



**IMPLEMENTASI METODE *TSUKAMOTO* DALAM SISTEM  
PENDUKUNG KEPUTUSAN UNTUK OPTIMASI JUMLAH  
PENGADAAN LAMPU LED**

skripsi

disajikan sebagai salah satu syarat  
untuk memperoleh gelar Sarjana Sains  
Program Studi Matematika

oleh  
Sri Hartatik  
4111410007

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI SEMARANG  
2017**

## PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, dan apabila dikemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, Agustus 2017



**UNNES**  
UNIVERSITAS NEGERI SEMARANG

## PENGESAHAN

Skripsi yang berjudul

Implementasi Metode *Tsukamoto* Dalam Sistem Pendukung Keputusan Untuk  
Optimasi Jumlah Pengadaan Lampu LED

disusun oleh

Sri Hartatik  
4111410007

Telah dipertahankan dihadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada  
tanggal 25 Agustus 2017.



Prof. Dr. Zaenuri, S.E, M.Si, Akt  
NIP. 196412231988031001

Sekretaris

Drs. Arief Agoestanto, M.Si  
NIP. 196807221993031005

Penguji I

Drs. Arief Agoestanto, M.Si  
NIP. 196807221993031005

Penguji II

Dr. Scolastika Mariani, M.Si  
NIP. 196502101991022001

Anggota Penguji/  
Pembimbing

Endang Sugiharti, S.Si., M.Kom  
NIP. 197401071999032001

## MOTTO DAN PERSEMBAHAN

### Motto

- *Bukanlah hidup kalau tidak ada masalah, bukanlah sukses kalau tidak melalui rintangan, bukanlah menang kalau tidak dengan pertarungan, bukanlah lulus kalau tidak ada ujian, dan bukanlah berhasil kalau tidak berusaha.*
- *Jika ingin mengubah derajat dunia harus dengan USAHA  
Jika ingin mengubah derajat kita di akhirat tingkatkan IMAN  
Jadilah yang lembut itu HATI, yang tipis itu BUDI, Yang tebal itu IMAN, yang tajam itu AKAL, Yang baik itu SIFAT dan yang manis itu SENYUMAN.*

### Persembahan

*Skripsi ini kupersembahkan untuk:*

- *Ibuku Rumini dan Bapakku Achmad Sutar atas kasih sayang, dukungan semangat dan doa yang tak hentinya dipanjatkan.*
- *Adikku Titin Fatimah dan mas Fitri Antoro yang senantiasa menemani dan menghiburku.*
- *Ruli, Istatik, Novi, Fitra, Michel, Eva, Putri Aida dan teman-teman Lollipop yang selalu memberi semangat dan selalu ada untukku saat senang maupun sedih.*
- *Teman-teman Proxima yang senantiasa berjuang bersama di UNNES tercinta.*
- *Almamaterku Universitas Negeri Semarang.*

## KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kehadirat Allah SWT atas segala rahmat, taufik dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Implementasi metode *tsukamoto* dalam sistem pendukung keputusan untuk optimasi jumlah pengadaan lampu LED**”.

Dalam penulisan skripsi ini penulis mendapat banyak bantuan dari berbagai pihak. Untuk itu perkenankanlah penulis mengucapkan terima kasih kepada:

1. Prof. Dr. Fathur Rokhman, M. Hum., Rektor Universitas Negeri Semarang.
2. Prof. Dr. Zaenuri, S.E, M.Si,Akt, Dekan FMIPA Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika FMIPA Universitas Negeri Semarang.
4. Drs. Mashuri, M.Si., Koordinator Prodi Matematika FMIPA Universitas Negeri Semarang.
5. Endang Sugiharti, S.Si., M.Kom., Dosen pembimbing yang telah memberikan bimbingan, motivasi dan semangat.
6. Ibu, Bapak dan adik tercinta yang selalu memberikan doa, semangat serta dukungan baik secara moral maupun spiritual.
7. Seluruh pihak di Toko KERTA Semarang yang telah memberikan izin hal pengambilan data.

## ABSTRAK

Hartatik,Sri. 2017. **Implementasi metode *tsukamoto* dalam sistem pendukung keputusan untuk optimasi jumlah pengadaan lampu led.** Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Endang Sugiharti S.Si, M.Kom.

Kata kunci : Optimasi, *Tsukamoto* , Sistem Pendukung Keputusan , *Software* MATLAB , Pengadaan, Lampu LED.

Sistem Pendukung Keputusan (SPK) yang dikembangkan bertujuan untuk menentukan jumlah lampu LED yang akan diadakan berdasarkan data persediaan dan jumlah permintaan. SPK ini menggunakan Sistem Inferensi *Fuzzy* atau *Fuzzy Inference System (FIS) Tsukamoto*. *Software* yang digunakan adalah MATLAB. Optimasi jumlah pengadaan lampu LED dilakukan dengan menggunakan tiga variabel, yaitu permintaan, persediaan dan pengadaan. Variabel permintaan terdiri dari tiga himpunan *fuzzy*, yaitu : sedikit, sedang dan banyak, variabel persediaan terdiri dari tiga himpunan *fuzzy*, yaitu : menurun, tetap dan meningkat, sedangkan variabel pengadaan terdiri dari tiga himpunan *fuzzy*, yaitu berkurang, normal dan bertambah. Dengan mengkombinasikan semua himpunan *fuzzy* tersebut, diperoleh sembilan aturan *fuzzy*, yang selanjutnya digunakan dalam setiap inferensi. Pada tahap inferensi, dicari nilai keanggotaan anteseden ( $\alpha$ ) dan nilai optimasi perkiraan pengadaan ( $z$ ) dari setiap aturan. Optimasi pengadaan barang ( $z$ ) dicari dengan metode defuzzifikasi rata-rata terpusat. Analisa dengan menggunakan metode *Tsukamoto* menghasilkan kondisi optimum pengadaan lampu LED mencapai 290 buah, ini mendekati perhitungan yang dihasilkan oleh Toko KERTA tanpa menggunakan metode *Tsukamoto*, yaitu 300. Analisa dengan menggunakan metode *Tsukamoto* ini memiliki tingkat validitas yang sangat baik yaitu mencapai 76% sehingga memperlihatkan kondisi rill yang harus dijalankan pihak penjual lampu LED di Toko KERTA dalam melakukan proses pengadaan barang supaya lebih tepat sasaran.

## DAFTAR ISI

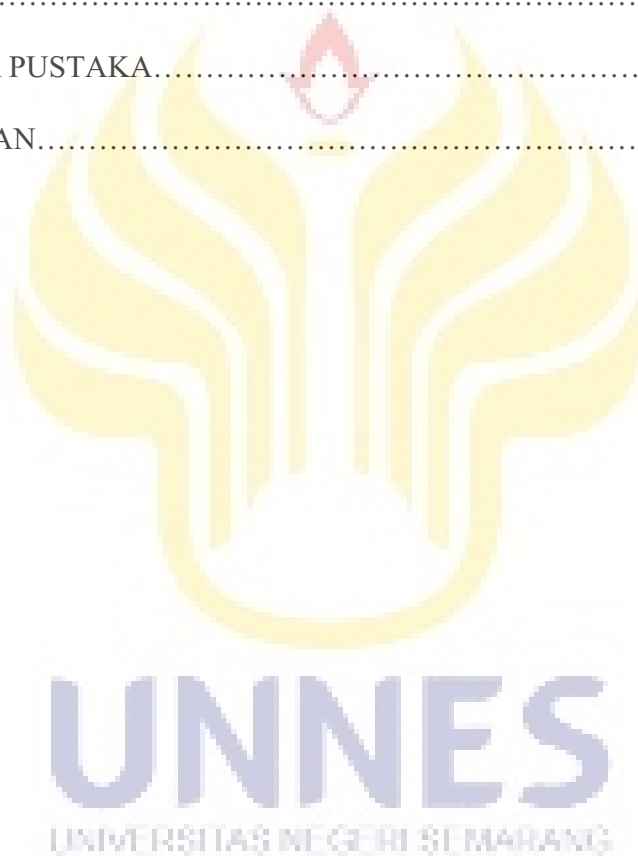
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
<b>BAB</b>	
<b>I PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	4
1.3 Batasan Masalah.....	5
1.4 Tujuan dan Manfaat	
1.4.1 Tujuan Penulisan.....	5
1.4.2 Manfaat Penulisan.....	6
1.5 Sistematika Penulisan	
1.5.1 Bagian Awal.....	6
1.5.2 Bagian Pokok.....	6
1.5.3 Bagian Akhir.....	7
<b>II LANDASAN TEORI</b>	

