

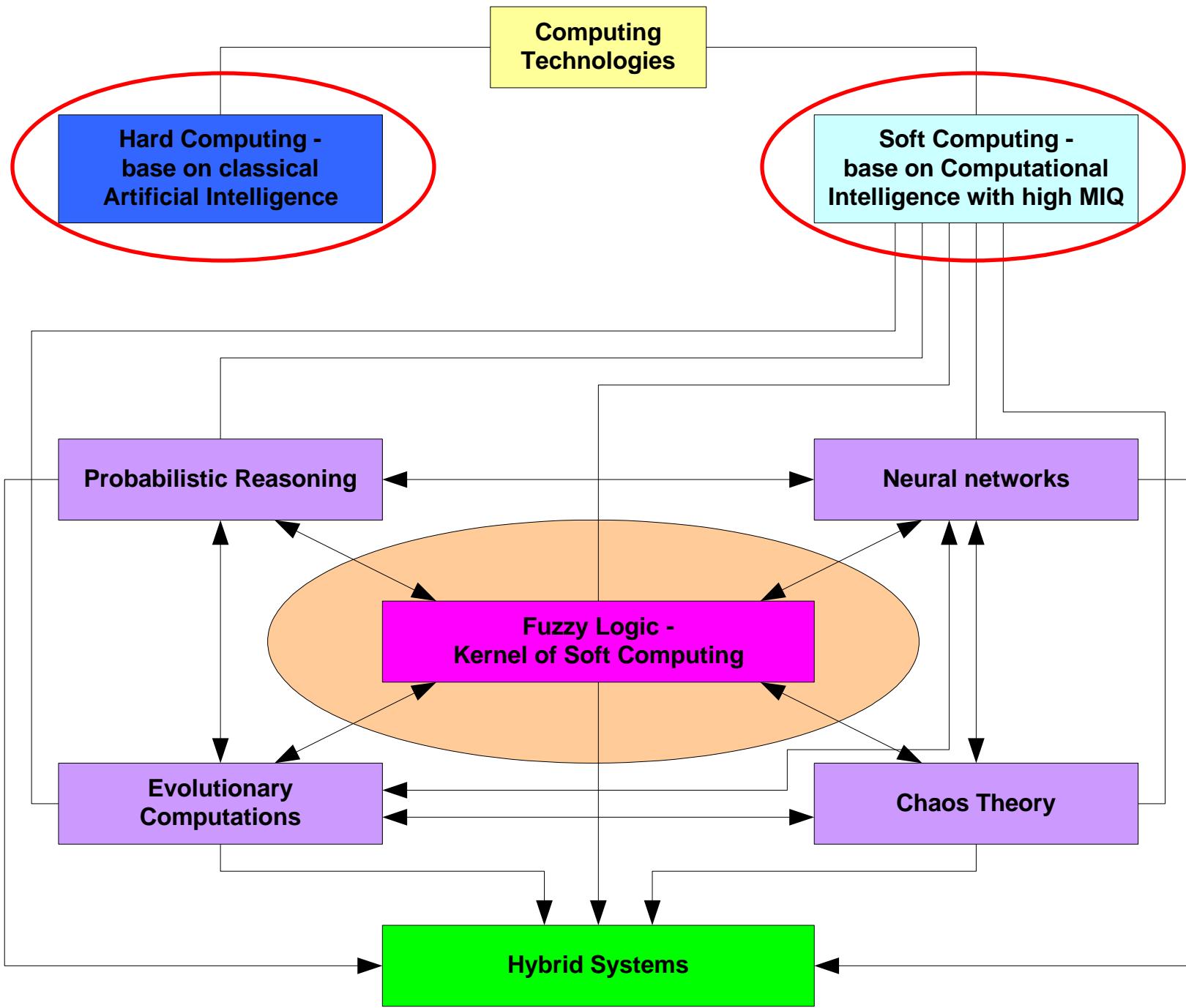
Pendahuluan

Dr. Suyanto, S.T., M.Sc.
HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

Apa itu *Soft Computing*

- **SC** is an *evolving collection of methodologies*, which aims to exploit tolerance for *imprecision, uncertainty*, and *partial truth* to achieve *robustness, tractability (TR)* and *low cost (LC)* [Zadeh, Lotfi A., 2006]
- **Tujuan SC:** terbentuknya *High Machine Intelligence Quotient* (HMIQ), sistem yang mampu mengolah informasi seperti cara berpikir manusia, mampu menyelesaikan permasalahan non-linier dan tidak ada model matematisnya (TR), serta dapat diimplementasikan dengan biaya rendah (LC).
- **Komponen utama SC:** *Fuzzy Logic*, *Neural Network*, dan *Probabilistic Reasoning*. Kemudian diikuti oleh *Belief Network*, *Genetic Algorithms*, *Chaos Theory* dan *Learning Theory*.
- **Yang terpenting dalam SC:** satu komponen dengan komponen lainnya saling melengkapi dan berkontribusi untuk menyelesaikan suatu permasalahan.





Teknik Dasar SC

- Awalnya, teknik dasar SC hanya empat:
 - *Fuzzy Logic*
 - *Evolutionary Computation* (EC)
 - *Neuro Computing* atau *Artificial Neural Networks* (ANN)
 - *Probabilistic Computing*
- Para ahli menambahkan teknik lain yang bisa digolongkan ke dalam SC, diantaranya:
 - *Support Vector Machine*
 - *Rough Sets*
 - *Multivalued Logic*
 - *Chaos Computing*
 - *Immune Network Theory*

Masalah pada metode SC

- Seleksi data masukan dan keluaran
- Pada pendekatan ANN, harus tersedia data yang digunakan untuk melatih sistem.
- Pada pendekatan Fuzzy Logic, harus tersedia pengetahuan tentang rule.

Masalah pada metode SC

- Pada Algoritma Genetika, penentuan fungsi obyektif yang menguji kualitas individu menjadi titik penting dari mekanisme pencarian solusi.
- Seleksi arsitektur sistem yang bersifat coba-coba, belum tersedianya formulasi walaupun untuk satu jenis aplikasi.
- Suatu metode baik untuk aplikasi tertentu tapi bisa sangat buruk untuk aplikasi lainnya.

Solusi: *Hybrid Systems*

- Fuzzy Logic yang digabungkan dengan ANN atau Evolutionary Algorithm (EA)
- ANN yang digabungkan dengan Fuzzy Logic atau EA
- EA yang digabungkan dengan Fuzzy Logic atau ANN

Hybrid Fuzzy Systems

- **Evolving Fuzzy:** Fuzzy Logic Control yang dibangkitkan dan di-*tune* menggunakan EA
- **Fuzzy Neural Systems:** ANN yang dimodifikasi oleh Fuzzy System
- **Neural Fuzzy Systems:** Fuzzy Logic Control yang di-*tune* menggunakan ANN

Hybrid ANN Systems

- **Fuzzy Neural Systems:** ANN yang dimodifikasi oleh Fuzzy System
- **Evolving Neural Networks:** *Weights* dan/atau topologi (arsitektur) ANN yang di-*tune* menggunakan Evolutionary Algorithm (EA)

Hybrid EA Systems

- Parameter EA dikontrol menggunakan Fuzzy logic
- Parameter EA dikontrol menggunakan EA juga

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.

Fuzzy Systems

Dr. Suyanto, S.T., M.Sc.
HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

Logic [RUS95]

Jenis <i>logic</i>	Apa yang ada di dunia nyata	Apa yang dipercaya <i>Agent</i> tentang fakta
<i>Propositional logic</i>	fakta	benar/salah /tidak diketahui
<i>First-order logic</i>	fakta, objek, relasi	benar/salah /tidak diketahui
<i>Temporal logic</i>	fakta, objek, relasi, waktu	benar/salah /tidak diketahui
<i>Probability theory</i>	fakta	derajat kepercayaan [0,1]
<i>Fuzzy logic</i>	derajat kebenaran	derajat kepercayaan [0,1]

Peringatan:

1 dari 50 botol ini berisi cairan kimia mematikan yang warna dan rasanya sama dengan air mineral. Anda akan mati seketika jika meminumnya.

Probability

1

Peringatan:

Satu plastik cairan kimia mematikan dicampurkan ke dalam 50 botol ini secara tidak merata. Anda tidak akan mati jika cuma minum satu botol, tetapi anda akan menderita pusing ringan/berat.

Fuzziness

2

Fuzziness & Probability

- Banyak peneliti berbeda pendapat tentang **teori fuzzy** dan **teori probabilitas**
- Sebenarnya, kedua teori tersebut memang sama-sama untuk menangani masalah ketidakpastian
- Tetapi, perbedaannya adalah pada **jenis ketidakpastian** yang ditangani

Fuzzy Systems

- Ide dasar *fuzzy systems* adalah *fuzzy logic* dan *fuzzy sets*
- *Fuzzy logic* sudah lama dipikirkan oleh para filsuf Yunani kuno
- Plato: filsuf pertama yang meletakkan fondasi *fuzzy logic*
- Plato: “Terdapat area ketiga selain Benar dan Salah”
- Rumi: “Jauh di balik benar dan salah, ada sebuah padang terbuka. Kutemui kau di sana.”
- *Fuzzy logic* menghilang selama 2 milenium
- Muncul kembali pada era 1960-an
- Konsep *fuzzy logic* yang sangat sistematis pertama kali diusulkan oleh Lotfi A. Zadeh, the University of California, Berkeley, Amerika Serikat

Classical Sets

- Teori himpunan klasik: suatu himpunan secara intuitif adalah setiap kumpulan elemen-elemen
- Himpunan klasik dikenal juga sebagai ***crisp set***
- *Crisp = clear and distinct* [Oxford 1995]
- *Crisp set* : himpunan yang membedakan anggota dan non anggotanya dengan batasan yang jelas

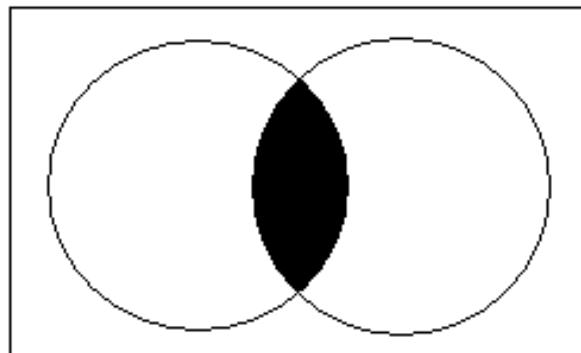
Contoh

$$A = \{x \mid x \text{ bilangan bulat}, x > 6\}$$

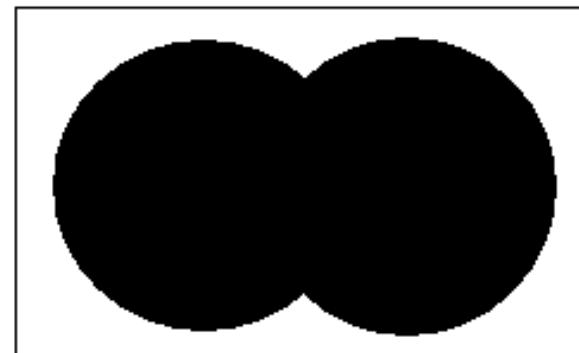
Anggota himpunan A adalah 7, 8, 9, ...

Bukan anggota A adalah 6, 5, 4, ...

Intersection, union, complement, difference



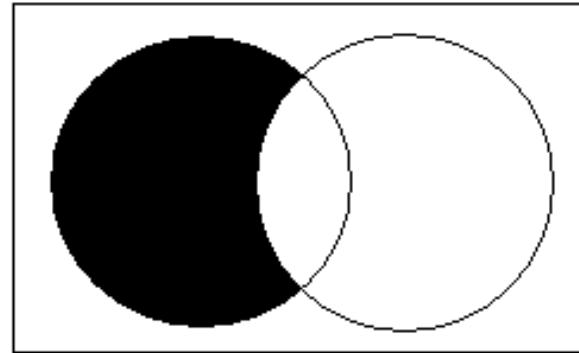
(a)



(b)



(c)



(d)

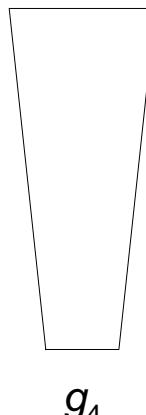
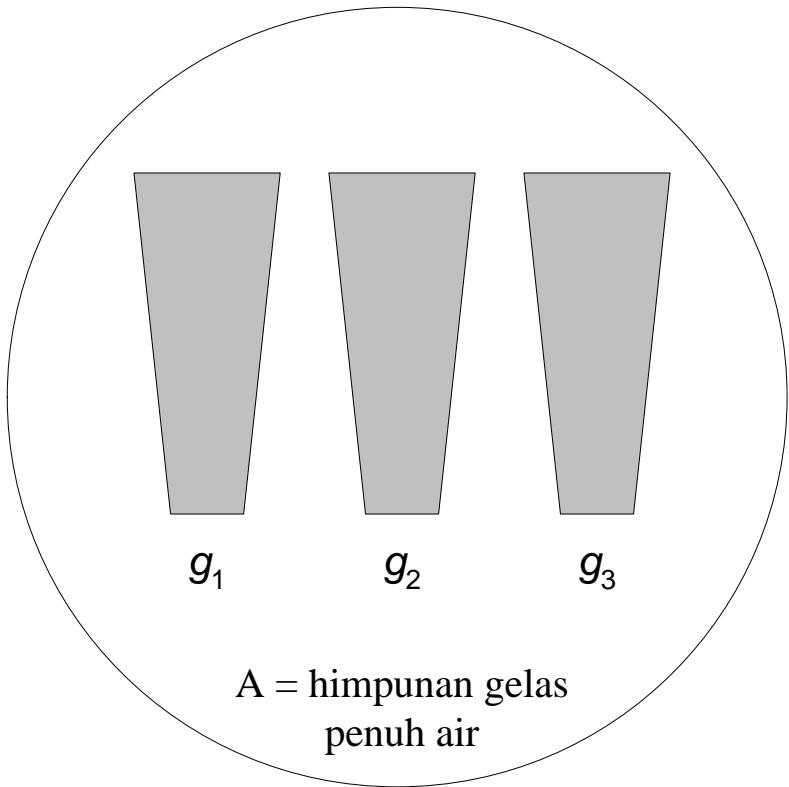
Excluded Middle Laws

$$A \cup \overline{A} = U$$

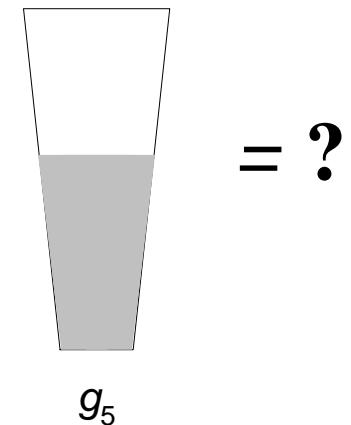
Law of Contradiction

$$A \cap \overline{A} = \emptyset$$

Semesta U



Gelas kosong
termasuk \bar{A}



Gelas yang berisi air
setengah bagian tidak
termasuk A maupun \bar{A}

Aku adalah pembohong.
Jangan percaya padaku.

Pernyataan P dan negasinya
sama-sama benar. $T(P) = T(\bar{P})$



Fungsi Karakteristik

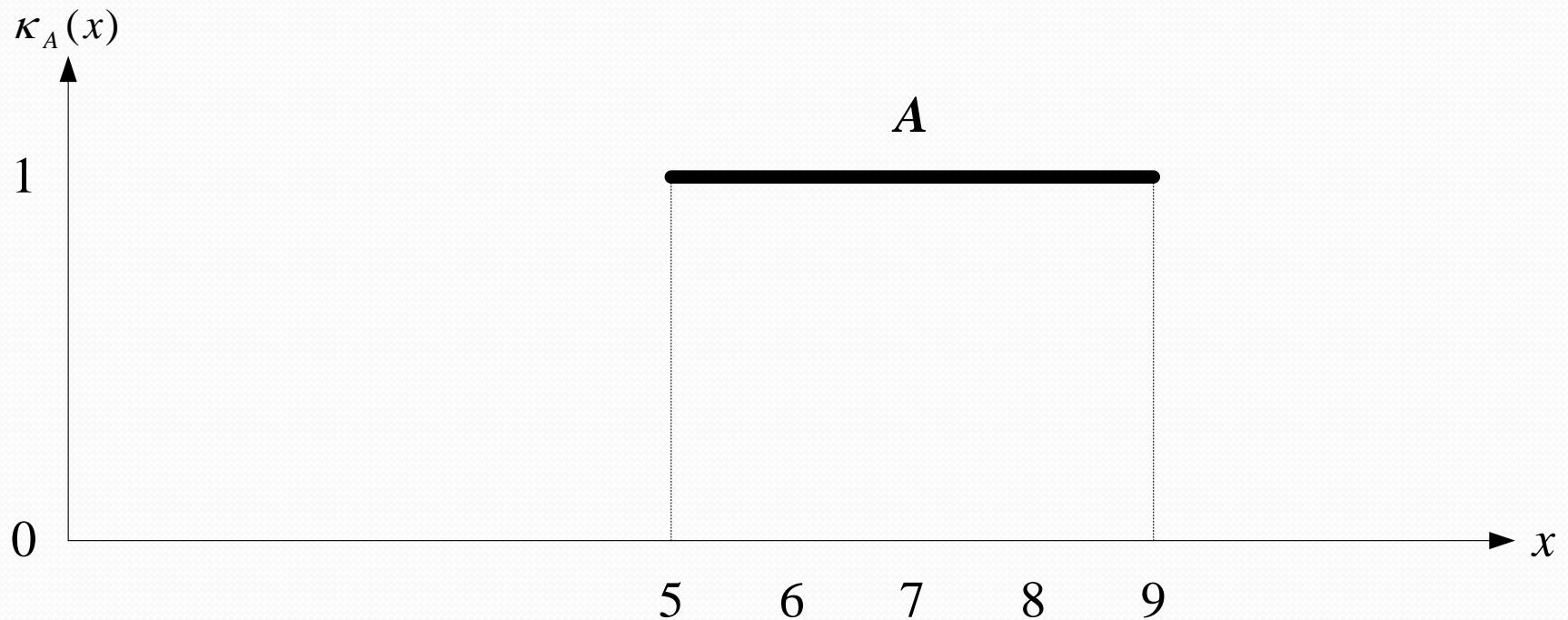
Fungsi karakteristik dari himpunan A adalah suatu pemetaan

$$\kappa_A : U \rightarrow \{0, 1\}$$

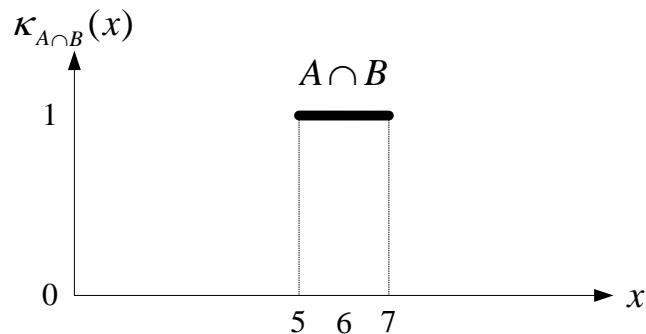
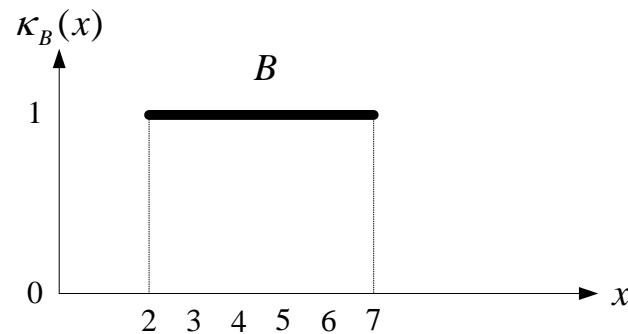
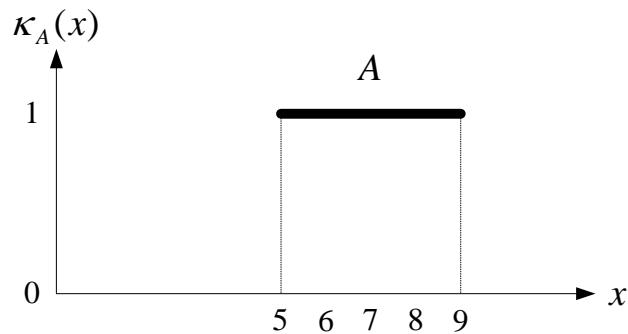
sedemikian hingga, untuk semua x ,

$$\kappa_A(x) = \begin{cases} 1, & \text{jika } x \in A; \\ 0 & \text{untuk kasus lainnya.} \end{cases}$$

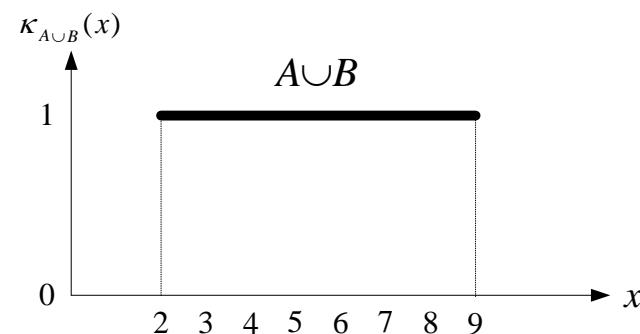
Fungsi Karakteristik *Classical Set*



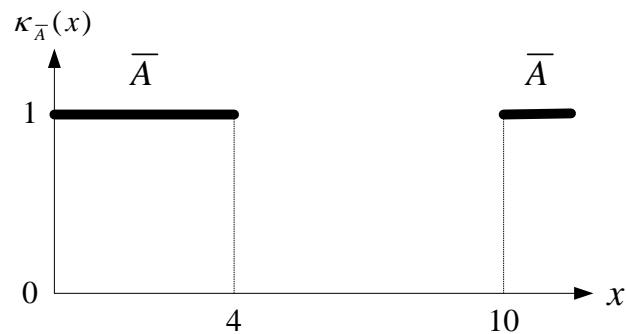
$A = \text{himpunan klasik semua bilangan bulat positif lebih dari } 4 \text{ dan kurang dari } 10 \text{ atau } \{5, 6, \dots, 9\}.$



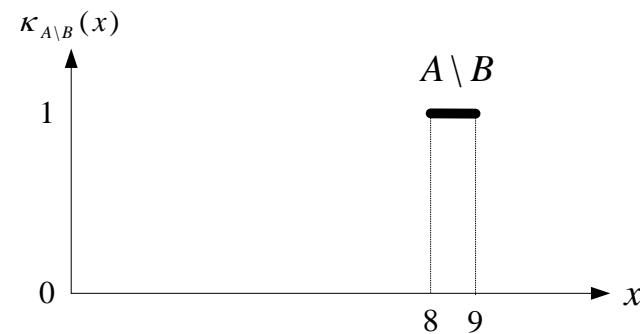
Intersection



Union



Complement



Difference

Kasus 1: Pemberian Beasiswa

Mahasiswa	IPK	Gaji Ortu (Rp/bulan)
A	3,00	10 juta
B	2,99	1 juta

if $IPK \geq 3,00$ and $Gaji \leq 10$ juta
then Dapat Beasiswa

Fuzzy Sets

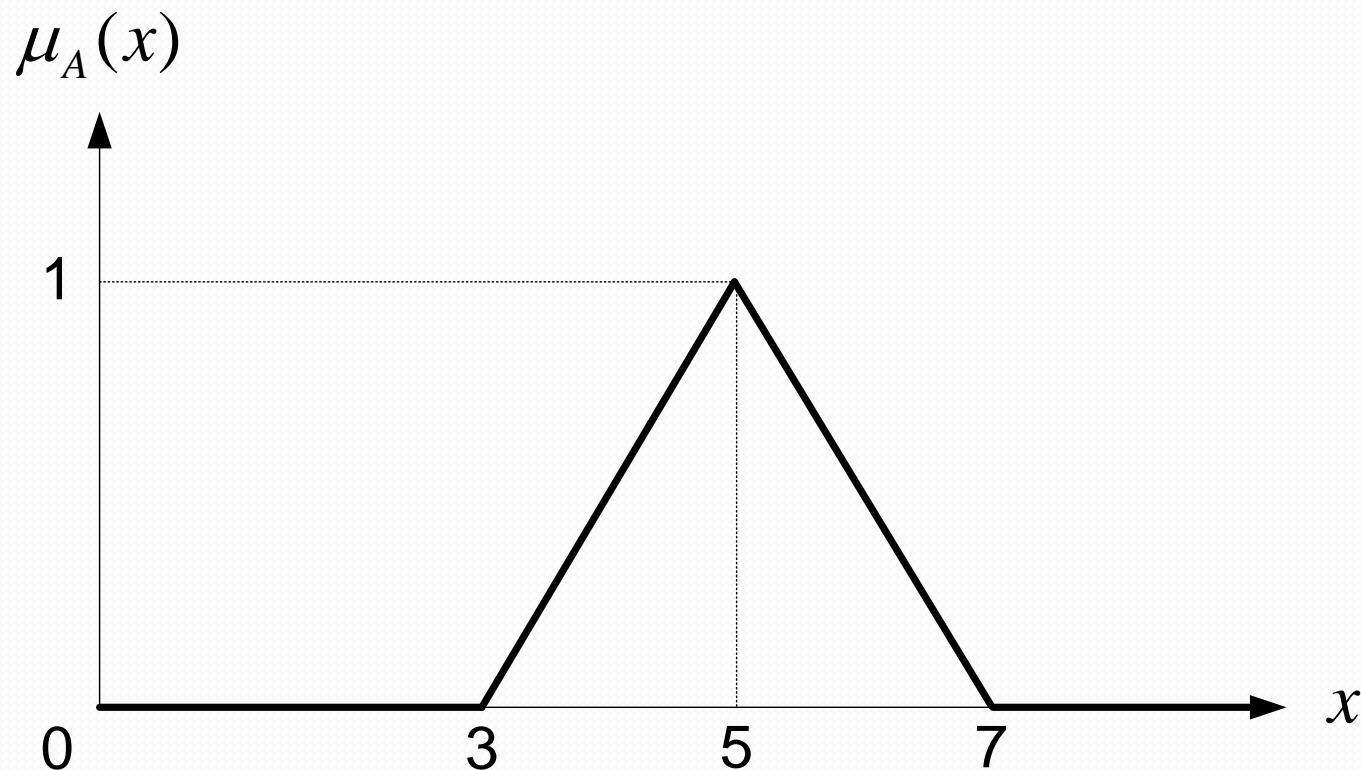
- Digunakan untuk penalaran yang lebih manusiawi.
- Misalkan U adalah *universe of discourse* (semesta pembicaraan) dan x adalah anggota U .
- Suatu *fuzzy set* A di dalam U didefinisikan sebagai suatu *membership function* atau fungsi keanggotaan, yang memetakan setiap objek di U menjadi suatu nilai real dalam interval $[0, 1]$.
- Nilai-nilai menyatakan derajat keanggotaan x di dalam A .

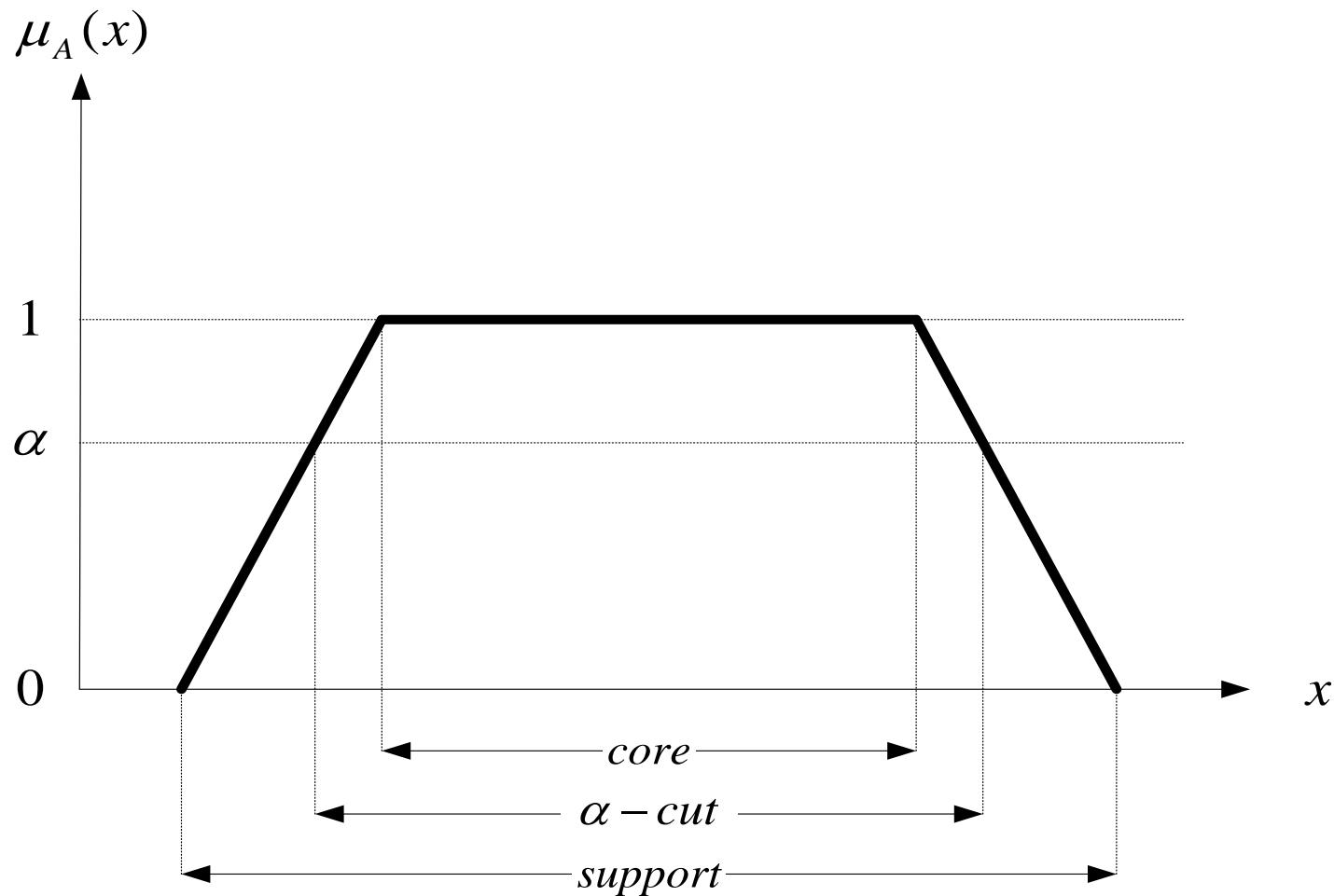
Suhu ($^{\circ}$ C)	Dingin	Hangat	Panas
5	1	0,1	0
15	0,9	0,8	0
25	0,5	1	0,6
35	0,1	0,6	0,9
45	0	0,2	1

Fuzzy Set

- Dingin = {5, 15, 25, 35} dan derajat keanggotaannya dinyatakan oleh $\mu_{Dingin} = \{1; 0,9; 0,5; 0,1\}$
- Hangat = {5, 15, 25, 35, 45} dan derajat keanggotaannya dinyatakan oleh $\mu_{Hangat} = \{0,1; 0,8; 1; 0,6; 0,2\}$
- Panas = {25, 35, 45} dan derajat keanggotaannya dinyatakan oleh $\mu_{Panas} = \{0,6; 0,9; 1\}$

Graphical Representation

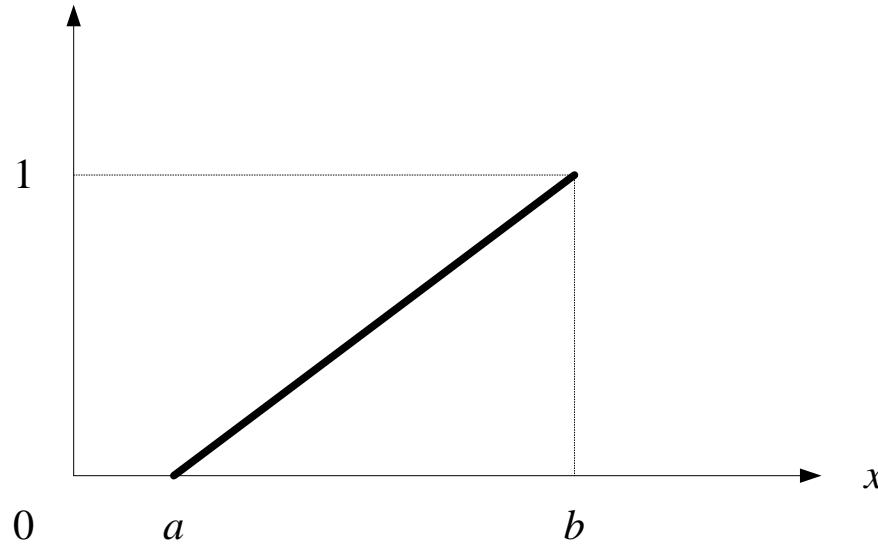




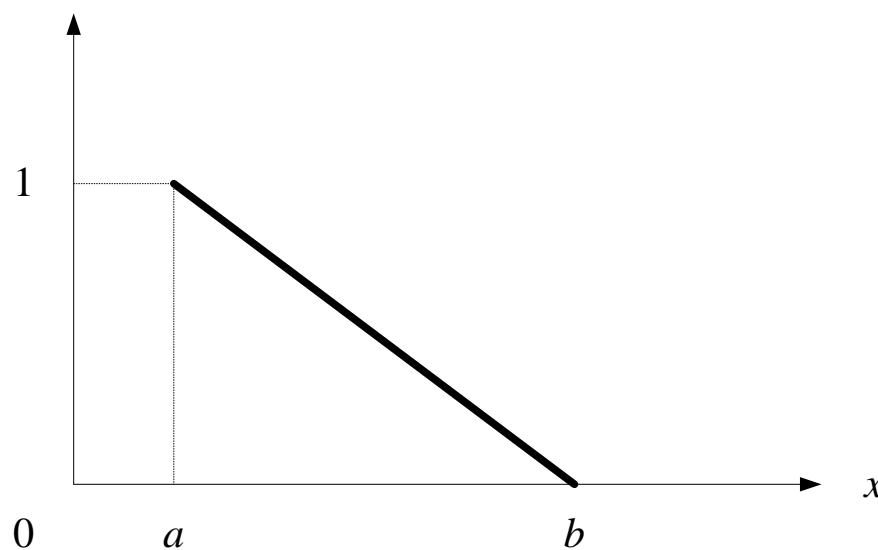
Fungsi keanggotaan μ_A dengan *core*, α - *cut* dan *support*

Bentuk Fungsi Keanggotaan

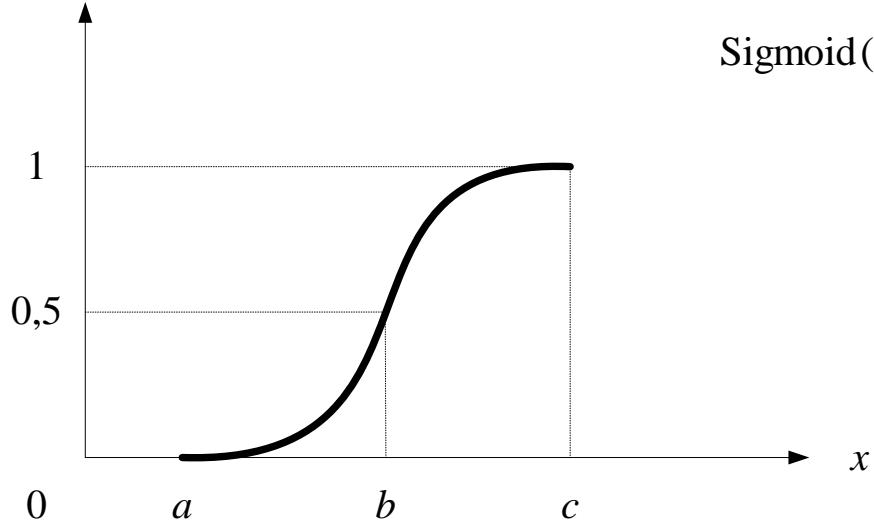
- Fungsi Linier
- Fungsi *Sigmoid*
- Fungsi Segitiga
- Fungsi Trapezium
- Fungsi Berbentuk *Bell*:
 - *Phi*
 - *Beta*
 - *Gauss*

$\mu(x)$ 

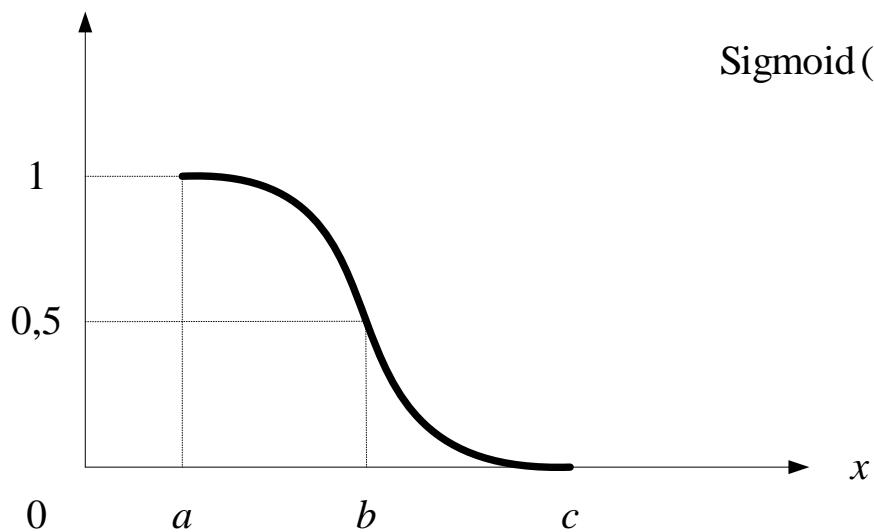
$$\text{LinierNaik}(x, a, b) = \begin{cases} 0, & x \leq a \\ (x - a)/(b - a), & a < x \leq b \end{cases}$$

 $\mu(x)$ 

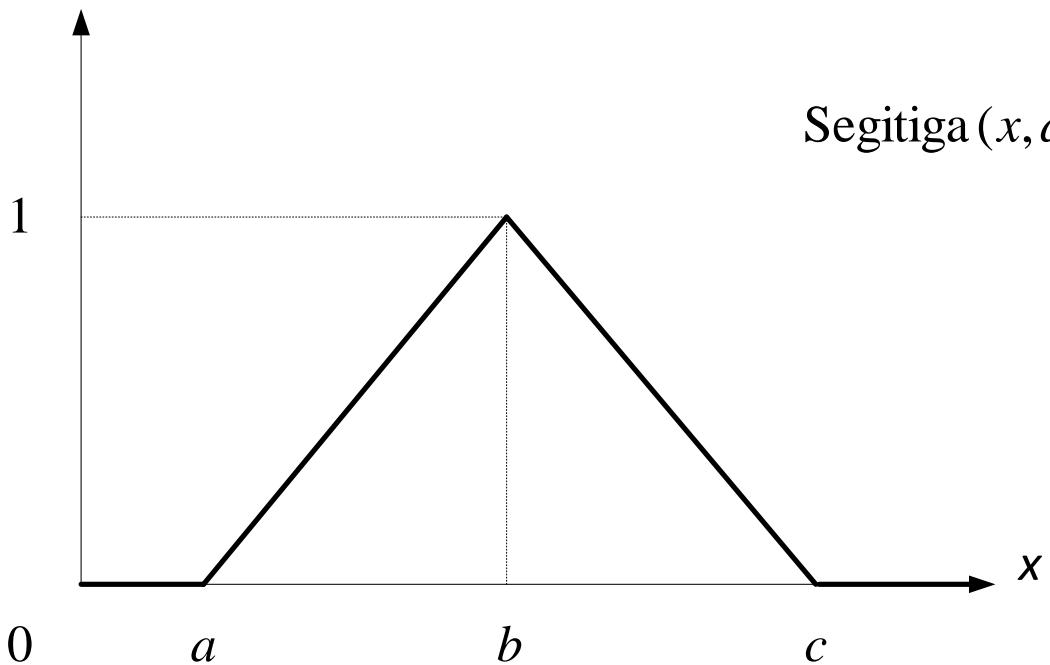
$$\text{LinierTurun}(x, a, b) = \begin{cases} (b - x)/(b - a), & a \leq x < b \\ 0, & x \geq b \end{cases}$$

$\mu(x)$ 

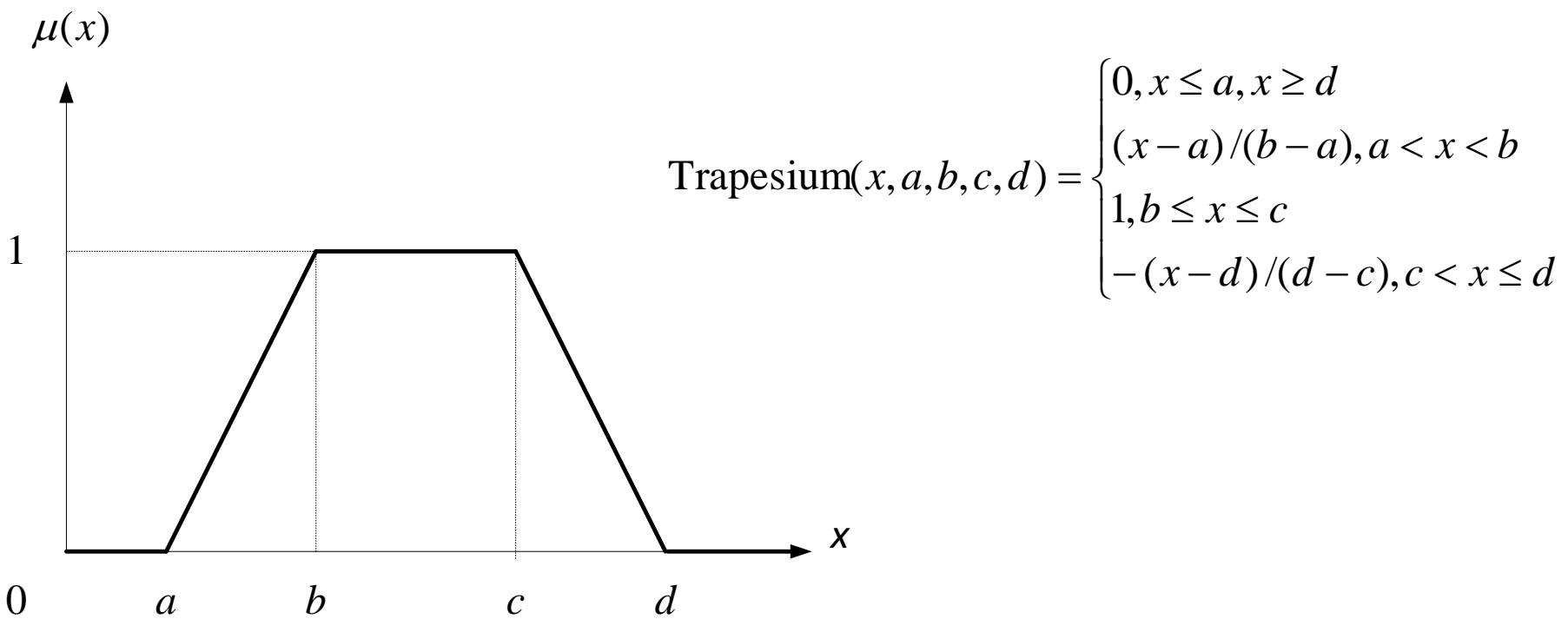
$$\text{Sigmoid}(x, a, b, c) = \begin{cases} 0, & x \leq a \\ 2((x-a)/(c-a))^2, & a < x \leq b \\ 1 - 2((c-x)/(c-a))^2, & b < x < c \\ 1, & c \leq x \end{cases}$$

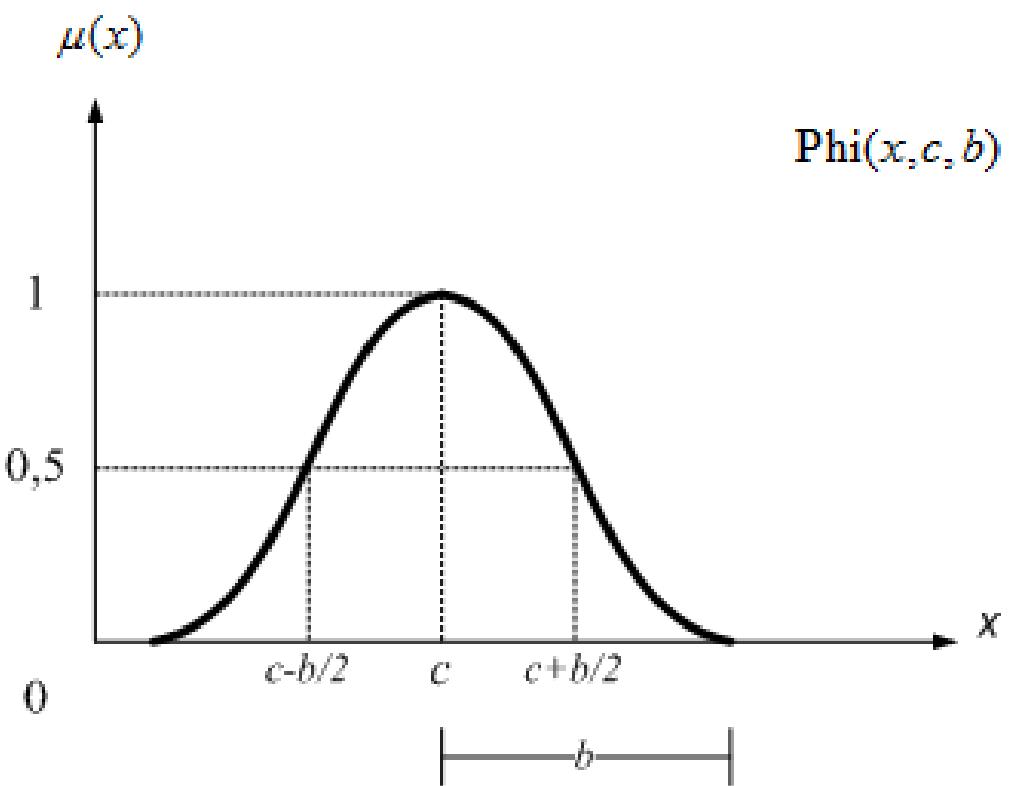
 $\mu(x)$ 

$$\text{Sigmoid}(x, a, b, c) = \begin{cases} 1, & x \leq a \\ 1 - 2((x-a)/(c-a))^2, & a < x \leq b \\ 2((c-x)/(c-a))^2, & b < x < c \\ 0, & x \geq c \end{cases}$$

$\mu(x)$ 

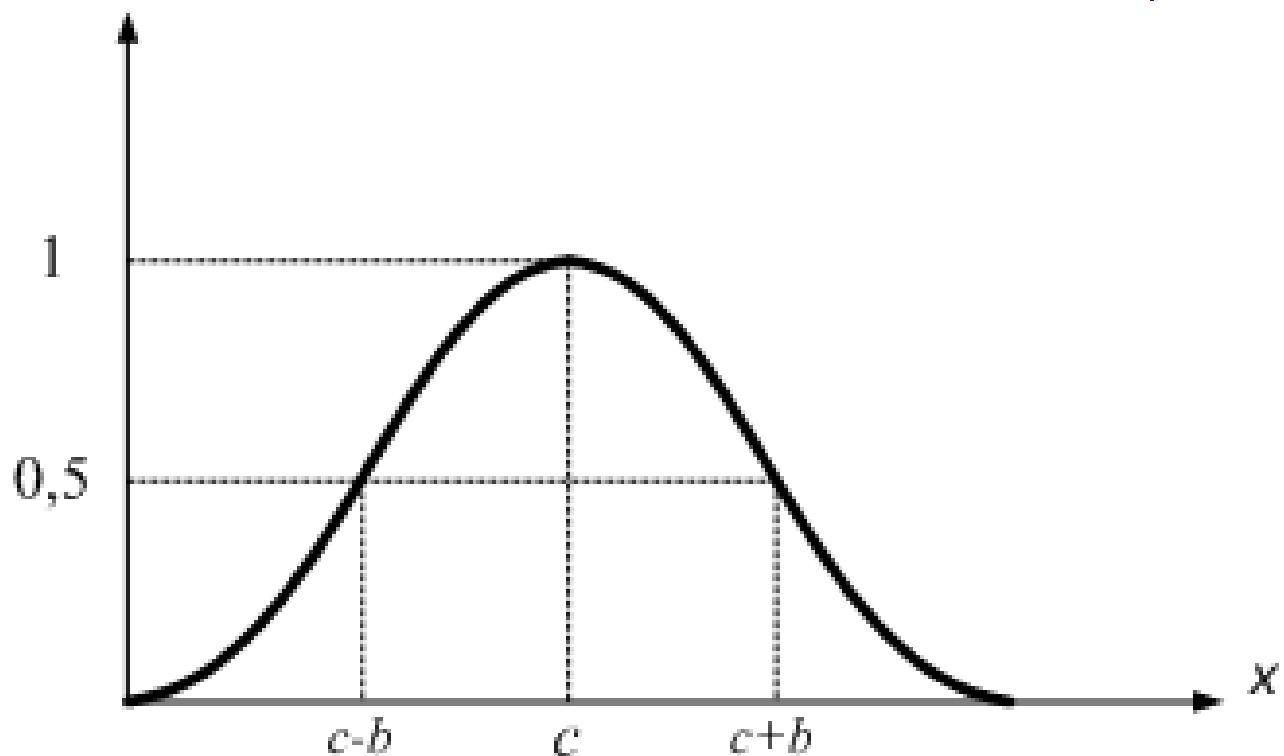
$$\text{Segitiga}(x,a,b,c) = \begin{cases} 0, & x \leq a, x \geq c \\ (x-a)/(b-a), & a < x \leq b \\ -(x-c)/(c-b), & b < x \leq c \end{cases}$$





$$\text{Phi}(x, c, b) = \begin{cases} \text{Sigmoid}\left(x, c-b, c-\frac{b}{2}, c\right), & x \leq c \\ 1 - \text{Sigmoid}\left(x, c, c+\frac{b}{2}, c+b\right), & x > c \end{cases}$$

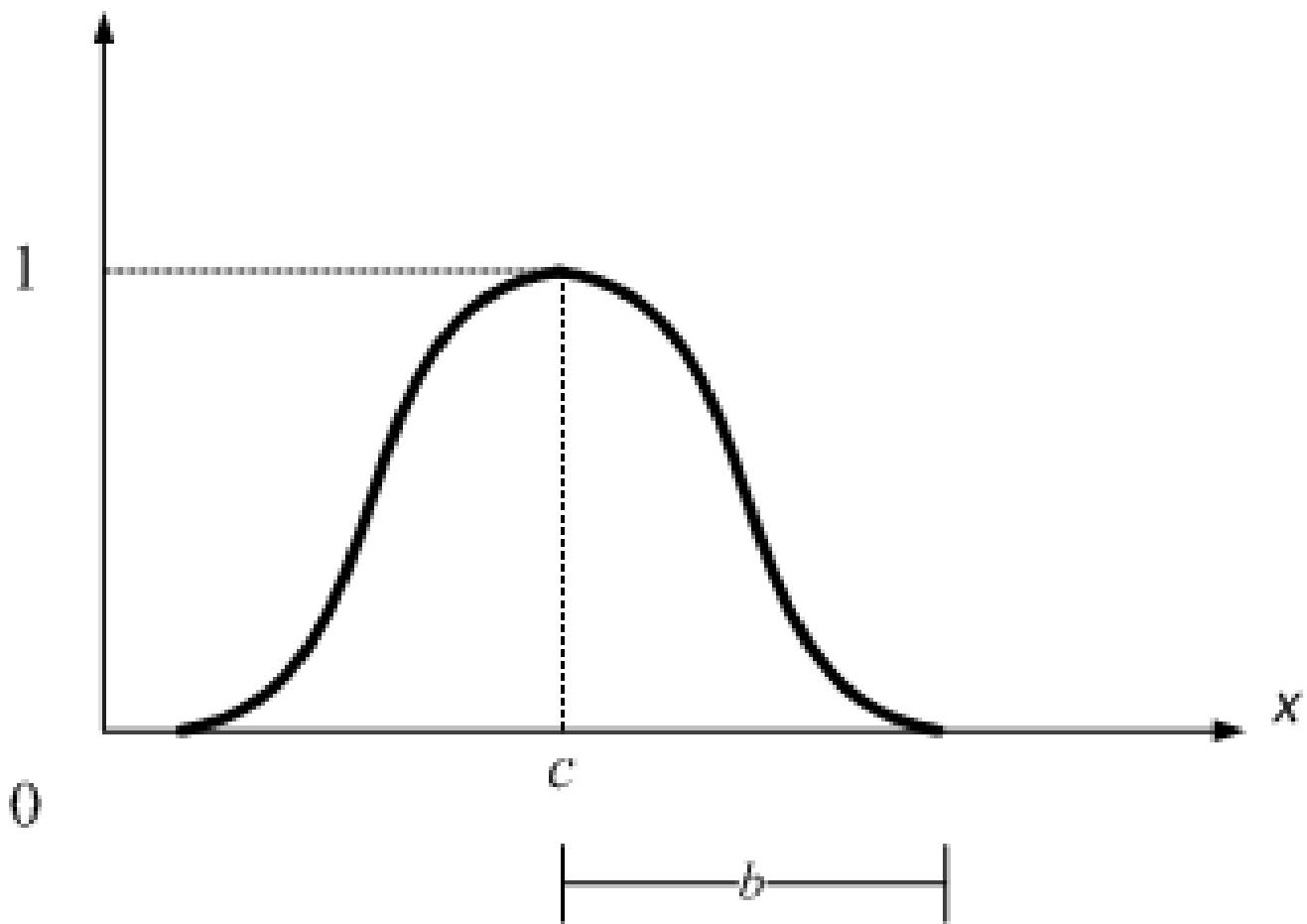
$\mu(x)$

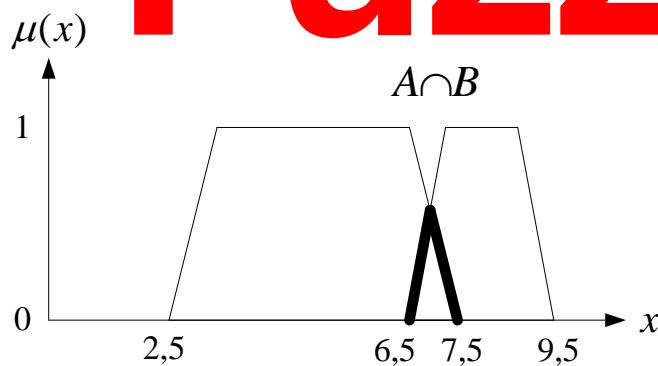
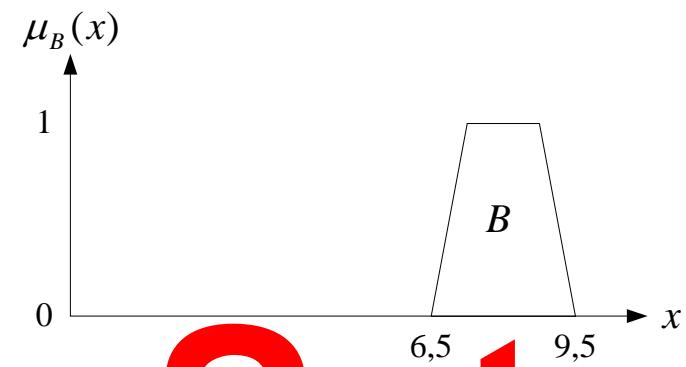
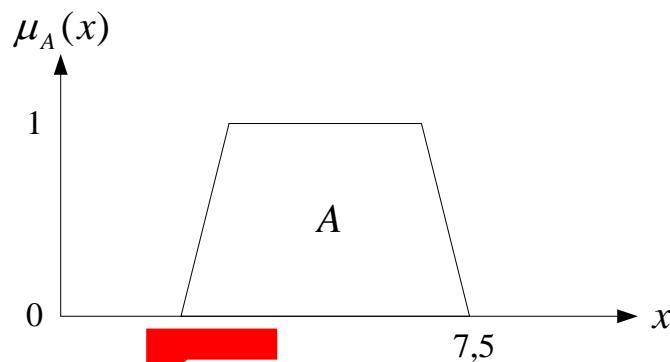


$$\text{Beta}(x, c, b) = \frac{1}{\left(1 + \left|\frac{x-c}{b}\right|^2\right)}$$

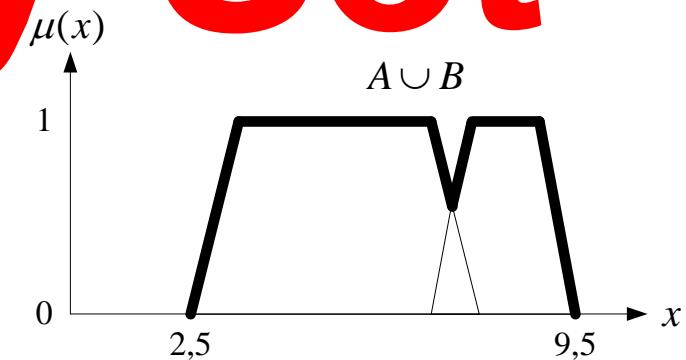
$\mu(x)$

$$\text{Gauss}(x, c, b) = e^{-b(c-x)^2}$$

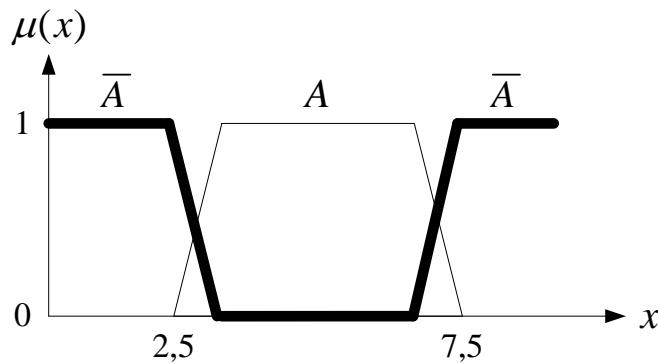




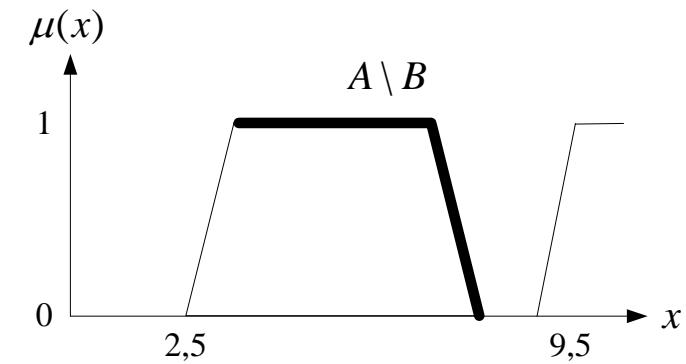
Intersection



Union



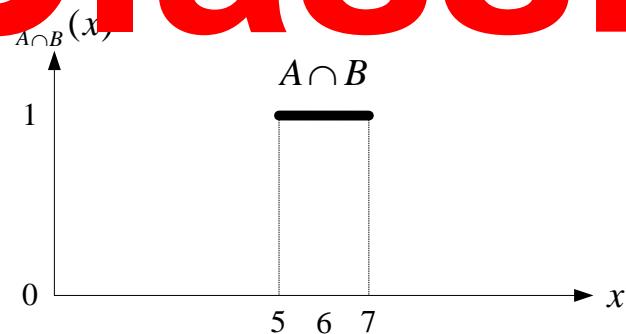
Complement



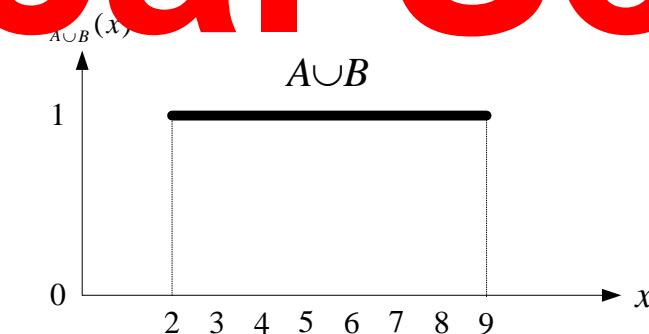
Difference

Fuzzy Set

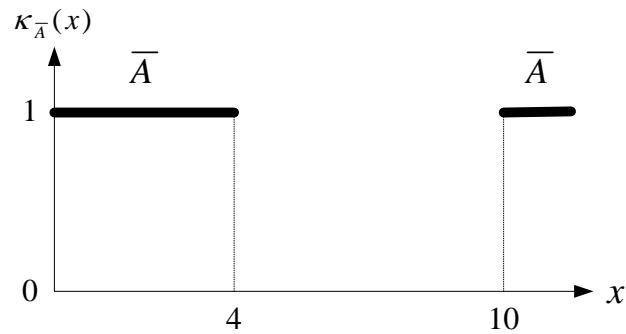
Classical Set



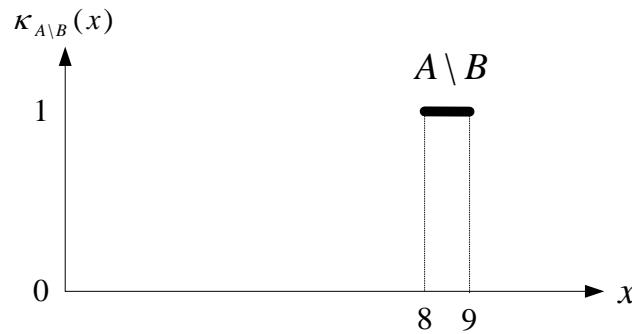
Intersection



Union



Complement



Difference

Logical connectives & Implication

- Dalam bahasa manusia, banyak percakapan yang menggunakan kalimat yang tidak pasti kebenarannya
 - '**Hampir semua** orang suka permen'
 - '**Sepertinya** dia anak yang pintar'
- Misalkan P adalah suatu *fuzzy logic proposition*
- Nilai kebenaran P adalah $[0, 1]$.
 - Nilai 0 menyatakan bahwa P adalah salah
 - Nilai 1 menyatakan bahwa P adalah benar

Logical connectives & Implication

$$T : P \rightarrow [0, 1]$$

T adalah fungsi kebenaran yang memetakan P ke suatu nilai dalam interval $[0, 1]$.

Logical Connectives

- *Negation* $T(\neg P) = 1 - T(P)$
- *Disjunction* $T(P \vee Q) = \max\{T(P), T(Q)\}$
- *Conjunction* $T(P \wedge Q) = \min\{T(P), T(Q)\}$

Approximate Reasoning

A : 'Apakah dia anak yang **pintar**?'

B : '**Sepertinya begitu.**'

A : 'Apakah Indeks Prestasi dan hasil tes psikologinya **bagus**?'

B : '**Ya, keduanya sangat bagus.**'

A : 'Apakah dia layak mendapatkan beasiswa?'

B : '**Ya, sepertinya itu adalah keputusan yang baik.**'

Approximate Reasoning

P_1 : Sebagian besar mahasiswa suka membaca

P_2 : Dani adalah mahasiswa

P_3 : Sepertinya Dani suka membaca

Reasoning yang Pasti

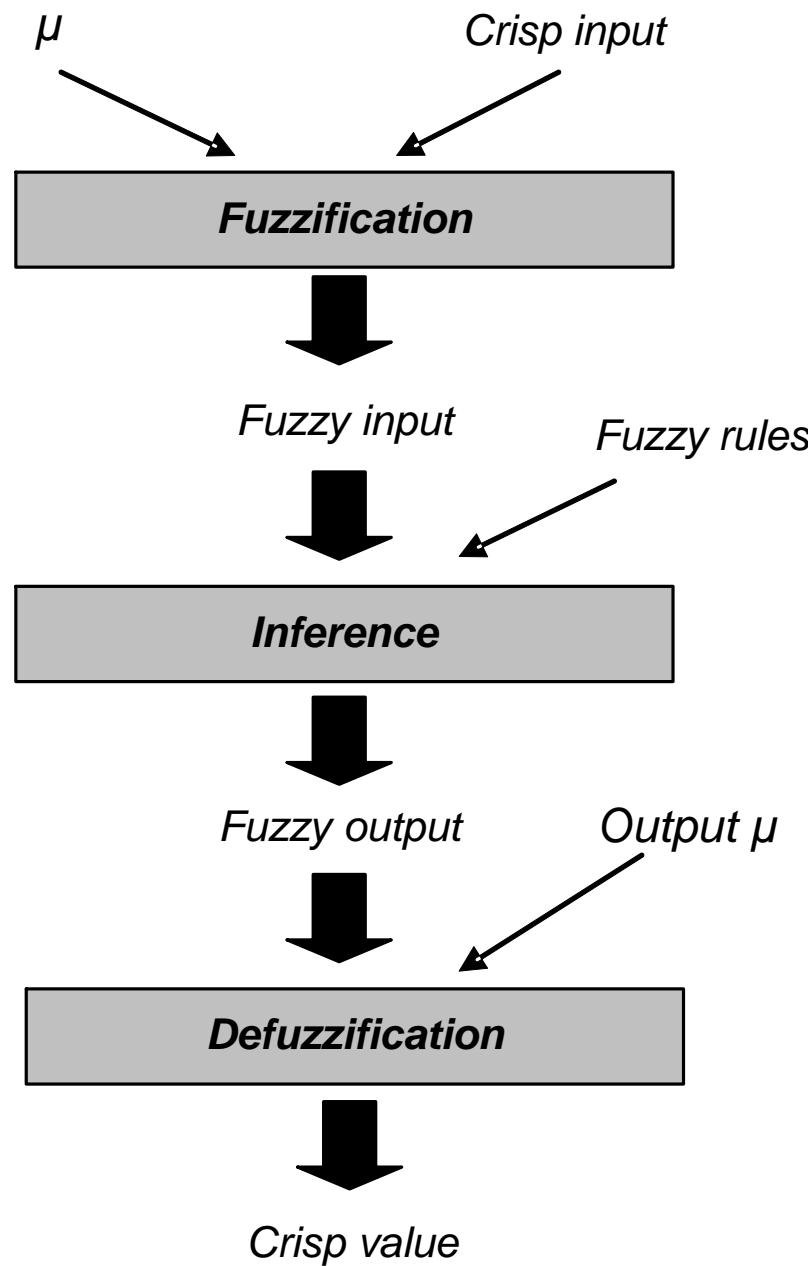
P_1 : Semua manusia pasti akan mati

P_2 : Dani adalah manusia

P_3 : Dani pasti akan mati

Fuzzy Inference Systems

- Variabel linguistik adalah suatu interval numerik dan mempunyai nilai-nilai linguistik, yang semantiknya didefinisikan oleh fungsi keanggotaannya.
- Misalnya, *Suhu* adalah suatu variabel linguistik yang bisa didefinisikan pada interval $[-10^{\circ}\text{C}, 40^{\circ}\text{C}]$.
- Variabel tersebut bisa memiliki nilai-nilai linguistik seperti 'Dingin', 'Hangat', 'Panas' yang semantiknya didefinisikan oleh fungsi-fungsi keanggotaan tertentu.



Model Inferensi

- Mamdani → *Intuitive*
- Sugeno → *Control*

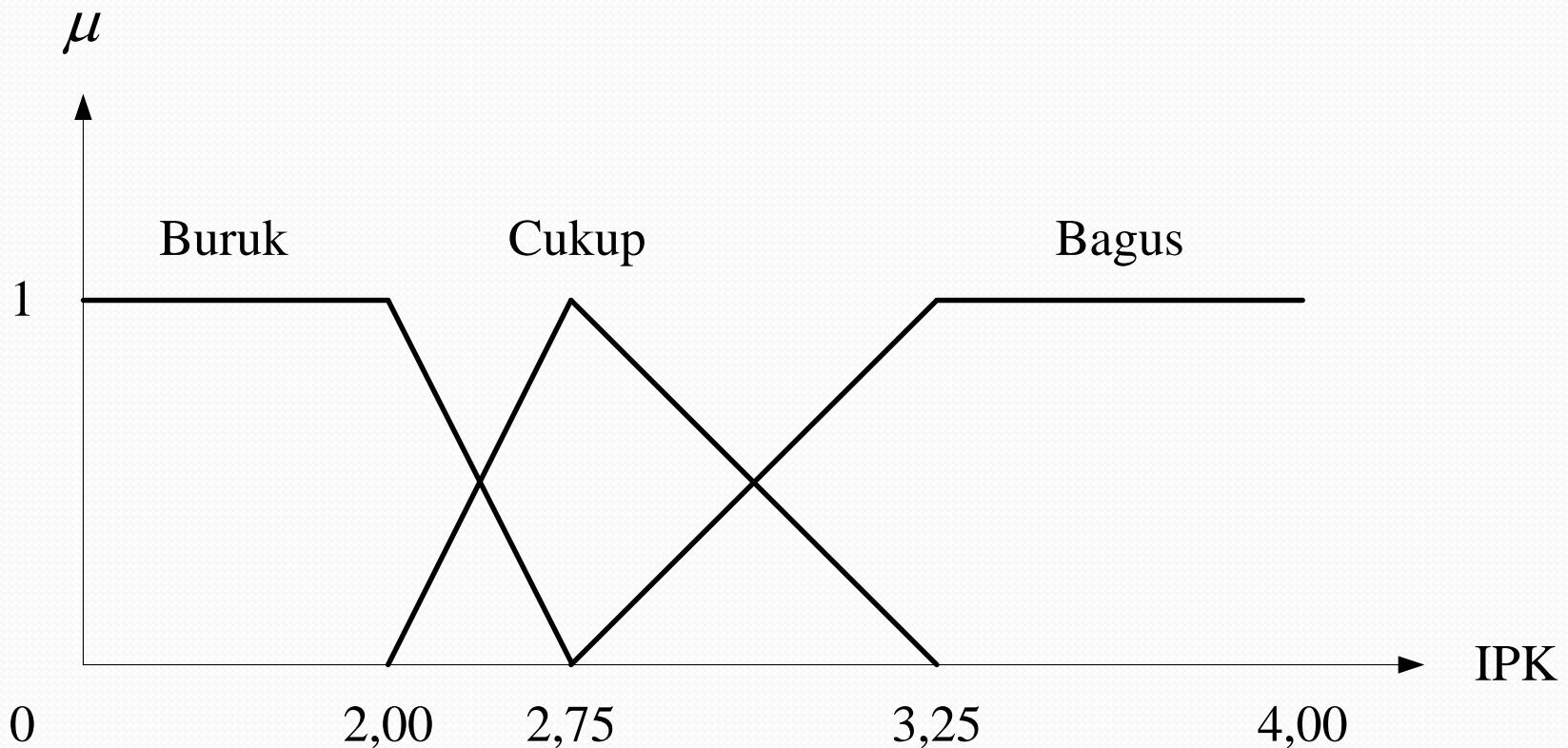
Kasus 1: Pemberian Beasiswa

Mahasiswa	IPK	Gaji Ortu (Rp/bulan)
A	3,00	10 juta
B	2,99	1 juta

Model Mamdani

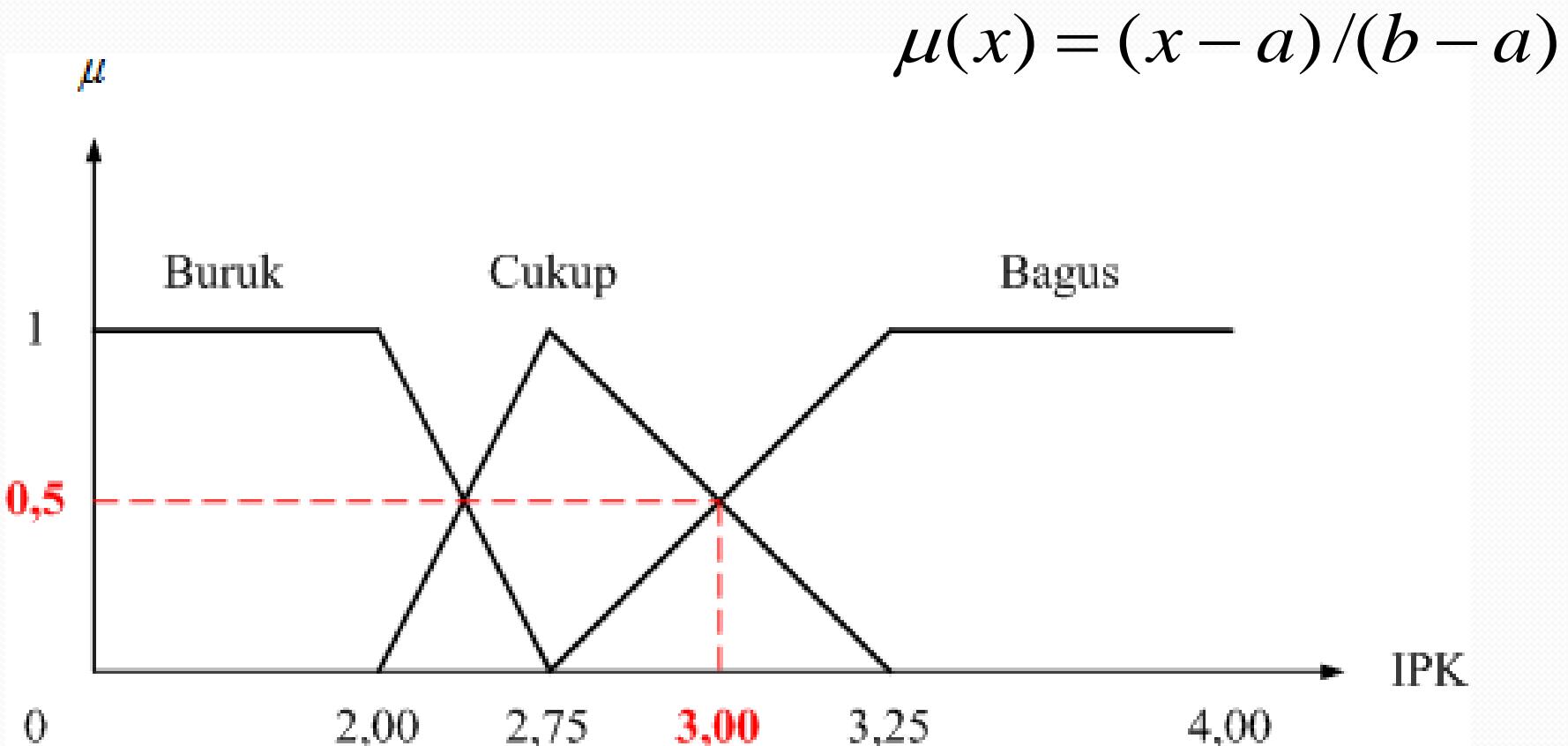
- Untuk membangun sistem yang penalarannya menyerupai perasaan manusia
- Perhitungannya kompleks sehingga membutuhkan waktu relatif lama
- Ketelitiannya tinggi

FK untuk IPK

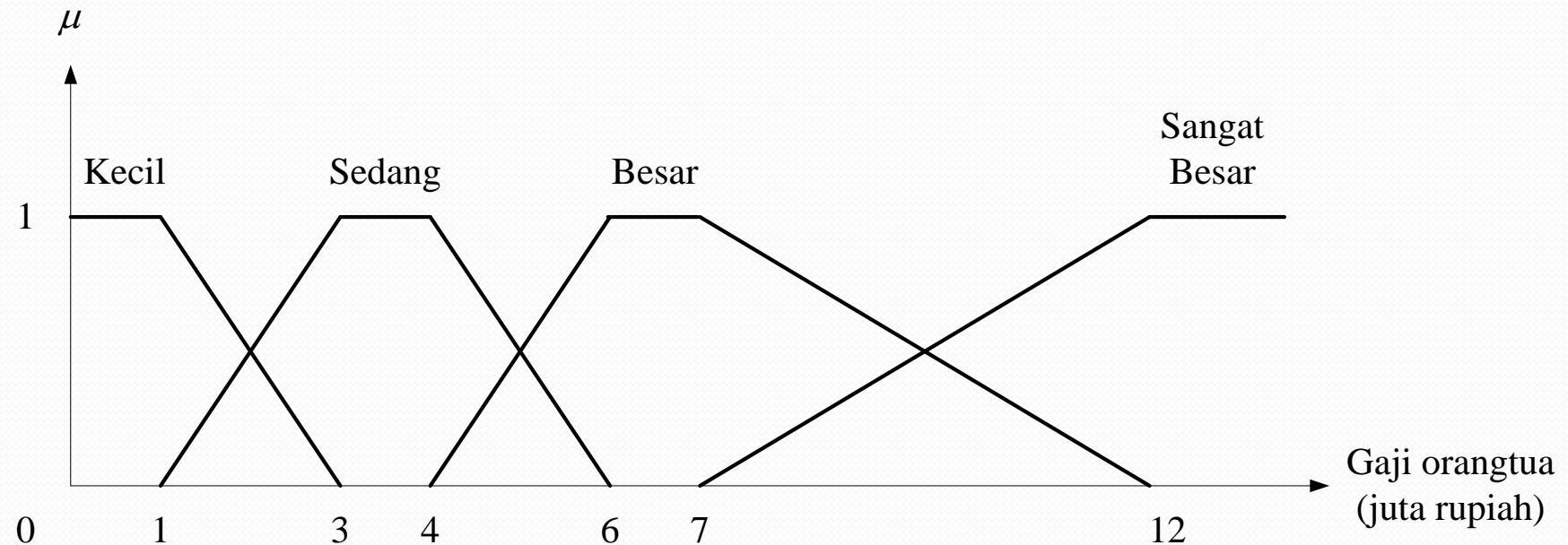


IPK mahasiswa A

$$\mu(x) = -(x - c)/(c - b)$$



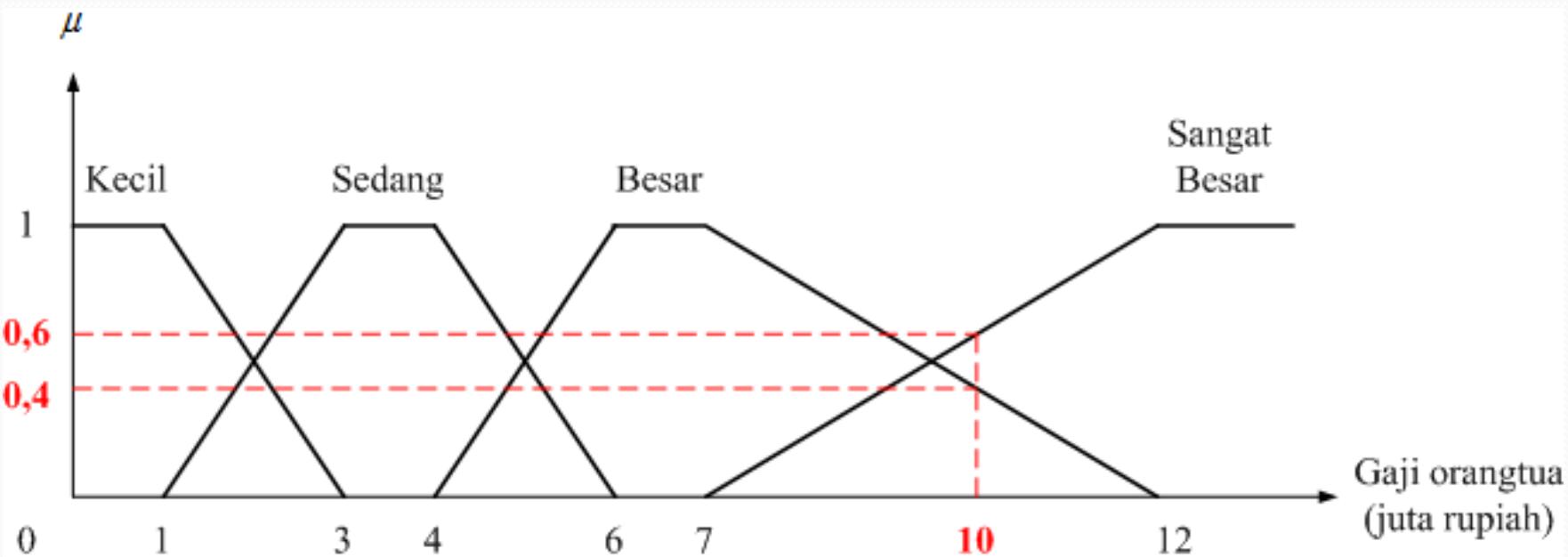
FK Gaji Orangtua



Gaji Ortu mhs A

$$\mu(x) = -(x - d)/(d - c), c < x \leq d$$

$$\mu(x) = (x - a)/(b - a), a < x < b$$



Fuzzification untuk mhs A

IPK = 3,00

Gaji Orangtua = 10 juta/bulan



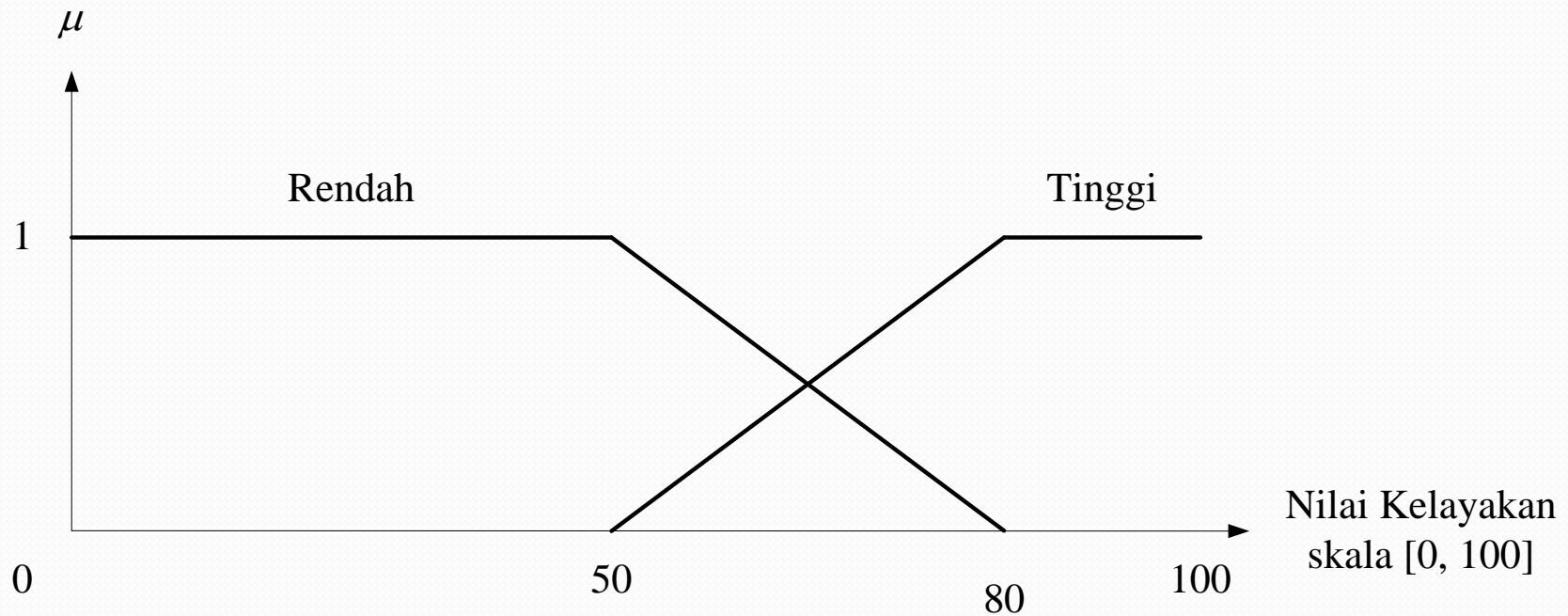
IPK = **Cukup** (0,5)

IPK = **Bagus** (0,5)

Gaji Orangtua = **Besar** (0,4)

Gaji Orangtua = **Sangat Besar** (0,6)

Fungsi Keanggotaan Nilai Kelayakan

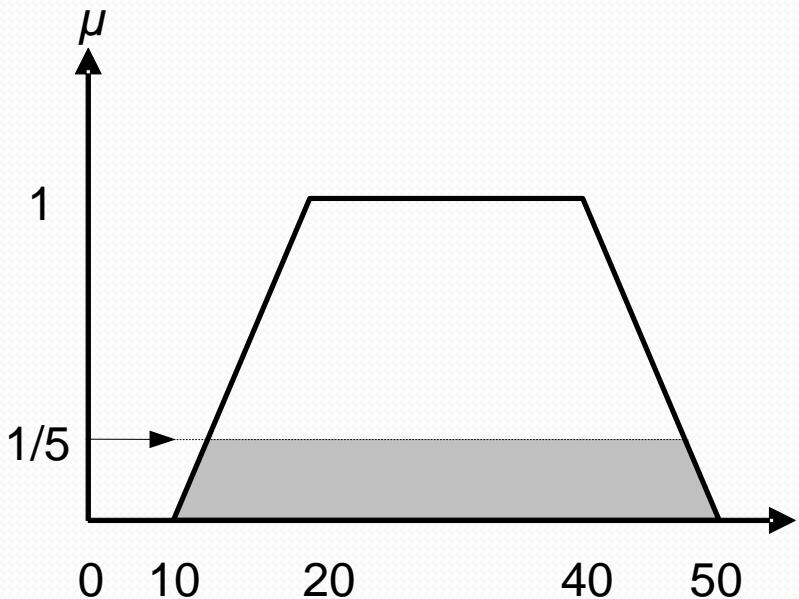


Aturan Fuzzy untuk Nilai Kelayakan

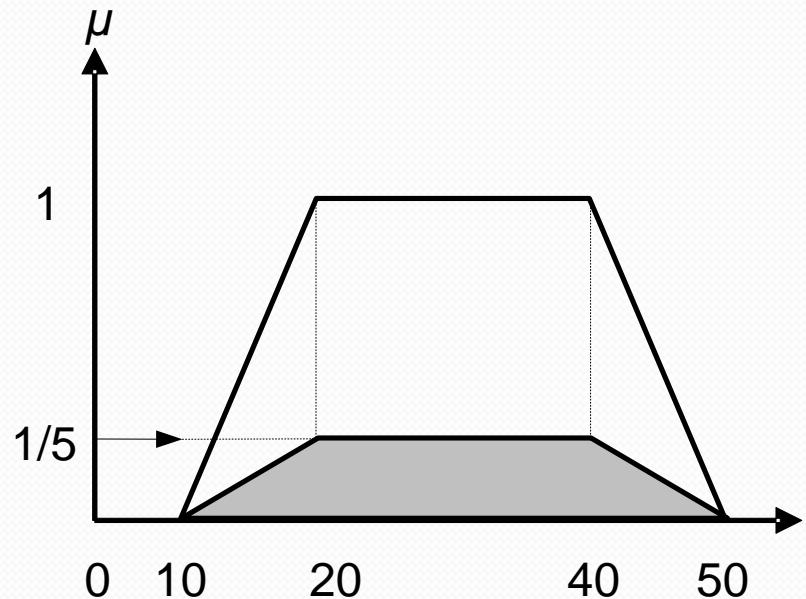
IPK \ Gaji	Kecil	Sedang	Besar	Sangat Besar
Buruk	Rendah	Rendah	Rendah	Rendah
Cukup	Tinggi	Rendah	Rendah	Rendah
Bagus	Tinggi	Tinggi	Tinggi	Rendah

1. IF IPK = Buruk AND $Gaji$ = Kecil THEN NK = Rendah
2. IF IPK = Buruk AND $Gaji$ = Sedang THEN NK = Rendah
3. IF IPK = Buruk AND $Gaji$ = Besar THEN NK = Rendah
4. IF IPK = Buruk AND $Gaji$ = Sangat Besar THEN NK = Rendah
5. IF IPK = Cukup AND $Gaji$ = Kecil THEN NK = Tinggi
6. IF IPK = Cukup AND $Gaji$ = Sedang THEN NK = Rendah
7. IF IPK = Cukup AND $Gaji$ = Besar THEN NK = Rendah
8. IF IPK = Cukup AND $Gaji$ = Sangat Besar THEN NK = Rendah
9. IF IPK = Bagus AND $Gaji$ = Kecil THEN NK = Tinggi
10. IF IPK = Bagus AND $Gaji$ = Sedang THEN NK = Tinggi
11. IF IPK = Bagus AND $Gaji$ = Besar THEN NK = Tinggi
12. IF IPK = Bagus AND $Gaji$ = Sangat Besar THEN NK = Rendah

Inferensi pada model Mamdani: *Clipping* dan *Scaling*



(a) *Clipping*



(b) *Scaling*

Aturan *fuzzy* yang diaplikasikan

7. IF $IPK = \text{Cukup}$ AND $Gaji = \text{Besar}$ THEN $NK = \text{Rendah}$
8. IF $IPK = \text{Cukup}$ AND $Gaji = \text{Sangat Besar}$ THEN $NK = \text{Rendah}$
11. IF $IPK = \text{Bagus}$ AND $Gaji = \text{Besar}$ THEN $NK = \text{Tinggi}$
12. IF $IPK = \text{Bagus}$ AND $Gaji = \text{Sangat Besar}$ THEN $NK = \text{Rendah}$

Nilai *fuzzy* untuk mhs A

IPK = 3,00

Gaji Orangtua = 10 juta/bulan



IPK = **Cukup** (0,5)

IPK = **Bagus** (0,5)

Gaji Orangtua = **Besar** (0,4)

Gaji Orangtua = **Sangat Besar** (0,6)

Conjunction (\wedge) & Disjunction (\vee)

IF $IPK = \text{Cukup}(0,5)$ AND $Gaji = \text{Besar}(0,4)$ THEN $NK = \text{Rendah}(0,4)$

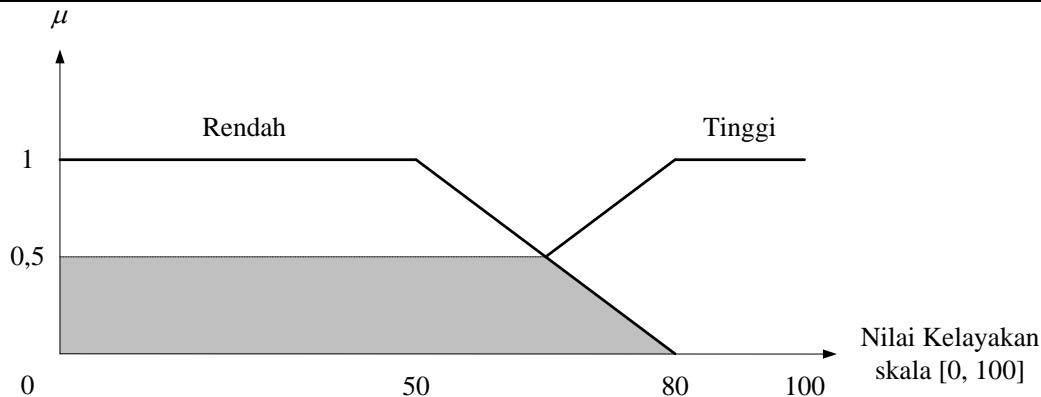
IF $IPK = \text{Cukup}(0,5)$ AND $Gaji = \text{Sangat Besar}(0,6)$ THEN $NK = \text{Rendah}(0,5)$

IF $IPK = \text{Bagus}(0,5)$ AND $Gaji = \text{Besar}(0,4)$ THEN $NK = \text{Tinggi}(0,4)$

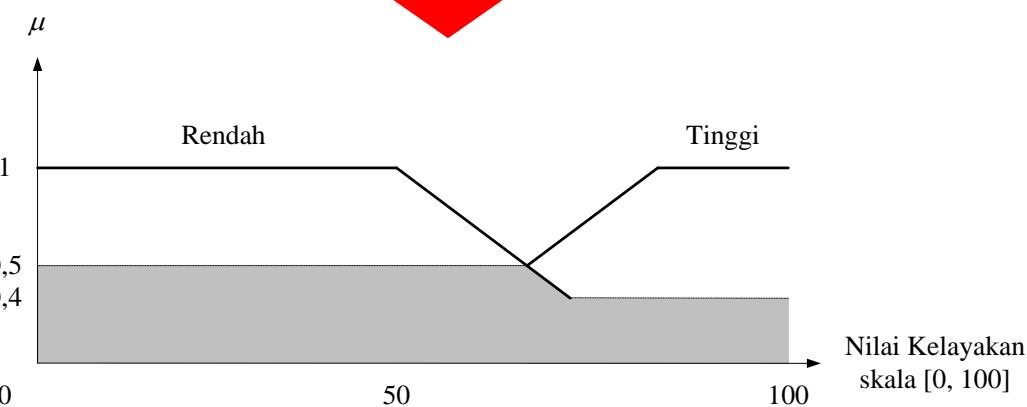
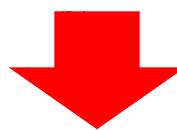
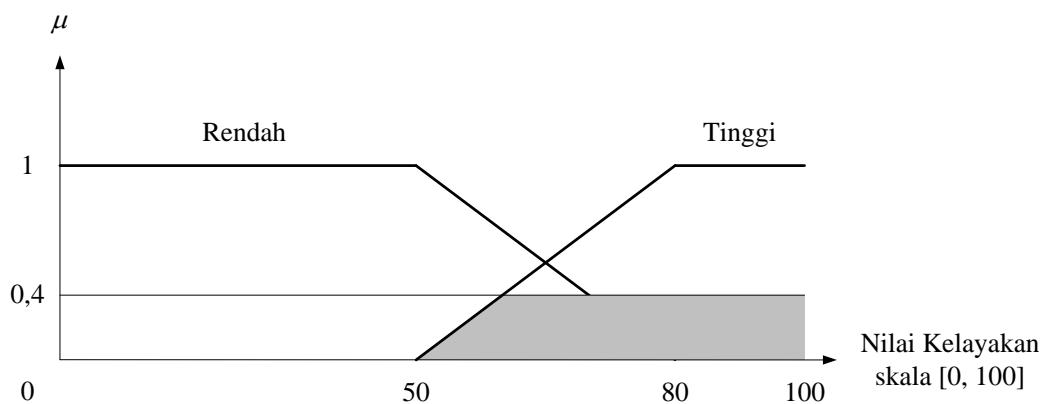
IF $IPK = \text{Bagus}(0,5)$ AND $Gaji = \text{Sangat Besar}(0,6)$ THEN $NK = \text{Rendah}(0,5)$

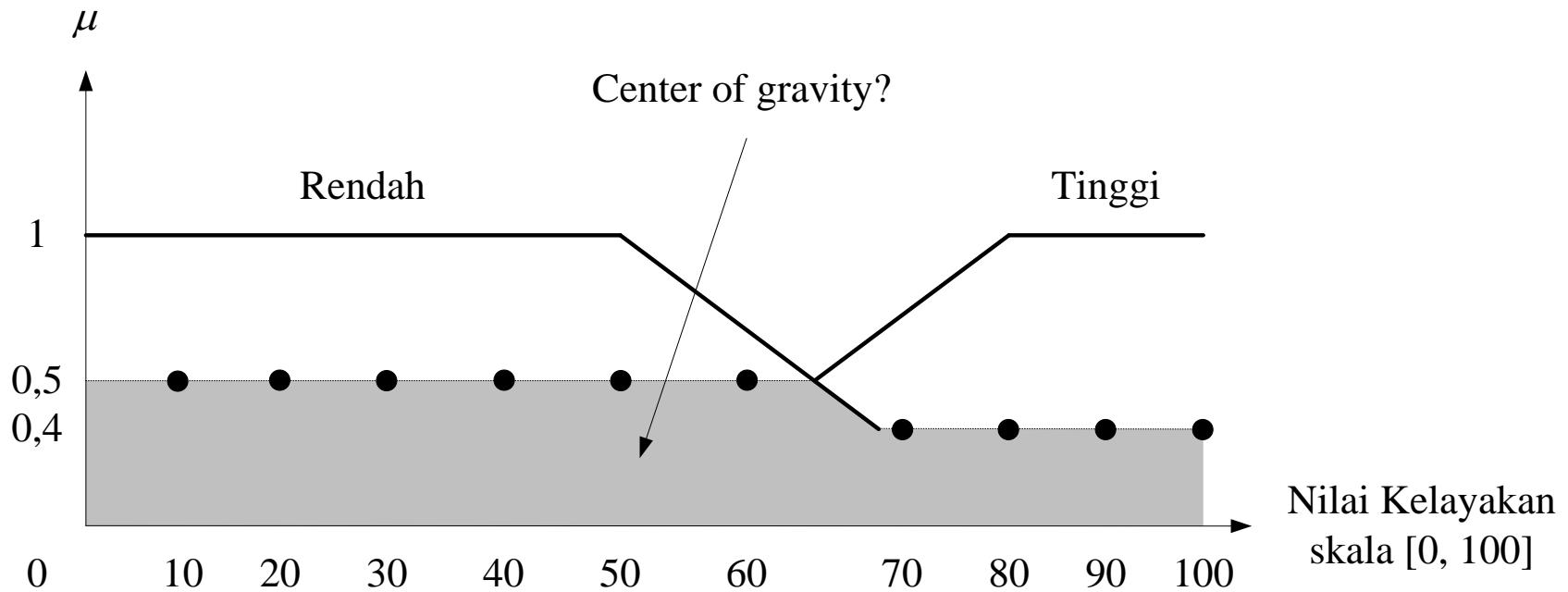


$NK = \text{Rendah} (0,5)$
 $NK = \text{Tinggi} (0,4)$



(a)

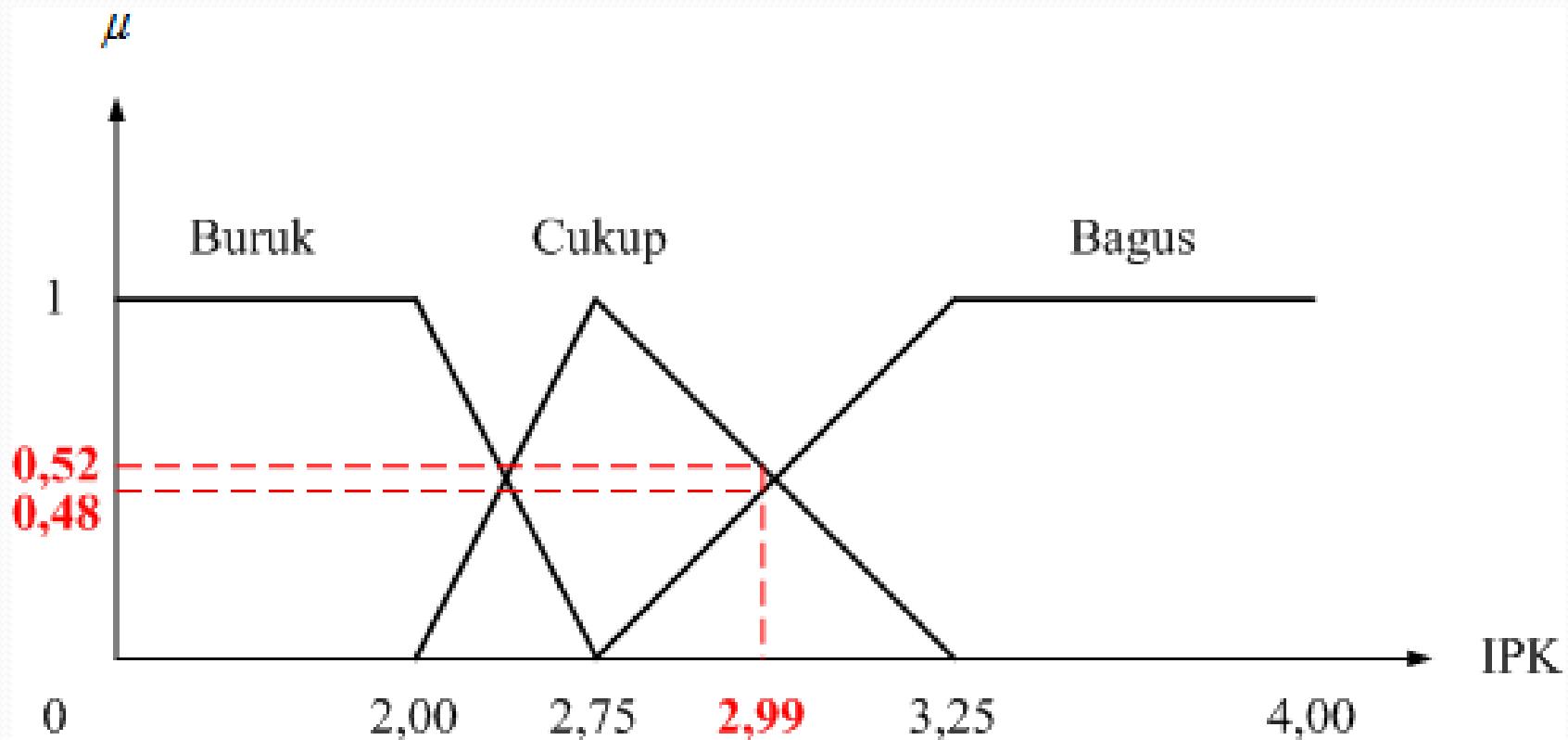




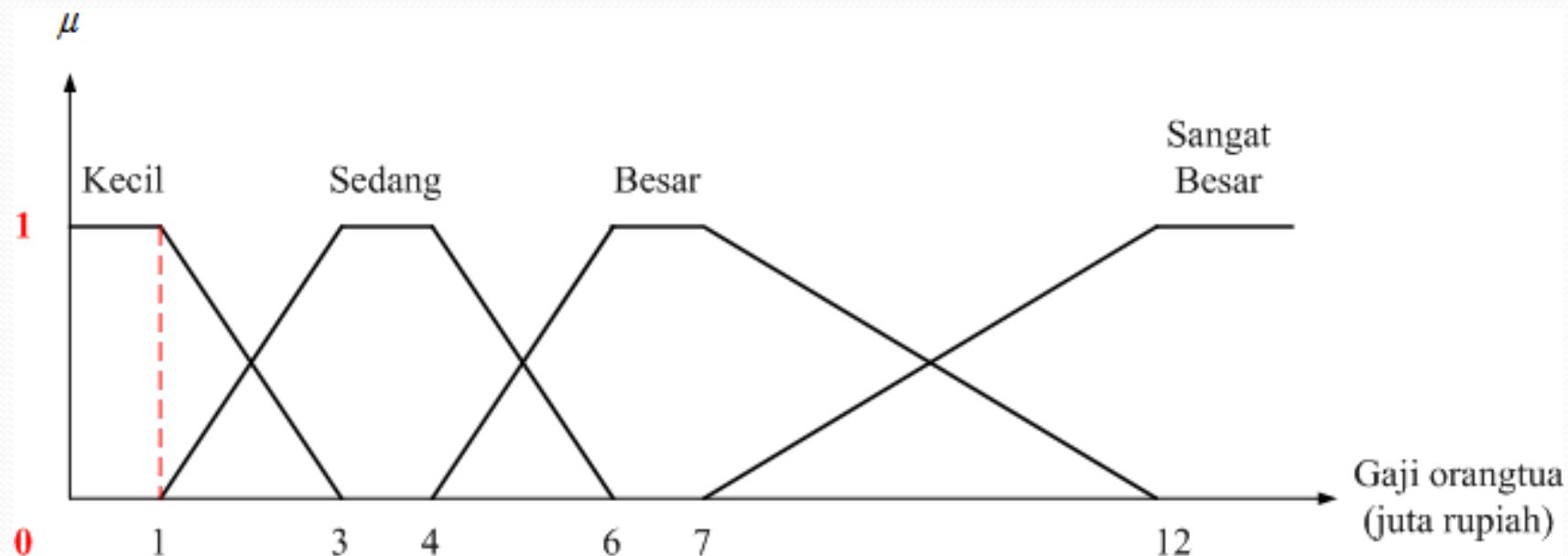
$$y^* = \frac{(10 + 20 + 30 + 40 + 50 + 60)0,5 + (70 + 80 + 90 + 100)0,4}{6(0,5) + 4(0,4)}$$

$$y^* = \frac{105 + 136}{4,6} = 52,39$$

IPK mahasiswa B



Gaji Orangtua mhs B



Conjunction (\wedge) & Disjunction (\vee)

IF $IPK = \text{Cukup}(0,52)$ AND $Gaji = \text{Kecil}(1)$ THEN $NK = \text{Tinggi}(0,52)$

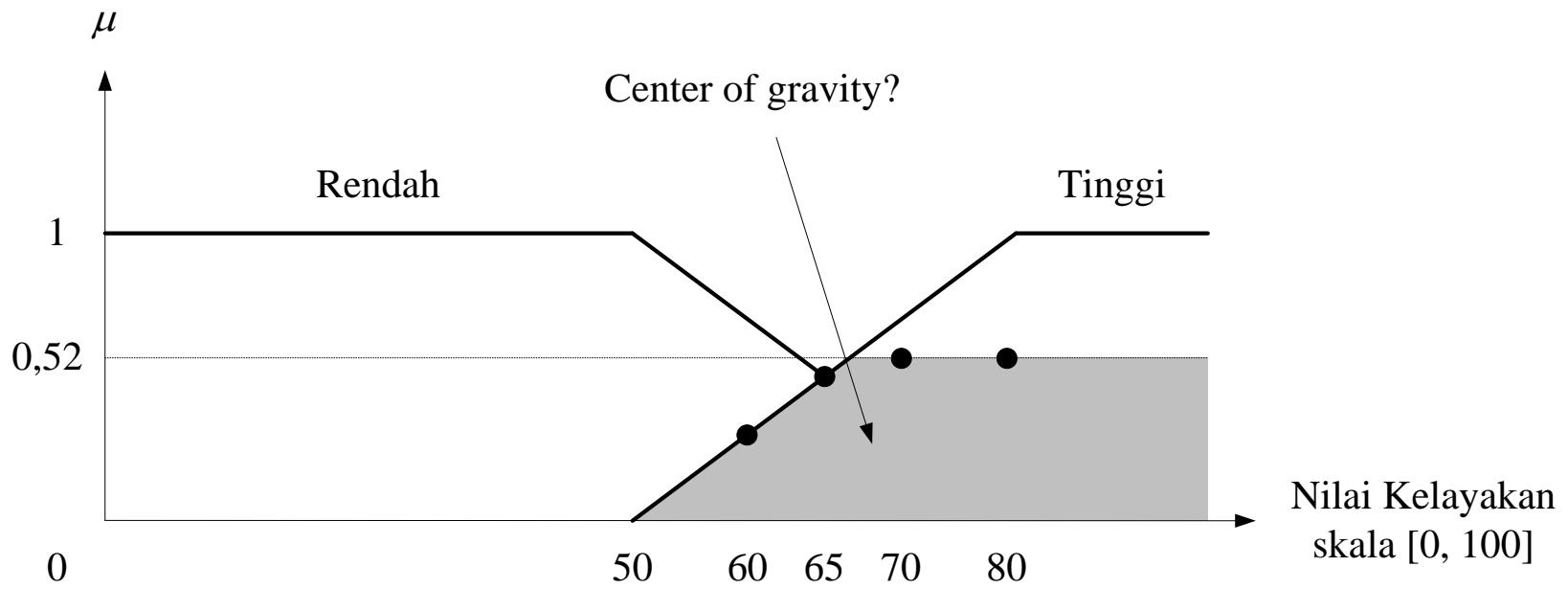
IF $IPK = \text{Cukup}(0,52)$ AND $Gaji = \text{Sedang}(0)$ THEN $NK = \text{Rendah}(0)$

IF $IPK = \text{Besar}(0,48)$ AND $Gaji = \text{Kecil}(1)$ THEN $NK = \text{Tinggi}(0,48)$

IF $IPK = \text{Besar}(0,48)$ AND $Gaji = \text{Sedang}(0)$ THEN $NK = \text{Tinggi}(0)$



$NK = \text{Rendah} (0)$
 $NK = \text{Tinggi} (0,52)$



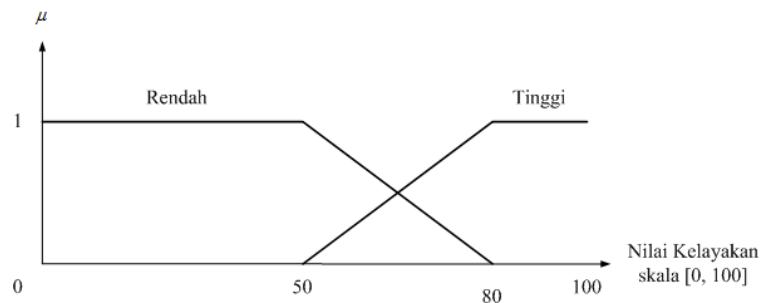
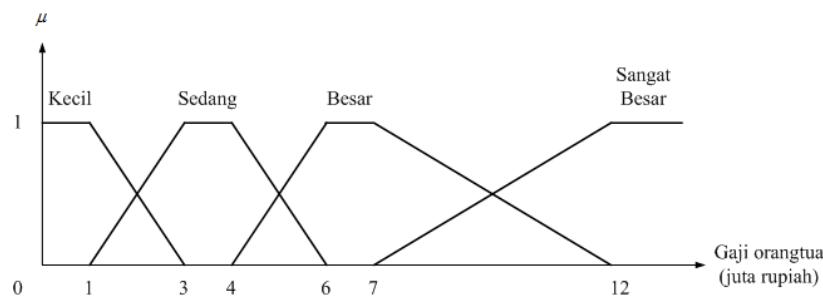
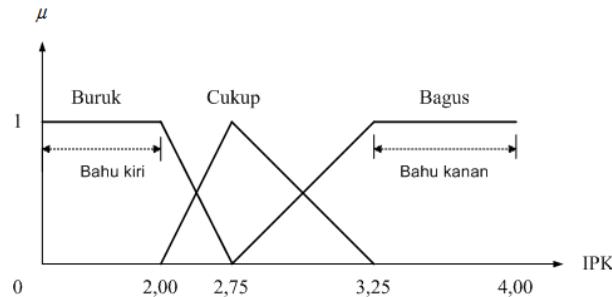
$$y^* = \frac{60(1/3) + 65(1/2) + (70 + 80)(0,52)}{(1/3) + (1/2) + (0,52)2}$$

$$y^* = \frac{20 + 32,5 + 78}{2,87334} = 69,66$$

Keputusan Model Mamdani

- Mahasiswa B dengan IPK = 2,99 dan Gaji orangtuanya sebesar 1 juta rupiah per bulan memperoleh Nilai Kelayakan sebesar **69,66**.
- Lebih besar dibandingkan dengan Nilai Kelayakan mahasiswa A yang sebesar **52,39**.
- Jadi, mahasiswa B layak mendapatkan beasiswa.

Model Mamdani



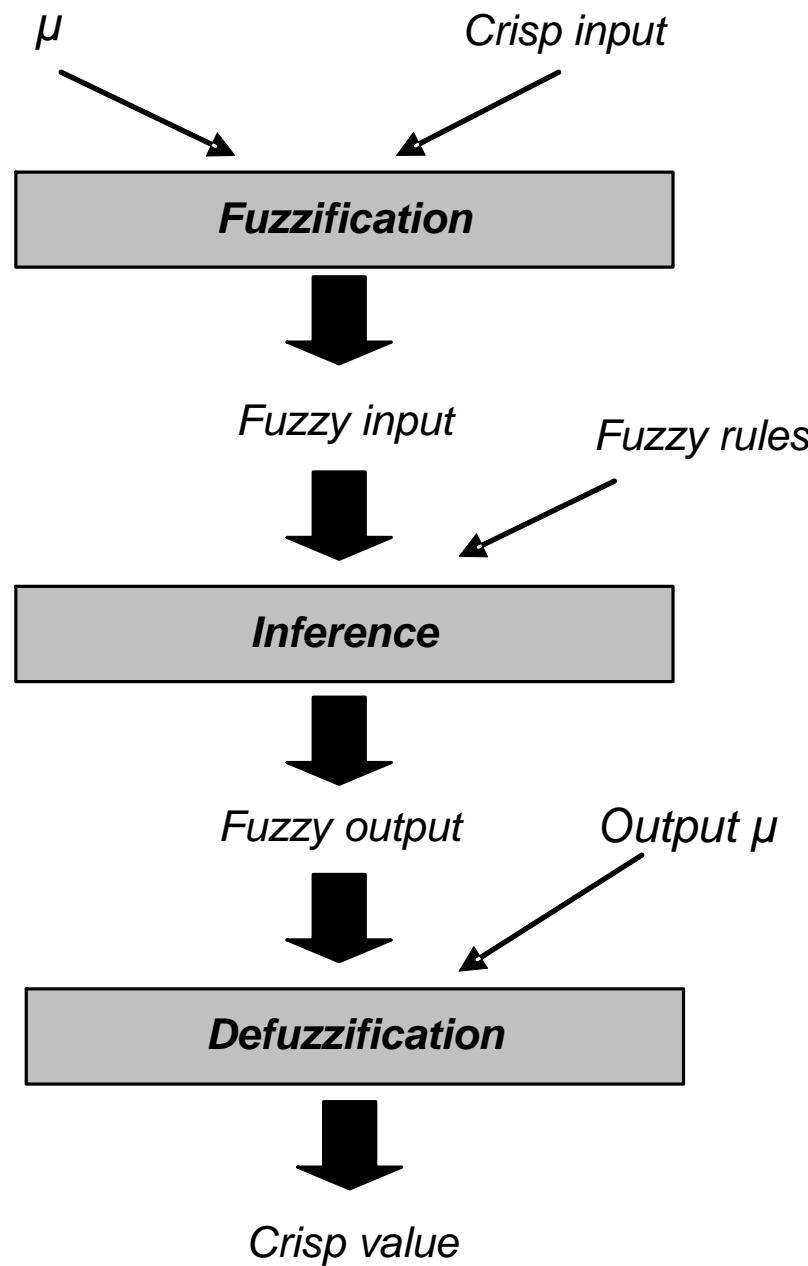
IPK \ Gaji	Kecil	Sedang	Besar	Sangat Besar
Buruk	Rendah	Rendah	Rendah	Rendah
Cukup	Tinggi	Rendah	Rendah	Rendah
Bagus	Tinggi	Tinggi	Tinggi	Rendah

Model Sugeno

- Model ini sering digunakan untuk membangun sistem kontrol yang membutuhkan respon cepat
- Proses perhitungannya sangat sederhana sehingga membutuhkan waktu relatif cepat sehingga sangat sesuai untuk sistem kontrol
- Bagaimana jika digunakan untuk masalah pemberian beasiswa?

Masalah: Pemberian Beasiswa

Mahasiswa	IPK	Gaji Ortu (Rp/bulan)
A	3,00	10 juta
B	2,99	1 juta



Fuzzification & Rule Evaluation

- Misalkan proses *fuzzification*-nya sama persis dengan model Mamdani.
- Misalkan *Rule* yang digunakan juga sama persis dengan model Mamdani.

Mahasiswa A

IF $IPK = \text{Cukup}(0,5)$ AND $Gaji = \text{Besar}(0,4)$ THEN $NK = \text{Rendah}(0,4)$

IF $IPK = \text{Cukup}(0,5)$ AND $Gaji = \text{Sangat Besar}(0,6)$ THEN $NK = \text{Rendah}(0,5)$

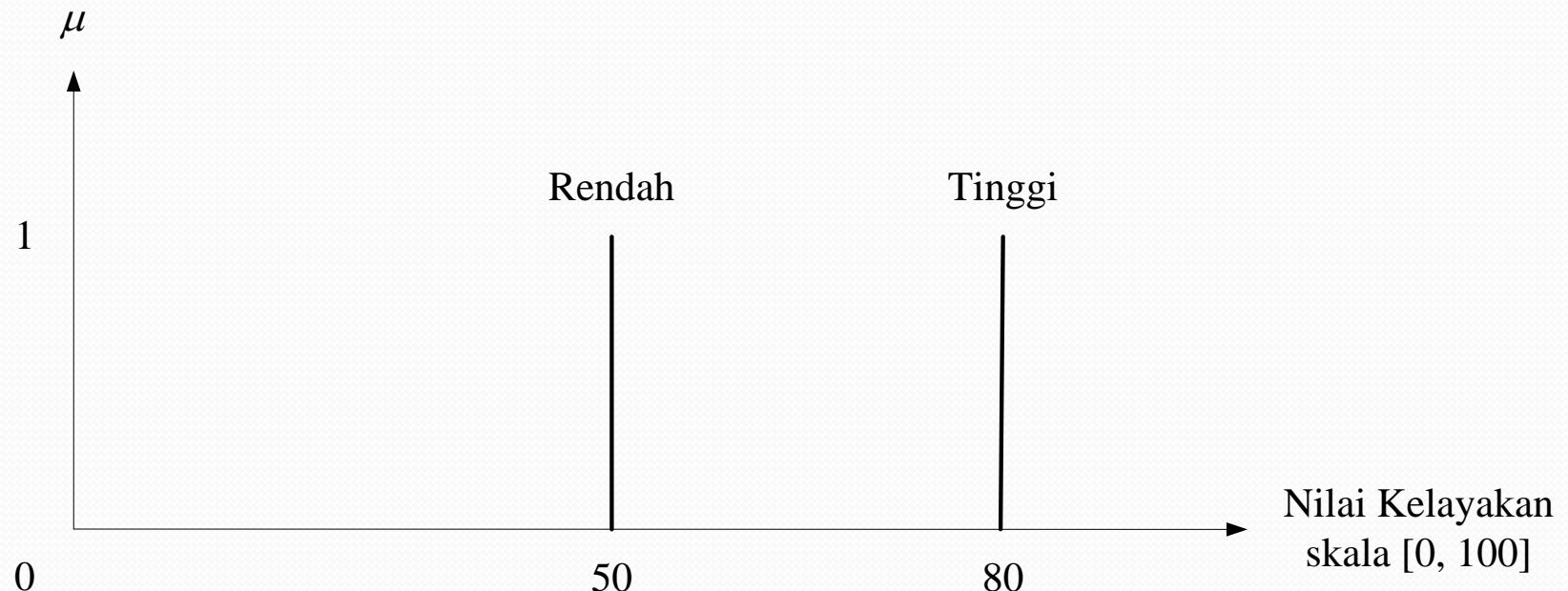
IF $IPK = \text{Bagus}(0,5)$ AND $Gaji = \text{Besar}(0,4)$ THEN $NK = \text{Tinggi}(0,4)$

IF $IPK = \text{Bagus}(0,5)$ AND $Gaji = \text{Sangat Besar}(0,6)$ THEN $NK = \text{Rendah}(0,5)$

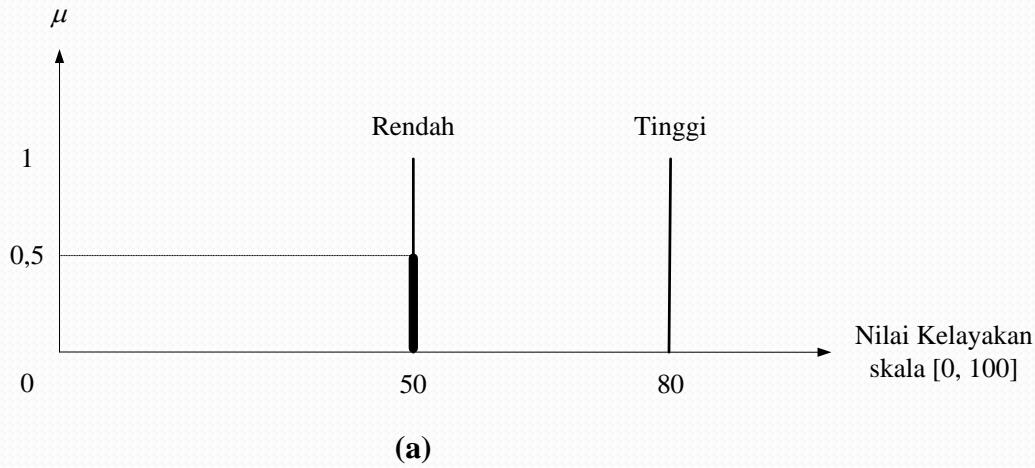


$NK = \text{Rendah} (0,5)$
 $NK = \text{Tinggi} (0,4)$

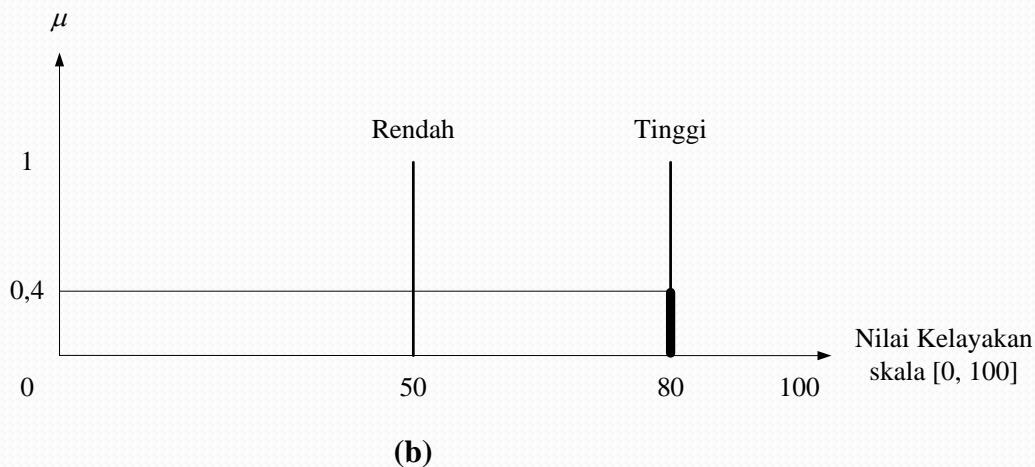
FK *singleton* untuk Nilai Kelayakan



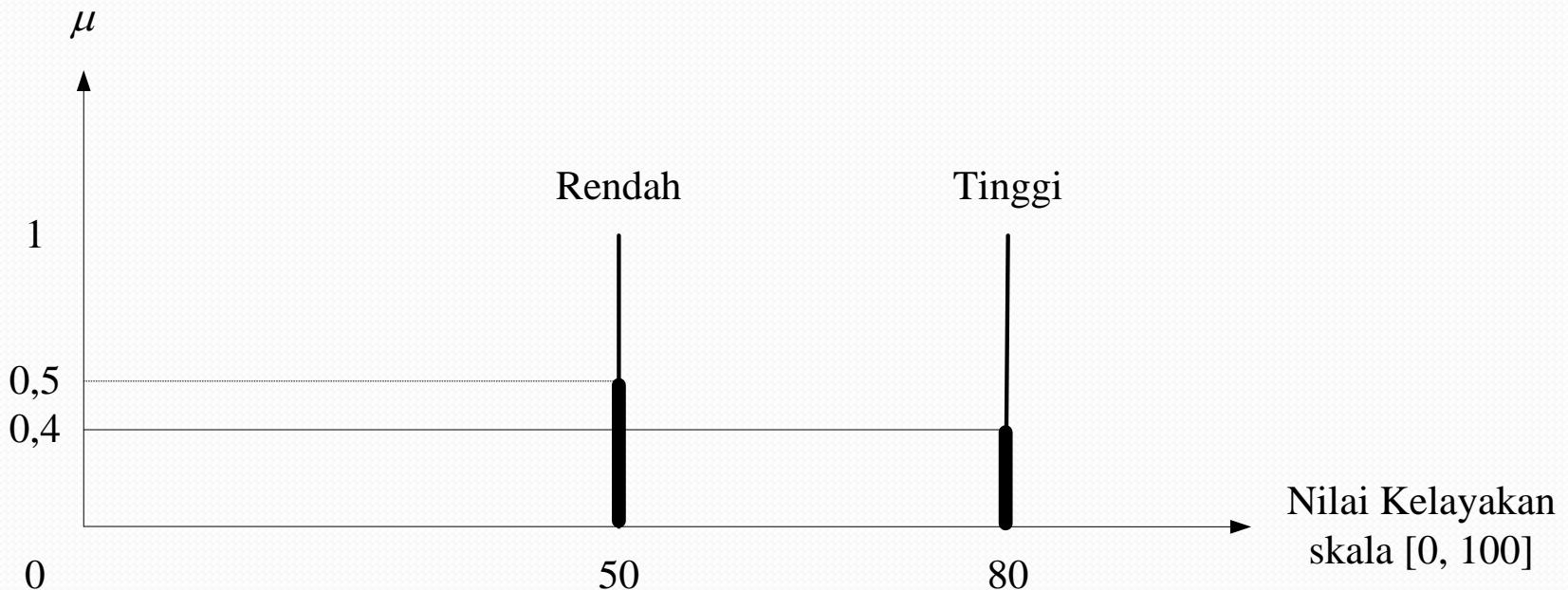
Untuk mahasiswa A



- NK = Rendah (0,5)
- NK = Tinggi (0,4)



Proses *Composition*



Defuzzification: Weighted Average

$$y^* = \frac{(0,5)50 + (0,4)80}{(0,5) + (0,4)} = 63,33$$

Mahasiswa B

IF $IPK = \text{Cukup}(0,52)$ AND $Gaji = \text{Kecil}(1)$ THEN $NK = \text{Tinggi}(0,52)$

IF $IPK = \text{Cukup}(0,52)$ AND $Gaji = \text{Sedang}(0)$ THEN $NK = \text{Rendah}(0)$

IF $IPK = \text{Besar}(0,48)$ AND $Gaji = \text{Kecil}(1)$ THEN $NK = \text{Tinggi}(0,48)$

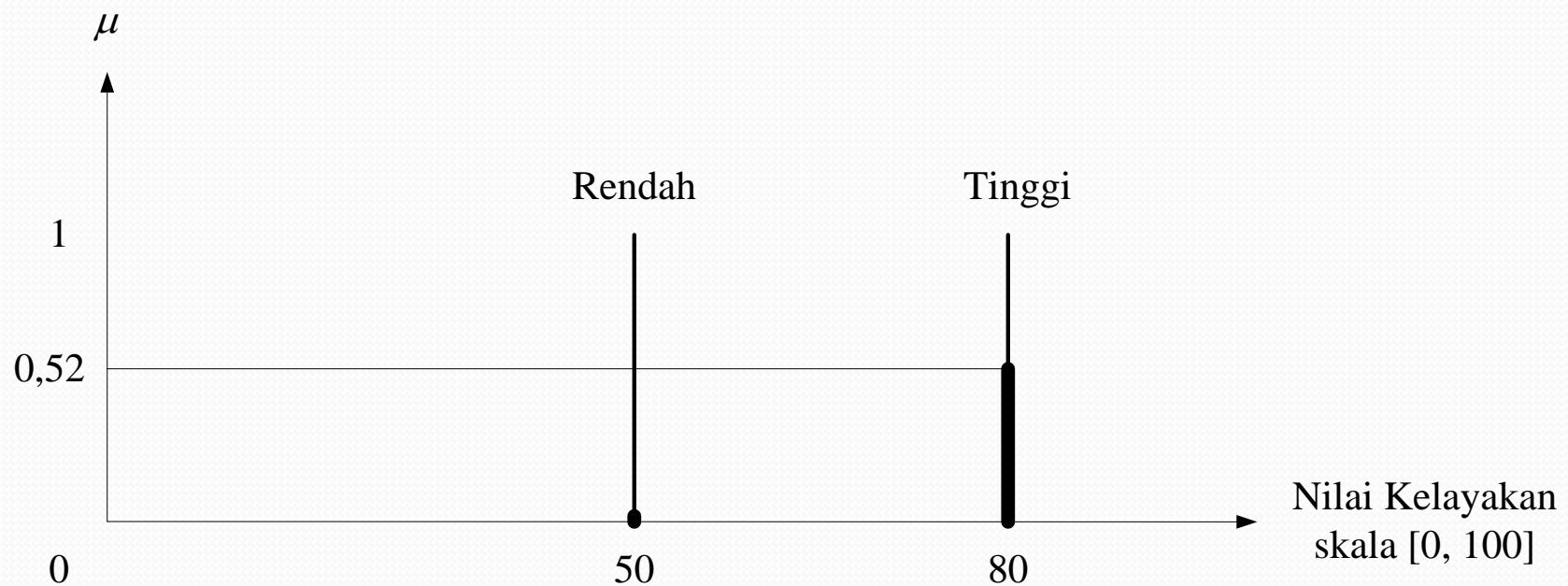
IF $IPK = \text{Besar}(0,48)$ AND $Gaji = \text{Sedang}(0)$ THEN $NK = \text{Tinggi}(0)$



$NK = \text{Rendah} (0)$
 $NK = \text{Tinggi} (0,52)$

Untuk Mahasiswa B

- NK = Rendah (0)
- NK = Tinggi (0,52)



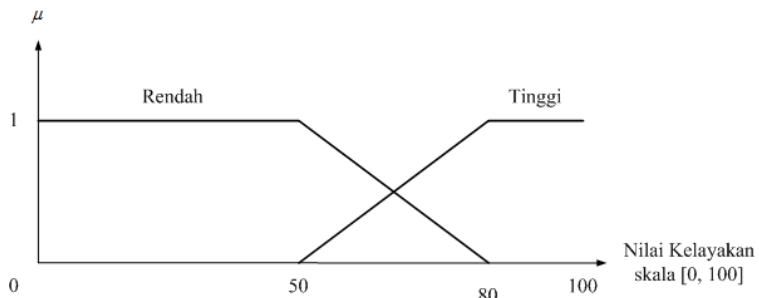
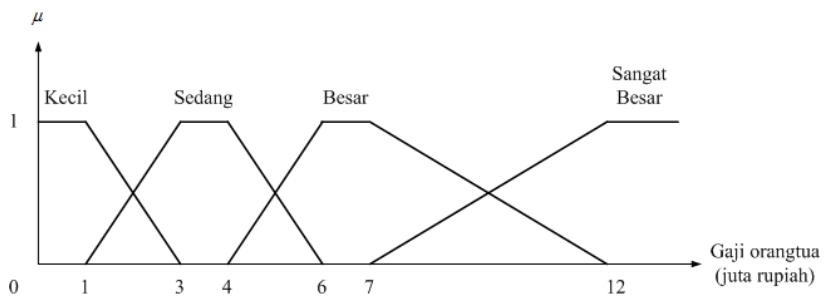
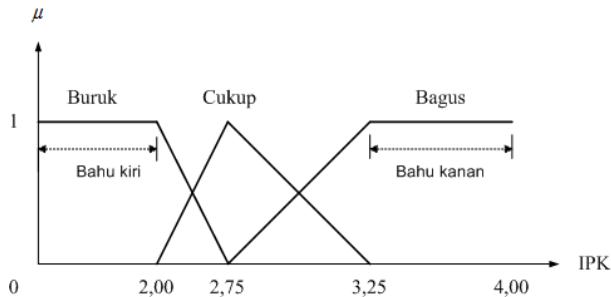
Defuzzification: Weighted Average

$$y^* = \frac{(0)50 + (0,52)80}{0 + 0,52} = 80$$

Keputusan Model Sugeno

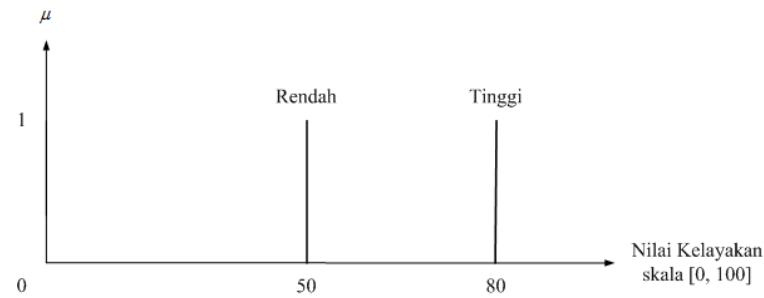
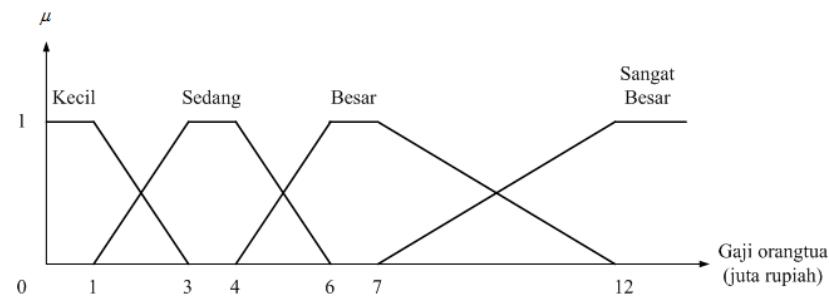
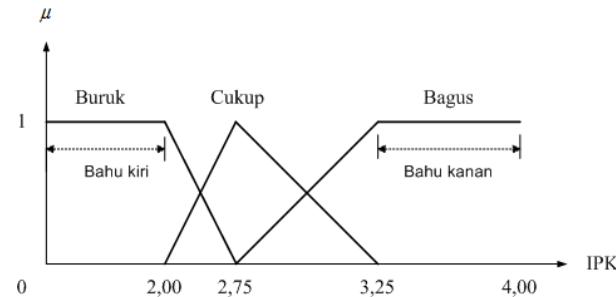
- Mahasiswa B dengan IPK = 2,99 dan Gaji orangtuanya sebesar Rp 1 juta per bulan memperoleh Nilai Kelayakan sebesar **80**.
- Lebih besar dibandingkan dengan Nilai Kelayakan mahasiswa A yang sebesar **63,33**.
- Jadi, mahasiswa B layak mendapatkan beasiswa.

