = \{(x, \mu[x]) \mid x \; X\}$$

Berdasarkan Klir and Bo, kisaran nilai fungsi keanggotaan yang paling umum digunakan adalah interval [0,1]. Dalam hal ini, masing-masing fungsi keanggotaan memetakan elemen-elemen dari himpunan semesta X yang diberikan, yang selalu

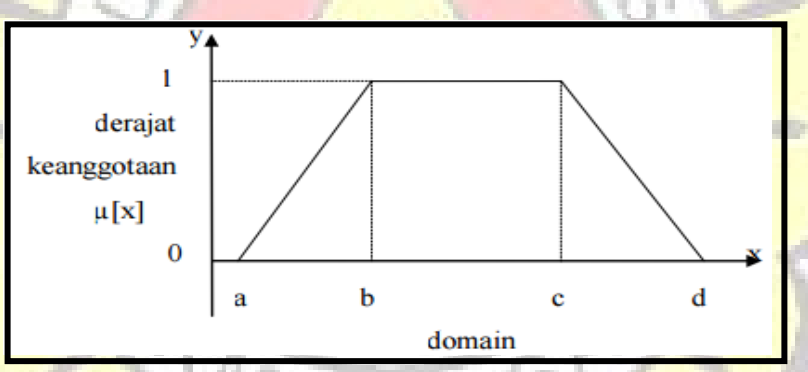
merupakan suatu himpunan *crisp*, ke dalam bilangan nyata dalam interval [0,1] (Arhami, 2005).

Ada beberapa fungsi yang digunakan yaitu sebagai berikut:

a. Representasi Kurva Trapesium Kurva Trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1.

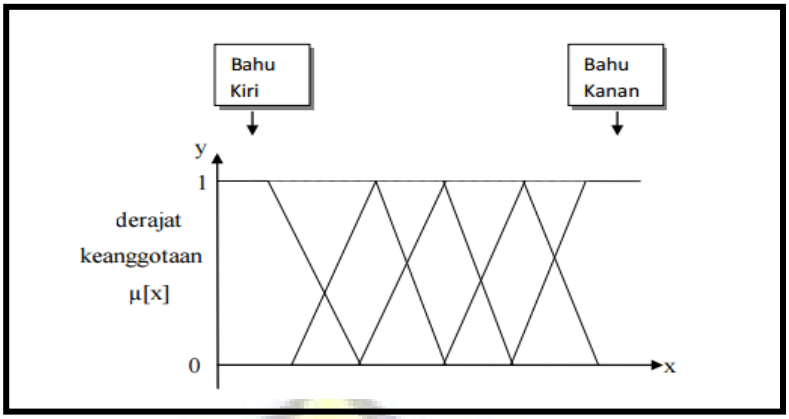
Fungsi keanggotaan:

$$\begin{cases} 0; & x \leq a \; atau \; x \geq d \\ \dfrac{x-a}{b-a}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \dfrac{d-x}{d-c}; & c \leq x \leq d \end{cases}$$



Gambar 2.2 Representasi kurva trapesium (Sri Kusuma Dewi, 2010)

b. Representasi Kurva Bahu Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Himpunan *fuzzy* 'bahu', bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar.



Gambar 2.3 Representasi kurva bahu (Sri Kusuma Dewi, 2010)

2.3.5 Operasi Himpunan *Fuzzy*

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasikan dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan yang dikenal dengan nama *fire strength* atau predikat. Menurut Wang, ada tiga operasi dasar dalam himpunan *fuzzy* ,yaitu *complement*, irisan (*intersection*) dan gabungan (*union*) (Arhami, 2005).

Tabel 2.1 Operasi- operasi dasar dalam himpunan *fuzzy* (Arhami, 2005)

Operasi	Fungsi Keanggotaan
Complement	$\mu_{A'}[x] = 1 - \mu_A[x]$
Intersection	$\mu_{(A \cap B)}[x] = min(\mu_A[x], \mu_B[x])$
Union	$\mu_{(A \cup B)}[x] = max(\mu_A[x], \mu_B[x])$

2.3.6 Sistem Inferensi *Fuzzy*

Sistem inferensi *fuzzy* merupakan kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk IF-THEN, dan penalaran *fuzzy*. 2.2.4.1 Metode Sugeno Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Michio Sugeno mengusulkan penggunaan singleton sebagai fungsi keanggotaan dari konsekuen. *Singleton* adalah sebuah himpunan *fuzzy* dengan fungsi keanggotaan yang pada titik tertentu mempunyai sebuah nilai dan 0 di luar titik tersebut.