2.1 Himpunan dan Logika <i>Fuzzy</i> .....	8
2.2 Fungsi keanggotaan.....	11
2.3 Teori Operasi Himpunan.....	15
2.4 Metode <i>Fuzzy Inference System Tsukamoto</i> .....	16
2.5 Metode <i>Tsukamoto</i> .....	18
2.6 Fungsi Keanggotaan.....	19
2.7 Operator Dasar Untuk Himpunan <i>Fuzzy</i> .....	22
2.8 Fungsi Implikasi.....	23
2.9 Optimasi.....	24
2.10 Sistem Pendukung Keputusan (SPK).....	24
2.11 MATLAB ( <i>Matrix Laboratory</i> )	
2.11.1 Mengenal MATLAB.....	25
2.11.1.1 Sekilas Tentang MATLAB.....	25
2.11.1.2 Menjalankan MATLAB.....	26
2.11.1.3 <i>Command Window</i> .....	27
2.11.1.4 Kesalahan Perintah.....	27
2.11.1.5 Menggunakan Variabel.....	28
2.11.1.6 Mengenal Variabel Khusus.....	29
2.11.1.7 Mengenal <i>Workspace</i> .....	30
2.11.2 Operasi <i>Array</i> .....	31
2.11.2.1 Dasar-Dasar <i>Array</i> .....	31
2.11.2.2 Operasi <i>Array</i> .....	32



2.11.2.3 Memanipulasi <i>Array</i> .....	37
2.11.3 Mengenal GUI	
2.11.3.1 Memanfaatkan GUIDE.....	40
2.11.3.2 Mengenal <i>Toolbar</i> .....	41
2.11.3.3 Merancang <i>Interface</i> .....	42
2.11.3.4 Menyimpan <i>Interface</i> .....	44
2.11.3.5 Menjalankan <i>Interface</i> .....	45
III METODE PENELITIAN	
3.1 Studi Pustaka.....	46
3.2 Perumusan Masalah.....	47
3.3 Pemecahan Masalah	
3.3.1 Pengumpulan dan Pengolahan Data.....	47
3.3.2 Perancangan, Pembuatan dan Pengujian Data.....	48
3.4 Penarikan Kesimpulan.....	48
IV HASIL DAN PEMBAHASAN	
4.1 Hasil.....	50
4.1.1 Model Base Metode <i>Tsukamoto</i> .....	50
4.1.2 Mendefinisikan Variabel <i>Fuzzy</i> .....	50
4.2 Inferensi.....	69
4.3 Menentukan <i>Output Crisp</i> .....	77
4.4 Perancangan Aplikasi Menggunakan MATLAB.....	78
4.5 Perhitungan Manual Metode <i>Tsukamoto</i> .....	82

4.6 Pembahasan.....	94
4.6.1 Perhitungan Jumlah Pengadaan Diselesaikan dengan SPK.....	94
<b>V PENUTUP</b>	
5.1 Simpulan.....	100
5.2 Saran.....	102
<b>DAFTAR PUSTAKA</b> .....	104
<b>LAMPIRAN</b> .....	107



## DAFTAR TABEL

## TABEL

2.1 Daftar Variabel Khusus pada MATLAB.....	29
2.2 Daftar Toolbar pada GUIDE MATLAB.....	41
4.1 Variabel-variabel Dalam Perhitungan <i>Tsukamoto</i> .....	69
4.2 Data Lampu LED.....	82
4.3 Data Minimum dan Maksimum.....	82
4.4 Hasil Uji Validitas SPK.....	95
4.5 Data Maksimum Dan Minimum.....	96
4.6 Data Maksimum Dan Minimum.....	97
4.7 Perbandingan Perhitungan Jumlah Pengadaan <i>Tsukamoto</i> dengan Data Pengadaan Toko KERTA.....	98

## DAFTAR GAMBAR

### GAMBAR

4.1 Fungsi Keanggotaan Dari Variable Permintaan.....	56
4.2 Fungsi Keanggotaan Dari Variable Persediaan .....	62
4.3 Fungsi Keanggotaan Dari Variable Pengadaan.....	68
4.4 Desain <i>Interface</i> Pengadaan Jumlah Lampu LED.....	79
4.5 DFD Level 0.....	79
4.6 DFD Level 1 Proses Pengadaan.....	80
4.7 Fungsi Keanggotaan Dari Variable Permintaan.....	83
4.8 Fungsi Keanggotaan Dari Variable Persediaan .....	85
4.9 Fungsi Keanggotaan Dari Variable Pengadaan.....	86
4.10 Tampilan Proses SPK.....	94

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Dengan perkembangan ilmu yang sangat pesat dapat membantu manusia dalam memecahkan segala masalah dan kebutuhan yang semakin banyak dan kompleks. Akan tetapi hal tersebut tidak bisa lepas dari peran berbagai cabang ilmu pengetahuan yang mendasarinya. Seperti matematika yang melahirkan berbagai cabang ilmu lain. Matematika sebagai bahasa simbol yang bersifat universal hubungannya sangat erat dengan kehidupan nyata. Fakta membuktikan bahwa matematika mempunyai peranan yang sangat penting dalam menyelesaikan masalah-masalah yang dijumpai dalam kehidupan sehari-hari. Masalah-masalah tersebut dapat diselesaikan dengan beberapa metode pada ilmu matematika.

Matematika adalah ilmu tentang logika mengenai bentuk, susunan, besaran dan konsep-konsep yang berhubungan satu dengan yang lainnya dengan jumlah yang banyak. Matematika tidak hanya terbatas dalam perhitungan angka saja, namun dari hal ini muncul keterkaitan yang bisa diaplikasikan dalam cabang ilmu lain. Di dalam kehidupan nyata seringkali kita menjumpai masalah yang tidak bisa diselesaikan secara langsung. Salah satunya yang menimbulkan kesamaran (*vagueness*). Kesamaran dinyatakan sebagai sebuah bahasa lazim yang diterima dengan arti yang berbeda di setiap tempat. Pada awalnya, masalah ini dapat diatasi dengan ilmu statistika dan teori peluang (Klirr dkk, 1997:5). Namun akhirnya, permasalahan

semakin bertambah rumit dan tidak terpecahkan, terutama masalah kesamaran terhadap kalimat dalam bahasa sehari-hari.

Misalnya ketika didefinisikan suatu himpunan  $U$  adalah himpunan semua makanan kaleng di Indonesia. Terdapat suatu himpunan  $A$  di dalam himpunan  $U$  yang dapat didefinisikan dengan syarat keanggotaan makanan kaleng yang berisi kornet. Jika didefinisikan himpunan lain di dalam  $U$  yang menyatakan himpunan makanan kaleng yang diproduksi di Indonesia, hal ini akan menyebabkan kesulitan. Hal ini dikarenakan tidak dapat diketahui secara tepat apakah setiap makanan kaleng yang diproduksi di Indonesia semua bahannya berasal dari Indonesia.

Pada dasarnya kesulitan tersebut adalah terdapat beberapa himpunan yang mempunyai daerah perbatasan yang tidak tegas, sedangkan dalam himpunan tegas setiap menyatakan suatu himpunan harus tegas keanggotaannya. Untuk itulah diperlukan konsep himpunan *fuzzy* untuk mengatasi hal tersebut. Pada tahun 1965, Profesor Lotfi Asker Zadeh, seorang guru besar *University of California* mempublikasikan karya ilmiah berjudul "*Fuzzy sets*". Dalam karya ilmiah tersebut, Zadeh membuat terobosan baru yang memperluas konsep himpunan tegas (*Crisp Sets*). Dalam perkembangannya, penggunaan teori himpunan *fuzzy* terbagi menjadi tiga periode yaitu fase belajar (1965-1977) yang ditandai dengan perkembangan dan perkiraan penggunaannya. Kemudian fase transisi (1978-1988) yang ditandai dengan perkembangan teori dan banyak sukses dalam praktek penggunaan. Yang terakhir fase ledakan *fuzzy* (*Fuzzy Boom*) (1989-sekarang) yang ditandai dengan peningkatan

sukses dalam penggunaan di bidang industri, ekonomi/bisnis, dan penggunaan perangkat lunak (*Soft Computing*) (Klirr dkk,1997: 215-216).

Salah satu penerapan logika *fuzzy* adalah dalam ilmu ekonomi, yaitu penggunaan Sistem Inferensi *Fuzzy* dalam penentuan jumlah pengadaan. Ilmu ekonomi yang mempelajari tentang perencanaan pengadaan dalam hal penentuan jumlah pengadaan adalah manajemen operasi. Secara umum, manajemen operasi diartikan sebagai pengarahan dan pengendalian berbagai kegiatan yang mengolah berbagai jenis sumber daya untuk membuat barang atau jasa tertentu (Pontas M.Pardede, 2005: 13).

