BAB II

TINJAUAN PUSTAKA

2.1 Gambaran Umum Perusahaan

Perusahaan listrik ini mengalami beberapa kali perubahan nama dan sekarang perusahaan ini diberi nama PT. PLN (Persero) Distribusi Jawa Barat dan Banten. Berikut ini adalah uraian tentang sejarah, struktur organisasi dan tugas pokok perusahaan.

2.1.1 Sejarah PT. PLN (Persero) Distribusi Jawa Barat dan Banten

a. Masa Penjajahan Belanda

Pada tahun 1905 Indonesia masih dalam penjajahan Hindia-Belanda. Pemerintah Hindia-Belanda memberi izin kepada pengusaha Belanda untuk mendirikan perusahaan listrik di daerah Bandung dengan nama *bandoengsche electriciteit maatschappij*, yang disingkat B.E.M dengan membangun jaringan-jaringan listrik di kota Bandung dan sekitarnya. Pada tahun 1913 pengusahaan pembangkitan dan penyaluran tenaga listrik dilaksanakan oleh I.W.D yang mempunyai wilayah kerja diseluruh Indonesia.

Pada tahun 1919 dibuat akta pendirian perusahaan pelistrikan dihadapan notaris Mr.Andrian Hendrik Van Ophusien No. 213 tanggal 31 Desember 1919. Maka perusahaan listrik pertama yang ada di Bandung dengan nama B.E.M dihapuskan, kemudian digabungkan kedalam suatu perusahaan terbatas dengan nama G.E.B.E.O.N.V (*gemeenschaappijk electriciteit bendrijk voor bandoengsch*

omterken namloose vennotschap) yang bertugas melaksanakan penguasaan distribusi tenaga listrik di Bandung dan sekitarnya. Dan diresmikan pada tanggal 1 Januari 1921 berdiri di Bandung yang merupakan perusahaan swasta yang mendapat izin konsensi di Jawa Barat kecuali kota Cirebon, Jakarta dan Tangerang.

b. Masa Penjajahan Jepang

Pada masa penguasaan pelistrikan beralih dari pemerintahan Hindia-Belanda ke pemerintahan Jepang, baik perindustrian maupun pembangkitan dan penyaluran tenaga listrik diusahakan oleh apa yang dinamakan *djawa denki djigyo sha bandoeng shisa*. Sedangkan untuk pembangkitan dan penyalurannya diusahakan oleh instansi-instansi pelaksana yaitu pada periode 1942 sampai dengan 1943 dilaksanakan oleh *seibo denki djigyo shi* dan pada periode 1943 sampai dengan 1945 oleh *denki kosha* dengan wilayah kerja diseluruh pulau Jawa.

Pada tahun 1945 sehubungan telah dibomnya kota Hiroshima dan Nagasaki oleh Amerika Serikat maka pemerintahan Jepang menyerah tanpa syarat kepada sekutu dan Indonesia pada tanggal 17 Agustus 1945 berkesempatan memproklamasikan kemerdekaannya.

Pada tahun 1948 terjadi agresi Belanda, sehubungan dengan itu pemerintah Republik Indonesia hijrah ke Yogyakarta maka penguasaan dan pendistribusian listrik khususnya di Jawa Barat termasuk Jakarta diusahakan oleh G.E.B.E.O.N.V kembali. Sedangkan untuk pembangkitan dan penyaluran tetap dikuasai oleh pemerintah Republik Indonesia yaitu oleh perusahaan negara untuk

pembangkitan tenaga listrik disingkat PENEPETEL dengan wilayah kerjanya diseluruh Jawa Barat termasuk Jakarta.

c. Setelah Proklamasi Kemerdekaan

Pada tahun 1957 terjadi nasionalisasi perusahaan milik asing, maka G.E.B.E.O.N.V diambil alih tepatnya pada tanggal 27 Desember 1957. sehubungan dengan nasionalisasi ini maka dikeluarkan Peraturan Pemerintah No. 86 Tahun 1958, Peraturan Pemerintah No. 18 Tahun 1959, Peraturan Pemerintah Pengganti Undang-undang No. 19 Tahun 1960 tentang Perusahaan Negara, maka berdasarkan Peraturan Pemerintah No. 18 Tahun 1959 tersebut maka G.E.B.E.O.N.V di Bandung dinyatakan menjadi Perusahaan Milik Negara dengan nama Perusahaan Listrik Negara (PLN).

Tahun 1961 dibentuk Badan Pimpinan Umum Perusahaan Listrik Negara (BPU-PLN) berdasarkan Undang-undang No. 19 Tanggal 30 April 1960. ditetapkan pula SK Menteri Pekerjaan Umum dan Tenaga Kerja No. 16/I/20 tanggal 20 April 1961 bahwa BPU-PLN terdiri atas XIV Exploitasi, yaitu sebagai berikut :

a. Exploitasi I sampai X, menguasai dan mengelola pembangkitan dan pengeluaran atau pendistribusian tenaga listrik diluar wilayah Jawa Barat.
b. Exploitasi XI sampai XIII, menguasai pembangkitan dan pengeluaran atau pendistribusian tenaga listrik wilayah Jawa Barat dan Jakarta Raya.
c. Exploitasi IX sampai XIV, menguasai dan mengelola Perusahaan Gas.