Model Mamdani



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

Model Sugeno



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

Nilai Kelayakan mahasiswa A & B

Mahasiswa	Nilai Kelayakan mendapat beasiswa	
	Model Mamdani	Model Sugeno
A	52,39	63,33
B	69,66	80
Selisih A dan B	17,72	16,67

Sistem Berbasis *Crisp Sets* dan FOL

IF $IPK = 2,57$ AND $Gaji = 1,20$ juta THEN $NK = 62,5$

IF $IPK = 2,57$ AND $Gaji = 1,25$ juta THEN $NK = 62,3$

...

...

...

Aturan FOL untuk proses inference

IPK \ Gaji	G1	G2	G3	G4	G5
P1	70	60	40	30	20
P2	80	70	50	40	30
P3	90	80	60	50	40
P4	100	90	70	60	50

Aturan FOL untuk proses inference

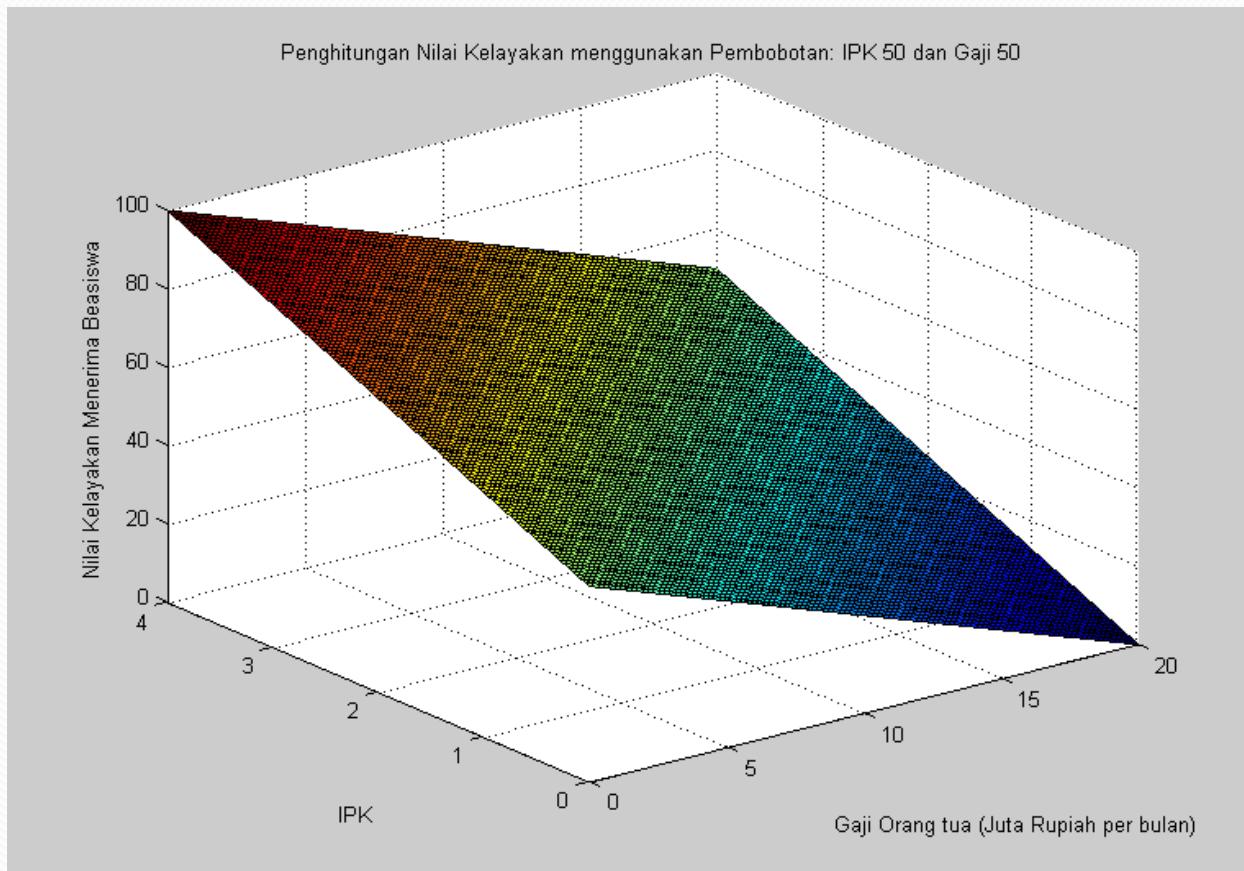
1. IF $\text{Interval}(IPK, P1) \wedge \text{Interval}(Gaji, G1) \Rightarrow NK = 70$
2. IF $\text{Interval}(IPK, P1) \wedge \text{Interval}(Gaji, G2) \Rightarrow NK = 60$
- ...
- ...
- ...
20. IF $\text{Interval}(IPK, P4) \wedge \text{Interval}(Gaji, G5) \Rightarrow NK = 50$

Sistem yang Linier

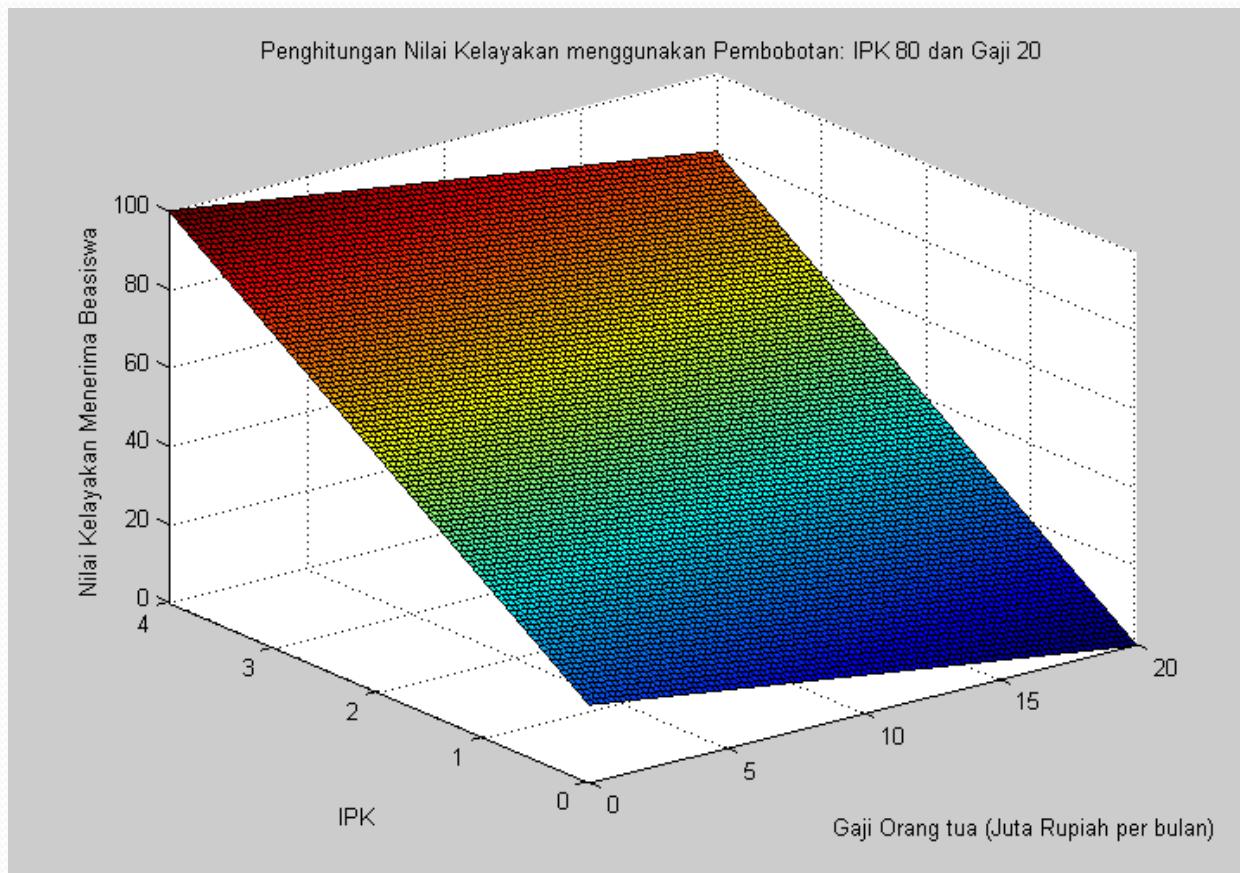
$$NK = W_1(IPK / 4) + W_2((20 - Gaji) / 20)$$

- W_1 adalah bobot untuk IPK, W_2 adalah bobot untuk Gaji.
- Asumsi: IPK maksimum adalah 4,00
- Asumsi: Gaji Orang tua maksimum adalah Rp 20 jt/bln.
- Karena skala untuk NK adalah [0, 100], maka $W_1 + W_2$ harus sama dengan 100.

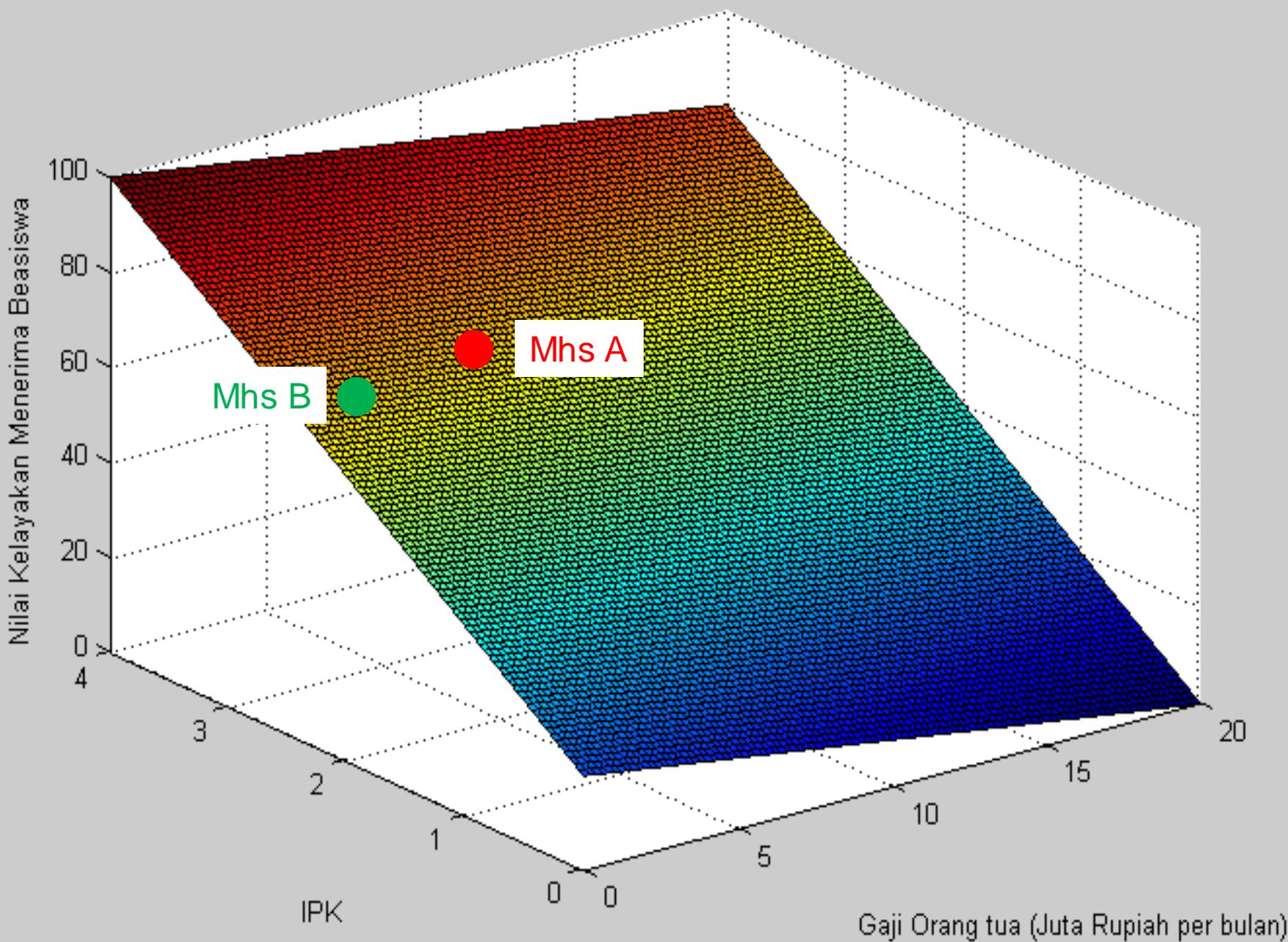
$W_1 = 50$ dan $W_2 = 50$



$W_1 = 80$ dan $W_2 = 20$

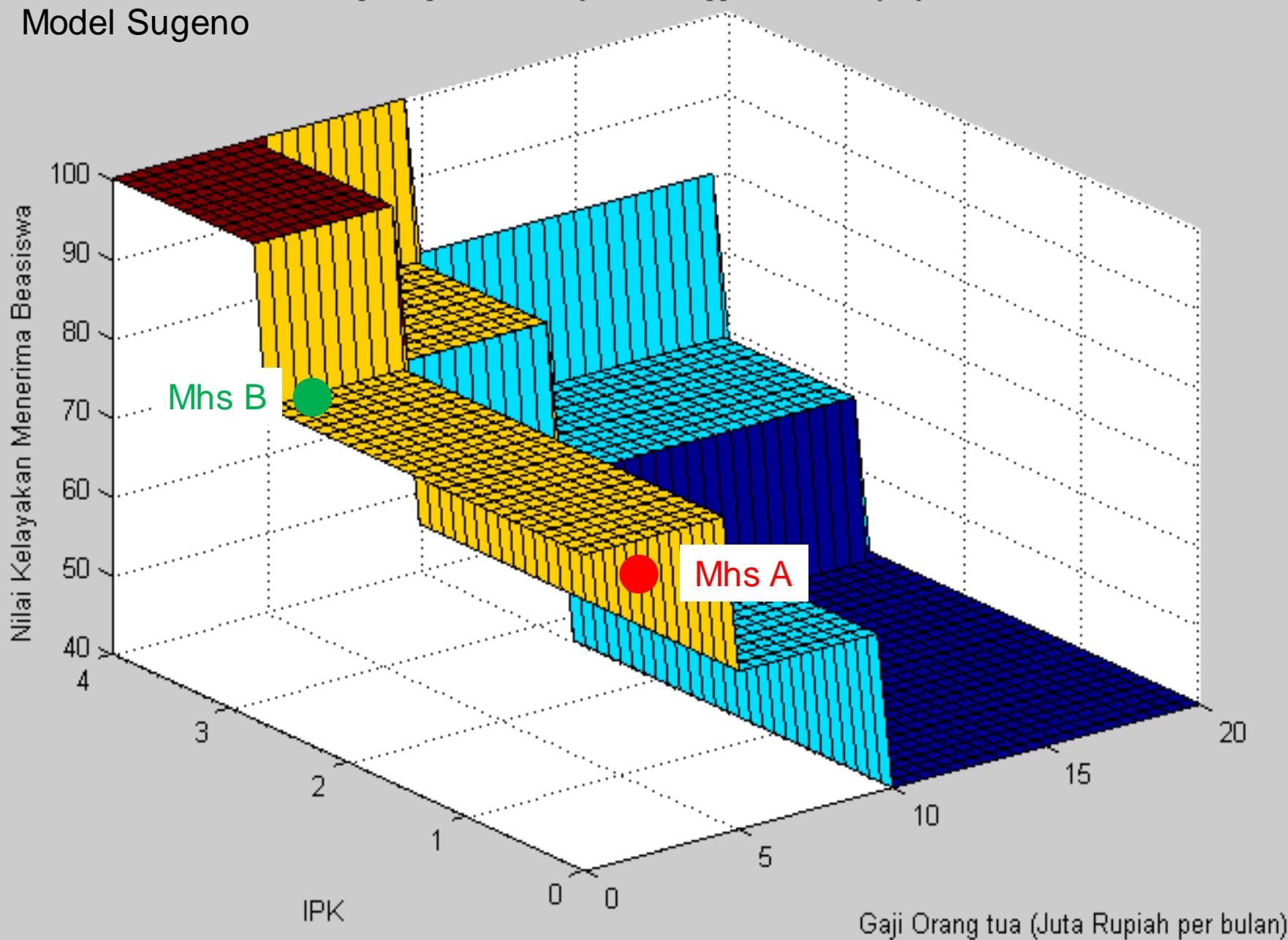


Penghitungan Nilai Kelayakan menggunakan Pembobotan: IPK 80 dan Gaji 20

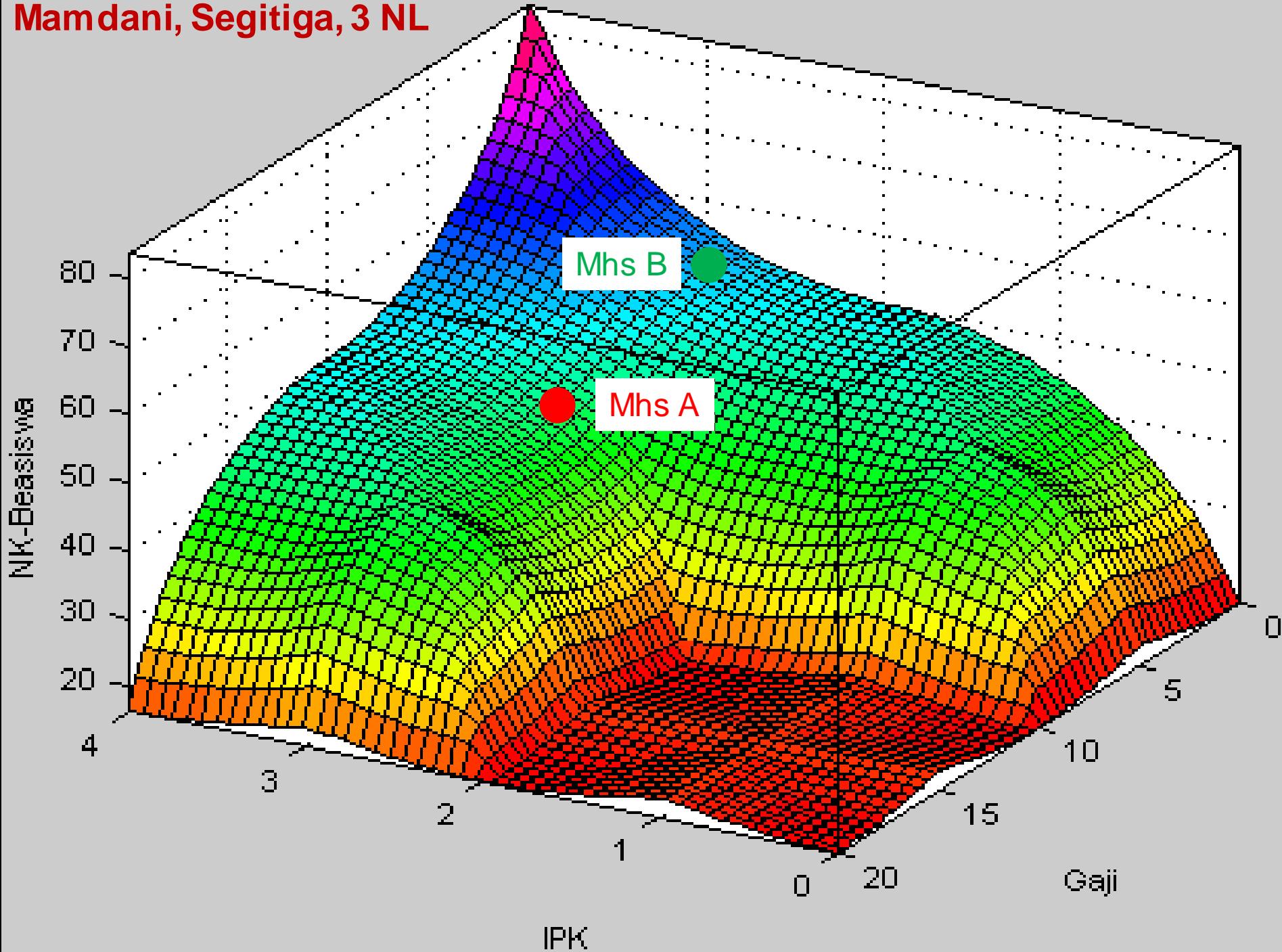


Penghitungan Nilai Kelayakan menggunakan Fuzzy Systems

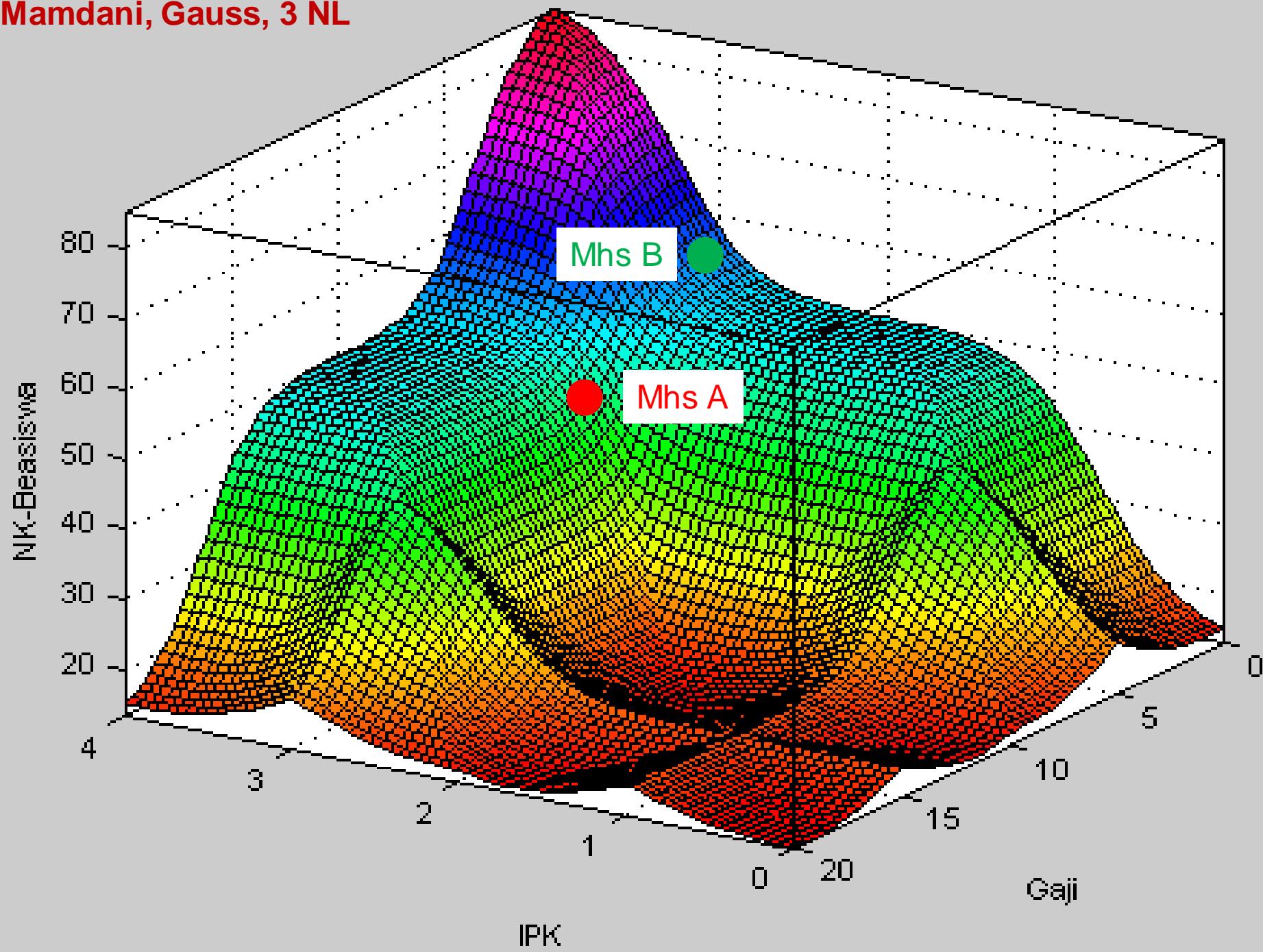
Model Sugeno



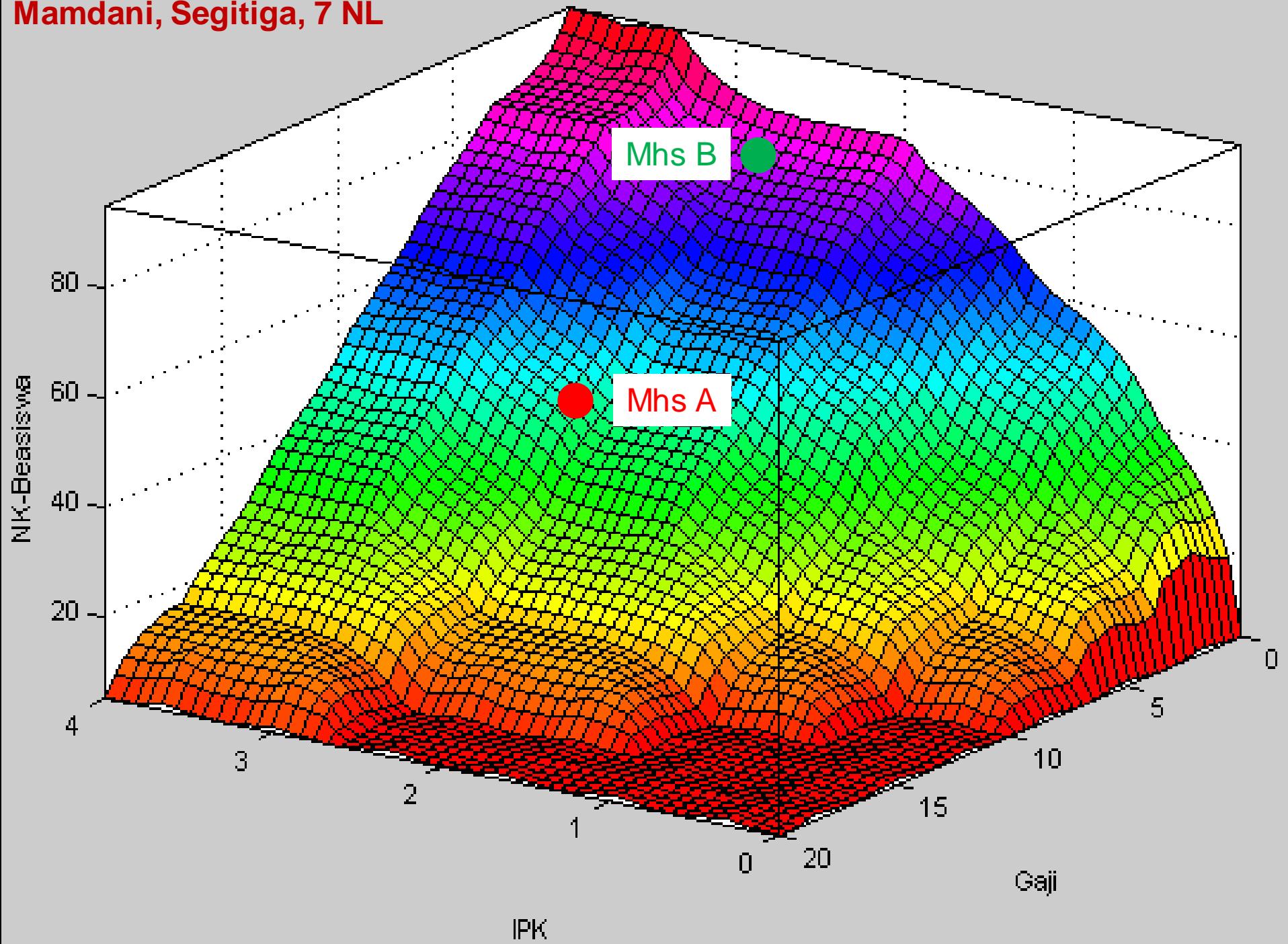
Mamdani, Segitiga, 3 NL



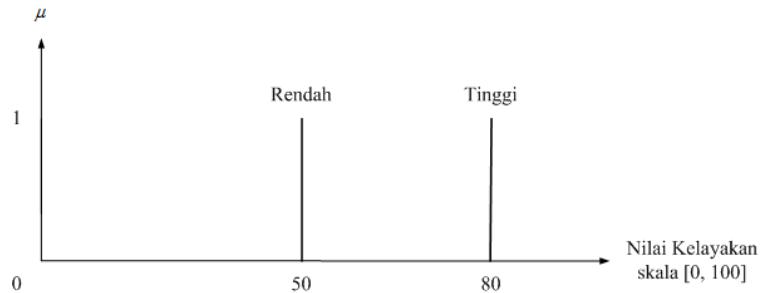
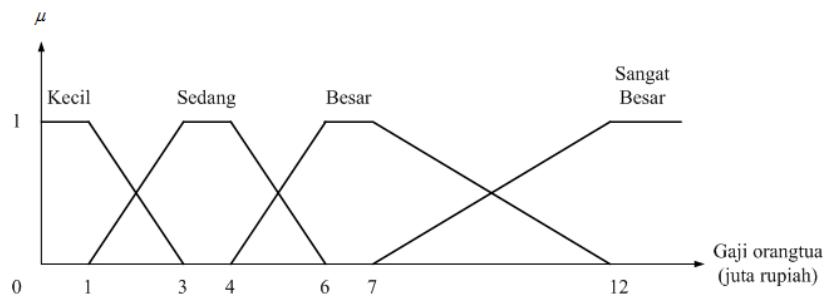
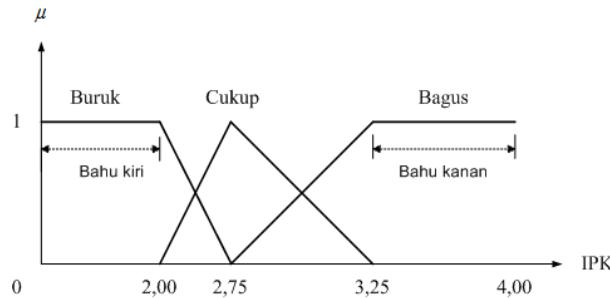
Mamdani, Gauss, 3 NL



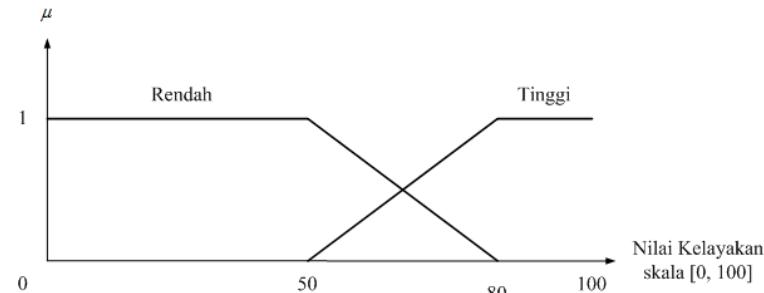
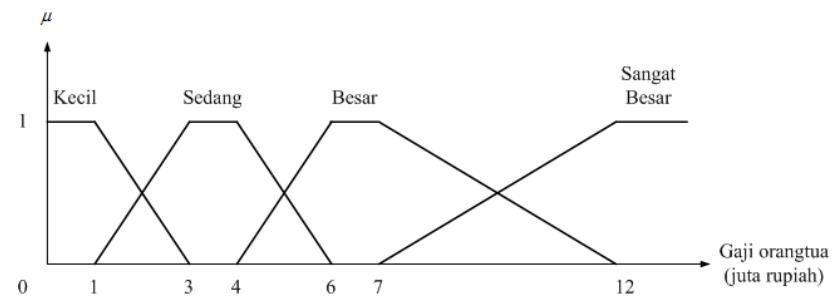
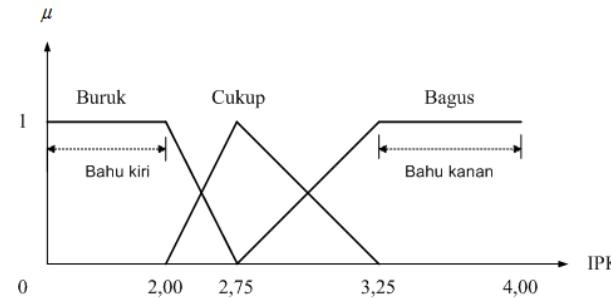
Mamdani, Segitiga, 7 NL



Model Sugeno



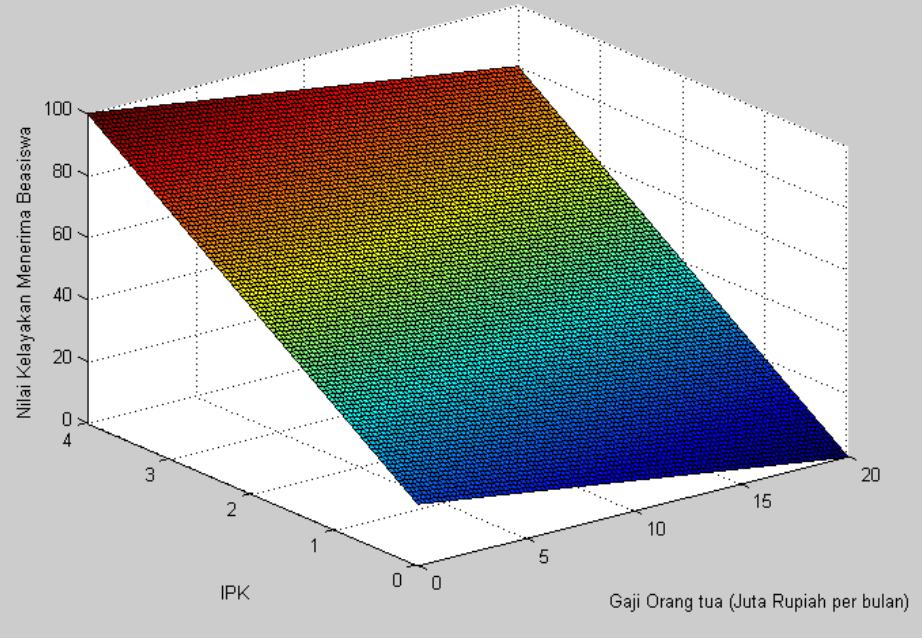
Model Mamdani



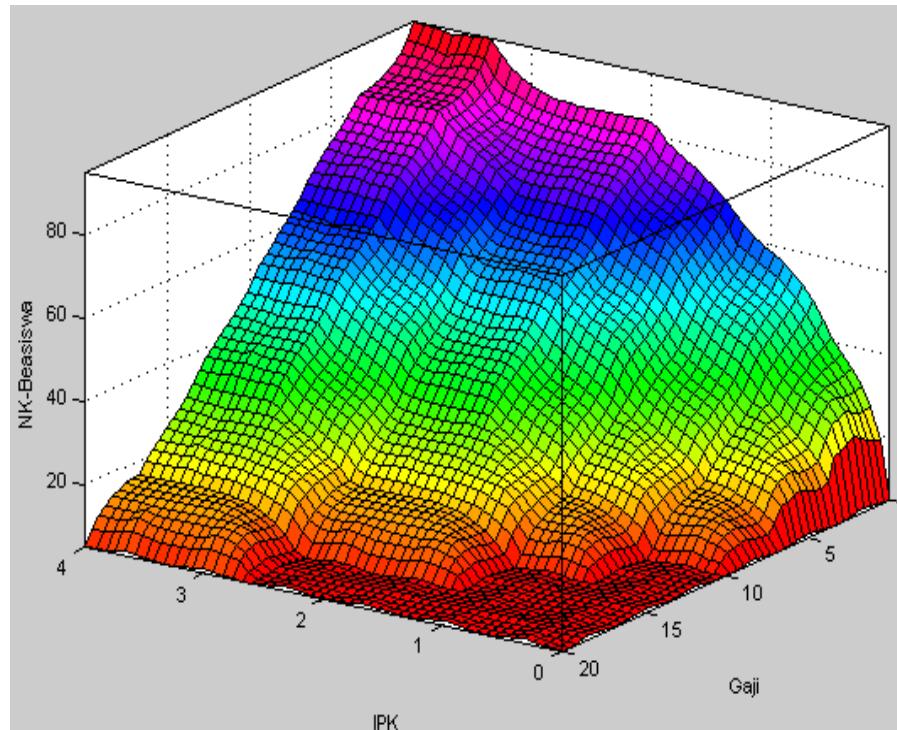
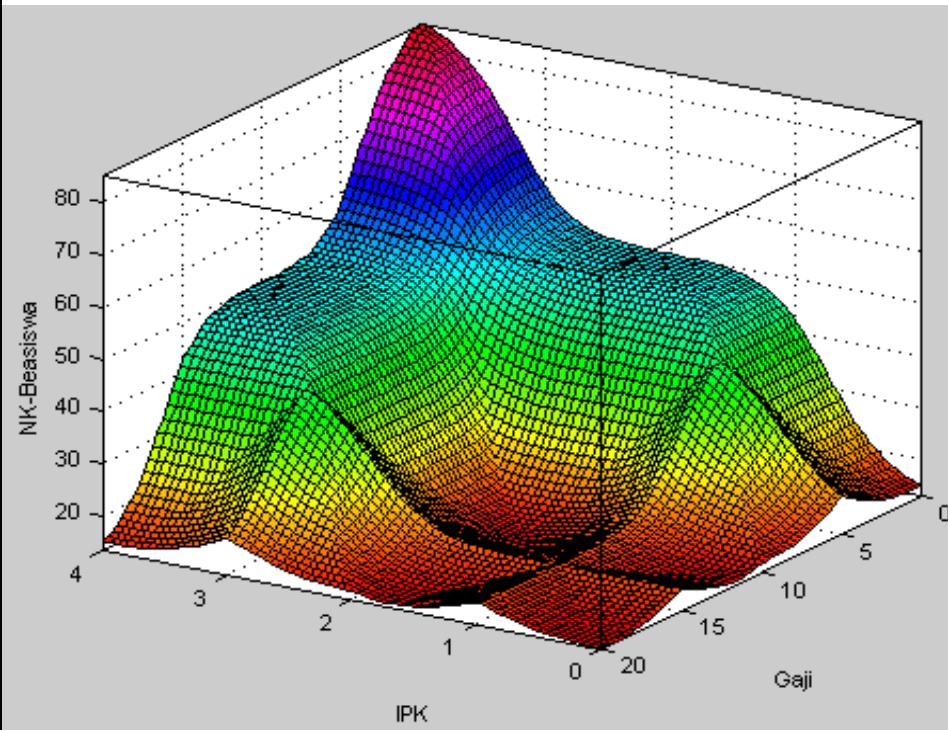
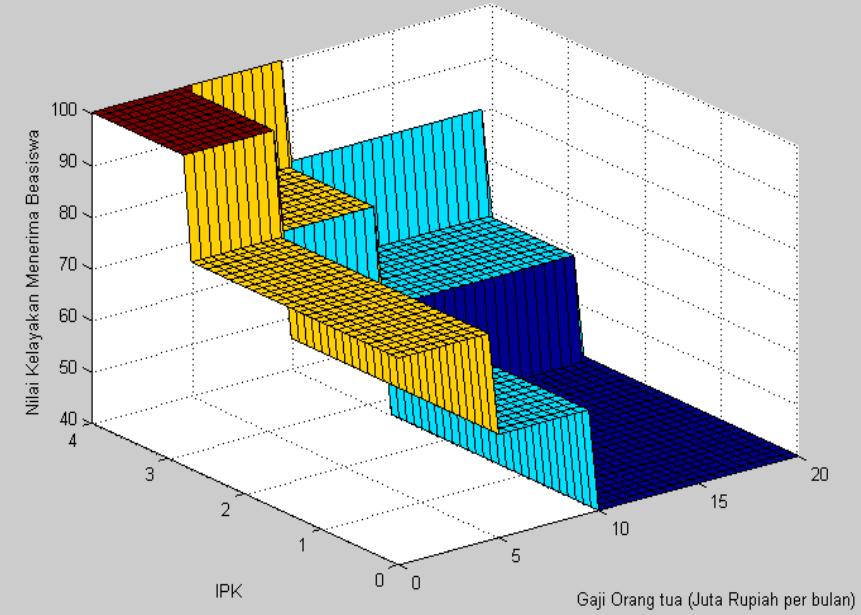
IPK \ Gaji	Kecil	Sedang	Besar	Sangat Besar
Buruk	Rendah	Rendah	Rendah	Rendah
Cukup	Tinggi	Rendah	Rendah	Rendah
Bagus	Tinggi	Tinggi	Tinggi	Rendah

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

Penghitungan Nilai Kelayakan menggunakan Pembobotan: IPK 80 dan Gaji 20



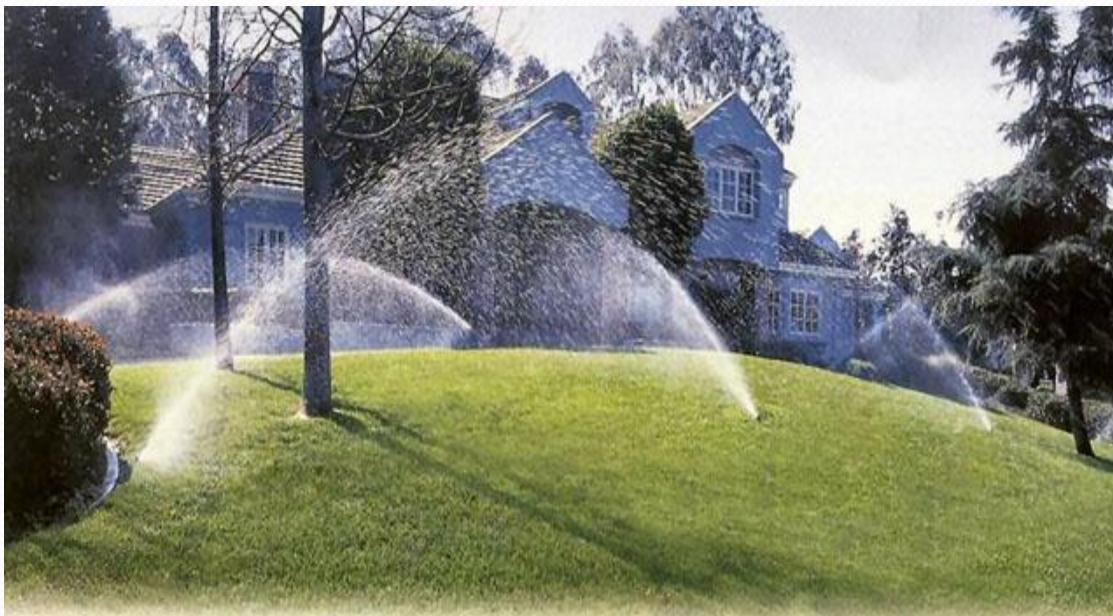
Penghitungan Nilai Kelayakan menggunakan Fuzzy Systems





Kasus 2: *Sprinkler CS*

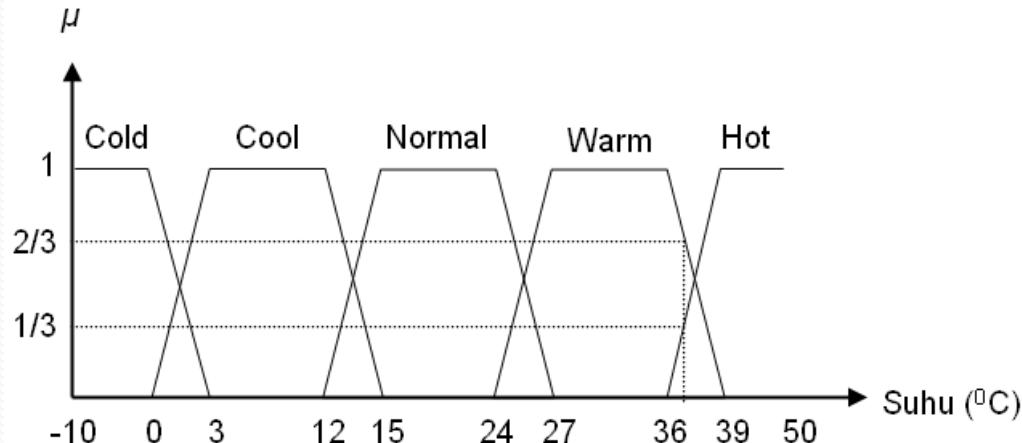
- Sistem kontrol untuk penyiraman air.
 - *Input:* Suhu Udara ($^{\circ}\text{C}$) & Kelembaban Tanah (%)
 - *Output:* Durasi Penyiraman (menit)
-
- Nilai *crisp* yang diterima oleh sensor suhu adalah 37°C dan sensor kelembaban adalah 12%.
 - Berapa lama durasi penyiraman yang dilakukan?



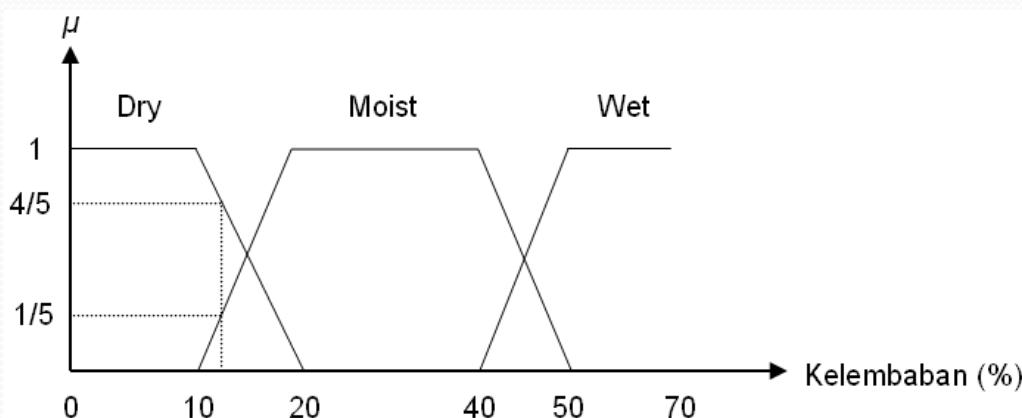
Kasus 2: *Sprinkler CS*

- Model Mamdani atau Sugeno?
- Jumlah Nilai Linguistik untuk setiap variabel?
- Fungsi Keanggotaan: Segitiga, trapesium, phi, ...?
- Batas-batas Nilai Linguistik?
- *Fuzzy rule* yang tepat?

Proses fuzzification



- Warm = $-(37-39)/(39-36) = 2/3$.
- Hot = $(37-36)/(39-36) = 1/3$.



- Dry = $-(12-20)/(20-10) = 4/5$
- Moist = $(12-10)/(20-10) = 1/5$

Proses inferensi

Antecedent 1 (Temperature)

Antecedent 2
(Moisture)

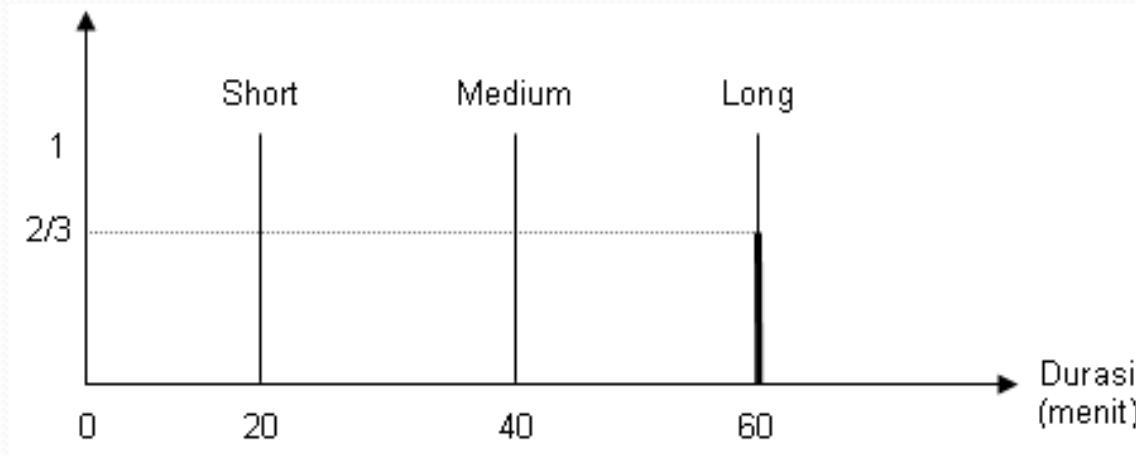
	Cold	Cool	Normal	Warm	Hot
Dry	L	L	L	L	L
Moist	L	M	M	M	M
Wet	S	S	S	S	S

Note: L = Long, M = Medium, S = Short

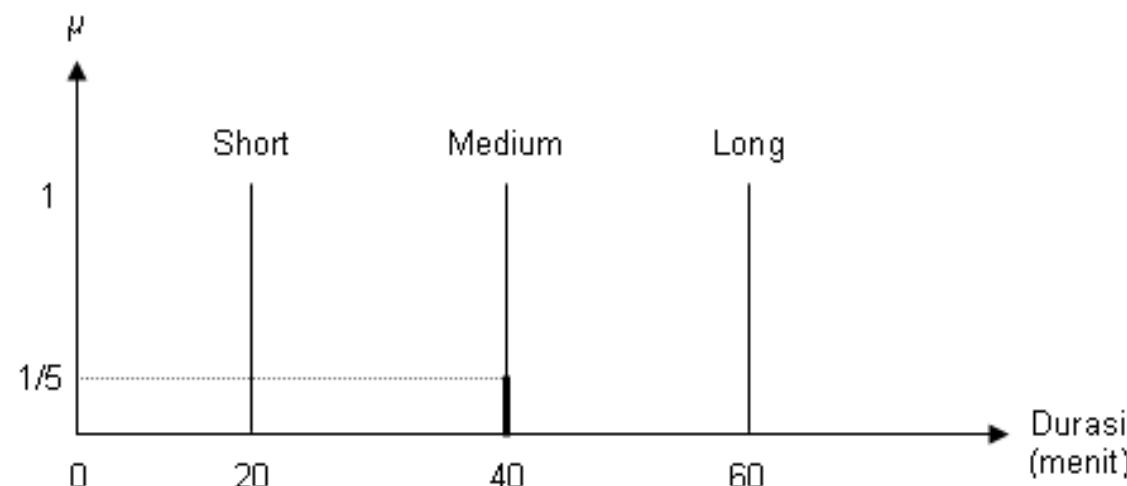
Proses *inference*

- Dari empat data fuzzy *input* di atas, Warm (2/3), Hot (1/3), Dry (4/5) dan Moist (1/5), maka kita mendapatkan 4 aturan (dari 15 aturan) yang dapat diaplikasikan:
 - IF Suhu is Warm (2/3) AND Kelembaban is Dry (4/5)
THEN Durasi is **Long** (2/3)
 - IF Suhu is Warm (2/3) AND Kelembaban is Moist (1/5)
THEN Durasi is **Medium** (1/5)
 - IF Suhu is Hot (1/3) AND Kelembaban is Dry (4/5)
THEN Durasi is **Long** (1/3)
 - IF Suhu is Hot (1/3) AND Kelembaban is Moist (1/5)
THEN Durasi is **Medium** (1/5)

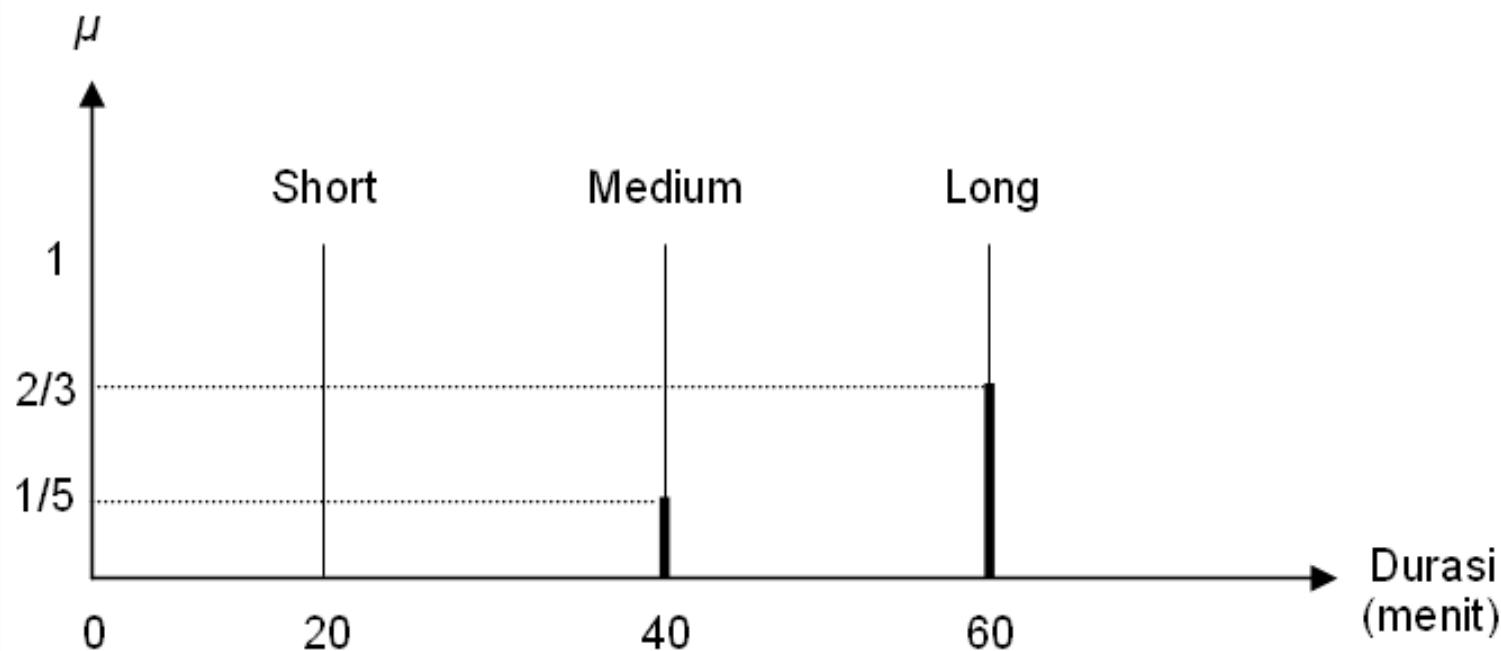
Inference: Model Sugeno



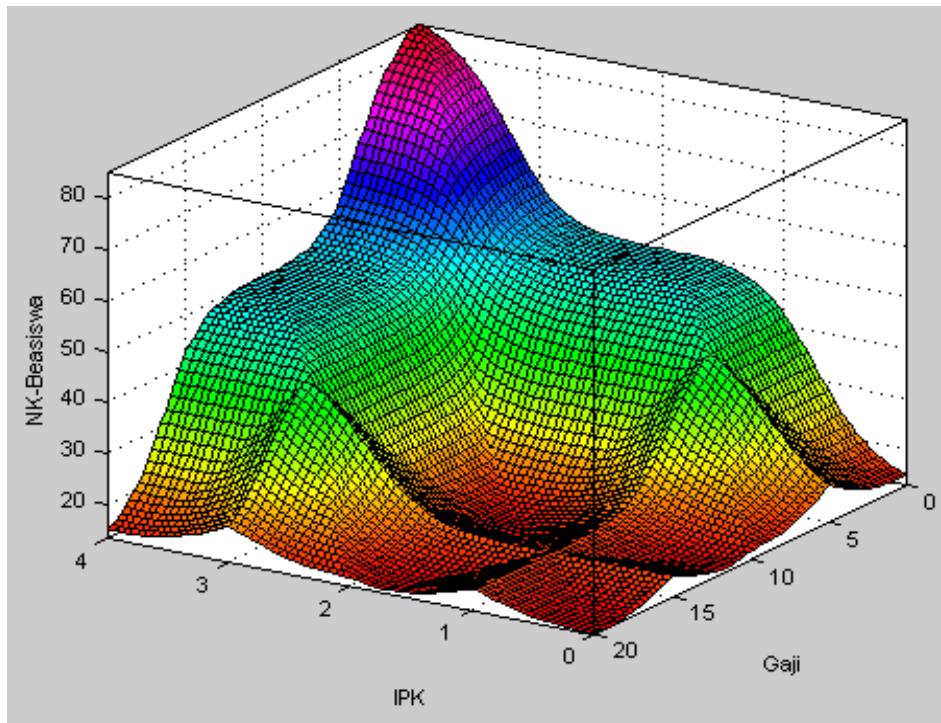
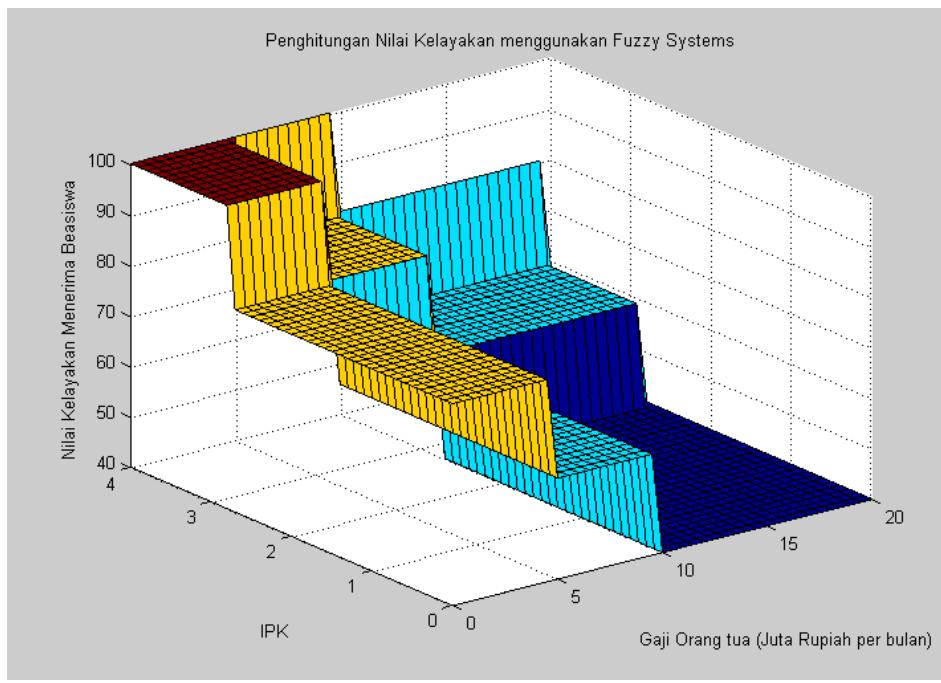
(a)



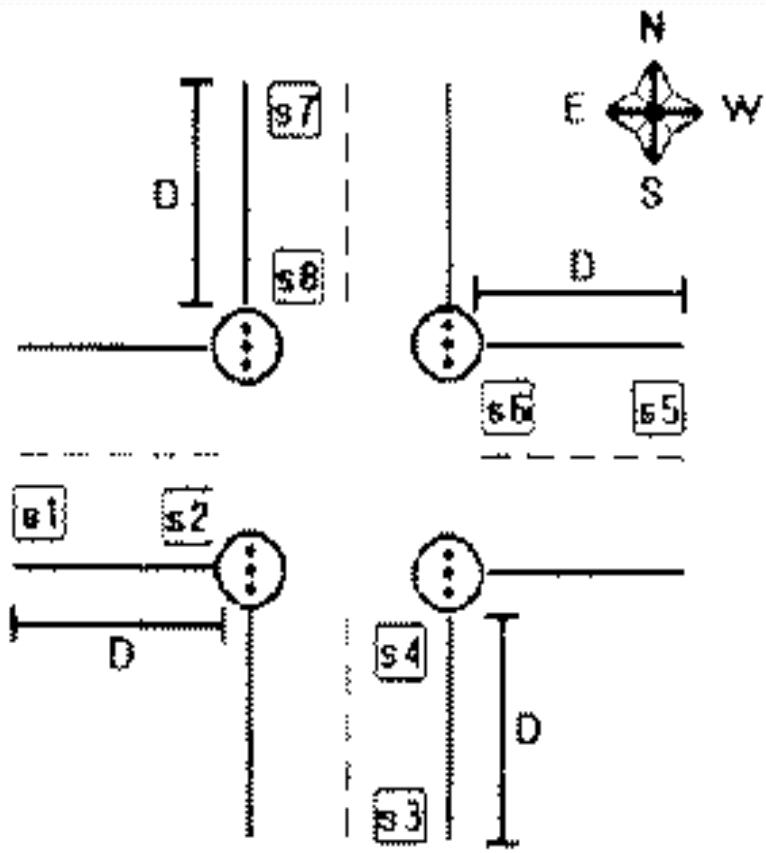
Defuzzyfication: Weighted Average



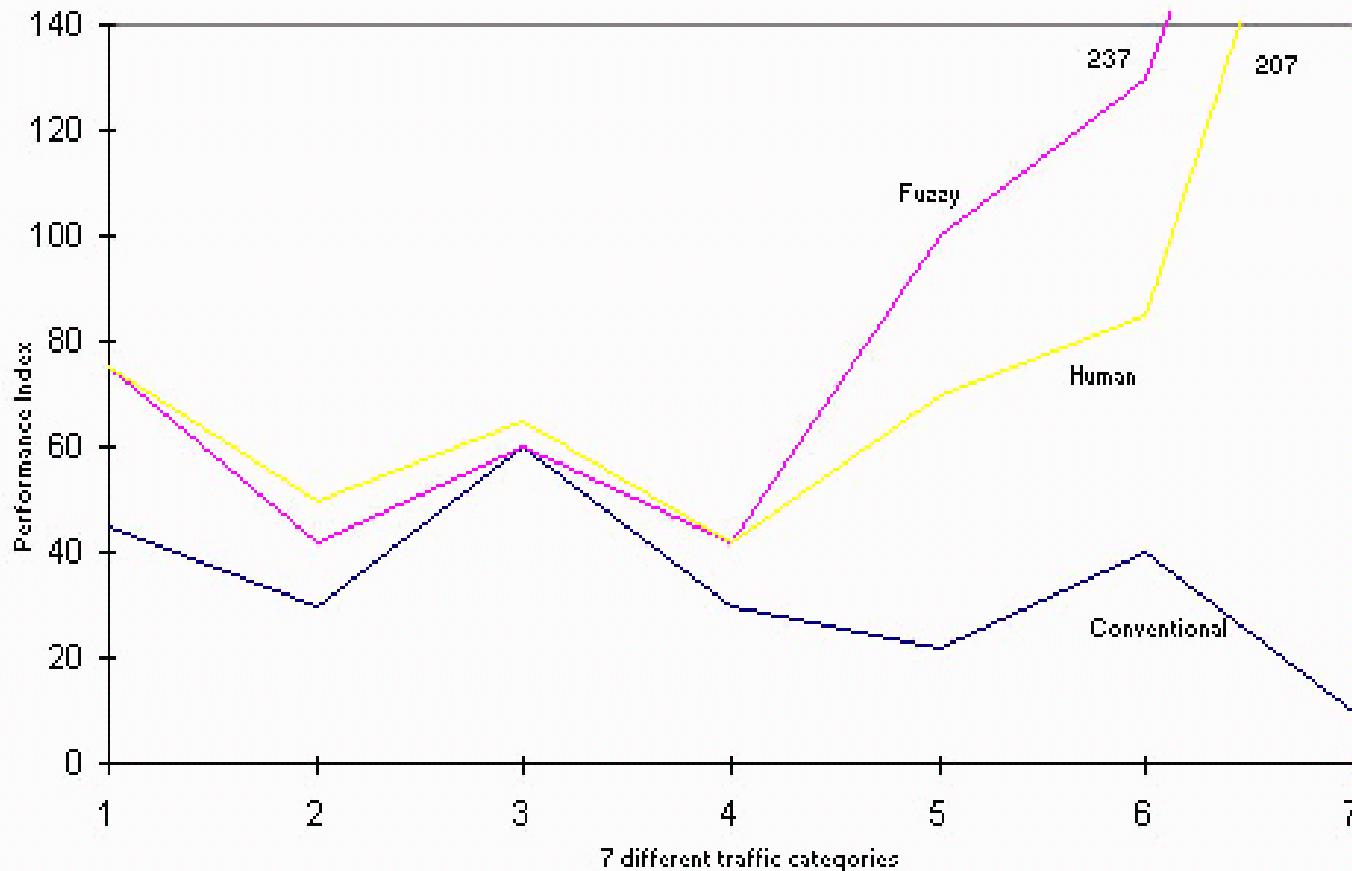
$$y^* = \frac{1/5(40) + 2/3(60)}{1/5 + 2/3} = 55,38$$



Kasus 3: *Traffic Light Controller*



Kasus 3: *Traffic Light Controller*



Kasus 4: Akreditasi Prodi

- Pendidikan dan Pengajaran
 - Pendidikan Dosen: S₁, S₂, S₃?
 - Jabatan Akademik: AA, L, LK, GB?
 - Rasio Dosen : Mhs?
- Penelitian
 - Dana proyek?
 - Paper Ilmiah?
- Pengabdian Masyarakat
 - Jumlah kegiatan per tahun?
 - Manfaat bagi masyarakat?

Kasus 5: *University Rankings*

- Bagaimana mengelompokkan universitas ke dalam ***cluster*** yang secara statistik mirip untuk semua kriteria yg ada?
- ***Clustering harus*** dilakukan tanpa asumsi apapun tentang tingkat kepentingan relatif dari semua kriteria.
- Kriteria:
 - Jumlah profesor?
 - Jumlah mhs asing?
 - Rasio Dosen:Mhs?
 - Besarnya Dana proyek?
 - Jumlah Paper Ilmiah?

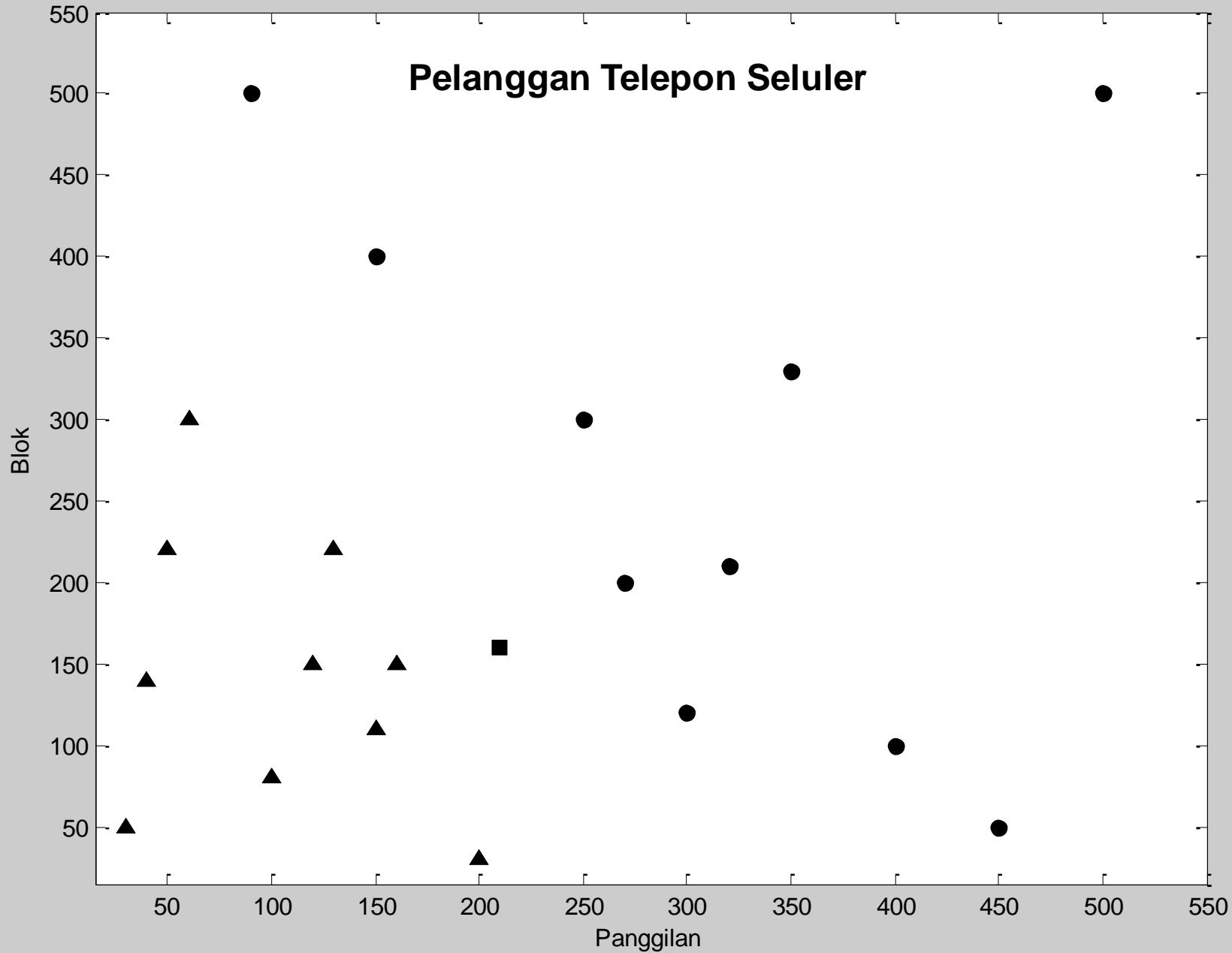
Kasus 5: *University Rankings*

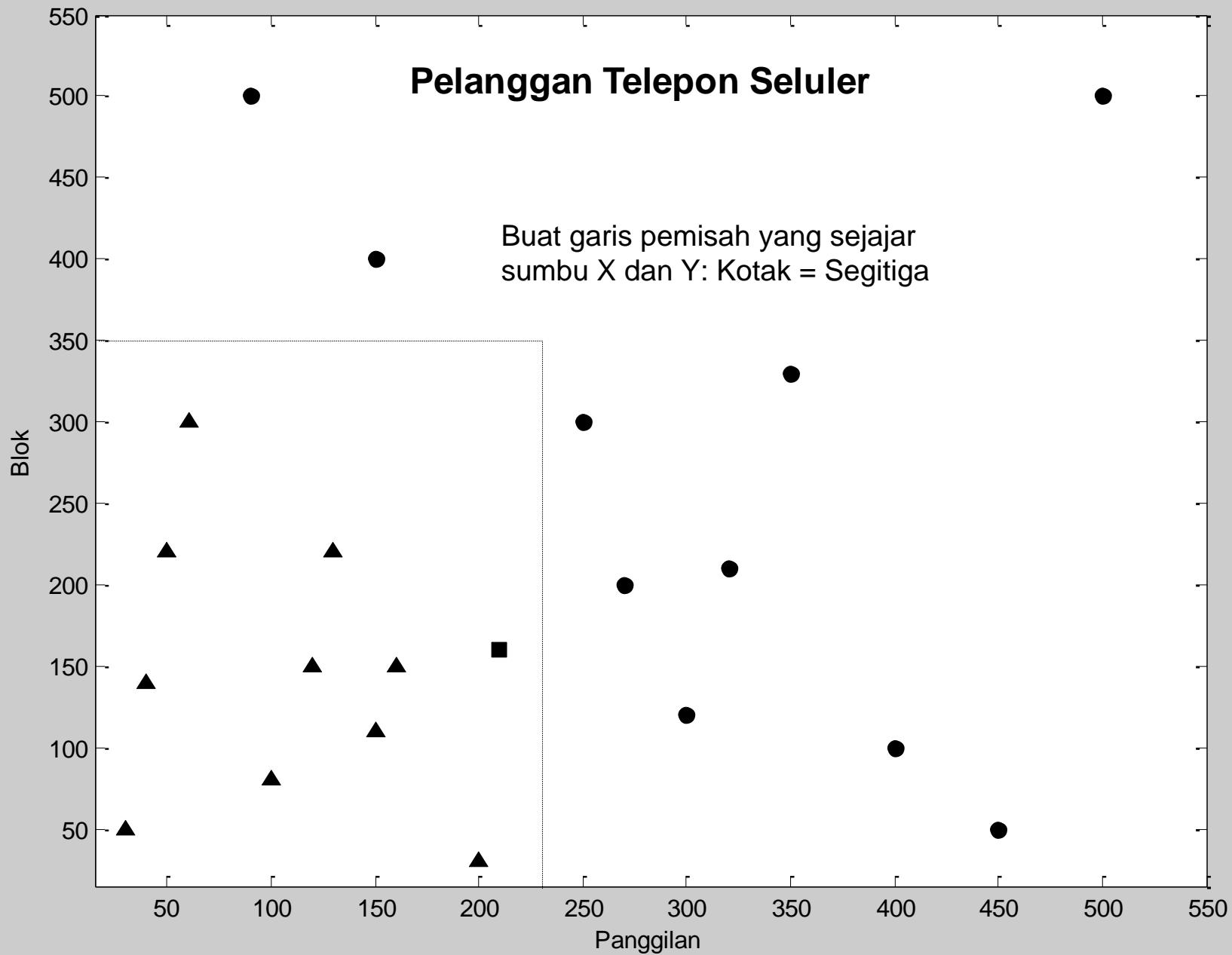
- Shanghai Jao Tong University (SJTU)
- The Time Higher League Table (THES)
- Metode: ***Fuzzy Clustering***
- Top 500 Univerisities in the world
 1. Harvard
 2. Stanford
 3. Cambridge

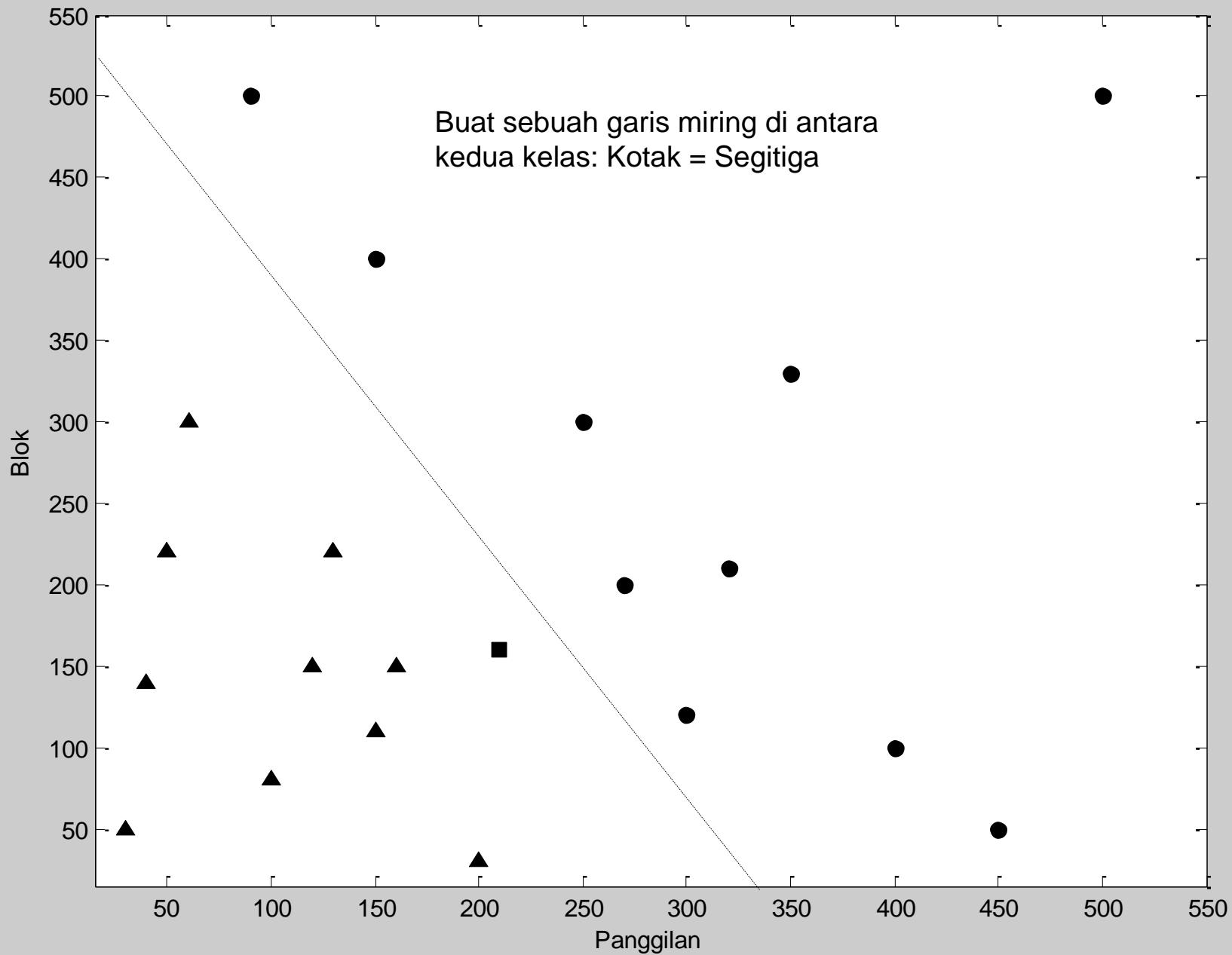
Data Pelanggan Telepon Seluler

Panggilan	Blok	Bonus
30	50	Tidak
40	140	Tidak
50	220	Tidak
60	300	Tidak
100	80	Tidak
120	150	Tidak
130	220	Tidak
150	110	Tidak
160	150	Tidak
200	30	Tidak

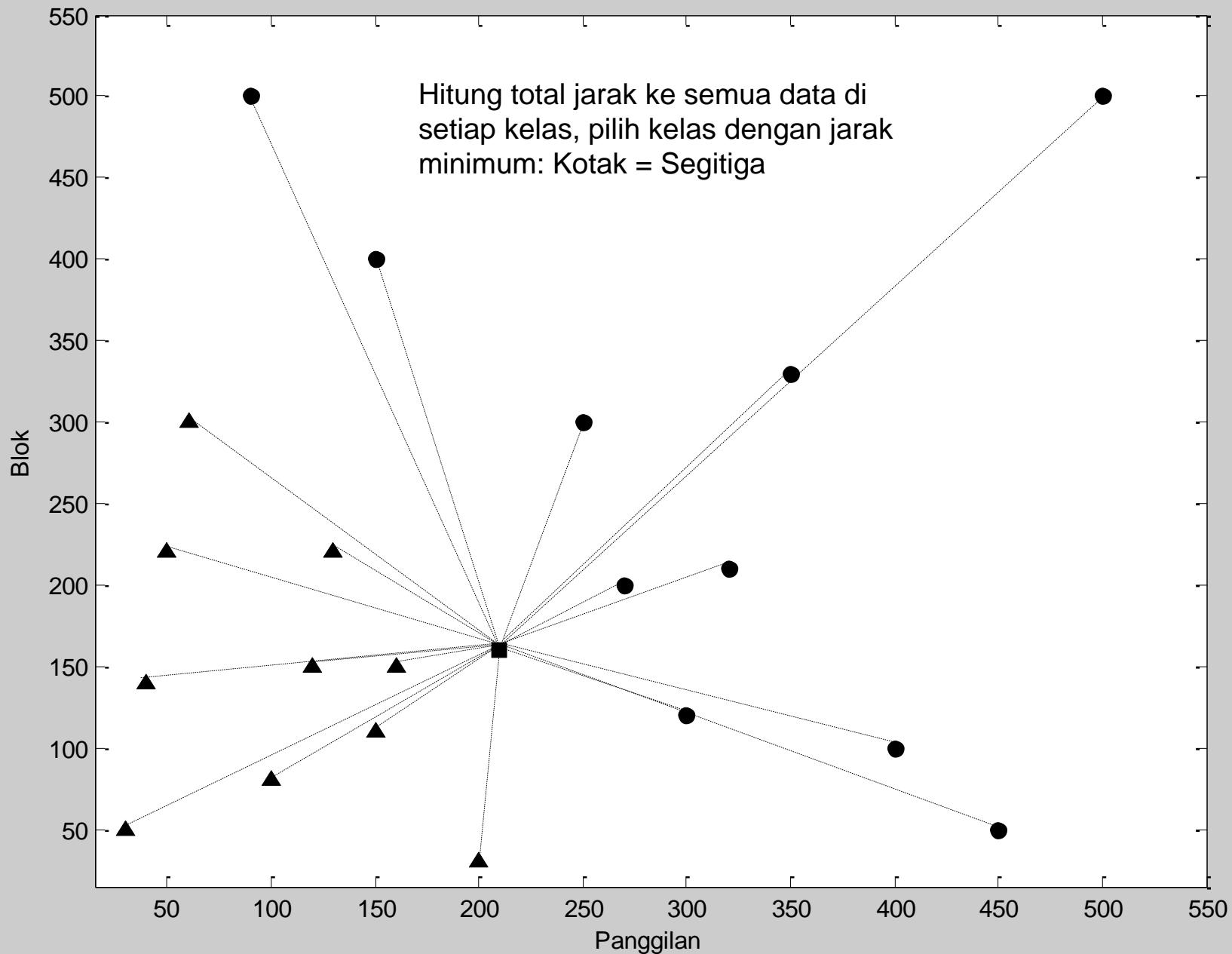
Panggilan	Blok	Bonus
90	500	Ya
150	400	Ya
250	300	Ya
270	200	Ya
300	120	Ya
320	210	Ya
350	330	Ya
400	100	Ya
450	50	Ya
500	500	Ya

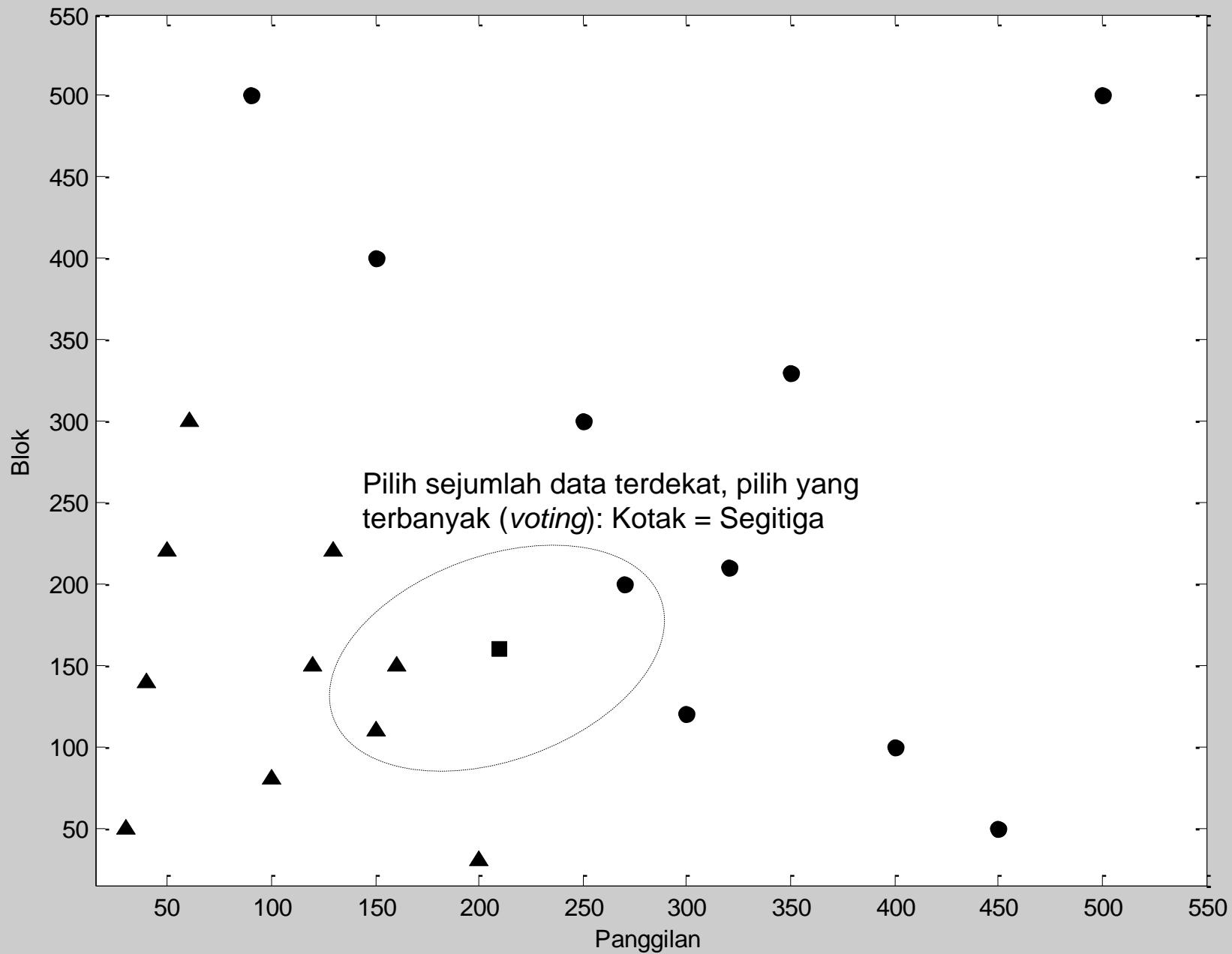


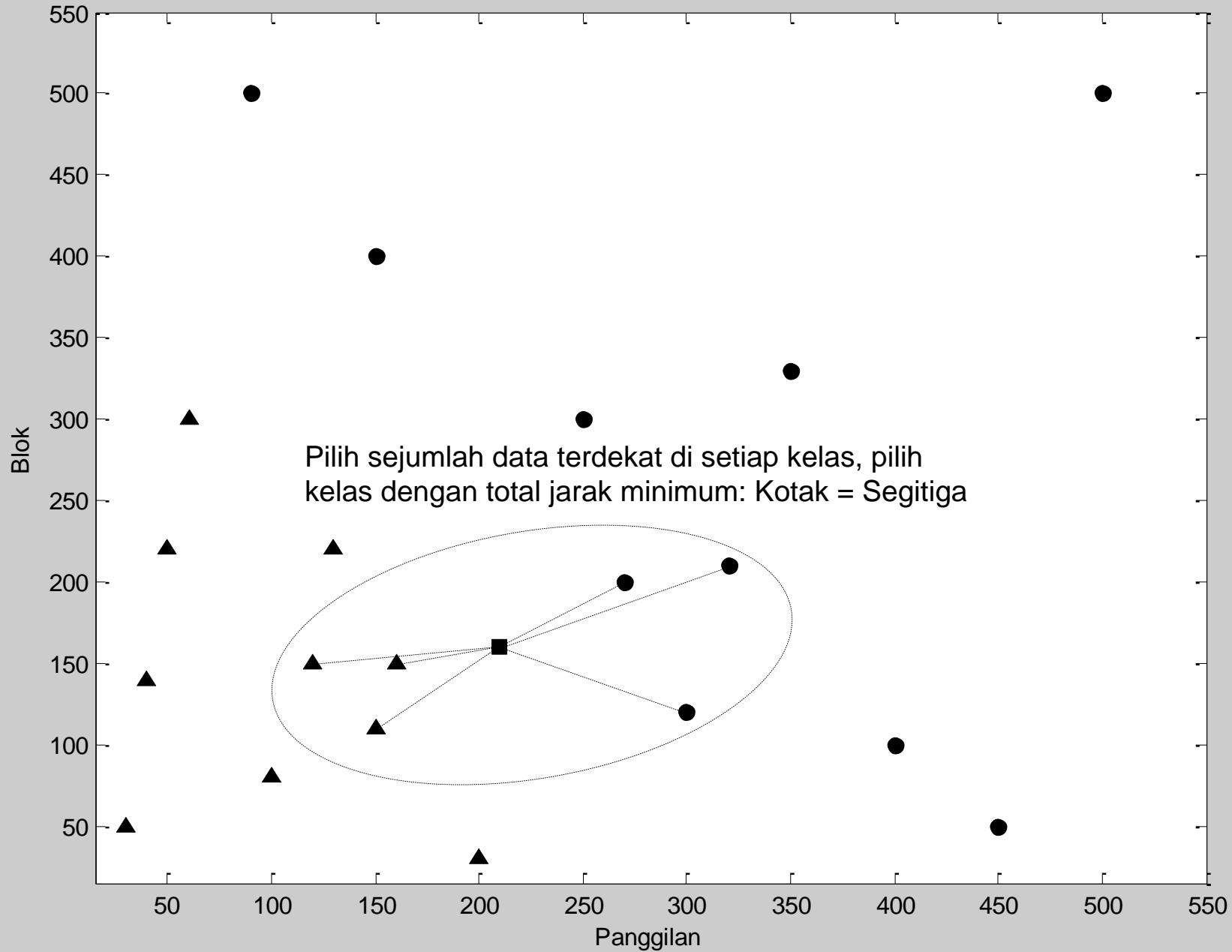


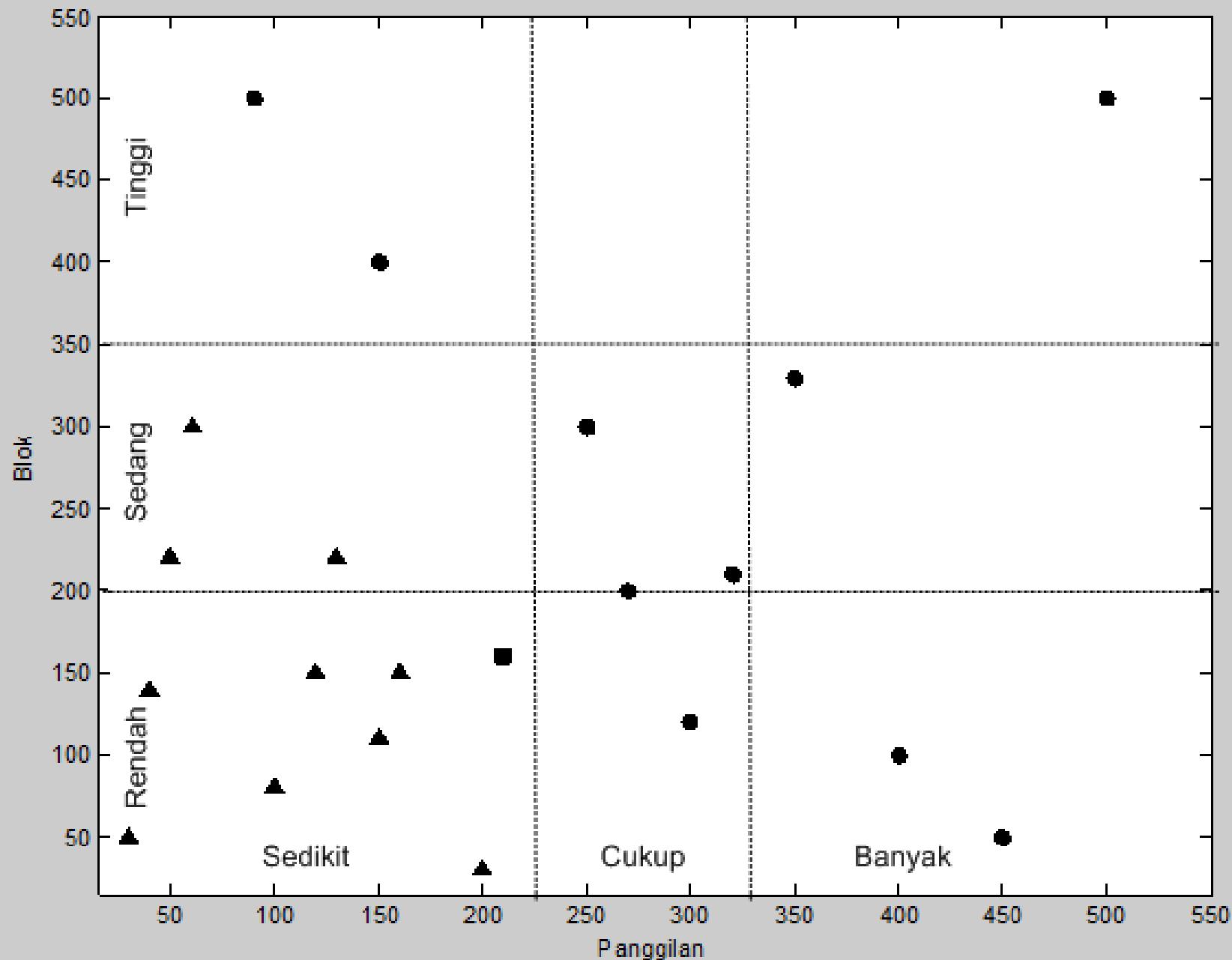


Buat sebuah garis miring di antara
kedua kelas: Kotak = Segitiga

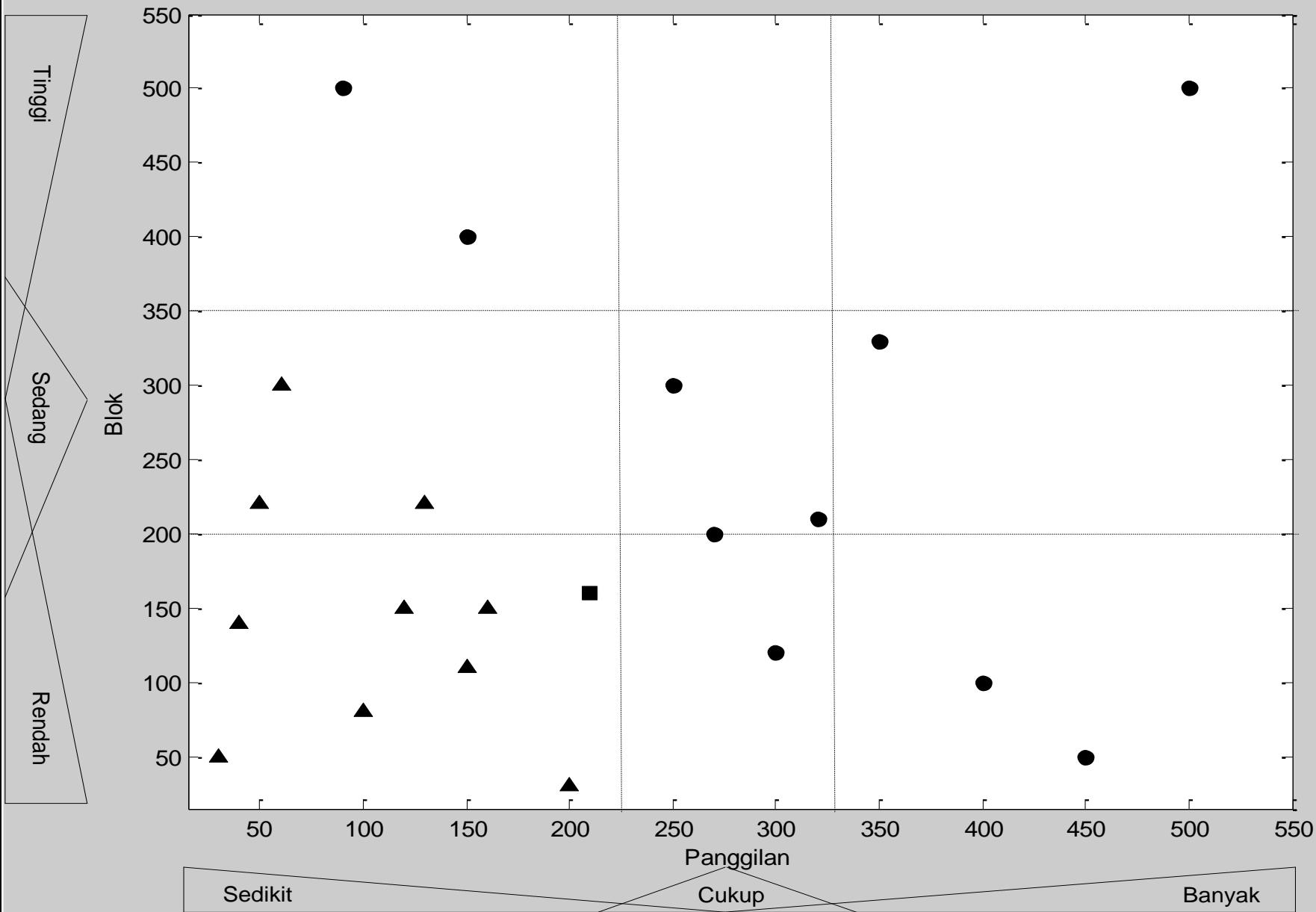




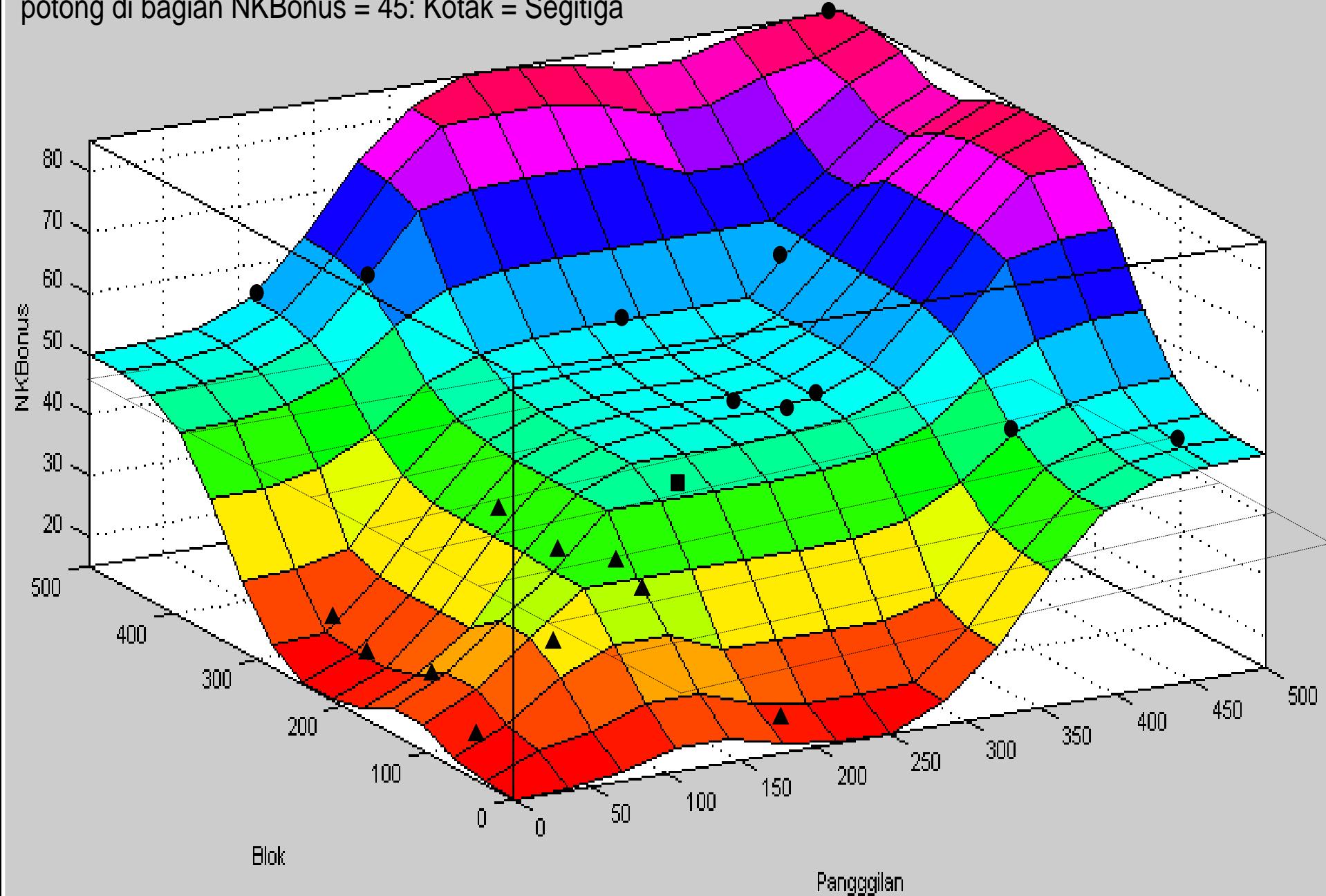


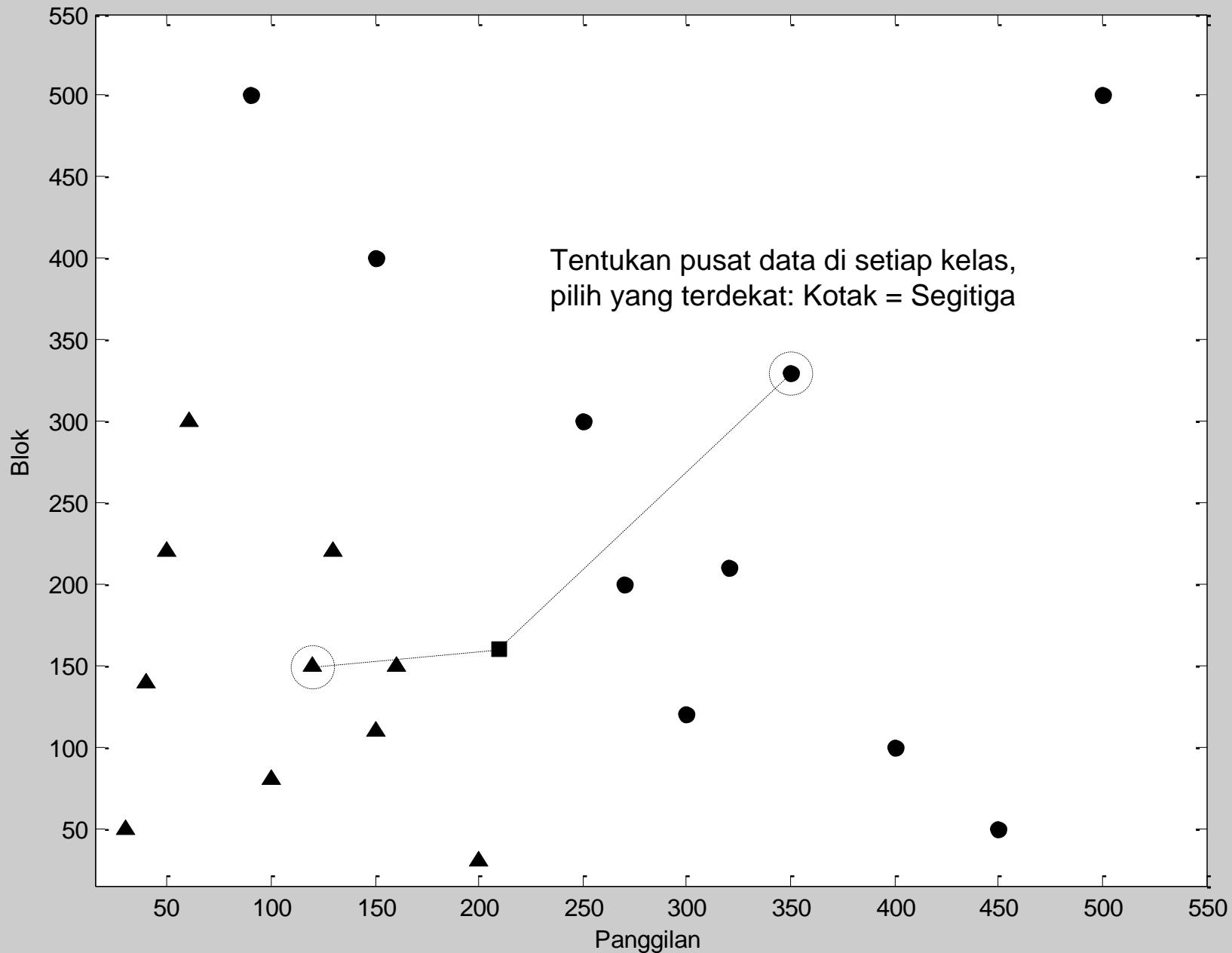


Bagi Panggilan dan Blok ke dalam tiga nilai linguistik, lalu lakukan *fuzzification* dan inferensi *berbasis fuzzy rules* untuk menentukan kelas. Kotak = Segitiga



Buat fungsi yang menarik semua data ke atas, lalu potong di bagian $NKBonus = 45$: Kotak = Segitiga



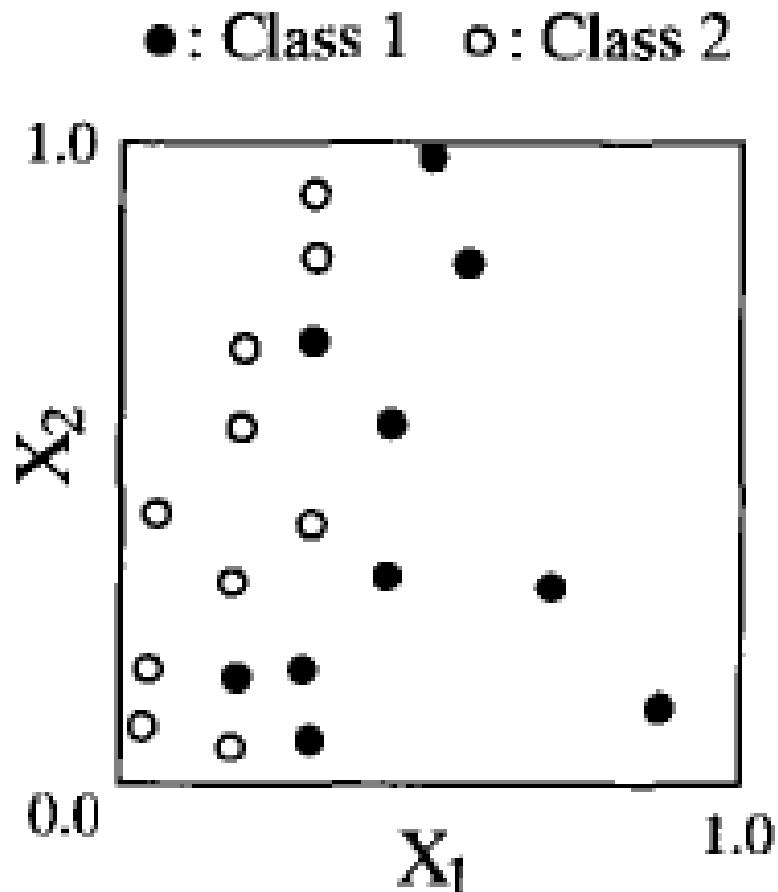


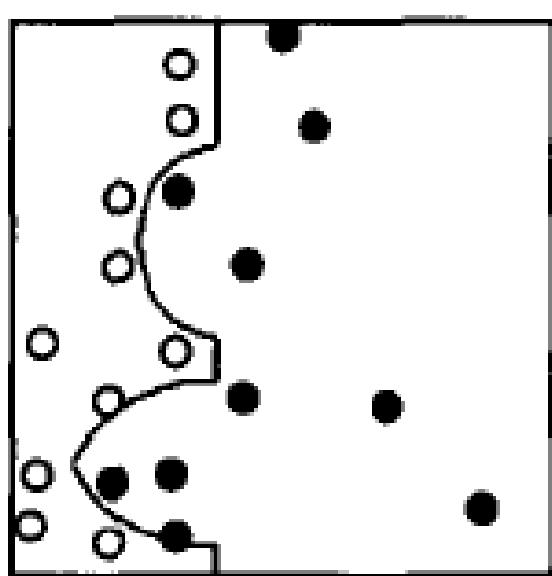
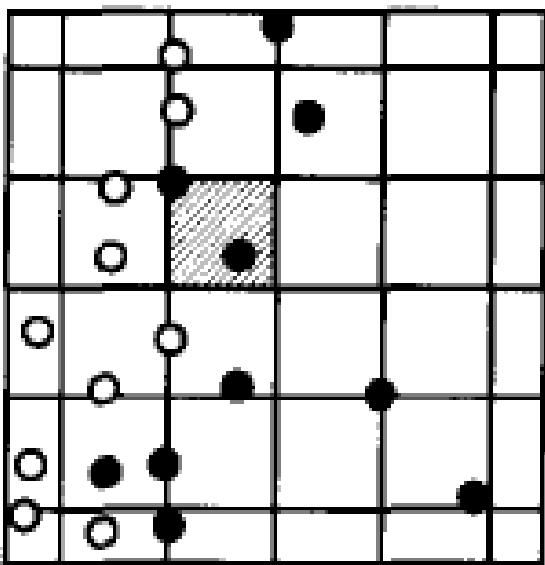
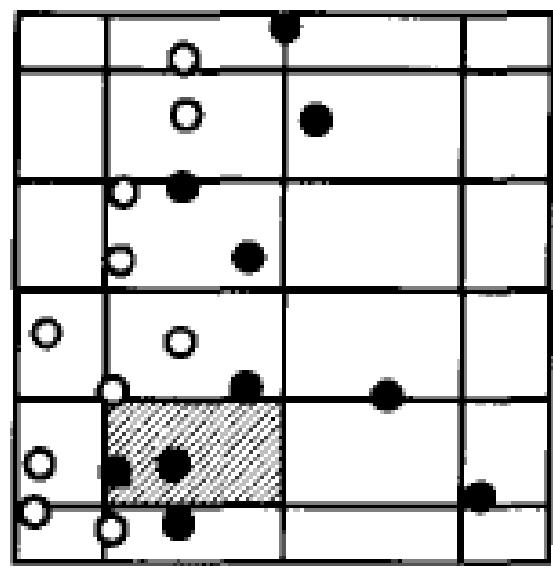
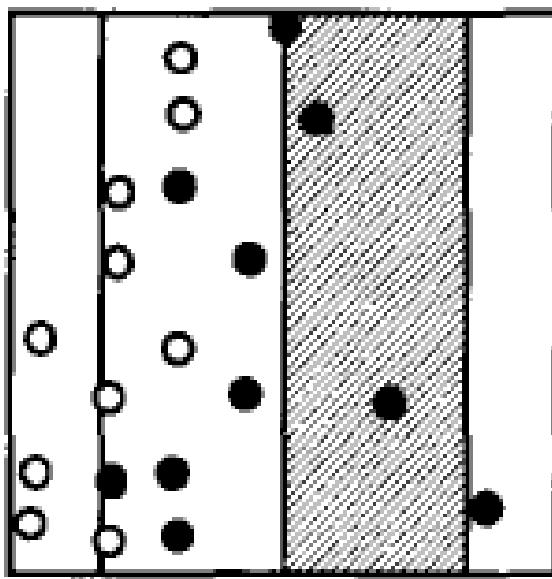
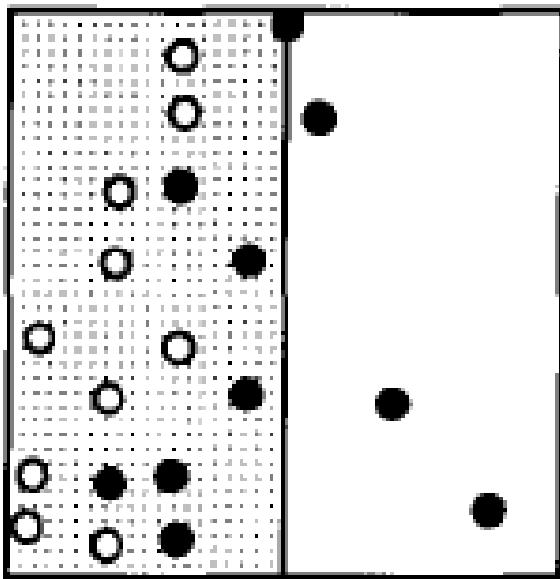
Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms

Hisao Ishibuchi*, Ken Nozaki, Naohisa Yamamoto, Hideo Tanaka

Department of Industrial Engineering, University of Osaka Prefecture, Gakuen-cho 1-1, Sakai, Osaka 593, Japan

Received January 1994; revised February 1994

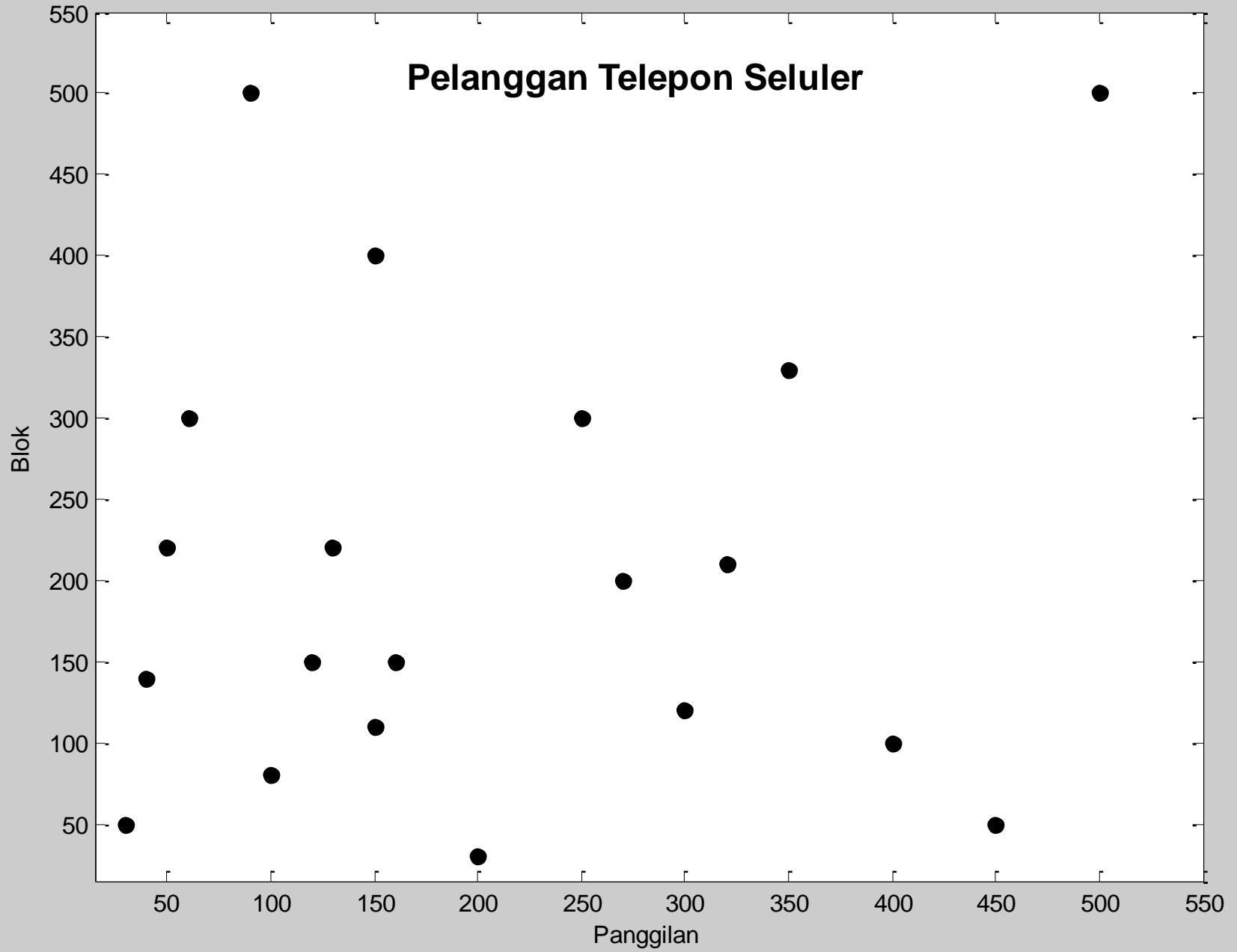


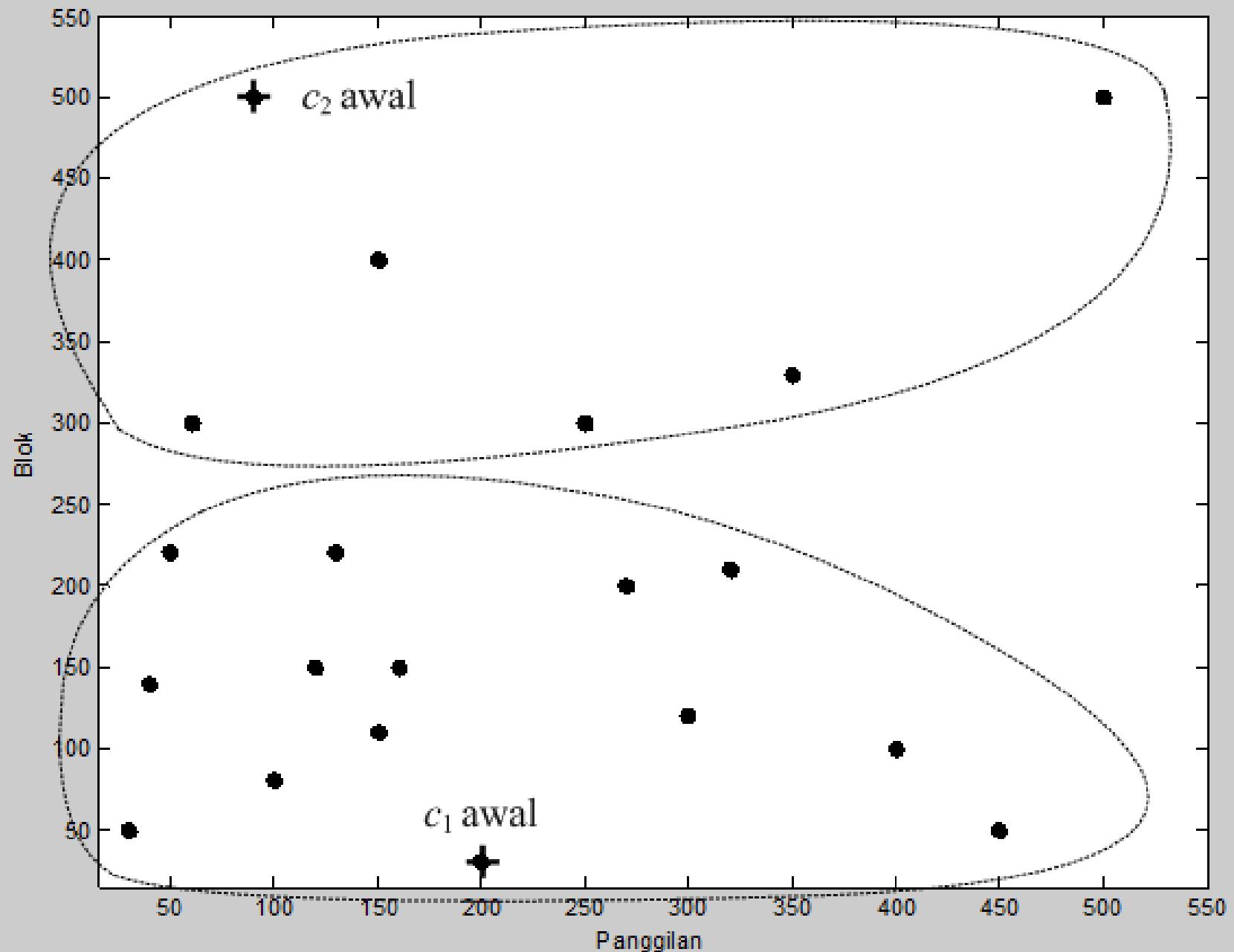


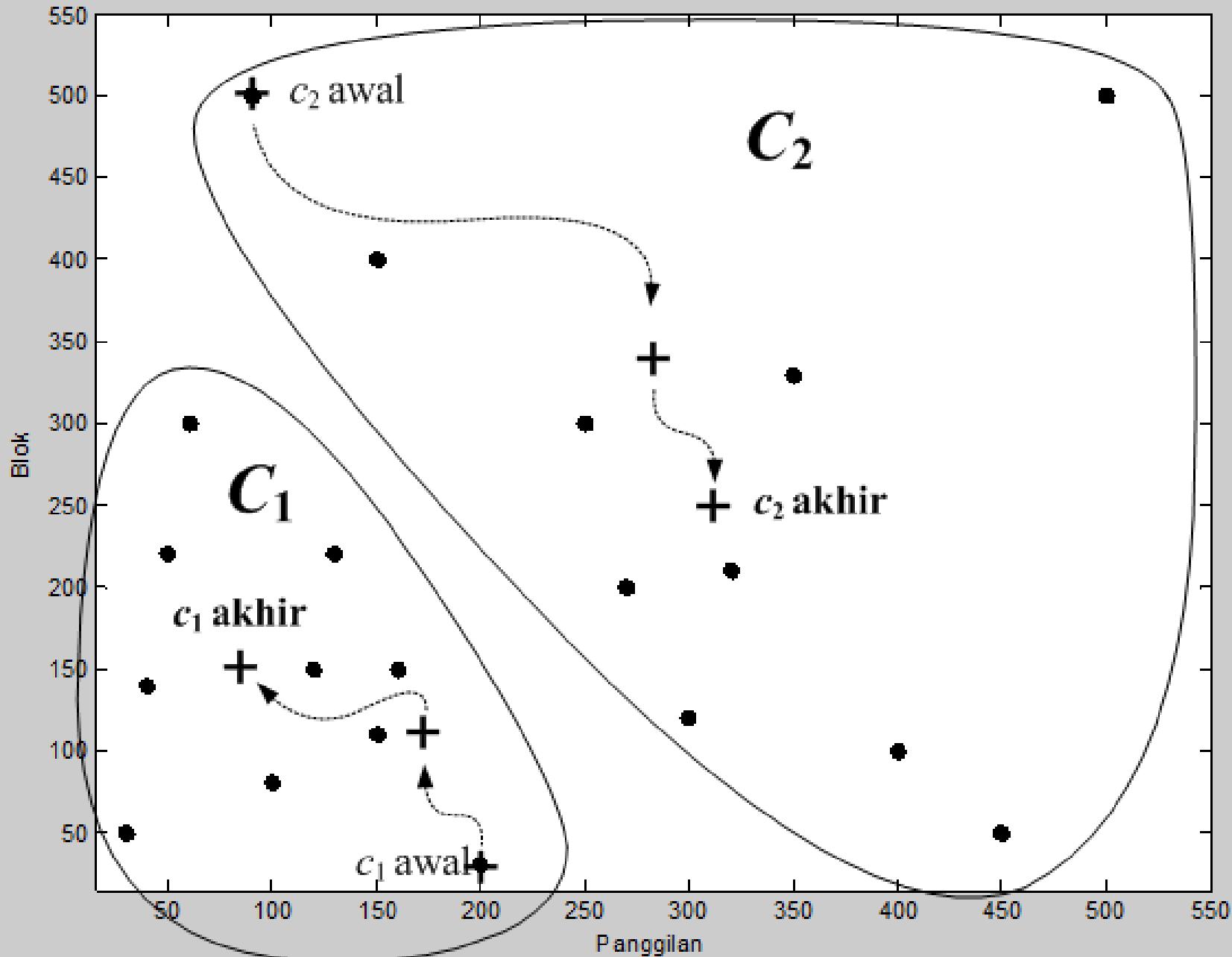
Data Pelanggan Telepon Seluler

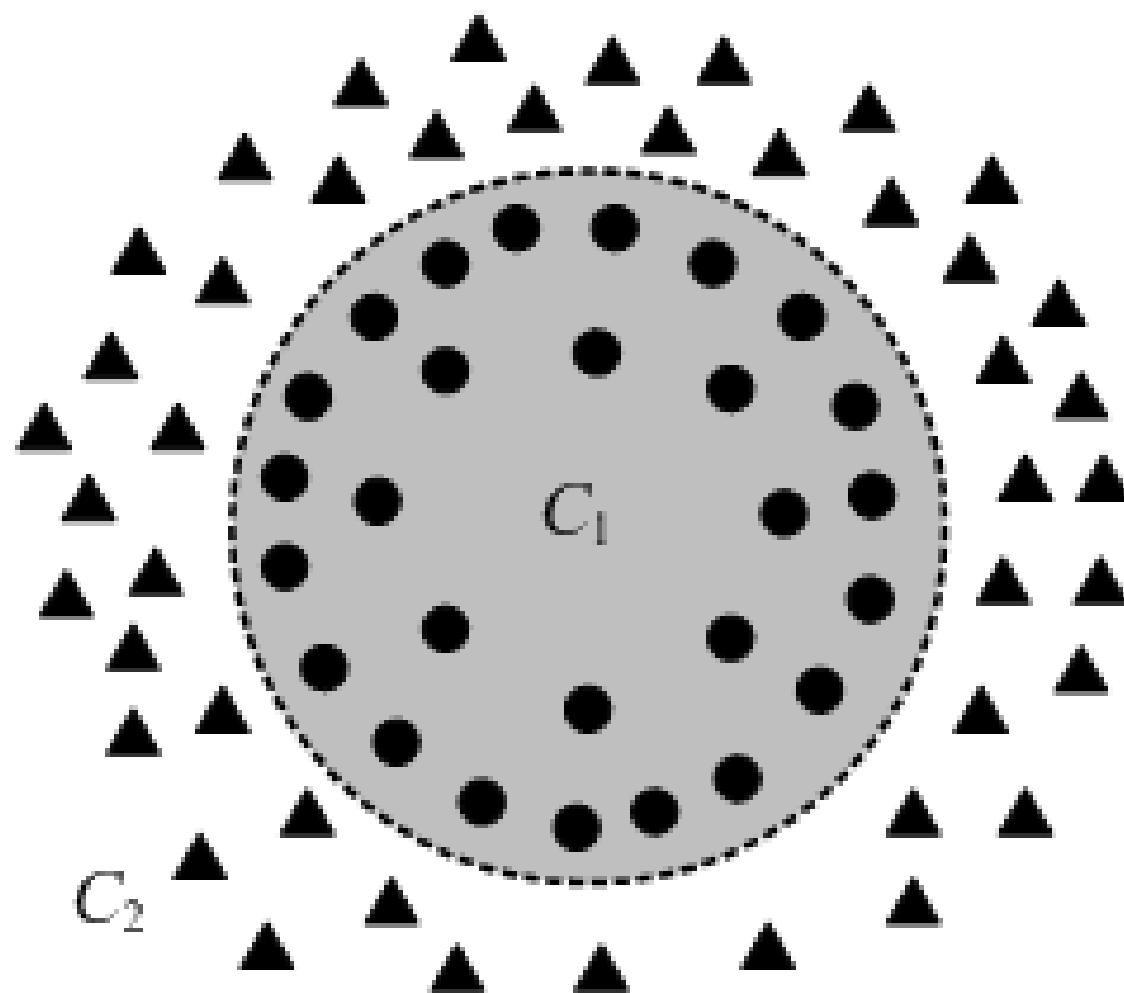
Panggilan	Blok
30	50
40	140
50	220
60	300
100	80
120	150
130	220
150	110
160	150
200	30

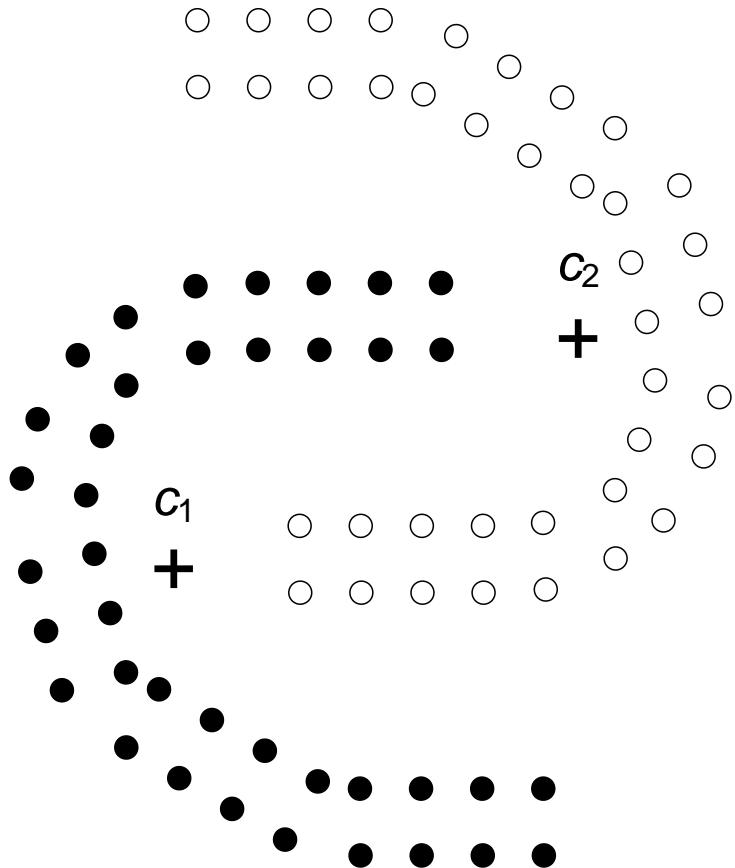
Panggilan	Blok
90	500
150	400
250	300
270	200
300	120
320	210
350	330
400	100
450	50
500	500

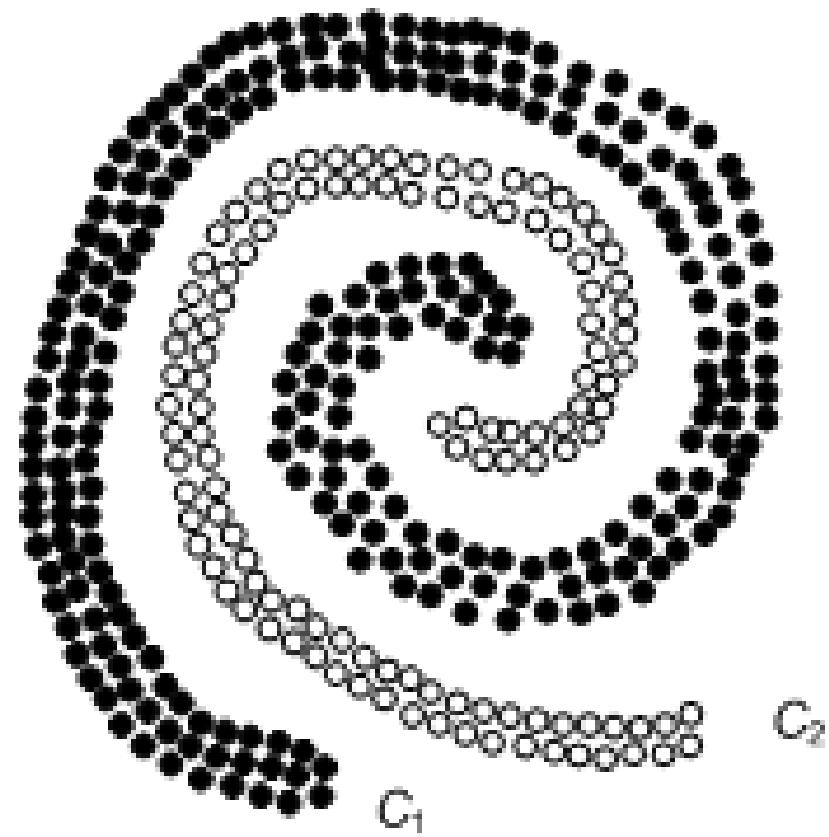












C_1

C_2

Fuzzy Clustering

- To extract rules from data
- Fuzzy c-means
- Aplikasi:
 - Cracked Tile Detection
 - Finding Cancer Cells
 - Image Segmentation
 - Document clustering

Cluster

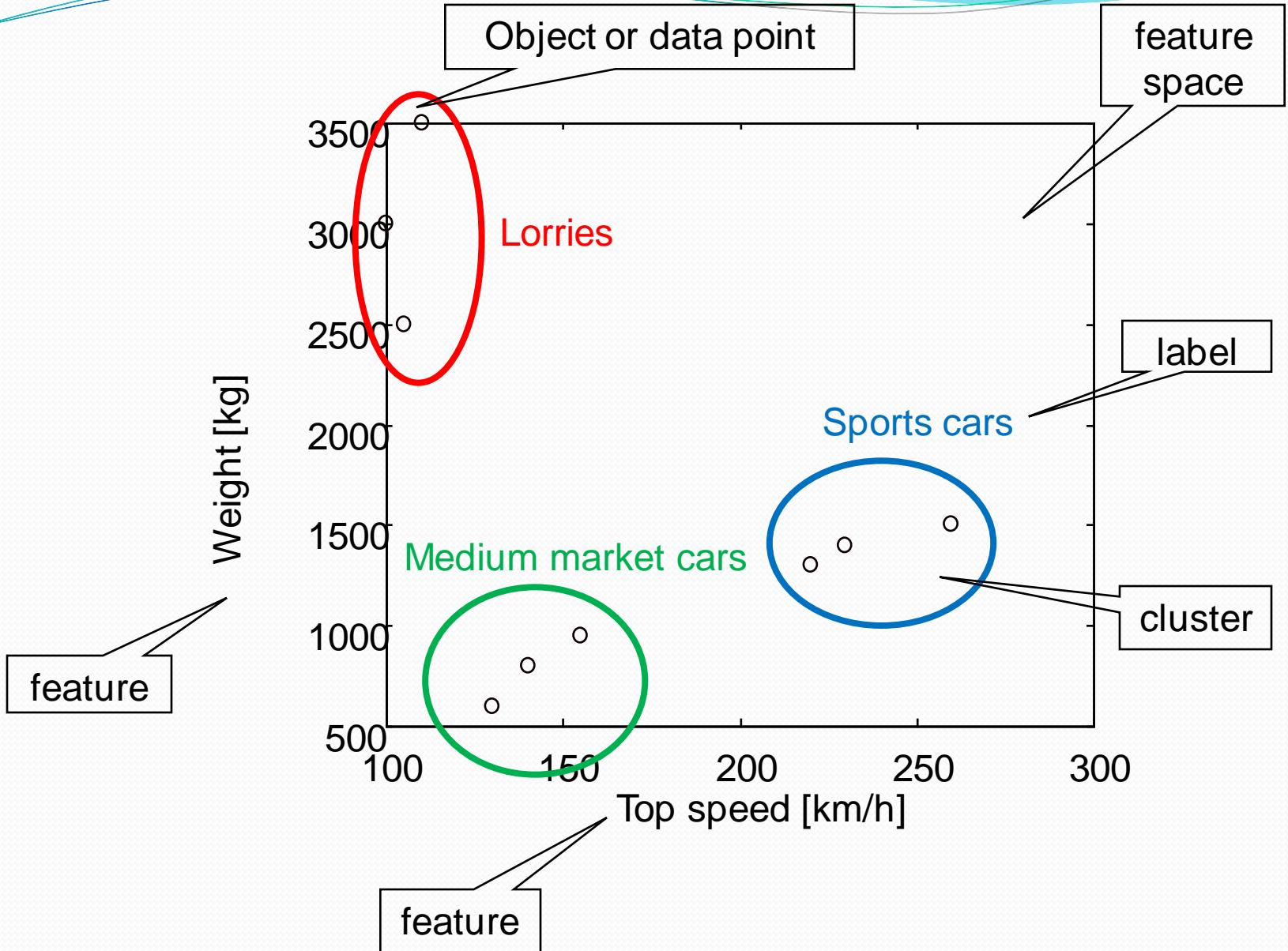
- Sejumlah individu yang memiliki sifat hampir sama atau terjadi bersama-sama
- Kelompok universitas kelas dunia
- Kelompok mobil kategori tertentu
- Kumpulan spam email
- Kumpulan tipe penyakit kanker

Cluster Analysis

- A statistical classification technique
- for discovering whether the individuals of a population fall into different groups
- by making **quantitative comparisons** of multiple characteristics.

