

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian yang dilakukan Hendra (2012) membandingkan hasil analisis menggunakan metode ARIMA (*Autoregressive Moving Average*) yang berdasarkan pada nilai RMSE (*Root Mean Square Error*) dan menggunakan data simulasi yaitu stasioner, stasioner dengan *outlier* dan nonstasioner pada metode ANFIS dan ternyata hasil yang didapat dari penelitian ini lebih bagus menggunakan metode ANFIS.

Hikmah (2007) melakukan pengambilan sampling retina menggunakan citra RGB yang dikonversi ke HSV (*Hue Saturation Value*). Program dirancang menggunakan suatu perangkat lunak analisa matematis dengan memanfaatkan fungsi-fungsi terapan pengolahan citra. Program menerima masukan berupa citra retina dalam format jpg. Metode yang digunakan dalam penelitian tersebut menggunakan metode ANFIS. Penelitian Hani'ah (2015) menggunakan model *Fuzzy Inference System* Takagi Sugeno Orde-1 untuk mengetahui peramalan pemakaian air di PDAM Tirta Moedal Semarang lalu diimplementasikan dengan menggunakan bahasa pemrograman MATLAB.

Fradyudha (2013) penggunaan Metode ANFIS (*Adaptive Neuro Fuzzy Inference System*) untuk program simulasi tanaman kedelai pada pemberian variasi pupuk urea dan pupuk organik menggunakan GroIMP,

masukannya dari sistem data hasil observasi berupa data morfologi tanaman. Data tersebut diantaranya tinggi tanaman, panjang batang, banyak cabang dan banyak daun. Data morfologi tersebut diolah dalam proses pengolahan *input* menggunakan ANFIS (*Adaptive Neuro Fuzzy Inference System*), sedangkan *output* berupa model simulasi morfologi tanaman yang datanya diperoleh dari hasil *output* proses.

Penelitian Mulato (2015) mengaplikasikan metode *fuzzy* untuk identifikasi kematangan buah jambu biji merah. Proses yang dilakukan adalah mengubah tipe gambar jambu biji merah dari tipe RGB ke tipe *grayscale* yang digunakan sebagai data penelitian, lalu dilakukan ekstraksi menggunakan bantuan MATLAB untuk memperoleh informasi dari gambar yaitu *contrast, correlation, energy, homogeneity, mean, variance, standard deviation, skewness, kurtosis, entropy dan inverse difference moment (IDM)*.

Penelitian Opeyemi dan Justice (2012) mengembangkan metode ANFIS untuk prediksi serangan jantung dirancang dengan delapan masukan yang terdiri dari denyut jantung, olahraga, tekanan darah, usia, kolesterol, jenis nyeri dada, darah gula dan seks lalu satu keluaran dengan variabel tingkat resiko. Pasien yang diklasifikasikan menjadi empat bidang yang berbeda yaitu sangat rendah, rendah, tinggi dan sangat tinggi. Pengumpulan data dilakukan dengan cara meng-ekstrak model dari database untuk menghasilkan struktur FIS. Jaringan dilatih dengan set data setelah itu diuji dan divalidasi dengan set data pengujian. Sistem ini dirancang dengan cara bahwa pasien dapat menggunakannya sendiri.

Pasien hanya perlu menyediakan beberapa bobot nilai yang berfungsi sebagai input untuk sistem yang akan diprediksi tingkat resiko pasien.

Penelitian Iraj, dkk (2012) mengklasifikasi siswa luar biasa untuk beasiswa menggunakan Adaptive Neuro Fuzzy. Sistem dibuat dengan metode ANFIS untuk memisahkan mahasiswa berdasarkan faktor pembelajaran. Fungsi keanggotaan yang di gunakan dalam ANFIS yaitu primf dengan jumlah 5 lalu keluaran menggunakan linear dan melakukan 34 *epoch*. Dalam sistem ANFIS memiliki sedikit kesalahan dan dapat digunakan sebagai sistem alternatif yang efektif untuk mengklasifikasikan siswa.

2.2 Landasan Teori

2.2.1 Logika *Fuzzy*

Logika *Fuzzy* adalah peningkatan dari logika boolean yang berhadapan dengan konsep kebenaran sebagian. Saat logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah biner (0 atau 1, hitam atau putih, ya atau tidak), logika *fuzzy* menggantikan kebenaran boolean dengan tingkat kebenaran. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*, pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dalam penalaran dengan logika *fuzzy* tersebut. Logika *fuzzy* dapat dianggap sebagai kotak hitam yang berhubungan antara ruang *input* menuju ruang *output*. Kotak hitam tersebut berisi cara atau metode yang dapat digunakan untuk mengolah

data *input* menjadi *output* dalam bentuk informasi yang baik (Kusumadewi dan Hartati, 2006).

a. Teori Himpunan *Fuzzy*

Pada dasarnya, teori himpunan *fuzzy* merupakan perluasan dari himpunan klasik. Pada teori himpunan klasik (*crisp*), keberadaan suatu elemen pada suatu himpunan (A) hanya akan memiliki dua kemungkinan keanggotaan, yaitu menjadi anggota (A) atau tidak menjadi anggota (A). Suatu nilai yang menunjukkan seberapa besar tingkat keanggotaan suatu elemen (x) dalam suatu himpunan (A), sering dikenal dengan nama nilai keanggotaan atau derajat keanggotaan, dinotasikan dengan $\mu_A(x)$. Himpunan klasik hanya ada dua nilai keanggotaan, yaitu $\mu_A(x) = 1$ untuk x menjadi anggota (A) dan $\mu_A(x) = 0$ untuk (x) bukan dari anggota (A).

