BAB II

LANDASAN TEORI

2.1 Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan.

2.1.1 Konsep Sistem

Suatu sistem pada dasarnya adalah sekolompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Dari defenisi ini dapat dirinci lebih lanjut pengertian sistem secara umu, yaitu :

a. Setiap sistem terdiri dari unsur-unsur
b. Unsur-unsur tersebut merupakan bagian terpadu sistem yang bersangkutan.
c. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem.
d. Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

2.1.2 Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu : tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta

lingkungan. Berikut penjelasan mengenai elemen-elemen yang membentuk sebuah sistem :

a. Tujuan

Setiap sistem memiliki tujuan (*goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

b. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa pelanggan).

c. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lbih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

d. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

e. Batas

Yang disebut batas (*boundary*) sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain. Pertumbuhan sebuah toko kelontong dipengaruhi oleh pembelian pelanggan, gerakan pesaing dan keterbatasan dana dari bank. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem. Sebagai contoh, dengan menjual saham ke publik, sebuah perusahaan dapat mengurangi keterbasatan dana.

f. Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

g. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem.

2.1.3 Klasifikasi Sistem

Beberapa Klasifikasi sistem

a. Sistem Abstrak (*Abstract System*)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. (Contoh : Sistem Teologia).

b. Sistem Fisik (*Physical System*)

Sistem fisik adalah sistem yang ada secara fisik. (Contoh : Sistem Komputer).

c. Sistem Alamiah (*Natural System*)

Sistem alamiah adalah sistem yang terjadi melalui proses alam dan tidak dibuat manusia. (Contoh : Sistem Perputaran Bumi).

d. Sistem Buatan Manusia (*Human Made System*)

Sistem buatan manusia adalah sistem yang dirancang oleh manusia dan melibatkan interaksi antara manusia dengan mesin. (Contoh : Sistem Informasi).

e. Sistem Tertentu (*Deterministic System*)

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi, interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan (Contoh : Sistem Komputer melalui program).

f. Sistem Tak Tentu (*Probabilistic System*)

Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

g. Sistem Tertutup (*Closed System*)

Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya (kenyataannya tidak ada sistem yang benar-benar tertutup), yang ada hanyalah relatively closed system.

h. Sistem Terbuka (*Open System*)

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya, sehingga harus memiliki sistem pengendalian yang baik.

2.2 Aplikasi

Aplikasi dapat dikatakan suatu prangkat lunak yang siap pakai dengan menjalankan intruksi-intruksi dari user atau pengguna, aplikasi banyak diciptakan guna membatu berbagai keperluan seperti untuk laporan, percetakan dan lain.

2.2.1 Konsep Aplikasi

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja *to apply* yang dalam bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu subkelas perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Contoh utama perangkat lunak aplikasi adalah program pengolah kata, lembar kerja, dan pemutar media.

2.2.2 Jenis - Jenis Aplikasi

Jenis-jenis perangkat lunak / aplikasi secara general :

a. Perangkat Lunak System (S*ystem Software*)

   Sistem Operasi yaitu program yang berfungsi untuk mengendalikan sistem kerja yang mendasar sehingga mengatur kerja media input, output, tabel pengkodean, memori, penjadwalan prosesor, dan lain-lain. Sistem operasi berfungsi sebagai penghubung antara manusia dengan perangkat keras dan perangkat lunak yang akan digunakan.

b. Program Lunak Aplikasi (*Application Software*)

   Program Aplikasi adalah perangkat lunak yang dirancang khusus untuk kebutuhan tertentu, misalnya program pengolah kata, mengelola lembar kerja, program presentasi, design grafis, dan lain-lain.

c. Bahasa Pemograman (*Program Language*)

   Perangkat lunak bahasa yaitu program yang digunakan untuk menerjemahkan instruksi-instruksi yang ditulis dalam bahasa

pemrograman ke bahasa mesin dengan aturan atau prosedur tertentu, agar diterima oleh komputer.

d. Program Bantu (*Utility*)

Perangkat Lunak merupakan perangkat lunak yang berfungsi sebagai aplikasi pembantu dalam kegiatan yang ada hubungannya dengan komputer, misalnya memformat disket, mengopi data, mengkompres file, dan lain-lain.

