

BAB 2

LANDASAN TEORI

2.1 Sistem Intelejen Visual

Berbicara tentang biometrik tentu tak akan lepas kaitannya dengan sistem visi komputer (*computer vision*) yang terdiri atas dasar bidang pengolahan citra (*image processing*) dan bidang pengenalan pola (*pattern recognition*). Komputer dapat melihat dan juga mengenali suatu objek benda yang sedang diamati/diobservasi dengan mengaplikasikan dan mengkombinasikan kedua proses bidang tersebut diatas.

2.1.1 Pengolahan citra

Untuk membuat komputer agar memiliki sebuah penglihatan yang dapat berguna melihat suatu objek gambar dengan jelas seperti selayaknya mata pada manusia, komputer harus melakukan proses pengolahan citra. Proses pengolahan citra merupakan bidang yang berhubungan dengan proses transformasi citra/gambar, membuat suatu citra agar dapat dikenali dengan baik oleh komputer. Tujuan utama proses ini adalah untuk mendapatkan kualitas citra yang lebih baik. Sedangkan aplikasi dari pengolahan citra dan *computer vision* dalam berbagai bidang atau disiplin ilmu pengetahuan, teknologi, dan kehidupan sehari-hari antara lain dapat dilihat dalam tabel:

Tabel 2.1 Aplikasi *Computer Vision* dan pengolahan citra

Pictorial Database	Quality Control of Food
Graphics Desain	Bacterial Growth Scanning
Engraving	Geological Modelling
Textile Design	Autonomous Navigation
Cartography	Cosmetics
Metallurgy	Astronomy
Materials Science	Defence Application
Medical Imaging	Particle Tracking
Object Recognition	Fishing Using Ultrasound
Microscopy	Pottery Blemish Identification
Satellite Picture Processing	Eye Retina Comparison
Particle Counting and Sizing	Fluid Mechanics
Typesetting	Journalism – Picture Enhancement
3D Reconstruction	Security – Picture Enhancement and Identification
Photographic Security	Weather Mapping
Fingerprint Matching	Document Transfer
Document Reading	Document Comparison

Setelah diperoleh citra digital yang berupa pola bit-bit di dalam memori komputer, maka analisis dan pengolahan dapat dilakukan. Pengolahan citra selalu melibatkan satu atau bahkan lebih algoritma yang akan

diimplementasikan terhadap citra. Ada beberapa tingkatan dalam pengolahan citra, yaitu tingkatan rendah, menengah, dan tinggi.

Pengolahan tingkat rendah dikaitkan dengan pengolahan pada citra biner, meningkatkan kualitas citra dengan satu atau beberapa algoritma khusus, dan merepresentasikan citra hasil pengolahan, memindahkan data yang tidak diinginkan, dan meningkatkan data yang diinginkan.

Sedangkan pengolahan tingkat menengah, mengidentifikasi bentuk yang terdapat dalam citra secara signifikan, daerah atau titik-titik dari citra biner.

Dan pengolahan tingkat tinggi, mengantar-mukakan citra dengan beberapa landasan informasi yang ada. Ini mengasosiasikan bentuk-bentuk yang mungkin ditemukan selama tahap-tahap pemrosesan menengah dengan bentuk yang ada dan nyata pada dunia sebenarnya. Hasil algoritma dari level ini dimodifikasi menjadi suatu prosedur non-citra yang memuat aksi-aksi yang mungkin terhadap analisis citra yang dijadikan referensi.

2.1.2 Pengenalan pola

Setelah komputer dapat melihat suatu objek gambar maka yang harus dilakukan agar objek yang dilihatnya bukan menjadi sekedar objek asal (*random object*), pengenalan pola harus diaplikasikan agar komputer dapat mendeteksi dan mengenali perbedaan tiap-tiap objek yang dilihatnya. Pengenalan pola adalah salah satu bidang visi komputer yang khususnya berhubungan dengan proses identifikasi objek yang terdapat pada suatu citra atau dengan kata lain interpretasi citra. Proses ini bertujuan untuk mendeteksi

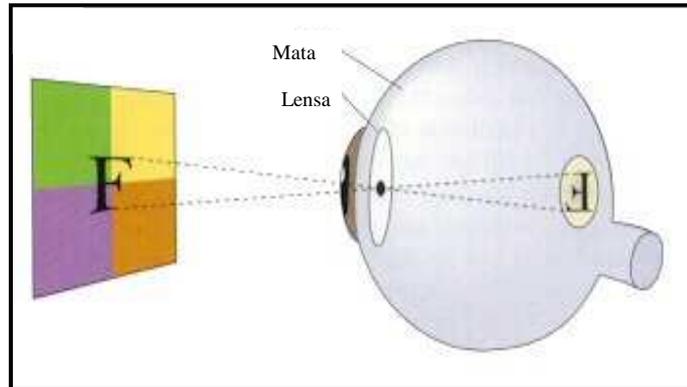
dan mengekstraksi informasi-informasi yang disampaikan oleh citra yang sedang dilihat.

Tanpa adanya mata manusia tidak akan bisa melihat, hal yang sama terjadi pada komputer jika tidak mempunyai mata. Agar dapat melihat komputer harus diberikan indera penglihatan, yaitu sebuah sistem penglihatan. Sistem penglihatan komputer harus memiliki fitur-fitur yang sama kegunaannya seperti yang ada pada sistem penglihatan manusia agar benar-benar dapat menunjukkan kegunaannya sebagai alat penglihatan.

Manusia memiliki indera penglihatan yang mempunyai beberapa fitur utama didalamnya yang dapat menjadikan penglihatannya bekerja dengan baik, fitur-fitur itu adalah:

- Mata, sebagai penangkap citra
- Retina, untuk pengaturan citra pada beberapa lapisan
- Pembeda warna
- Ukuran penglihatan yang dinamis

Untuk lebih jelasnya silahkan perhatikan gambar 2.1 berikut ini.

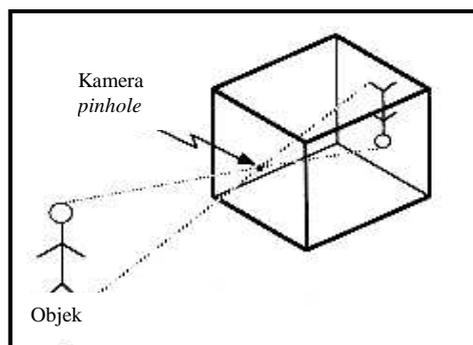


Gambar 2.1 Indera penglihatan pada manusia

Sedangkan sistem penglihatan pada komputer juga memiliki fitur-fitur juga bekerja selayaknya fitur yang ada pada mata manusia, pada umumnya adalah seperti ini:

- Kamera *pinhole*, sebagai penangkap citra
- Susunan CCD, untuk pengaturan informasi citra
- Alat RGB, sebagai pembeda warna
- Jangkauan penglihatan dapat diatur secara geometris

Penjelasan lebih lanjut dapat melihat pada gambar 2.2 di bawah.



Gambar 2.2 Sistem penglihatan pada computer

Sistem visi komputer masih terus dikembangkan agar dapat memiliki kemampuan tingkat tinggi (*high level capability*) sebagaimana yang dimiliki oleh sistem visual manusia. Kemampuan tingkat tinggi mencakup bidang seperti pendeteksian objek, pengenalan objek, pendeskripsian objek, pembuatan objek tiga dimensi dari citra dua dimensi, dan penerjemahan/mengartikan suatu gerakan.

Jika sistem visi komputer dapat melihat dan mengenali objek serta mengekstraksi informasi yang ada pada sebuah objek, maka untuk mengidentifikasi ataupun mengautentifikasi citra yang sedang diobservasi dan membuat sebuah objek asal menjadi sebuah objek yang mempunyai identitas yang dapat dibedakan satu dengan lainnya dibutuhkan inteligensia semu (*artificial intelligence*) untuk dapat membedakannya. Dengan penggunaan inteligensia semu komputer dapat menentukan identitas sebuah objek dengan menggunakan berbagai informasi yang didapatnya sehingga komputer dapat melihat dan mampu membedakan bermacam-macam citra seperti halnya manusia yang melihat dengan mata dan menggunakan otak untuk membedakan objek-objek yang dilihatnya.

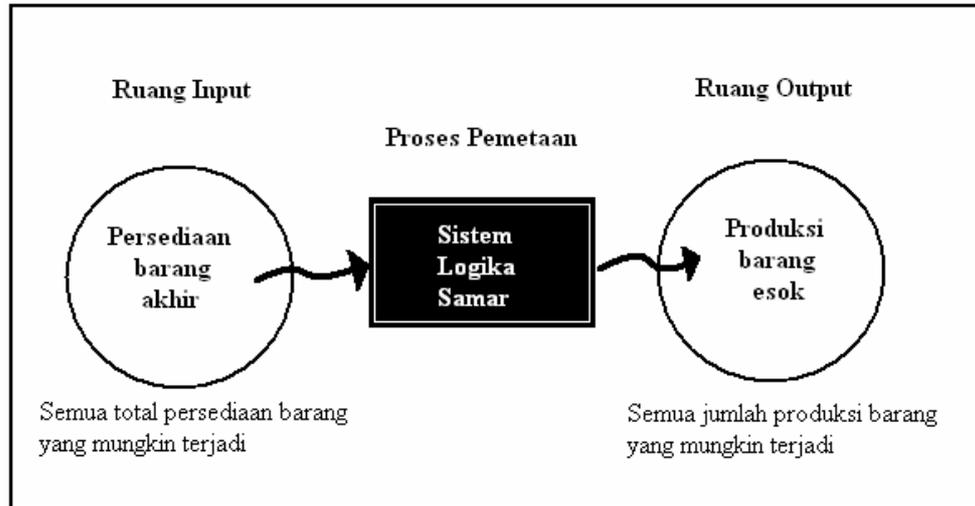
Sistem seperti ini, yang menggabungkan sistem penglihatan komputer dengan sistem inteligensia semu, adalah sistem yang dinamakan sistem intelijen visual (*visual intelligence system*). Dengan kombinasi antara kedua sistem “mata” dengan sistem “otak” pada komputer, maka sistem intelijen visual ini akan dapat melahirkan sebuah teknologi canggih yang sering kita sebut dengan teknologi biometrik.

2.2 Logika samar

Salah satu pendekatan yang dapat digunakan sebagai intelijensia semu pada sistem intelijen visual adalah penggunaan logika samar (*fuzzy logic*). Logika samar merupakan kemampuan penalaran bagi komputer selayaknya rasa nalar bagi seorang manusia, yaitu dapat menarik kesimpulan dari suatu kumpulan masalah tanpa harus memerlukan data yang lengkap.

Secara teknis, logika samar adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam ruang output. Dengan berbagai variasi input yang mungkin terjadi, logika samar akan membuat peta yang tepat agar jalur output yang dihasilkan sesuai dengan inputnya. Sebagai contohnya:

- Manajer pergudangan yang memeriksa berapa banyak persediaan barang yang ada, kemudian diberitahukan kepada manajer produksi agar jumlah barang yang akan diproduksi dapat ditetapkan berdasarkan jumlah stok barang yang ada. Jika stok barang di gudang sedikit maka barang yang diproduksi akan meningkat, begitu pula sebaliknya.
- Pelayan di restoran yang memberikan pelayanan kepada seorang tamu. Kemudian tamu akan memberikan tip, sesuai dengan pelayanan yang diberikan. Jumlah tip tergantung dari pelayanan yang diberikan, semakin baik pelayanan maka tip yang diberikan semakin besar.



Gambar 2.3 Contoh pemetaan input output

Logika samar banyak digunakan orang karena memiliki beberapa keunggulan diantaranya adalah:

- Konsep logika samar mudah dimengerti, konsep matematis yang mendasari penalaran ini sangat sederhana dan mudah dimengerti.
- Logika samar sangat fleksibel.
- Logika samar memiliki toleransi terhadap data-data yang tidak lengkap.
- Logika samar mampu memodelkan fungsi-fungsi non-linier yang sangat kompleks.
- Logika samar dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa proses pelatihan.
- Logika samar dapat bekerja sama dengan teknik-teknik kendali secara konvensional.
- Logika samar didasarkan pada bahasa alami.

2.3 Himpunan samar

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

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

Contoh dari himpunan tegas:

Jika diketahui:

$S = (1, 2, 3, 4, 5, 6)$ adalah semesta pembicaraan

$A = (1, 2, 3)$

$B = (3, 4, 5)$

Bisa dikatakan bahwa:

- ❖ Nilai keanggotaan 2 pada himpunan A , $\mu_A[2]=1$, karena $2 \in A$
- ❖ Nilai keanggotaan 3 pada himpunan A , $\mu_A[3]=1$, karena $3 \in A$
- ❖ Nilai keanggotaan 4 pada himpunan A , $\mu_A[4]=0$, karena $4 \notin A$
- ❖ Nilai keanggotaan 2 pada himpunan B , $\mu_B[2]=0$, karena $2 \notin B$
- ❖ Nilai keanggotaan 3 pada himpunan B , $\mu_B[3]=1$, karena $3 \in B$

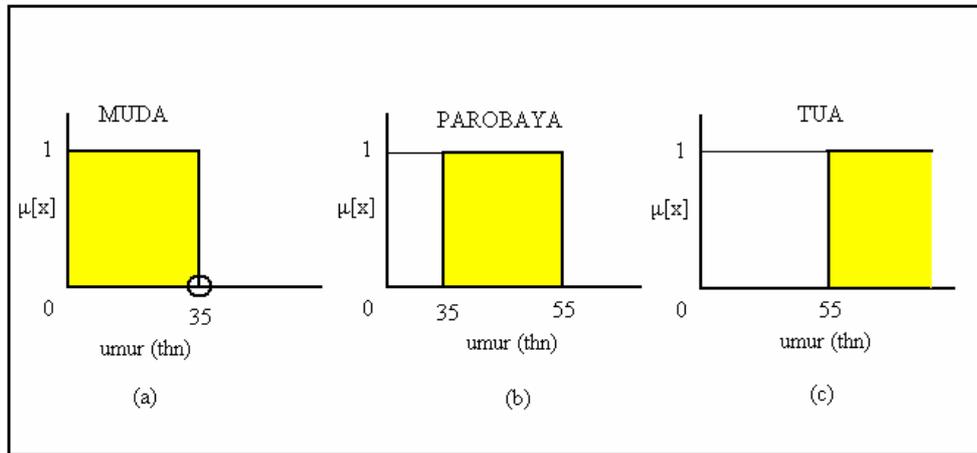
Contoh lainnya jika memakai variabel umur yang dibagi menjadi 3 kategori, yaitu:

MUDA umur < 35 tahun

PAROBAYA 35 tahun \leq umur \leq 55 tahun

TUA umur > 55 tahun

Nilai keanggotaan secara grafis, himpunan MUDA, PAROBAYA, dan TUA ini dapat dilihat pada gambar 2.4.



