

# Neuro-Fuzzy

Dr. Suyanto, S.T., M.Sc.

HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)  
Informatics faculty – Telkom University

# Perbandingan ANN dan Fuzzy Systems

Kriteria	ANN	<i>Fuzzy Systems</i>
Sangat baik untuk masalah dengan informasi yang kurang presisi dan memiliki kebenaran parsial?	<b>Tidak</b>	Ya
Memiliki kemampuan untuk menjelaskan proses penalaran?	<b>Tidak</b>	Ya
Bisa <i>learning</i> ?	Ya	<b>Tidak</b>

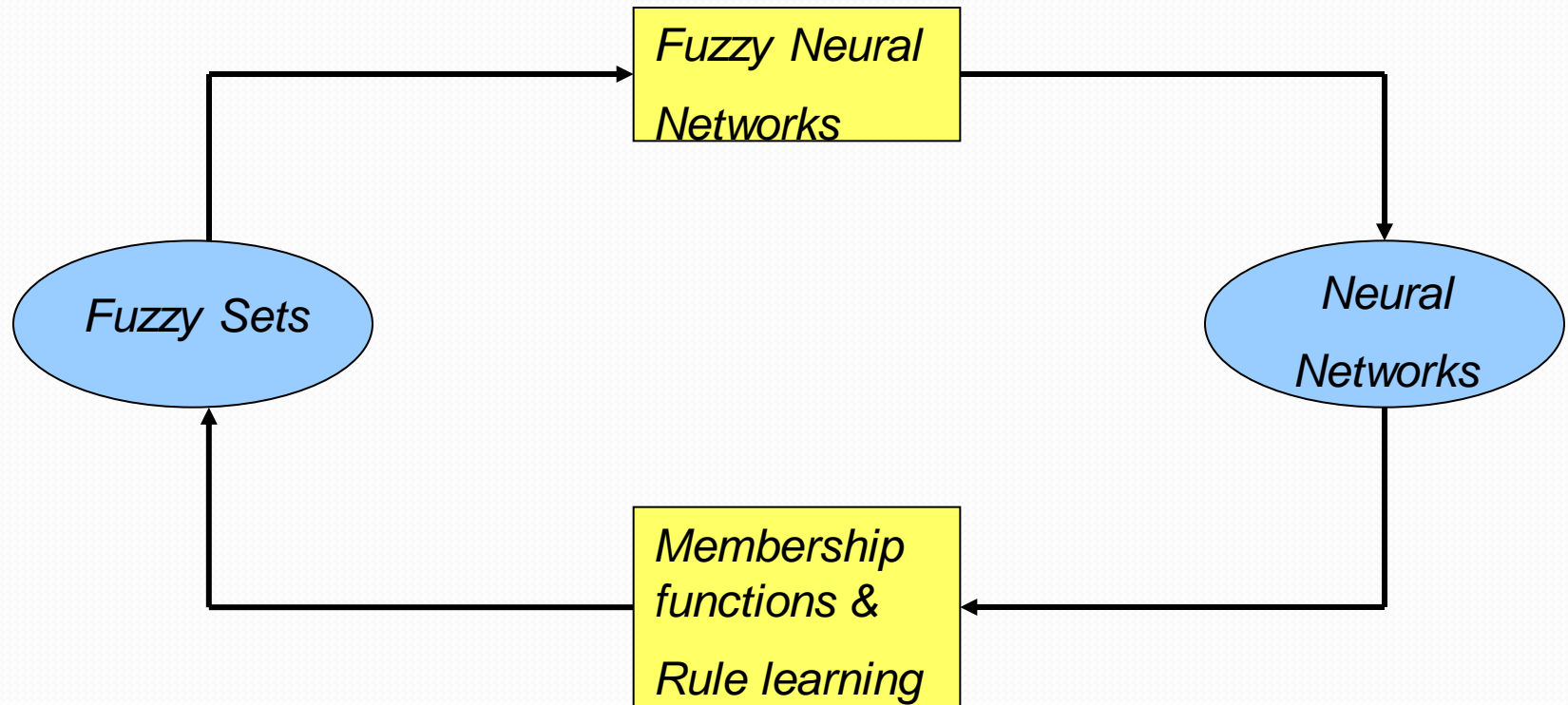
# Permasalahan *Fuzzy Systems*

- Fungsi keanggotaan: **bentuk** & **kemiringan**
- Bagaimana membangun aturan *fuzzy*?

## Solusi?

Gunakan **ANN** untuk mendapatkan fungsi keanggotaan dan/atau aturan *fuzzy* yang optimal.

# Interaksi ANN dan FS [TET01]



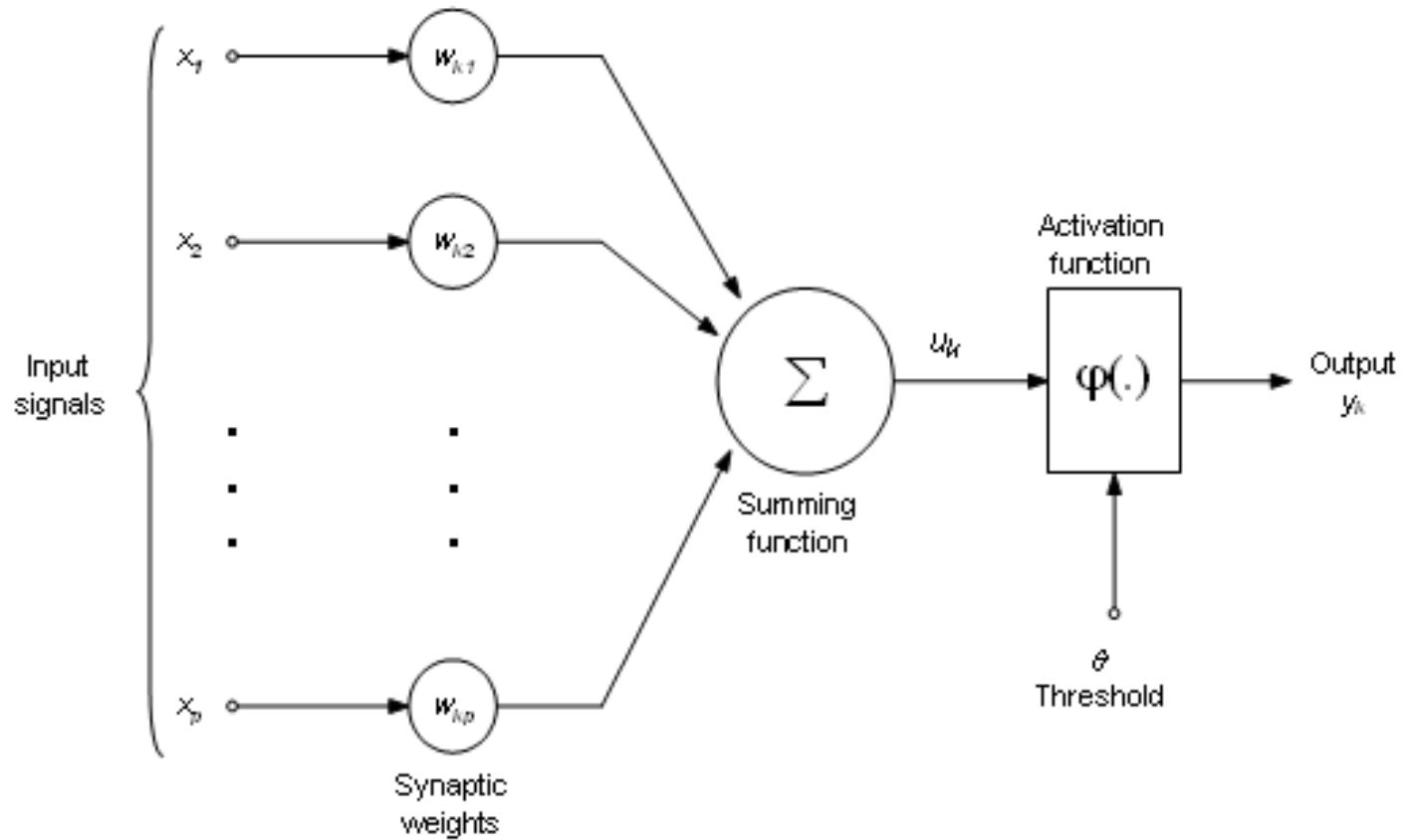
# Tiga kombinasi ANN & Fuzzy [TET01]

- **Co-operative**
  - Off-line: ANN mempelajari fungsi keanggotaan dan/atau aturan sistem fuzzy **hanya sekali untuk selamanya**.
  - On-line: ANN mempelajari fungsi keanggotaan dan/atau aturan sistem fuzzy pada **saat sistem tersebut beroperasi**.
- **Concurrent** (lebih tepat sekuensial) dimana ANN diaplikasikan sebagai **pre** atau **post processing**.
- **Hybrid** dimana fuzzy system direpresentasikan sebagai struktur jaringan (**yang bisa belajar**).

# *Fuzzy Neural Network*

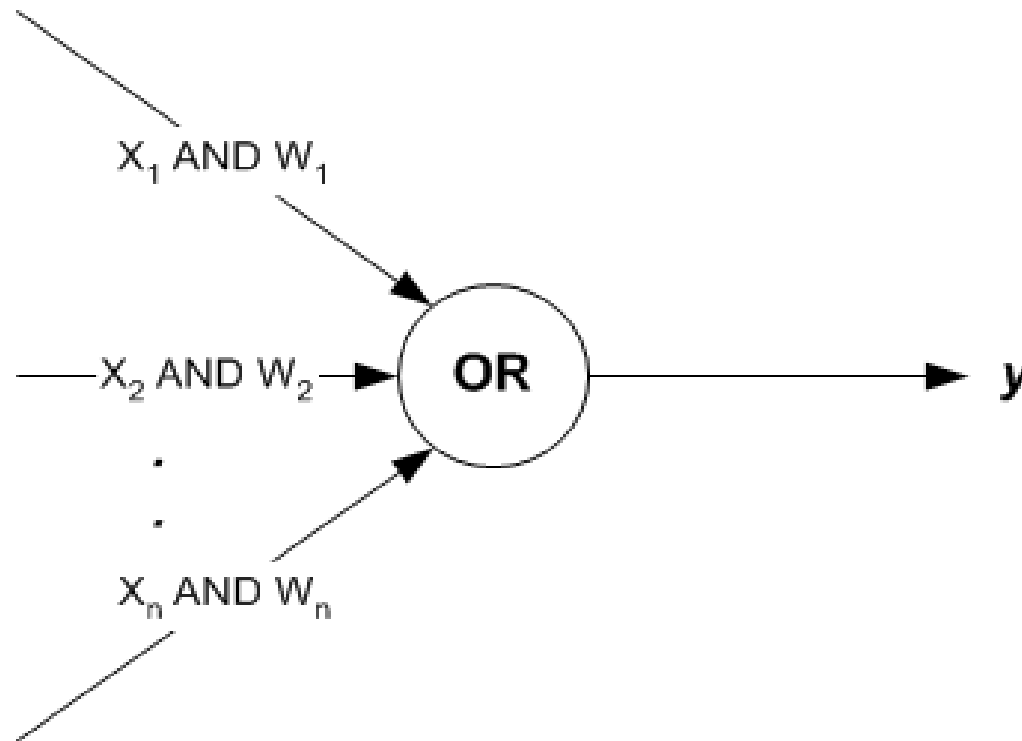
- *Fuzziness* dapat aplikasikan pada ANN dalam beberapa cara:
  - *Fuzzy neuron*
  - *Multilayered Fuzzy Neural Network*

# Neuron konvensional



$$y = g(A(w, x))$$

# OR *fuzzy neuron* [TET01]

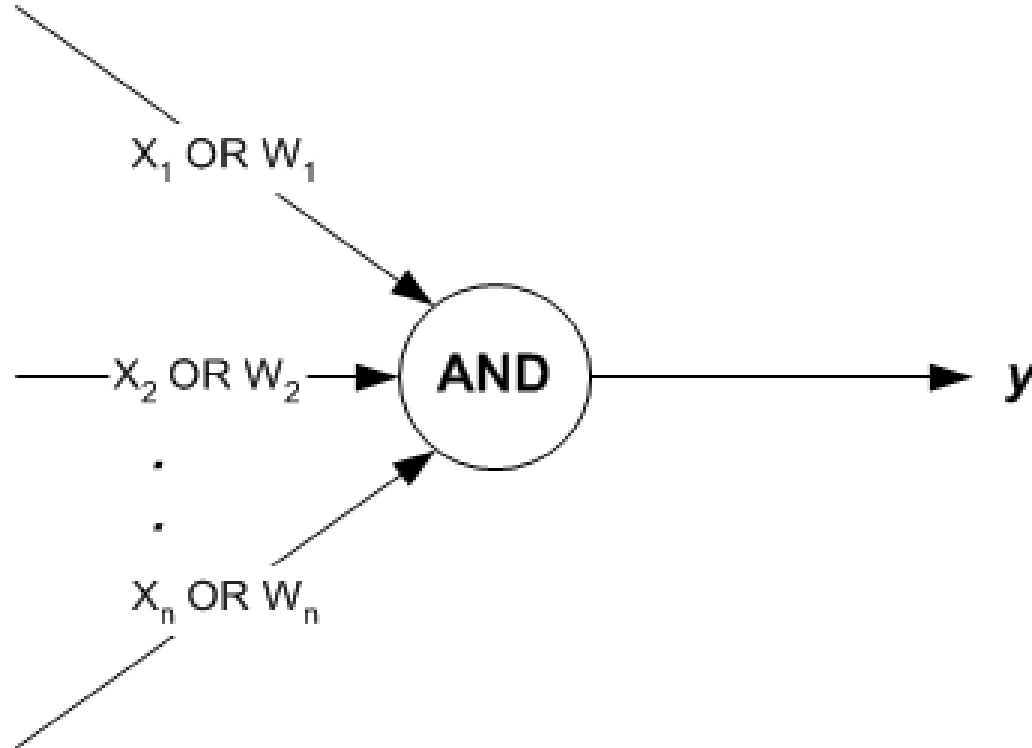


$$y = OR(x_1 \text{ AND } w_1, x_2 \text{ AND } w_2, \dots, x_n \text{ AND } w_n)$$

$$y = \bigvee_{j=1}^n (x_j \Delta w_j)$$



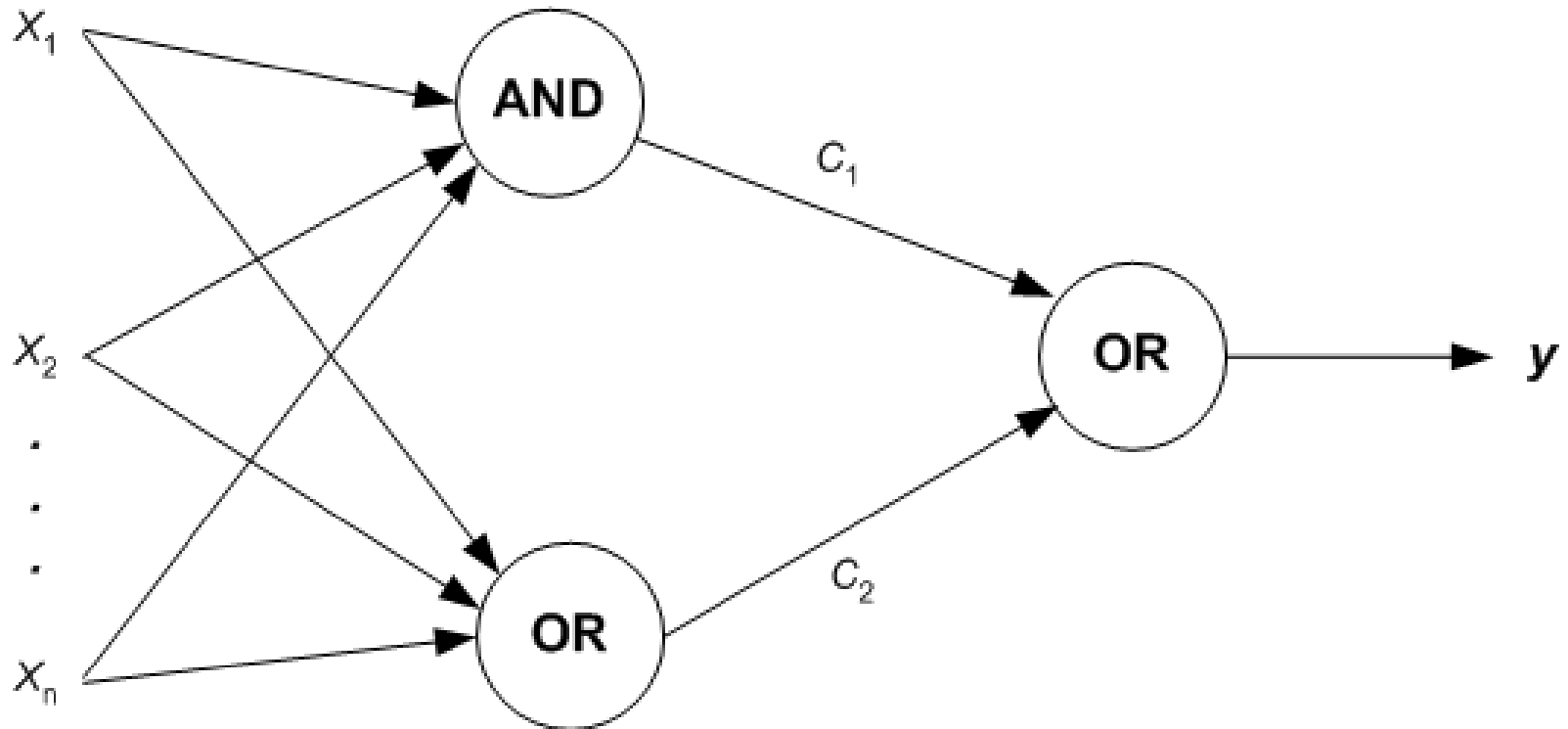
# AND fuzzy neuron [TET01]