2.2.3 Fungsi atau Manfaat Aplikasi

Fungsi atau manfaat aplikasi secara general :

a. Perangkat Lunak System (S*ystem Software*)

Fungsi utama sistem operasi adalah :

1. Menyimpan program dan aksesnya
2. Membagi tugas di dalam CPU
3. Mengalokasikan tugas-tugas penting

b. Bahasa Pemograman (*Program Language*)

Fungsi dan manfaatnya adalah :

1. Menyusun statemen demi statemen yang mengarah pada skenario pengolahan data.
2. Melakukan *compiler* terhadap susunan statemen yang telah disusun.
3. Membuat program aplikasi akuntansi, perbankan, games permanan, sistem cerdas, dan sebagainya.

c. Program Bantu (*Utility*)

Fungsi dan manfaatnya adalah :

1. Mengatur tata ruang data mendia penyimpanan.
2. Membuat duplikat/ salinan media penyimpanan
3. Melakukan defragmenter
4. Mengontrol unit kerja system.

2.3 Bantuan Siswa Miskin

Program Bantuan Siswa Miskin (BSM) adalah bantuan yang diberikan kepada siswa dari keluarga kurang mampu untuk dapat melakukan kegiatan belajar di sekolah. Bantuan ini memberikan peluang bagi siswa untuk mengikuti pendidikan di level yang lebih tinggi.

2.3.2 Tujuan

Agar siswa dari kalangan tidak mampu dapat terus melanjutkan pendidikan di sekolah. Selain itu juga bertujuan untuk mengurangi jumlah siswa putus sekolah akibat permasalahan biaya pendidikan.

2.3.3 Pemanfaatan

Dana BSM dapat dimanfaatkan untuk:

a. Pembelian perlengkapan siswa (misalnya buku pelajaran, alat tulis, sepatu dan tas).
b. Biaya transportasi siswa ke sekolah / amdrasah
c. Uang saku siswa untuk sekolah

2.3.4 Mekanisme Pembayaran

Mekanisme pembayaran, masing-masing penyelenggara program menggunakan mekanisme yang sedikit berbeda satu dengan yang lain, yaitu sebagai berikut:

a. Direktorat SD menunjuk PT. Pos Indonesia untuk membayar manfaat BSM secara langsung kepada para siswa.
b. Direktorat SD menunjuk PT. Pos Indonesia untuk membayar manfaat BSM secara langsung kepada para siswa.
c. Direktorat SD menunjuk PT. Pos Indonesia untuk membayar manfaat BSM secara langsung kepada para siswa.

Untuk Kemenag, manfaat BSM disalurkan kepada Madrasah Negeri maupun Swasta. Untuk Madrasah Swasta, Kemenag menunjuk PT Pos Indonesia untuk menyalurkan manfaat BSM kepada siswa secara langsung, sementara untuk Madrasah Negeri, dana BSM langsung disalurkan ke masing-masing rekening bank Madrasah. Tabel di bawah ini memperlihatkan rincian pagu 2012 dan pagu yang direncanakan untuk 2013 berdasarkan informasi yang disediakan oleh kedua Kementerian.

2.4 Fuzzy

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang *output*. Titik awal dari konsep modern mengenai ketidakpastian adalah paper yang dibuat oleh Lofti A Zadeh(1965), dimana Zadeh memperkenalkan teori yang memiliki obyek-obyek dari himpunan *fuzzy* yang memiliki batasan yang tidak presisi dan keanggotaan dalam himpunan *fuzzy*,

dan bukan dalam bentuk logika benar (*true*) atau salah (*false*), tapi dinyatakan dalam derajat (*degree*). Konsep seperti ini disebut dengan *Fuzziness* dan teorinya dinamakan *Fuzzy Set Theory*. *Fuzziness* dapat didefinisikan sebagai logika kabur berkenaan dengan semantik dari suatu kejadian, fenomena atau pernyataan itu sendiri. Seringkali ditemui dalam pernyataan yang dibuat oleh seseorang, evaluasi dan suatu pengambilan keputusan. Sebagai contoh:

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.
2. Pelayan restoran memberikan pelayanan terhadap tamu, kemudian tamu akan memberikan tip yang sesuai atas baik tidaknya pelayanan yang diberikan.
3. Anda mengatakan pada saya seberapa sejuk ruangan yang anda inginkan, saya akan mengatur putaran kipas yang ada pada ruangan ini.