Dalam aktivitas ekonomi/bisnis, ada beberapa toko/waralaba yang belum memanfaatkan kecanggihan teknologi. Hal ini ditunjukkan dalam pengelolaan data barang yang masih dilakukan secara manual sehingga kurang efektif dalam kinerjanya, maka dari itu toko/waralaba membutuhkan suatu sistem yang bisa mengelola semua transaksi dengan efisien. Selanjutnya seringkali barang menumpuk dan kosong, terkadang harga dan kualitas barang juga turun. Oleh karena itu, bantuan komputer akan sangat membantu dalam transaksi dan mengatur persediaan barang tidak sekedar mengandalkan buku catatan. Dalam skripsi ini akan ditentukan optimasi perhitungan jumlah pengadaan barang yang memenuhi kondisi optimum.

Untuk mempermudah pekerjaan, dalam hal pengadaan barang untuk menghemat waktu dan memperkecil kesalahan dalam perhitungan, selanjutnya dibutuhkan Sistem Pendukung Keputusan (SPK). Sehingga pembuat keputusan cukup menginputkan data-data yang diperlukan oleh SPK, yang selanjutnya disebut variabel input, yaitu: persediaan barang maksimum satu periode tertentu, persediaan barang minimum satu

periode tertentu, permintaan maksimum satu periode tertentu, permintaan minimum satu periode tertentu, pengadaan maksimum satu periode tertentu, pengadaan minimum satu periode tertentu, permintaan saat ini, dan persediaan saat ini. Kemudian SPK akan mengolah data-data tersebut dengan metode *Tsukamoto* dan akan menampilkan keluaran (*output*) berupa jumlah barang yang akan didatangkan.

### 1.2 Rumusan Masalah

Berdasarkan pemikiran di atas maka rumusan masalahnya adalah:

1. Bagaimana proses membangun sistem pendukung keputusan untuk menentukan jumlah pengadaan lampu LED dengan metode *Tsukamoto* berdasarkan data persediaan dan jumlah permintaan?
2. Bagaimana tingkat validitas SPK dengan metode *Tsukamoto* untuk menentukan optimasi jumlah pengadaan berdasarkan data persediaan dan jumlah permintaan?
3. Bagaimana perbandingan jumlah perhitungan pengadaan dengan metode *Tsukamoto* dengan hasil pengadaan toko?

### 1.3 Pembatasan Masalah

Dari latar belakang di atas, agar pembahasan tidak terlalu luas maka diperlukan pembatasan masalah sebagai berikut:



1. Penentuan optimasi jumlah pengadaan berdasarkan data persediaan dan data jumlah permintaan, faktor-faktor lain yang mempengaruhi pengadaan tidak dibahas dalam penulisan ini.
2. Data-data yang digunakan untuk mengambil keputusan hanyalah data-data sebagai berikut: persediaan maksimum satu periode tertentu, persediaan minimum satu periode tertentu, permintaan maksimum satu periode tertentu, permintaan minimum satu periode tertentu, pengadaan maksimum satu periode tertentu, pengadaan minimum satu periode tertentu, permintaan saat ini dan persediaan saat ini.

#### 1.4 Tujuan Penulisan

Beberapa tujuan dari penulisan ini adalah:

1. Menerapkan metode *Tsukamoto* dalam menentukan jumlah optimasi pengadaan lampu LED berdasarkan persediaan dan jumlah permintaan.
2. Membangun sebuah Sistem Pendukung Keputusan (SPK) dengan metode *Tsukamoto* untuk menentukan jumlah optimasi pengadaan lampu LED berdasarkan persediaan dan jumlah.

#### 1.5 Sistematika Penulisan

Secara garis besar penulisan skripsi ini terjadi atas 3 bagian, yaitu bagian awal, bagian pokok dan bagian akhir yang masing-masing diuraikan sebagai berikut.

### 1.5.1 Bagian awal

Dalam penulisan skripsi ini, bagian awal berisi halaman judul, halaman pengesahan, motto dan persembahan, kata pengantar, abstrak, daftar isi, daftar Tabel, daftar Gambar, dan daftar lampiran.

### 1.5.2 Bagian pokok

Bagian pokok dari penulisan skripsi ini adalah isi skripsi yang terdiri atas 5 bab, yaitu:

#### 1. BAB I PENDAHULUAN

Berisi latar belakang, perumusan masalah, batasan masalah, tujuan, serta sistematika penulisan skripsi.

#### 2. BAB II LANDASAN TEORI

Berisi tentang Himpunan dan Logika *Fuzzy*, *Fuzzy Inferece System Tsukamoto*, Optimasi, Sistem Pendukung Keputusan, dan MATLAB.

#### 3. BAB III METODE PENELITIAN

Berisi tentang prosedur/langkah-langkah yang dilakukan dalam penelitian ini meliputi studi pustaka, perumusan masalah, pemecahan masalah yang meliputi pengumpulan dan pengolahan data serta perancangan, pembuatan dan pengujian sistem. Selanjutnya langkah terakhir dalam metode penelitian yaitu penarikan kesimpulan.

#### 4. BAB IV PEMBAHASAN

Berisi tentang model base *Tsukamoto*, inferensi, menentukan *output Crips*, perancangan sistem, pengujian aplikasi untuk optimasi pengadaan lampu

LED, perbandingan perhitungan manual dengan hasil dari aplikasi, validitas SPK.

## 5. BAB V PENUTUP

Berisi kesimpulan hasil penelitian dan saran-saran peneliti.

### 1.5.3 Bagian Akhir

Berisi daftar pustaka sebagai acuan penulisan yang mendukung kelengkapan skripsi ini dan lampiran-lampiran yang melengkapi uraian pada bagian isi.



## BAB II

### LANDASAN TEORI

#### 2.1 Himpunan dan Logika *Fuzzy*

##### a. Dari Himpunan Klasik ke Himpunan Samar (*fuzzy*)

Misalkan  $U$  sebagai semesta pembicaraan (himpunan semesta) yang berisi semua anggota yang mungkin dalam setiap pembicaraan atau aplikasi. Misalkan himpunan tegas  $A$  dalam semesta pembicaraan  $U$ . Dalam matematika ada tiga metode atau bentuk untuk menyatakan himpunan, yaitu metode pencacahan, metode pencirian dan metode keanggotaan. Metode pencacahan digunakan apabila suatu himpunan didefinisikan dengan mancah atau mendaftar anggota-anggotanya. Sedangkan metode pencirian, digunakan apabila suatu himpunan didefinisikan dengan menyatakan sifat anggota-anggotanya. (Setiadji, 2009: 8). Dalam kenyataannya, cara pencirian lebih umum digunakan, kemudian setiap himpunan  $A$  ditampilkan dengan cara pencirian sebagai berikut:

$$A = \{x \in U \mid x \text{ memenuhi suatu kondisi}\} \quad (2.1)$$

Metode ketiga adalah metode keanggotaan yang mempergunakan fungsi keanggotaan nol-satu untuk setiap himpunan  $A$  yang dinyatakan sebagai  $\mu_A(x)$ .

$$\mu_A(x) = \begin{cases} 1, & \text{jika } x \in A \\ 0, & \text{jika } x \notin A \end{cases} \quad (2.2)$$

Menurut Nguyen dkk (2003: 86) fungsi pada persamaan (2.2) disebut fungsi karakteristik atau fungsi indikator. Suatu himpunan *fuzzy*  $A$  di dalam semesta pembicaraan  $U$  didefinisikan sebagai himpunan yang bercirikan suatu fungsi keanggotaan  $\mu_A$ , yang mengawankan setiap  $x \in U$  dengan bilangan real di dalam interval  $[0,1]$ , dengan nilai  $\mu_A(x)$  menyatakan derajat keanggotaan  $x$  di dalam  $A$ . Dengan kata lain jika  $A$  adalah himpunan tegas, maka nilai keanggotaannya hanya terdiri dari dua nilai yaitu 0 dan 1. Sedangkan nilai keanggotaan di himpunan fuzzy adalah interval tertutup  $[0,1]$ . Artinya nilai keanggotaan tersebut menunjukkan bahwa suatu item tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar, dan masih ada nilai-nilai yang terletak antara benar dan salah.

b. Atribut

Himpunan *fuzzy* memiliki 2 atribut (Sri Kusumadewi dan Hari Purnomo, 2004: 6), 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 menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb.

c. Istilah-istilah dalam logika *fuzzy*

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

1. Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy* (Sri Kusumadewi dan Hari Purnomo, 2004: 6). Contoh: Umur, Temperatur, Permintaan, Persediaan, Produksi, dan sebagainya.

