Fuzzy Logic

6.1 Pengantar

Fuzzy Logic adalah metodologi pemecahan masalah dengan beribu-ribu aplikasi dalam pengendali yang tersimpan dan pemrosesan informasi. *Fuzzy logic* menyediakan cara sederhana untuk menggambarkan kesimpulan pasti dari informasi yang ambigu, samar-samar atau tidak tepat. Sedikit banyak, *fuzzy logic* menyerupai pembuatan keputusan pada manusia dengan kemampuannya untuk bekerja dari data yang ditafsirkan dan mencari solusi yang tepat.

6.2 Sejarah *Fuzzy Logic*

Konsep *fuzzy logic* diperkenalkan oleh Prof. Lotfi Zadeh dari Universitas California di Berkeley pada 1965 dan dipresentasikan bukan sebagai suatu metodologi kontrol, tetapi sebagai suatu cara pemrosesan data dengan memperkenankan penggunaan *partial set membership* dibanding *crisp set membership* atau *non-membership*. Pendekatan pada *set* teori ini tidak diaplikasikan pada sistem kontrol sampai tahun 70-an karena kemampuan komputer yang tidak cukup pada saat itu.

Profesor Zadeh berpikir bahwa orang tidak membutuhkan kepastian, masukan informasi numerik dan belum mampu terhadap kontrol adaptif yang tinggi.

Konsep *fuzzy logic* kemudian berhasil diaplikasikan dalam bidang kontrol oleh E.H. Mamdani. Sejak saat itu aplikasi *fuzzy* berkembang kian pesat. Di tahun 1980-an negara Jepang dan negara-negara di Eropa secara agresif membangun produk nyata sehubungan dengan konsep logika *fuzzy* yang diintegrasikan dalam produk-produk kebutuhan rumah tangga seperti *vacum cleaner*, *microwave oven* dan kamera video. Sementara pengusaha di Amerika Serikat tidak secepat itu mencakup teknologi ini. *Fuzzy logic* berkembang pesat selama beberapa tahun terakhir. Terdapat lebih dari dua ribu produk di pasaran yang menggunakan konsep *fuzzy logic*, mulai dari mesin cuci hingga kereta berkecepatan tinggi. Setiap aplikasi tentunya menyadari beberapa keuntungan dari *fuzzy logic* seperti performa, kesederhaan, biaya rendah dan produktifitasnya.

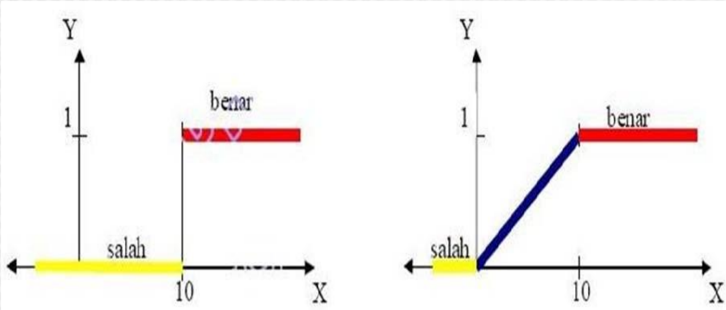
6.3 Definisi Logika *Fuzzy*

Logika *fuzzy* merupakan sebuah logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antara benar dan salah secara bersamaan namun berapa besar kebenaran dan kesalahan suatu nilai tergantung kepada bobot keanggotaan yang dimilikinya. Adapun perbedaan logika fuzzy dengan logika tegas adala:

- logika tegas memiliki nilai

tidak = 0.0, dan ya = 1.0

- *fuzzy* memiliki nilai antara 0,0 hingga 1,0



Gambar 6.1 Nilai Tegas dan *Fuzzy*

Fuzzy logic menawarkan beberapa karakteristik unik yang menjadikannya suatu pilihan yang baik untuk banyak masalah kontrol. Karakteristik tersebut antara lain:

1. Sudah menjadi sifatnya yang kuat selama tidak membutuhkan ketepatan, *input* yang bebas derau dan dapat diprogram untuk gagal dengan aman jika sensor arus balik dimatikan atau rusak. *Control output* adalah fungsi kontrol halus meskipun jarak variasi *input* yang cukup besar.
2. Selama *fuzzy logic controller* memproses aturan-aturan yang dibuat *user* yang memerintah target sistem kontrol, maka dapat dimodifikasi dengan mudah untuk meningkatkan atau mengubah secara drastis performa sistem. Sensor yang baru dapat dengan mudah digabungkan ke dalam sistem secara sederhana dengan menghasilkan aturan memerintah yang sesuai.

3. *Fuzzy logic* tidak terbatas pada sedikit masukan umpan-balik dan satu atau dua *output control*, tidak juga penting untuk menilai atau menghitung parameter rata-rata perubahan dengan tujuan agar diimplementasikan. Sensor data yang menyediakan beberapa indikasi untuk aksi dan reaksi sistem sudah cukup. Hal ini memungkinkan sensor menjadi murah dan tidak tepat sehingga menghemat biaya sistem keseluruhan dan kompleksitas rendah.
4. Karena operasi-operasi yang berbasiskan aturan, jumlah *input* yang masuk akal dapat diproses (1 sampai 8 atau lebih) dan banyak *output* (1 sampai 4 atau lebih) dihasilkan, walaupun pendefinisian *rulebase* secara cepat menjadi rumit jika terlalu banyak *input* dan *output* dipilih untuk implementasi tunggal selama pendefinisian *rules* (aturan), hubungan timbal baliknya juga harus didefinisikan. Akan lebih baik jika memecah sistem ke dalam potongan-potongan yang lebih kecil dan menggunakan *fuzzy logic controller* yang lebih kecil untuk didistribusikan pada sistem, masing-masing dengan tanggung jawab yang lebih terbatas.
5. *Fuzzy logic* dapat mengontrol sistem nonlinear yang akan sulit atau tidak mungkin untuk dimodelkan secara matematis. Hal ini membuka pintu bagi sistem kontrol yang secara normal dianggap tidak mungkin untuk otomatisasi.