Vehicle

Vehicle	Top speed km/h	Colour	Air resistance	Weight Kg
V1	220	red	0.30	1300
V2	230	black	0.32	1400
V3	260	red	0.29	1500
V4	140	gray	0.35	800
V5	155	blue	0.33	950
V6	130	white	0.40	600
V7	100	black	0.50	3000
V8	105	red	0.60	2500
V9	110	gray	0.55	3500



Kasus 6: *Tile Clustering*

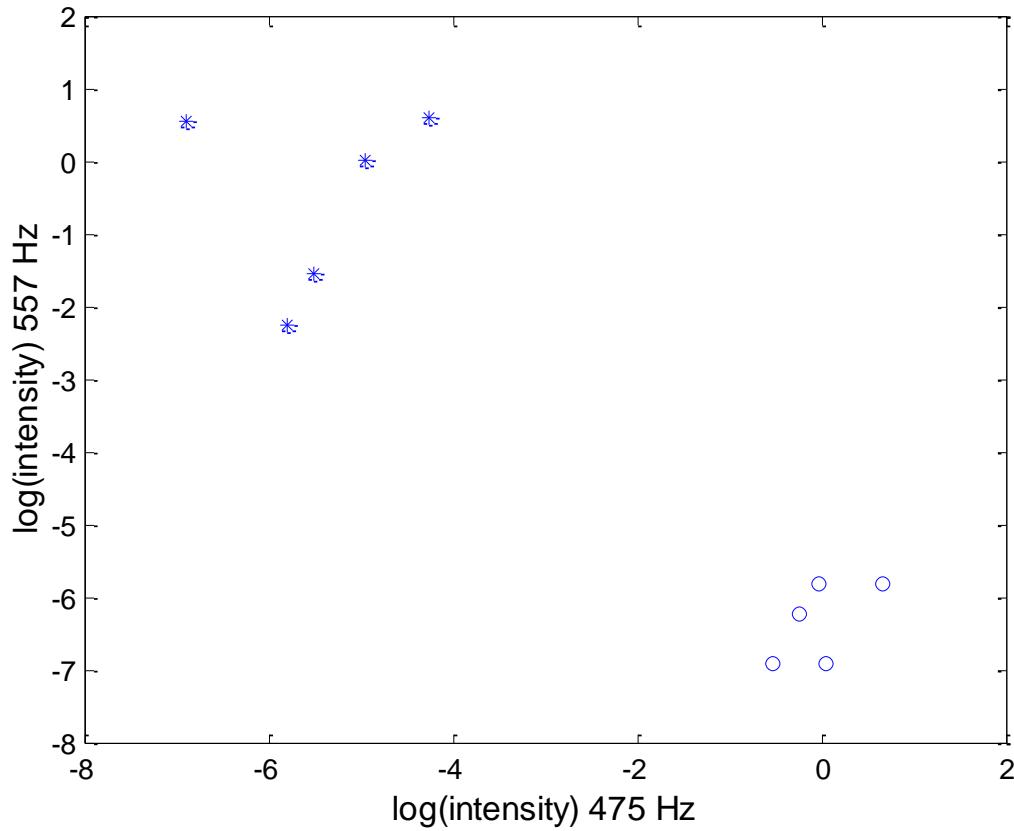
- Tiles are made from clay moulded into the right shape, brushed, glazed, and baked.
- Unfortunately, the baking may produce invisible cracks.
- Operators can detect the cracks by hitting the tiles with a hammer, and in an automated system the response is recorded with a microphone, filtered, Fourier transformed, and normalised.
- A small set of data is given in the TABLE below.

Kasus 6: *Tile Clustering*

475Hz	557Hz	Ok?
0.958	0.003	Yes
1.043	0.001	Yes
1.907	0.003	Yes
0.780	0.002	Yes
0.579	0.001	Yes
0.003	0.105	No
0.001	1.748	No
0.014	1.839	No
0.007	1.021	No
0.004	0.214	No

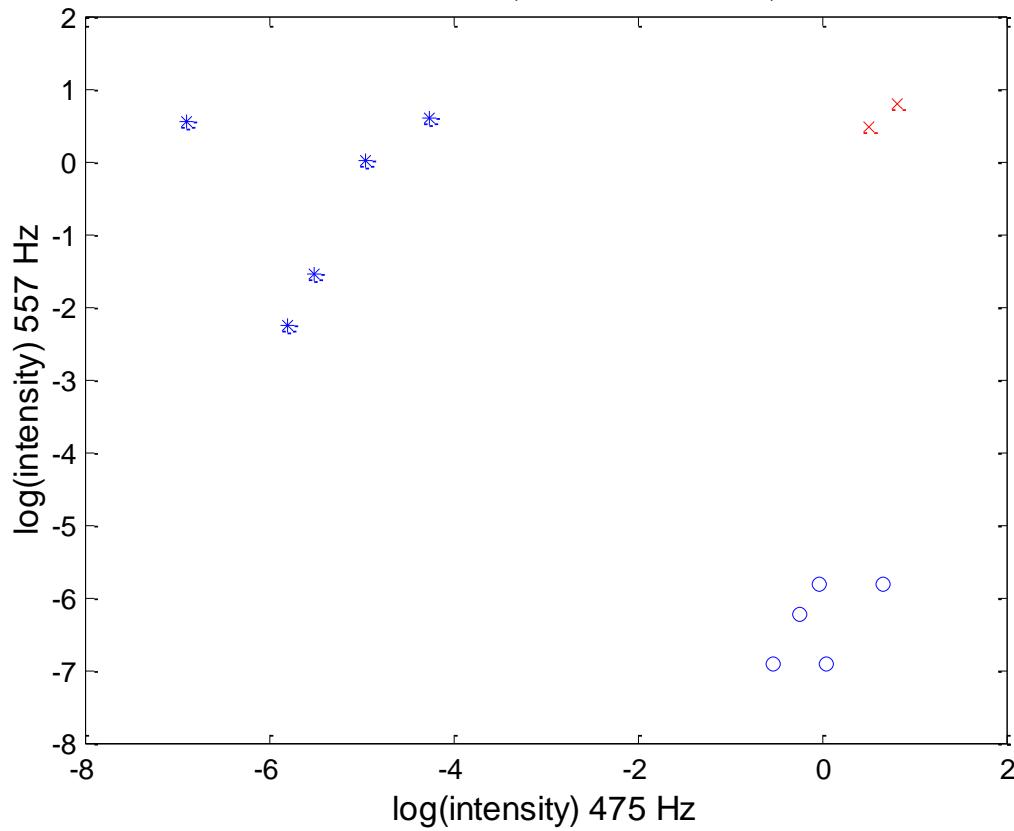
Hard c-means (HCM) (biasa disebut *k-means*)

Tiles data: o = whole tiles, * = cracked tiles, x = centres

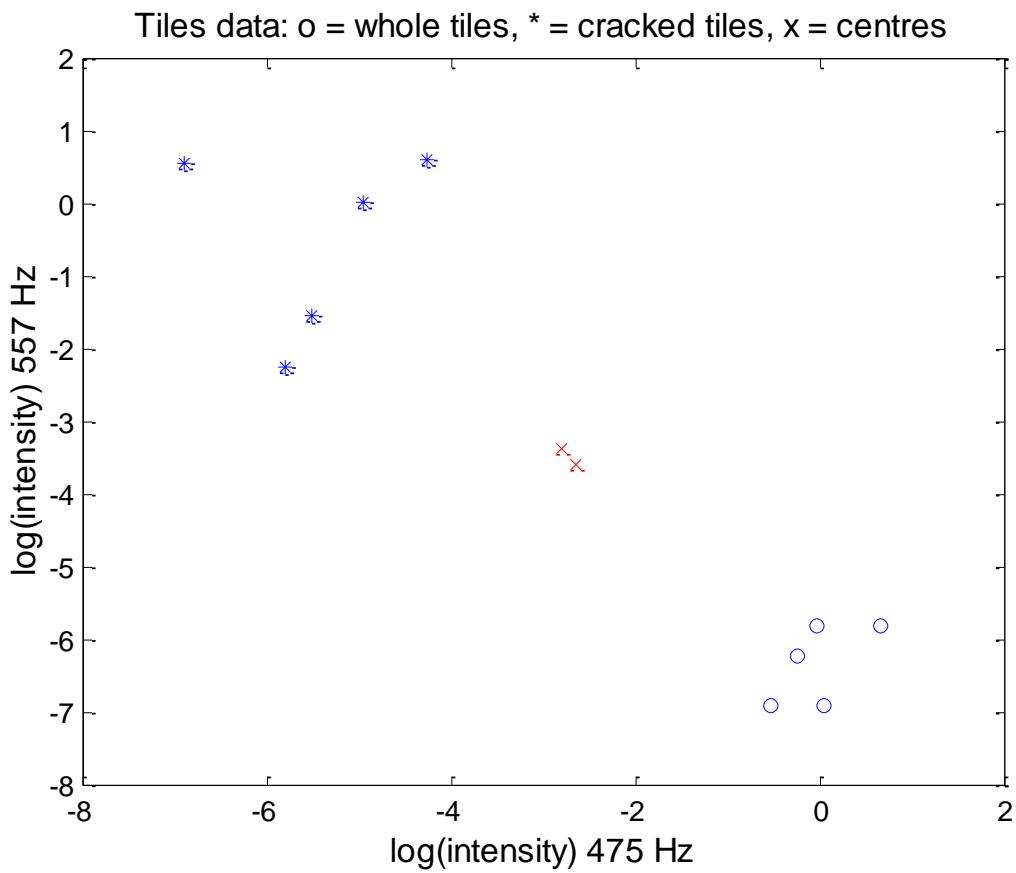


Plot of tiles by frequencies (logarithms). The whole tiles (o) seem well separated from the cracked tiles (*). The **objective** is to find the two clusters.

Tiles data: o = whole tiles, * = cracked tiles, x = centres

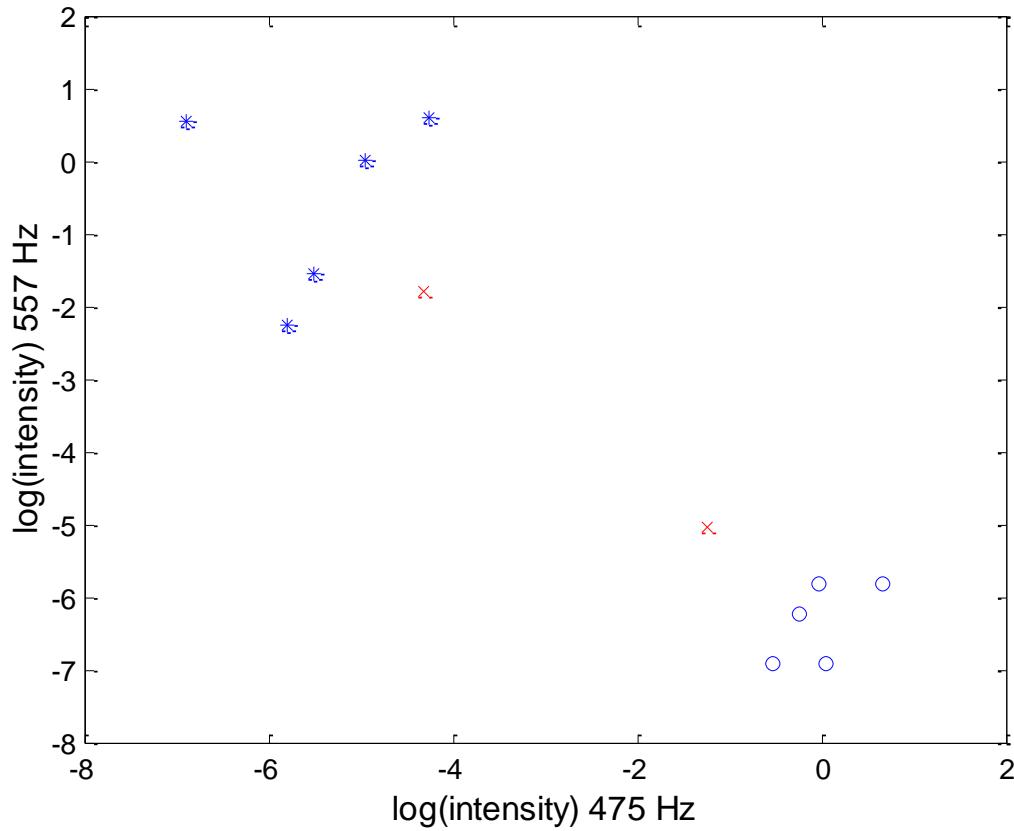


1. Place two cluster centres (x) at random.
2. Assign each data point (*) and (o) to the nearest cluster centre (x)



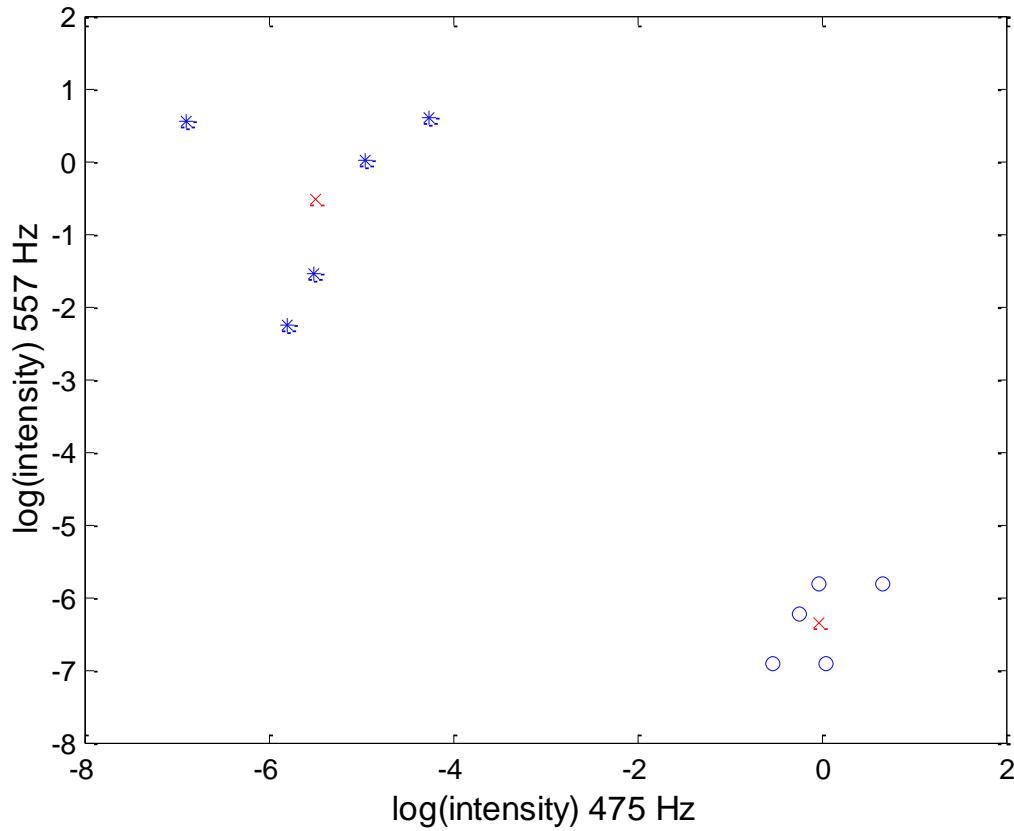
1. Compute the new centre of each class
2. Move the crosses (x)

Tiles data: o = whole tiles, * = cracked tiles, x = centres

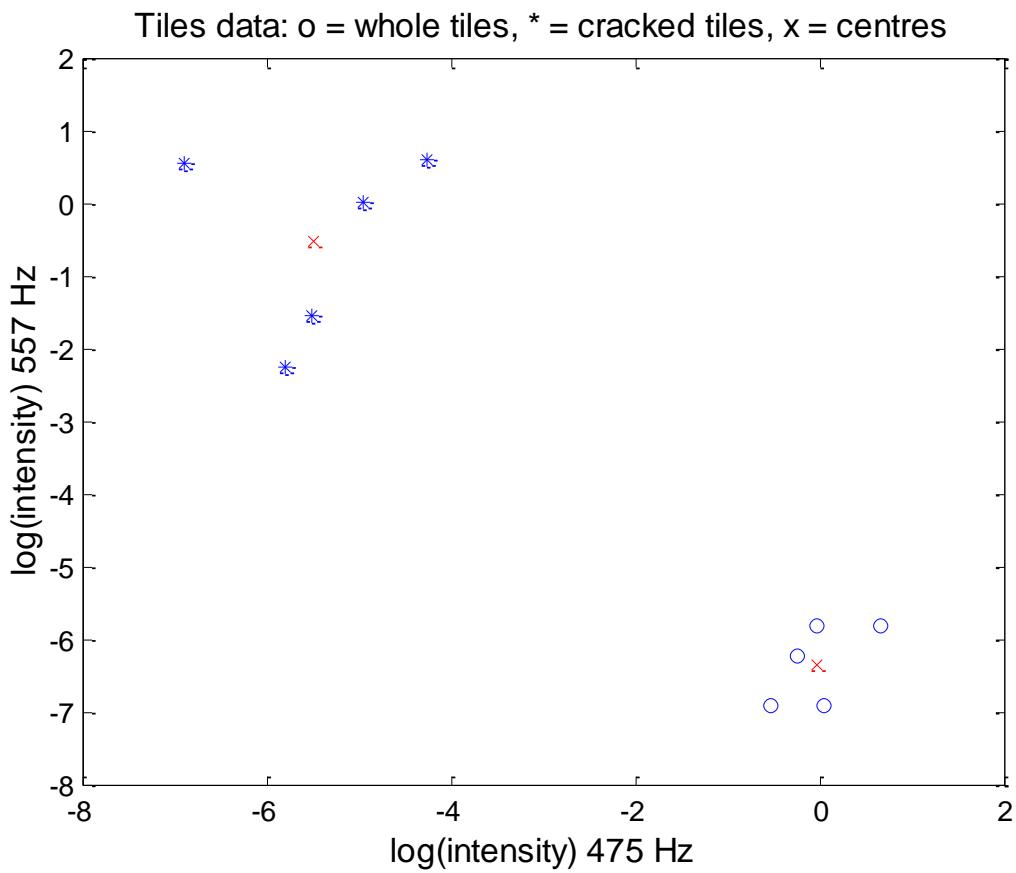


Iteration 2

Tiles data: o = whole tiles, * = cracked tiles, x = centres



Iteration 3



Iteration 4 (then stop, because no visible change)
Each data point belongs to the cluster defined by the nearest centre

M =

0.0000	1.0000
0.0000	1.0000
0.0000	1.0000
0.0000	1.0000
0.0000	1.0000
1.0000	0.0000
1.0000	0.0000
1.0000	0.0000
1.0000	0.0000
1.0000	0.0000

The membership matrix M:

1. The last five data points (rows) belong to the first cluster (column)
2. The first five data points (rows) belong to the second cluster (column)

Membership matrix \mathbf{M}

$$m_{ik} = \begin{cases} 1 & \text{if } \|\mathbf{u}_k - \mathbf{c}_i\|^2 \leq \|\mathbf{u}_k - \mathbf{c}_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$

Diagram illustrating the components of the membership matrix formula:

- data point k** : Points to the variable \mathbf{u}_k .
- cluster centre i** : Points to the term \mathbf{c}_i .
- cluster centre j** : Points to the term \mathbf{c}_j .
- distance**: Points to the squared distance term $\|\mathbf{u}_k - \mathbf{c}_i\|^2$.

c-partition

All clusters C together fills the whole universe U

$$\bigcup_{i=1}^c C_i = U$$

Clusters do not overlap

$$C_i \cap C_j = \emptyset \quad \text{for all } i \neq j$$

$$\emptyset \subset C_i \subset U \quad \text{for all } i$$

A cluster C is never empty and it is smaller than the whole universe U

$$2 \leq c \leq K$$

There must be at least 2 clusters in a c-partition and at most as many as the number of data points K

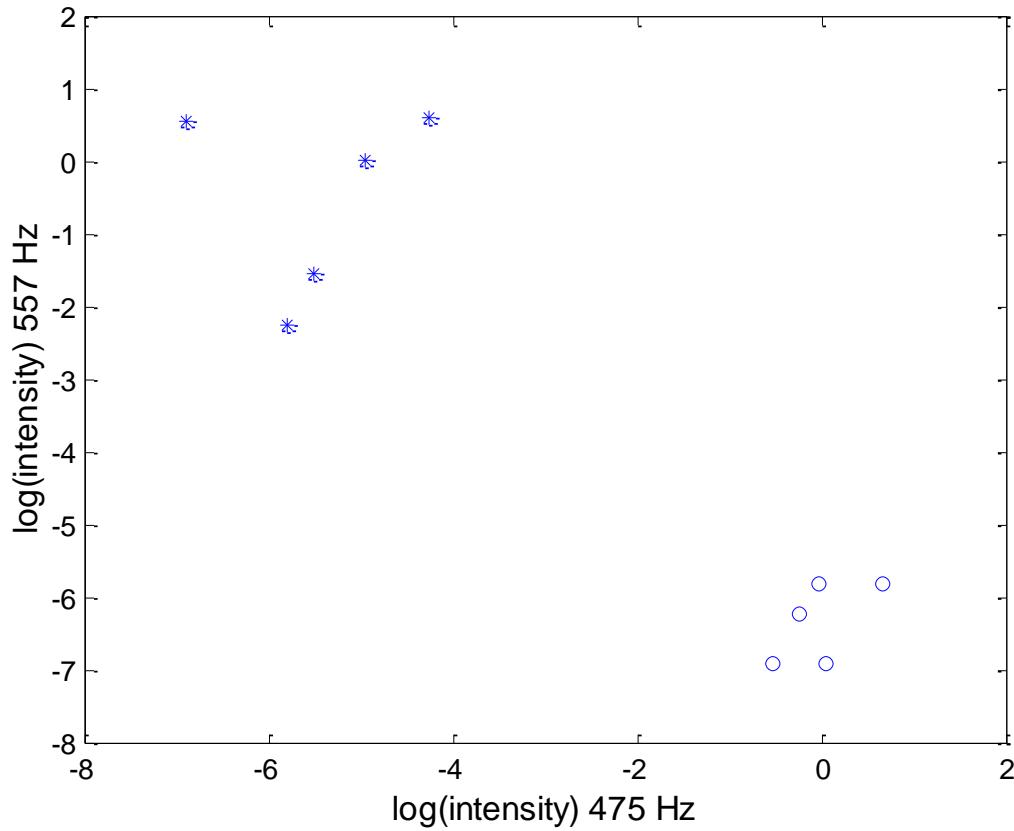
Objective function

Minimise the total sum
of all distances

$$J = \sum_{i=1}^c J_i = \sum_{i=1}^c \left(\sum_{k, \mathbf{u}_k \in C_i} \|\mathbf{u}_k - \mathbf{c}_i\|^2 \right)$$

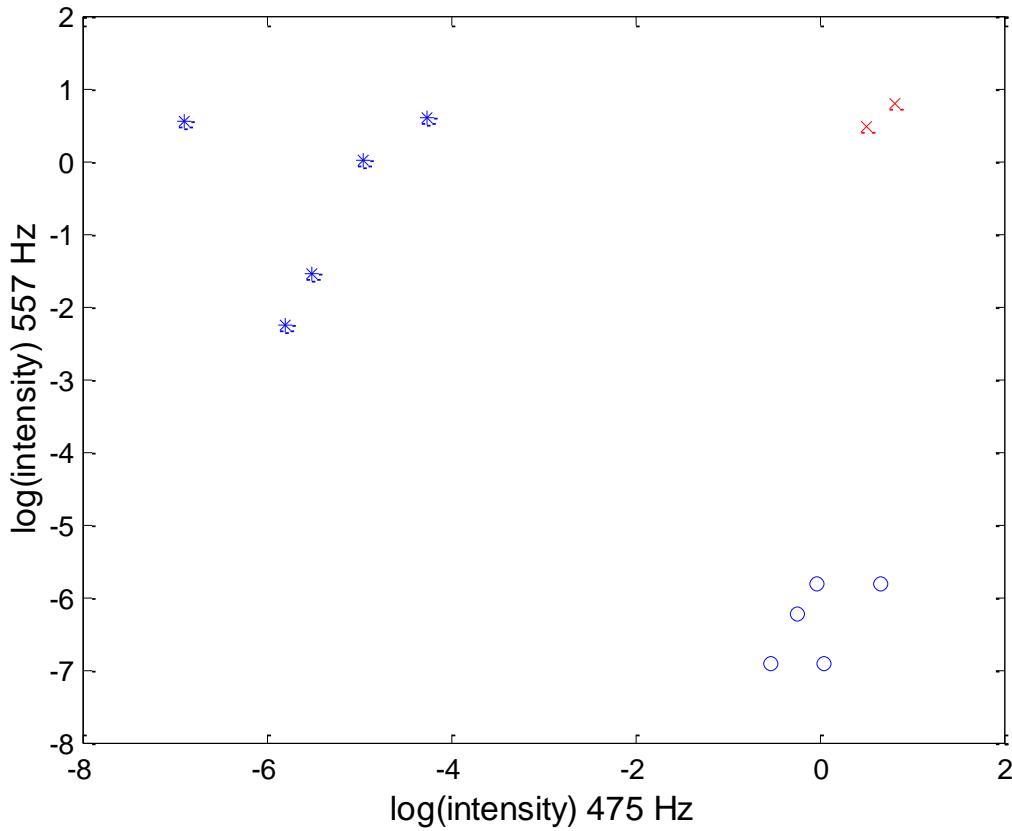
Fuzzy c-means (FCM)

Tiles data: o = whole tiles, * = cracked tiles, x = centres



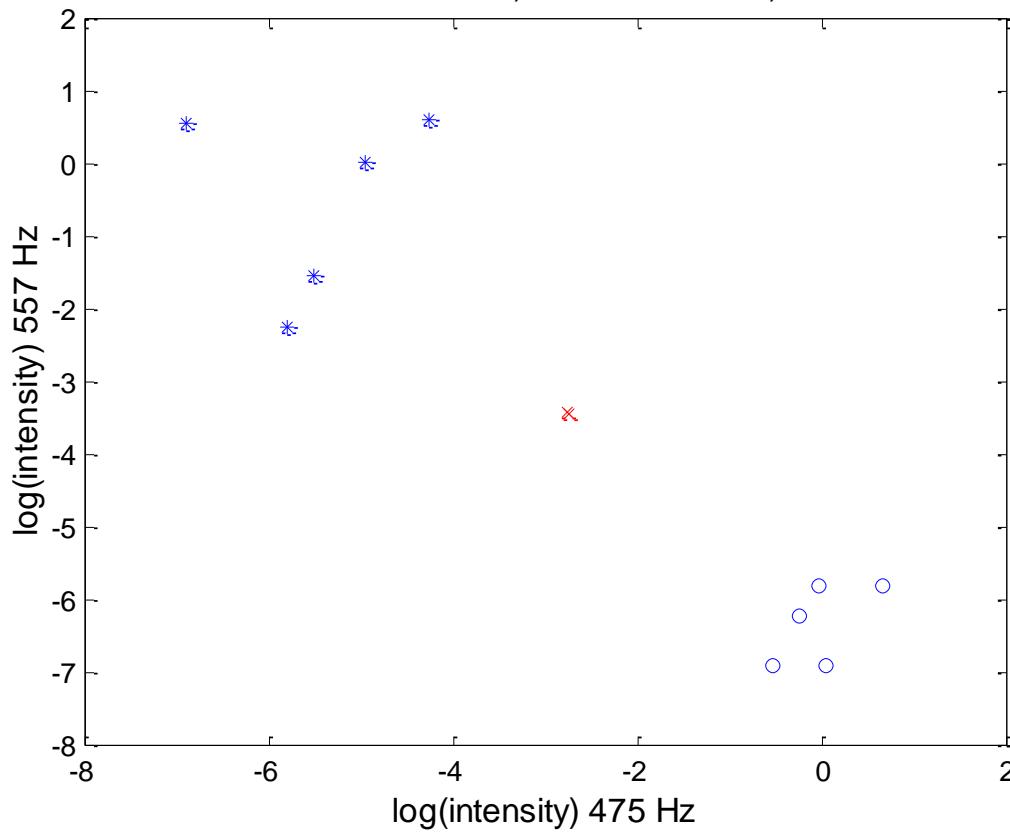
Each data point belongs to two clusters to different degrees

Tiles data: o = whole tiles, * = cracked tiles, x = centres



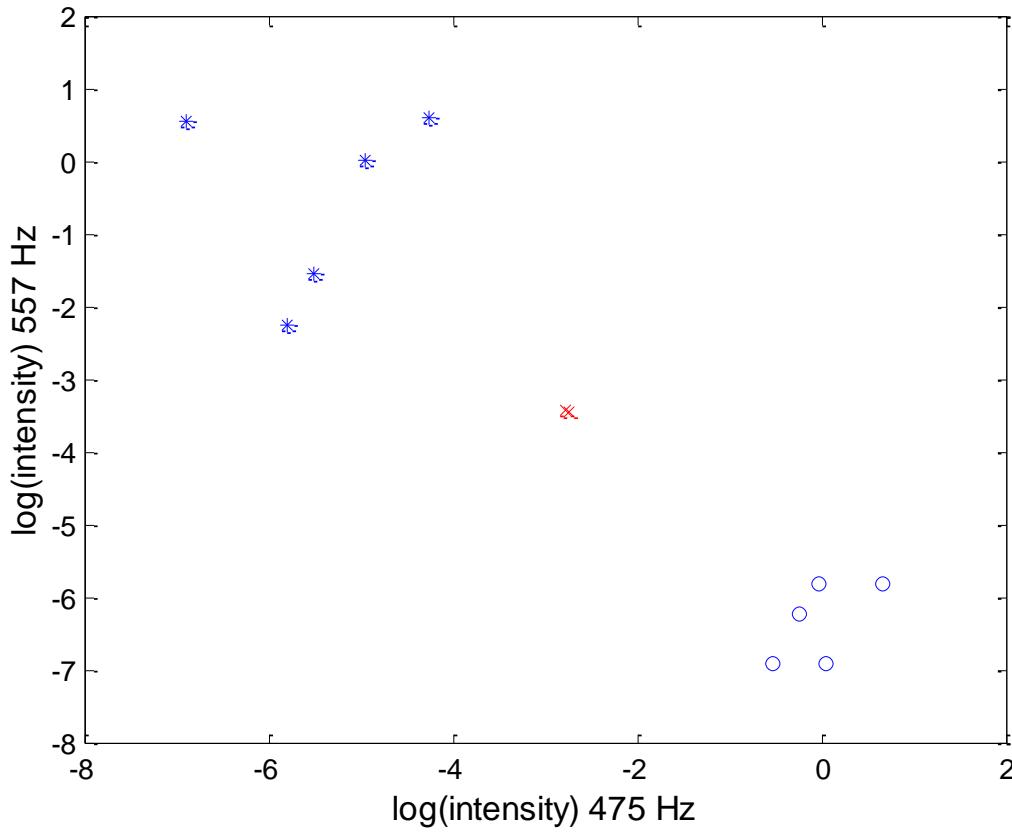
1. Place two cluster centres
2. Assign a fuzzy membership to each data point depending on distance

Tiles data: o = whole tiles, * = cracked tiles, x = centres



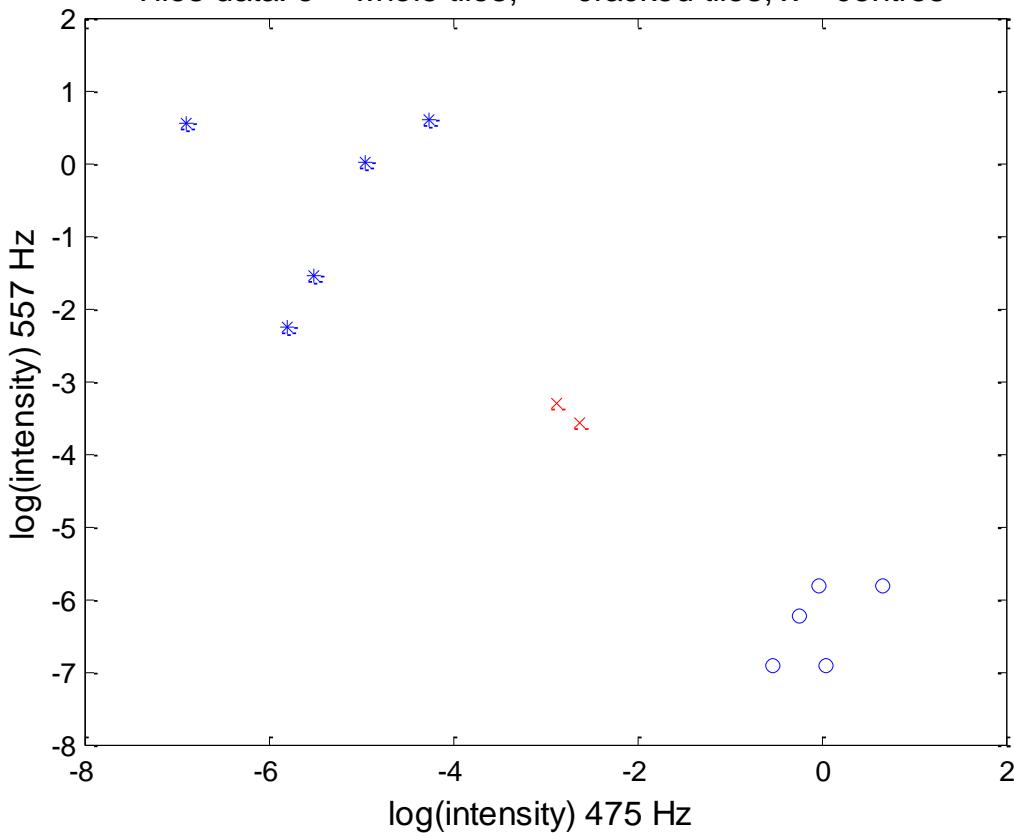
1. Compute the new centre of each class
2. Move the crosses (x)

Tiles data: o = whole tiles, * = cracked tiles, x = centres



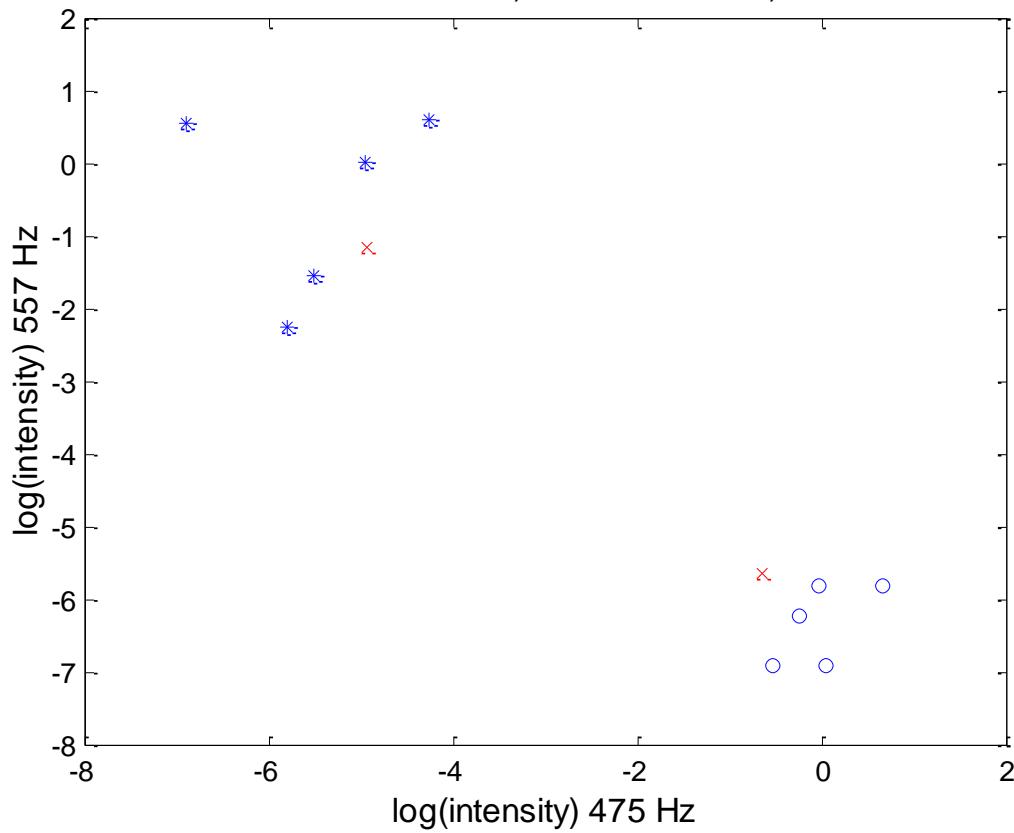
Iteration 2

Tiles data: o = whole tiles, * = cracked tiles, x = centres



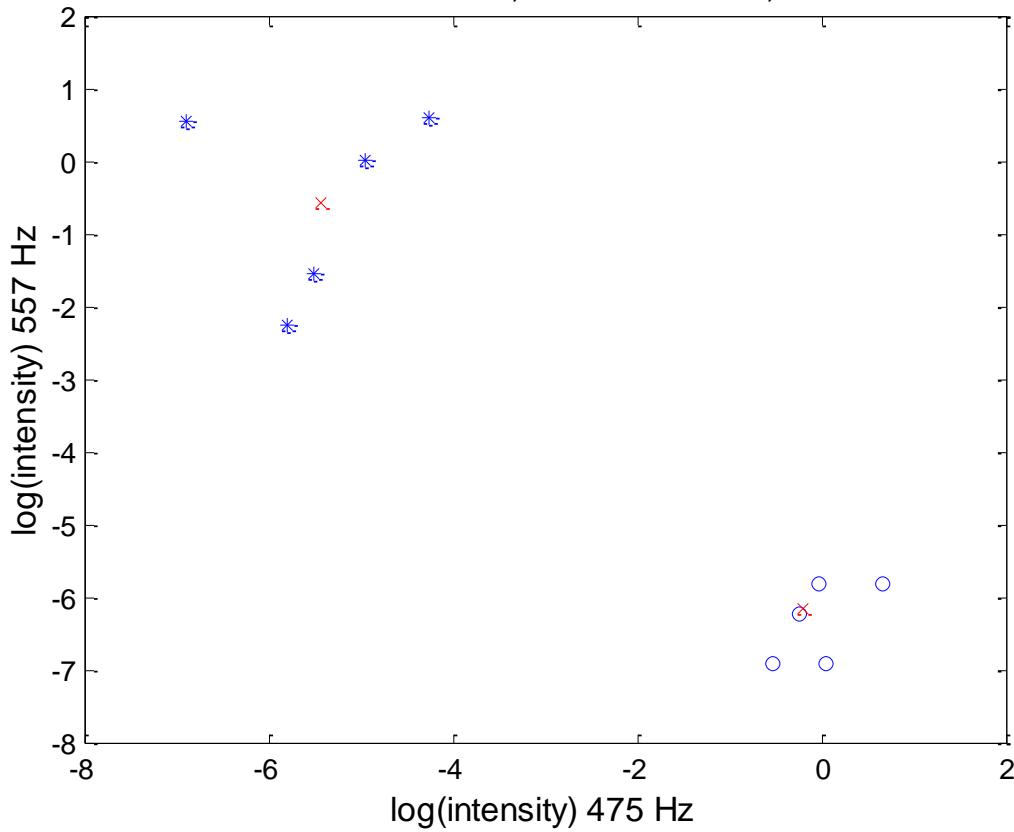
Iteration 5

Tiles data: o = whole tiles, * = cracked tiles, x = centres



Iteration 10

Tiles data: o = whole tiles, * = cracked tiles, x = centres



Iteration 13 (then stop, because no visible change)
Each data point belongs to the two clusters to a degree

M =

0.0025	0.9975
0.0091	0.9909
0.0129	0.9871
0.0001	0.9999
0.0107	0.9893
0.9393	0.0607
0.9638	0.0362
0.9574	0.0426
0.9906	0.0094
0.9807	0.0193

The membership matrix M:

1. The last five data points (rows) belong mostly to the first cluster (column)
2. The first five data points (rows) belong mostly to the second cluster (column)

Fuzzy membership matrix \mathbf{M}

Point k 's
membership
of cluster i

$$m_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(q-1)}}$$

Fuzziness
exponent

Distance from point
 k to current cluster
centre i

$$d_{ik} = \|\mathbf{u}_k - \mathbf{c}_i\|$$

Distance from point
 k to other cluster
centres j

Fuzzy membership matrix \mathbf{M}

$$\begin{aligned}m_{ik} &= \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(q-1)}} \\&= \frac{1}{\left(\frac{d_{ik}}{d_{1k}} \right)^{2/(q-1)} + \left(\frac{d_{ik}}{d_{2k}} \right)^{2/(q-1)} + \cdots + \left(\frac{d_{ik}}{d_{ck}} \right)^{2/(q-1)}} \\&= \frac{\frac{1}{d_{ik}^{2/(q-1)}}}{\frac{1}{d_{1k}^{2/(q-1)}} + \frac{1}{d_{2k}^{2/(q-1)}} + \cdots + \frac{1}{d_{ck}^{2/(q-1)}}}\end{aligned}$$

Gravitation to
cluster i relative to
total gravitation

Fuzzy c-partition

All clusters C together fill the whole universe U .
Remark: The sum of memberships for a data point is 1, and the total for all points is K

$$\bigcup_{i=1}^c C_i = U$$

$$C_i \cap C_j = \emptyset \quad \text{for all } i \neq j$$

$$\emptyset \subset C_i \subset U \quad \text{for all } i$$

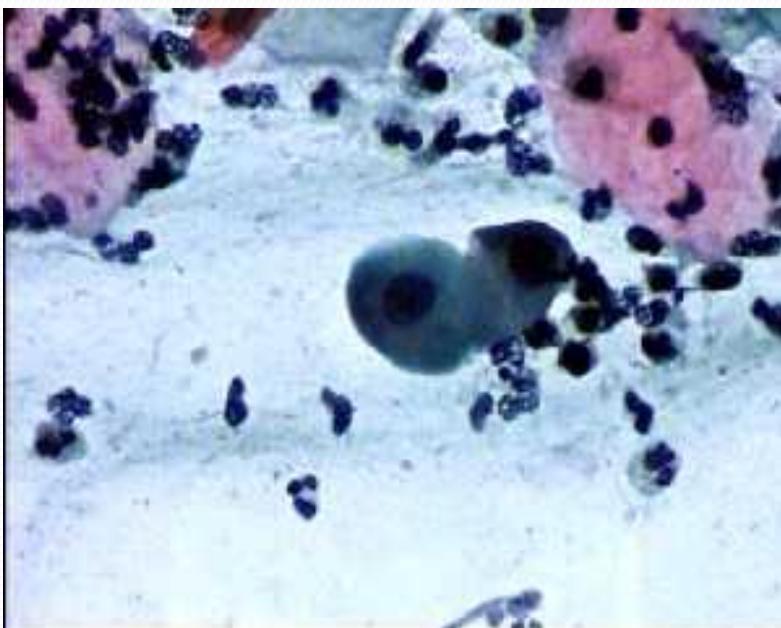
A cluster C is never empty and it is smaller than the whole universe U

$$2 \leq c \leq K$$

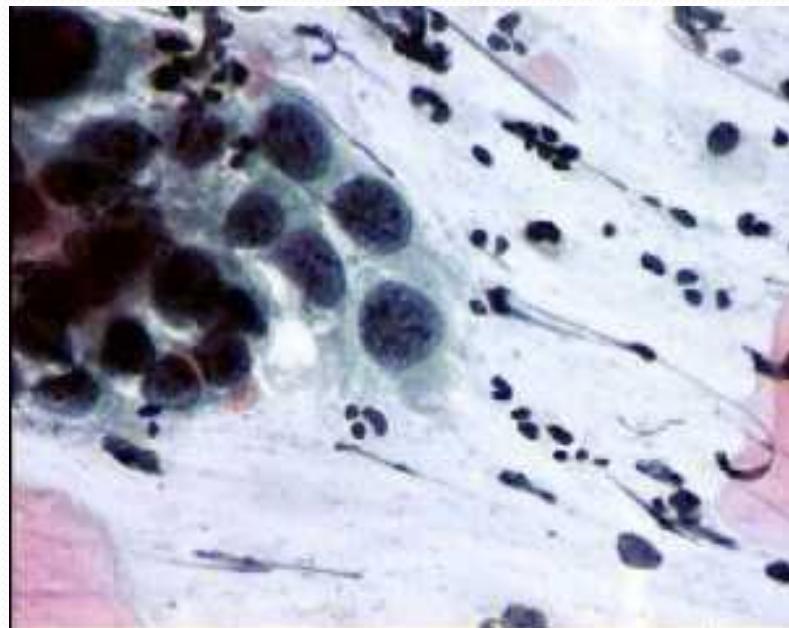
*Not valid:
Clusters do overlap*

There must be at least 2 clusters in a c-partition and at most as many as the number of data points K

Kasus 7: Klasifikasi Sel Kanker



Normal smear

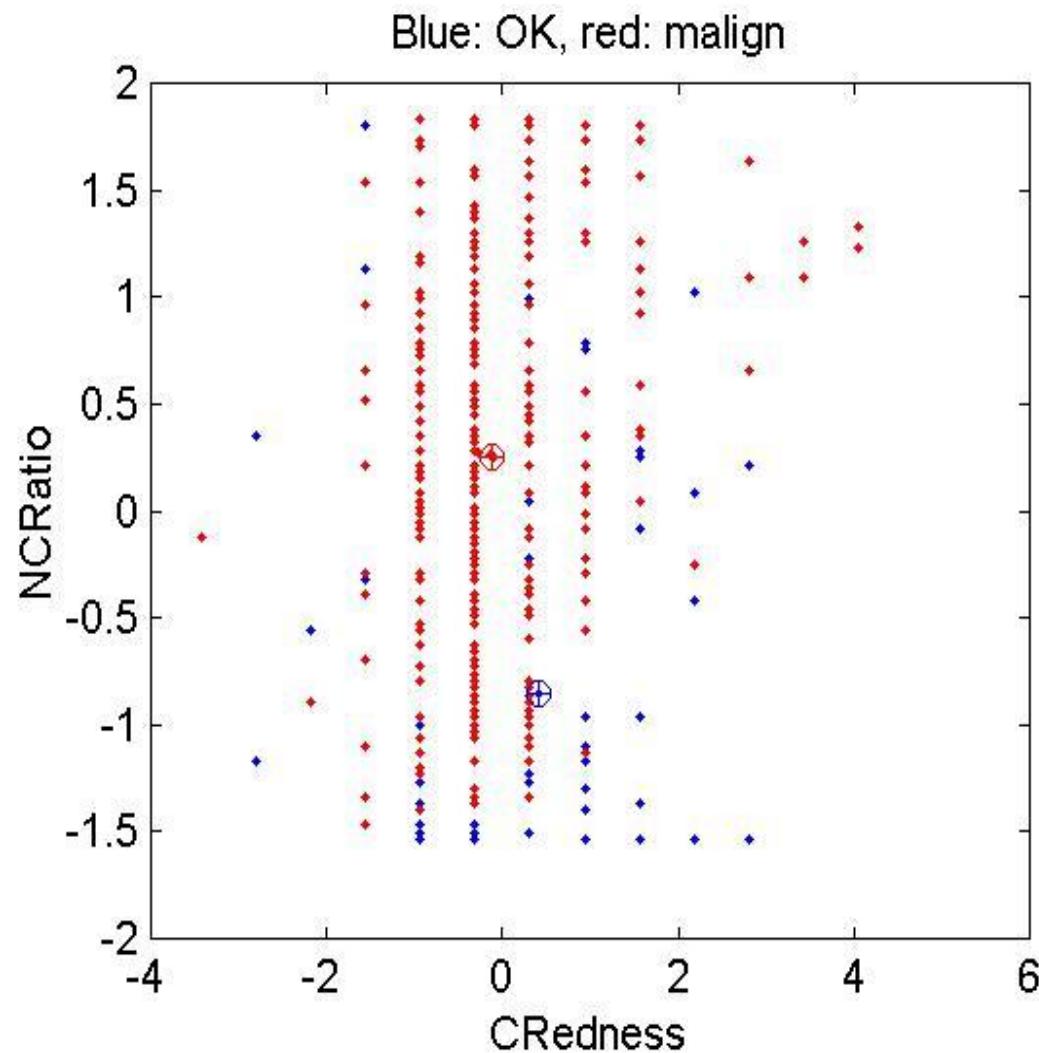


Severely dysplastic smear

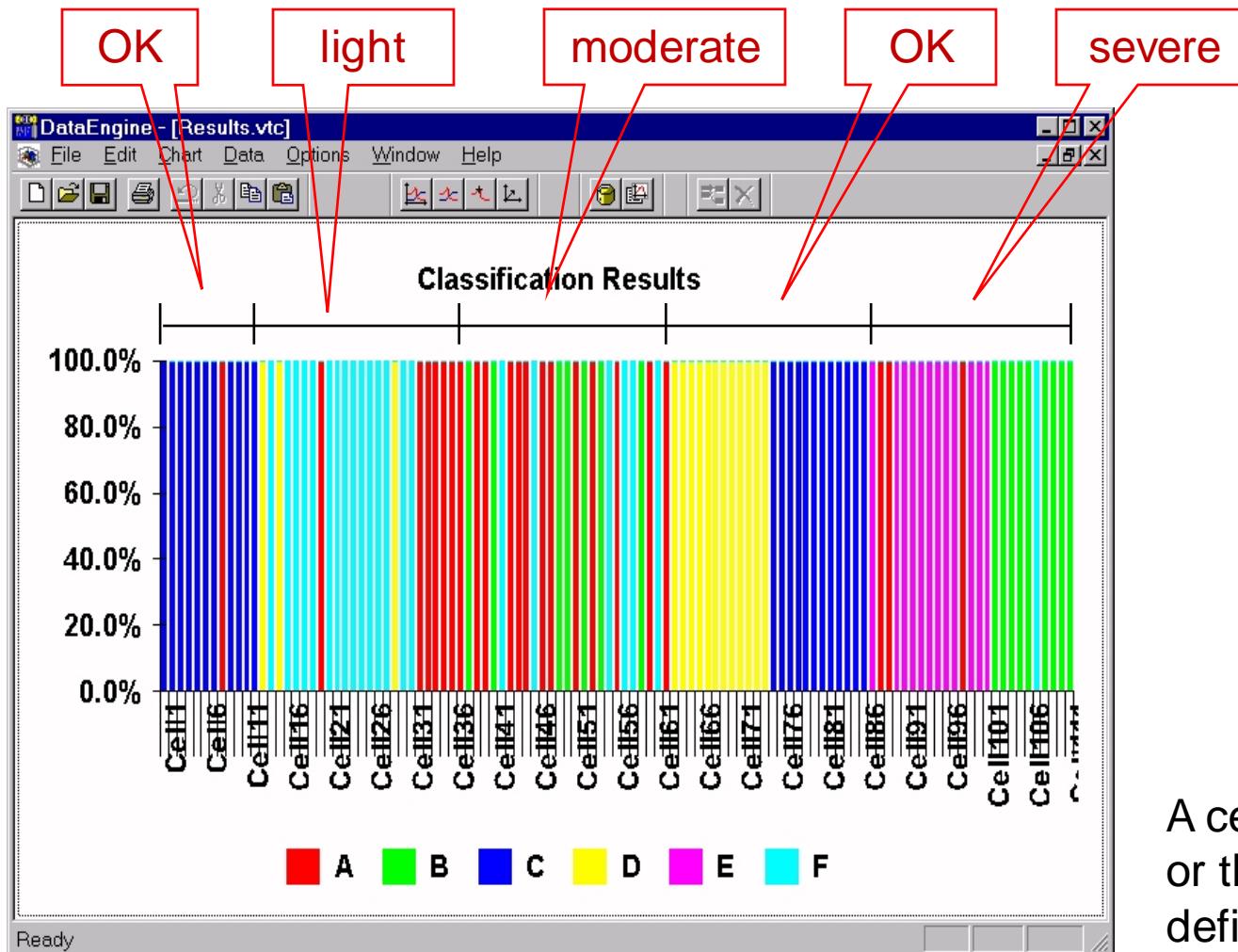
Possible Features

- Nucleus and cytoplasm area
- Nucleus and cyto brightness
- Nucleus shortest and longest diameter
- Cyto shortest and longest diameter
- Nucleus and cyto perimeter
- Nucleus and cyto no of maxima
- ...

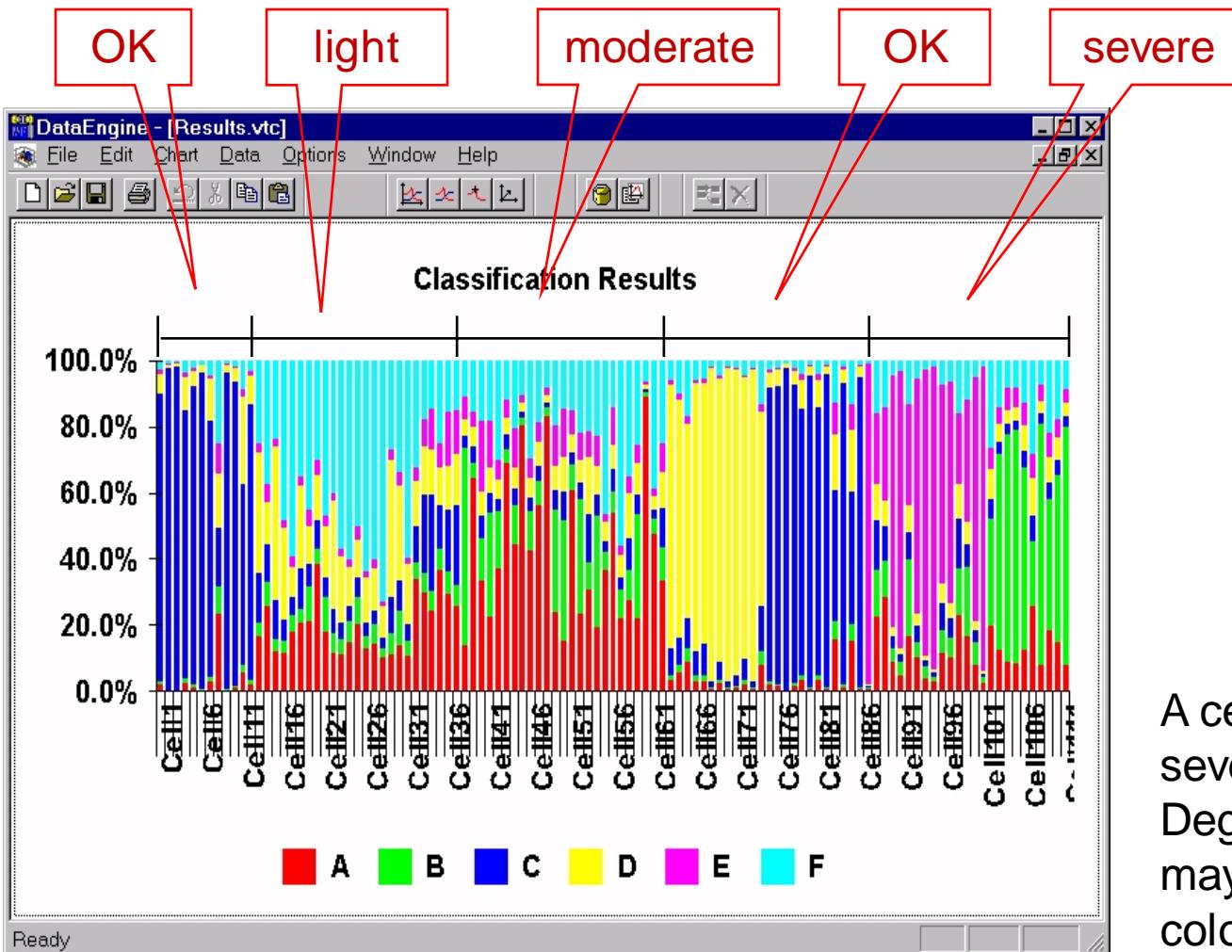
Classes are nonseparable



Hard Classifier (HCM)



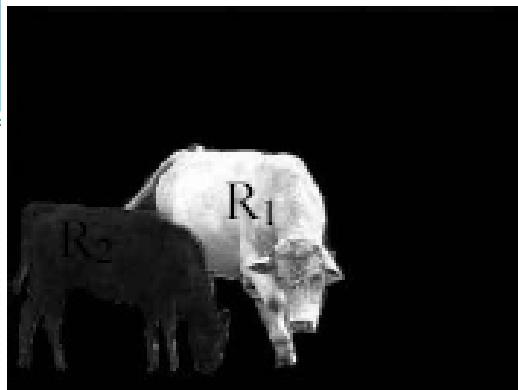
Fuzzy Classifier (FCM)



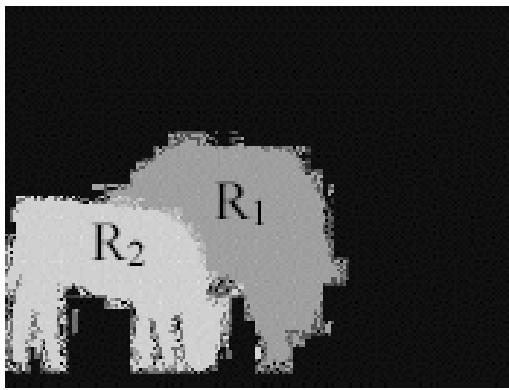
A cell can belong to several classes to a Degree, i.e., one column may have several colours.

Kasus 8: *Image Segmentation*

- Sangat penting untuk *image analysis, understanding, dan coding*
- Bagaimana memisahkan satu objek yang diinginkan dari objek2 lainnya?
- Pada image terdapat banyak ambiguitas



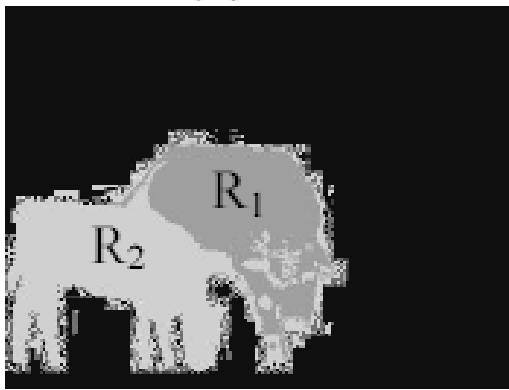
(a)



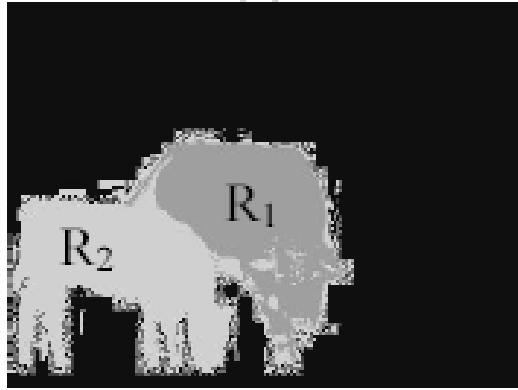
(b)



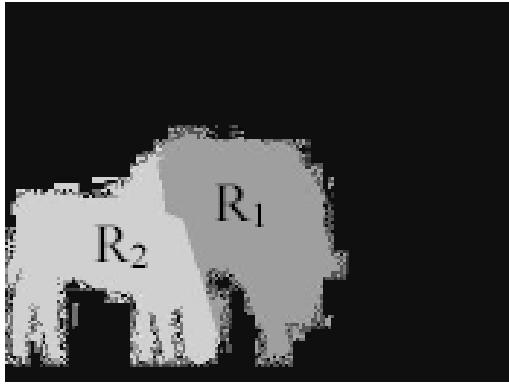
(c)



(d)



(e)



(f)

- (a) Original cow image,
- (b) Manually segmented reference image for (a).
- (c) FCM with pixel locations
- (d) FCM with pixel intensity
- (e) FCM with both features
- (f) Fuzzy Clustering Incorporating Spatial Information (FCSI).

[M. Ameer Ali, Gour C Karmakar and Laurence S Dooley, "Image Segmentation Using Fuzzy Clustering Incorporating Spatial Information, Monash University, Australia]

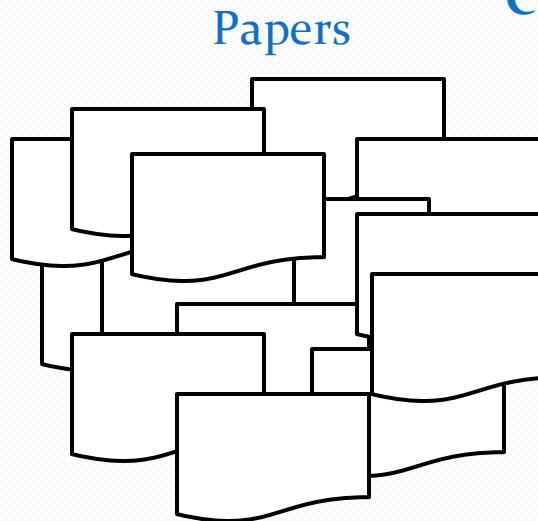
Kasus 9: *Documents Clustering*

- Text represented as an unordered collection of words
- Using tf-idf (term frequency-inverse document frequency)
- Document = one vector in high dimensional space
- Similarity = cosine similarity between vectors

PROBLEM

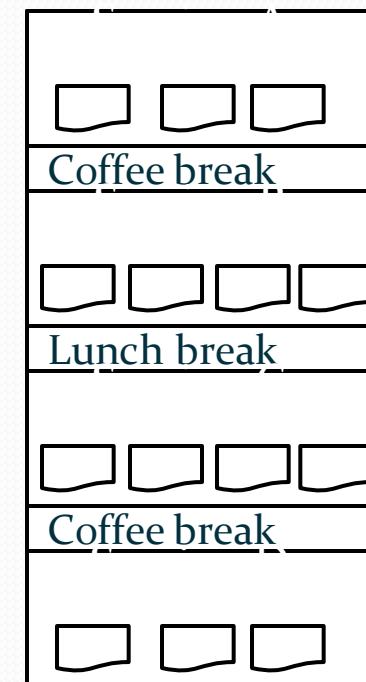
Grouping conference papers with regard to their contents into predefined sessions schedule.

EXAMPLE



Constraint-based clustering

Sessions schedule



Combining CBC & Fuzzy Clustering

- **PHASE 1 SOLUTION**

- constrained-based clustering (CBC)

- **DIFFICULTIES**

- CBC can get stuck in local minimum
 - often low quality result (created schedule)
 - user interaction needed to repair schedule

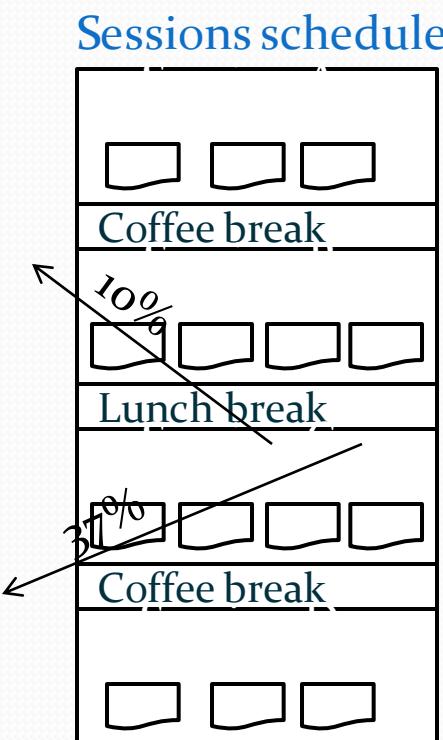
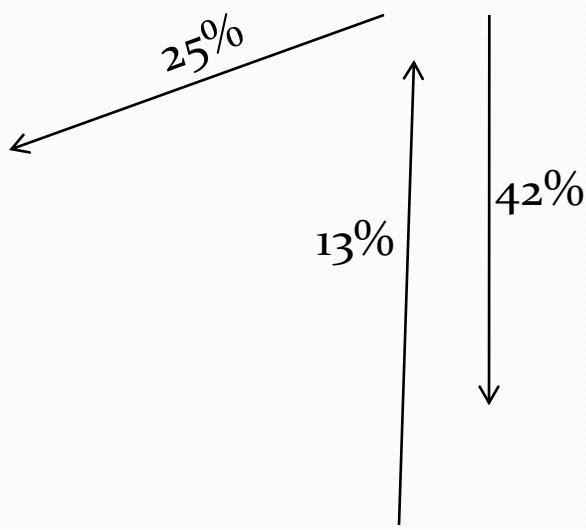
- **PHASE 2 NEEDED**

- run fuzzy clustering (FC) with initial clusters from CBC
 - if output clusters of FC differ from CBC repeat everything
 - if the clusters of FC equal to CBC show new info to user

RUN FUZZY CLUSTERING ON PHASE 1 RESULTS

- insight into result quality
- identify problematic papers

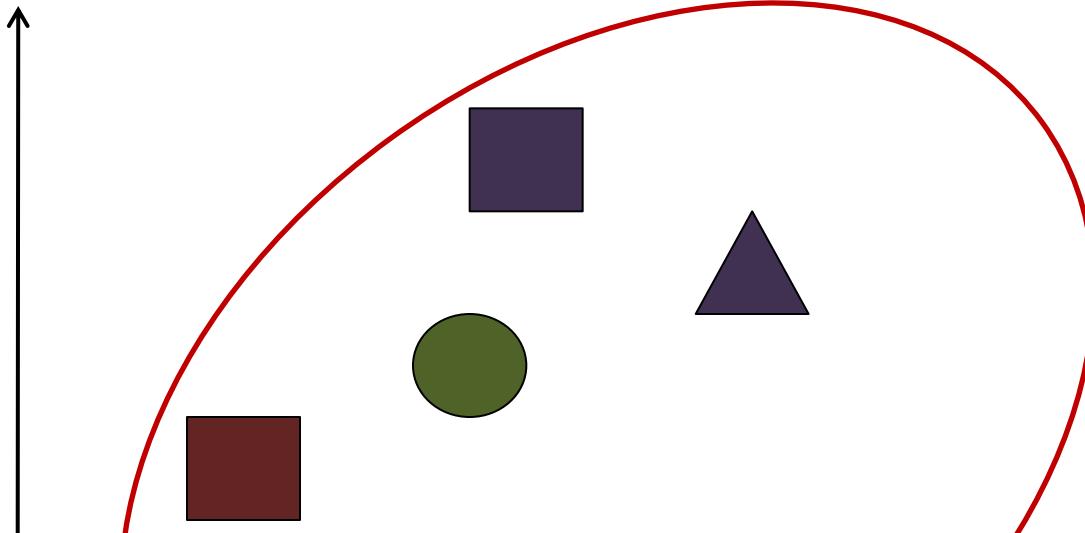
EXAMPLE



Masalah Tugas Akhir IF

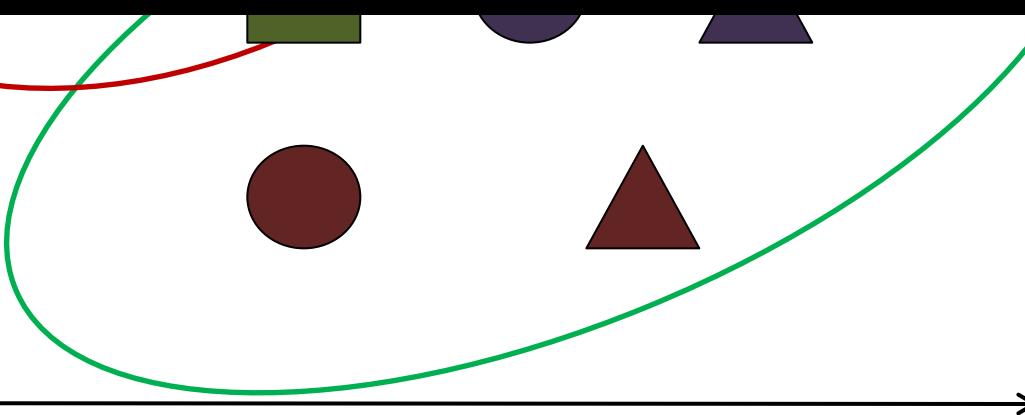
- Clustering dokumen TA berdasarkan KBK
- Clustering Dosen berdasarkan Keahliannya
- Penentuan Dosen Penguji Sidang TA

Hard Classifier



Ketidakpastian didekati dengan Kepastian

→ Ketidakpastian



Fuzzy Classifier



Kesulitan *Fuzzy Systems*

- Model Mamdani atau Sugeno atau model lain?
- Jumlah Nilai Linguistik untuk setiap variabel?
- Fungsi Keanggotaan: Segitiga, trapesium, phi, ...?
- Batas-batas Nilai Linguistik?
- *Fuzzy rule* yang tepat?

Solusi

- Semua komponen *fuzzy* bisa didefinisikan secara otomatis menggunakan EAs atau ANN.
- *Evolving Fuzzy Systems*
- *Neuro-Fuzzy*
- *Fuzzy Systems* yang bisa **LEARNING**
- Perlu data latih yang **MEMADAI**

Pemberian Beasiswa

NIM	IPK	Gaji Ortu (juta rupiah)	Nilai Kelayakan
070001	2,0	16	45
070002	2,5	12	50
070003	3,6	30	55
070004	1,5	140	10
070005	2,7	10	40
070006	3,9	0,5	95
070007	1,9	1	52
070008	2,8	8	68
070009	3,5	6	72
070010	2,0	7	53

Sprinkler Control System

Tanggal	Waktu	Suhu (°C)	Kelembaban Tanah (%)	Durasi Penyiraman (menit)
01-02-2006	08:00	20	16	55,7
01-02-2006	13:00	25	12	59,3
01-02-2006	18:00	16	30	5,6
02-02-2006	07:00	15	14	30,1
02-02-2006	12:00	27	10	43,4
02-02-2006	19:00	12	19	18,6
03-02-2006	06:30	19	12	22,1
03-02-2006	10:00	28	8	47,3
03-02-2006	13:30	35	6	76,4
03-02-2006	16:00	20	17	20,9

Evolving Fuzzy Systems

- Representasi Individu?
- Fungsi fitness-nya?
- Algoritma EAs?
- Operator-operator evolusi?

Masa Depan *Fuzzy*

- Computers that understand and respond to normal human language.
- Machines that write interesting novels and screenplays in a selected style, such as Hemingway's.
- Huge expert decision makers, theoretically able to extract the wisdom of every document ever written.
- Molecule-sized soldiers of health that will roam the blood-stream, killing cancer cells and slowing the aging process.

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.

Artificial Neural Network

Dr. Suyanto, S.T., M.Sc.
HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

Bagaimana manusia belajar?



Menangis?

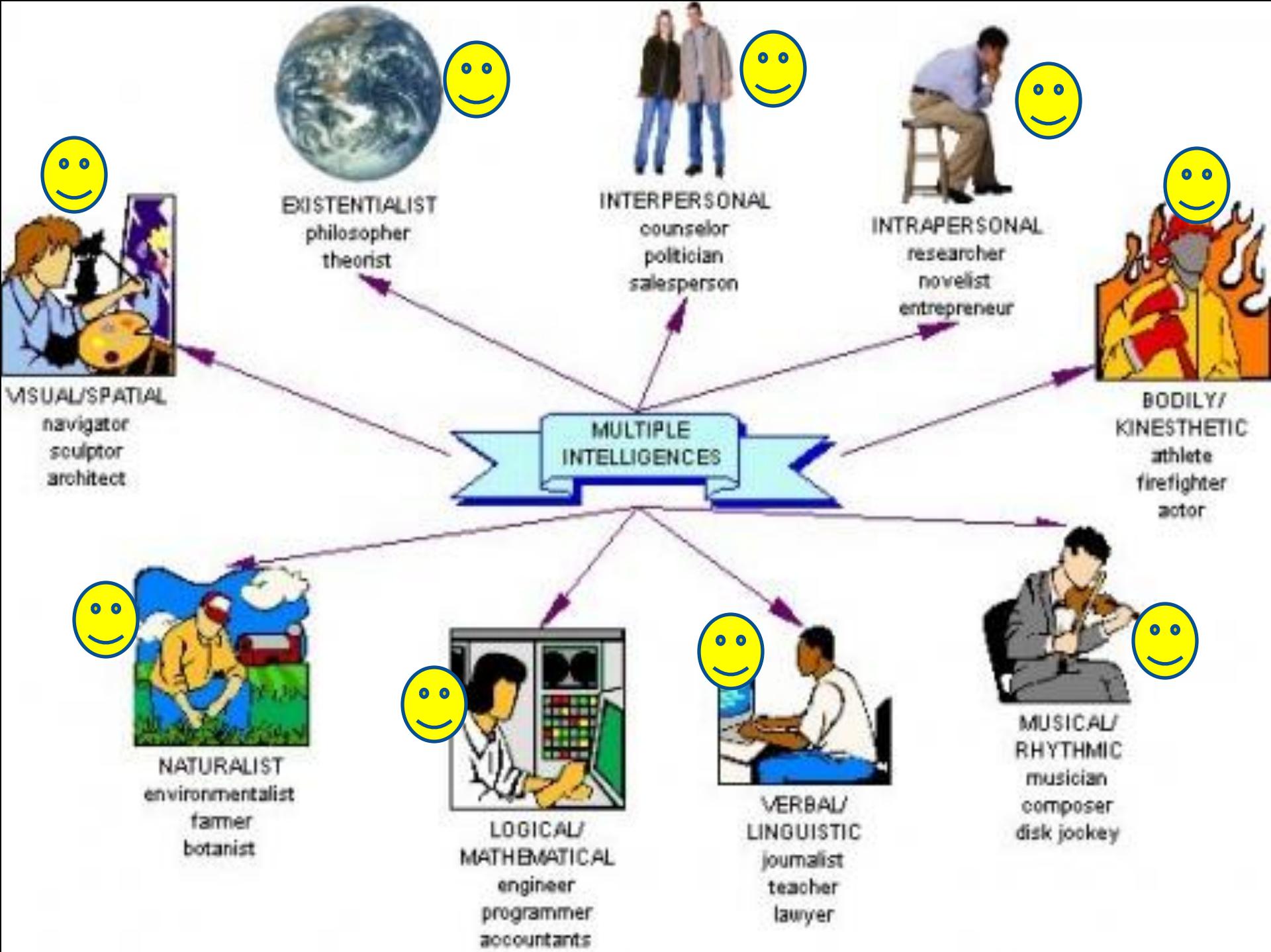
Merayap - Berjalan?

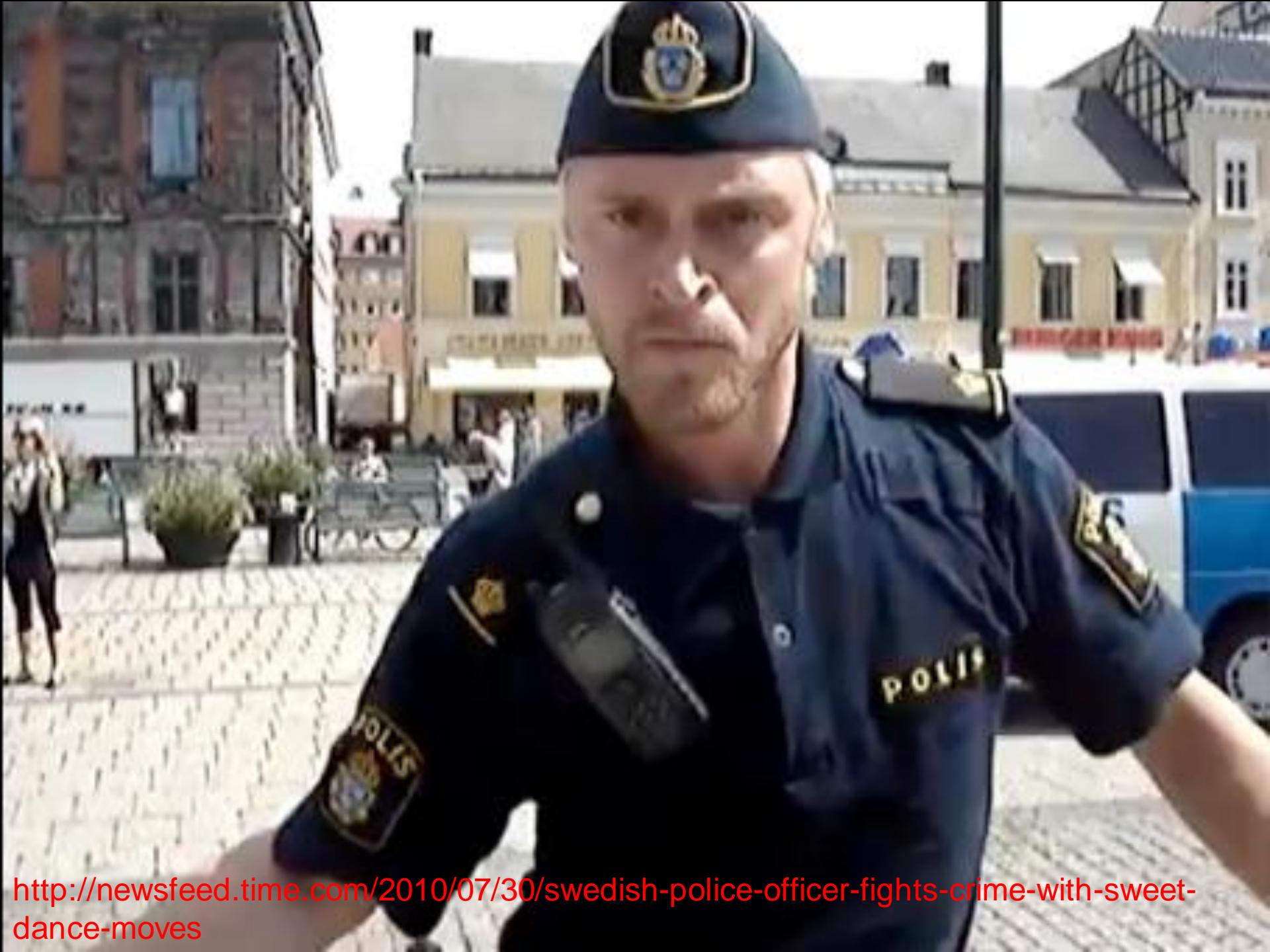
Arah?

Berbahasa?

Logika?

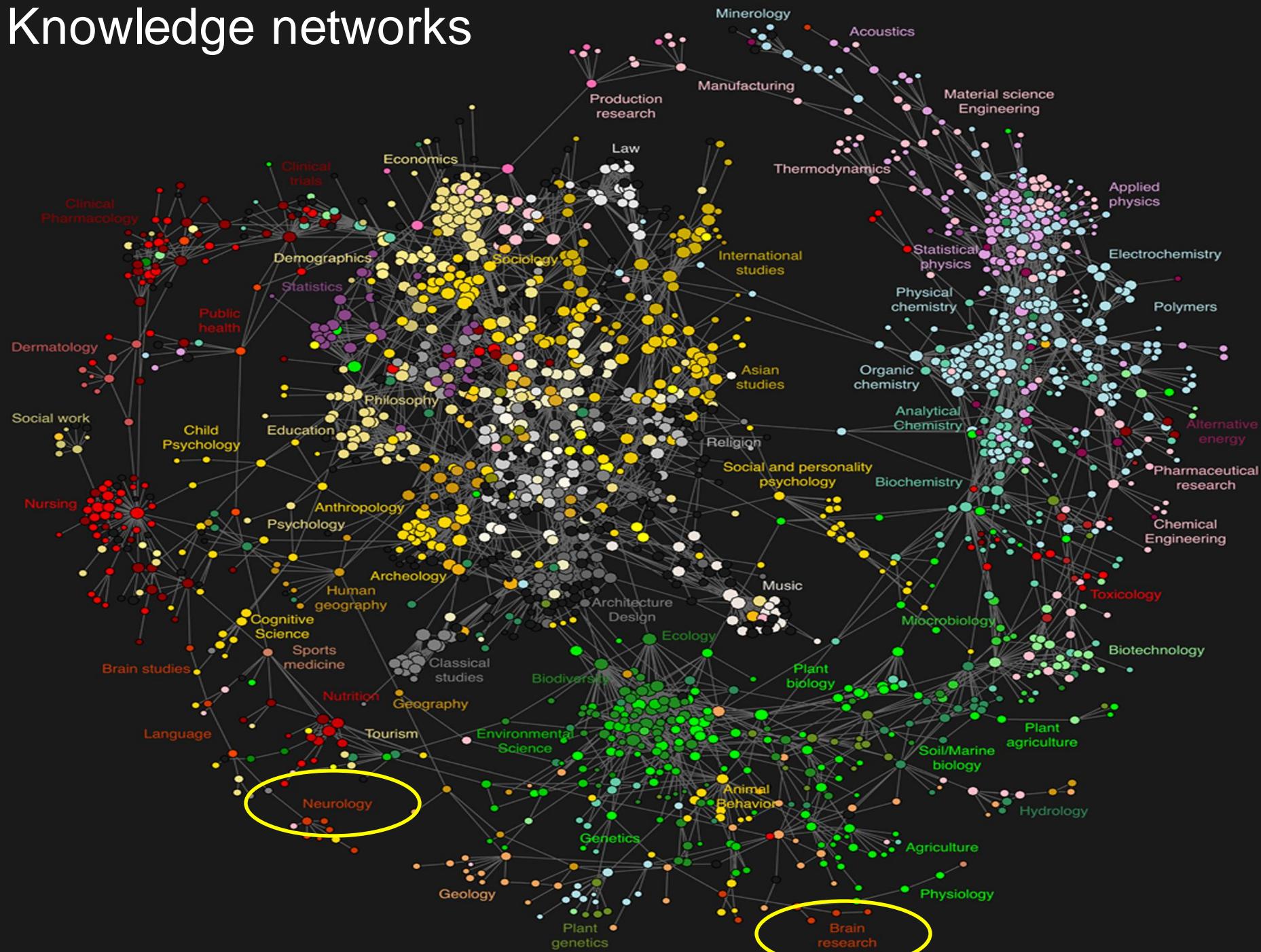
Multiple Intelligence !!!



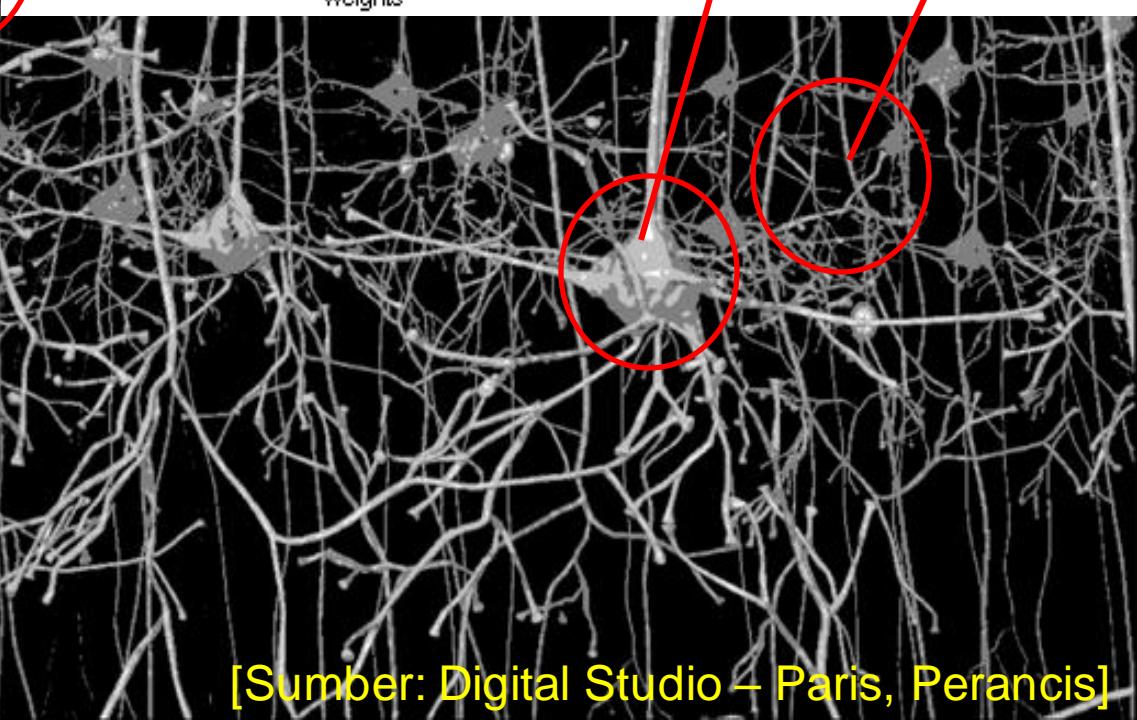
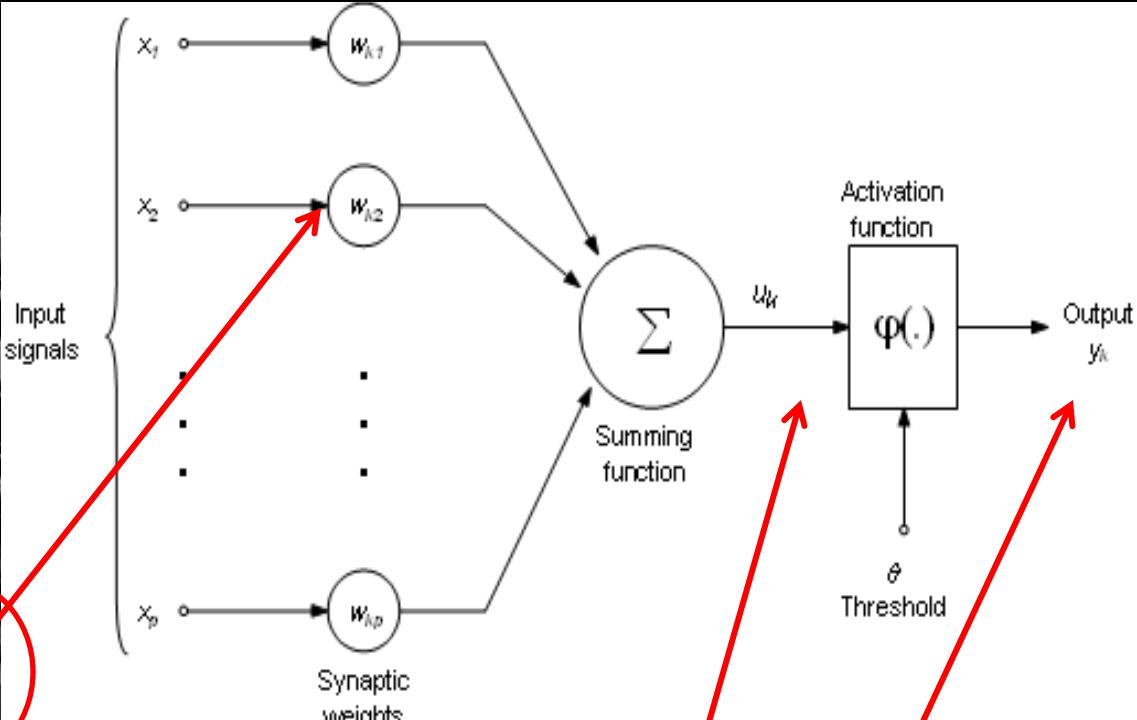
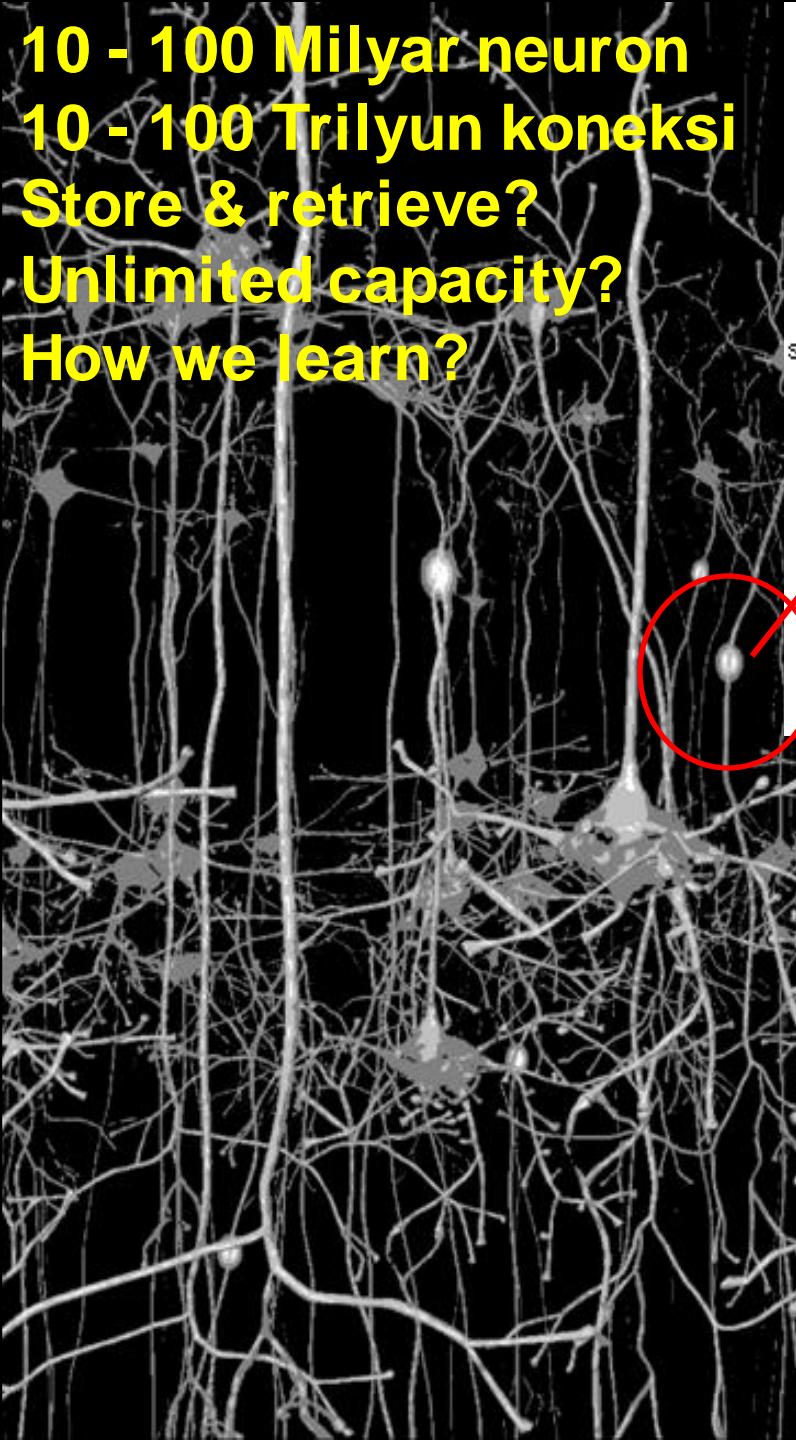


<http://newsfeed.time.com/2010/07/30/swedish-police-officer-fights-crime-with-sweet-dance-moves>

Knowledge networks



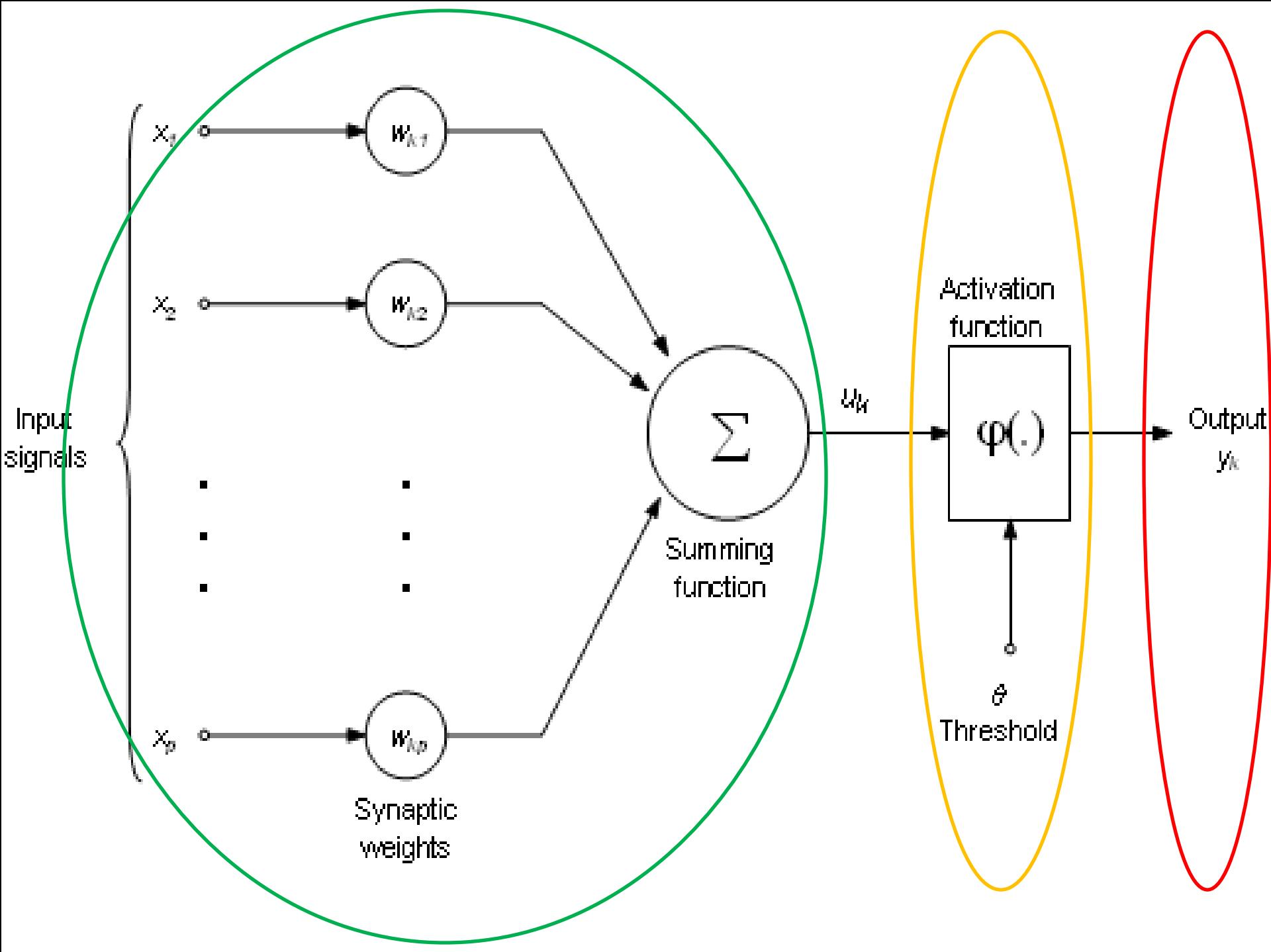
**10 - 100 Milyar neuron
10 - 100 Trilyun koneksi
Store & retrieve?
Unlimited capacity?
How we learn?**



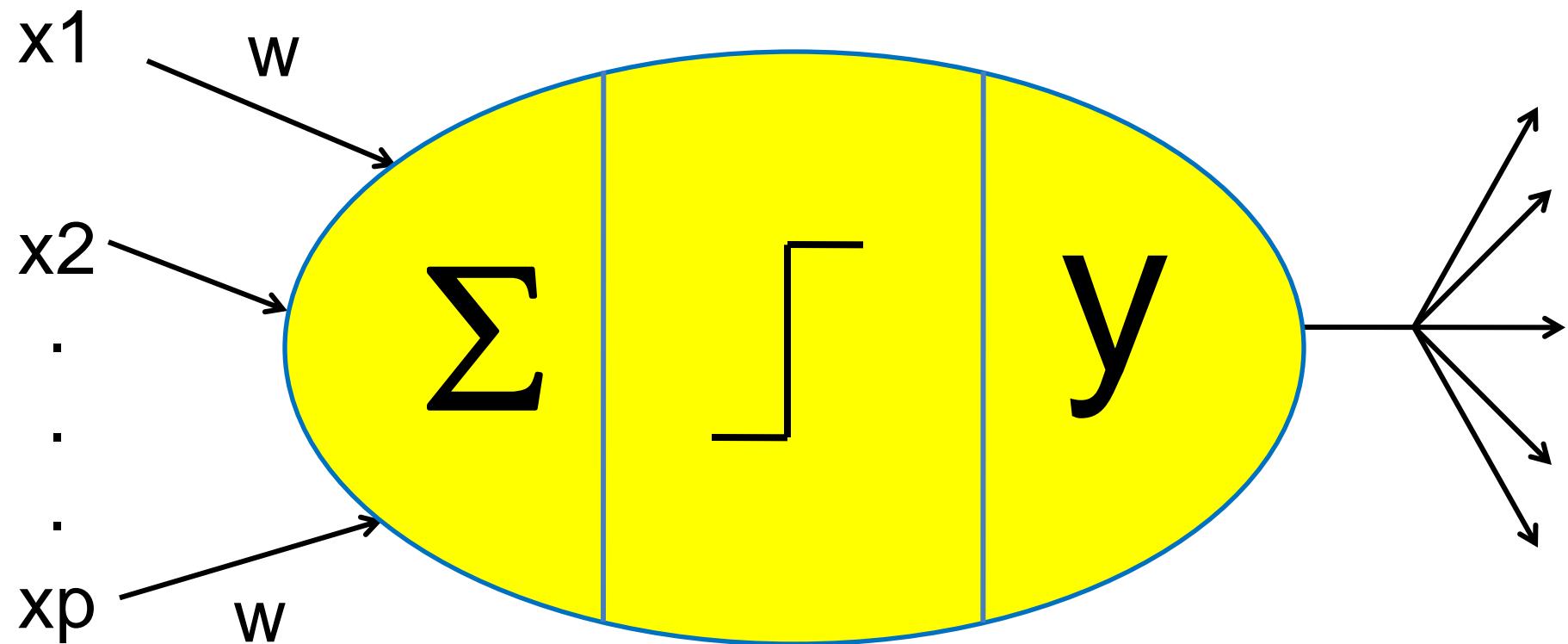
[Sumber: Digital Studio – Paris, Perancis]

Perceptron

- **Neuron:** Sel syaraf biologis
- **Perceptron:** Sel syaraf buatan
 - Input function
 - Activation function
 - Output



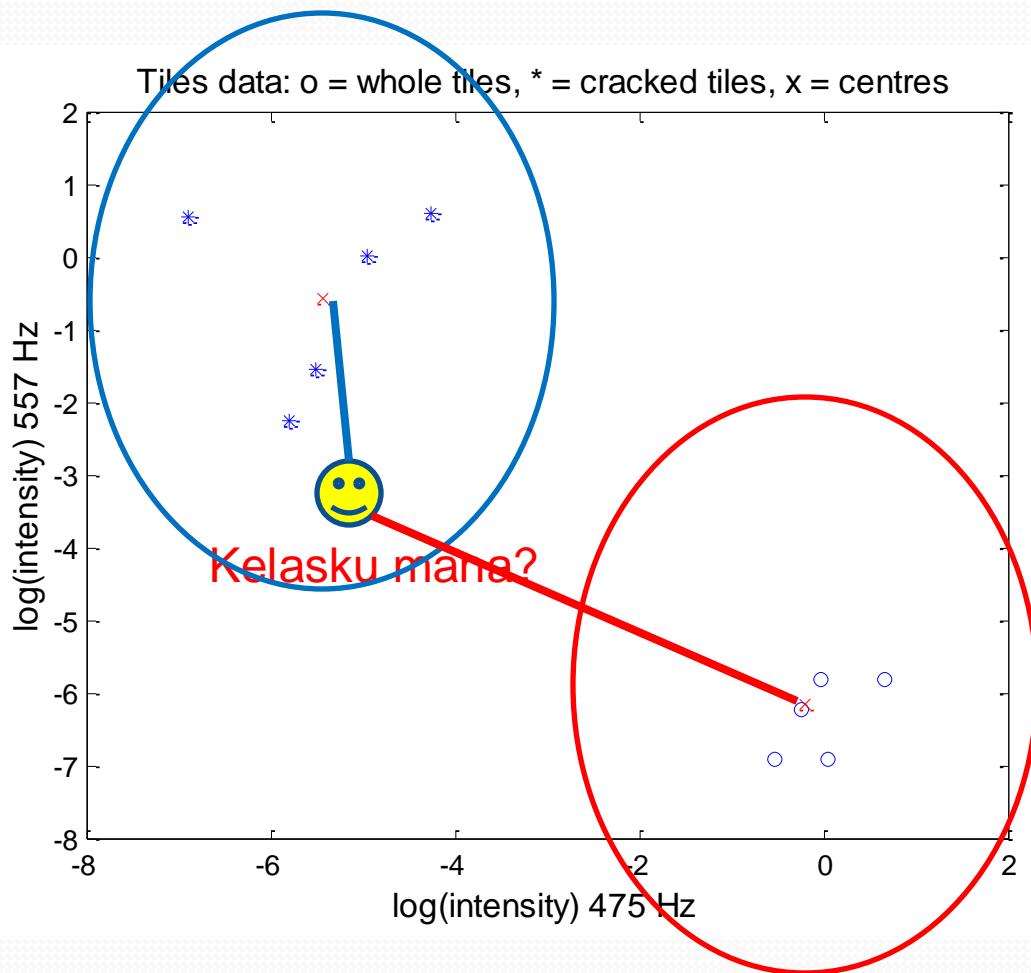
Perceptron



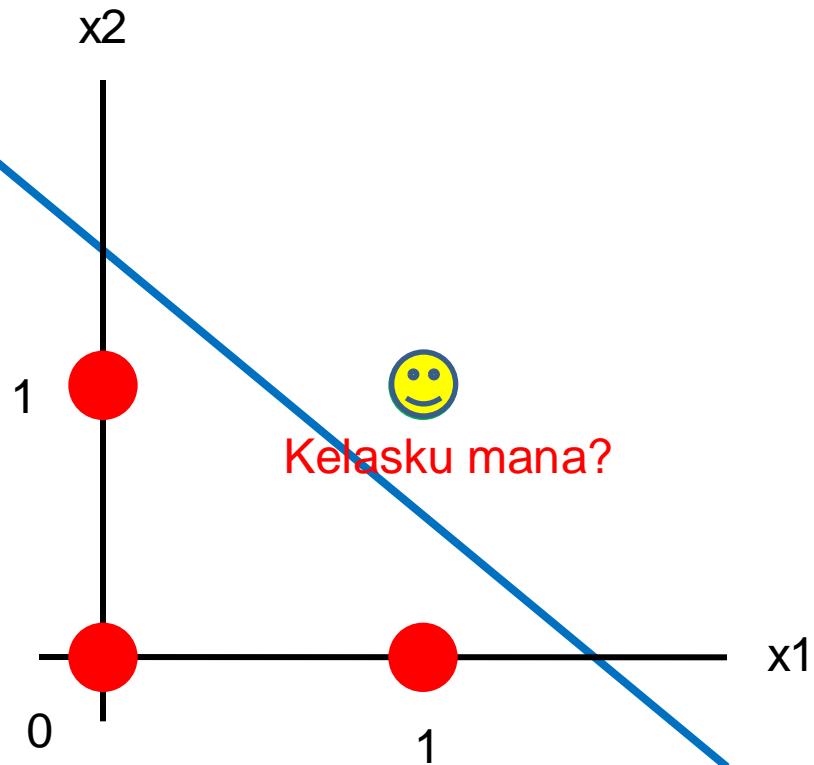
Perceptron

- Jika manusia punya 10 milyar neuron, apa yang bisa dilakukan?
 - Sangat banyak hal bisa dilakukan
 - Apalagi jika Multiple Intelligence
- 
- Perceptron = MODEL SEDERHANA dari neuron
 - Apa yang bisa dilakukan oleh satu perceptron?
 - **Klasifikasi**
 - **Prediksi**
 - **Optimasi, ...**

Fuzzy C-Means (FCM)



AND



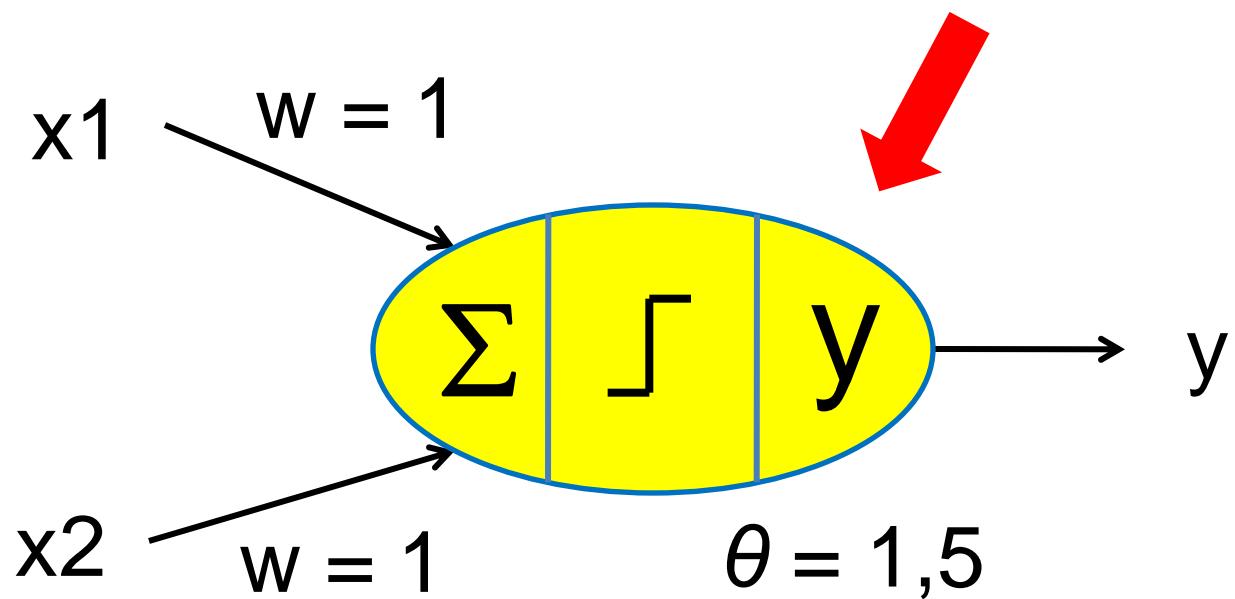
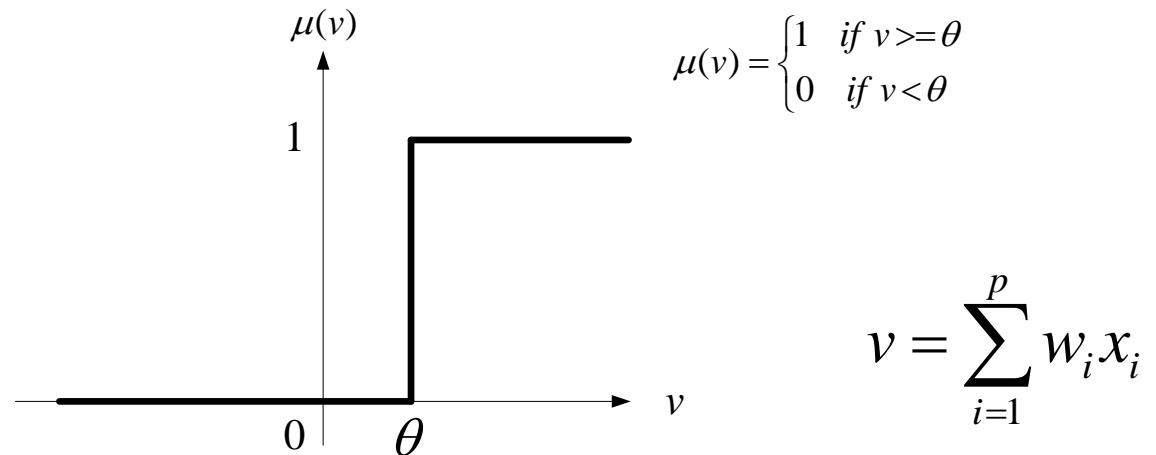
x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

$$x_1 + x_2 - 1,5 = 0$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 - 1,5 = 0$$

AND

x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1



OR

x_2

1

0



1

0

1

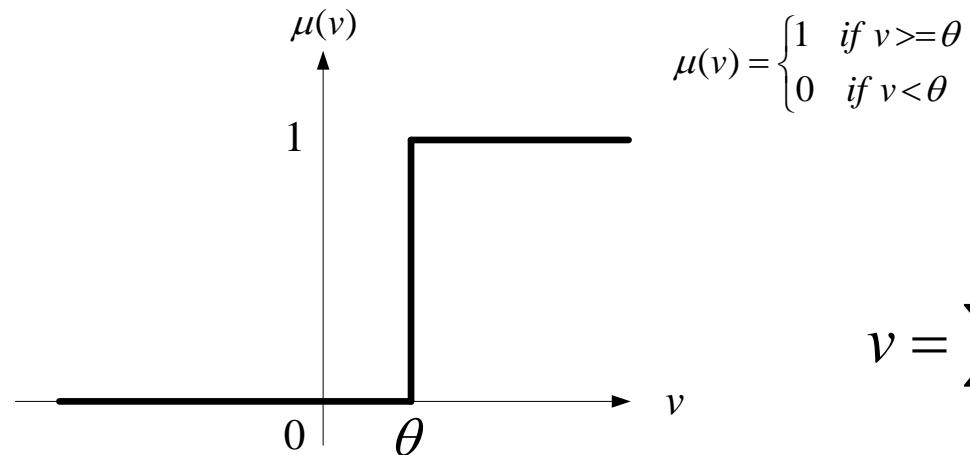
x_1

$$x_1 + x_2 - 0,5 = 0$$

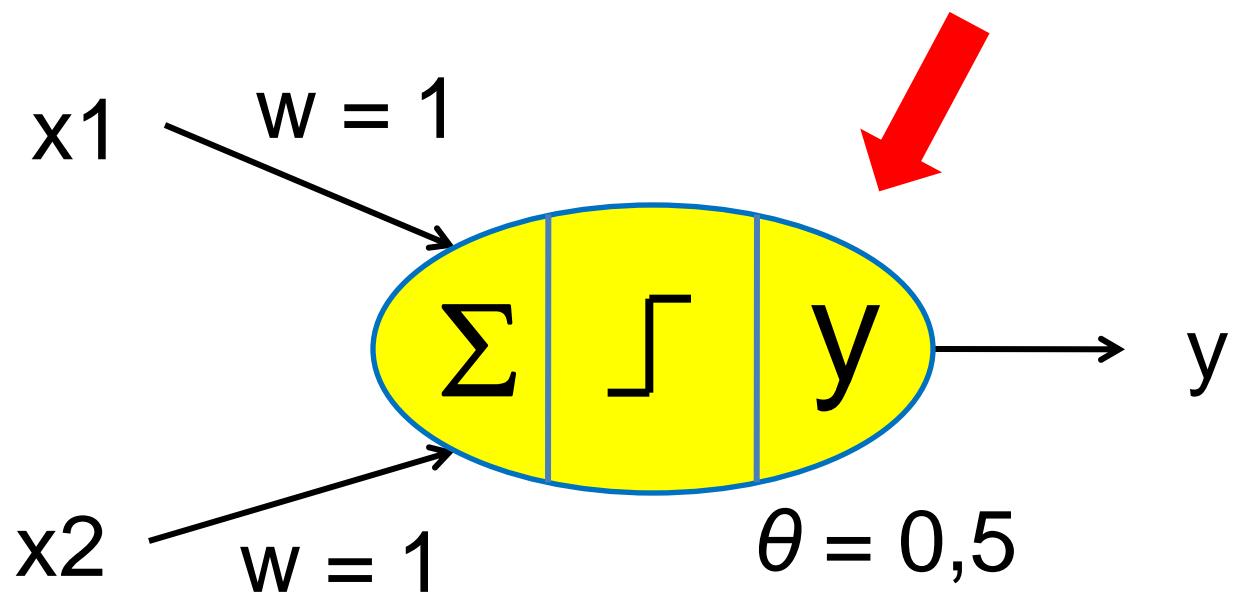
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

OR

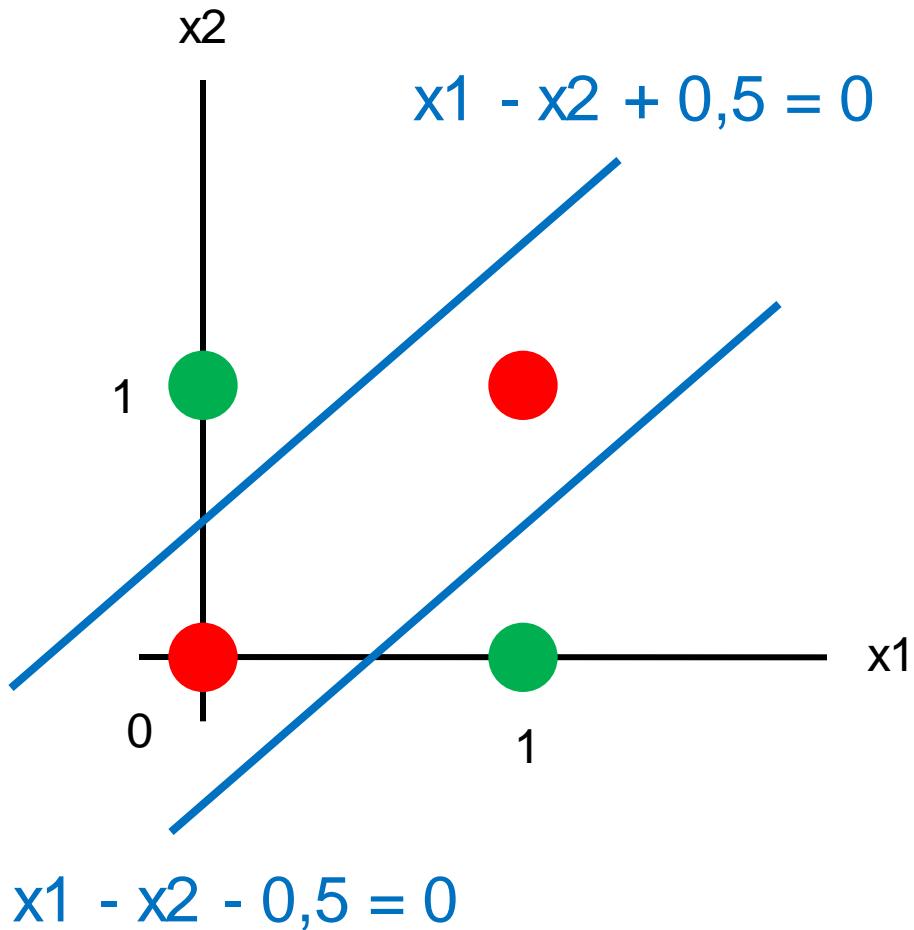
x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	1



$$v = \sum_{i=1}^p w_i x_i$$



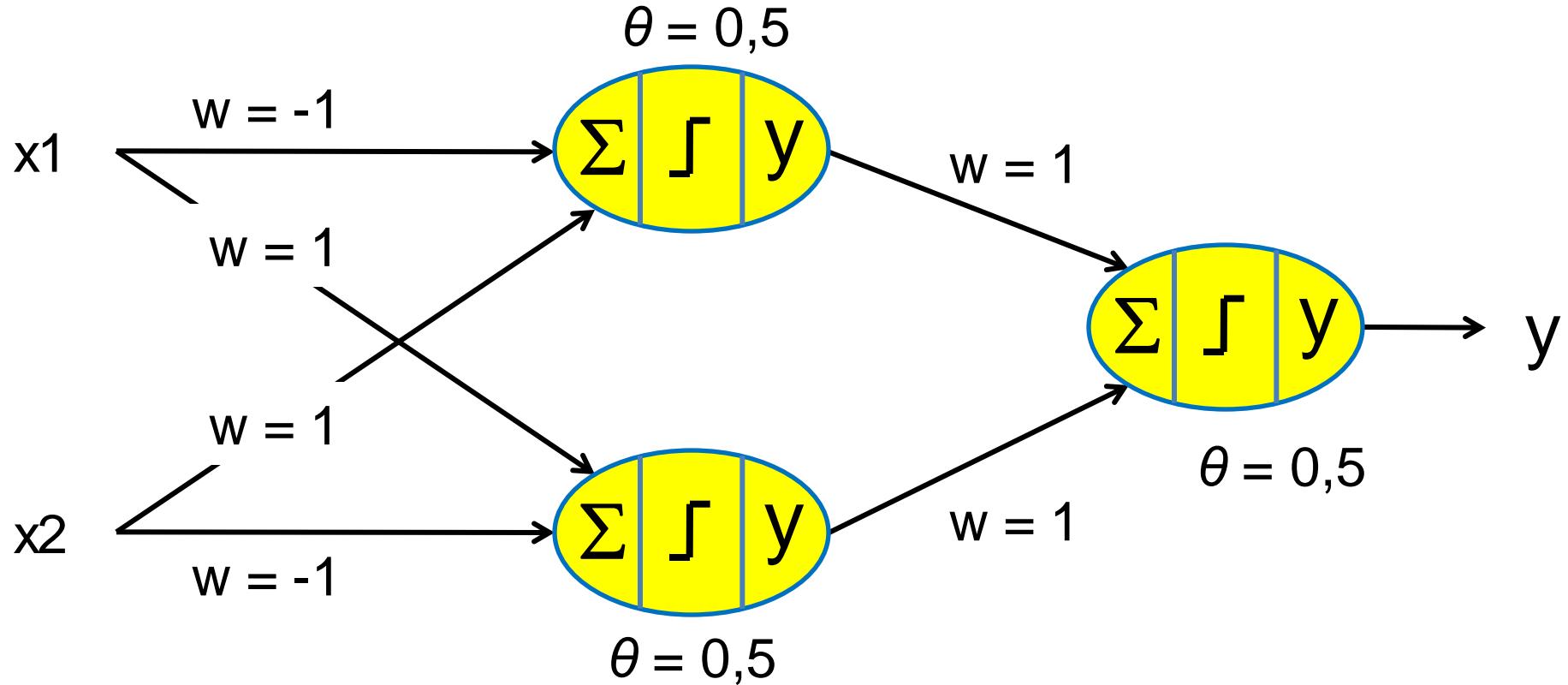
XOR



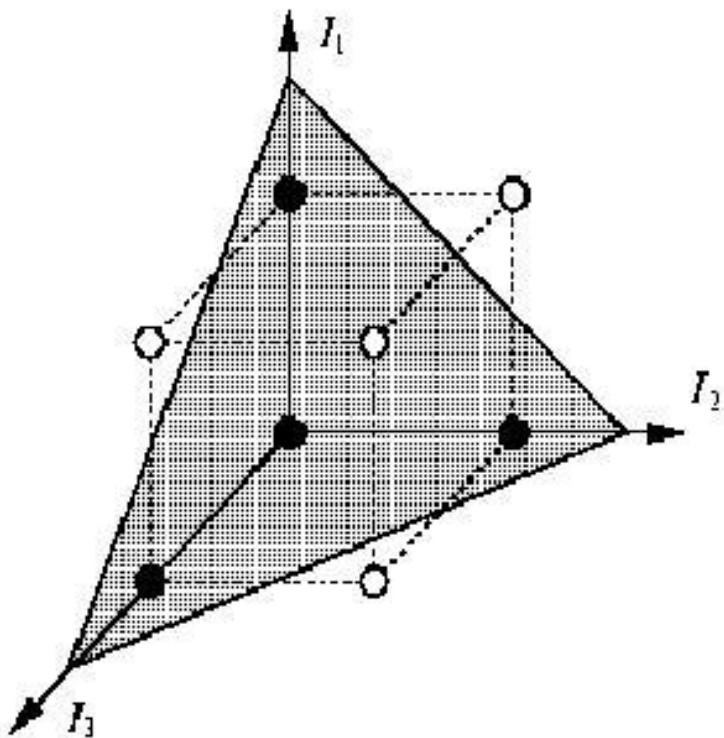
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

XOR

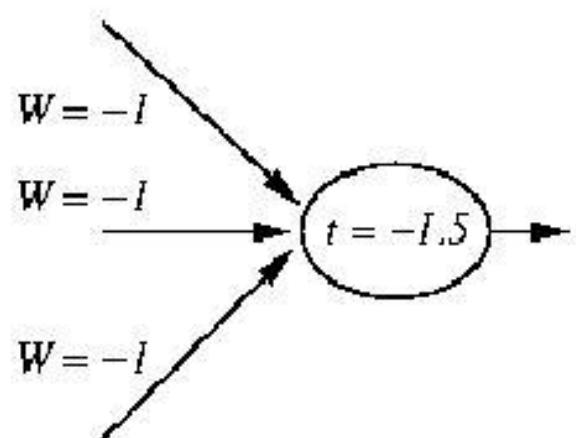
x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0



3 elemen input → 3 dimensi

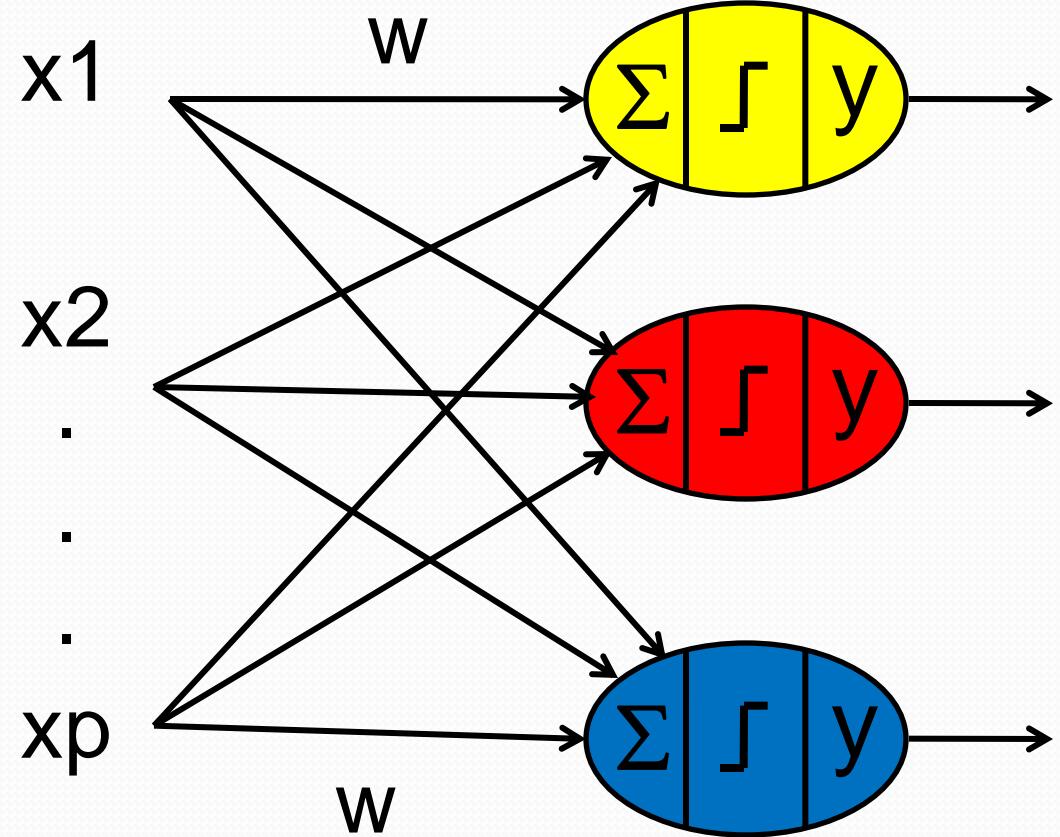
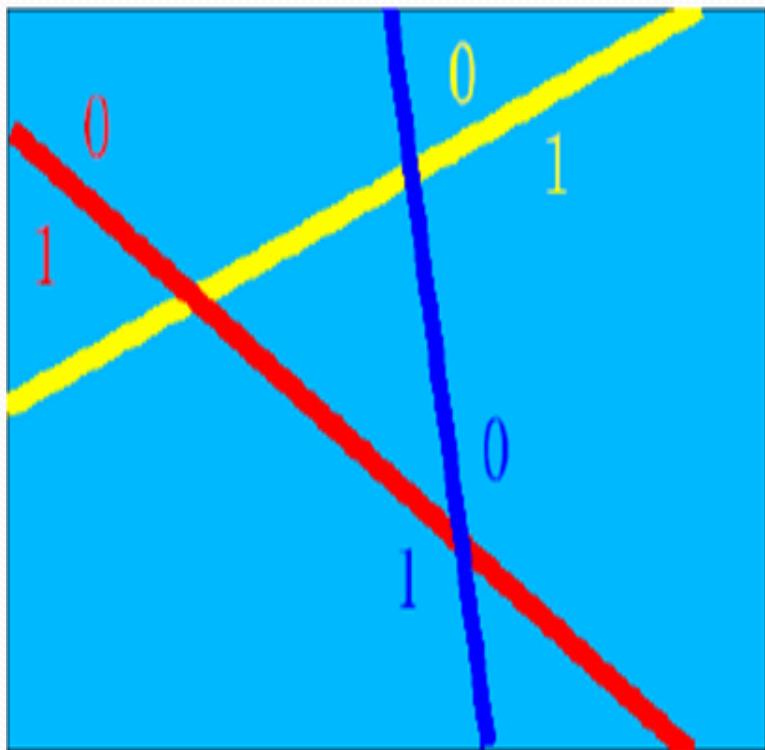


(a) Separating plane



(b) Weights and threshold

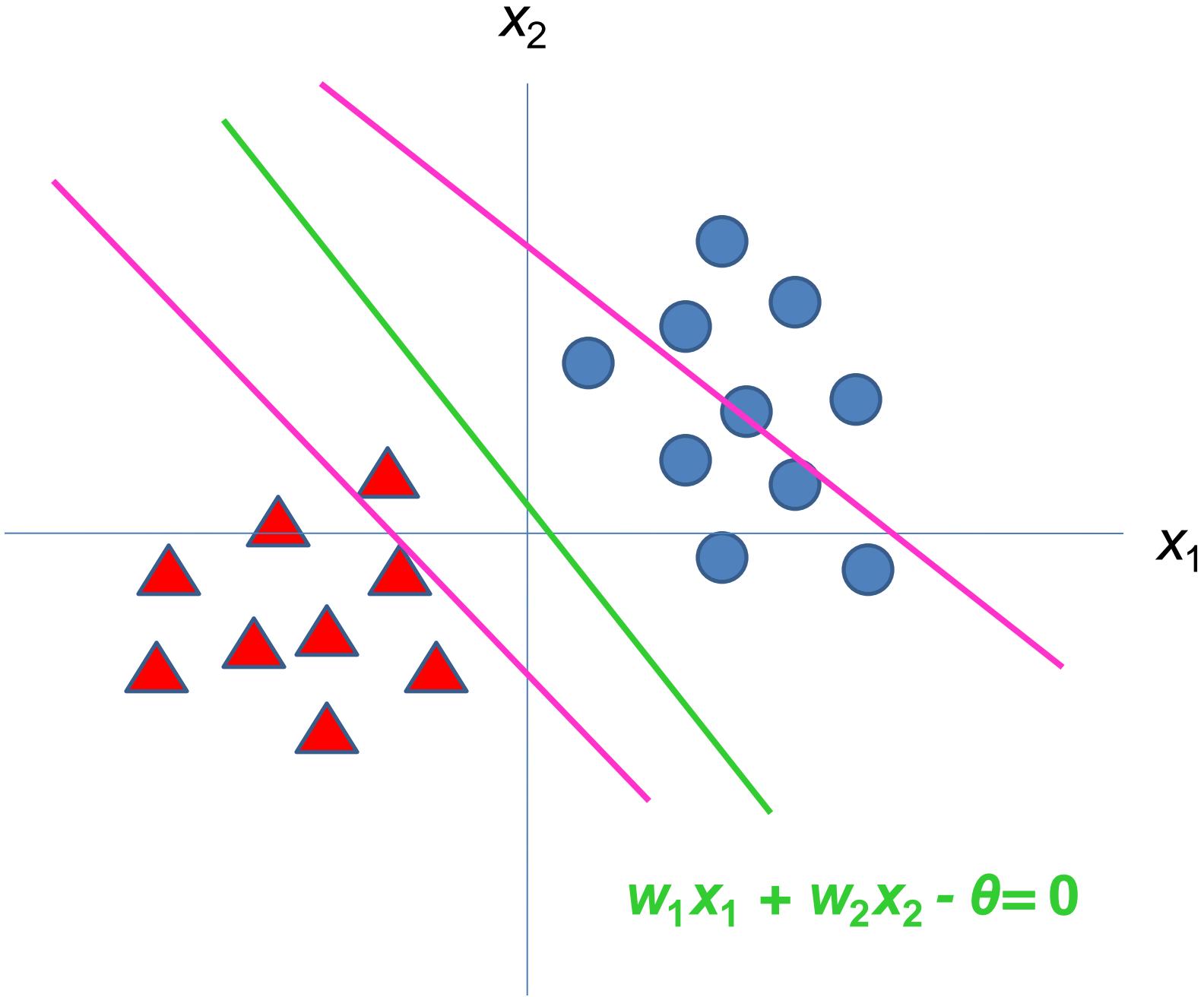
Perceptron Network

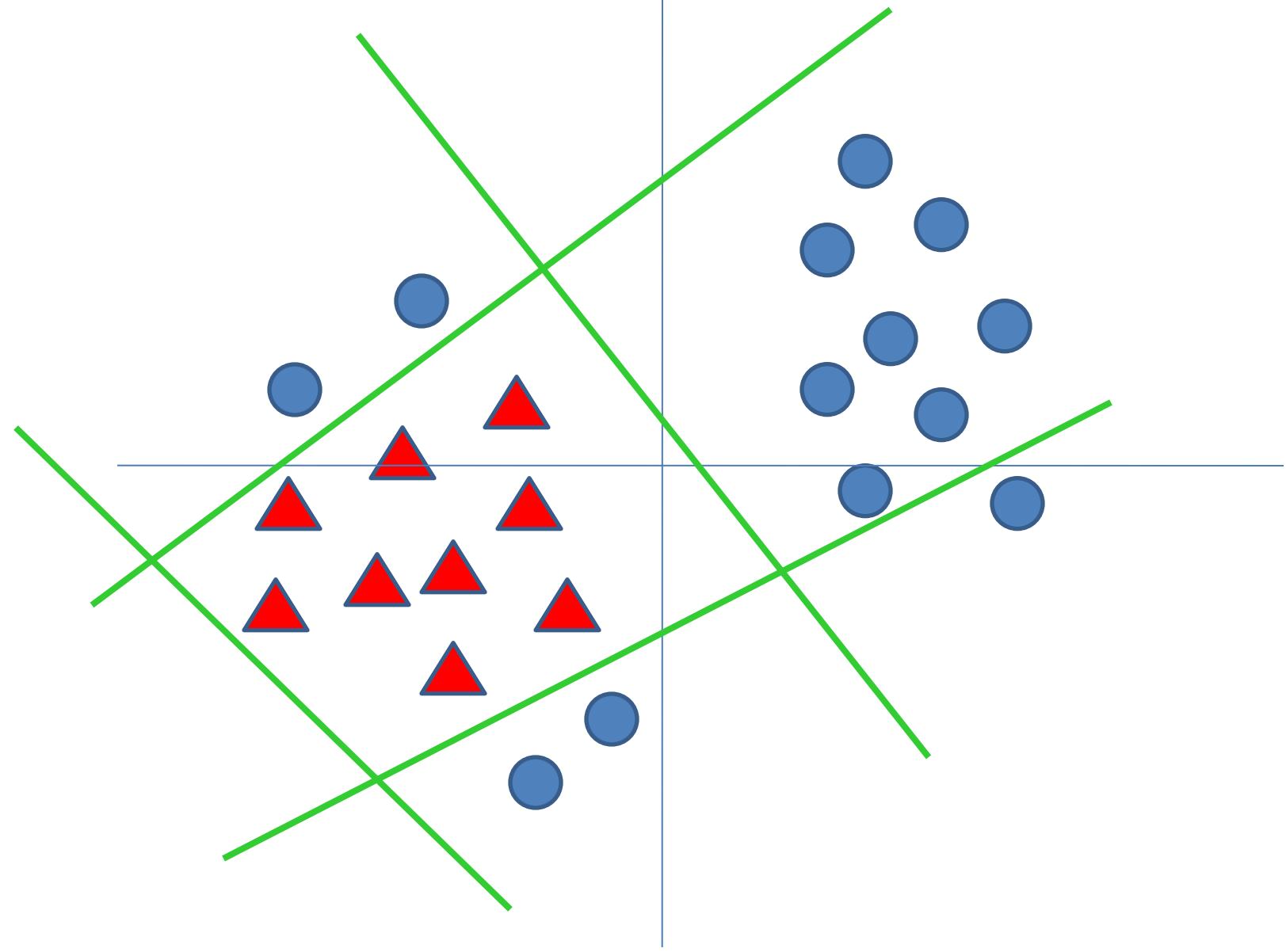


Learning

Bagaimana menemukan
weights yang tepat?