## 2. Himpunan *fuzzy*

Misalkan  $X$  semesta pembicaraan, terdapat  $A$  di dalam  $X$  sedemikian sehingga:

$$A = \{ x, \mu_A[x] \mid x \in X, \mu_A: x \rightarrow [0,1] \} \quad (2.3)$$

Suatu himpunan *fuzzy*  $A$  di dalam semesta pembicaraan  $X$  didefinisikan sebagai himpunan yang bercirikan suatu fungsi keanggotaan  $\mu_A$ , yang mengawankan setiap  $x \in X$  dengan bilangan real di dalam interval  $[0,1]$ , dengan nilai  $\mu_A(x)$  menyatakan derajat keanggotaan  $x$  di dalam  $A$  (Athia Saelan, 2009: 2). Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Misalkan  $X = \text{Umur}$  adalah variabel *fuzzy*. Maka dapat didefinisikan himpunan “Muda”, “Parobaya”, dan “Tua” (Jang dkk, 1997:17).

## 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: semesta pembicaraan untuk variabel umur:  $[0, +\infty)$ . (Sri Kusumadewi dan Hari Purnomo, 2004:7). Sehingga semesta pembicaraan dari variable umur adalah  $0 \leq \text{umur} < +\infty$ . Dalam hal ini, nilai yang diperbolehkan untuk dioperasikan dalam variable umur adalah lebih besar dari atau sama dengan 0, atau kurang dari positif tak hingga.

#### 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*: Muda =  $[0, 45]$  (Sri Kusumadewi dan Hari Purnomo, 2004: 8).

## 2.2 Fungsi Keanggotaan

Jika  $X$  adalah himpunan objek-objek yang secara umum dinotasikan dengan  $x$ , maka himpunan *fuzzy*  $A$  di dalam  $X$  didefinisikan sebagai himpunan pasangan berurutan (Jang dkk, 1997:14):

$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (2.4)$$

$\mu_A(x)$  disebut derajat keanggotaan dari  $x$  dalam  $A$ , yang mengindikasikan derajat  $x$  berada di dalam  $A$  (Lin dan Lee, 1996: 10). Dalam himpunan *fuzzy* terdapat beberapa representasi dari fungsi keanggotaan, salah satunya yaitu 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, yaitu representasi linear naik dan representasi linear turun.

#### 1. Representasi linear NAIK

Pada representasi linear NAIK, kenaikan nilai derajat keanggotaan himpunan *fuzzy* ( $\mu[x]$ ) dimulai pada nilai domain yang memiliki derajat keanggotaan nol  $[0]$  bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Fungsi keanggotaan representasi linear naik dapat dicari dengan cara sebagai berikut:

Himpunan *fuzzy* pada representasi linear NAIK memiliki domain  $(-\infty, \infty)$  terbagi menjadi tiga selang, yaitu:  $[0, a]$ ,  $[a, b]$ , dan  $[b, \infty)$ .

##### a) Selang $[0, a]$

Fungsi keanggotaan himpunan *fuzzy* pada representasi linear NAIK pada selang  $[0, a]$  memiliki nilai keanggotaan = 0.

##### b) Selang $[a, b]$

Pada selang  $[a, b]$ , fungsi keanggotaan himpunan *fuzzy* pada representasi linear NAIK direpresentasikan dengan garis lurus yang melalui dua titik,



yaitu dengan koordinat  $(a,0)$  dan  $(b,1)$ . Misalkan fungsi keanggotaan *fuzzy* NAIK dari  $x$  disimbolkan dengan  $\mu[x]$ , maka persamaan garis lurus tersebut adalah:

$$\frac{\mu[x]-0}{1-0} = \frac{x-a}{b-a}$$

$$\Leftrightarrow \mu[x] = \frac{x-a}{b-a} \quad (2.5)$$

c) Selang  $[b,\infty)$

Fungsi keanggotaan himpunan *fuzzy* pada representasi linear NAIK pada selang  $[x_{\max}, \infty)$  memiliki nilai keanggotaan = 0.

Dari uraian di atas, fungsi keanggotaan himpunan *fuzzy* pada representasi linear NAIK, dengan domain  $(-\infty,\infty)$  adalah:

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x \geq b \end{cases} \quad (2.6)$$

2. Representasi linear TURUN

Sedangkan pada representasi linear TURUN, garis lurus dimulai dari nilai domain dengan derajat keanggotaan himpunan *fuzzy* ( $\mu[x]$ ) tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan himpunan *fuzzy* lebih rendah. Fungsi keanggotaan representasi linear TURUN dapat dicari dengan cara sebagai berikut:

Himpunan *fuzzy* pada representasi linear TURUN memiliki domain  $(-\infty,\infty)$  terbagi menjadi tiga selang, yaitu:  $[0, a]$ ,  $[a, b]$ , dan  $[b, \infty)$ .

a) Selang  $[0, a]$

Fungsi keanggotaan himpunan *fuzzy* pada representasi linear TURUN pada selang  $[0, a]$  memiliki nilai keanggotaan=0

b) Selang  $[a, b]$

Pada selang  $[a, b]$ , fungsi keanggotaan himpunan *fuzzy* pada representasi linear TURUN direpresentasikan dengan garis lurus yang melalui dua titik, yaitu dengan koordinat  $(a, 1)$  dan  $(b, 0)$ . Misalkan fungsi keanggotaan *fuzzy* TURUN dari  $x$  disimbolkan dengan  $\mu[x]$ , maka persamaan garis lurus tersebut adalah:

$$\frac{\mu[x]-0}{1-0} = \frac{x-b}{a-b}$$

$$\Leftrightarrow \mu[x] = \frac{x-b}{a-b}$$

Karena pada selang  $[a, b]$ , gradien garis lurus=-1, maka persamaan garis lurus tersebut menjadi:

$$\mu[x] = (-1)\left(\frac{x-b}{a-b}\right)$$

$$\mu[x] = \frac{b-x}{b-a}$$

c) Selang  $[b, \infty)$

Fungsi keanggotaan himpunan *fuzzy* pada representasi linear TURUN pada selang  $[b, \infty)$  memiliki nilai keanggotaan = 0.

Dari uraian di atas, fungsi keanggotaan himpunan *fuzzy* pada representasi linear TURUN, dengan domain  $(-\infty, \infty)$  adalah:

$$\mu_A(x) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a \leq x \leq b \\ 0, & x \geq b \end{cases}$$

### 2.3 Teori Operasi Himpunan

Menurut Lin dan Lee (1996: 27) Ada dua operasi pokok dalam himpunan fuzzy, yaitu:

#### 1. Konjungsi *fuzzy*

Konjungsi *fuzzy* dari A dan B dilambangkan dengan  $A \wedge B$  dan didefinisikan oleh:

$$\mu_{A \wedge B} = \mu_A(x) \cap \mu_B(y) = \min(\mu_A(x), \mu_B(y)) \quad (2.7)$$

#### 2. Disjungsi *fuzzy*

Disjungsi *fuzzy* dari A dan B dilambangkan dengan  $A \vee B$  dan didefinisikan oleh:

$$\mu_{A \vee B} = \mu_A(x) \cup \mu_B(y) = \max(\mu_A(x), \mu_B(y)) \quad (2.8)$$

### 2.4 Metode *Fuzzy Inference System* (FIS) Tsukamoto

Inferensi adalah proses penggabungan banyak aturan berdasarkan data yang tersedia. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Dua pendekatan untuk menarik kesimpulan pada *IF-THEN rule* (aturan

jika-maka) adalah *forward chaining* dan *backward chaining* (Turban dkk, 2005:726).

**a. *Forward chaining***

*Forward chaining* mencari bagian JIKA terlebih dahulu. Setelah semua kondisi dipenuhi, aturan dipilih untuk mendapatkan kesimpulan. Jika kesimpulan yang diambil dari keadaan pertama, bukan dari keadaan yang terakhir, maka ia akan digunakan sebagai fakta untuk disesuaikan dengan kondisi JIKA aturan yang lain untuk mendapatkan kesimpulan yang lebih baik. Proses ini berlanjut hingga dicapai kesimpulan akhir .

**b. *Backward chaining***

*Backward chaining* adalah kebalikan dari *forward chaining*. Pendekatan ini dimulai dari kesimpulan dan hipotesis bahwa kesimpulan adalah benar. Mesin inferensi kemudian mengidentifikasi kondisi JIKA yang diperlukan untuk membuat kesimpulan benar dan mencari fakta untuk menguji apakah kondisi JIKA adalah benar. Jika semua kondisi JIKA adalah benar, maka aturan dipilih dan kesimpulan dicapai. Jika beberapa kondisi salah, maka aturan dibuang dan aturan berikutnya digunakan sebagai hipotesis kedua. Jika tidak ada fakta yang membuktikan bahwa semua kondisi JIKA adalah benar atau salah, maka mesin inferensi terus mencari aturan yang kesimpulannya sesuai dengan kondisi JIKA yang

tidak diputuskan untuk bergerak kan untuk bergerak satu langkah ke depan memeriksa kondisi tersebut. Proses ini berlanjut hingga suatu set aturan didapat untuk mencapai kesimpulan atau untuk membuktikan tidak dapat mencapai kesimpulan.