Sedangkan karakteristik utama dari *fuzzy logic* yang ditemukan oleh Prof. Lotfi A. Zadeh adalah sebagai berikut:

- Dalam *fuzzy logic*, penalaran tepat dipandang sebagai suatu kasus terbatas dari penalaran kira-kira.
- Dalam *fuzzy logic* segala sesuatunya adalah masalah derajat.
- Sistem logis manapun dapat difuzzifikasi.
- Dalam *fuzzy logic*, pengetahuan diinterpretasikan sebagai koleksi dari *fuzzy* yang dipaksakan pada sekumpulan variabel.
- Kesimpulan dipandang sebagai sebuah proses dari perkembangan pembatas elastis.

Adapun langkah-langkah penggunaan *fuzzy logic* adalah sebagai berikut:

- Definisikan objektif dan kriteria kontrol:

1. Apa yang kita coba kontrol ?
2. Apa yang harus kita lakukan untuk mengontrol sistem ?
3. Respon seperti apa yang kita butuhkan ?
4. Apa mode kegagalan sistem yang mungkin ?

- Tentukan hubungan antara *input* dan *output* serta memilih jumlah minimum variabel *input* pada mesin *fuzzy logic* (secara khusus *error* dan rata-rata perubahan *error*).
- Dengan menggunakan struktur berbasis aturan dari *fuzzy logic*, jabarkan permasalahan kontrol ke dalam aturan IF X AND Y THEN Z yang mendefinisikan respon *output system* yang diinginkan untuk kondisi *input system* yang diberikan. Jumlah dan kompleksitas dari *rules* bergantung

pada jumlah parameter *input* yang diproses dan jumlah variabel *fuzzy* yang bekerjasama dengan tiap-tiap parameter. Jika mungkin, gunakan setidaknya satu variabel dan turunan waktunya. Walaupun mungkin untuk menggunakan sebuah parameter tunggal yang *error* saat itu juga tanpa mengetahui rata-rata perubahannya, hal ini melumpuhkan kemampuan sistem untuk meminimalisasi keterlampauan untuk sebuah tingkat *input*.

- Buat fungsi keanggotaan yang menjelaskan nilai *input* atau *output* yang digunakan di dalam *rules*.
- Buat rutinitas proses awal dan akhir yang penting jika diimplementasikan dalam *software*, sebaliknya program rules ke dalam mesin *hardware fuzzy logic*.
- *Test system*, evaluasi hasil, atur *rules* dan fungsi keanggotaan, dan *retest* sampai hasil yang memuaskan didapat.

6.4 Himpunan *Fuzzy*

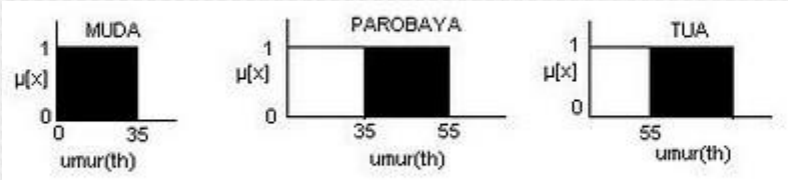
Sangat penting sekali bagi kita untuk terlebih dahulu mengetahui apa itu *crisp set* atau yang dikenal juga dengan *conventional set*, sebelum kita mengarah pada bagaimana himpunan *fuzzy* dibuat untuk kekurangan pada *crisp set*.

a. Crisp Set

Dalam kebanyakan jenis pemikiran setiap harinya dan refleksi bahasa darinya, orang-orang menggunakan *crisp set* untuk mengelompokkan sesuatu. Menjadi anggota dari *crisp set* adalah seluruhnya berhubungan atau tidak sama sekali. Seorang wanita dikatakan hamil ataupun tidak, ia tidak pernah “hamil sebagian” atau “sedikit hamil”. Berpikir dengan *crisp set* menjadikan segala sesuatunya lebih sederhana, karena sesuatu bisa merupakan anggota dari suatu *crisp set* atau tidak. *Crisp set* dapat digunakan untuk merepresentasikan gambaran pengertian hitam dan putih. Seringkali juga, saat sesuatu itu merupakan anggota dari sebuah *crisp set* maka ia kemudian (pada waktu yang sama) bukan merupakan anggota dari *crisp set* manapun. Kembali hal ini menyederhanakan penggunaan logika dengan proses pemikiran semacam ini. Konstruksi linguistik yang menggambarkan jenis pemikiran ini dapat benar-benar berguna, terutama saat kategori *crisp* digunakan. Pada himpunan tegas (*crisp*), nilai keanggotaan suatu 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.