$$y = \text{AND}(x_1 \text{ OR } w_1, x_2 \text{ OR } w_2, \dots, x_n \text{ OR } w_n)$$

$$y = \Delta_{j=1}^n (x_j \nabla w_j)$$

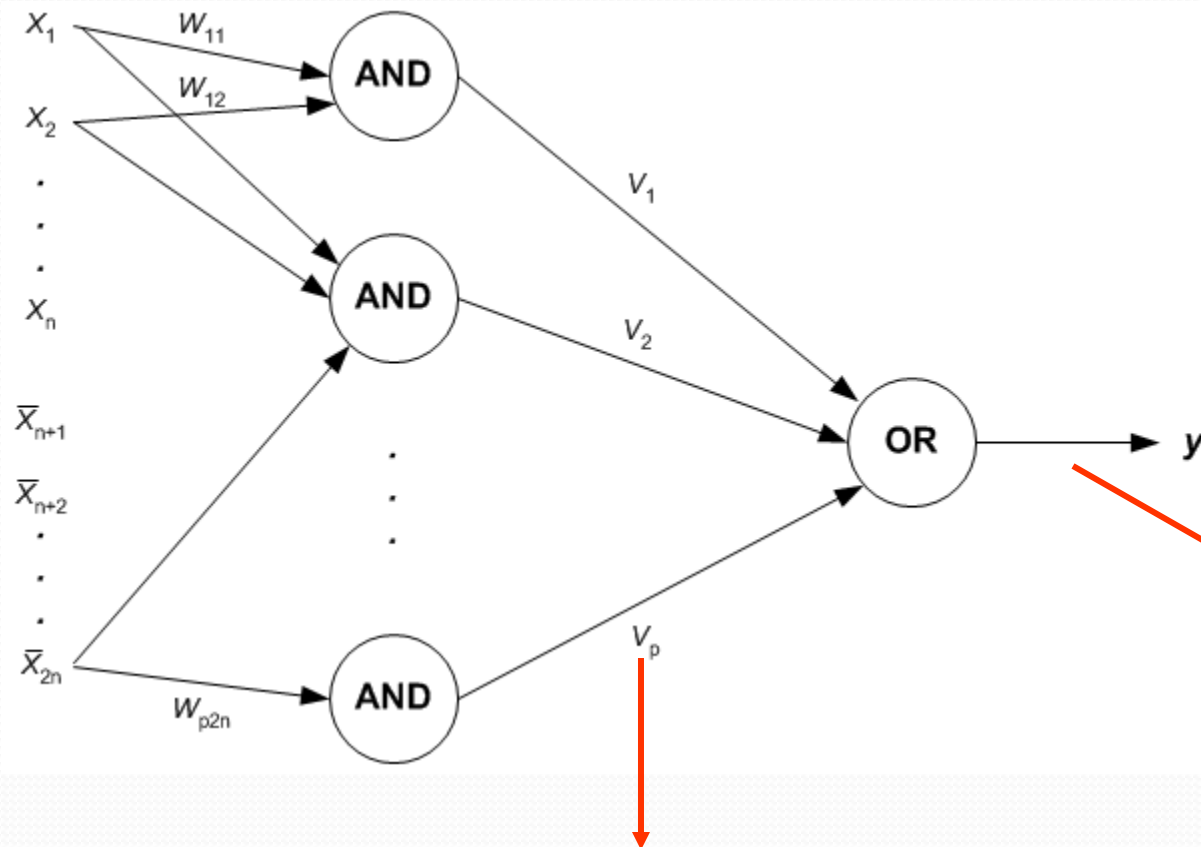
# OR/AND *fuzzy neuron*



Jika  $C_1 = 1$  dan  $C_2 = 0$ , maka akan menjadi AND fuzzy neuron

Jika  $C_1 = 0$  dan  $C_2 = 1$ , maka akan menjadi OR fuzzy neuron

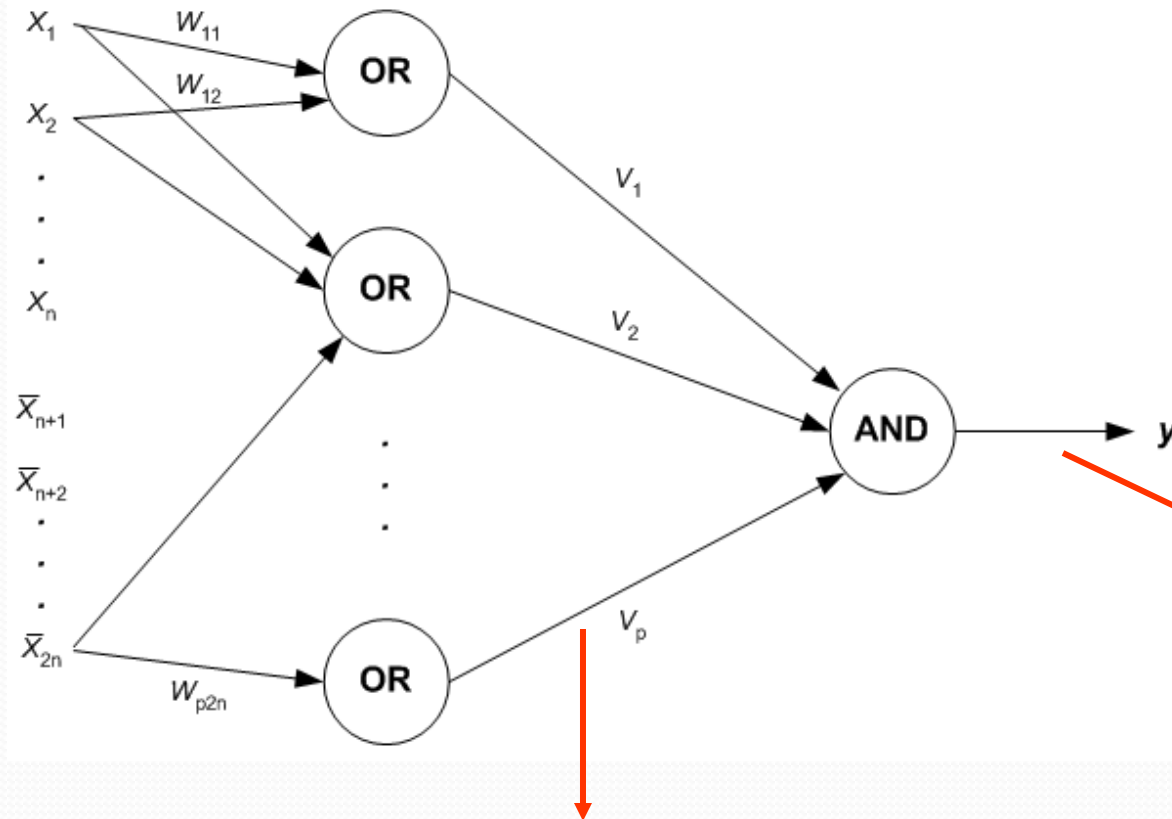
# 3 Layer ANN dengan *AND fuzzy units* pada *hidden layer*



$$y = \nabla_{j=1}^p (x_j \Delta v_j)$$

$$v_k = \left[ \Delta_{j=1}^n (x_j \nabla w_{k,j}) \right] \Delta \left[ \Delta_{j=1}^n (\bar{x}_j \nabla w_{k,(n+j)}) \right], \quad k = 1, 2, \dots, p$$

# 3 Layer ANN dengan *OR fuzzy units* pada *hidden layer*



$$y = \Delta_{j=1}^p (x_j \nabla v_j)$$

$$v_k = [\nabla_{j=1}^n (x_j \Delta w_{k,j})] \nabla [\nabla_{j=1}^n (\bar{x}_j \Delta w_{k,(n+j)})], \quad k = 1, 2, \dots, p$$

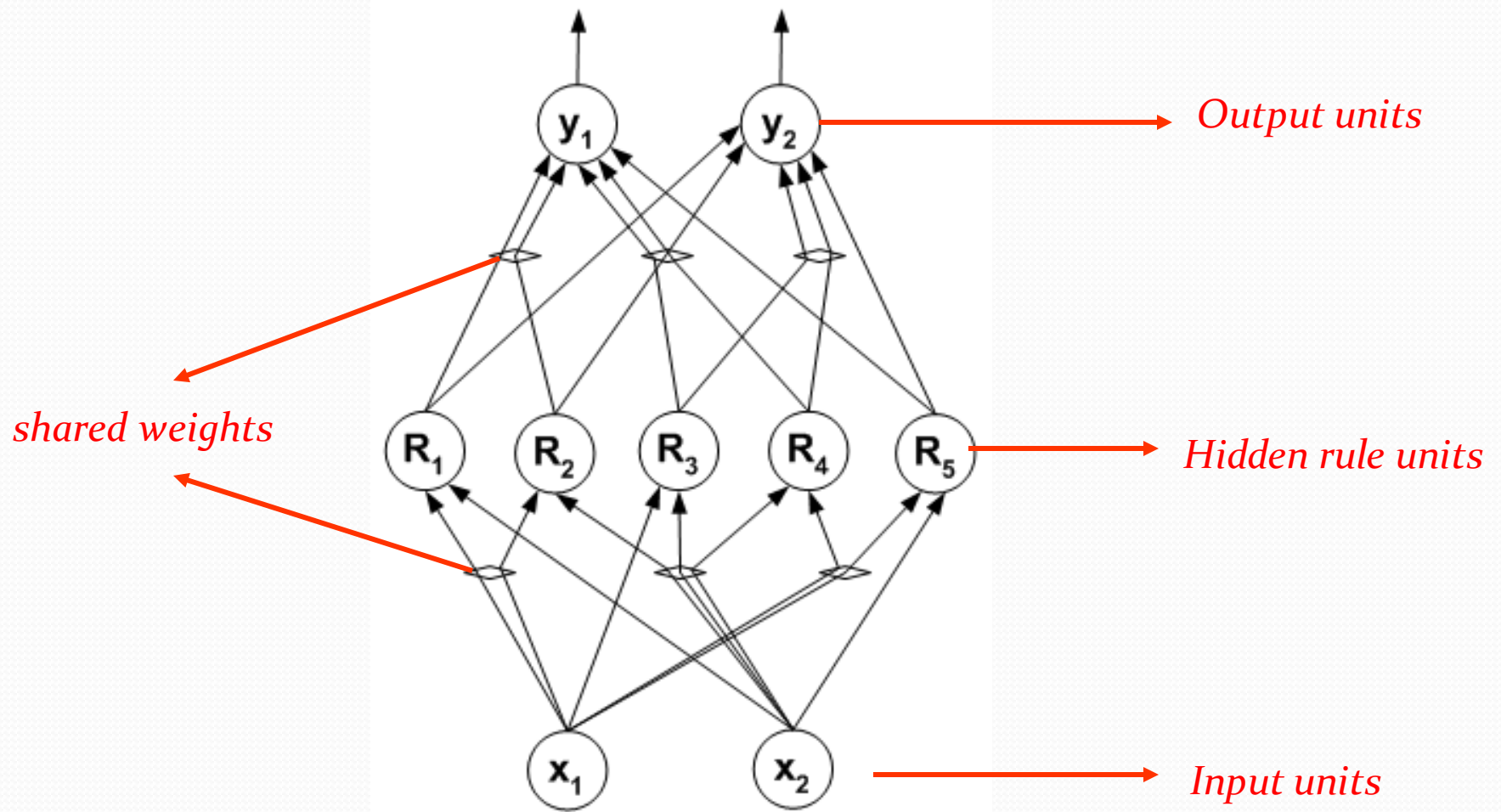
# NEFPROX (NEuro-Fuzzy function apPROXimator)

- Sesuai dengan namanya, NEFPROX digunakan untuk *approximating a continuous unknown function specified by sample input/output data pairs*.
- Feedforward Network dengan 3 layer
- Tidak ada *cycles*
- Tidak ada koneksi antara layer  $n$  dan layer  $n + j$  (dimana  $j > 1$ )
- Jaringan ini mampu belajar dan memberikan *fuzzy inference path*

# Karakteristik NEFPROX

- Input unit diberi label  $x_1, x_2, \dots, x_n$
- Hidden rule units dinyatakan  $R_1, R_2, \dots, R_k$
- Output units dilambangkan dengan  $y_1, y_2, \dots, y_m$
- Setiap koneksi diberi bobot dgn suatu himpunan fuzzy dan diberi label dengan istilah linguistik.
- Semua koneksi yang berasal dari unit input yang sama dan mempunyai label sama diberi bobot yang sama menggunakan suatu *shared weight*.
- Tidak ada dua aturan dgn *antecedents* yang sama.

# NEFPROX



# Proses belajar pada NEFPROX

- Algoritma belajar terdiri dari dua bagian:
  - A structure-learning (*fuzzy rule*)
  - Parameter-learning (*connection weight*)
- Jika kita **tidak memiliki pengetahuan** tentang aturan, maka jaringan **tidak memiliki *hidden rule units*** pada awal proses belajar.



# *Structure-learning algorithm* [TET01]

1. Pilih pola latih  $(s, t)$  dari *training set*
2. Untuk setiap input unit  $x_i$  cari fungsi keanggotaan  $\mu_{ji}^{(i)}$  sehingga

$$\mu_{ji}^{(i)}(s_i) = \max_{j \in \{1, \dots, p_i\}} \left\{ \mu_j^{(i)}(s_i) \right\}$$

# *Structure-learning algorithm* [TET01]

3. Jika tidak ada rule  $R$  dengan bobot

$$W(x_1, R) = \mu_{ji}^{(1)}, \dots, W(x_n, R) = \mu_{jn}^{(1)}$$

maka buat *node* untuk rule tersebut dan hubungkan node tersebut ke semua *output nodes*.

# Structure-learning algorithm [TET01]

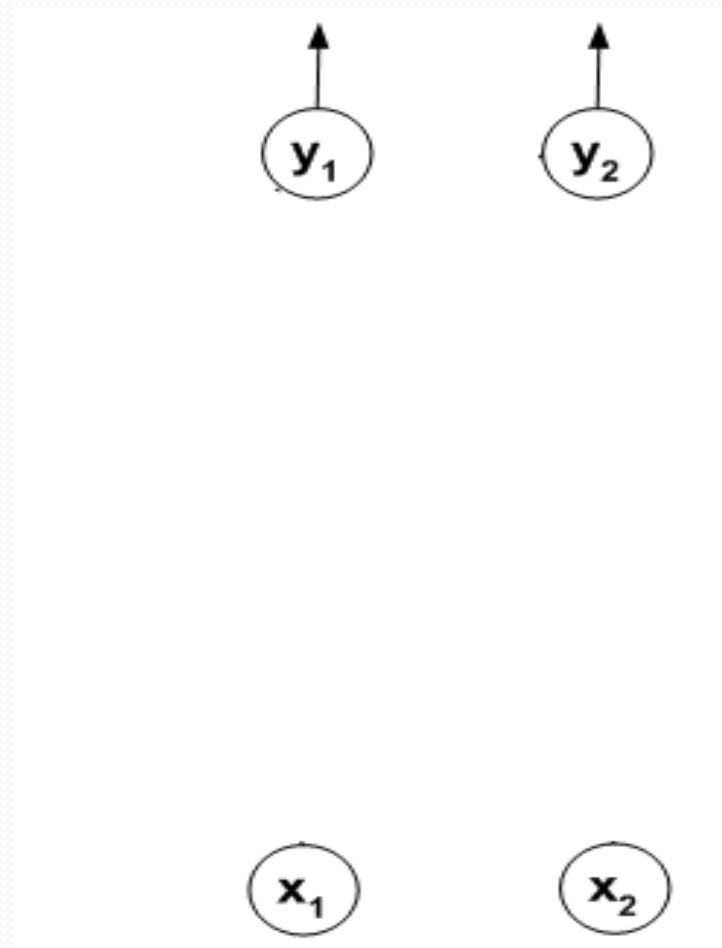
4. Untuk setiap koneksi dari *rule node* baru ke *output nodes*, cari fuzzy weight  $v_{ji}^{(i)}$  yang sesuai menggunakan fungsi-fungsi keanggotaan yang di-*assigned* ke output units  $y_i$  sehingga

$$v_{ji}^{(i)}(t_i) = \max_{j \in \{1, \dots, q_i\}} \{v_j^{(i)}(t_i)\} \quad \text{dan} \quad v_j^{(i)}(t_i) \geq 0.5$$

Jika himpunan fuzzy tidak terdefinisi, maka buat himpunan fuzzy baru  $v_{new}^{(i)}(t_i)$  untuk variabel output  $y_i$  dan set  $W(R, y_i) = v_{new}^{(i)}$