Menurut Sri Kusumadewi dan Sri Hartati (2006:34) sistem inferensi *fuzzy* merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* yang berbentuk *IF-THEN*, dan penalaran *fuzzy*.

Sistem inferensi *fuzzy* menerima *input crisp*. Input ini kemudian dikirim ke basis pengetahuan yang berisi n aturan *fuzzy* dalam bentuk *IF-THEN*. *Fire strength* (nilai keanggotaan anteseden atau  $\alpha$ ) akan dicari pada setiap aturan. Apabila aturan lebih dari satu, maka akan dilakukan agregasi semua aturan. Selanjutnya pada hasil agregasi akan dilakukan *defuzzy* untuk mendapatkan nilai *crisp* sebagai output sistem. Salah satu metode FIS yang dapat digunakan untuk pengambilan keputusan adalah metode *Tsukamoto*.

## 2.5 Metode Tsukamoto

Sistem Inferensi *Fuzzy* merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk *IF-THEN*, dan penalaran *fuzzy*. Selama ini telah dikenal beberapa metode dalam FIS, seperti 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 monoton. Sebagai hasilnya, keluaran hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan  $\alpha$ -predikat (*fire strength*). Hasil akhir menggunakan rata-rata terbobot. (Setiadji, 2009: 200).

Bentuk model *fuzzy Tsukamoto* adalah : IF (X IS A) and (Y IS B) Then (Z IS C). Dimana A,B dan C adalah himpunan *fuzzy*. Misalkan diketahui 2 *rule* berikut

IF (X is A<sub>1</sub>) AND (Y is B<sub>1</sub>) THEN (Z is C<sub>1</sub>)

IF (X is A<sub>2</sub>) AND (Y is B<sub>2</sub>) THEN (Z is C<sub>2</sub>)

Dalam inferensinya, metode *Tsukamoto* menggunakan tahapan berikut :

1. *Fuzzyfikasi*
2. Pembentukan basis pengetahuan *Fuzzy* (Rule dalam bentuk IF....THEN)
3. Mesin Inferensi, menggunakan fungsi implikasi MIN untuk mendapatkan nilai  $\alpha$ -predikat tiap-tiap rule ( $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$ ). Kemudian masing-masing nilai  $\alpha$ -predikat ini digunakan untuk menghitung keluaran hasil inferensi secara tegas (*crisp*) masing-masing rule ( $z_1, z_2, z_3, \dots, z_n$ ).

#### 4. *Defuzzyfikasi*

Menggunakan metode rata-rata (*Average*)

$$Z = \frac{\sum \alpha_i z_i}{\sum \alpha_i} \quad (2.8)$$

Keterangan :

$Z$  = Variabel *output*

$\alpha_i$  = Nilai  $\alpha$  predikat

$z_i$  = Nilai variabel *output*

## 2.6 Fungsi Keanggotaan

Fungsi Keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam 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. Ada beberapa fungsi yang bisa digunakan. (Dhansipp, 2009)

### 1. Representasi Linear

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

$$\mu[x, a, b] = \begin{cases} 1 & ; x \leq 0 \\ \frac{x-a}{b-a} & ; 0 \leq x \leq 1 \\ 0 & ; x \geq 1 \end{cases} \quad (2.9)$$

Keterangan:

$a$  = nilai domain yang mempunyai derajat keanggotaan nol

$b$  = nilai domain yang mempunyai derajat keanggotaan satu

$x$  = nilai *input* yang akan di ubah ke dalam bilangan *fuzzy*

## 2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (*linear*). Nilai-nilai disekitar  $b$  memiliki derajat keanggotaan turun cukup tajam (menjauhi 1).

$$\mu[x, a, b] = \begin{cases} 0 & ; x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a} & ; a \leq x \leq b \\ \frac{c-x}{c-b} & ; b \leq x \leq c \end{cases} \quad (2.10)$$

Keterangan:

$a$  = nilai domain terkecil yang mempunyai derajat keanggotaan nol

$b$  = nilai domain yang mempunyai derajat keanggotaan satu

$c$  = nilai domain terbesar yang mempunyai derajat keanggotaan nol

## 3. Representasi Kurva Trapesium

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

$$\mu[x, a, b] = \begin{cases} 0 & ; x \leq a \\ \frac{x-a}{b-a} & ; a \leq x \leq b \\ 1 & ; b \leq x \leq c \\ \frac{d-x}{d-c} & ; c \leq x \leq d \\ 0 & ; x \geq d \end{cases} \quad (2.11)$$



Keterangan:

a = nilai domain terkecil yang mempunyai derajat keanggotaan nol

b = nilai domain terkecil yang mempunyai derajat keanggotaan satu

c = nilai domain terbesar yang mempunyai derajat keanggotaan satu

d = nilai domain terbesar yang mempunyai derajat keanggotaan nol

x = nilai input yang akan di ubah ke dalam bilangan *fuzzy*

#### 4. Kurva Bentuk Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. 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.

$$\mu[x, a, b] = \begin{cases} 0 & ; x \leq a \\ \frac{b-x}{b-a} & ; a \leq x \leq b \\ 1 & ; x \geq a \\ 0 & ; x \leq a \\ \frac{x-a}{b-a} & ; a \leq x \leq b \\ 1 & ; x \geq b \end{cases} \quad (2.12)$$

#### 2.7 Operator Dasar Untuk Himpunan *Fuzzy*

Misalkan A dan B adalah himpunan *fuzzy*.

### Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y]) \quad (2.13)$$

### Operator OR

Operator ini berhubungan dengan operasi union pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y]) \quad (2.14)$$

### Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan pada himpunan yang bersangkutan dari

$$1. \mu_{A'} = 1 - \mu_A[x] \quad (2.15)$$

## 2.8 Fungsi Implikasi

Tiap-tiap aturan pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah: Misalkan A dan B adalah himpunan *Fuzzy*

$$IF\ x\ is\ A\ THEN\ y\ is\ B \quad (2.16)$$

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

$$IF\ (x_1\ is\ A_1) \cdot (x_2\ is\ A_2) \cdot (x_3\ is\ A_3) \cdot \dots \cdot (x_N\ is\ A_N)\ THEN\ y\ is\ B \quad (2.17)$$

dengan  $\cdot$  adalah operator (misal: OR atau AND).

## 2.9 Optimasi

Optimasi adalah suatu proses untuk mencapai hasil yang ideal atau optimasi (nilai efektif yang dapat dicapai). Optimasi dapat diartikan sebagai suatu bentuk mengoptimalkan sesuatu hal yang sudah ada, ataupun merancang dan membuat sesuatu secara optimal.

## 2.10 Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan (*Decision Support System*) adalah sistem informasi berbasis komputer yang menyediakan dukungan informasi yang

interaktif bagi manajer dan praktisi bisnis selama proses pengambilan keputusan (O'Brien, 2005: 448). SPK dibangun tentunya mempunyai tujuan yang ingin dicapai oleh seorang pembuat keputusan. Menurut Aji Supriyanto (2005:260) tujuan SPK adalah sebagai "*second opinion*" atau "*information sources*" sebagai bahan pertimbangan seorang manajer sebelum memutuskan kebijakan tertentu.

## 2.11 MATLAB

### 2.11.1 MATLAB (*Matrix Laboratory*)

### 2.11.2 Mengenal MATLAB

#### Sekilas tentang MATLAB

MATLAB adalah *software* buatan The MathWork, Inc., yang sangat bermanfaat untuk menyelesaikan berbagai masalah komputasi numerik. Perangkat lunak ini menawarkan kemudahan dan kesederhanaan dalam menyelesaikan permasalahan yang berhubungan dengan vektor dan matriks. Memperoleh inversi matriks dan menyelesaikan persamaan linier merupakan contoh permasalahan yang dapat dipecahkan dengan cara yang sangat singkat dan mudah sekali (Irawan, 2012:12).

Untuk menangani persoalan-persoalan yang spesifik, MATLAB menyediakan sejumlah *Toolbox*. Contoh *toolbox* :

- a. *Image Processing* : ditujukan secara khusus untuk melakukan pengolahan citra;
- b. *Signal Processing* : ditujukan untuk menangani pengolahan isyarat;
- c. *Neural Network* : menyediakan berbagai fungsi yang terkait dengan jaringan syaraf tiruan.

