

BAB II

LANDASAN TEORI

2.1 *Clustering*

“*Clustering* atau analisis *cluster* adalah proses pengelompokan satu set benda- benda fisik atau abstrak ke dalam kelas objek yang sama” (Han, 2006).

Baskoro (2010) menyatakan bahwa :

Clustering atau clusterisasi adalah salah satu alat bantu pada data mining yang bertujuan mengelompokkan obyek-obyek ke dalam *cluster-cluster*. *Cluster* adalah sekelompok atau sekumpulan obyek-obyek data yang similar satu sama lain dalam *cluster* yang sama dan dissimilar terhadap obyek-obyek yang berbeda cluster. Obyek akan dikelompokkan ke dalam satu atau lebih *cluster* sehingga obyek-obyek yang berada dalam satu *cluster* akan mempunyai kesamaan yang tinggi antara satu dengan lainnya. Obyek-obyek dikelompokkan berdasarkan prinsip memaksimalkan kesamaan obyek pada *cluster* yang sama dan memaksimalkan ketidaksamaan pada cluster yang berbeda. Kesamaan obyek biasanya diperoleh dari nilai-nilai atribut yang menjelaskan obyek data, sedangkan obyek-obyek data biasanya direpresentasikan sebagai sebuah titik dalam ruang multidimensi.

Dengan menggunakan clusterisasi, kita dapat mengidentifikasi daerah yang padat, menemukan pola-pola distribusi secara keseluruhan, dan menemukan keterkaitan yang menarik antara atribut-atribut data. Dalam data mining, usaha difokuskan pada metode-metode penemuan untuk *cluster* pada basis data berukuran besar secara efektif dan efisien. Beberapa kebutuhan clusterisasi dalam data mining meliputi skalabilitas, kemampuan untuk menangani tipe atribut yang berbeda, mampu menangani dimensionalitas yang tinggi, menangani data yang mempunyai *noise*, dan dapat diterjemahkan dengan mudah.

2.1.1 *Metode Clustering*

Secara garis besar, terdapat beberapa metode clusterisasi data. Pemilihan metode clusterisasi bergantung pada tipe data dan tujuan clusterisasi itu sendiri.

Metode - metode beserta algoritma yang termasuk didalamnya meliputi (Baskoro, 2010):

a. *Partitioning Method*

Membangun berbagai partisi dan kemudian mengevaluasi partisi tersebut dengan beberapa kriteria, yang termasuk metode ini meliputi algoritma K-Means, K-Medoid, proclus, Clara dan PAM.

b. *Hierarchical Methods*

Membuat suatu penguraian secara hierarkikal dari himpunan data dengan menggunakan beberapa kriteria. Metode ini terdiri atas dua macam, yaitu *Agglomerative* yang menggunakan strategi *bottom-up* dan *Disisive* yang menggunakan strategi top-down. Metode ini meliputi algoritma Birch, Cure, dan Chameleon.

c. *Density-based Methods*

Metode ini berdasarkan konektivitas dan fungsi densitas. Metode ini meliputi algoritma Dbscan, Optic, dan Denclu.

d. *Grid-based Methods*

Metode ini berdasarkan suatu struktur granularitas multi-level. Metode clusterisasi ini meliputi algoritma Sting, WaveCluster, dan Clique.

e. *Model-based Methods*

Suatu model dihipotesakan untuk masing-masing cluster dan ide untuk mencari best fit dari model tersebut untuk masing-masing yang lain. Metode clusterisasi ini meliputi pendekatan statistik, yaitu algoritma COBWEB dan jaringan syaraf tiruan, yaitu SOM.

2.1.2 **Klasifikasi Algoritma Clustering**

“Algoritma *clustering* secara luas diklasifikasikan menjadi dua algoritma, yaitu *hierarchical clustering* dan *non-hierarchical clustering*” (Henjaya, 2010).

Han and Kamber (2006) menyatakan bahwa *hierarchical clustering* adalah sebuah metode hierarkis yang menciptakan komposisi hierarkis yang diterapkan pada objek data, sehingga akan menghasilkan *cluster-cluster* yang bersarang. Algoritma *hierarchical clustering* mengatur seluruh objek dalam sebuah pohon untuk melihat hubungan antara setiap objek (Henjaya, 2010). Contoh algoritma

hierarchical clustering adalah HAC (*Hierarchical Agglomerative Clustering*) (Karhendana, 2008).

Non-hierarchical clustering pada umumnya disebut algoritma *partitional clustering*, memberikan sejumlah n objek dan k yang merupakan jumlah dari cluster yang terbentuk. Algoritma *partitional clustering* mengolah objek ke dalam kelompok berdasarkan kriteria optimasi tertentu, dimana setiap kelompok merupakan representasi sebuah cluster. Han and Kamber (2006) menyatakan bahwa contoh algoritma *partitional clustering* antara lain *Fuzzy C Means*.