Fuzzy system (sistem kabur) didasari atas konsep himpunan kabur yang memetakan domain *input* kedalam domain *output*. Perbedaan mendasar himpunan tegas dengan himpunan kabur adalah nilai keluarannya. Himpunan tegas hanya memiliki dua nilai *output* yaitu nol atau satu, sedangkan himpunan kabur memiliki banyak nilai keluaran yang dikenal dengan nilai derajat keanggotaannya.

Logika *fuzzy* adalah peningkatan dari logika *Boolean* yang berhadapan dengan konsep kebenaran sebagian. Dimana logika klasik (*crisp*) menyatakan bahwa segala hal dapat diekspresikan dalam istilah *binary* (0 atau 1,

hitam atau putih, ya atau tidak). Logika *fuzzy* menggantikan kebenaran Boolean dengan tingkat kebenaran. Logika *fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk *linguistic*, konsep tidak pasti seperti "sedikit", "lumayan", dan "sangat". Logika ini diperkenalkan oleh Dr. Lotfi Zadeh dari Universitas California, Barkeley pada tahun 1965. Logika *fuzzy* telah digunakan pada bidang-bidang seperti taksonomi, topologi, linguistik, teori automata, teori pengendalian, psikologi, *pattern recognition*, pengobatan, hukum, decision analysis, *system theory and information retrieval*. Pendekatan *fuzzy* memiliki kelebihan pada hasil yang terkait dengan sifat kognitif manusia, khususnya pada situasi yang melibatkan pembentukan konsep, pengenalan pola, dan pengambilan keputusan dalam lingkungan yang tidak pasti atau tidak jelas.

Ada beberapa alasan mengapa orang menggunakan logika fuzzy antara lain:

1. Konsep logika *fuzzy* mudaah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi *nonlinear* yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengimpikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.

6. Logika *fuzzy* dapat bekerjasama dengan teknik- teknik kendali secara konvensional
7. Logika *fuzzy* didasarkan pada bahasa alami.

2.4.1 Konsep Logika Fuzzy

Teori logika *fuzzy* yang diajukan oleh Zadeh pada pertengahan tahun 1960 memberikan suatu pemecahan masalah terhadap persoalan yang tidak pasti ini. Sehingga sistem informasi yang akan dibuat menggunakan model DBMS dan *query* yang berbasis *fuzzy* karena model DBMS konvensional, non *fuzzy* kurang dapat memenuhi kebutuhan sistem informasi ini. Banyak model DBMS dan query fuzzy yang ada, salah satunya adalah model Tahani yang ditemukan pada tahun 1977. Prof. Lutfi Zadeh berpendapat bahwa logika benar dan salah dari logika boolean/konvensional tidak dapat mengatasi masalah gradasi yang ada di dunia nyata. Untuk mengatasi masalah gradasi tersebut maka ia mengembangkan sebuah himpunan samar (*fuzzy*).

2.4.2 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan μA[x], memiliki 2 kemungkinan (Kusumadewi S, Purnomo H, 2010) yaitu:

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval [0,1], namun interprestasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan bernilai suatu himpunan *fuzzy* USIA adalah 0,9; maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,9 usia berarti 10% dari himpunan tersebut diharapkan tidak muda.

Himpunan *fuzzy* memiliki 2 atribut yaitu :

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA.
2. Numeris, yaitu suatu nilai (angka) yang menunjukan ukuran dari suatu variable seperti : 40,25, 50 dsb..