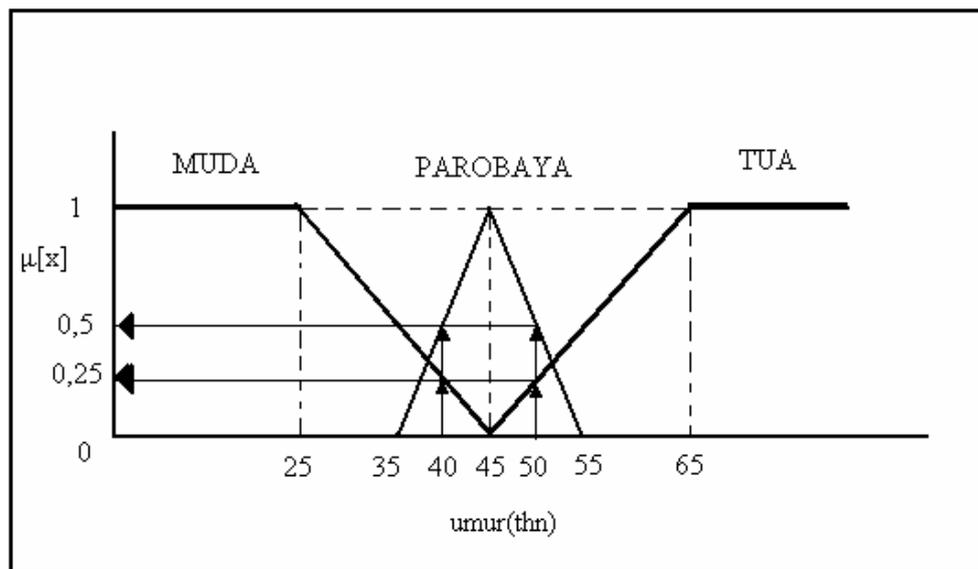
Gambar 2.4 Himpunan: (a) MUDA, (b) PAROBAYA, dan (c) TUA

Pada gambar 2.4 dapat dilihat bahwa:

- ❖ Apabila seseorang berusia 34 tahun, maka ia dikatakan MUDA ($\mu_{\text{MUDA}}[34] = 1$);
- ❖ Apabila seseorang berusia 35 tahun, maka ia dikatakan TIDAK MUDA ($\mu_{\text{MUDA}}[35] = 0$);
- ❖ Apabila seseorang berusia 35 tahun, maka ia dikatakan PAROBAYA ($\mu_{\text{PAROBAYA}}[35] = 1$);
- ❖ Apabila seseorang berusia 34 tahun, maka ia dikatakan TIDAK PAROBAYA ($\mu_{\text{PAROBAYA}}[34] = 0$);

Dapat terlihat bahwa pemakaian himpunan tegas untuk menyatakan umur sangat tidak adil, adanya perubahan kecil saja pada suatu nilai dapat mengakibatkan perbedaan kategori yang cukup signifikan.

Himpunan samar digunakan untuk mengantisipasi hal tersebut. Seseorang dapat masuk dalam dua himpunan yang berbeda, MUDA dan PAROBAYA ataupun PAROBAYA dan TUA. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya. Pada gambar 2.5 menunjukkan himpunan samar untuk variabel umur.



Gambar 2.5 Himpunan samar untuk variabel umur

Kita dapat melihat dari gambar diatas dengan jelas, bahwa seseorang dapat termasuk ke dalam dua himpunan umur:

- ❖ Seseorang yang berumur 40 tahun, termasuk dalam himpunan MUDA dengan $\mu_{MUDA}[40]=0,25$; namun dia juga termasuk ke dalam himpunan PAROBAYA dengan $\mu_{PAROBAYA}[40]=0,5$.
- ❖ Seseorang yang berumur 50 tahun, termasuk dalam himpunan PAROBAYA dengan $\mu_{PAROBAYA}[50]=0,5$; namun dia juga termasuk ke dalam himpunan TUA dengan $\mu_{TUA}[50]=0,25$.

Jika pada himpunan tegas, nilai keanggotaan hanya ada 2 kemungkinan, yaitu nol (0) atau satu (1), pada himpunan samar nilai keanggotaan terletak pada rentang antara nol (0) sampai satu (1). Apabila x memiliki nilai keanggotaan samar $\mu_A[x]=0$ berarti x tidak menjadi anggota himpunan A , demikian juga apabila x memiliki nilai keanggotaan samar $\mu_A[x]=1$ yang berarti x menjadi anggota penuh pada himpunan A .

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

Himpunan samar memiliki dua buah atribut, yaitu linguistik dan numeris. Linguistik merupakan penamaan suatu himpunan yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA. Sedangkan numeris, seperti asal katanya yaitu merupakan suatu nilai atau angka yang menunjukkan ukuran dari suatu variabel seperti: 25, 40, 50, dan sebagainya.

Dalam memahami sistem logika samar, perlu mengetahui beberapa hal tentang pengetahuan dasar dalam sistem samar, yaitu:

a. Variabel samar

Variabel samar merupakan variabel yang hendak dibahas dalam suatu sistem logika samar.

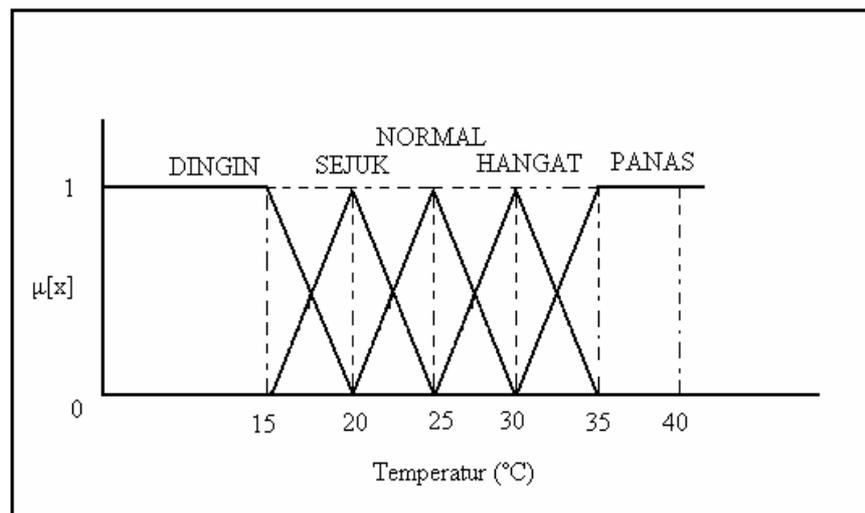
Contoh: umur, temperatur, permintaan, dan sebagainya.

b. Himpunan samar

Himpunan samar merupakan suatu himpunan atau grup yang mewakili suatu kondisi atau keadaan tertentu dalam variabel samar.

Contoh:

- Variabel umur, terbagi menjadi 3 himpunan samar, yaitu: MUDA, PAROBAYA, dan TUA. (Gambar 2.5)
- Variabel temperatur, yang terbagi menjadi 5 himpunan samar, yaitu: DINGIN, SEJUK, NORMAL, HANGAT, PANAS. (Gambar 2.6)



Gambar 2.6 Himpunan samar pada variabel temperatur.

c. Semesta pembicaraan

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

Contoh:

- Semesta pembicaraan untuk variabel umur: $[0 +\infty]$
- Semesta pembicaraan untuk variabel temperatur: $[0 40]$

d. Domain

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

Contoh domain himpunan samar:

- MUDA = $[0 45]$
- PAROBAYA = $[35 55]$
- TUA = $[45 +\infty]$
- DINGIN = $[0 20]$
- SEJUK = $[15 25]$

- NORMAL = [20 30]
- HANGAT = [25 35]
- PANAS = [30 40]

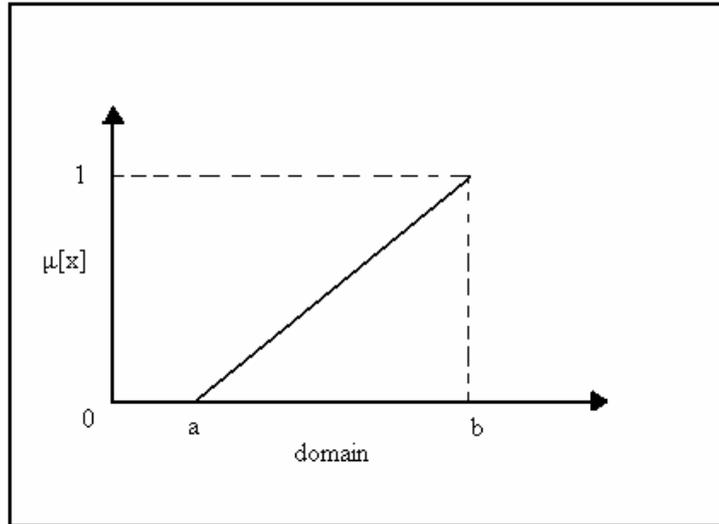
2.4 Fungsi keanggotaan

Membership function atau fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval antara nol sampai satu, nilai keanggotaan sering disebut juga sebagai derajat keanggotaan. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa pendekatan fungsi yang biasa digunakan:

a. Representasi linear

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

Ada dua keadaan himpunan samar 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.



Gambar2.7 Representasi linear naik

Fungsi Keanggotaan:

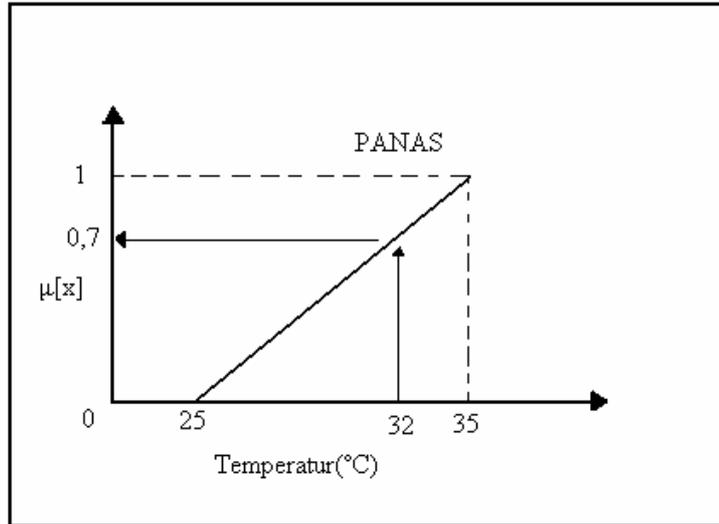
$$\mu[x] = \{ 0; x \leq a \}; \text{ atau}$$

$$\mu[x] = \{ (x - a) / (b - a); a \leq x \leq b \}; \text{ atau}$$

$$\mu[x] = \{ 1; x \geq b \}$$

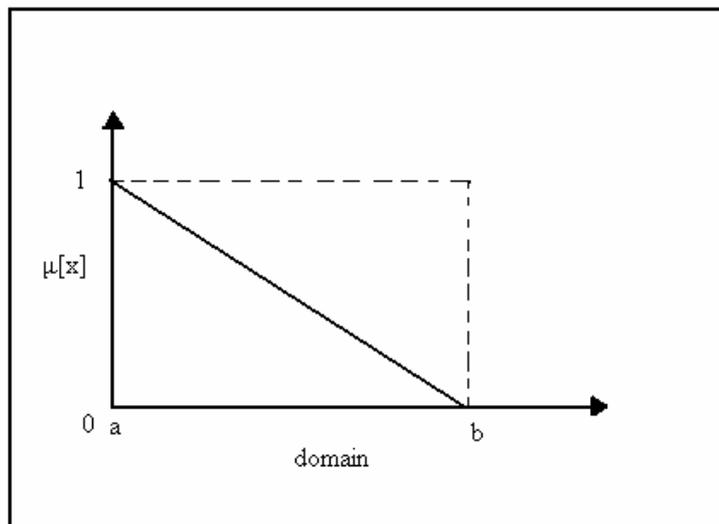
Contoh representasinya pada fungsi keanggotaan untuk himpunan PANAS pada variabel temperatur ruangan seperti terlihat pada gambar 2.8.

$$\begin{aligned} \mu_{\text{PANAS}}[32] &= (32-25)/(35-25) \\ &= 7/10 = 0,7 \end{aligned}$$



Gambar2.8 Himpunan samar: PANAS

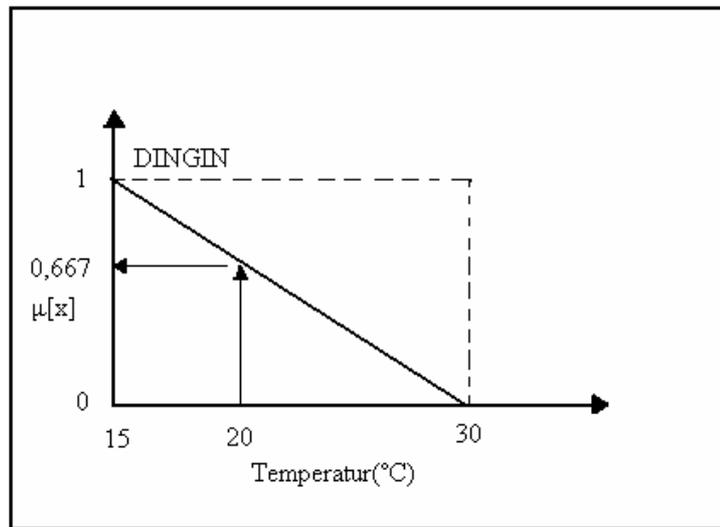
Sedangkan keadaan himpunan samar linear yang kedua adalah merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun menuju nilai domain yang memiliki derajat keanggotaan lebih rendah.



Gambar 2.9 Representasi linear turun

Contohnya dapat dilihat pada fungsi keanggotaan untuk himpunan DINGIN pada variabel temperatur ruangan seperti terlihat pada gambar 2.10.