Untuk lebih jelasnya, bisa dilihat dari contoh berikut:



Gambar 6.2 Nilai Keanggotaan

Dari Gambar 6.2 dapat dijelaskan bahwa:

- Apabila seseorang berusia 34 tahun, maka dikatakan MUDA (μMUDA[34] = 1);
- Apabila seseorang berusia 35 tahun, maka dikatakan TIDAK MUDA (μMUDA[35] = 0);
- Apabila seseorang berusia 35 tahun kurang 1 hari, maka dikatakan TIDAK MUDA (μMUDA[35 – 1hr] = 0);
- Apabila seseorang berusia 35 tahun, maka dikatakan PAROBAYA (μPAROBAYA[35] = 1);
- Apabila seseorang berusia 34 tahun, maka dikatakan TIDAK PAROBAYA (μPAROBAYA[34] = 0);
- Apabila seseorang berusia 55 tahun, maka ia dikatakan PAROBAYA (μPAROBAYA[55] = 1);
- Apabila seseorang berusia 35 tahun kurang 1 hari, maka dikatakan TIDAK PAROBAYA (μPAROBAYA[35 – 1hr] = 0);

Dari sini bisa katakan bahwa pemakaian himpunan *crisp* untuk menyatakan umur sangat tidak adil, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan. Oleh karena itu digunakanlah himpunan *fuzzy* untuk mengantisipasi hal tersebut.

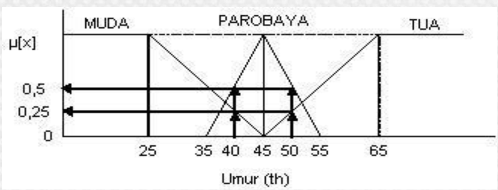
b. Fuzzy Set

Logika *fuzzy* lahir berdasarkan fenomena-fenomena alam yang serba tidak tepat dan samar ditinjau dari cara berpikir manusia, di mana pada kenyataannya tidak ada suatu kondisi atau pernyataan yang tepat 100% benar atau 100% salah. Prof. Lotfi A. Zadeh mengemukakan bahwa *true* atau *false* dalam logika Boolean tidak dapat merepresentasikan pernyataan yang tidak pasti yang berada di antara pernyataan *true* atau *false* tadi, seperti yang sering terjadi dalam dunia nyata. Untuk merepresentasikan nilai ketidakpastian antara *true* atau *false* tersebut, Prof. Lotfi A. Zadeh mengembangkan suatu teori berdasarkan *conventional set* yang disebutnya *fuzzy set* (himpunan *fuzzy*). Sebagai ganti dari pernyataan dengan nilai seluruhnya *true* atau semuanya *false*, logika *fuzzy* memberikan nilai yang spesifik pada setiap nilai di antara pernyataan *true* atau *false* dengan menentukan fungsi keanggotaan (*membership function*) bagi tiap nilai *input* dari proses *fuzzy* (*crisp input*) dan derajat keanggotaan (*degree of membership*) yaitu menyatakan derajat dari *crisp input* sesuai *membership function* antara 0 sampai 1, sehingga memungkinkan bagi suatu persamaan memiliki nilai *true* dan *false* secara bersamaan.

Menurut Prof. Lotfi A Zadeh, *fuzzy set* adalah sebuah kelas dari objek dengan serangkaian kesatuan dari *grades of membership* (nilai keanggotaan). Sebuah *set* dikarakterisasikan oleh sebuah fungsi keanggotaan (karakteristik) yang memberikan tiap objek sebuah nilai keanggotaan yang rentang nilainya antara 0 dan 1. Gagasan pencantuman (*inclusion*), penyatuan (*union*), persimpangan (*intersection*), pelengkap (*complement*), hubungan (*relation*), kecembungan (*convexity*), dan sebagainya diberikan pada *set* tersebut dan berbagai macam sifat dari pemikiran ini dalam konteks dari *fuzzy set* dibangun. Secara khusus, dalil untuk *fuzzy set* cembung dibuktikan tanpa perlu *fuzzy set* terputus. Aturan umum untuk teori *fuzzy set* dituliskan sebagai berikut:

$$f:[0,1]^n \rightarrow [0,1] \quad (6.1)$$

di mana *n* merupakan jumlah kemungkinan. Persamaan (6.1) menyatakan bahwa kita dapat mengambil *n* jumlah *event* yang mungkin dan menggunakan *f* untuk menghasikan hasil tunggal yang mungkin. Untuk lebih jelasnya mengenai himpunan *fuzzy* dapat dilihat pada contoh persoalan berikut:



Gambar 6.3 Himpunan *Fuzzy*

Dengan adanya himpunan *fuzzy* memungkinkan seseorang untuk dapat masuk ke dalam 2 himpunan yang berbeda, MUDA dan PAROBAYA, PAROBAYA dan TUA, dan sebagainya. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya. Dari Gambar 6.3, dapat dilihat bahwa:

- Seseorang yang berumur 40 tahun, termasuk dalam himpunan MUDA dengan µMUDA[40] = 0,25; namun dia juga termasuk dalam himpunan PAROBAYA dengan µPAROBAYA[40] = 0,5.
- Seseorang yang berumur 50 tahun, termasuk dalam himpunan MUDA dengan µMUDA[50] = 0,25; namun dia juga termasuk dalam himpunan PAROBAYA dengan µPAROBAYA[50] = 0,5.

Kalau pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1, pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila *x* memiliki nilai keanggotaan *fuzzy* µA[*x*] = 0 berarti *x* tidak menjadi anggota himpunan A, demikian pula apabila *x* memiliki nilai keanggotaan *fuzzy* µA[*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. Misalnya, jika nilai keanggotaan suatu himpunan *fuzzy* 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 *fuzzy* memiliki 2 atribut, yaitu:

- Linguistik, penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA.
- Numerik, suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb.