Himpunan *fuzzy* (*fuzzy set*) adalah sekumpulan obyek (x) dengan masing-masing obyek memiliki nilai keanggotaan (*membership function*) μ atau disebut juga dengan nilai kebenaran. Jika $Z_{i,t}$ adalah sekumpulan obyek $Z_{i,t} = (Z_{1,t}, Z_{2,t}, \dots, Z_{m,t})$ dan anggotanya dinyatakan dengan (Z) maka himpunan *fuzzy* dari (A) di dalam (Z) adalah himpunan dengan sepasang anggota atau dapat dinyatakan sebagai persamaan (2.1) berikut.

$$F = \{(Z, \mu_F(Z)) \mid Z \in Z_{i,t}\} \quad (2.1)$$

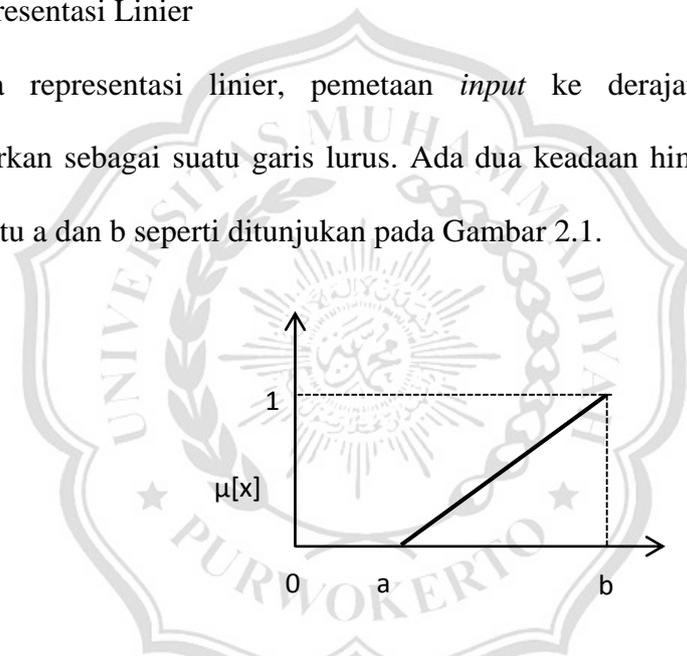
F adalah notasi himpunan *fuzzy*, $\mu_F(x)$ adalah derajat keanggotaan dari (Z) (nilai antara 0 sampai 1) (Kusumadewi dan Hartati, 2006).

b. Fungsi Keanggotaan *Fuzzy*

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan antara lain.

1. Representasi Linier

Pada representasi linier, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Ada dua keadaan himpunan *fuzzy* yang linier yaitu a dan b seperti ditunjukkan pada Gambar 2.1.



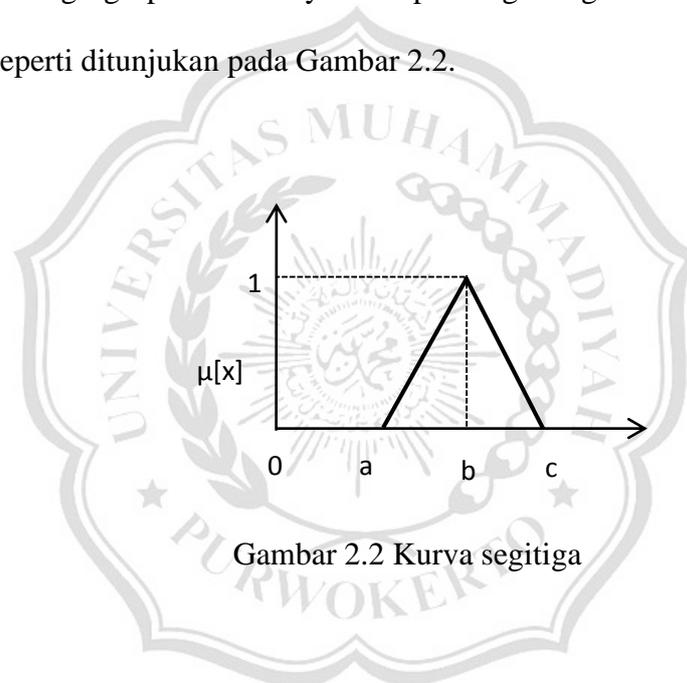
Gambar 2.1 Representasi linier naik

Ada dua keadaan himpunan *fuzzy* yang linier. Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju nilai domain yang memiliki derajat keanggotaan satu [1], fungsi keanggotaannya ditunjukkan pada persamaan (2) sebagai berikut.

$$\mu[x] = \begin{cases} 0; & \text{jika } x \leq a \\ \frac{x-a}{b-a}; & \text{jika } a \leq x \leq b \\ 1; & \text{jika } x \geq b \end{cases} \quad (2.2)$$

2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara dua garis lurus (linear) seperti ditunjukkan pada Gambar 2.2.