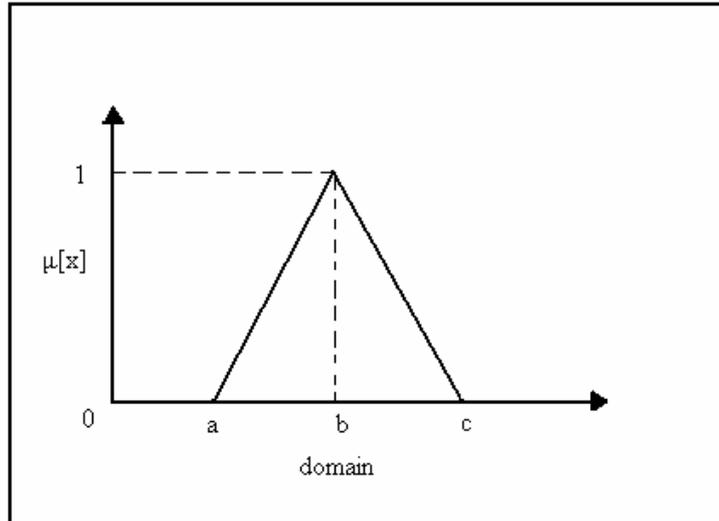
$$\begin{aligned}\mu_{\text{DINGIN}}[20] &= (30-20)/(30-15) \\ &= 10/15 = 0,667\end{aligned}$$



Gambar 2.10 Himpunan samar: DINGIN

b. Representasi kurva segitiga

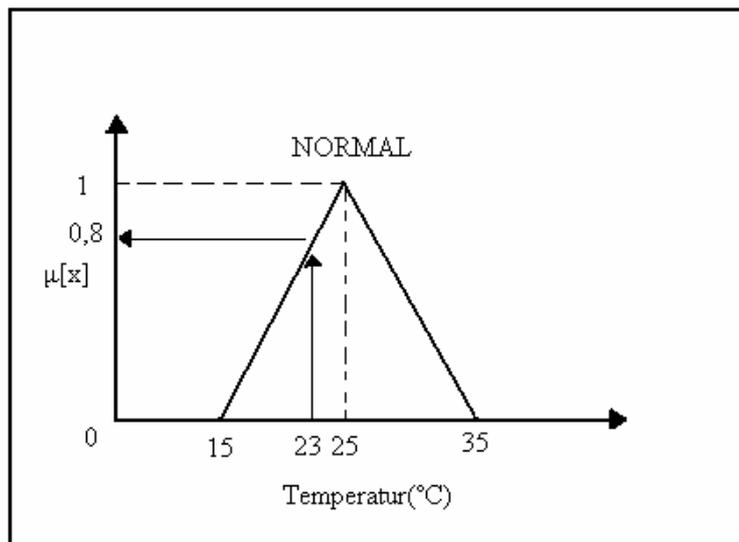
Kurva segitiga pada dasarnya merupakan gabungan antara dua garis linear yang saling bertemu pada ujungnya dan membentuk segitiga.



Gambar 2.11 Kurva segitiga

Contoh fungsi keanggotaan untuk himpunan NORMAL pada variabel temperatur ruangan:

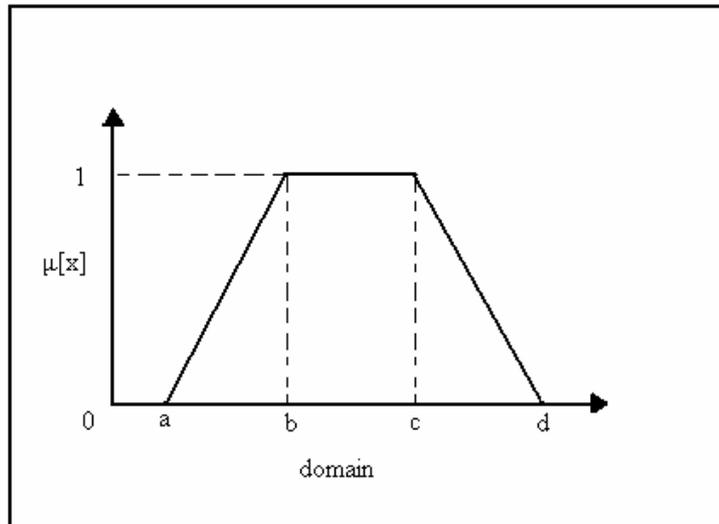
$$\begin{aligned} \mu_{\text{NORMAL}}[23] &= (23-15)/(25-15) \\ &= 8/10 = 0,8 \end{aligned}$$



Gambar 2.12 Kurva segitiga himpunan samar: NORMAL

c. Representasi kurva trapesium

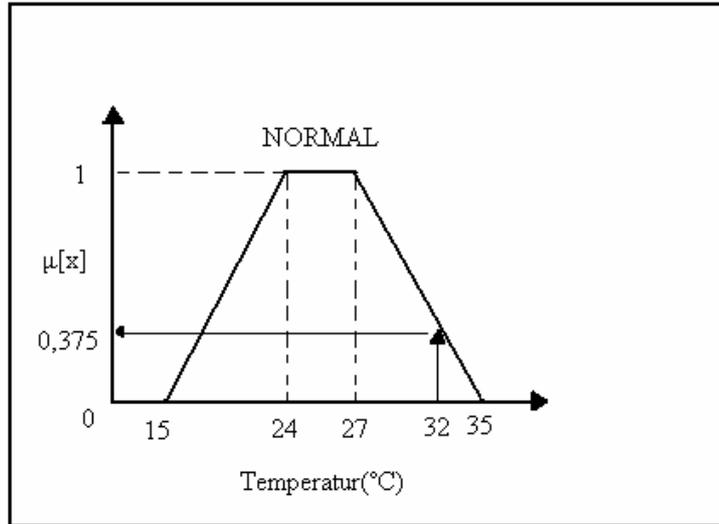
Kurva trapesium ini pada dasarnya seperti bentuk kurva segitiga, hanya saja kurva ini memiliki beberapa titik yang memiliki derajat keanggotaan bernilai satu.



Gambar 2.13 Kurva trapesium

Contoh pada fungsi keanggotaan untuk himpunan NORMAL pada variabel temperatur ruangan:

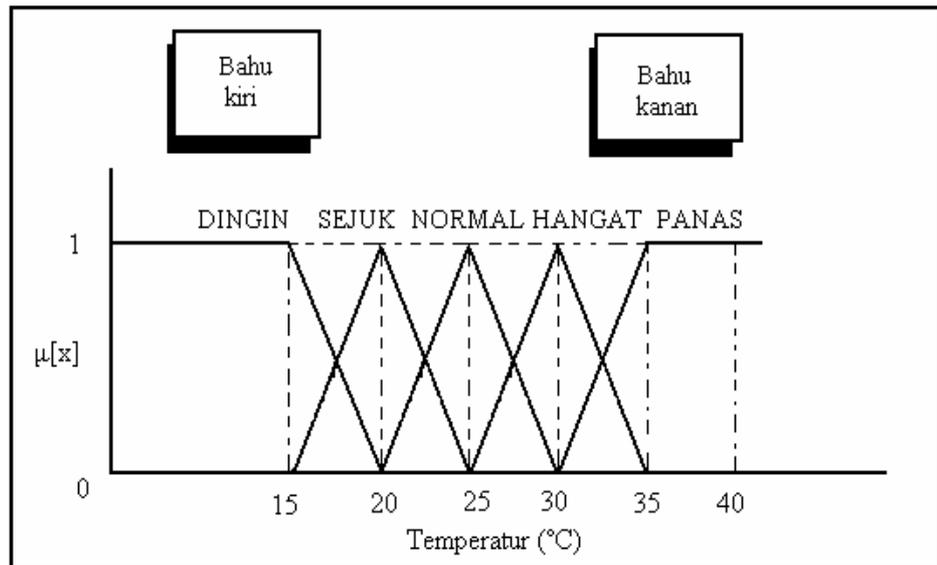
$$\begin{aligned}\mu_{\text{NORMAL}}[32] &= (35-32)/(35-27) \\ &= 3/8 = 0,375\end{aligned}$$



Gambar 2.14 Kurva trapesium himpunan samar: NORMAL

d. Representasi kurva bahu

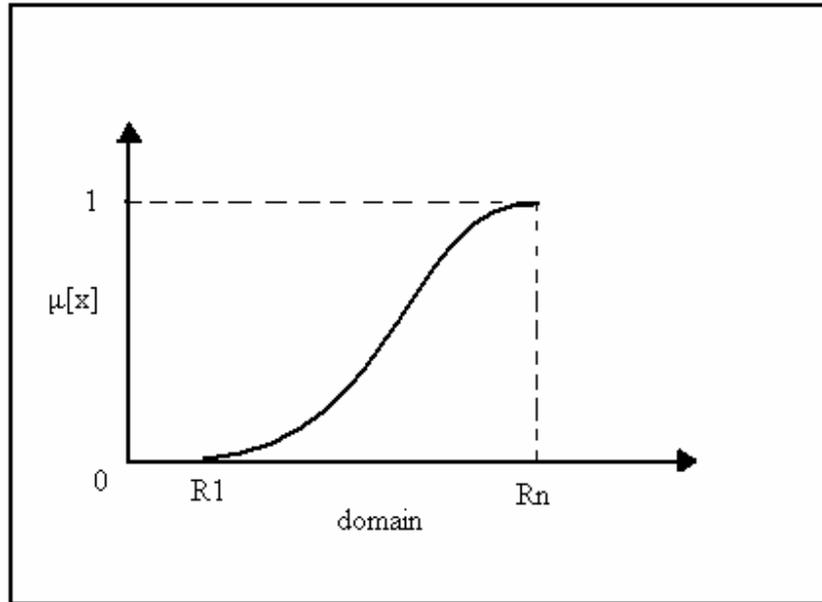
Daerah yang terletak ditengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun (misalkan: DINGIN bergerak ke SEJUK bergerak ke HANGAT bergerak ke PANAS). Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Sebagai contoh, apabila telah mencapai kondisi PANAS, kenaikan temperatur akan tetap berada pada kondisi PANAS. Himpunan samar bahu digunakan untuk mengakhiri variabel suatu daerah samar. Bahu kiri bergerak dari benar ke salah, demikian sebaliknya bahu kanan bergerak dari salah ke benar. Contoh daerah bahu dapat dilihat pada variabel temperatur yang direpresentasikan dengan kurva bahu.



Gambar 2.15 Daerah bahu pada variabel temperatur.

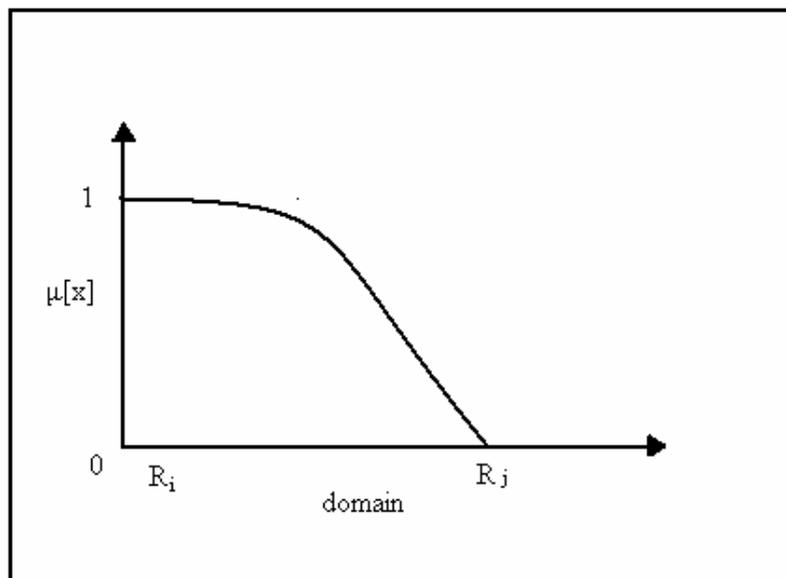
e. Representasi kurva-s

Kurva-s atau *sigmoid* merupakan kurva PERTUMBUHAN dan PENYUSUTAN yang berhubungan dengan kenaikan dan penurunan permukaan secara tak linear. Kurva-s untuk PERTUMBUHAN akan bergerak dari sisi paling kiri dengan nilai keanggotaan nol, menuju sisi paling kanan dengan nilai keanggotaan satu. Fungsi keanggotaannya akan tertumpu pada 50 % nilai keanggotaannya yang sering disebut dengan titik infleksi (*crossover*).



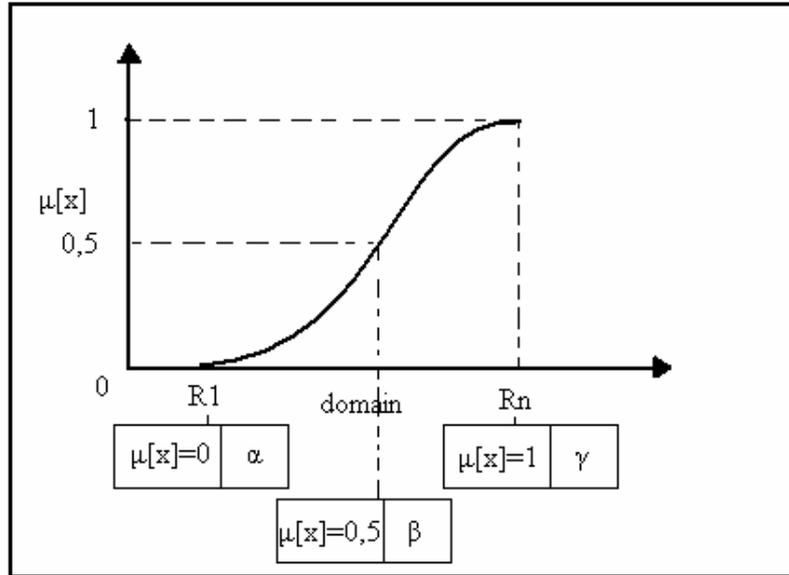
Gambar 2.16 Himpunan samar: PERTUMBUHAN

Kurva-s untuk PENYUSUTAN akan bergerak dari sisi kiri dengan nilai keanggotaan satu, menuju ke sisi kanan dengan nilai keanggotaan nol.



Gambar 2.17 Himpunan samar: PENYUSUTAN

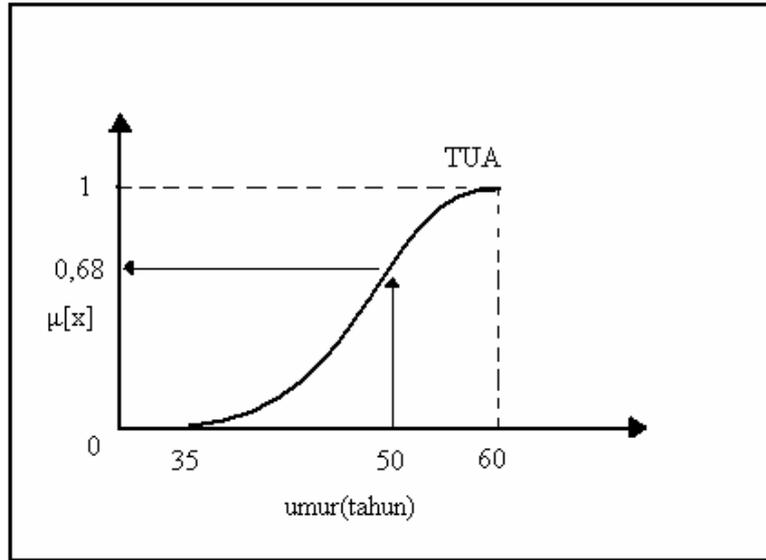
Kurva-s didefinisikan dengan menggunakan 3 parameter, yaitu: nilai keanggotaan nol (α), nilai keanggotaan lengkap (γ), dan titik infleksi (β) titik yang memiliki domain 50% benar.