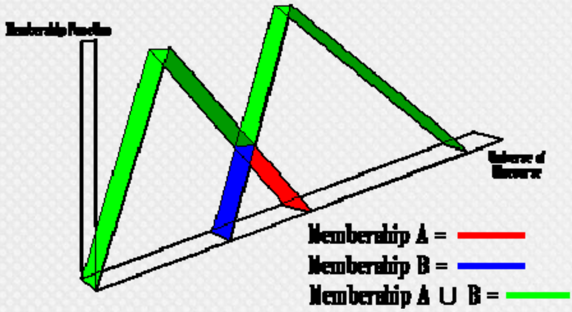
c. Fuzzy Set Operation

Fuzzy set operation adalah operasi yang dilakukan pada *fuzzy set*. Operasi-operasi ini merupakan generalisasi dari operasi *crisp set*. Terdapat lebih dari satu generalisasi yang mungkin. Operasi-operasi yang paling banyak digunakan secara luas disebut *standard fuzzy set operations*. Terdapat tiga operasi yaitu: *fuzzy unions*, *fuzzy intersections*, dan *fuzzy complements*.

d. Fuzzy Unions

Fungsi keanggotaan dari *union* dari dua *fuzzy set* A dan B dengan fungsi keanggotaan μA dan μB berturut-turut ditetapkan sebagai maksimum dari dua fungsi keanggotaan tersendiri. Ini disebut standar maksimum.

$$\mu_{A \cup B} = \max\left(\mu_A, \mu_B\right) \tag{6.2}$$



Gambar 6.4 Keanggotaan $A \cup B$

Operasi *union* dalam teori *fuzzy set* ekuivalen dengan operasi OR pada aljabar Boolean. Sifat (*property*) dari *fuzzy union* mencakup:

- *Boundary Condition*

  u(a,0) = a

- *Monotonicity*

  b ≤ d secara tidak langsung menyatakan u(a,b) ≤ u(a,d)

- *Commutativity*

  u(a,b) = u(b,a)

- *Associativity*

  u(a,u(b,d)) = u(u(a,b),d)

- *Continuity*

  u adalah fungsi yang berkelanjutan (*continuous*)
- *Superidempotency*

  u(a,a) > a
- *Strict monotonicity*

  $a_1 < a_2$ dan $b_1 < b_2$ secara tidak langsung menyatakan bahwa

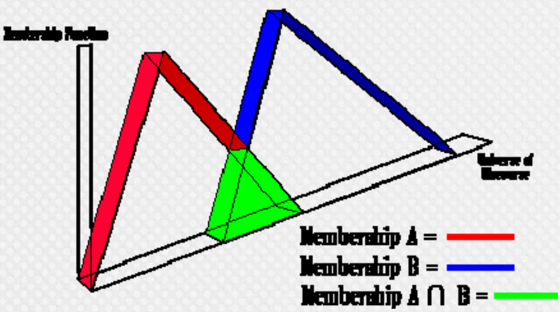
  $u(a_1,b_1) < u(a_2,b_2)$

Keterangan: u menyatakan *union* a, b, d, a_1, a_2, b_1, b_2 menyatakan *fuzzy set*.

e. Fuzzy Intersections

Fungsi keanggotaan dari *intersection* dari dua *fuzzy set* A dan B dengan fungsi keanggotaan μA dan μB berturut-turut ditetapkan sebagai minimum dari dua fungsi keanggotaan tersendiri. Ini disebut standar minimum.

$$\mu_{A \cap B} = \min\left(\mu_A, \mu_B\right) \tag{6.3}$$



Gambar 6.5 Keanggotaan $A \cap B$

Operasi *intersection* dalam teori *fuzzy set* ekuivalen dengan operasi AND pada aljabar Boolean. Sifat (*property*) dari *fuzzy intersection* mencakup:

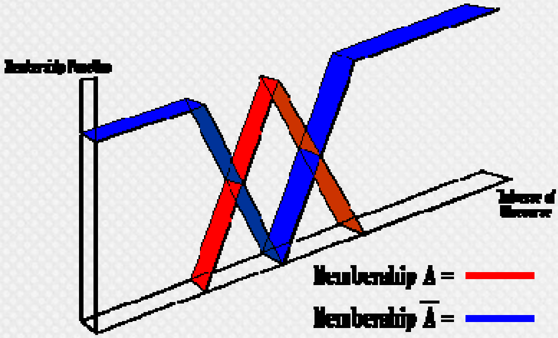
- *Boundary Condition*
  i(a,1) = a
- *Monotonicity*
  b ≤ d secara tidak langsung menyatakan i(a,b) ≤ i(a,d)
- *Commutativity*
  i(a,b) = i(b,a)
- *Associativity*
  i(a,i(b,d)) = i(i(a,b),d)
- *Continuity*
  i adalah fungsi yang berkelanjutan (*continuous*)
- *Subidempotency*
  i (a,a) ≤ a

Keterangan: i menyatakan *intersection*; a, b, d, a_1, a_2, b_1, b_2 menyatakan *fuzzy set*.

f. Fuzzy Complement

Fungsi keanggotaan dari *intersection* dari sebuah *fuzzy set* A dengan fungsi keanggotaan µA ditetapkan sebagai negasi dari fungsi keanggotaan yang ditentukan. Ini disebut standar negasi.

$$\mu_{\overline{A}} = 1 - \mu_A \tag{6.4}$$



Gambar 6.6 Keanggotaan $\overline{A}$

Operasi *complement* dalam teori *fuzzy set* ekuivalen dengan operasi NOT pada aljabar Boolean. Sifat (*property*) dari *fuzzy complement* mencakup:

- *Boundary Condition*

  c(0) = 1 dan c(1) = 0
- *Monotonicity*

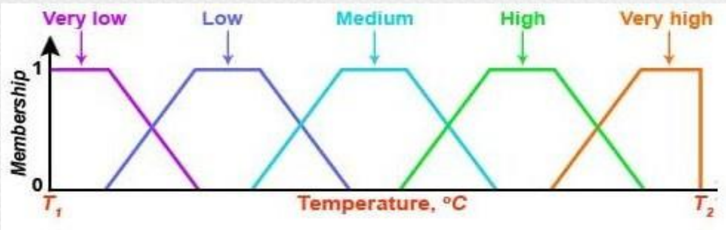
  Untuk semua a,b[0,1], jika a ≤ b maka c(a) ≥ c(b)
- *Continuity*

  c adalah fungsi yang berkelanjutan (*continuous*)
- *Involutions*

  c adalah suatu *involution*, yang berarti bahwa c(c(a)) = a untuk setiap a[0,1]

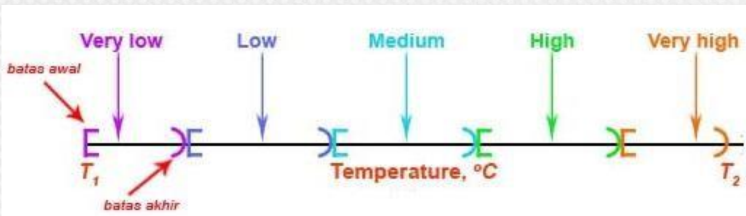
g. Himpunan Klasik vs Himpunan Fuzzy

Perbedaan himpunan *fuzzy* dengan himpunan klasik dapat diilustrasikan pda Gambar 6.7. Dari gambar tersebut dapat terlihat himpunan *fuzzy* memiliki batas yang tidak jelas, sedangkan himpunan klasik memiliki batas yang jelas. Pada gambar tanda ‘)’ mengartikan batas akhir dari sebuah *scope* dan tanda ‘[‘ mengartikan batas awal sebuah *scope* dari himpunan klasik. Rentang suhu yang dinyatakan dalam himpunan *fuzzy*:



Gambar 6.7 Himpunan *Fuzzy* Suhu

Rentang suhu yang dinyatakan dalam Himpunan Klasik:



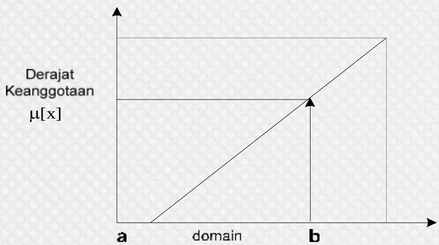
Gambar 6.8 Himpunan Klasik Suhu

6.5 Dasar Logika Fuzzy

- Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1 atau dapat dinyatakan dengan notasi $0 \leq \mu \leq 1$. Fungsi keanggotaan dapat dinyatakan dengan dua cara yaitu secara numerik dan fungsi.
- Domain adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam himpunan *fuzzy*.
- Ada beberapa fungsi yang dapat digunakan dalam merepresentasikan fungsi keanggotaan, yaitu:

1. Representasi Linear, ada 2 kemungkinan himpunan *fuzzy linear* yaitu:

a. **Representasi Linear Naik**, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju nilai domain yang memiliki derajat keanggotaan lebih tinggi.

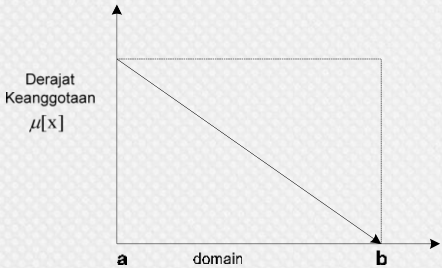


Gambar 6.9 Fungsi Keanggotaan Linear Naik

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \tag{6.5}$$

b. Representasi Linear Turun, 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.

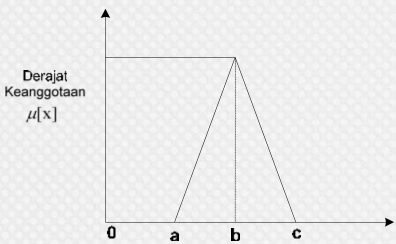


Gambar 6.10 Fungsi Keanggotaan Linear Turun

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} (b-x)/(b-a) & ; a \leq x \leq b \\ 0 & ; x \geq b \end{cases} \tag{6.6}$$

2. Representasi Kurva Segitiga

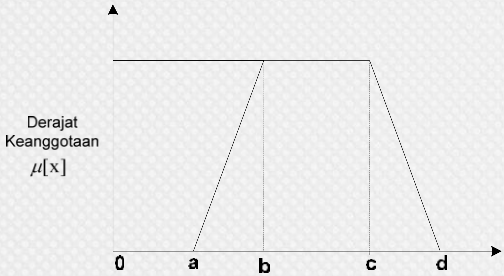


Gambar 6.11 Fungsi Keanggotaan Kurva Segitiga

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} 0 & ; x \le a \text{ atau } x \ge c \\ (x-a)/(b-a) & ; a \le x \le b \\ (c-x)/(c-b) & ; b \le x \le c \end{cases} \qquad (6.7)$$