2.1.3 Fuzzy Cluster Means (FCM)

Fuzzy C-Means (FCM) adalah suatu teknik pengelompokan data yang mana keberadaan tiap-tiap titik data dalam suatu kelompok (*cluster*) ditentukan oleh derajat keanggotaan. Kelebihan dari *Fuzzy C-Means* adalah dapat melakukan *clustering* lebih dari satu variabel secara sekaligus. Teknik ini pertama kali diperkenalkan oleh Jim Bezdek pada tahun 1981. (kesumadewi, 2010).

Konsep dasar FCM, pertama kali adalah menentukan pusat *cluster*, yang akan menandai lokasi rata-rata untuk tiap-tiap kelompok (*cluster*). Pada kondisi awal, pusat *cluster* ini masih belum akurat. Tiap-tiap titik data memiliki derajat keanggotaan untuk tiap-tiap kelompok (*cluster*). Dengan memperbaiki pusat *cluster* dan derajat keanggotaan tiap-tiap titik data secara berulang, maka akan dapat dilihat bahwa pusat *cluster* akan bergerak menuju lokasi yang tepat. Perulangan ini didasarkan pada minimalisasi fungsi obyektif yang menggambarkan jarak dari titik data yang diberikan ke pusat *cluster* yang terbobot oleh derajat keanggotaan titik data tersebut.

Output dari *Fuzzy C-Means* bukan merupakan *fuzzy inference system*. Namun merupakan deretan pusat *cluster* dan beberapa deret keanggotaan untuk tiap-tiap titik data. Informasi ini dapat digunakan untuk membangun suatu *fuzzy inference system*. FCM menggunakan model pengelompokan *fuzzy* sehingga data dapat menjadi anggota dari semua kelas atau *cluster* tersebut dengan derajat keanggotaan yang berbeda antara 0 hingga 1.

Metode Fuzzy clustering mengijinkan obyek untuk menjadi bagian dari beberapa kelompok secara bersamaan dengan perbedaan level keanggotaan. Sebagai contoh dalam hard clustering, misal himpunan data $Z = \{z_1, z_2, z_3, \dots, z_{10}\}$ di jika dibagi menjadi 2 kelompok maka himpunan U yang merupakan matriks partisi yang menunjukkan level keanggotaan elemen himpunan Z dalam kelompok A_1 atau A_2 akan berwujud sebagai berikut :

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Baris atas matriks partisi U menunjukkan level keanggotaan elemen himpunan Z dalam A_1 dan baris bawah menunjukkan level keanggotaan elemen himpunan Z dalam A_2 . Tampak bahwa setiap elemen himpunan Z secara khusus atau penuh akan menjadi anggota kelompok (A_1 atau A_2) dengan level keanggotaan 1. Dan tidak menjadi anggota dalam suatu kelompok dengan level keanggotaan 0. x_1, x_2, x_3, x_4, x_5 dan x_6 secara khusus merupakan anggota dari A_2 , x_7, x_8, x_9 dan x_{10} secara khusus merupakan anggota dari kelompok A_2 .

Sementara itu dalam fuzzy clustering level keanggotaan data dalam suatu kelompok bukan hanya 0 dan 1 akan tetapi dapat memiliki nilai antara interval $[0,1]$. Baris ke i dalam matriks partisi mengandung level keanggotaan I terhadap A_i . Nilai level keanggotaan dalam setiap kolom matriks partisi yang berarti nilai keanggotaan data dalam setiap kelompok akan selalu berjumlah 1.

Sebagai contoh : himpunan $Z = \{z_1, z_2, z_3, \dots, z_{10}\}$. Dalam pembagian samar jika himpunan Z dibagi menjadi 2 kelompok Z_1 dan Z_2 , maka matriks partisi U dapat dituliskan seperti berikut :

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 & 0,7 & 0,4 & 0,2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,3 & 0,6 & 0,8 & 1 & 1 & 1 \end{bmatrix}$$

Baris atas matriks partisi U menunjukan level keanggotaan elemen himpunan Z dalam A_1 dan baris bawah menunjukkan level keanggotaan elemen himpunan Z dalam A_2 . Dalam hard clustering ini tampak bahwa setiap elemen himpunan Z dapat menjadi anggota kelompok (A_1 atau A_2) dengan level

keanggotaan antara 0 sampai dengan 1. x_1, x_2, x_3, x_4 menjadi anggota A_1 dengan level keanggotaan 1.0 dan menjadi anggota A_2 dengan level keanggotaan 0 sedangkan x_5 menjadi anggota A_1 dengan level keanggotaan 0.7 dan menjadi anggota A_2 dengan level keanggotaan 0.3 dan seterusnya.