Karena BPU-PLN mencakup Perusahaan Gas, maka berdasarkan Peraturan Menteri Pekerjaan Umum dan Tenaga Kerja No.9/PRT/1964 tanggal 28

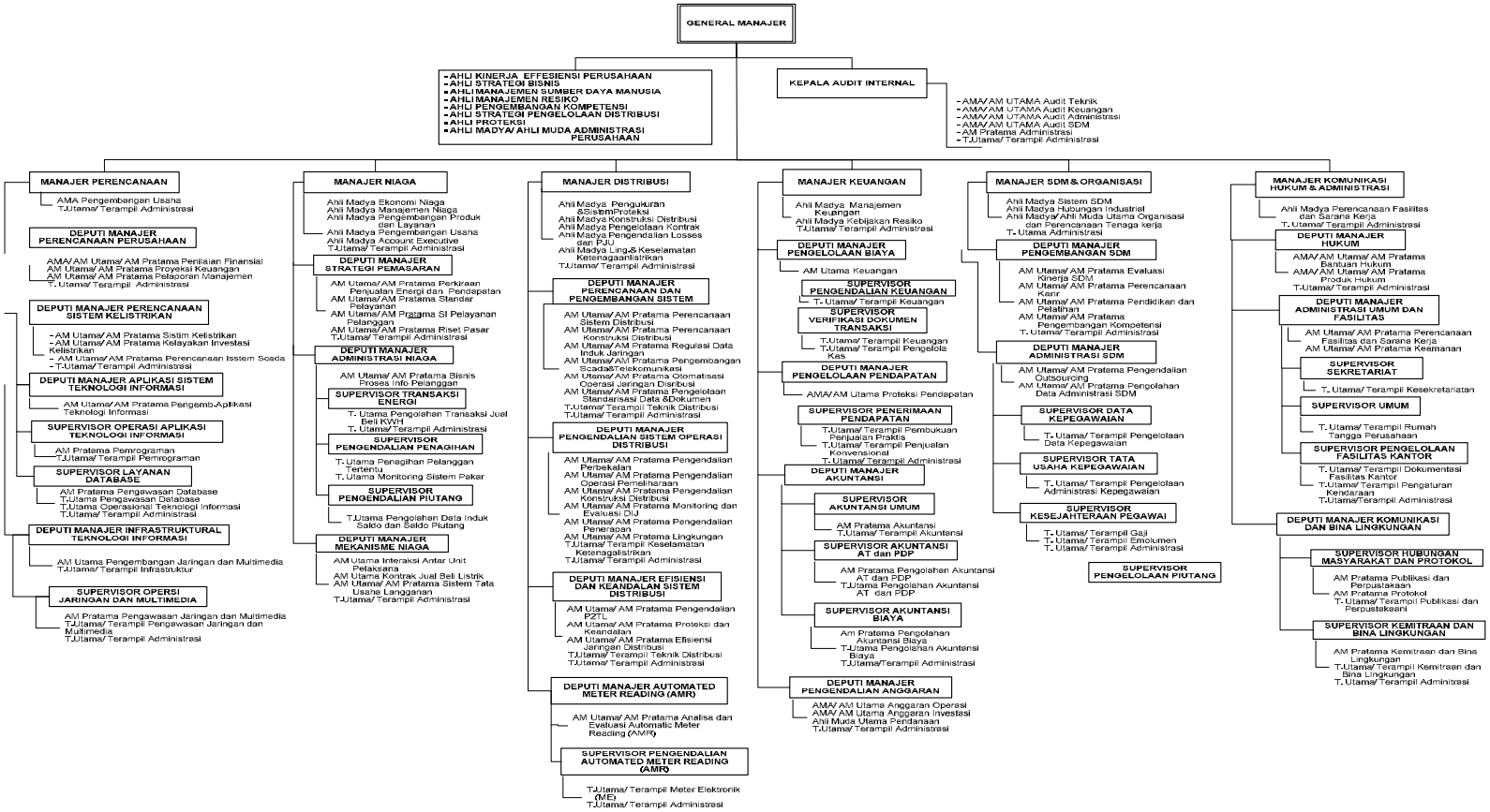
Desember 1964, BPU-PLN dibebaskan dan sekaligus mendirikan perusahaan Listrik Negara (PLN) dan Perusahaan Gas Negara (PGN) yang masing-masing direksinya terpisah.

Pada tahun 1972 terjadi perubahan dari Perusahaan Listrik Negara (PLN) menjadi Perusahaan Umum Listrik Negara (PULN). Hal ini diatur dalam Peraturan Pemerintah No. 18 Tahun 1972 tentang Perusahaan Listrik Negara.

PLN yang didirikan dengan Peraturan Pemerintah No. 19 Tahun 1965 ditegaskan statusnya menjadi Perusahaan Umum (Perum) sebagaimana yang tercantum dalam pasal 2 ayat 2 Undang-undang No. 9 Tahun 1969 dengan nama Perusahaan Listrik Negara (PLN). Berdasarkan Peraturan Menteri PUTL No. 013/PRT/1975 tanggal 8 September 1975, PLN mengadakan reorganisasi yang menyangkut nama, tugas, dan wilayah-wilayah kerja di daerah. Kemudian berdasarkan pengumuman PLN Exploitasi XI No. 05/DIII/Sek/1975 tanggal 14 Juli 1975, PLN exploitasi XI diubah namanya menjadi Perusahaan Umum Listrik Negara Distribusi Jawa Barat. Dengan adanya Peraturan Pemerintah Republik Indonesia No. 23 Tahun 1994 tanggal 16 Juni 1994 tentang pengalihan bentuk Perusahaan Umum Listrik Negara menjadi Perusahaan Perseroan (Persero) Distribusi Jawa Barat sejak tanggal 30 Juli 1994.