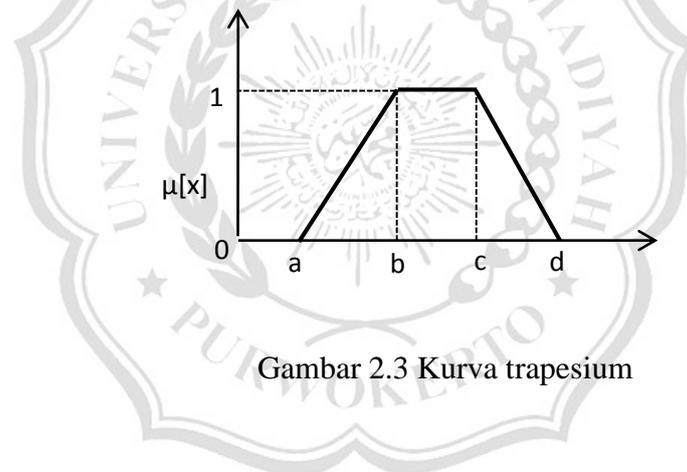
Gambar 2.2 Kurva segitiga

Ada tiga keadaan himpunan *fuzzy* yang linier pada kurva segitiga. Kenaikan himpunan dimulai pada domain a yang memiliki derajat keanggotaan nol [0] bergerak menuju nilai domain b yang memiliki derajat keanggotaan satu [1] lalu bergerak turun menuju domain c yang memiliki derajat keanggotaan nol [0], fungsi keanggotaannya ditunjukkan pada persamaan (3) sebagai berikut.

$$\mu[x] = \begin{cases} 0; & \text{jika } x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; & \text{jika } a \leq x \leq b \\ \frac{b-x}{c-b}; & \text{jika } b \leq x \leq c \end{cases} \quad (2.3)$$

3. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan satu [1] seperti ditunjukkan pada Gambar 2.3.



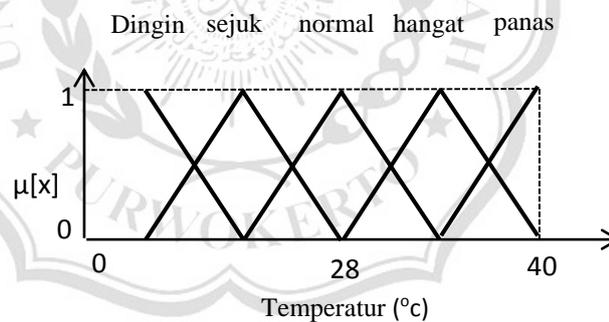
Gambar 2.3 Kurva trapesium

Ada empat keadaan himpunan *fuzzy* pada kurva trapezium. Kenaikan himpunan dimulai pada domain a yang memiliki derajat keanggotaan nol [0] bergerak menuju nilai domain b yang memiliki derajat keanggotaan satu [1] lalu bergerak lagi menuju domain c yang memiliki derajat keanggotaan satu [1] yang terakhir bergerak menuju domain d yang memiliki derajat keanggotaan nol [0], fungsi keanggotaannya ditunjukkan pada persamaan (4) sebagai berikut.

$$\mu[x] = \begin{cases} 0; & \text{jika } x \leq a \text{ atau } x \geq d \\ \frac{x-a}{b-a}; & \text{jika } a \leq x \leq b \\ 1; & \text{jika } b \leq x \leq c \\ \frac{b-x}{c-b}; & \text{jika } c \leq x \leq d \end{cases} \quad (2.4)$$

4. Representasi Kurva Bentuk Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun seperti ditunjukkan pada Gambar 2.4 menunjukkan variabel temperatur dengan daerah bahunya.

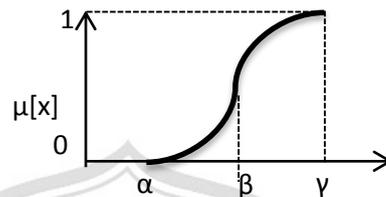


Gambar 2.4 Kurva bahu

Himpunan *fuzzy* bahu bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar.

5. Representasi Kurva S

Kurva S untuk pertumbuhan bergerak dari sisi paling kiri (nilai keanggotaan = 0) ke sisi paling kanan (nilai keanggotaan = 1) seperti ditunjukkan pada Gambar 2.5.



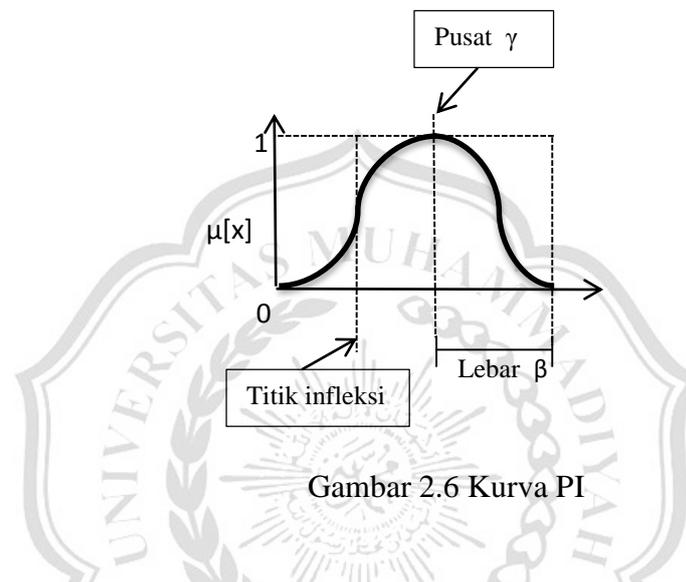
Gambar 2.5 Kurva S pertumbuhan

Kurva S diidentifikasi dengan menggunakan tiga parameter, yaitu nilai keanggotaan nol (α), nilai keanggotaan lengkap (γ), dan titik infleksi atau *crossover* (β) yaitu titik yang memiliki domain benar 50%, fungsi keanggotaannya akan tertumpu pada 50% nilai keanggotaannya.