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy (Aplikasi logika *fuzzy* untuk pendukung keputusan, Sri KusumadewiGraha Ilmu, 2010), yaitu:

a. Variabael *fuzzy*

   Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: umur, temperature, permintaan, dsb.

b. Himpunan *fuzzy*

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

Contoh :

a. Variable mahasiswa, terbagi menjadi 5 himpunan *fuzzy*, yaitu kurang sekali, kulang, cukup, baik dan baik sekali.
b. Variable dosen, terbagi menjadi 3 himpunan *fuzzy*, yaitu : cukup, baik, dan baik sekali.
c. Semesta Pembicara

   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. Ada kalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya

   Contoh :

   1. Semesta Prmbicaraan untuk variable mahasiswa : [0 50]
   2. Semesta pembicaraan untuk variable dosen [0 50]

d. 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 dan bilangan negatif.

Contoh domain himpunan *fuzzy*:

a. Kurang sekali [0 15]

b. Kurang [5 25]

c. Cukup [15 35]

d. Baik [25 45]

e. Baik Sekali [35 50]

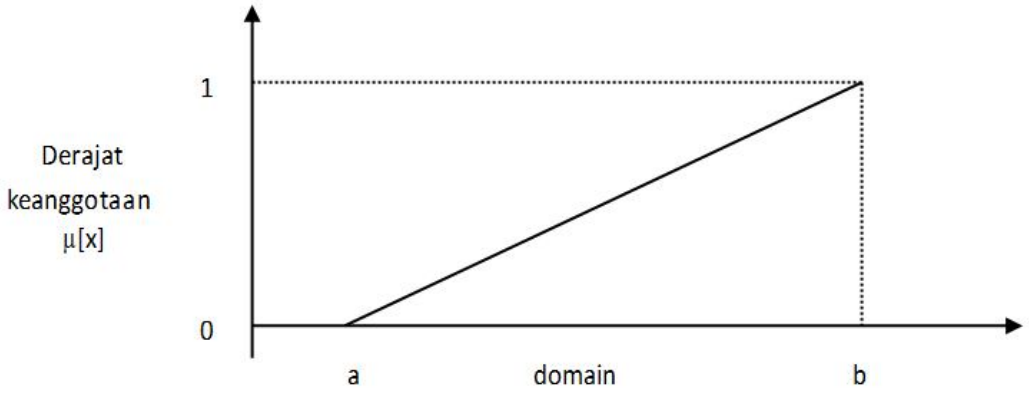
2.4.3 Fungsi Keanggotaan

Fungsi keanggotan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Apabila U menyatakan himpunan universal dan A adalah himpunan fungsi *fuzzy* dalam U, maka A dapat dinyatakan sebagai pasangan terurut (Wang, 1997 dari Wulandari, F., 2005). Ada beberapa fungsi yang bisa digunakan.

a. Representasi Linear

   Pada representasi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada 2 keadaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaanya nol(0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi .



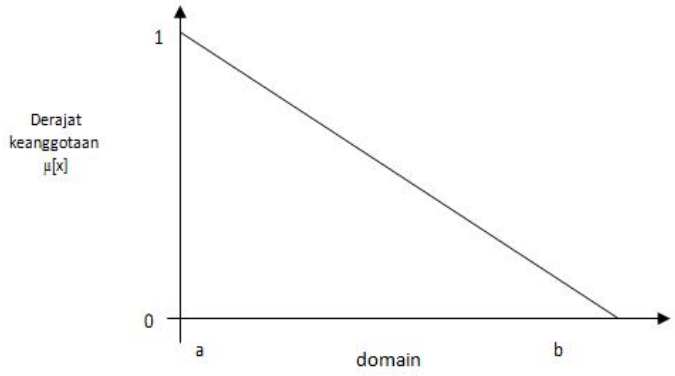
Gambar 2.1 Representasi linear Naik

Sumber : Sri Kusumadewi, Sri Hartati, Agus Harjoko, dan Retantyo Wardoyo. *Fuzzy Multi-Attribute Decision Making (FUZZY MADM)*. (2006:9)

Fugnsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x-a) / (b-a); & a<x<b \\ 1; & x \geq b \end{cases}$$

Kedua, merupakan kebalikan dari 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. Seperti terlihat pada gambar 2.2



Gambar 2.2 Representasi linier turun

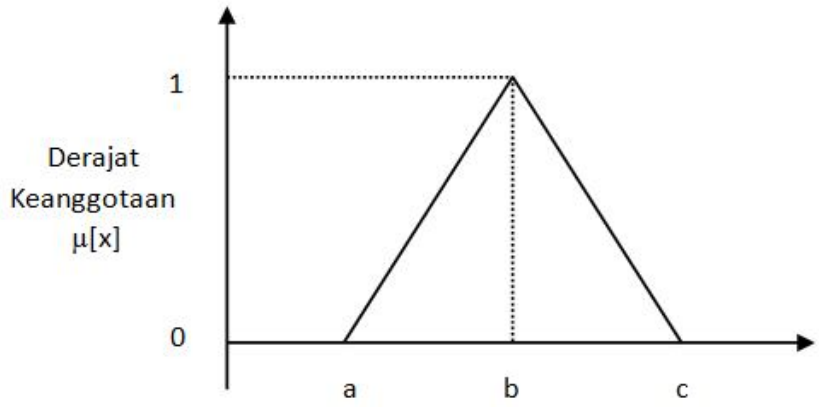
Sumber : Sri Kusumadewi, Sri Hartati, Agus Harjoko, dan Retantyo Wardoyo. *Fuzzy Multi-Attribute Decision Making (FUZZY MADM)*. (2006:10)

Fungsi Keanggotaan

$$\mu[x] = \begin{cases} 0; & x \geq b \\ (b-x) / (b-a); & a<x<b \\ 1; & x \leq a \end{cases}$$

b. Representasi kurva segitiga

Kurva segitiga pad dasarnya merupakan gabungan antara 2 garis (linear). Seperti terlihat pada gambar 2.3



Gambar 2.3 Kurva Segitiga

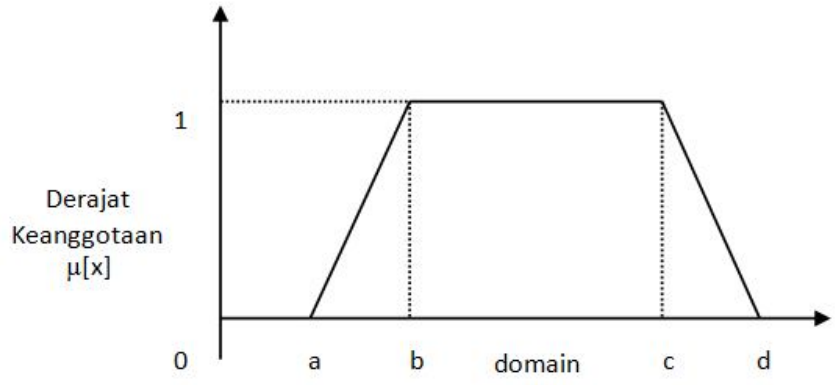
Sumber : Sri Kusumadewi, Sri Hartati, Agus Harjoko, dan Retantyo Wardoyo. *Fuzzy Multi-Attribute Decision Making (FUZZY MADM)*. (2006:11)

File keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \geq c \text{ ata } x \leq a \\ (x-a) / (b-a); & a \leq x \leq b \\ (c-x) / (c-b); & b < x < c \end{cases}$$

c. Representase kurva trapesium

(Kusmadewi S, Purnomo H, 2010) Kurva trapesium pada daasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan 1. Seperti terlihat pada gambar 2.4.