2.1.3.1 Algoritma *Fuzzy C Means*

Algoritma *Fuzzy C-Means* (FCM) adalah sebagai berikut (Zimmerman, 1996); (Yan, 1994)

1. Input data yang akan dicluster X , berupa matriks $n \times m$ (n = jumlah sampel data, m = atribut setiap data), X_{ij} = data sampel ke i ($i = 1, 2, \dots, n$).
2. Tentukan:
 - a. Jumlah cluster = c ;
 - b. Pangkat (bobot) = w ;
 - c. Maksimum iterasi = $MaxIter$;
 - d. Error terkecil yang dihadapkan = ;
 - e. Fungsi obyektif awal = $P_0 = 0$;
 - f. Iterasi awal = $t = 1$;
3. Bangkitkan bilangan random μ_{ik} , $i = 1, 2, \dots, n$; $k = 1, 2, \dots, c$; sebagai elemen-elemen matriks partisi awal U .

$$U_0 = \begin{bmatrix} \mu_{11}(x_1) & \mu_{12}(x_2) & \dots & \mu_{1c}(x_c) \\ \vdots & \vdots & & \vdots \\ \mu_{n1}(x_1) & \mu_{n2}(x_2) & \dots & \mu_{nc}(x_c) \end{bmatrix} \quad (2.1)$$

Matriks partisi pada dari *fuzzy clustering* harus memenuhi kondisi sebagai berikut:

$$\mu_{ik} \in [0,1], 1 \leq i \leq n; 1 \leq k \leq c \quad (2.2)$$

$$\sum_{i=1}^n \mu_{ik} = 1; 1 \leq k \leq c \quad (2.3)$$

$$0 < \sum_{k=1}^c \mu_{ik} < c, 1 \leq i \leq n \quad (2.4)$$

4. Hitung pusat *cluster* ke-k: v_{kj} , dengan $k=1, 2, \dots, c$; dan $j=1, 2, \dots, m$.

$$V_{ij} = \frac{\sum_{k=1}^n (\mu_{ik}) \cdot X_{kj}}{\sum_{k=1}^n (\mu_{ik})^w} \quad (2.5)$$

5. Perbaiki derajat keanggotaan tiap data pada setiap *cluster* (perbaiki matriks partisi)

$$V_{ij} = \left[\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(w-1)} \right]^{-1} \quad (2.6)$$

Dengan

$$V_{ij} = d(X_k - V_i) = \left[\sum_{j=1}^c (X_{kj} - V_{ij}) \right]^{1/2} \quad (2.7)$$

6. Hitung fungsi obyektif pada iterasi ke- t , Pt:

$$J(U, V; X) = \sum_{k=1}^n \sum_{j=1}^c (\mu_{ik})^w (d_{ik})^2 \quad (2.8)$$

7. Cek kondisi berhenti

Jika: $(|Pt - Pt-1| < \xi)$ atau $(t > \maxIter)$ maka berhenti; (2.9)

Jika tidak: $t = t + 1$, ulangi langkah ke- 4

2.1.4 Rata-Rata (*Mean*)

$$X = \frac{x_1 + x_2 + x_2 \dots \dots \dots + x_n}{n} = \frac{\sum_{i=1}^n X_i}{n} \quad (2.10)$$

Keterangan:

n = banyak data

x_i = nilai data ke i

2.1.5 Standar Deviasi (*Deviation Standard*)

$$\left(\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}} \quad (2.11)$$

Keterangan:

n = banyak data

x_i = nilai data ke i

\bar{x} = nilai rata-rata data

2.2 Peramalan

Peramalan adalah perhitungan yang objektif dan dengan menggunakan data-data masa lalu, untuk menentukan sesuatu di masa yang akan datang. Untuk melakukan peramalan memerlukan ilmu pengetahuan statistik dan teknologi (Sumayang, 2003).

Peramalan merupakan suatu proses untuk memperkirakan apa yang akan terjadi di masa yang akan datang. Proses peramalan dilakukan dengan metode ilmiah dan secara sistematis. Sifat kualitatif seperti perasaan, pengalaman dan lain-lain merupakan hal penting dalam proses peramalan selain menggunakan prosedur ilmiah atau terorganisir. Jika ingin memperkirakan suatu variable harus diperhatikan dan dipelajari di waktu sebelumnya. Untuk mempelajari bagaimana sejarah perkembangan dari suatu variabel, akan diamati deretan nilai-nilai variabel itu menurut waktu (Ari, 2011)

Metode peramalan merupakan suatu cara melakukan peramalan secara kuantitatif apa yang akan terjadi dimasa yang akan datang berdasarkan data yang relevan pada masa lalu. Metode peramalan sangat besar manfaatnya karena akan membantu dalam mengadakan pendekatan analisis terhadap tingkah laku atau pola yang lalu, sehingga dapat memberikan cara pemikiran, pengerjaan atau pemecahan masalah yang sistematis dan pragmatis, dan memberikan tingkat keyakinan yang lebih besar atau ketepatan hasil dari peramalan yang dibuat.

2.2.1 Teknik Peramalan

Situasi peramalan sangat beragam dalam horizon waktu peramalan, faktor yang menentukan hasil sebenarnya, tipe pola data dan berbagai aspek lainnya. Untuk menghadapi penggunaan yang luas seperti itu, beberapa teknik telah dikembangkan. Teknik tersebut dibagi dalam dua kategori utama, yaitu:

1. Peramalan Kuantitatif

Peramalan kuantitatif adalah peramalan yang didasarkan atas data kuantitatif masa lalu. Hasil peramalan yang dibuat sangat bergantung pada metode yang dipergunakan dalam peramalan tersebut.

