

BAB 2

LANDASAN TEORI

2.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan atau *Artificial Intelligence* (AI) adalah salah satu cabang ilmu komputer yang berhubungan dengan tingkah laku yang pintar, belajar dan dapat beradaptasi pada mesin. Penelitian pada *Artificial Intelligence* dipusatkan pada penghasilan mesin yang mengotomatisasikan tugas-tugas yang memerlukan tingkah laku yang pintar. Contohnya kontrol, perencanaan dan penjadwalan, kemampuan menjawab diagnostik dan pertanyaan-pertanyaan pelanggan, handwriting, suara dan pengenalan wajah. Untuk itu, Artificial Intelligence telah menjadi disiplin ilmu, difokuskan pada penyediaan solusi masalah-masalah kehidupan nyata, aplikasi software, game strategi seperti catur komputer dan video game lainnya.

2.1.1 Sejarah Kecerdasan Buatan

Kecerdasan buatan termasuk bidang ilmu yang relatif muda. Pada tahun 1950-an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang dikerjakan oleh manusia. Alan Turing, seorang matematikawan dari Inggris pertama kali mengusulkan adanya tes untuk melihat bisa tidaknya sebuah mesin dikatakan cerdas. Hasil tes tersebut kemudian dikenal dengan Turing Test.

Kecerdasan buatan itu atau "*artificial intelligence*" itu sendiri muncul dari seorang profesor dari *Massachusetts Institute of Technology* yang bernama John McCarthy pada tahun 1956 pada Dartmouth Conference yang dihadiri oleh para

peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan dari kecerdasan buatan, yaitu: mengetahui dan memodelkan proses-proses berfikir manusia dan mendesign mesin agar dapat menirukan kelakuan manusia tersebut. Beberapa program AI yang mulai dibuat pada tahun 1956-1966, antara lain :

1. Logic Theorist, diperkenalkan pada Dartmouth Conference, program ini dapat membuktikan teorama-teorama matematika.
2. Sad Sam, diprogram oleh Robert K. Lindsay (1960). Program ini dapat mengetahui kalimat-kalimat sederhana yang ditulis dalam bahasa inggris dan mampu memberikan jawaban dari fakta-fakta yang didengar dalam sebuah percakapan.
3. ELIZA, diprogram oleh Joseph Weizenbaum (1967). Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan.

2.1.2 Perbedaan Kecerdasan Buatan Dan Kecerdasan Alami

Jika dibandingkan dengan kecerdasan alami (kecerdasan yang dimiliki oleh manusia), kecerdasan buatan memiliki beberapa kelebihan secara komersial antara lain :

- a. Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini dimungkinkan karena kemampuan manusia untuk mengingat sesuatu cukup terbatas. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer dan program tidak di ubah.
- b. Kecerdasan buatan lebih mudah diduplikasi dan disebar. Menduplikasikan pengetahuan manusia dari satu orang ke orang lain

membutuhkan proses yang sangat lama, dan juga suatu keahlian itu tidak akan pernah dapat diduplikasi dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dengan mudah dipindahkan ke komputer yang lain.

- c. Kecerdasan buatan akan lebih murah dibanding dengan kecerdasan alami. Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama
- d. Kecerdasan buatan bersifat konsisten. Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer sedangkan kecerdasan alami akan senantiasa mengalami perubahan.
- e. Kecerdasan buatan dapat didokumentasikan. Keputusan yang dibuat oleh komputer dapat didokumentasikan dengan mudah dengan melacak setiap aktivitas dari sistem tersebut.
- f. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding kecerdasan alami.
- g. Kecerdasan buatan dapat mengerjakan pekerjaan lebih teliti dan lebih baik dibanding kecerdasan alami.

Sedangkan keuntungan dari kecerdasan alami adalah :

- a. Kreatif. Kemampuan untuk menambah ataupun memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.

- b. Kecerdasan alami memungkinkan seseorang untuk menggunakan pengalaman secara langsung. Sedangkan pada kecerdasan buatan harus bekerja dengan input-input simbolik.
- c. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

2.1.3 Klasifikasi Ilmu *Artificial Intelligence*

Artificial Intelligence dibagi dalam dua bidang studi yaitu: *Artificial Intelligence* Konvensional dan *Computational Intelligence* (CI). *Artificial Intelligence* Konvensional terutama meliputi metode yang diklasifikasikan sebagai mesin yang belajar (*machine learning*), dicirikan dengan analisis statistika. Ini juga dikenal sebagai simbolik *Artificial Intelligence*, logikal *Artificial Intelligence*, neat *Artificial Intelligence* (menyatakan solusi harus luwes, jelas dan benar) dan *Good Old Fashioned Artificial Intelligence* (GOFAI). Metodenya meliputi :

- Sistem Pakar (*Expert System*): menerapkan kemampuan menalar untuk mencapai kesimpulan. Sebuah sistem pakar dapat memproses sejumlah besar informasi yang diketahui dan menyediakan kesimpulan berdasarkan informasi tersebut.
- *Case based reasoning* : merupakan proses penyelesaian masalah baru berdasarkan pada solusi dari masalah lampau yang serupa. *Case based reasoning* dapat disebut juga menganalogikan masalah yang mirip dengan masalah lampau.

- *Bayesian networks* : merupakan grafik asiklik tertuju dimana tiap node-nya mewakili satu variabel, dan garis penghubung tiap node disebut hubungan ketergantungan statistik diantara variabel tersebut dan nilai distribusi probabilitas lokal untuk tiap variabel yang diberikan dari node sebelumnya.
- *Behavior based Artificial Intelligence* : merupakan kecerdasan buatan dimana kecerdasannya dibuat dari banyak elemen modul yang relatif sederhana dalam membuatnya. Setiap elemen berfungsi hanya pada konteks khusus, yang hanya dikenali oleh modul tersebut.
- *Computation Intelligence* meliputi pengembangan iterative atau belajar. Pembelajaran didasarkan pada kumpulan data-data empiris yang dihubungkan dengan kecerdasan buatan non-simbolik. Beberapa metode pembelajaran tersebut adalah sebagai berikut:
 - Neural Network: sistem dengan kemampuan pengenalan pola.
 - Fuzzy System: teknik untuk menalar dibawah ketidakpastian, secara luas digunakan pada industri modern dan sistem kontrol produk konsumen.
 - Evolutionary computation: menerapkan konsep yang terinspirasi dari permasalahan biologi seperti populasi, mutasi dan *survival of the fittest* untuk menghasilkan solusi yang lebih baik. Metode ini dapat dibagi lagi menjadi *evolutionary algorithm* (contoh *genetic algorithm*) dan *swarm intelligence* (contoh *ant algorithm*).

Dengan *hybrid intelligent system* berusaha untuk membuat kombinasi dari dua group ini. Aturan kesimpulan pakar dapat dihasilkan melalui neural network atau aturan produksi dari pembelajaran statistika seperti ACT-R. Otak manusia

menggunakan multiple teknik untuk bersama-sama memformulasikan dan menghasilkan *cross-check*. Jadi, integrasi terlihat menjanjikan dan mungkin perlu dalam *artificial intelligence* yang sebenarnya.

Satu metode baru yang menjanjikan disebut *intelligence amplification*, dimana mencoba mencapai tingkat *artificial intelligence* dalam proses pengembangan yang evolusioner sebagai penjelasan kecerdasan manusia melalui teknologi.

2.1.4 Penggunaan *Artificial Intelligence* pada Bisnis

Bank menggunakan sistem kecerdasan buatan untuk mengorganisasikan operasi, penanaman saham dan mengatur properti. Pada agustus 2001, robot melebihi manusia dalam simulasi kompetisi pertukaran keuangan (BBC news, 2001). Klinik medikal dapat menggunakan sistem kecerdasan buatan untuk mengatur dan menjadwalkan, membuat rotasi pegawai dan menyediakan informasi secara periodik. Banyak juga aplikasi tergantung pada jaringan syaraf tiruan (ANN – *artificial neural network*), yaitu jaringan yang mempolakan organisasi dalam cara meniru otak manusia. Aplikasi ini mempunyai banyak kebaikan dalam hal pengenalan pola. Institusi keuangan telah lama menggunakan sistem seperti ini untuk mendeteksi klaim di luar kewajaran. ANN juga digunakan secara luas pada *homeland security*, suara dan pengenalan teks, diagnosis medical, data mining dan *e-mail spam filtering*. *Homeland security* mempunyai fungsi untuk mencegah, mendeteksi, menanggapi dan memperbaiki suatu tingkah laku kegiatan seperti teroris dan juga bencana alam.