Gambar 2.4 Representasi kurva trapezium

Sumber : Sri Kusumadewi, Sri Hartati, Agus Harjoko, dan Retantyo Wardoyo. *Fuzzy Multi-Attribute Decision Making (FUZZY MADM)*. (2006:12)

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \geq d \text{ atau } x \leq a \\ (x-a) / (b-a); & a<x<b \\ (d-x) / (d-c); & c<x<d \\ 1; & b \leq x \leq c \end{cases}$$

2.4.4 Multi Attribute Decision Making

Multiple Criteria Decision Making (MCDM) adalah suatu metode pengambilan keputusan untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Kriteria biasanya berupa ukuran-ukuran, aturan-aturan atau standar yang digunakan dalam pengambilan keputusan. Berdasarkan tujuannya, MCDM dapat dibagi menjadi 2 model (Zeimmermann, 1991): *Multi Attribute Decision Making* (MADM); dan *Multi Objective Decision Making* (MODM).

MADM digunakan untuk menyeleksikan masalah-masalah dalam ruang diskret, sedangkan MODM digunakan untuk menyelesaikan masalah-masalah pada ruang kontinyu (seperti permasalahan pada pemograman matematis). Secara umum dapat dikatakan bahwa, MADM menyeleksi alternatif terbaik dari sejumlah alternatif, sedangkan MODM merancang alternatif terbaik. Perbedaan mendasar terlihat pada Tabel 2.1.

Tabel 2.1 Perbedaan antara MADM dan MODM

Sumber : Sri Kusumadewi, Sri Hartati, Agus Harjoko, dan Retantyo Wardoyo. *Fuzzy Multi-Attribute Decision Making (FUZZY MADM)*. (2006:70)

	MADM	MODM
Kriteria (didefinisikan oleh)	Atribut	Tujuan
Tujuan	Implisit	Eksplisit
Atribut	Eksplisit	Implisit
Alternatif	Diskret, dalam jumlah terbatas	Kontinu, dalam jumlah tak terbatas
Kegunaan	Seleksi	Desain

2.4.4.1 Klasifikasi Metode MCDM

Ada beberapa cara dalam mengklasifikasikan metode MCDM. Menurut tipe data yang digunakan, MCDM dapat dibagi berdasarkan tipe data deterministik, stokastik atau fuzzy. Menurut jumlah pengambilan keputusan yang terlibat dalam proses pengambilan keputusan. MCDM dapat dibagi berdasarkan pengambil keputusan. MCDM dapat dibagi berdasarkan pengambilan keputusan satu orang atau pengambilan keputusan dalam bentuk group (kelompok).

2.4.4.2 Klasifikasi Solusi MCDM

Masalah MCDM tidak selalu memberikan suatu solusi unik, perbedaan tipe bisa jadi akan memberikan perbedaan solusi (Yoon, 1981).

a. Solusi ideal, kriteria atau atribut dapat dibagi menjadi 2 kategori, yaitu kriteria yang nilainya akan dimaksimumkan (kategori kriteria keuntungan), dan kriteria yang nilainya akan diminimumkan (kategori kriteria biaya). Solusi ideal akan memaksimumkan semua kriteria keuntungan dan meminimumkan semua kriteria biaya.
b. Solusi non-dominated, solusi ini sering juga dikenal dengan nama solusi Pareto-optimal. Solusi feasible MCDM dikatakan non-dominated jika tidak ada solusi feasible yang lain yang akan menghasikan perbaikan terhadap suatu atribut tanpa menyebabkan degenerasi pada atribut lainnya.
c. Solusi yang memuaskan, solusi yang memuaskan adalah himpunan bagian dari solusi-solusi feasible dimana setiap alternatif dimana setiap alternatif melampaui semua kriteria yang diharapkan.
d. Solusi yang lebih disukai, solusi yang disukai adalah solusi non-dominated yang paling banyak memuaskan pengambil keputusan.

2.4.5 Konsep Dasar Multi-Attribute Decision Making (MADM)

Multiple Attribute Decision Making (FMADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan.

Pada dasarnya, ada tiga pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif dan pendekatan integrasi antara subyektif dan obyektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif dapat ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan (Kusumadewi, 2007).