Ada 2 model *fuzzy* dengan metode Sugeno yaitu sebagai berikut:

a. Model *Fuzzy* Sugeno Orde-Nol Secara umum bentuk model *fuzzy* Sugeno Orde Nol adalah: IF (x1 is A1) o (x2 is A2) o (x3 is A3) o… o (xN is AN) THEN z=k dengan Ai adalah himpunan *fuzzy* ke-I sebagai antesenden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

b. Model *Fuzzy* Sugeno Orde-Satu Secara umum bentuk model *fuzzy* Sugeno Orde-Satu adalah: IF (x1 is A1) o… o (xN is AN) THEN z = p1*x1+… + pN*xN+q dengan Ai adalah himpunan *fuzzy* ke-i sebagai antesenden, dan pi adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen.

Tahapan-tahapan dalam metode Sugeno yaitu sebagai berikut:

1. Pembentukan himpunan *Fuzzy*

   Pada tahapan ini variabel *input* (*crisp*) dari sistem *fuzzy* ditransfer ke dalam himpunan *fuzzy* untuk dapat digunakan dalam perhitungan nilai kebenaran

dari premis pada setiap aturan dalam basis pengetahuan. Dengan demikian tahap ini mengambil nilai-nilai *crisp* dan menentukan derajat di mana nilai-nilai tersebut menjadi anggota dari setiap himpunan *fuzzy* yang sesuai.

2. Aplikasi fungsi implikasi

   Tiap-tiap aturan (proposisi) pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah sebagai berikut:

   $$IF\ x\ is\ A\ \ THEN\ y\ is\ B$$

   Dengan x dan y adalah skalar, dan A dan B adalah himpunan *fuzzy*. Proposisi yang mengikuti IF disebut sebagai antesenden sedangkan proposisi yang mengikuti THEN disebut konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator *fuzzy* seperti,

   IF(x1 is A1) o (x2 is A2) o (x3 is A3) o…o (xN is AN) THEN y is B dengan o adalah operator (misal: OR atau AND).

   Secara umum fungsi implikasi yang dapat digunakan yaitu sebagai berikut:

   a. Min (minimum)

      Fungsi ini akan memotong *output* himpunan *fuzzy*.

   b. Dot (product)

      Fungsi ini akan menskala *output* himpunan *fuzzy*.

   Pada metode Sugeno ini , fungsi implikasi yang digunakan adalah fungsi min.

3. Defuzzifikasi (*Defuzzification*)

   Input dari proses defuzzifikasi adalah himpunan *fuzzy* yang dihasilkan dari proses komposisi dan *output* adalah sebuah nilai (*crisp*). Untuk aturan IF-

THEN *fuzzy* dalam persamaan RU(k) = IF x1 is A1 k and… and xn is An k THEN y is Bk , dimana A1 k dan Bk berturut-turut adalah himpunan *fuzzy* dalam Ui R (U dan V adalah domain fisik), i = 1, 2, … , n dan x = (x1, x2, … , xn) U dan y V berturut-turut adalah variabel *input* dan *output* (*linguistic*) dari sistem *fuzzy* (Li, 2006).

Menurut Wang, defuzzifier pada persamaan di atas didefenisikan sebagai suatu pemetaan dari himpunan *fuzzy* Bk dalam V R (yang merupakan *output* dari inferensi *fuzzy*) ke titik *crisp* y * V (Arhami, 2005). Pada metode Sugeno defuzzification dilakukan dengan perhitungan *Weight Average* (WA) :

$$WA = \frac{a_1 z_1 + a_2 z_2 + a_3 z_3 + \ldots + a_n z_n}{a_1 + a_2 + a_3 + \ldots + a_n}$$

2.4 Basis Data

Basis data (*database*) adalah suatu kumpulan data yang disusun dalam bentuk tabel-tabel yang saling berkaitan maupun berdiri sendiri dan disimpan secara bersama-sama pada suatu media. Basis data dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya. Terdapat beberapa aturan yang harus dipatuhi pada file basis data agar dapat memenuhi kriteria sebagai suatu basis data, yaitu:

a. Kerangkapan data, yaitu munculnya data-data yang sama secara berulang-ulang pada file basis data,

b. Inkonsistensi data, yaitu munculnya data yang tidak konsisten pada field yang sama untuk beberapa file dengan kunci yang sama,

c. Data terisolasi, disebabkan oleh pemakaian beberapa file basis data. Program aplikasi tidak dapat mengakses file tertentu dalam sistem basis data tersebut, kecuali program aplikasi dirubah atau ditambah sehingga seolah-olah ada file yang terpisah atau terisolasi terhadap file yang lain,

d. Keamanan data, berhubungan dengan masalah keamanan data dalam sistem basis data. Pada prinsipnya file basis data hanya boleh digunakan oleh pemakai tertentu yang mempunya wewenang untuk mengakses,

e. Integrasi data, berhubungan dengan unjuk kerja sistem agar dapat melakukan kendali atau kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh. (Janner Simarmata, 2008)