Meminimumkan error

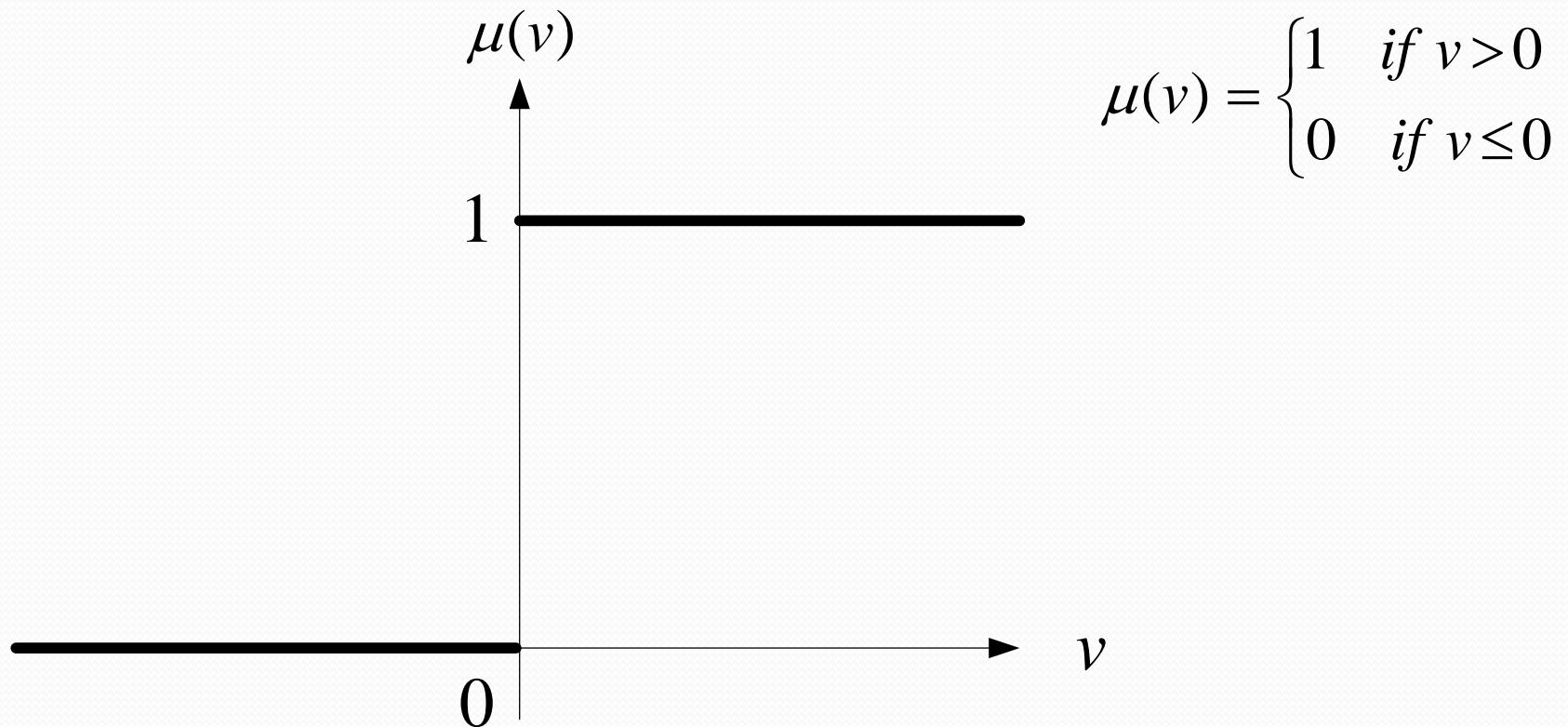


x_2 x_1 

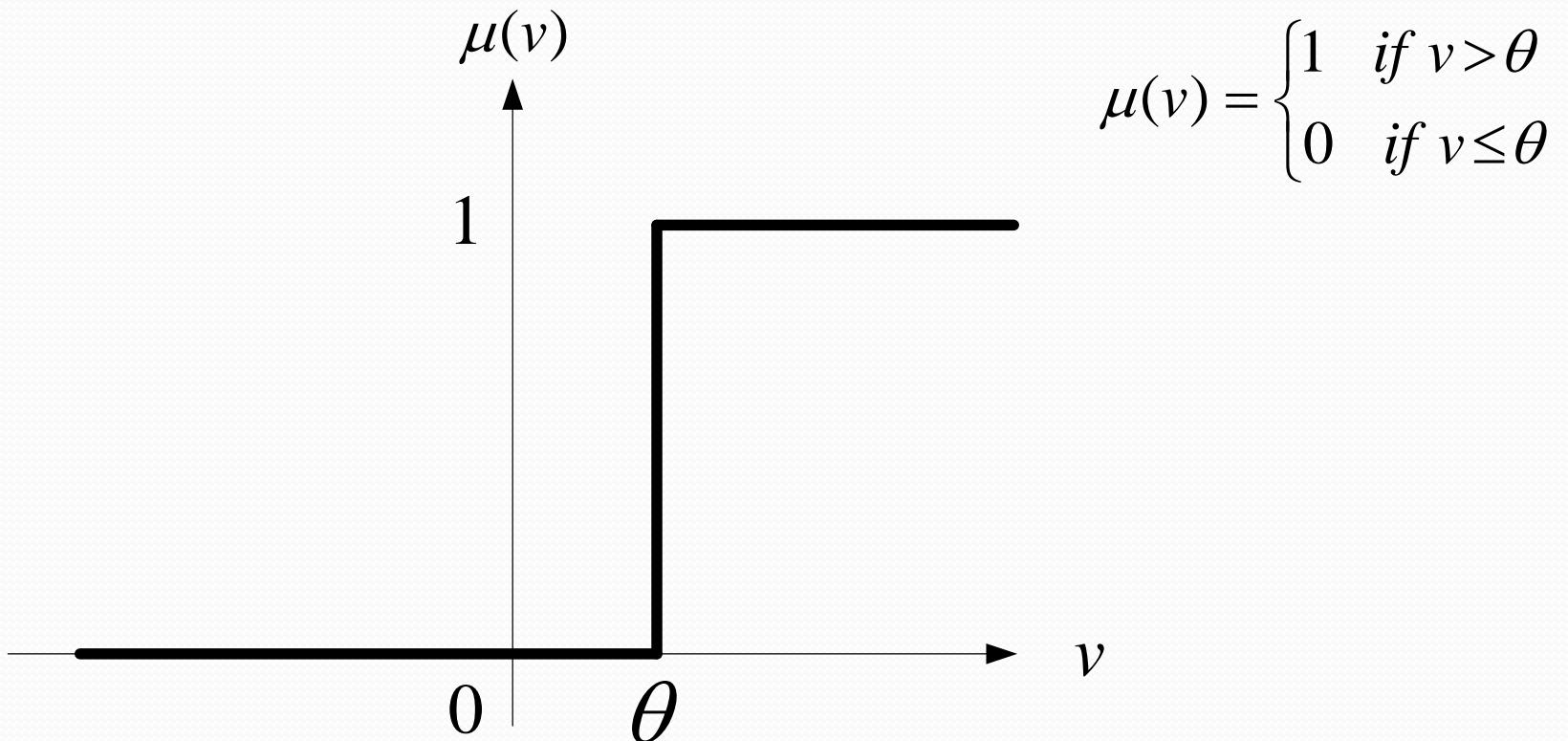
Activation Functions

- *Hard Limit*
- *Threshold*
- *Linear (Identity)*
- *Sigmoid*
- *Radial Basis Function (RBF)*
- ...

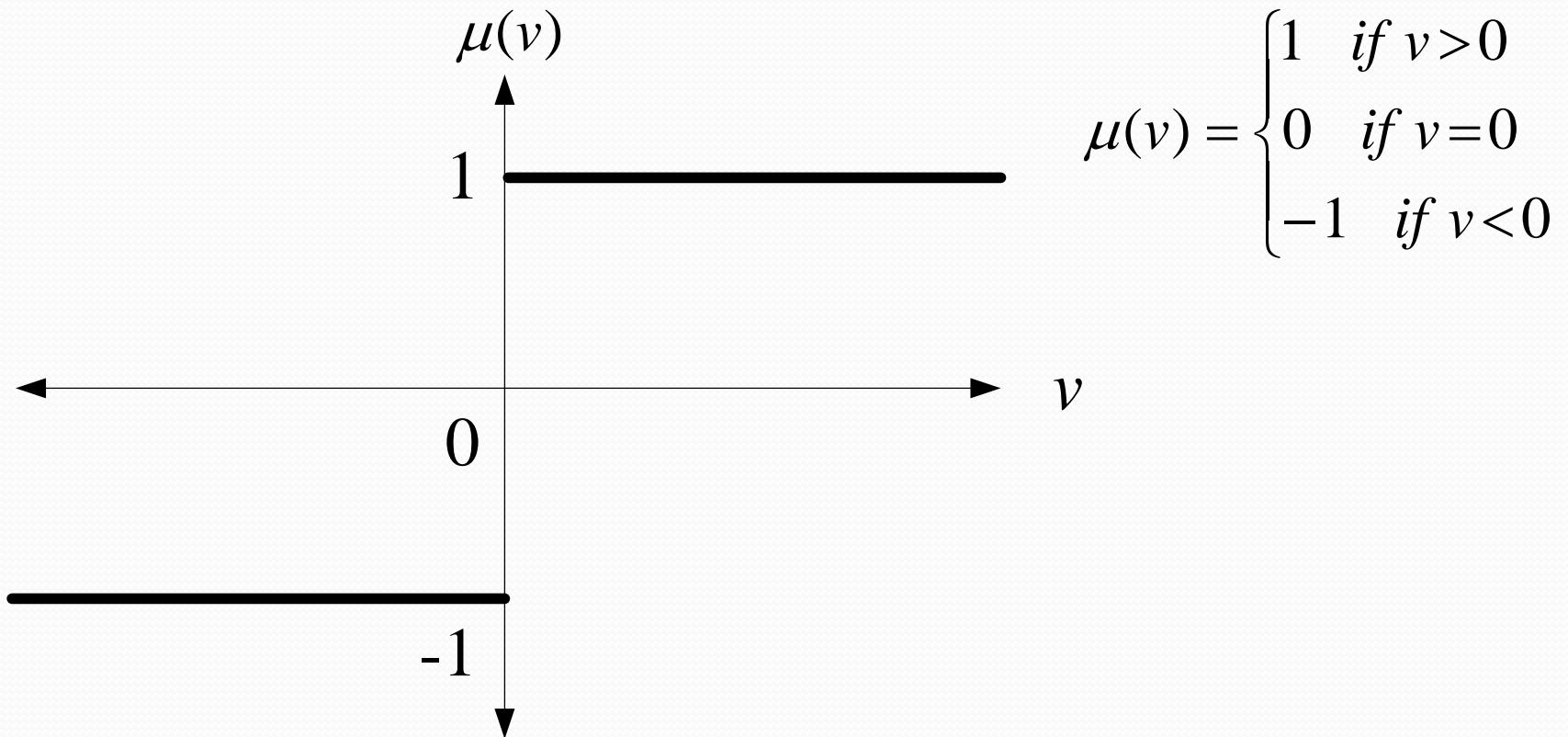
Hard Limit



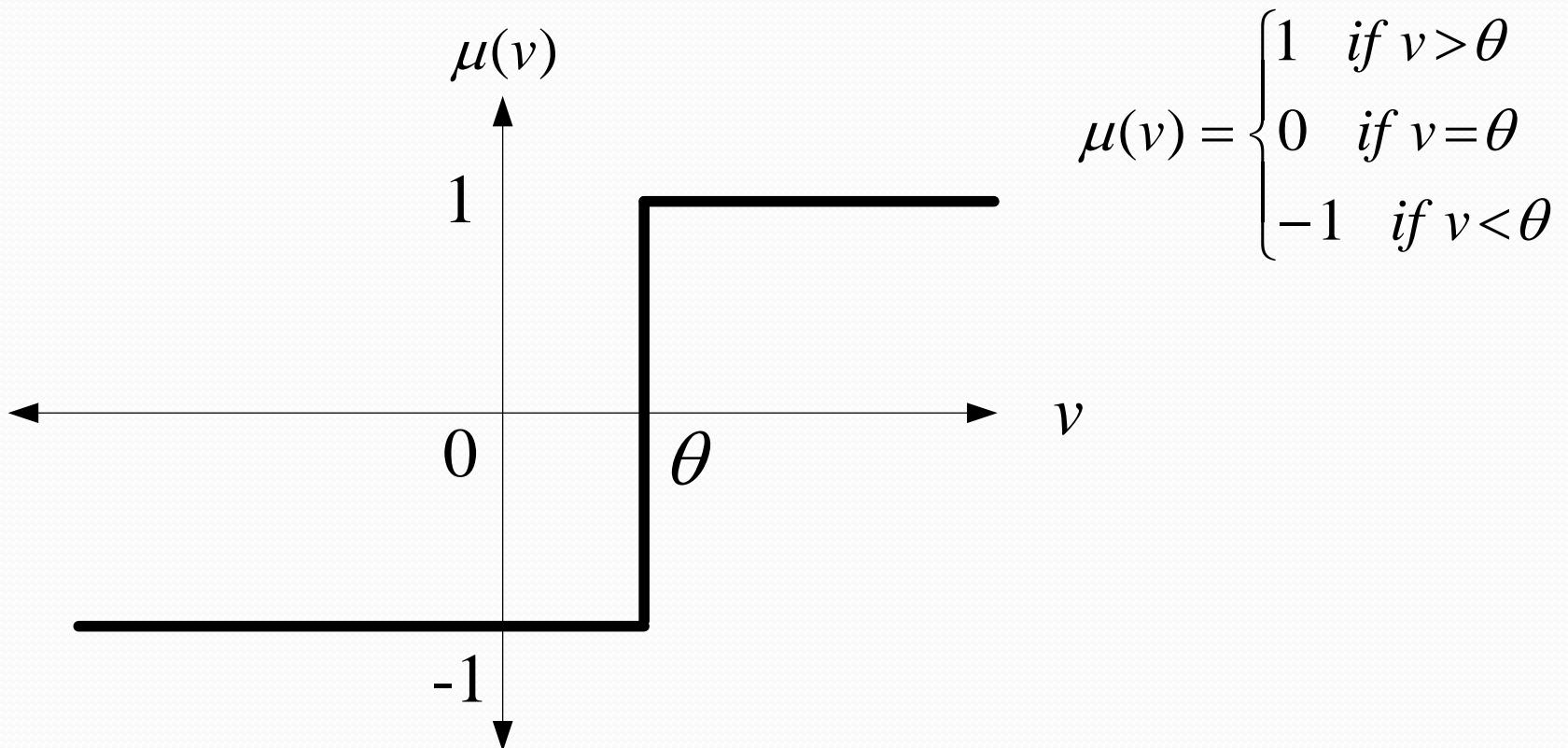
Threshold



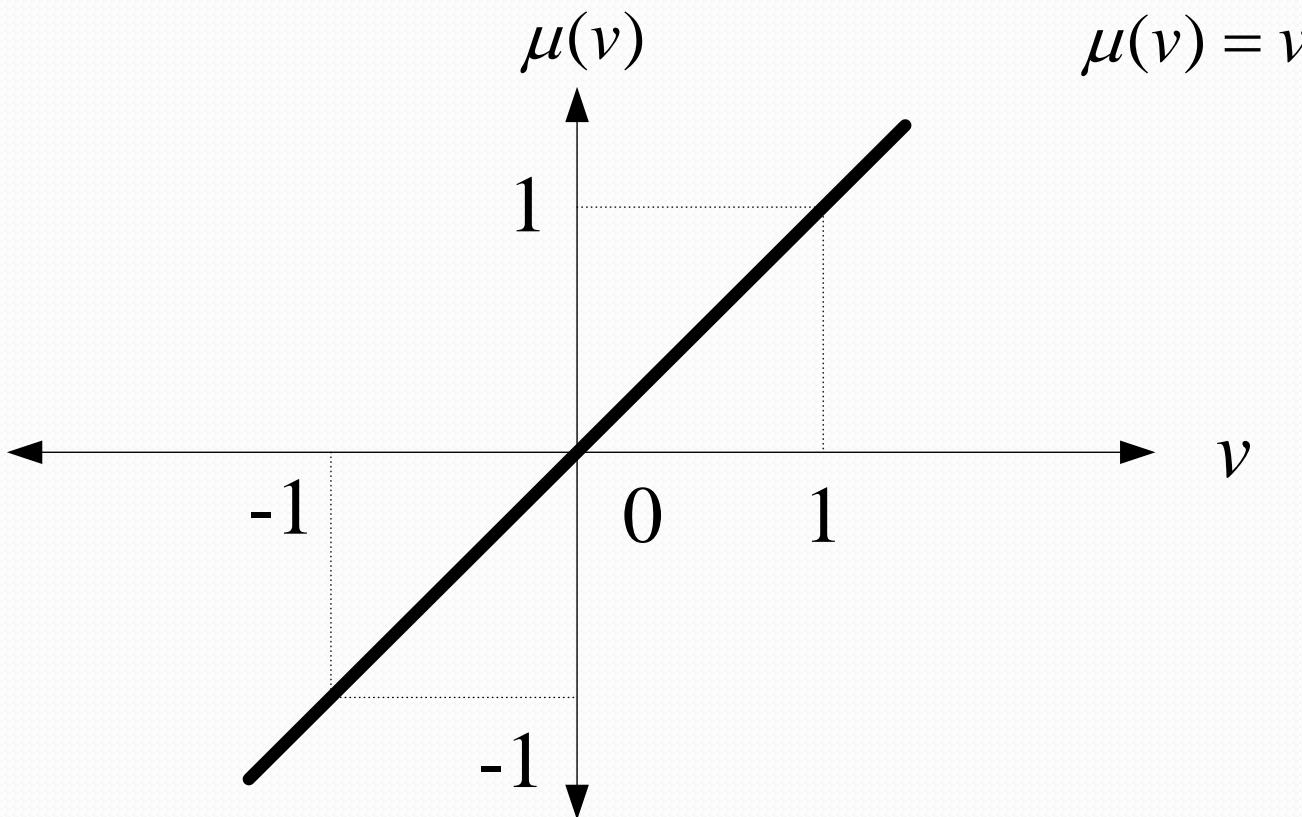
Symmetric Hard Limit



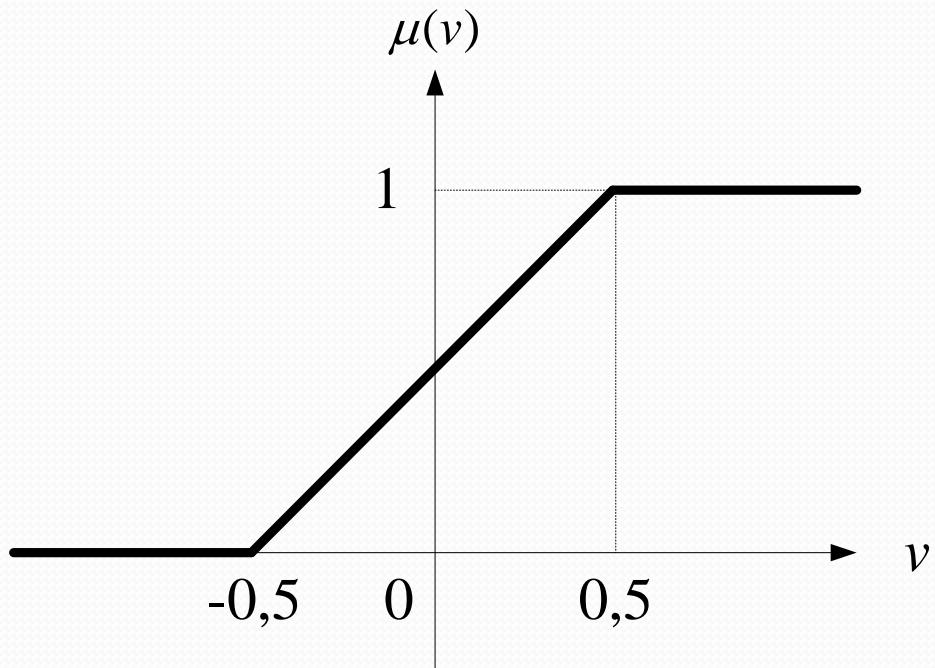
Bipolar Threshold



Linear (Identity)

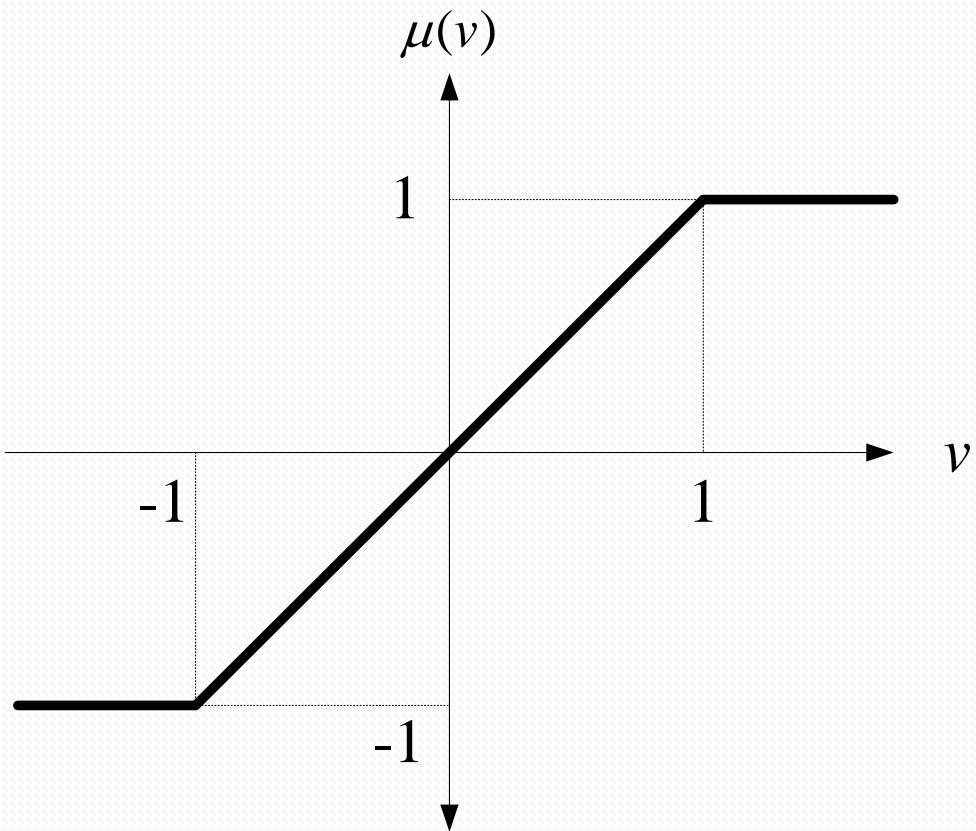


Piecewise-linear



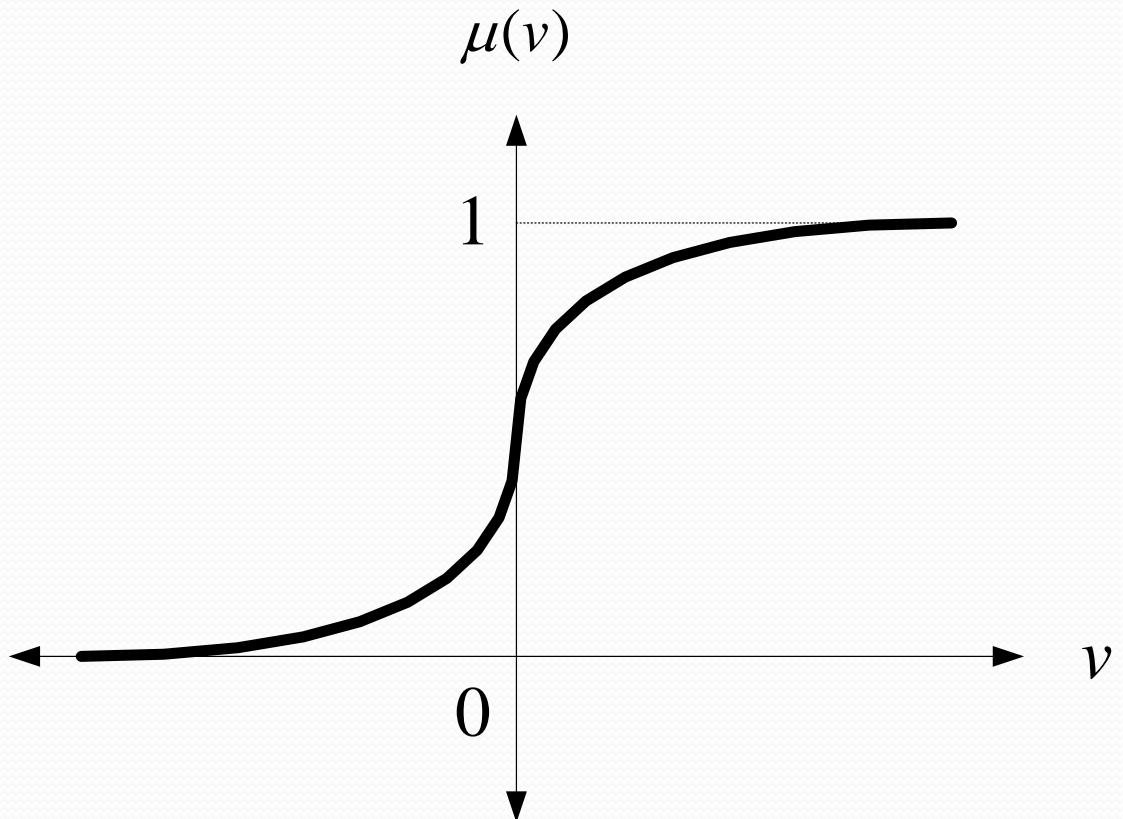
$$\mu(v) = \begin{cases} 1 & \text{jika } v \geq 0,5 \\ v + 0,5 & \text{jika } 0,5 > v > -0,5 \\ 0 & \text{jika } v \leq -0,5 \end{cases}$$

Symmetric Piecewise-linear



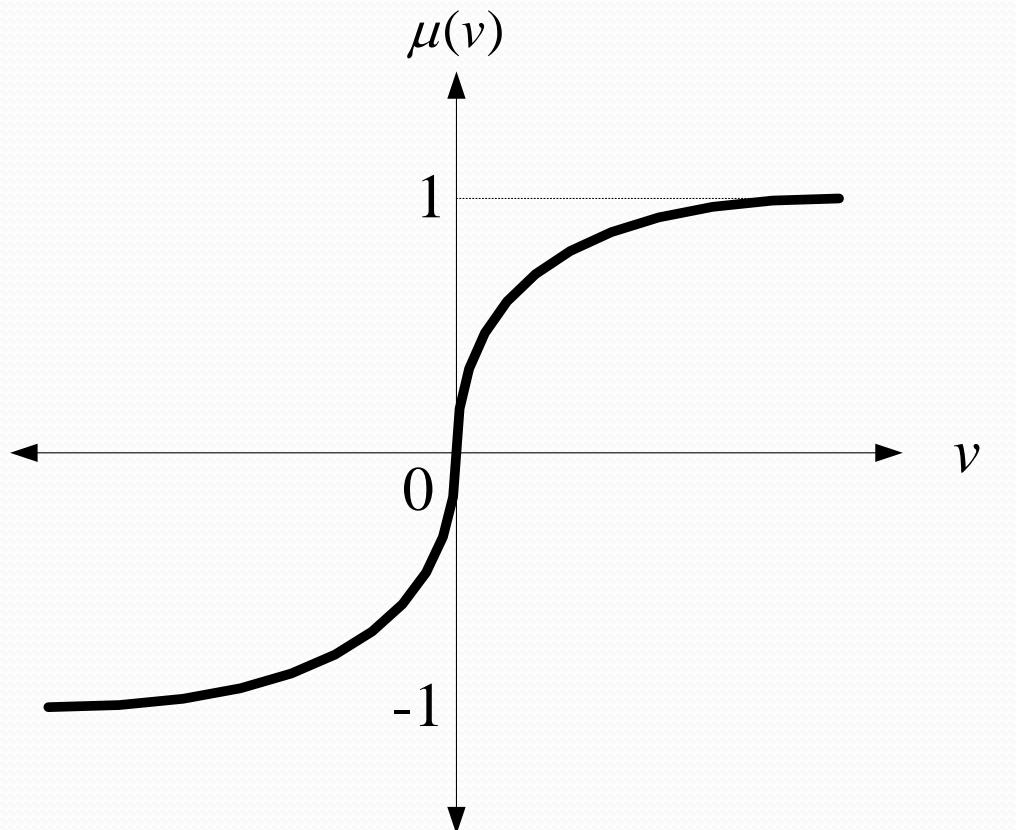
$$\mu(v) = \begin{cases} 1 & \text{jika } v \geq 1 \\ v & \text{jika } -1 < v < 1 \\ -1 & \text{jika } v \leq -1 \end{cases}$$

Sigmoid



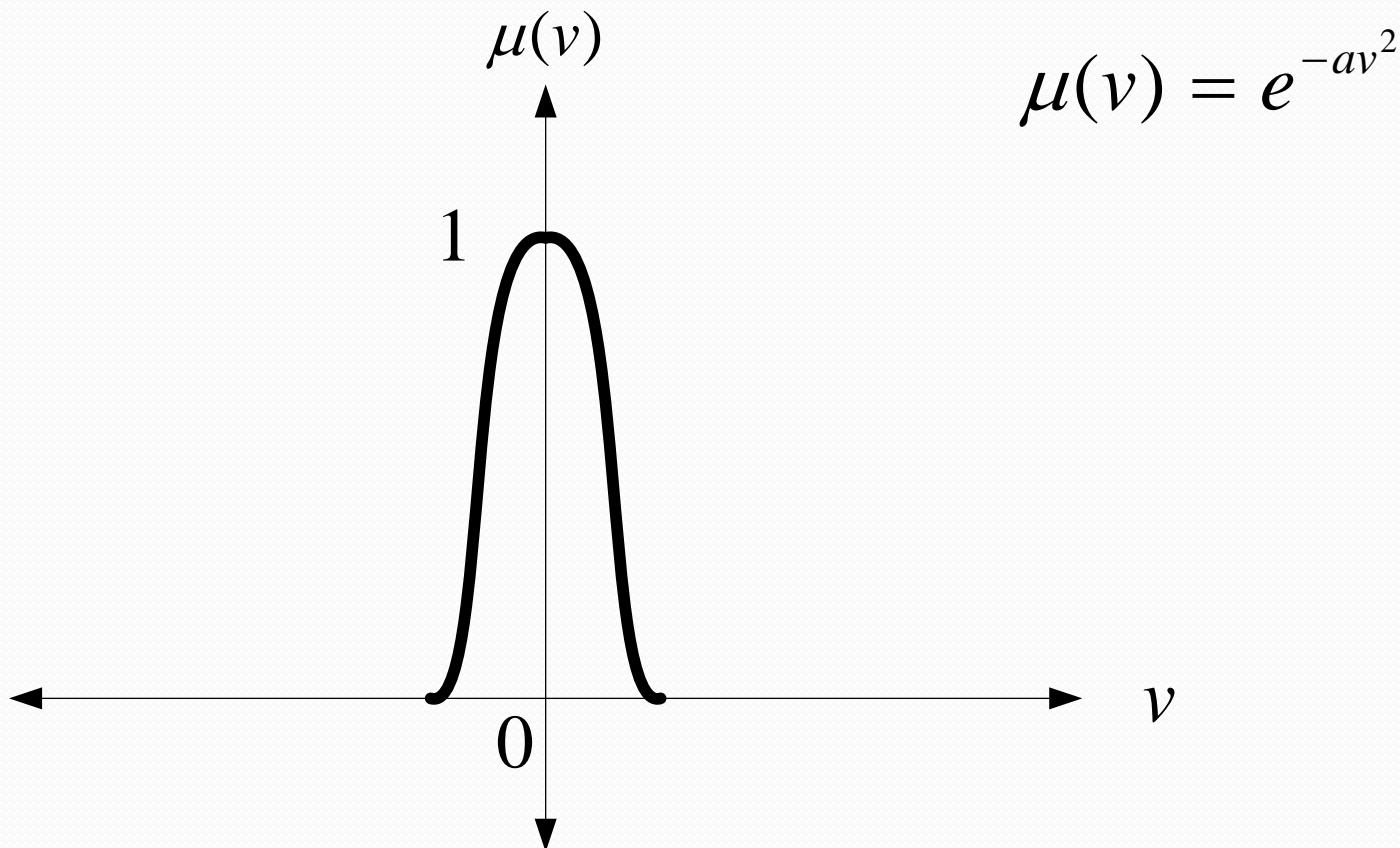
$$\mu(v) = \frac{1}{1+e^{-av}}$$

Symmetric (Bipolar) Sigmoid



$$\mu(v) = \frac{1 - e^{-av}}{1 + e^{-av}}$$

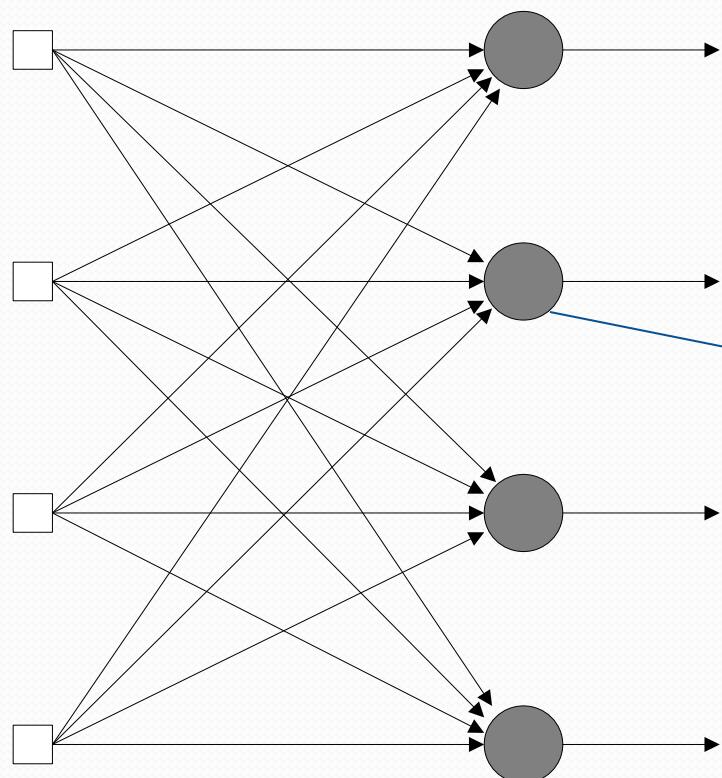
Radial Basis Function (RBF)



Arsitektur ANN

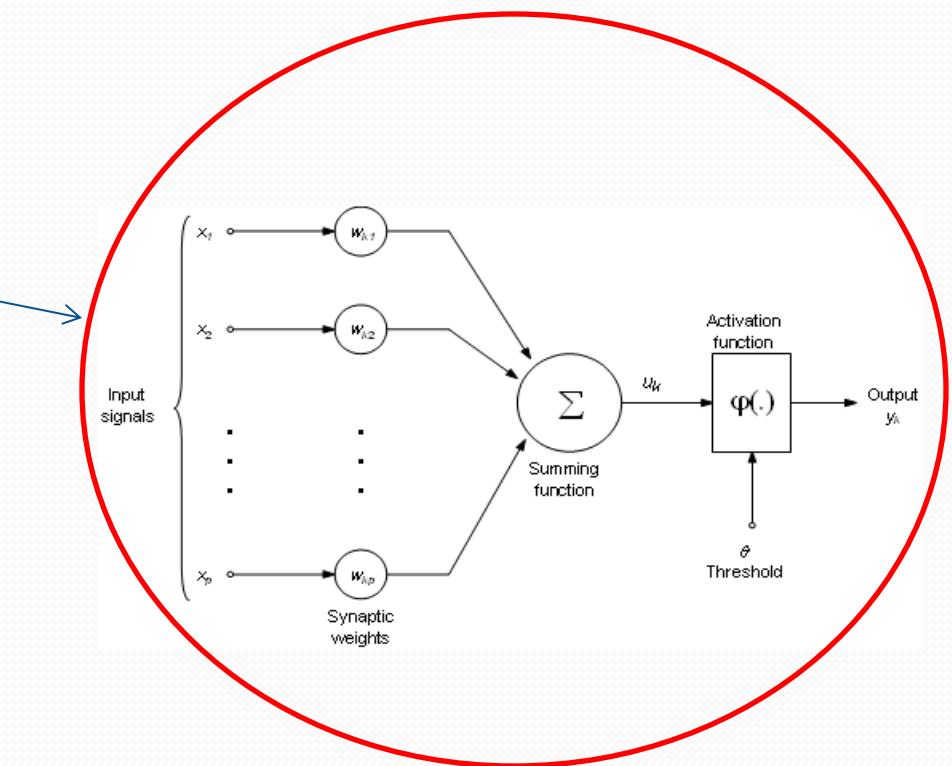
- Para ahli memodelkan sel syaraf otak manusia ke dalam berbagai arsitektur ANN (susunan *neuron*) yang berbeda-beda.
- Masing-masing arsitektur menggunakan algoritma belajar khusus.

Single-Layer Feedforward Networks

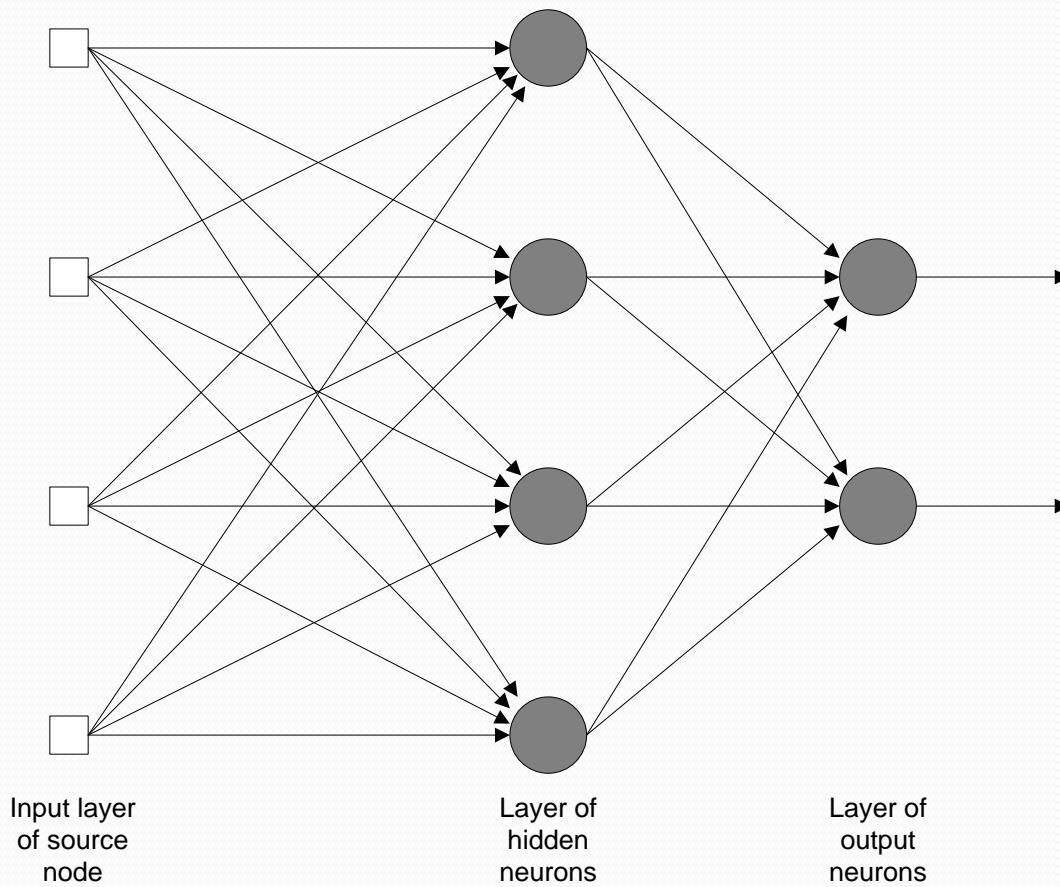


Input layer
of source node

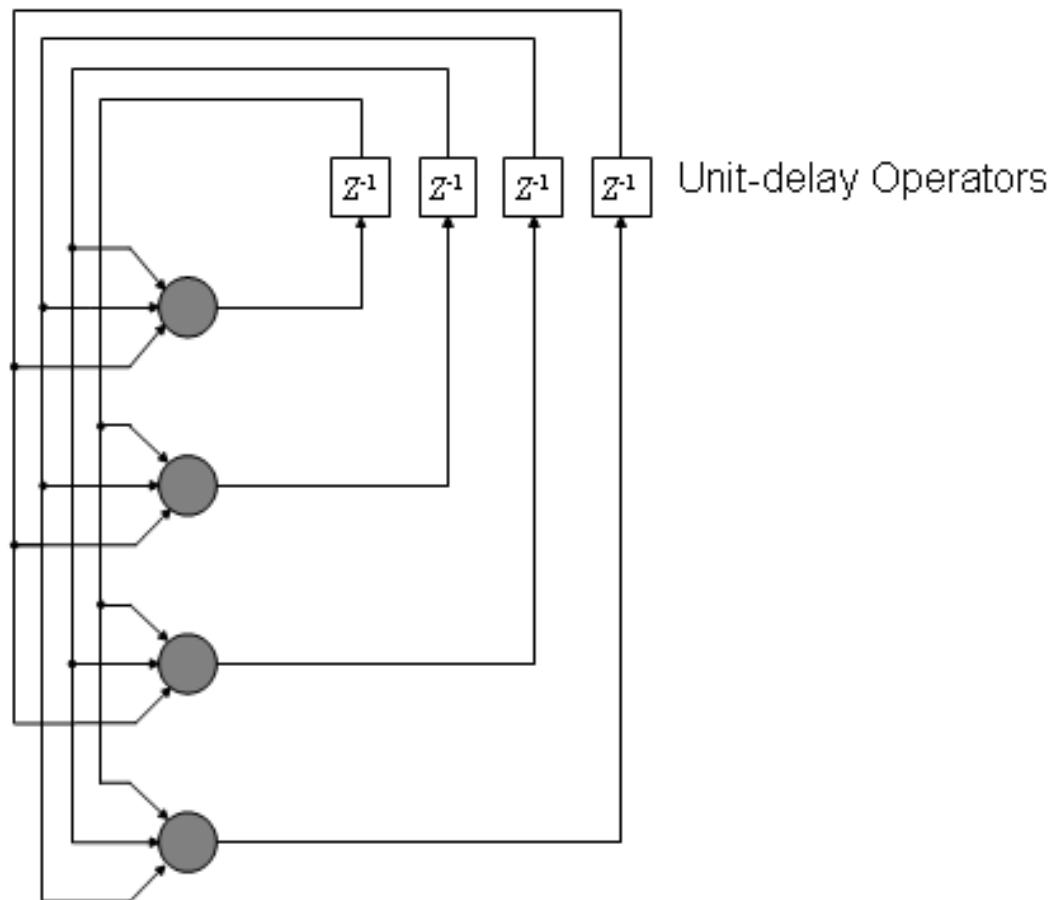
Output layer
of neurons



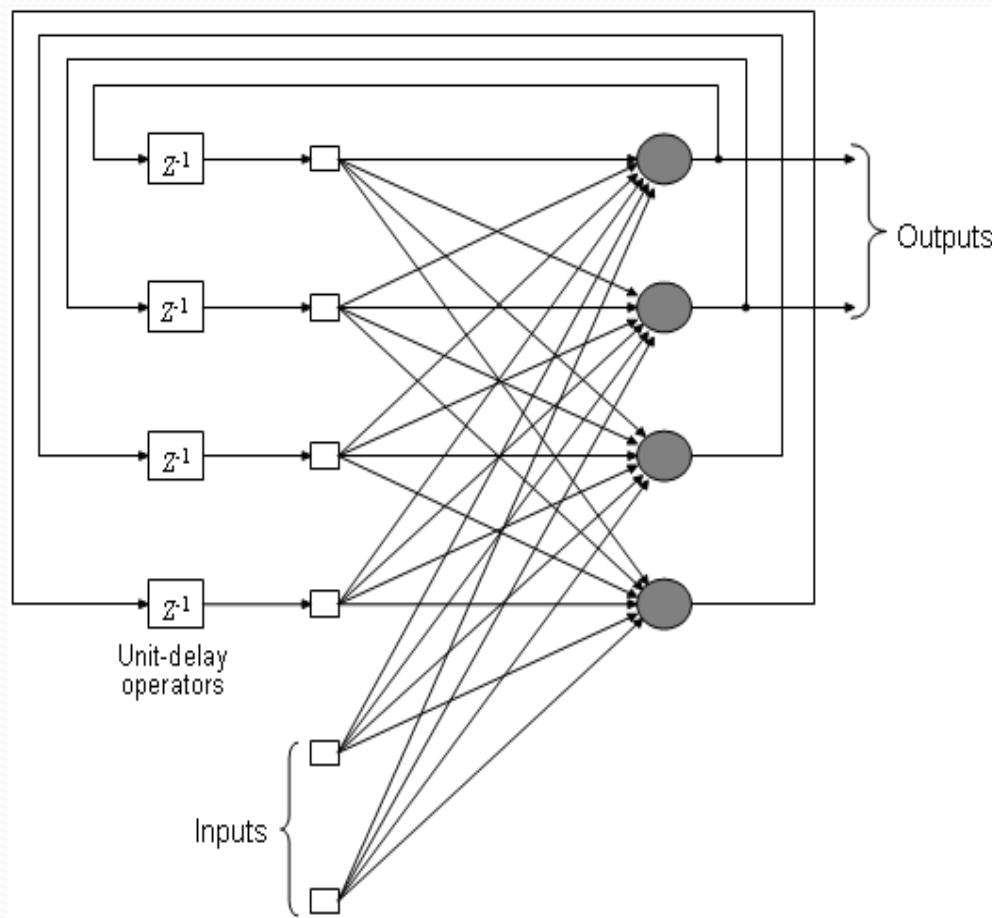
Multi-Layer Feedforward Networks



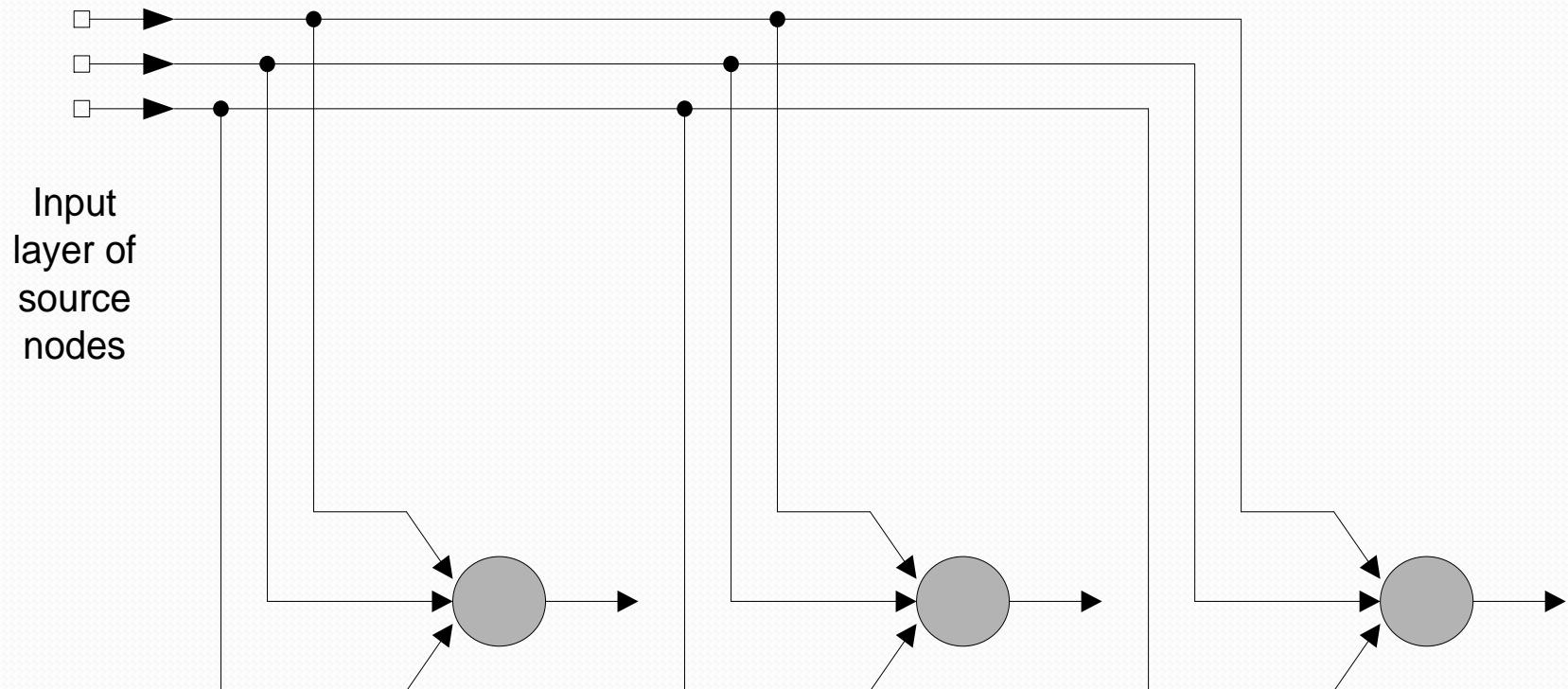
Recurrent Networks (tanpa hidden neurons)



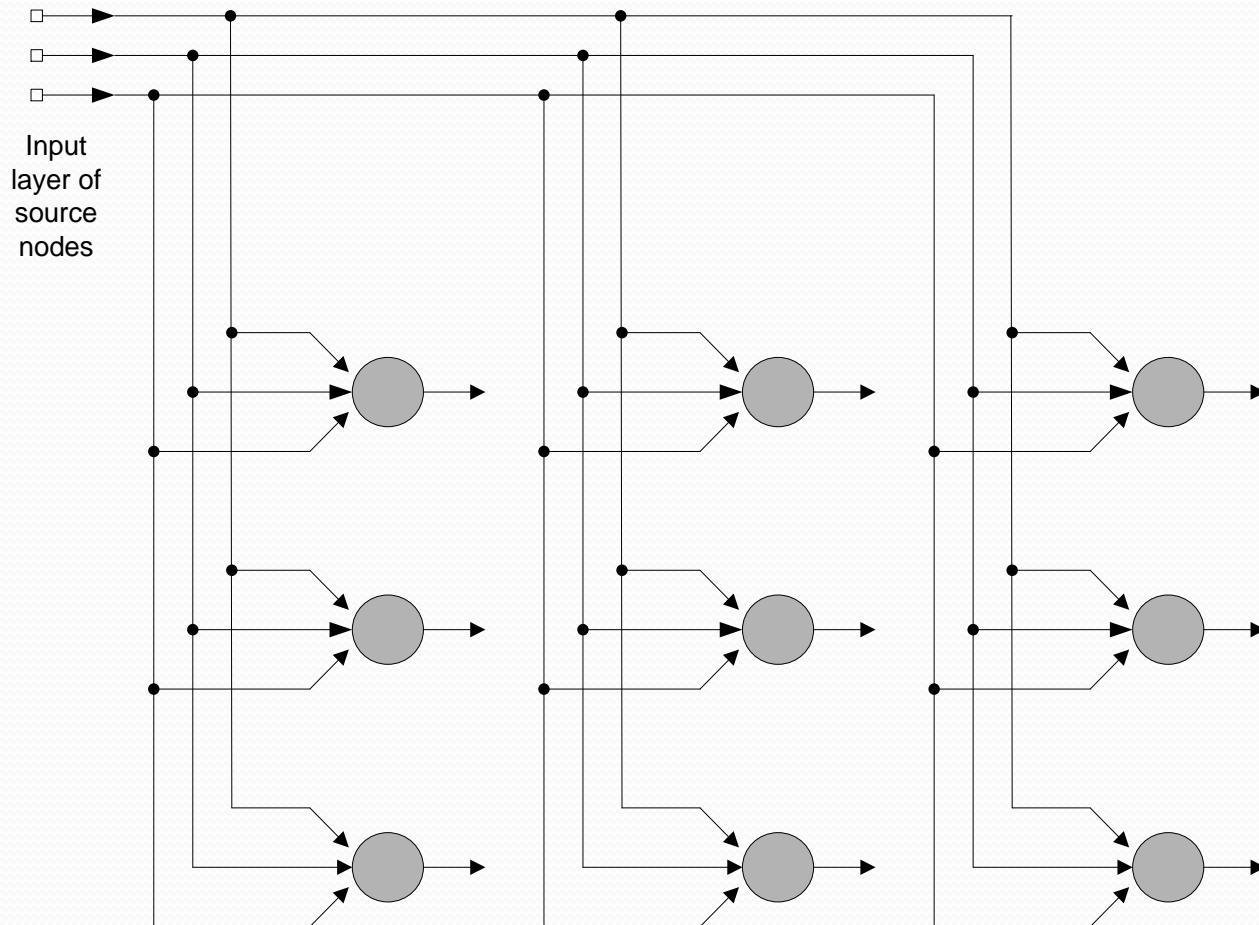
Recurrent Networks (dengan *hidden neurons*)



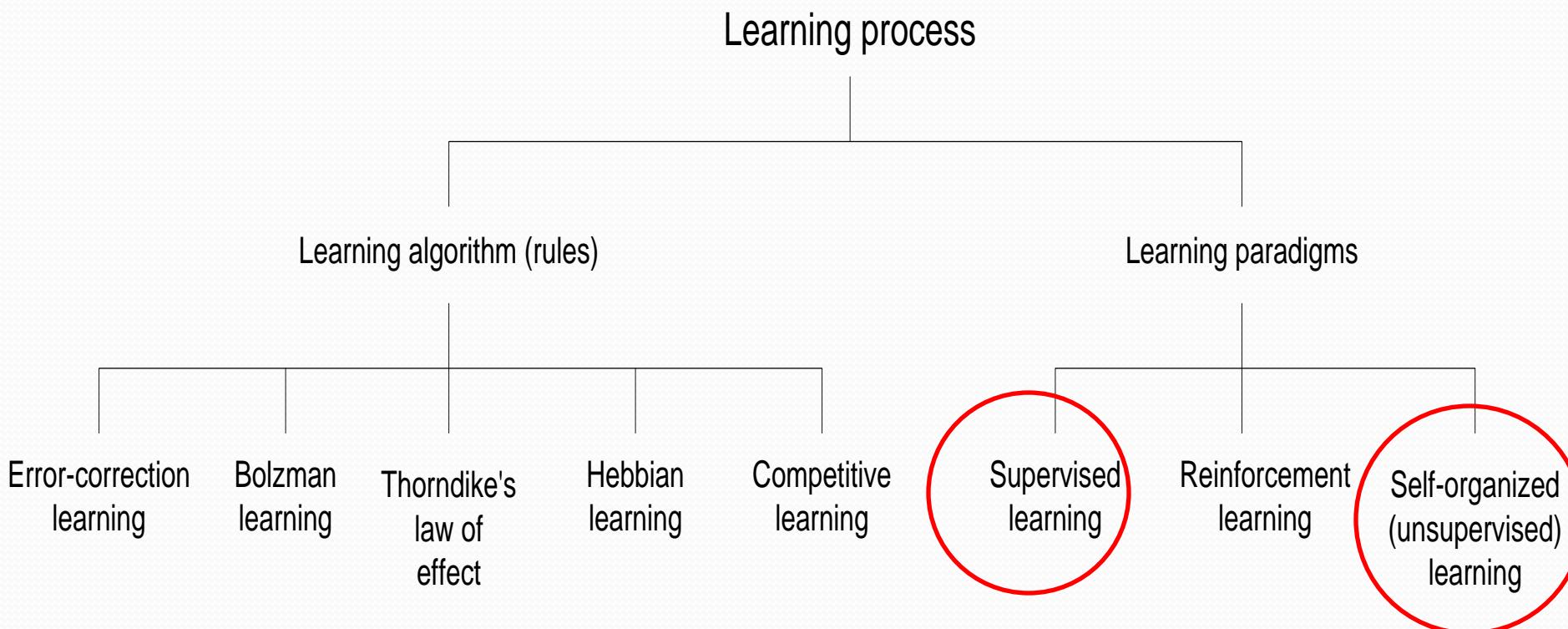
Lattice Structure (satu dimensi, 3 neurons)



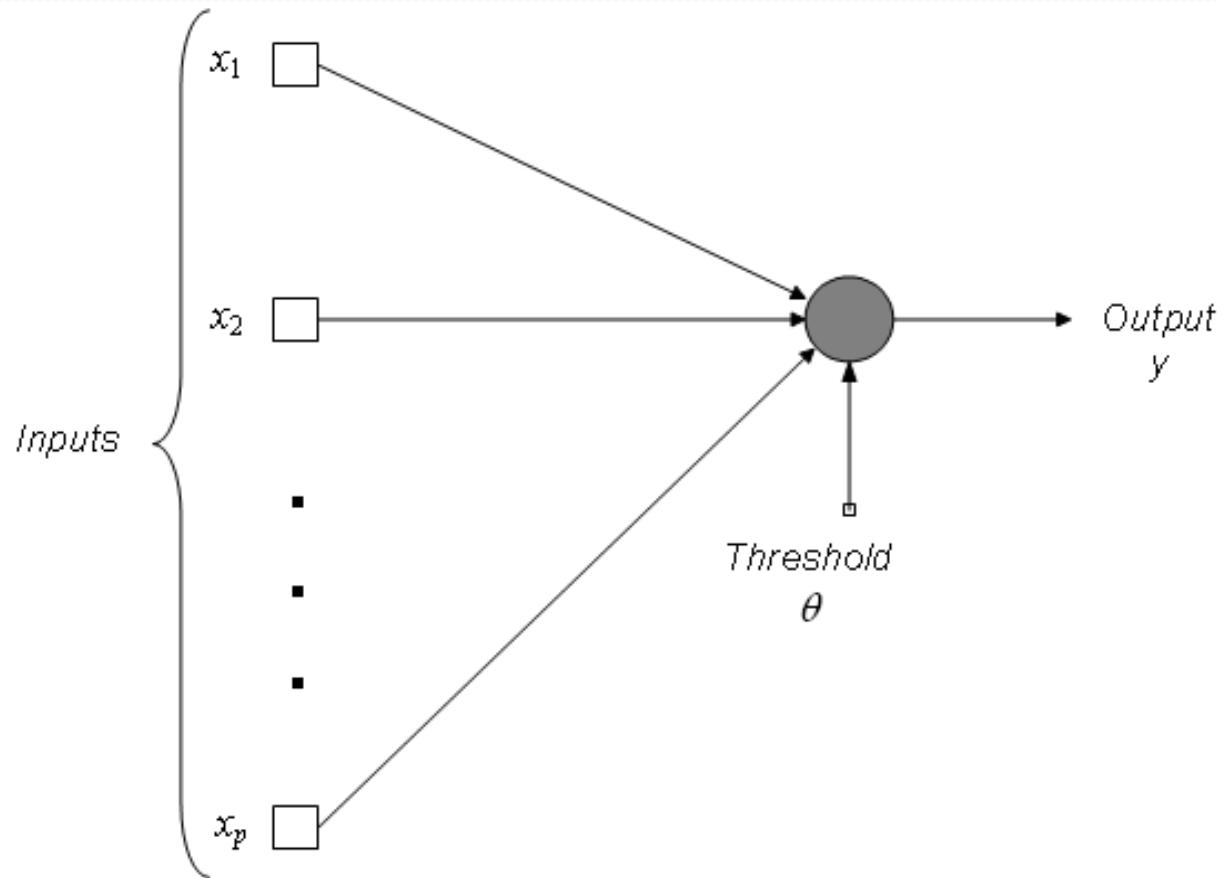
Lattice Structure (dua dimensi, 3x3 neurons)



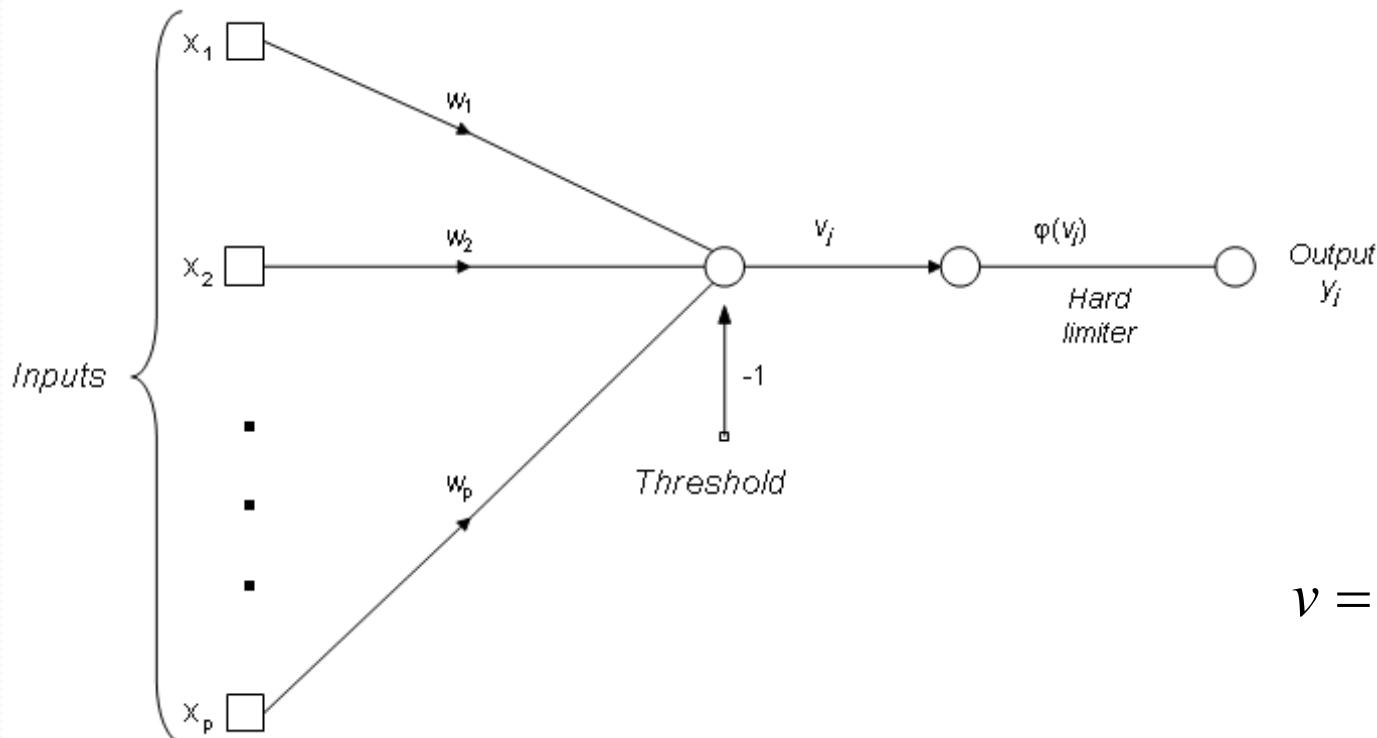
Proses Belajar (*Learning*)



Perceptron: Model

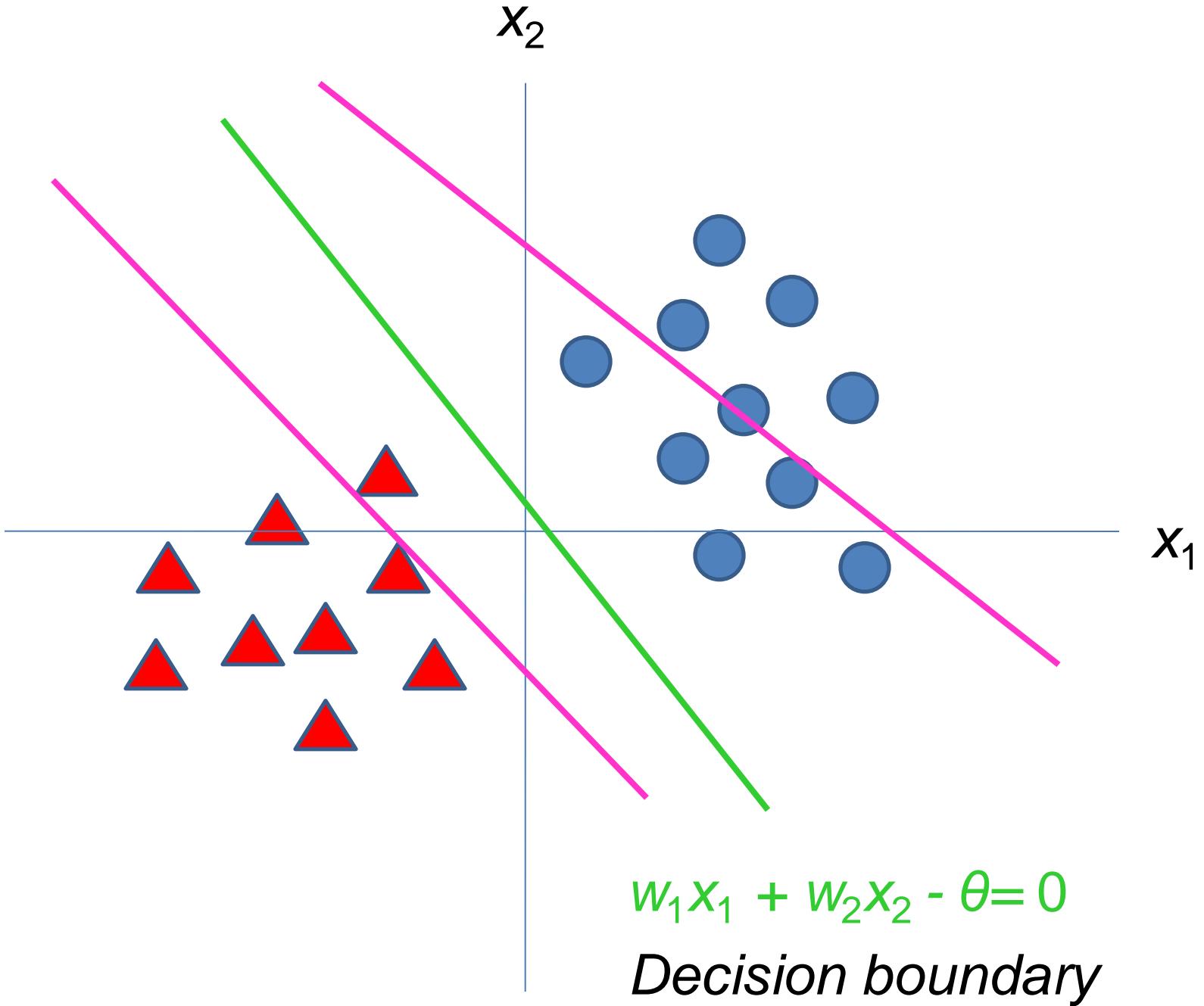


Perceptron: Signal-Flow Graph

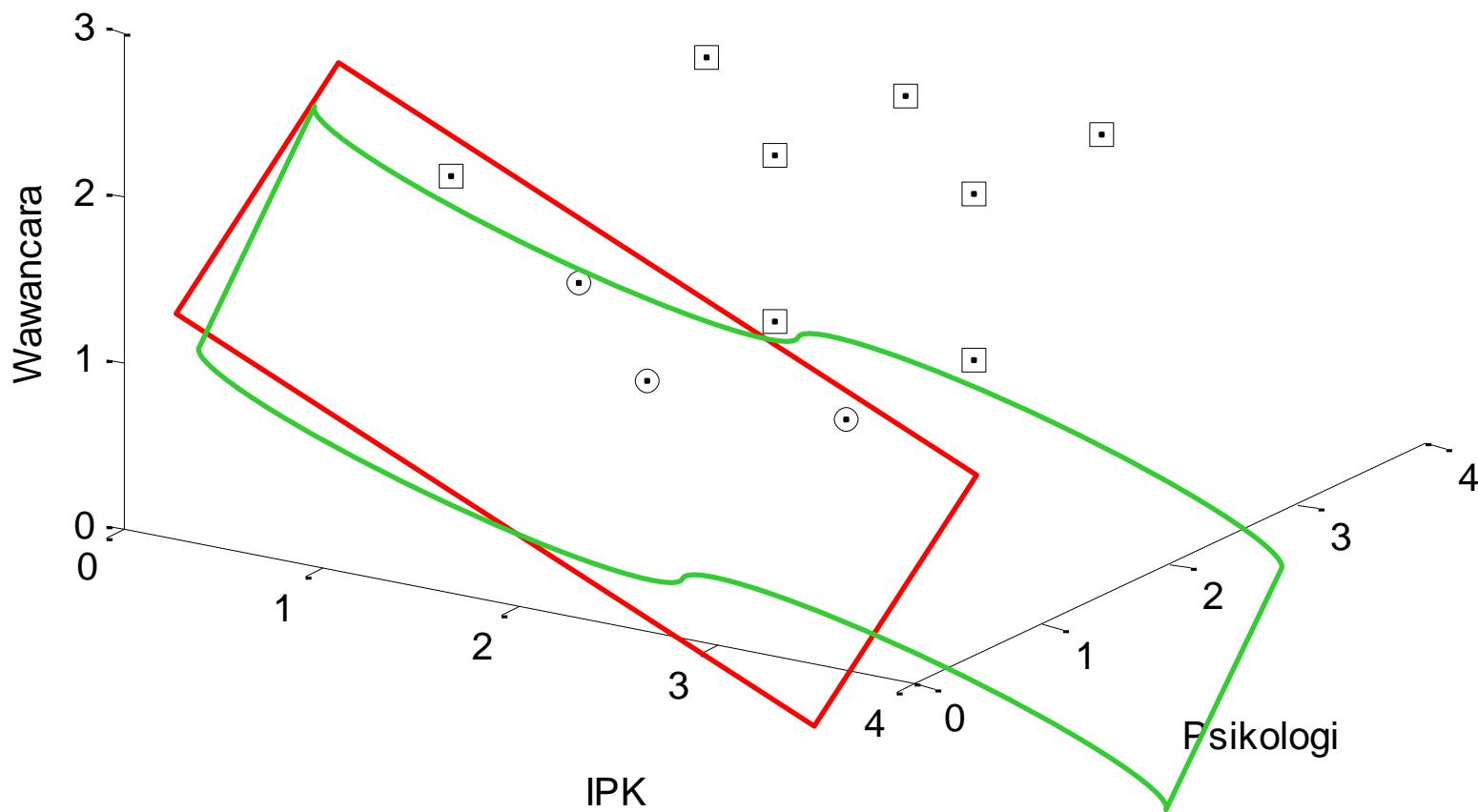


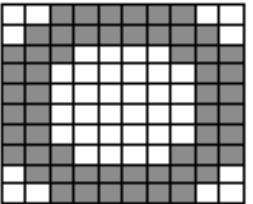
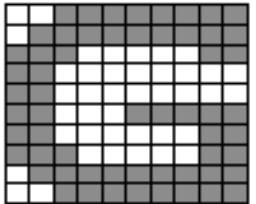
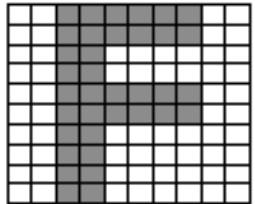
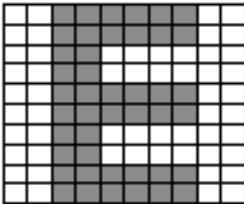
$$v = \sum_{i=1}^p w_i x_i - \theta$$

Decision boundary $\rightarrow \sum_{i=1}^p w_i x_i - \theta = 0$

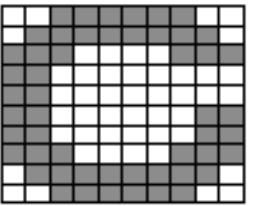
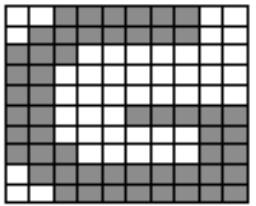
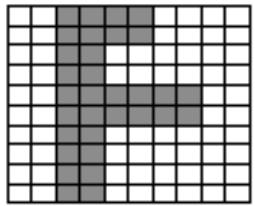
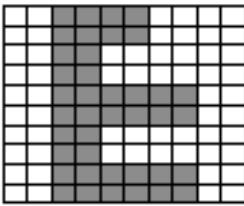


Pelamar	IPK	Psikologi	Wawancara	Diterima
P1	3	3	2	1
P2	3	2	2	1
P3	3	2	1	1
P4	3	1	1	0
P5	2	3	2	1
P6	2	2	2	1
P7	2	2	1	1
P8	2	1	1	0
P9	1	3	2	1
P10	1	2	1	0
P11	1	1	2	1

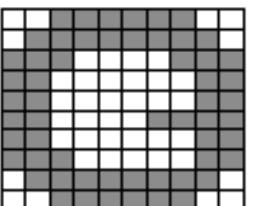
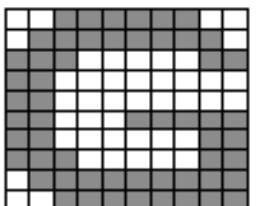
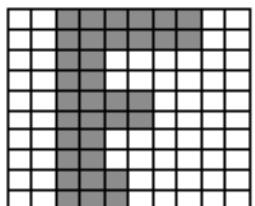
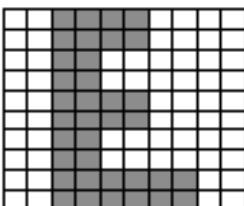
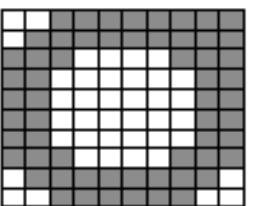
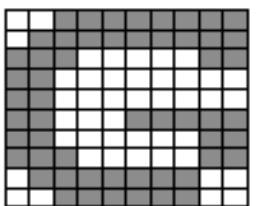
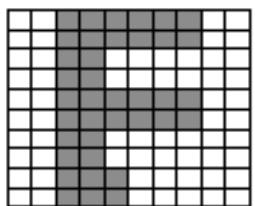
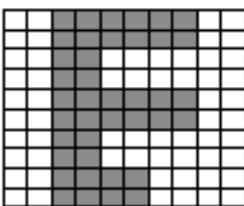
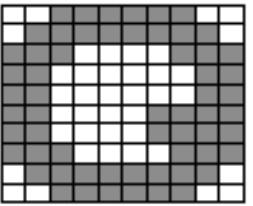
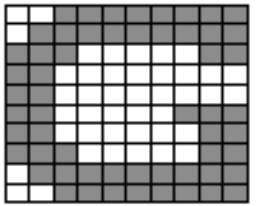
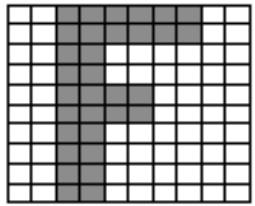
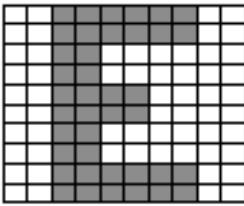




Visualisasi 100 dimensi?

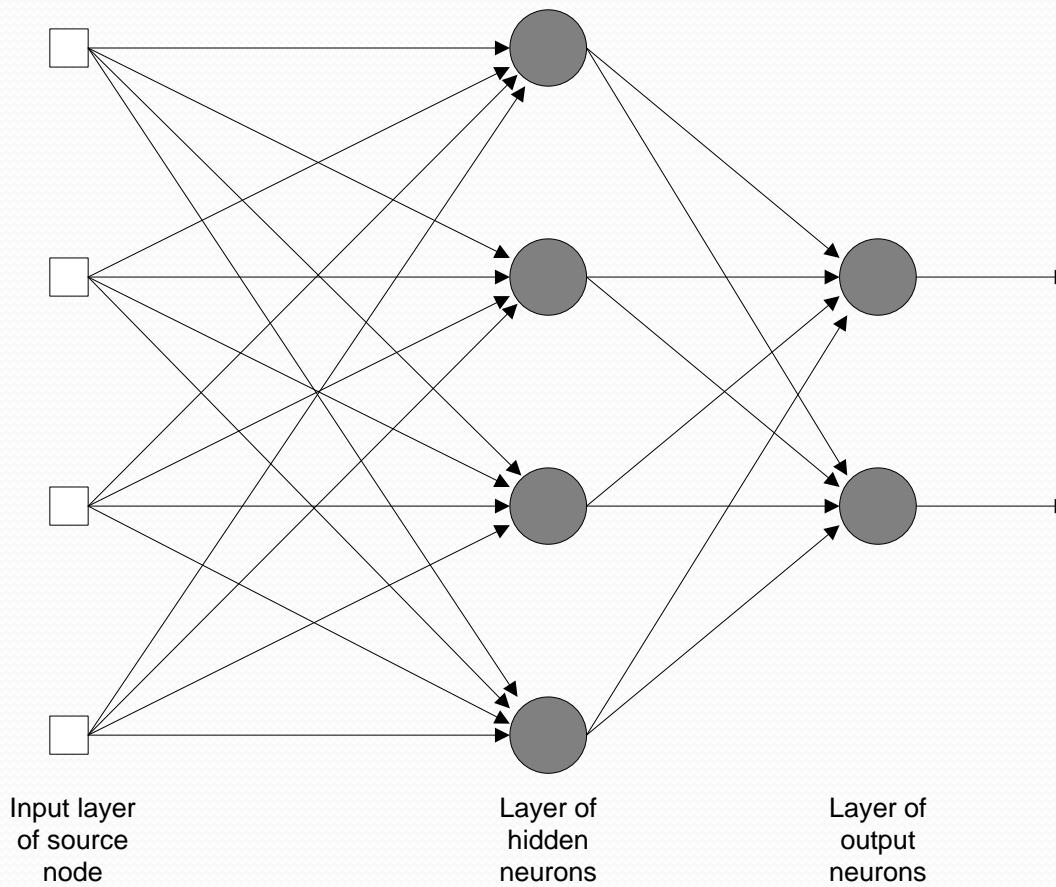


Bisa dengan Perceptron?



Pola	Pix 1	Pix 2	Pix 3	Pix 4	Pix 5	...	Pix 100
E1	0	0	1	1	1	...	0
F1	0	0	1	1	1	...	0
G1	0	1	1	1	1	...	1
O1	0	1	1	1	1	...	1
..							
O5	0	1	1	1	1	...	1

Multi-Layer Perceptron (MLP)



Algoritma Belajar Propagasi Balik

- **Definisikan masalah**
 - Matriks pola masukan (P)
 - matriks target (T)
- **Inisialisasi parameter jaringan**
 - Arsitektur jaringan (misalkan I-H-O)
 - *Synaptic weights* → acak (atau dengan metode tertentu)
 - *Learning rate (lr)* → laju belajar
 - *Threshold MSE* → untuk menghentikan *learning*

Algoritma Belajar Propagasi Balik

- Pelatihan Jaringan
 - Perhitungan Maju

$$A1 = \frac{1}{1 + e^{(W1 * P + B1)}}$$

$$A2 = W2 * A1 + B2$$

$$E = T - A2$$

$$MSE = \frac{\sum E^2}{N}$$

Algoritma Belajar Propagasi Balik

- Pelatihan Jaringan
 - Perhitungan Mundur

$$D2 = (1 - A2^2) * E$$

$$D1 = (1 - A1^2) * (W2 * D2)$$

$$dW1 = dW1 + (lr * D1 * P)$$

$$dB1 = dB1 + (lr * D1)$$

$$dW2 = dW2 + (lr * D2 * P)$$

$$dB2 = dB2 + (lr * D2)$$

Algoritma Belajar Propagasi Balik

- Pelatihan Jaringan
 - Perhitungan Mundur

$$W1 = W1 + dW1$$

$$B1 = B1 + dB1$$

$$W2 = W2 + dW2$$

$$B2 = B2 + dB2$$

Algoritma Belajar Propagasi Balik

- Langkah-langkah di atas adalah untuk satu kali siklus pelatihan (satu *epoch*).
- Biasanya, pelatihan harus diulang-ulang lagi hingga jumlah siklus tertentu atau telah tercapai MSE yang diinginkan.
- Hasil akhir dari pelatihan jaringan adalah bobot-bobot W_1 , W_2 , B_1 dan B_2 .

Pengenalan Karakter E, F, G, O

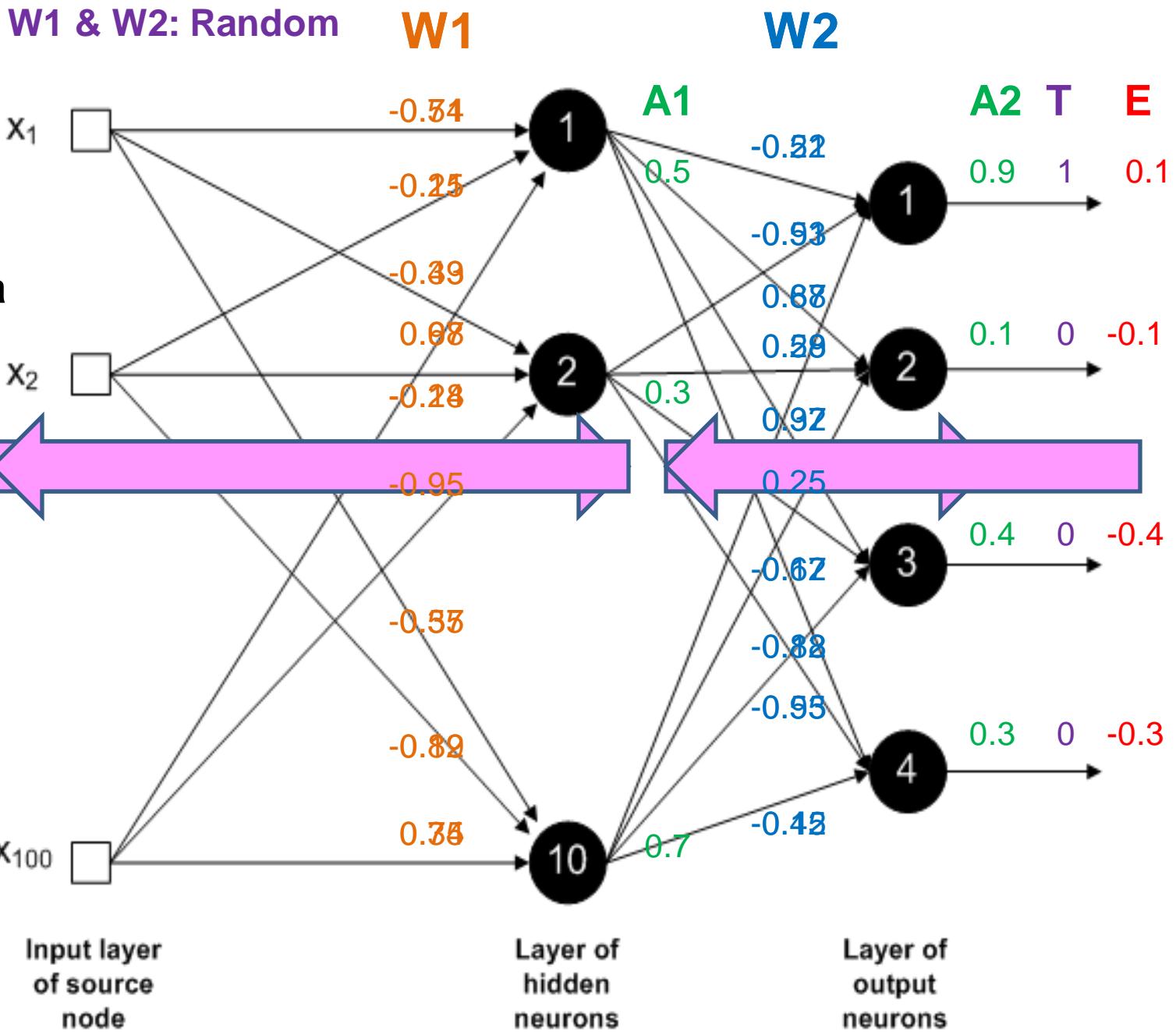
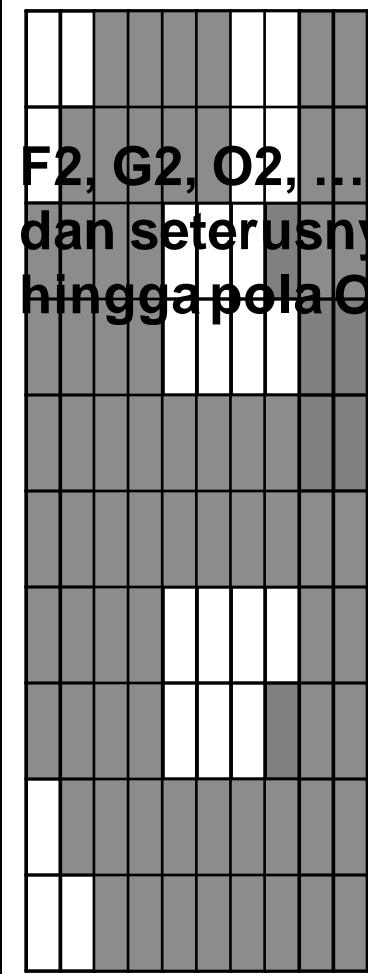
Matriks P

Pola	Pix 1	Pix 2	Pix 3	Pix 4	Pix 5	...	Pix 100
E1	0	0	1	1	1	...	0
F1	0	0	1	1	1	...	0
G1	0	1	1	1	1	...	1
O1	0	1	1	1	1	...	1
E2	0	0	1	1	1	...	0
...
O5	0	1	1	1	1	...	1

Matriks T

N1	N2	N3	N4	Kelas
1	0	0	0	E
0	1	0	0	F
0	0	1	0	G
0	0	0	1	O
1	0	0	0	E
...
0	0	0	1	O

Training



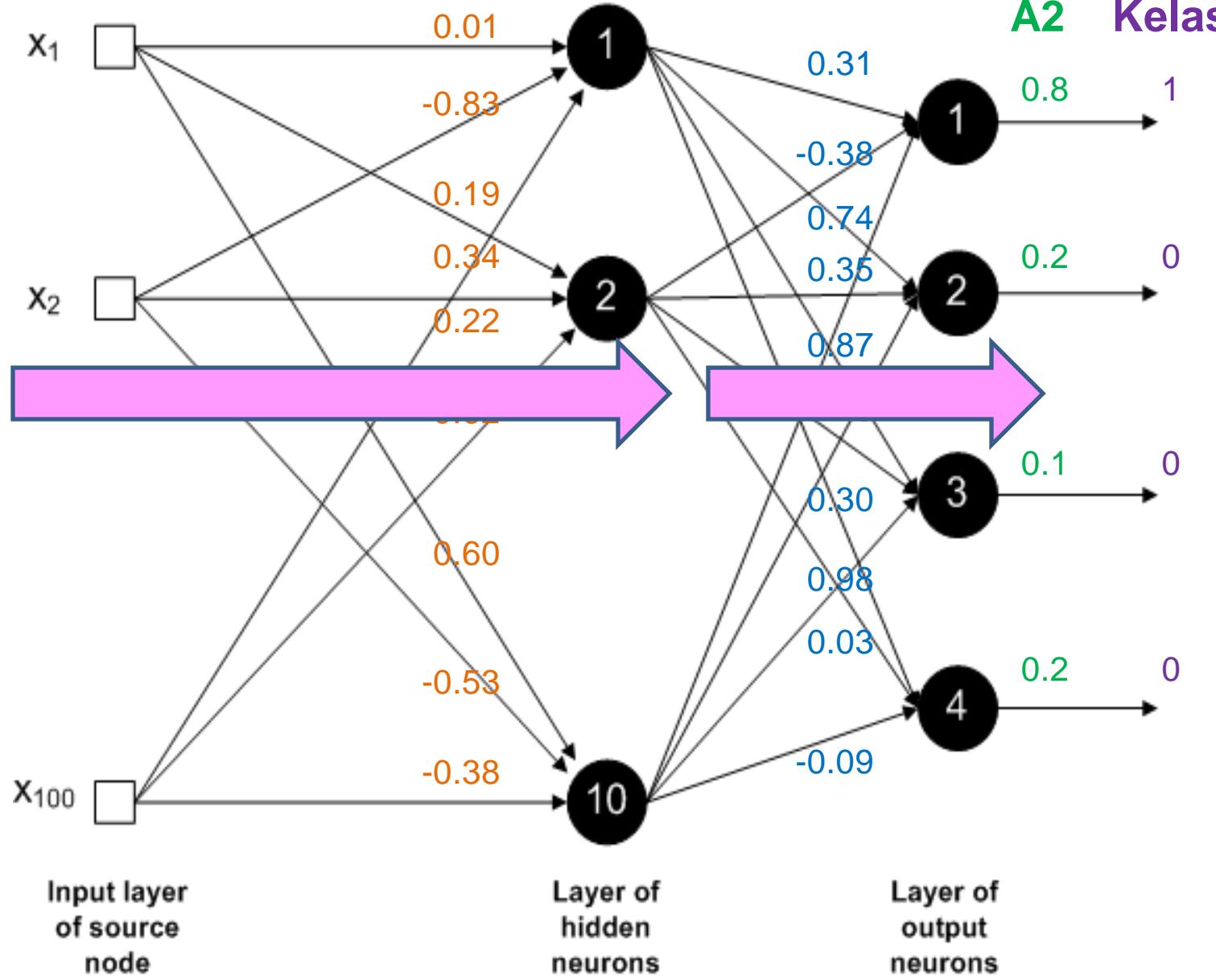
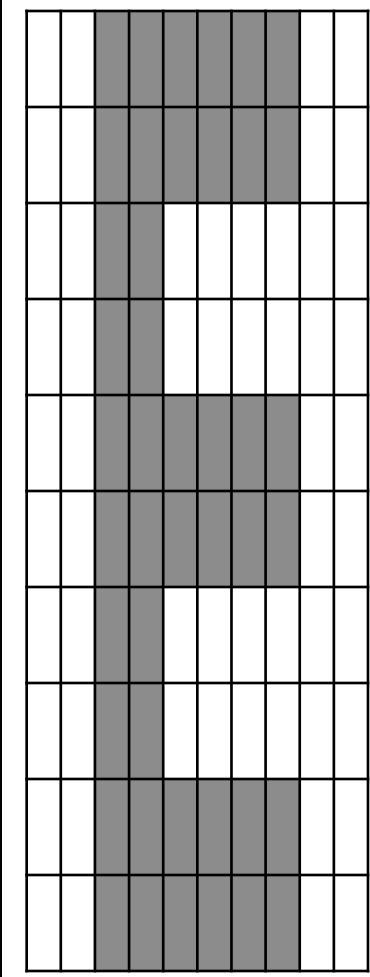
Testing

W1 & W2: Trained

W1

W2

A2 Kelas



Demo Program

- Pengenalan Karakter
- 4 kelas huruf: E, F, G, O
- Data Latih
 - 4 kelas, setiap kelas berisi 5 pola → 20 pola huruf
 - MLP = 100-10-4
 - Algoritma Learning Propagasi Balik

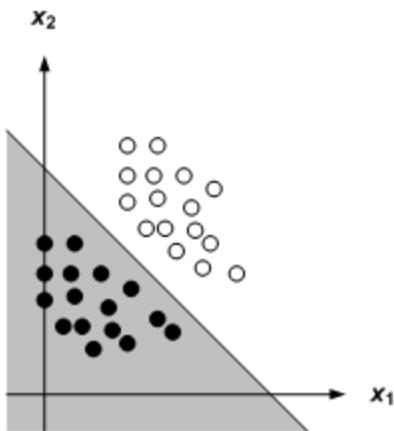
**Unlimited capacity?
We LEARN, not just store & retrieve !**

Updating synaptic weights

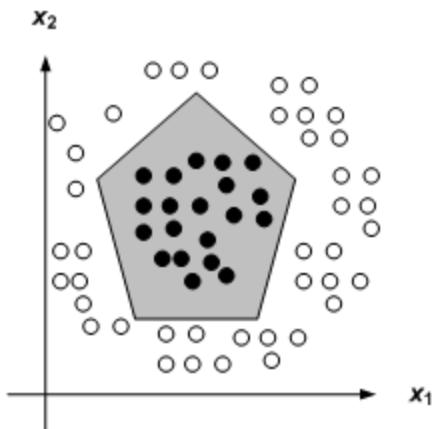
Permasalahan pada MLP

- Bagaimana struktur ANN yang optimal?
 - Jumlah *hidden layer*
 - Jumlah neuron pada hidden layer
 - Jumlah neuron pada output layer
 - Fungsi aktivasi yang optimal
- *Learning Rate*
- Kapan Menghentikan *Learning*

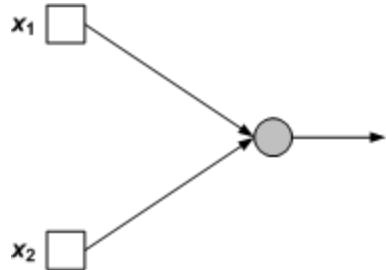
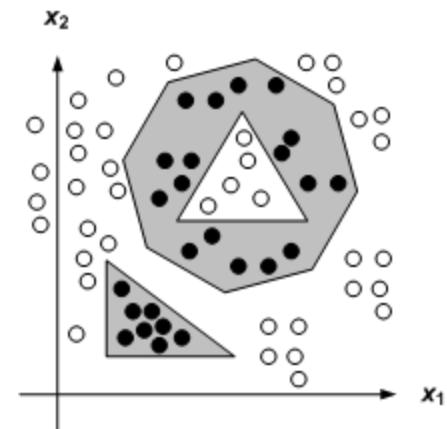
Klasifikasi *Linearly Separable*



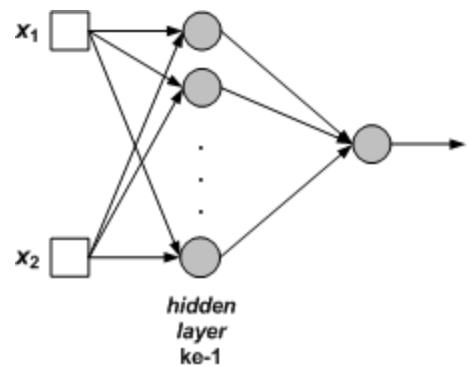
Klasifikasi yang Kompleks



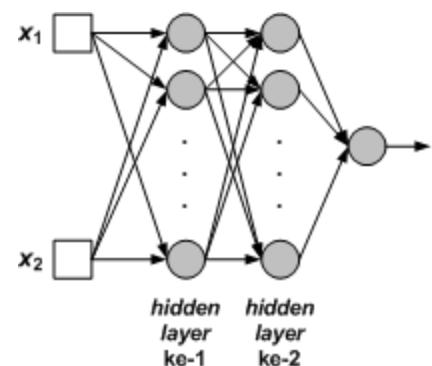
Klasifikasi Sangat Kompleks



Perceptron



MLP dengan 1 *hidden layer*



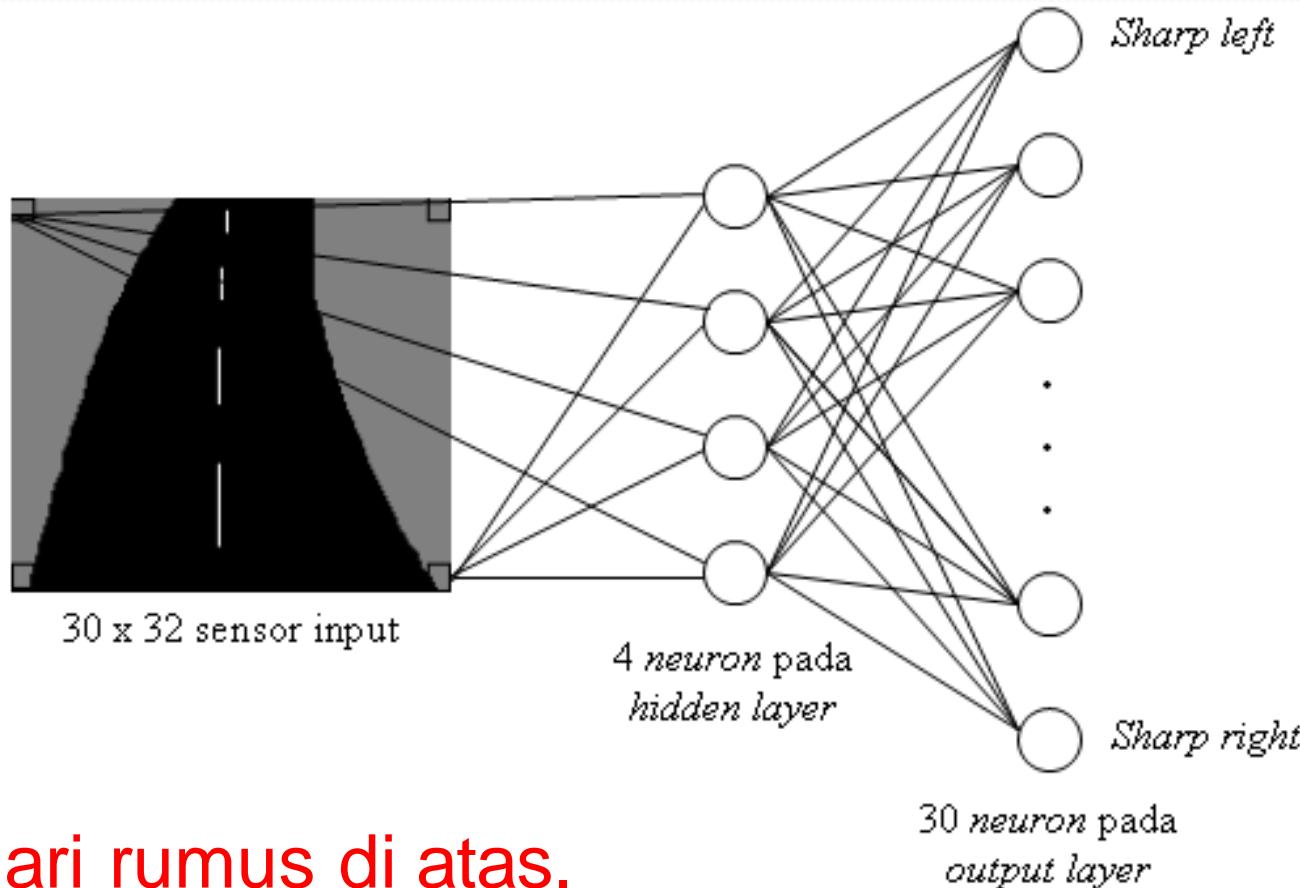
MLP dengan 2 *hidden layer*

Jumlah neuron pada hidden layer?

$$N_H = \sqrt{N_I N_O}$$

Perhatian: Rumus ini hanya perkiraan (tidak pasti).

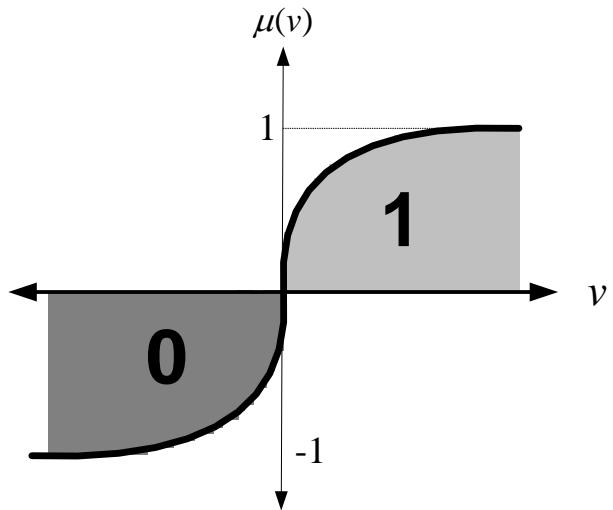
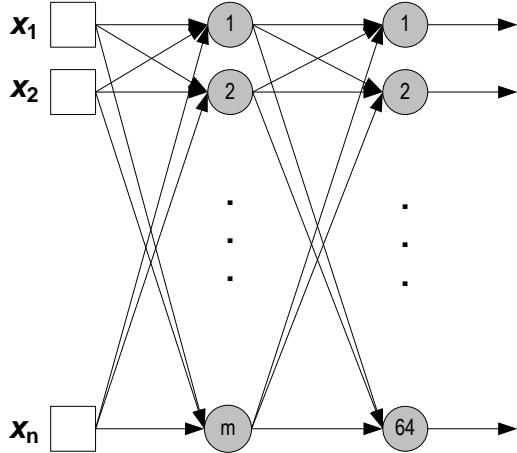
ALVINN: MLP 960-4-30



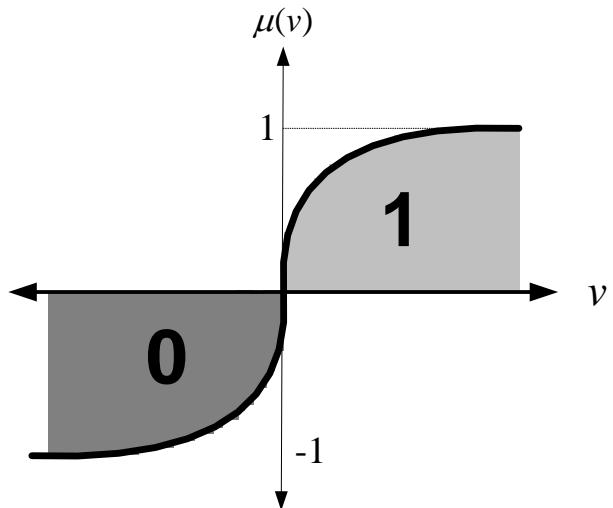
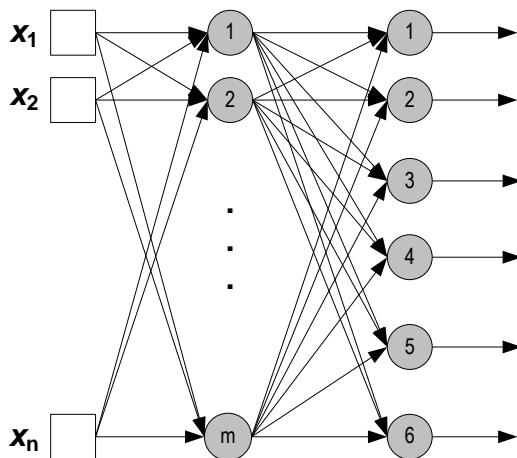
Jauh dari rumus di atas.

Jumlah neuron pada output layer

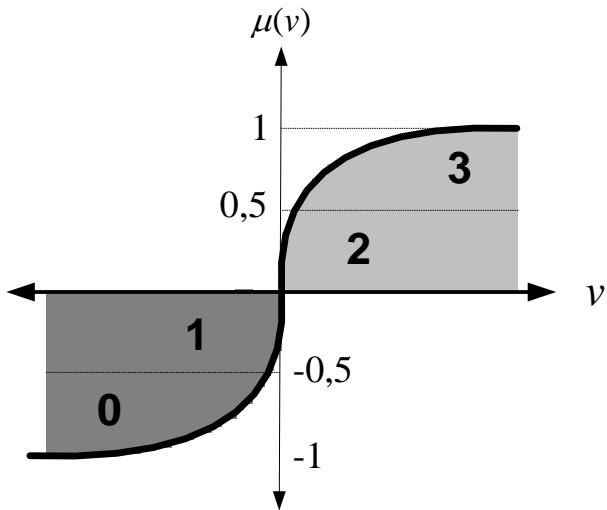
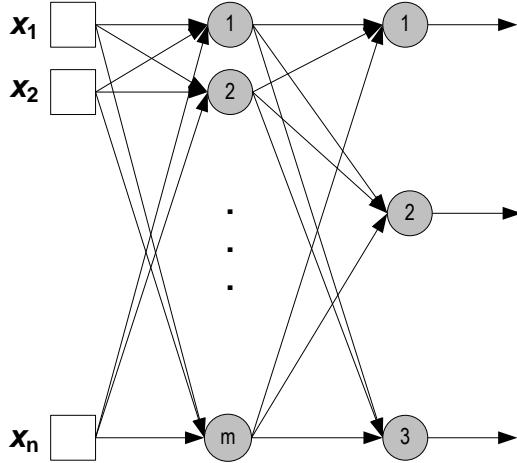
- **Deterministik:** mudah dihitung berdasarkan permasalahan yang dihadapi.
- Untuk pengenalan karakter dengan **64 kelas:** ('a', 'b', ..., 'z', 'A', 'B', ..., 'Z', 'o', 'i', ... '9', '-', '+'), perlu berapa output neuron?