2.4.6 Metode Fuzzy Multi-Attribute Decision Making

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah FMADM, antara lain (Kusumadewi, 2006) :

1. *Simple Additive Weighting* (SAW)
2. *Weighted Product* (WP)
3. ELECTRE
4. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)
5. *Analytic Hierarchy Process* (AHP)

Algoritma *Fuzzy Multip Attribute Decision Making* adalah :

1. Memberikan nilai setiap alternatif (Ai) pada setiap kriteria (Cj) yang sudah ditentukan, dimana nilai tersebut diperoleh berdasarkan nilai *crisp*; i=1,2,…m dan j=1,2,…n.
2. Memberikan nilai bobot (W) yang juga didapatkan berdasarkan nilai *crisp*.

3. Melakukan normalisasi matriks dengan cara menghitung nilai rating kinerja ternormalisasi (rij) dari alternatif (Ai) pada atribut (Cj) berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan / benefit = MAKSIMUM atau atribut biaya / cost = MINIMUM). Apabila berupa atribut keuntungan maka nilai crisp (xij) dari setiap kolom atribut dibagi dengan nilai crisp MAX (MAX xij) dari tiap kolom, sedangkan untuk atribut biaya, nilai crisp MIN (MIN xij) dari tiap kolom atribut dibagi dengan nilai crisp (xij) setiap kolom.
4. Melakukan proses perankingan dengan cara mengalikan matriks ternormalisasi (R) dengan nilai bobot (W).
5. Melakukan nilai preferensi untuk setiap alternatif (Vi) dengan cara menjumlahkan hasil kali antara matriks ternormalisasi (R) dengan nilai bobot (W). Nilai Vi yang lebih besar mengindikasikan bahwa alternatif lebih terpilih (Kusumadewi, 2007).

2.5 Konsep Permograman Berorientasi Objek

Konsep ini membagi program menjadi objek-objek yang saling berinteraksi satu sama lain.

Objek adalah kesatuan entitas yang memiliki sifat dan tingkah laku. Dalam kehidupan sehari-hari, objek adalah benda, baik benda berwujud nyata seperti manusia, hewan, mobil, komputer, handphone, pena, ataupun benda yang tidak nyata atau konsep, seperti tabungan bank, sistem antrian, sistem internet banking, dan sebagainya.

Keuntungan OOP :

1. Alami (*natural*)
2. Dapat diandalkan (*reliable*)
3. Dapat digunakan kembali (*reusable*)
4. Mudah untuk di-maintain (*maintable*)
5. Dapat diperluas (*extendable*)
6. Efisiensi waktu

Berikut ini beberapa bahasa pemograman yang menggunakan konsep OOP :

1. C++
2. Visual C++
3. Delphi
4. Visual basic
5. Java

2.6 JAVA

Java merupakan bahasa pemograman yang *multi platform* dan *multi device*. Aplikasi dengan berbasis Java ini dikompulasikan ke dalam p-code dan bisa dijalankan dengan *Jaa Virtual Mechine*.

2.6.1 Sejarah Java

Sejarah java berawal pada tahun 1991 ketika perusahaan *Sun Microsystem* memulai *green project*, yakni projek penelitian untuk membuat bahasa yang akan digunakan pada chip-chip *embedded* untuk *device intellegent consumer electronic*. Bahasa tersebut haruslah bersifat *platform,* tidak tergantung kepada vendor yang manufaktur *chip* tersebut.

Dalam penelitiannya, *project green* berhasil membuat *prototype* semacam PDA (*Personal Data Asistance*) yang dapat berkomunikasi antara satu dengan yang lain dan diberi nama *star* 7. Ide berawal untuk membuat sistem operasi bagi *star 7* berbasis C dan C++. Setelah berjalan beberapa lama. James Gosling, salah seorang anggota team, merasa kurang puas dengan beberapa karakteristik dari kedua bahasa tersebut berusaha mengembangakan bahasa lain. Bahasa tersebut kemudian di namakan Oak, diinspirasi ketika melihat pohon di seberang kaca ruang kantornya. Belakangan Oak beralih nama menjadi Java.