2. Peramalan Kualitatif atau teknologi

Peramalan kualitatif adalah peramalan yang didasarkan atas data kualitatif pada masa lalu. Hasil peramalan yang dibuat sangat bergantung pada orang yang menyusunnya. Hal ini penting karena hasil peramalan tersebut ditentukan berdasarkan pemikiran yang intuisi, pendapat dan pengetahuan serta pengalaman penyusunnya.

Menurut Makridakis, Whellwright dan McGee (1995:8) peramalan kuantitatif dapat diterapkan bila terdapat kondisi berikut:

- a. Tersedia informasi (data) tentang masa lalu.
- b. Informasi (data) tersebut dapat dikuantitatifkan dalam bentuk data numerik.
- c. Dapat diasumsikan bahwa beberapa aspek pola masa lalu akan terus berlanjut pada masa yang akan datang.

2.2.2 Metode Peramalan

Ada berbagai metode yang dapat digunakan untuk proses prediksi adalah sebagai berikut:

1. *Grammatical Evolution*

Evolutionary Algorithms (EAs) merupakan algoritma-algoritma komputasi yang berbasis evolusi biologi yang ada di dunia nyata. Salah satu algoritmanya yaitu *Grammatical Evolution* (GE). GE merupakan pengembangan dari *Genetic Programming* (GP). Perbedaannya terletak pada representasi individu. GE menggunakan representasi individu yang dapat digunakan untuk mengevolusi program yang bebas bahasa. (Herrera, 1999).

2. *Fuzzy Time Series (FTS)*

Fuzzy Time series (FTS) adalah metode prediksi data yang menggunakan prinsip-prinsip fuzzy sebagai dasarnya. Sistem prediksi dengan FTS menangkap pola dari data yang telah lalu kemudian digunakan untuk memproyeksikan data yang akan datang. Pertama kali dikembangkan oleh Q. Song and B.S. Chissom pada tahun 1993. Metode ini sering digunakan oleh para peneliti untuk menyelesaikan masalah prediksi. Prosesnya juga tidak membutuhkan suatu sistem pembelajaran dari sistem yang rumit sebagaimana yang ada pada algoritma genetika dan jaringan syaraf sehingga mudah untuk digunakan dan dikembangkan (Robandi, 2006 dikutip dari M. Syauqi Haris, 2010).

3. *Adaptive Neuro-Fuzzy Inference System (ANFIS)*

ANFIS merupakan gabungan dari *Artificial Neural Network (ANN)* dan *Fuzzy Inference Systems (FIS)* yang merupakan jaringan adaptif yang berbasis pada system inference fuzzy (Kusumadewi dan Hartati, 2006).

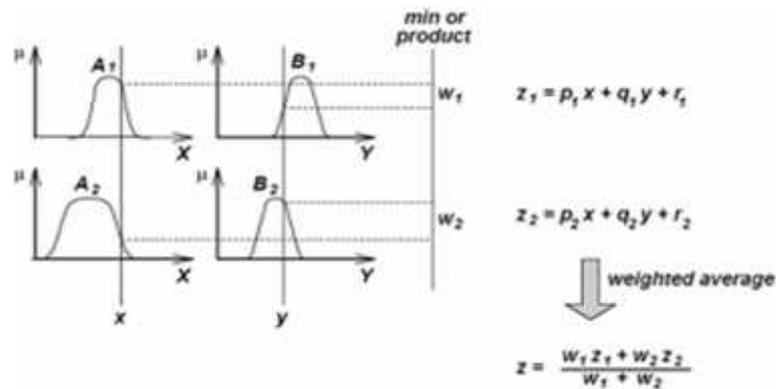
2.2.3 *Adaptive Neuro Fuzzy Inference System (ANFIS)*

Adaptive Neuro Fuzzy Inference System (ANFIS) adalah jaringan yang berbasis pada *system inference fuzzy*. Parameter ANFIS dapat dipisahkan menjadi dua, yaitu parameter premis dan konsekuen yang dapat diadaptasikan dengan pelatihan *hybrid*. Pelatihan *hybrid* dilakukan dalam dua langkah yaitu langkah maju dan mundur. (Widodo, 2005)

ANFIS dapat menggabungkan mekanisme *fuzzy inference system* yang digambarkan dalam arsitektur jaringan syaraf. Jang memperkenalkan arsitektur 2 input 1 output dengan model fuzzy tipe Takagi Sugeno Kang membuat 2 buah *rule*. *Neuro fuzzy* model ANFIS ini terdiri dari 5 *layer* atau lapisan jaringan untuk menerapkan sebuah fuzzy sistem tipe Takagi Sugeno Kang. Sistem ANFIS ini dapat memproses *fuzzy rule* dengan jumlah yang besar.

2.2.3.1 Arsitektur ANFIS

ANFIS adalah penggabungan mekanisme *fuzzy inference system* yang digambarkan dalam arsitektur jaringan syaraf. Sistem inferensi *fuzzy* yang digunakan adalah sistem inferensi fuzzy model Takagi-Sugeno-Kang (TSK) orde satu dengan pertimbangan kesederhanaan dan kemudahan komputasi. Salah satu contoh ilustrasi mekanisme inferensi fuzzy TSK orde satu dengan dua masukan x_1 dan x_2 .