Robot juga telah umum digunakan di banyak industri. Mereka sering diberikan pekerjaan yang dipertimbangkan berbahaya bagi manusia. Robot juga terbukti efektif pada pekerjaan yang sangat repetitive yang mungkin menyebabkan kesalahan

atau kecelakaan tugas karena kehilangan konsentrasi, jika dikerjakan oleh manusia. Contohnya General Motor (GM) menggunakan 16.000 robot untuk tugas *painting*, *welding* dan *assembly*. Jepang merupakan negara pemimpin dalam penggunaan robot, dari data yang didapat pada 1995, 700.000 robot digunakan secara luas diseluruh dunia dan lebih dari 500.000 berasal dari Jepang (Encarta, 2006).

2.2 Fuzzy Logic

Logika fuzzy atau *fuzzy logic* bermula dari kenyataan bahwa dunia nyata sangat kompleks. Kompleksitas ini muncul dari ketidakpastian dalam bentuk informasi *imprecision* (ketidakpastian). Mengapa komputer yang dibuat oleh manusia tidak mampu menangani persoalan yang kompleks dan tidak presisi ini sedangkan manusia bisa. Jawabanya adalah manusia mempunyai kemampuan untuk menalar (*Reasoning*) dengan baik yaitu kemampuan yang komputer tidak mempunyainya. Pada suatu sistem jika kompleksitasnya berkurang, maka persamaan matematik dapat digunakan dan ketelitian yang dihasilkan menjadi sangat berguna dalam pemodelan sistem tetapi jika kompleksitasnya bertambah dimana persamaan matematik tidak dapat digunakan, logika fuzzy menjadi salah satu alternatif penyelesaiannya. Logika fuzzy merupakan alternatif cara berpikir yang dapat memodelkan kompleks sistem menggunakan pengetahuan dan pengalaman yang dipunyai logika dimulai sebagai studi tentang bahasa dalam argument dan persuasif, dan hal itu kemungkinan digunakan untuk menilai kebenaran dari rantai reasoning, dalam bentuk matematikanya sebagai contoh adalah sebagai berikut : Dalam dua nilai logika proporsinya bisa benar atau salah, tetapi tidak keduanya. Kebenaran

ditandai dengan *statement*(pernyataan) nilai kebenaran. Dalam fuzzy logic proporsi kemungkinan benar atau salah atau mempunyai setengah nilai kebenaran , seperti kemungkinan benar.

Ide pemikiran pada sistem fuzzy yaitu nilai kebenaran (dalam *fuzzy logic*) atau nilai fungsi keanggotaan (*Membership function*) dalam *fuzzy set* diindikasikan oleh nilai pada range $[0.0,1.0]$, dengan 0.0 mewakili kesalahan absolut dan 1.0 mewakili kebenaran absolut. Dalam kasus ini berhubungan dengan *multivalued logic*(logika bernilai banyak).

Sebagai contoh , kalimat :”Arthur is old.”

Jika umur Arthur 80, kemungkinan akan memberikan *statement* nilai kebenaran 0.80. *Statement* dapat diartikan menjadi set terminologi sebagai berikut:”Arthur adalah anggota dari set orang tua .” Suatu logika berdasar pada dua nilai kebenaran *true* and *false* kadang-kadang tidak cukup manakala menguraikan dan menggambarkan pemikiran manusia. Logika fuzzy menggunakan keseluruhan interval 0(*False*) dan 1(*True*) untuk menguraikan pemikiran manusia . Sebagai hasilnya logika fuzzy diterapkan dalam mengatur sebuah sistem karena sifatnya yang lebih sesuai dengan cara berpikir manusia.

2.2.1 Sejarah *Fuzzy Logic*

Sistem fuzzy merupakan suatu alternatif untuk pemikiran tradisional dari set keanggotaan dan logika yang berasal dari filosofi yunani dan aplikasi pada intelegensia semu. Logika fuzzy pertama kali diperkenalkan oleh seseorang profesor

computer science dari *University of California* di Berkeley yang bernama Lofti A.Zadeh pada tahun 1965 dan berhasil diaplikasikan dalam bidang kontrol oleh E.H.Mamdani. Sejak itu aplikasi dari logika fuzzy ini berkembang pesat terutama di negara Jepang dengan dihasilkannya ribuan paten mulai dari bermacam-macam produk elektronik sampai aplikasi pada kereta api di kota Sendai. Logika fuzzy pada dasarnya merupakan logika bernilai banyak (*Multivalued Logic*) yang dapat mendefinisikan nilai diantara keadaan yang biasa dikenal seperti ya atau tidak, hitam atau putih, benar atau salah. Logika fuzzy menirukan cara manusia mengambil keputusan dengan kemampuannya bekerja dari data yang samar atau tidak rinci dan menemukan penyesuaian yang tepat.

2.2.2 Teori Set Fuzzy (*Fuzzy Set Theory*)

Teori set fuzzy atau *Fuzzy set theory* adalah perpanjangan dari teori himpunan crisp yang konvensional. *Fuzzy set theory* mengatur konsep dari *partial truth* (nilainya antara 1(True) dan 0(False)). Dikenalkan oleh Prof. Lotfi A. Zadeh di UC/Berkeley pada tahun 1965 untuk dimaksudkan sebagai memodelkan kesamaran dan ambiguitas dalam sistem yang kompleks.

Ide dari fuzzy set sendiri adalah simpel dan alami. Sebagai contoh, kita mau mendefinisikan sebuah set (Himpunan) dari *gray level* yang propertinya *dark*. Dalam teori set yang klasik, kita harus mendeterminasikan sebuah *threshold*, katakanlah *gray level*nya bernilai 100. Semua *gray level* yang nilainya antara 0 dan 100 adalah elemen dari set ini, yang lainnya tidak termasuk dalam set ini (Gambar 2.1 bagian kiri). Tetapi kegelapannya hanya masalah pada derajat. Jadi sebuah fuzzy set bisa memodelkan properti ini dengan lebih baik. Untuk mendefinisikan himpunan ini,

kita juga membutuhkan dua *threshold*, katakanlah *gray level* 50 dan 150. Semua *gray level* yang dibawah 50 adalah full member dari himpunannya, *gray level* yang di atas 150 bukan merupakan member dari himpunannya. *Gray level* diantara 50 dan 150, mempunyai sebagian *membership* dalam himpunan (Gambar 2.1 bagian kanan).



Gambar 2.1 Representation of "dark gray-levels" with a crisp and a fuzzy set (Adapted from: Tizhoosh, Fuzzy Image Processing, ©CopyRight Springer,1997)

2.2.3 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik – titik input data ke dalam derajat keanggotaannya yang memiliki interval antara 0 sampai 1. Pada dasarnya ada dua cara mendefinisikan keanggotaan dari *Fuzzy Set*, yaitu secara numeris dan fungsional. Definisi numeris menyatakan fungsi derajat keanggotaan sebagai vektor jumlah yang tergantung pada tingkat diskretisasi. Misalnya, jumlah elemen diskret dalam semesta pembicaraan. Definisi fungsional menyatakan derajat keanggotaan sebagai batasan ekspresi analitis yang dapat dihitung. Standar atau ukuran tertentu pada fungsi keanggotaan secara umum berdasar atas semesta X bilangan *real*.