Dalam rangka program restrukturisasi, menjadi Strategis Businness Unit (SBU). PT. PLN (Persero) Distribusi Jawa Barat melalui keputusan Direksi No. 161.K/1010/DIR/2000 tanggal 20 September 2000 mendapat status baru menjadi PT. PLN (Persero) Distribusi Jawa Barat dan Banten.

2.1.2 Struktur Organisasi PT. PLN (Persero) Distribusi Jawa Barat dan Banten



Gambar 2.1 Struktur Organisasi Perusahaan

2.1.3 Visi, Misi dan Filosofi PT. PLN (Persero) Distribusi Jawa Barat dan Banten

Visi berisi pandangan jauh kedepan mengenai cita dan citra yang ingin diwujudkan oleh unit kerja yang bersangkutan pada masa yang akan datang. Visi memberikan gambaran kearah mana unit kerja akan dibawa dan bagaimana agar unit kerja tetap eksis, konsisten, antisipasif, inovatif dan produktif.

Guna mencapai visi tersebut maka dibutuhkan misi, karena misi adalah suatu cara bagaimana perusahaan mencapai visi/ tujuan suatu organisasi/ perusahaan dan misi tersebut dapat diwujudkan dengan strategi, cara, sarana dan pedoman berfikir sebagai langkah untuk menuju kondisi dimasa depan. Misi berisi suatu yang harus diemban oleh unit kerja sesuai dengan visinya. Berdasarkan definisi visi dan misi diatas, serta mengacu pada visi dan misi PT. PLN secara nasional. Berikut ini adalah visi, misi dan filosofi PT. PLN (Persero).

A. Visi PT. PLN (Persero) Distribusi Jawa Barat dan Banten

1. Mempertahankan posisi sebagai market share
2. Mewujudkan Perusahaan setara dengan kelas dunia
3. Sumber Daya Manusia yang Profesional
4. Aktivitas usaha akrab lingkungan

B. Misi PT. PLN (Persero) Distribusi Jawa Barat dan Banten

1. Memberikan kontribusi dalam pembangunan Nasional
2. Melakukan usaha sesuai dengan kaidah ekonomi yang sehat
3. Menjaga kualitas produk
4. Memuaskan Pelanggan

C. Filosofi PT. PLN (Persero) Distribusi Jawa Barat dan Banten

Filosofi adalah landasan utama sebagai keyakinan untuk mewujudkan visi dan misi perusahaan. Landasan filosofi PT. PLN (Persero) adalah “Mempunyai komitmen yang tinggi terhadap kepentingan pelanggan dengan menjadikan Sumber Daya Manusia sebagai sumber daya penting perusahaan”.

2.1.4 Tugas Pokok dan Fungsi PT. PLN (Persero) Distribusi Jawa Barat dan Banten

Tugas pokok PT. PLN (Persero) adalah memberikan pelayanan kepada masyarakat umum yaitu berusaha untuk meningkatkan kesejahteraan bidang kelistrikan sesuai dengan kemajuan teknologi. Realisasinya adalah memperkuat daerah jangkuaan listrik sampai ke pedesaan. Untuk menyelenggarakan tugas tersebut, PT. PLN (Persero) mempunyai fungsi sebagai berikut :

1. Menyediakan listrik bagi kepentingan umum dan sekaligus memupuk keuntungan berdasarkan prinsip pengelolaan perusahaan.
2. Mengusahakan penyediaan tenaga listrik dalam jumlah dan mutu yang memadai.
3. Merintis kegiatan-kegiatan usaha penyediaan tenaga listrik dengan tujuan untuk :
   a. Meningkatkan kesejahteraan dan kemakmuran rakyat secara adil dan merata serta mendorong kegiatan ekonomi.
   b. Mengusahakan keuntungan agar dapat membiayai pengembangan penyediaan tenaga listrik untuk melayani kebutuhan masyarakat.

2.2 Soft Computing

Soft Computing merupakan inovasi baru dalam membangun sistem cerdas. Sistem cerdas ini merupakan sistem yang memiliki keahlian seperti manusia pada domain tertentu, mampu beradaptasi dan belajar agar dapat bekerja lebih baik jika terjadi perubahan lingkungan.[2]

Menurut Prof. Lotfi A.Zadeh yang dikutip Sri Kusumadewi (2003:p.7) mengungkapkan :

"*Soft Computing* adalah Koleksi dari beberapa metodologi yang bertujuan untuk mengekspoitasi adanya toleransi terhadap ketidaktepatan, ketidakpastian dan kebenaran parsial untuk dapat diselesaikan dengan mudah dan biaya penyelesaian yang murah"

2.3 Logika *Fuzzy*

Sistem *fuzzy* pertama kali ditemukan oleh Prof. Lotfi Zadeh pada pertengahan tahun 1960 di Universitas California, Berkeley. Sistem ini diciptakan karena boolean logic tidak mempunyai ketelitian yang tinggi, hanya mempunyai logika 0 dan 1 saja. Sehingga untuk membuat sistem yang mempunyai ketelitian yang tinggi maka kita tidak dapat menggunakan boolean logic.