#### 2.11.1.2. Menjalankan MATLAB

Sebelum kita mencoba untuk menggunakan MATLAB, kita perlu menjalankan program tersebut terlebih dulu. Pada lingkungan windows, program MATLAB dijalankan dengan cara seperti berikut.

- a) Klik pada tombol *START*.
- b) Pilih *All Programs*.
- c) Klik pada folder MATLAB.
- d) Klik pada *icon* MATLAB R2009a (atau yang sesuai pada komputer kita).

Beberapa bagian penting yang terdapat pada *interface* MATLAB adalah seperti berikut (Irawan, 2010:12-13).

- a) *Command Window* atau jendela perintah adalah jendela yang dipakai untuk memberikan perintah secara manual.
- b) *Workspace* berisi daftar variabel yang diciptakan oleh pemakai dan masih ada dalam memori.

- c) *Command History* mencantumkan perintah-perintah yang pernah diberikan oleh pemakai.
- d) *Current Directory* menyatakan direktori kerja.

### 2.11.1.3 *Command Window*

MATLAB memungkinkan kita bekerja pada jendela *command* (*command window*). Pada jendela ini, kita bisa memberikan perintah MATLAB secara interaktif. Begitu kita menekan tombol Enter, MATLAB akan mengeksekusi perintah yang telah kita berikan.

Pada jendela *command*, tanda `>>` menyatakan bahwa MATLAB siap menerima perintah dari pemakai.

Setiap perintah yang dapat dieksekusi oleh MATLAB disebut sebagai pernyataan (*statement*). Umumnya pernyataan berbentuk :

Variabel = ekspresi

atau

Ekspresi.



### 2.11.1.4 Kesalahan Perintah

Apabila perintah yang kita berikan tidak dikenal, MATLAB akan menampilkan pesan kesalahan. Sebagai contoh : ketikkan

`>> 5 + 4a`

Begitu tombol Enter kita tekan, MATLAB segera menampilkan kesalahan seperti berikut:

```
???5+4a (Error : Unexpected MATLAB
expression).
```

Ini berarti MATLAB menunjukkan posisi yang salah. Untuk memperbaiki ekspresi yang salah, gunakan tombol  $\uparrow$  untuk menampilkan perintah. Kemudian, lakukan pengeditan dan akhiri dengan menekan tombol Enter (Irawan, 2012:14).

#### 2.11.1.5 Menggunakan Variabel

Pada *Command Window*, kita bisa menggunakan variabel. Variabel adalah suatu nama yang dapat dipakai untuk menyimpan suatu nilai dan nilai yang ada di dalamnya bisa diubah sewaktu-waktu.

Sebelum mempraktekkan penggunaan variabel, aturan tentang cara menamakan variabel perlu diketahui terlebih dahulu. Aturan dalam memberikan nama terhadap variabel adalah sebagai berikut.

1. MATLAB membedakan huruf kecil dan huruf kapital pada penamaan variabel. Dengan demikian, *bilangan* dan *Bilangan* adalah dua variabel yang berbeda.
2. Nama variabel harus diawali dengan huruf, sedangkan kelanjutannya dapat berupa huruf, angka atau tanda garis bawah (`_`).

3. Panjang nama variabel dapat mencapai 31 karakter. Jika nama variabel lebih dari 31 karakter, maka karakter ke-32 dan seterusnya diabaikan.
4. Nama variabel bersifat *case sensitive*. Artinya, huruf kapital dan huruf kecil dibedakan. Jadi, nama seperti Bil dan bil dianggap berbeda.

#### 2.11.1.6 Mengenal Variabel Khusus

MATLAB menyediakan sejumlah variabel khusus, yaitu variabel yang dipakai oleh MATLAB dan memiliki makna secara khusus. Sejumlah daftar variabel khusus seperti pada Tabel 2.1.

Tabel 2.1 Daftar Variabel Khusus pada MATLAB

Variabel	Keterangan
Ans	Menampung hasil suatu ekspresi yang dijadikan sebagai sebuah pernyataan
Eps	Bilangan terkecil dalam komputer yang apabila ditambah dengan satu maka akan bernilai lebih besar daripada satu
Flops	Jumlah operasi bilangan pecahan
Inf	Berasal dari kata " <i>infinitive</i> " yang artinya adalah



tak berhingga (hasil 1/0)

Nan Berasal dari “Not-a-Number” atau “bukan sebuah bilangan. Misalnya hasil dari 0/0”

Pi Menyatakan bilangan  $\pi$

Realmin Bilangan real positif terkecil

Realmax Bilangan real positif terbesar

#### 2.11.1.7 Mengenal *Workspace*

Ketika kita bekerja dengan MATLAB, semua variabel yang kita ciptakan akan tersimpan dalam memori komputer. MATLAB menggunakan istilah *workspace* untuk menyimpan variabel-variabel tersebut (Irawan, 2012:24-25).

Untuk mengetahui variabel apa saja yang terdapat pada *workspace* tanpa melihat pada jendela *workspace*, kita bisa menggunakan perintah **who**.

```
>> who
```

Sebuah variabel membutuhkan memori komputer. Agar memori yang dipakai tidak membesar, variabel-variabel yang tidak digunakan bisa kita hapus. Caranya dengan menggunakan perintah **clear**. Jika ingin menghapus beberapa variabel maka ketikkan

```
>> clear var1 var2 ...
```

Akan tetapi jika ingin menghapus semua variabel maka ketikkan

```
>> clear
```

## 2.11.2 Operasi Array

### 2.11.2.1 Dasar –Dasar array

Variabel dengan tipe data tunggal (skalar) hanya dapat digunakan untuk menyimpan sebuah nilai saja, sehingga untuk menyimpan beberapa nilai sekaligus dalam suatu variabel khusus dibutuhkan variabel *array* atau variabel berindeks. Variabel *array* dapat digunakan untuk menampung banyak data yang sejenis (*numeric / string*). Sebagai alternatif lain dalam membuat *array* yang perlu dilakukan hanyalah mengetikkan kurung kotak kiri "[", memasukkan elemen-elemen dengan dipisahkan oleh spasi atau koma, kemudian menutup *array* dengan kurung kotak kanan "]". Untuk memisahkan elemen *array* yang satu dengan lainnya digunakan spasi atau koma atau titik koma. Bila menggunakan koma maka akan menghasilkan *array* baris, lalu bila digunakan titik koma akan menghasilkan *array* kolom. Untuk membuat *array* yang mempunyai banyak elemen digunakan notasi kolom, sebagai contoh :

```
x=1:10
```

```
x =
```

```
1 2 3 4 5 6 7 8 9 10
```

### Cara Untuk Membuat *Array*

```
x=[ 2 3 4 ]
```

membuat vektor baris *x* yang memuat elemen-elemen yang diberikan

```
x=awal : akhir
```

membuat vektor baris *x* dimulai dengan awal, interval satu, diakhiri dengan akhir.

```
x=awal:kenaikan:akhir
```

membuat vektor baris dimulai dengan awal, interval sebesar kenaikan, diakhiri pada atau sebelum akhir.

```
x=linspace(awal,akhir,n)
```

membuat vektor baris diawali dengan awal, berakhir dengan akhir, mempunyai *n* elemen.

```
x=logspace(awal,akhir,n)
```

#### 2.11.2.2 Operasi *Array*

Dari contoh diatas kita selalu membuat *array* berbentuk vektor baris , dalam MATLAB dimungkinkan juga untuk membuat suatu *array* berbentuk vektor kolom. Dalam hal ini manipulasi *array* tidak menimbulkan perubahan, satu-satunya perbedaan hanyalah hasilnya ditampilkan sebagai kolom bukan sebagai baris. Seperti telah dijelaskan pada bagian sebelumnya untuk membuat vektor kolom elemen-elemen *array* dipisahkan dengan titik koma " ; ".

Cara lain adalah membentuk vektor baris kemudian di transpose mejadi vektor kolom menggunakan notasi " ' ". Sebagai contoh :

```
x=1:10
```

```
x =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
y=x'
```

```
y =
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

### Operasi Array-Skalar

Semua operasi matematika sederhana antara skalar dan *array* mempunyai sifat yang sama (+, -, :, \*). Semua operasi *array* dengan skalar akan dikenakan pada semua elemen *array*.