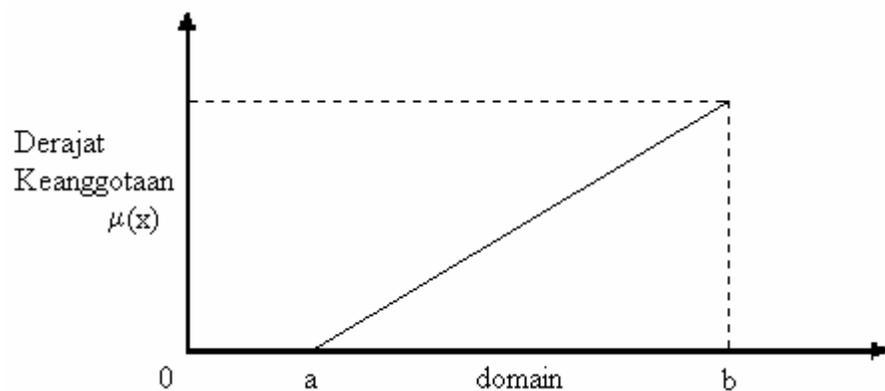
Fungsi keanggotaan (*membership function*) yang sering digunakan terdiri dari beberapa jenis, yaitu :

- Fungsi Linear

Fungsi keanggotaan dari fungsi linear adalah :

$$\mu(x) = \begin{cases} 0 & \text{jika } x < a, x > c \\ (x - a)/(b - a) & \text{jika } a \leq x \leq b \\ 1 & \text{jika } x \geq b \end{cases}$$

Gambar grafik fungsi keanggotaannya adalah :



Gambar 2.2 Grafik Fungsi keanggotaan Linear

- Fungsi-S (S-function)

Fungsi-S atau Sigmoid merupakan kurva yang dibentuk sehubungan dengan kenaikan dan penurunan nilai yang tidak linear. Fungsi keanggotaannya akan tertumpu pada 50% nilai keanggotaan yang sering disebut dengan titik infleksi.

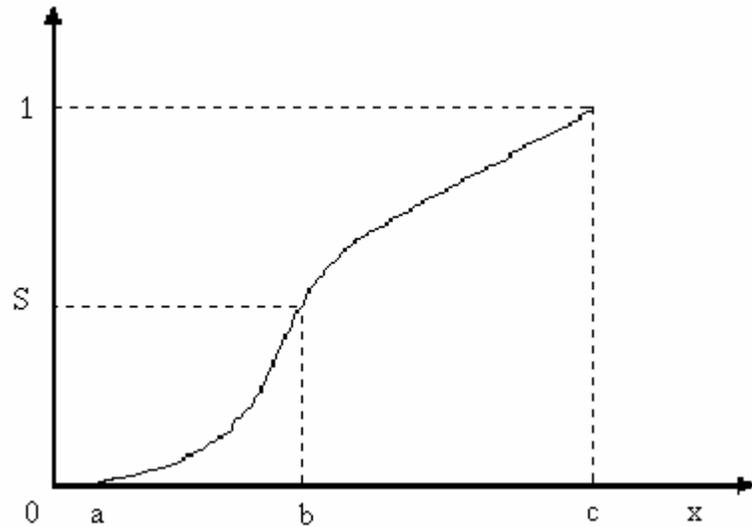
Persamaan dari Fungsi-S ini adalah :

$$0 \quad \text{jika } x < a$$

$$\begin{aligned}
 S(x,a,b) &= 2[(x-a)/(c-a)]^2 && \text{jika } a \leq x \leq b \\
 &1 - 2[(x-a)/(c-a)]^2 && \text{jika } b \leq x \leq c \\
 &1 && \text{jika } x > c
 \end{aligned}$$

Dengan $b = a+c/2$.

Gambar grafik fungsi keanggotaannya adalah :



Gambar 2.3 Grafik fungsi keanggotaan S

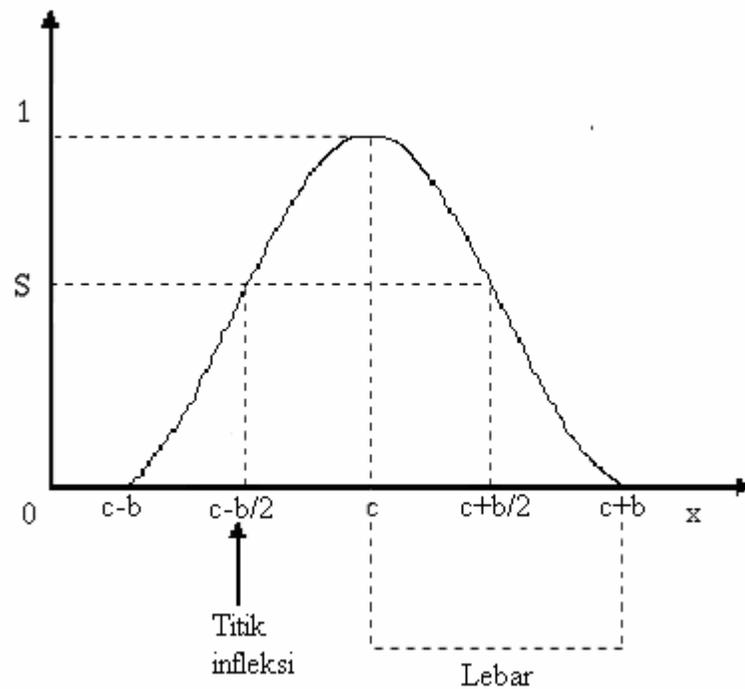
- Fungsi-PI (*PI-function*)

Fungsi-PI merupakan salah satu kurva berbentuk lonceng, di mana derajat keanggotaan yang bernilai 1 tepat terletak pada pusat domain c (lihat gambar 2.4).

Persamaan fungsi-PI diperoleh dari persamaan fungsi-S, yaitu :

$$\begin{aligned}
 PI(x,b,c) &= S(x,c-b,c-b/2,c) && \text{jika } x \leq c \\
 &1 - S(x;c,c+b/2,c+b) && \text{jika } x \geq c
 \end{aligned}$$

Gambar grafik fungsi keanggotaannya adalah :



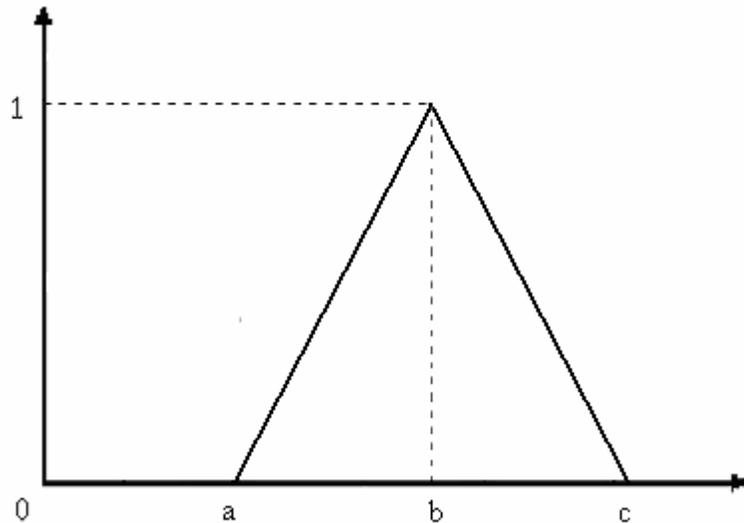
Gambar 2.4 Grafik fungsi keanggotaan PI

Fungsi keanggotaan segitiga (*Triangular membership function*)

Persamaan untuk bentuk segitiga ini adalah :

$$T(x;a,b,c) = \begin{cases} 0 & \text{jika } x < a, x > c \\ (x - a)/(b - a) & \text{jika } a \leq x \leq b \\ (c - x)/(c - b) & \text{jika } b \leq x \leq c \end{cases}$$

Gambar grafik fungsi keanggotaannya adalah :



Gambar 2.5 Grafik fungsi keanggotaan segitiga

Fungsi keanggotaan trapesium (*Trapezoidal membership function*)

Persamaan untuk bentuk trapezium ini adalah :

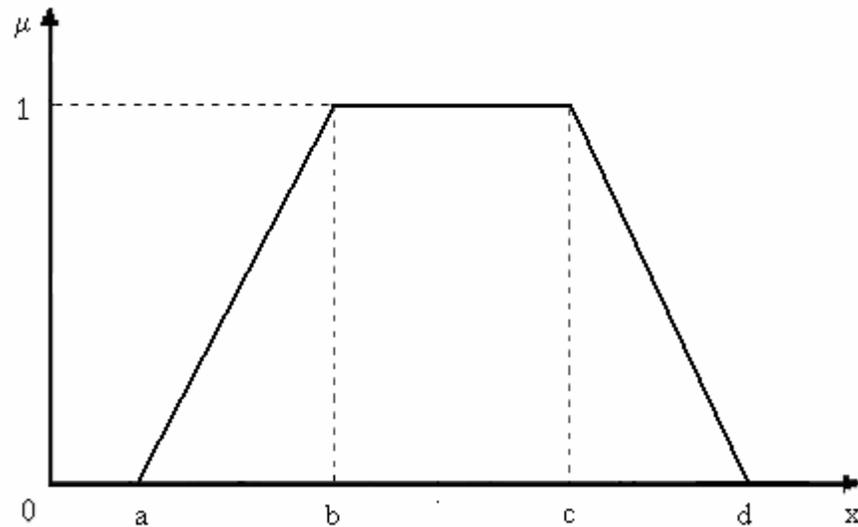
0 jika $x < a$, $x > d$

$Z(x;a,b,c,d) = 1$ jika $b \leq x \leq c$

$(x - a)/(b - a)$ jika $a \leq x \leq b$

$(d - x)/(d - c)$ jika $c \leq x \leq d$

Gambar grafik fungsi keanggotaannya adalah :