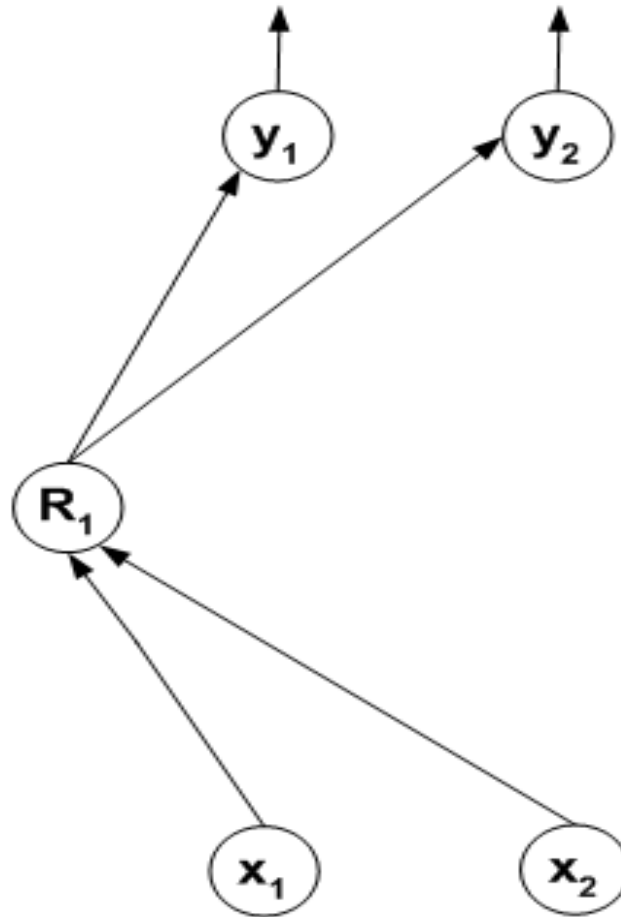
# Ilustrasi (0)

## *Structure-learning algorithm*



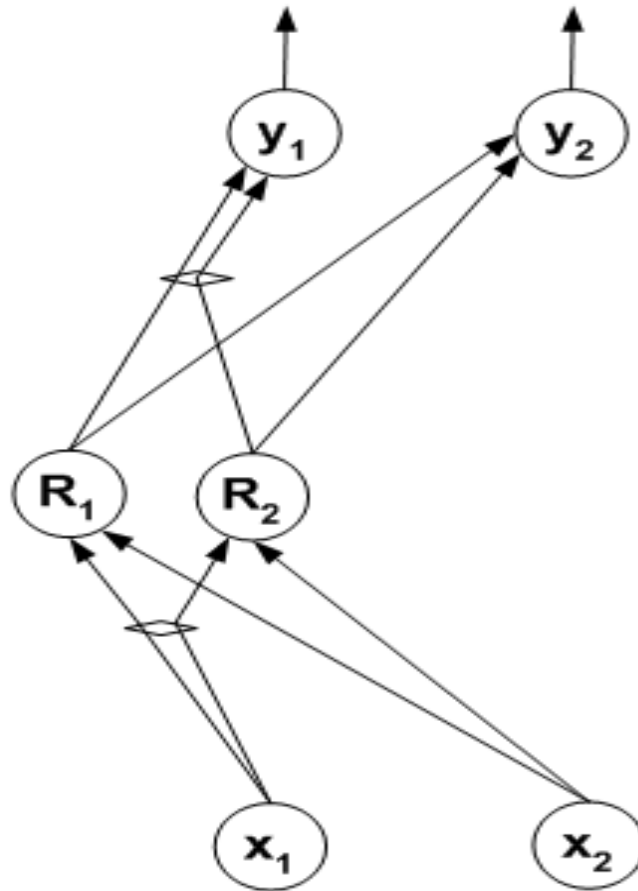
# Ilustrasi (1)

## *Structure-learning algorithm*



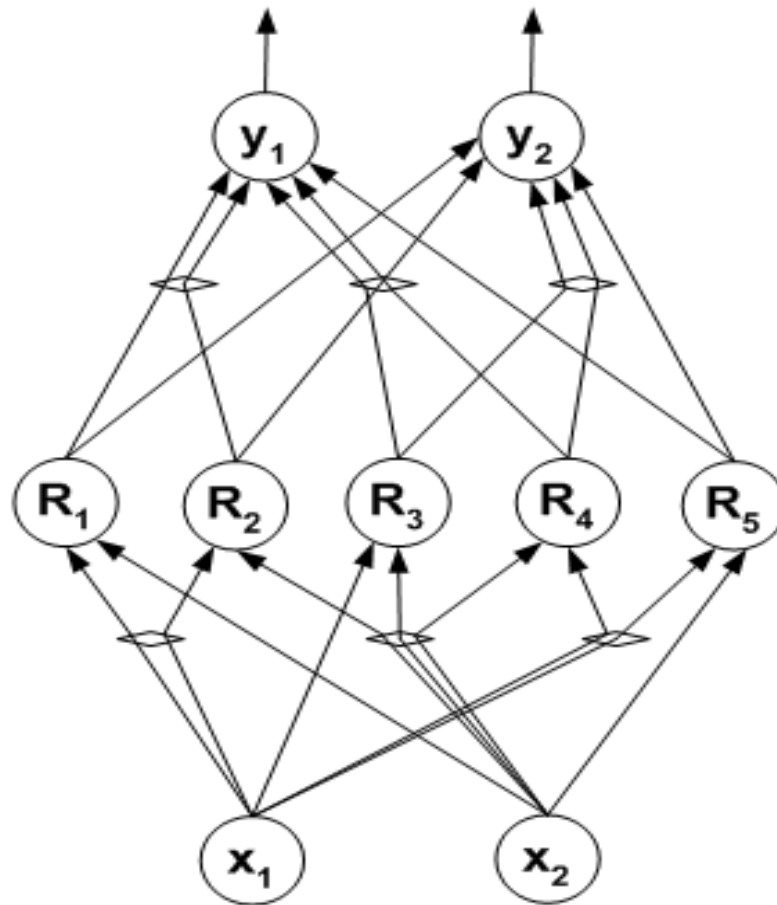
## Ilustrasi (2)

### *Structure-learning algorithm*



## Ilustrasi (3)

### *Structure-learning algorithm*



## *Parameter-learning algorithm* [TET01]

1. Pilih pola latih  $(s, t)$  dari *training set* dan letakkan pada input layer
2. Propagasikan maju pola tersebut melalui hidden layer sampai *output units* menghasilkan vektor output  $\mathbf{o}$ .
3. Untuk setiap *output unit*  $y_i$  tentukan *error*-nya

$$\delta_{yi} = t_i - o_{yi}$$



## *Parameter-learning algorithm* [TET01]

4. Untuk setiap rule unit  $R$  dengan output  $\mathbf{o}_R > 0$  lakukan
  - *Update* parameter2 himpunan fuzzy  $W(R, y_i)$  menggunakan *learning rate*  $\sigma > 0$
  - *Tentukan perubahan*  $\delta_R = o_R(1 - o_R) \cdot \sum_{y \in \text{outputlayer}} (2W(R, y)(t_i) - 1) \cdot |\delta_y|$
  - *Update* parameter2 himpunan fuzzy  $W(x, R)$  menggunakan  $\delta_R$  dan  $\sigma$  untuk menghitung variasi-variasi tersebut.
5. Jika kriteria konvergen telah tercapai, maka berhenti. Jika tidak, kembali ke langkah 1.

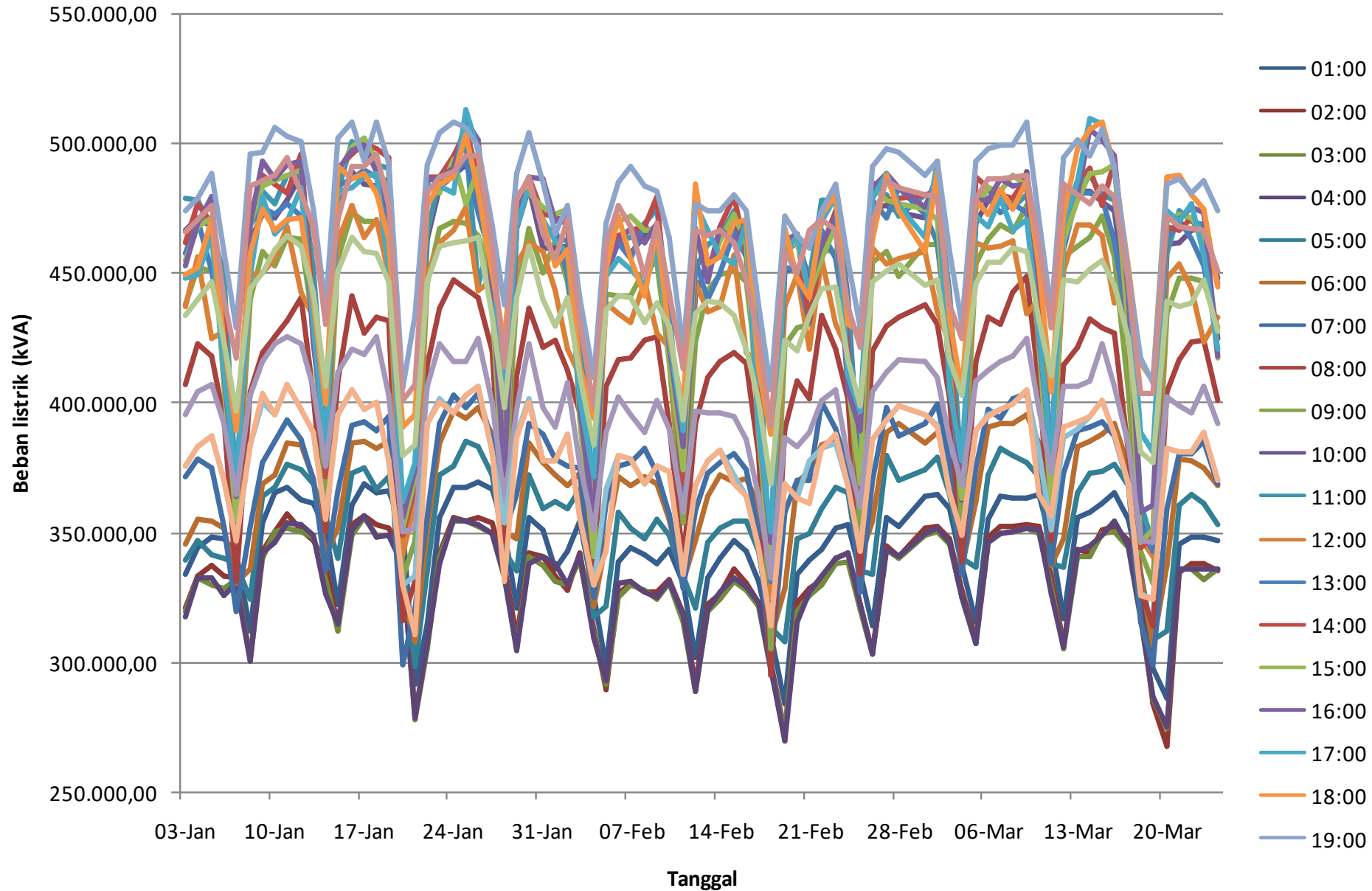
# *Parameter-learning algorithm*

- Prosedur belajar berbasis pada *simple heuristic method*.
- Prosedur ini menghasilkan pergeseran fungsi-fungsi keanggotaan.

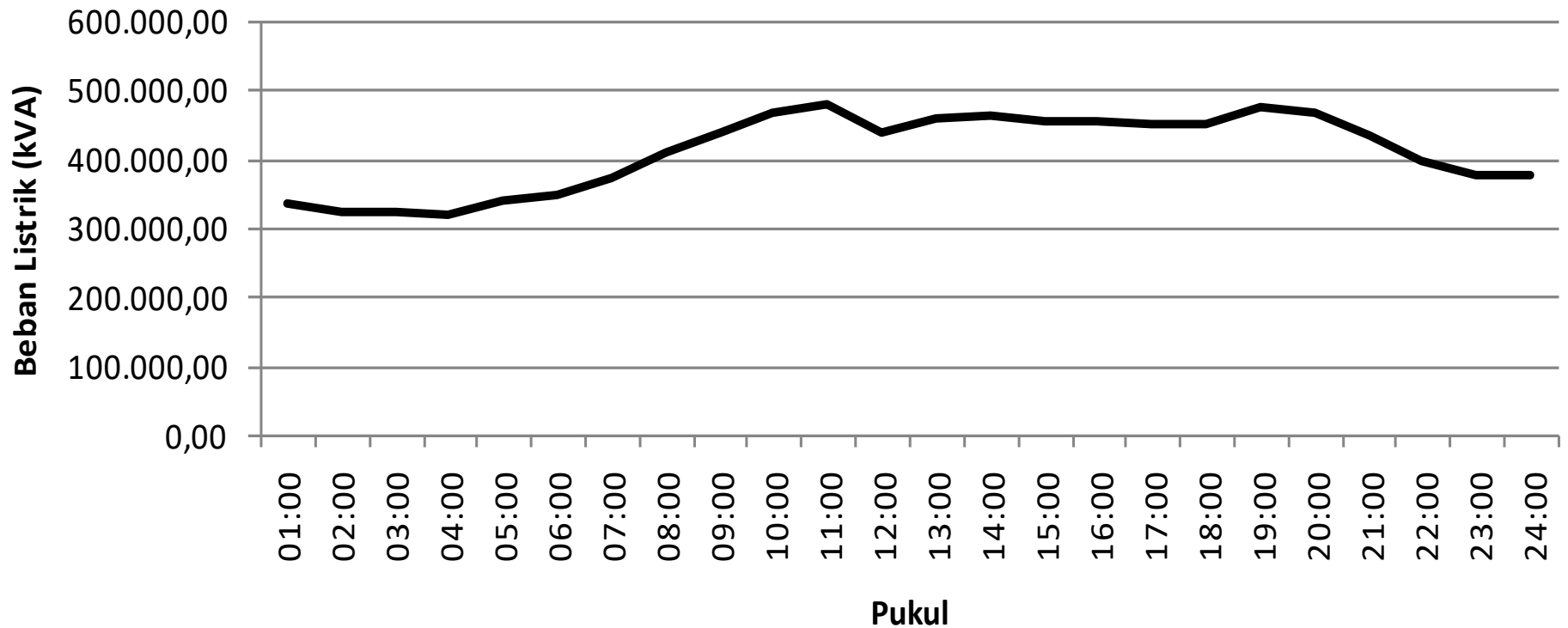
## Kasus: Prediksi Beban listrik jangka pendek

- Tenaga listrik harus tersedia pada saat dibutuhkan
- Tenaga listrik tidak dapat disimpan dalam skala besar
- Berapa tenaga listrik yang perlu dibangkitkan untuk memenuhi kebutuhan tenaga listrik konsumen setiap jam (24 jam) untuk beberapa hari yang akan datang?

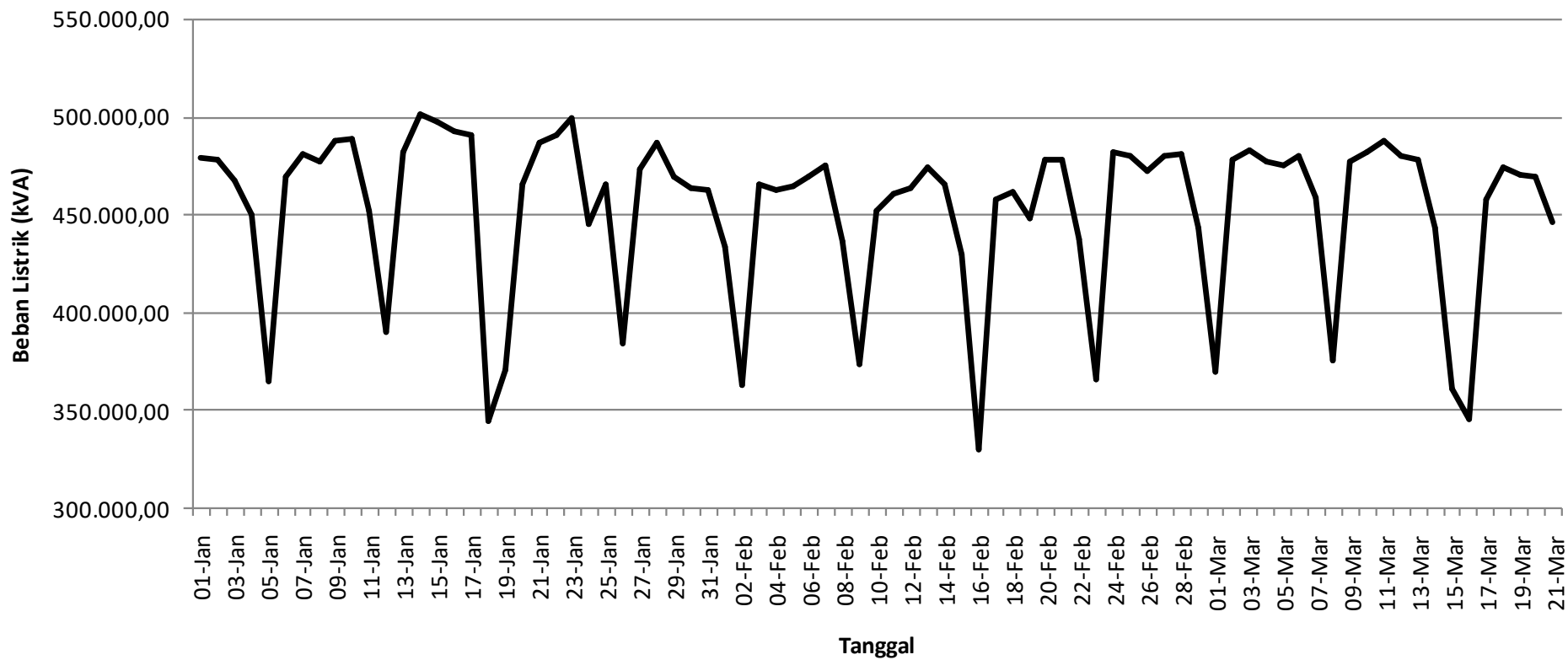
# Data Beban Listrik per jam (pukul 01:00 - 24:00) Periode 03 Januari - 24 Maret 2007 di suatu APJ