```
y=x-2
```

```
y =
```

```
-1 0 1 2 3 4 5 6 7 8
```

```
y=3*x/2 -5
```

```

y =
Columns 1 through 7
-3.5000 -2.0000 -0.5000 1.0000 2.5000 4.0000
5.5000
Columns 8 through 10
7.0000 8.5000 10.0000

```

### Operasi Array-Array

Operasi antara *array* tidak sama dengan operasi diatas. Logika sederhana dalam operasi *array-array* adalah logika operasi matematika antara vektor atau matriks. Syarat-syarat operasi matematika antara vektor atau matriks berlaku juga pada *array*. Sebagai contoh

```
x=[10 20 30 10;40 50 60 20;70 80 90 30]
```

```

x =
10 20 30 10
40 50 60 20
70 80 90 30

```

```
y=[50 60 70 5 ;1 2 3 6 ;40 5 20 7]
```

```

y =
50 60 70 5
1 2 3 6
40 5 20 7

```

```
x+y
```

```

ans =
60 80 100 15
41 52 63 26

```

```

110 85 110 37

x-y

ans =

-40 -40 -40 5

39 48 57 14

30 75 70 23

x*y
??? Error using ==> *
Inner matrix dimensions must agree.

x.*y

ans =

500 1200 2100 50

40 100 180 120

2800 400 1800 210

```

Dalam contoh diatas untuk perkalian menggunakan " . \* " ini berarti perkalian elemen dengan elemen. Perkalian tanpa titik berarti perkalian matriks sehingga syarat-syarat perkalian matriks harus terpenuhi.

```

x^2
??? Error using ==> ^
Matrix must be square.

x.^2

% mengangkat setiap elemen array

ans =

100 400 900 100

1600 2500 3600 400

4900 0 8100 900

```

### 2.11.2.3 Memanipulasi *Array*

*Array* dan matriks merupakan hal yang mendasar dalam MATLAB, maka ada banyak cara untuk memanipulasinya. Sekali matriks dibuat, MATLAB menyediakan cara untuk meyisipkan, mengambil dan mengatur kembali isi matriks tersebut. Penguasaan akan hal ini merupakan kunci untuk menggunakan MATLAB secara efisien. Sebagai contoh manipulasi matriks atau *array* perhatikan contoh berikut : membuat *array* dengan elemen bilangan satu atau nol.

```
x=ones(3,4) % membuat array dengan ukuran 3
```

```
baris empat kolom
```

```
x =
```

```
1 1 1 1
```

```
1 1 1 1
```

```
1 1 1 1
```

```
size(x) % mengetahui ukuran x
```

```
ans =
```

```
3
```

```
4
```

```
zeros(4)
```

```
ans =
```

```
0 0 0 0
```

```
0 0 0 0
```

```
0 0 0 0
```

```
0 0 0 0
```

Jika digunakan argumen tunggal, *ones(n)* atau *zeros(n)* MATLAB akan membuat *array* (n X n). Jika dipanggil dengan dua argumen misal *ones(r,c)* maka MATLAB akan membuat array dengan r baris dan c kolom.

```
A=[1 2 3;4 5 6; 7 8 9] % membuat array baru
```

```
A =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
A(2,3)=0; % mengubah elemen baris 2 kolom 3  
menjadi nol
```

```
A(1,1)=100; % mengubah elemen baris 1 kolom  
1 menjadi 100
```

```
A =
```

```
100 2 3
```

```
4 5 0
```

```
7 8 9
```

Untuk menghapus elemen matriks digunakan tanda " [ ] ". Tanda ini juga berguna untuk membuat matriks kosong. Contoh :