Karena pada awalnya ditujukan untuk pemograman device kecil, Java memiliki karakteristik berukuran kecil, efisien, dan *portable* untuk berbagai *hardware*. Projek *green* sempat terancam terhenti karena dalam perkembangannya, device ini belum memiliki pasar seperti yang diramalkan semula. Selanjutnya java diarahkan untuk pemograman interntet. Secara kebetulan fitur-fitur java yang telah disebutkan sebelumnya sangat sesuai bagi pengembangan internet sehingga dalam beberapa tahun belakangan ini Java telah menjadi primadona untuk pemogaman yang berbasis internet.

2.6.2 Keunggulan Java

Java memiliki beberapa keunggulan bila dibandingkan dengan bahasa pemograman lainnya. Ada beberapa aspek, yaitu :

a. Java bersifat sederhana dan relatif mudah

   Java dimodelkan sebagian dari bahasa C++, namun dengan memperbaiki beberapa karakteristik C++, seperti mengurangi kompleksitas beberapa

fitur, penambahan funsionalitas, serta penghasilan beberapa aspek pemicu ketidakstabilan sistem pada C++

b. Java berorientasi pada objek (*Object Oriented*)

Java adalah bahasa pemograman yang bersifat objek (OOP), bukan seperti pascal, basic, C yang berbasis prosedural. Dalam memecahkan masalah, java membagi program menjadi objek-bjek, kemudian memodelkan sifat dan tingkah laku masing-masing. Selanjutnya, java menentukan dan mengatur interaksi antara objek yang satu dengan lainnya.

c. Java bersifat terdistribusi

Pada dekade awal perkembangan PC (*Personal Computer*), komputer lainnya bersifat sebagai workstation tunggal, tidak terhubung satu sama lain. Saat omo, sistem komputerasisasi cenderung terdistribusi, mulai dari *workstation client*, *email server, database server, web server, proxy server,* dan sebagainya.

d. Java bersifat multiplatform

Pada umumnya, program yang dibuat dan dikompile di suatu platform hanya bisa dijalankan di *platform* tersebut. Java bersifat *multi-platform*, yakni dapat di terjemahkan oleh Java *Interpreter* pada berbagai operasi.

e. Java bersifat MultiThread

Thread adalah proses yang dapat dikerjakan oleh program dalam suatu waktu. Java bersifat *Multithreaded,* artinya dapat mengerjakan beberapa proses dalam waktu yang hampir bersamaan.

2.7 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat mendownload langsung dari web resminya.

2.7.1 Sejarah XAMPP

XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan *project* non-profit yang di kembangkan oleh *Apache Friends* yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang pada tahun 2002, *project* mereka ini bertujuan mempromosikan pengunaan *Apache web server*.

2.7.2 Manfaat XAMPP

Manfaat atau kegunaan XAMPP adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

2.7.3 Kelebihan XAMPP

Kelebihan :

1. Bisa berperan sebagai server web Apache untuk simulasi pengembangan website. Tool pengembangan web ini mendukung teknologi web populer seperti PHP, MySQL, dan Perl.

2. Melalui program ini, programmer web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet.
3. XAMPP juga dilengkapi fitur manajemen database PHPMyAdmin seperti pada server hosting sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis database secara mudah.
4. XAMPP dapat dijalankan di sistem operasi Windows 2000/XP/Vista/7 dan sistem operasi lain.

2.8 Netbeans

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah!). *Sun Microsystems* mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama.

Saat ini terdapat dua produk : NetBeans IDE dan NetBeans Platform

NetBeans IDE adalah sebuah lingkungan pengembangan - sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan.

2.9 Ireport

iReport adalah visual tool untuk membuat *reports, file* hasil dari iReport adalah jrxml, *file* xml yang berisi content desain dari report nya. Jasper Report

adalah *engine* yang menghasilkan xml tersebut menjadi bisa dilihat, di *print* atau di convert ke dokumen format yang lain seperti pdf, openoffice doc, rtf. Dan servlet adalah web yang berbasis kan class java. Ini saya maunya menjeleskan cara merelasikan ketiga nya ini.