$$\mu[x, \alpha, \beta, \gamma] = \begin{cases} 0; & \text{jika } x \leq \alpha \\ \frac{2(x-\alpha)}{\gamma-\alpha^2}; & \text{jika } \alpha \leq x \leq \beta \\ \frac{1-2(\gamma-x)}{\gamma-\alpha}; & \text{jika } \beta \leq x \leq \gamma \\ 1; & \text{jika } x \geq \gamma \end{cases} \quad (2.5)$$

6. Representasi Kurva Bentuk Lonceng

Kurva PI berbentuk lonceng dengan derajat keanggotaan satu [1] terletak pada pusat dengan domain (γ), dan lebar kurva (β) seperti ditunjukkan pada Gambar 2.6.



Gambar 2.6 Kurva PI

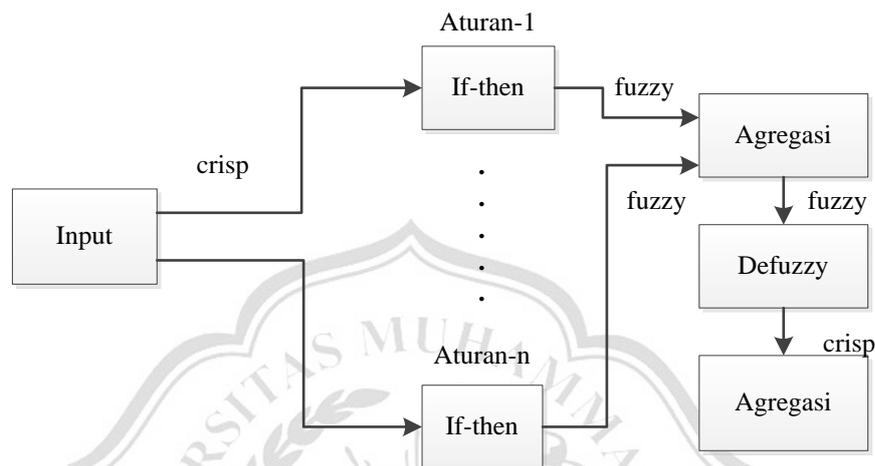
Kurva ini didefinisikan dengan dua parameter, yaitu nilai pada domain yang menunjukkan pusat kurva (γ), dan setengah lebar kurva (β), fungsi keanggotaannya ditunjukkan pada persamaan (6) sebagai berikut.

$$\mu[x, \beta, \gamma] = \begin{cases} s\left(x; \gamma - \beta, \gamma - \frac{\beta}{2}\right); & \text{jika } x \leq \gamma \\ 1 - s\left(x; \gamma, \gamma + \frac{\beta}{2} + \beta\right); & \text{jika } x > \gamma \end{cases} \quad (2.6)$$

2.2.2 FIS (FUZZY INFERENCE SYSTEM)

Kusumadewi (2006) mengatakan bahwa Sistem Inferensi *Fuzzy* (*Fuzzy Inference System* atau FIS) merupakan suatu kerangka komputasi yang

didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk *if-then*, dan penalaran *fuzzy*. Secara garis besar, diagram blok proses *inference fuzzy* seperti ditunjukkan pada Gambar 2.7.



Gambar 2.7 Diagram blok *Fuzzy Inference System*

FIS menerima *input crisp*, *input* ini kemudian dikirim ke basis pengetahuan yang berisi (n) aturan *fuzzy* dalam bentuk *if-then*. *Fire strength* (bobot) akan dicari pada setiap aturan. Apabila jumlah aturan lebih dari satu, maka akan dilakukan agregasi dari semua aturan. Pada hasil agregasi kemudian dilakukan *defuzzy* untuk mendapatkan nilai *crisp* sebagai keluaran sistem.

a. Metode Tsukamoto

Sitem inferensi *fuzzy* didasarkan pada konsep penalaran monoton. Pada metode penalaran secara monoton, nilai *crisp* pada daerah konsekuen dapat diperoleh secara langsung berdasarkan *fire strength* pada antesedennya. Salah satu syarat yang harus dipenuhi pada metode penalaran ini adalah himpunan

fuzzy pada konsekuennya harus bersifat monoton (baik monoton naik maupun monoton turun).

b. Metode Sugeno (TSK)

Sistem inferensi *fuzzy* menggunakan metode sugeno memiliki karakteristik yaitu konsekuen tidak merupakan himpunan *fuzzy*, namun merupakan suatu persamaan linier yang sesuai dengan variabel-variabel masukannya. Ada dua model untuk sistem inferensi *fuzzy* dengan menggunakan metode TSK, yaitu model TSK orde-0 dan model TSK orde-1.

1. Model TSK Orde-0

Secara umum bentuk model *fuzzy* Sugeno orde-0 adalah IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z = k$. A_i adalah himpunan *fuzzy* ke- i sebagai anteseden, \circ adalah operator *fuzzy* (seperti AND atau OR), dan k adalah suatu konstanta (tegas) sebagai konsekuen.