Gambar 2.18 Karakteristik fungsi kurva-s

Contoh fungsi keanggotaan untuk himpunan TUA pada variabel umur:

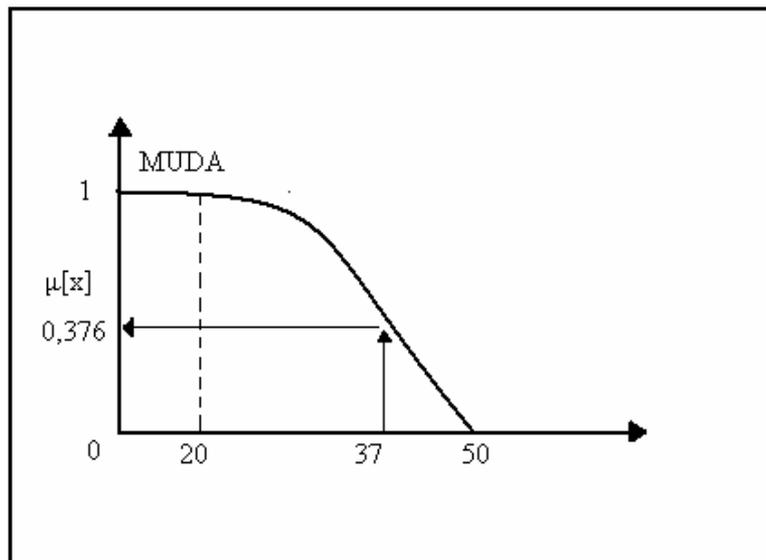
$$\begin{aligned} \mu_{TUA}[50] &= 1-2((60-50)/(60-35))^2 \\ &= 1-2(10/25)^2 \\ &= 0,68 \end{aligned}$$



Gambar 2.19 Himpunan samar: TUA

Contoh fungsi keanggotaan untuk himpunan MUDA pada variabel:

$$\begin{aligned} \mu_{\text{MUDA}}[37] &= 2((50-37)/(50-20))^2 \\ &= 2(13/30)^2 \\ &= 0,376 \end{aligned}$$



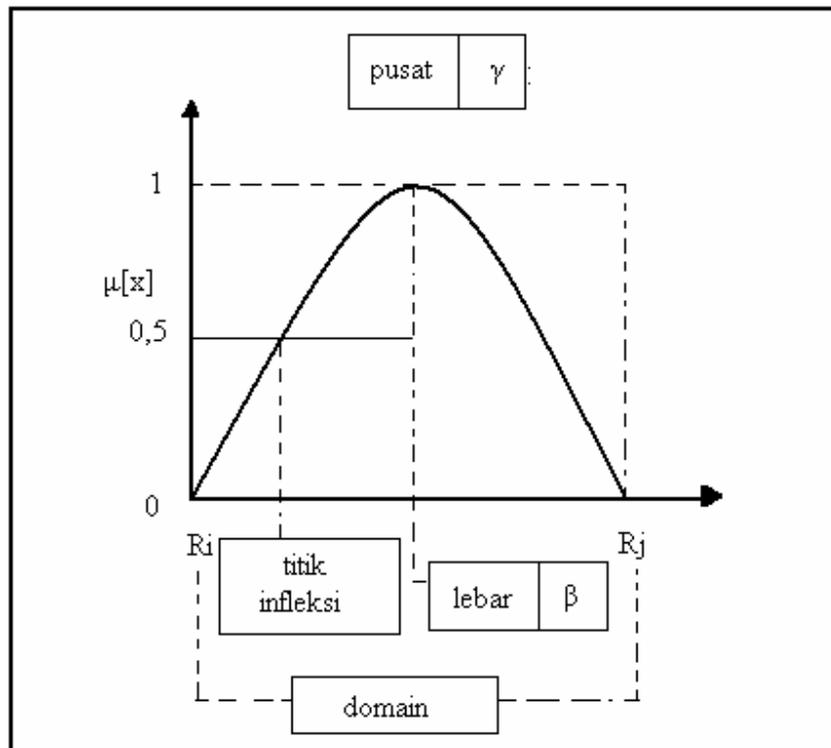
Gambar 2.20 Himpunan samar: MUDA

f. Representasi kurva lonceng

Bell curve atau kurva lonceng biasa digunakan untuk mempresentasikan bilangan samar. Kurva berbentuk lonceng ini terbagi atas tiga kelas, yaitu: himpunan samar Pi, Beta, dan Gauss. Perbedaan ketiga kurva ini terletak pada gradiennya.

(i) Kurva Pi

Kurva Pi berbentuk lonceng dengan derajat keanggotaan satu terletak pada pusat dengan domain (γ), dan lebar kurva (β).



Gambar 2.21 Karakteristik fungsional kurva Pi

Contoh fungsi keanggotaan untuk himpunan PAROBAYA pada variabel umur seperti terlihat pada gambar 2.22.

$$\mu_{\text{PAROBAYA}}[42] = 1 - 2 \left(\frac{45 - 42}{45 - 35} \right)^2$$

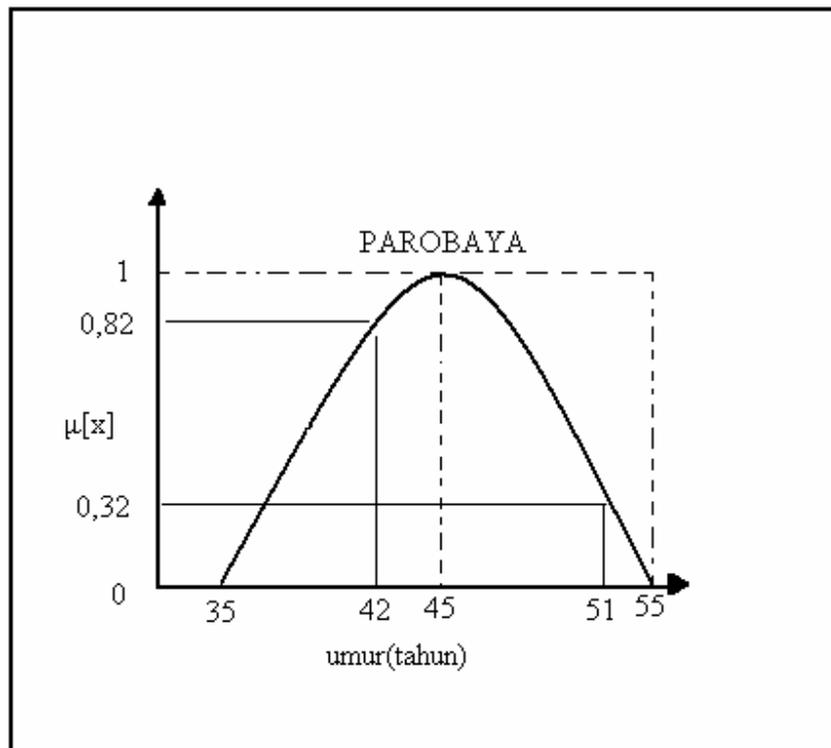
$$= 1 - 2\left(\frac{3}{10}\right)^2$$

$$= 0,82$$

$$\mu_{\text{PAROBAYA}}[51] = 2\left(\frac{55-51}{55-45}\right)^2$$

$$= 2\left(\frac{4}{10}\right)^2$$

$$= 0,32$$

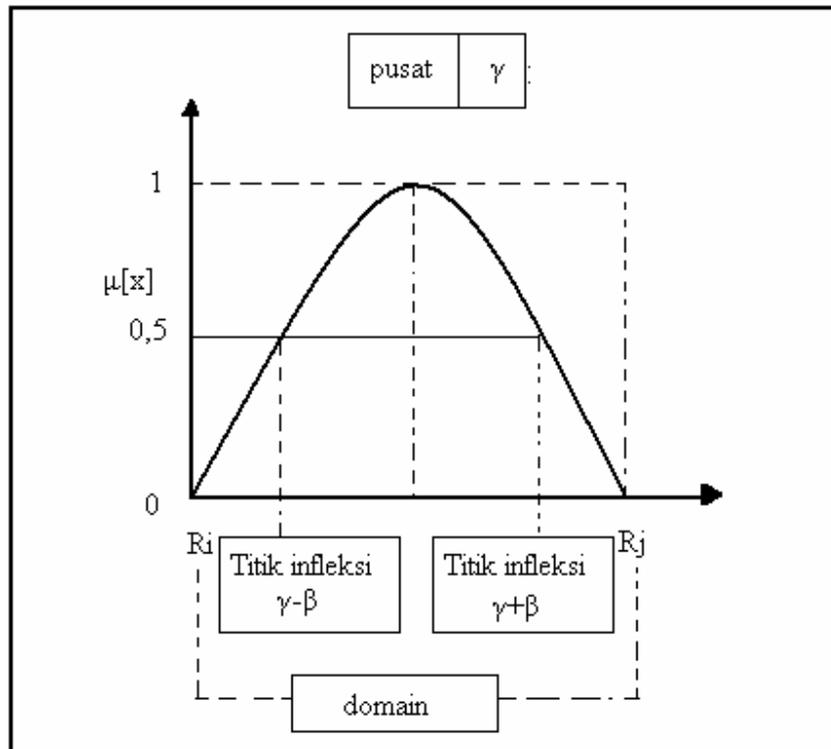


Gambar 2.22 Himpunan samar kurva Pi: PAROBAYA

(ii) Kurva Beta

Seperti halnya kurva Pi, kurva Beta juga berbentuk lonceng namun lebih rapat. Salah satu perbedaan mencolok kurva Beta dari kurva Pi adalah fungsi keanggotaannya akan mendekati nol hanya jika nilai (β) sangat besar. Kurva ini juga di definisikan dengan 2 parameter, yaitu nilai pada domain yang menunjukkan pusat kurva (γ), dan setengah

lebar kurva (β). Nilai kurva untuk nilai domain x diberikan sebagai berikut:

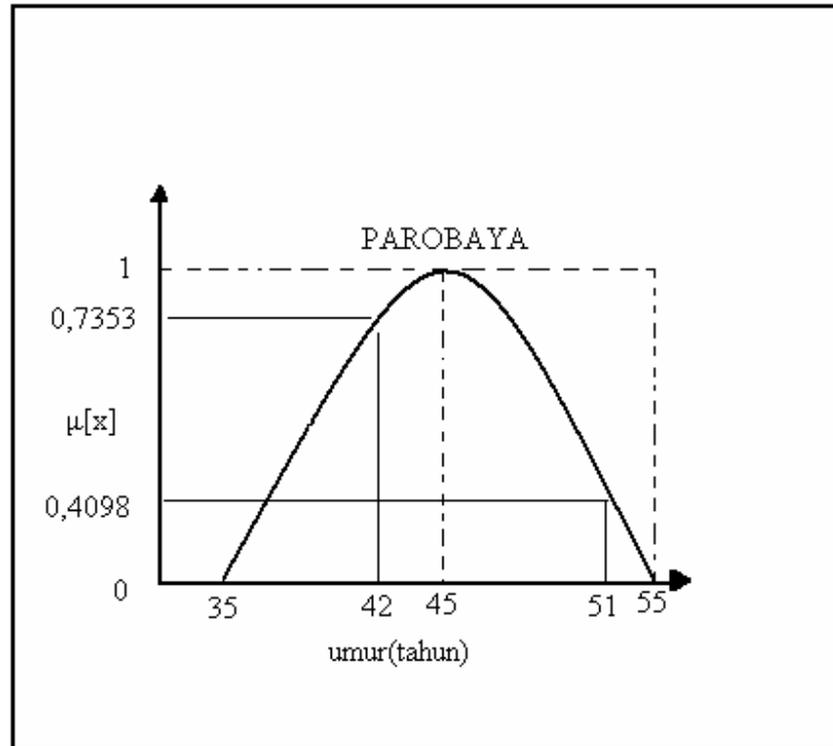


Gambar 2.23 Karakteristik fungsional kurva Beta

Contoh fungsi keanggotaan untuk himpunan PAROBAYA pada variabel umur:

$$\begin{aligned} \mu_{\text{PAROBAYA}}[42] &= 1/(1+((42-45)/5)^2) \\ &= 0,7353 \end{aligned}$$

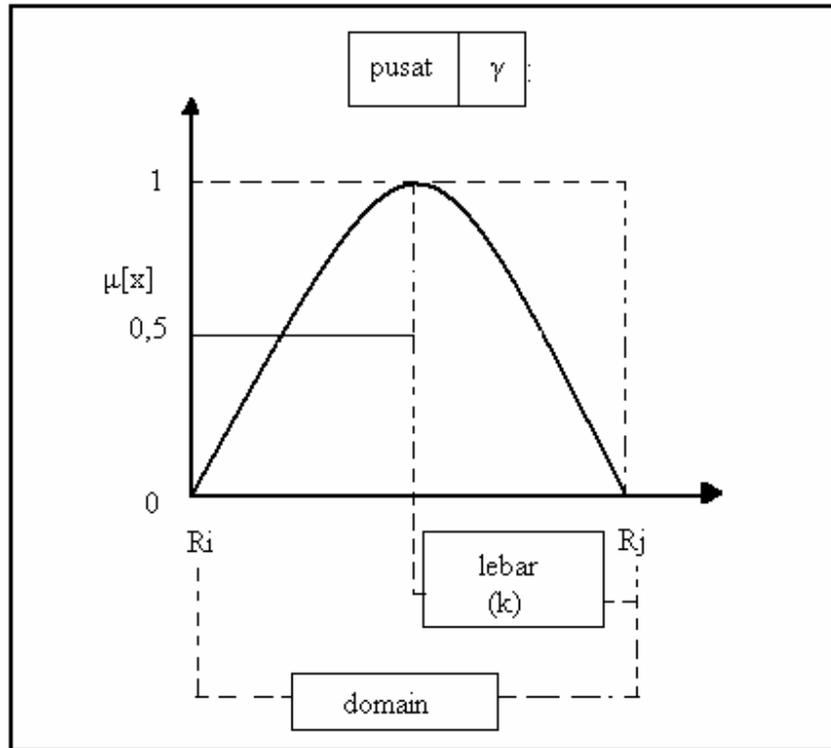
$$\begin{aligned} \mu_{\text{PAROBAYA}}[51] &= 1/(1+((51-45)/5)^2) \\ &= 0,4098 \end{aligned}$$



Gambar 2.24 Himpunan samar kurva Beta: PAROBAYA

(iii) Kurva Gauss

Jika kurva Pi dan Beta menggunakan 2 parameter yaitu (γ) dan (β), kurva Gauss juga menggunakan (γ) untuk menunjukkan nilai domain pada pusat kurva, dan (k) yang menunjukkan lebar kurva.