Penerapan teori logika ini dianggap mampu menciptakan sebuah revolusi dalam teknologi. Sebagai contoh, mulai tahun 90-an para manufaktur industri yang bergerak di bidang *Distributed Control system (DCSs), Programmable Controller (PLCs) dan Microcontroller (MCUs)* telah menyatukan sistem logika *fuzzy* pada barang produksi mereka dan memiliki prospek ekonomi yang baik. Pada saat yang bersamaan, pertumbuhan yang luar biasa terjadi pada industri

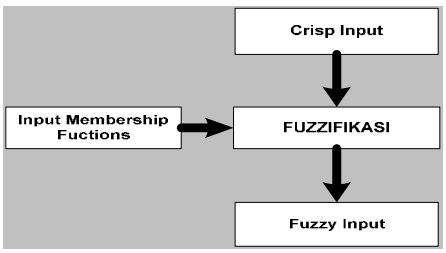
perangkat lunak yang menawarkan kemudahan penggunaan logika *fuzzy* dan penerapannya pada setiap aspek kehidupan sehari-hari.

2.3.1 Pengertian Logika *Fuzzy*

Logika *fuzzy* adalah suatu metode yang mengadopsi penilaian yang dilakukan manusia terhadap suatu kebenaran [2]. Yang diekspresikan dalam fungsi kontinue dari 0 sampai 1. Berbeda dengan logika klasik yang menyatakan segala hal dalam istilah *binary* (0 atau 1, ya atau tidak).

a. Fuzzifikasi

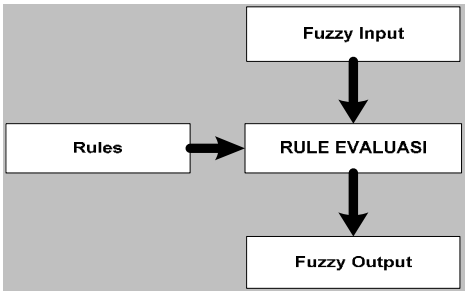
Merupakan suatu proses untuk mengubah suatu masukan dari bentuk tegas (*crisp*) menjadi peubah *fuzzy* (*variable linguistik)* yang biasanya disajikan dalam bentuk himpunan-himpunan *fuzzy* dengan fungsi keanggotaannya masing-masing.



Gambar 2.2 Proses Fuzzifikasi

b. Evaluasi Kaidah

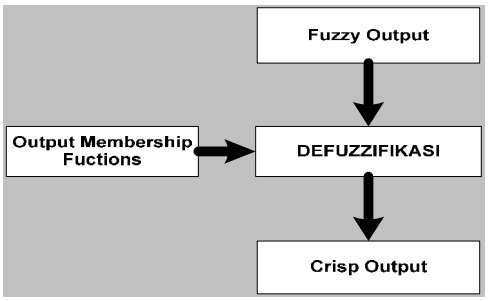
Merupakan proses pengambilan keputusan (*inference)* yang berdasarkan aturan-aturan yang ditetapkan pada basis aturan *(rules base)* untuk menghubungkan antara peubah-peubah *fuzzy* masukan dan peubah *fuzzy* keluaran. Aturan-aturan ini berbentuk IF…THEN. Proses ini berfungsi untuk mencari suatu nilai *fuzzy output* dari *fuzzy input*.



Gambar 2.3 Proses Rule Evaluasi

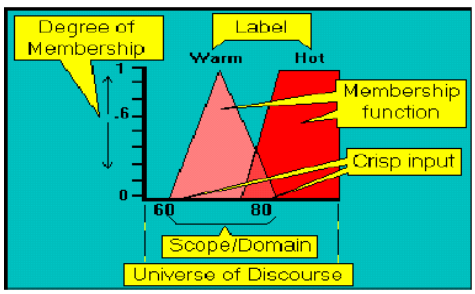
c. Defuzzifikasi

Merupakan proses pengubahan besaran *fuzzy* yang disajikan dalam bentuk himpunan-himpunan *fuzzy* keluaran dengan fungsi keanggotaannya untuk mendapatkan kembali bentuk tegasnya. Hal ini diperlukan karena *plant* hanya mengenal nilai tegas sebagai besaran sebenarnya untuk regulasi prosesnya. Proses ini berfungsi untuk menentukan suatu nilai *crisp output*.



Gambar 2.4 Proses Defuzzifikasi

Secara keseluruhan istilah yang digunakan dalam sistem *fuzzy* adalah sebagai berikut :



Gambar 2.5 Istilah yang digunakan dalam fuzzy

***a. Degree of membership* (Derajat Keanggotaan)**

Fungsi dari derajat keanggotaan adalah untuk memberikan bobot pada suatu input yang kita berikan, sehingga input tadi dapat dinyatakan dengan nilai. Batas dari derajat keanggotaan adalah dari 0 sampai 1.

b. *Scope* (Domain)

Merupakan suatu batas dari kumpulan *input* tertentu. Misalnya suhu dingin adalah 10-50 derajat, sangat cepat adalah dari 200-500 rpm.

c. Label (Himpunan)

Adalah kata-kata untuk memberikan suatu keterangan pada Domain. Contohnya : panas, dingin, cepat, sangat cepat.

***d. Membership Function* (Fungsi Keanggotaan)**

Suatu bentuk bangun yang merepresentasikan suatu batas dari domain.

e. Crisp Input

Nilai input analog yang kita berikan untuk mencari derajat keanggotaan.