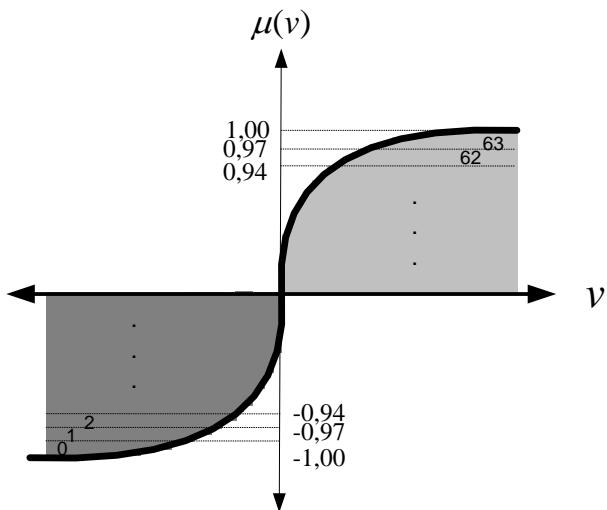
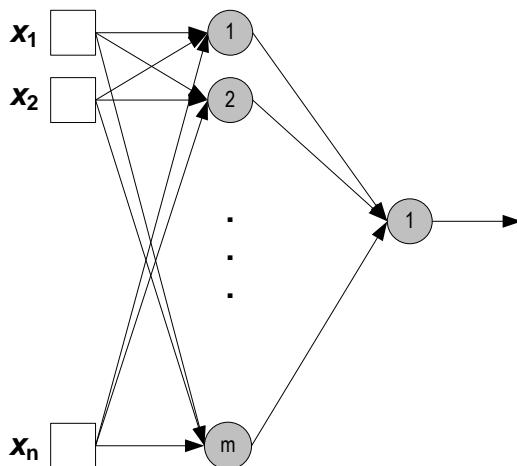
Output dari neuron ke-								Kelas
1	2	3	4	5	...	64		
1	0	0	0	0	...	0	a	
0	1	0	0	0	...	0	b	
0	0	1	0	0	...	0	c	
0	0	0	1	0	...	0	d	
0	0	0	0	1	...	0	e	
...	
0	0	0	0	0	...	1	+	



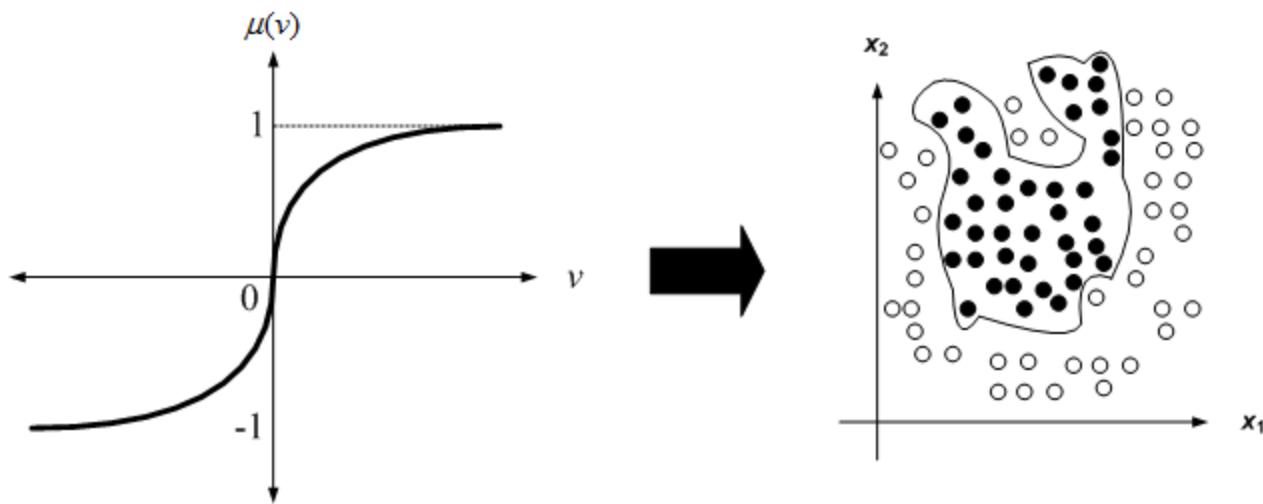
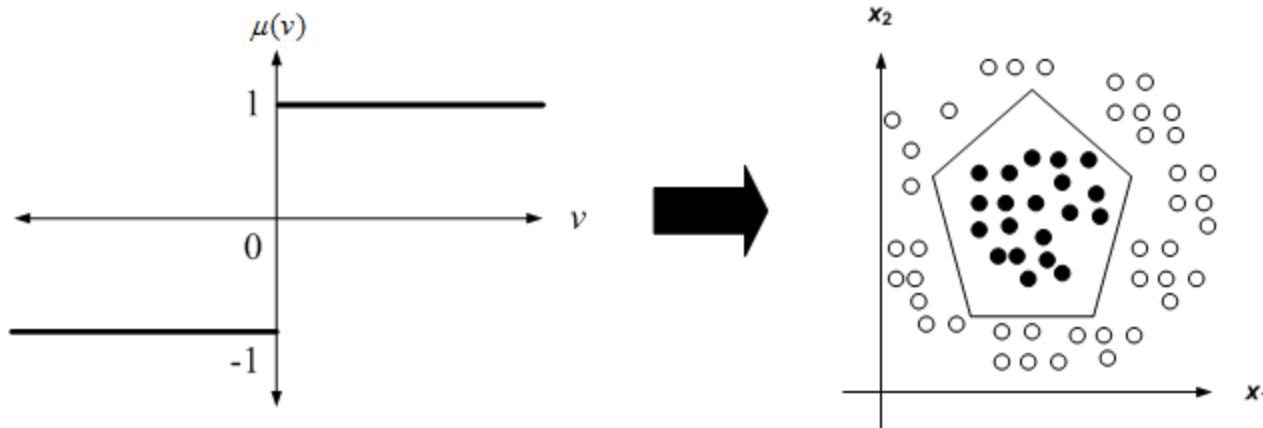
Output neuron ke-						Kelas
1	2	3	4	5	6	
0	0	0	0	0	0	a
0	0	0	0	0	1	b
0	0	0	0	1	0	c
0	0	0	0	1	1	d
0	0	0	1	0	0	e
...
1	1	1	1	1	1	+



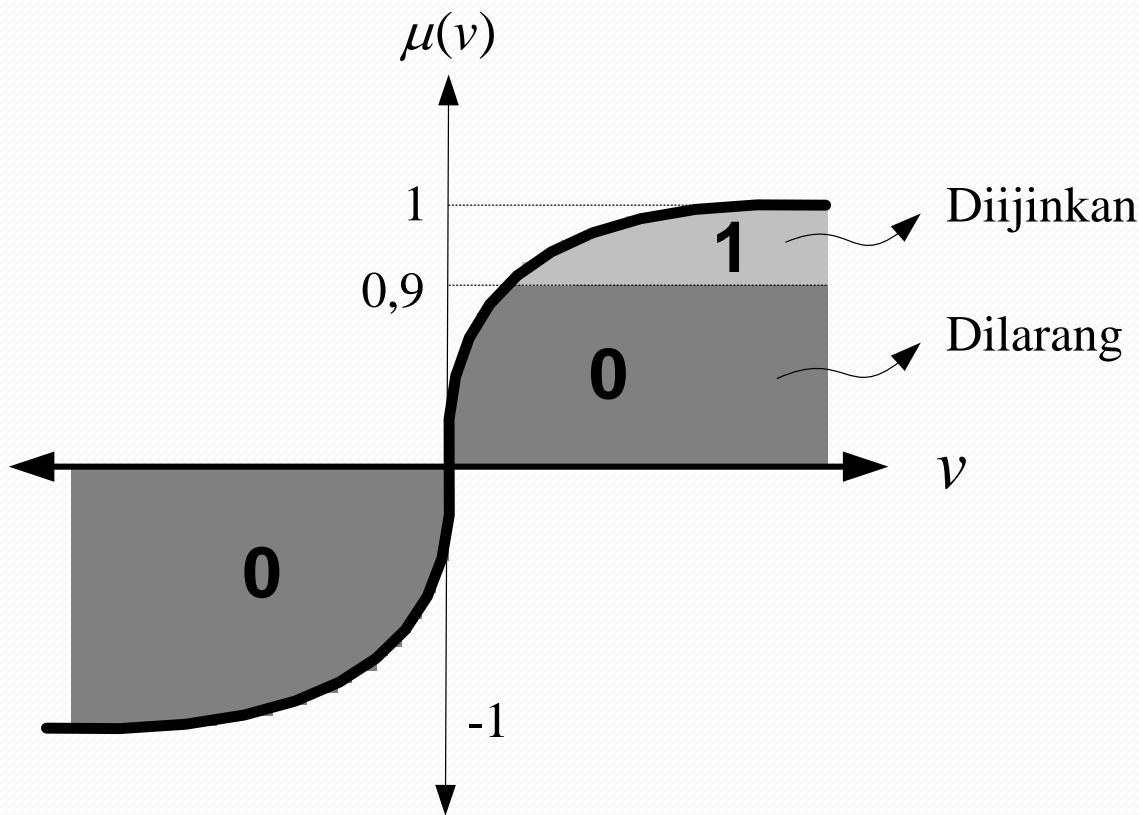
Output neuron ke-			Kelas
1	2	3	
0	0	0	a
0	0	1	b
0	0	2	c
0	0	3	d
0	1	0	e
...
3	3	3	+



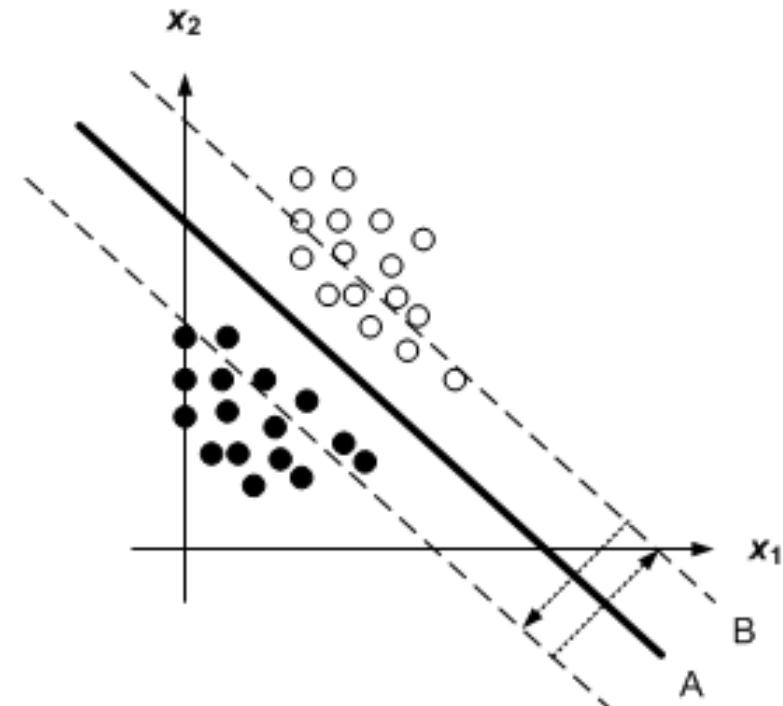
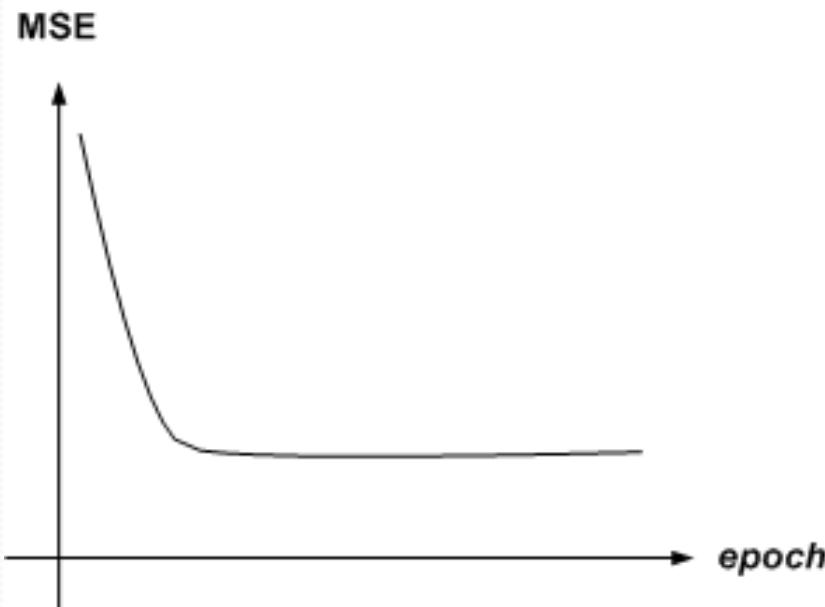
Output neuron ke-		Kelas
1	2	
0		a
1		b
2		c
3		d
4		e
...		...
63		+

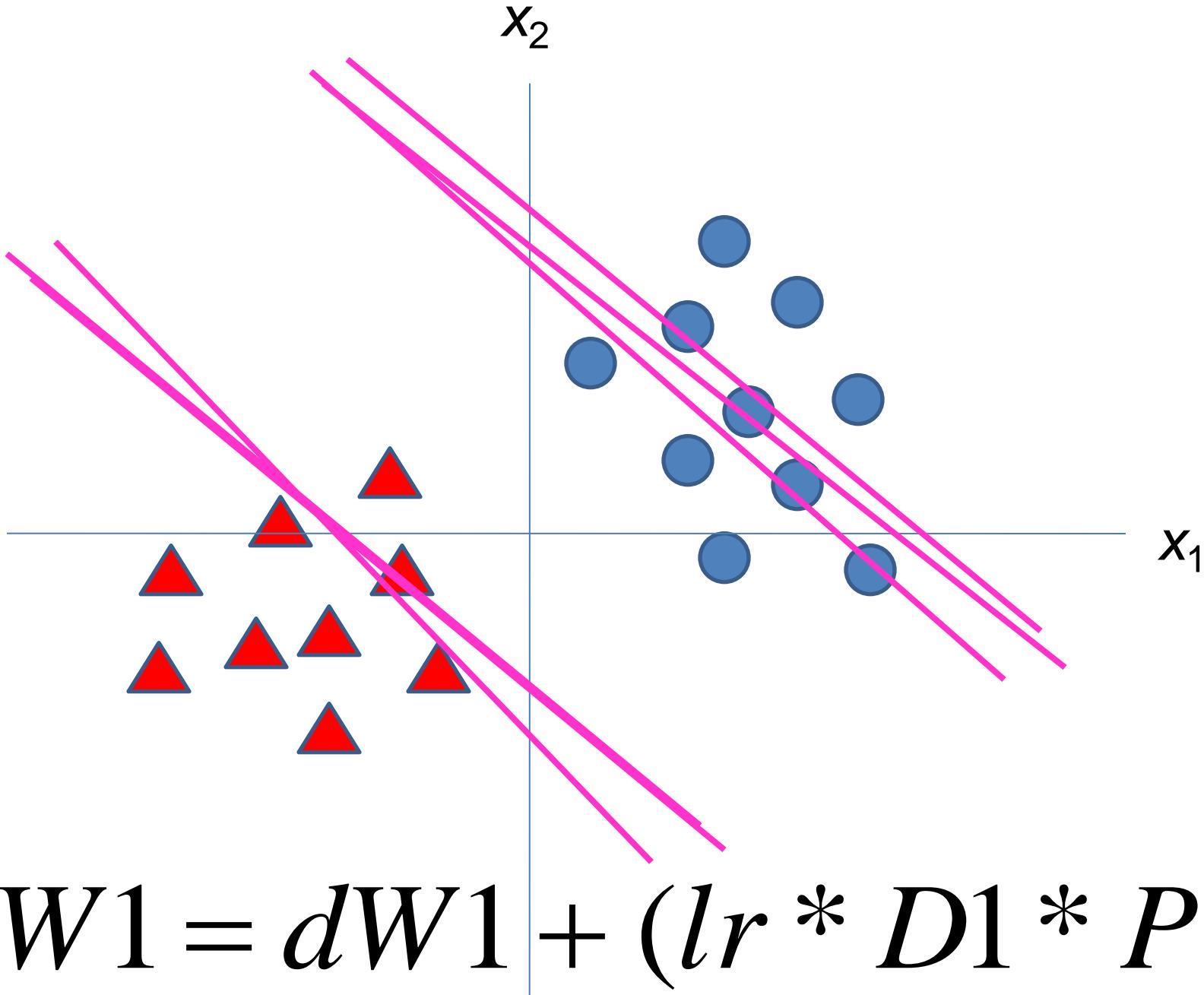


Security Systems



Learning Rate: Besar





$$dW1 = dW1 + (lr * D1 * P)$$

Algoritma Belajar Propagasi Balik

- Pelatihan Jaringan
 - Perhitungan Mundur

$$D2 = (1 - A2^2) * E$$

$$D1 = (1 - A1^2) * (W2 * D2)$$

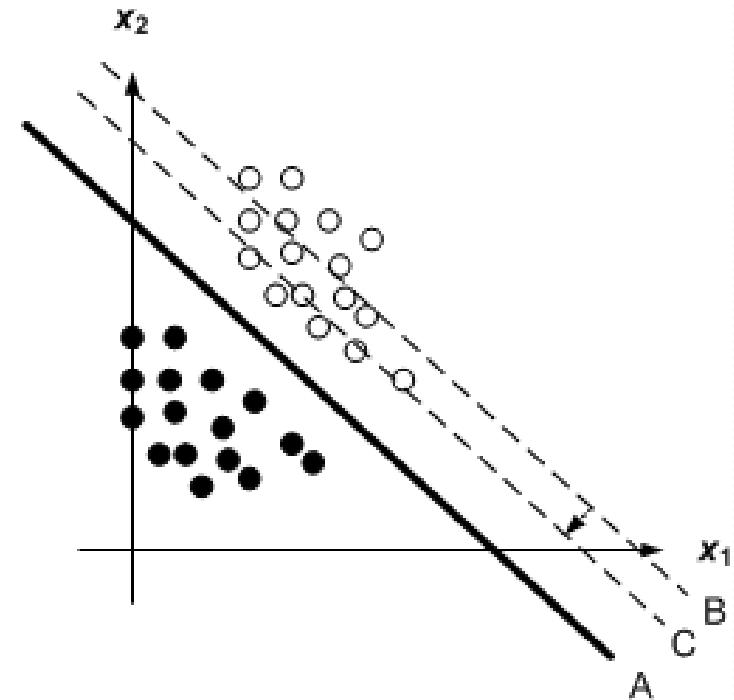
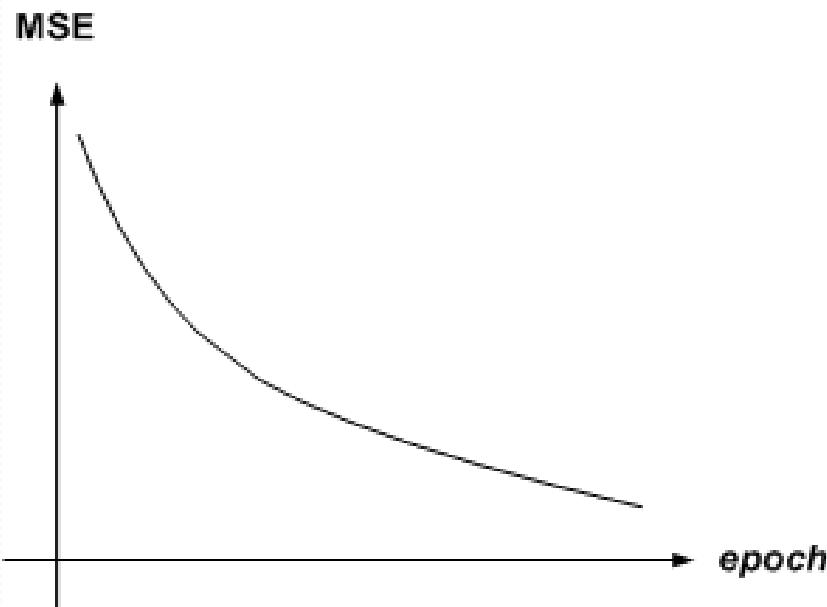
$$dW1 = dW1 + (lr * D1 * P)$$

$$dB1 = dB1 + (lr * D1)$$

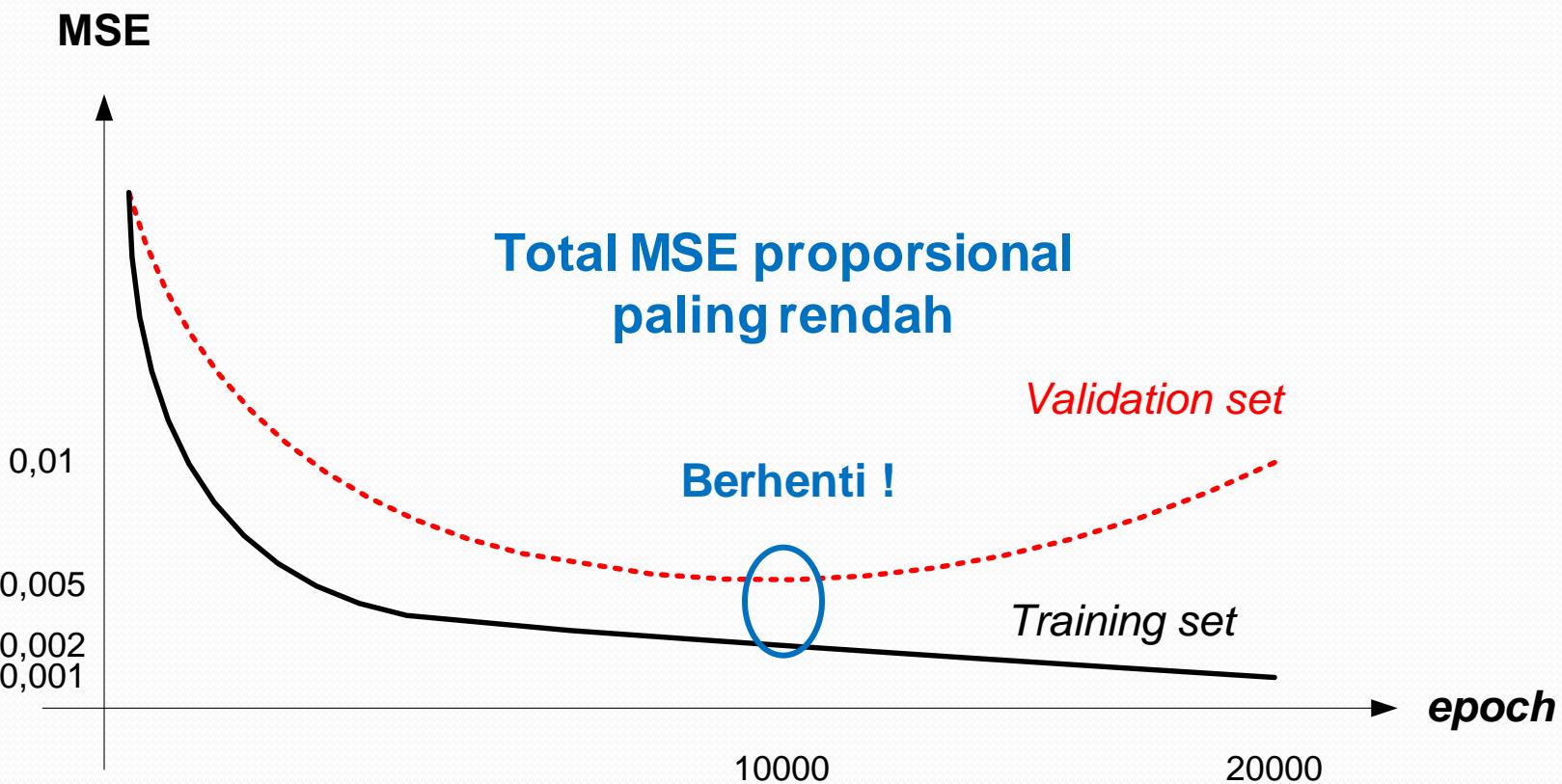
$$dW2 = dW2 + (lr * D2 * P)$$

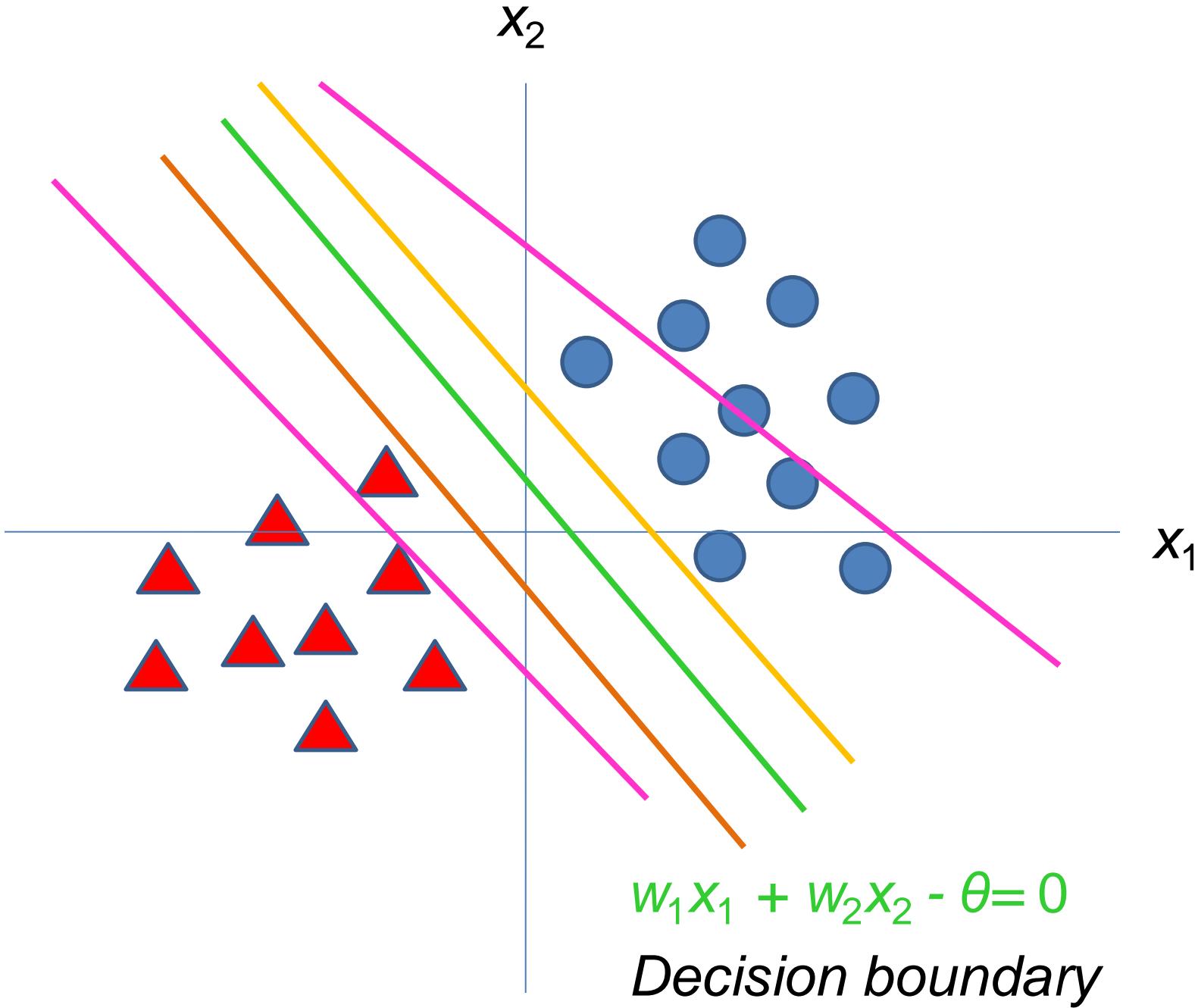
$$dB2 = dB2 + (lr * D2)$$

Learning Rate: Kecil

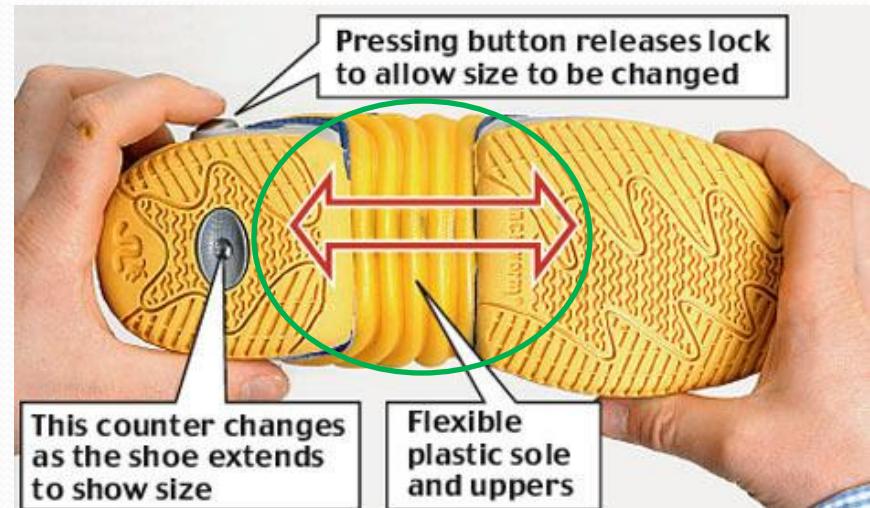


Kapan Menghentikan *Learning*?





Overfit, Oversize, Flexible



Strategi penggunaan ANN

- **Cara memandang masalah:**
 - Klasifikasi
 - Sekuriti
 - Prediksi
 - Optimasi
- **Teknik learning:** Supervised/Unsupervised
- **Desain Arsitektur**
 - Jumlah layer
 - Jumlah neuron
 - Pemetaan output
- **Strategi learning**
 - Penyiapan data: filterisasi data, pembagian data (training, validasi, test)
 - Parameter: inisialisasi (acak atau memakai algoritma), laju belajar, dsb.
 - Penghentian learning

Studi Kasus

- Klasifikasi
- Verifikasi
- Sekuriti
- Prediksi
- Deteksi
- Optimasi

Kasus 1 Verifikasi tandatangan

- OFFLINE atau ONLINE?
 - Citra: 100 x 100 pixel grayscale
 - Satu juta tandatangan?
-
- Input & Output?
 - P dan T?
 - Struktur dan parameter MLP?
 - Perlu *preprocessing*?



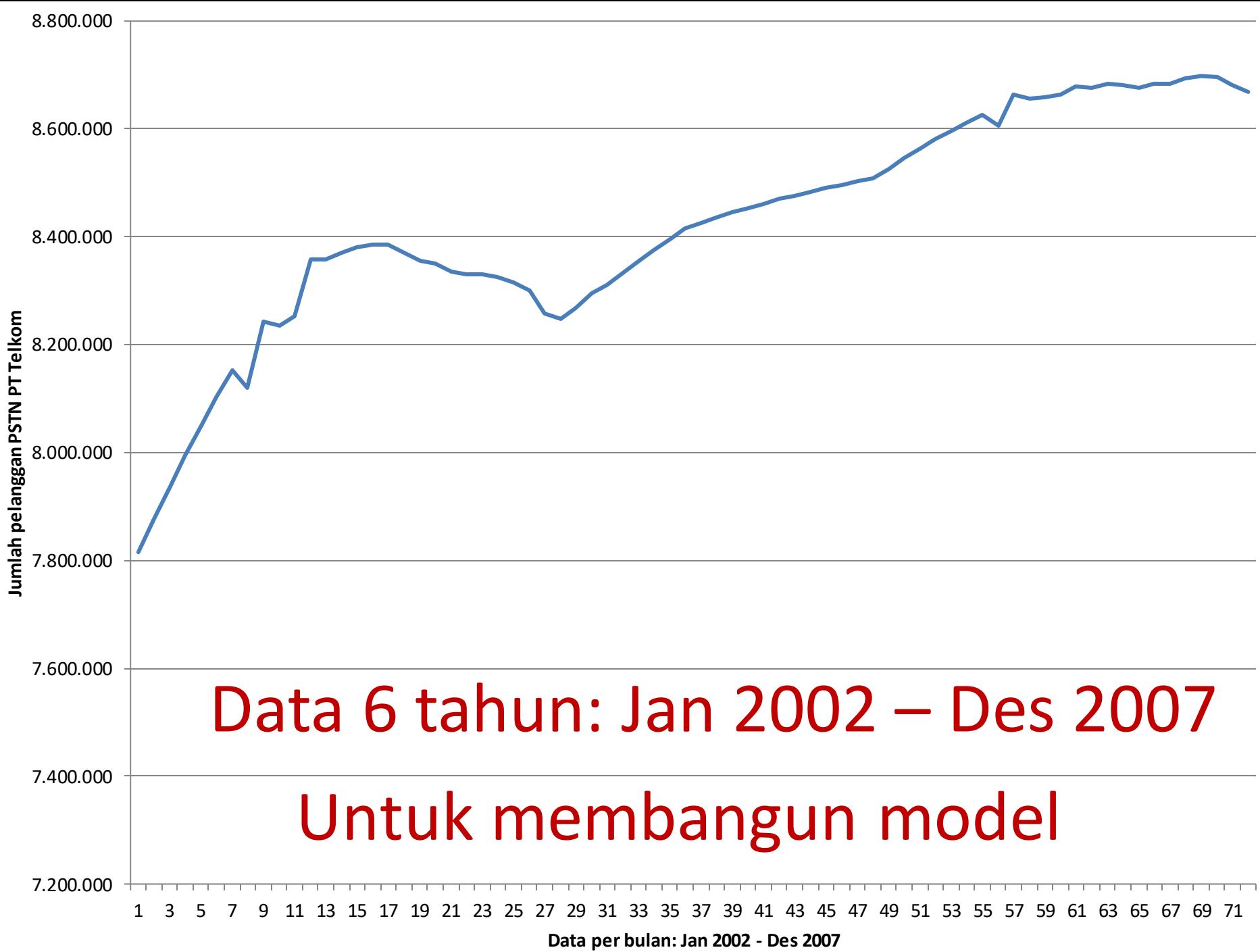
Kasus 2: Sistem keamanan

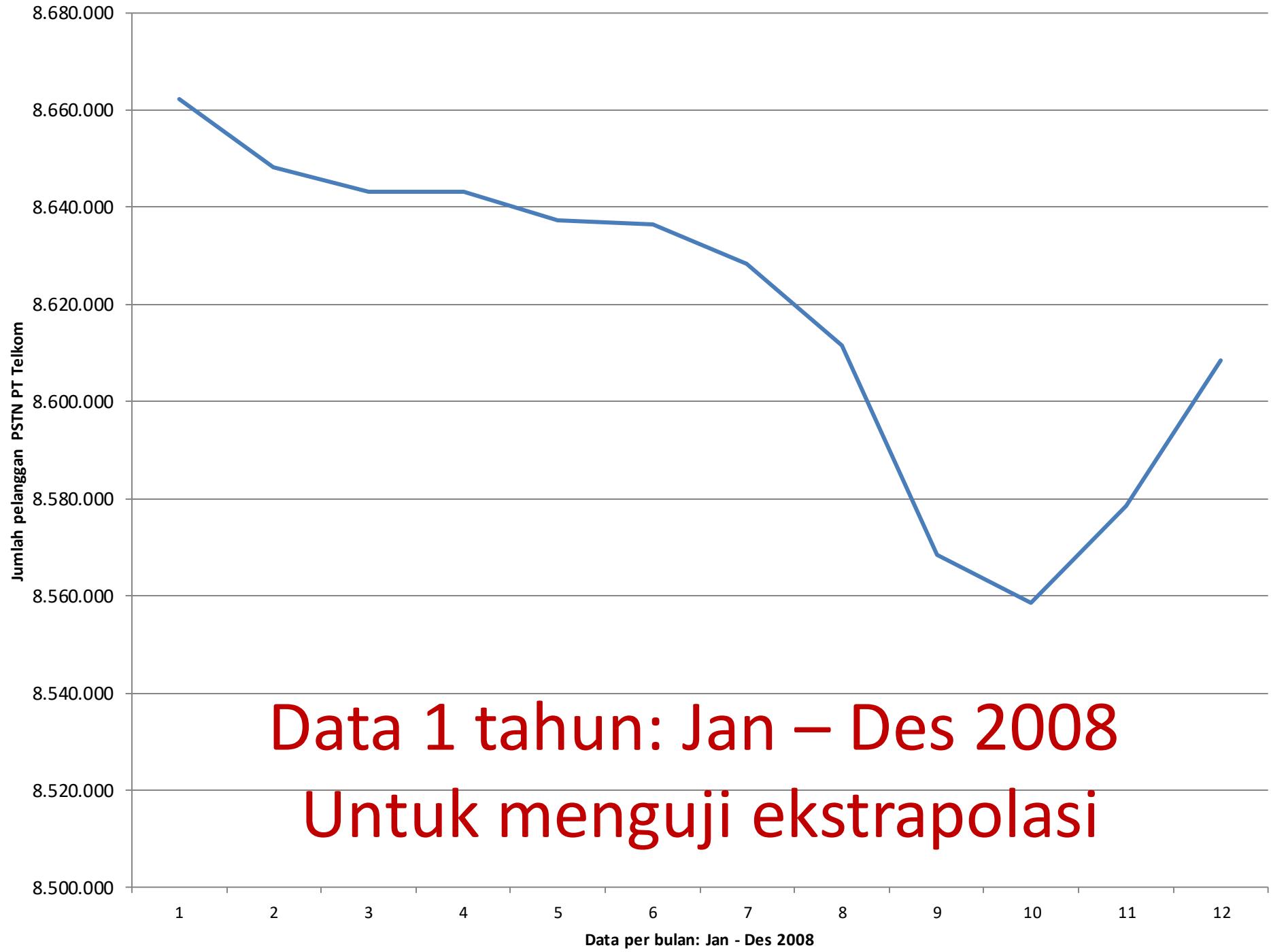
- Satu ruangan hanya 10 orang yang boleh masuk
 - Setiap orang yang akan masuk ruangan harus menempelkan ibu jari untuk diverifikasi sidik jarinya
 - Citra: 300 x 300 pixels
-
- Input & Output?
 - P dan T?
 - Struktur dan parameter MLP?
 - Perlu *preprocessing*?

Kasus 3: Prediksi pelanggan PSTN

- Data riil dari PT Telkom
 - Jumlah pelanggan bulanan selama 7 tahun
 - Error harus < 1 %
-
- Input & Output?
 - P dan T?
 - Struktur dan parameter MLP?
 - Perlu *preprocessing*?







Data 1 tahun: Jan – Des 2008
Untuk menguji ekstrapolasi

Pembangunan Model

- **Skenario 1**
 - Data *training* : 24 bulan (2002 - 2003)
 - Data *validation* : 24 bulan (2004 - 2005)
 - Data *testing* : 24 bulan (2006 - 2007)
- **Skenario 2**
 - Data *training* : 12 bulan (2002)
 - Data *validation* : 12 bulan (2003)
 - Data *testing* : 48 bulan (2004 - 2007)
- **Skenario 3**
 - Data *training* : 48 bulan (2002 - 2005)
 - Data *validation* : 12 bulan (2006)
 - Data *testing* : 12 bulan (2007)

Formulasi Masalah

- Input & Output?
- P dan T?
- Struktur dan parameter MLP?
- Perlu *preprocessing*?

Normalisasi

$$Xn_i = \left(\frac{(X_i - \min(X))}{(\max(X) - \min(X))} \times 0,8 \right) + 0,1$$

- Xn_i = data aktual normalisasi ke-i
- X_i = data aktual dengan range data asli ke-i
- X = data aktual dengan range data asli

Denormalisasi

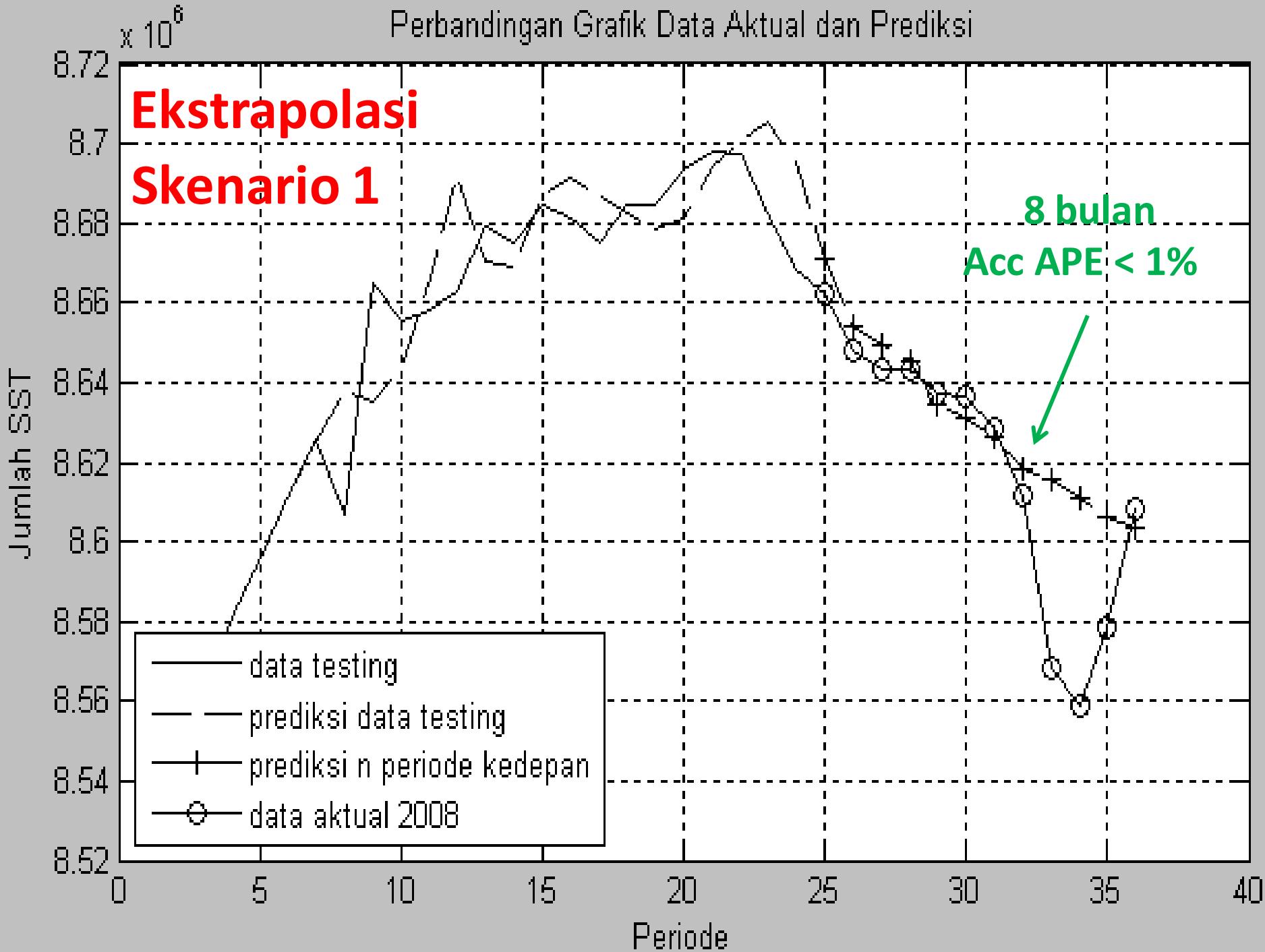
$$F_i = \left[\frac{((F'_i) - 0,1)}{0,8} \right] \times (\max(X) - \min(X)) + \min(X)$$

- F_i = nilai prediksi dengan range nilai asli
- F'_i = nilai prediksi dari hasil data yang dinormalisasi
- X = data aktual

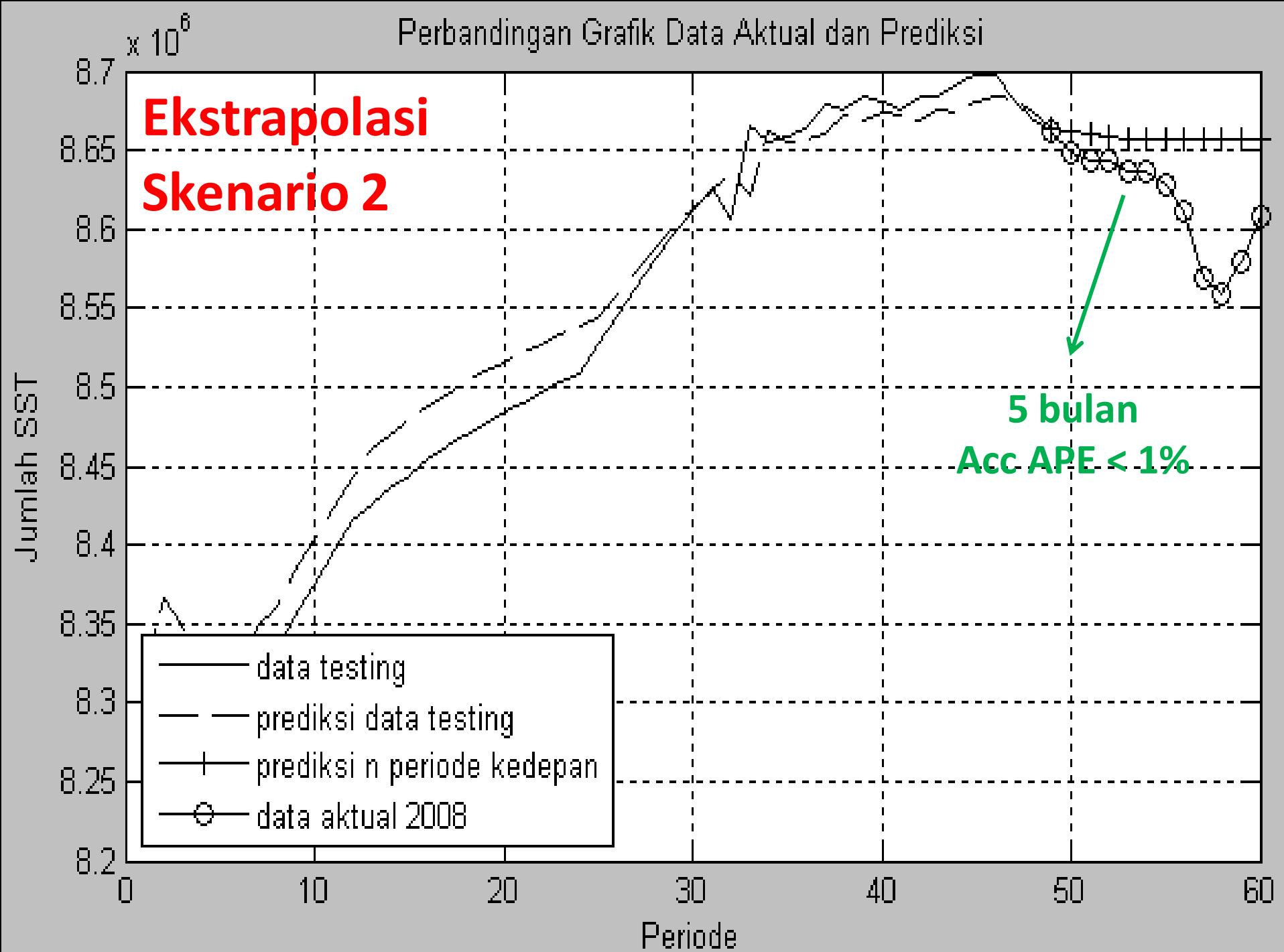
Pembangunan Model

- **Skenario 1**
 - Data *training* : 24 bulan (2002 - 2003)
 - Data *validation* : 24 bulan (2004 - 2005)
 - Data *testing* : 24 bulan (2006 - 2007)
- **Skenario 2**
 - Data *training* : 12 bulan (2002)
 - Data *validation* : 12 bulan (2003)
 - Data *testing* : 48 bulan (2004 - 2007)
- **Skenario 3**
 - Data *training* : 48 bulan (2002 - 2005)
 - Data *validation* : 12 bulan (2006)
 - Data *testing* : 12 bulan (2007)

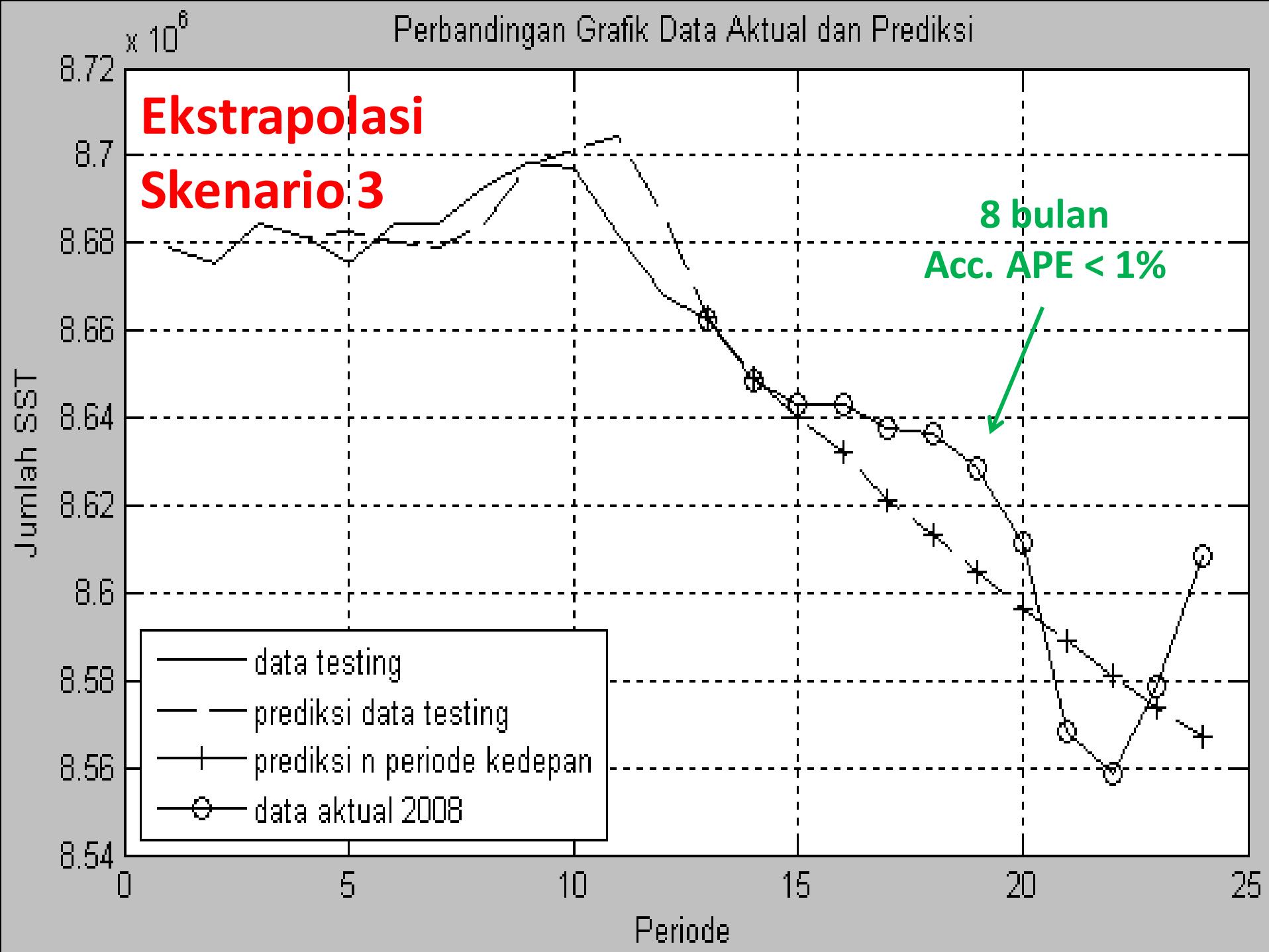
Perbandingan Grafik Data Aktual dan Prediksi



Perbandingan Grafik Data Aktual dan Prediksi



Perbandingan Grafik Data Aktual dan Prediksi



Kasus 4: Deteksi Kecurangan

- Jumlah pelanggan: 10 juta
- Data yg tersedia: tagihan bulanan selama 5 tahun
- Kecurangan:
 - Jika tagihan pada bulan ini jauh lebih sedikit atau lebih besar dibandingkan bulan-bulan sebelumnya
 - Jika tunggakan lebih dari 3 bulan dengan total tagihan jauh lebih besar dibandingkan bulan-bulan sebelumnya

Formulasi Masalah

- Input & Output?
- P dan T?
- Struktur dan parameter MLP?
- Perlu *preprocessing*?

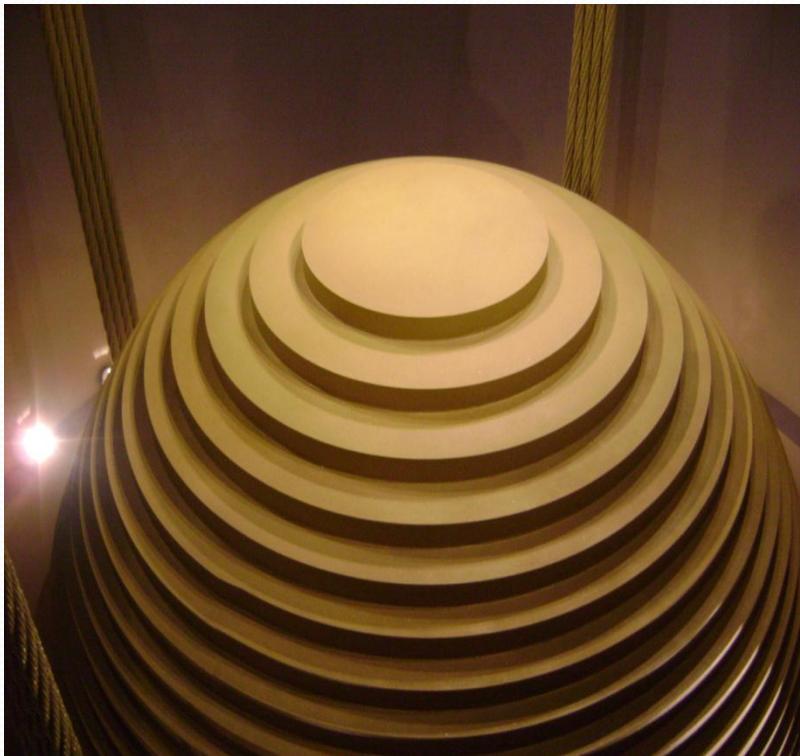
Kasus 5: Deteksi Churn

Kota	Tipe	Tipe Pembayaran	Tagihan Bulanan	Jumlah Panggilan	Panggilan TidakNormal	Churn
Jakarta	Pemerintah	Cash	Besar	10	Sedikit	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	8	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Kecil	5	Banyak	Tidak
Surabaya	Corporate	Cash	Kecil	3	Banyak	Ya
Surabaya	Corporate	Cash	Kecil	2	Banyak	Ya
Surabaya	Corporate	Kartu Kredit	Besar	1	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	9	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	7	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	6	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	4	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	9	Sedang	Tidak

Formulasi Masalah

- Input & Output?
- P dan T?
- Struktur dan parameter MLP?
- Perlu *preprocessing*?

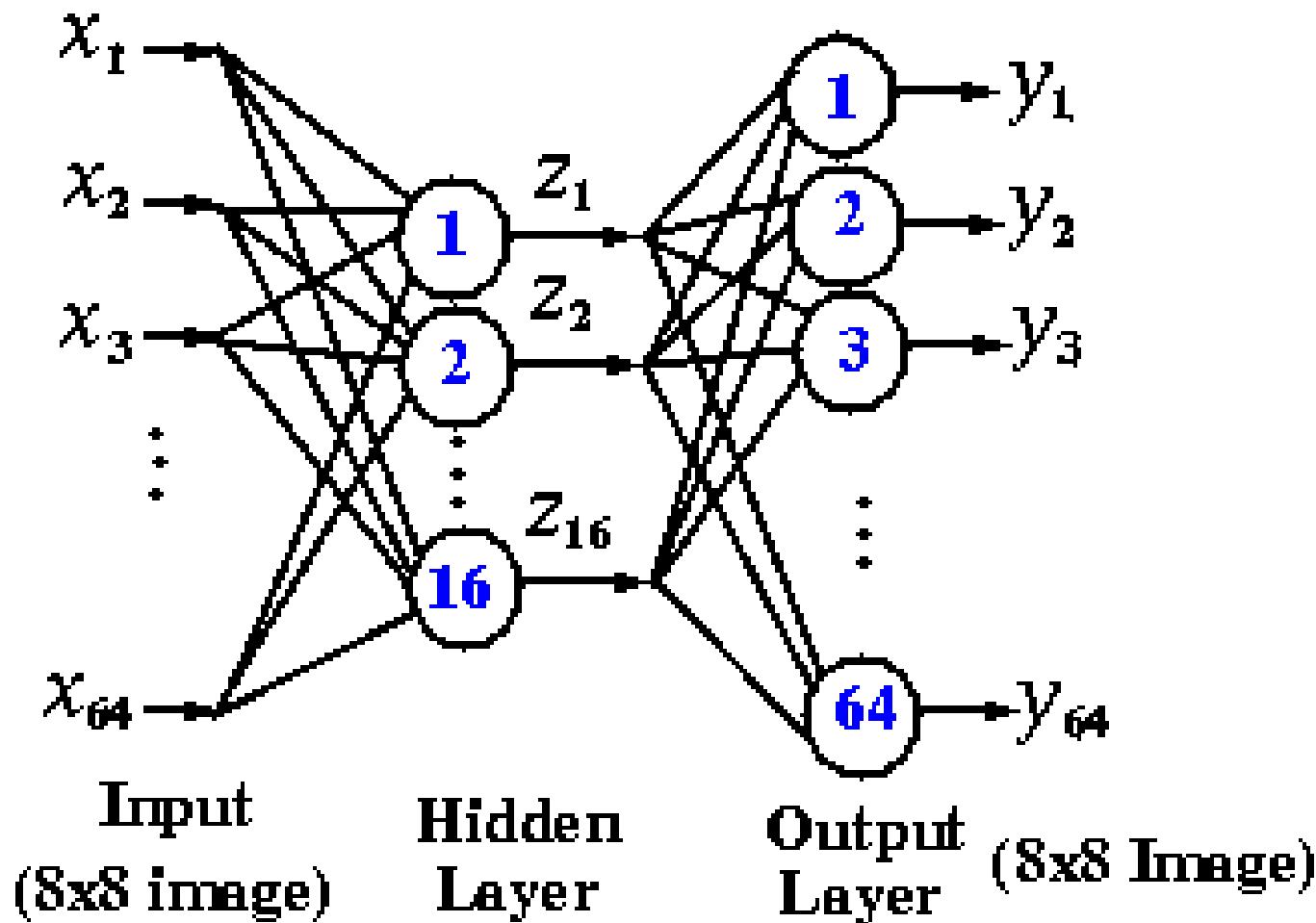
Kasus 6: Kompresi Citra

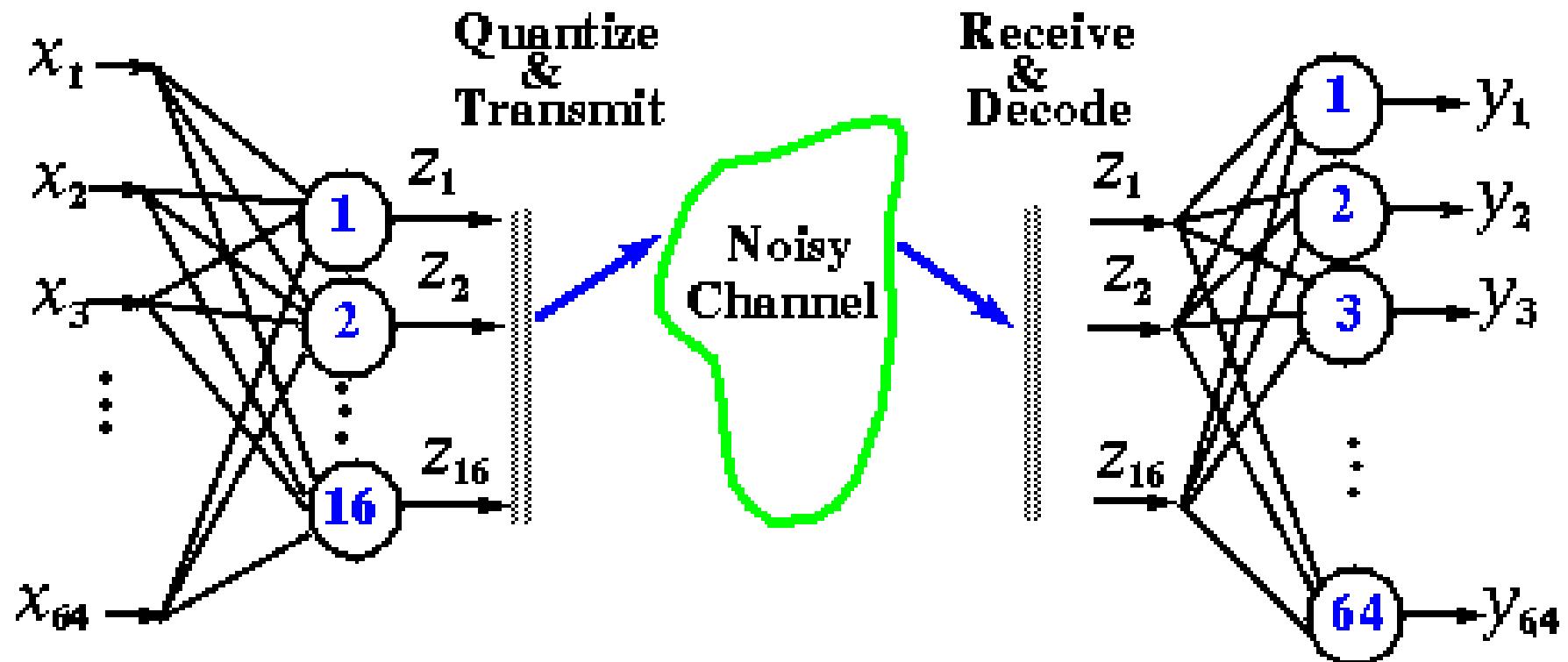


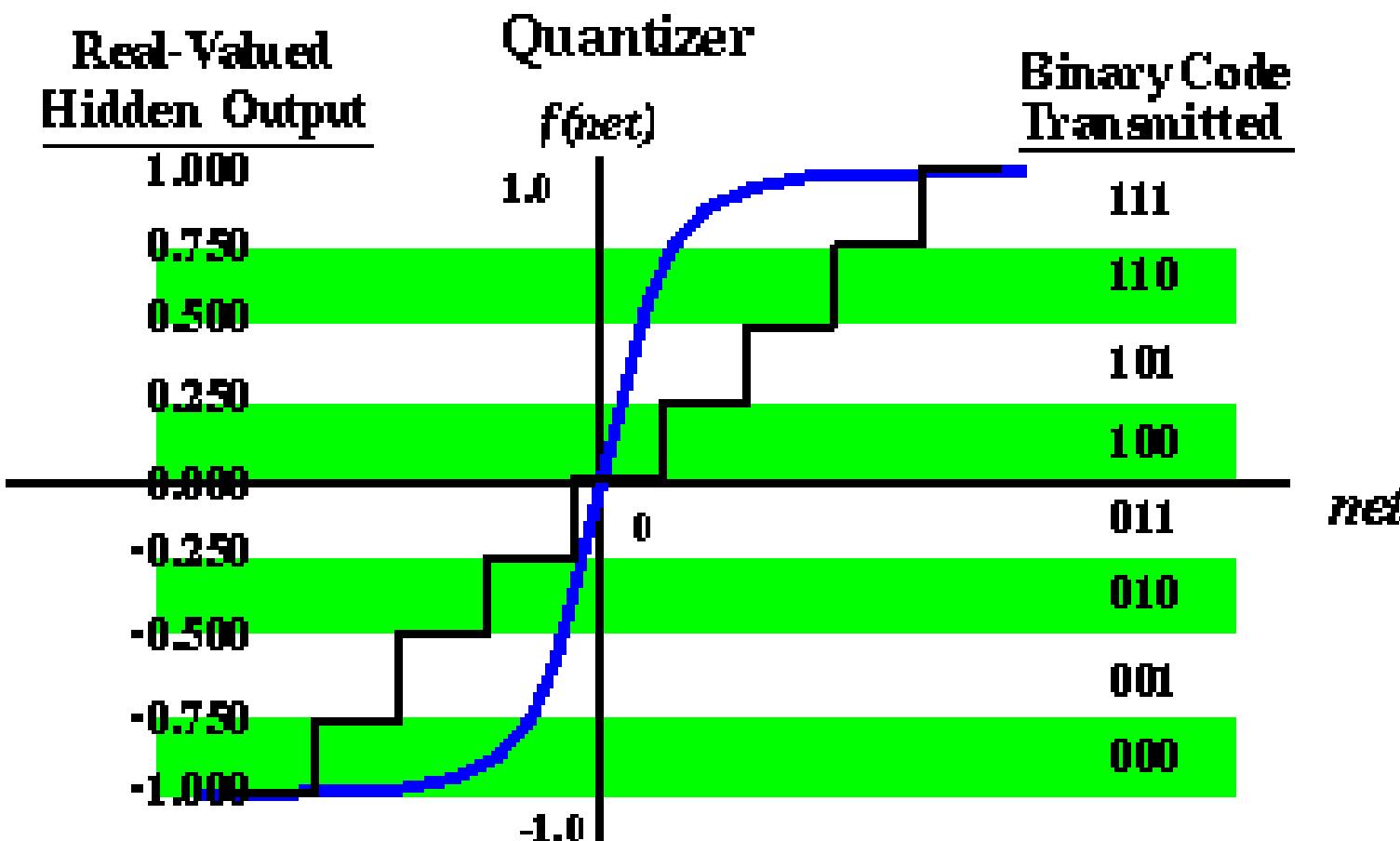
1024x1024 pixel, size: **3 MB**

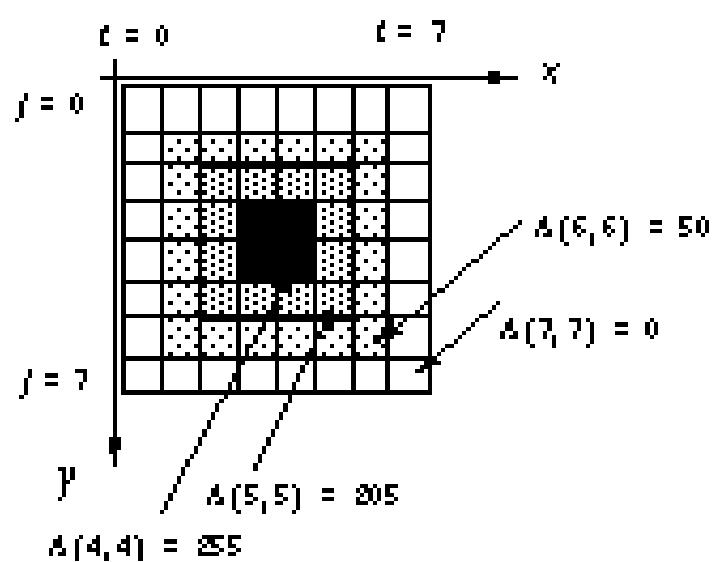
Formulasi Masalah

- Input & Output?
- P dan T?
- Struktur dan parameter MLP?
- Perlu *preprocessing*?

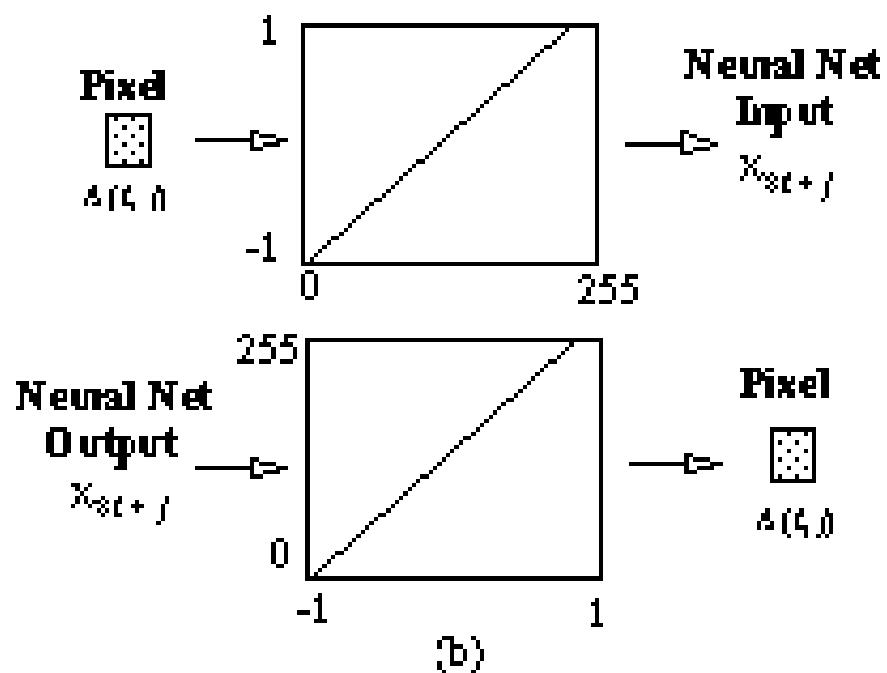








(a)



MLP

- Fungsi Aktivasi: umumnya $(0, 1)$ atau $(-1, 1)$
 - Preprocessing
 - Normalisasi Data Input
 - Denormalisasi Data Output
- Output dari suatu neuron bisa biner (0 atau 1) maupun real dalam interval $(0, 1)$ atau $(-1, 1)$
 - Klasifikasi → setiap kelas dikodekan dalam biner
 - Prediksi → output didenormalisasi
 - Optimasi → preprocessing yang agak rumit

Supervised vs Unsupervised

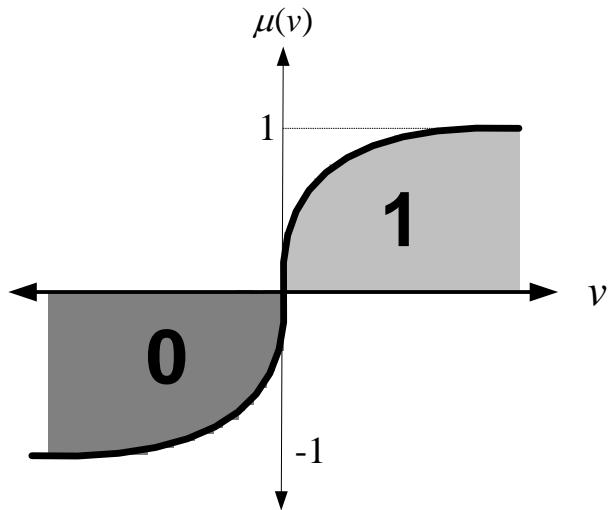
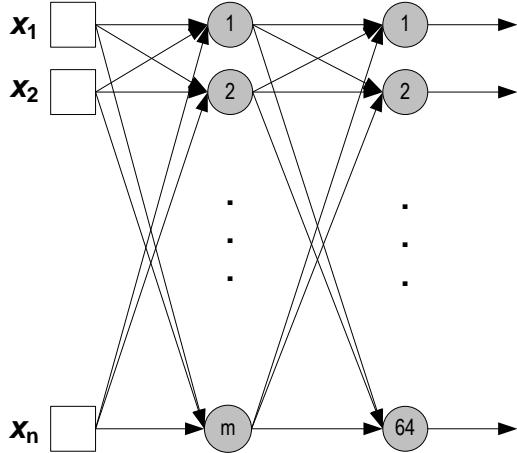
Supervised	Unsupervised
Klasifikasi	Clustering
Kelas harus diketahui	Kelas tidak harus diketahui
Waktu training lambat	Waktu training cepat

ANN

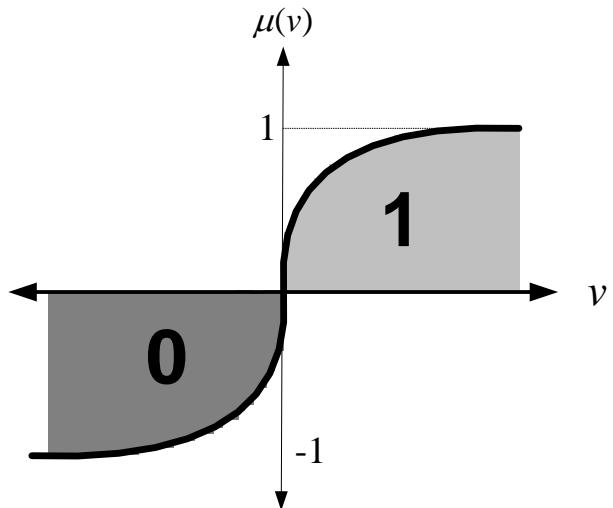
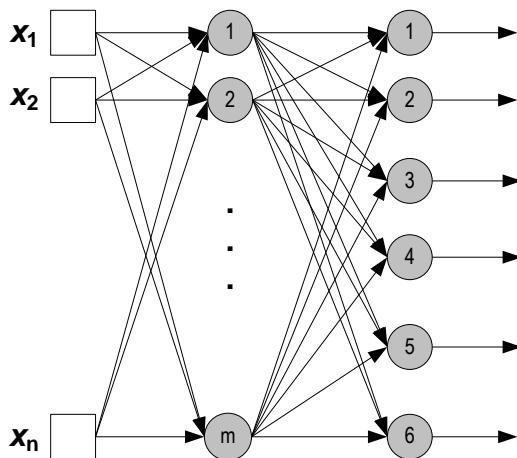
- Klasifikasi atau Clustering
- Mudah implementasi
- Akurasi tinggi
- Tahan *noise*
- Implementasi hardware (CHIP)
- Harus tersedia data latih dengan kelas yang jelas
- Waktu training lama
- Training ulang
- Penalarannya tidak bisa dijelaskan (*Weights*)

Jumlah neuron pada output layer

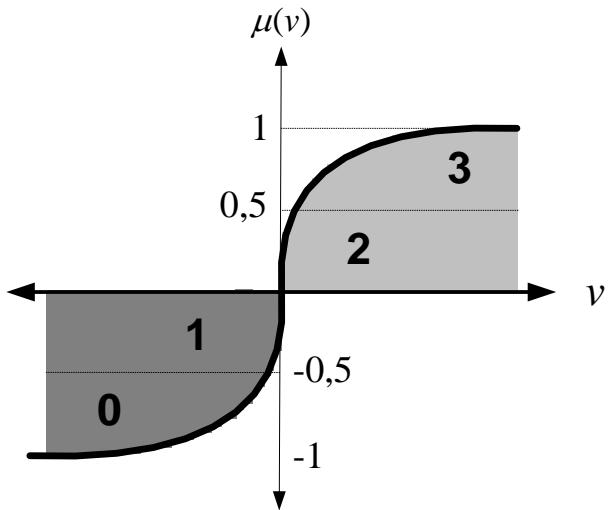
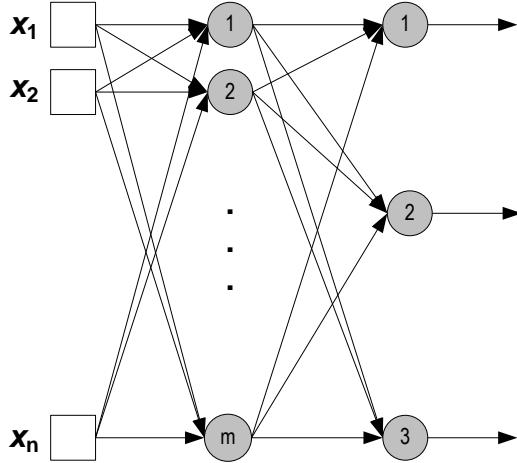
- **Deterministik:** mudah dihitung berdasarkan permasalahan yang dihadapi.
- Untuk pengenalan karakter dengan **64 kelas:** ('a', 'b', ..., 'z', 'A', 'B', ..., 'Z', 'o', 'i', ... '9', '-', '+'), perlu berapa output neuron?



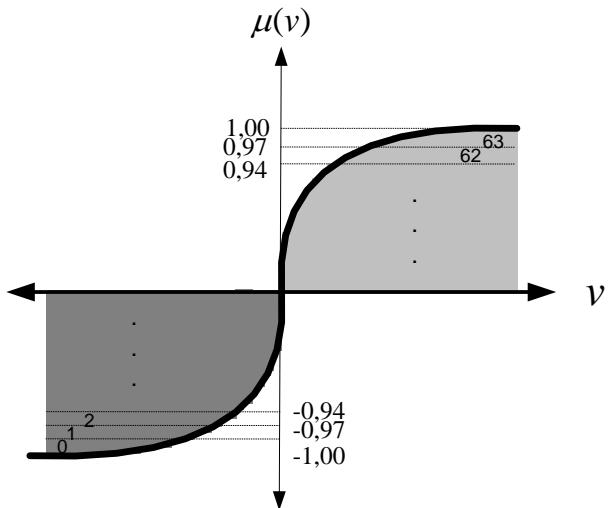
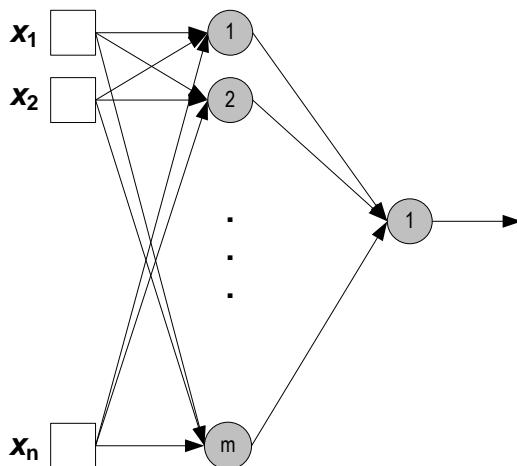
Output dari neuron ke-								Kelas
1	2	3	4	5	...	64		
1	0	0	0	0	...	0	a	
0	1	0	0	0	...	0	b	
0	0	1	0	0	...	0	c	
0	0	0	1	0	...	0	d	
0	0	0	0	1	...	0	e	
...	
0	0	0	0	0	...	1	+	



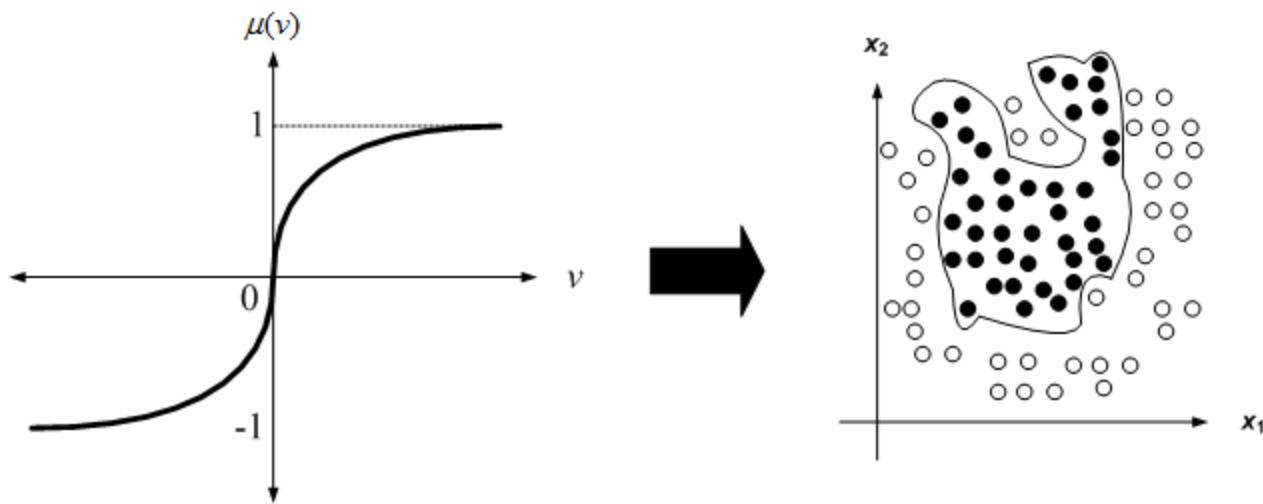
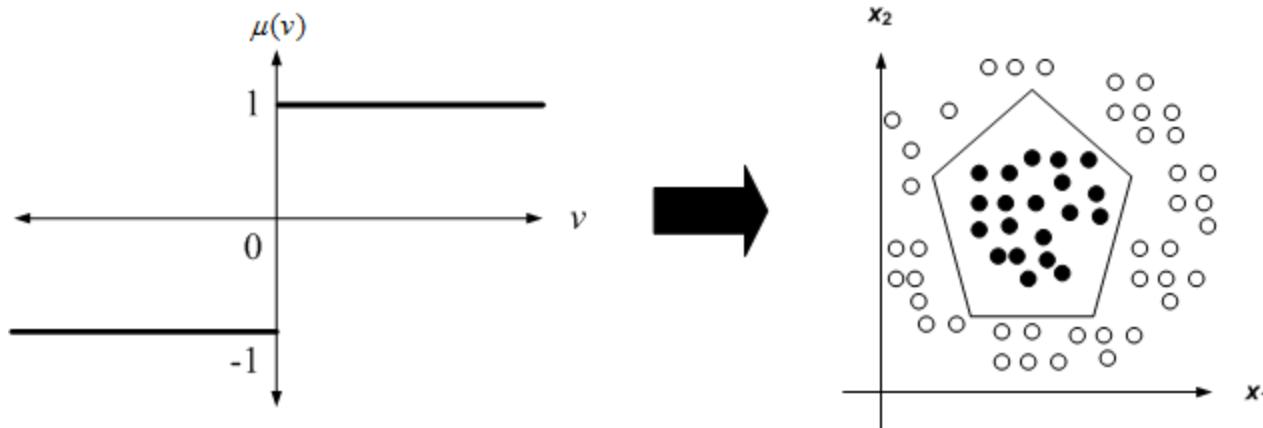
Output neuron ke-						Kelas
1	2	3	4	5	6	
0	0	0	0	0	0	a
0	0	0	0	0	1	b
0	0	0	0	1	0	c
0	0	0	0	1	1	d
0	0	0	1	0	0	e
...
1	1	1	1	1	1	+



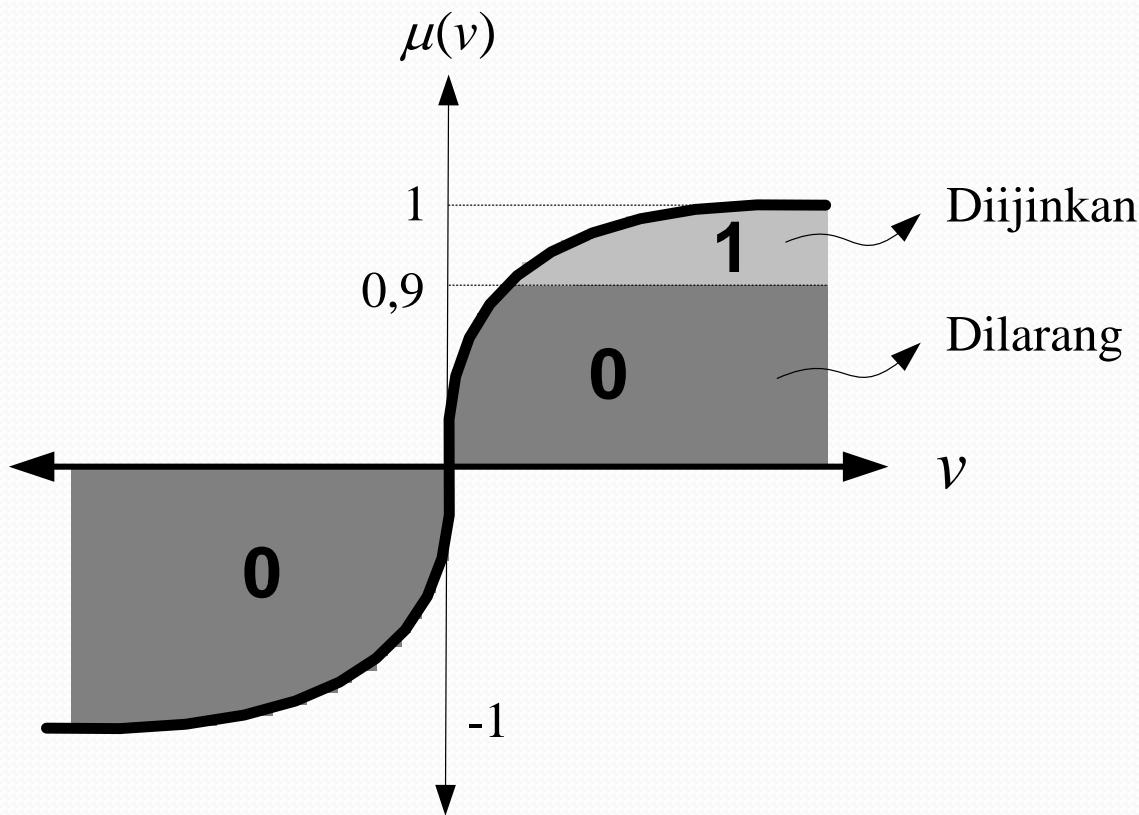
Output neuron ke-			Kelas
1	2	3	
0	0	0	a
0	0	1	b
0	0	2	c
0	0	3	d
0	1	0	e
...
3	3	3	+



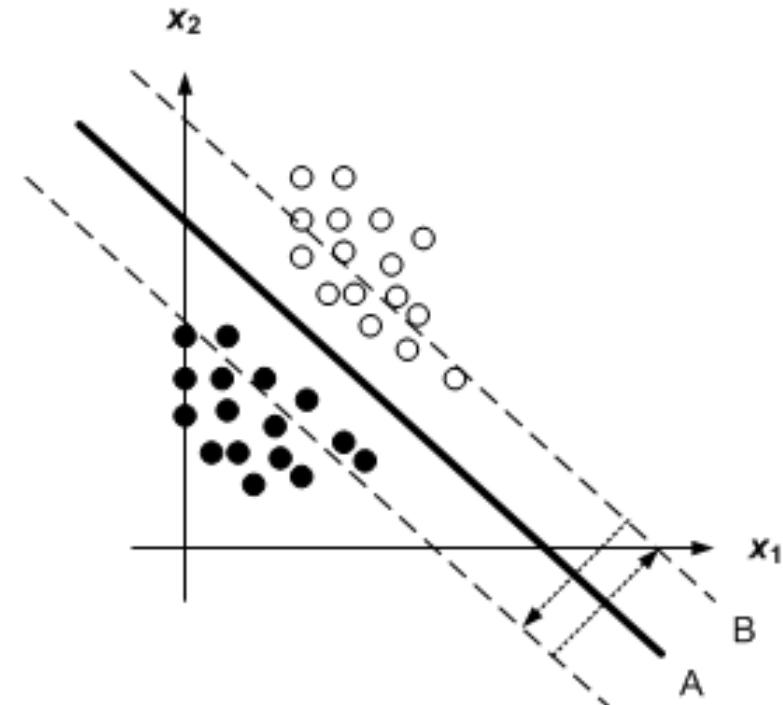
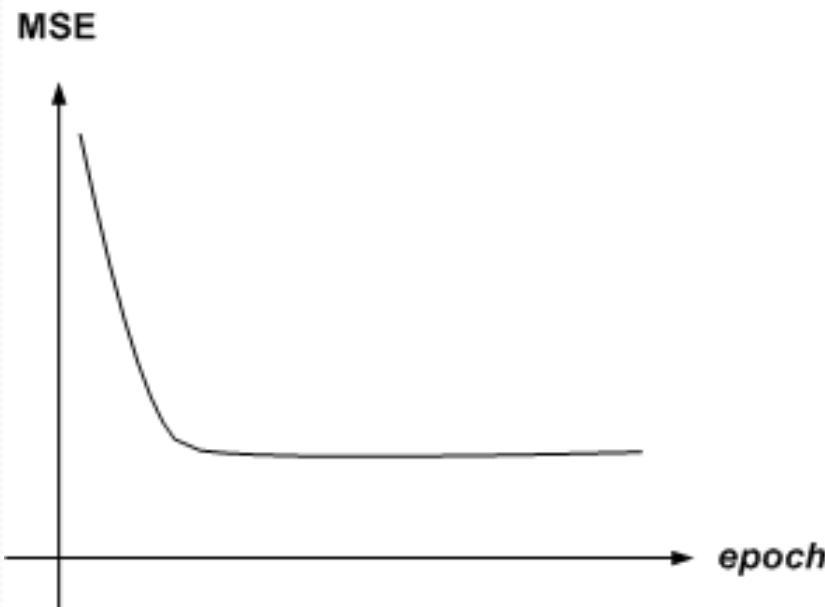
Output neuron ke-		Kelas
1	2	
0		a
1		b
2		c
3		d
4		e
...		...
63		+

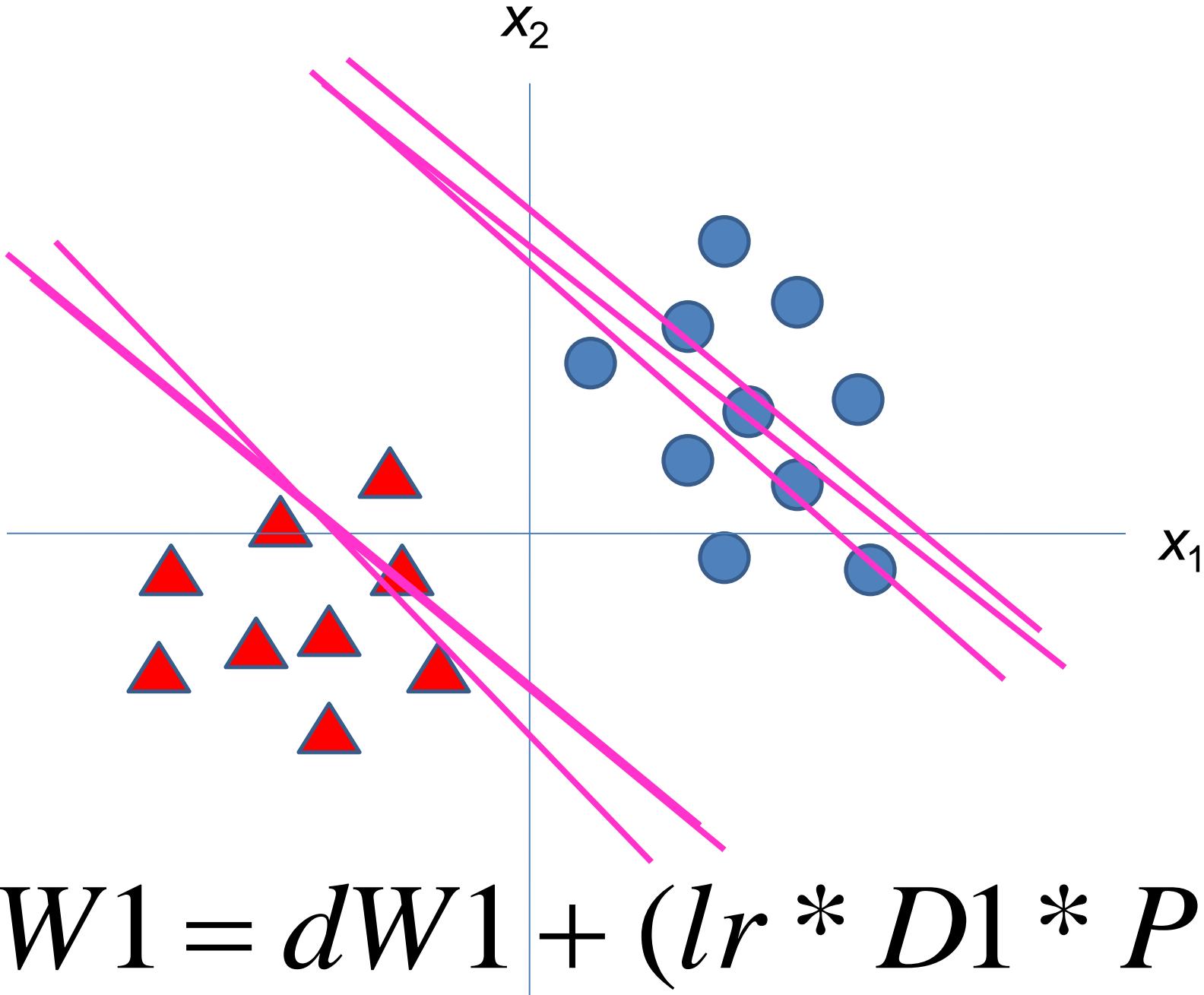


Security Systems



Learning Rate: Besar





$$dW1 = dW1 + (lr * D1 * P)$$

Algoritma Belajar Propagasi Balik

- Pelatihan Jaringan
 - Perhitungan Mundur

$$D2 = (1 - A2^2) * E$$

$$D1 = (1 - A1^2) * (W2 * D2)$$

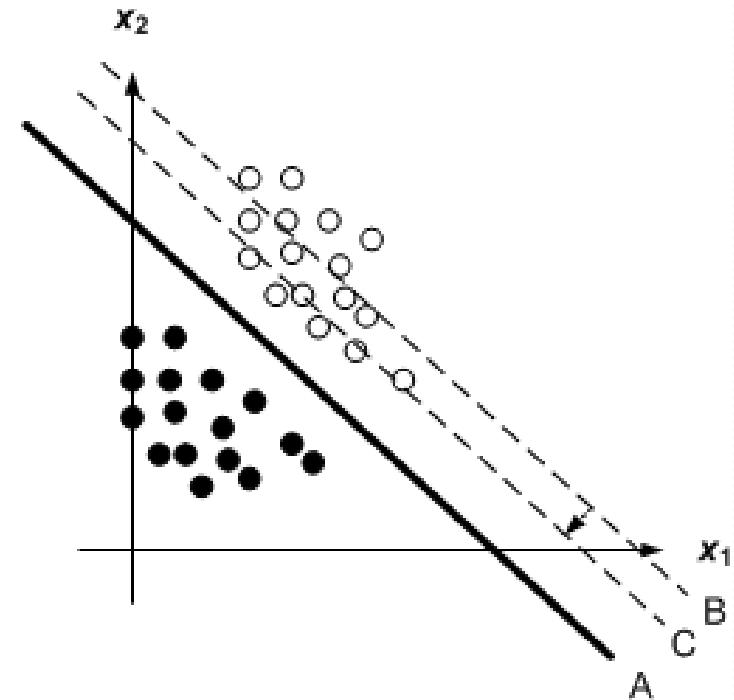
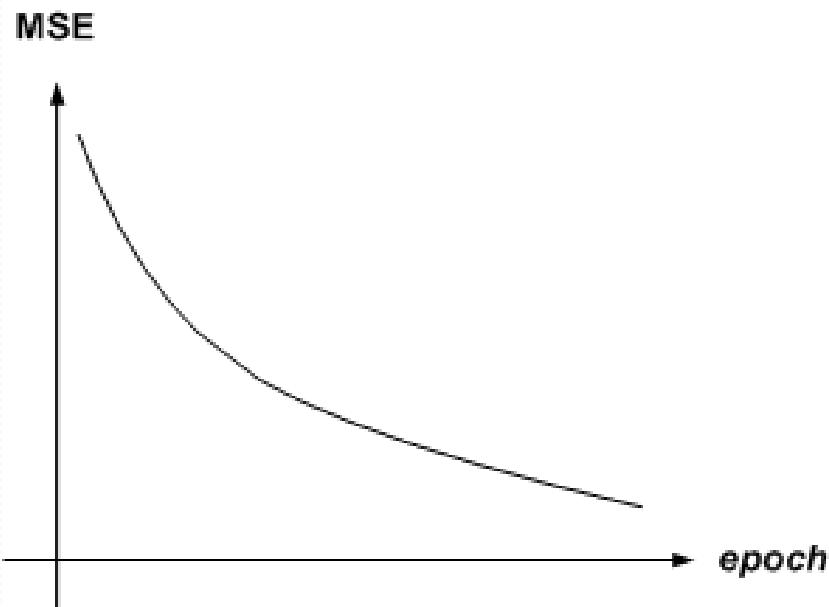
$$dW1 = dW1 + (lr * D1 * P)$$

$$dB1 = dB1 + (lr * D1)$$

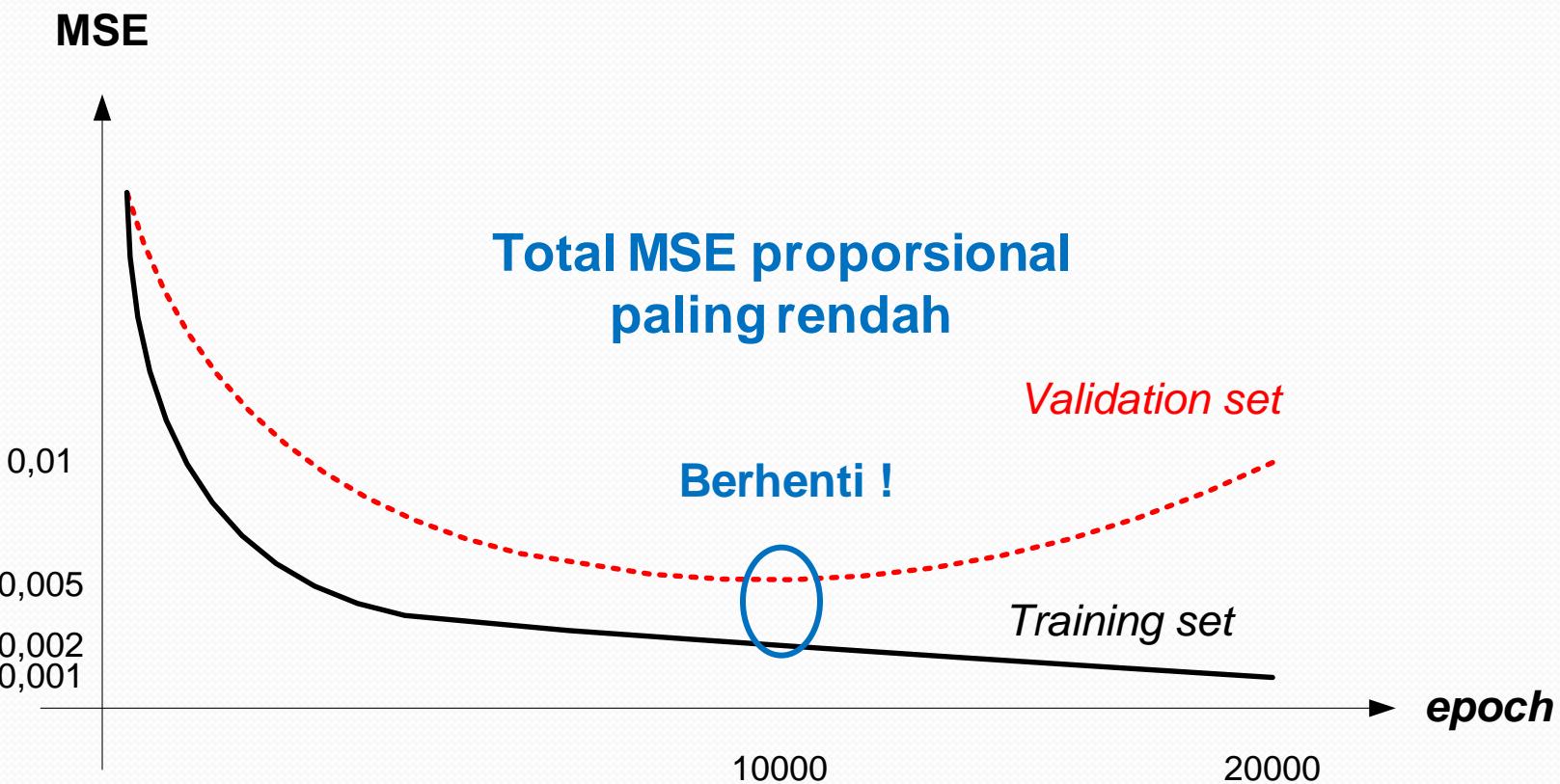
$$dW2 = dW2 + (lr * D2 * P)$$

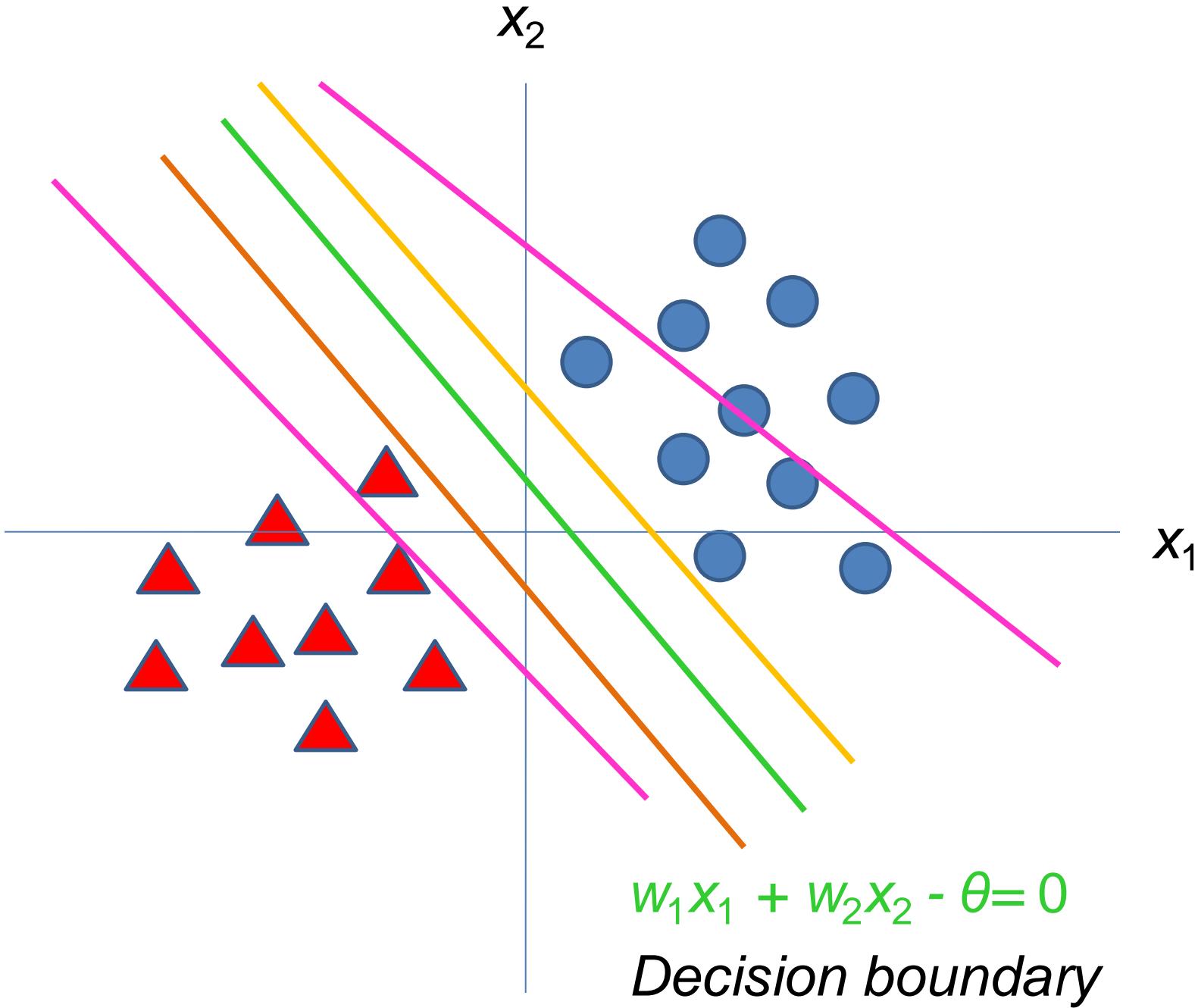
$$dB2 = dB2 + (lr * D2)$$

Learning Rate: Kecil

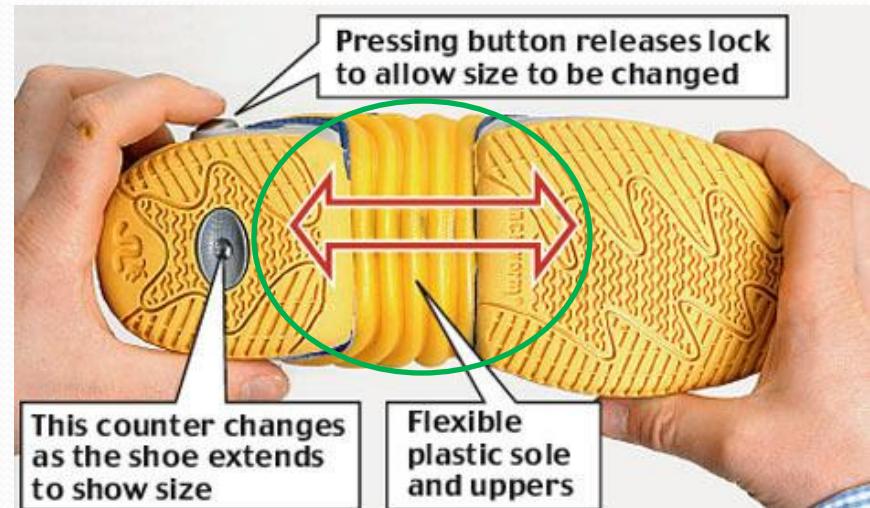


Kapan Menghentikan *Learning*?





Overfit, Oversize, Flexible



Strategi penggunaan ANN

- **Cara memandang masalah:**
 - Klasifikasi
 - Sekuriti
 - Prediksi
 - Optimasi
- **Teknik learning:** Supervised/Unsupervised
- **Desain Arsitektur**
 - Jumlah layer
 - Jumlah neuron
 - Pemetaan output
- **Strategi learning**
 - Penyiapan data: filterisasi data, pembagian data (training, validasi, test)
 - Parameter: inisialisasi (acak atau memakai algoritma), laju belajar, dsb.
 - Penghentian learning



Menemukan persamaan dari objek berbeda?

Menemukan perbedaan dari objek mirip?

Menemukan fitur / ciri penting?

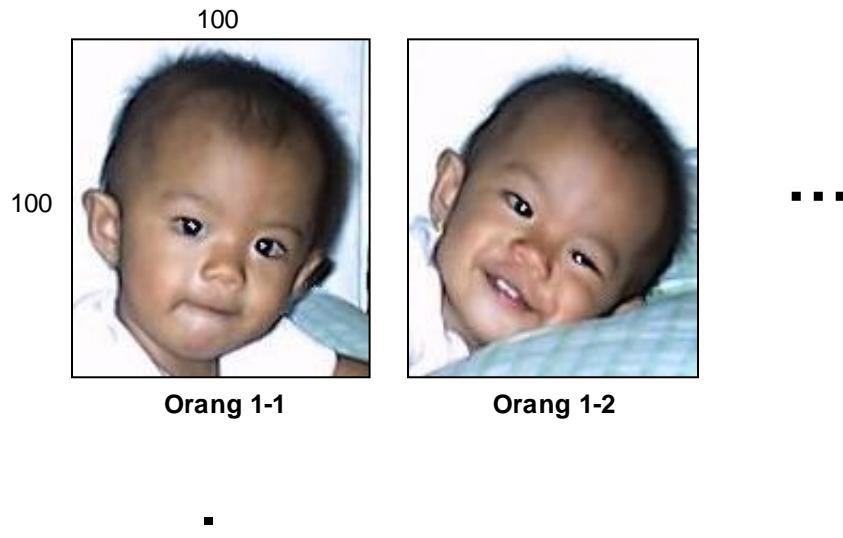
Tidak pernah diajari !!!

UNSUPERVISED Learning

Perbedaan?



- Wajah
- Suara
- Bau
- Gesture
- Sidik Jari?
- Iris Mata?
- DNA?



Face Recognition

- 50 orang
- @ 10 variation
- 100x100 pixels
- Matris 500x10000

- Efisien?
- Bisa direduksi?

PCA

PCA

- Principle Component Analysis
- Reduksi dimensi data
- Analisa data
- Ekstraksi ciri
- Visualisasi
- Mengurangi noise

Objek	Panjang	Lebar	Tinggi	Kelas
Objek 1	2,1	1,5	0,8	Meja
Objek 2	2,3	1,7	0,8	Meja
Objek 3	2,1	1,3	0,8	Kursi
Objek 4	1,6	1,5	0,8	Kursi
Objek 5	2,5	1,9	0,8	Meja

↑
Tidak penting

Bisa direduksi menjadi 2 dimensi? Bisa.

Objek	Panjang	Lebar	Kelas	
Objek 1	2,1	1,5	Meja	
Objek 2	2,3	1,7	Meja	
Objek 3	2,1	1,3	Kursi	
Objek 4	1,6	1,5	Kursi	
Objek 5	2,5	1,9	Meja	

Bisa direduksi menjadi 1 dimensi? Tidak bisa. Ada data overlap.

Lebar

Y



Domain PCA
Sumbu Y Kita mengingat?

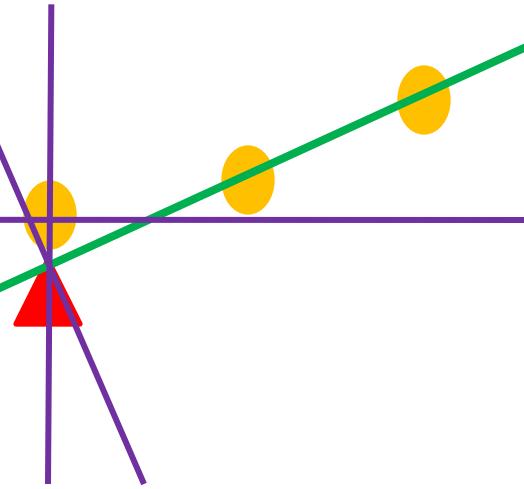
Bisa direduksi menjadi 1 dimensi?

X

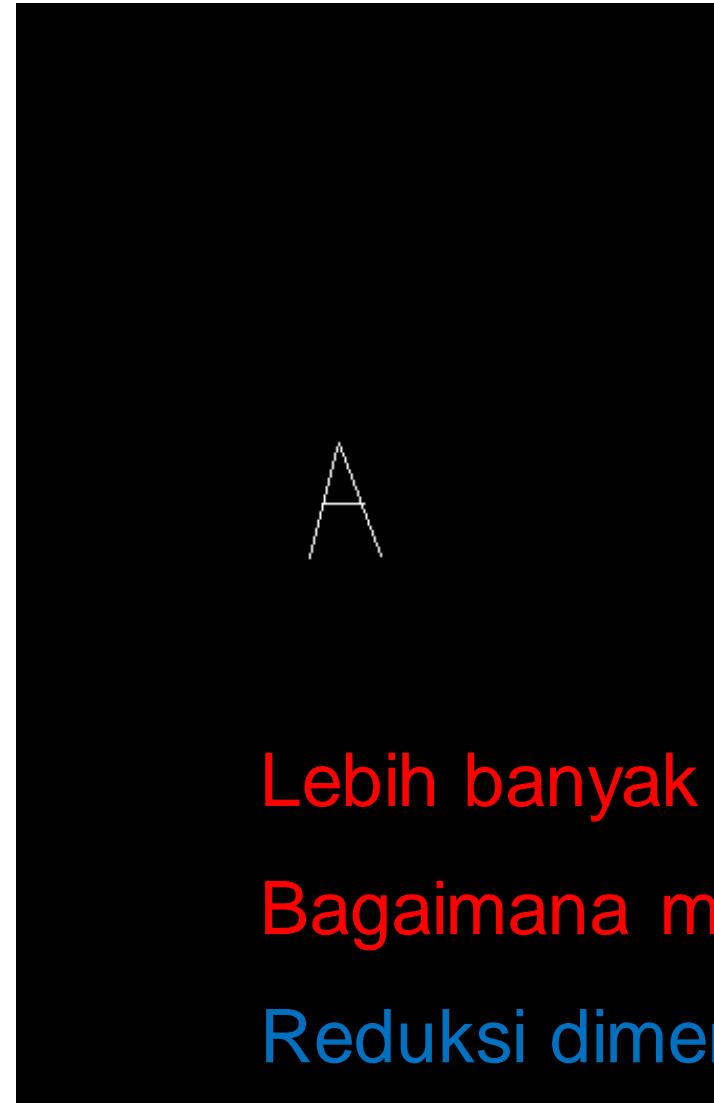
1,5

2,1

Panjang



200



300

A

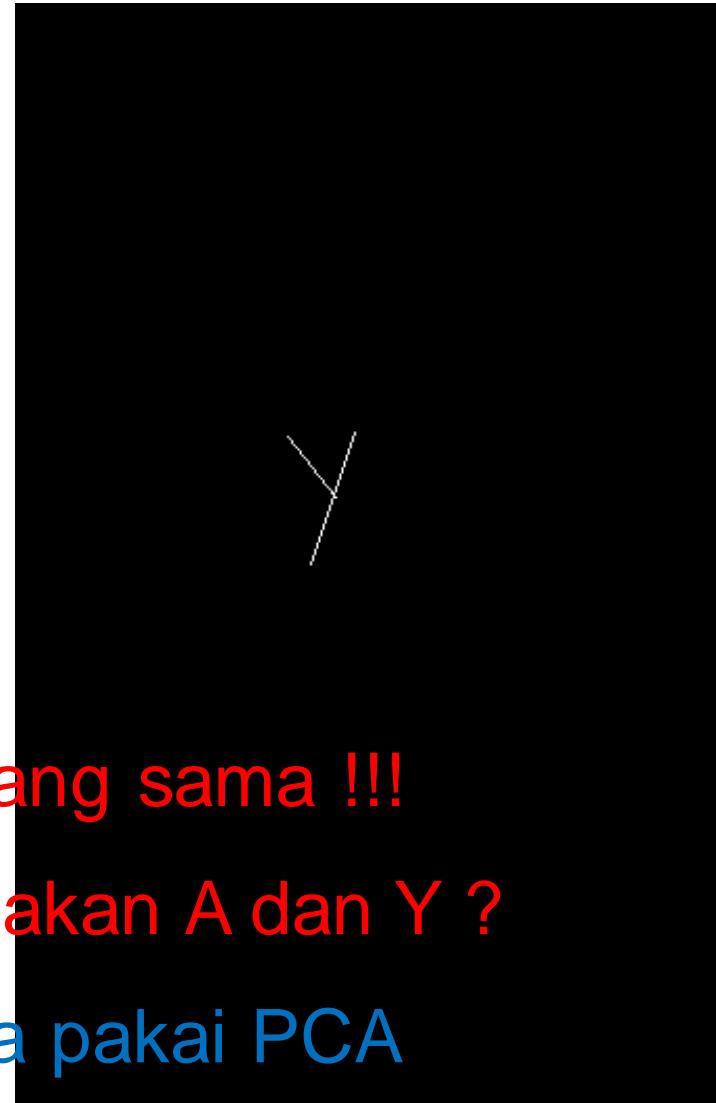
Lebih banyak pixel yang sama !!!

Bagaimana membedakan A dan Y ?

Reduksi dimensi data pakai PCA

60000 dimensi → 40 dimensi

200



300

Y

$$B = \begin{bmatrix} 3 & 5 & 4 & 5 & 2 \\ 2 & 7 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix}$$

Rata-rata pada setiap dimensi

Transpose

Jumlah data (observasi)

$$Cov(B) = \frac{(B - \bar{B}) * (B - \bar{B})^T}{N - 1}$$

Vektor Covarian

$$C_B = Cov(B) = \begin{bmatrix} 1,7 & 1,6 & 0 \\ 1,6 & 7,3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Dari matriks covarian tersebut, dapat dihitung suatu basis orthogonal dengan mencari *EIGEN VALUE* dan *EIGEN VECTOR*-nya.
- *Eigenvectors* v_i dan *eigenvalues* L_i , adalah solusi dari persamaan:

$$C_B v_i = L_i v_i \quad , i=1,2,3,\dots,n \longrightarrow \text{dimensi}$$

Jika L_i sudah ditemukan, maka *eigen vector* dapat ditemukan dari persamaan:

$$\left| C_B - L_i I \right| \rightarrow \begin{array}{l} \text{Matriks identitas yang berukuran sama dengan } C_B \\ \text{Determinant dari matriks di dalamnya} \end{array}$$

$$B = \begin{bmatrix} 3 & 5 & 4 & 5 & 2 \\ 2 & 7 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix}$$



$$[v \ L] = \text{eig}(C_B)$$

$$v = \begin{bmatrix} v_3 & v_2 & v_1 \\ 0,0000 & -0,9648 & 0,2631 \\ 0,0000 & 0,2631 & 0,9648 \\ 1,0000 & 0,0000 & 0,0000 \end{bmatrix} \quad L = \begin{bmatrix} L_3 & L_2 & L_1 \\ 0,0000 & 0,0000 & 0,0000 \\ 0,0000 & 1,2500 & 0,0000 \\ 0,0000 & 0,0000 & 7,7500 \end{bmatrix}$$

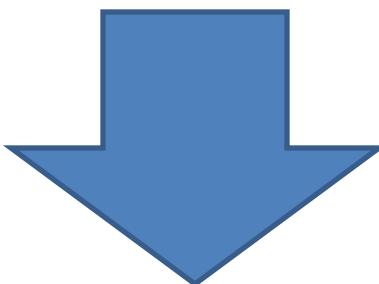
Jika v_{row} didapat dari v^T yang diurutkan berdasarkan *eigenvalues* terbesar

$$v_{row} = \begin{bmatrix} 0,2631 & 0,9648 & 0,000 \\ -0,9648 & 0,2631 & 0,0000 \\ 0,0000 & 0,0000 & 1,0000 \end{bmatrix}$$

maka matriks B dapat ditransformasikan ke dalam representasi PCA:

$$Y = v_{row} * B$$

$$v_{row} = \begin{bmatrix} 0,2631 & 0,9648 & 0,000 \\ -0,9648 & 0,2631 & 0,0000 \\ 0,0000 & 0,0000 & 1,0000 \end{bmatrix} * B = \begin{bmatrix} 3 & 5 & 4 & 5 & 2 \\ 2 & 7 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix}$$

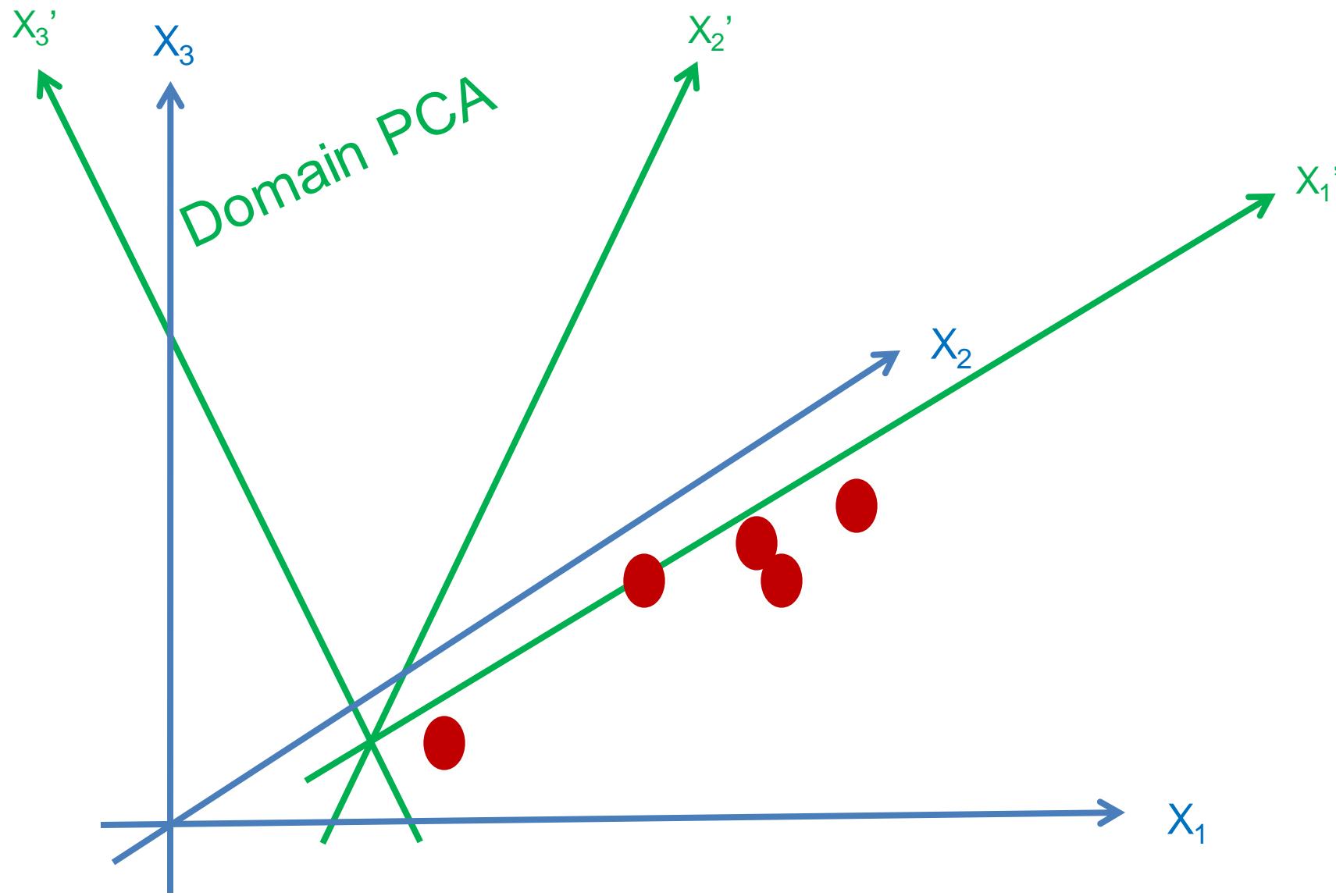


PCA



$$Y = \begin{bmatrix} 2,7189 & 8,0689 & 8,7706 & 9,9985 & 7,2796 \\ -2,3681 & -2,9820 & -1,7541 & -2,4558 & -0,0877 \\ 0,0000 & 0,0000 & 0,0000 & 0,0000 & 0,0000 \end{bmatrix}$$

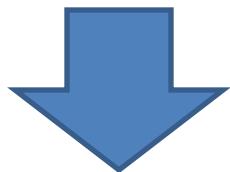
$$B = \begin{bmatrix} 3 & 5 & 4 & 5 & 2 \\ 2 & 7 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{aligned} x_1 \\ x_2 \\ x_3 \end{aligned} \quad Y = \begin{bmatrix} 2,7189 & 8,0689 & 8,7706 & 9,9985 & 7,2796 \\ -2,3681 & -2,9820 & -1,7541 & -2,4558 & -0,0877 \\ 0,0000 & 0,0000 & 0,0000 & 0,0000 & 0,0000 \end{bmatrix}$$



Jika v_{row} dibangun dari ketiga *eigenvector* yang ada, maka matriks B dapat dibangun kembali dengan tepat (**tanpa kesalahan**) menggunakan Y dan v_{row}

$$B_{rev} = v_{row}^T * Y$$

$$v_{row}^T = \begin{bmatrix} 0,0000 & -0,9648 & 0,2631 \\ 0,0000 & 0,2631 & 0,9648 \\ 1,0000 & 0,0000 & 0,0000 \end{bmatrix} * Y = \begin{bmatrix} 2,7189 & 8,0689 & 8,7706 & 9,9985 & 7,2796 \\ -2,3681 & -2,9820 & -1,7541 & -2,4558 & -0,0877 \\ 0,0000 & 0,0000 & 0,0000 & 0,0000 & 0,0000 \end{bmatrix}$$



$$B_{rev} = \begin{bmatrix} 3 & 5 & 4 & 5 & 2 \\ 2 & 7 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 3 & 5 & 4 & 5 & 2 \\ 2 & 7 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Jika v_{row} dibangun dari **satu eigenvector** yang eigenvalue-nya **terbesar** (v_1), maka matriks B dapat dibangun kembali dengan kesalahan (galat) sebesar $\text{norm}(B - B_{\text{rev}})$.

$$v_{\text{row}}^T = \begin{bmatrix} 0,2631 \\ 0,9648 \\ 0,0000 \end{bmatrix} * Y = \boxed{\begin{bmatrix} 2,7189 & 8,0689 & 8,7706 & 9,9985 & 7,2796 \end{bmatrix}}$$

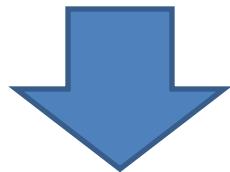


$$B_{\text{rev}} = \begin{bmatrix} 0,7154 & 2,1231 & 2,3077 & 2,6308 & 1,9154 \\ 2,6231 & 7,7846 & 8,4615 & 9,6462 & 7,0231 \\ 0,0000 & 0,0000 & 0,0000 & 0,0000 & 0,0000 \end{bmatrix}$$

$$B = \boxed{\begin{bmatrix} 3 & 5 & 4 & 5 & 2 \\ 2 & 7 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}$$

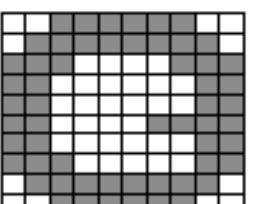
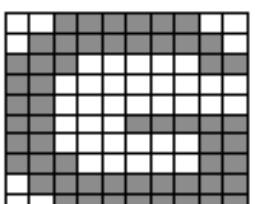
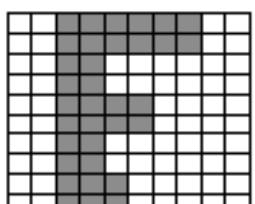
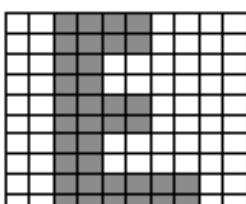
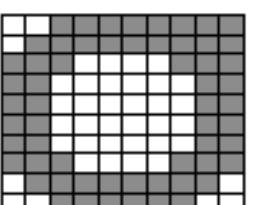
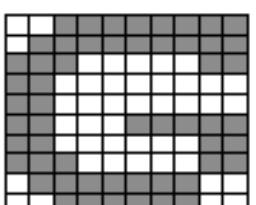
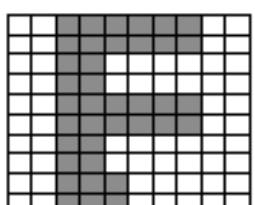
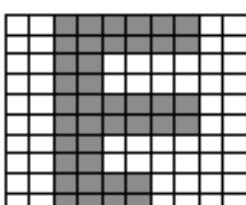
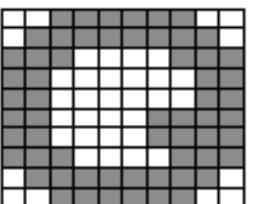
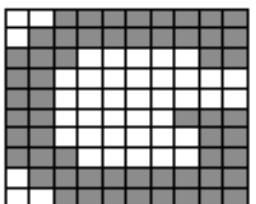
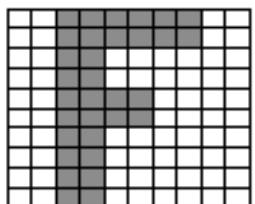
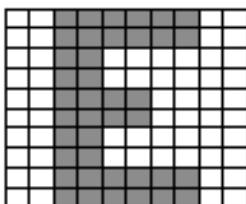
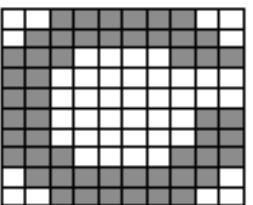
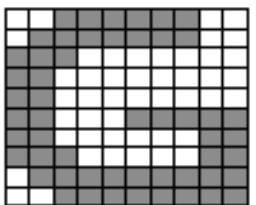
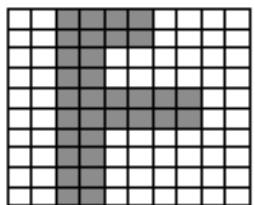
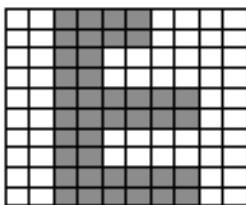
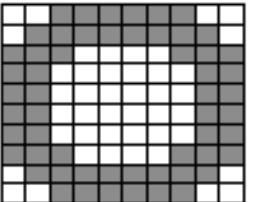
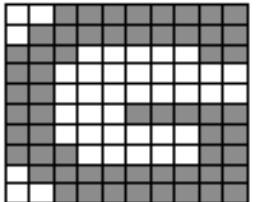
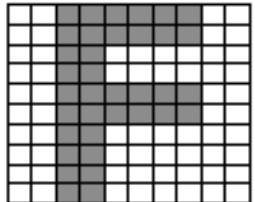
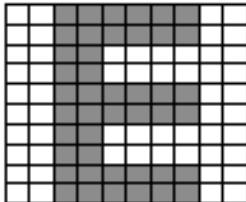
Jika v_{row} dibangun dari **dua eigenvector** yang **eigenvalue-nya terbesar**, (v_1 dan v_2), maka matriks B dapat dibangun kembali dengan kesalahan (galat) yang lebih kecil dibandingkan dengan jika v_{row} dibangun dari satu **eigenvector**.

$$v_{\text{row}}^T = \begin{bmatrix} 0,2631 & -0,9648 \\ 0,9648 & 0,2631 \\ 0,0000 & 0,000 \end{bmatrix} * Y = \begin{bmatrix} 2,7189 & 8,0689 & 8,7706 & 9,9985 & 7,2796 \\ -2,3681 & -2,9820 & -1,7541 & -2,4558 & -0,0877 \end{bmatrix}$$



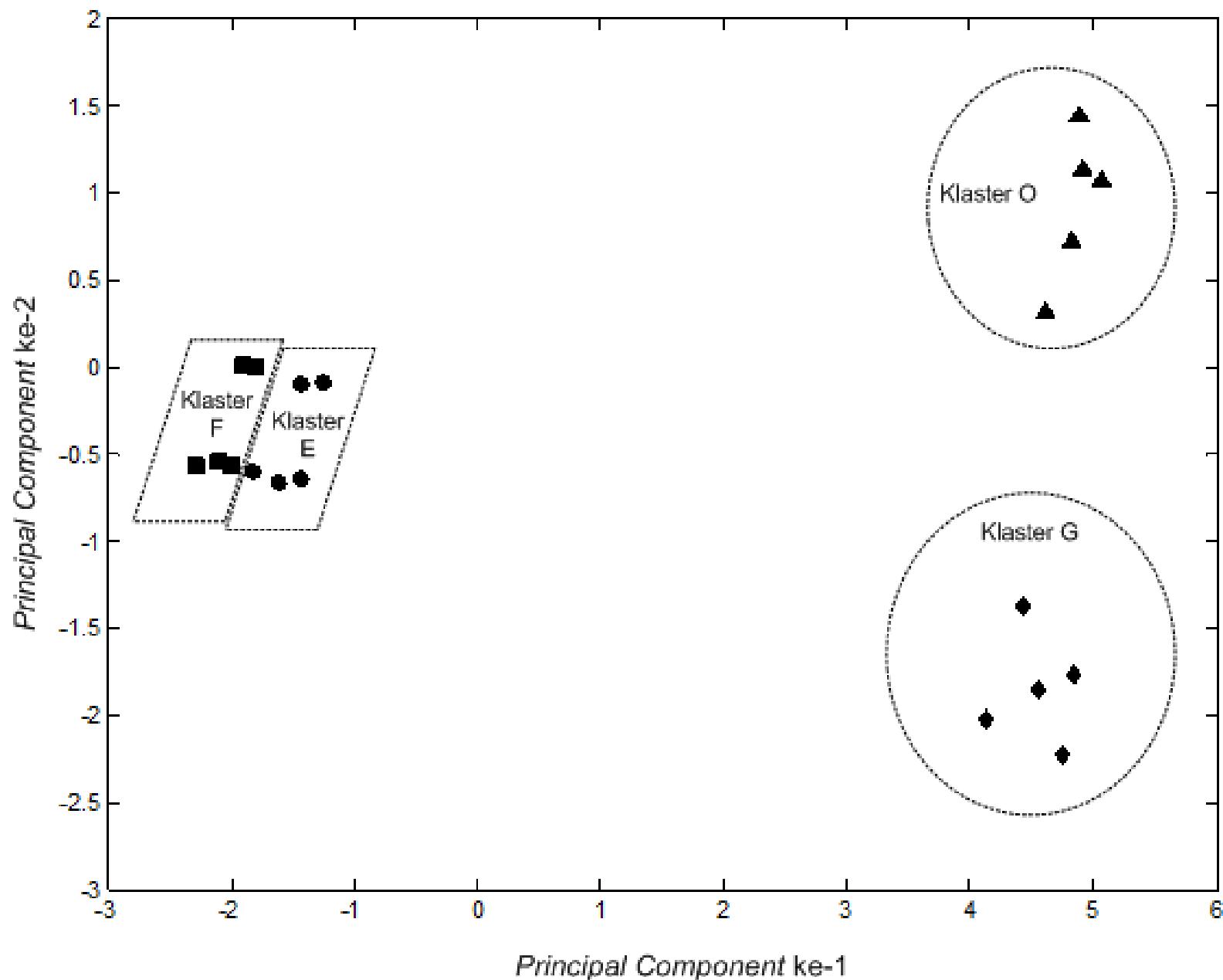
$$B_{\text{rev}} = \begin{bmatrix} 3,0001 & 5,0000 & 3,9999 & 5,0000 & 1,9999 \\ 2,0001 & 7,0003 & 8,0004 & 9,0004 & 7,0003 \\ 0,0000 & 0,0000 & 0,0000 & 0,0000 & 0,0000 \end{bmatrix}$$

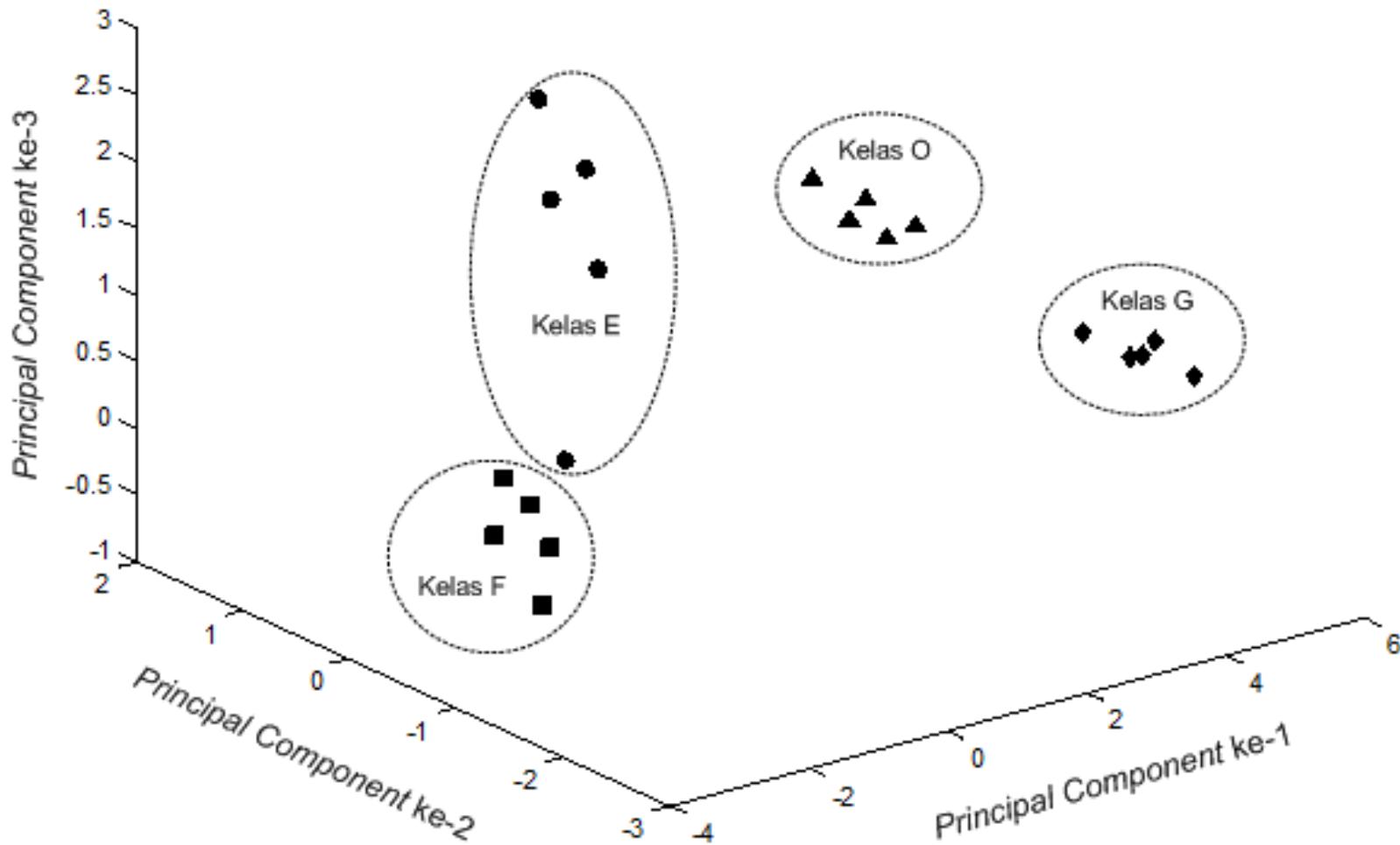
$$B = \begin{bmatrix} 3 & 5 & 4 & 5 & 2 \\ 2 & 7 & 8 & 9 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



- Visualisasi 100 dimensi?
- Tidak bisa dilakukan
- Reduksi $100 \rightarrow 2$ dimensi
- Gunakan PCA dari dua *eigenvector* yang *eigenvalue*-nya terbesar.