Gambar 2.1 Sistem Inferensi Fuzzy Sugeno Orde Satu Dua Input

Misalkan ada 2 input x_1, x_2 dan satu output z . Ada 2 aturan pada basis aturan model sugeno:

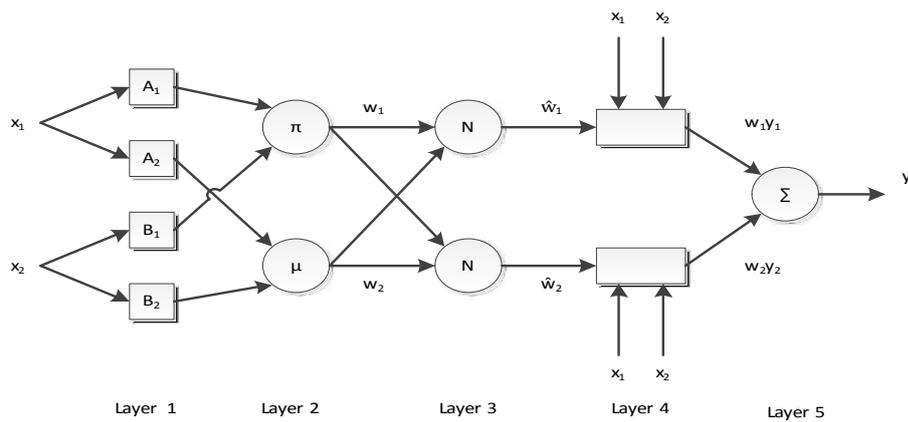
Jika x_1 is A_1 and x_2 is B_1 maka $z_1 = p_1 x_1 + q_1 x_2 + r_1$

Jika x_1 is A_2 and x_2 is B_2 maka $z_2 = p_2 x_1 + q_2 x_2 + r_2$

Jika a predikat untuk aturan kedua adalah w_1 dan w_2 , maka dapat dihitung rata-rata terbobot:

$$y = \frac{w_1 \hat{y}_1 + w_2 \hat{y}_2}{w_1 + w_2} = \hat{w}_1 \hat{y}_1 + \hat{w}_2 \hat{y}_2 \tag{2.12}$$

Adaptive Neuro Fuzzy Inference System (ANFIS) memiliki arsitektur yang dapat terlihat dalam struktur dibawah ini, dengan sistem inferensi fuzzy yang diterapkan adalah inferensi fuzzy model Takagi Sugeno Kang



Gambar 2.2 Struktur ANFIS

Dari gambar diatas dapat diketahui bahwa arsitektur *Adaptive Neuro Fuzzy Inference System (ANFIS)* terdiri atas lima lapisan, dan setiap lapis terdapat node/simpul. Terdapat dua macam node yaitu node adaptif (bersimbol kotak) dan node tetap (bersimbol lingkaran). Fungsi dari setiap lapis adalah sebagai berikut: (Suyanto, 2008)

1). Lapisan 1

Lapisan 1 berfungsi sebagai proses fuzzyfikasi. Output dari node I pada lapisan 1 dinotasikan sebagai $O_{1,i}$. Setiap node 1 pada lapis 1 ini adalah node adaptif (parameter dapat berubah) dengan fungsi node sebagai berikut.

$$O_{1,i} = \mu_{A_i}(x_1) \text{ untuk } i = 1, 2 \text{ dan}$$

$$O_{1,i} = \mu_{B_{i-2}}(x_2) \text{ untuk } i = 3, 4$$

Dari gambar di atas tiap-tiap input tersebut dibagi menjadi 2 fungsi keanggotaan, x_1 dibagi dalam A_1 dan A_2 ; misalnya A_1 menyatakan kecil dan A_2 menyatakan besar. Begitu juga x_2 dibagi dalam fungsi keanggotaan B_1 yang menyatakan kecil dan B_2 yang menyatakan besar. Dari pemetaan tersebut x_1 dan x_2 sudah menjadi variabel fuzzy yang masing-masing mempunyai nilai μ kecil dan besar tertentu. x_1 mempunyai nilai μ_{A_1} dan μ_{A_2} sedangkan x_2 punya nilai μ_{B_1} dan μ_{B_2} .

Fungsi keanggotaan yang digunakan adalah fungsi Generalized-Bell yang dirumuskan:

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \tag{2.13}$$

Di mana $\{a_i, b_i, c_i\}$ adalah himpunan parameter. Parameter pada lapisan ini disebut parameter - parameter premis.