2.4.1 *Fuzzy Database*

Sistem basis data (*database* system) adalah suatu sistem informasi yang mengintegrasikan kumpulan data yang saling berhubungan dan membuatnya tersedia untuk beberapa aplikasi (Kusumadewi S, Purnomo H, 2010).

Database adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

Fuzzifikasi *query* diasumsikan sebuah *query* konvensional (non*fuzzy*), DBMS yang akan mencoba membuat dan menerapkan sebuah sistem dasar logika *fuzzy query*

(*fuzzy logic based query*ing system). Kelebihan *query* fuzzifikasi yaitu dapat mencapai kelenturan (flexibility) dari DBMS, penanganan *error* otomatis, pencarian yang fleksibel, dan kesanggupan merespon kosong.

Awal penanganan ketidakpastian dengan manajemen basis data dikembangkan di dalam kerangka manajemen sistem basis data yang bukan *fuzzy*. Biasanya, sistem ini berhadapan dengan evaluasi dan konstruksi tentang *fuzzy query* dengan *database* yang bersifat tegas, dan mengabaikan permasalahan dalam penyajian langsung dari data *fuzzy* di DBMS (Mashkuri Hj Yaacob, 1997:43 dalam Setiyowati, M.I, Seta, B.A, 2007).

Sebagian besar basis data *fuzzy* merupakan perluasan dari model basis data relasional, namun dikemas dalam formulasi yang berbeda tergantung pada tipe ambiguitas yang akan diekspresikan dan dimanipulasi. Tahani mendeskripsikan suatu metode untuk melakukan pengolahan *query fuzzy* didasarkan pada manipulasi data. Disini konsep teori *fuzzy* lebih banyak digunakan untuk melakukan pengolahan *query*. Basis data yang diusulkan oleh Zadeh, mengekspresikan ambiguitas data dengan cara memperluas model data. Perluasan dilakukan dengan cara menggunakan relasi *fuzzy* berupa *grade* yang ditambahkan pada relasi standar (Kusumadewi S, 2007).

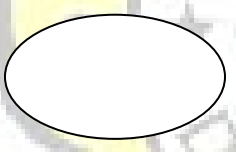
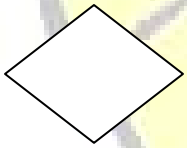
2.4.2 *Entity Relationship* Diagram (ERD)

Entity Relationship Diagram (*ERD*) atau disebut diagram ER (Dr. Sularso Budilaksono, 2009) merupakan diagram yang berfungsi sebagai alat bantu dalam memodelkan rancangan basis data. Diagram ER membantu untuk merancang

keterhubungan antar entitas sehinggga entitas itu dapat dirinci secara logika dalam diagram ER.

Diagram ER memerlukan simbol – simbol sederhana untuk memodelkan rancangan basis data. Simbol – simbol ini diperlukan untuk menyeragamkan diagram ER. Berikut ini adalah simbol atau lambing yang digunakan dalam membuat Diagram ER:

Tabel 2.2 Simbol Diagram ER (Dr. Sularso Budilaksono, 2009)

Simbol	Keterangan
ENTITAS 	Digunakan untuk menggambarkan obyek yang diidentifikasikan ke dalam lingkungan.
ATRIBUT 	Digunakan untuk menggambarkan elemen-elemen dari suatu entity, yang menggambarkan karakter entity.
RELASI 	Entity dapat berhubungan satu sama lain. Hubungan ini disebut dengan relationship.
GARIS	Digunakan untuk menghubungkan entity dengan relasi/hubungan, maupun entity dengan atribut.