3. Representasi Kurva Trapesium

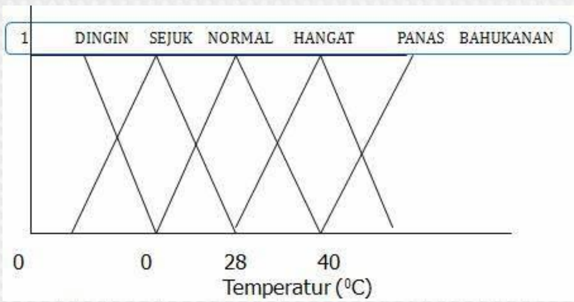


Gambar 6.12 Fungsi Keanggotaan Kurva Trapesium

Fungsi Keanggotaan:

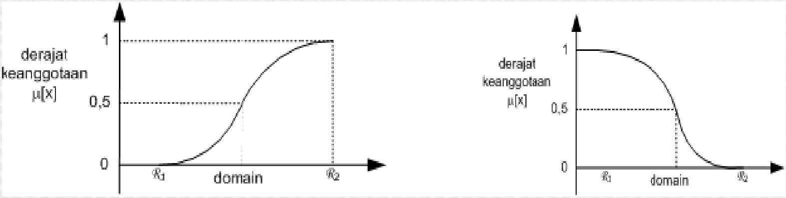
$$\mu(x) = \begin{cases} 0 & ; x \le a \text{ atau } x \ge d \\ (x-a)/(b-a) & ; a \le x \le b \\ 1 & ; b \le x \le c \\ (d-a)/(d-c) & ; c \le x \le d \end{cases} \qquad (6.8)$$

4. Representasi Kurva Bentuk Bahu



Gambar 6.13 Fungsi Keanggotaan Kurva Bentuk Bahu

5. Representasi Kurva – S



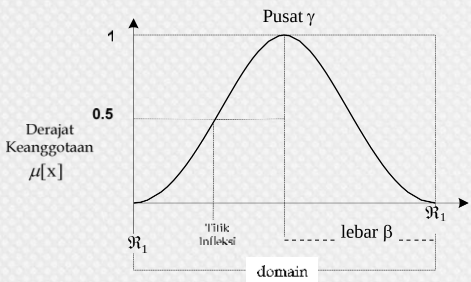
Gambar 6.14 Fungsi Keanggotaan Kurva S

Fungsi Keanggotaan:

$$S(x;\alpha,\beta,\gamma) = \begin{cases} 0; & x \le \alpha \\ 2((x-\alpha)/(\gamma-\alpha))^2; & \alpha \le x \le \beta \\ 1-2((\gamma-x)/(\gamma-\alpha))^2; & \beta \le x \le \gamma \\ 1; & x \ge \gamma \end{cases} \tag{6.9}$$

$$S(x;\alpha \quad \begin{cases} 1; & x \le \alpha \\ 1-2((x-\alpha)/(\gamma-\alpha))^2 & \alpha \le x \le \beta \\ 2((\gamma-x)/(\gamma-\alpha))^2 & \beta \le x \le \gamma \\ 0; & x \ge \gamma \end{cases} \tag{6.10}$$

6. Representasi Kurva Bentuk Lonceng



Gambar 6.15 Fungsi Keanggotaan Kurva Bentuk Lonceng

Fungsi Keanggotaan:

$$\mu(x) = \begin{cases} S\left(x;\gamma - \beta, \gamma - \dfrac{\beta}{2}, \gamma\right); & x \leq \gamma \\ 1 - S\left(x;\gamma, \gamma + \dfrac{\beta}{2}, \gamma + \beta\right); & x > \gamma \end{cases} \qquad (6.11)$$

- **Aritmatika Logika Fuzzy**
  - Gabungan (*union*), dalam sistem logika *fuzzy* dikenal dengan istilah *Max*, operasi kesamaan dinyatakan dengan persamaan:

    $\mu A \cup B$ = max($\mu A[x]$, $\mu B[y]$) (6.12)
  - Irisan (*intersection*), dalam sistem logika *fuzzy* dikenal dengan istilah *Min*, operasi kesamaan dinyatakan dengan persamaan:

    $\mu A \cap B$ = min($\mu A[x]$, $\mu B[y]$) (6.13)
  - Kesamaan (*equality*), operasi kesamaan dinyatakan dengan persamaan:

    $\mu A(x) = \mu B(x);\ x \in U$ (6.14)
  - Produk (Product), operasi produk dinyatakan dengan persamaan:

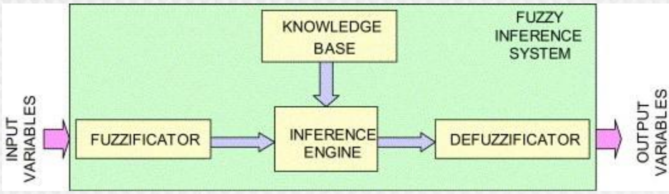
    $\mu A'(x) = 1 - \mu A(x);\ x \in U$ (6.15)
  - Komplemen (*complement*), operasi komplemen dinyatakan dengan persamaan:

    $\mu(A.B)(x) = \mu A(x).\mu B(x);\ x \in U$ (6.16)

6.6 Proses Sistem Kontrol Logika Fuzzy

Dalam sistem kontrol logika *fuzzy* (Gambar 6.16) terdapat beberapa tahapan operasional yang meliputi:

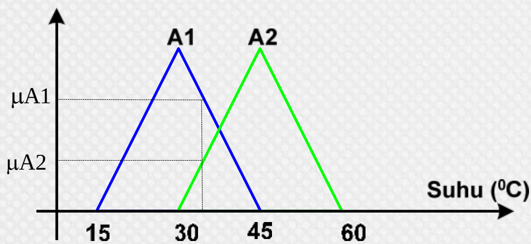
1. Fuzzifikasi
2. Aturan Dasar (*rule base*)
3. Penalaran (*inference machine*)
4. Defuzzifikasi



Gambar 6.16 Diagram Kontrol Logika *Fuzzy*

a. Fuzzifikasi

Fuzzifikasi adalah suatu proses pengubahan dari *input* (nilai tegas) yang ada menjadi nilai fungsi keanggotaan. Misal: merujuk pada Gambar 6.17 berikut, fuzzifikasi dari suhu 35^{0}C.