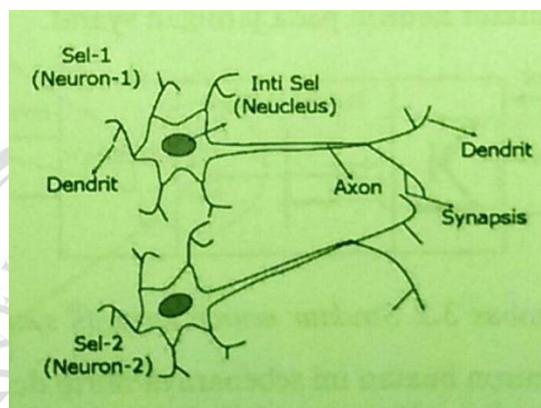
2. Model TSK Orde-1

Secara umum bentuk model *fuzzy* Sugeno orde-1 adalah IF $(x_1 \text{ is } A_1) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z = p_1 * x_1 + \dots + p_N * x_n + q$. A_i adalah himpunan ke- i sebagai anteseden, \circ adalah operator *fuzzy* (seperti AND atau OR), p_i adalah suatu konstanta ke- i dan q juga merupakan konstanta dalam konsekuen (Kusumadewi, 2006).

2.2.3 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (*Artificial Neural Network*) atau disingkat JST adalah sistem komputasi dimana arsitektur dan operasi didasari dari pengetahuan

sel saraf biologis didalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasikan proses pembelajaran pada otak manusia tersebut. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi non-linier, klasifikasi data *cluster* dan regresi non-parametrik atau sebuah simulasi dari koleksi model syaraf biologi (Hermawan, 2006).



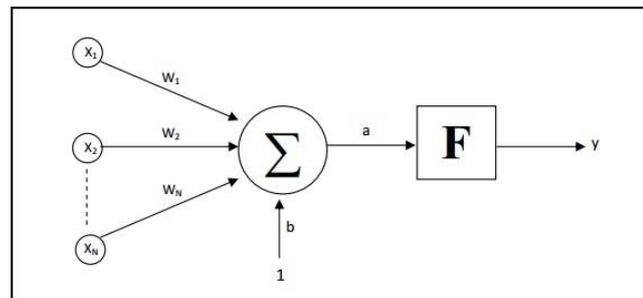
Gambar 2.8 Syaraf secara biologis (Kusumadewi, 2006)

Hubungan antar *neuron* dalam jaringan syaraf mengikuti pola tertentu tergantung pada arsitektur jaringan syarafnya. Pada dasarnya ada tiga macam arsitektur jaringan syaraf, yaitu:

1. Jaringan syaraf dengan lapisan tunggal

Jaringan syaraf tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima *input* kemudian secara langsung mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi, dengan kata lain ciri-ciri dari arsitektur jaringan syaraf dengan lapisan tunggal adalah

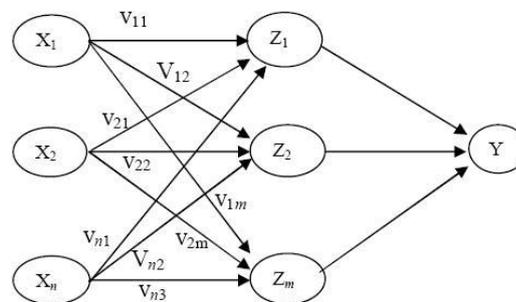
hanya terdiri dari satu lapisan *input* dan satu lapisan *output* tanpa lapisan tersembunyi.



Gambar 2.9 Jaringan syaraf lapisan tunggal

2. Jaringan syaraf dengan banyak lapisan

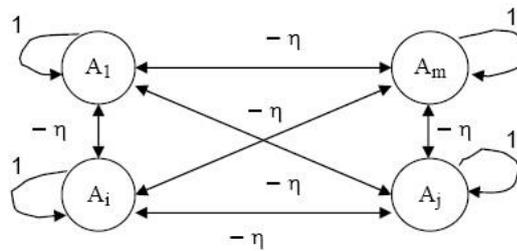
Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan *input* dan lapisan *output* (memiliki satu atau lebih lapisan tersembunyi). Ada lapisan bobot-bobot yang terletak antara dua lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit (Kusumadewi dan Hartati, 2006).



Gambar 2.10 Jaringan syaraf banyak lapisan

3. Jaringan syaraf dengan lapisan kompetitif

Arsitektur ini memiliki bentuk yang berbeda, dimana antara *neuron* dapat saling dihubungkan seperti yang ditunjukkan pada Gambar 2.11.