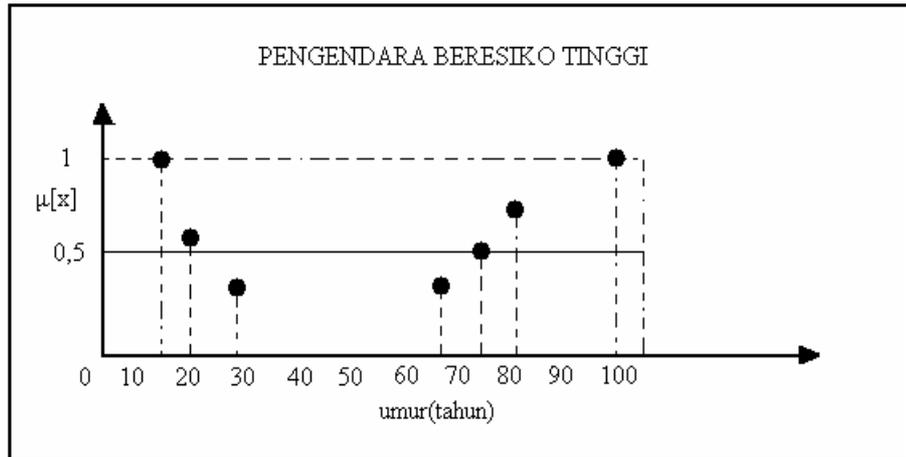
Gambar 2.25 Karakteristik fungsional kurva Gauss

g. Koordinat keanggotaan

Himpunan samar berisi urutan pasangan berurutan yang berisi nilai domain dan kebenaran nilai keanggotaannya dalam bentuk:

$$\text{Skalar}(i) / \text{Derajat}(i)$$

Skalar adalah suatu nilai yang digambar dari domain himpunan samar, sedangkan derajat skalar merupakan derajat keanggotaan himpunan samarnya.

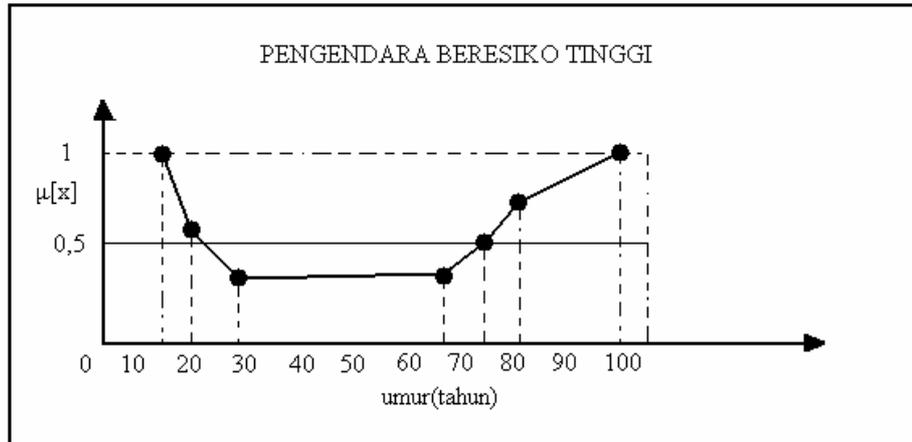


Gambar 2.26 Titik-titik koordinat keanggotaan PENGENDARA BERESIKO TINGGI

Pada himpunan samar PENGENDARA BERESIKO TINGGI di atas menunjukkan contoh penerapan pada sistem asuransi yang akan menanggung resiko seorang pengendara kendaraan bermotor berdasarkan usianya, akan terlihat seperti membentuk huruf 'U'. Koordinatnya dapat dituliskan dengan tujuh pasangan berurutan sebagai berikut:

16/1 21/0,6 28/0,3 68/0,3 76/0,5 80/0,7 96/1

Koordinat tersebut menspesifikasikan titik-titik sepanjang domain himpunan samar. Semua titik harus ada di domain dan paling sedikit harus ada satu titik yang memiliki nilai kebenaran sama dengan satu. Apabila titik-titik tersebut telah digambarkan, maka digunakan interpolasi linear untuk mendapatkan permukaan samarnya. (Gambar 2.27)



Gambar 2.27 Kurva pada himpunan PENGENDARA BERESIKO TINGGI

2.5 Operator dasar Zadeh

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan samar. Nilai keanggotaan sebagai hasil dari operasi dua himpunan sering dikenal dengan nama *fire strength* atau α -predikat. Ada tiga 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.

$$M_A \cap_B = \min(\mu_A[x], \mu_B[y])$$

Sebagai contohnya, misalkan nilai keanggotaan 27 tahun pada himpunan MUDA adalah 0,6 ($\mu_{MUDA}[27]=0,6$); dan nilai keanggotaan Rp. 2.000.000,-

pada himpunan penghasilan TINGGI adalah 0,8 ($\mu_{\text{GAJITINGGI}}[2 \times 10^6] = 0,8$);
 maka α -predikat untuk usia MUDA dan berpenghasilan TINGGI adalah:

$$\begin{aligned}\mu_{\text{MUDA}} \cap \mu_{\text{GAJITINGGI}} &= \min(\mu_{\text{MUDA}}[27], \mu_{\text{GAJITINGGI}}[2 \times 10^6]) \\ &= \min(0,6; 0,8) \\ &= 0,6\end{aligned}$$

b) Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. A-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari satu (1).

$$M_A \cup B = \max(\mu_A[x], \mu_B[y])$$

Sebagai contoh untuk menghitung nilai α -predikat untuk usia MUDA atau berpenghasilan TINGGI adalah:

$$\begin{aligned}\mu_{\text{MUDA}} \cup \mu_{\text{GAJITINGGI}} &= \max(\mu_{\text{MUDA}}[27], \mu_{\text{GAJITINGGI}}[2 \times 10^6]) \\ &= \max(0,6; 0,8) \\ &= 0,8\end{aligned}$$

c) Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. A-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari satu (1).

$$M_A' = 1 - \mu_A[x]$$

Sebagai contohnya untuk menghitung nilai α -predikat untuk usia TIDAK MUDA adalah:

$$\mu_{\text{MUDA}}'[27] = 1 - \mu_{\text{MUDA}}[27]$$

$$= 1 - 0,6$$

$$= 0,4$$

2.6 Penalaran Monoton

Metode penalaran secara monoton digunakan sebagai dasar untuk teknik implikasi samar. Meskipun penalaran ini sudah jarang digunakan, namun terkadang masih digunakan untuk penskalaan samar. Jika dua daerah samar direlasikan dengan implikasi sederhana sebagai berikut:

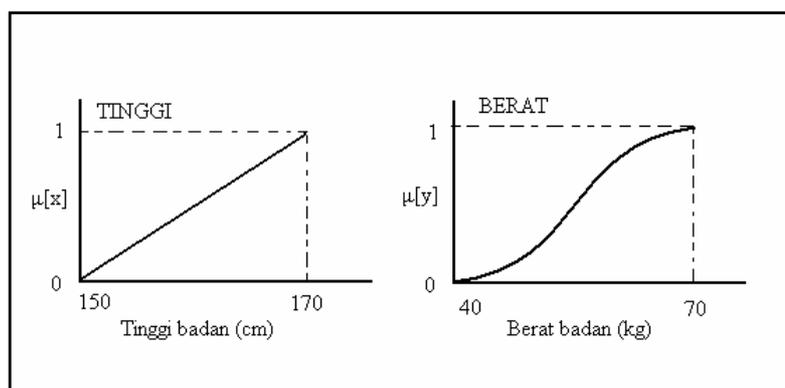
IF x is A THEN y is B

transfer fungsi:

$$y = f((x, A), B)$$

Maka sistem samar dapat berjalan tanpa harus melalui komposisi dan dekomposisi samar. Nilai output dapat diestimasi secara langsung dari nilai keanggotaan yang berhubungan dengan antesedennya.

Untuk lebih jelasnya kita ambil contoh dari dua himpunan samar: TINGGI dan BERAT, yang masing-masing menunjukkan tinggi badan dan berat badan orang Indonesia. Keduanya diperlihatkan pada kurva berikut:



Gambar 2.28 Himpunan samar: TINGGI dan BERAT

Relasi antara kedua himpunan tersebut diekspresikan dengan aturan tunggal sebagai berikut:

IF TinggiBadan is TINGGI THEN BeratBadan is BERAT

Implikasi secara monoton akan menyeleksi daerah samar A dan B dengan algoritma sebagai berikut:

- Untuk suatu elemen x pada domain A, tentukan nilai keanggotaannya dalam daerah samar A, yaitu: $\mu_A[x]$;
- Pada daerah samar B, nilai keanggotaan yang berhubungan dengan penentuan permukaan samarnya. Tarik garis lurus ke arah domain. Nilai pada sumbu domain y , merupakan solusi dari fungsi implikasi tersebut.

Fungsi ini dapat dituliskan dengan: $y_B = f(\mu_A[x], D_B)$

Sebagai contoh untuk menunjukkan kerja algoritma tersebut; misalkan seseorang memiliki tinggi badan 165 cm, memiliki derajat keanggotaan 0,75 pada daerah samar TINGGI, yang diperoleh dari:

$$\begin{aligned}\mu_{\text{TINGGI}}[165] &= (165-150)/(170-150) \\ &= 15/20 \\ &= 0,75\end{aligned}$$

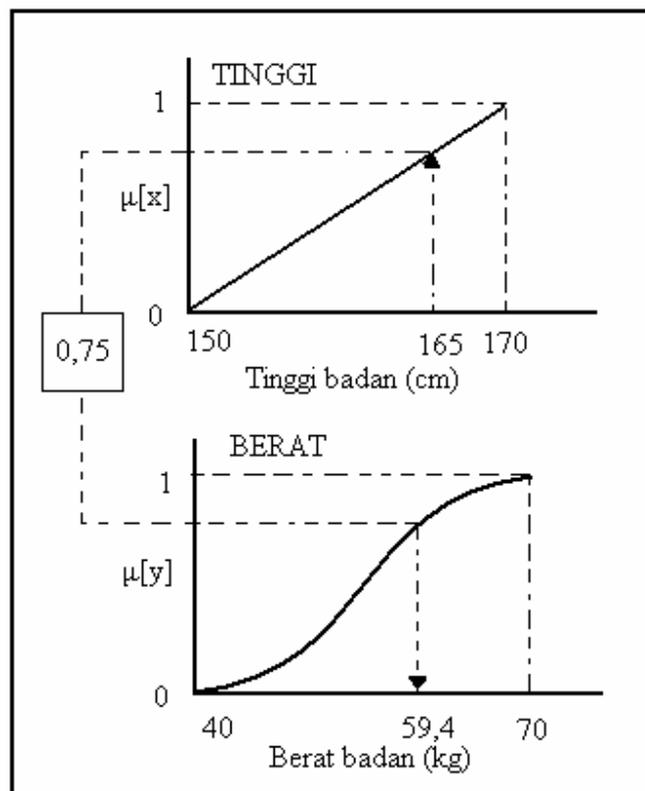
Nilai ini dipetakan ke daerah samar BERAT yang akan memberikan solusi berat badan orang tersebut yaitu 59,4 kg, yang diperoleh dari:

$$\begin{aligned}\mu_{\text{BERAT}}[y] &= S(y; 40, 55, 70) \\ &= 0,75\end{aligned}$$

Karena $0,75 > 0,5$ maka letak y adalah antara 52,5 sampai 70, sehingga:

$$\begin{aligned}1-2[(70-y)/(70-40)]^2 &= 0,75 \\ \rightarrow 1-2(70-y)^2/900 &= 0,75\end{aligned}$$

$$\begin{aligned} \rightarrow 2(70-y)^2/900 &= 0,25 \\ \rightarrow (70-y)^2 &= 112,5 \\ \rightarrow (70-y) &= \pm\sqrt{(112,5)} \\ \rightarrow y &= 70 \pm 10,6 \\ \rightarrow y &= 59,4 \end{aligned}$$



Gambar 2.29 Implikasi monoton: TINGGI ke BERAT

2.7 Fungsi implikasi

Tiap-tiap aturan atau proposisi pada basis pengetahuan samar akan berhubungan dengan suatu relasi samar. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah:

IF x is A THEN y is B

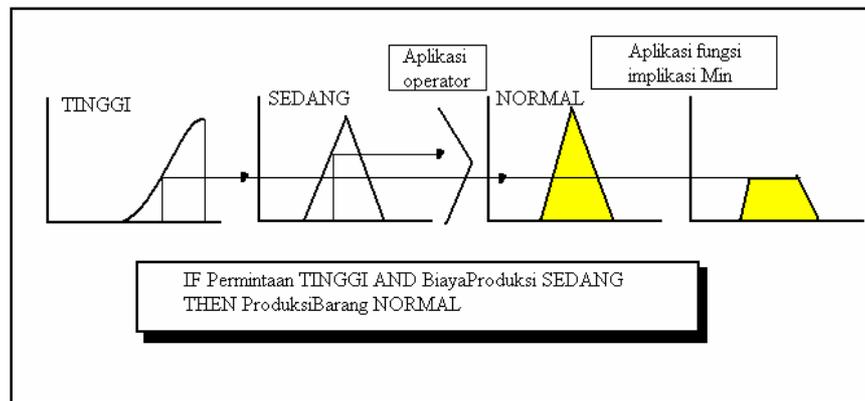
Dengan x dan y adalah skalar, A dan B adalah himpunan samar. Proposisi yang mengikuti IF disebut sebagai anteseden, sedangkan proposisi yang mengikuti THEN disebut sebagai konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator samar, seperti:

IF $(x_1 \text{ is } A_1) \bullet (x_2 \text{ is } A_2) \bullet (x_3 \text{ is } A_3) \bullet \dots \bullet (x_N \text{ is } A_N)$ THEN $y \text{ is } B$

Dengan \bullet adalah operator baik itu AND atau OR.

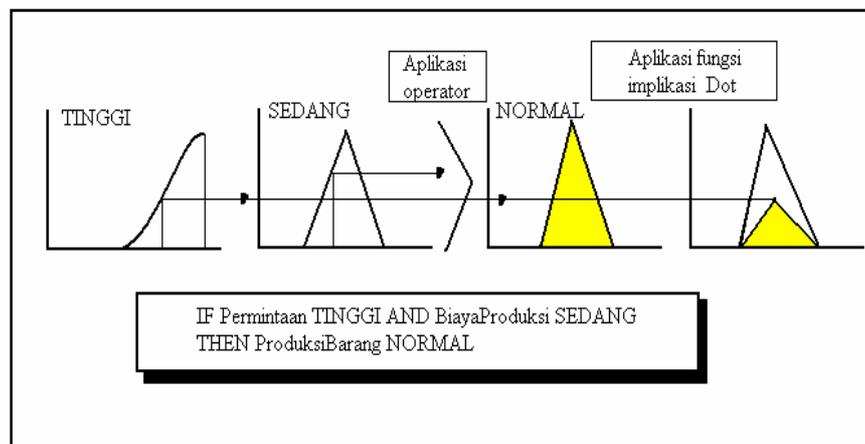
Secara umum ada dua fungsi implikasi yang dapat digunakan, yaitu:

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



Gambar 2.30 Fungsi implikasi: MIN

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

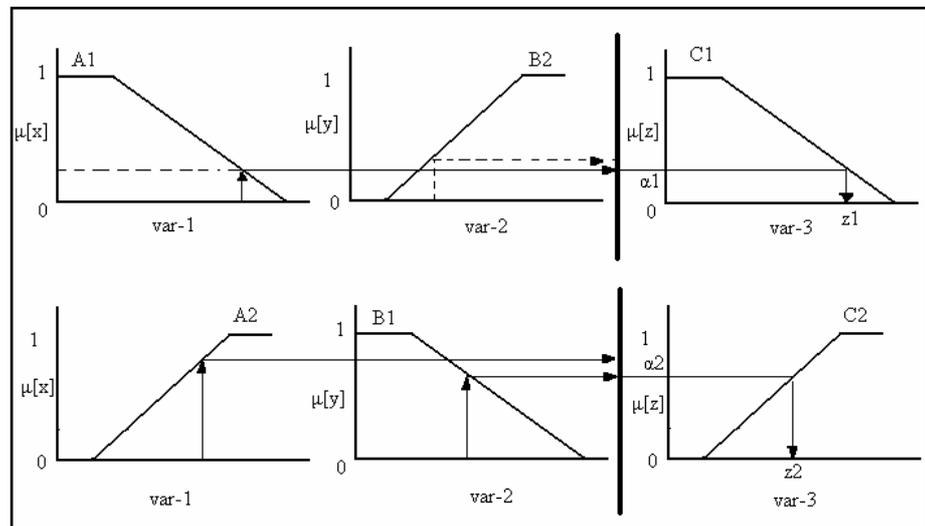


Gambar 2.31 fungsi implikasi: DOT

2.8 Sistem inferensi logika samar

2.8.1 Metode Tsukamoto

Pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-THEN harus direpresentasikan dengan suatu himpunan samar dengan fungsi keanggotaan yang monoton. 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.