Gambar 6.17 Diagram Kontrol Logika *Fuzzy*

Pada Persamaan 6.17 contoh perhitungan fuzzifikasi dapat ditunjukkan sebagai berikut:

$$\begin{aligned} \mu A_1 &= \frac{c-x}{c-b} = \frac{45-35}{45-30} = \frac{2}{3} \\ \mu A_2 &= \frac{x-a}{b-a} = \frac{35-30}{45-30} = \frac{1}{3} \end{aligned} \tag{6.17}$$

b. Konstruksi Dasar Aturan Fuzzy (*Fuzzy Rule Base*)

Dasar aturan *fuzzy* terdiri atas satu *set* aturan *if-then fuzzy*, di mana dasar aturan ini merupakan inti dari sistem *fuzzy* yang merupakan sensor terhadap semua komponen lain yang digunakan untuk menerapkan suatu aturan yang memiliki alasan mendasar dan efisien. Hal penting yang harus dipertimbangkan dalam membangun sebuah dasar aturan *fuzzy* untuk sistem *fuzzy* maupun kontrol *fuzzy*, yaitu:

1. Variabel *input* dan *output*, pemilihan variabel *input* dan *output* yang tepat sangat menentukan kinerja dari model sistem yang akan dibuat.
2. Jangkauan nilai linguistik, nilai linguistik berhubungan dengan fungsi keanggotaan dari data. Untuk menghasilkan nilai batas lingustik yang tepat, maka dibutuhkan proses pengaturan sehingga dapat dihasilkan kinerja model sistem yang baik.
3. Penurunan aturan *fuzzy*, beberapa faktor yang sangat membantu ketika menurunkan suatu dasar aturan *fuzzy*, yaitu:

- Pengalaman atau pengetahuan
- Model *fuzzy*
- Model matematika

Interpretasi Aturan Implikasi *Fuzzy*:

- Fungsi implikasi adalah fungsi yang menyatakan hubungan akibat dari penerapan aturan *if-then* (jika-maka).
- Fungsi implikasi dalam matematika biasanya dinotasikan dengan (jika *p* maka *q*).

Metode untuk menentukan fungsi implikasi dalam sistem dan kontrol *fuzzy*:

1. Implikasi Dienes-Rescher
2. Implikasi Lukasiewicz
3. Implikasi Zadeh
4. Implikasi Gödel
5. Implikasi Mamdani

c. Penalaran (Inference Machine/Engine)

Fuzzy inference engine adalah prinsip logika *fuzzy* yang digunakan untuk mengkombinasikan aturan *if-then fuzzy* dalam suatu dasar aturan *fuzzy* pada suatu pemetaan *fuzzy set* pada himpunan *input* dengan *fuzzy* set pada himpunan *output*. Secara garis besar berdasarkan kombinasi yang digunakan, maka ada dua metode *inference engine*, yaitu:

1. *Inference engine* yang menggunakan kombinasi gabungan, berdasarkan dasar aturan individu *inference* dengan menggunakan kombinasi gabungan, maka terdapat dua metode *inference engine* yaitu:

- *Product Inference Engine*
- *Minimum Inference Engine*

2. *Inference engine* yang menggunakan kombinasi irisan. Berdasarkan dasar aturan individu dengan menggunakan kombinasi irisan maka terdapat tiga metode *inference engine*, yaitu:
   - Lukasiewicz *Inference Engine*
   - Zadeh *Inference Engine*
   - Dienes-Rescher *Inference Engine*

d. Defuzzifikasi

Defuzzy adalah suatu mekanisme untuk mengkonversi atau mengubah hasil *output fuzzy* menjadi suatu *output nonfuzzy* yang tegas. *Output* dalam bentuk *fuzzy* belum dapat hasil yang dapat langsung digunakan. Dalam teori logika *fuzzy*, terdapat beberapa metode *defuzzy* yang digunakan, yaitu:

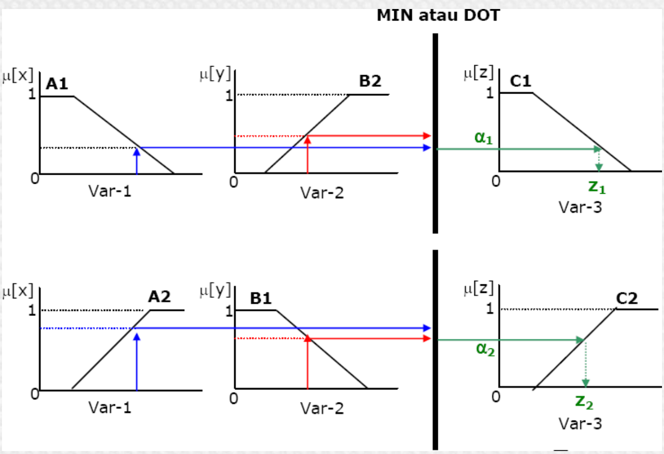
1. *Center of Gravity*
2. *Center of Average*
3. *Bisector*
4. MOM (*Mean of Maximum*)
5. LOM (*Largest of Maximum*)
6. SOM (*Smallest of Maximum*)