## Beban listrik pada tanggal 03 Jan 2007



## Beban Listrik pada pukul 11:00 untuk Periode 03 Jan - 24 Mar 2007



# NEFPROX

- *fuzzy set* yang digunakan adalah segitiga
- *Jumlah fuzzy set* = 4 atau 5
- Jumlah input = 4 atau 5
- *Learning rate* ( $\sigma$ ) = 0; 0,0025; 0,006; dan 0,01
- FK memiliki *intersect* 0,5 (setengah bagian overlap)
- Domain fungsi keanggotaan dalam interval tertentu

Pukul	Domain (kVA)	Pukul	Domain (kVA)
01:00	270000 - 390000	13:00	320000 - 510000
02:00	250000 - 370000	14:00	320000 - 535000
03:00	250000 - 370000	15:00	320000 - 535000
04:00	250000 - 370000	16:00	330000 - 530000
05:00	280000 - 415000	17:00	340000 - 540000
06:00	275000 - 420000	18:00	375000 - 525000
07:00	275000 - 420000	19:00	375000 - 530000
08:00	280000 - 475000	20:00	370000 - 510000
09:00	290000 - 500000	21:00	345000 - 475000
10:00	310000 - 510000	22:00	320000 - 440000
11:00	310000 - 520000	23:00	295000 - 435000
12:00	300000 - 500000	24:00	300000 - 420000

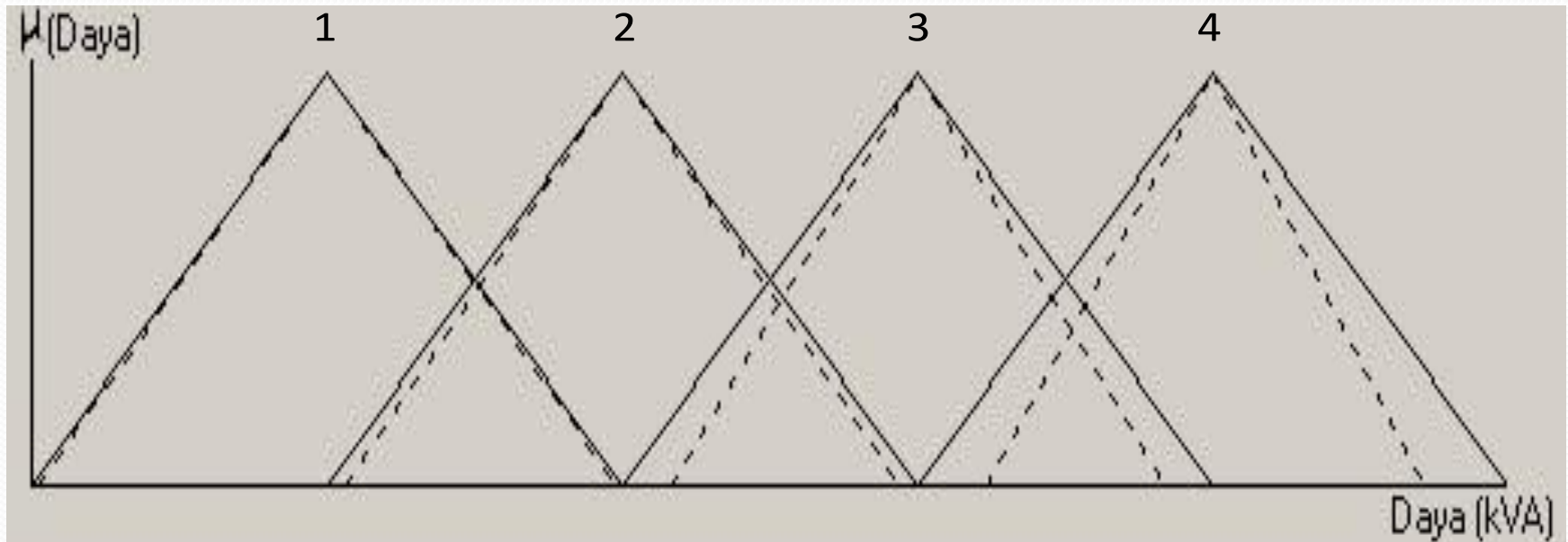


# Structure Learning

- Jika *jumlah fuzzy set* ( $F$ ) = 4 dan *jumlah input* ( $P$ ) = 4, maka total aturan maksimum = 256
- Jika *jumlah fuzzy set* ( $F$ ) = 5 dan *jumlah input* ( $P$ ) = 5, maka total aturan maksimum = 3125

Pukul	Jumlah <i>input</i> optimal	Jumlah <i>Fuzzy Set</i> optimal	$\sigma$	Jumlah Aturan yang dihasilkan
01:00	5	4	0.01	40
02:00	5	5	0.006	36
03:00	5	5	0.01	39
04:00	5	5	0.01	43
05:00	5	5	0.01	42
06:00	5	5	0.01	44
07:00	5	5	0.006	49
08:00	5	5	0.01	43
09:00	5	5	0.006	35
10:00	5	5	0.01	35
11:00	5	5	0.006	36
12:00	5	5	0.006	38
13:00	4	5	0.0025	24
14:00	5	5	0.006	45
15:00	5	4	0.0025	43
16:00	5	4	0.0025	35
17:00	4	4	0.006	30
18:00	5	4	0.0025	51
19:00	5	4	0.006	34
20:00	5	4	0.006	43
21:00	5	4	0.006	35
22:00	4	4	0.006	36
23:00	4	4	0.006	34
24:00	4	5	0.01	45

# Parameter Learning



# Akurasi NEFPROX

$$MAPE = \frac{1}{P} \sum_{i=1}^P \frac{|T(i) - O(i)|}{|T(i)|} * 100\%$$

- $T(i)$  adalah beban listrik sebenarnya (target)
- $O(i)$  adalah nilai prediksi yang dihasilkan NEFPROX
- $P$  adalah jumlah pasangan data (*input* dan targetnya)
- Semakin kecil MAPE yang dihasilkan berarti semakin bagus performansi NEFPROX.

MAPE untuk *validation* set pukul 01:00 sampai 03:00.

Jumlah <i>Input</i>	$\sigma$	MAPE (%)					
		Pukul01:00		Pukul02:00		Pukul03:00	
		Jumlah <i>fuzzy set</i>		Jumlah <i>fuzzy set</i>		Jumlah <i>fuzzy set</i>	
		4	5	4	5	4	5
4	0	2,0537	4,0982	4,5634	1,5700	2,6726	1,5364
	0,0025	2,0191	4,0331	4,6225	1,5469	2,5959	1,5167
	0,006	1,7833	4,4739	4,6514	1,6057	2,6137	1,4374
	0,01	1,8292	4,5383	4,5684	1,6422	2,7300	1,5295
5	0	1,1728	4,2584	4,4250	1,0040	2,1827	1,0479
	0,0025	1,3648	4,1078	4,4126	0,8095	2,2589	0,8846
	0,006	0,7216	2,7773	4,2754	0,7087	2,4307	0,6565
	0,01	0,6208			0,7511	2,6310	0,6437

MAPE untuk *validation set* pukul 07:00 sampai 09:00.

Jumlah <i>Input</i>	$\sigma$	MAPE (%)					
		Pukul07:00		Pukul08:00		Pukul09:00	
		Jumlah <i>fuzzy set</i>		Jumlah <i>fuzzy set</i>		Jumlah <i>fuzzy set</i>	
		4	5	4	5	4	5
4	0	3,6292	3,2427	4,1060	3,3450	3,5731	2,5512
	0,0025	3,5908	3,1867	3,2290	2,4405	3,2101	2,5337
	0,006	3,5931	2,8979	3,3454	1,5548	3,3943	2,6345
	0,01	3,2748	2,7334	3,2876	1,5179		2,8007
5	0	3,0657	2,4545	4,1402	2,2193	4,4337	2,3040
	0,0025	3,0204	2,3033	3,3906	1,6534	3,1605	1,7558
	0,006	3,0670	2,0763	3,4176	1,6592	3,0359	1,3242
	0,01	3,1921		3,3253	1,4529		1,4536

MAPE untuk *validation set* pukul 19:00 sampai 21:00.

Jumlah <i>Input</i>	$\sigma$	MAPE (%)					
		Pukul 19:00		Pukul 20:00		Pukul 21:00	
		Jumlah <i>fuzzy set</i>		Jumlah <i>fuzzy set</i>		Jumlah <i>fuzzy set</i>	
		4	5	4	5	4	5
4	0	2,6541	3,2915	1,8599	1,7885	2,6374	3,8766
	0,0025	2,3836	2,9800	1,5248	1,9846	2,5538	3,7370
	0,006	1,7777	2,3619	1,4934	2,7109	2,3680	3,3896
	0,01	1,6005	1,8084	1,5067	2,7749	2,1762	2,6152
5	0	1,9521	2,0966	1,4194	1,0239	2,1370	4,4388
	0,0025	1,5541	2,0265	1,0113	1,1898	2,0251	4,3872
	0,006	0,7032	1,8351	0,9703		1,8923	4,2276
	0,01	0,7976	1,4855	0,9931		1,8973	3,3241

Kesalahan Prediksi  
dan MAPE untuk  
Test Set  
(11-17 MAR 2007)

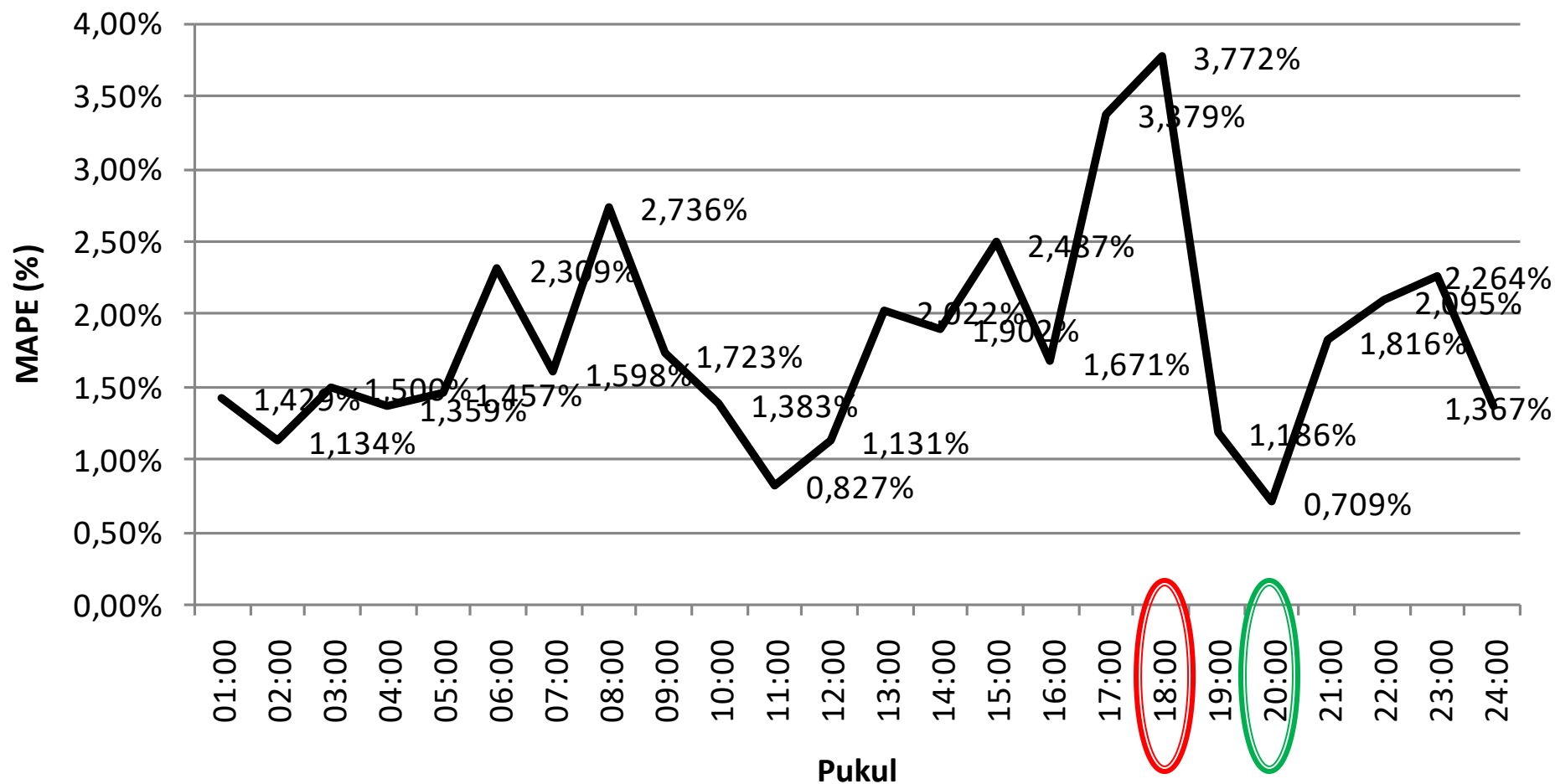
Pukul	Error (%)							MAPE
	11 Mar	12 Mar	13 Mar	14 Mar	15 Mar	16 Mar	17 Mar	
01:00	-1,426	0,290	2,190	2,096	1,103	0,044	2,852	1,429 %
02:00	-0,969	1,108	1,809	2,182	-0,319	-0,731	0,821	1,134 %
03:00	0,329	1,463	2,432	2,402	-1,974	-0,395	1,505	1,500 %
04:00	1,0516	1,291	1,932	1,380	-1,067	-1,380	1,410	1,359 %
05:00	3,435	0,019	-1,97	-1,215	-1,252	-2,128	0,179	1,457 %
06:00	2,986	-0,0006	3,303	-1,203	-4,118	0,942	3,614	2,309 %
07:00	2,815	-0,402	1,750	1,295	-1,501	-1,355	-2,068	1,598 %
08:00	-0,733	-1,233	4,943	2,366	3,213	3,686	-2,981	2,736 %
09:00	1,032	1,999	1,086	0,264	-1,490	1,97	4,218	1,723 %
10:00	2,657	1,941	-1,021	-0,975	-0,199	0,497	2,390	1,383 %
11:00	2,018	0,421	-0,138	-0,520	0,972	0,719	0,999	0,827 %
12:00	0,199	3,429	-0,452	-0,510	0,458	-1,262	-1,607	1,131 %
13:00	3,225	4,034	-2,666	-0,575	-0,265	3,243	-0,147	2,022 %
14:00	-0,053	4,867	2,738	-1,588	-2,717	1,099	0,249	1,902 %
15:00	2,276	4,852	-4,056	-0,794	-2,398	0,002	-3,033	2,487 %
16:00	-1,255	-1,130	0,899	-4,16	-2,329	-1,189	0,738	1,671 %
17:00	5,234	0,973	-2,992	-3,824	-1,795	-6,201	2,634	3,379 %
18:00	3,978	-2,923	-6,231	-5,871	-2,667	-1,951	2,785	3,772 %
19:00	0,470	0,837	-0,583	0,693	-1,298	1,573	2,848	1,186 %
20:00	-0,421	-0,619	0,292	0,936	-0,527	0,488	1,683	0,709 %
21:00	-0,242	-3,893	-1,575	-0,808	-1,504	0,545	4,146	1,816 %
22:00	0,258	-3,694	-0,805	-2,560	-3,419	0,135	3,792	2,095 %
23:00	0,165	5,037	4,210	-0,988	-1,134	2,351	1,963	2,264 %
24:00	0,865	-0,353	1,831	-4,849	-0,620	0,005	1,046	1,367 %



Beban listrik pukul 18:00 sulit diprediksi: Sangat Fluktuatif

Beban listrik pukul 20:00 mudah diprediksi: Relatif periodik

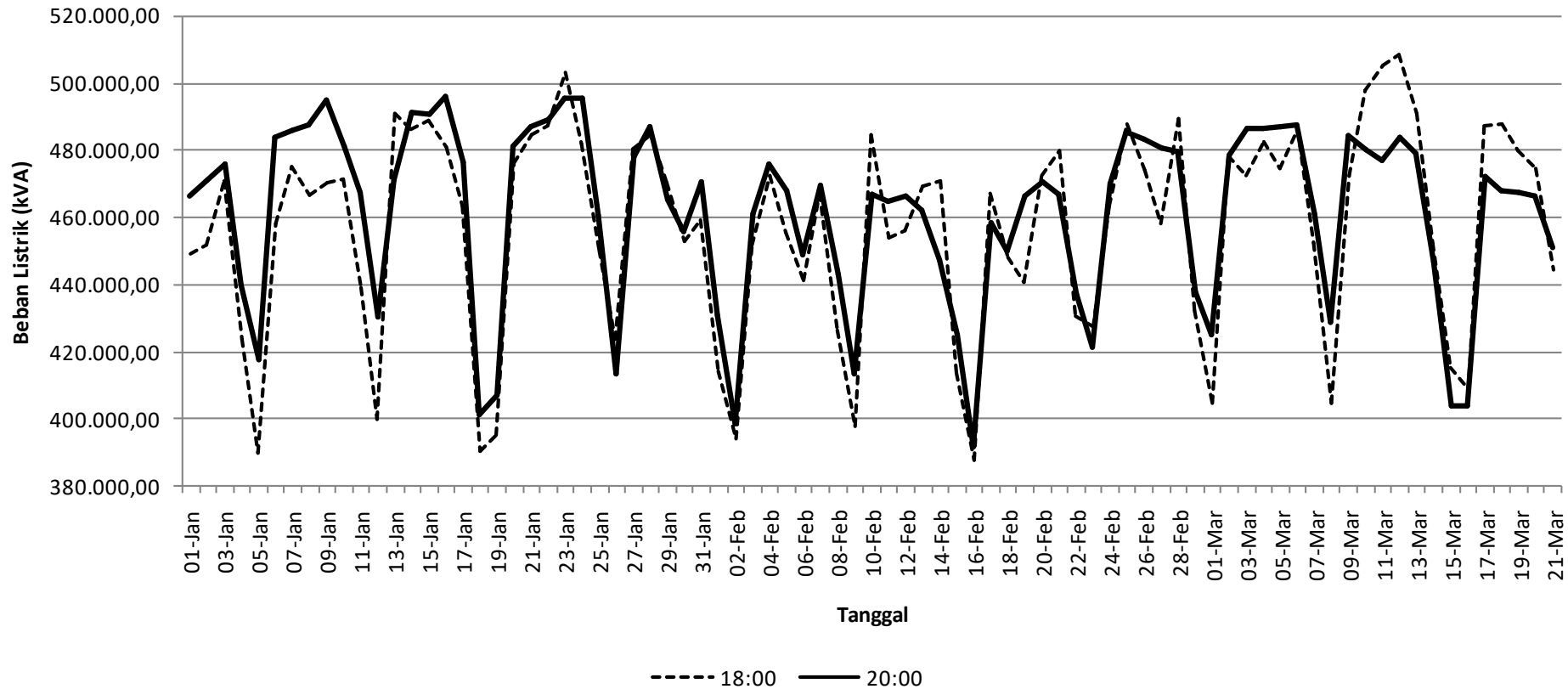
## MAPE untuk data *Test Set*: 11-17 Mar 2007