2.5 UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*.

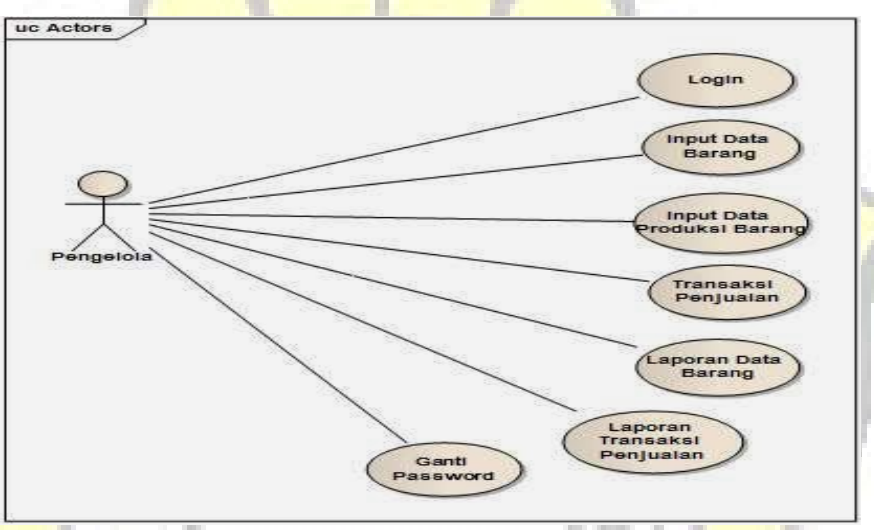
Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

2.5.1 Diagram-Diagram UML (*Unified Modelling Language*)

UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Karena ini merupakan sebuah bahasa, UML mempunyai aturan untuk menggabungkan dan mengkombinasikan elemen-elemen tersebut (Dharwiyanti, 2003).

A. *Usecase* Diagram

Diagram *Usecase* merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing-masing diagram *usecase* menunjukkan sekumpulan *usecase*, aktor, dan hubungannya. Diagram *usecase* adalah penting untuk memvisualisasikan, mespesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. Diagram *usecase* merupakan pusat pemodelan perilaku sistem, subsistem, kelas. (Wahono,2003). Berikut ini adalah contoh *usecase*.

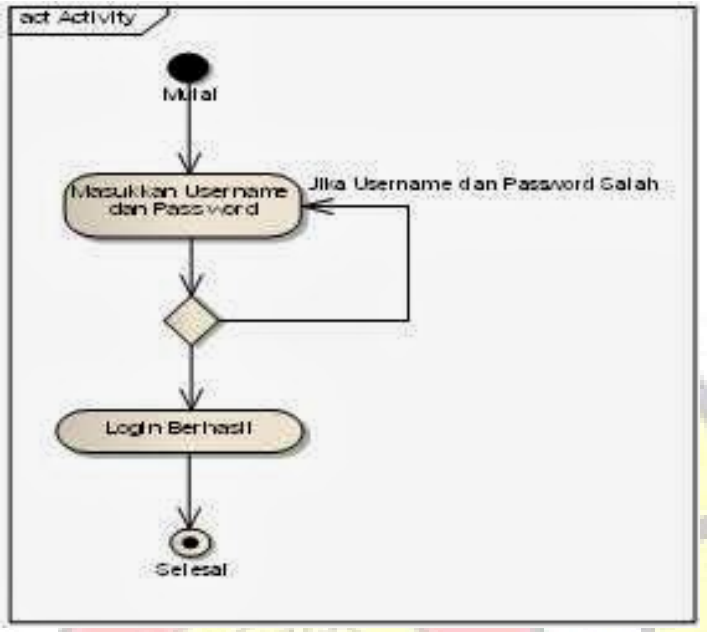


Gambar 2.4 Contoh *usecase* (Wahono, 2003)

B. *Activity* Diagram

Activity diagram menggambarkan berbagai alir aktifitas dalam sebuah 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 tidak menggambarkan sifat internal dari sebuah sistem dan interaksi

antara beberapa sub sistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Berikut ini adalah contoh *activity* diagram.



Gambar 2.5 Contoh *activity* diagram (Wahono, 2003)