Gambar 2.32 inferensi dengan menggunakan metode Tsukamoto

Contohnya, sebuah perusahaan makanan kaleng akan memproduksi makanan jenis XYZ. Dari data 1 bulan terakhir, permintaan terbesar hingga mencapai 500 kemasan/hari. Persediaan barang digudang terbanyak sampai 600 kemasan/hari, dan terkecil sampai 100 kemasan/hari. Dengan segala keterbatasannya, sampai saat ini, perusahaan baru mampu memproduksi barang maksimum 7000 kemasan/hari, serta demi efisiensi mesin dan SDM tiap hari diharapkan perusahaan memproduksi

paling tidak 2000 kemasan. Apabila proses produksi perusahaan tersebut menggunakan 4 aturan logika samar sebagai berikut:

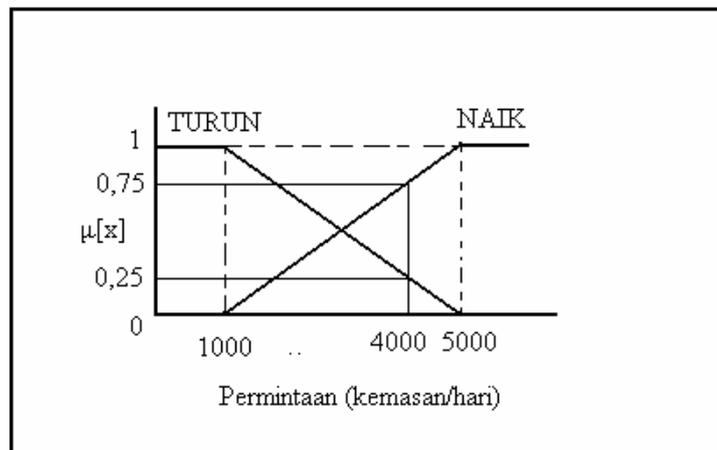
- [R1] IF Permintaan TURUN AND Persediaan BANYAK
THEN Produksi Barang BERKURANG;
- [R2] IF Permintaan TURUN AND Persediaan SEDIKIT
THEN Produksi Barang BERKURANG;
- [R3] IF Permintaan NAIK AND Persediaan BANYAK
THEN Produksi Barang BERTAMBAH;
- [R4] IF Permintaan NAIK AND Persediaan SEDIKIT
THEN Produksi Barang BERTAMBAH;

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

Solusi:

Ada tiga variabel yang harus dimodelkan, yaitu:

- Permintaan; terdiri atas dua himpunan samar, yaitu: NAIK dan TURUN.



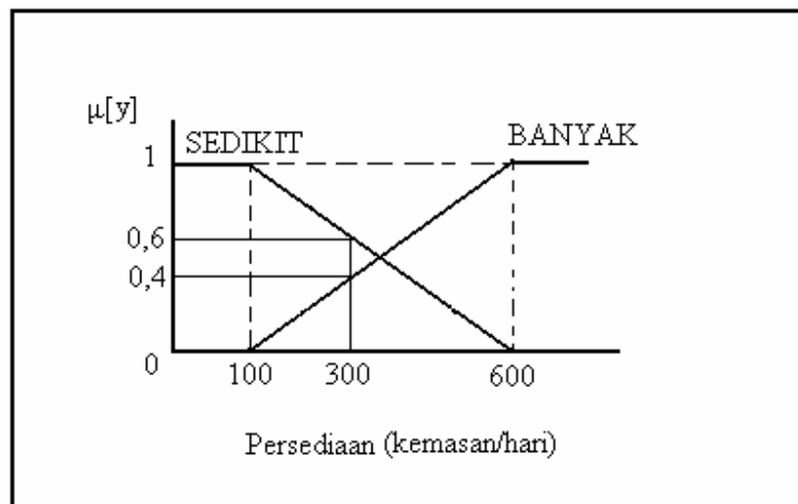
Gambar 2.33 Fungsi keanggotaan PERMINTAAN

Nilai keanggotaan dari himpunan ini:

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

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

- Persediaan; terdiri atas dua himpunan samar, yaitu: **SEDIKIT** dan **BANYAK**.



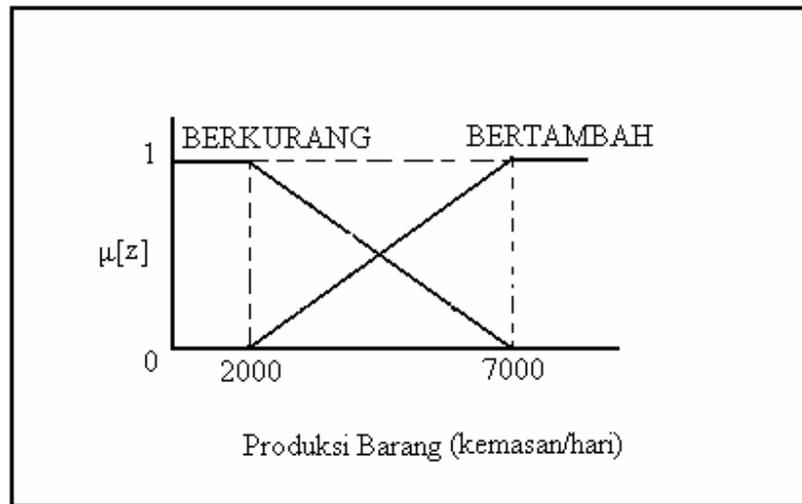
Gambar 2.34 Fungsi keanggotaan PERSEDIAAN

Dengan nilai keanggotaan:

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

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

- Produksi barang; terdiri atas dua himpunan samar, yaitu: **BERKURANG** dan **BERTAMBAH**.



Gambar 2.35 Fungsi keanggotaan PRODUKSI BARANG

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

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

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

Lihat pada himpunan Produksi Barang BERKURANG,

$$(7000-z)/5000 = 0,25 \rightarrow z_1 = 5750$$

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

$$\begin{aligned}
 \alpha\text{-predikat} &= \mu_{\text{TURUN}} \cap \mu_{\text{SEDIKIT}} \\
 &= \min(\mu_{\text{TURUN}}[4000], \mu_{\text{SEDIKIT}}[300])
 \end{aligned}$$

$$= \min(0,25;0,6)$$

$$= 0,25$$

Lihat pada himpunan Produksi Barang BERKURANG,

$$(7000-z)/5000 = 0,25 \rightarrow z_2 = 5750$$

[R3] IF Permintaan NAIK AND Persediaan BANYAK

THEN Produksi Barang BERTAMBAH;

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

Lihat pada himpunan Produksi Barang BERTAMBAH,

$$(z-2000)/5000 = 0,4 \rightarrow z_3 = 4000$$

[R4] IF Permintaan NAIK AND Persediaan SEDIKIT

THEN Produksi Barang BERTAMBAH;

$$\begin{aligned}\alpha\text{-predikat} &= \mu_{\text{NAIK}} \cap \mu_{\text{SEDIKIT}} \\ &= \min(\mu_{\text{NAIK}}[4000], \mu_{\text{SEDIKIT}}[300]) \\ &= \min(0,75;0,6) \\ &= 0,6\end{aligned}$$

Lihat pada himpunan Produksi Barang BERTAMBAH,

$$(z-2000)/5000 = 0,6 \rightarrow z_2 = 5000$$

dari sini kita dapat menentukan berapakah nilai z, yaitu dengan cara sebagai

berikut:

$$z = \frac{\alpha\text{pred}_1 * z_1 + \alpha\text{pred}_2 * z_2 + \alpha\text{pred}_3 * z_3 + \alpha\text{pred}_4 * z_4}{\alpha\text{pred}_1 + \alpha\text{pred}_2 + \alpha\text{pred}_3 + \alpha\text{pred}_4}$$

$$\alpha\text{pred}_1 + \alpha\text{pred}_2 + \alpha\text{pred}_3 + \alpha\text{pred}_4$$

$$z = \frac{0,25 * 5750 + 0,25 * 5750 + 0,4 * 4000 + 0,6 * 5000}{0,25 + 0,25 + 0,4 + 0,6}$$

$$0,25 + 0,25 + 0,4 + 0,6$$

$$z = 7575 / 1,5$$

$$z = 4983$$

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

2.8.2 Metode Mamdani

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

1. Pembentukan himpunan samar

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

2. Aplikasi fungsi implikasi (aturan)

Fungsi yang digunakan pada metode Mamdani adalah Min.

3. Komposisi aturan

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

- a. Metode Max (maksimum)

Pada metode ini, solusi himpunan samar diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah samar, dan menggunakannya untuk memodifikasi

daerah samar, dan mengaplikasikannya ke output dengan menggunakan operator OR (union). Jika semua proposisi telah dievaluasi, maka output akan berisi suatu himpunan samar yang merefleksikan kontribusi 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 samar sampai aturan ke-i;

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

Misalkan ada 3 aturan sebagai berikut:

[R1] IF Biaya Produksi RENDAH AND Permintaan NAIK

THEN Produksi Barang BERTAMBAH;

[R2] IF Biaya Produksi STANDAR

THEN Produksi Barang NORMAL;

[R3] IF Biaya Produksi TINGGI AND Permintaan TURUN

THEN Produksi Barang BERKURANG;

Proses inferensi dengan menggunakan metode Max dalam melakukan komposisi aturan akan membuat range output yang terkoordinasi. Dan apabila digunakan implikasi MIN, maka metode komposisi sering disebut dengan nama MAX-MIN atau MIN-MAX atau Mamdani.

b. Metode Sum (*additive*/penambahan)

Pada metode ini, solusi himpunan samar diperoleh dengan cara melakukan *bounded-sum* terhadap semua output daerah samar. 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 samar sampai aturan ke-i;

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

c. Metode Probabilistik OR (probor)

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

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

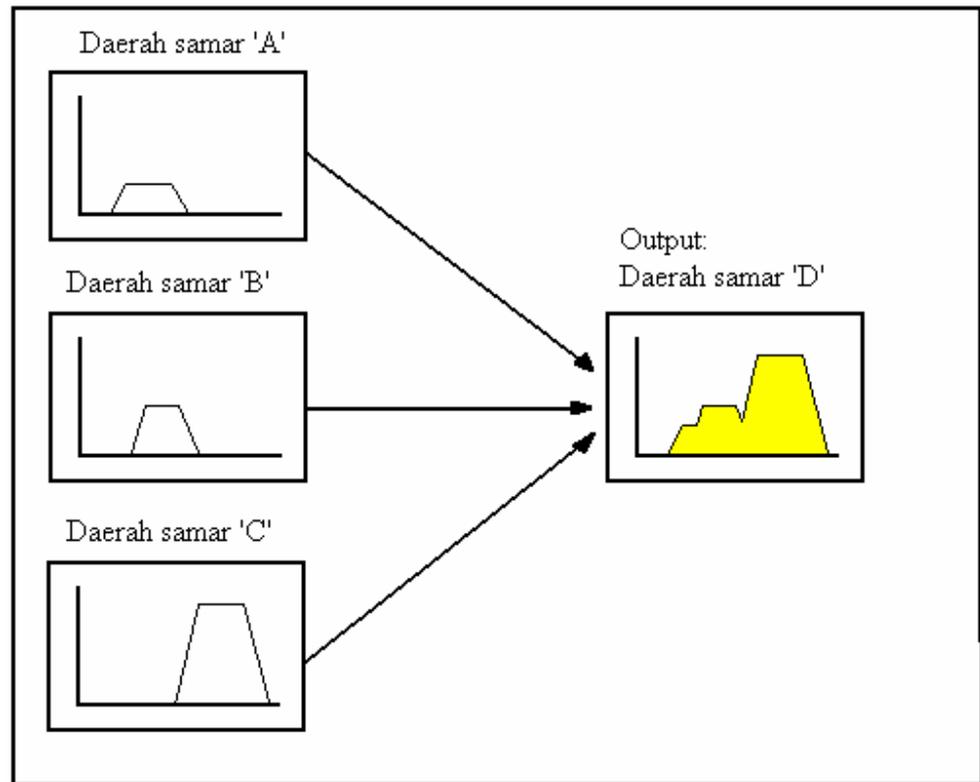
dengan:

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

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

4. Penegasan (*defuzzy*)

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



Gambar 2.36 Proses defuzzifikasi

Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, antara lain:

- a. Metode Centroid (*Composite moment*)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat pada daerah samar.

- b. Metode Bisektor

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

- c. Metode *Mean of Maximum* (MOM)

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

d. Metode *Largest of Maximum* (LOM)

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

e. Metode *Smallest of Maximum* (SOM)

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

2.8.3 Metode Sugeno

Metode ini diperkenalkan oleh Takagi Sugeno Kang pada tahun 1985. Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja output sistem tidak berupa himpunan samar, melainkan berupa konstanta atau persamaan linear.

a. Model logika samar Sugeno orde-nol

Secara umum bentuk model samar Sugeno orde-nol adalah:

$$\text{IF } (x_1 \text{ is } A_1) \bullet (x_2 \text{ is } A_2) \bullet \dots \bullet (x_N \text{ is } A_N) \text{ THEN } z = k$$

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

b. Model logika samar Sugeno orde-satu

Secara umum bentuk model samar Sugeno orde-satu adalah:

$$\text{IF } (x_1 \text{ is } A_1) \bullet \dots \bullet (x_N \text{ is } A_N) \text{ THEN } z = p_1 * x_1 + \dots + p_N * x_N + q$$

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

Dengan menggunakan metode Sugeno sebagai komposisi aturan, maka defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya.