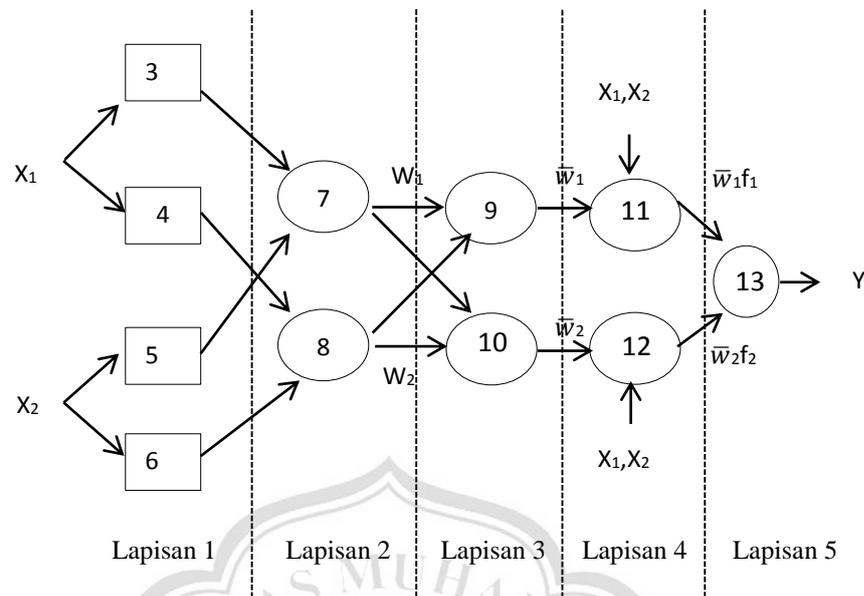
Gambar 2.11 Jaringan syaraf lapisan kompetitif

2.2.4 ANFIS (*Adaptive Neuro Fuzzy Inference System*)

Model *fuzzy* dapat digunakan sebagai pengganti dari persepsi dengan banyak lapisan. Sistem dapat dibagi menjadi dua grup, yaitu satu grup berupa jaringan syaraf dengan bobot-bobot *fuzzy* dan fungsi aktivasi *fuzzy*, dan grup kedua berupa jaringan syaraf dengan *input* yang difuzzykan pada lapisan pertama atau kedua, namun bobot-bobot pada jaringan syaraf tersebut tidak difuzzykan. *Neuro fuzzy* termasuk kelompok yang kedua.

Adaptive Neuro Fuzzy Inference System (ANFIS) adalah arsitektur yang secara fungsional sama dengan *fuzzy rule base* model Sugeno. Arsitektur ANFIS juga sama dengan jaringan syaraf dengan fungsi radial dengan sedikit batasan tertentu (Kusumadewi dan Hartati, 2006).

Jaringan ANFIS terdapat dua *input* x_1 , x_2 dan satu *output* y pada jaringan ANFIS. Jaringan ANFIS terdiri dari lapisan lapisan yang ditunjukkan pada Gambar 2.12.



Gambar 2.12 Jaringan ANFIS

1. Pada lapisan pertama adaptif terhadap parameter suatu fungsi aktivasi *output* dari tiap *neuron* berupa derajat keanggotaan yang diberikan oleh fungsi keanggotaan *input*, yaitu: $\alpha_{A1}(x_1)$, $\alpha_{B1}(x_2)$, $\alpha_{A2}(x_1)$ atau $\alpha_{B2}(x_2)$. Sebagai contoh, misalkan fungsi keanggotaan diberikan sebagai persamaan (7) berikut.

$$\mu(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2.7)$$

Dimana (a, b ,c) adalah parameter-parameter, biasanya b=1. Jika nilai parameter-parameter ini berubah, maka bentuk kurva yang terjadipun akan ikut berubah. Parameter-parameter pada lapisan ini bisa dikenal dengan nama *premise parameters*.

2. Pada lapisan kedua tiap-tiap *neuron* berupa *neuron* yang tetap *outputnya* adalah hasil dari *input*, biasanya digunakan operator AND tiap-tiap *node* merepresentasikan α predikat dari aturan ke i .
3. Tiap-tiap *neuron* pada lapisan ketiga berupa *node* tetap yang merupakan hasil perhitungan rasio dari α predikat (w), dari aturan ke- i terhadap jumlah dari keseluruhan α predikat.

$$\bar{w}_i = \frac{w}{w_1 + w_2}, \text{ dengan } i = 1,2 \quad (2.8)$$

Hasil ini dikenal dengan nama *normalized firing strength*.

4. Tiap-tiap *neuron* pada lapisan keempat merupakan *node* adaptif terhadap satu *output*.

$$\bar{w}_1 y_1 = \bar{w}_1 (c_{i1}x_1 + c_{i2}x_2 + c_{i0}); \text{ dengan } i = 1,2 \quad (2.9)$$

\bar{w}_1 adalah *normalised firing strength* pada lapisan ketiga yaitu parameter-parameter pada neuron tersebut. Parameter tersebut disebut dengan nama *consequent parameters*.

5. Tiap-tiap *neuron* pada lapisan kelima adalah *node* tetap yang merupakan jumlah dari semua masukan (Kusumadewi dan Hartati, 2006).

2.3 Ekstraksi Ciri

Ekstraksi ciri merupakan langkah awal dalam melakukan klasifikasi dan interpretasi citra. Proses ini berkaitan dengan kuantitasi karakteristik citra ke dalam sekelompok nilai ciri yang sesuai. Berdasarkan orde statistiknya, analisis tekstur dapat dikategorikan menjadi tiga, yaitu analisis tekstur orde satu, orde dua dan orde tiga (Putra, 2009).