Beban listrik pukul 18:00 sulit diprediksi: Sangat Fluktuatif

Beban listrik pukul 20:00 mudah diprediksi: Relatif periodik

Beban Listrik pada pukul 18:00 dan 20:00



# ANFIS (Adaptive Network-based Fuzzy Inference System)

Pada *first order Sugeno model*, himpunan aturan menggunakan kombinasi linier dari input-input yang ada dapat diekspresikan sbb:

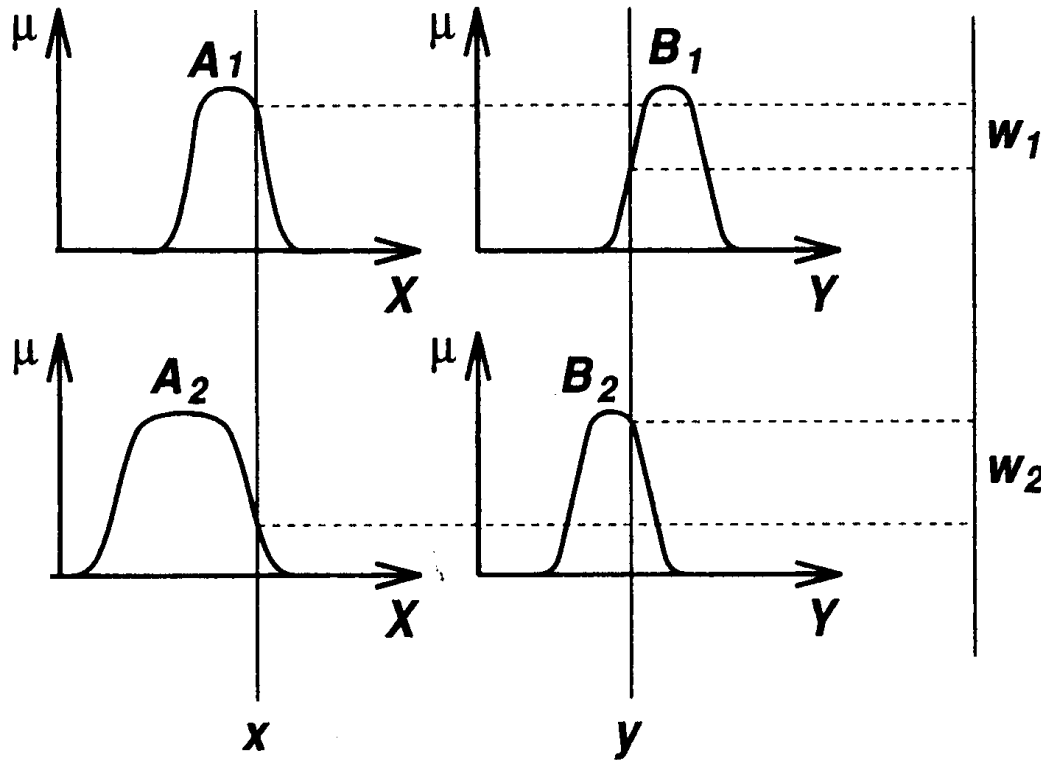
**IF**  $x$  is  $A_1$  **AND**  $y$  is  $B_1$  **THEN**  $f_1 = p_1x + q_1y + r_1$

**IF**  $x$  is  $A_2$  **AND**  $y$  is  $B_2$  **THEN**  $f_2 = p_2x + q_2y + r_2$

Mekanisme penalaran pada model ini adalah sbb:

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} = \bar{w}_1 + \bar{w}_2$$

**Min or  
Product**



$$z_1 = p_1 x + q_1 y + r_1$$

$$z_2 = p_2 x + q_2 y + r_2$$

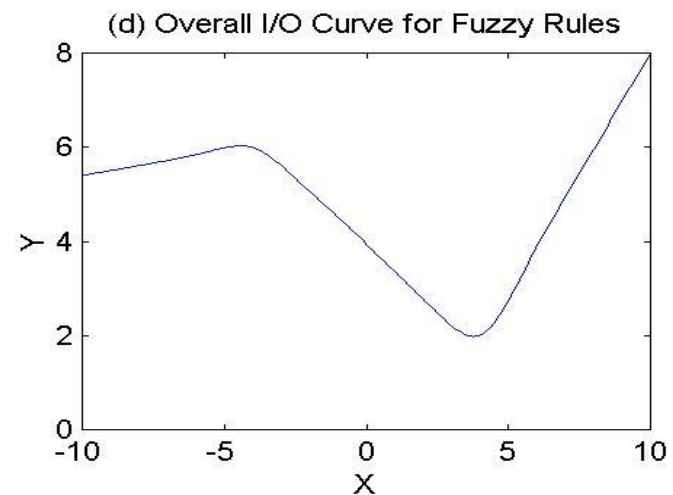
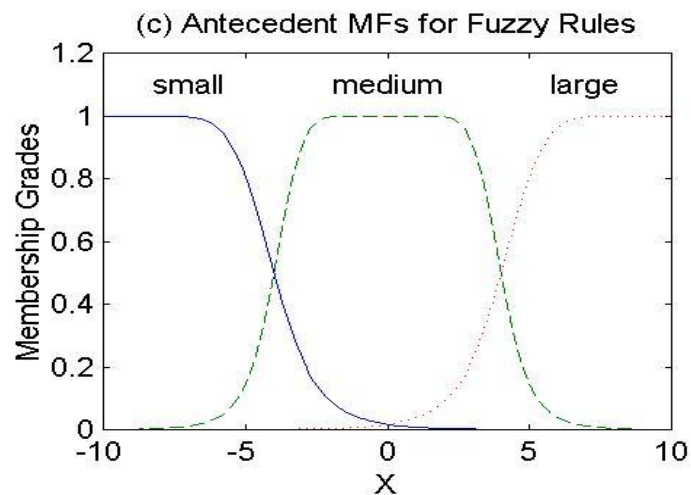
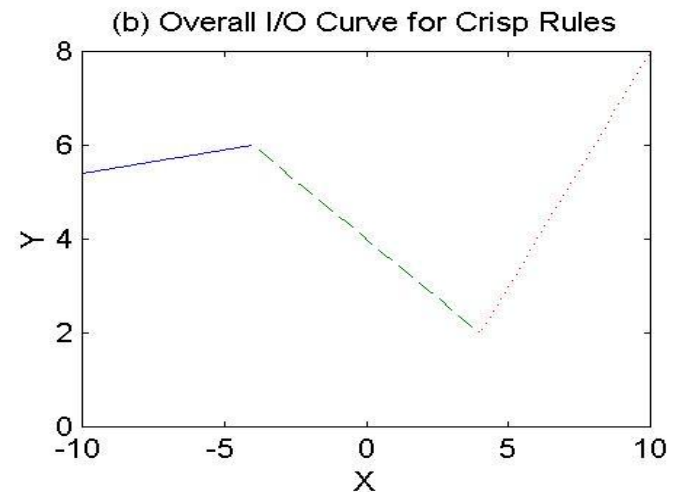
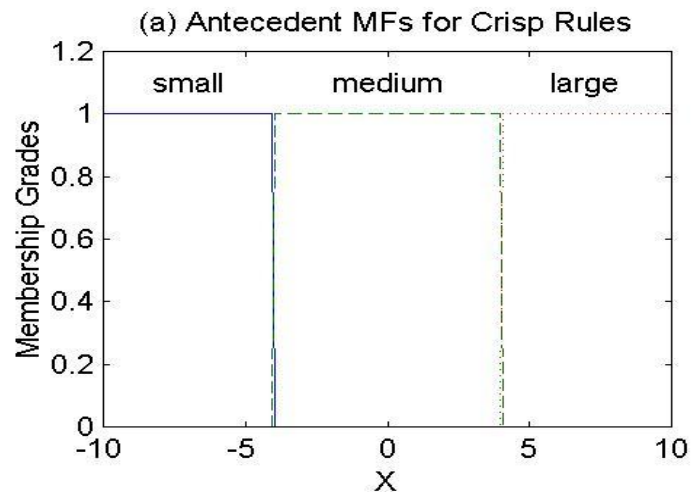


**Weighted Average**

$$z = \frac{w_1 z_1 + w_2 z_2}{w_1 + w_2}$$

**Figure 4.8.** *The Sugeno fuzzy model.*

If  $X$  is small then  $Y = 0.1X + 6.4$   
If  $X$  is medium then  $Y = -0.5X + 4$   
If  $X$  is large then  $Y = X - 2$

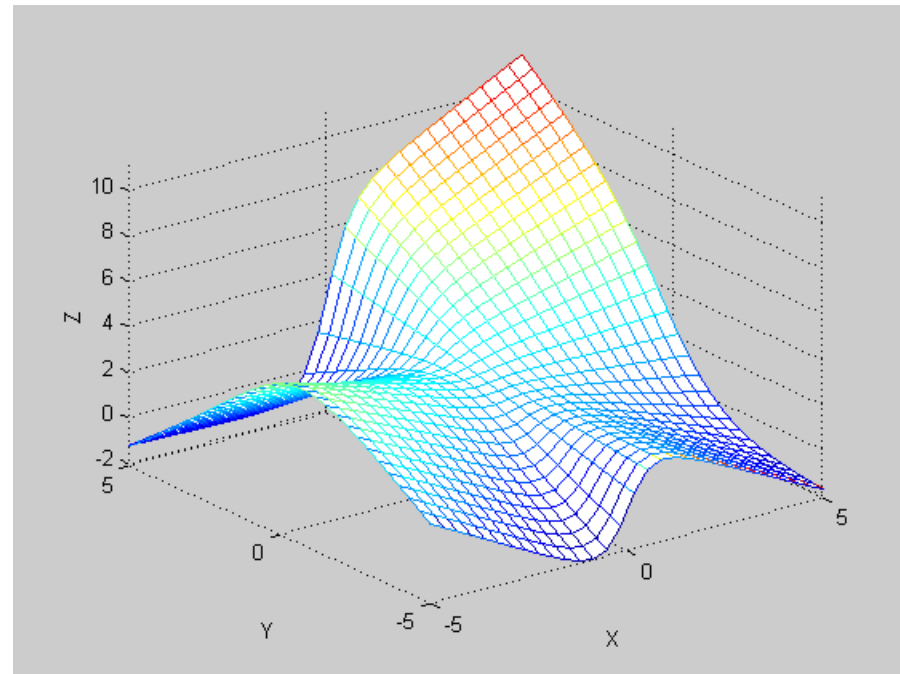
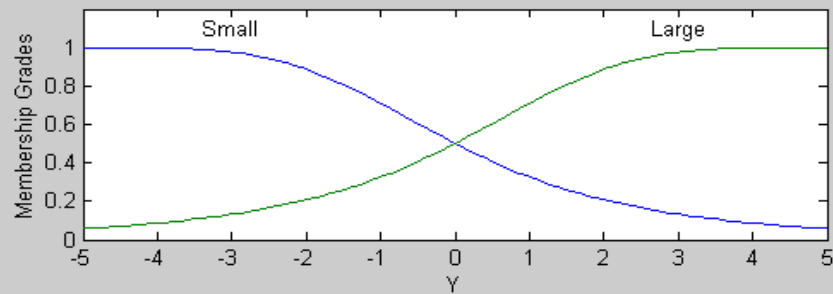
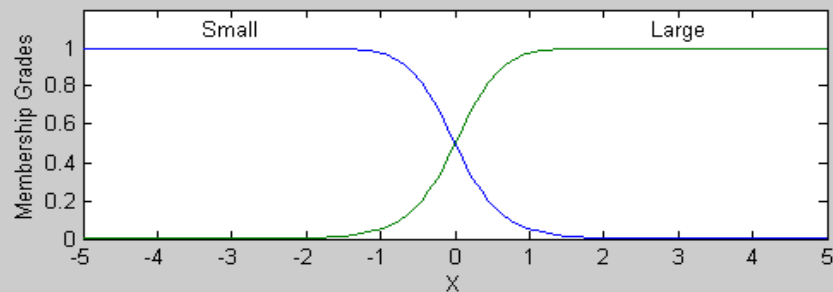


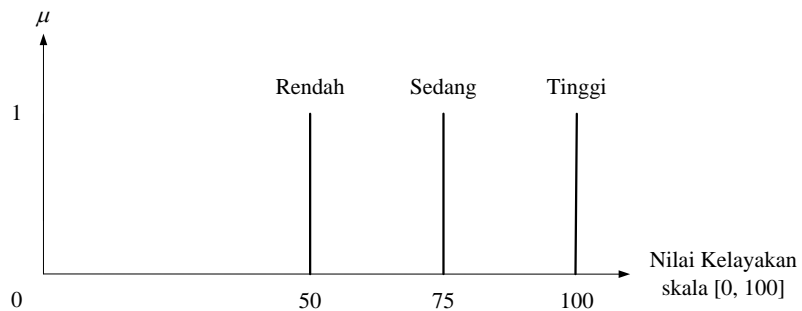
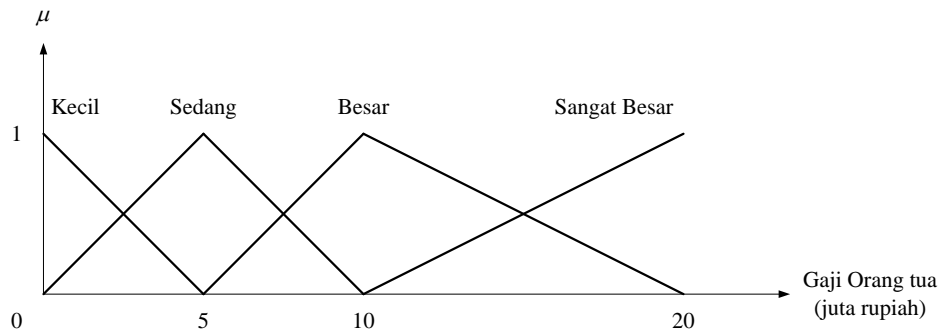
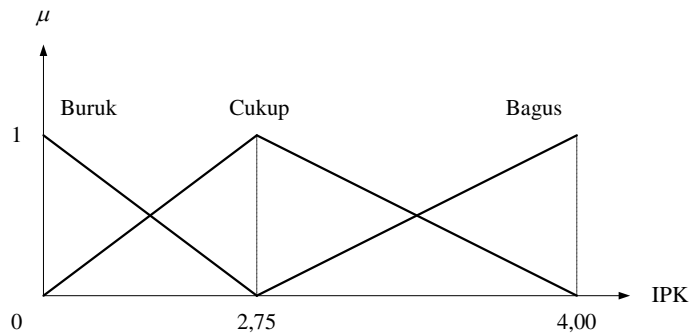
If  $X$  is small and  $Y$  is small then  $z = -x+y+1$

If  $X$  is small and  $Y$  is large then  $z = -y+3$

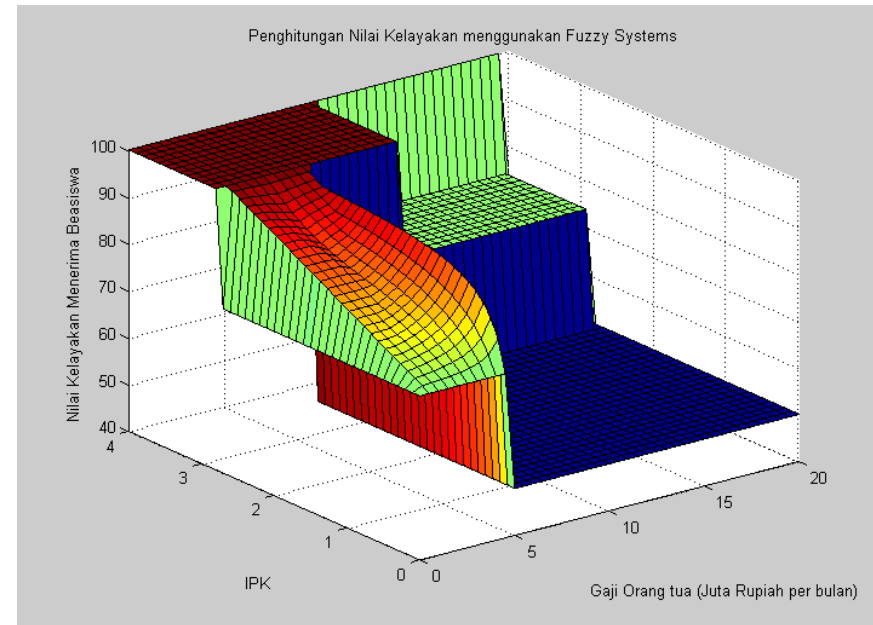
If  $X$  is large and  $Y$  is small then  $z = -x+3$