Gambar 2.6 Grafik fungsi keanggotaan trapesium

2.2.4 Variabel Fuzzy (Variabel Linguistik)

Inti dari teknik pemodelan fuzzy adalah nama suatu *fuzzy set* yang disebut variabel linguistik. Variabel linguistik merupakan variabel yang bernilai kata/kalimat, bukan angka. Sebagai alasan menggunakan kata/kalimat daripada angka karena informasi yang disampaikan akan menjadi lebih informatif, meskipun kenyataannya peranan linguistik kurang spesifik dibandingkan dengan angka. Variabel linguistik merupakan konsep penting dalam *Fuzzy Logic* dan memegang peranan penting dalam beberapa aplikasi.

Misal, jika “kecepatan” adalah suatu variabel linguistik, maka nilai linguistik untuk variabel kecepatan tersebut antara lain “lambat”, “sedang”, dan “cepat“. Hal ini sesuai dengan kebiasaan manusia sehari – hari dalam menilai sesuatu, misalnya : “Ia mengendarai mobil dengan **cepat**“, tanpa memberikan nilai berapa kecepatannya.

Konsep tentang variabel linguistik ini juga pertama kali diperkenalkan oleh Prof. Lofti Zadeh. Variabel linguistik ini menurut Zadeh dikarakteristikan dengan

$$(X, T(x), U, G, M)$$

dimana :

X = Nama variabel (variabel linguistik).

$T(x)$ = Semesta pembicaraan untuk x atau disebut juga nilai linguistik dari x .

U = Jangkauan dari setiap nilai fuzzy untuk x yang dihubungkan dengan variabel dasar U .

G = Aturan sintaksis untuk memberikan nama (x) pada setiap nilai X .

M = Aturan semantik yang menghubungkan setiap X dengan artinya.

Sebagai contoh, jika :

X = “kecepatan”, U $[0,100]$, dan $T(\text{kecepatan}) = \{\text{lambat, sedang, cepat}\}$

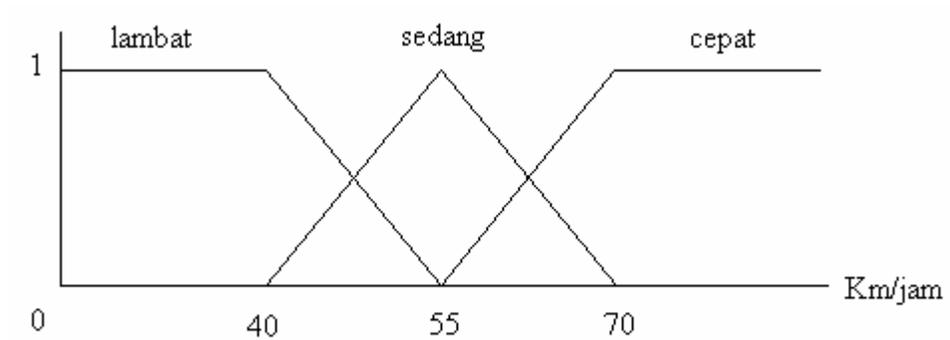
Maka M untuk setiap X adalah :

$M(\text{lambat})$ = Fuzzy setnya “kecepatan dibawah 40 km/jam” dengan fungsi keanggotaan lambat.

$M(\text{sedang})$ = Fuzzy setnya “kecepatan mendekati 55 km/jam” dengan fungsi keanggotaan sedang.

$M(\text{cepat})$ = Fuzzy setnya „kecepatan diatas 70 km/jam“ dengan fungsi keanggotaan cepat.

Gambar grafik fungsi keanggotaannya sebagai berikut :



Gambar 2.7 Grafik fungsi keanggotaan kecepatan

2.2.5 Diagram Alir *Fuzzy Set*

Dalam teori *Fuzzy Set* untuk mendapatkan solusi yang eksak, maka ada tiga langkah umum yang dapat dilakukan :

- Fuzzifikasi (*fuzzification*)

Fuzzifikasi adalah fase pertama dari perhitungan fuzzy yaitu perubahan nilai tegas ke nilai fuzzy. Proses fuzzifikasi dapat dinyatakan sebagai berikut

:

$$x = \text{fuzzifier}(x_0)$$

dengan x_0 adalah sebuah vektor nilai tegas dari satu variabel input, x adalah vektor *fuzzy set* yang didefinisikan sebagai variabel, dan *fuzzifier* adalah sebuah operator fuzzifikasi yang mengubah nilai tegas ke *fuzzy set*.

- Penalaran/Evaluasi Kaidah (*rule evaluation*)

Aturan pada suatu model fuzzy menunjukkan bagaimana suatu sistem beroperasi. Secara umum aturan dituliskan sebagai :

$$IF (X_1 \text{ is } A_1) * (X_2 \text{ is } A_2) * (X_3 \text{ is } A_3) * \dots * (X_n \text{ is } A_n) THEN Y \text{ is } B$$

Dengan * adalah operator (misal : OR atau AND), X_i adalah skalar dan A_i adalah variabel linguistik. Apabila sistem tidak menggunakan hedge, maka variabel linguistiknya sama dengan himpunan fuzzy.

Untuk menulis aturan perlu diperhatikan hal-hal berikut ini :

- Kelompokkan semua aturan yang memiliki solusi pada variabel yang sama.
- Urutkan aturan sehingga mudah dibaca.
- Gunakan identitas untuk memperlihatkan struktur aturan.
- Gunakan penamaan yang umum untuk mengidentifikasi variabel-variabel pada kelas yang berbeda.
- Gunakan komentar untuk mendeskripsikan tujuan dari suatu atau sekelompok aturan.
- Berikan spasi antar aturan.
- Tulis variabel dengan huruf besar-kecil, himpunan fuzzy dengan huruf besar, dan elemen-elemen bahasa lainnya dengan huruf kecil.
- Defuzzifikasi (*Defuzzification*)

Input dari proses defuzzifikasi adalah suatu fuzzy set yang diperoleh dari komposisi aturan – aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain fuzzy set tersebut. Sehingga jika

diberikan suatu fuzzy set dalam range tertentu, maka harus dapat diambil suatu nilai crisp tertentu sebagai output.

Pengendali logika fuzzy harus mengubah variabel keluaran kabur menjadi nilai-nilai tegas yang dapat digunakan untuk mengendalikan sistem. Proses ini disebut penegasan (*Defuzzification*). Telah dikembangkan banyak metode untuk melakukan penegasan ini, diantaranya adalah :

1. Metode *Centroid (CompositeMoment)*

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah fuzzy, dirumuskan:

$$X \leftarrow \frac{\sum_{i=0}^n di * \mu A(di)}{\sum_{i=0}^n \mu A(di)}$$

Dengan di adalah nilai domain ke- i dan $\mu(di)$ adalah nilai keanggotaan titik tersebut.