***f. Universe of discourse* (Semesta Pembicaraan)**

Batas input yang telah kita berikan dalam merancang suatu *fuzzy* sistem. Batas ini berbeda dengan batas *scope*/ domain. *Universe of discourse* adalah batas semua input yang akan diberikan sedangkan *scope*/ domain adalah suatu batas yang menentukan bahwa *input* tersebut dinyatakan panas, dingin, cepat.

2.3.2 Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan satu item x dalam suatu himpunan A, yang sering ditulis dengan $\mu_A[x]$, memiliki 2 kemungkinan, yaitu :

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

Jika pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 dan 1, pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1. apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A[x]=0$ berarti x tidak menjadi anggota himpunan A, demikian pula apabila x memiliki nilai keanggotaan *fuzzy* $\mu_A[x]=1$ berarti x menjadi anggota penuh pada himpunan A.

Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval [0,1], namun interpretasi 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.

Himpunan *fuzzy* memiliki dua atribut, yaitu :

a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : Dingin, Sejuk, Hangat, dan Panas
b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel, seperti 40, 25, 50 dsb.

2.3.2.1 Operator Dasar Zadeh Untuk Operasi Himpunan *Fuzzy*

Ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α-predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu :

a. Operator AND

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

$$\mu A \cap B = \min(\mu A[x], \mu B[y])$$

b. Operator OR

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

$$\mu A \cup B = \max(\mu A[x], \mu B[y])$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

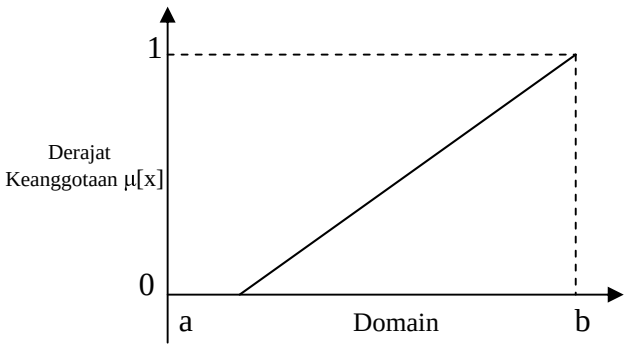
$$\mu A' = 1 - \mu A[x]$$

2.3.3 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (derajat keanggotaan) yang memiliki interval antara 0 dan 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan, diantaranya adalah sebagai berikut :

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 keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.

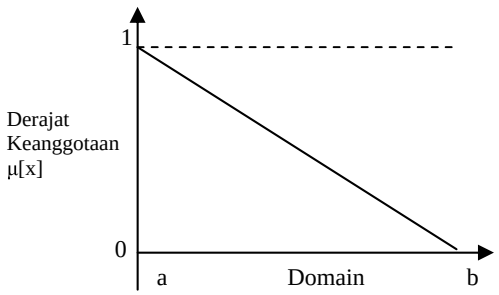


Gambar 2.6 Representasi Linear Naik

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \le a \\ (x-a)/(b-a); & a \le x \le b \\ 1; & x \ge b \end{cases} \quad (2.i)$$

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.



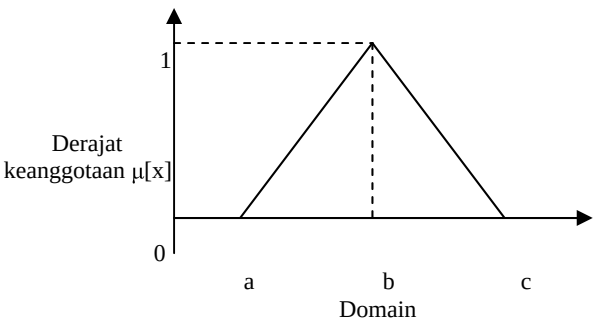
Gambar 2.7 Representasi Linear

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} (b-x)/(b-a); & a \le x \le b \\ 0; & x \ge b \end{cases} \qquad \text{(2.ii)}$$

b. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear).



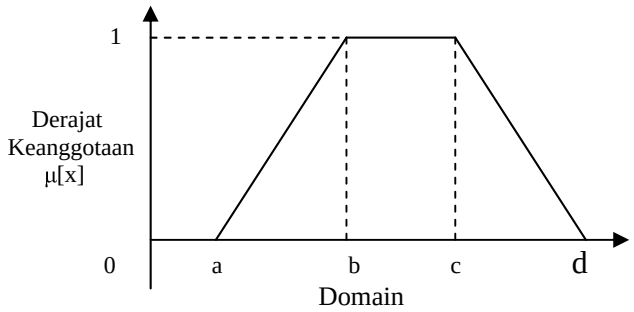
Gambar 2.8 Kurva Segitiga

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \le a, x \ge c \\ (x-a)/(b-a); & a \le x \le b \\ (c-x)/(c-b); & b \le x \le c \end{cases} \qquad \text{(2.iii)}$$