If  $X$  is large and  $Y$  is large then  $z = x+y+2$



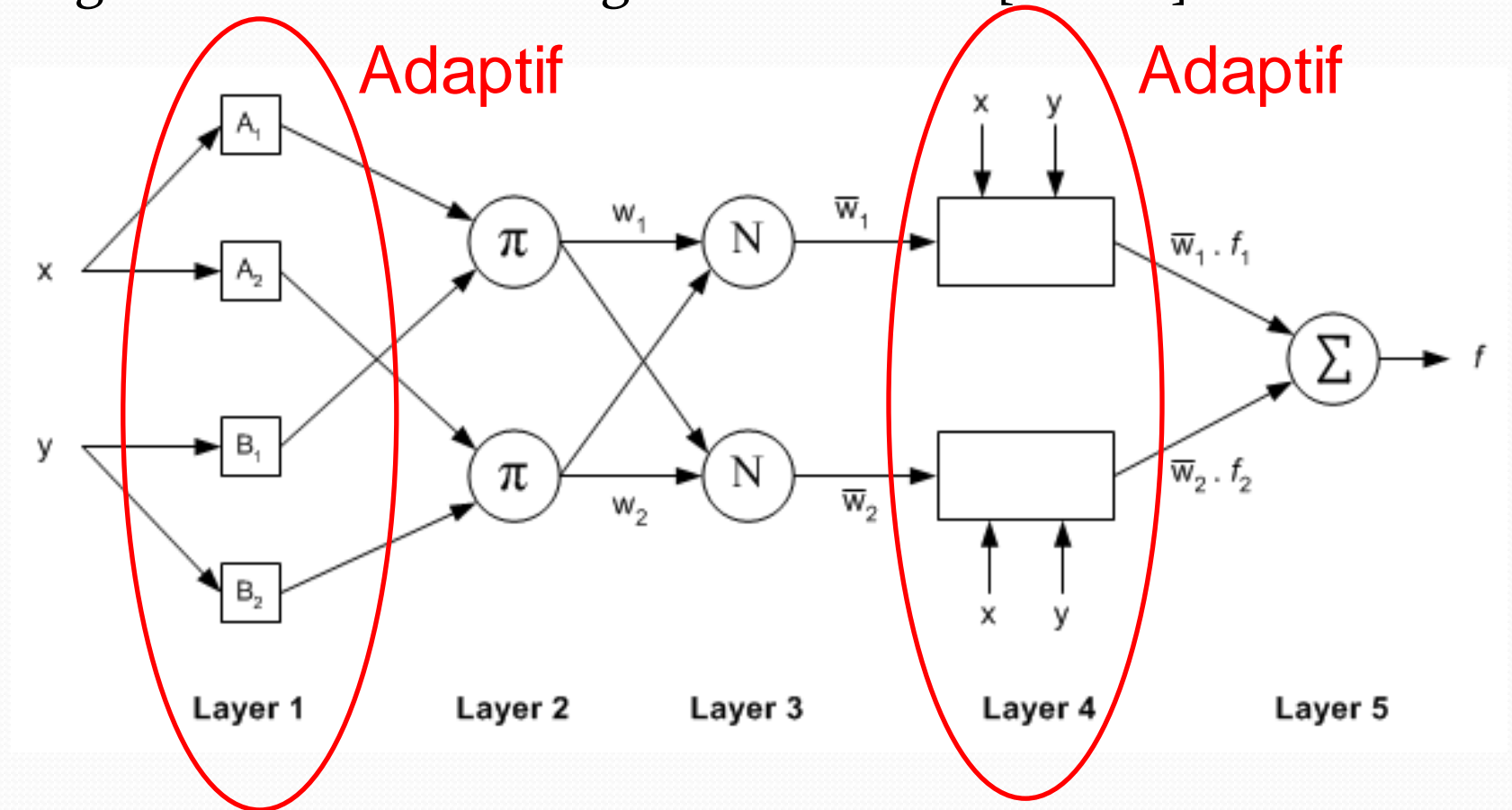


IPK \ Gaji	Kecil	Sedang	Besar	Sangat Besar
Buruk	Sedang	Rendah	Rendah	Rendah
Cukup	Tinggi	Tinggi	Sedang	Rendah
Bagus	Tinggi	Tinggi	Tinggi	Sedang



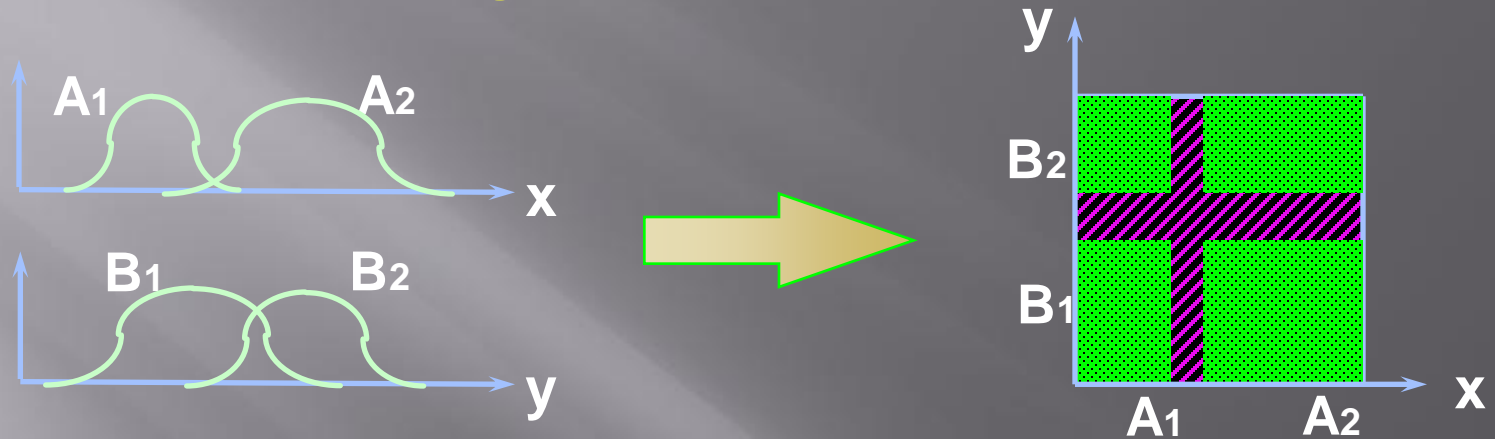
# ANFIS (Adaptive Network-based Fuzzy Inference System)

Arsitektur jaringan ANFIS yang berhubungan dengan Sugeno model di atas digambarkan sbb [TETo1]:

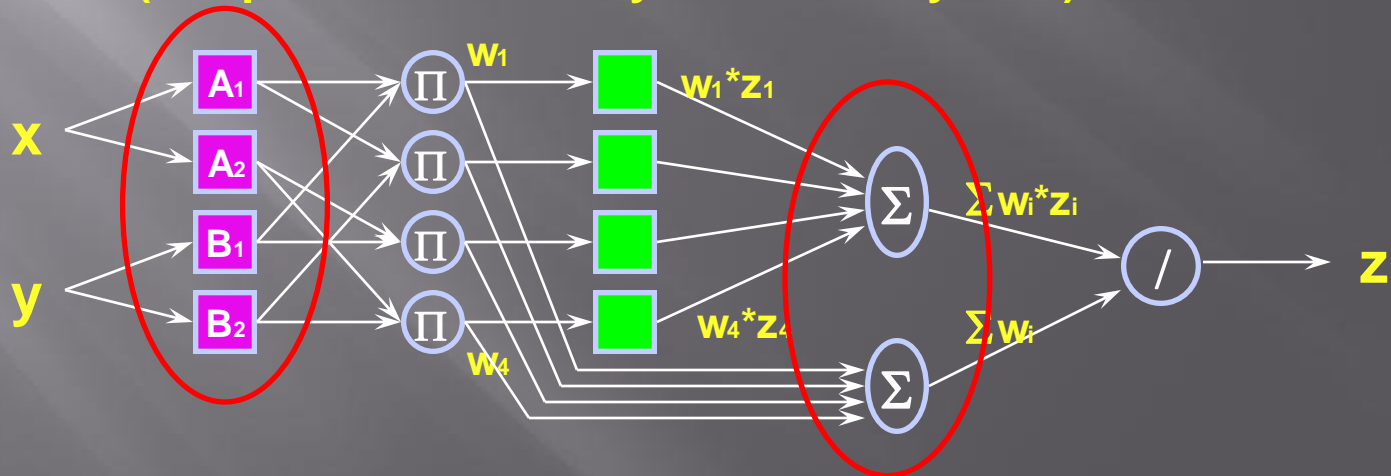




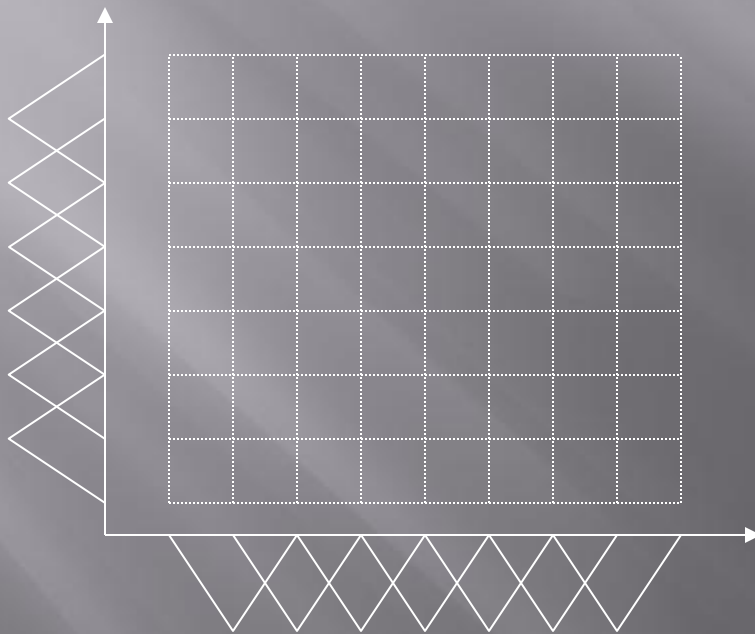
- Input space partitioning



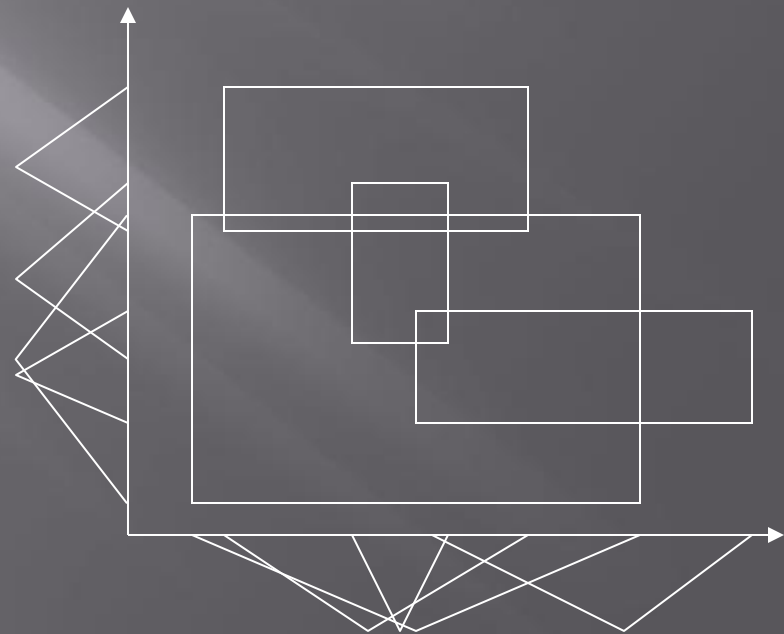
- ANFIS (Adaptive Neuro-Fuzzy Inference System)

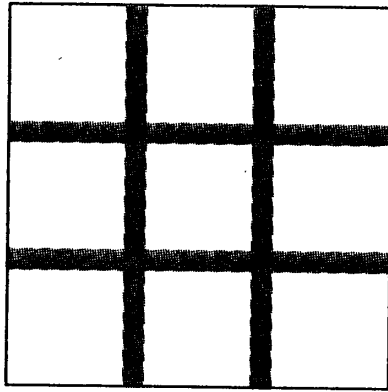


Regular grid

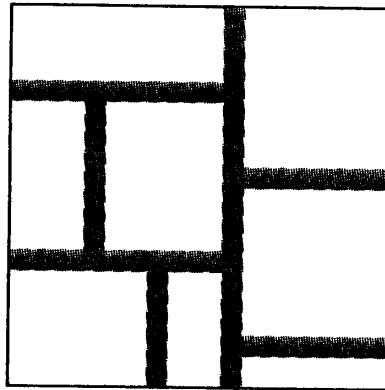


Independent functions

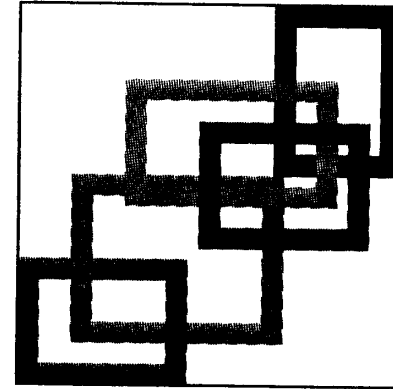




(a)



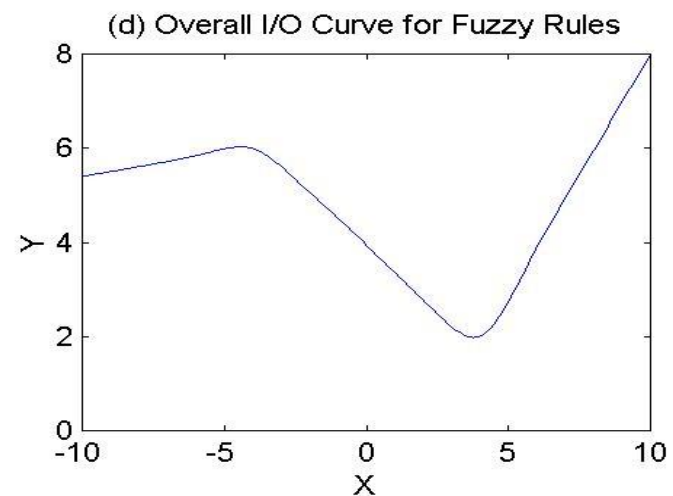
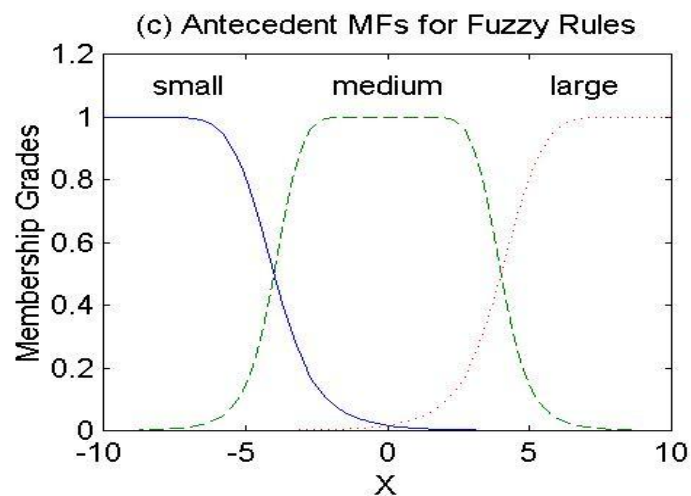
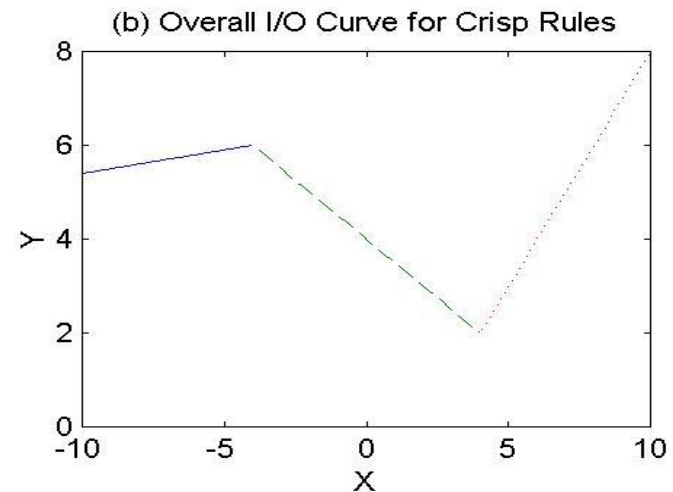
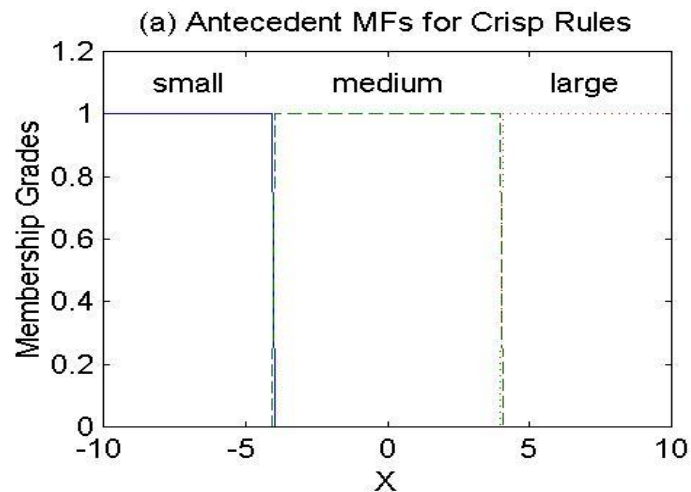
(b)



(c)

**Figure 4.13.** *Various methods for partitioning the input space: (a) grid partition; (b) tree partition; (c) scatter partition.*

If  $X$  is small then  $Y = 0.1X + 6.4$   
If  $X$  is medium then  $Y = -0.5X + 4$   
If  $X$  is large then  $Y = X - 2$



## ANFIS (Adaptive Network-based Fuzzy Inference System)

- **Layer 1:** Dinotasikan  $O_{l,i}$  (output dari node  $i$  pada layer  $l$ ). Masing-masing node pada layer  $i$  adalah *adaptive unit* dengan output:

$$O_{1,i} = \mu_{A_i}(x), \quad i = 1,2$$

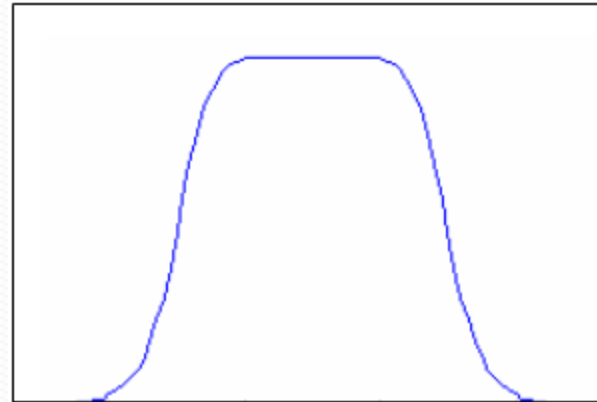
$$O_{1,i} = \mu_{B_i}(y), \quad i = 3,4$$

dimana  $x$  dan  $y$  adalah nilai-nilai input untuk *node* tersebut dan  $A_i$  atau  $B_{i-2}$  adalah himpunan *fuzzy*. Jadi, masing-masing node pada layer 1 berfungsi membangkitkan derajat keanggotaan bagian *premise*.