2. Metode Bisektor

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

$$X = \text{disdh} \mu(di) = \frac{1}{2} \sum_{i=0}^n \mu A(di)$$

3. Metode *Mean Of Maximum* (MOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum, dirumuskan sebagai berikut:

$$X = \text{mean} \{ di | \mu(di) = \text{maximum} \mu A \}$$

4. Metode *Largest Of Maximum* (LOM)

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

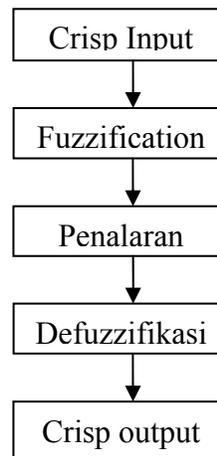
$$X = \max \{ \text{abs}(di) | \mu(di) = \text{maximum} \mu A \}$$

5. Metode *Smallest Of Maximum* (SOM)

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

$$X = \min \{ \text{abs}(d_i) \mid \mu(d_i) = \text{maximum} \mu A \}$$

Diagram alir proses pengaturan dalam teori fuzzy set dapat digambarkan sebagai berikut:



Gambar 2.8 Diagram alir proses pengaturan himpunan fuzzy

2.3 Pemrosesan Citra (*Image Processing*)

Pemrosesan citra atau *image processing* adalah sebuah teknik dimana data dari sebuah citra didigitisasi dan beberapa operasi matematika diaplikasikan ke dalam data, biasanya dengan komputer digital, untuk membuat sebuah citra yang lebih berguna atau menyenangkan pada mata manusia, atau untuk mepresentasikan beberapa dari tugas interpretasi dan rekonisi yang biasanya dilakukan oleh manusia.

Lebih luasnya lagi, *image processing* adalah bentuk dari *information processing* dimana kedua *input* dan *output* berupa citra, seperti foto atau frame dari video.

Kebanyakan teknik *image processing* memperlakukan citra sebagai signal dua dimensi dan mengaplikasikan teknik *signal processing* yang standard kedalamnya.

2.3.1 Definisi Citra

Sebuah citra adalah kumpulan piksel-piksel yang disusun dalam larik dua dimensi. Piksel adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Indeks baris dan kolom (x,y) dari sebuah piksel dinyatakan dalam bilangan bulat. Piksel (0,0) terletak pada sudut kiri atas pada citra, indeks x bergerak ke kanan dan indeks y bergerak ke bawah. Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer.



Gambar 2.9 Perbedaan letak titik origin pada kordinat grafik dan pada citra

(a) koordinat pada grafik matematika (b) koordinat pada citra

Pengertian citra berdasarkan sifatnya dapat dikategorikan menjadi dua , yaitu citra analog dan citra digital.

2.3.1.1 Citra Analog

Citra Analog merupakan kesan visual yang dihasilkan oleh sistem saraf penglihatan manusia akibat stimulasi cahaya. Cahaya merupakan gelombang elektromagnetik yang berada didalam jangkauan 350nm sampai 780nm dari spektrum elektromagnetik dan diekspresikan sebagai distribusi energi $L(\lambda)$. Cahaya yang diterima oleh sebuah objek disebut iluminasi dan dapat ditulis sebagai:

$$I(\lambda) = p(\lambda)L(\lambda)$$

Dimana $p(\lambda)$ adalah tingkat refleksitas objek, $L(\lambda)$ adalah distribusi energi cahaya dan $I(\lambda)$ adalah iluminasi yang dihasilkan. Iluminasi yang dihasilkan akan ditangkap oleh sensor visual mata yang disebut batang dan kerucut. sensor batang berjumlah sekitar seratus juta dan bersifat peka terhadap nilai iluminasi yang rendah (gelap), sedangkan sensor kerucut berjumlah sekitar 6,5 juta, peka terhadap nilai iluminasi yang tinggi (terang) dan warna (Jain, 1989, P49-P50).

2.3.1.2 Citra Digital

Berdasarkan Foley (1995, P816) citra adalah sebuah kumpulan nilai-nilai yang tersusun secara dua dimensi, yaitu tersusun atas baris dan kolom. Setiap nilai dari kumpulan ini disebut elemen citra (*picture elemen* atau piksel). Pada umumnya nilai-nilai ini merupakan bilangan bulat dan memiliki jangkauan nilai minimal maupun maksimal, serta memiliki tingkat intensitas salah satu komponen warna pada model warna tertentu (misalnya model warna merah, hijau, dan biru). Dimensi dari susunan

nilai ini disebut lebar dan tinggi dari citra sedangkan banyaknya bit (*binary digit*) untuk menyatakan sebuah nilai disebut kedalaman citra.

Citra digital dihasilkan melalui proses digitalisasi, yang terdiri atas proses *sampling* dan *quantizing*. Proses *sampling* adalah proses penangkapan gelombang cahaya oleh serangkaian sensor yang peka terhadap gelombang elektromagnetik cahaya tertentu, memiliki ukuran tertentu, dan disusun atas baris dan kolom, ukuran dan bentuk sensor, serta banyaknya sensor secara horizontal dan vertical menentukan resolusi citra yang dihasilkan. Sensor-sensor ini akan menghasilkan tegangan-tegangan listrik yang sebanding dengan besar intensitas cahaya yang diterima, yang kemudian akan dikirimkan ke *quantizer* untuk ditentukan nilai-nilainya. Nilai-nilai inilah yang disebut dengan piksel.

2.3.1.3 Citra Biner

Citra Biner (*Binary Image*) adalah citra digital yang piksel-pikselya hanya terdiri dari dua nilai saja yaitu 0 dan 1

2.3.2 Piksel

Gambar yang bertipe *bitmap* tersusun dari piksel-piksel. Piksel disebut juga dengan *dot*. Piksel berbentuk bujur sangkar dengan ukuran relatif kecil yang merupakan penyusun atau pembentuk gambar *bitmap*. Banyaknya piksel tiap satuan luas tergantung pada resolusi yang digunakan. Keanekaragaman warna piksel tergantung pada *bit depth* yang dipakai. Semakin banyak jumlah piksel tiap satu satuan luas, semakin baik kualitas gambar yang dihasilkan dan tentu akan semakin besar ukuran *file*-nya. Hubungan yang ada antara piksel-piksel, yaitu *neighbours of a*

pixel. Dimisalkan piksel F terletak di koordinat (x,y) , maka yang disebut delapan tetangga (neighbours of a piksel) atau N8 (F) adalah piksel-piksel yang memiliki koordinat:

$$(x+1,y),(x-1,y),(x,y+1),(x,y-1),$$

$$(x+1,y+1), (x-1,y+1),(x+1,y-1),(x-1,y-1)$$

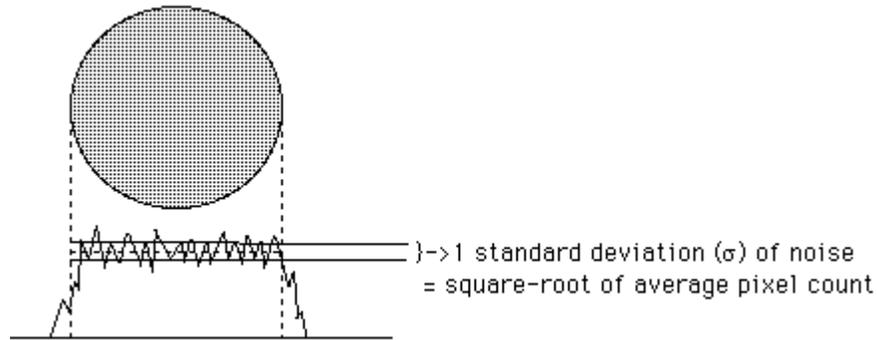
$(x-1,y-1)$	$(x,y-1)$	$(x+1,y-1)$
$(x-1,y)$	(x,y)	$(x+1,y)$
$(x-1,y+1)$	$(x,y+1)$	$(x+1,y+1)$

Gambar: Posisi neighbours of a piksel.

2.3.3 Image Noise

Image noise adalah derajat dari variasi dari nilai piksel yang disebabkan oleh statistika alam dari kerusakan radioaktif dan proses pendeteksi. Walaupun jika kita mendapat sebuah citra dari sumber yang seragam pada gamma kamera yang ideal dengan keseragaman dan efisiensi yang sempurna, angka yang terhitung yang terdeteksi dalam semua piksel dari citra tidak akan sama.

Noise yang terhitung dalam pengobatan nuklir adalah Poisson noise, jadi nilai noise piksel variance sama dengan nilai rata-rata terhitung yang diekspektasikan dalam wilayah yang diberikan dalam citra. Standard deviation adalah akar dari variance.