2.9 Basisdata sistem logika samar

Sebagian besar basis data standar diklasifikasikan berdasarkan bagaimana data tersebut dipandang oleh user. Misalkan kita memiliki data karyawan yang tersimpan pada tabel DT_KARYAWAN dengan field NIP, nama, tanggal lahir, tahun masuk, dan gaji per bulan.

Tabel 2.2 Data karyawan

NIP	Nama	Tgl. Lahir	Thn. Masuk	Gaji/bln
01	Lia	03-06-1977	2001	750.000
02	Iwan	23-09-1959	1990	1.500.000
03	Sari	12-12-1972	1993	1.255.000
04	Andi	06-03-1970	2003	1.040.000
05	Budi	04-12-1965	1995	950.000
06	Amir	18-11-1968	1994	1.600.000
07	Rian	28-05-1970	2002	1.250.000
08	Kiki	09-07-1976	2006	550.000
09	Alda	14-08-1972	2004	735.000
10	Yoga	17-09-1982	2005	860.000

Kemudian dari tabel DT_KARYAWAN, diolah menjadi suatu tabel temporer untuk menghitung umur karyawan dan masa kerjanya. Tabel tersebut diberi nama dengan tabel KARYAWAN.

Tabel 2.3 Data karyawan setelah diolah

NIP	Nama	Umur (thn)	Masa Kerja (thn)*	Gaji/bln
01	Lia	30	6	750.000
02	Iwan	48	17	1.500.000
03	Sari	36	14	1.255.000
04	Andi	37	4	1.040.000
05	Budi	42	12	950.000
06	Amir	39	13	1.600.000
07	Rian	37	5	1.250.000
08	Kiki	32	1	550.000
09	Alda	35	3	735.000
10	Yoga	25	2	860.000

* Dengan dasar tahun 2007

Dengan menggunakan basis data standar, kita dapat mencari data-data karyawan dengan spesifikasi tertentu dengan menggunakan query. Misalkan kita ingin mendapatkan informasi tentang nama-nama karyawan yang usianya kurang dari 35 tahun, maka kita bisa ciptakan suatu query, sebagai berikut:

```
SELECT NAMA  
  
FROM KARYAWAN  
  
WHERE (Umur < 35)
```

Sehingga muncul nama-nama Lia, Kiki, dan Yoga. Apabila kita ingin mendapatkan informasi tentang nama-nama karyawan yang gajinya lebih dari satu juta rupiah, maka kita bisa ciptakan suatu query sebagai berikut:

```
SELECT NAMA  
FROM KARYAWAN  
WHERE (Gaji > 1000000)
```

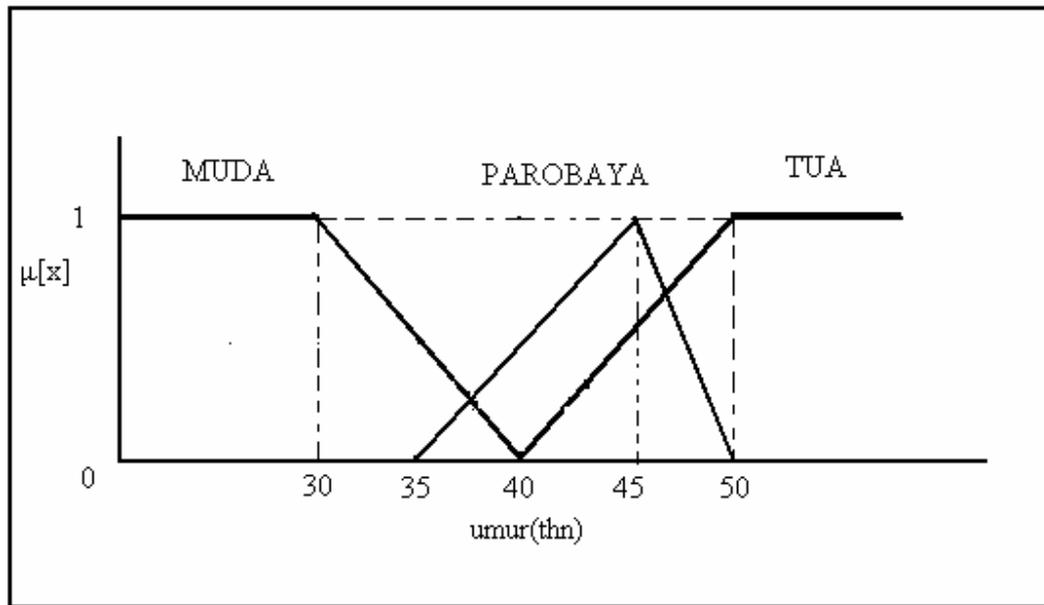
Sehingga muncul nama-nama Iwan, Sari, Andi, Amir dan Rian. Apabila kita ingin mendapatkan informasi tentang nama-nama karyawan yang masa kerjanya kurang dari atau sama dengan lima tahun tetapi gajinya sudah lebih dari satu juta rupiah, maka query-nya adalah:

```
SELECT NAMA  
FROM KARYAWAN  
WHERE (MasaKerja <= 5) and (Gaji > 1000000)
```

Sehingga nama yang akan muncul adalah Andi dan Rian.

Ada kenyataannya, seseorang kadang membutuhkan informasi dari data-data yang bersifat ambigu. Apabila hal ini terjadi, maka gunakan basis data dengan logika samar. Selama ini sudah ada beberapa penelitian tentang basis data logika samar, salah satunya adalah model Tahani. Basis data logika samar model Tahani masih tetap menggunakan relasi standar, hanya saja model Tahani menggunakan teori himpunan samar untuk mendapatkan informasi pada query-nya.

Misalkan sebagai contoh, dibuat kategori usia karyawan menjadi tiga himpunan: MUDA, PAROBAYA, dan TUA.



Gambar 2.37 Fungsi keanggotaan untuk variabel usia

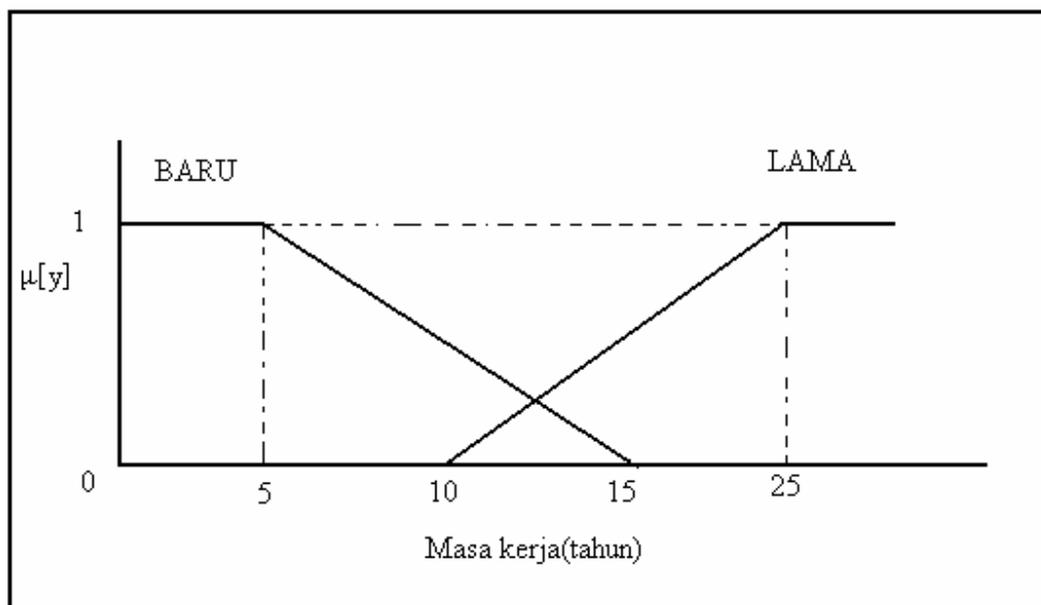
Kemudian didapatkan tabel karyawan berdasarkan umur dengan derajat keanggotaannya pada setiap himpunan.

Tabel 2.4 Tabel Karyawan berdasar umur

NIP	Nama	Umur	Derajat Keanggotaan ($\mu[x]$)		
			MUDA	PAROBAYA	TUA
01	Lia	30	1	0	0
02	Iwan	48	0	0,4	0,8
03	Sari	36	0,4	0,1	0
04	Andi	37	0,3	0,2	0
05	Budi	42	0	0,7	0,2
06	Amir	39	0,1	0,4	0
07	Rian	37	0,3	0,2	0

08	Kiki	32	0,8	0	0
09	Alda	35	0,5	0	0
10	Yoga	25	1	0	0

Kemudian variabel Masa Kerja dapat dikategorikan kepada himpunan samar: BARU dan LAMA.



Gambar 2.38 Fungsi keanggotaan untuk variabel Masa Kerja

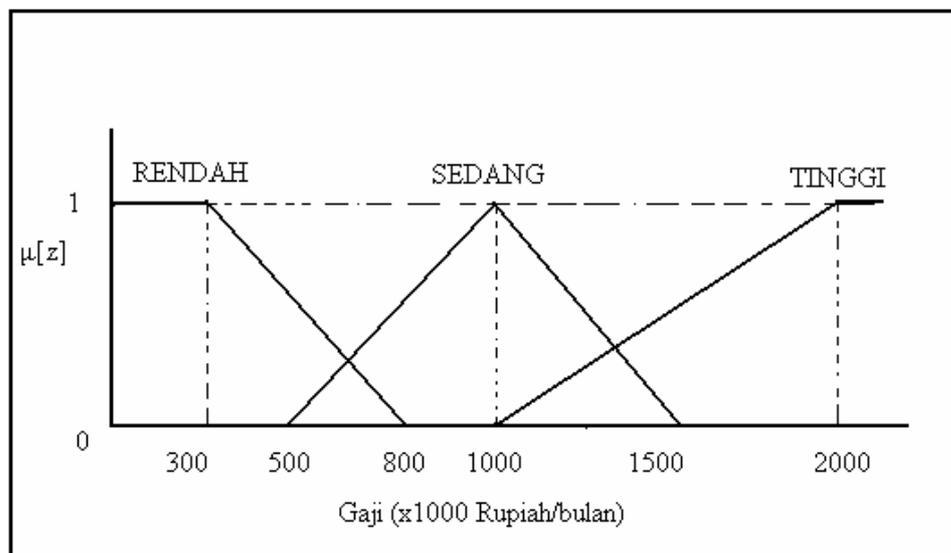
Kemudian didapatkan tabel karyawan berdasarkan masa kerja dengan derajat keanggotaannya pada setiap himpunan.

Tabel 2.5 Tabel Karyawan berdasarkan Masa Kerja

NIP	Nama	Masa Kerja	Derajat Keanggotaan ($\mu[y]$)	
			BARU	LAMA
01	Lia	6	0,9	0

02	Iwan	17	0	0,467
03	Sari	14	0,1	0,267
04	Andi	4	1	0
05	Budi	12	0,3	0,133
06	Amir	13	0,2	0,2
07	Rian	5	1	0
08	Kiki	1	1	0
09	Alda	3	1	0
10	Yoga	2	1	0

Lalu variabel untuk himpunan samar Gaji dapat dikategorikan sebagai:
RENDAH, SEDANG, dan TINGGI.



Gambar 2.39 Fungsi keanggotaan untuk variabel Gaji

Maka didapatkanlah tabel karyawan berdasarkan gaji dengan derajat keanggotaannya pada setiap himpunan dari fungsi tersebut di atas.

Tabel 2.6 Tabel karyawan berdasarkan Gaji.

NIP	Nama	Gaji/bln	Derajat Keanggotaan ($\mu[z]$)		
			RENDAH	SEDANG	TINGGI
01	Lia	750.000	0,1	0,5	0
02	Iwan	1.500.000	0	0,49	0,255
03	Sari	1.255.000	0	0	0,5
04	Andi	1.040.000	0	0,92	0,04
05	Budi	950.000	0	0,9	0
06	Amir	1.600.000	0	0	0,6
07	Rian	1.250.000	0	0,5	0,25
08	Kiki	550.000	0,5	0	0
09	Alda	735.000	0,13	0	0
10	Yoga	860.000	0	0	0

Dari ketiga database logika samar itu ada beberapa query yang bisa diberikan, misalnya:

- ❖ Siapa sajakah karyawan yang masih muda, tetapi memiliki gaji yang tinggi?

Maka query-nya adalah:

```
SELECT NAMA
```

```
FROM KARYAWAN
```

```
WHERE (Umur = "MUDA") and (Gaji = "TINGGI")
```

Maka hasil pencarian dari query ini akan terlihat seperti tabel 2.6 di bawah ini.

Tabel 2.7 Tabel hasil query MUDA dan TINGGI

NIP	NAMA	UMUR	GAJI	Derajat Keanggotaan		
				MUDA	TINGGI	MUDA & TINGGI
07	Rian	37	1.250.000	0,3	0,5	0,3
03	Sari	36	1.255.000	0,4	0,25	0,25
06	Amir	39	1.600.000	0,1	0,6	0,1
04	Andi	37	1.040.000	0,3	0,04	0,04
01	Lia	30	750.000	1	0	0
02	Iwan	48	1.500.000	0	0,255	0
05	Budi	42	950.000	0	0	0
08	Kiki	32	550.000	0,8	0	0
09	Alda	35	735.000	0,5	0	0
10	Yoga	25	860.000	1	0	0

Berarti terlihat ada empat karyawan yang memenuhi kategori masih MUDA dan berpenghasilan TINGGI, yaitu Rian, Sari, Amir, dan Andi. Dan Rian adalah karyawan yang memiliki nilai terbesar dalam kategori tersebut.

- ❖ Siapa sajakah karyawan yang masih muda atau karyawan yang memiliki gaji tinggi?

Maka query-nya adalah:

SELECT NAMA

FROM KARYAWAN

WHERE (Umur = "MUDA") or (Gaji = "TINGGI")

Maka hasil pencarian dari query ini akan terlihat seperti tabel 2.7 di bawah ini.

Tabel 2.8 Tabel hasil query MUDA atau TINGGI

NIP	NAMA	UMUR	GAJI	Derajat Keanggotaan		
				MUDA	TINGGI	MUDA or TINGGI
01	Lia	30	750.000	1	0	1
10	Yoga	25	860.000	1	0	1
08	Kiki	32	550.000	0,8	0	0,8
06	Amir	39	1.600.000	0,1	0,6	0,6
03	Sari	36	1.255.000	0,4	0,25	0,4
09	Alda	35	735.000	0,5	0	0,5
04	Andi	37	1.040.000	0,3	0,04	0,3
07	Rian	37	1.250.000	0,3	0,5	0,5
02	Iwan	48	1.500.000	0	0,255	0,255
05	Budi	42	950.000	0	0	0

Maka dari sepuluh karyawan yang ada hanya ada satu yang tidak memenuhi kategori tersebut, yaitu Budi yang derajat keanggotaannya adalah

nol (0). Sedangkan yang derajat keanggotaannya terbesar ada dua orang, yaitu Lia dan Yoga, masing-masing memiliki derajat keanggotaan satu (1).