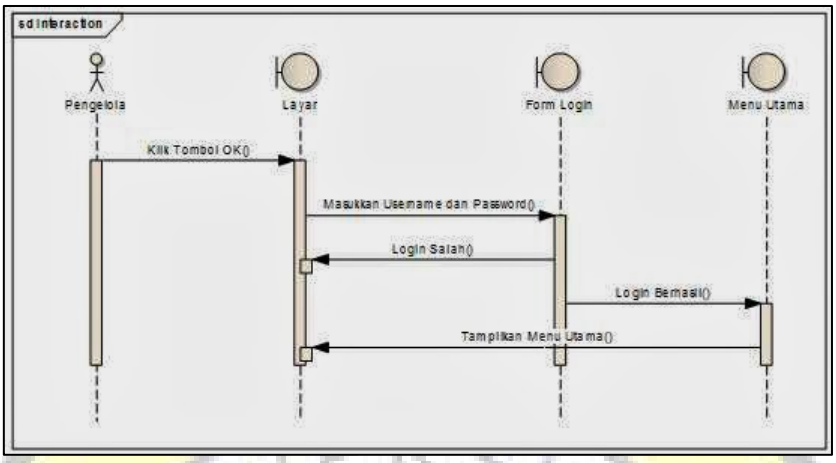
C. *Sequence* Diagram

Sequence Diagram adalah suatu diagram yang digunakan untuk memodelkan skenario penggunaan. Skenario penggunaan adalah barisan kejadian yang terjadi selama satu kesekusi sistem. *Sequence* diagram digunakan untuk :

1. *Overview* perilaku sistem.
2. Menunjukkan objek-objek yang diperlukan.
3. Mendokumentasikan skenario dari suatu *usecase*

4. Memeriksa jalur-jalur pengaksesan

Berikut ini adalah contoh dari *sequence* diagram.



Gambar 2.6 Contoh *sequence* diagram (Wahono, 2003)

UML (*Unified Modeling Language*) adalah bahasa grafis untuk menspesifikasikan, membangun, dan mendokumentasikan sistem perangkat lunak. UML menyediakan diagram-diagram yang sangat kaya dan dapat diperluas sesuai kebutuhan. Diagram adalah representasi grafis elemen-elemen tertentu beserta hubungan-hubungannya. (Hariyanto, 2004).

2.6 Microsoft Visual Basic.Net

Microsoft Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET Framework, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi Windows Forms, Aplikasi web berbasis ASP.NET, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk

lainnya (seperti Microsoft Visual C++, Visual C#, atau Visual J#), atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio .NET. Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas .NET Framework.

2.7 Microsoft SQL *Server* 2012

Pada dasarnya pengertian dari SQL Server itu sendiri adalah bahasa yang dipergunakan untuk mengakses data dalam basis data *relation*. Bahasa ini secara *defacto* adalah bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua *server* basis data yang ada mendukung bahasa ini dalam manajemen datanya. SQL server 2012 merupakan salah satu produk dari *Relational Database Management System* (RDBMS). SQL Server 2012 terdiri atas beberapa komponen sebagai berikut:

a. *Relational Database Engine* : komponen utama atau jantung SQL Server 2005.
b. *Analysis Services* : Basis dari solusi intelijen bisnis yang ampuh (*powerful*) dan mendukung aplikasi-aplikasi OLAP (*online analytical processing*), serta data *minning*.

c. *Data Transformation Service* (DTS): sebuah mesin untuk membuat solusi ekspor dan impor data, serta untuk mentransformasi data ketika data tersebut ditransfer.

d. *Notification Services*: sebuah framework untuk solusi dimana pelanggan akan dikirimi notifikasi ketika sebuah *event* muncul.

e. *Reporting Services* : *service* yang akan mengambil data dari SQL Server, dan menghasilkan laporan-laporan.

f. *Service broker* : sebuah mekanisme antrian yang akan menangani komunikasi berbasis pesan diantara *service*.

g. *Native HTTP Support*: dukungan yang memungkinkan SQL server 2005 yang (jika diinstall pada Windows Server 2003) akan merespon *request* terhadap *HTTP endpoint*, sehingga memungkinkan pembangunan sebuah web *service* untuk SQL *Server* tanpa menggunakan IIS.

h. SQL server Agent : akan mengotomatiskan perawatan *database* dan mengatur *task*, *event* dan *alert*.

i. .NET CLR (*Common Language Runtime*): akan memungkinkan pembuatan solusi menggunakan *managed code* yang ditulis dalam salah satu bahasa .NET.

j. *Replication*: serangkaian teknologi untuk menjalin dan mendistribusikan data dan obyek *database* dari sebuah *database* ke *database* lain, dan melakukan sinkronisasi untuk menjaga konsistensinya.

k. *Full-Text Search*: memungkinkan pengindeksan yang cepat dan flexibel untuk *query* berbasis kata kunci (terhadap data teks yang disimpan dalam *database*).

Dalam hal ini fuzzy database memiliki kaitan dengan SQL Server 2012 adalah sebagai media penyimpanan data-data fuzzy yang diinput melalui aplikasi atau program.