Gambar 2.10 Standard deviasi

Jika terdapat lesion, maka signal-to-noise ratio (SNR) adalah ukuran dari seberapa jauh (dalam satuan dari standard derivations dari noise) lesion diatas (atau dibawah) rata-rata level background. Sebuah lesion SNR dari sekitar dua akan terdeteksi secara marginal. Sebuah lesion SNR dari sekitar tiga akan terdeteksi secara mudah. Untuk menurunkan persentasi dari image noise, kita harus mendapatkan hitungan sebanyak mungkin:

$$\% \text{ image noise} = \left(\frac{\sigma}{N} \right) \times 100\% = \frac{100}{\sqrt{N}}$$

Limitasinya (seperti biasa) adalah waktu scan dan dosis pasien. Dengan begitu imaging systems yang ideal akan seefisien mungkin.

2.3.4 Pendeteksi Tepi (*Edge Detection*)

Pendeteksi tepi (*Edge detection*) adalah sebuah metode dalam *image processing* untuk mendeteksi suatu sisi (pinggir) dari sebuah citra. *Goal* dari *edge detection* adalah untuk memberi tanda pada poin dalam *digital image* yang dimana intensitas cahayanya berubah secara tajam. Perubahan ketajaman dalam properti citra biasanya merefleksikan kejadian-kejadian yang penting dan perubahan dalam properti dunia.

Ini termasuk (i)diskontuinitas dalam kedalaman, (ii)diskontuinitas dalam orientasi permukaan, (iii)perubahan dalam properti material dan (iv)beberapa variasi dalam skema iluminasi. *Edge detection* adalah sebuah riset dalam area *image processing* dan *computer vision*, dan juga dalam area *feature extraction*.

Edge detection dari sebuah citra berkurang secara signifikan tergantung oleh banyaknya data dan informasi filter yang bisa dianggap sebagai kurang relevan, menyimpan struktur properti yang penting dari sebuah citra. Ada banyak metode untuk melakukan *edge detection*, tapi beberapa bisa dibagi dalam dua kategori, *search-based* dan *zero-crossing based*. Dalam metode *search-based* mendeteksi sisi dengan melihat maxima dan minima dalam derivatif pertama dari citra, biasanya *local directional maxima* dari magnitude gradien. Dalam metode *zero-crossing based* mencari *zero crossing* dalam derivatif kedua dari citra untuk mencari sisi, biasanya *zero crossing* dari *Laplacian* atau *zero crossing* dari ekspresi diferensial yang non-linear.

2.3.5 Image Quality

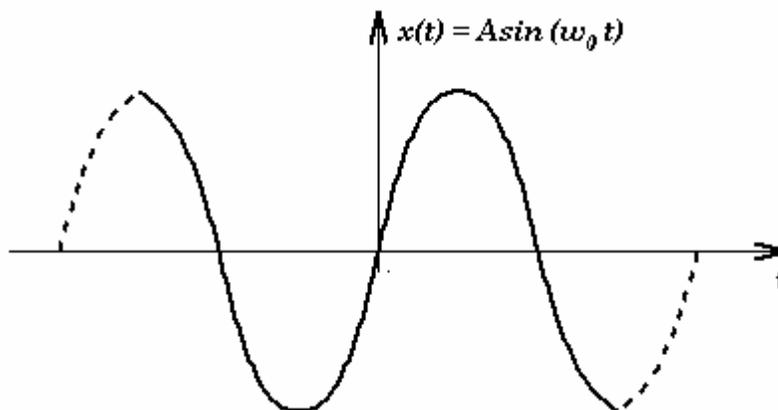
Kualitas citra atau *Image quality* dan *clarity* adalah definisi yang digunakan untuk mengacu kepada derajat dari informasi yang relevan dalam citra yang ditampilkan. Walaupun definisi tersebut tidak begitu jelas, banyak physicist bekerja pada model matematika dari proses pencitraan yang mencoba untuk menggabungkan banyak atribut-atribut citra yang spesifik (sebagai contoh, resolusi, contrast, dan noise) dalam ke sebuah *single image metric* yang yang dimana sangat layak untuk mengerjakan tugas-tugas pencitraan yang spesifik (seperti deteksi, klarifikasi, aktifitas atau kuantisasi ukuran, dll.). Image matriknya sendiri biasanya sejenis

signal-to-noise ratio. Dalam skripsi ini digunakan lima metode untuk mengukur kualitas citra yaitu: Time untuk mengukur kecepatan pemrosesan, FFT, *Index of fuzziness*, MSE, dan PSNR.

2.3.5.1 Fast Fourier Transform (FFT)

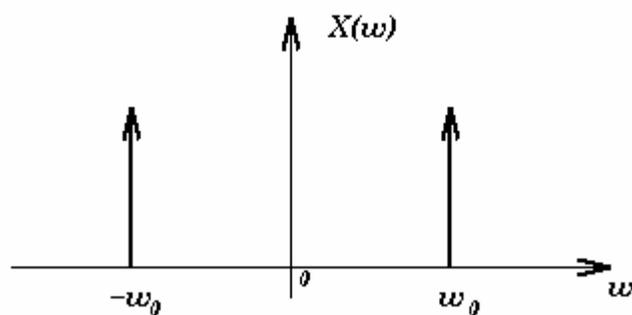
Discrete Fourier Transform (DFT), biasanya juga disebut sebagai *finite Fourier Transform*, merupakan transformasi Fourier yang banyak digunakan dalam pemrosesan sinyal dan bidang-bidang terkait seperti menganalisa frekuensi pada sinyal yang sudah disampling, menyelesaikan persamaan diferensial parsial, dan melakukan operasi konvolusi. Sedangkan Fast Fourier Transform (FFT) merupakan suatu algoritma DFT yang dapat mengurangi sejumlah perhitungan yang diperlukan untuk N nilai, dari $2N^2$ menjadi $2N \log N$, dimana \log merupakan logaritma berbasis-2. Jika fungsi yang ditransformasikan tidak berhubungan dengan frekuensi sampling, tanggapan FFT berbentuk seperti fungsi sinus.

Jika kita memiliki sebuah sinyal $x(t) = A \sin(\omega t)$ dengan $-\infty < t < +\infty$ seperti ditunjukkan pada gambar dibawah ini,



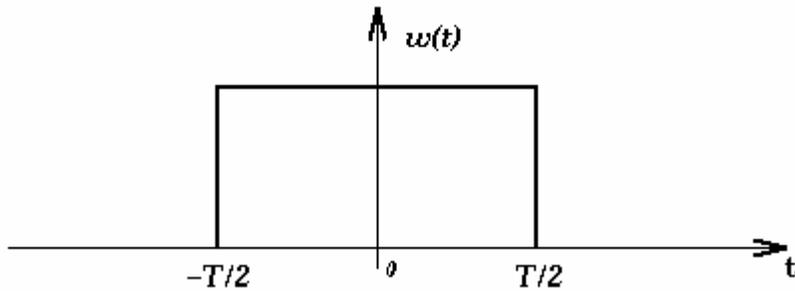
Gambar 2.11 Bentuk Gelombang Sinusoidal

Selanjutnya dari continuous Fourier Transform akan digunakan untuk menemukan frekuensi batas yang dikehendaki. Dalam domain frekuensi ini diperoleh dua pulsa seperti disajikan pada gambar dibawah ini.



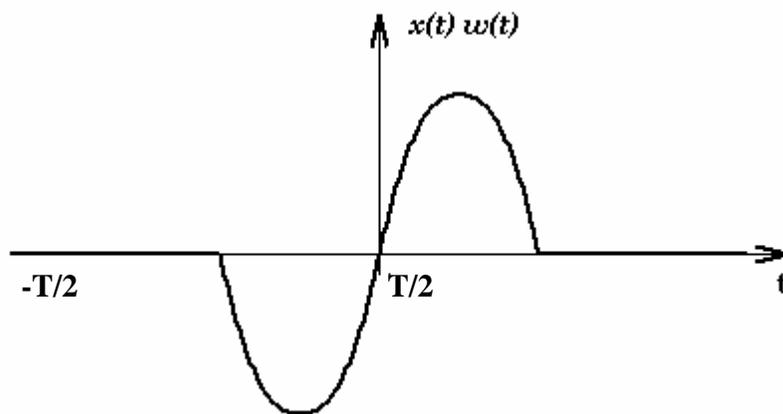
Gambar 2.12 Pulsa dalam domain frekuensi

Kondisi ini benar hanya jika terdapat sinyal yang tanpa batas, tetapi tidak pernah dijumpai gelombang sinyal sinus yang tanpa batas. Kenyataan yang ada adalah sinyal akan ada untuk waktu T tertentu. Oleh karena itu dapat diperoleh dengan mengalikan sinyal tanpa batas dengan fungsi window $w(t)$ sehingga yang ada hanya pada waktu T, seperti disajikan pada gambar dibawah ini.