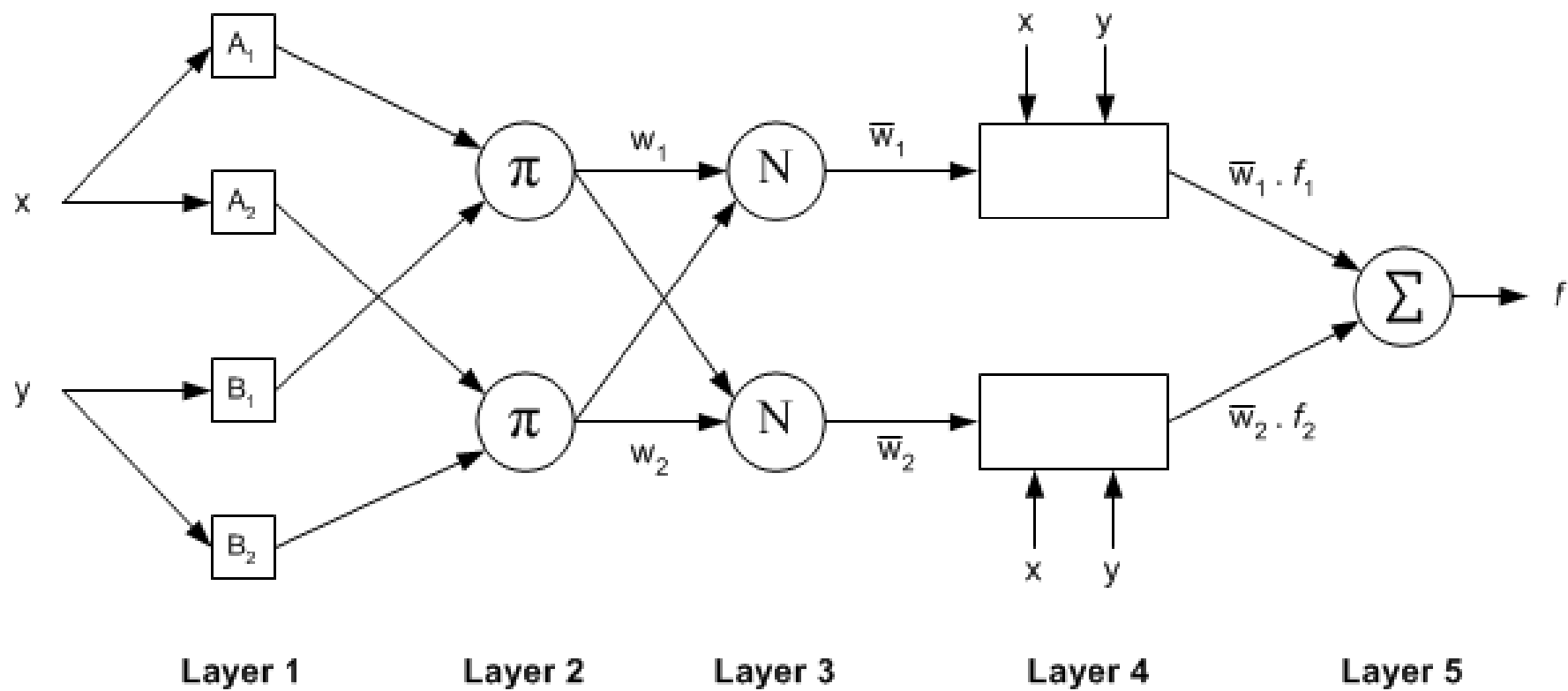
# ANFIS (Adaptive Network-based Fuzzy Inference System)

Misalkan, pada layer 1, kita menggunakan fungsi *Bell* sebagai fungsi keanggotaan  $\mu$  dengan rumus dan grafik sebagai berikut:

$$bell(x; a, b, c) = \frac{1}{\left(1 + \left|\frac{x - c}{a}\right|^{2b}\right)}$$



Dimana  $a$ ,  $b$ , dan  $c$ , yang biasa disebut sebagai *premise parameters*, sangat menentukan kemiringan fungsi *Bell* tersebut. Parameter  $b$  harus bernilai positif. Jika  $b$  bernilai negatif, maka fungsi *Bell* menjadi terbalik.



# ANFIS (Adaptive Network-based Fuzzy Inference System)

- **Layer 2:** Dinotasikan  $\pi$ . Setiap node pada layer ini berfungsi untuk menghitung *firing strength* dari setiap *rule* sebagai *product* dari semua input yang masuk atau operator **t-norm** (*triangular norm*):

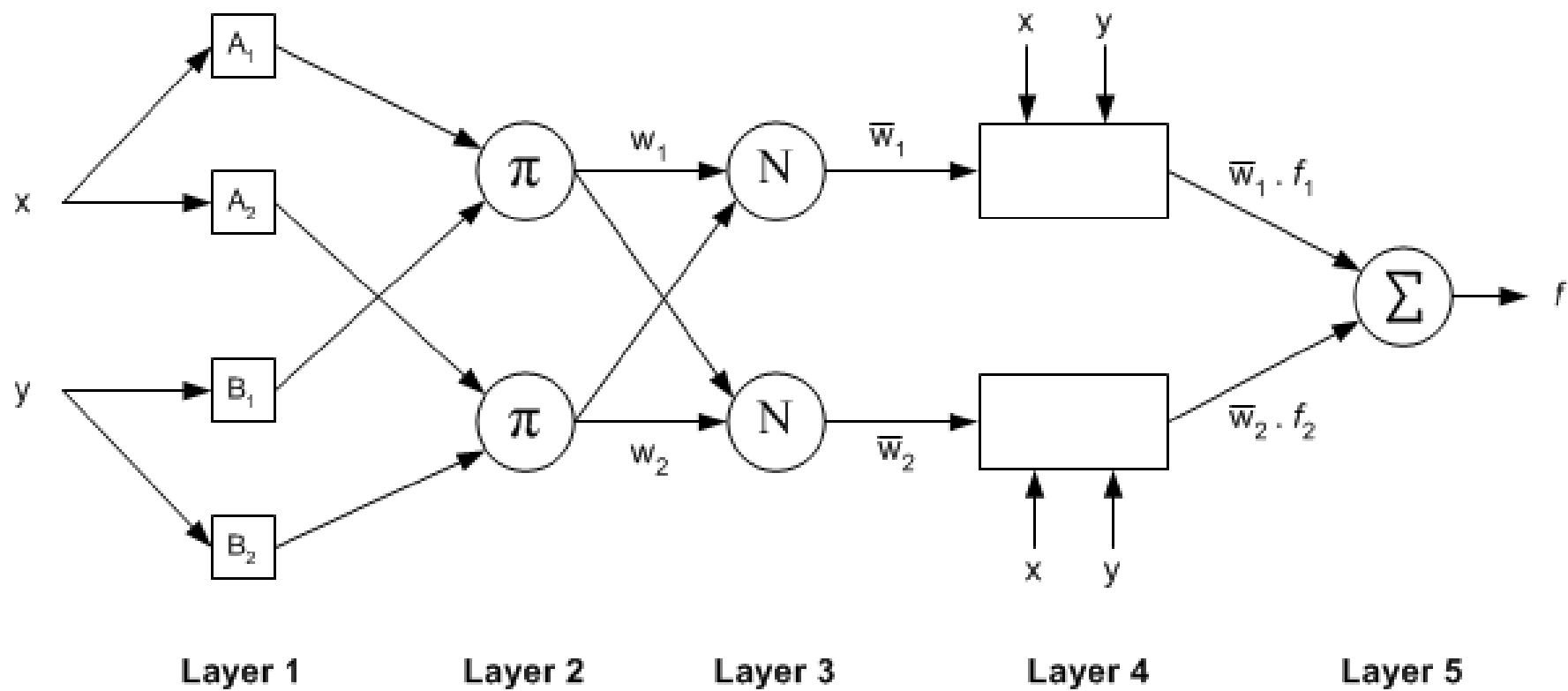
$$O_{2,i} = w_i = \mu_{A_i}(x) \Delta \mu_{B_i}(y), \quad i = 1, 2$$

Sehingga

$$w_1 = \mu_{A_1}(x) \text{ AND } \mu_{B_1}(y)$$

$$w_2 = \mu_{A_2}(x) \text{ AND } \mu_{B_2}(y)$$

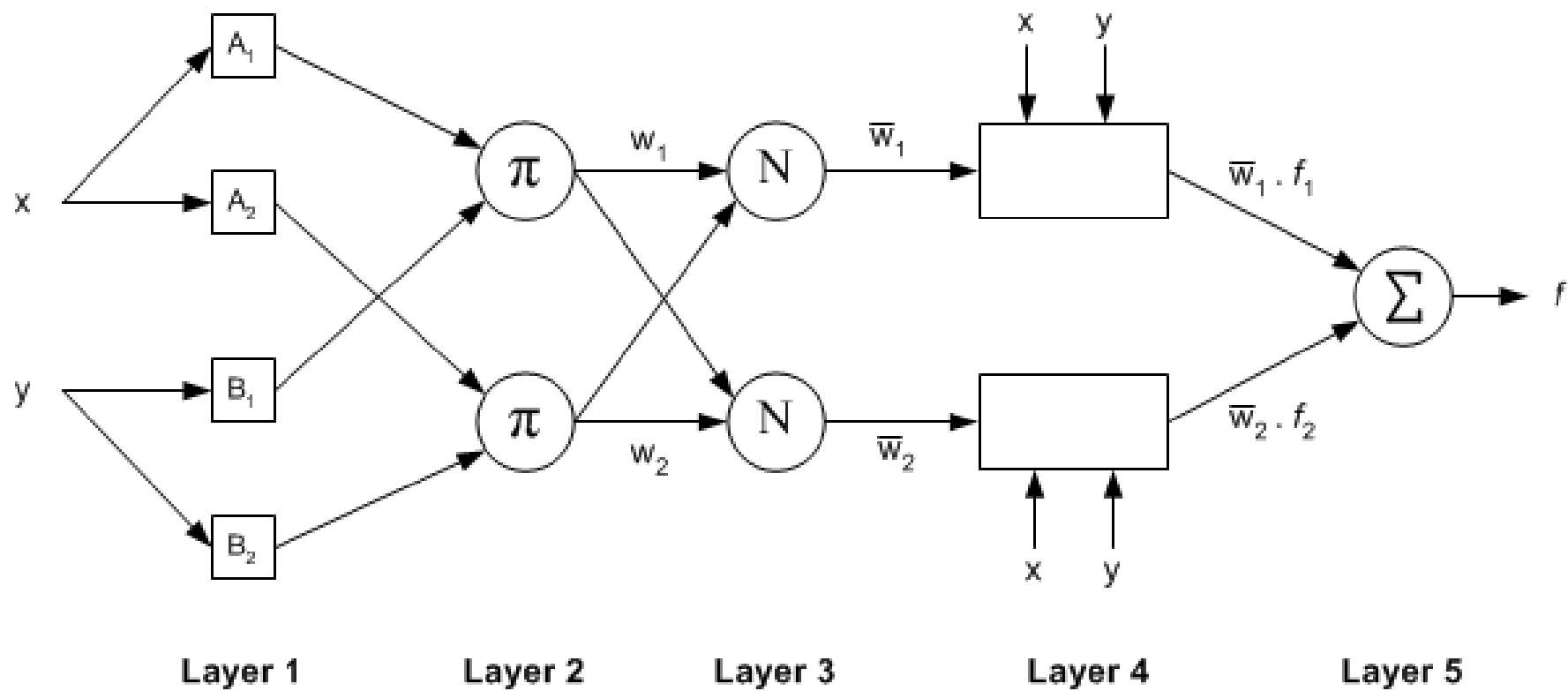




# ANFIS (Adaptive Network-based Fuzzy Inference System)

- **Layer 3:** Dilambangkan dengan  $N$ . Berfungsi untuk menghitung rasio dari *firing strength* dari rule ke- $i$  terhadap total *firing strength* dari semua rule:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2.$$



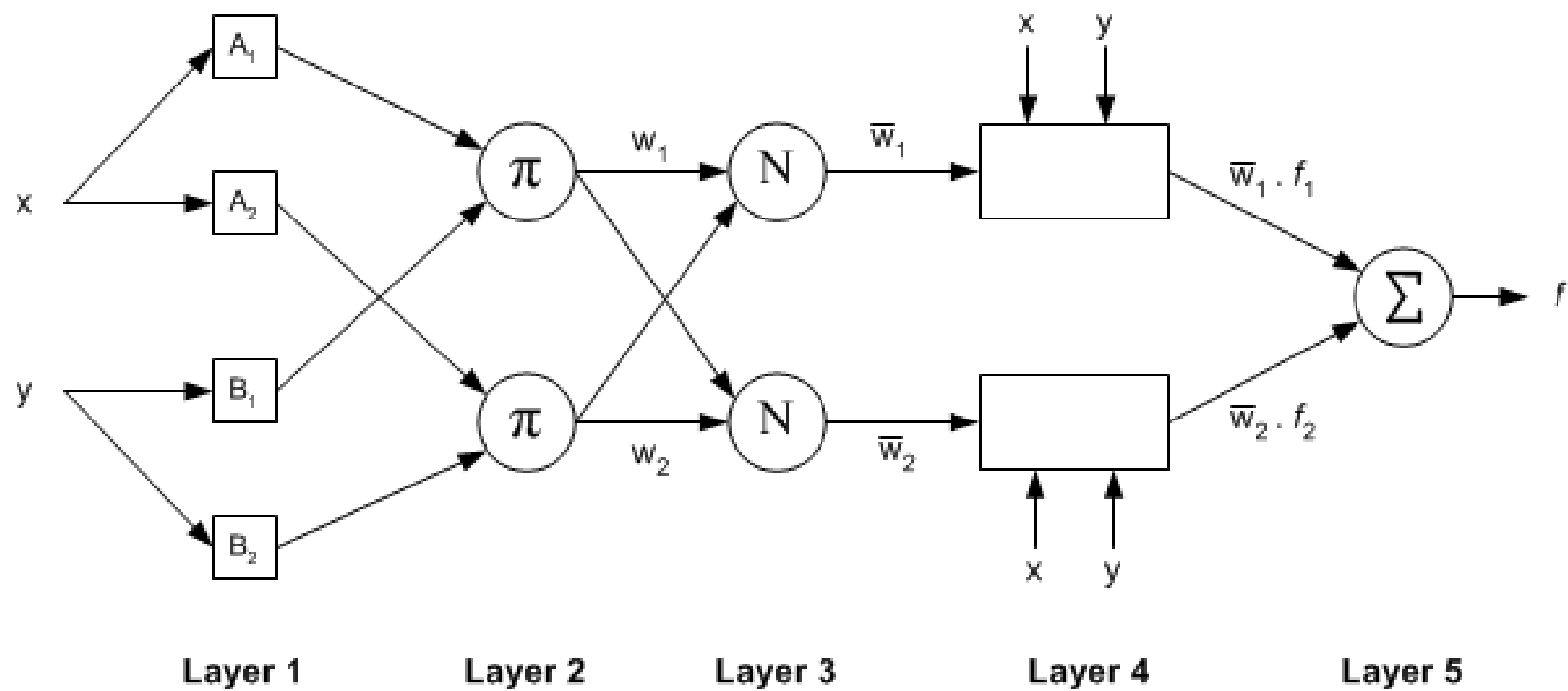
## ANFIS (Adaptive Network-based Fuzzy Inference System)

- **Layer 4:** Setiap node pada layer ini berfungsi sebagai:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i)$$

$\bar{w}_i$  adalah output dari layer 3

$\{p_i x + q_i y + r_i\}$  adalah himpunan parameter pada fuzzy dengan model *first-order* Sugeno.