- ❖ Siapa sajakah karyawan yang masih muda tetapi masa kerjanya sudah lama?

Maka query-nya adalah:

```
SELECT NAMA
FROM KARYAWAN
WHERE (Umur = "MUDA") and (MasaKerja = "LAMA")
```

Maka hasil pencarian dari query ini akan terlihat seperti tabel 2.8 di bawah ini.

Tabel 2.9 Tabel hasil query MUDA dan LAMA

NIP	NAMA	UMUR	MASA KERJA	Derajat Keanggotaan		
				MUDA	LAMA	MUDA & LAMA
03	Sari	36	14	0,4	0,267	0,267
06	Amir	39	13	0,1	0,2	0,1
01	Lia	30	6	1	0	0
02	Iwan	48	17	0	0,467	0
04	Andi	37	4	0,3	0	0
05	Budi	42	12	0	0,133	0
07	Rian	37	5	0,3	0	0
08	Kiki	32	1	0,8	0	0
09	Alda	35	3	0,5	0	0

10	Yoga	25	2	1	0	0
----	------	----	---	---	---	---

Berarti hanya ada dua karyawan saja yang memenuhi kategori tersebut, yaitu Sari dan Amir.

2.10 Konsep pengenalan sidik jari

Pengenalan sidik jari dianggap sebagai metode yang paling tua di antara jajaran teknologi biometrik lain yang ada, tetapi tidak dapat disangkal teknologi ini merupakan yang paling populer digunakan pada saat ini. Sidik jari memiliki banyak kelebihan dalam berbagai hal yang membuat biometrik ini lebih nyaman dan aman.

Kelebihan dari biometrik pengenalan sidik jari didukung dengan beberapa hal, seperti:

- Keunikan sidik jari, masing-masing dari sepuluh jari tangan kita memiliki bentuk sidik jari yang berbeda satu sama lain dan tentunya juga berbeda dengan milik orang lain bahkan sampai anak kembar sekalipun akan memiliki sidik jari yang unik.
- Tahan lama, sidik jari seseorang tidak akan pernah berubah kecuali karena mengalami kecelakaan yang parah sehingga mengakibatkan rusaknya sidik jari.
- Praktis, tidak seperti password, kode PIN, dan kartu pintar, sidik jari tidak mungkin ketinggalan atau hilang.

Perbedaan sidik jari ditemukan dari macam-macam pola yang terbentuk pada jari kita. Pola yang tercetak pada sidik jari pada dasarnya dapat dibagi menjadi tiga pola utama, yaitu putaran, lengkungan dan lingkaran.

Pola sidik jari yang berbentuk putaran merupakan pola yang paling umum dan paling banyak ditemukan, contoh bentuk sidik jari berpola putaran bisa dilihat pada gambar 2.40.



Gambar 2.40 Pola putaran

Berikutnya adalah pola yang berbentuk lengkungan. Ada dua jenis lengkungan, yang pertama adalah lengkungan biasa (gambar 2.41 (a)) dan lengkungan yang memiliki sudut lebih kecil dan tajam yang biasa disebut *tented arch* (gambar 2.41 (b))



(a)

(b)

Gambar 2.41 Pola lengkungan, (a) lengkungan biasa, dan (b) lengkungan tajam

Yang terakhir adalah alur sidik jari yang berpola lingkaran. Seperti namanya maka pola ini paling tidak memiliki satu cetakan pada sidik jari yang membentuk lingkaran. Contoh pola lingkaran dapat dilihat pada gambar 2.42.



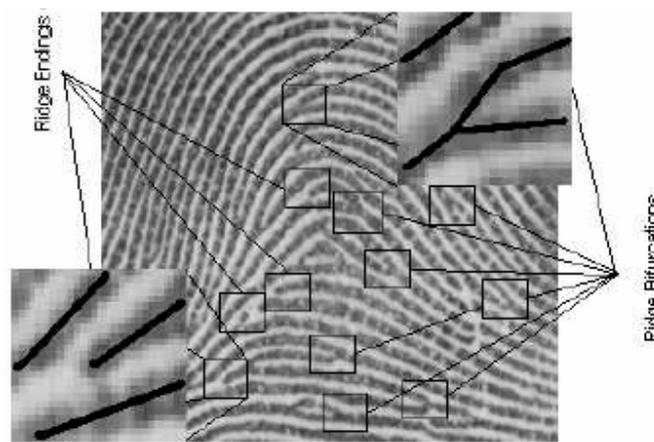
Gambar 2.42 Pola lingkaran

Bentuk-bentuk pola seperti ini yang akan mengidentifikasi seorang individu, walaupun mempunyai pola yang sama tetapi karakteristik pola itu pasti akan berlainan baik dari arah putaran, besarnya sudut, maupun banyaknya lingkaran. Oleh karena itu penggunaan sidik jari sebagai jawaban dari suatu masalah keamanan merupakan solusi yang tepat, selain dapat langsung mengidentifikasi seseorang dengan tingkat keamanan yang bisa diandalkan, juga dapat meningkatkan kenyamanan karena tak perlu merepotkan diri membawa berbagai macam barang untuk dijadikan alat identifikasi.

2.11 Proses pengenalan sidik jari

Sidik jari bisa menjadi salah satu alat pengaman yang paling personal dalam penggunaannya, hal ini dikarenakan struktur sidik jari setiap manusia bersifat pribadi dan pasti berbeda pada masing-masing individu. Perbedaan yang dimaksud ialah perbedaan pola alur yang terdapat di ujung jari, alur ini berbentuk seperti lembah panjang yang berbukit-bukit dan membentuk sebuah pola unik pada sidik jari.

Pada alur tersebut terdapat pola khusus yang bisa digunakan untuk mengenali sidik jari, pola ini membentuk titik penting yang disebut titik minutiae. Titik minutiae merupakan titik yang dihasilkan dari dua jenis pola khusus, yaitu; *ridge-ending* (ujung) dan *ridge-bifurcation* (cabang). Titik ujung adalah titik minutiae yang didapatkan dari ujung/akhir dari sebuah alur, sedangkan titik cabang adalah titik minutiae yang didapatkan dari tempat bersatunya dua alur yang terpisah.



Gambar 2.43 Titik minutiae, ridge ending dan bifurcation

Secara spesifik, konfigurasi global didefinisikan oleh struktur ridge yang digunakan untuk mengklasifikasikan suatu kelas dari sidik jari, pada saat pendistribusian titik-titik minutiae digunakan untuk membandingkan atau menyepadankan (*match*) dan membentuk kesamaan ciri atau pola diantara dua sidik jari sampel. Sistem identifikasi otomatis untuk sidik jari, yang berfungsi membandingkan sidik jari sebagai input dengan sekumpulan data sidik jari dalam suatu database, menyandarkan pada pola-pola ridge didalam suatu citra query untuk membatasi atau menspesifikasi pencarian (*searching*) di dalam suatu database yang

merupakan fingerprint indexing dan pada titik-titik minutiae untuk memastikan perbandingan sidik jari yang relevan.

Teknik matching untuk sidik jari secara luas dapat diklasifikasikan sebagai pengaitan keberadaan minutiae dan *correlation-based* pada sidik jari tersebut. Teknik yang diketahui yang mendasari pengidentifikasian dengan mengaitkan keberadaan minutiae, dilakukan dengan cara mencoba menyatukan dua himpunan titik-titik minutiae dan kemudian menentukan jumlah total dari minutiae yang sepadan. Sedangkan teknik *correlation-based* sebaliknya, membandingkan pola ridge dan jalur-jalur global pada sidik jari, untuk melihat apakah pola atau jalur tersebut sepadan atau tidak. Berdasarkan ujicoba, diketahui bahwa teknik matching yang berdasarkan minutiae memiliki keakuratan dalam mendeteksi titik-titik minutiae dan menggunakan teknik tersebut untuk membandingkan atau menyepadankan dua field minutiae yang menjalani suatu pentransformasian yang *non-rigid* (tidak kaku). Sedangkan unjuk kerja dari teknik *correlation-based* selalu dipengaruhi oleh distorsi yang tidak linier dan noise yang hadir di dalam citra. Secara umum, teknik yang berdasarkan minutiae lebih handal daripada teknik *correlation-based*.

2.12 Gabor Filtering dan Segmentasi Citra

Filter gabor 2D dapat diasumsikan sebagai gelombang pada suatu bidang yang kompleks yang dimodulasikan oleh pendekatan gaussian 2D. Filter tersebut secara optimal menangkap informasi frekuensi dan orientasi lokal. Dengan menyesuaikan filter gabor terhadap frekuensi dan arah yang spesifik, maka informasi frekuensi dan

orientasi lokal dapat diperoleh. Oleh karena itu mereka disesuaikan untuk informasi ekstraksi tekstur dari suatu citra.

Secara umum bentuk simetris filter gabor didalam wilayah spasial dapat dinyatakan sebagai berikut:

$$G_{\theta, f}(x, y) = \exp \left\{ -\frac{1}{2} \left[\left(\frac{x'}{\delta_x} \right)^2 + \left(\frac{y'}{\delta_y} \right)^2 \right] \right\} \cos(2\pi f x'),$$

$$x' = x \sin \theta + y \cos \theta,$$

$$y' = x \cos \theta - y \sin \theta$$

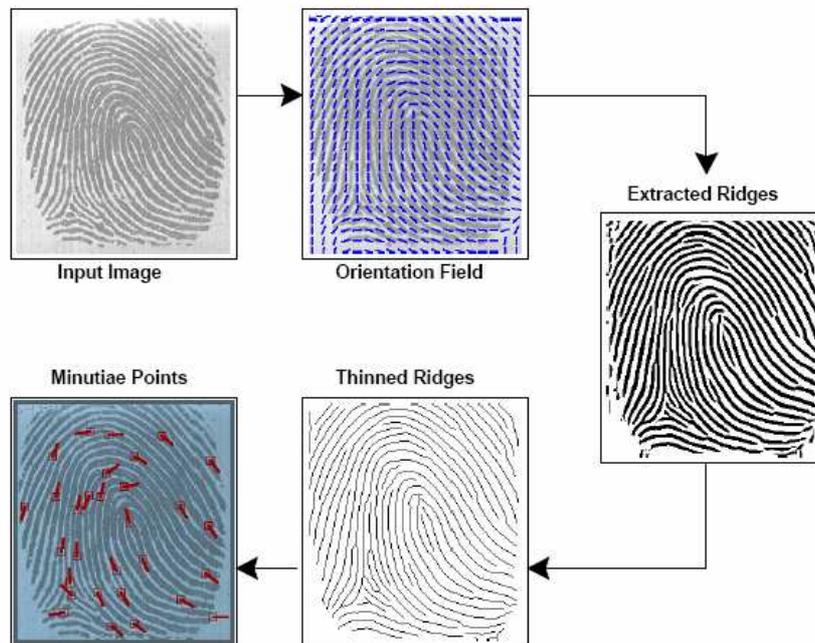
dimana f merupakan suatu frekuensi gelombang bidang sinusoid dengan sudut θ terhadap sumbu $-x$, dan δ_x dan δ_y merupakan standar deviasi dari kemas Gaussian sepanjang sumbu $-x$ dan sumbu $-y$.

Dengan demikian maka citra sidik jari dapat diperbaiki/memperjelas struktur ridge serta meningkatkan jalur-jalur tertentu didalam sidik jari. Perbaikan kualitas citra juga meliputi antara lain, pengurangan noise pada citra yang dapat berakibat pemberian informasi ridge yang salah, atau yang paling buruk kehilangan informasi jumlah minutiae pada citra sidik jari, juga mencegah munculnya minutiae-minutiae palsu, termasuk didalamnya adalah proses segmentasi citra sidik jari yaitu proses pemisahan informasi background dan foreground didalam citra sidik jari, untuk mencegah sel-sel yang terbentuk dari proses matching didominasi oleh informasi-informasi yang diperoleh dari background, yang berakibat pada nilai-nilai fitur yang dihitung akan memiliki ketidakakuratan indikasi kekuatan dari ridge-ridge sidik jari. Foreground berkaitan erat dengan suatu wilayah yang berisikan informasi yang dibutuhkan untuk proses matching, dan didalamnya memuat ridge-ridge dan titik-titik minutiae, sedangkan background tidak diperlukan didalam kasus ini karena tidak memuat informasi apapun yang diperlukan untuk proses matching. Segmentasi

dilakukan dengan mengamati variasi intensitas lokal di dalam citra grey-level original.

2.13 Ekstraksi Minutiae dan Matching Sidik Jari

Ekstraksi minutiae adalah proses dimana titik minutiae dideteksi didalam suatu citra sidik jari. Masing-masing minutiae dikarakteristik dengan lokasi (x, y) didalam citra. Orientasi sebesar θ dari ridge-ridge yang mana mengandung minutiae yang terdeteksi. Informasi ridge didalam region 64×64 disekitar titik (x, y) diasosiasikan terhadap keseluruhan minutiae yang sangat berguna didalam mencocokkan dua kumpulan minutiae. Skema ekstraksi minutiae dapat dilihat pada gambar berikut.



Gambar 2.44 Alur data dalam ekstraksi minutiae

Secara lebih luas ekstraksi minutiae dapat diklasifikasikan kedalam tahap-tahap berikut :

- i.** Aproksimasi medan orientasi
Orientasi dari citra sidik jari dihitung didalam berbagai blok yang tidak tumpang-tindih dengan pengujian gradien intensitas pixel di berbagai arah x maupun y didalam blok.
- ii.** Deteksi ridge
Ridge-ridge hadir didalam citra sidik jari yang diidentifikasi dengan pengaplikasian mask-mask yang mampu mempertajam atau menonjolkan nilai-nilai grey-level lokal maksimum sepanjang arah normal dari arah lokal ridge.
- iii.** Perampingan ridge
Suatu peta ridge yang terbentuk oleh stadium akhir dari stadium ekstraksi minutiae digunakan untk menghasilkan citra perampingan ridge.
- iv.** Pendeteksian minutiae
Sekumpulan aturan yang diaplikasikan terhadap ridge yang mengalami perampingan untuk memberikan label atau tanda ke setiap titik minutiae. Pada tahap penutup, suatu algoritma yang berkaitan dapat diaplikasikan untuk pengujian ke setiap titik-titik minutiae untuk mengetahui apakah titik-titik minutiae tersebut palsu atau tidak.

Setelah informasi fitur diekstraksi dari citra, maka kemudian dilakukan proses pencocokan (*matching*) sidik jari yang meliputi perbandingan suatu cetak query

dengan sekumpulan, satu atau lebih, cetak template. Kemudian matcher sidik jari melakukan pendekatan matching dengan memanfaatkan dua kumpulan informasi sidik jari yang berbeda untuk mencocokkan berbagai sidik jari, dimana informasi tersebut berupa fitur-fitur minutiae dan peta fitur ridge.