Pola	Pix 1	Pix 2	Pix 3	Pix 4	Pix 5	...	Pix 100
E1	0	0	1	1	1	...	0
F1	0	0	1	1	1	...	0
G1	0	1	1	1	1	...	1
O1	0	1	1	1	1	...	1
..							
O5	0	1	1	1	1	...	1

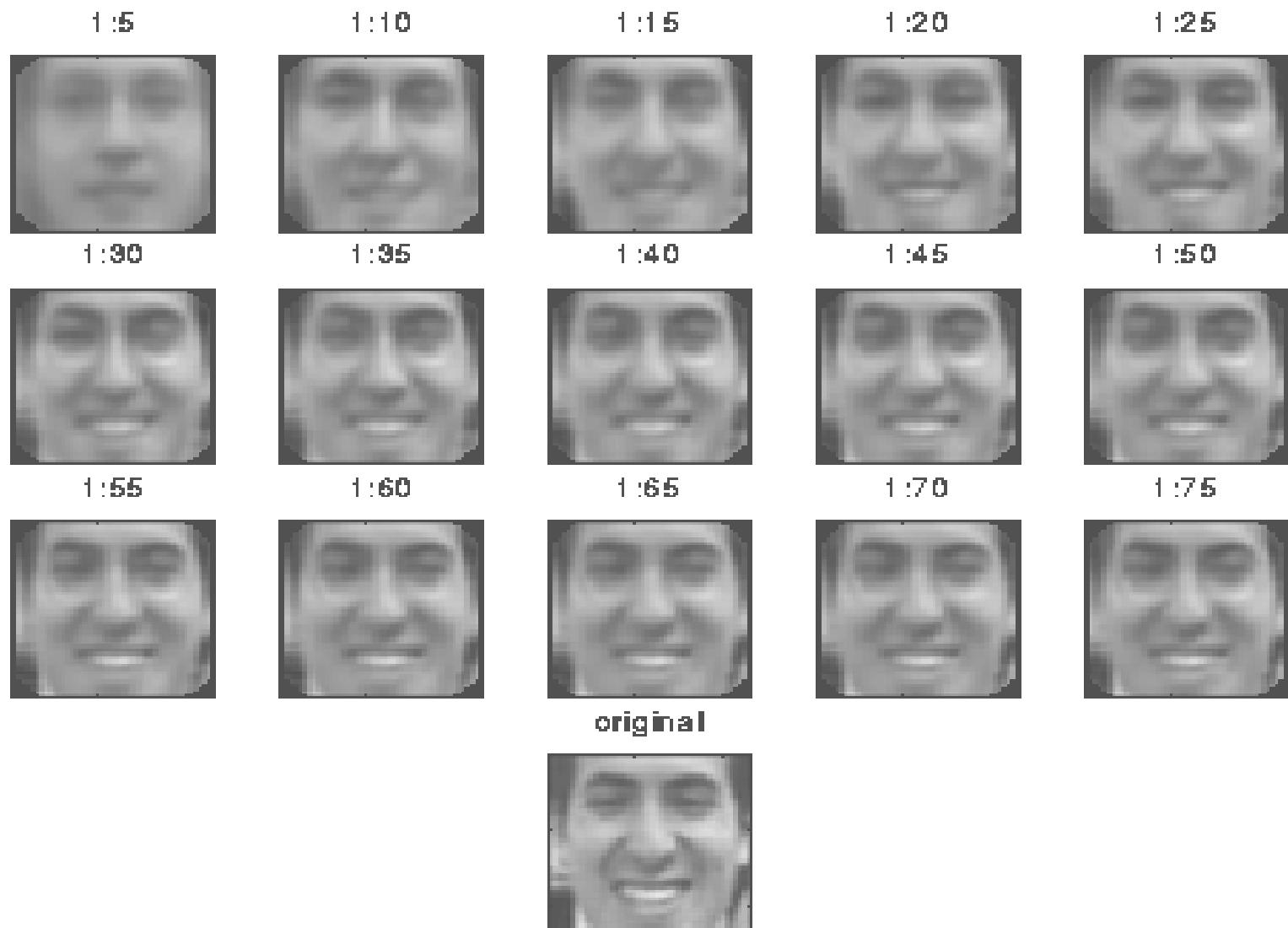






**The first 20 eigen faces with
the highest eigen values**

**Eigenfaces with eigenvalues
ranked from 141 to 160**



Faces reconstructed using eigenfaces with high eigenvalues. The label above each face is the range of eigenfaces used.

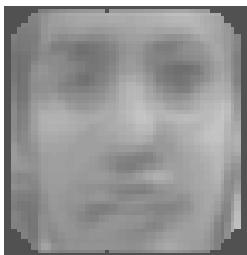
11:60



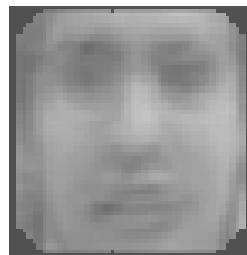
21:70



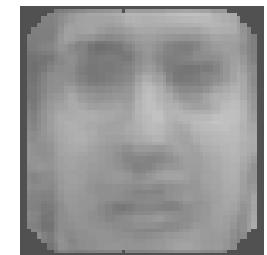
31:80



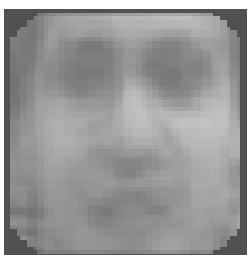
41:90



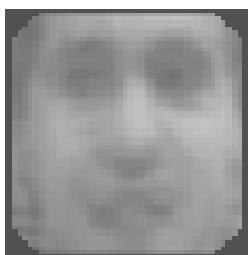
51:100



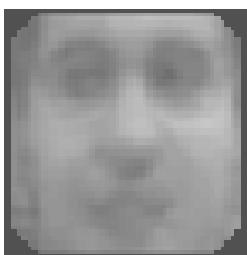
61:110



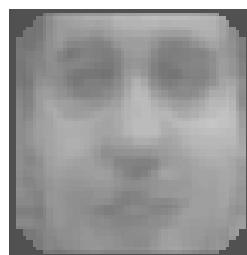
71:120



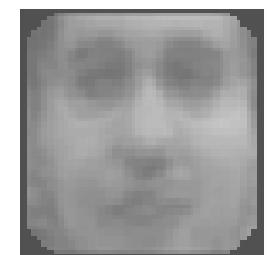
81:130



91:140



101:150



original



Faces reconstructed using eigenfaces with low eigenvalues. The label above each face is the range of eigenfaces used.

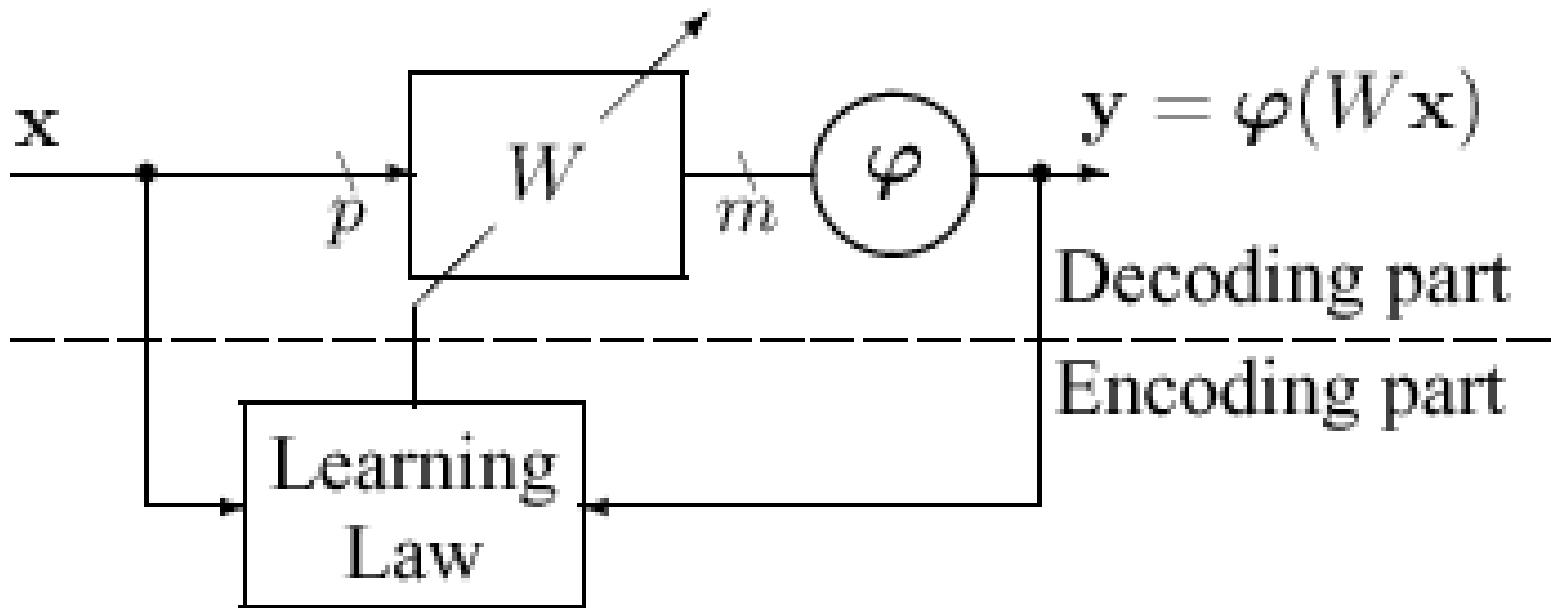
Perhitungan PCA

$$C_B = Cov(B) = \frac{(B - \bar{B}) * (B - \bar{B})^T}{N - 1}$$

$$C_B v_i = L_i v_i \quad , i = 1, 2, 3, \dots, n$$

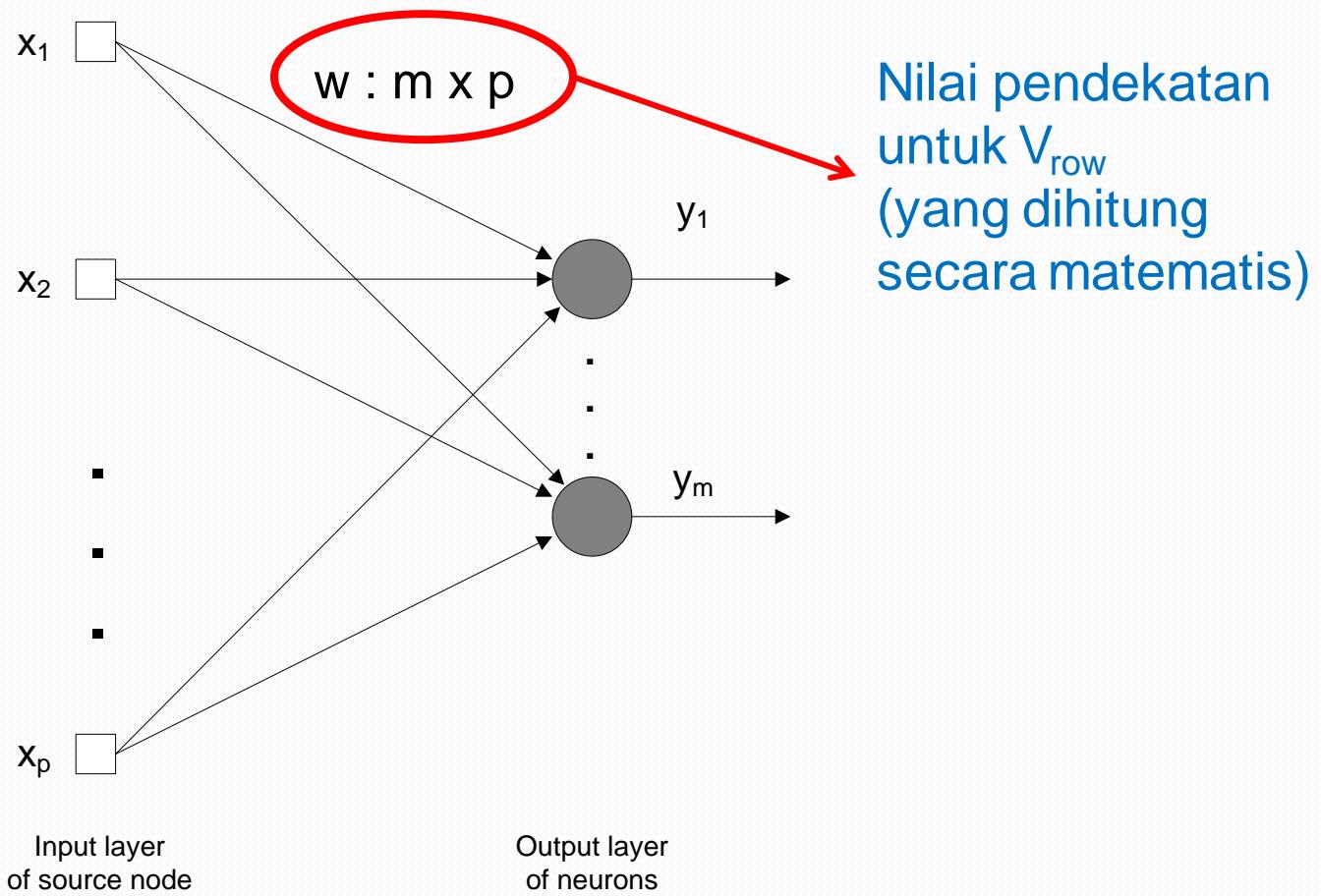
$$| C_B - L_i I |$$

- Sejuta citra @ 800 x 600 pix, kompleksitas & memory?
- Kompleksitas tinggi → prosesnya lama
- Memory besar
- Solusinya?
- ANN: Hebbian Learning



- Decoding: matrix W berukuran $m \times p$
- Satu baris *weight vector* = satu neuron

Arsitektur Hebbian Network



Hebbian-Based PCA

- Generalized Hebbian Algorithm (GHA)
- Adaptive Principal Components Extraction (APEX)

Generalized Hebbian Algorithm (GHA)

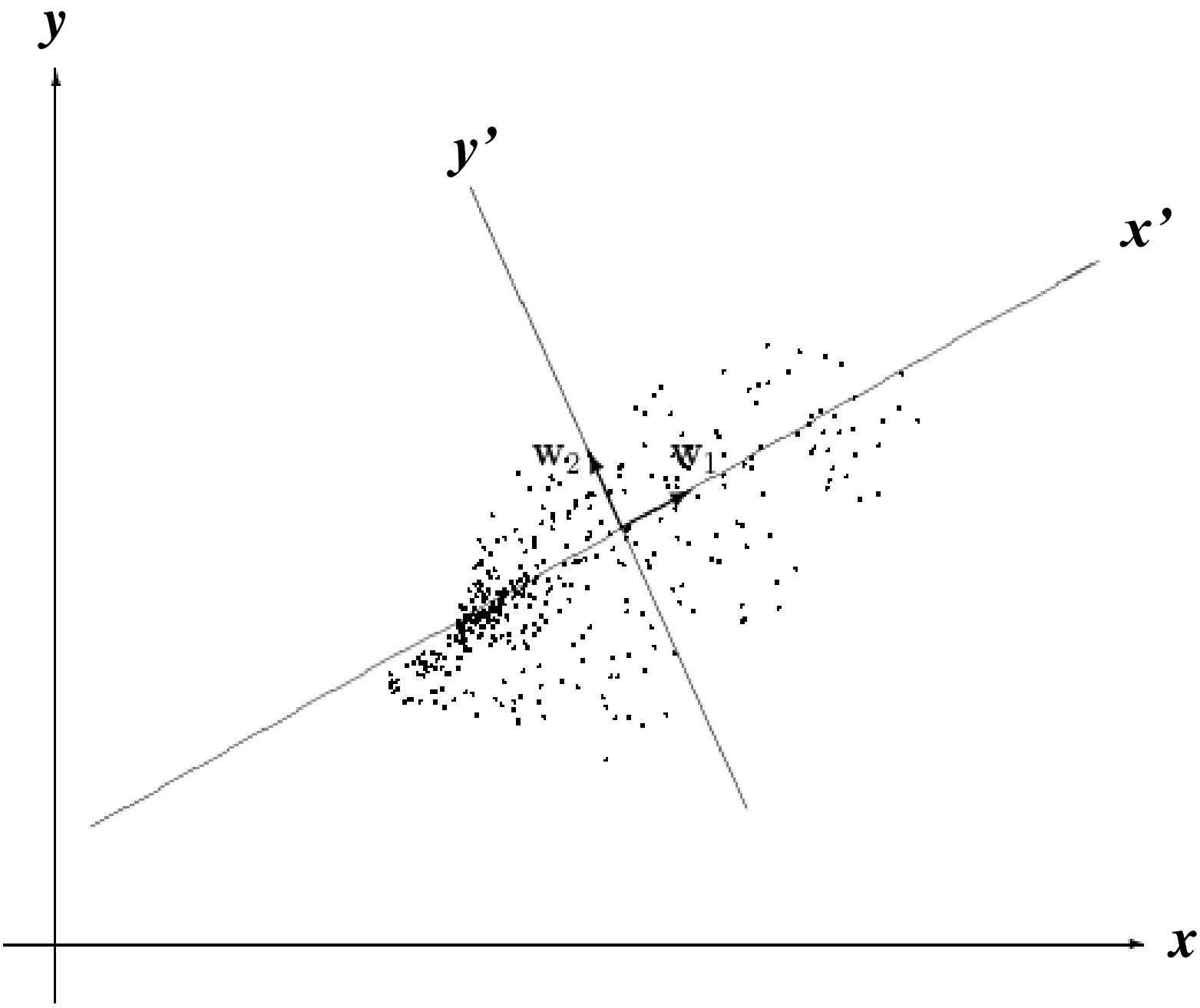
1. Initialize the weights to a small random values at $n = 1$.
Assign a small positive value to the learning-rate.
2. Compute

$$y_j(n) = \sum_{i=1}^m w_{ji}(n)x_i(n),$$

$$\Delta w_{ji}(n) = \eta \left[y_j(n)x_i(n) - y_j(n) \sum_{k=1}^j w_{ki}(n)y_k(n) \right],$$

for $j = 1, 2, \dots, l$, and $i = 1, 2, \dots, m..$

3. Increment n by 1, and go to step 2. Repeat until all synaptic weights reach steady-state values ($\|w\| \rightarrow 1$).

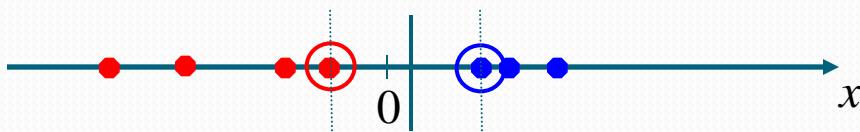


Real Problems

- Reduksi dimensi
 - Ekstraksi fitur: PCA, ICA, Fourier, dsb.
 - Klasifikasi menjadi lebih mudah
- Penambahan dimensi
 - Support Vector Machine (SVM)
 - Fungsi Kernel
 - Klasifikasi menjadi lebih mudah
- Reduksi dan Penambahan

SVM

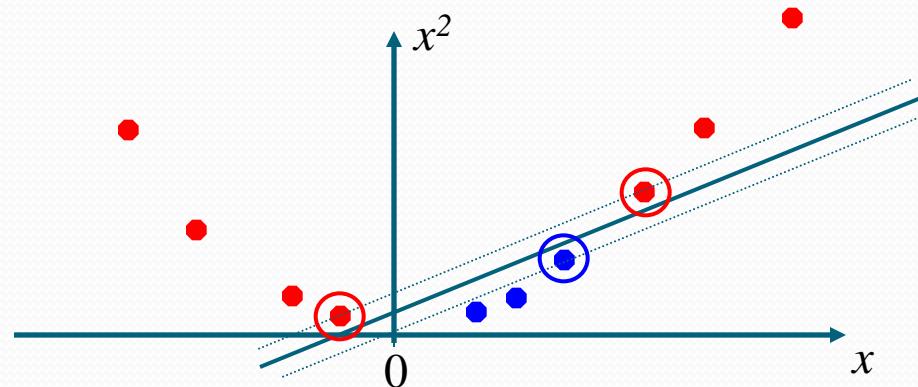
- Datasets yang *linearly separable* masih bisa diselesaikan:



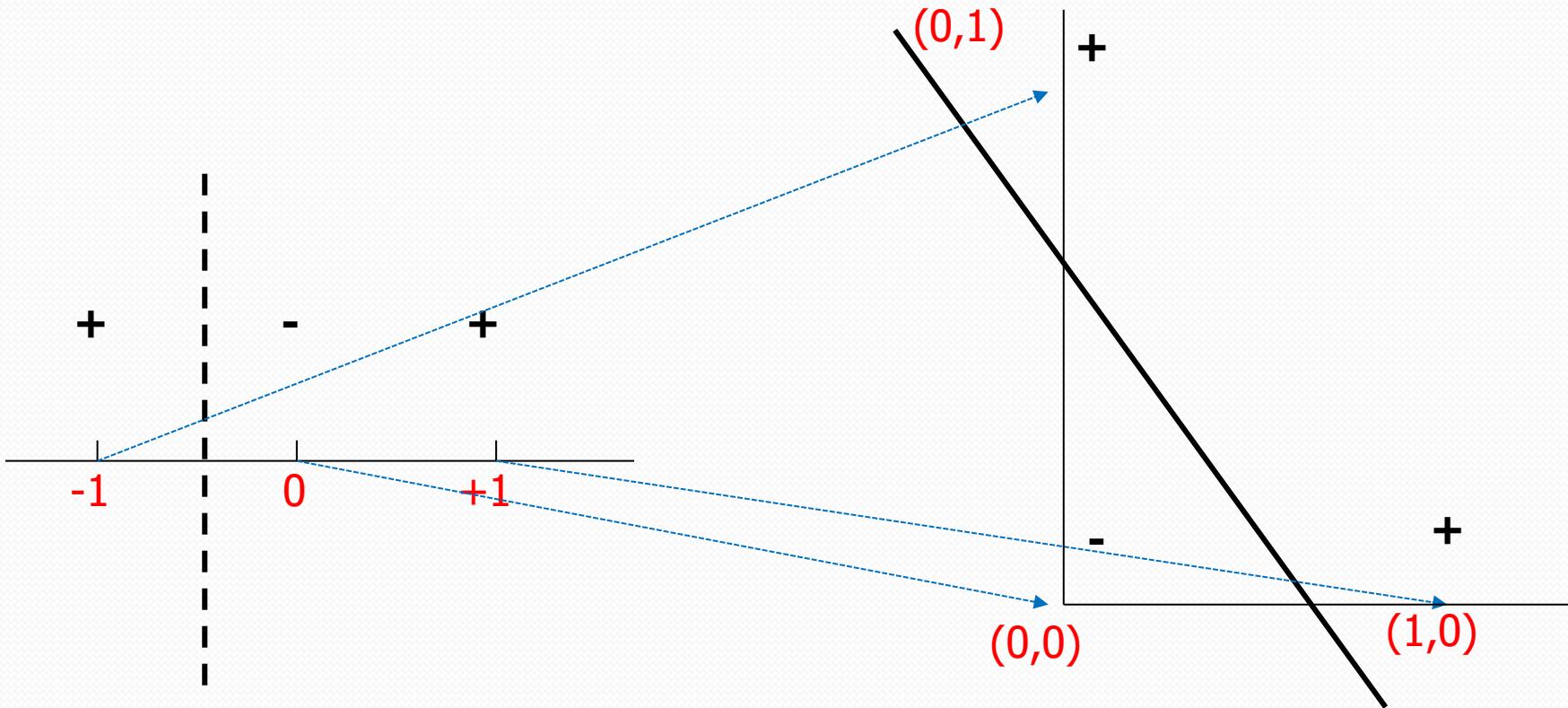
- Tetapi, apa yang bisa dilakukan jika dataset-nya tidak *linearly separable*?



- Petakan dataset ke ruang berdimensi lebih tinggi:



SVM



Strategi penggunaan ANN

- **Cara memandang masalah:**
 - Klasifikasi
 - Sekuriti
 - Prediksi
 - Optimasi
- **Teknik learning:** Supervised/Unsupervised
- **Desain Arsitektur**
 - Jumlah layer
 - Jumlah neuron
 - Pemetaan output
- **Strategi learning**
 - Penyiapan data: filterisasi data, pembagian data (training, validasi, test)
 - Setting parameter: inisialisasi, laju belajar, dsb.
 - Penghentian learning

Case 1: Spam Filtering

- KDD: Knowledge Discovery from Data
- International competition about Personalised Spam Filtering held in Humboldt-Universität zu Berlin, Germany
- Find set of rules or strategy to distinguish an email as spam or not

From: DOUGLAS M ROBIN douglas_markrobin@yahoo.co.uk
Subject: COMPLETE THE FORM WITH YOUR PASSPORT PHOTO ATTACHED
The National Lottery
PAYMENT/PROCESSING OFFICE, LONDON, UK.
3240 RIDGE-WAY,LONDON NW71RN. 00447040112422

Batch/074/05/ZY3
Ticket/ 5647600545188
Serial No /5073-11

Dear (),

I acknowledge the receipt of your mail, as regard your request the reason is that over the years we have had cases of double claim of winnings and with the help of the verification form its earlier to detect the winners in various category.

Your information is need to process other vital document before your cash prize can be release to you for claim.

I need those vital information alongside passport photo to proceed with the processing of your winnings.I need urgent response in 24 hrs because you have less 2 weeks .

Regards,
Douglas Robin.
(FIDUCIARY OFFICER)

Data

- 4000 emails as training set for learning. 2000 spam and 2000 non-spam.
- 4000 emails as tune set (validation set). 2000 spam and 2000 non-spam.
- 2500 emails as validation set. This data is from a user inbox containing 1250 spam and 1250 non-spam.
- 7500 emails as test set, without label of spam nor non-spam. The data are from 3 inboxes. Each box contains 2500 email, 1250 spam and 1250 non-spam.

Rules of the game

- All 18.000 emails are encoded as numerical data
- Firstly, build a wordlist atau dictionary from all emails and produce a dictionary of 200.000 words
- Secondly, each email is parsed and each word is replaced by the suitable index of word in the dictionary and calculate the word frequency
- Email spam is labeled by 1, email non-spam -1, and unknown email 0.

Example: Wordlist (Dictionary)

No	Word
1	attached
2	document
3	form
4	lottery
5	national
6	open
7	passport
8	ticket
...	...
200.000	urgent

Example: Spam Email

- 1 35:1 73:1 77:1 206:1 16176:1
- 1 2058:1 27162:1 49588:1
- 1 9:3 94:1 109:1 163:1 405:1 406:1 415:2 416:1
435:3 436:3 437:4 440:4 450:3 456:1 457:1
461:1 466:1 467:1 477:1 478:1 751:1 1015:1
1034:14 1041:1 1216:1 1226:1 1231:1 1666:1
2344:1 2345:1 2505:1 2528:1 3498:1 4339:1
4463:1 7480:1 8143:1 15050:1 17176:1 19051:1
20895:1 22963:1 35908:1 48253:1 49469:1
60004:1 78684:1 84924:1 85550:1 93429:1
95839:1 106782:1 106783:4 106784:2 106785:2
106786:1 106788:1 106802:1 106803:1

Example: Non-Spam Email

- **-1** 9:1 82:1 92:1 104:1 231:1 308:1 338:1 351:1
390:1 440:2 693:1 933:1 975:1 984:1 1631:1
2404:2 2560:2 2589:2 3361:1 3630:1 4042:1
4059:1 6515:1 7851:1 8762:1 10427:1 16178:1
37517:1 44973:1 53347:1 109089:2 109090:1
110944:1 111668:1 133323:1 140060:1 155590:1

Example: Unlabeled Email

- o 94:1 204:1 257:1 582:1 4898:1 6371:1

Classification technique?

- ID₃
- Bayesian Learning
- Genetic Algorithm
- Support Vector Machine
- ANN: MLP
- Or other techniques?

Feature Extraction?

- Information Gain
- PCA
- ICA
- Atau teknik lainnya

Training Set

Email ke-	K ₁	K ₂	K ₃	K ₄	K ₅	K ₆	...	K ₂₀₀₀₀₀	Kelas
1	3	9	0	0	0	2	...	1	1
2	7	3	0	2	0	1	...	0	-1
3	2	0	17	1	0	8	...	0	1
4	9	2	4	16	5	7	...	8	1
5	1	0	2	6	0	5	...	16	-1
6	7	0	0	0	0	0	...	0	1
7	0	3	0	0	0	8	...	0	-1
8	5	12	3	1	0	0	...	0	1
9	6	8	0	18	0	0	...	5	1
...
4000	8	2	0	23	0	1	...	9	-1

Information Gain (IG)

- Untuk memilih kata-kata yang paling efektif dalam mengklasifikasikan data (*spam* atau *non-spam*).
- IG digunakan untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan data dengan cara:
 1. Hitung IG dari semua kata yang ada (sekitar 200.000)
 2. Urutkan kata-kata tersebut berdasarkan IG-nya, dari yang terbesar sampai yang terkecil
 3. Ambil sejumlah kata (yang IG-nya paling besar)

Email spam:

1 35:11 73:5 77:9 206:12 16176:6

Look up

Daftar 60 nomor kata yang
Information Gain-nya paling besar

- Jumlah kata?
- Trial & error
- Misal: 60 kata

No. Urut	Nomor kata
1	9
2	35
3	73
4	206
5	1.190
6	2.657
7	13.890
8	16.176
...	
60	199.576

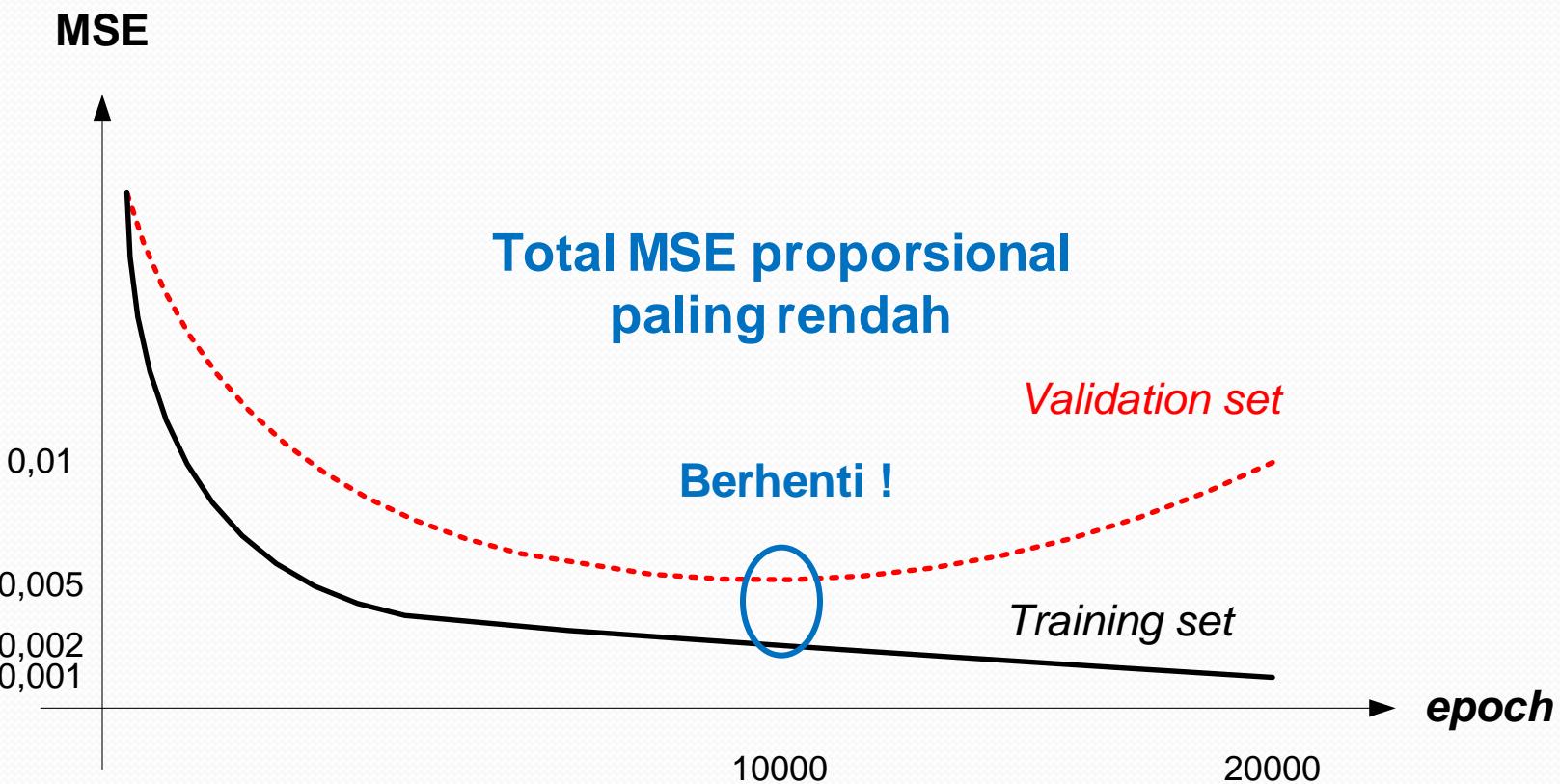
Feature yang dihasilkan:

0	11	5	12	0	0	0	6	0	0	...	0
1	2	3	4	5	6	7	8	9	10	...	60

PCA untuk Ekstraksi Fitur?

- 4.000 records dan 200.000 kolom
- Data: **4.000 x 200.000**
- Matriks Covarian: $200.000 \times 200.000 = 40$ milyar
- Jika 1 elemen = 2 byte, maka memory = **80 GB**
- PCA dengan cara matematis
 - Deterministik
 - Prosesnya lama dan perlu memory besar
- Hebbian Learning
 - Hasilnya berupa pendekatan (tidak pasti)
 - Lebih cepat dan memory kecil

Kapan Menghentikan *Learning*?



Kasus 2: Optimasi Pembelian Buku

- Data Mining Cup 2009
- Di Jerman: 96 ribu judul buku per tahun
- Pembelian yang optimal
 - Judul mana yang sebaiknya dibeli?
 - Berapa kuantitasnya untuk setiap judul tersebut?
- Kasus riil di perusahaan Libri (Jerman)
 - 2.418 cabang
 - Untuk 8 judul tertentu, berapa jumlah yang sebaiknya dibeli oleh setiap cabang?
 - Learning berdasarkan data histori?

Karakteristik Data

Feature	Type	Description	Attributes
ID	Integer	<ul style="list-style-type: none">• Unique location id	random unique key
WGxxxxx	Integer	<ul style="list-style-type: none">• Number of total items sold within 12 months within a category• Categories with 5 digits• First digit: Information about the type of product e.g. hardcover vs. paperback• Second to fifth digit: Hierarchical information about the type of content e.g. second digit <u>Fiction</u> and third digit subcategory <u>Science-Fiction</u>	independent variables
T1...T8	Integer	<ul style="list-style-type: none">• Number of items sold within 12 months per title	target value

Contoh Data Training

Teknik Klasifikasi?

- Fuzzy
- ANN: MLP, CNN, ...
- EAs: GA, ES, GE, ...
- SVM
- ...

Feature Extraction?

- Information Gain
- PCA
- ICA
- ...

IG atau PCA?

- IG
 - Dunia Nyata
 - Menemukan atribut yang paling efektif
- PCA
 - Konversi ke dunia lain
 - Menemukan atribut yang paling efektif
- *Feature Extraction* yang baik:
 - **Inter-class** → jaraknya diperlebar
 - **Intra-class** → jaraknya dipersempit

Lebar

Y

Domain PCA
Sumbu Y Kecualian penting?

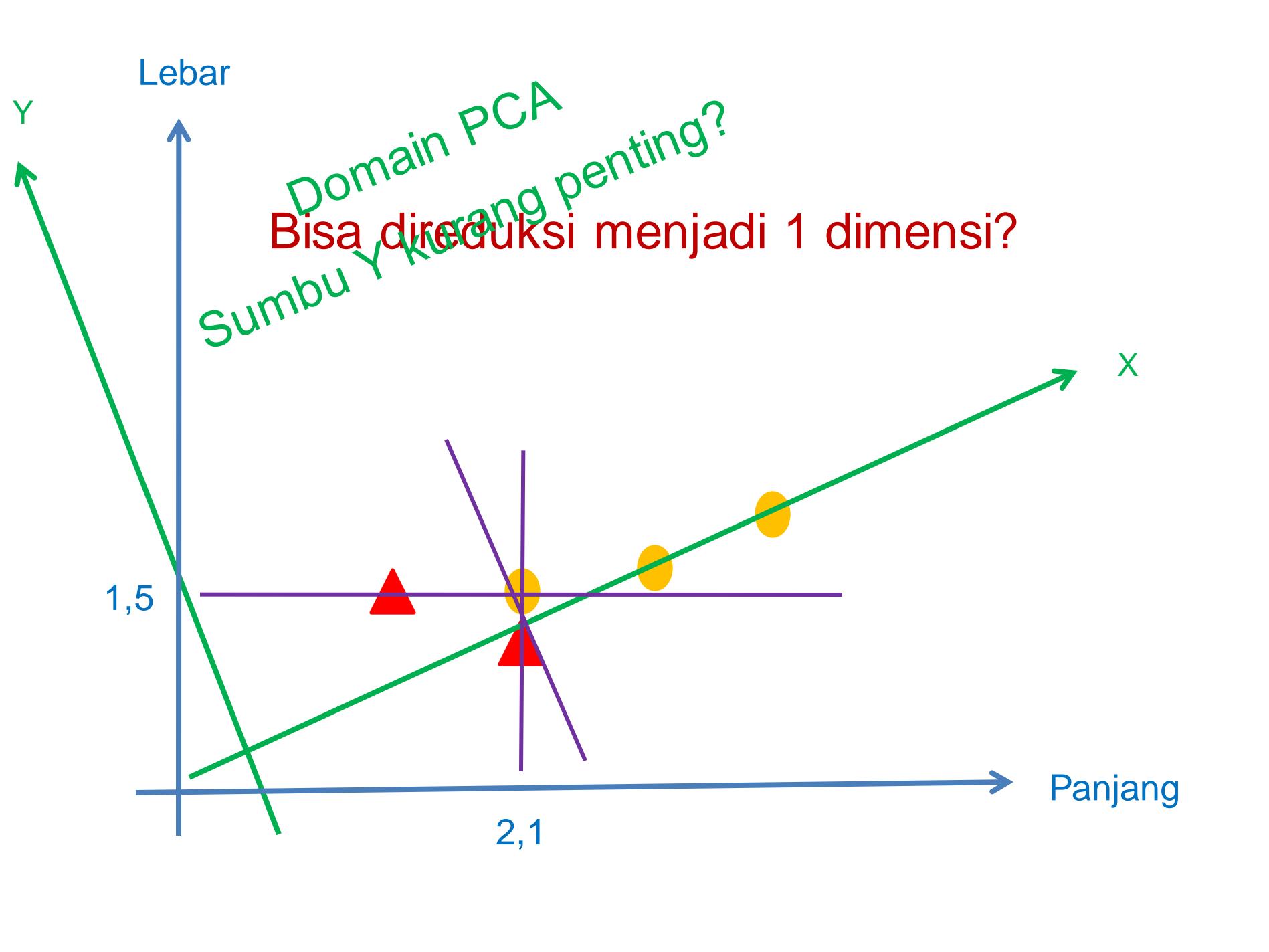
Bisa direduksi menjadi 1 dimensi?

X

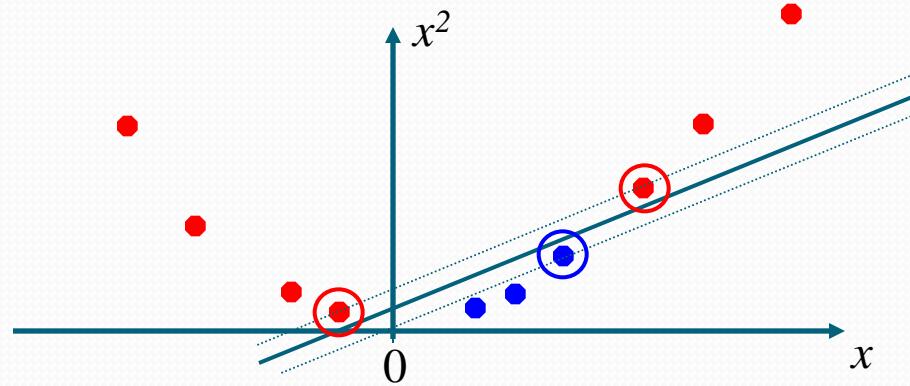
1,5

2,1

Panjang



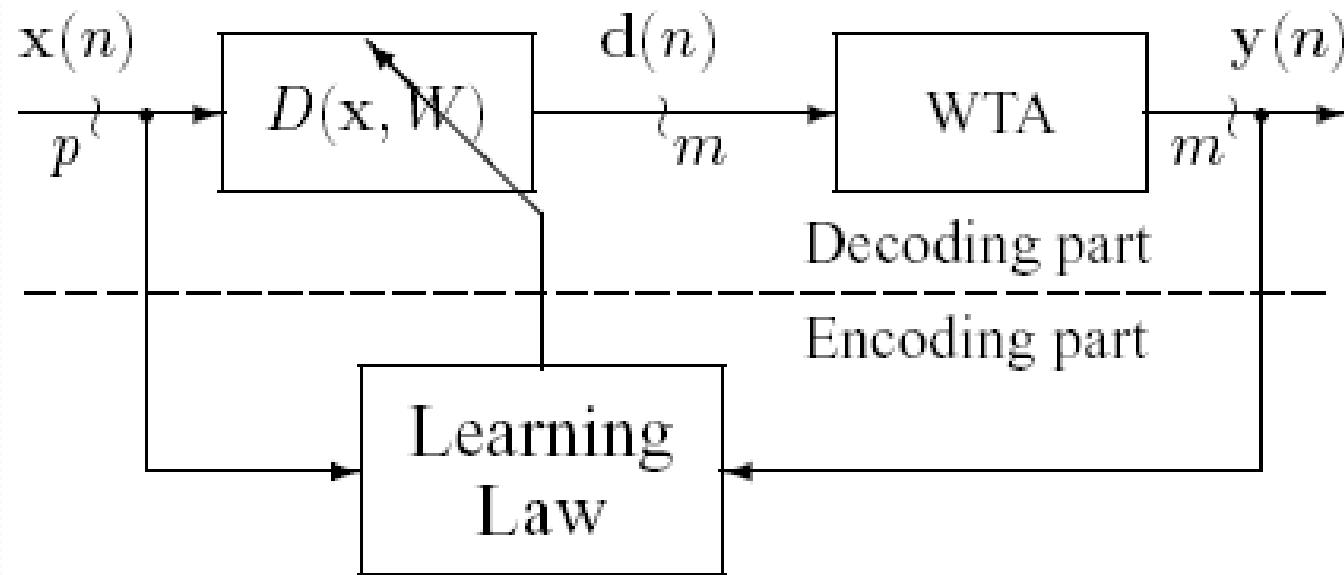
SVM



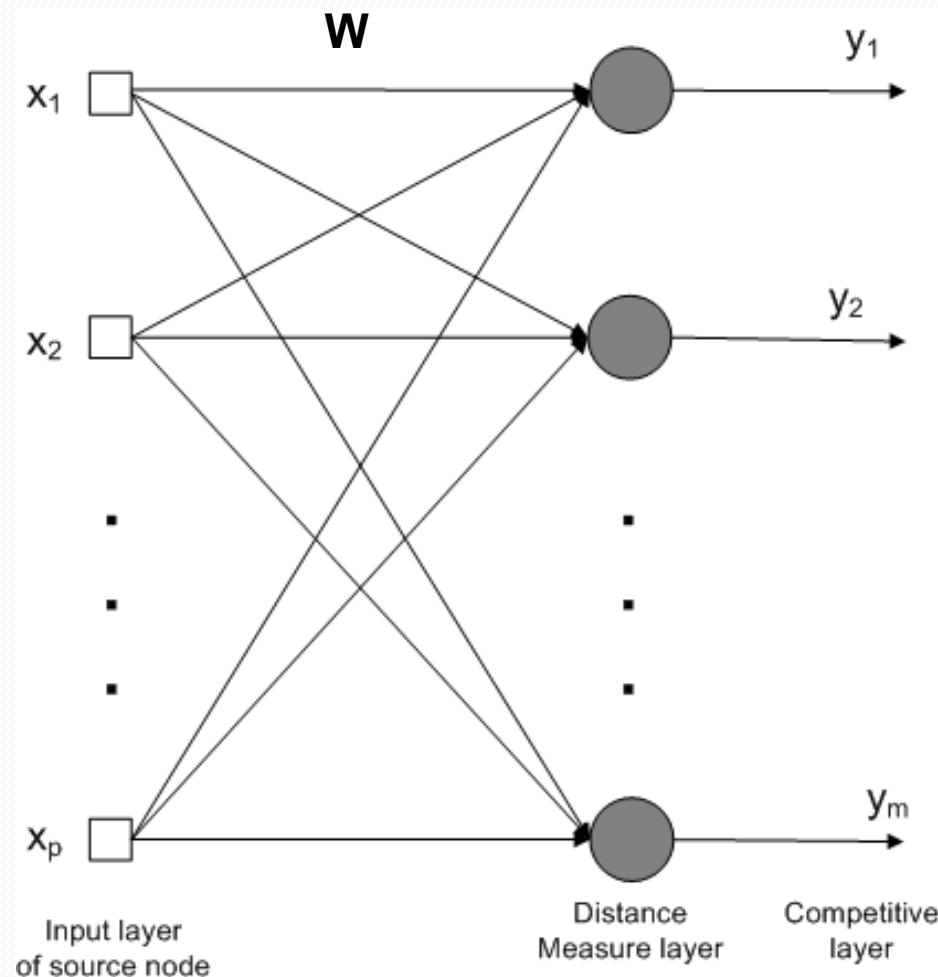
Competitive Neural Networks (CNN)

CNN terdiri dari dua lapis neurons:

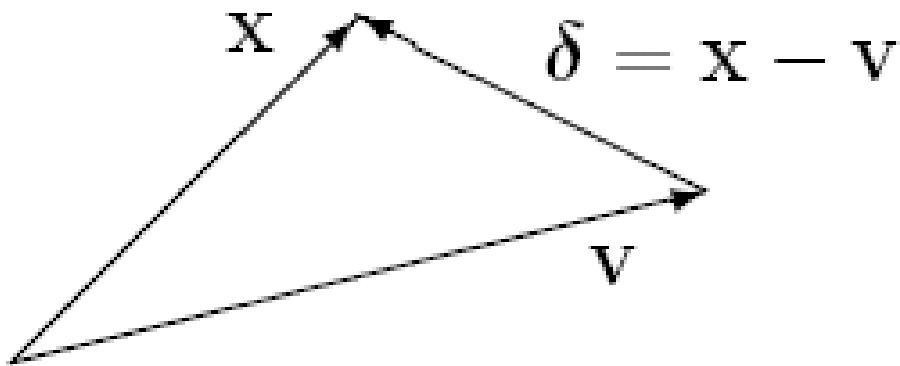
- Distance-measure layer
- Competitive layer atau “Winner-Takes-All” (WTA)



Arsitektur WTA Network



Euclidean norm



$$d = \|\mathbf{x} - \mathbf{v}\| = \|\delta\| = \sqrt{\delta_1^2 + \dots + \delta_p^2} = \sqrt{\delta^T \cdot \delta}$$

The Competitive Layer

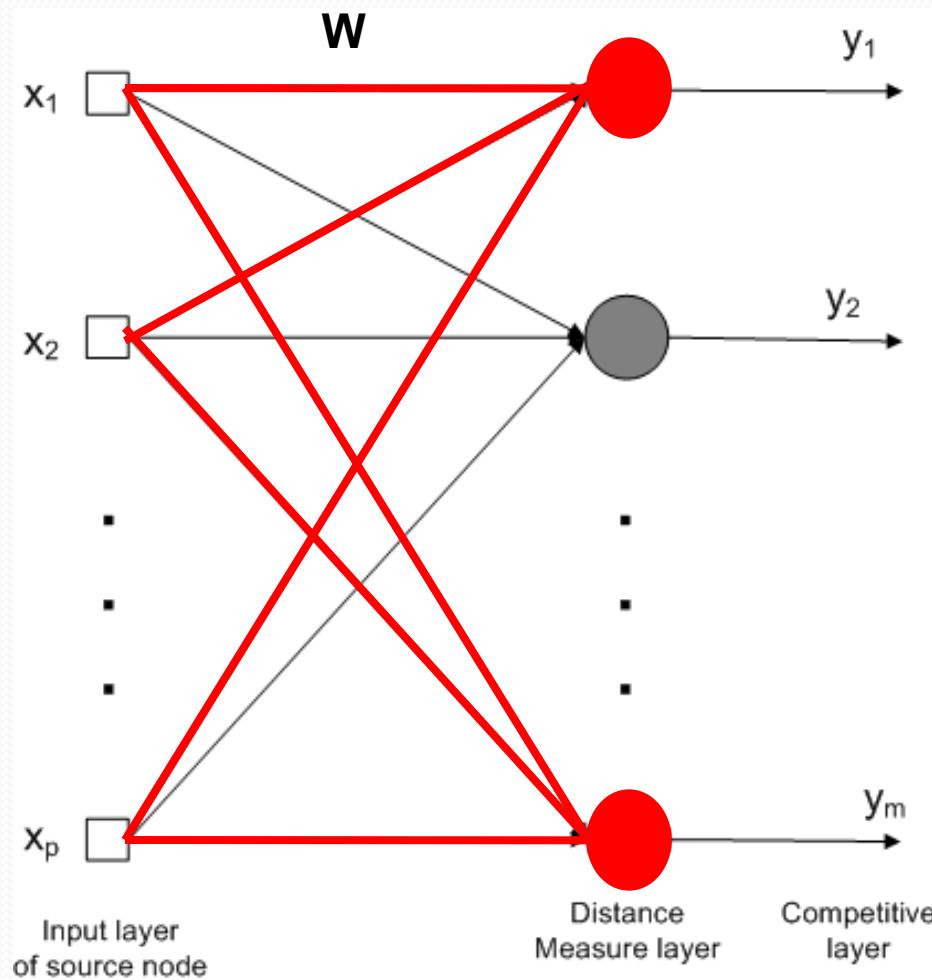
$$y_j = \begin{cases} 1 & \text{if } j = \arg \min_k D(\mathbf{x}, \mathbf{w}_k) \\ 0 & \text{otherwise} \end{cases}$$

Training

Inisialisasi: random Training CNN

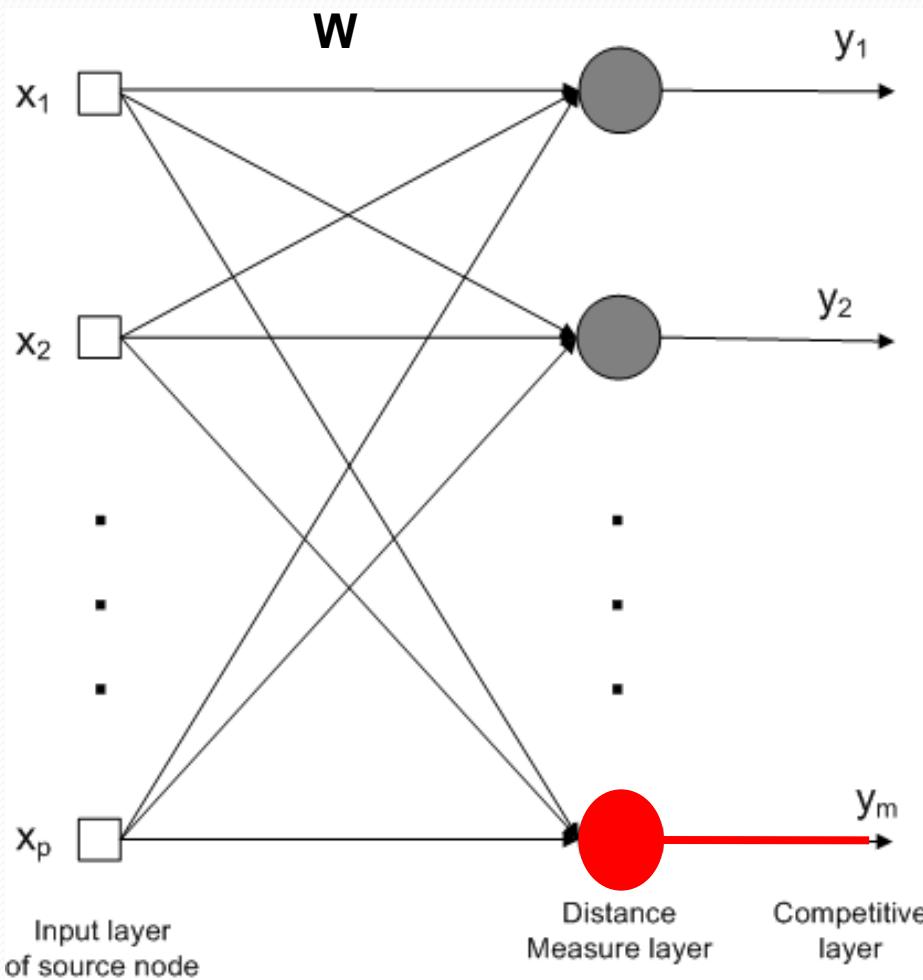
$W \rightarrow$ titik pusat cluster
yang "TEPAT"

Pola 1



Testing

Pola X



Jika pola Y bukan termasuk dalam cluster yang ada, bagaimana cara **menolak**?

% Initialisation of competitive learning

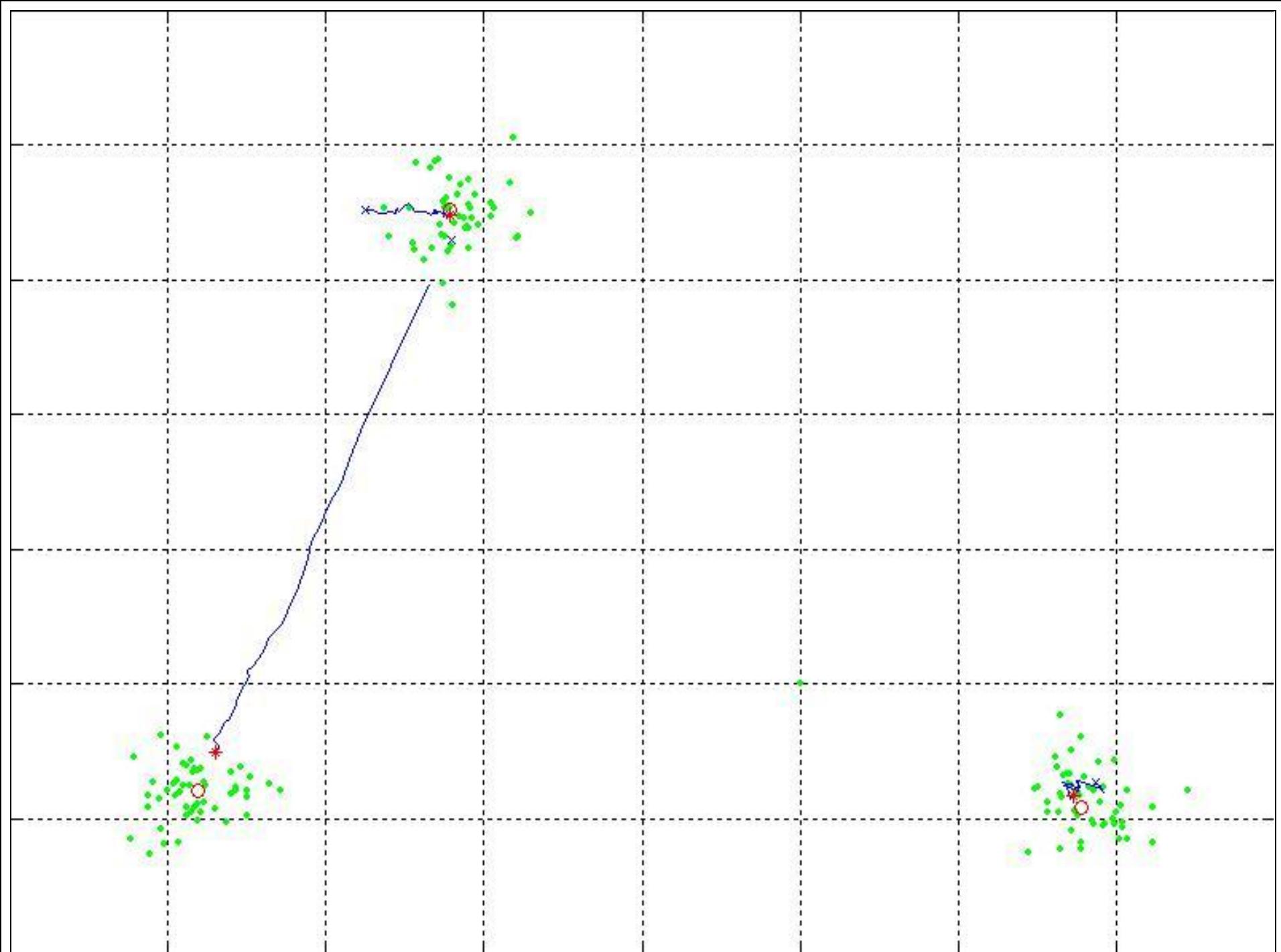
```
p = 2;                                % Dimension of input = p
m = 3;                                % Number of clusters = m
clst = randn(p, m);                   % cluster centroids
Nk = 50;                               % points per cluster
N = m*Nk;                             % total number of points
sprd = 0.1;                            % a relative spread of the Gaussian "blobs"
X = zeros(p,N+m);                     % X is p by m+N input data matrix
wNk = ones(1, Nk);
for k = 1:m                            % generation of m Gaussian "blobs"
    xc = clst(:,k);
    X(:,(1+(k-1)*Nk):(k*Nk))=sprd*randn(p,Nk)+xc(:,wNk);
End
```

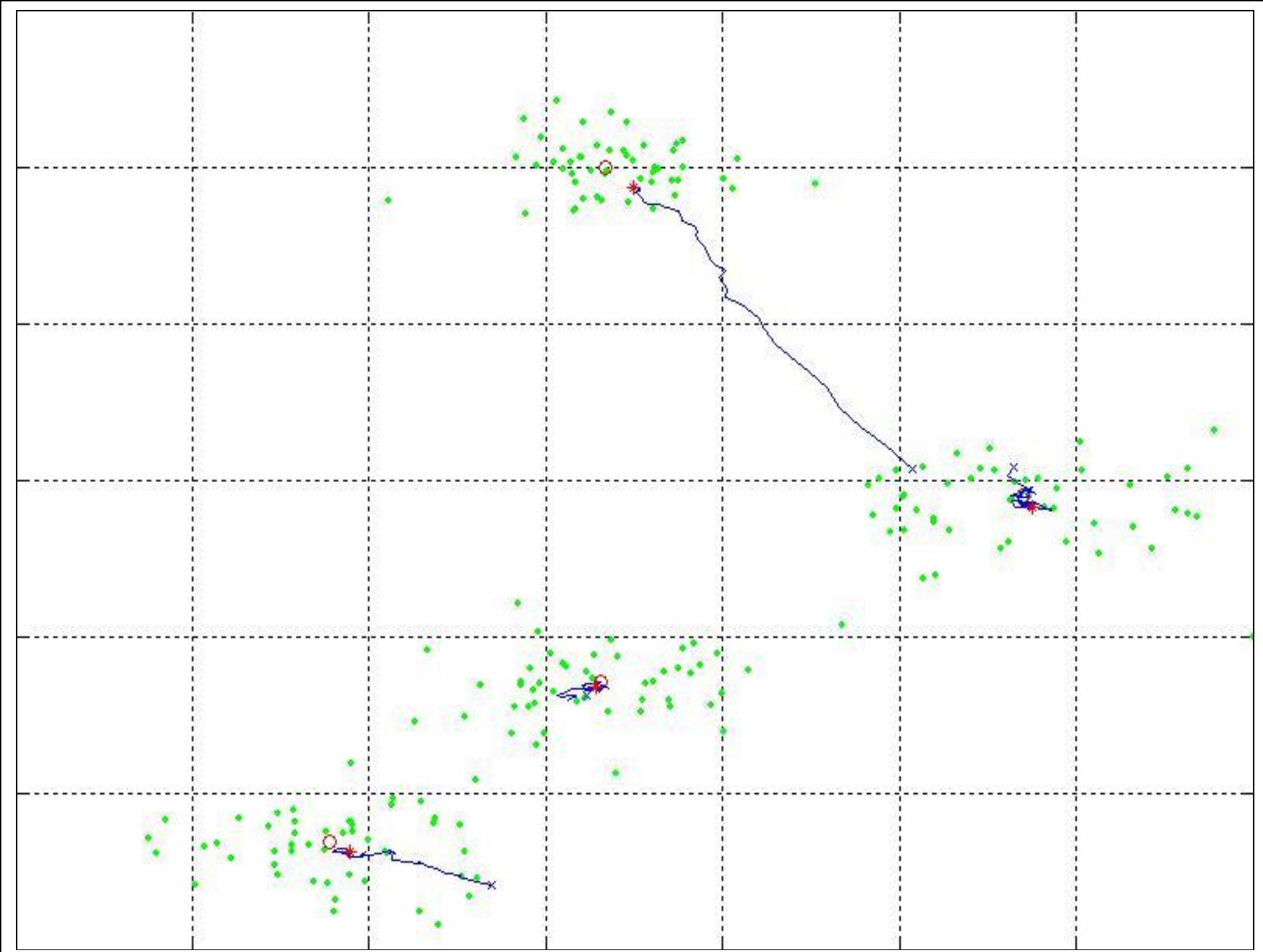
```
[xc k] = sort(rand(1,N+m));
X = X(:, k);                          % input data is shuffled randomly
winit = X(:,1:m)';                   % Initial values of weights
X = X(:,m+1:N+m);                  % Data matrix is p by N
```

% Competitive learning

```
W = winit;
V = zeros(N, m, p);      % to store all weights
V(1,:,:)=W ;
wnm = ones(m,1) ;
eta = 0.08 ;              % learning gain
deta = 1-1/N ;            % learning gain decaying factor
for k = 1:N               % main training loop
    xn = X(:,k)';
    xmw = xn(wnm,:)-W;    % the current vector is compared to all weight vectors
    [win jwin] = min(sum((xmw.^2),2));
    W(jwin,:)=W(jwin,:)+eta*xmw(jwin,:); % update weights of the winning neurons
    V(k,:,:)=W;
    eta = eta*deta;
End

plot (X(1,:),X(2,:),'g.', clst(1,:),clst(2,:),'ro', winit(:,1),winit(:,2),'bx' , ...
V(:,:,1), V(:,:,2), 'b', W(:,1),W(:,2) , 'r*'), grid
```





Data Training 1

- Jumlah data: 10 ribu records
- Dimensi: 3 ribu
- Jumlah kelas: 4
- Semua data memiliki kelas yang valid
- **Metode:** Supervised atau Unsupervised?
- Bisa Supervised maupun Unsupervised

Data Training 2

- Jumlah data: 60 juta records
- Dimensi: 200
- Jumlah kelas: tidak diketahui
- **Metode:** Supervised atau Unsupervised?
- Unsupervised
- Supervised tidak bisa dipakai

Supervised vs Unsupervised

- **Supervised → Klasifikasi** → User mengajari ANN
 - Jumlah kelas **diketahui**
 - **Tersedia** data latih yang **VALID**
- **Unsupervised → Clustering** → ANN memberitahu user
 - Jumlah kelas bisa **tidak diketahui**
 - **Tidak tersedia** data latih yang **VALID**



Menangis?

Berjalan?

Arah?

Berbahasa?

Logika?

Multiple Intelligence!

Supervised dan Unsupervised!!

Molecular Memory

- Setiap sel punya memory?
- Transplantasi organ tubuh???
- “The Eye”

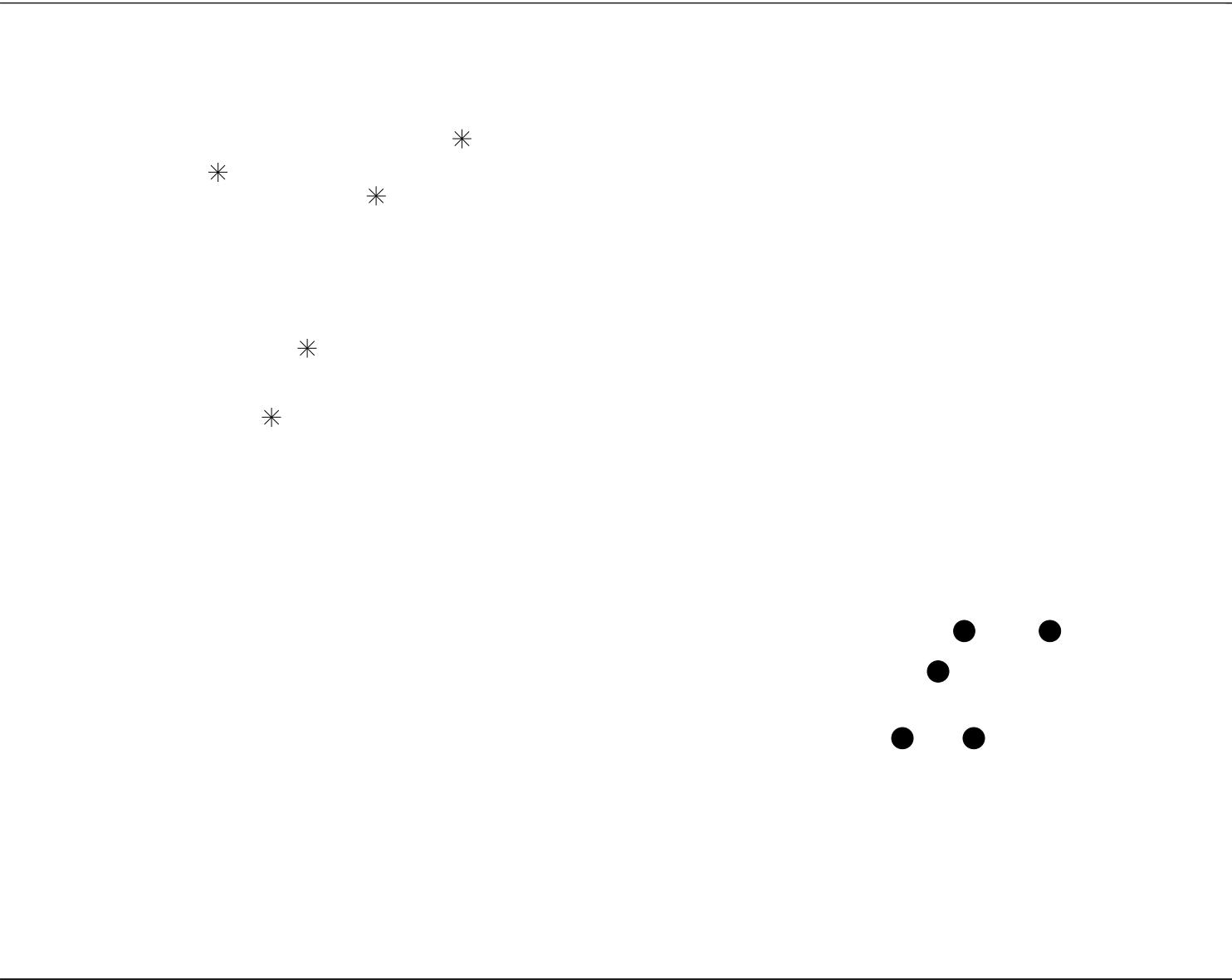
Referensi

- [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.

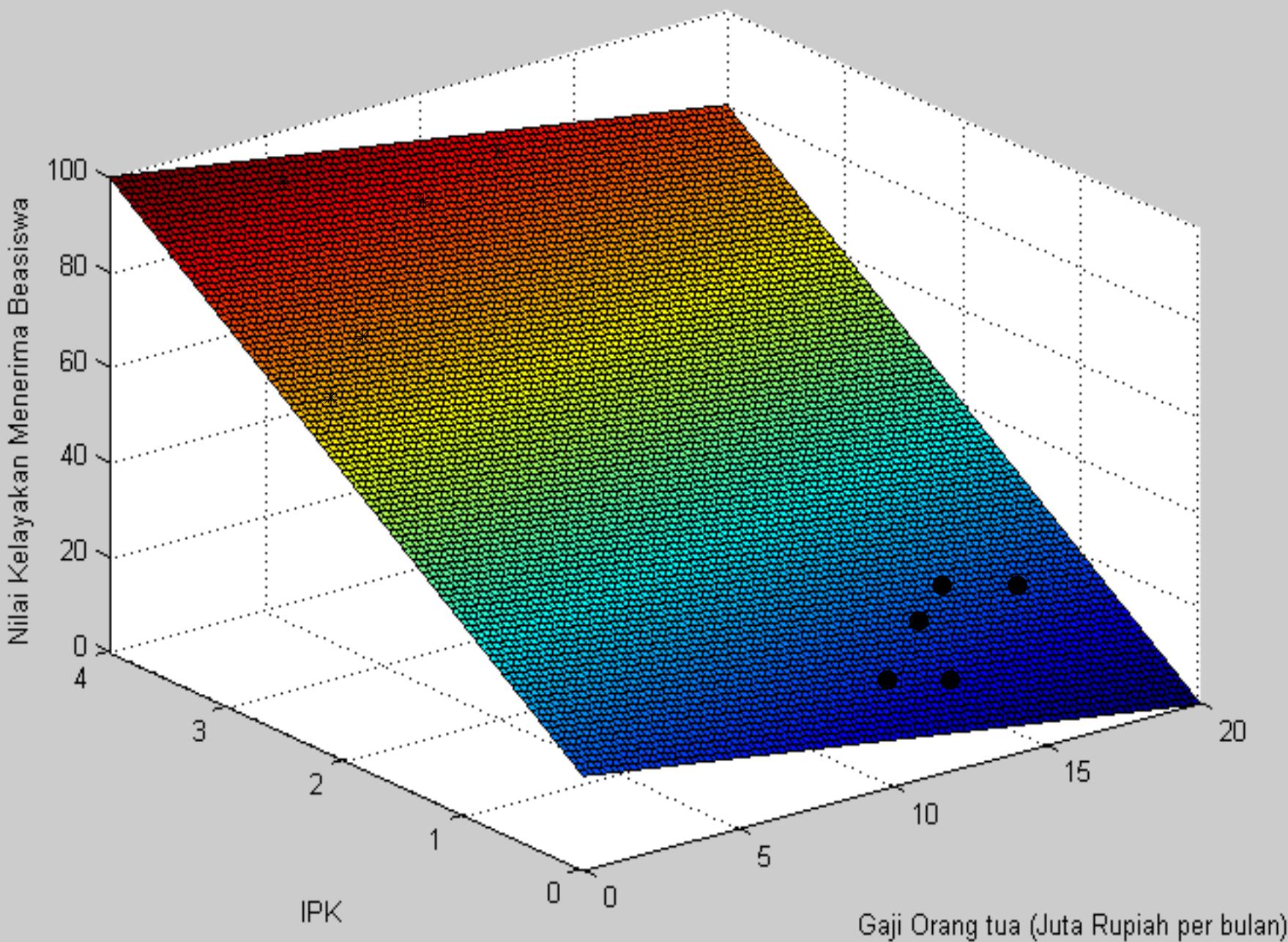
Evolutionary Computation

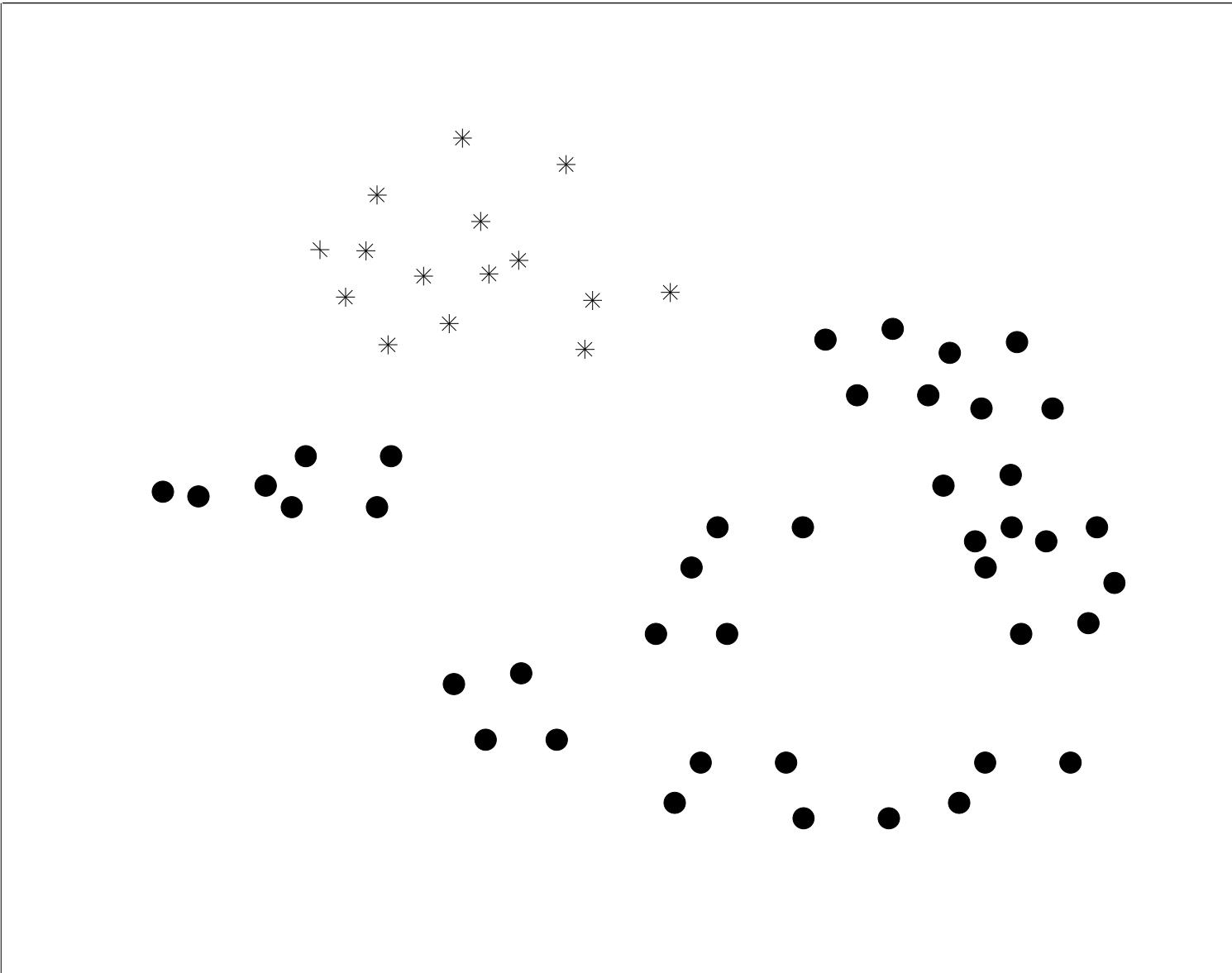
Dr. Suyanto, S.T., M.Sc.
HP/WA: 0812 845 12345

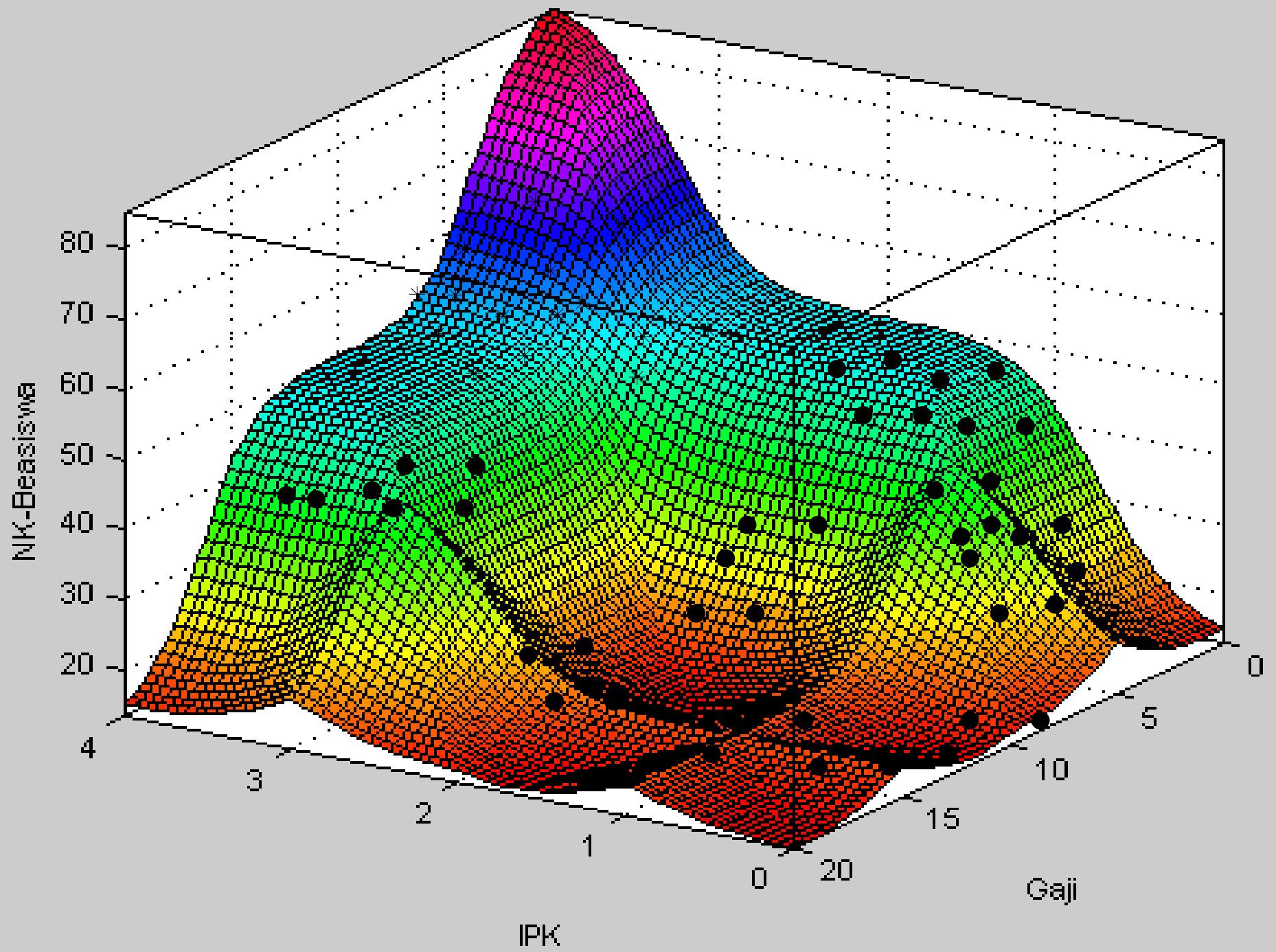
Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

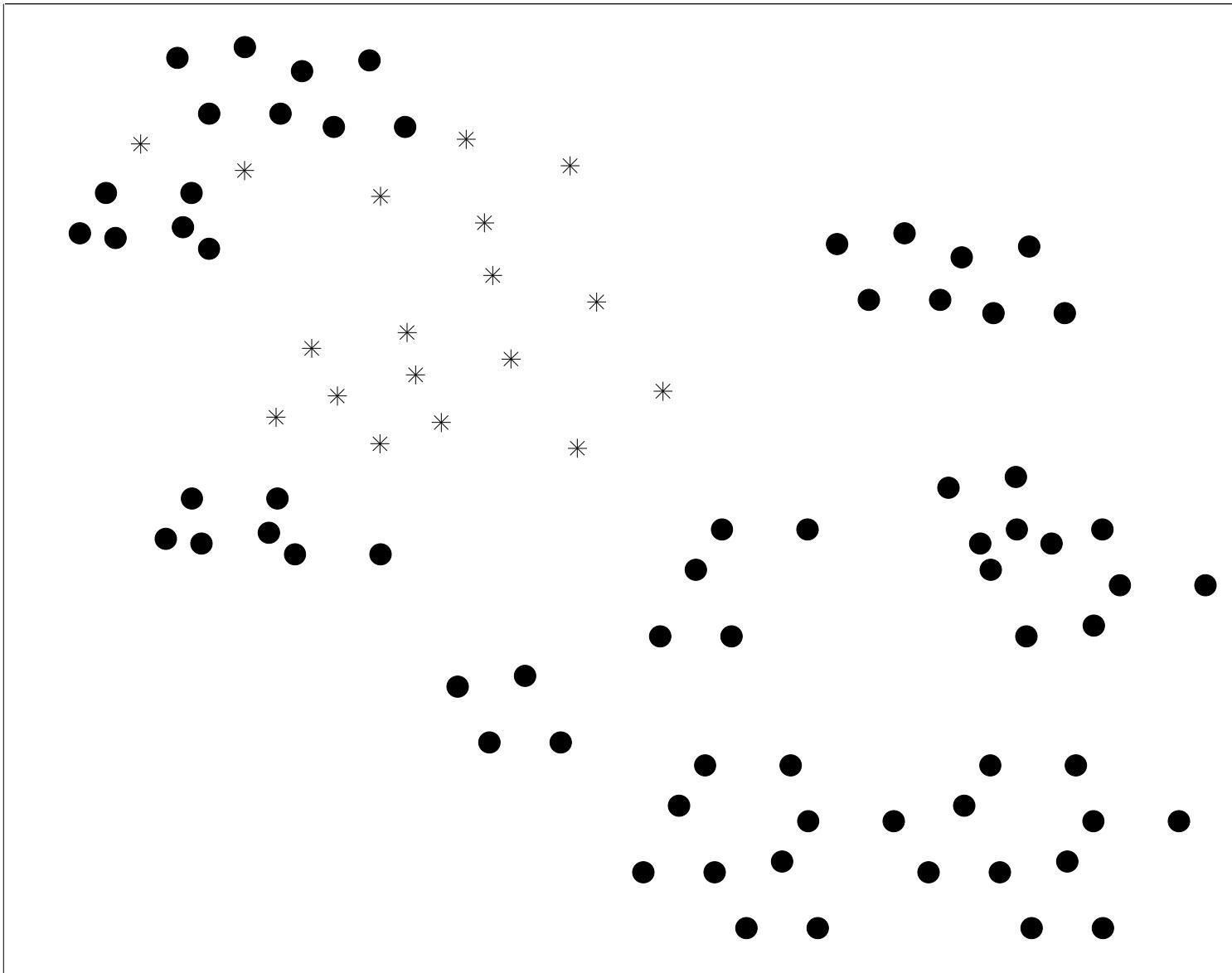


Penghitungan Nilai Kelayakan menggunakan Pembobotan: IPK 80 dan Gaji 20

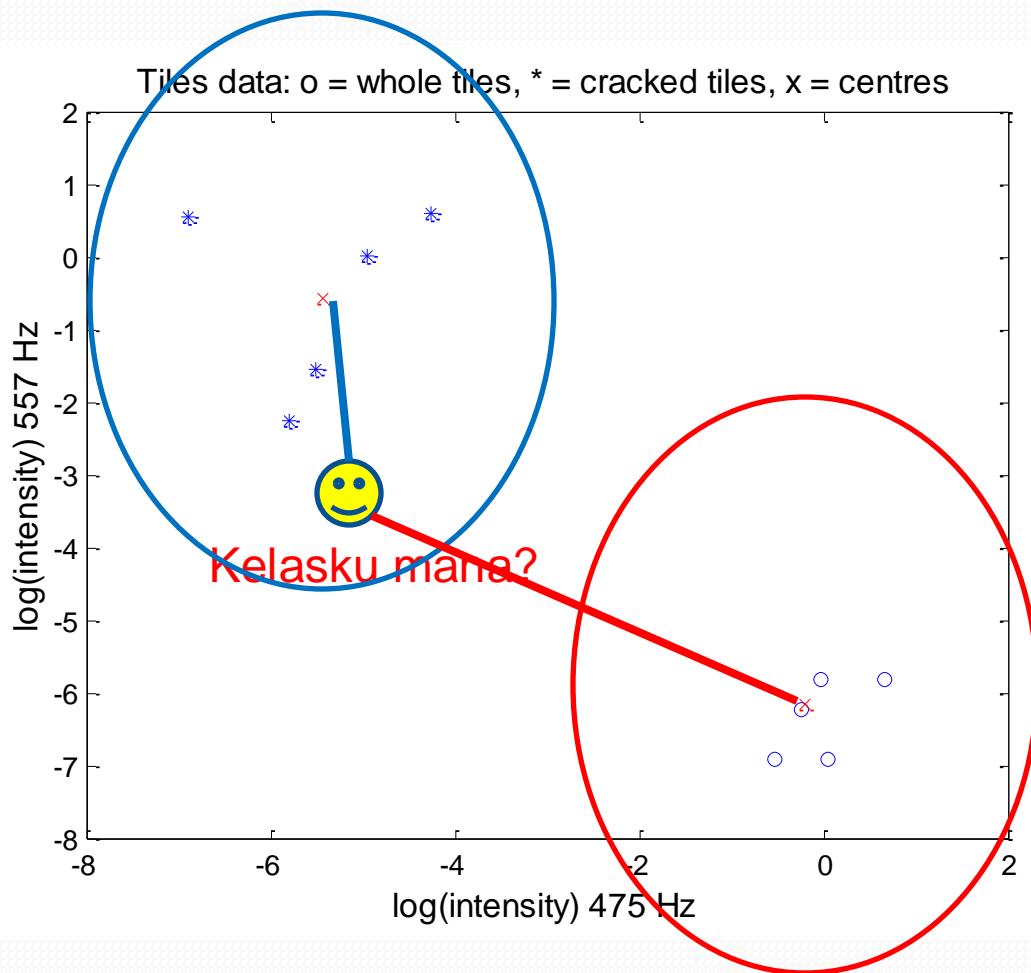




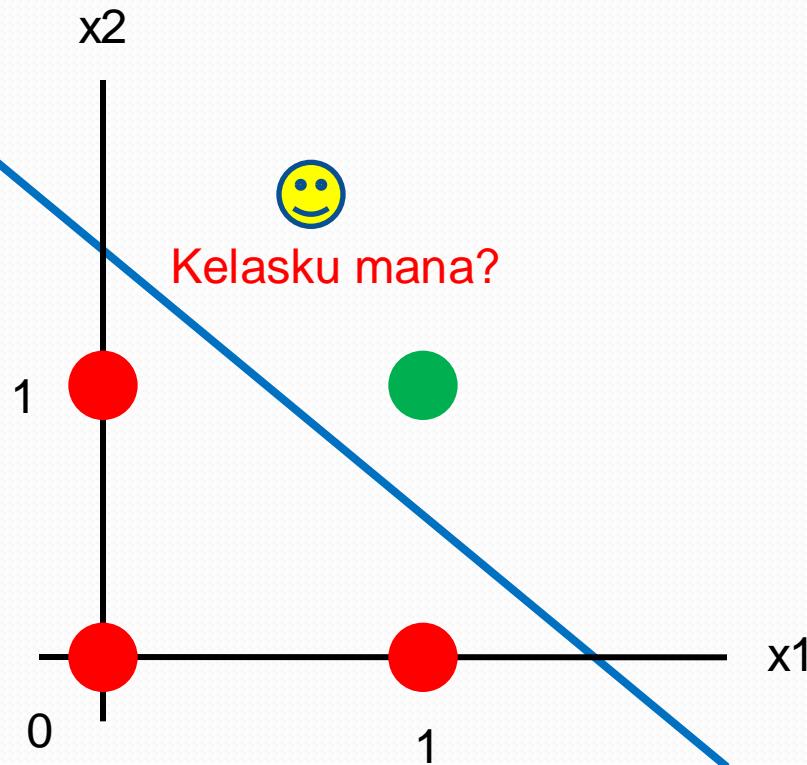




Fuzzy C-Means (FCM)



AND

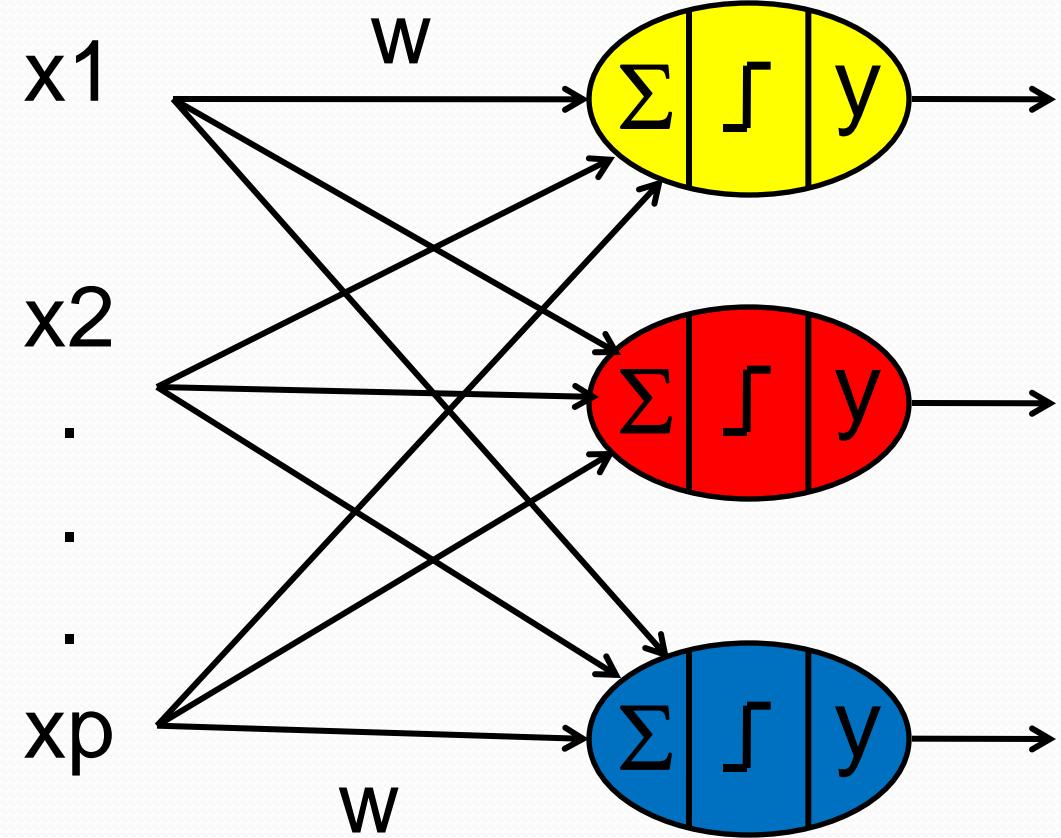
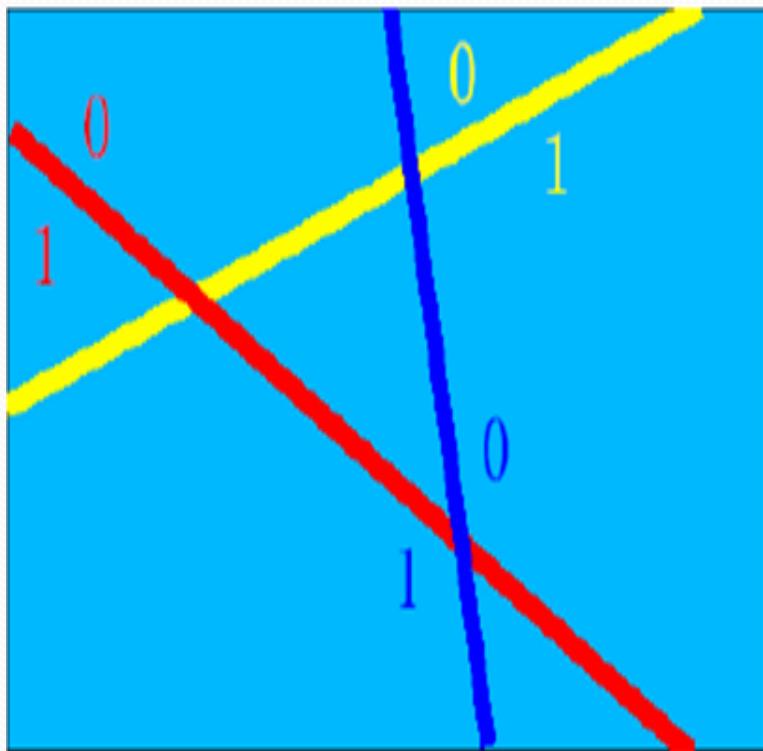


x1	x2	y
0	0	0
0	1	0
1	0	0
1	1	1

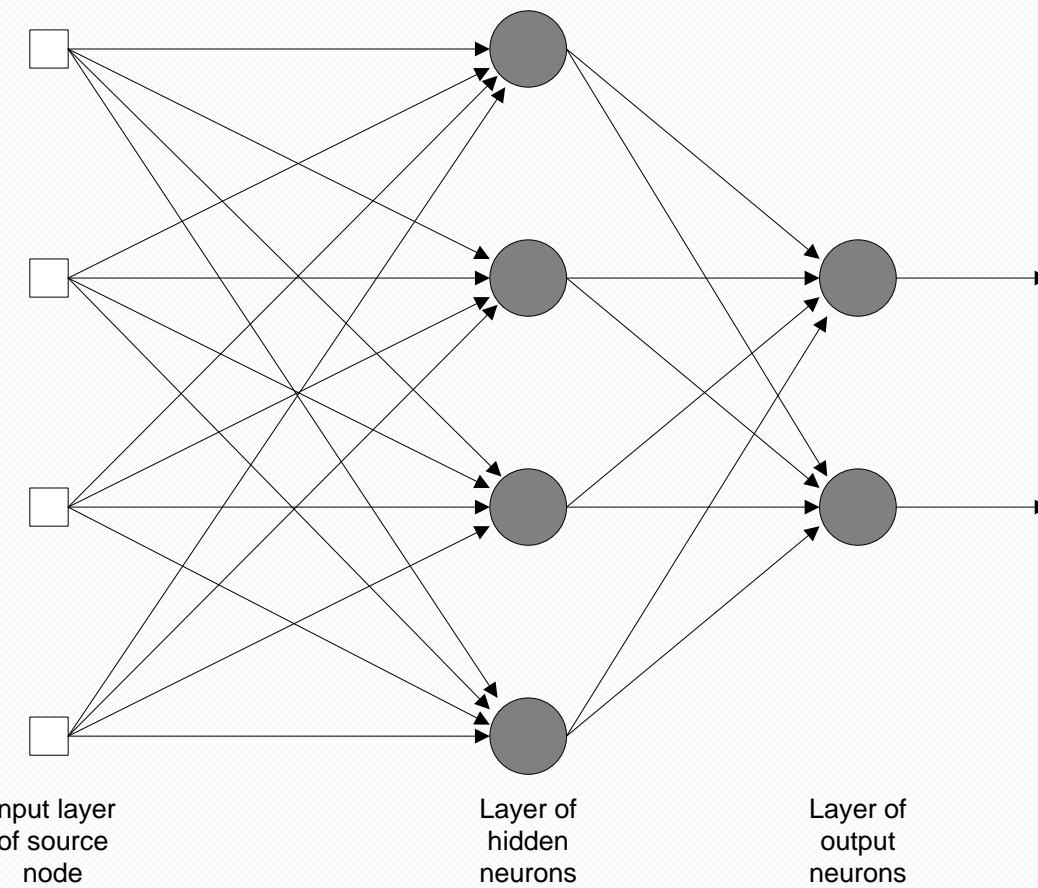
$$x_1 + x_2 - 1,5 = 0$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 - 1,5 = 0$$

Perceptron Network



ANN: MLP



Fuzzy, ANN, dan EC

- **Fuzzy**
 - Reasoning → Logika & Inferensi
 - Kontrol
 - Klasifikasi, prediksi, optimasi, dsb.
- **ANN**
 - Learning → Sistem kerja otak
 - Klasifikasi
 - ~~Prediksi, optimasi, kontrol, dsb.~~
- **Evolutionary Computation**
 - Searching → “Teori Evolusi”
 - Optimasi
 - Prediksi, klasifikasi, kontrol, dsb.

Berbeda konsep,
tetapi saling
mendukung dan
bisa digabungkan

Apa itu EC?

Evolutionary Computation is an abstraction from the theory of **biological evolution** that is used to create **optimization** procedures or methodologies, usually implemented on computers, that are used to solve problems“ [JULo7].

Apa itu EAs?

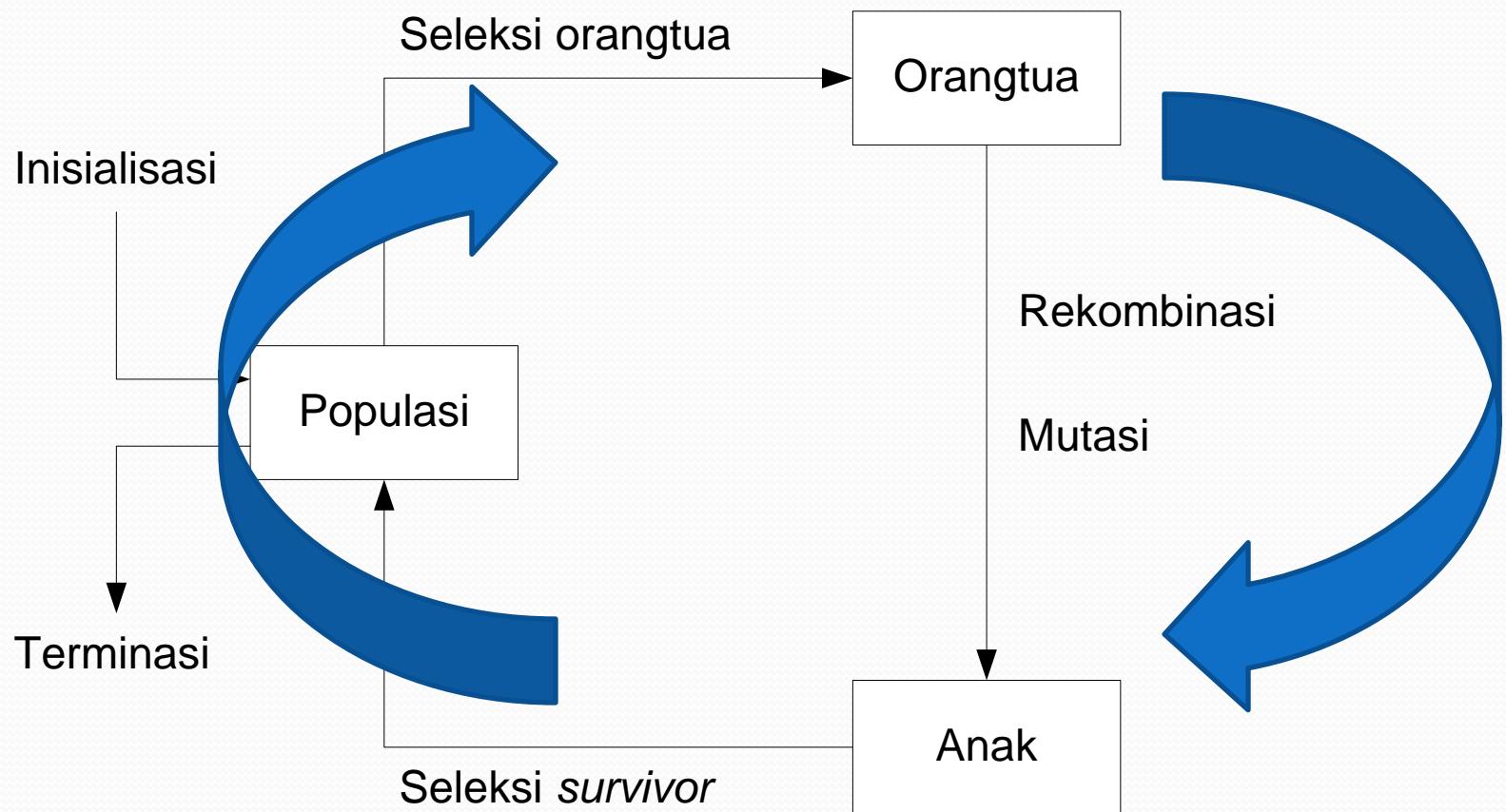
Evolutionary Algorithms are generic, **population-based meta-heuristic** optimization algorithms that use **biology-inspired** mechanisms like **mutation**, **crossover**, **natural selection** and **survival of the fittest**.

EAs = algoritma2 yang mengimplementasikan abstraksi **EC**

Yang termasuk EAs:

1. Genetic Algorithms (GA): binary strings
2. Evolution Strategies (ES): real-valued vectors
3. Evolutionary Programming (EP): finite state machines
4. Genetic Programming (GP): LISP trees
5. Differential Evolution (DE): Perkembangan dari ES
6. Grammatical Evolution (GE) ← Perkembangan GP

Skema umum EAs



Komponen GA

- Inisialisasi Populasi (N kromosom)
- Evaluasi Individu (berbasis fungsi fitness)
- Seleksi Ortu
- Rekombinasi/Cross-over
- Mutasi
- Seleksi Survivor

Bekerja dengan GA

- Buat skema pengkodean Individu → Kromosom
- Bangun fungsi fitness
- Definisikan operator GA

Studi kasus: Minimasi fungsi

Nilai minimum $h = ?$

$$h(x_1, x_2) = x_1^2 + x_2^2$$

$$x_1, x_2 \in [-5,12;5,12]$$

Individu

x ₁										x ₂									
1	1	0	1	0	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1
g_1										g_{10}	g_{11}							g_{20}	

Fitness

$$f = \frac{1}{(x_1^2 + x_2^2) + 0,01}$$

Jika nilai minimum = 0, nilai maks $f = ?$

Generasi 1

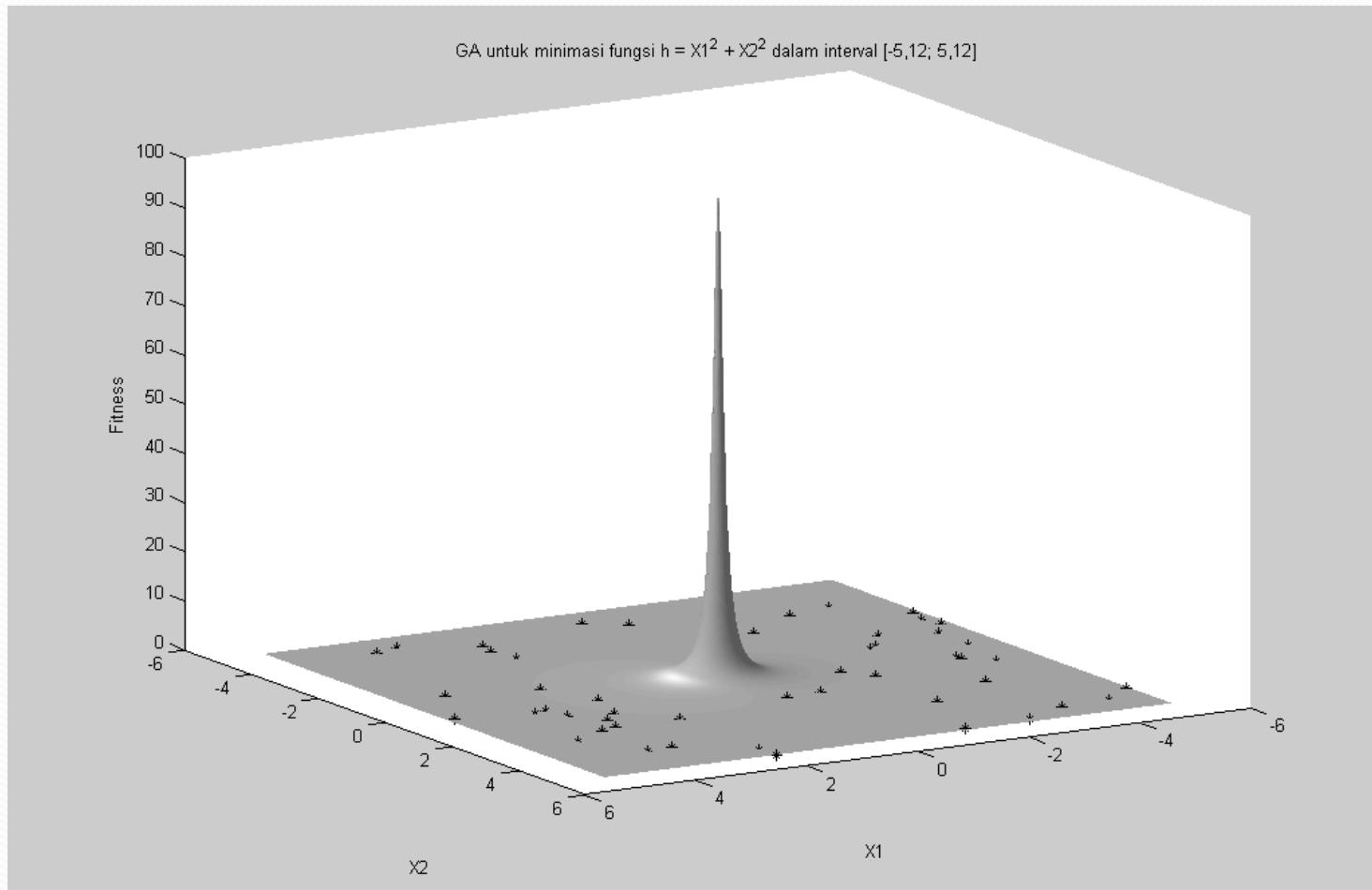
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_N \cdot 2^{-N})$$

$$x_1, x_2 \in [-5,12; 5,12]$$



No	Genotype	Phenotype		Nilai fitness
	kromosom biner	X1	X2	
1	00010011011001101110	-4.35	1.1	0.049646
2	11001101110001000011	3.11	-4.45	0.033916
3	10110010111111001110	2.03	4.62	0.039254
4	11001110001101111101	3.12	3.81	0.041219
5	11001110101011011001	3.14	2.17	0.068594
6	00101110000110110110	-3.28	-0.74	0.08837
7	01111011111010110010	-0.17	1.78	0.31179
50	11010110011000111011	3.45	0.59	0.081562

Generasi 1



Generasi 10

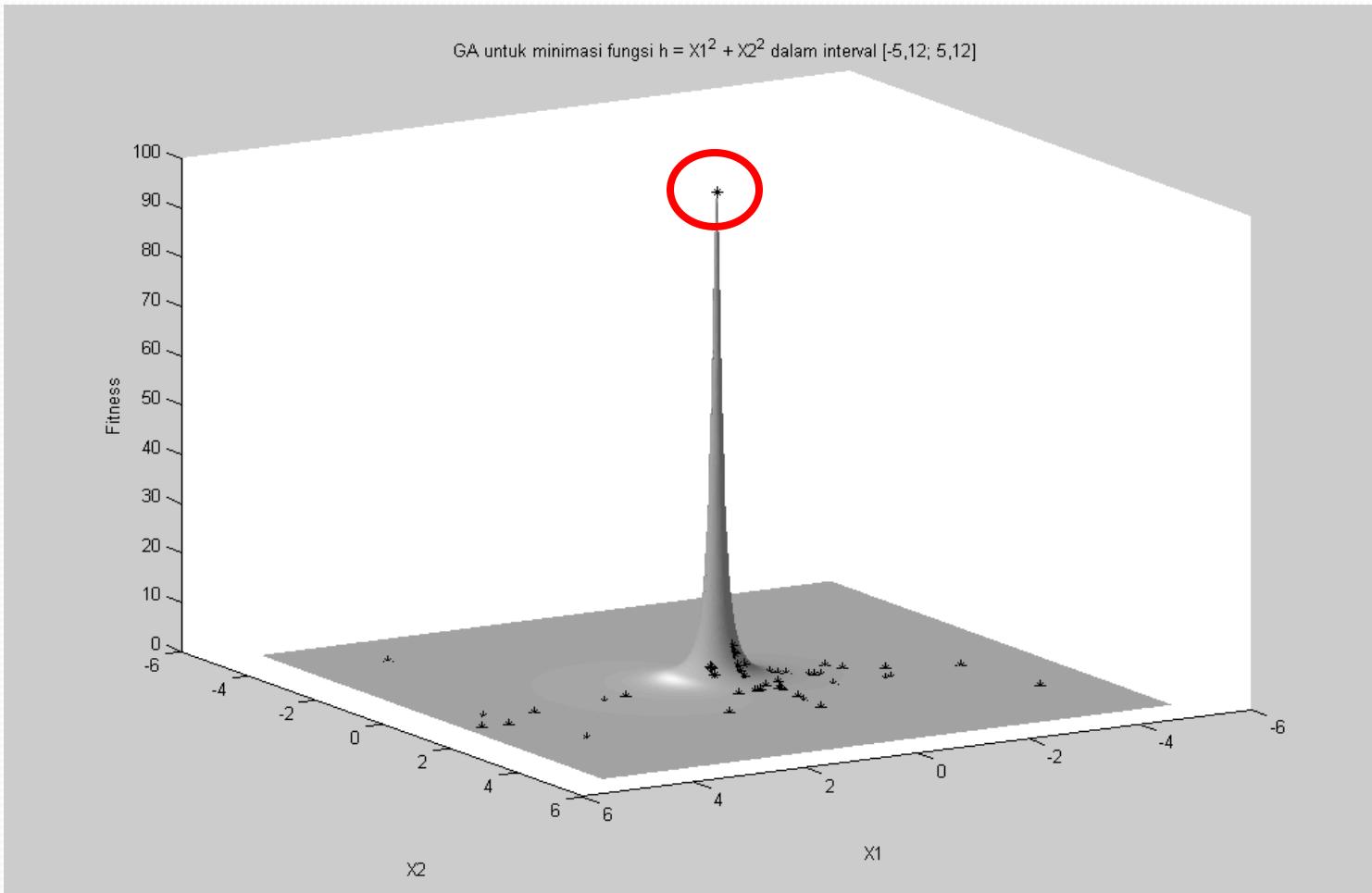
$$x = r_b + \frac{(r_a - r_b)}{\sum_{i=1}^N 2^{-i}} (g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_N \cdot 2^{-N})$$

$$x_1, x_2 \in [-5,12; 5,12]$$

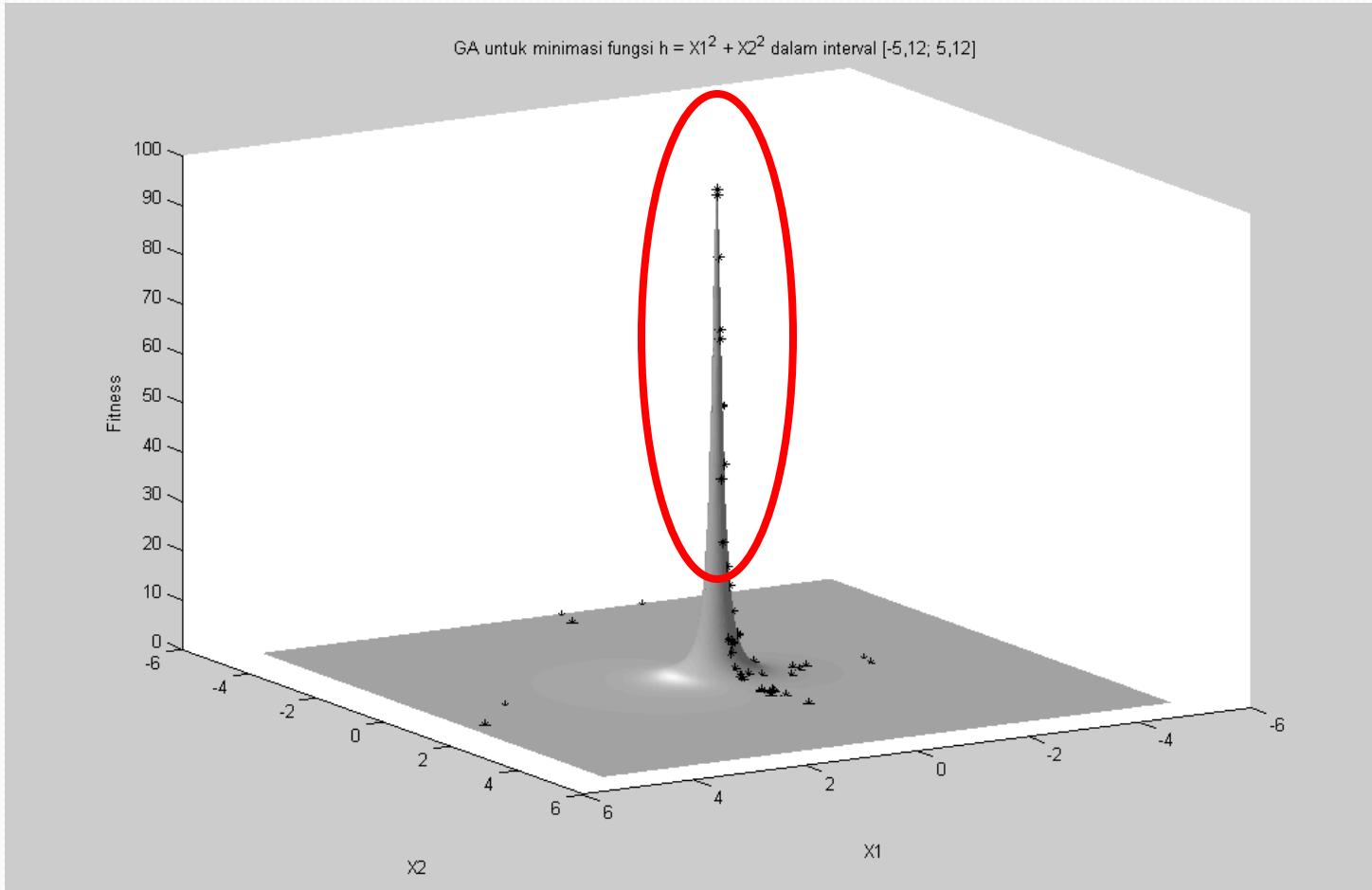


No	Genotype	Phenotype		Nilai fitness
	kromosom biner	X1	X2	
1	01111111110000000000	-0.01	0	99.01
2	01111111110000000000	-0.01	0	99.01
3	01111101010001000001	-3.77	1.03	0.065429
4	01111101011001110001	-2.4	1.2	0.1387
5	01111001111000100001	3.58	0.52	0.076355
6	01011101111000101010	4.83	1.01	0.041053
7	01111101111000100001	-1.38	1.2	0.29812
...				
50	01111101111000000001	-1.93	0.02	0.26772

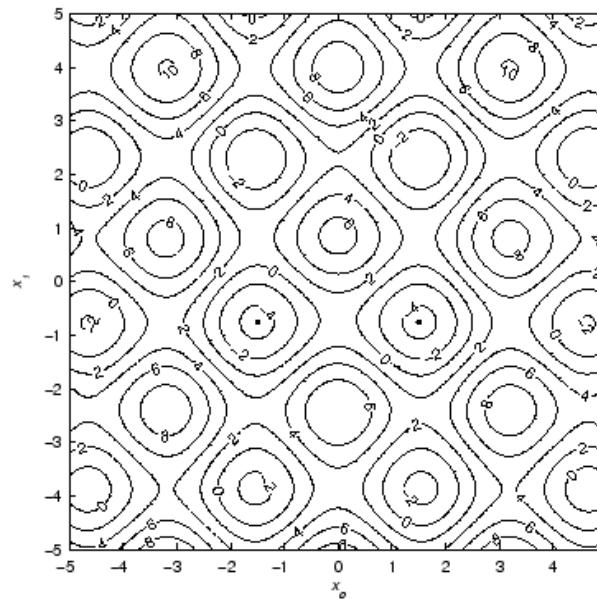
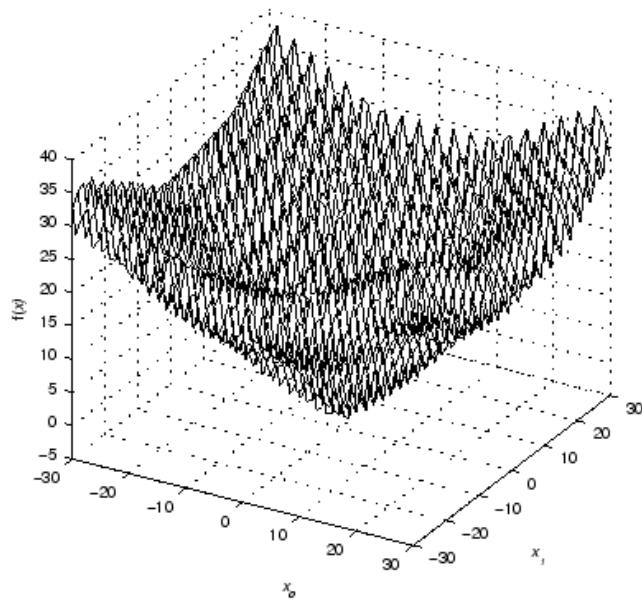
Generasi 10



Generasi 100



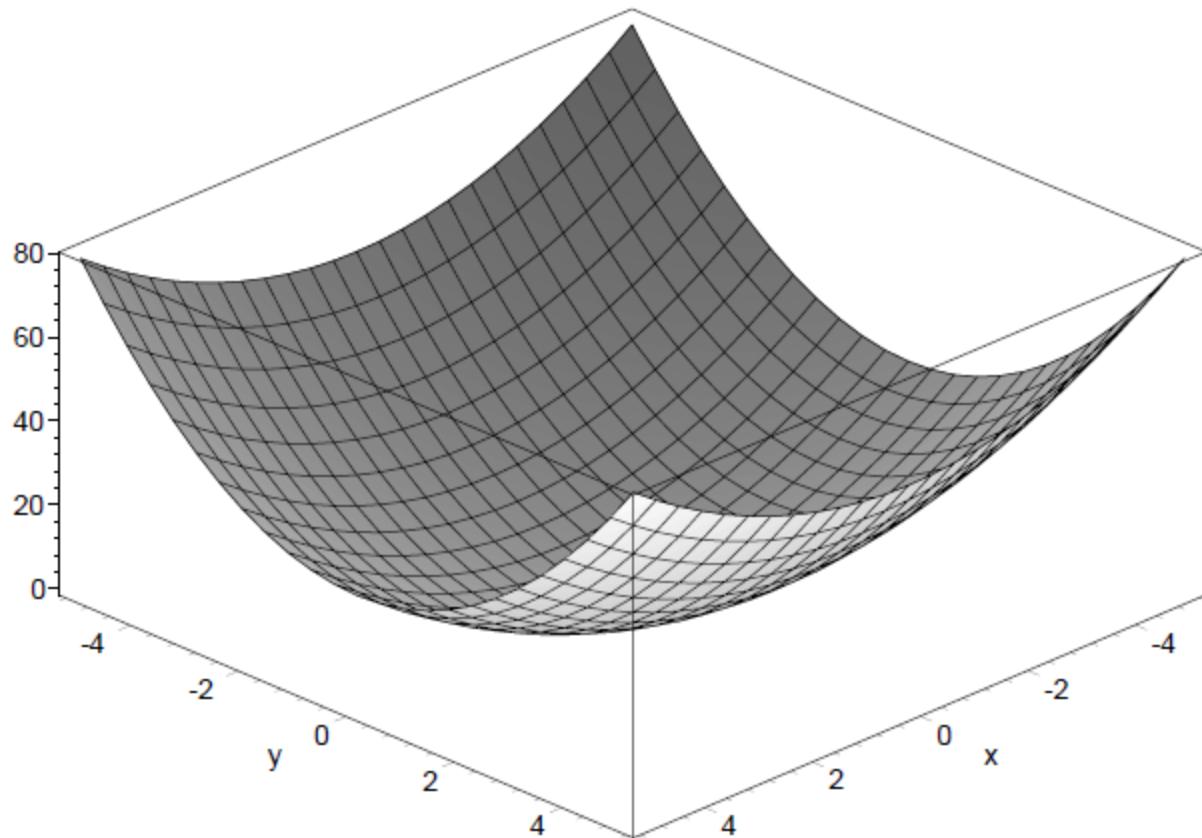
$$f(\vec{x}) = \sum_{i=0}^{D-1} \left(e^{-0.2} \sqrt{x_{i-1}^2 + x_i^2} + 3(\cos(2x_{i-1}) + \sin(2x_i)) \right)$$

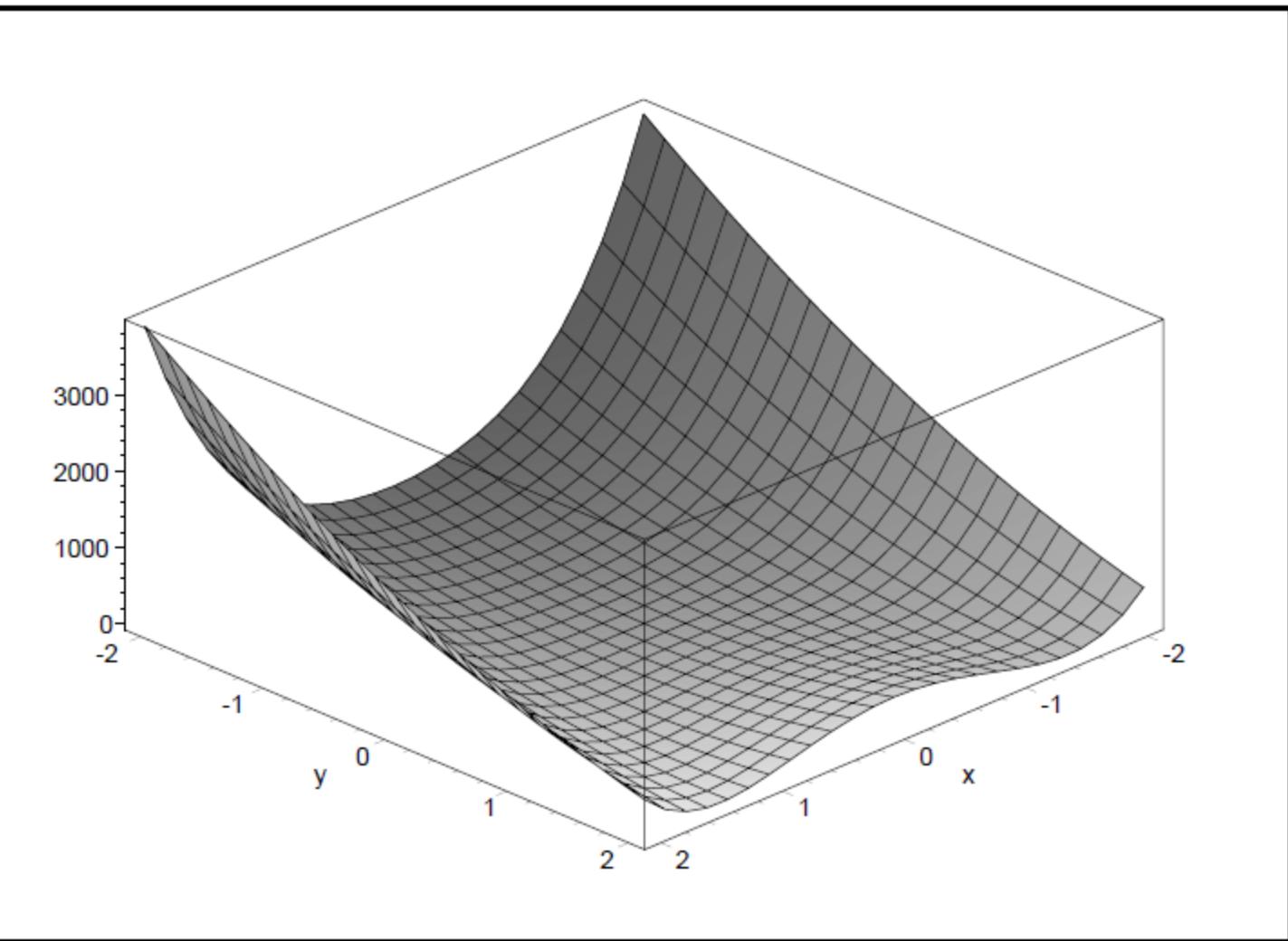


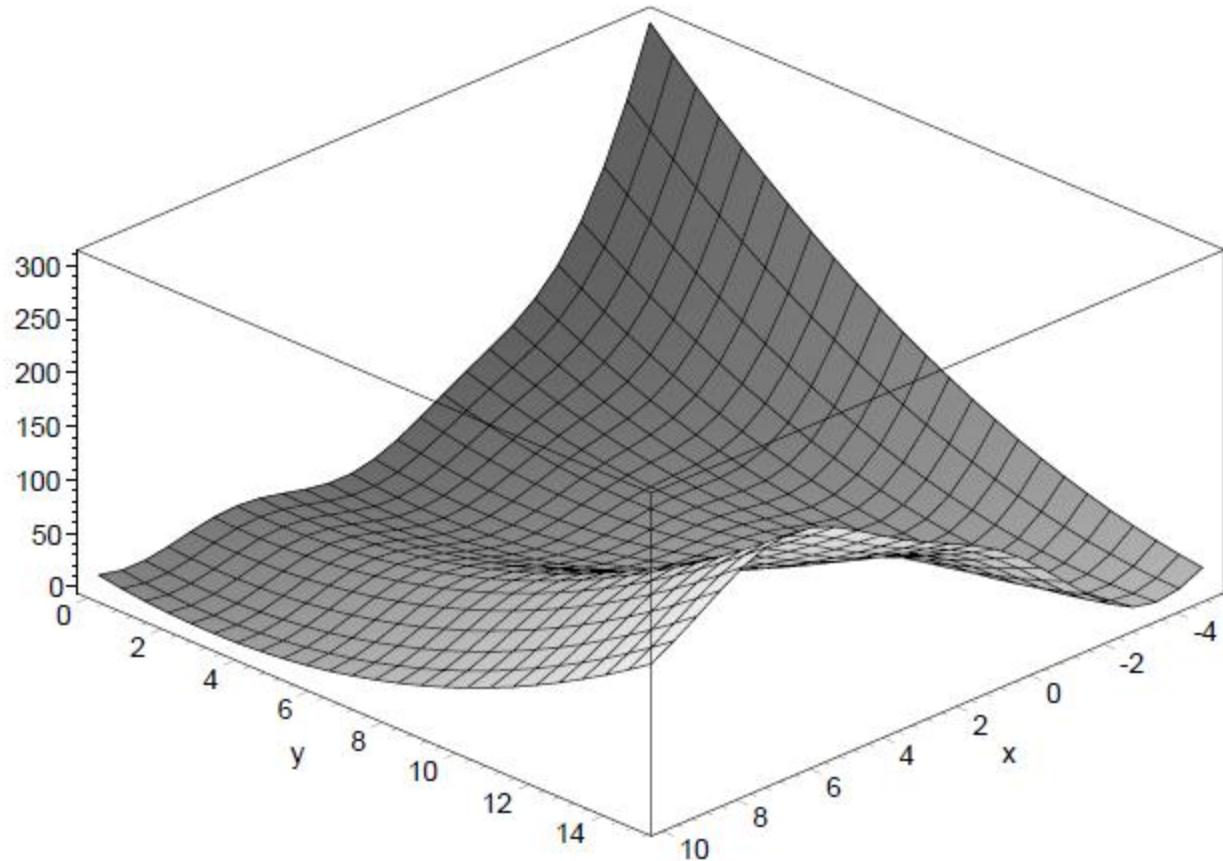
- Untuk presisi 10^{-9} → Berapa bit?
- Bisa menggunakan kromosom Real?