1. Statistik orde kesatu merupakan metode pengambilan ciri yang didasarkan pada karakteristik histogram citra. Histogram menunjukkan probabilitas kemunculan nilai derajat keabuan piksel pada suatu citra dengan mengabaikan hubungan antar piksel tetangga. Analisis tekstur orde satu lebih baik dalam mempresentasikan tekstur citra dalam parameter-parameter terukur, seperti *mean*, *skewness*, *variance*, *kurtosis*, dan *Entropy*.
2. Statistik orde kedua mempertimbangkan hubungan antara dua piksel (piksel yang bertetangga) pada citra. Kebutuhan analisisnya, analisis tekstur orde dua memerlukan bantuan matriks korekuensi (*matrix co-occurrence*) untuk citra keabuan, biasanya disebut GLCM. Analisis tekstur orde dua lebih baik dalam mempresentasikan tekstur citra dalam parameter-parameter terukur, seperti *contrast*, *correlation*, *energy* dan *homogeneity*.
3. Statistik orde ketiga dan yang lebih tinggi mempertimbangkan hubungan antara tiga atau lebih piksel, hal ini secara teoritis memungkinkan tetapi belum bisa diterapkan.

2.4 Gray Level Co-occurrence Matrix (GLCM)

Gray Level Co-Occurrence Matrix adalah matriks yang dibangun menggunakan histogram tingkat dua. Matriks *co-occurrence* merupakan matriks berukuran $L \times L$ (L menyatakan banyaknya tingkat keabuan) dengan elemen $P(x_1, x_2)$ yang merupakan distribusi probabilitas bersama (*joint probability distribution*) dari pasangan titik-titik dengan tingkat keabuan x_1 yang berlokasi pada koordinat (j, k) dengan x_2 yang berlokasi pada koordinat (m, n) . Koordinat pasangan titik-titik tersebut berjarak r dengan sudut Θ . Histogram tingkat kedua $P(x_1, x_2)$ dihitung dengan pendekatan sebagai berikut.

$$P(x_1, x_2) = \frac{\text{banyaknya pasangan titik-titik dengan tingkat keabuan } x_1 \text{ dan } x_2}{\text{banyak titik pada daerah suatu citra}} \quad (2.10)$$

(Putra, 2009).

2.5 Red Green Blue (RGB)

Model warna RGB (*red green blue*) adalah contoh dari sebuah model warna, yaitu model yang formal untuk mendefinisikan dan menampilkan warna-warna pada monitor komputer dan televisi. Sebenarnya banyak model warna yang sudah dikembangkan oleh para ahli tetapi hanya dua macam model yang banyak digunakan secara luas dalam dunia komputer grafik, yaitu model RGB (*red green blue*) dan model CMY(K) (*cyan magenta yellow*). Model warna lain dikembangkan untuk keperluan lain seperti pengemasan gelombang warna untuk dipancarkan oleh stasiun televisi atau untuk identifikasi warna menggunakan tabel warna seperti ditunjukkan pada Tabel 2 memperlihatkan beberapa model warna yang penting beserta deskripsi dari model warna tersebut.

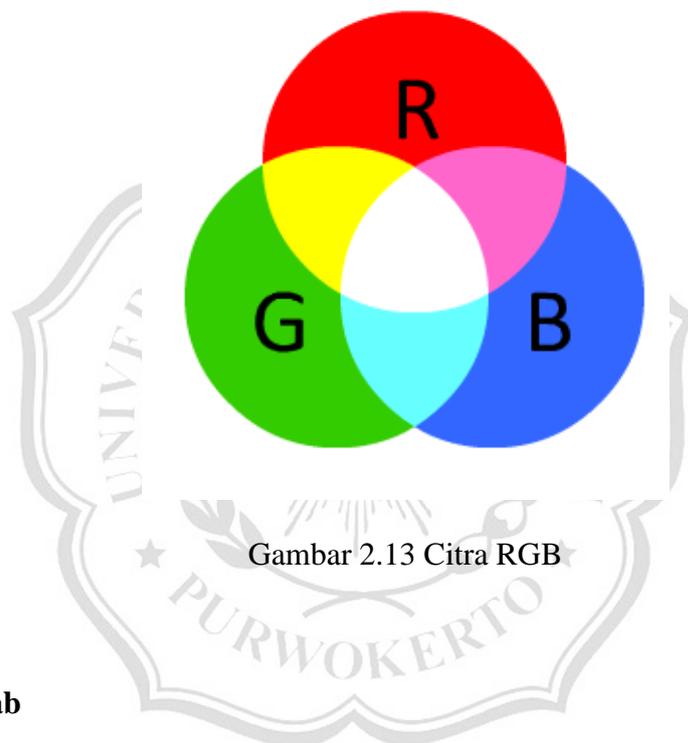
Tabel 2.1 Model warna dan deskripsinya

Model Warna	Deskripsi
RGB	<i>Red, Green</i> dan <i>Blue</i> adalah sebuah model warna pokok adaptif yang digunakan pada sistem display.
CMY(K)	<i>Cyan, Magenta, Yellow</i> (kuning dan hitam). Sebuah model warna subtraktif yang digunakan pada mesin <i>printer</i> .
YCbCr	Luminasi (Y) dan dua komponen kromasiti (Cb dan Cr). Digunakan dalam siaran gelombang televisi.
HIS	Hue, Saturasi, dan Intensitas berdasarkan persepsi manusia terhadap warna.

Berdasarkan cara pembentukan komponen warnanya, model warna dapat dibedakan menjadi dua macam yaitu model warna adaptif dan model warna subtraktif. Model warna adaptif adalah suatu warna dibentuk dengan mengkombinasikan energi cahaya dari ketiga warna pokok dalam berbagai perbandingan. Dominasi warna pokok akan menggiring warna yang dihasilkan ke arah salah satu warna pokok, bila ketiga warna pokok mempunyai kontribusi yang sama kuatnya, warna tidak akan muncul dan sebagai gantinya yang muncul adalah hitam, abu-abu atau putih, tergantung pada kekuatan atau intensitas ketiga warna pokok tadi.