6.7 *Fuzzy Inference System* (FIS)

FIS merupakan salah satu metode logika *fuzzy* yang dapat digunakan untuk pengambilan suatu keputusan. Metode ini banyak digunakan sebagai sistem kontrol karena kemampuan metode ini untuk menggantikan seorang operator dalam mengambil keputusan.

a. Metode Tsukamoto

Pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-Then harus direpresentasikan dengan suatu himpunan *fuzzy* 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 6.18 Inferensi Menggunakan Metode Tsukamoto

$$\Downarrow \text{ rata-rata terbobot}$$

$$z = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2} \tag{6.18}$$

b. 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 *fuzzy*
2. Aplikasi fungsi implikasi (aturan)
3. Komposisi aturan
4. Penegasan (*deffuzy*)

1. Pembentukan himpunan *fuzzy*

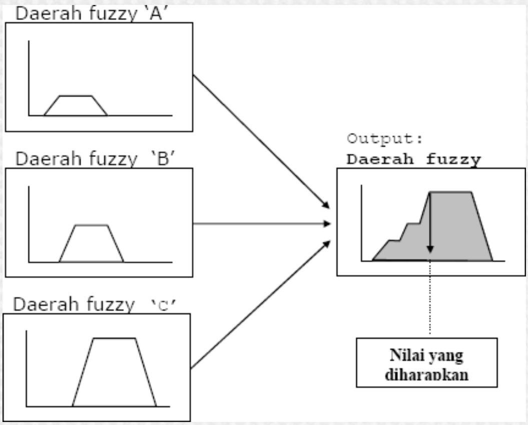
   Pada metode Mamdani, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.
2. Aplikasi fungsi implikasi

   Pada metode Mamdani, fungsi implikasi yang digunakan adalah Min.
3. Komposisi Aturan

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

4. Penegasan (*defuzzy*)

   Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut, sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output* seperti terlihat pada Gambar 6.19.



Gambar 6.19 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 (z*) daerah *fuzzy*. Secara umum dirumuskan:

$$Z^* = \frac{\int_Z Z\mu(Z)\,dz}{\int_z \mu(z)\,dz}$$

$$Z^* = \frac{\sum_{j=1}^{n} z_j \mu(z_j)}{\sum_{j=1}^{n} \mu(z_j)} \tag{6.19}$$

b. Metode Bisektor

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan separo dari jumlah total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan:

$$z_p \text{ sedemikian hingga } \int_{R1}^{p} \mu(z)\,dz = \int_{p}^{Rn} \mu(z)\,dz \tag{6.20}$$

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 dari domain yang memiliki nilai keanggotaan maksimum.

e. Metode Smallest of Maximum (SOM)

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

b. 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 olehTakagi-Sugeno Kang pada tahun 1985.

- Model Fuzzy Sugeno Orde-Nol

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

$$\text{IF } (x_1 \text{ is } A_1) \bullet (x_2 \; is \; A_2) \bullet (x_3 \text{ is } A_3) \bullet \ldots \bullet (x_n \text{ is } A_n) \text{ THEN z} = \text{k}$$

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

- Model Fuzzy Sugeno Orde-Satu

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

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

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. Apabila komposisi aturan menggunakan metode Sugeno, maka deffuzifikasi dilakukan dengan cara mencari nilai rata-ratanya.

c. Basis Data Fuzzy

- Model Tahani
- Model Umano

6.8 Aplikasi

Beberapa aplikasi logika fuzzy, antara lain:

1. Pada tahun 1990 pertama kali dibuat mesin cuci dengan logika *fuzzy* di Jepang (Matsushita Electric Industrial Company). Sistem *fuzzy* digunakan untuk menentukan putaran yang tepat secara otomatis berdasarkan jenis dan banyaknya kotoran serta jumlah yang akan dicuci. Input yang digunakan adalah: seberapa kotor, jenis kotoran, dan banyaknya yang dicuci. Mesin ini menggunakan sensor optik, mengeluarkan cahaya ke air dan mengukur bagaimana cahaya tersebut sampai ke ujung lainnya. Makin kotor, maka sinar yang sampai makin redup. Di samping itu, sistem juga dapat menentukan jenis kotoran (daki atau minyak).

2. Transmisi otomatis pada mobil. Mobil Nissan telah menggunakan sistem *fuzzy* pada transmisi otomatis, dan mampu menghemat bensin 12 – 17%.
3. Kereta bawah tanah Sendai mengontrol pemberhentian otomatis pada area tertentu.
4. Ilmu kedokteran dan biologi, seperti sistem diagnosis yang didasarkan pada logika *fuzzy*, penelitian kanker, manipulasi peralatan prostetik yang didasarkan pada logika *fuzzy*.
5. Manajemen dan pengambilan keputusan, seperti manajemen basisdata yang didasarkan pada logika *fuzzy*, tata letak pabrik yang didasarkan pada logika *fuzzy*, sistem pembuat keputusan di militer yang didasarkan pada logika *fuzzy*, pembuatan *games* yang didasarkan pada logika *fuzzy*.
6. Ekonomi, seperti pemodelan *fuzzy* pada sistem pemasaran yang kompleks.
7. Klasifikasi dan pencocokan pola.
8. Psikologi, seperti logika *fuzzy* untuk menganalisis kelakuan masyarakat, pencegahan dan investigasi kriminal.
9. Ilmu-ilmu sosial, terutama untuk pemodelan informasi yang tidak pasti.
10. Ilmu lingkungan, seperti kendali kualitas air, prediksi cuaca.