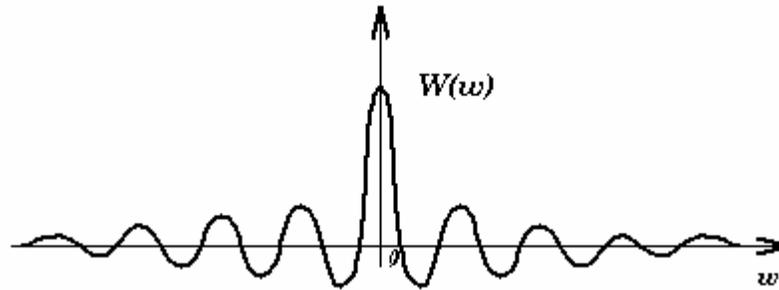
Gambar 2.13 Kondisi sinyal untuk waktu T tertentu

Hasil perkalian $x(t)$ dengan $w(t)$ disajikan pada gambar dibawah ini.



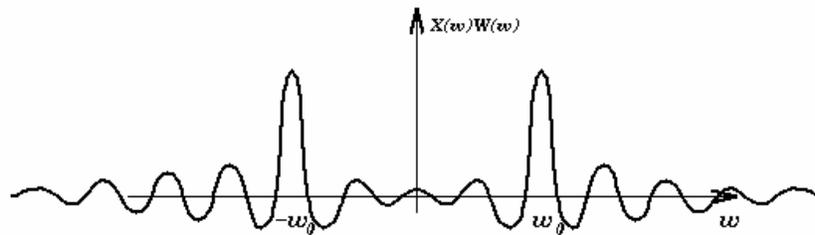
Gambar 2.14 Fungsi sinyal $x(t)w(t)$

Perkalian $x(t)$ dan $w(t)$ sebagai sinyal waktu akan menghasilkan frekuensi baru yang merupakan gulungan frekuensi $x(t)$ dan $w(t)$. Hasil frekuensi dari fungsi window seperti disajikan pada gambar dibawah ini:



Gambar 2.15 Frekuensi fungsi *window*

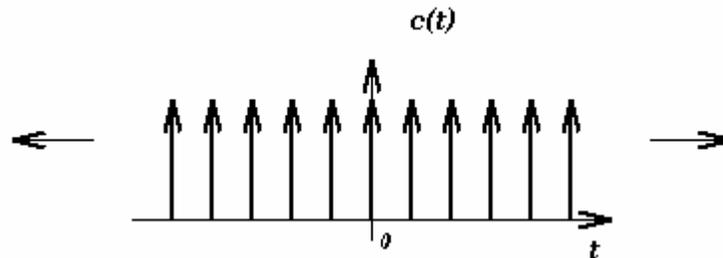
Jika dilakukan konvolusi terhadap sinyal $X(w)$, dihasilkan sinyal berbentuk berikut:



Gambar 2.16 Frekuensi dari gelombang sinyal yang terbatas

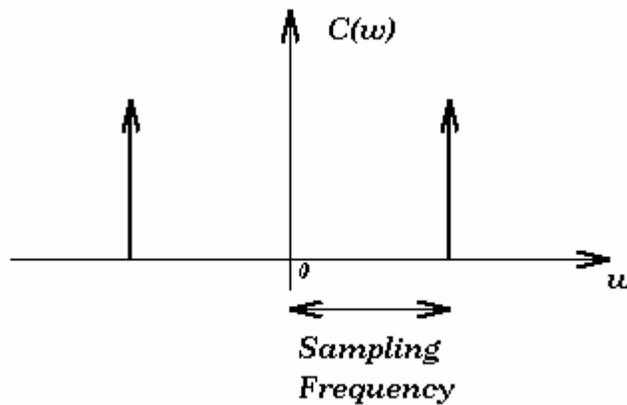
Uraian diatas adalah perkiraan dari frekuensi yang ideal, sementara kondisi demikian jarang dijumpai. Cara untuk memperbaiki resolusi yaitu dengan menaikkan panjang data dan panjang fungsi window. Sejauh ini hanya berhubungan dengan waktu yang kontinyu, untuk mengetahui apa yang terjadi dengan sinyal diskrit diperlukan langkah-langkah sebagai berikut:

- Pertama, Sampling atau pencuplikan data; hal ini dilakukan dengan cara mengalikan sinyal dalam domain waktu $x(t)$ dengan rangkaian pulsa $c(t)$. Fungsi pulsa $c(t)$ disajikan pada gambar di bawah ini:



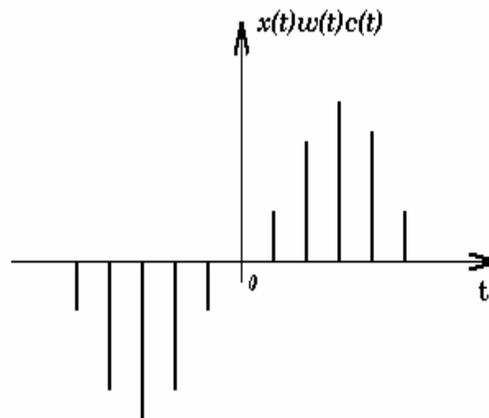
Gambar 2.17 Fungsi Pulsa $c(t)$

Frekuensi tanggapan (frekuensi respon) dari pulsa $c(t)$ disajikan pada gambar dibawah ini:



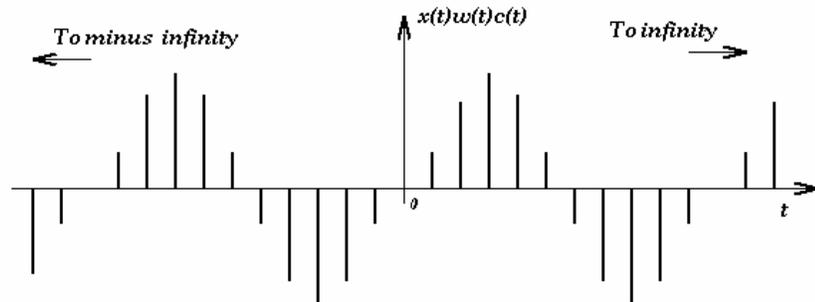
Gambar 2.18 Respon sinyal pulsa dalam domain frekuensi

Sampling sinyal sinus terbatas memberikan sinyal $x(t)*w(t)*c(t)$ yang memberikan frekuensi tanggapan $X(w)*W(w)*C(w)$, seperti disajikan pada gambar berikut ini:



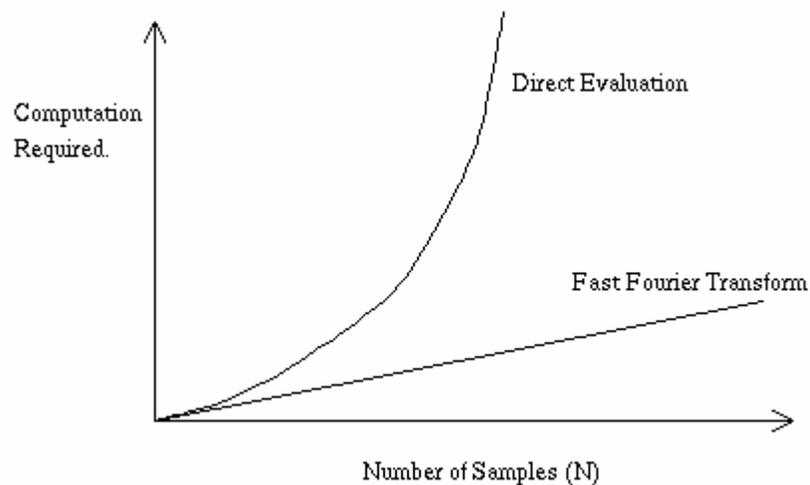
Gambar 2.19 Sinyal tersampling $x(t)w(t)c(t)$

- Kedua, mengalikan dengan rangkaian pulsa. Jika dilihat frekuensi tanggapan dari sinyal sampel, frekuensi sampling sudah periodik. Namun masih ada masalah, yaitu frekuensi tanggapan masih kontinyu walaupun sinyal waktu asli diskrit. Oleh karena itu diperlukan frekuensi tanggapan untuk penggunaan komputer. Hal ini diperoleh dengan cara mengalikan frekuensi domain dengan rangkaian pulsa, seperti disajikan pada Gambar 2.15.
- Ketiga, mengalikan dengan frekuensi domain, dimana diperlukan konvolusi rangkaian waktu diskrit dengan rangkaian pulsa pada domain waktu. Apa yang terjadi, rangkaian sampel menjadi periodik panjang dari rangkaian input. Jika input sinyal berupa input seperti Gambar 3.18, maka rangkaian sampel diskrit untuk DFT disajikan pada Gambar 3.19



Gambar 2.20 Rangkaian sinyal DFT

Pada Discrete Fourier Transform terlihat sangat kompleks dan sangat sulit untuk dilakukan karena meliputi banyak penjumlahan dan perkalian. Sebagai contoh untuk sinyal tersampling yang sederhana, akan diperlukan 49 perkalian kompleks dan 56 penjumlahan. Dengan DFT. Jika memiliki 1024 sample, maka diperlukan kompleksitas penjumlahan dan perkalian yang sangat tinggi, sehingga jumlah perhitungan memerlukan peningkatan yang tidak terkendali lagi.