## ANFIS (Adaptive Network-based Fuzzy Inference System)

- **Layer 5:** Satu node tunggal yang dilambangkan  $\Sigma$  pada layer ini berfungsi **mengagregasikan seluruh output** (yang didefinisikan sebagai **penjumlahan** dari semua sinyal yang masuk):

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

# Learning pada ANFIS

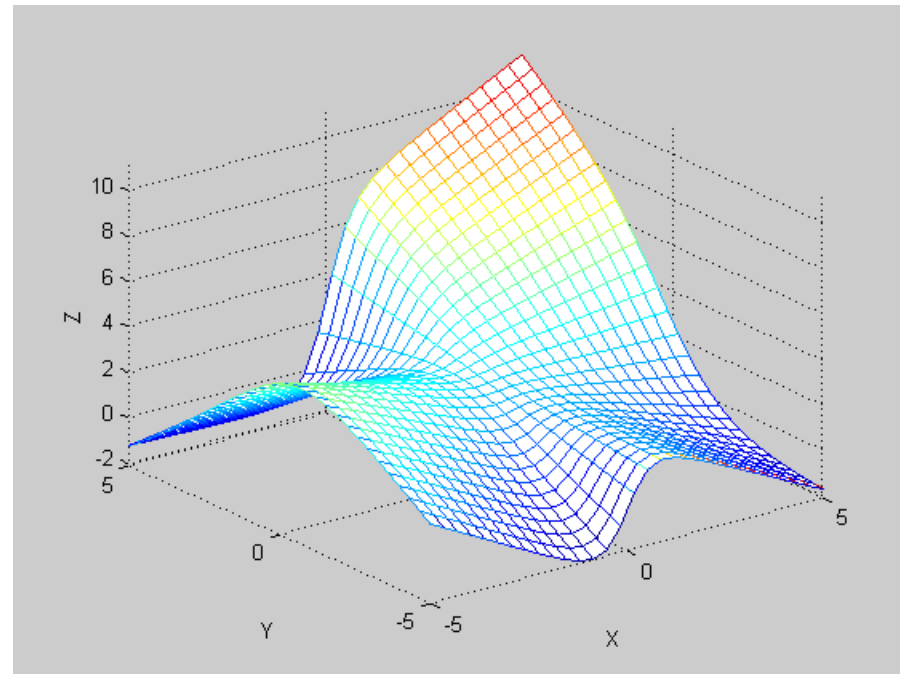
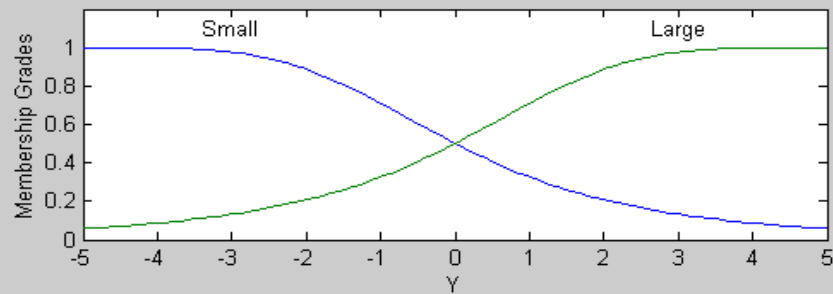
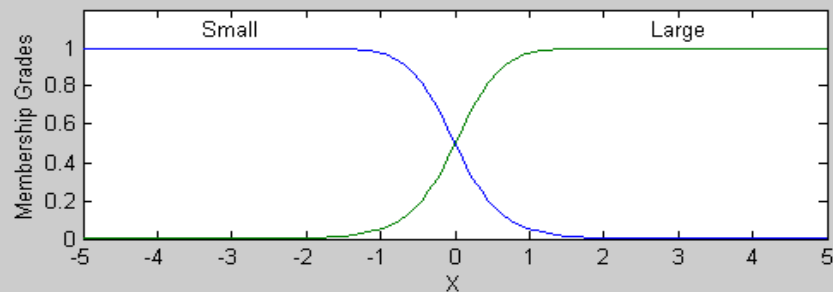
- Algoritma *learning* pada ANFIS adalah *hybrid supervised method* yang berbasis pada dua metode: *least-squares* dan *gradient descent*.
- Pada tahap maju, sinyal-sinyal merambat maju sampai layer 4 dan *consequent parameters* di-update menggunakan metode *least-square*
- Pada tahap mundur, sinyal-sinyal *error* dirambatkan mundur dan *premise parameters* di-update menggunakan *gradient descent*.

If  $X$  is small and  $Y$  is small then  $z = -x+y+1$

If  $X$  is small and  $Y$  is large then  $z = -y+3$

If  $X$  is large and  $Y$  is small then  $z = -x+3$

If  $X$  is large and  $Y$  is large then  $z = x+y+2$

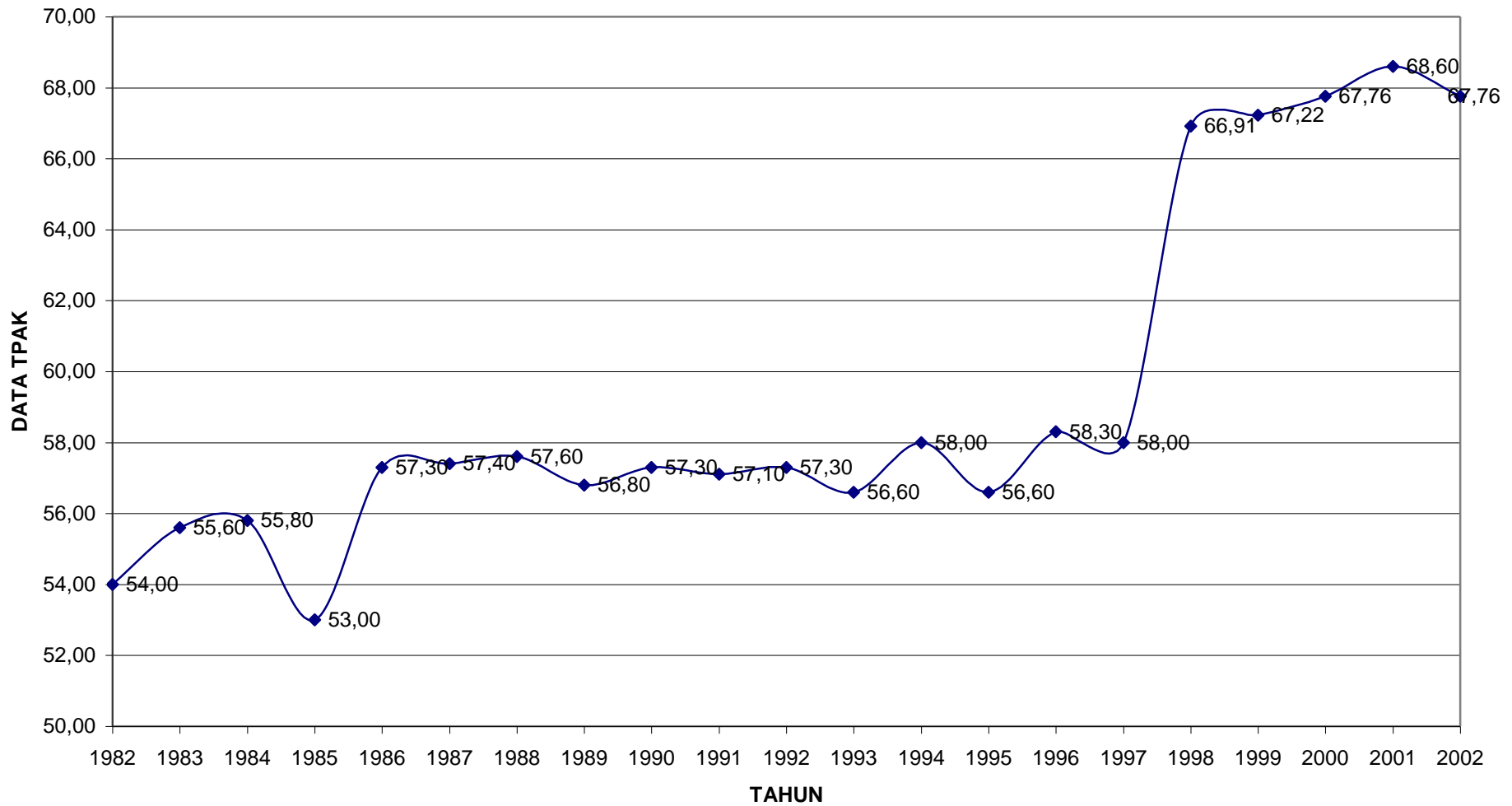


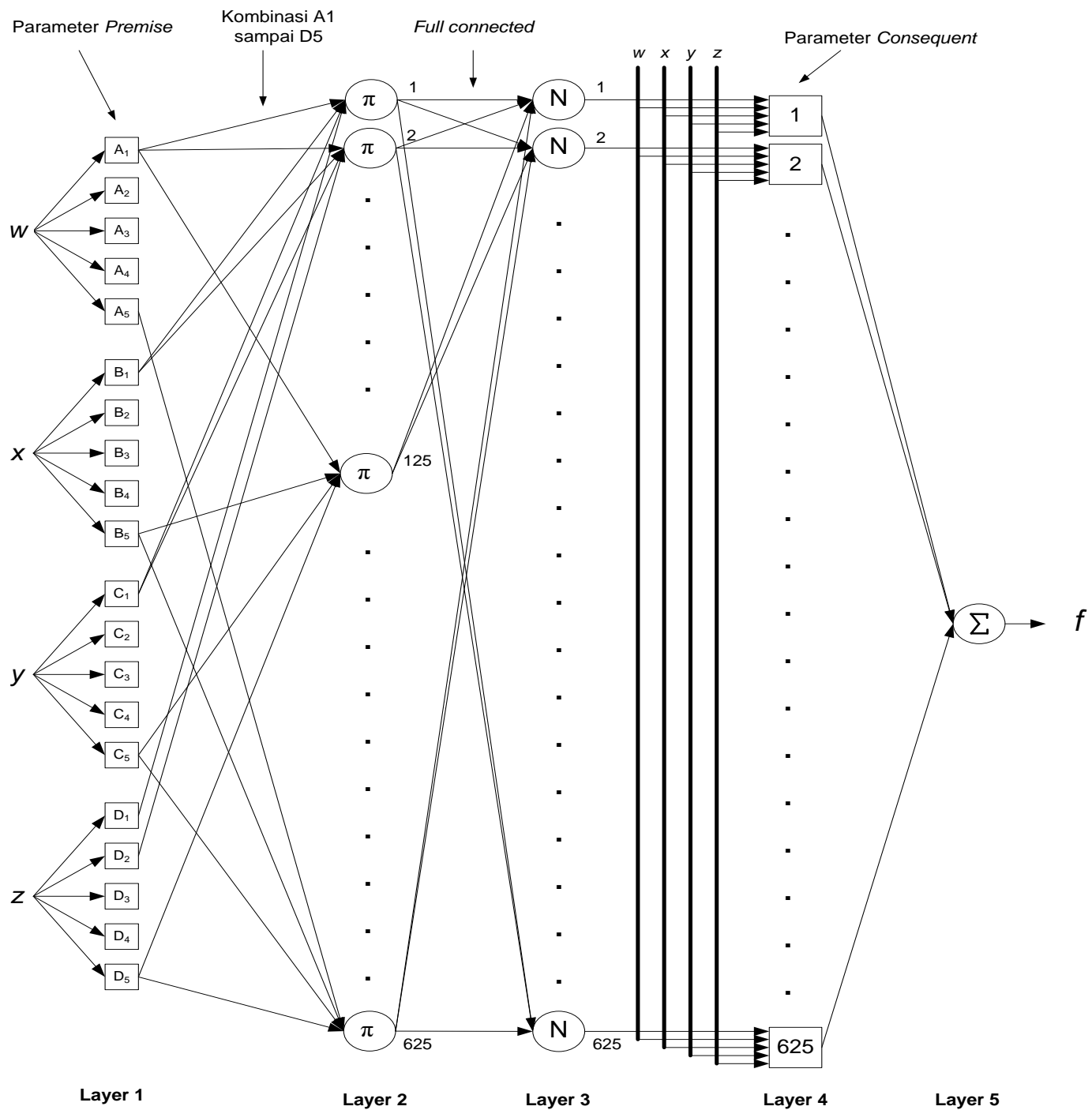


# Prediksi Tingkat Partisipasi Angkatan Kerja

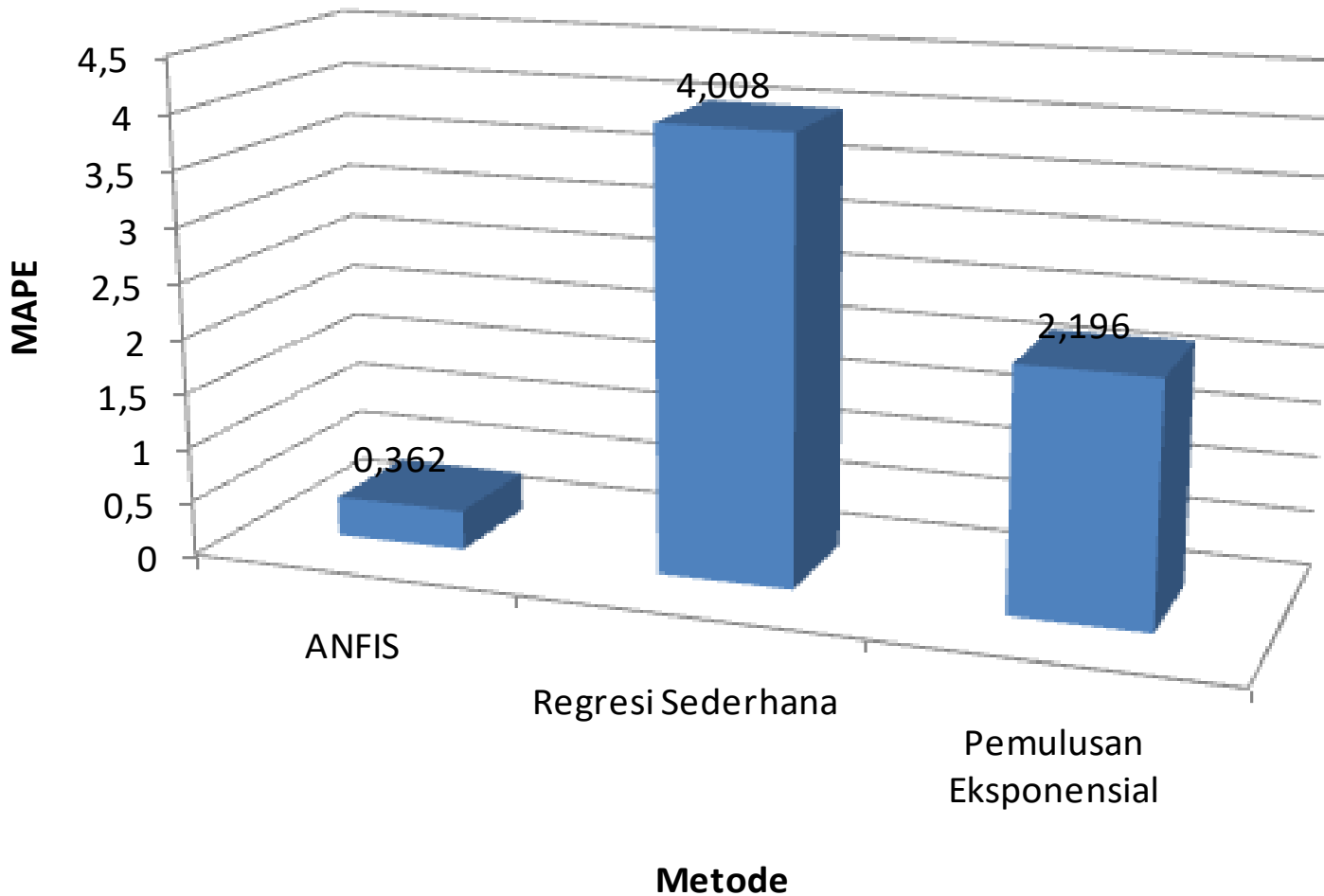
- Data dari Badan Pusat Statistik
- Data time series per tahun: 1982 – 2002 (21 sampel)
- Bagaimana penggunaan ANFIS?

**GRAFIK POLA DATA TPAK NASIONAL PERIODE 1982-2002**





# MAPE untuk ANFIS, Regresi Sederhana, dan Pemulusan Eksponensial



# Kesimpulan

- Sinergi ANN dan FS bisa menghasilkan sistem *soft computing* yang lebih baik.
- Implementasi *Neuro-Fuzzy* bisa dilakukan menggunakan berbagai macam cara tergantung pada masalah yang dihadapi.
- NEFPROX
- ANFIS

# Referensi

- [SUYo8] Suyanto, 2008, “Soft Computing: Membangun Mesin Ber-IQ Tinggi”, Informatika, Bandung Indonesia. ISBN: 978-979-1153-49-2.
- [TETo1] Tettamanzi A., Tomassini M., ”Soft Computing”. Springer-Verlag Berlin Heidelberg, 2001. Printed in Germany.
- [MIT97] Mitchell M. Tom. 1997. ”Machine Learning”. McGraw-Hill International Editions. Printed in Singapore.