```
A(:,2)=[ ]
```

```
A =
```

```
1 3
```

```
4 6
```

```
7 9
```



Argumen diatas maksudnya hapus semua elemen di kolom 2. Untuk mencari elemen *array* digunakan perintah *find*, fungsi *find* mempunyai argumen sebagai berikut :

$$i=find(x)$$

menghasilkan indeks dari *array* x di mana elemen-elemennya tidak nol

$$[r,c]=find(x)$$

menghasilkan indeks baris dan kolom dari *array* x di mana elemen-elemennya tidak nol (Pujiriyanto, 2004).

### 2.11.3 Mengenal GUI

#### 2.11.3.1 Memanfaatkan GUIDE

Sejauh ini, kita bekerja pada *command line*. Sesungguhnya, MATLAB memungkinkan kita untuk membuat aplikasi yang berbasis grafik. Cara termudah untuk membuat *interface* berbasis grafis (GUI) yaitu dengan menggunakan fasilitas GUIDE (*Graphical User Interface Development Environment*). Caranya, pada promp MATLAB, ketikkan *guide* seperti berikut :

$$>>guide \text{ (enter)}$$

Sesaat kemudian, akan muncul tampilan GUIDE *Quick Start*. Untuk kepentingan latihan yang akan dibahas, biarkan *Blank GUI* dalam keadaan terpilih. Kemudian klik pada tombol OK. Kita akan menjumpai tampilan jendela aplikasi (*untitled1.fig*).

### 2.11.3.2 Mengenal *Toolbar*

Pada jendela GUIDE terdapat bagian yang dinamakan *toolbar*. Letaknya di bagian kiri. Di dalam *toolbar* tercantum komponen-komponen yang bisa diletakkan ke dalam aplikasi. Komponen-komponen yang terdapat pada *toolbar* dijelaskan dalam Tabel 2.2. Pengenalan terhadap elemen-elemen tersebut sangat penting sebelum memulai penyertaan komponen dalam aplikasi yang akan kita buat.

Tabel 2.2 Daftar *Toolbar* pada GUIDE MATLAB

Komponen	Keterangan
Select	Untuk membatalkan pemilihan terhadap komponen
Push Botton	Komponen berupa tombol
Radio Button	Komponen tombol radio
Edit Text	Komponen yang memungkinkan pemakai memasukkan nilai dari keyboard
Pop-up Menu	Berguna untuk membuat menu
Toggle Button	Tombol yang bersifat on/off
Axes	Berguna untuk membuat grafik

Button Group	Komponen yang berisi sejumlah tombol radio
Slider	Tombol penggeser
Check Box	Kotak cek
Static Text	Berguna untuk membuat label
Listbox	Digunakan untuk menyajikan sejumlah pilihan
Tabel	Untuk membuat Tabel
Panel	Kotak dengan judul berada di bagian atas
ActiveX Control	Kontrol ActiveX

---

### 2.11.3.3 Merancang *Interface*

#### 1. Menambahkan *Static Text*

Berikut adalah langkah untuk menambah sebuah komponen *static text* ke dalam *interface* yang masih kosong.

- a) Klik pada *Static Text* pada *toolbar*
- b) Letakkan penunjuk *mouse* pada posisi lalu di klik
- c) Klik pada *icon Property Inspector*. Maka akan muncul jendela *Inspector*
- d) Letakkan penunjuk *mouse* pada posisi dikanan *property* bernama *String*

- e) Kliklah. Lalu, gantilah *static text* menjadi kata yang diinginkan ;  
kemudian tekan tombol enter.

## 2. Menambahkan *Edit Text*

Berikut adalah langkah untuk menambah sebuah komponen *Edit Text*.

- a) Klik pada *Edit Text* pada *toolbar*
- b) Letakkan penunjuk *mouse* pada posisi yang diinginkan lalu klik.
- c) Kotak *edit (Edit Text)* yang baru saja kita tambahkan perlu diberi nama. Cara memberikan nama:
  1. Aktifkan *Property Inspector*
  2. Gantilah isi *property Tag* menjadi misal *edit\_.....*
  3. Kosongkan isi *property String*

dengan cara seperti itu, nama untuk kotak edit diberi nama *edit\_...* dan isinya dikosongkan.

## 3. Menambahkan *Push Button*

Berikut adalah langkah untuk menambah sebuah komponen *Push Button*

- a) Klik pada *Push Botton* pada *toolbar*
- b) Letakkan penunjuk *mouse* pada posisi yang diinginkan kemudian klik.
- c) Tombol yang baru saja kita tambahkan perlu diberi nama dan dilegkapi dengan judul. Cara memberikan nama dan judul:
  1. Aktifkan *Property Inspector*

2. Gantilah isi *property Tag* menjadi *pushbutton\_hitung*
3. Isikan *hitung* pada *property String* dan kemudian tekanlah tombol Enter.

#### 2.11.3.4 Menyimpan *Interface*

Setelah desain *interface* dilakukan. Kita perlu menyimpannya terlebih dahulu. Caranya seperti berikut.

- a) Klik pada menu *File*
- b) Klik pada pilihan *Save*
- c) Ketikkan nama *file* pada *file name*
- d) Klik pada tombol *Save*


Dengan cara seperti itu, judul pada *interface* berubah menjadi nama *file* yang kita simpan (...).fig

Setelah penyimpanan dilakukan, MATLAB membentuk skrip dengan nama (...).

Melalui skrip tersebut, penambahan kode secara manual akan dilakukan.

#### 2.11.3.5 Menjalankan *Interface*

Untuk menguji *interface* yang telah kita buat, kita bisa mengetikkan file yang akan di uji pada Command Windows dan menekan tombol Enter.

Cara yang lebih gampang klik saja pada ikon  yang terdapat pada *interface* yang kita buat (Irawan, 2012:321).

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dari hasil penelitian dan pembahasan tentang implementasi metode *Tsukamoto* dalam sistem pendukung keputusan untuk optimasi jumlah pengadaan lampu LED diperoleh kesimpulan sebagai berikut.

1. Sistem pendukung keputusan untuk optimasi jumlah pengadaan barang menggunakan algoritma *fuzzy* metode *Tsukamoto* ini di bangun dengan tiga himpunan yaitu permintaan, persediaan, dan pengadaan lampu LED di Toko KERTA Semarang. Selanjutnya dari himpunan tersebut tentukan variabel-variabel *fuzzy*. Himpunan permintaan ada tiga variabel yaitu SEDIKIT, SEDANG dan BANYAK. Himpunan persediaan ada tiga variabel yaitu MENURUN, TETAP dan MENINGKAT. Himpunan pengadaan ada tiga variabel yaitu BERKURANG, NORMAL dan BERTAMBAH. Dari variabel-variabel tersebut selanjutnya dilakukan inferensi sehingga menghasilkan 9 aturan *fuzzy*. Aturan tersebut digunakan untuk membangun sistem aplikasi untuk pengadaan lampu LED. Aplikasi tersebut dibuat menggunakan *software* MATLAB. Data diolah melalui beberapa tahapan, yaitu pertama data permintaan dan persediaan diinputkan, setelah semua data diinputkan maka selanjutnya dihitung

menggunakan algoritma *Tsukamoto* yang telah dimasukkan dalam kode program. Lalu akan keluar hasil perhitungan jumlah lampu LED yang akan diadakan. Dalam pembuatan aplikasi, tahap yang harus dilakukan mulai dari pembuatan desain program menggunakan *Graphic User Interface* (GUI), kemudian dilanjutkan dengan melengkapi kode pada MATLAB agar desain aplikasi bisa berfungsi. Setelah proses selesai, maka aplikasi dapat digunakan untuk optimasi pengadaan lampu LED.

2. Dari 18 data percobaan yang telah dilakukan, lalu dihitung validitas hasil kerja sistem. Diperoleh hasil persentasi validitas sebesar 76% dari membandingkan hasil kerja sistem dengan hasil perhitungan manual. Ini berarti aplikasi tersebut dapat digunakan sebagai sistem pendukung keputusan untuk optimasi pengadaan lampu LED sehingga tidak ada permintaan yang tidak terpenuhi.
3. Dalam membandingkan jumlah pengadaan hasil perhitungan metode *Tsukamoto* dengan jumlah pengadaan Toko, diambil dua contoh kasus. Kasus yang pertama, SPK kelebihan sisa pengadaan sebesar 22 buah lampu LED. Pada kasus kedua, SPK juga kelebihan sisa pengadaan 2 buah lampu LED. Perbandingan yang lainnya disajikan dalam Tabel 4.7.

## 5.2 Saran

Berikut ini adalah saran yang mungkin dapat digunakan untuk pengembangan sistem lebih lanjut.

1. Dalam upaya meningkatkan ketepatan dalam menentukan perkiraan pengadaan lampu LED disarankan untuk mengkaji faktor yang berpengaruh terhadap pengadaan dan menambahkan faktor daya tampung tempat penyimpanan, serta melakukan perbaikan terhadap aplikasi dengan melakukan perubahan interval pada fungsi keanggotaan untuk mendapatkan nilai terbaik.
2. Untuk mendapatkan perbandingan hasil yang terbaik menggunakan logika *fuzzy*, perlu dilanjutkan dengan menggunakan metode yang lain seperti metode Sugeno dan metode Mamdani.



## DAFTAR PUSTAKA

- Aji Supriyanto. 2005. *Pengantar Teknologi Informasi*. Jakarta: Salemba Infotek.
- Arman Hakim Nasution. 2008. *Perencanaan & Pengendalian Produksi*. Yogyakarta: Graha Ilmu.
- Athia.Saelan.2009.*Logika Fuzzy*.  
<http://www.informatika.org/~rinaldi/Matdis/2009-2010/MakalahStrukdis0910.pdf>. Tanggal akses: 22 April 2017.
- Hendra Kusuma. 2004. *Manajemen Produksi, Perencanaan & Pengendalian Produksi*. Yogyakarta: Graha Ilmu.
- Hicks, James O. & Leininger, Wayne E. 1986. *Accounting Information Systems*. Minnesota: West Publishing Co.
- Indira Rakanita. 2008. *Aplikasi Pengujian EPrT (English Proficiency Test) dan Analisator Terhadap Nilai EPrT (English Proficiency Test) Trial Application And EPrT Score Analysis Through Software*.  
[http://www.ittelkom.ac.id/library/index.php?view=article&catid=13%3Aapl&id=332%3Abasis-data-atau-database&option=com\\_content&Itemid=15](http://www.ittelkom.ac.id/library/index.php?view=article&catid=13%3Aapl&id=332%3Abasis-data-atau-database&option=com_content&Itemid=15). Tanggal akses: 21 Mei 2017.
- Jang, J.S.R. et al. 1997. *Neuro-Fuzzy and Soft Computing*. London: Prentice Hall.
- Jogiyanto. 1989. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi.
- Jogiyanto. 2005. *Pengenalan Komputer*. Yogyakarta: Andi Offset.

Klir, George J. , Clair, U.S. , and Yuan, Bo. 1997. *Fuzzy Set Theory, Foundation and Application*. New Jersey: Prentice Hall International, Inc.

Khikmiah,Lailatul. 2012. “Prediksi Permintaan Gas Cair Menggunakan Fuzzy Inference Model pada PT Air Products Gresik”.*Jurnal Teknik POMITS*, Volume 11No.1/2012. <http://digilib.its.ac.id/public/ITS-paper-24363-5208100093-Paper.pdf> diakses tanggal 22 Agustus 2017

Kusrini. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*.Yogyakarta:Andi.

Lin, Chin Teng & Lee, GS George. 1996. *Neural Fuzzy Systems*.London: Prentice Hall.

Moscove, Stephen A. & Simkin, Mark G. 1984. *Accounting Information Sitems Concepts and Practice for Effective Decision Making, second edition*. New York: John Willey and Sons.

Nguyen, Hung T, et al. 2003. *A First Course in Fuzzy and Neural Control*.USA: Chapman & Hall/CRC.

O'Brien, James A. 2005. *Introduction to Information System, 12thedition. (Pengantar Sistem Informasi Perspektif Bisnis dan Manajerial*.Jakarta: PT Salemba Empat (Emban Patria).

Paulus Bambangwirawan. 2004. *Struktur Data dengan C++*.Yogyakarta.

Pontas M. Pardede. 2005. *Manajemen Operasi dan Produksi*. Yogyakarta.

- Sri Kusumadewi & Hari Purnomo. 2004. *Aplikasi Logika Fuzzy Untuk Sistem Pendukung Keputusan Edisi Pertama*. Yogyakarta: Graha Ilmu.
- Sri Kusumadewi & Sri Hartati. 2006. *Neuro Fuzzy-Integrasi Sistem Fuzzy dan Jaringan Syaraf*. Yogyakarta: Graha Ilmu.
- Tata Sutabri. 2005. *Sistem Informasi Manajemen*. Yogyakarta: Andi.
- Tedy Rismawanet al. 2008. *Sistem Pendukung Keputusan Berbasis Pocket PC Sebagai Penentu Status Gizi Menggunakan Metode KNN (K-Nearest Neighbor)*. <http://journal.uii.ac.id/index.php/jurnal-teknoin/article/viewFile/793/711>. Tanggal akses: 21 Mei 2017.
- Turban, E. 1995. *Decision Support and Expert System, fourth edition*. New Jersey: Prentice-Hall International Inc.
- Turban, E, Aronson, Jay E & Liang, Teng-Ping. 2005. *Decision Support Systems and Intelligent Systems Edisi 7 Jilid 2*. Yogyakarta: Andi.
- Turban E, Rainer, & Potter. 2006. *Introduction to Information Technology (Pengantar Teknologi Informasi)*. Jakarta: Salemba Infotek.
- Verzello, Robert J. & Reuter II, John. 1982. *Data Processing: Systems and Concepts*. Tokyo: McGraw-Hill Kogakusha, Ltd.
- Wang, Lin Xin. 1997. *A Course in Fuzzy Systems and Control*. Upper Sadle River, New Jersey: Prentice Hall.