2). Lapisan 2

Semua node pada lapisan ini adalah nonadaptif (parameter tetap). Fungsi node ini adalah mengalikan setiap sinyal masukan yang datang

$$O_{2,i} = w_i = \mu_{A_i}(x_1) \mu_{B_i}(x_2), i = 1, 2 \quad (2.14)$$

3). Lapisan 3

Lapisan Setiap node pada lapisan ini adalah node nonadaptif yang menampilkan fungsi derajat pengaktifan ternormalisasi (*normalized firing strength*) yaitu rasio keluaran node ke-i pada lapisan sebelumnya terhadap seluruh keluaran lapisan sebelumnya, dengan bentuk fungsi node:

$$i = \frac{w_i}{w_1 + w_2} \quad (2.15)$$

4). Lapisan 4

Setiap node pada lapis ini adalah adaptif dengan fungsi node

$$O_{4,i} = i \cdot i = i (p_i x_1 + q_i x_2 + r_i) \quad (2.16)$$

Dengan w_i : kuat penyulutan ternormalisasi dari lapisan 3

$\{p_i, q_i, r_i\}$: himpunan parameter dari node ini. Parameter pada lapis ini disebut parameter konsekuen.

5). Lapisan 5

Pada lapisan ini hanya ada satu node tetap yang fungsinya untuk menjumlahkan semua masukan. Fungsi simpul:

$$O_{5,i} = i \cdot i = \frac{i \cdot i}{i} \quad (2.17)$$

Jaringan adaptif dengan lima lapisan tersebut ekuivalen dengan sistem inferensi fuzzy TSK.

2.2.3.2 Algoritma Pembelajaran Hibrida

Metode pembelajaran hibrida dipergunakan untuk mengatur parameter-parameter ANFIS secara maju (*forward*) ataupun secara mundur (*backward*).

Tabel 2.1 Proses Pembelajaran Hibrida ANFIS

	Arah Maju	Arah Mundur
Parameter Premis	Tetap	<i>Gradient Descent</i>
Parameter Konsekuen	RLSE	Tetap
Sinyal	Keluaran Simpul	Laju Kesalahan

1. Pembelajaran arah maju (*forward pass*) dengan metode *Least Squares Estimator (LSE) Recursive*

Dari arsitektur ANFIS (Gambar 2.2), apabila nilai parameter premis tetap, maka keluaran keseluruhan dapat dinyatakan dengan kombinasi linear dari parameter konsekuen. (Widodo, 2005)

Berikut adalah persamaannya:

$$\begin{aligned}
 y &= \frac{w_1}{w_1 + w_2} y_1 + \frac{w_2}{w_1 + w_2} y_2 \\
 &= \hat{w}_1(p_1x_1 + q_1x_2 + r_1) + \hat{w}_2(p_2x_1 + q_2x_2 + r_2) \\
 &= (\hat{w}_1x_1)p_1 + (\hat{w}_1x_2)q_1 + \hat{w}_1r_1 + (\hat{w}_2x_1)p_2 + (\hat{w}_2x_2)q_2 + \hat{w}_2r_2
 \end{aligned}
 \tag{2.18}$$

Untuk p pasangan data pelatihan x_1, x_2 dan keluaran y , maka diperoleh persamaan linear simultan berikut:

$$\begin{aligned}
 y_1 &= (\hat{w}_1x_1)p_1 + (\hat{w}_1x_2)q_1 + \hat{w}_1r_1 + (\hat{w}_2x_1)p_2 + (\hat{w}_2x_2)q_2 + \hat{w}_2r_2 \\
 y_2 &= (\hat{w}_1x_1)p_1 + (\hat{w}_1x_2)q_1 + \hat{w}_1r_1 + (\hat{w}_2x_1)p_2 + (\hat{w}_2x_2)q_2 + \hat{w}_2r_2 \\
 &\dots = \dots + \dots + \dots + \dots + \dots + \dots \\
 y_n &= (\hat{w}_1x_1)p_1 + (\hat{w}_1x_2)q_1 + \hat{w}_1r_1 + (\hat{w}_2x_1)p_2 + (\hat{w}_2x_2)q_2 + \hat{w}_2r_2
 \end{aligned}
 \tag{2.19}$$

Jika dimisalkan

$$A = [px6] = \begin{bmatrix}
 {}_1x_1 & {}_1x_2 & {}_11 & {}_2x_1 & {}_2x_2 & {}_22 \\
 {}_1x_1 & {}_1x_2 & {}_11 & {}_2x_1 & {}_2x_2 & {}_22 \\
 {}_1x_1 & {}_1x_2 & {}_11 & {}_2x_1 & {}_2x_2 & {}_22 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 {}_1x_1 & {}_1x_2 & {}_11 & {}_2x_1 & {}_2x_2 & {}_22
 \end{bmatrix} \rightarrow \text{data ke } i \tag{2.20}$$

$$\begin{matrix}
 p_1 \\
 q_2
 \end{matrix}
 \left(\begin{matrix} \\ \\ \end{matrix} \right)$$

$$= [6 \times 1] = \begin{pmatrix} r_1 \\ p_2 \\ q_2 \\ r_2 \end{pmatrix}, \quad y = [p \times 1] = \dots \quad (2.21)$$