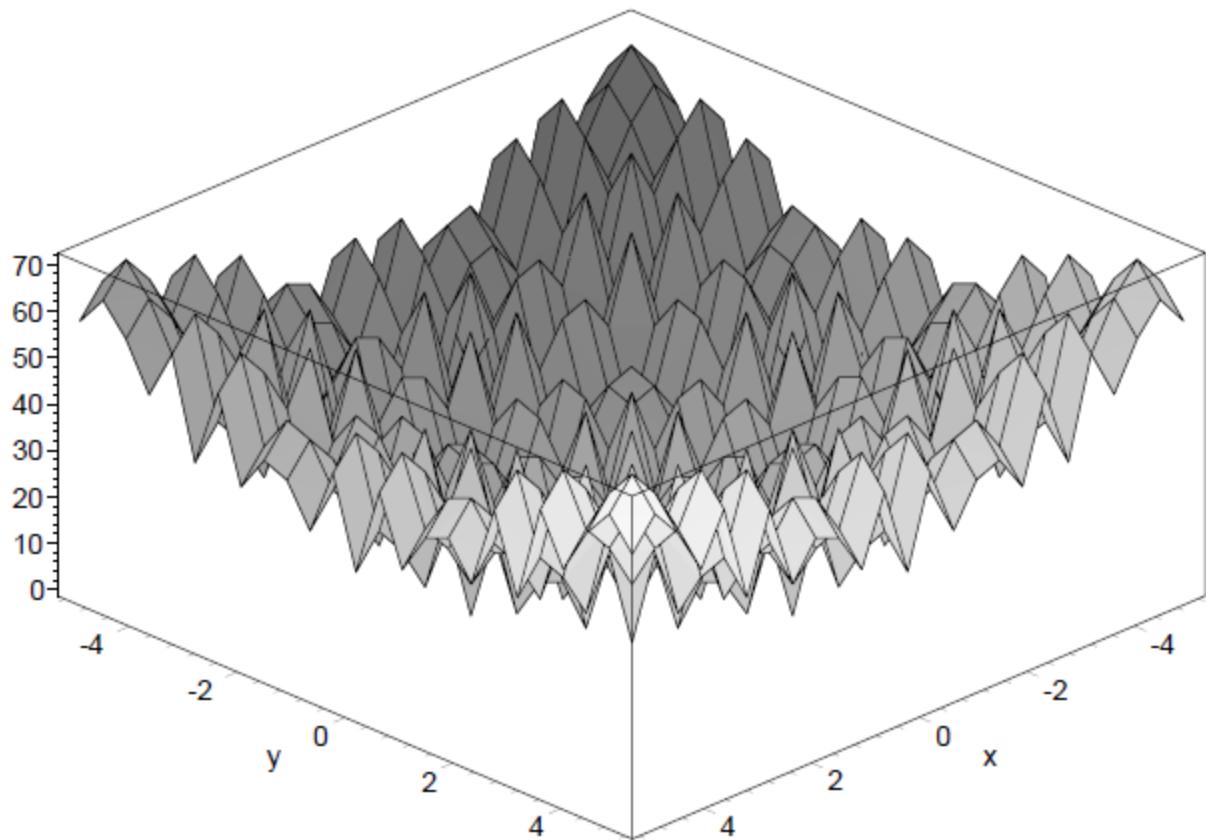
Representasi individu → kromosom

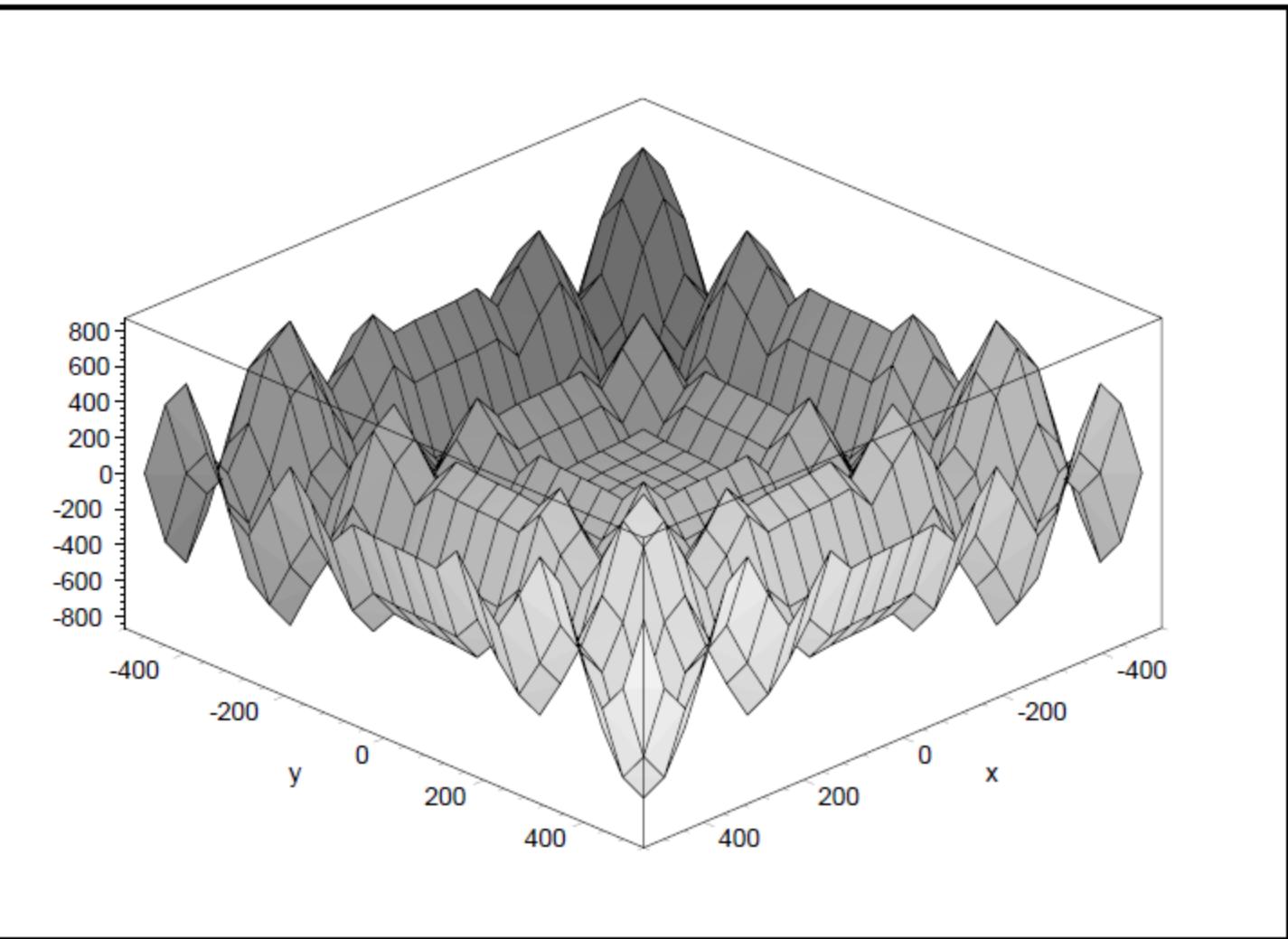
- Representasi Biner
- Representasi Integer
- Representasi Real
- Representasi Permutasi

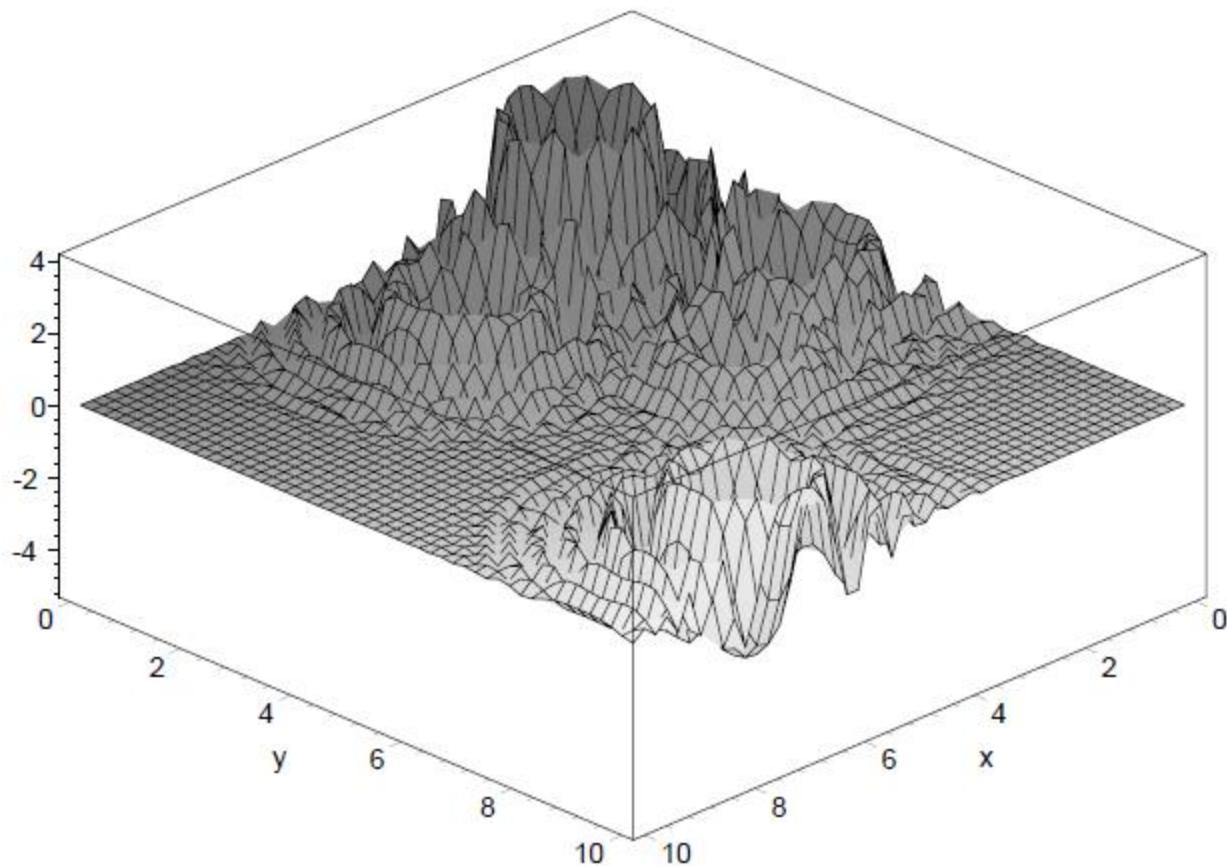


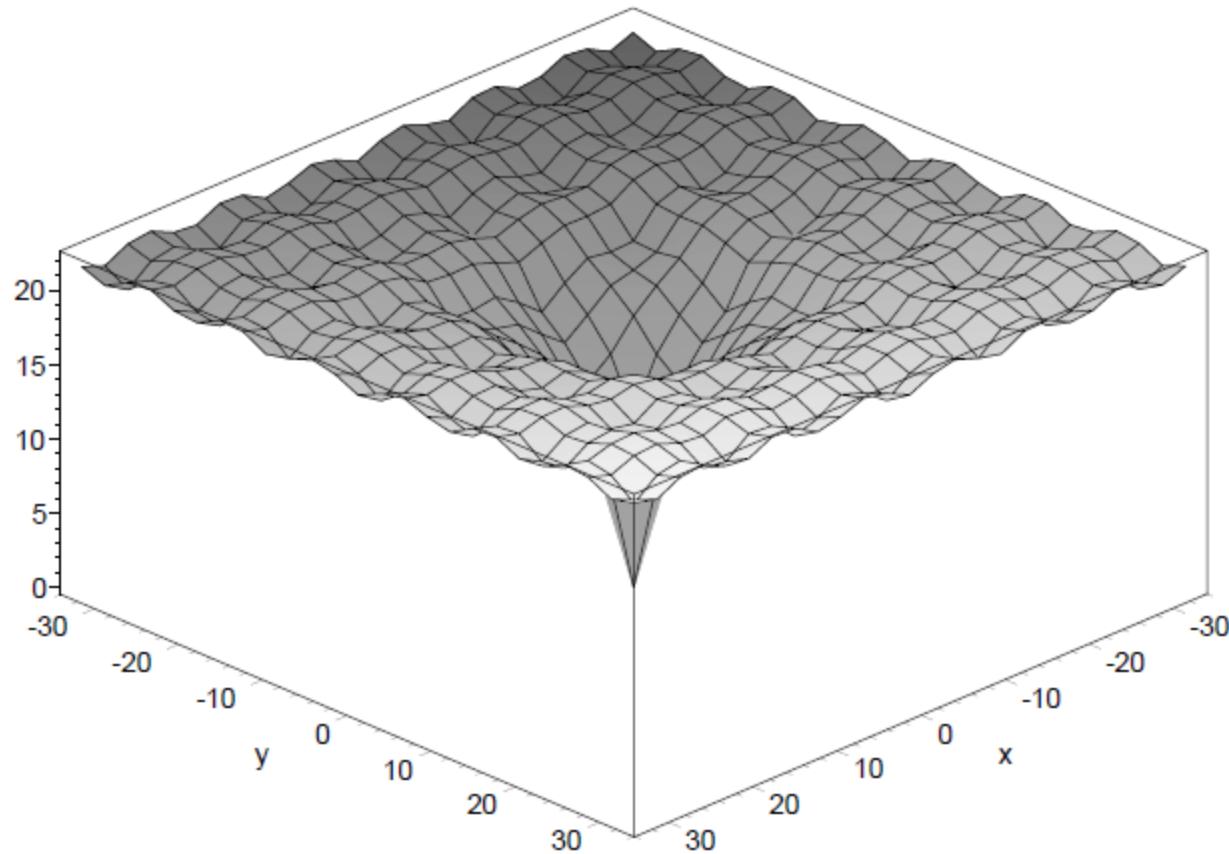


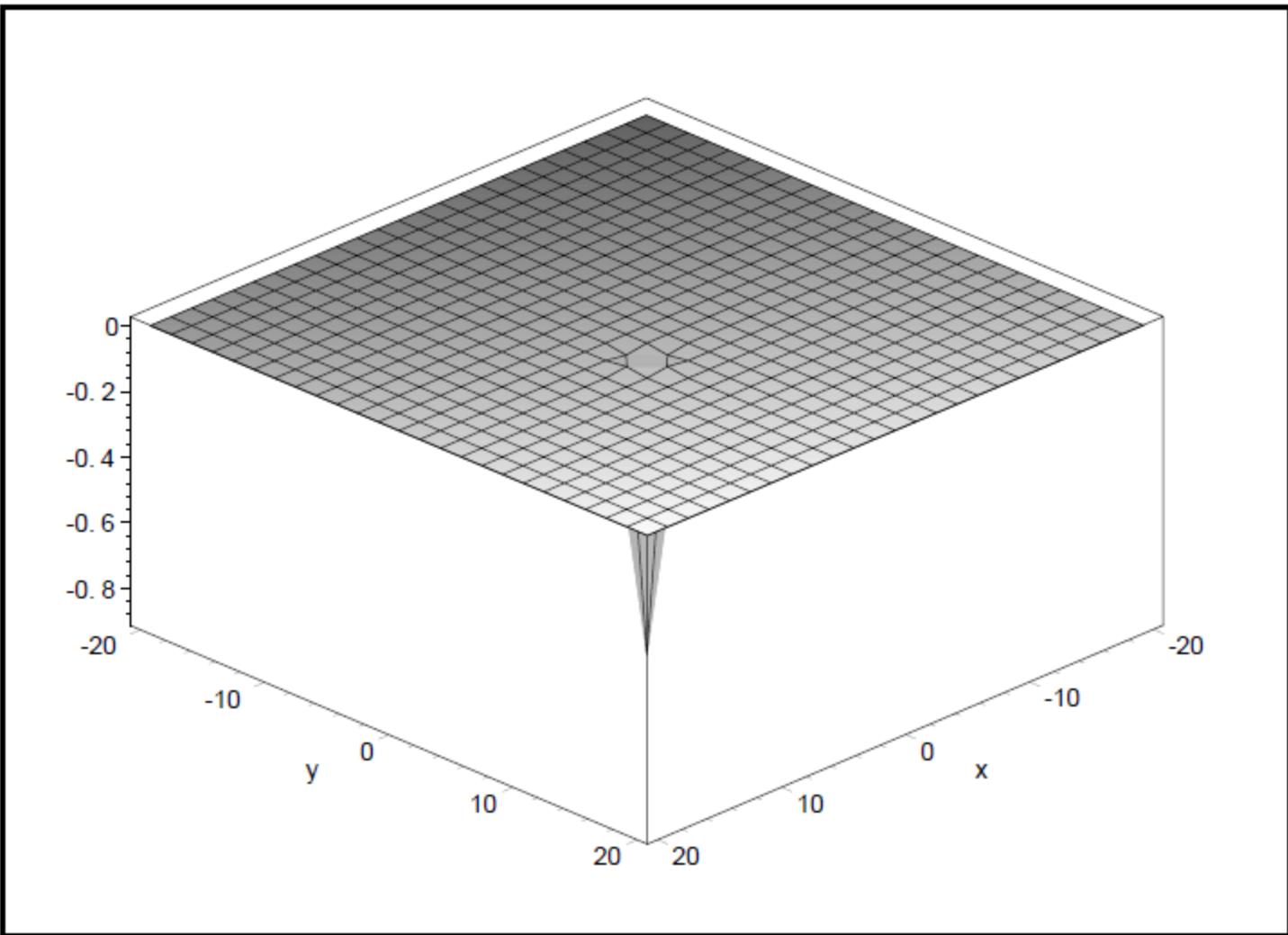


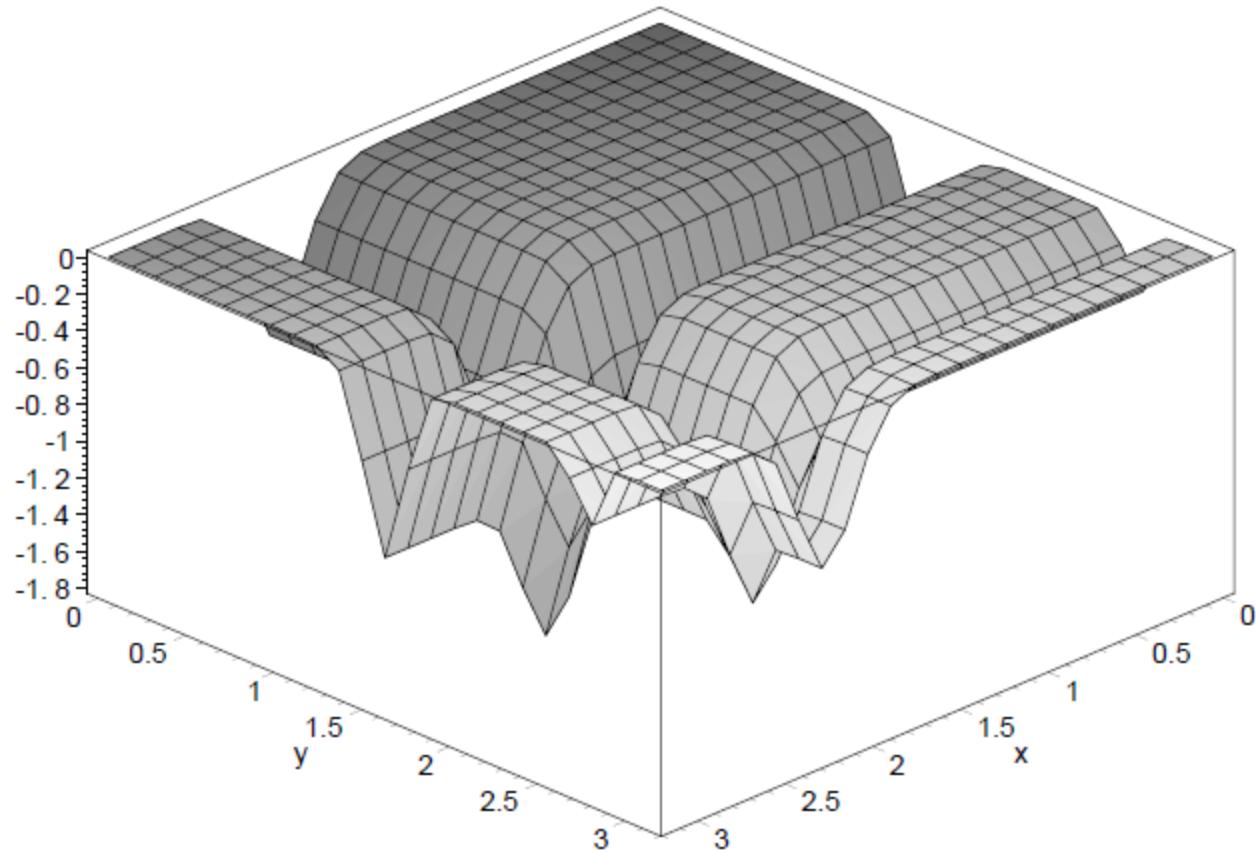


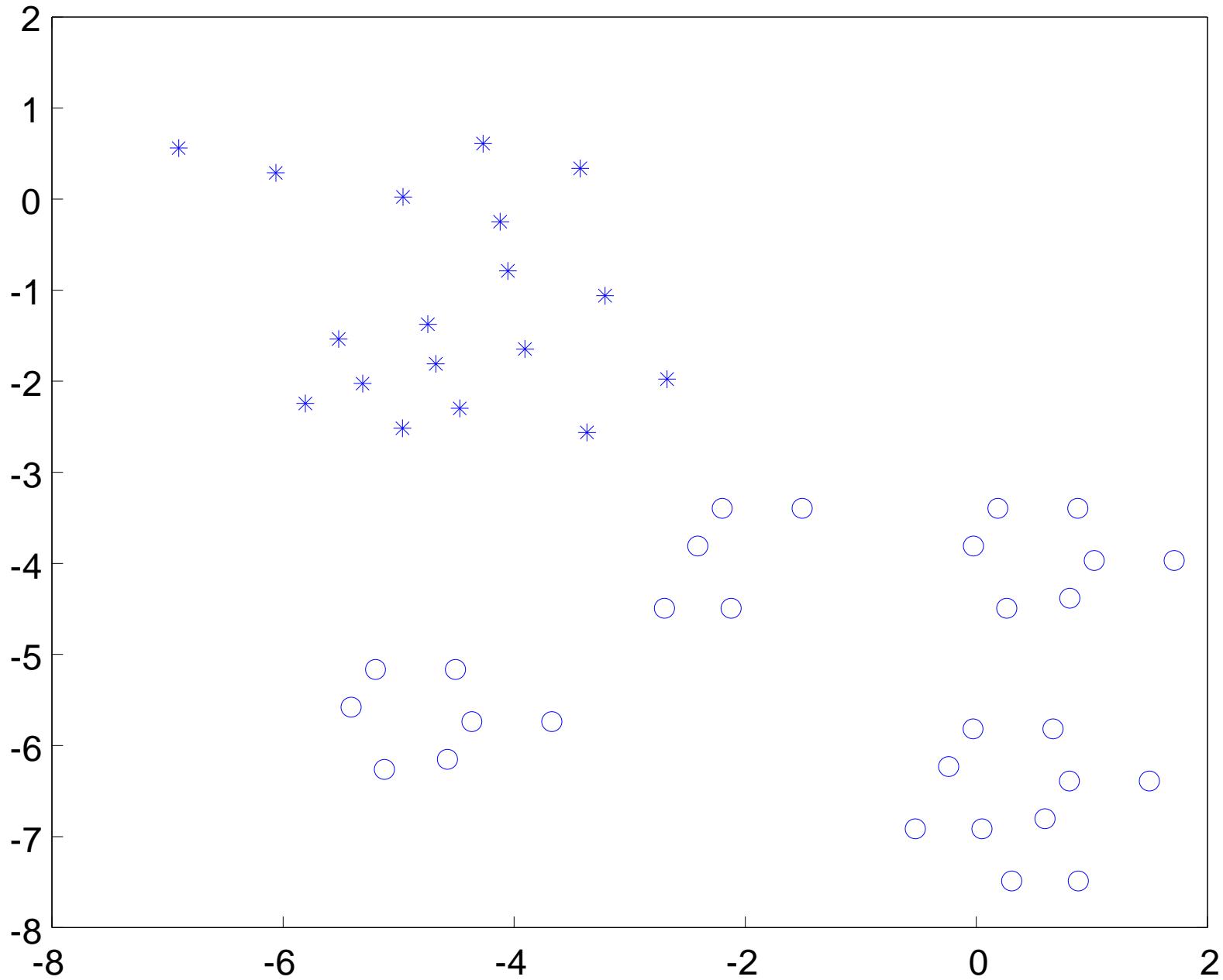












Setting parameter GA

- Tidak ada panduan yang pasti
 - Umumnya:
 - Representasi kromosom = biner/integer/real/permutasi
 - Jumlah bit per variabel → presisi yang diinginkan
 - Ukuran Populasi = $50 - 100$
 - Probabilitas Crossover (Pc) = $0,8$
 - Probabilitas Mutasi (Pm) = $1/NL$ sampai $1/L$
- N = Ukuran Populasi
- L = Panjang Kromosom (Jumlah Gen)

Desain GA

- Representasi kromosom yang sesuai masalah
- Fungsi fitness yang efisien
- Operator evolusi yang sesuai dengan kromosom & fungsi fitness
- Advanced GA
- Penambahan algoritma deterministik

Observasi parameter GA

- Minimasi fungsi $h = x_1^2 + x_2^2$, x_1 dan x_2 elemen $[-10, 10]$
- Fitness = $1/(x_1^2 + x_2^2 + 0.001)$

- Ukuran Populasi = [50 100 200]
- Jumlah bit = [10 50 90]
- Prob Rekombinasi = [0.5 0.7 0.9]
- Prob Mutasi = [0.5/JumGen 1/JumGen 2/JumGen]

- Jumlah Individu maksimum = 20000 (*fairness*)
- Jumlah running/percobaan = 30 (*valid*)

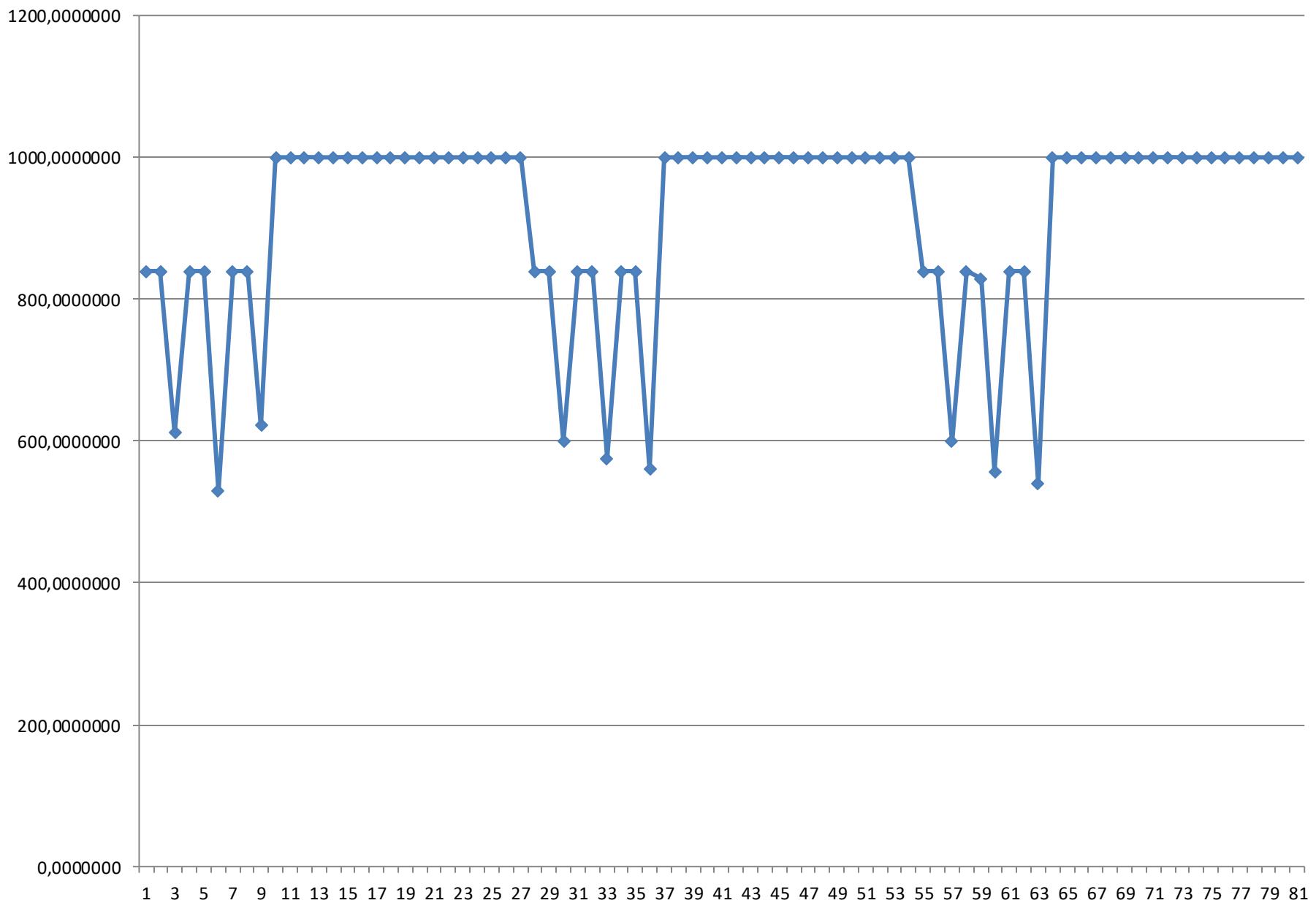
No Observasi	Ukuran populasi	Jumlah bit	Probabilitas Pdh Silang	Probabilitas Mutasi	Rata-rata Fitness terbaik	Rata-rata Jml Individu yang dievaluasi
1	50	10	0,5	0,0250	839,5544749	20000,0000
2	50	10	0,5	0,0500	839,5544749	20000,0000
3	50	10	0,5	0,1000	611,0770624	20000,0000
4	50	10	0,7	0,0250	839,5544749	20000,0000
5	50	10	0,7	0,0500	839,5544749	20000,0000
6	50	10	0,7	0,1000	528,7161733	20000,0000
7	50	10	0,9	0,0250	839,5544749	20000,0000
8	50	10	0,9	0,0500	839,5544749	20000,0000
9	50	10	0,9	0,1000	622,2201392	20000,0000
10	50	50	0,5	0,0050	1000,0000000	8301,6667
11	50	50	0,5	0,0100	1000,0000000	20000,0000
12	50	50	0,5	0,0200	999,9987777	20000,0000
13	50	50	0,7	0,0050	1000,0000000	8013,3333
14	50	50	0,7	0,0100	1000,0000000	20000,0000
15	50	50	0,7	0,0200	999,9982015	20000,0000
16	50	50	0,9	0,0050	1000,0000000	8133,3333
17	50	50	0,9	0,0100	1000,0000000	20000,0000
18	50	50	0,9	0,0200	999,9988782	20000,0000
19	50	90	0,5	0,0028	1000,0000000	8361,6667
20	50	90	0,5	0,0056	1000,0000000	8796,6667
21	50	90	0,5	0,0111	1000,0000000	20000,0000
22	50	90	0,7	0,0028	1000,0000000	8151,6667
23	50	90	0,7	0,0056	1000,0000000	8780,0000
24	50	90	0,7	0,0111	1000,0000000	20000,0000
25	50	90	0,9	0,0028	1000,0000000	7538,3333
26	50	90	0,9	0,0056	1000,0000000	8995,0000
27	50	90	0,9	0,0111	1000,0000000	20000,0000
28	100	10	0,5	0,0250	839,5544749	20000,0000
29	100	10	0,5	0,0500	839,5544749	20000,0000
30	100	10	0,5	0,1000	599,4452769	20000,0000

Paket parameter terbaik untuk kasus di atas

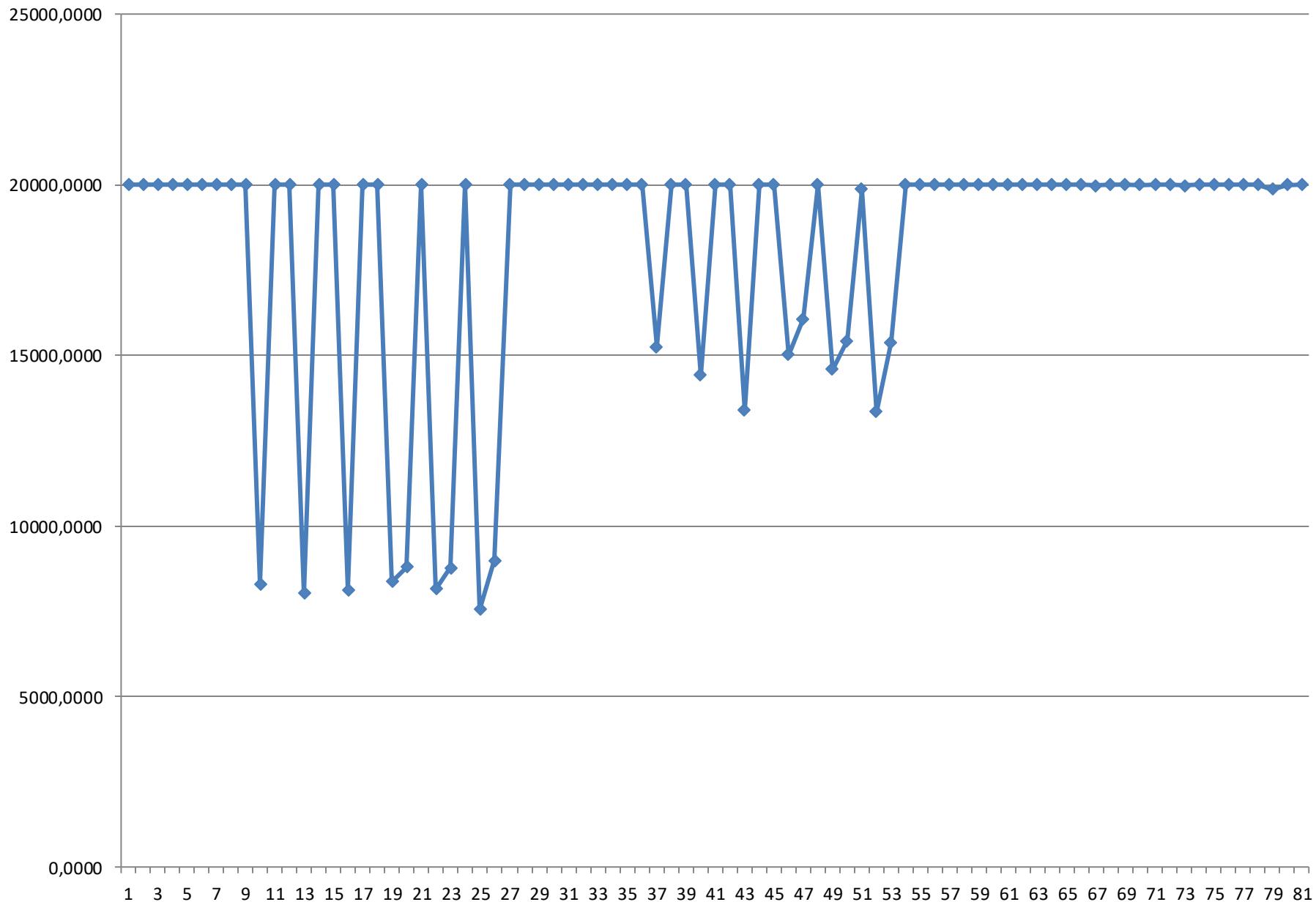
No Observasi	Ukuran populasi	Jumlah bit	Probabilitas Pdh Silang	Probabilitas Mutasi	Rata-rata Fitness terbaik	Rata-rata Jml Individu yang dievaluasi
31	100	10	0,7	0,0250	839,5544749	20000,0000
32	100	10	0,7	0,0500	839,5544749	20000,0000
33	100	10	0,7	0,1000	575,8472869	20000,0000
34	100	10	0,9	0,0250	839,5544749	20000,0000
35	100	10	0,9	0,0500	839,5544749	20000,0000
36	100	10	0,9	0,1000	559,6804844	20000,0000
37	100	50	0,5	0,0050	1000,0000000	15246,6667
38	100	50	0,5	0,0100	1000,0000000	20000,0000
39	100	50	0,5	0,0200	999,9986429	20000,0000
40	100	50	0,7	0,0050	1000,0000000	14416,6667
41	100	50	0,7	0,0100	1000,0000000	20000,0000
42	100	50	0,7	0,0200	999,9988459	20000,0000
43	100	50	0,9	0,0050	1000,0000000	13390,0000
44	100	50	0,9	0,0100	1000,0000000	20000,0000
45	100	50	0,9	0,0200	999,9987118	20000,0000
46	100	90	0,5	0,0028	1000,0000000	15010,0000
47	100	90	0,5	0,0056	1000,0000000	16056,6667
48	100	90	0,5	0,0111	1000,0000000	20000,0000
49	100	90	0,7	0,0028	1000,0000000	14580,0000
50	100	90	0,7	0,0056	1000,0000000	15430,0000
51	100	90	0,7	0,0111	1000,0000000	19860,0000
52	100	90	0,9	0,0028	1000,0000000	13346,6667
53	100	90	0,9	0,0056	1000,0000000	15390,0000
54	100	90	0,9	0,0111	1000,0000000	20000,0000
55	200	10	0,5	0,0250	839,5544749	20000,0000
56	200	10	0,5	0,0500	839,5544749	20000,0000
57	200	10	0,5	0,1000	599,0108676	20000,0000
58	200	10	0,7	0,0250	839,5544749	20000,0000
59	200	10	0,7	0,0500	828,6149185	20000,0000
60	200	10	0,7	0,1000	557,3828866	20000,0000

No Observasi	Ukuran populasi	Jumlah bit	Probabilitas Pdh Silang	Probabilitas Mutasi	Rata-rata Fitness terbaik	Rata-rata Jml Individu yang dievaluasi
61	200	10	0,9	0,0250	839,5544749	20000,0000
62	200	10	0,9	0,0500	839,5544749	20000,0000
63	200	10	0,9	0,1000	539,1055371	20000,0000
64	200	50	0,5	0,0050	1000,0000000	20000,0000
65	200	50	0,5	0,0100	999,9999995	20000,0000
66	200	50	0,5	0,0200	999,9986789	20000,0000
67	200	50	0,7	0,0050	1000,0000000	19966,6667
68	200	50	0,7	0,0100	999,9999997	20000,0000
69	200	50	0,7	0,0200	999,9947933	20000,0000
70	200	50	0,9	0,0050	1000,0000000	19986,6667
71	200	50	0,9	0,0100	999,9999996	20000,0000
72	200	50	0,9	0,0200	999,9939550	20000,0000
73	200	90	0,5	0,0028	999,9999988	19966,6667
74	200	90	0,5	0,0056	999,9999999	20000,0000
75	200	90	0,5	0,0111	999,9999976	20000,0000
76	200	90	0,7	0,0028	999,9999995	20000,0000
77	200	90	0,7	0,0056	1000,0000000	20000,0000
78	200	90	0,7	0,0111	999,9999979	20000,0000
79	200	90	0,9	0,0028	1000,0000000	19866,6667
80	200	90	0,9	0,0056	1000,0000000	19993,3333
81	200	90	0,9	0,0111	999,9999988	20000,0000

Rata-rata Fitness Terbaik



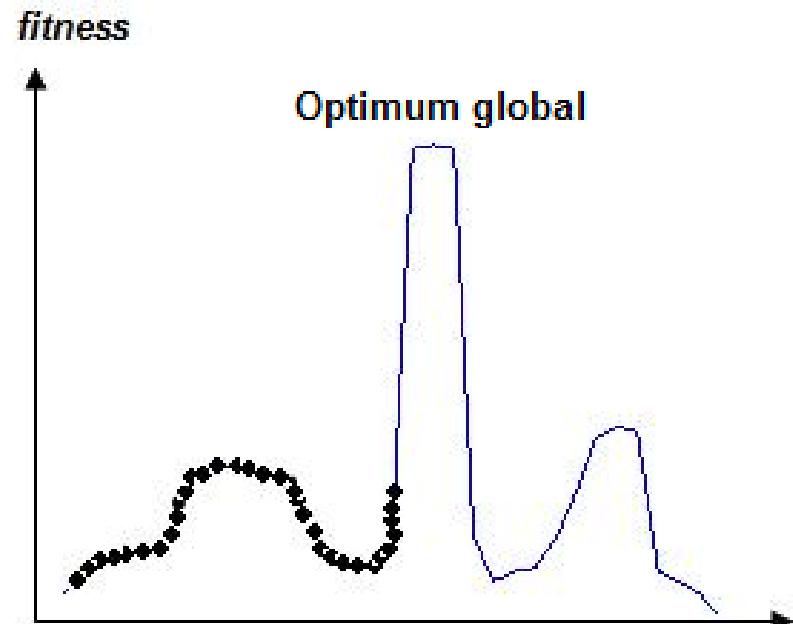
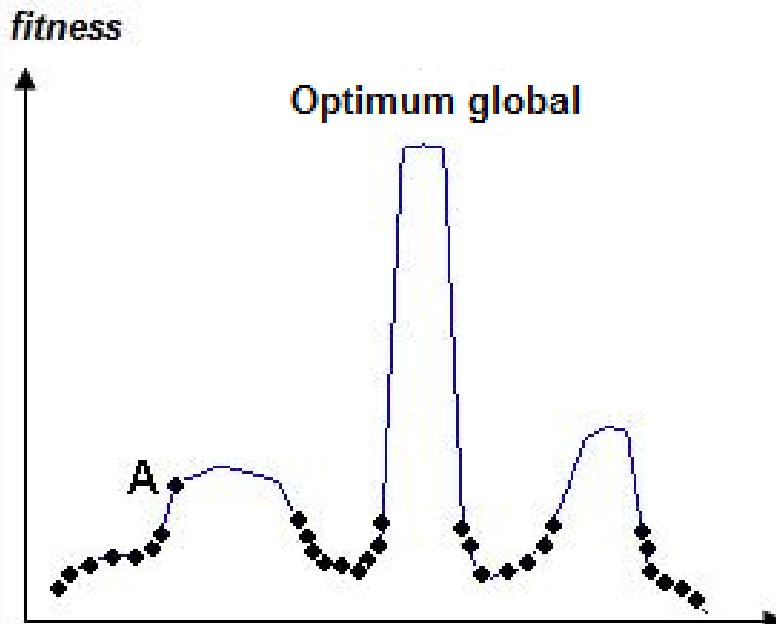
Rata-rata Jumlah Individu yg Dievaluasi



Advanced GA

- Konvergensi Prematur (KP)
- *Gray Coding*
- *Messy Encoding*
- *Fitness Ranking*
- *Island Model*
- *Adaptive GA*
- *Grid-Based Crossover*
- *Grammatical Encoding*

Konvergensi Prematur (KP)



Pencegahan KP

- *Gray Coding*
- *Messy Encoding*
- *Fitness Ranking*
- *Island Model*
- *Adaptive GA*

Hamming Distance

Integer

14

0	1	1	1	0
---	---	---	---	---

Biner



Berbeda 1 bit

15

0	1	1	1	1
---	---	---	---	---



Berbeda 5 bit

16

1	0	0	0	0
---	---	---	---	---

Binary Coding

- Jika solusi maksimum yang dicari adalah **10000** (16)
- Individu terbaik saat ini **01111** (15)
- Sampai beberapa generasi berikutnya ternyata individu terbaik tetap **01111** (15).
- Mengapa?
- **01111** → **10000** memerlukan mutasi 5 gen.
- Padahal probabilitas mutasi biasanya dibuat sangat kecil, biasanya $1/NL$ sampai $1/L$, dimana N adalah ukuran populasi, L panjang kromosom.
- SOLUSINYA?

Gray Coding

Integer	0	1	2	3	4	5	6	7
Binary coding	000	001	010	011	100	101	110	111
Gray coding	000	001	011	010	110	111	101	100

Gray Coding

individu: $x_1 = 5$ dan $x_2 = 3$



x_1			x_2		
1	0	1	0	1	1

Binary coding



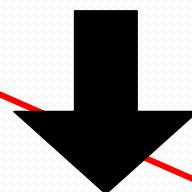
x_1			x_2		
1	1	1	0	1	0

Kromosom: *Gray coding*

Messy Encoding

Messy encoding:

3	0	8	0	2	1	6	1	1	0	4	1	5	1	7	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

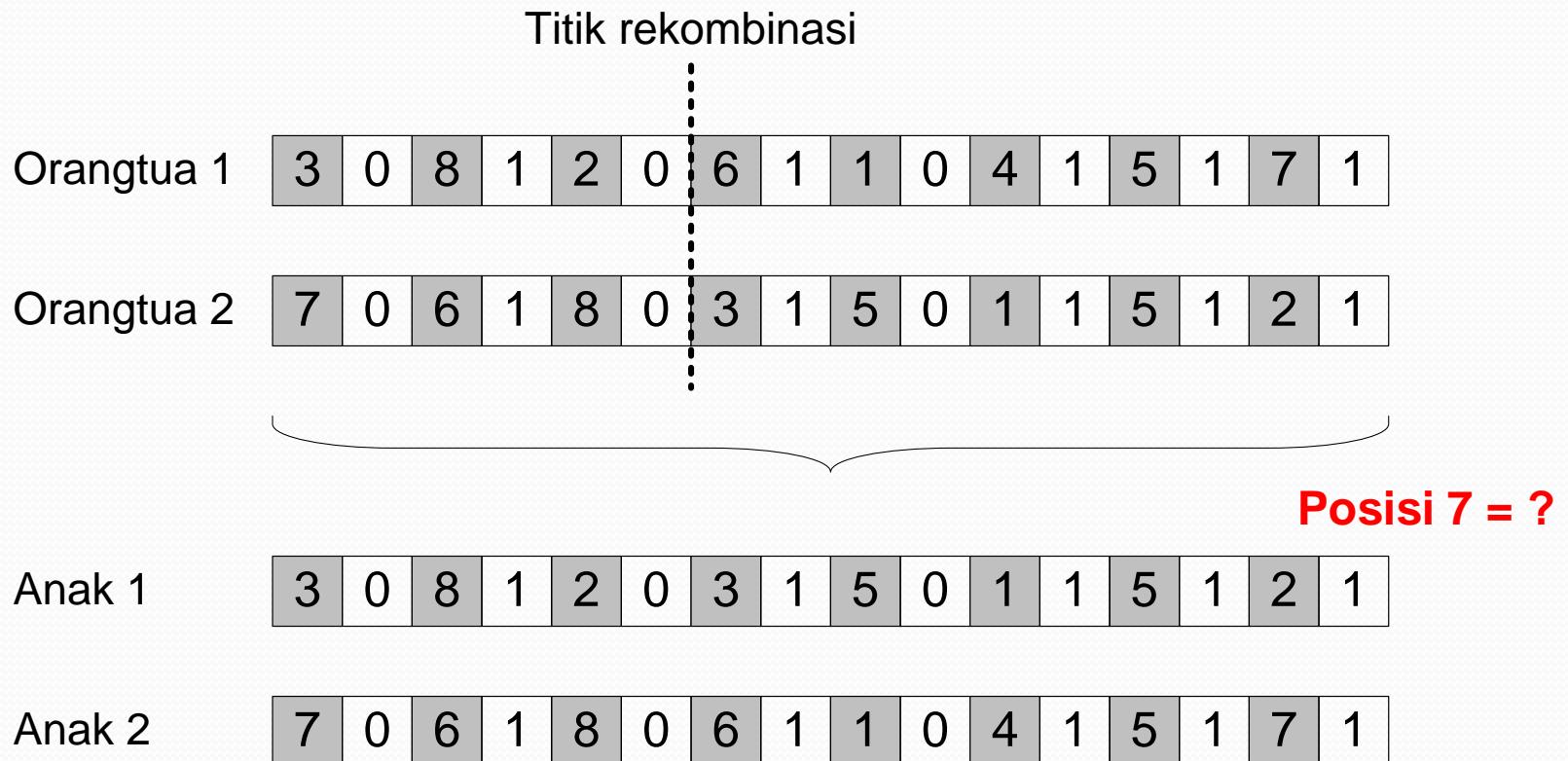


Binary encoding:

0	1	0	1	1	1	1	0
---	---	---	---	---	---	---	---

1 2 3 4 5 6 7 8

Crossover

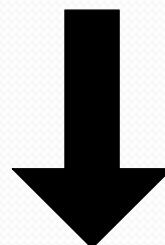


Messy encoding:

3	0	8	1	2	0	3	1	5	0	1	1	5	1	2	1
3	0	8	1	2	0	3	1	5	0	1	1	5	1	2	1

dipanjangkan
n kali semula

6	0	4	1	8	1	3	1	2	0	5	1	4	1	5	1
6	0	4	1	8	1	3	1	2	0	5	1	4	1	5	1



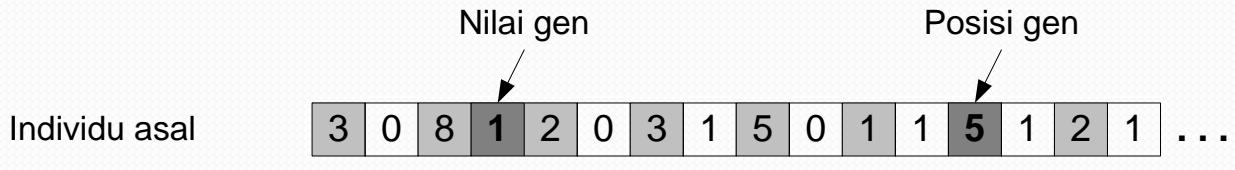
Nilai gen untuk posisi 7 = 0
(dibangkitkan secara acak)

Binary encoding:

1	0	0	1	0	0	0	1
1	0	0	1	0	0	0	1

Mutasi

Messy encoding:



Binary encoding:

1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---



1	0	0	1	1	0	1	0
---	---	---	---	---	---	---	---

Linear Fitness Ranking

Maksimasi h dimana x_1 dan x_2 adalah real $[-2, 2]$)?

$$h(x_1, x_2) = 100000 + 2x_1 + x_2$$

Linear Fitness Ranking

- Pada fungsi di atas, nilai-nilai h berada dalam interval 99994 sampai 100006.
- Dengan demikian, semua individu memiliki nilai *fitness* yang hampir sama dalam kisaran 100000.
- Hal ini akan berakibat buruk pada proses seleksi orangtua secara proporsional terhadap *fitness*-nya.
- Bagaimana Solusinya?

Linear Fitness Ranking

$$f_{LR}(i) = f_{\max} - (f_{\max} - f_{\min}) \left(\frac{R(i)-1}{N-1} \right)$$

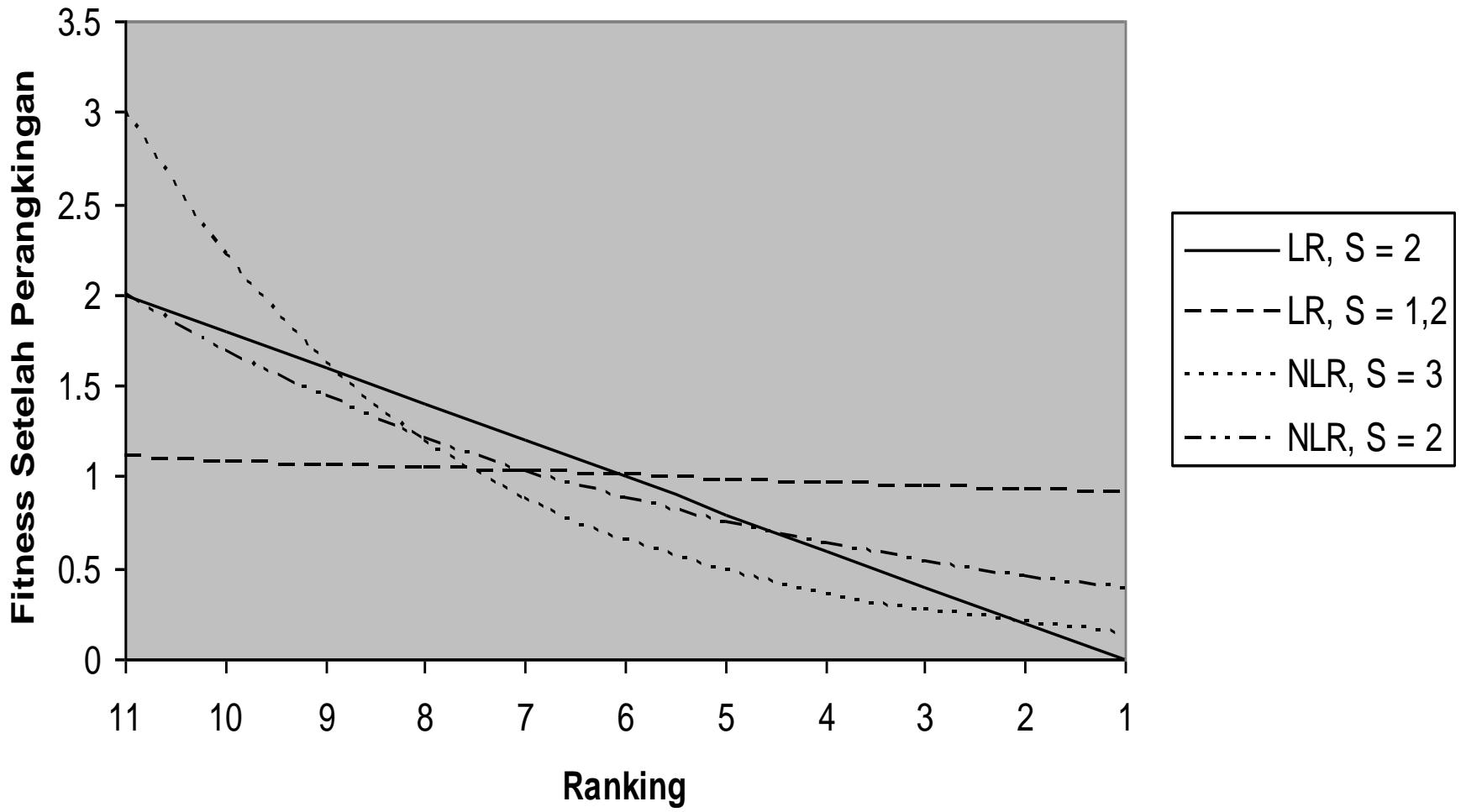
$f_{LR}(i) \rightarrow$ nilai *fitness* individu ke- i yang sudah diskalakan
 $N \rightarrow$ jumlah individu dalam populasi.

$R(i) \rightarrow$ ranking individu ke i .

$f_{\min} \rightarrow$ nilai *fitness* terkecil

$f_{\max} \rightarrow$ nilai *fitness* terbesar

Individu ke	<i>fitness</i>	Ranking $R(i)$	f_{LR}
1	100004,00	1	100004,00
2	100003,99	2	100003,56
3	100003,98	3	100003,12
4	100003,97	4	100002,68
5	100003,96	5	100002,26
6	100003,95	6	100001,82
7	100003,94	7	100001,38
8	100003,93	8	100000,84
9	100003,92	9	100000,44
10	100000,00	10	100000,00



Fitness Perangkingan

<i>Fitness</i>	<i>Posisi</i>	<i>Fitness Perangkingan</i>			
		<i>LR, S = 2</i>	<i>LR, S = 1,2</i>	<i>NLR, S = 3</i>	<i>NLR, S = 2</i>
90,11	11	2	1,1	3	2
90,09	10	1,8	1,08	2,21	1,69
90,08	9	1,6	1,06	1,62	1,43
90,06	8	1,4	1,04	1,19	1,21
90,05	7	1,2	1,02	0,88	1,03
89,97	6	1	1	0,65	0,87
89,96	5	0,8	0,98	0,48	0,74
89,95	4	0,6	0,96	0,35	0,62
79,94	3	0,4	0,94	0,26	0,53
79,93	2	0,2	0,92	0,19	0,45
79,91	1	0	0,9	0,14	0,38

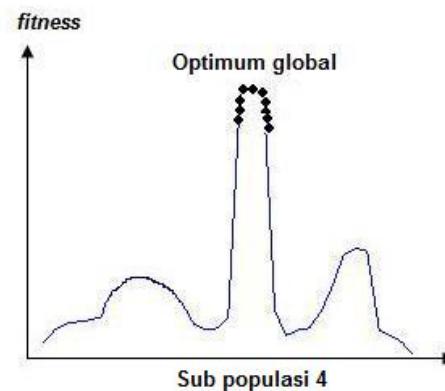
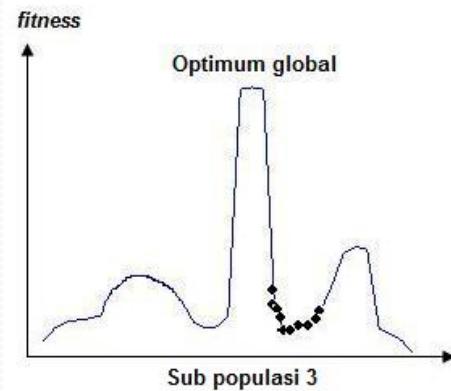
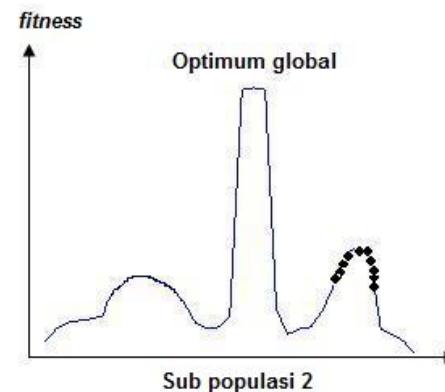
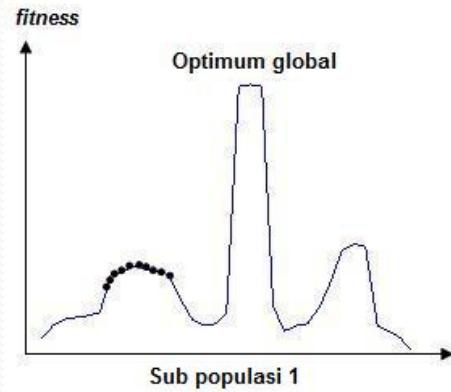
Island Models (Sub Population)

N kromosom dalam satu populasi dibagi menjadi N_k kelompok. Masing-masing kelompok berisi:

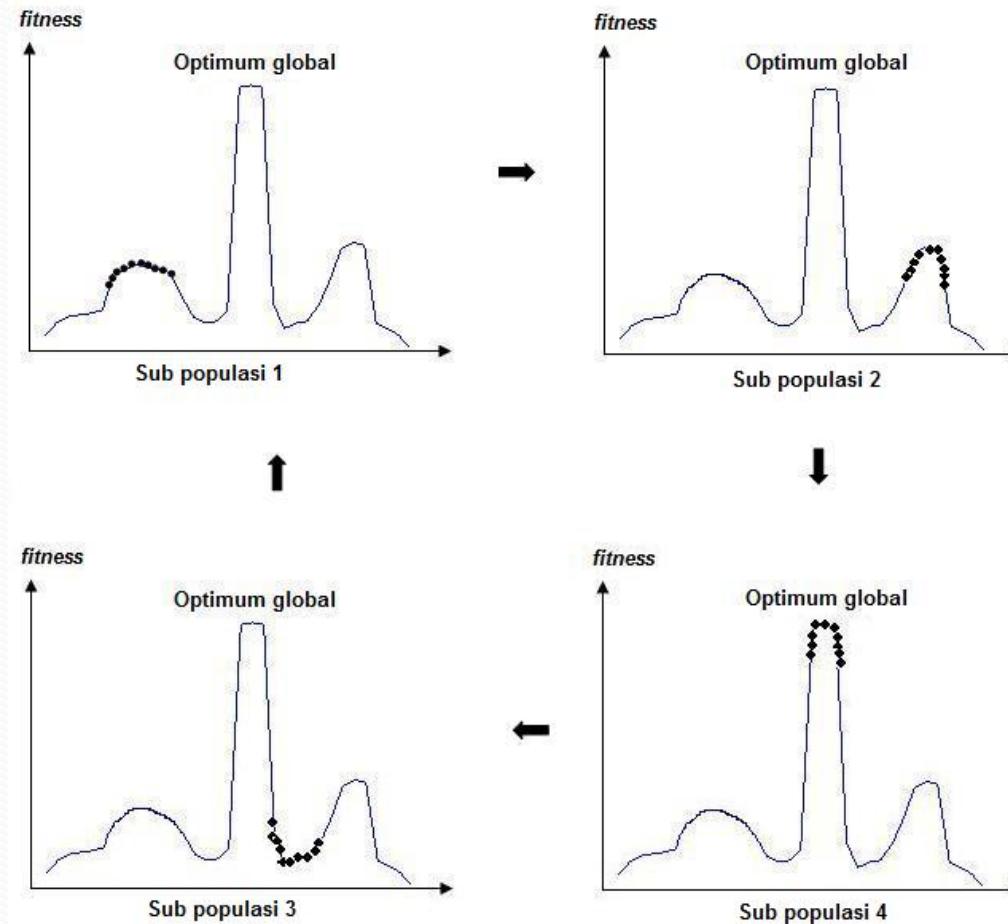
$$v = \frac{N}{N_k}$$

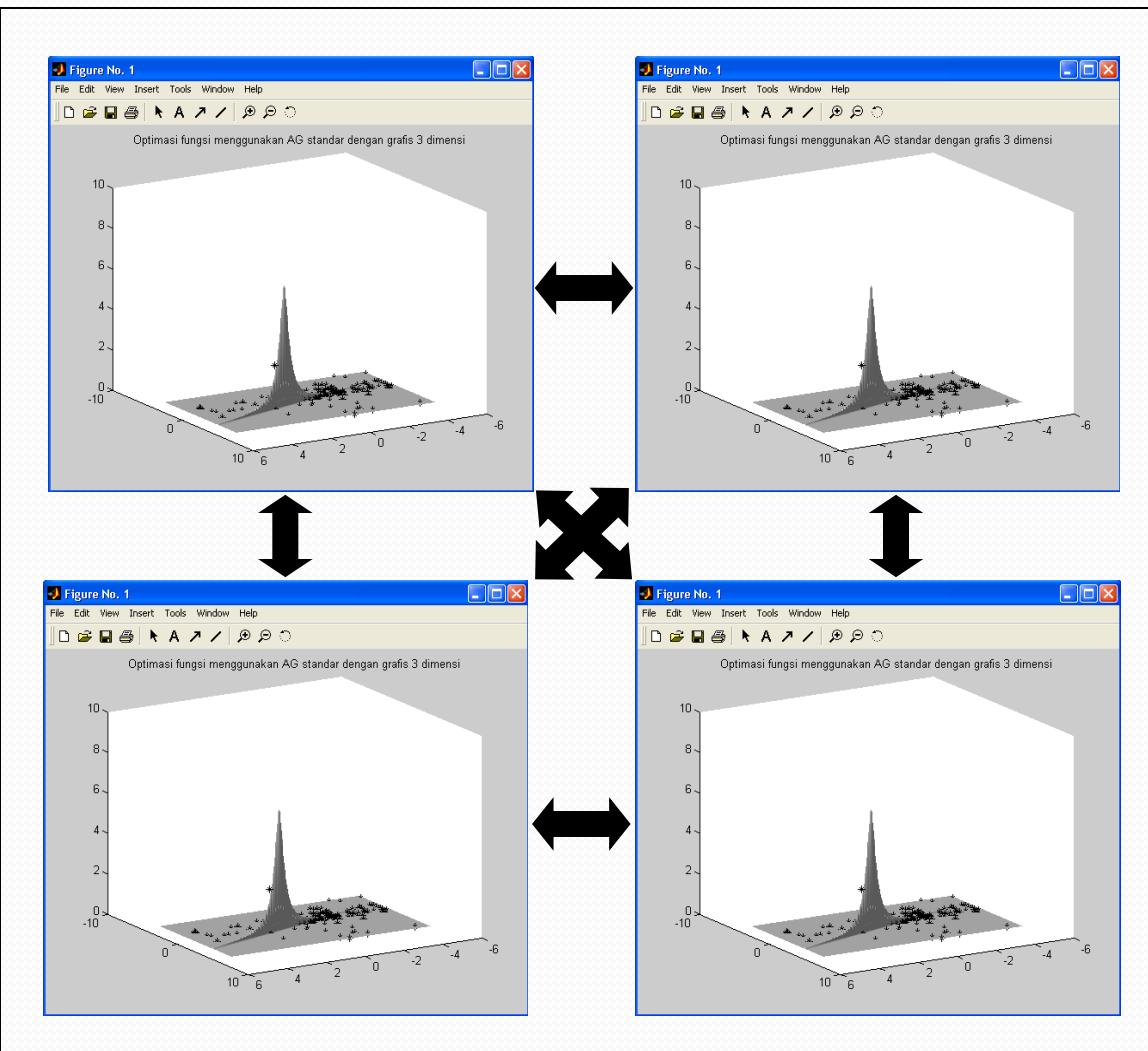
Suatu individu bisa dipindah ke sub populasi lain berdasarkan *tunneling probability* p_t

Island model EAs



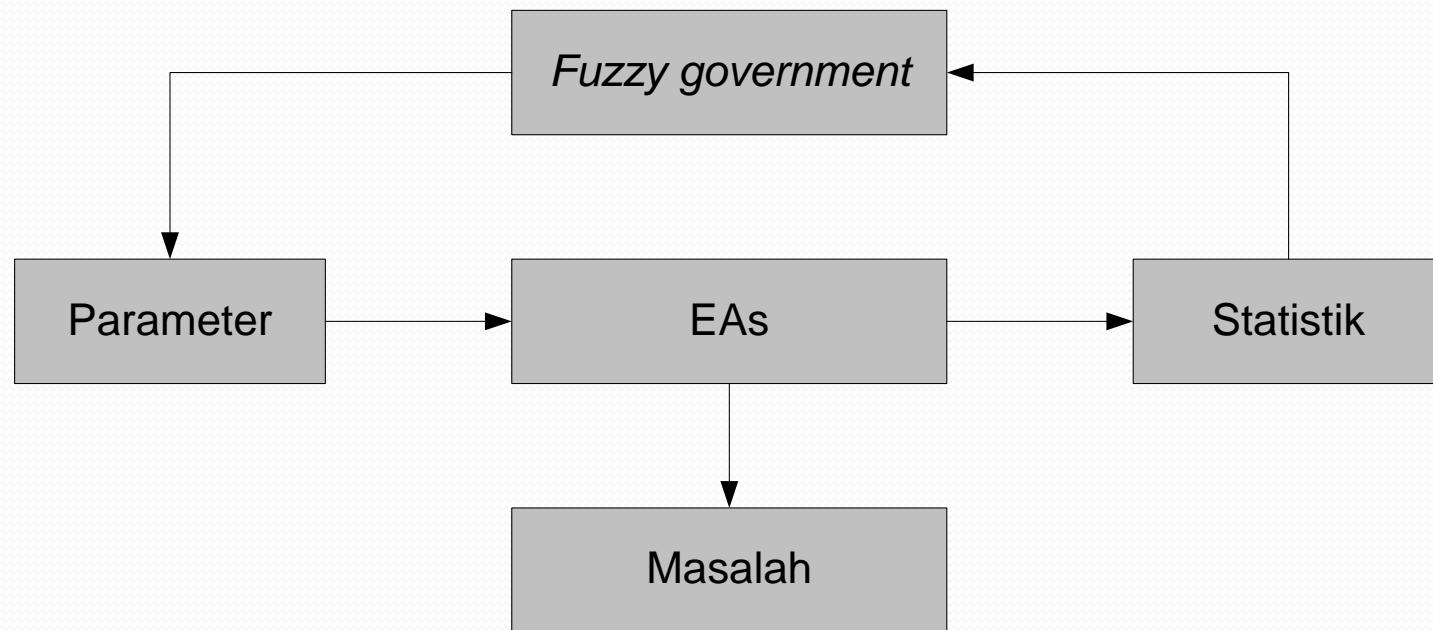
Island model EAs



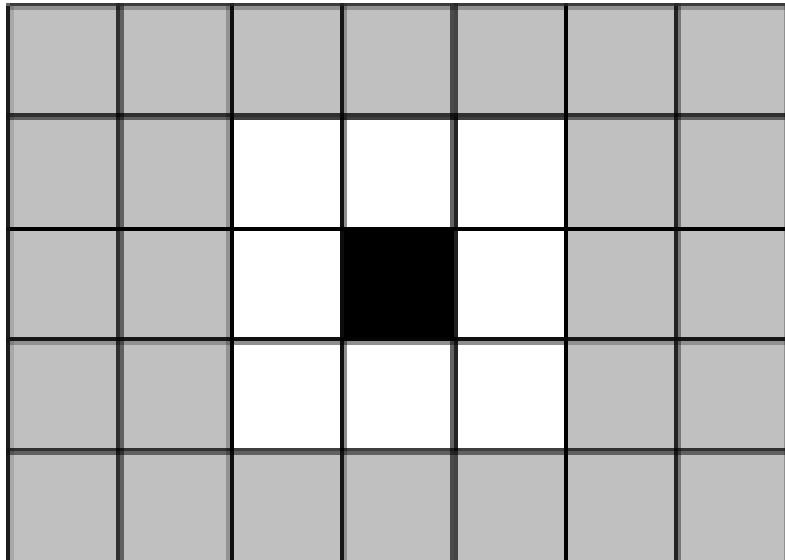


The best individual

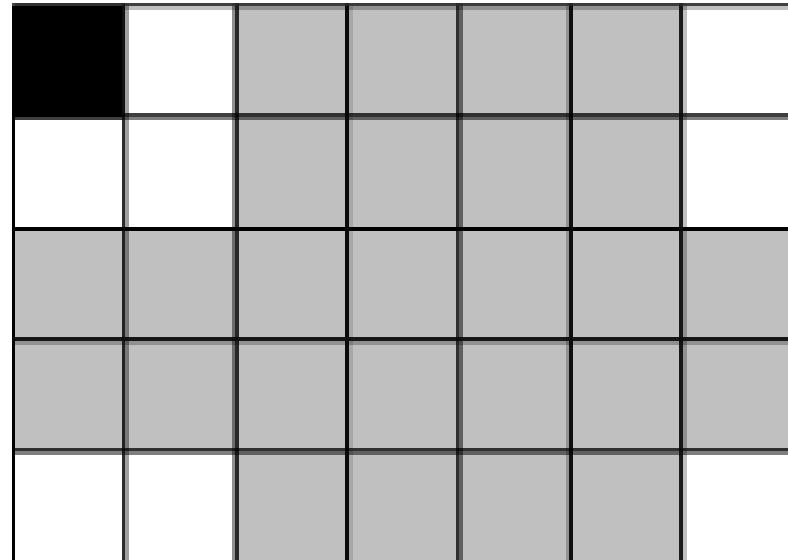
Adaptive EAs



Grid-based Crossover



(a)



(b)

- Individu-individu diletakkan dalam suatu ***toroidal space***, dimana ujung-ujung kotak tersebut disatukan dan membentuk ruang tiga dimensi seperti bola.
- Individu hanya bisa di-crossover dengan individu2 tetangganya.
- Individu hitam hanya boleh crossover dengan 1 dari 8 individu tetangga.

100	91	92	93	94	95	96	97	98	99	100	91
10	1	2	3	4	5	6	7	8	9	10	1
20	11	12	13	14	15	16	17	18	19	20	11
30	21	22	23	24	25	26	27	28	29	30	21
40	31	32	33	34	35	36	37	38	39	40	31
50	41	42	43	44	45	46	47	48	49	50	41
60	51	52	53	54	55	56	57	58	59	60	51
70	61	62	63	64	65	66	67	68	69	70	61
80	71	72	73	74	75	76	77	78	79	80	71
90	81	82	83	84	85	86	87	88	89	90	81
100	91	92	93	94	95	96	97	98	99	100	91
10	1	2	3	4	5	6	7	8	9	10	1

GA untuk melatih FFNN (MLP)

- Menemukan weights secara otomatis
- Menggunakan representasi biner
- Bagaimana performansi Advanced GA?

Masalah 3-Parity

Tabel Kebenaran XOR untuk tiga masukan X_1 , X_2 , dan X_3 .

X_1	X_2	X_3	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

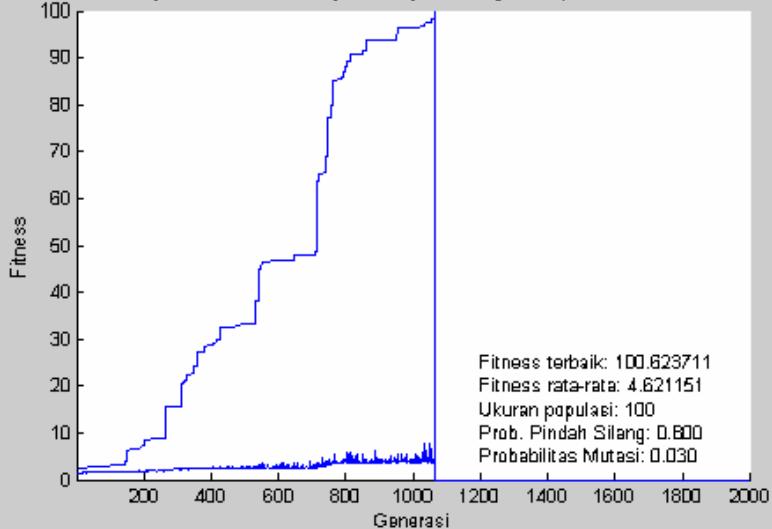
Fungsi Fitness

$$f = 1/delta$$

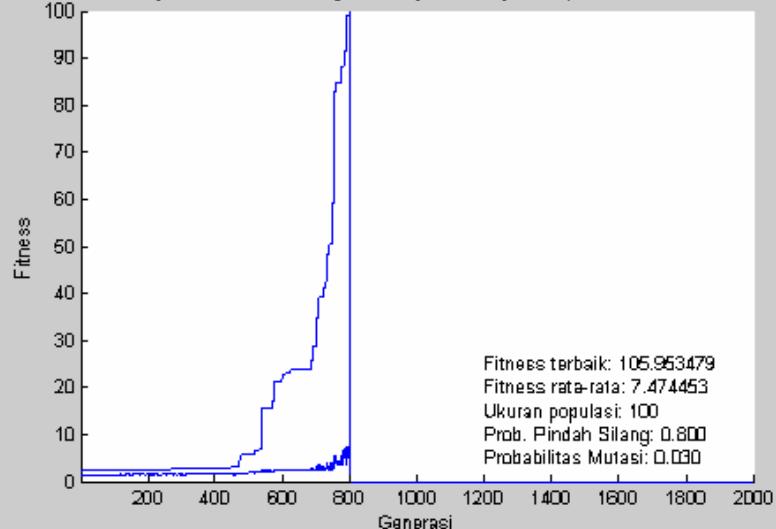
$$\text{delta} = \sqrt{\frac{1}{m} \sum_{i=1}^m (d(i) - y(i))^2}$$

- $m = 2^n = 2^3 = 8$
- $d(i)$ adalah output yang diharapkan.
- Pola masukan pertama, $X_1 = 0$, $X_2 = 0$, dan $X_3 = 0$, maka $d(1) = 0$.
- Pola masukan ke dua, $X_1 = 0$, $X_2 = 0$, dan $X_3 = 1$, maka $d(2) = 1$, dst.
- $y(i)$ adalah output aktual yang dihasilkan.

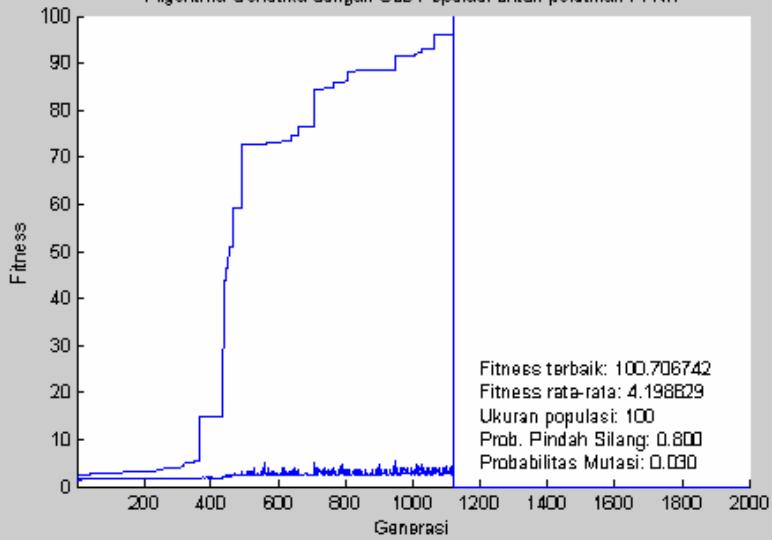
Algoritma Genetika dengan Binary Encoding untuk pelatihan FFNN



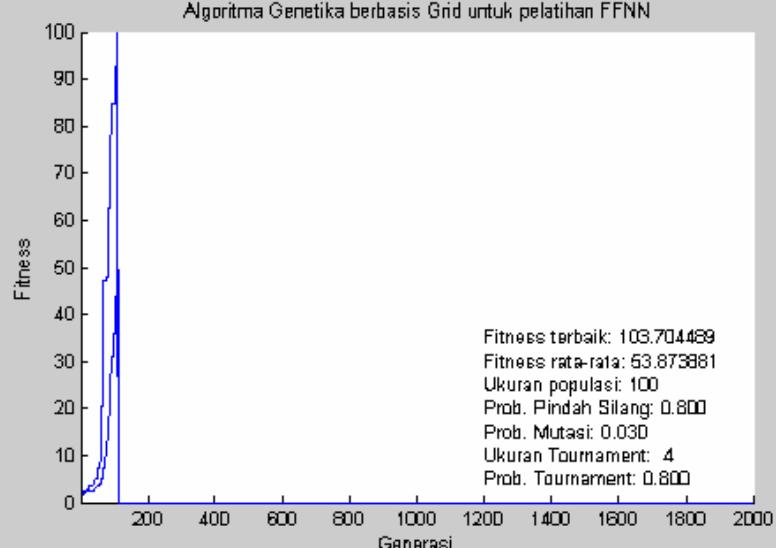
Algoritma Genetika dengan Messy Encoding untuk pelatihan FFNN



Algoritma Genetika dengan Sub Populasi untuk pelatihan FFNN



Algoritma Genetika berbasis Grid untuk pelatihan FFNN



Grammatical Encoding

- Otak manusia merupakan suatu komputer sangat kompleks yang terdiri dari sekitar 10^{11} elemen komputasi (*neurons*).
- Terdapat sekitar 10^{14} sampai 10^{15} koneksi antar *neurons*, atau sekitar 1000 sampai 10000 koneksi per *neuron*.
- Jika setiap koneksi dikodekan ke dalam kromosom, maka informasi yang mengisi kromosom akan sekitar **10⁵ GB**, dimana bobot-bobot sinaptik dikodekan menggunakan hanya 1 *byte*.
- Tetapi, pada kenyataanya ukuran *genome* manusia hanya sekitar **3 GB**.
- Oleh karena itu para peneliti percaya bahwa pengkodean otak manusia bukanlah menggunakan pengkodean langsung, melainkan pengkodean **prosedur** dimana otak dibentuk.

Grammatical Encoding

- Pada skema ini, kromosom dipandang sebagai kalimat yang diekspresikan menggunakan *grammar* (tata bahasa).
- Ketika sebuah kalimat dibaca (kromosom didekodekan), maka individu dibangkitkan menggunakan *grammar* tsb.
- Contoh: skema Kitano yang digunakan untuk mengkodekan ANN yang berisi maksimum 8 *neurons*

Skema Kitano

S	ACBA	Aadfb	Bbefd	Dfanp	Bahjm	Ckhgf	...
---	------	-------	-------	-------	-------	-------	-----

$$S \rightarrow \begin{pmatrix} A & C \\ B & A \end{pmatrix}$$

$$A \rightarrow \begin{pmatrix} a & d \\ f & b \end{pmatrix} \quad B \rightarrow \begin{pmatrix} b & e \\ f & d \end{pmatrix}$$

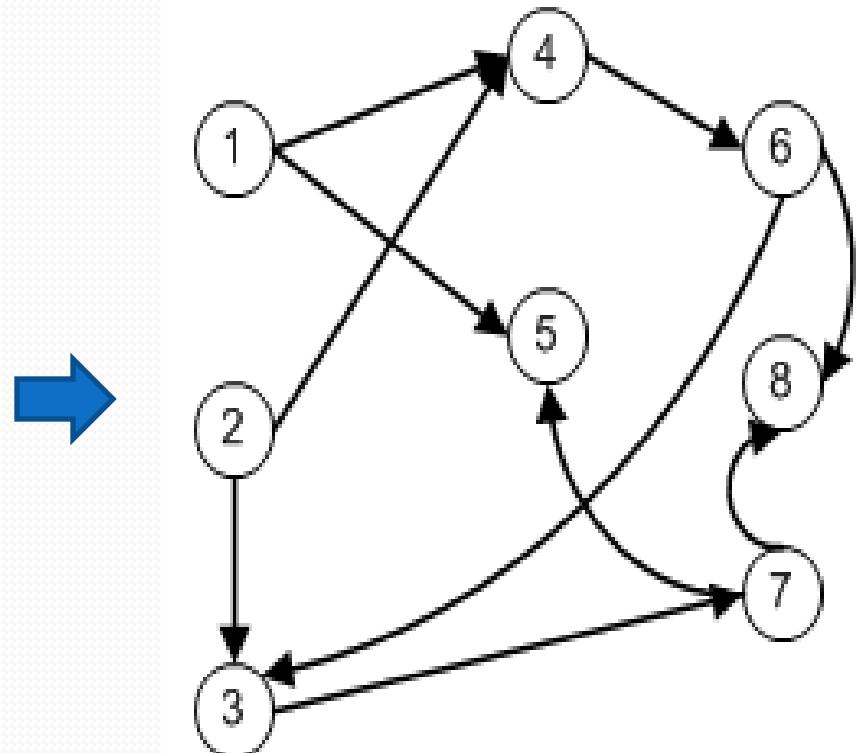
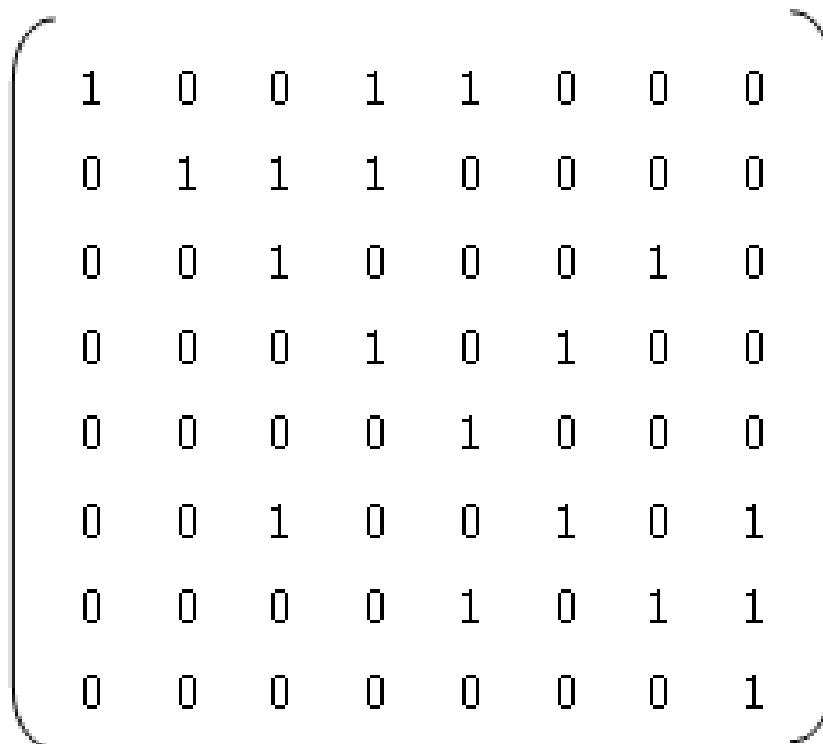
$$a \rightarrow \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, b \rightarrow \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, c \rightarrow \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \dots p \rightarrow \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Skema Kitano

S	ACBA	Aadfb	Bbefd	Dfanp	Bahjm	Ckhgf	...
---	------	-------	-------	-------	-------	-------	-----

$$\begin{pmatrix} A & C \\ B & A \end{pmatrix} \xrightarrow{\quad} \begin{pmatrix} a & d & k & h \\ f & b & g & f \\ b & e & a & d \\ f & d & f & b \end{pmatrix} \xrightarrow{\quad} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Skema Kitano



Demo Program

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.

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

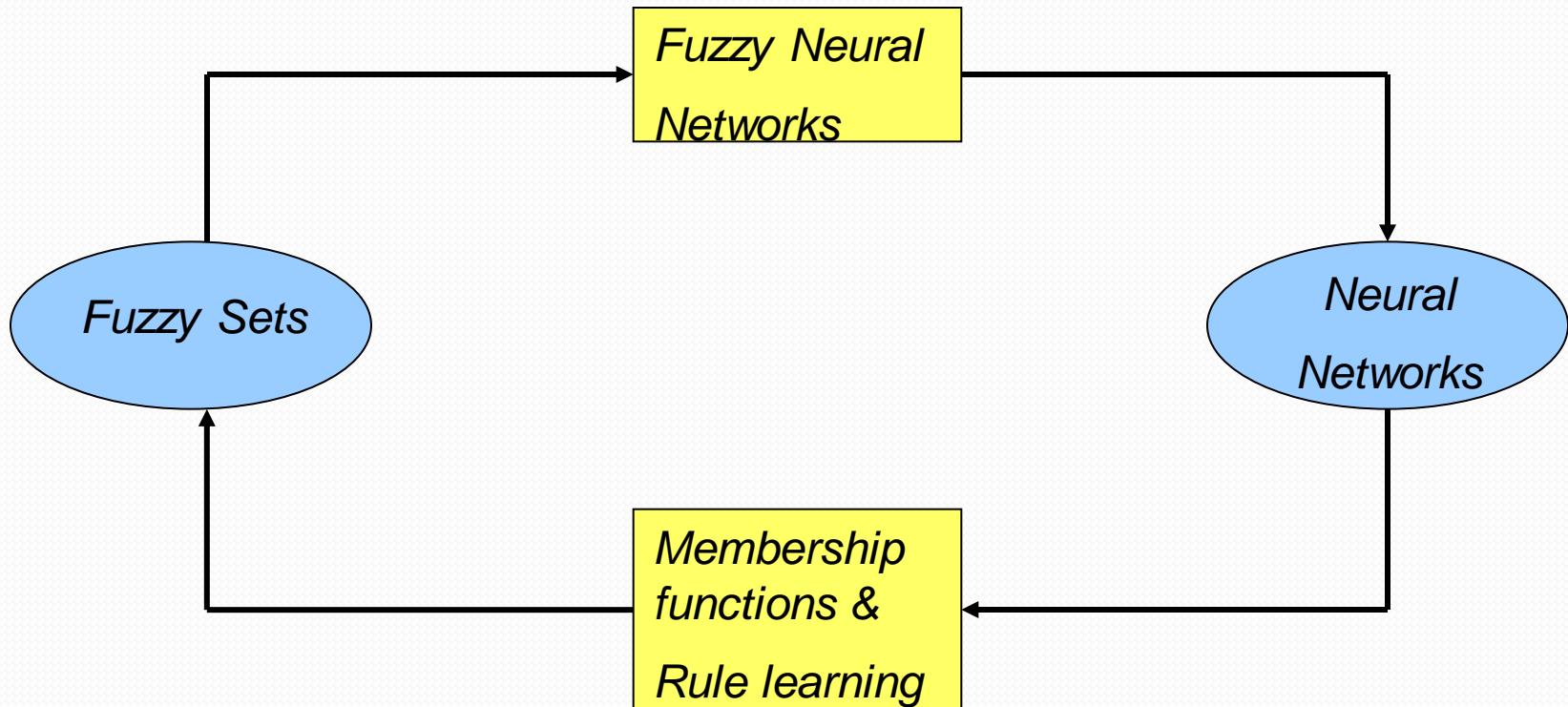
Permasalahan *Fuzzy Systems*

- 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]



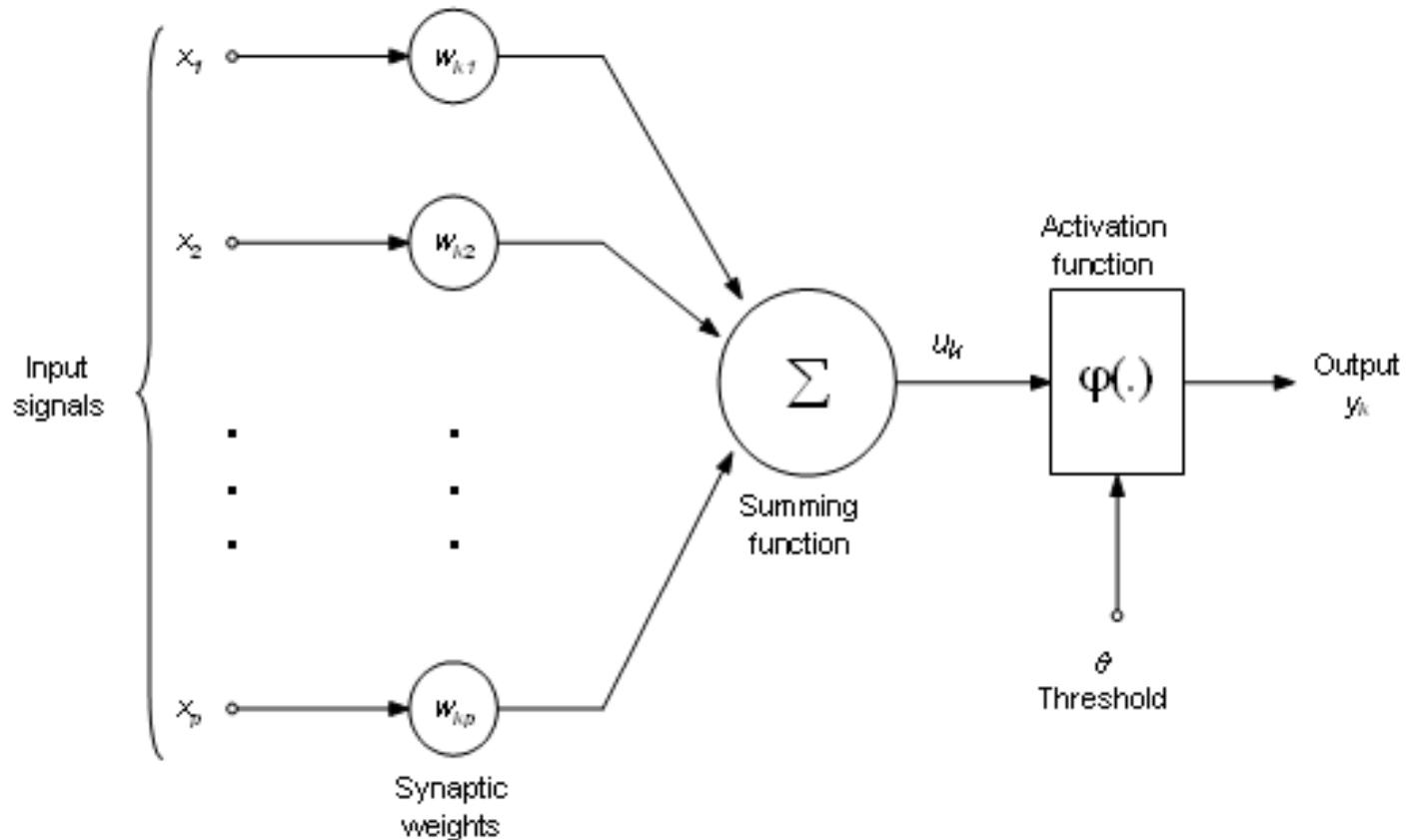
Tiga kombinasi ANN & Fuzzy [TET01]

- ***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

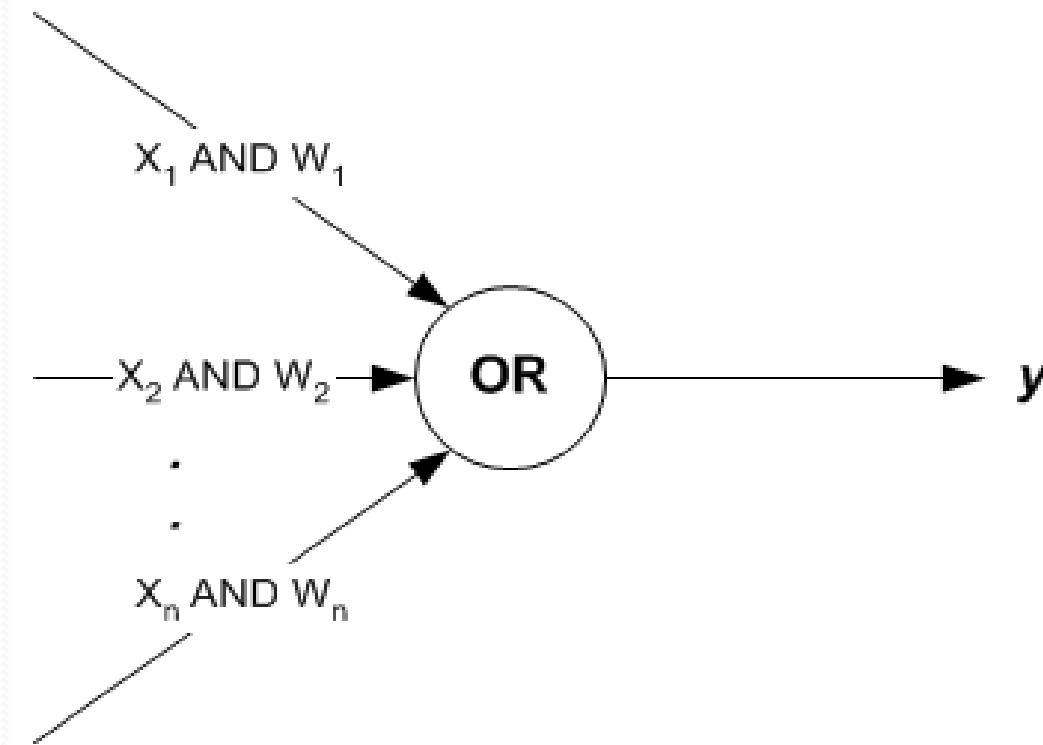
- *Fuzziness* dapat aplikasikan pada ANN dalam beberapa cara:
 - *Fuzzy neuron*
 - *Multilayered Fuzzy Neural Network*

Neuron konvensional



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

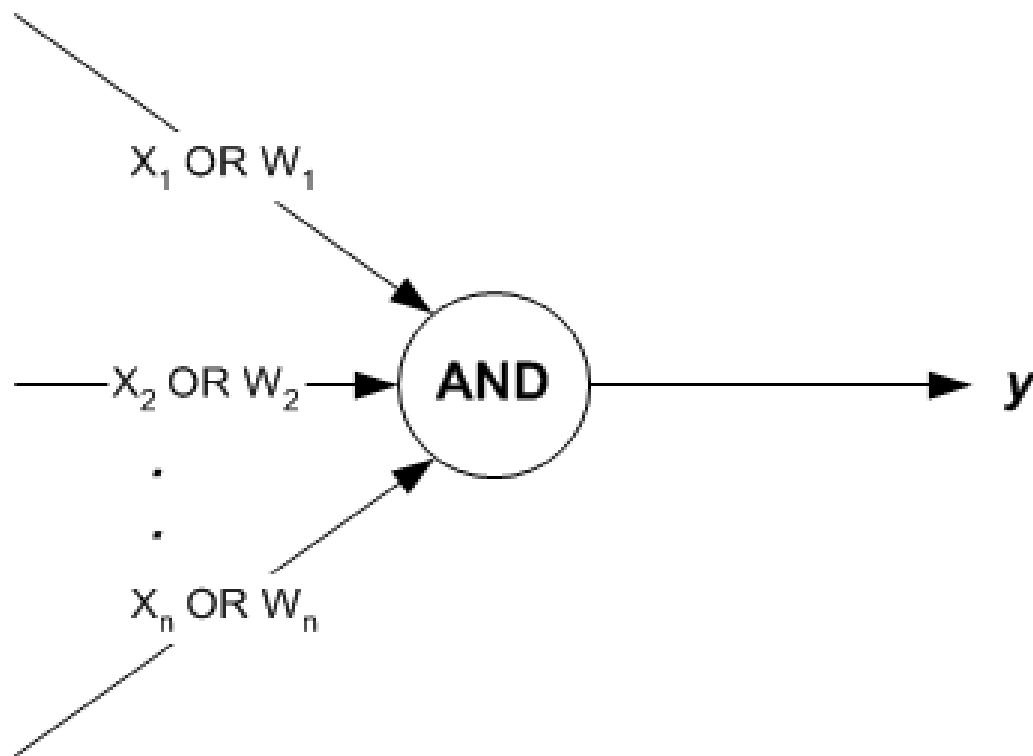
OR fuzzy neuron [TET01]



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

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

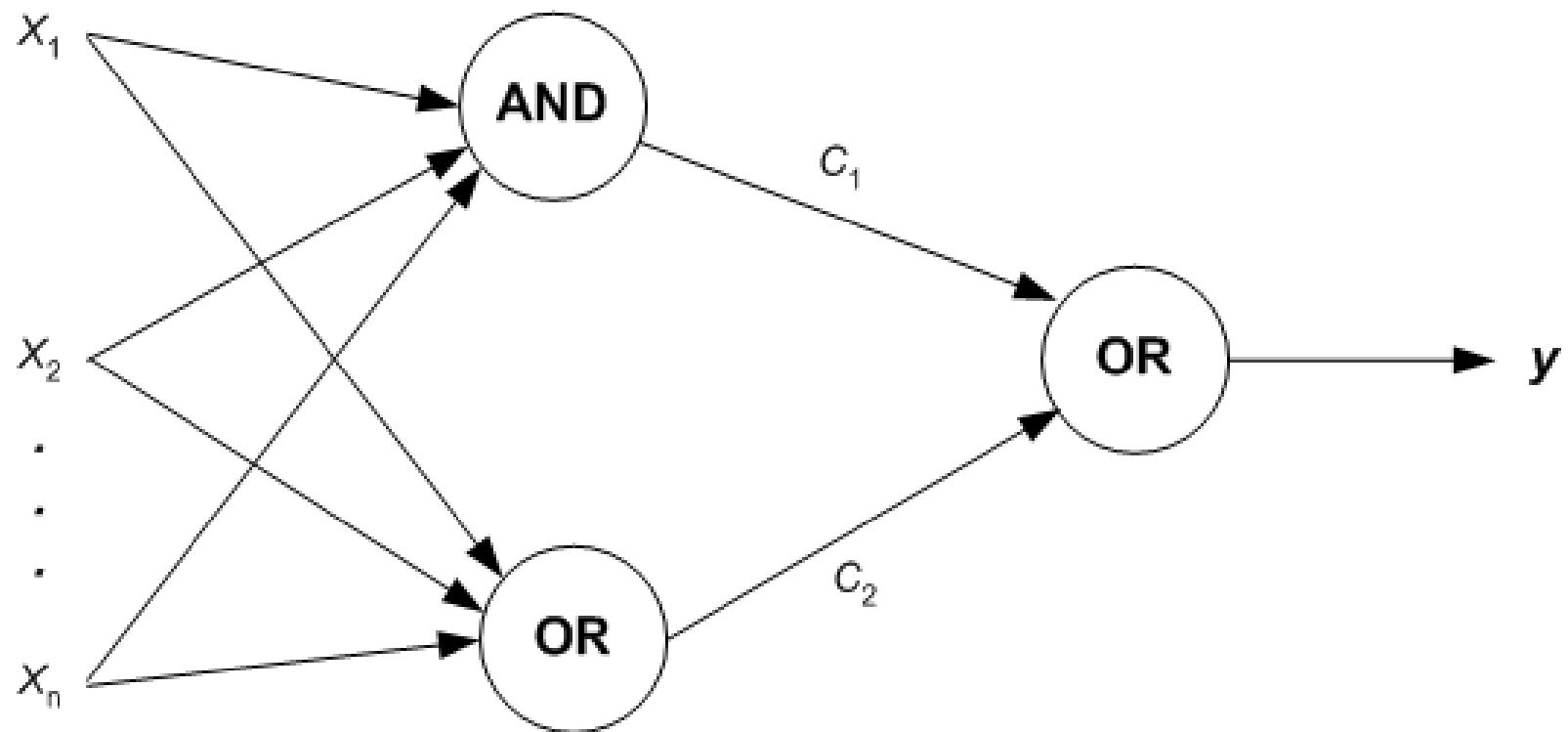
AND fuzzy neuron [TET01]



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

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

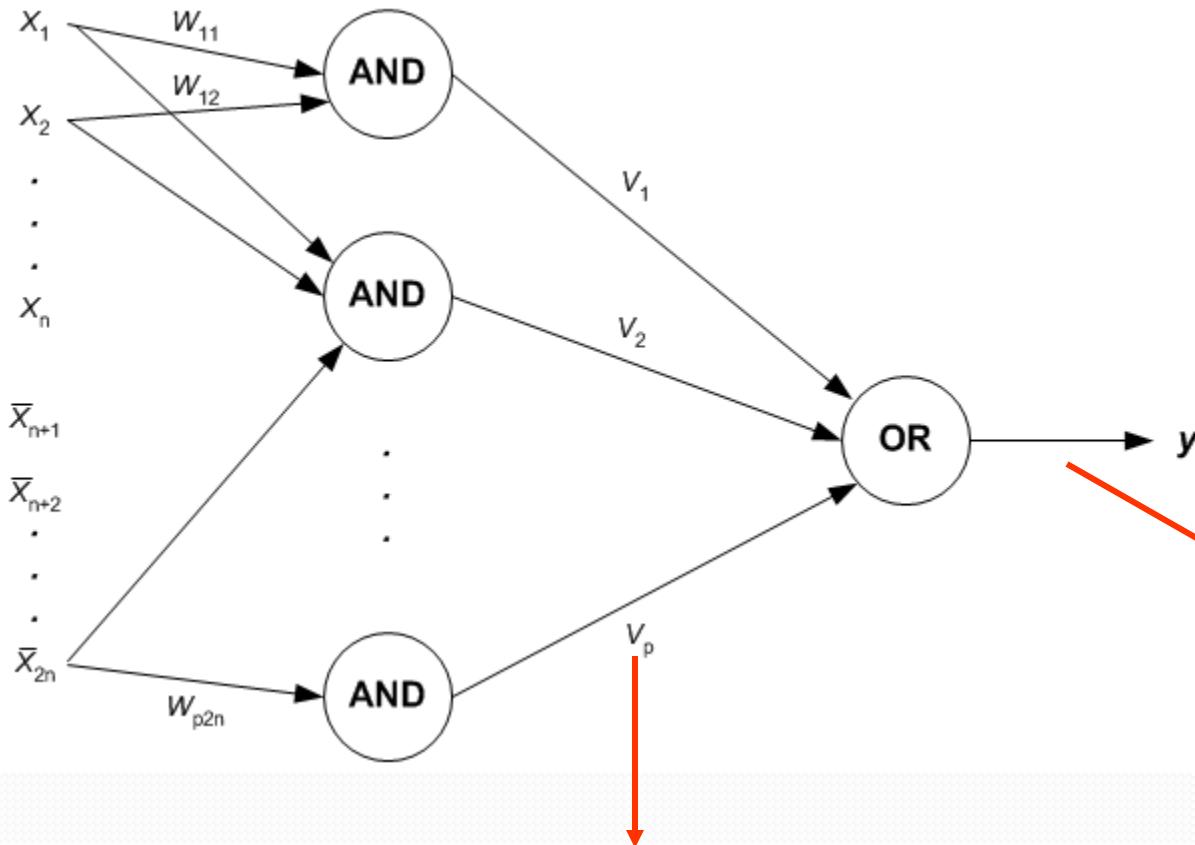
OR/AND fuzzy neuron



Jika C₁ = 1 dan C₂ = 0, maka akan menjadi AND fuzzy neuron

Jika C₁ = 0 dan C₂ = 1, maka akan menjadi OR fuzzy neuron

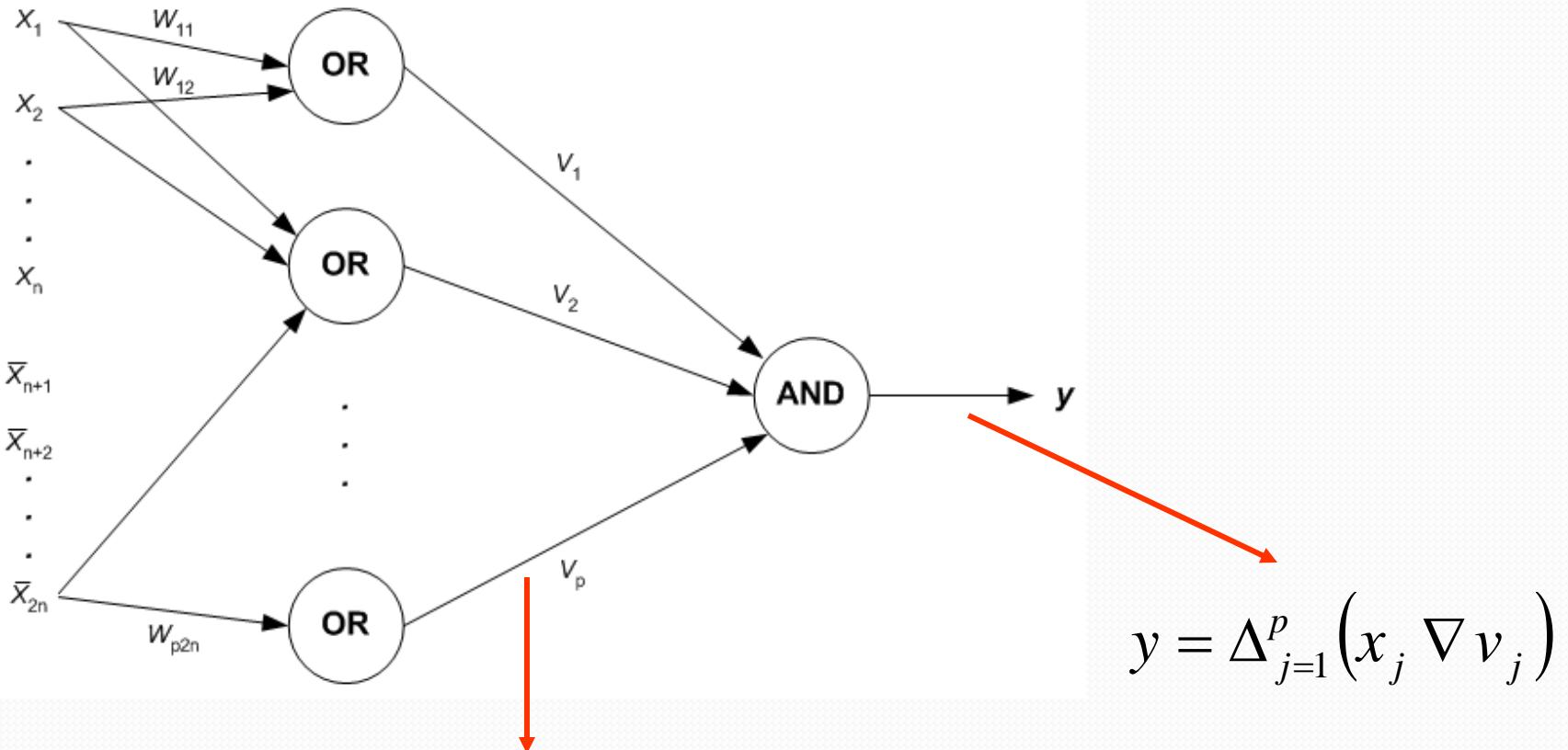
3 Layer ANN dengan *AND fuzzy units* pada *hidden layer*



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

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

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



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

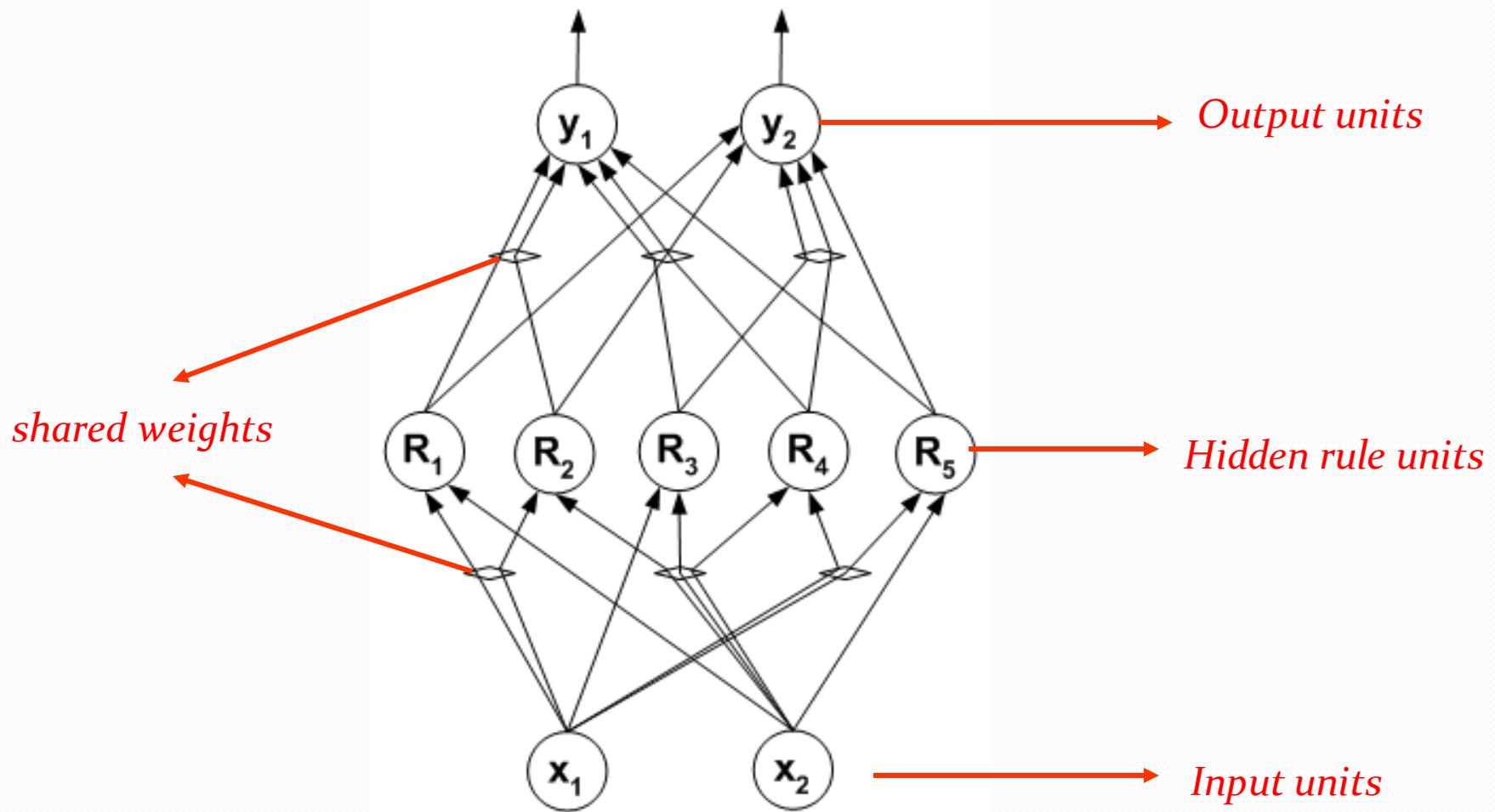
NEFPROX (NEuro-Fuzzy function apPROXimator)

- 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

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

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]

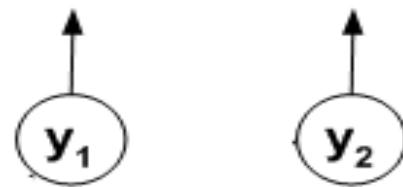
4. Untuk setiap koneksi dari *rule node* baru ke *output nodes*, cari fuzzy weight $v_{ji}^{(i)}$ yang sesuai menggunakan fungsi-fungsi keanggotaan yang *assigned* ke output units y_i sehingga

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

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

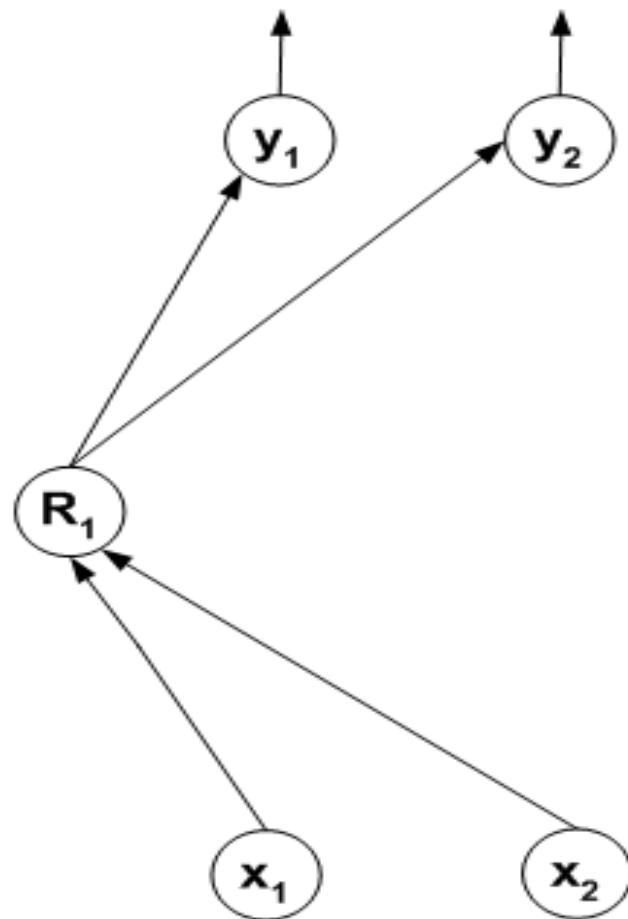
Ilustrasi (0)

Structure-learning algorithm



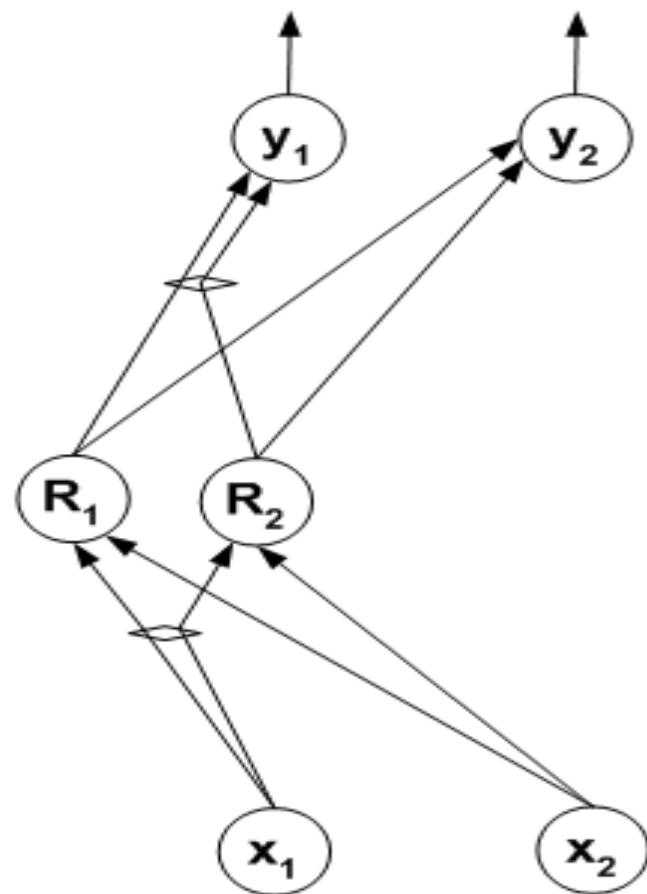
Ilustrasi (1)

Structure-learning algorithm



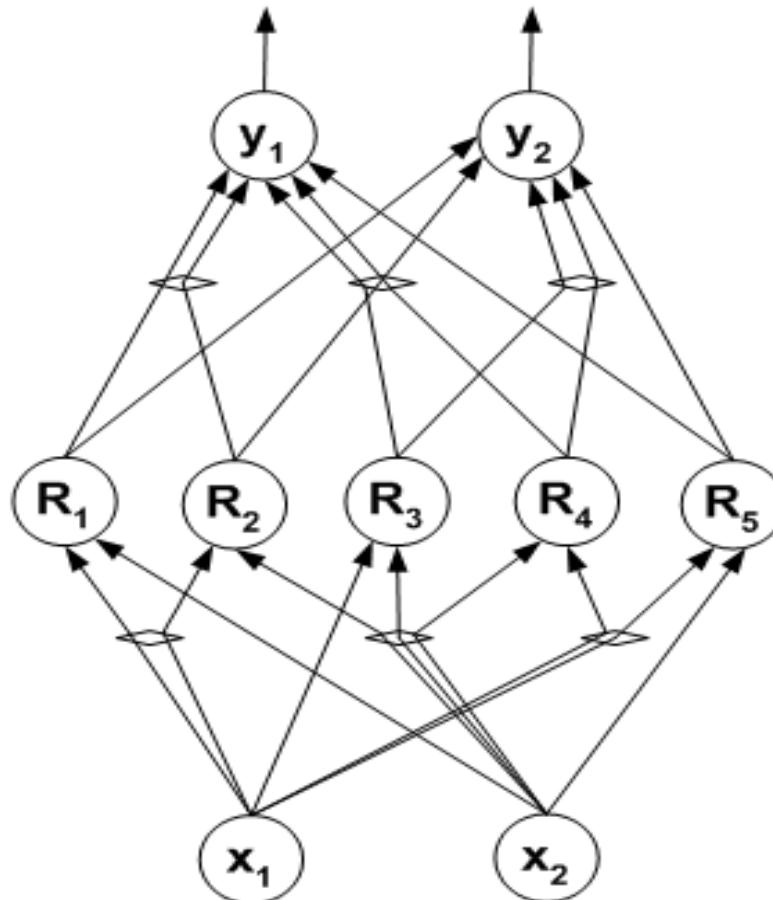
Ilustrasi (2)

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 $\mathbf{o}_R > \mathbf{o}$ lakukan

- *Update* parameter2 himpunan fuzzy $W(R, y_i)$ menggunakan *learning rate* $\sigma > 0$
- *Tentukan perubahan* $\delta_R = o_R(1 - o_R) \bullet \sum_{y \in \text{outputlayer}} (2W(R, y)(t_i) - 1) \bullet |\delta_y|$
- *Update* parameter2 himpunan fuzzy $W(x, R)$ menggunakan δ_R dan σ untuk menghitung variasi-variasi tersebut.

5. Jika kriteria konvergen telah tercapai, maka berhenti. Jika tidak, kembali ke langkah 1.

Parameter-learning algorithm

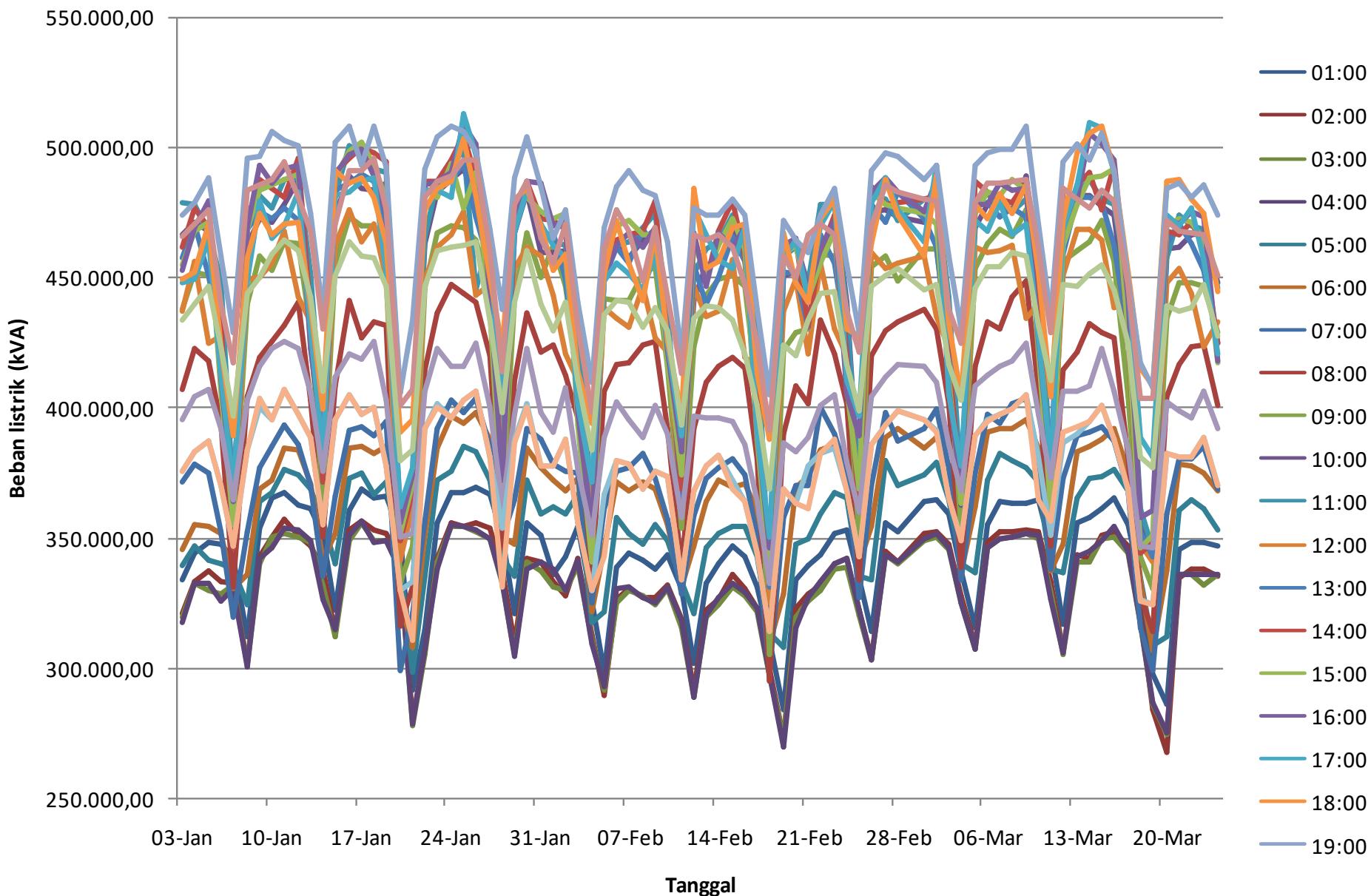
- Prosedur belajar berbasis pada *simple heuristic method*.
- Prosedur ini menghasilkan pergeseran fungsi-fungsi keanggotaan.

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?