c. Representasi Kurva Trapesium

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



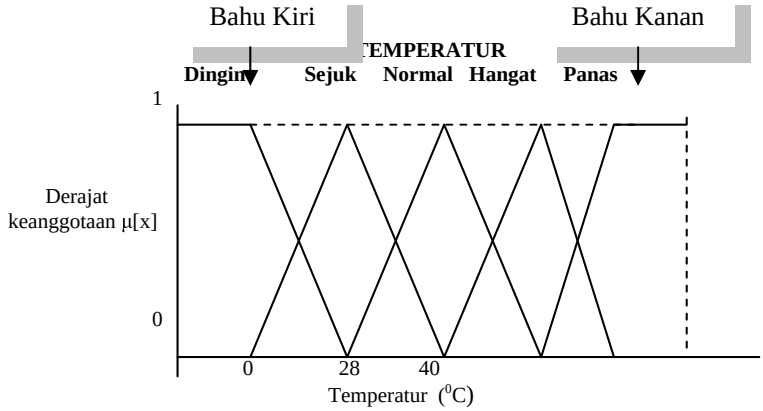
Gambar 2.9 Kurva Trapesium

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \le a, x \ge c \\ (x-a)/(b-a); & a \le x \le b \\ 1; & b \le x \le c \\ (d-x)/(d-c); & x \ge d \end{cases} \qquad \text{(2.iv)}$$

d. Representasi Kurva Bentuk Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan 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 salah ke benar.



Gambar 2.10 Daerah 'bahu' pada variabel

e. Representasi Kurva-S

Kurva PERTUMBUHAN dan PENYUSUTAN merupakan kurva S atau sigmoid yang berhubungan dengan kenaikan dan penurunan permukaan secara tak linear. Kurva S untuk pertumbuhan akan bergerak dari sisi paling kiri (nilai keanggotaan=0) ke sisi paling kanan (nilai keanggotaan=1). Fungsi keanggotaannya akan tertumpu pada 50% nilai keanggotaannya yang sering disebut dengan titik infleksi. Kurva untuk PENYUSUTAN akan bergerak dari sisi paling kanan (nilai keanggotaan = 1) ke sisi paling kiri (nilai keanggotaan = 0). Kurva S didefinisikan dengan menggunakan 3 parameter, yaitu : nilai keanggotaan nol (α), nilai keanggotaan lengkap (γ), dan titik infleksi atau *crossover* (β) yaitu titik yang merupakan domain 50% benar.

f. Representasi Kurva Bentuk Lonceng (*Bell Curve*)

Untuk merepresentasikan bilangan *fuzzy*, biasanya digunakan kurva berbentuk lonceng. Kurva berbentuk lonceng ini terbagi atas 3 kelas, yaitu : himpunan *fuzzy* PI, beta dan Gauss

g. Koordinat Keanggotaan

Himpunan *fuzzy* berisi urutan pasangan berurutan yang berisi nilai domain dan kebenaran nilai keanggotaanya dalam bentuk :

Skalar (i) / Derajat (i)

'Skalar' adalah suatu nilai yang digambar dari domain himpunan *fuzzy*, sedangkan 'Derajat' skalar merupakan derajat keanggotaan himpunan *fuzzy*nya.

2.3.4 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 adaah :

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 anteseden, sedangkan proposisi yang mengikuti THEN disebut sebagai konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator *fuzzy*, seperti :

IF (X_1 is A_1) • (X_2 is A_2) • (X_3 is A_3) •…..• (X_N is A_N) THEN y is B

dengan • adalah operator (misal : OR atau AND).

Secara umum, ada 2 fungsi implikasi yang dapat digunakan, yaitu :

a. Min (*minimum)*. Fungsi ini akan memotong *output* himpunan *fuzzy*.

b. Dot (*product)*. Fungsi ini akan menskala output himpunan fuzzy.

2.3.5 Sistem Interferensi *Fuzzy*

2.3.5.1 Metode Tsukamoto

Pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-THEN harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton (Gambar 2.16). sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α-predikat (*fire strength)*. Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

2.3.5.2 Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama Metode Max-Min. metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapat *output*, diperlukan 4 tahapan :

a. Pembentukan himpunan *fuzzy*

   Pada metode Mamdani, baik variabel input maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.

b. Aplikasi fungsi implikasi (aturan)

   Pada Metode Mamdani, fungsi implikasi yang digunakan adalah Min.

c. Komposisi aturan

   Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antara aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu : max, additive dan probabilistik OR (probor).

1. Metode Max (*Maximum*)

   Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan menggunakannya operator OR (union). Jika semua proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan konstribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan :

   $$\mu_{sf}[x_i] \leftarrow \max(\mu_{sf}[x_i], \mu_{kf}[x_i])$$

   dengan :

   $\mu_{sf}[x_i]$= nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

   $\mu_{kf}[x_i]$= nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

   Apabila digunakan fungsi implikasi MIN, maka metode komposisi ini sering disebut dengan nama MAX-MIN atau MIN-MAX atau MAMDANI.

2. Metode Additive (*Sum*)

   Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua output daerah *fuzzy*.

   Secara umum dituliskan :

   $$\mu_{sf}[x_i] \leftarrow \min(1, \mu_{sf}[x_i] + \mu_{kf}[x_i])$$

   dengan :

   $\mu_{sf}[x_i]$=nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

   $\mu_{kf}[x_i]$= nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

3. Metode Probabilistik (*probor*)

   Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *product* terhadap semua output daerah *fuzzy*. Secara umum dituliskan :

$$\mu_{sf}[x_i] \leftarrow (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i] * \mu_{kf}[x_i])$$

   dengan :

   $\mu_{sf}[x_i]$=nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

   $\mu_{kf}[x_i]$= nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

d. Penegasan (*defuzzy*)

   Input dari proses defuzifikasi 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 sebagai *output* .