(y_p)

maka baris ke p dari matriks data $[A:y]$ yang dinotasikan dengan $(a_p^T:y_p)$. Karena hampir semua perhitungan berdasarkan pada matriks A dan vektor y , maka $(a_p^T:y_p)$ disebut pasangan data ke p dari himpunan data pelatihan. Bila $m = n$ (A persegi) dan A nonsingular, maka q dapat diperoleh dari persamaan berikut:

$$\theta = A^{-1}y \quad (2.22)$$

tetapi biasanya $m > n$, berarti pasangan data pelatihan lebih banyak dari parameter fitting. Dalam kasus ini solusi eksak yang memenuhi m persamaan tidak selalu mungkin, karena data bisa terkontaminasi dengan derau, atau model tidak sesuai untuk menggambarkan sistem target. Maka θ merupakan vektor parameter konsekuen dan y adalah vektor keluaran

$$\theta = [p_1 \quad q_1 \quad r_1 \quad p_2 \quad q_2 \quad r_2]^T \quad (2.23)$$

$$y = [y_1 \quad \dots \quad y_n]$$

dalam bentuk persamaan matriks akan diperoleh :

$$A\theta = Y$$

$$\theta^* = (A^T A)^{-1} A^T Y \quad (2.24)$$

Penyelesaian untuk mencari nilai θ terbaik adalah dengan menggunakan metode RLSE untuk meminimumkan nilai $abs(A\theta = Y)$.

Rumus iterasi yang digunakan adalah:

$$\begin{aligned} n_{i+1} &= n_i + P_{i+1} a_{i+1} (y_{i+1}^T - a_{i+1}^T n_i) \\ P_{i+1} &= P_i - \frac{P_i a_{i+1} a_{i+1}^T P_i}{1 + a_{i+1}^T P_i a_{i+1}} \end{aligned} \quad (2.25)$$

dengan a_{i+1}^T adalah vektor baris ke- $i+1$ dari matriks A , y_{i+1} adalah elemen ke- $i+1$ dari vektor y . Syarat awal yang dibutuhkan adalah $n_0 = 0$ dan $P_0 = mI$, dengan I adalah matriks identitas dengan orde sejumlah parameter konsekuen yang diestimasi dan m adalah suatu bilangan yang besar.

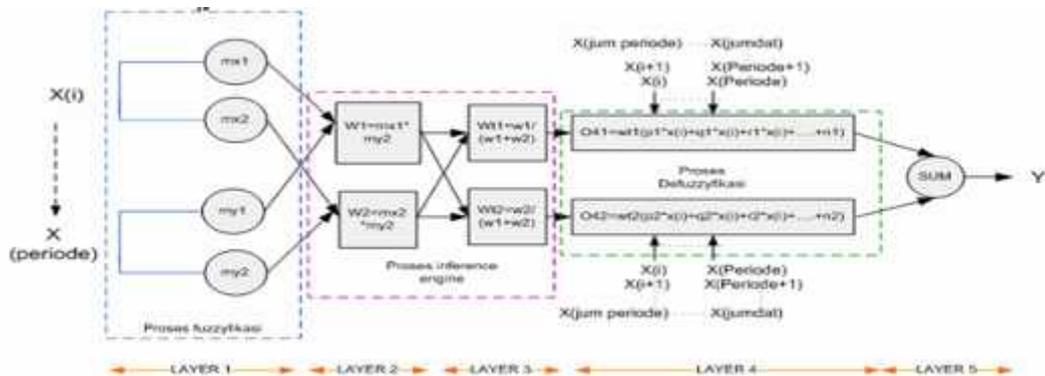
2.3 Peramalan menggunakan *Adaptif Neuro Fuzzy Inference System* (ANFIS)

Pada peramalan dengan metode ANFIS terbagi menjadi 3 proses yaitu: proses Inisialisasi awal, proses pembelajaran (*learning*), dan proses peramalan. Penentuan periode input dan periode training dilakukan saat inisialisasi awal dimana tiap-tiap periode input memiliki pola atau *pattern* yang berbeda.

Data yang digunakan untuk proses pembelajaran (*training*) terdiri dari data input, parameter ANFIS, dan data test yang berada pada periode training ANFIS.

Training dengan ANFIS menggunakan algoritma pembelajaran hibrida, dimana dilakukan penggabungan metode *Least-squares estimator* (LSE) pada alur maju dan *error backpropagation* (EBP) pada alur mundur. Pada algoritma pembelajaran ini nilai parameter premis akan tetap saat alur maju, namun sebaliknya parameter konsekuen akan berubah saat alur maju.

2.3.1 Tahapan Peramalan dengan Metode ANFIS menggunakan Pembelajaran Alur Maju



Gambar 2.3 Blok Diagram Alur Maju ANFIS untuk Proses Peramalan

Pada blok diagram Gambar 2.3 digambarkan mengenai proses alur maju dari sebuah sistem ANFIS yang terdiri dari beberapa layer. Pada layer pertama data input pada masing masing periode akan dilakukan proses *fuzzyfikasi*. Proses ini adalah untuk memetakan inputan data kedalam himpunan *fuzzy* sesuai dengan klasifikasi yang dipilih. Dalam proses ini inputan akan dilakukan perhitungan fungsi keanggotaan *fuzzy* untuk mentransformasi masukan himpunan klasik (*crisp*) ke derajat tertentu. Fungsi keanggotaan yang digunakan adalah jenis gaussian dimana pada fungsi keanggotaan ini terdapat dua parameter yaitu mean dan varian, parameter tersebut dalam metode ANFIS disebut sebagai parameter premis.