Gambar 2.21 Perbandingan jumlah sampling dengan kompleksitas perhitungan

Untuk mengurangi kesulitan tersebut, digunakan Fast Fourier Transform yang merupakan metode sederhana pada perhitungan, sehingga bisa lebih cepat untuk nilai besar N, dengan N adalah jumlah sample. Ide dasar dari FFT adalah menggunakan metode divide and conquer, yakni dengan cara memecahkan N nilai sampel menjadi dua (N/2) rangkaian. Hal ini membuat rangkaian permasalahan lebih sederhana dan lebih mudah dicari solusinya dibandingkan dengan satu permasalahan yang besar. DFT memerlukan $(N-1)^2$ kompleksitas perkalian dan $N(N-1)$ kompleksitas penjumlahan. Dengan metode FFT suatu permasalahan dipecah menjadi dua kelompok nilai sampel yang hanya memerlukan satu perkalian dan dua pertambahan serta kombinasi ulang yang nilainya minimal.

2.3.5.2 Index Of Fuzziness

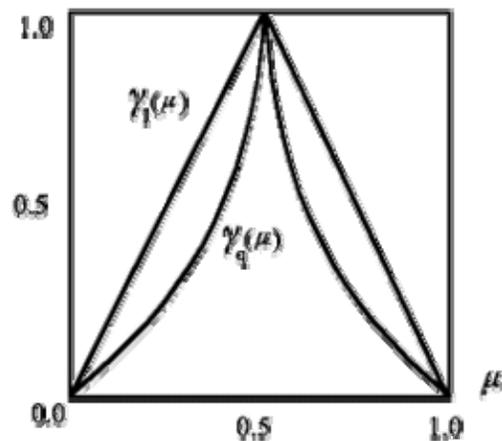
Index of fuzziness mengkalkulasikan ke-fuzzy-an (*fuzziness*) dari citra yang berukuran MxN, tergantung kepada perbedaan pada nilai keanggotaan masing-masing piksel. Formula berikut diperkenalkan oleh Arnold Kaufmann:

$$\gamma_1 = \frac{2}{MN} \sum_m \sum_n \min(\mu_{mn}, 1 - \mu_{mn})$$

Sedangkan rumus kuadratik dari *index of fuzziness* bisa didefinisikan sebagai berikut:

$$\gamma_q = \frac{2}{\sqrt{MN}} \left[\sum_m \sum_n \{ \min(\mu_{mn}, 1 - \mu_{mn}) \}^2 \right]^{\frac{1}{2}} .$$

Jumlah ke-fuzzy-an sama dengan nol (0) jika semua nilai keanggotaannya adalah 0 atau 1. Nilai ke-fuzzy-an akan mencapai nilai maksimum jika semua nilai keanggotaannya sama dengan 0.5 (Gambar 2.21).



Gambar 2.22 Linear and quadratic index of fuzziness. (Adapted from: Tizhoosh, *Fuzzy Image Processing*, ©CopyRight Springer 1997)

2.3.5.3 Mean Squared Error

Mean squared error atau disingkat MSE dari sebuah estimator adalah nilai yang diharapkan (*expected value*) yang dihasilkan dari akar nilai "error" tersebut. Error yang dimaksud disini adalah jumlah dari perbedaan antara estimator dan kuantitas

yang diestimasi. Perbedaan terjadi karena keacakan atau karena estimator tersebut tidak punya informasi untuk bisa menghasilkan estimasi yang lebih akurat lagi. Berikut adalah rumus MSE yang digunakan untuk mengukur kualitas sebuah citra:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I(i, j) - K(i, j)||^2$$

2.3.5.4 Peak Signal-to-Noise Ratio

Peak signal-to-noise ratio (PSNR) adalah rasio antara kekuatan maksimum dari sebuah sinyal dan kekuatan dari noise yang mempengaruhi citra. Dikarenakan banyak sinyal yang mempunyai jarak dinamik (*dynamic range*) yang sangat lebar. PSNR biasanya diekspresikan dalam skala *logarithmic decibel*. PSNR banyak digunakan sebagai ukuran dari kualitas dari rekonstruksi dalam pemrosesan citra, dan didefinisikan sangat mudah melalui rumus MSE yang ada pada bagian 2.3.5. PSNR-nya sendiri didefinisikan sebagai berikut:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

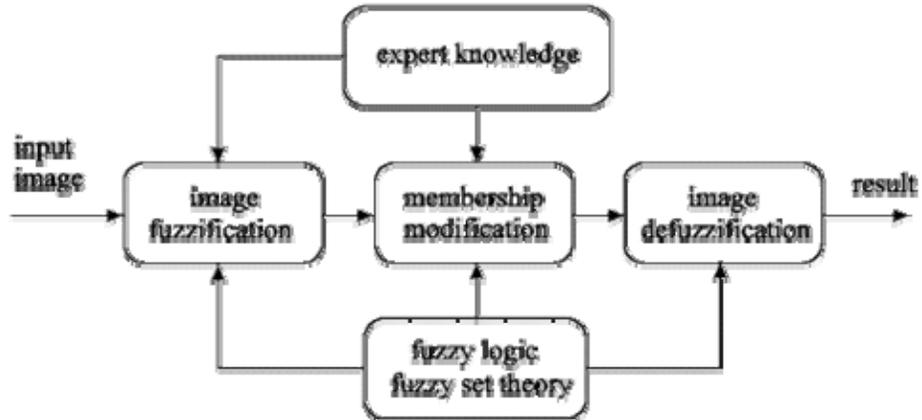
2.3.6 Image Enhancement

Adalah proses dimana sebuah citra dimanipulasikan untuk meningkatkan banyaknya informasi yang ditampilkan secara visual. Dalam *computer graphics*, proses untuk meningkatkan kualitas dari citra yang disimpan secara digital dengan memanipulasikan citra dengan software. Sebagai contoh, untuk membuat sebuah citra menjadi lebih terang atau lebih gelap, atau untuk menaikkan atau menurunkan kontras gambar. *Software image enhancement* yang canggih juga mensupport banyak filter untuk merubah citra dalam banyak cara yang bervariasi. Program yang ahli untuk mengerjakan *image enhancement* biasanya disebut *image editor*

2.4 Fuzzy Image Processing

Fuzzy image processing bukanlah sebuah teori yang unik. *Fuzzy image processing* merupakan sebuah koleksi dari pendekatan fuzzy yang berbeda pada *image processing*. Walaupun demikian bisa didefinisikan seperti berikut ini:

Fuzzy image processing adalah sebuah koleksi dari semua pendekatan yang mengerti, merepresentasikan dan memproses citra, segmennya dan fituranya sebagai himpunan fuzzy. Representasi dan pemrosesannya tergantung pada teknik fuzzy yang telah dipilih dan pada masalah yang untuk dipecahkan. (Tizhoosh, *Fuzzy Image Processing*, Springer, 1997). *Fuzzy image processing* mempunyai tiga langkah utama: fuzzifikasi (*fuzzification*), modifikasi dari nilai keanggotaan (*membership value*), dan jika diperlukan, defusifikasi citra (*image defuzzification*) (Gambar 2.22).



Gambar 2.23 The general structure of fuzzy image processing.