   Ada beberapa metode defuzifikasi pada komposisi aturan MAMDANI, antara lain :

   1. Metode Centroid (*Composite Moment)*

      Pada metode ini, *crisp* diperoleh dengan cara mengambil titik pusat (z*) daerah *fuzzy*.

      Secara umum dirumuskan :

$$z^* = \frac{\int_z z\,\mu(z)\,dz}{\int_z \mu(z)\,dz} \longrightarrow \text{Untuk semesta kontinue}$$

$$z^* = \frac{\sum_{j=1}^{n} z_j\,\mu(z_j)}{\sum_{j=1}^{n} \mu(z_j)} \longrightarrow \text{Untuk semesta diskret}$$

2. Metode Bisektor

   Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan sebagian dari jumlah total nilai keanggotaan pada daerah *fuzzy*.

   Secara umum dituliskan :

   Z_p sedemikian hingga $\int_{R_n}^{p} \mu(z)dz = \int_{p}^{R_n} \mu(z)dz$

3. Metode *Mean Of Maximum* (MOM)

   Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

4. Metode *Largest of Maximum* (LOM)

   Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

5. Metode *Smallest of Maximum* (SOM)

   Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

2.3.5.3 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. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Apabila komposisi aturan menggunakan

metode SUGENO, maka defuzifikasi dilakukan dengan cara mencari nilai rata-ratanya.

a. Model *Fuzzy* Sogeno Orde-Nol

   Secara umum bentuk model *fuzzy* SUGENO Orde-Nol adalah :

   IF (X_1 is A_1)• (X_1 is A_1) •(X_1 is A_1) •…•(X_N is A_N) THEN z=k

   Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

b. Model *Fuzzy* Sugeno Orde-Satu

   Secara umum bentuk model *fuzzy* SUGENO Orde-Satu adalah :

   Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan p_i adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen.

   IF (X_1 is A_1)•…•(X_N is A_N) THEN z=p_1*x1 + … + $p_N X_N$ + q

   Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan pi adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen.

2.3.6 Database Fuzzy

Database (basisdata) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.[2] Sistem basis data *(database system)* adalah suatu sistem informasi yang mengintegrasikan kumpulan data yang saling berhubungan satu dengan yang lainnya.[2] Desain database dimaksudkan untuk mengidentifikasikan sebuah *file-file* database yang diperlukan untuk sistem informasi dan untuk mengidentifikasikan isi struktur dari tiap-tiap *file* yang diidentifikasikan, elemen-elemen data di suatu *file database*

harus dapat digunakan untuk pembuatan suatu *output*. Demikian juga dengan input yang akan direkam didalam *database*, *file-file database* harus mempunyai elemen untuk menampung input yang dimasukkan. Dengan demikian struktur dari suatu *file database* tergantung dari arus data masuk dan arus data keluar, dari atau ke *file* tersebut.

2.3.6.1 *Database Fuzzy* Model Tahani

Sebagian besar *database* standar diklasifikasikan berdasarkan bagaimana data tersebut dipandang oleh user. Dengan menggunakan *database* standar, kita dapat mencari data-data dengan spesifikasi tertentu dengan menggunakan *query*. Pada kenyataanya, seseorang kadang membutuhkan informasi dari data-data yang bersifat ambigu. Apabila hal ini terjadi, maka kita menggunakan *database fuzzy*. Selama ini, sudah ada beberapa penelitian tentang *database fuzzy*. *Database fuzzy* model Tahani masih tetap menggunakan relasi standar, hanya saja model ini menggunakan teori himpunan *fuzzy* untuk mendapatkan informasi pada *query*-nya.

2.3.6.2 *Database Fuzzy* Model Umano

Pada *database fuzzy* Model Umano, data-data yang ambigu diekspresikan dengan menggunakan distribusi posibilitas. Distribusi posibilitas merupakan nilai atribut dari suatu model relasi.

2.4 Borland Delphi 7.0

2.3.1 Sejarah Perkembangan Delphi

Delphi adalah bahasa pemrograman visual berbasis *windows* yang menggunakan bahasa pascal sebagai *compiler*. Keberadaan bahasa pemrograman delphi tidak dapat dipisahkan dari bahasa turbo pascal, karena delphi merupakan generasi penerus dari turbo pascal yang diluncurkan pada tahun 1983 oleh *Borland International Incorporation.*

Kecenderungan pengguna bahasa pemrograman visual untuk membangun sebuah aplikasi telah mendorong Borland membuat bahasa pemrograman yang baru. Pada tahun 1995 di perkenalkan kepada pengguna komputer, sebuah bahasa pemrograman visual berbasis bahasa pascal. Bahasa pemrograman tersebut diberi nama Borland Delphi. Pengguna delphi dapat mempersingkat waktu pemrograman karena kita tidak perlu lagi menulis kode program yang rumit dan panjang untuk menggambar, meletakkan dan mengatur komponen. Selain itu kita juga dapat menyusun aplikasi yang lebih interaktif.

Delphi menyediakan fasilitas yang lengkap untuk membangun suatu program aplikasi misalkan IDE (*Integrated Development Environment).* Dengan IDE maka kita akan sangat terbantu, karena semua kebutuhan pemrograman telah disediakan dalam suatu tampilan. IDE delphi terdiri dari Menu, SpeedBox, Component Pallete, Objek Inspector Form dan Edit Code.