Pada layer kedua dan ketiga dilakuakn proses *inference engine* (system *inference fuzzy*) ditentukan rule fuzzy untuk dilakukan proses perhitungan selanjutnya. Pada proses ini digunakan model takagi sugeno. Pada proses ini digunakan dua rule yaitu: jika mx_1 bertemu my_1 maka akan dilanjutkan ke W_1 , dan jika mx_2 bertemu my_2 maka akan dilanjutkan ke W_2 . Nilai W_1 dan W_2 didapat dari hasil pencarian nilai minimum untuk masing – masing input keanggotaan fuzzy.

Pada layer 4 dilakukan proses defuzzyfikasi dilakukan perhitungan untuk mentransformasi hasil *fuzzy* ke bentuk keluaran yang *crisp*. Pada layer ini dilakukan perhitungan LSE untuk mendapatkan nilai parameter konsekuen.

Pada layer 5 dilakukan proses *summary* dari dua output pada layer 4. Pada ANFIS, sistem fuzzy terletak pada layer 1,2,3 dan 4. dimana sistem fuzzy ini adalah sebagai penentu *hidden node* yang terdapat pada sistem *neural network*.

2.4 Ukuran Ketepatan Peramalan

Dalam pemodelan deret berkala, sebagian data yang diketahui dapat digunakan untuk meramalkan data berikutnya. Selisih besaran (ukuran kesalahan peramalan) data peramalan terhadap data aktual yang terjadi merupakan suatu data penting untuk menilai ketepatan suatu metode peramalan. Dengan membandingkan ukuran kesalahan beberapa metode peramalan, akan diperoleh metode mana yang mempunyai ukuran kesalahan terkecil, sehingga nilai peramalan dapat dipakai sebagai acuan dalam menentukan kebutuhan-kebutuhan dimasa yang akan datang.

Menurut Makridakis, Whellwright dan McGee (1995:32,43) terdapat beberapa ukuran kesalahan dalam peramalan antara lain RMSE dan MAPE.

- a. Nilai tengah kesalahan akar kuadrat atau *Root Mean Squared Error* (RMSE).
RMSE adalah rata-rata kuadrat dari perbedaan nilai estimasi dengan nilai observasi suatu variabel. Jika nilai RMSE semakin kecil maka estimasi model atau variabel tersebut semakin valid.
- b. Mean Absolute Deviation (MAD).
Mean Absolute Deviation (MAD) mengukur ketepatan ramalan dengan merata-rata kesalahan dugaan (nilai absolut masing-masing kesalahan). MAD paling berguna ketika orang yang menganalisa ingin mengukur kesalahan ramalan dalam unit yang sama sebagai deret asli.
- c. Nilai tengah kesalahan persentase absolut atau *Mean Absolute Percentage Error* (MAPE).
MAPE merupakan persentase yang dihitung dari nilai absolut kesalahan di masing-masing periode dan dibagi dengan nilai data aktual periode tersebut, kemudian dicari rata-rata kesalahannya.

2.4.1 Mean Absolute Percentage Errors (MAPE)

Persentase kesalahan absolut rata-rata atau *Mean Absolute Percentage Error* (MAPE) dihitung dengan menemukan kesalahan absolut setiap periode, kemudian membaginya dengan nilai observasi pada periode tersebut, dan akhirnya merata-ratakan persentase absolut ini. Pendekatan ini sangat berguna jika ukuran variabel peramalan merupakan faktor penting dalam mengevaluasi akurasi peramalan tersebut. MAPE memberikan petunjuk seberapa besar kesalahan peramalan dibandingkan dengan nilai sebenarnya dari series tersebut. MAPE juga dapat digunakan untuk memperbandingkan akurasi dari tehnik yang sama atau berbeda pada dua series yang berbeda (Arsyad, 2001).

Mean absolute percentage error (MAPE) adalah cara yang paling akurat untuk menghitung *error*, karena menyatakan persentase kesalahan hasil ramalan terhadap keadaan aktual selama periode tertentu yang memberikan informasi persentase terlalu tinggi atau terlalu rendah. Suatu model mempunyai kinerja sangat bagus jika nilai MAPE berada di bawah 10%, dan mempunyai kinerja bagus jika nilai MAPE berada di antara 10% dan 20% (Alda, 2009).

Rumus *Mean absolute percentage error* (MAPE) (Sumayang, 2003).

$$e_t = X_t - WMA_t \quad (2.26)$$

$$\%E_{n+1} = \frac{e}{X} \times 100\% \quad (2.27)$$

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{e_t}{X_t} \right| 100\%}{n} \quad (2.28)$$

Keterangan rumus:

e_t = *Error* untuk periode waktu t

n = banyak data

$X_t \cdot WMA_t$ = *Error* adalah perbedaan hasil ramalan nilai dengan nilai yang sesungguhnya

$MAPE$ = Rata-rata persen *error*