Sebaliknya pada warna subtraktif, pembentukan suatu warna dilakukan dengan menyerap (menghilangkan) beberapa komponen dari cahaya putih. Komponen warna tertentu yang dihilangkan dalam model warna subtraktif ini

tidak akan berkontribusi dalam pembentukan suatu warna. Bila semua komponen warnanya dihilangkan yang muncul adalah hitam, sedangkan bila semua komponen warna dibiarkan memantul yang muncul adalah putih, contoh model warna subtraktif adalah model warna CMY(K), yang digunakan pada mesin pencetak berwarna (Ahmad, 2005).



Gambar 2.13 Citra RGB

2.6 Matlab

MATLAB (*Matrix Laboratory*) adalah sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh *The Math Works*. MATLAB memungkinkan manipulasi matriks, pemplotan fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan pengantarmukaan dengan program dalam bahasa lainnya. Sebuah paket tambahan, *simulink*, menambahkan simulasi grafis multiranah dan desain berdasar model untuk sistem terlekat dan dinamik (Sugiharto, 2006).

2.6.1 M-File Matlab

M-file merupakan editor *text*, latar dasar yang digunakan untuk meletakkan fungsi-fungsi *callback* sebagai pengendali *GUI (Graphical User Interface)*. *Callback* merupakan bagian-bagian kode data yang dibuat pada *M-File*, setiap *callback* akan diimplementasikan sebagai fungsi dalam *M-file*. *M-file* memiliki titik masukan tunggal (*single entry point*) yang dapat menginisialisasi GUI. Kemampuan GUI matlab tergantung pada fungsi yang dibuat pada M.File.

GUI secara otomatis akan memberikan nama sebuah fungsi *callback* dan menambahkannya ke dalam aplikasi *M-file*. Kemudian GUI mengatur nilai *callback property* ke *string* yang menyebabkan fungsi akan tereksekusi ketika kita menjalankan program.

2.6.2 Guide Matlab

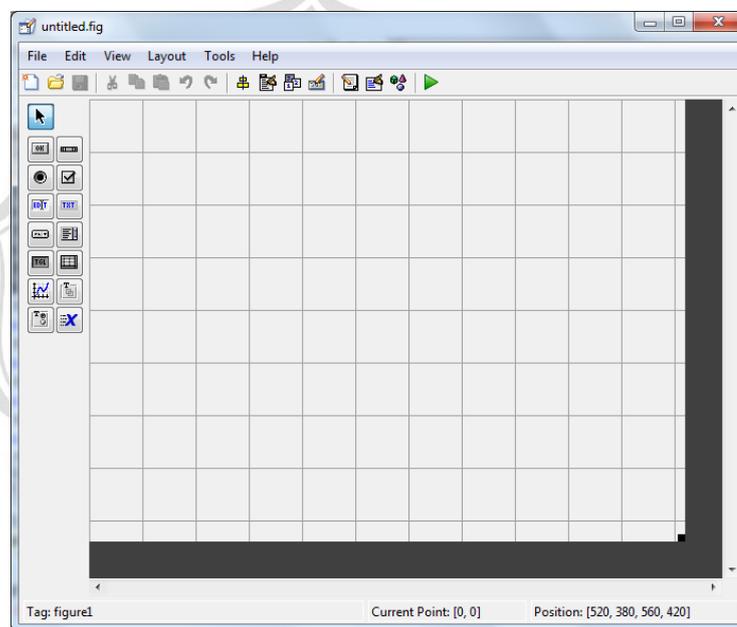
GUIDE atau GUI (*Graphical User Interface*) pada matlab merupakan perangkat antarmuka yang digunakan untuk memudahkan pengguna dalam mengakses program yang dibuat dengan tampilan yang mudah dipahami dan sederhana. *GUIDE* Matlab dapat digunakan untuk membuat aplikasi pengolahan citra digital. Tampilan dalam GUI dapat disesuaikan dengan keinginan. Dalam aplikasinya GUI dapat terdiri dari beberapa komponen *user interface* yang saling berinteraksi, sehingga membentuk sebuah program aplikasi.

2.6.3 Memulai Guide

Memulai GUI matlab, ada 2 cara yang dapat dilakukan, yaitu:

1. Pada *Command Window*, ketikkan *GUIDE*, kemudian tekan *ENTER*
2. Klik *File* pilih *New* kemudian pilih *GUI*

Setelah melakukan salah satu dari cara di atas, maka akan tampil jendela *GUIDE Quick Start*, pilih *Create New GUI*, kemudian di dalam *templates*, pilih *Blank GUI (default)* untuk memulai. Tampilan awal yang muncul pada GUI ditunjukkan pada Gambar 2.14.



Gambar 2.14 Tampilan awal GUI

Matlab mengimplementasikan GUI sebagai sebuah *figure* yang terdapat fasilitas *uicontrol (user interface control)* untuk memprogram masing-masing objek. Hal dasar yang digunakan untuk membuat aplikasi GUI yaitu mengatur *layout* komponen GUI dan memprogram komponen GUI agar dapat bekerja sesuai yang diharapkan.