Umumnya delphi lebih banyak digunakan untuk pengembangan aplikasi desktop dan *enterprise* berbasis *database*, tapi sebagai perangkat pengembangan yang bersifat *general-purpose* delphi juga mampu dan digunakan dalam berbagai

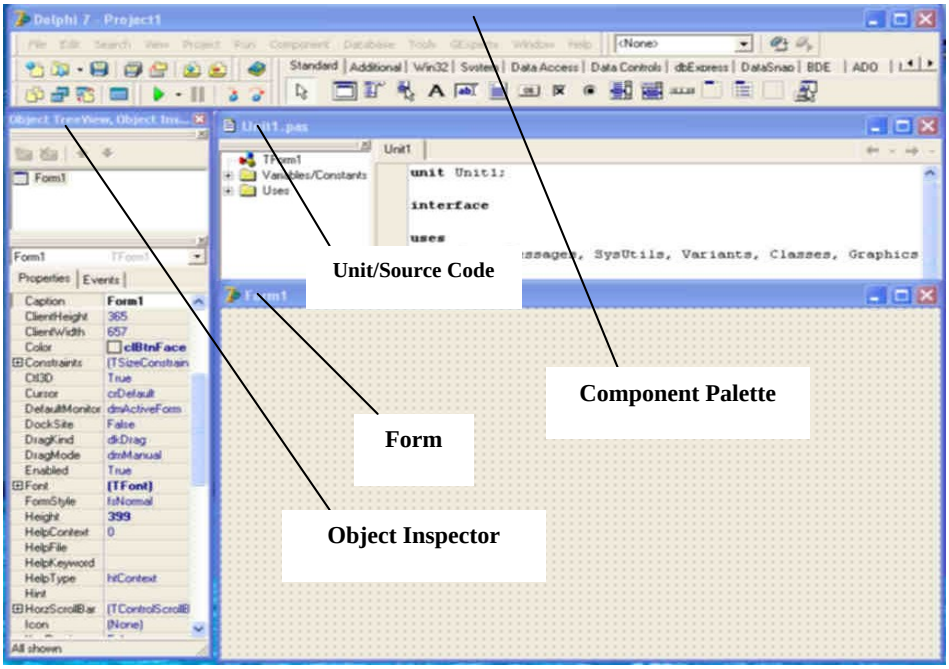
jenis proyek pengembangan *software*. Delphi juga yang dikenal sebagai salah satu yang membawa istilah *RAD (Rapid Application Development)* tool, saat dirilis tahun 1995 untuk *windows* 16-bit. Delphi 2, dirilis setahun kemudian, mendukung lingkungan *windows* 32-bit, dan versi c++, C++Builder, dirilis beberapa tahun kemudian. Pada tahun 2001 sebuah versi linux yang dikenal sebagai Kylix tersedia. Dengan satu rilis baru setiap tahunnya, pada tahun 2002 dukungan untuk Linux (melalui Kylix dan CLX component library) ditambahkan dan tahun 2003 .NET mulai didukung dengan munculnya Delphi.Net (Delphi 8).

Chief Architect yang membidani Delphi, dan pendahulunya Turbo Pascal, adalah Anders Hejlsberg sampai kemudian ia pindah ke Microsoft tahun 1996 di mana ia sebagai *chief designer* C# dan termasuk orang kunci dalam perancangan Microsoft.Net Framework. Dukungan penuh untuk .Net ditambahkan pada Delphi 8 (dirilis pada bulan Desember 2003) dengan penampilan user interface (*look and feel*) mirip dengan Microsoft Visual Studio.NET.

Delphi 2005 (nama lain dari Delphi 9) mendukung *code generation* baik untuk win32 maupun .NET, dan seperti yang telah dikenal, fitur-fitur manipulasi data secara *live* dari database secara *design-time*. Ia juga membawa banyak pembaruan pada IDE secara signifikan.

Para penganjur delphi mengklaim dengan bahasa pemrograman Delphi, IDE dan component library (VCL/CLX) yang disediakan oleh vendor tunggal memungkinkan satu paket yang lebih konsisten dan mudah dikenali. Produk delphi ini didistribusikan dalam beberapa rancangan: *Personal*, *Professional*, *Enterprise* (sebelumnya *Client/Server*) dan *Architect*.

2.3.2 Struktur Program Borland Delphi 7.0



Gambar 2.11 Struktur Program Delphi 7.0

Fungsi dari elemen-elemen di atas adalah :

a. Object Inspector : suatu window yang berguna untuk mengatur suatu object baik properti, events dan method.

b. Form : digunakan sebagai layar/window yang digunakan sebagai lembar kerja kita. Di form-lah semua komponen seperti tombol dan komponen lainnya disimpan.

c. Window Unit/ Source Code : *Window*/layar yang berisi perintah-perintah yang akan dieksekusi oleh komputer. Di layar inilah mengisikan program-program

d. Component Palette : layar yang berisikan komponen-komponen yang dipakai dalam program.

2.4 MySQL

MySQL adalah *multi user database* yang menggunakan bahasa *Struktured Query Language* (SQL). MySQL mampu menangani data yang cukup besar.

SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user - friendly* dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah - perintah pemrograman.

Secara umum SQL dapat digunakan untuk melakukan proses penambahan data, penghapusan data, perbaikan data, melihat-lihat data yang tersimpan dan membentuk laporan, serta melakukan ekstraksi informasi dalam suatu database.

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.