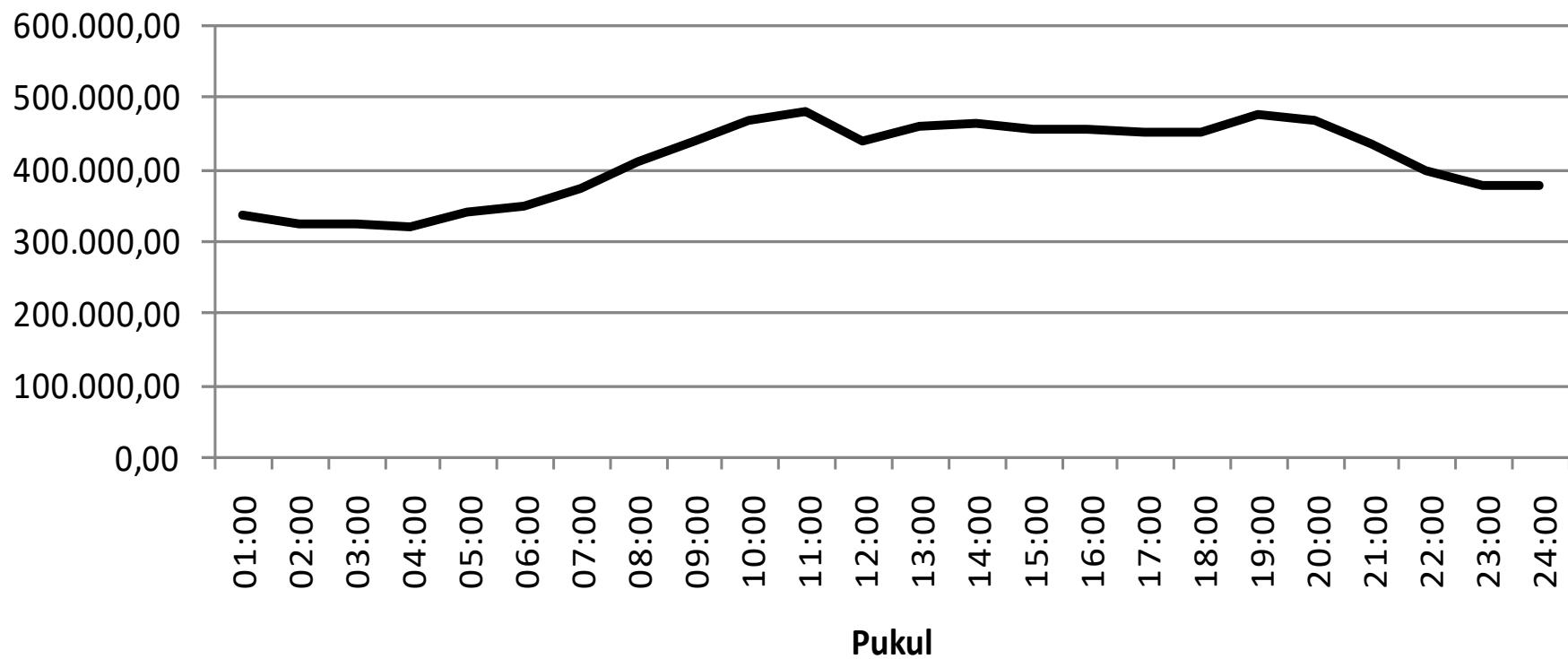
Data Beban Listrik per jam (pukul 01:00 - 24:00)

Periode 03 Januari - 24 Maret 2007 di suatu APJ

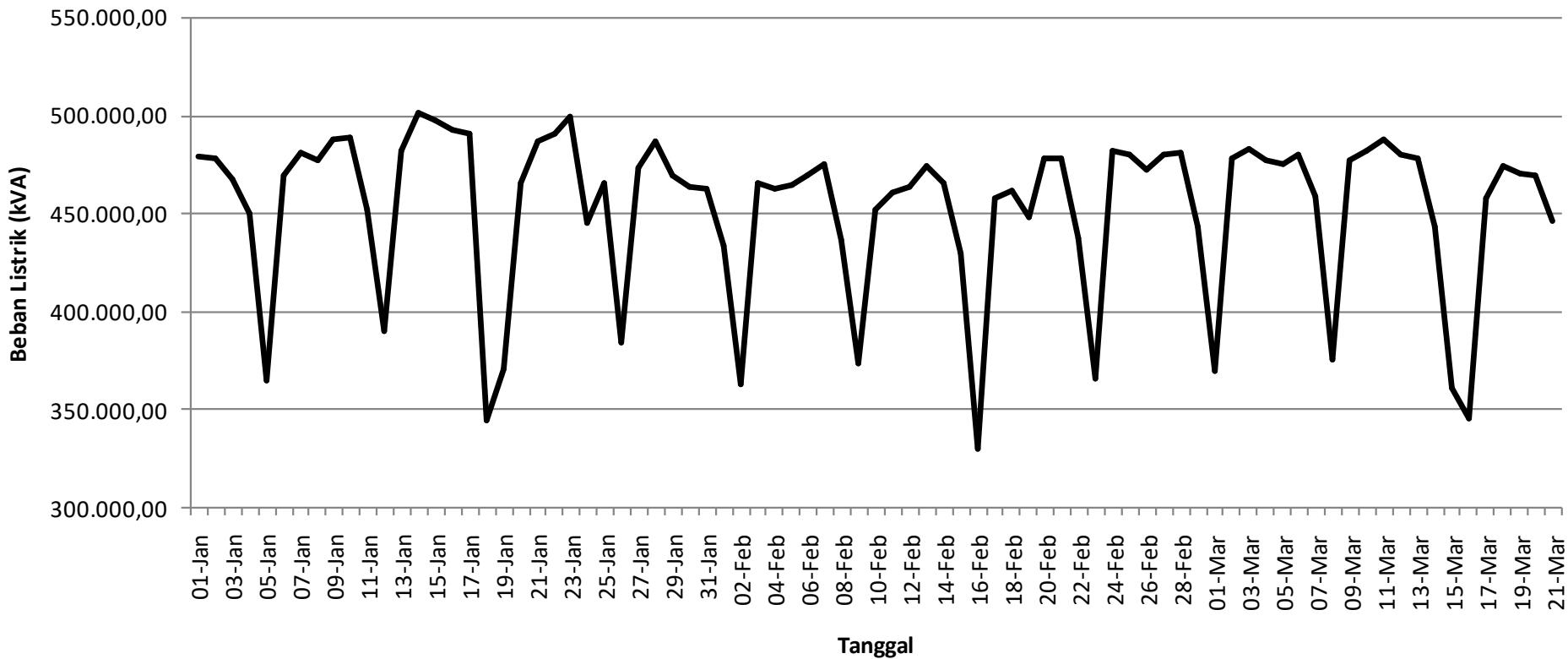


Beban listrik pada tanggal 03 Jan 2007

Beban Listrik (kVA)



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 (σ)* = 0; 0,0025; 0,006; dan 0,01
- FK memiliki *intersect* 0,5 (setengah bagian overlap)
- Domain fungsi keanggotaan dalam interval tertentu

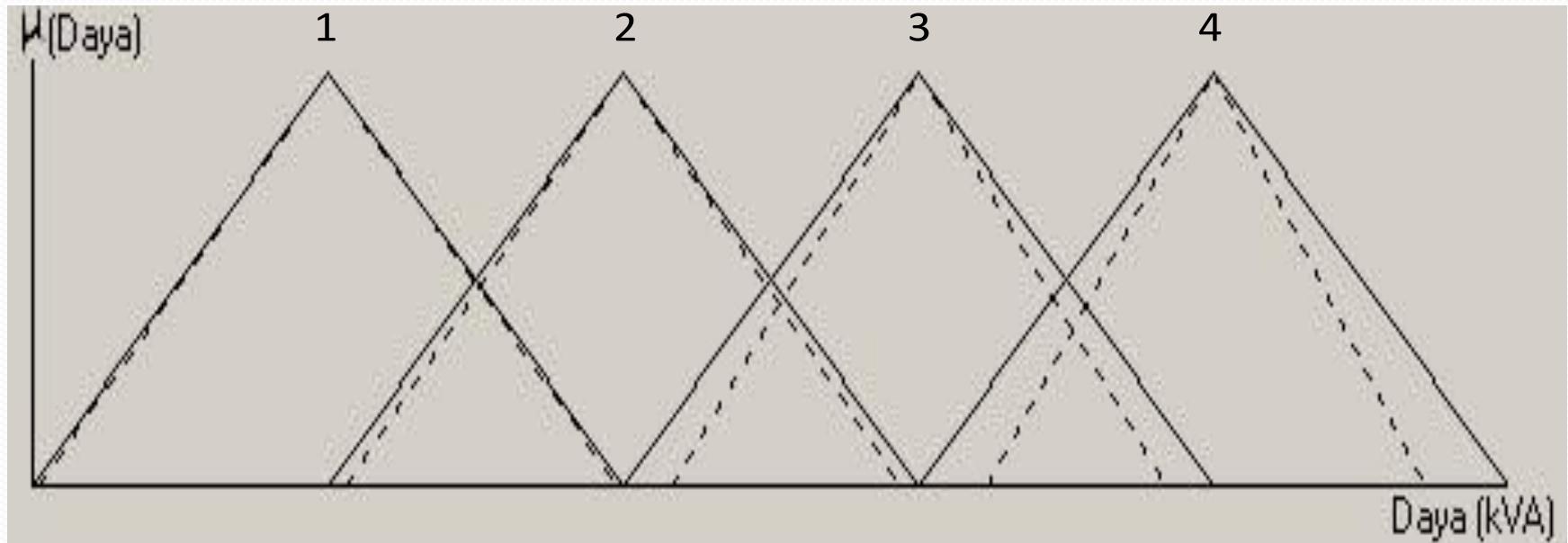
Pukul	Domain (kVA)	Pukul	Domain (kVA)
01:00	270000 - 390000	13:00	320000 - 510000
02:00	250000 - 370000	14:00	320000 - 535000
03:00	250000 - 370000	15:00	320000 - 535000
04:00	250000 - 370000	16:00	330000 - 530000
05:00	280000 - 415000	17:00	340000 - 540000
06:00	275000 - 420000	18:00	375000 - 525000
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 Fuzzy Set optimal	σ	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 Input	σ	MAPE (%)					
		Pukul 01:00		Pukul 02:00		Pukul 03:00	
		Jumlah fuzzy set		Jumlah fuzzy set		Jumlah fuzzy set	
		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 Input	σ	MAPE (%)					
		Pukul 07:00		Pukul 08:00		Pukul 09:00	
		Jumlah fuzzy set		Jumlah fuzzy set		Jumlah fuzzy set	
		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 Input	σ	MAPE (%)					
		Pukul 19:00		Pukul 20:00		Pukul 21:00	
		Jumlah fuzzy set		Jumlah fuzzy set		Jumlah fuzzy set	
		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

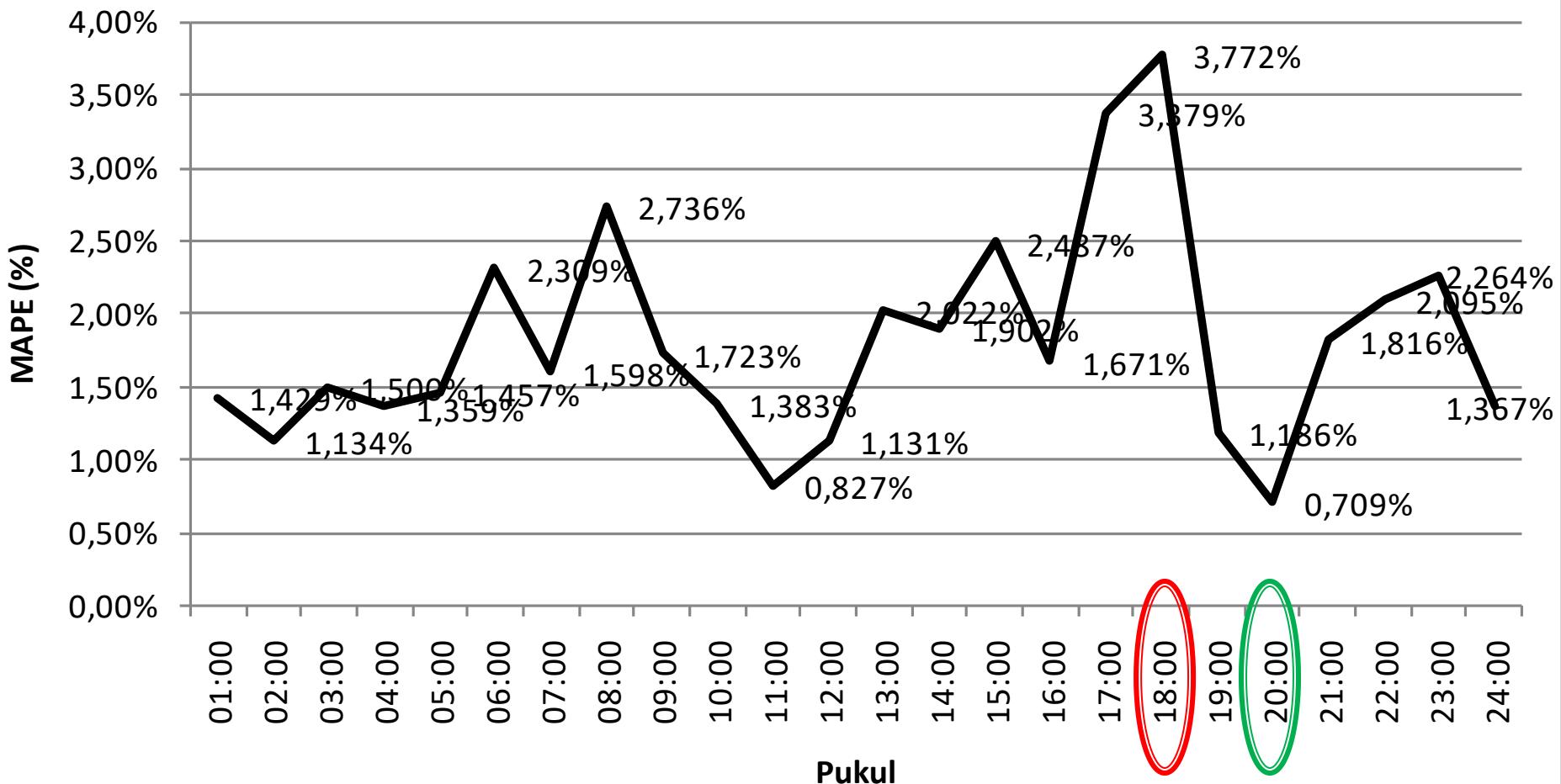
Kesalahan Prediksi dan MAPE untuk *Test Set* (11-17 MAR 2007)

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

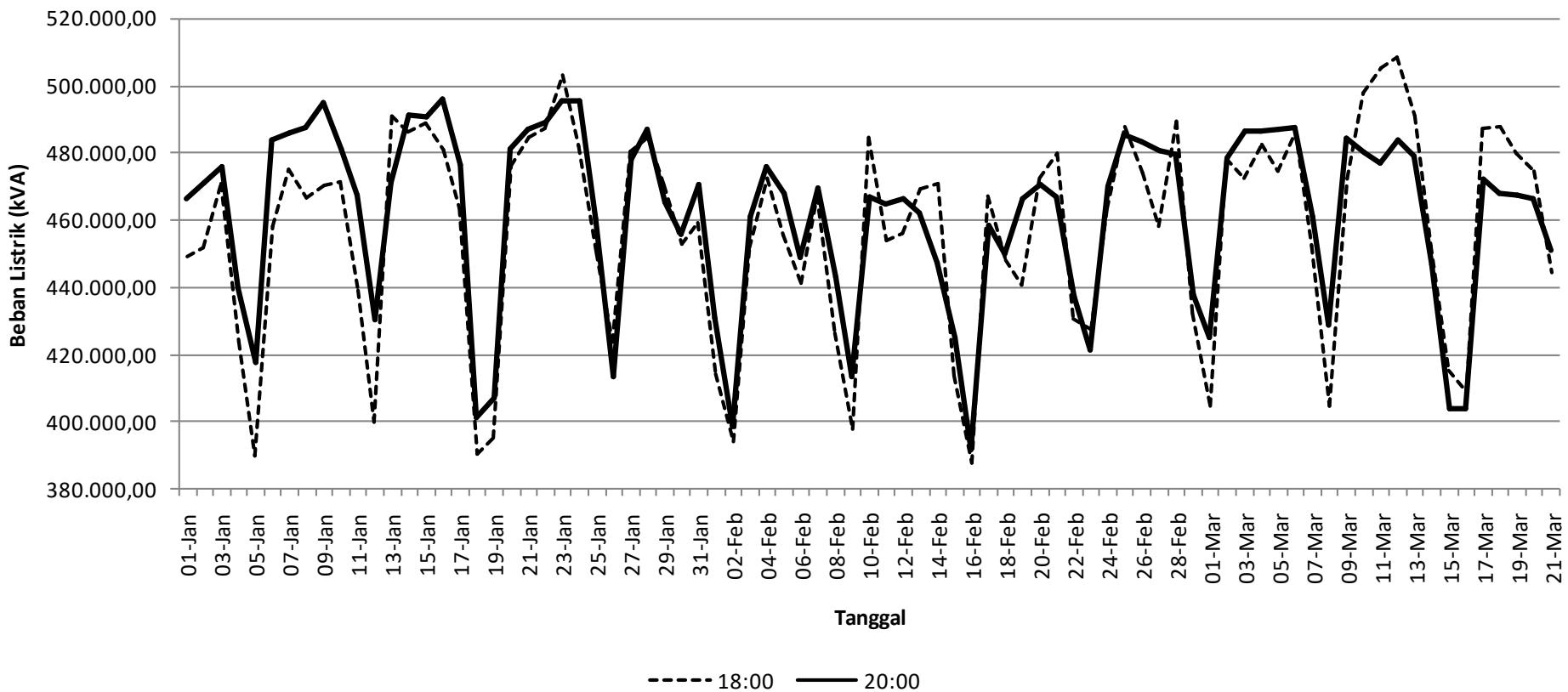
MAPE untuk data *Test Set*: 11-17 Mar 2007



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:

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

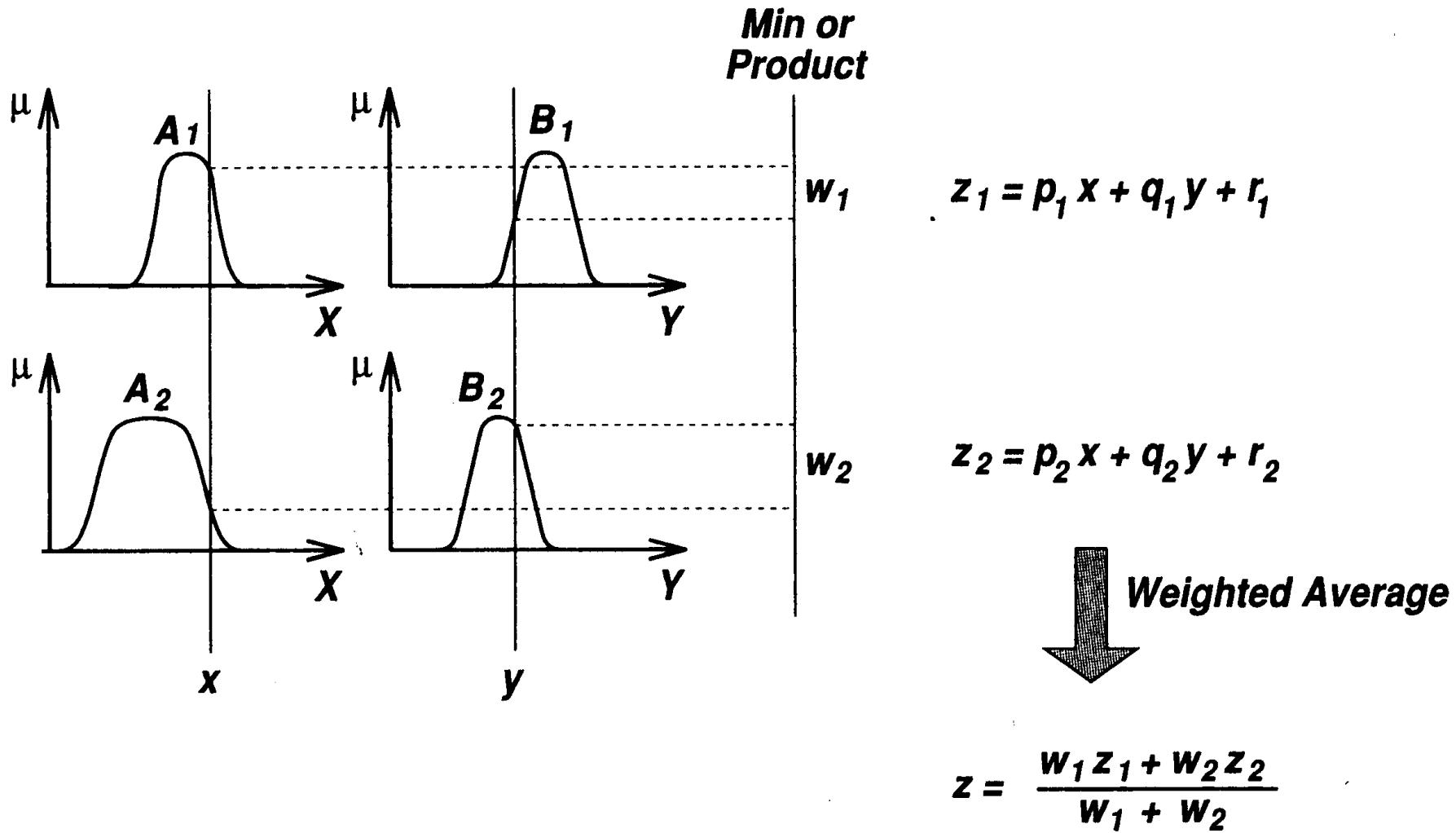
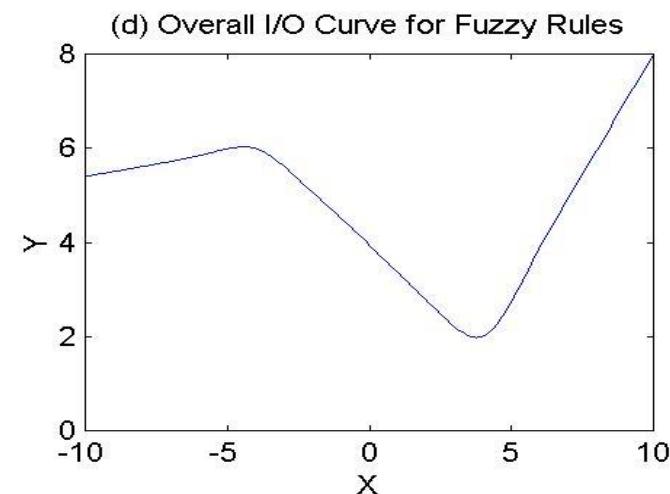
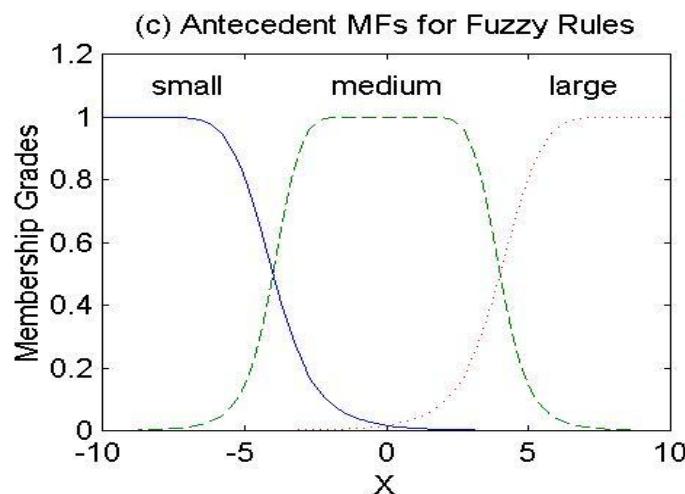
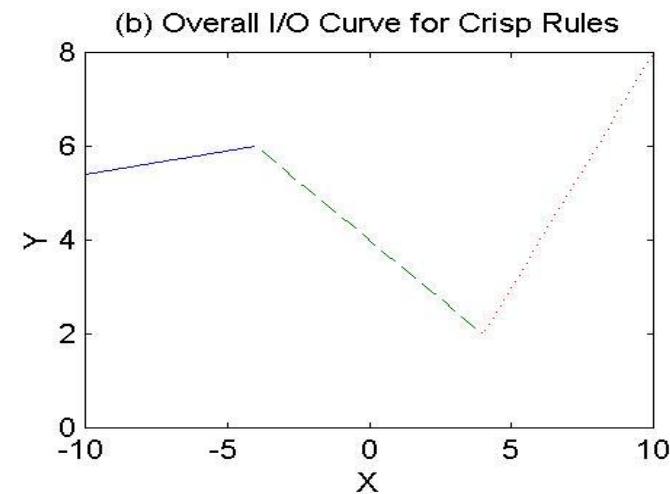
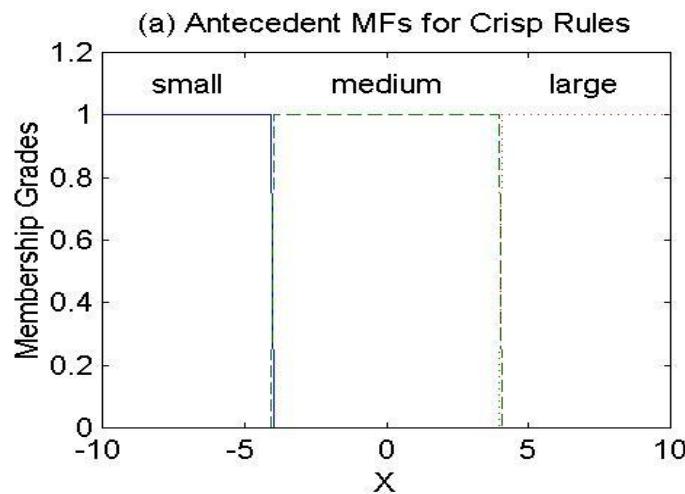


Figure 4.8. *The Sugeno fuzzy model.*

If X is small then $Y = 0.1X + 6.4$

If X is medium then $Y = -0.5X + 4$

If X is large then $Y = X - 2$

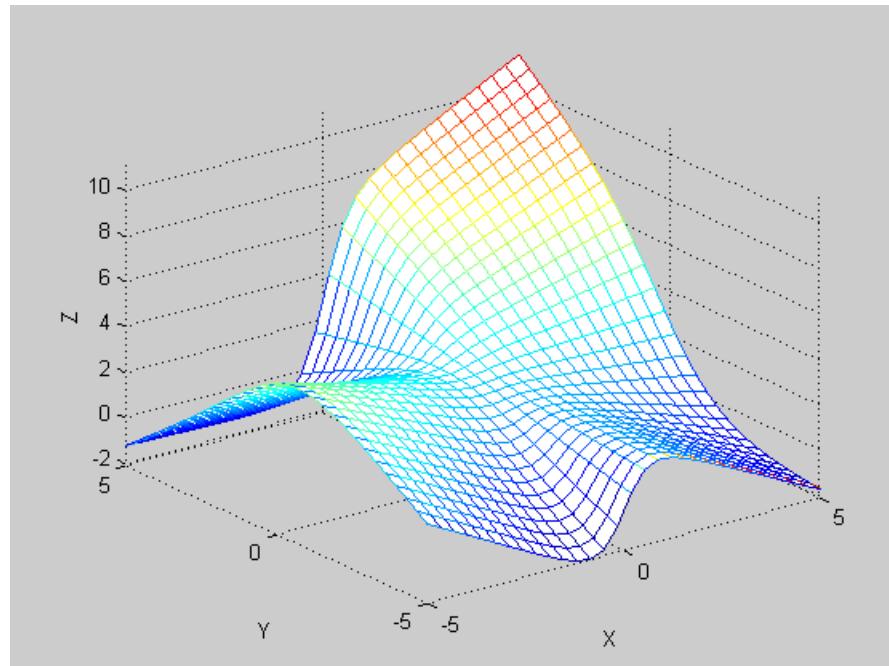
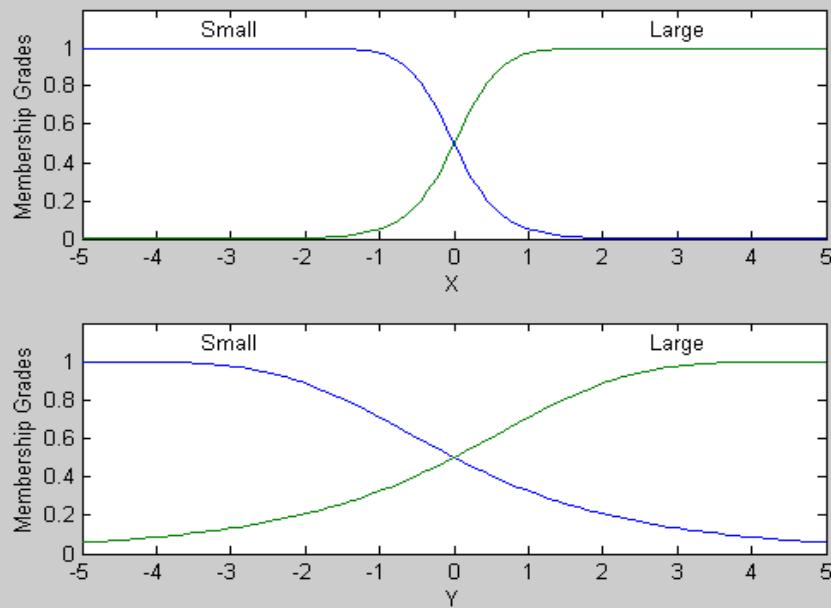


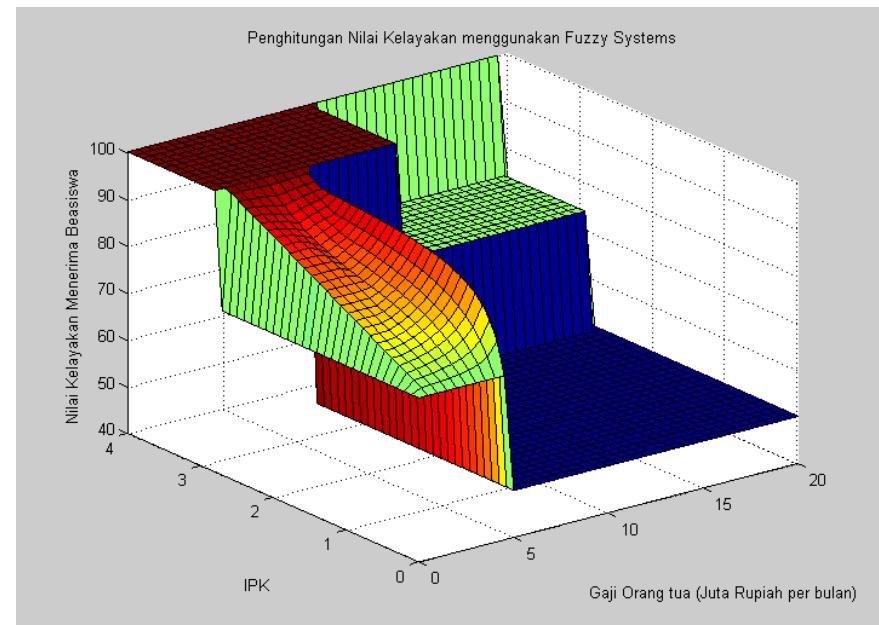
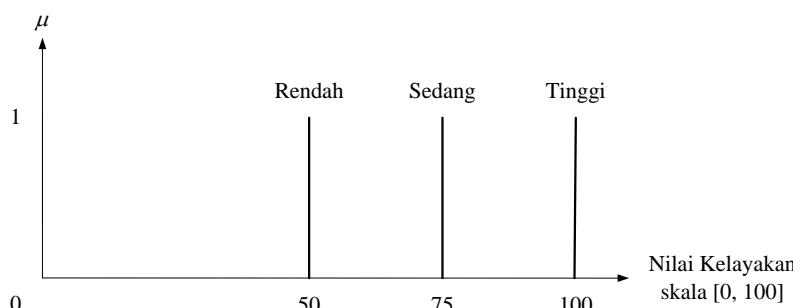
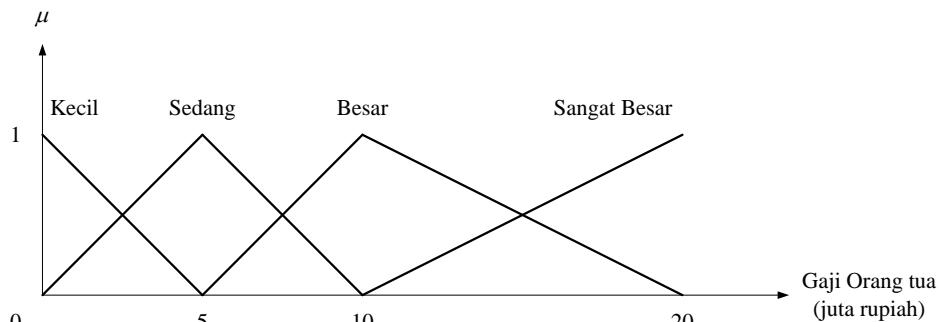
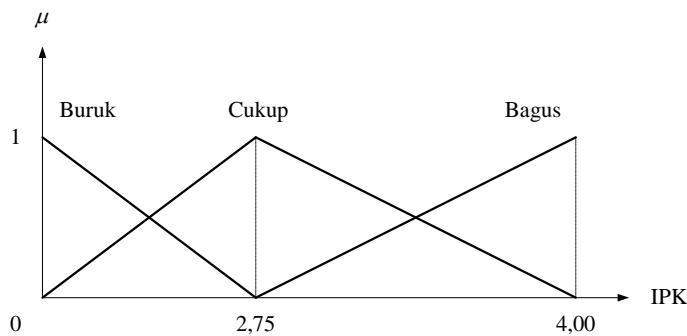
If X is small and Y is small then $z = -x+y+1$

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 $z = x+y+2$

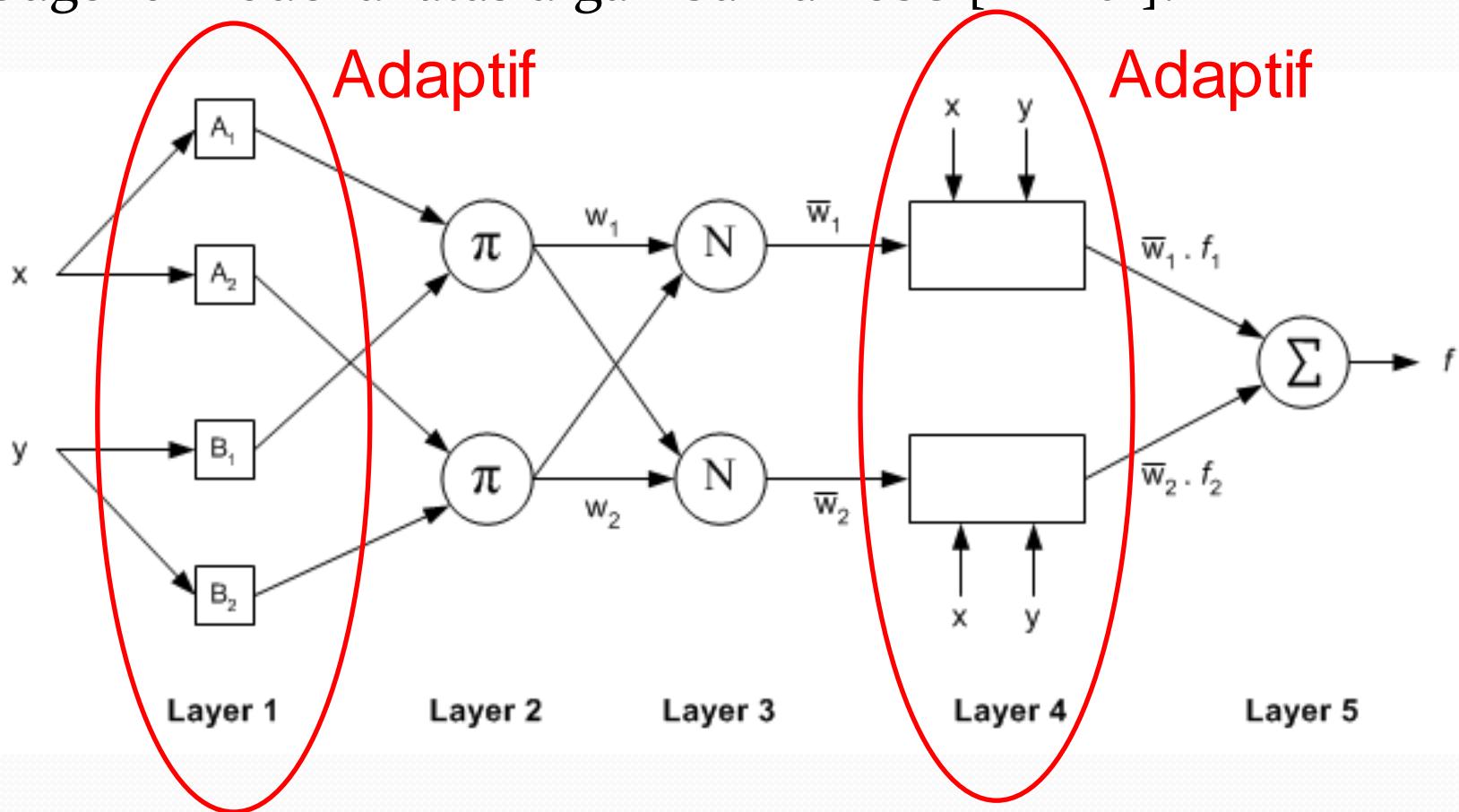




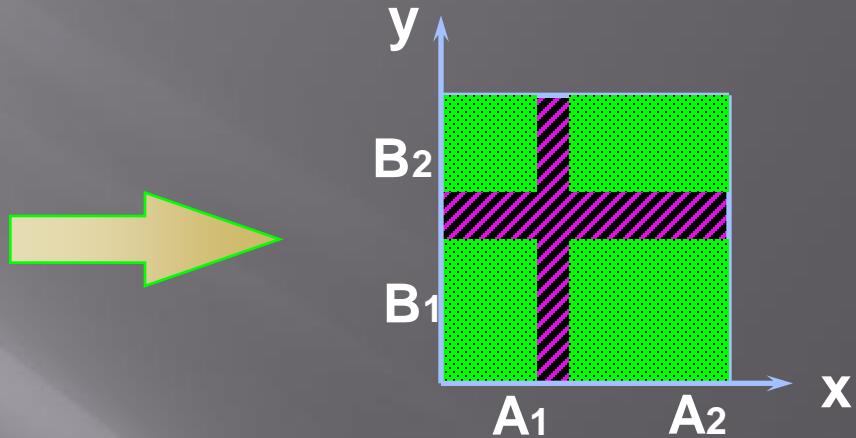
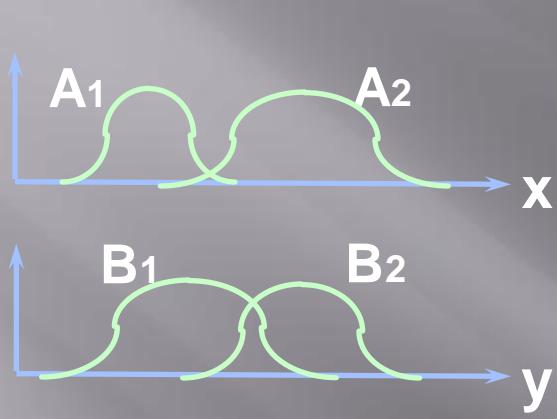
IPK \ Gaji	Kecil	Sedang	Besar	Sangat Besar
Buruk	Sedang	Rendah	Rendah	Rendah
Cukup	Tinggi	Tinggi	Sedang	Rendah
Bagus	Tinggi	Tinggi	Tinggi	Sedang

ANFIS (Adaptive Network-based Fuzzy Inference System)

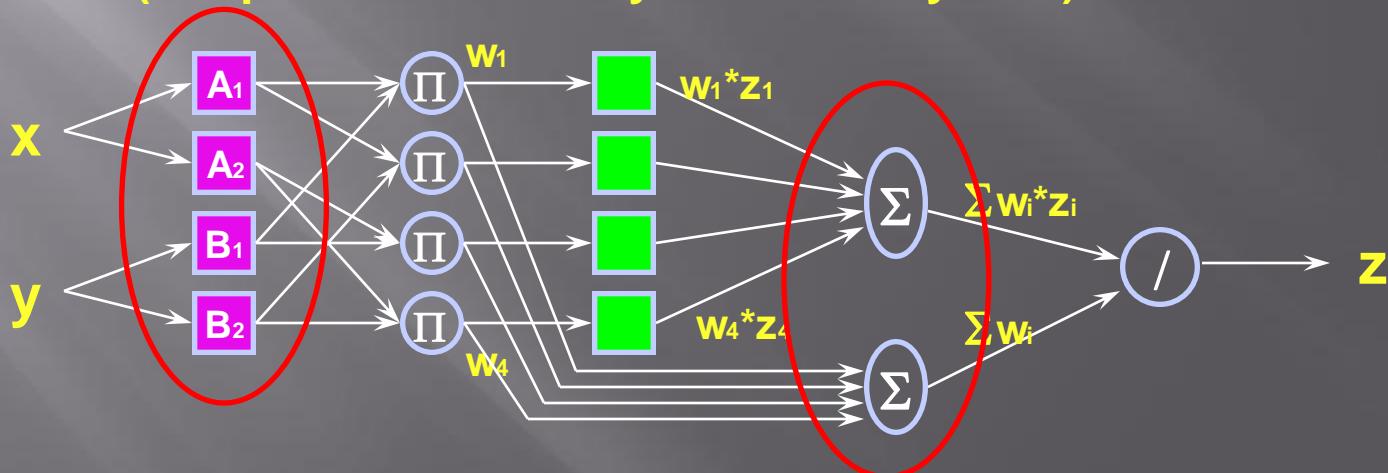
Arsitektur jaringan ANFIS yang berhubungan dengan Sugeno model di atas digambarkan sbb [TET01]:



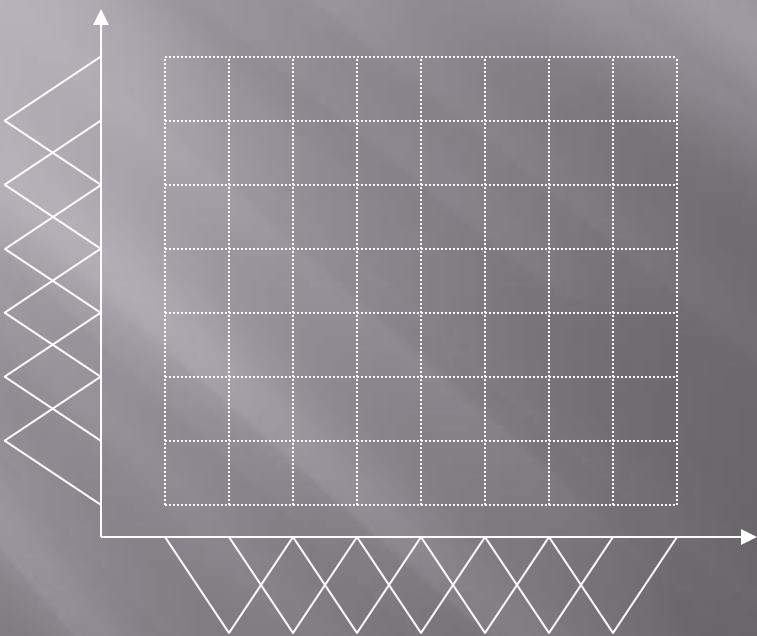
- Input space partitioning



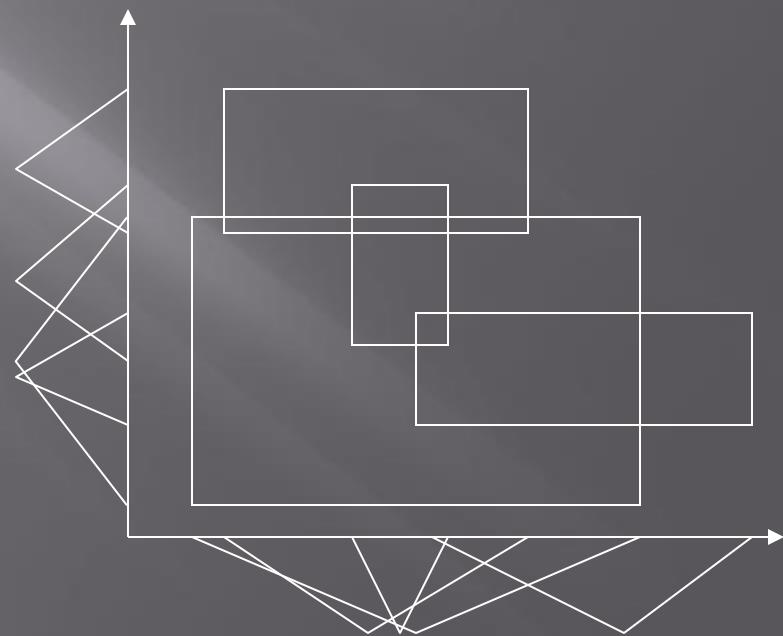
- ANFIS (Adaptive Neuro-Fuzzy Inference System)

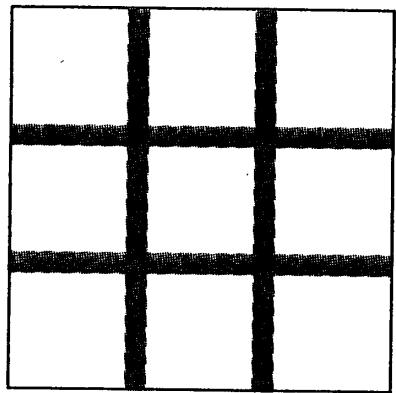


Regular grid

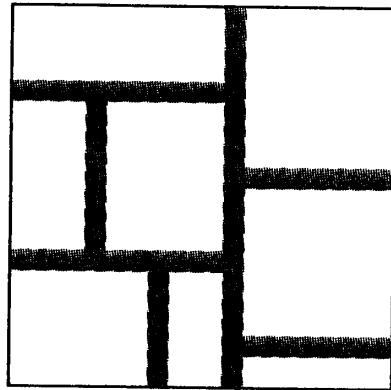


Independent functions

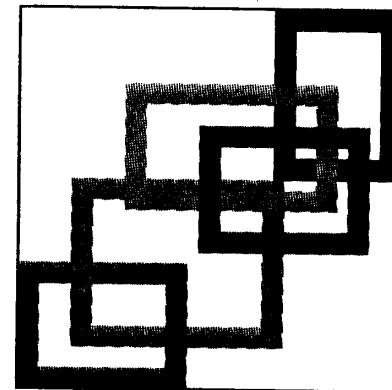




(a)



(b)



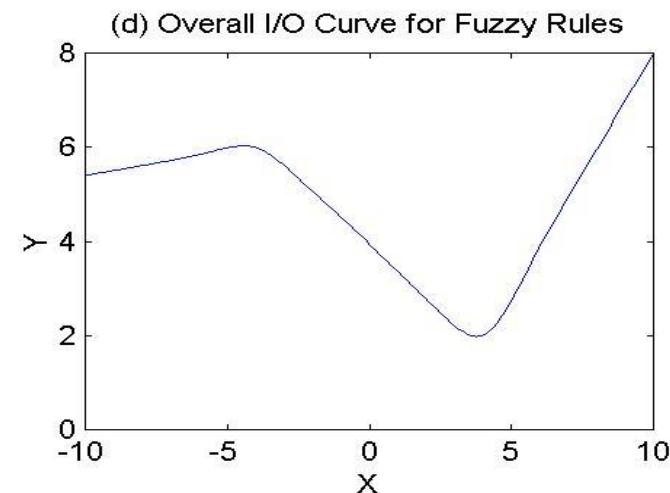
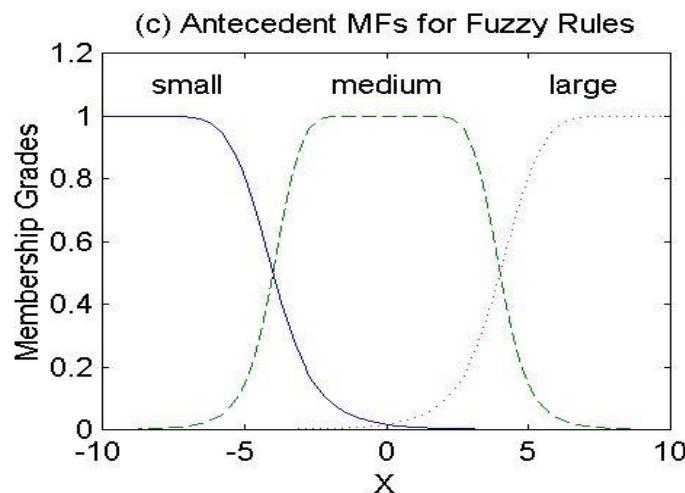
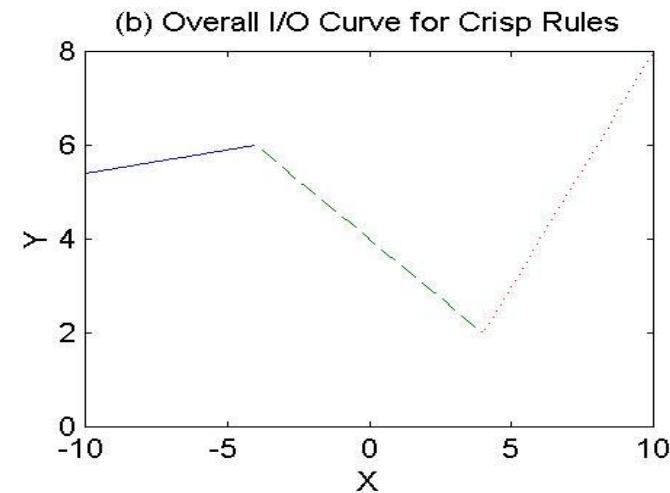
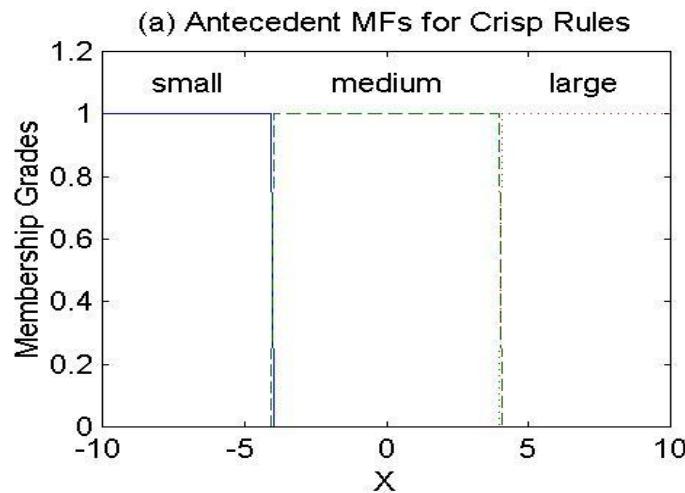
(c)

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:

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

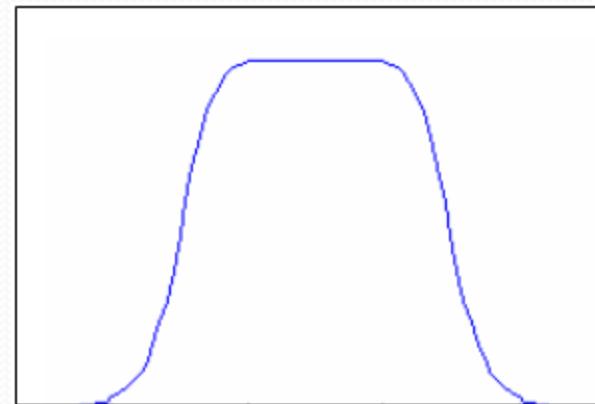
$$O_{1,i} = \mu_{B_i}(y), \quad i = 3, 4$$

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

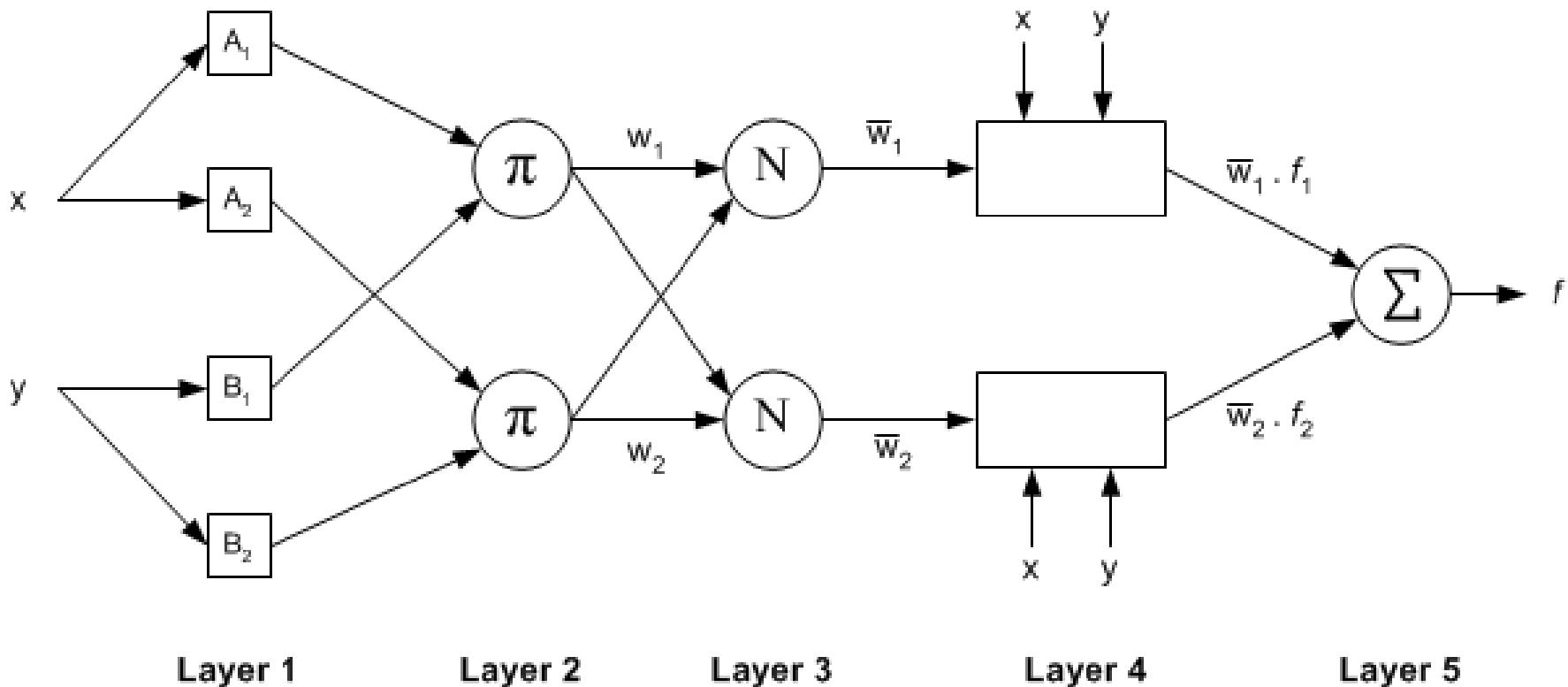
ANFIS (Adaptive Network-based Fuzzy Inference System)

Misalkan, pada layer 1, kita menggunakan fungsi *Bell* sebagai fungsi keanggotaan μ 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)

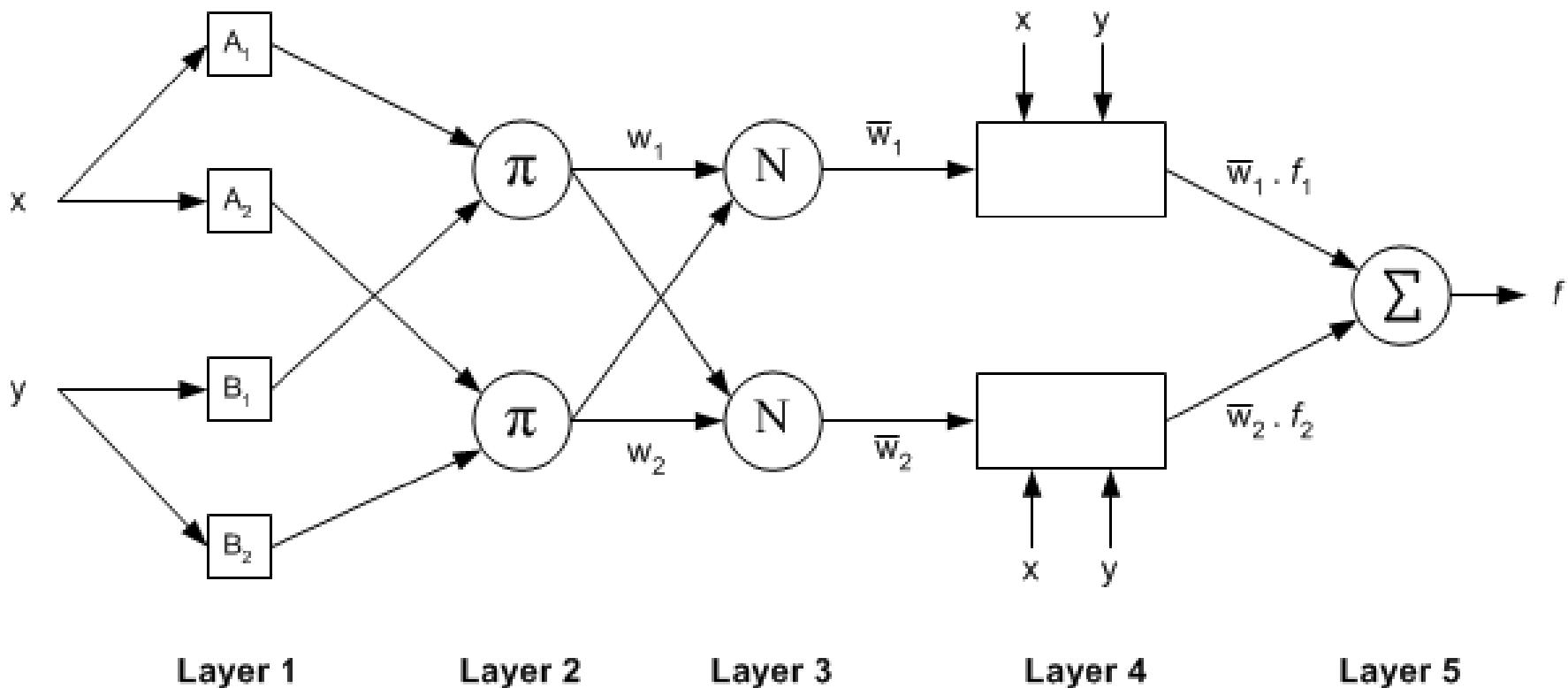
- Layer 2: Dinotasikan π . Setiap node pada layer ini berfungsi untuk menghitung *firing strength* dari setiap *rule* sebagai *product* dari semua input yang masuk atau **operator t-norm** (*triangular norm*):

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

Sehingga

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

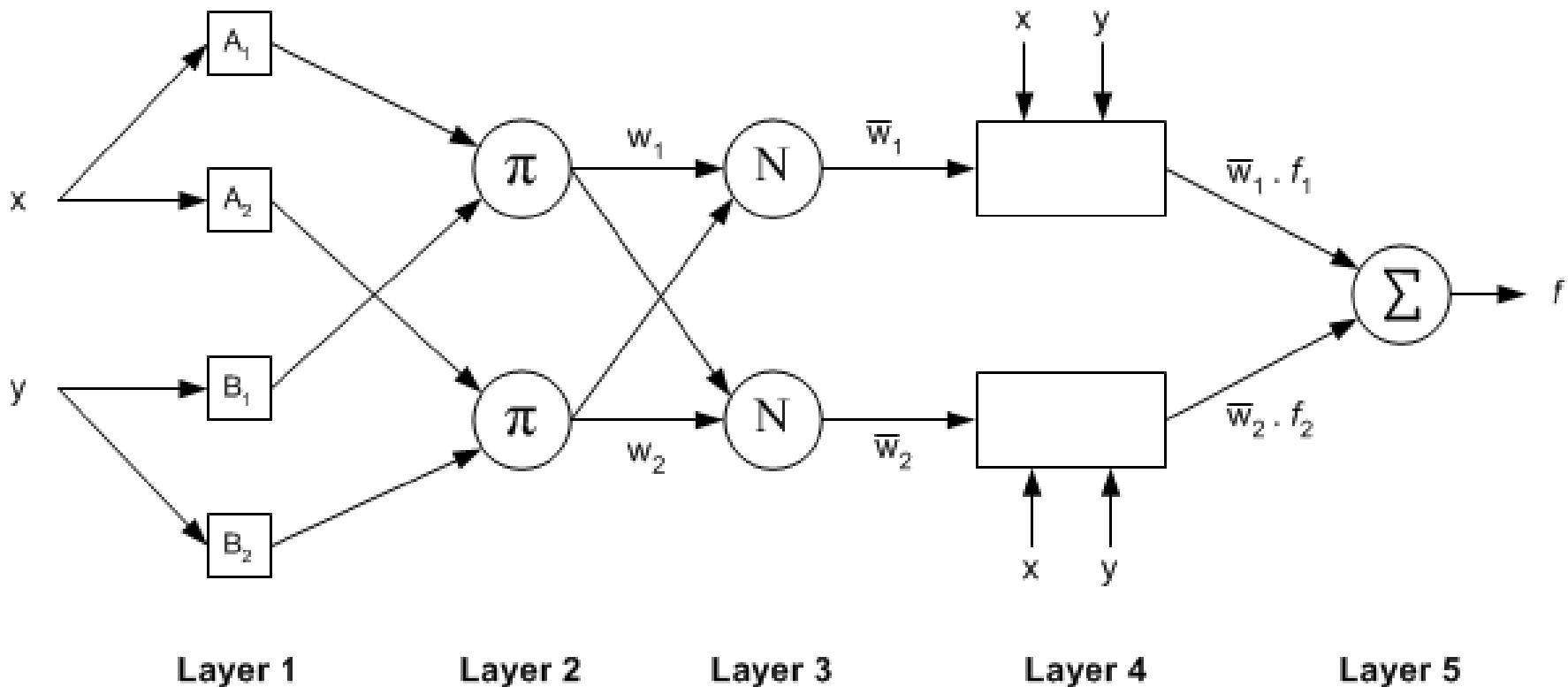
$$w_2 = \mu_{A_2}(x) \text{ AND } \mu_{B_2}(y)$$



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:

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



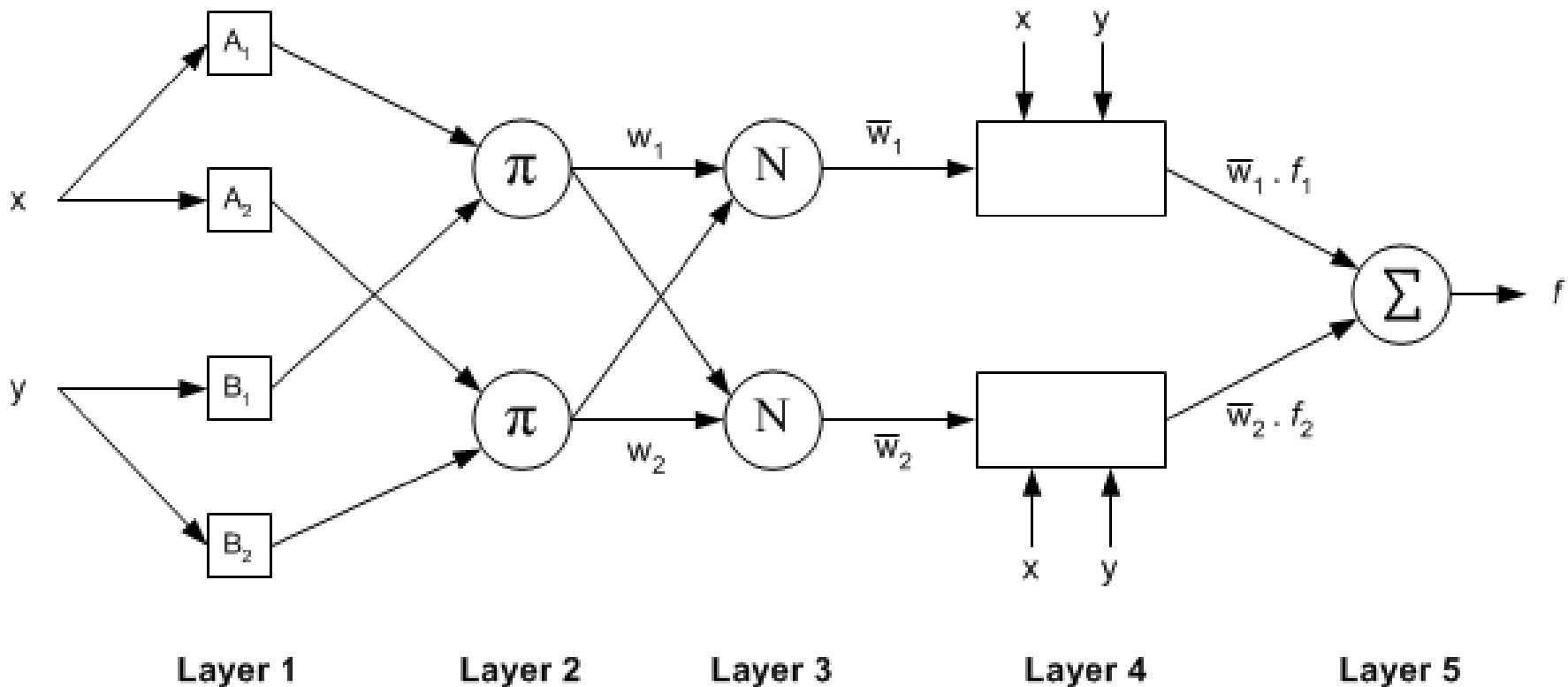
ANFIS (Adaptive Network-based Fuzzy Inference System)

- **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 Σ 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

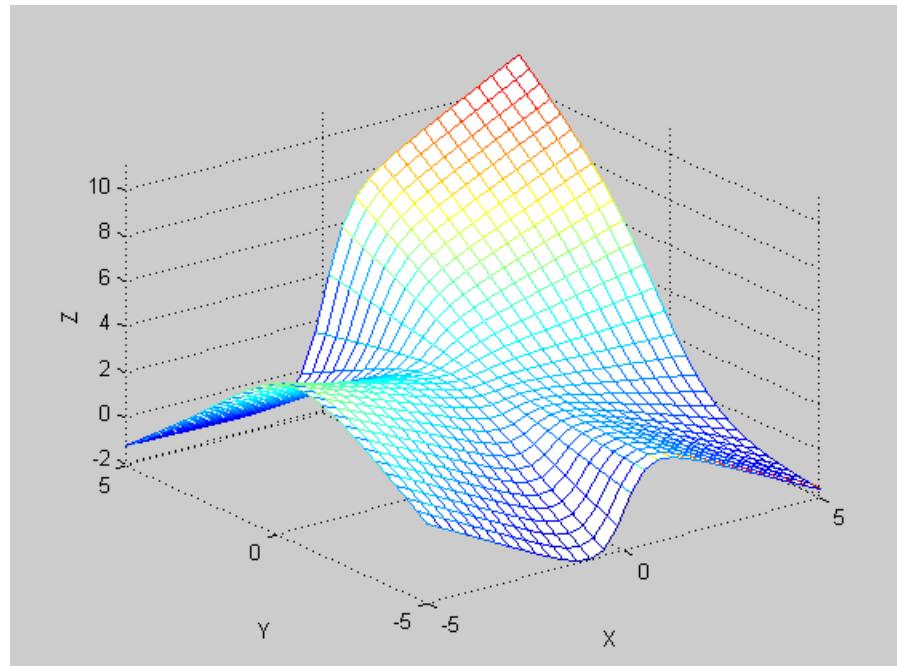
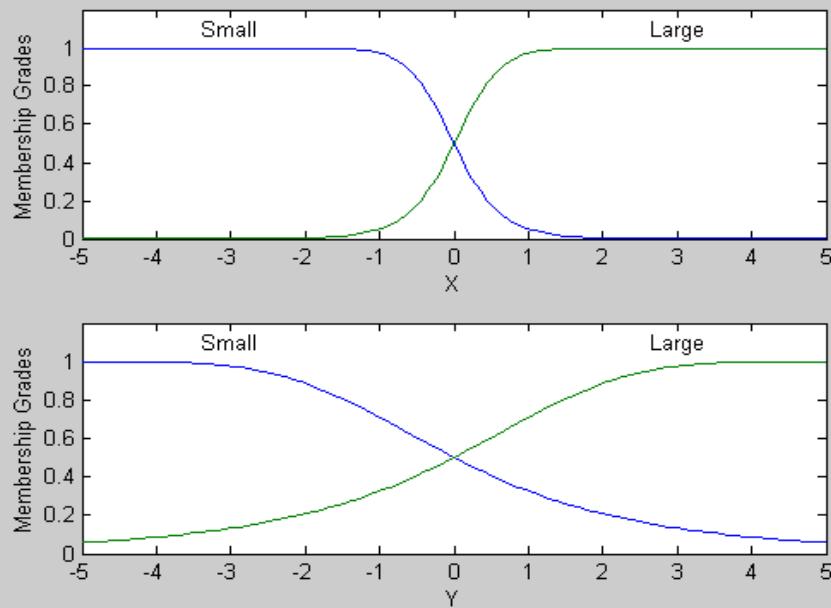
- Algoritma *learning* pada ANFIS adalah *hybrid supervised method* yang berbasis pada dua metode: *least-squares* dan *gradient descent*.
- Pada tahap maju, sinyal-sinyal merambat maju sampai layer 4 dan *consequent parameters* di-update menggunakan metode *least-square*
- Pada tahap mundur, sinyal-sinyal 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 $z = -x+3$

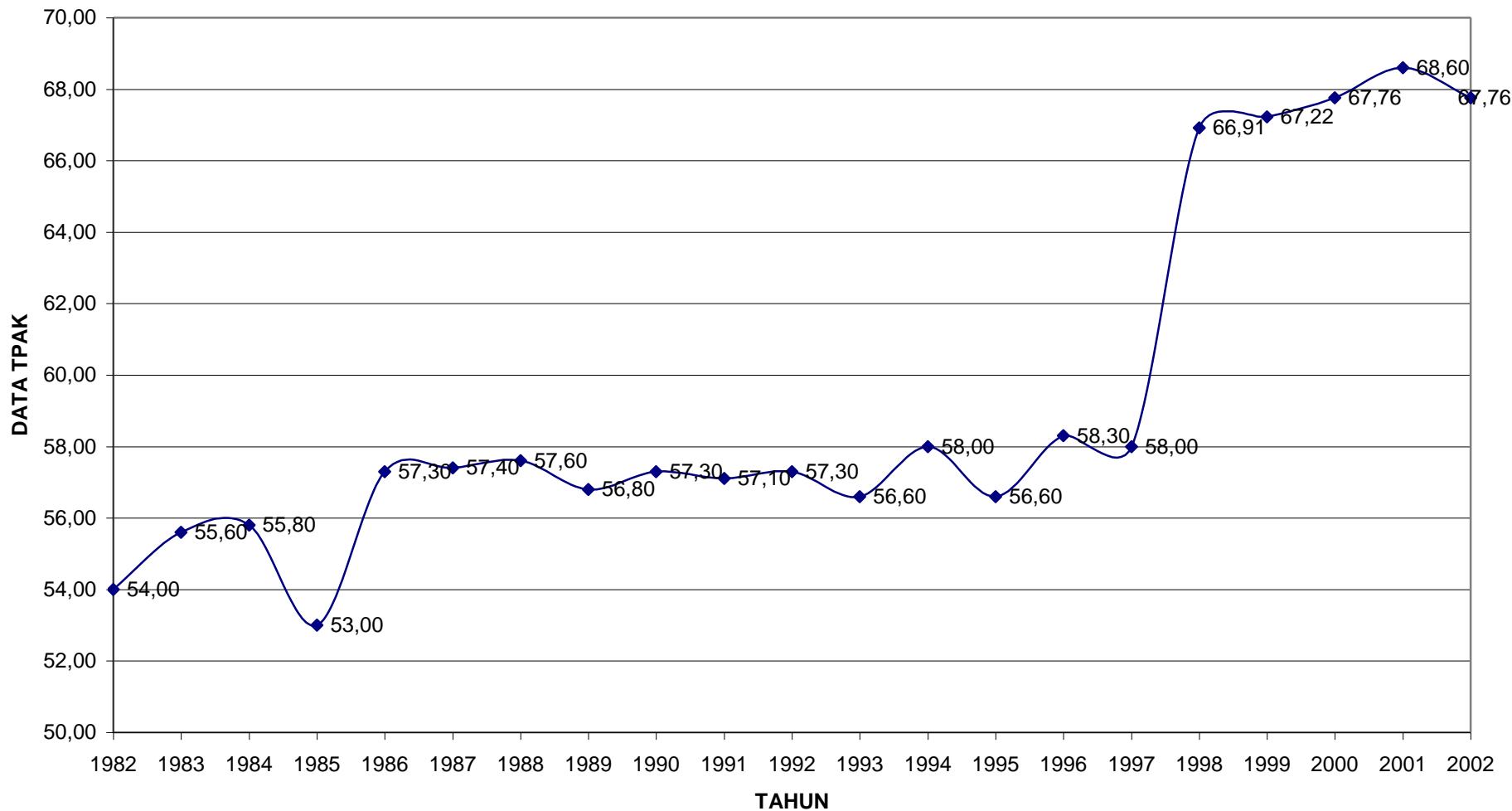
If X is large and Y is large then $z = x+y+2$

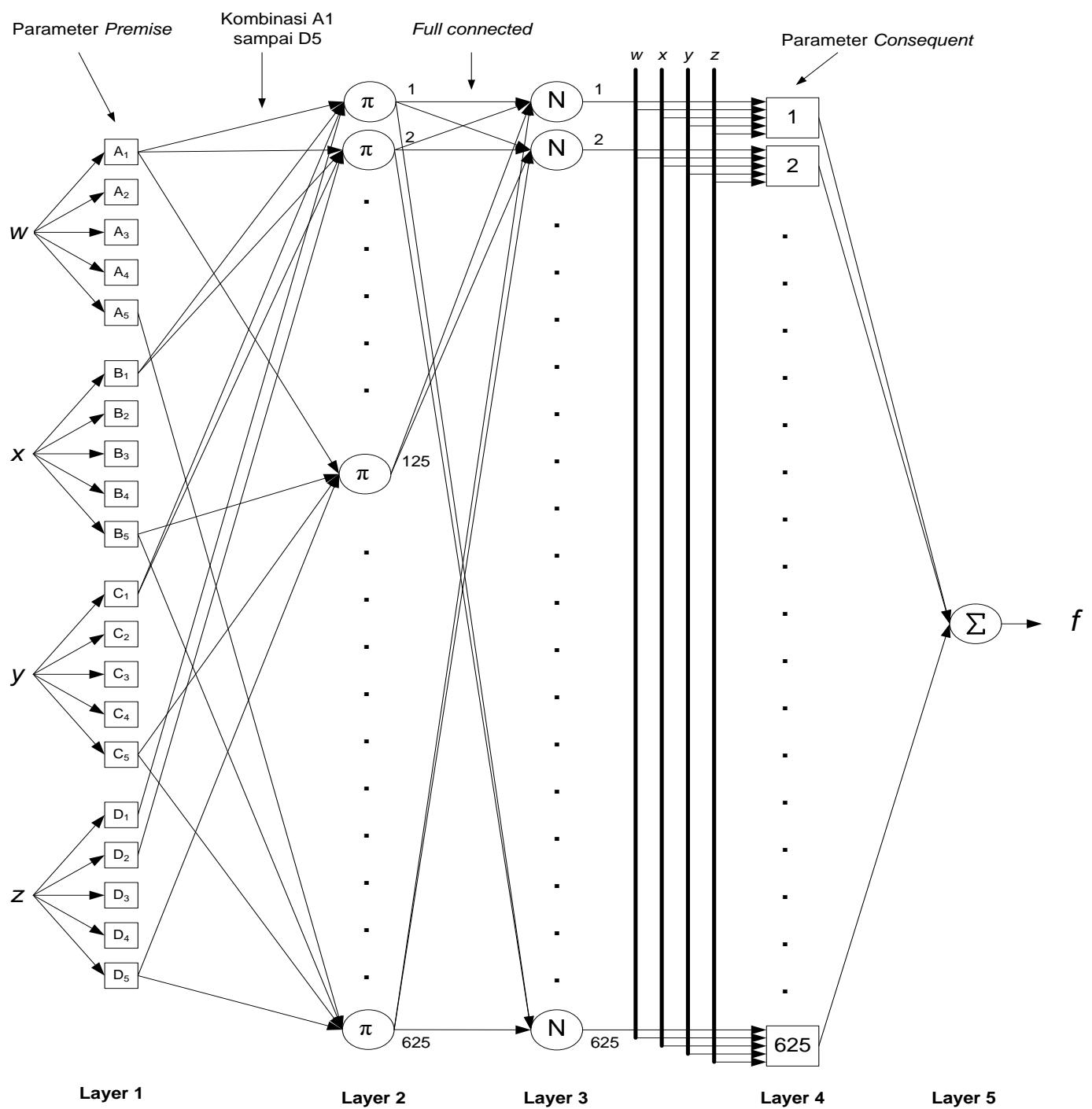


Prediksi Tingkat Partisipasi Angkatan Kerja

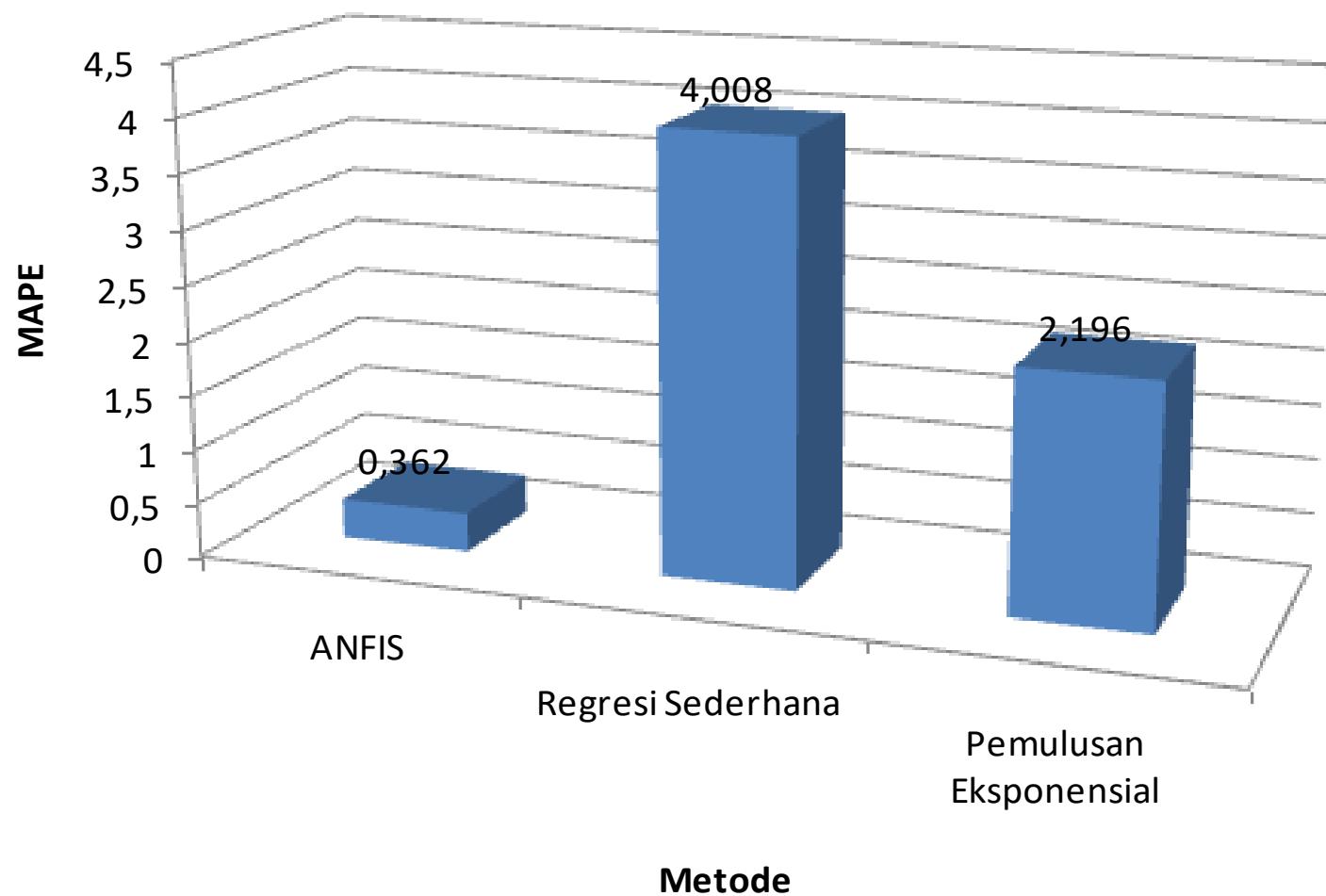
- Data dari Badan Pusat Statistik
- Data time series per tahun: 1982 – 2002 (21 sampel)
- Bagaimana penggunaan ANFIS?

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.

Evolving ANN

Dr. Suyanto, S.T., M.Sc.
HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

Perbandingan EAs dan ANN

Kriteria	EAs	ANN
Sangat baik untuk klasifikasi atau pengenalan pola?	Tidak	Ya
Sangat baik untuk optimasi, khususnya permasalahan kombinatorial?	Ya	Tidak
Bisa <i>learning</i> ?	Ya	Ya

Kekurangan ANN

- Arsitektur optimal?
 - Berapa jumlah layer yang optimal?
 - Berapa jumlah neuron pada setiap *layer*-nya?
 - Fungsi aktivasi?
- Berapa *learning rate* yang baik?
- Kapan menghentikan *learning*?

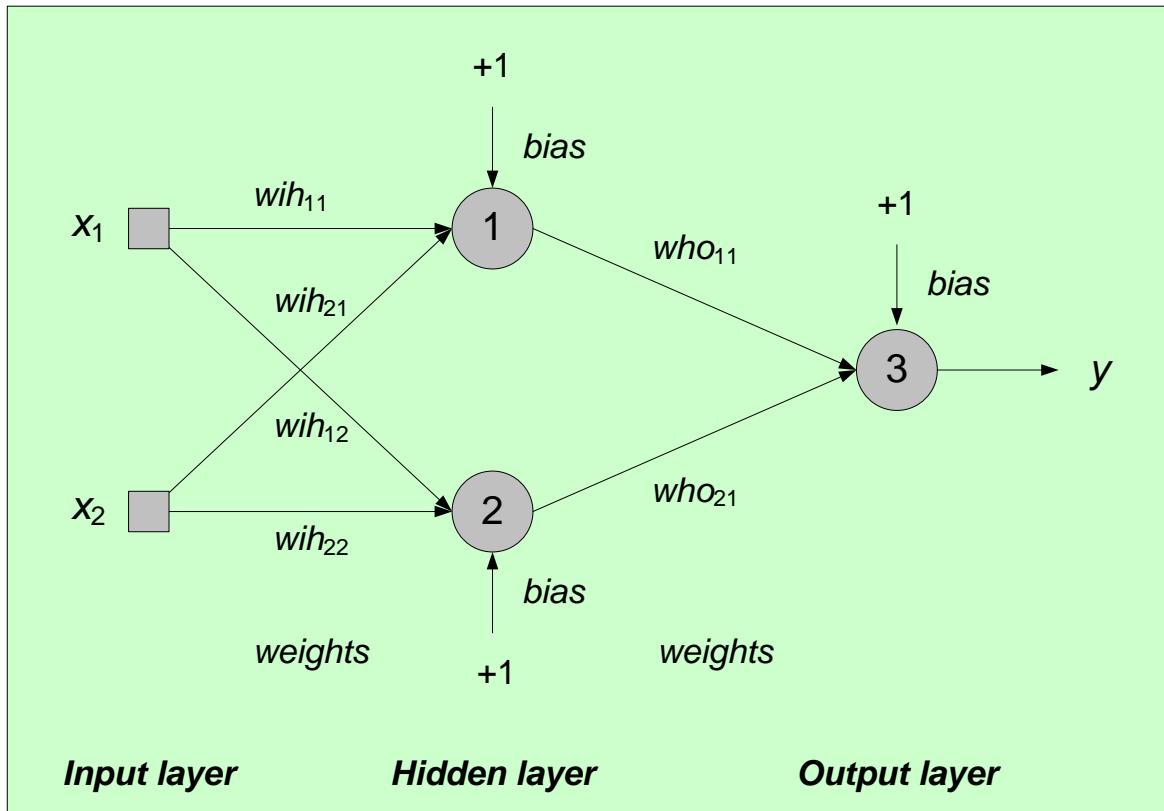
EAs untuk melatih ANN

- Banyak algoritma learning untuk ANN:
 - Back Propagation
 - Hebbian Learning
 - Competitive Learning, dsb.
- EAs bisa digunakan learning?
- Bisa.
- Bagaimana representasi kromosomnya?
- Bagaimana fungsi fitnessnya?

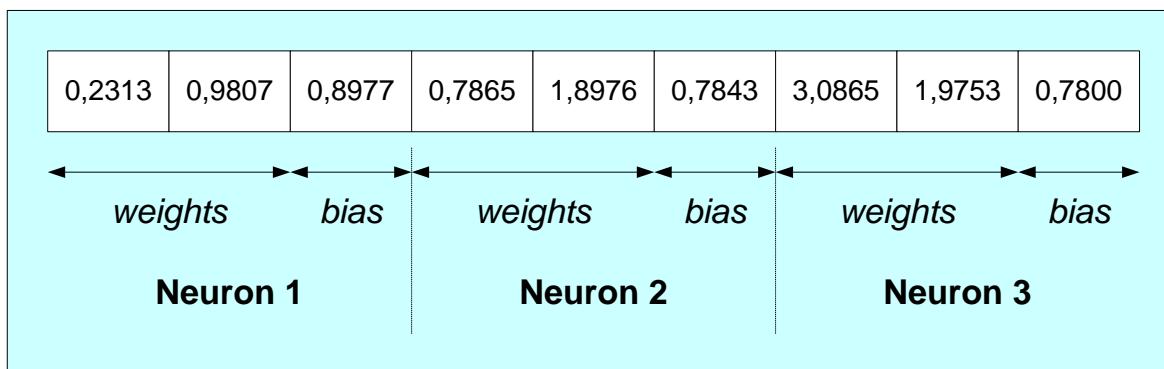
Representasi Kromosom

- Biner
- Integer
- Real

Multi Layer Perceptron (MLP):



Kromosom pada GA:

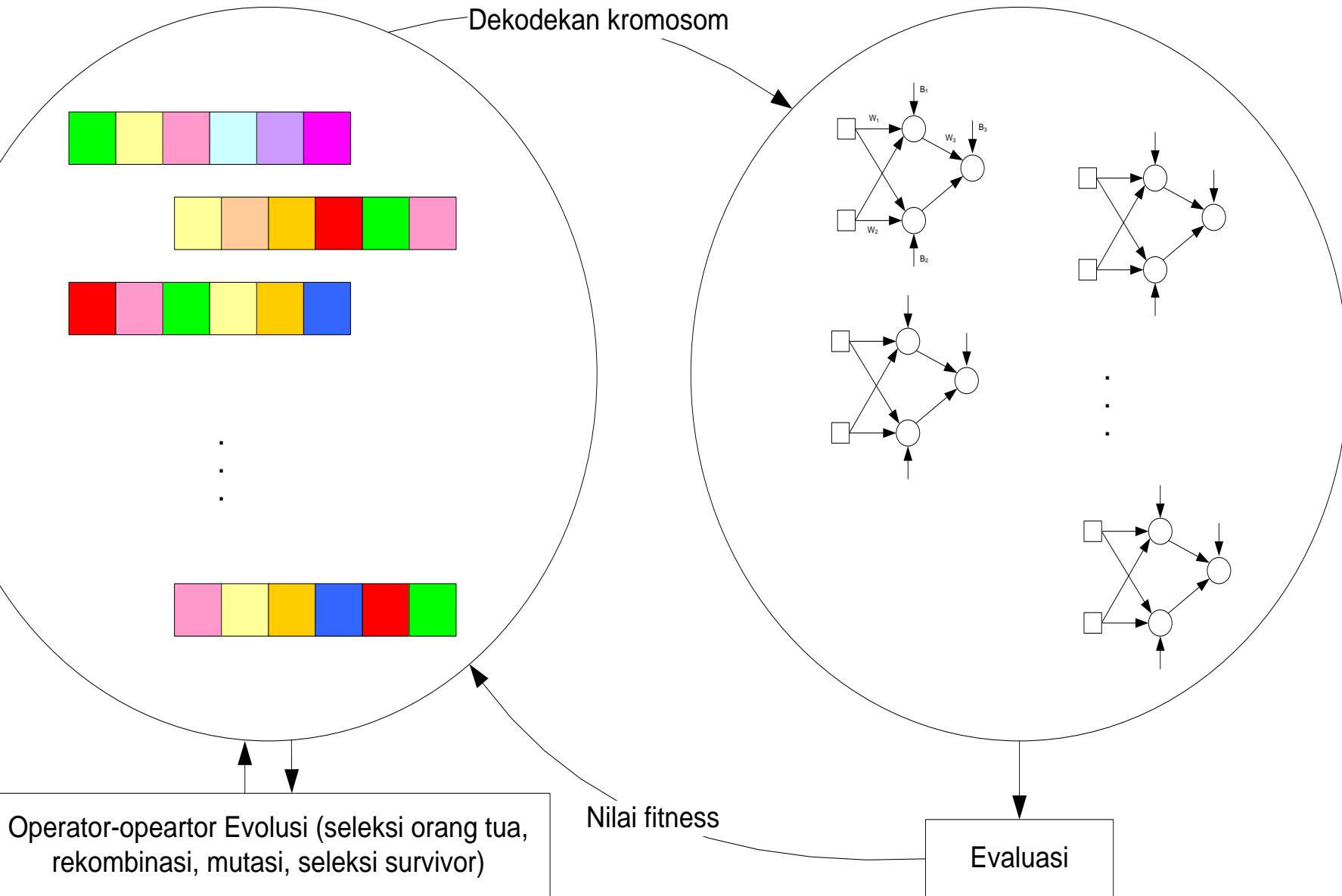


Fungsi Fitness

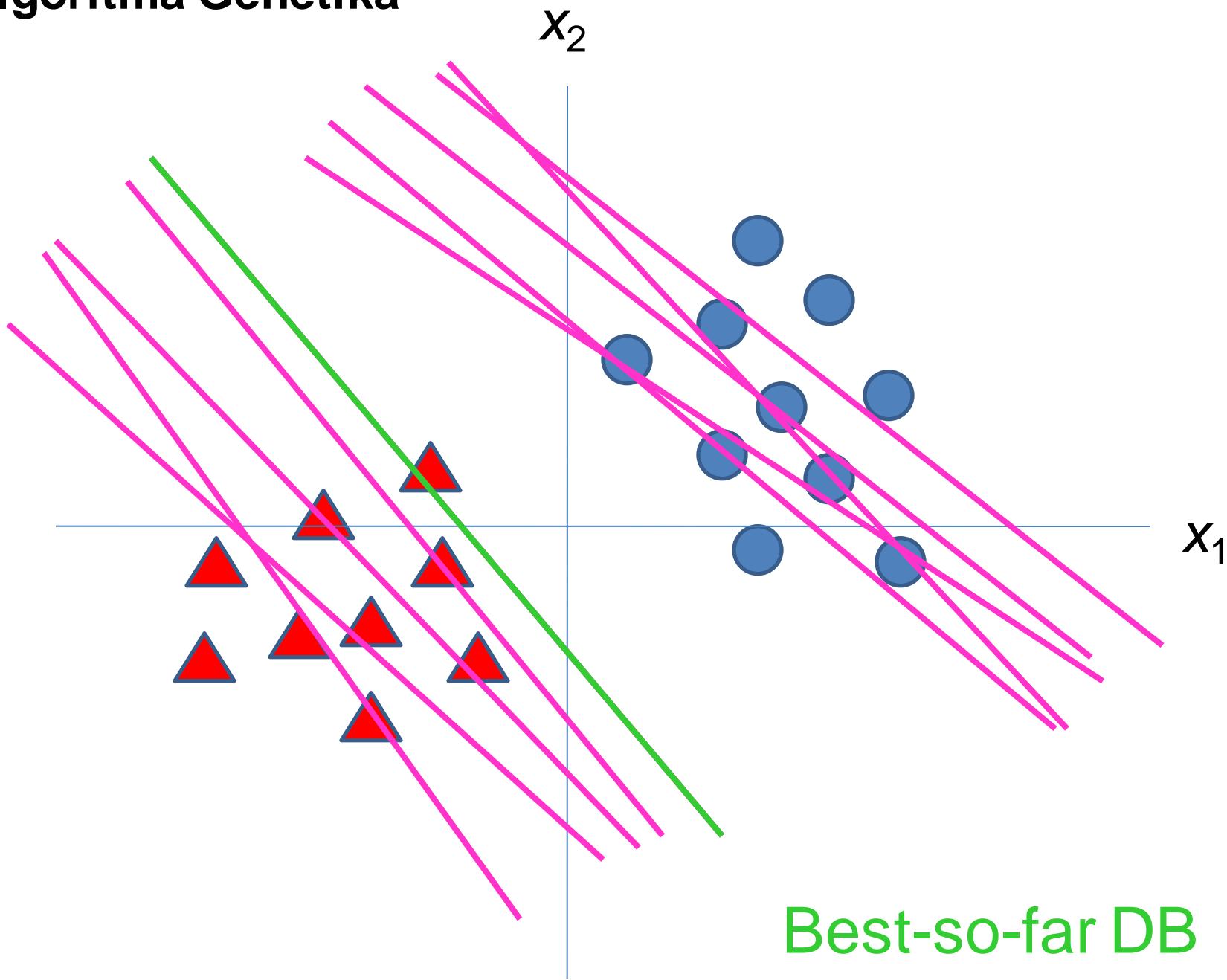
- *Back propagation*: maju dan mundur
- GA hanya: maju saja.
- *Fitness* dihitung berdasarkan rata-rata *error* (MSE) antara target dan keluaran untuk semua data latih
- MSE kecil → *fitness* tinggi

$$f = \frac{1}{MSE}$$

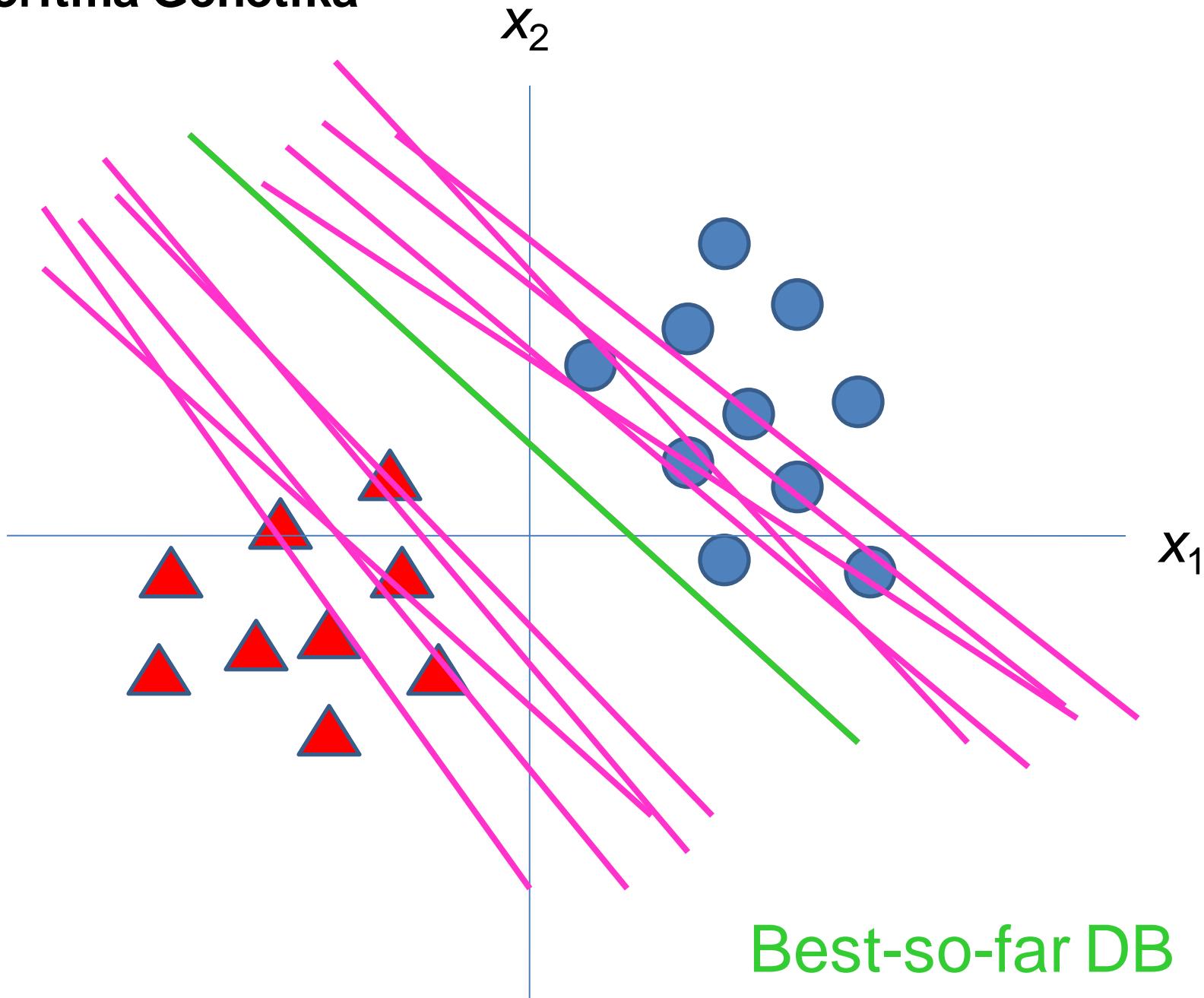
$$f = \frac{1}{MSE + a}$$



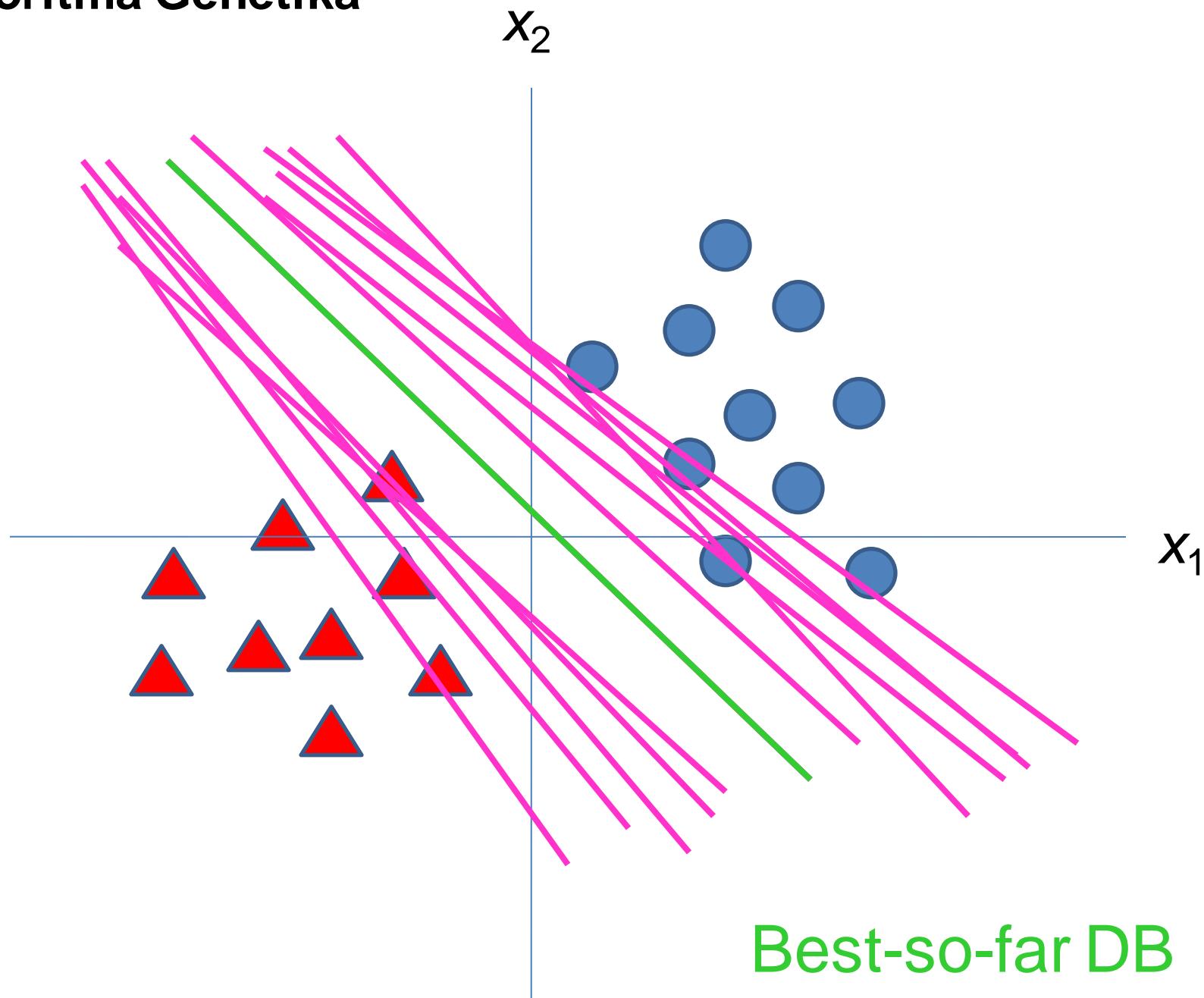
Algoritma Genetika



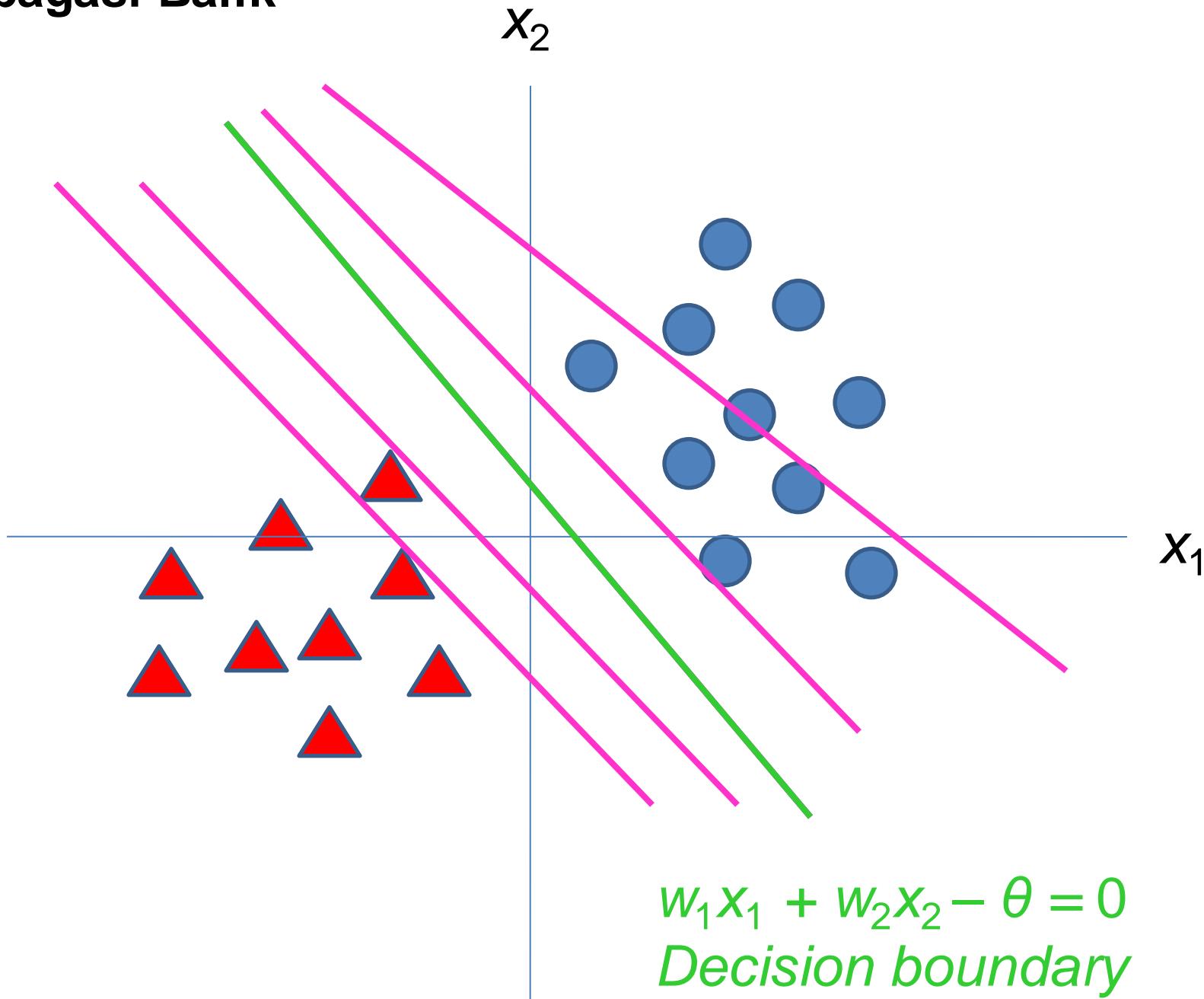
Algoritma Genetika



Algoritma Genetika

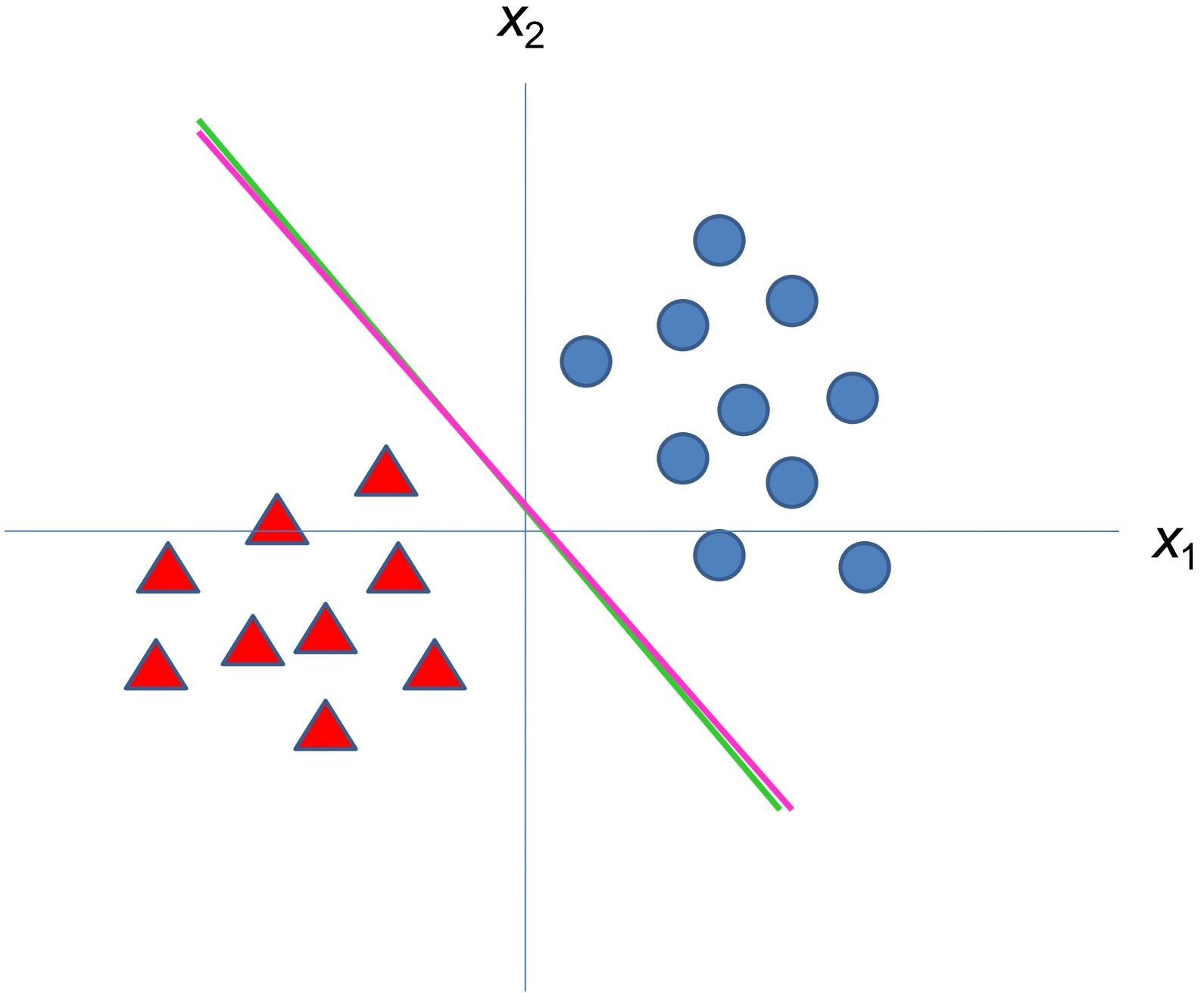


Propagasi Balik



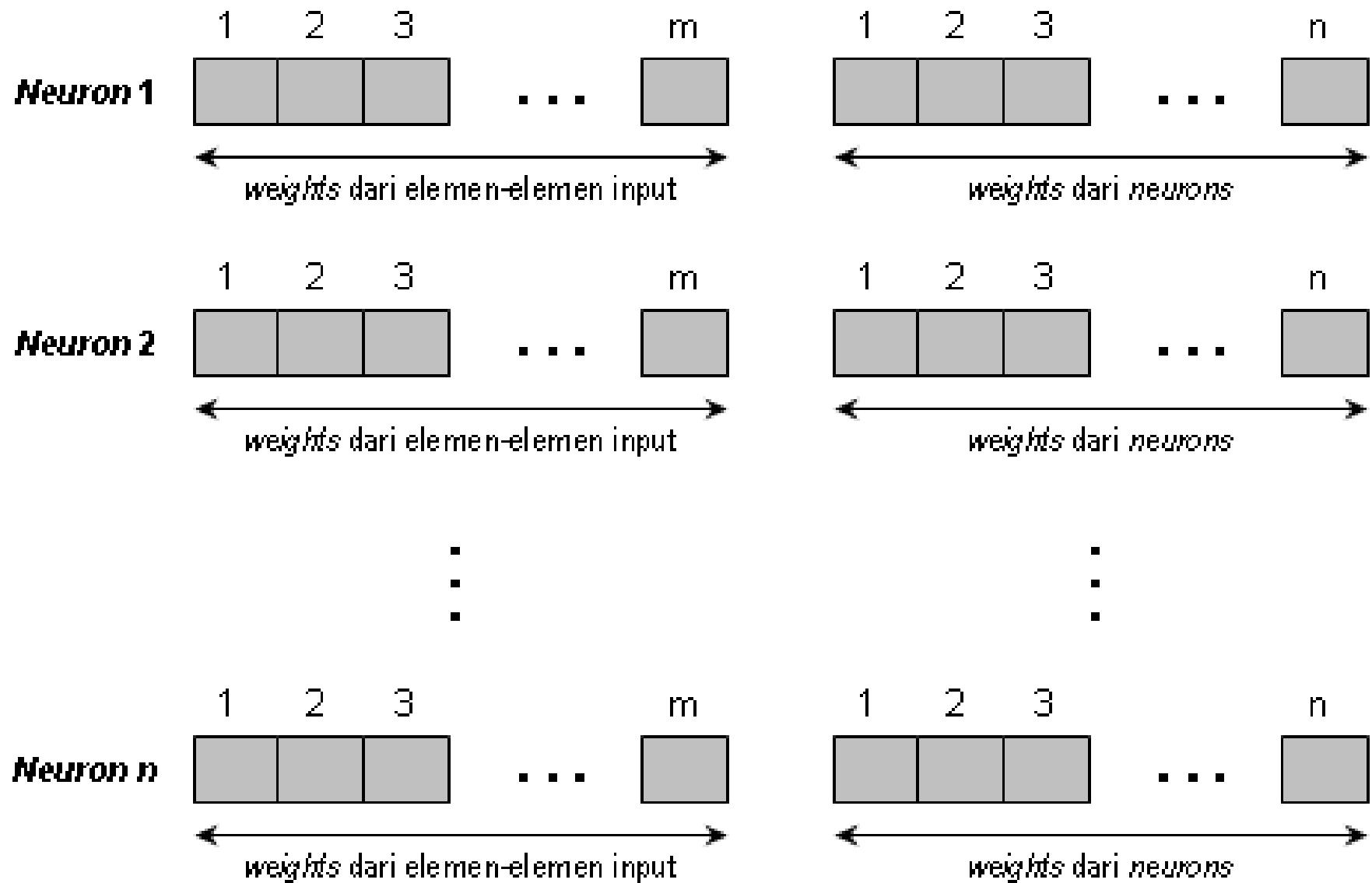
Diskusi

- AG lebih baik dibandingkan PB?
- PB
 - Tipe data real
- AG
 - Kromosom biner
 - Presisi bisa diatur

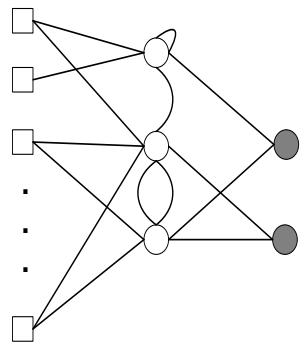


EAs untuk optimasi struktur ANN

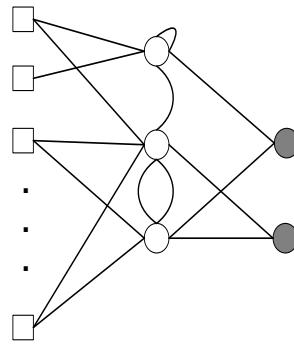
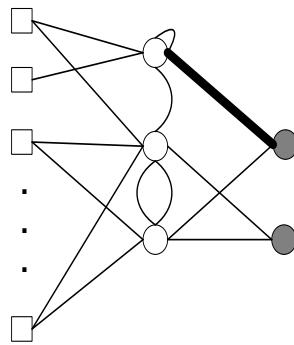
- Banyak struktur ANN:
 - Single Layer Perceptron
 - Multi Layer Perceptron
 - Recurrent Network, dsb.
- EAs untuk optimasi struktur ANN?
- Bagaimana representasi kromosomnya?
- Bagaimana fungsi fitnessnya?



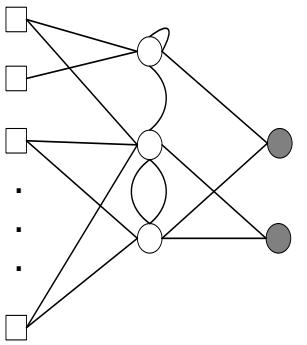
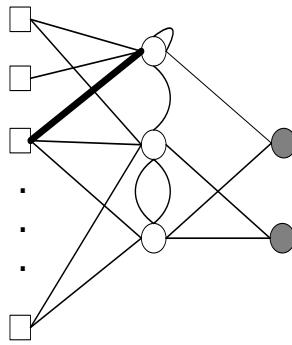
Kromosom Real



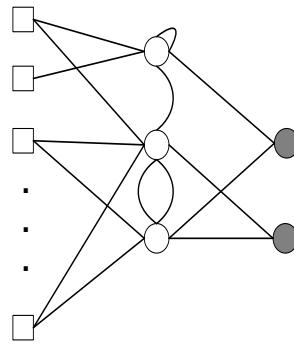
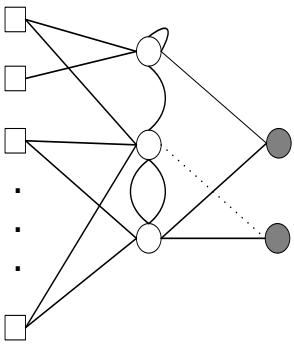
M1, M2



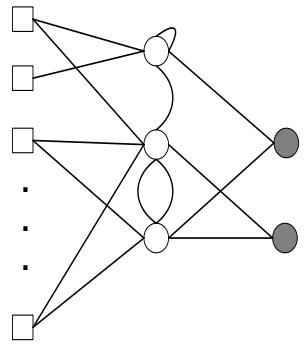
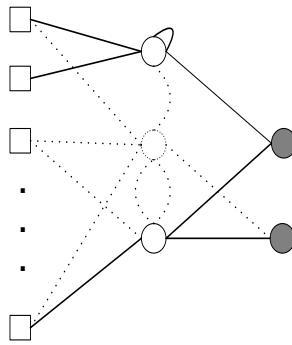
M3



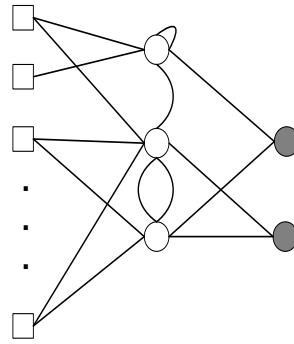
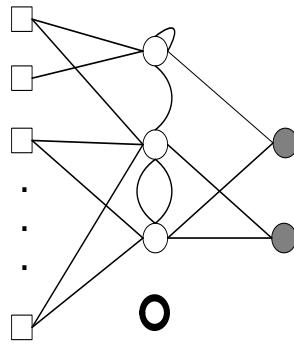
M4



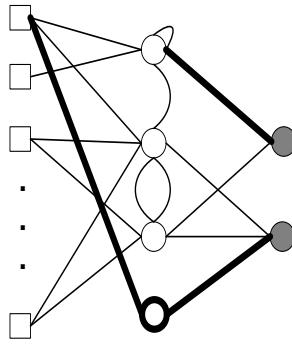
M5



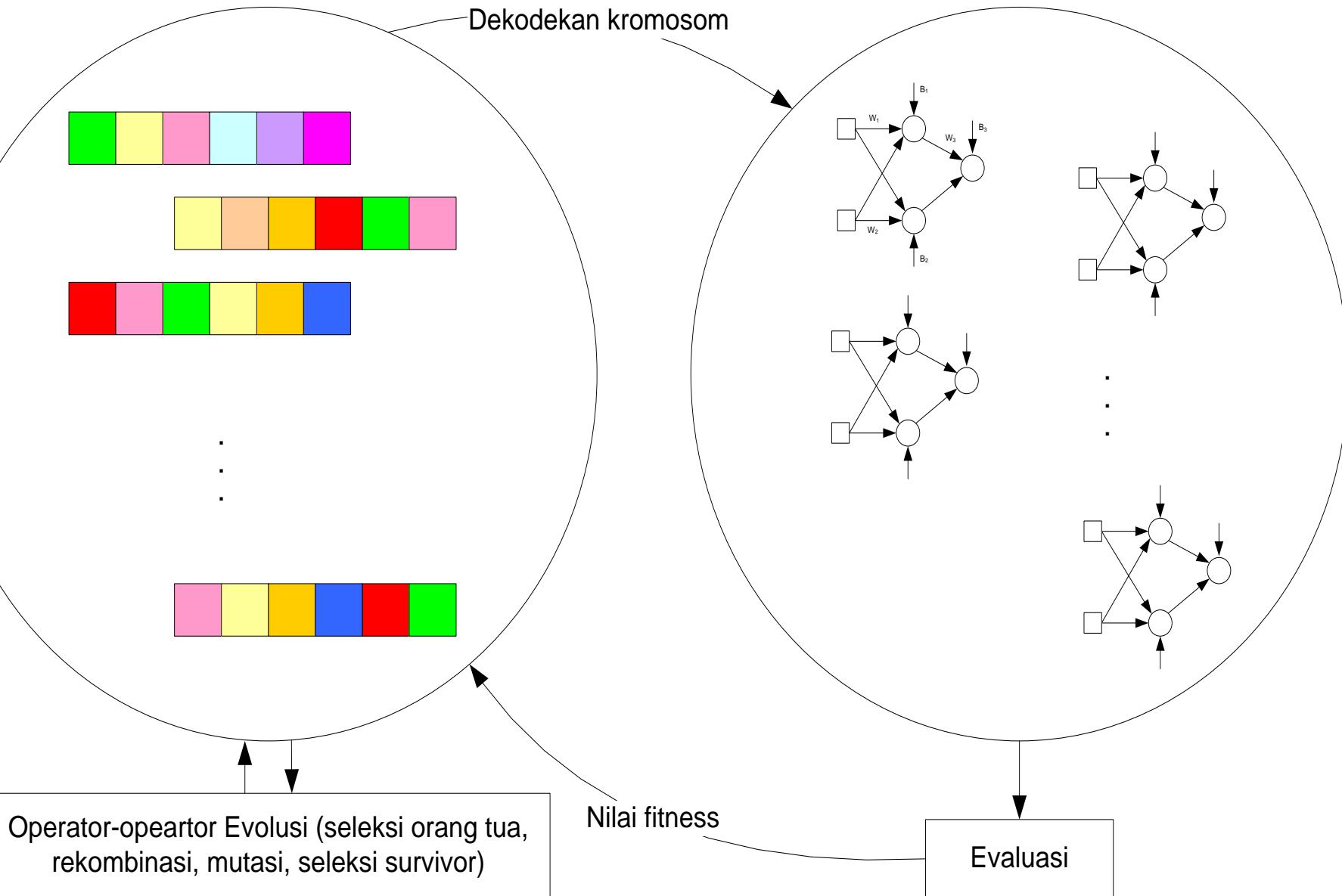
M6

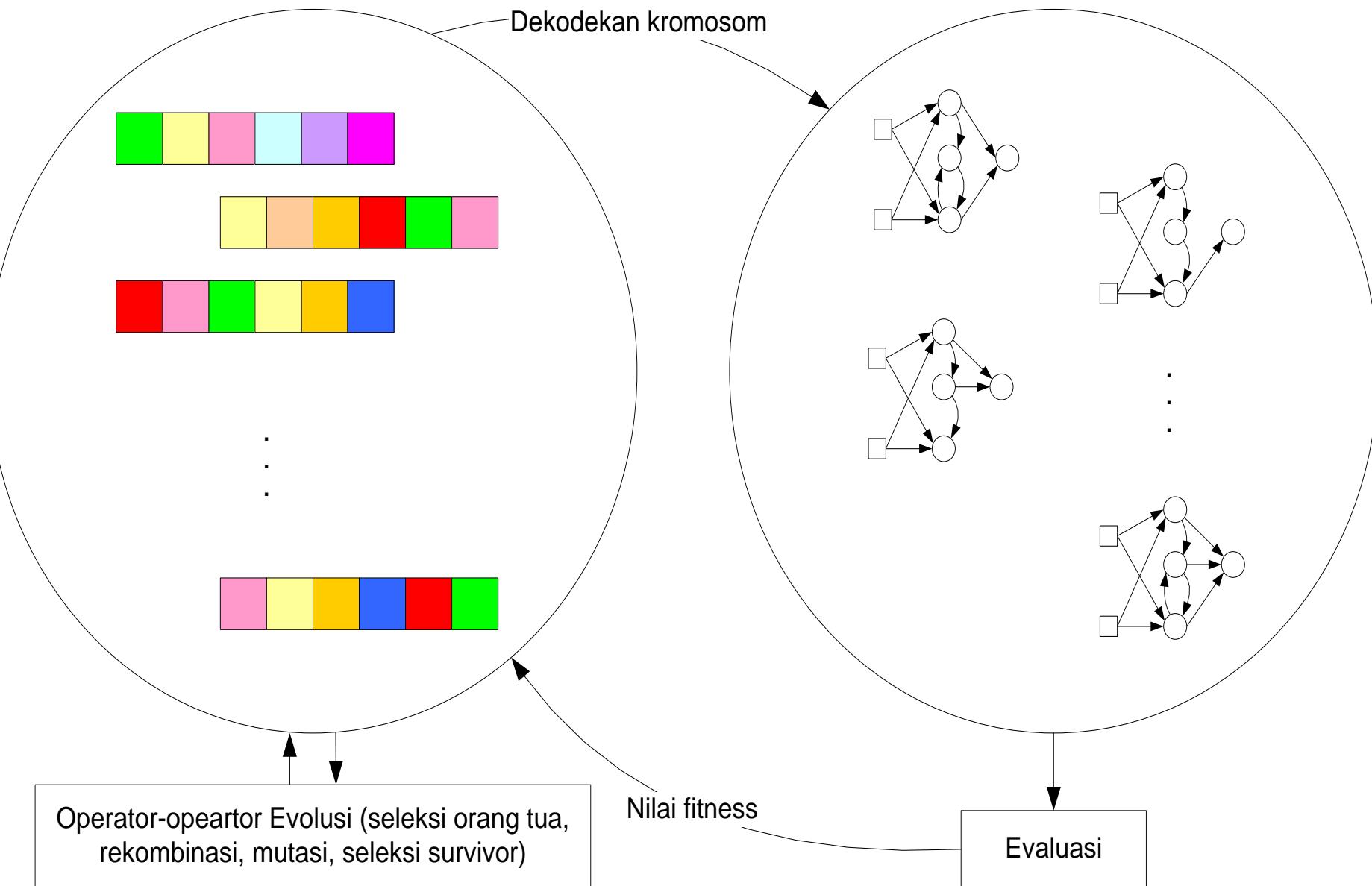


M7



Operator Mutasi





Fungsi Fitness

- Setiap kromosom dilatih menggunakan BP (atau EAs) dengan semua data latih untuk sejumlah *epoch* tertentu, misalnya 100 *epoch*.
- MSE kecil → *fitness* tinggi

$$f = \frac{1}{MSE}$$

Fungsi Fitness

- Setiap kromosom dilatih menggunakan BP (atau EAs) dengan semua data latih sampai menghasilkan MSE tertentu, misalnya 0,001.
- Jumlah *epoch* sedikit → *fitness* tinggi

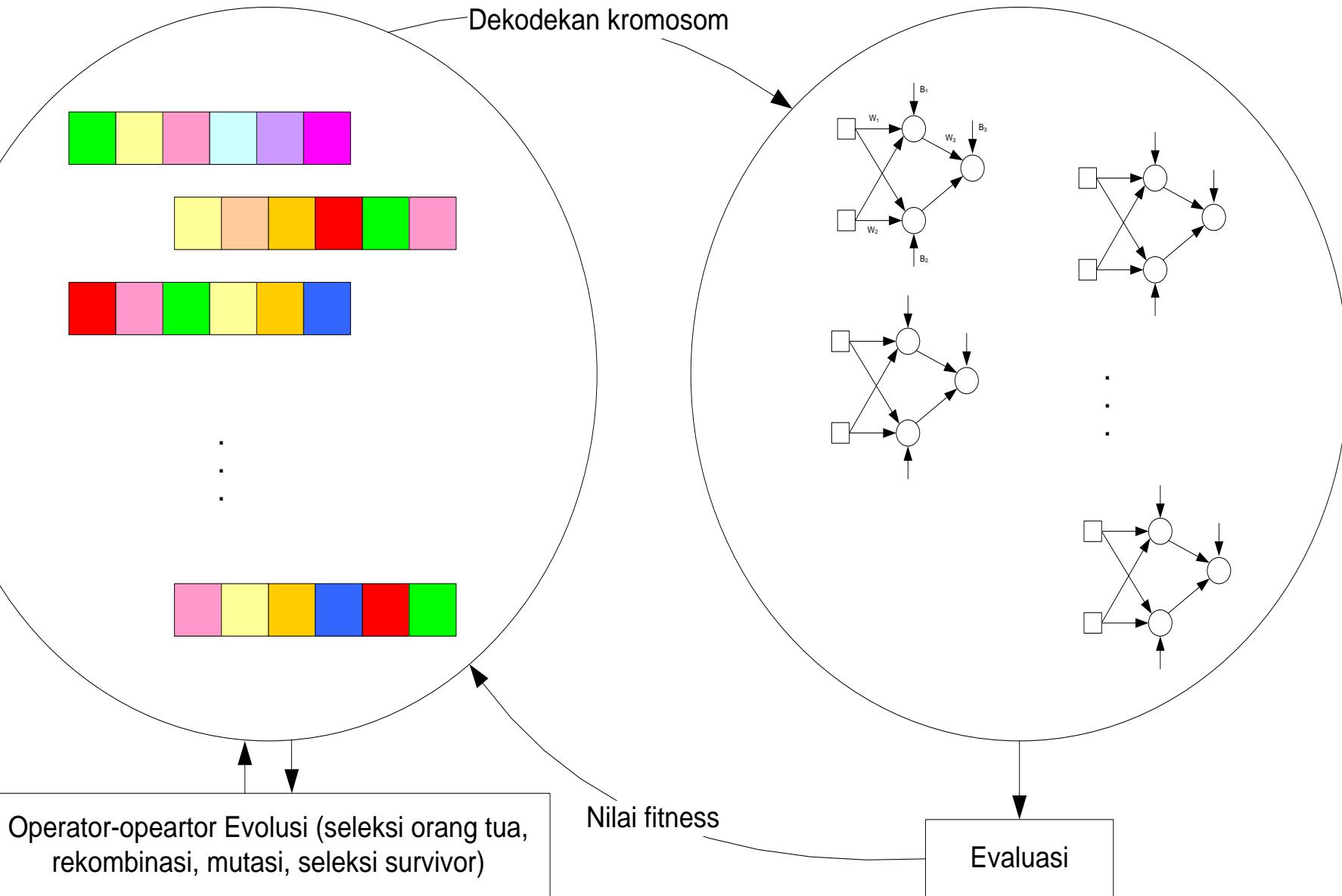
$$f = \frac{1}{epoch}$$

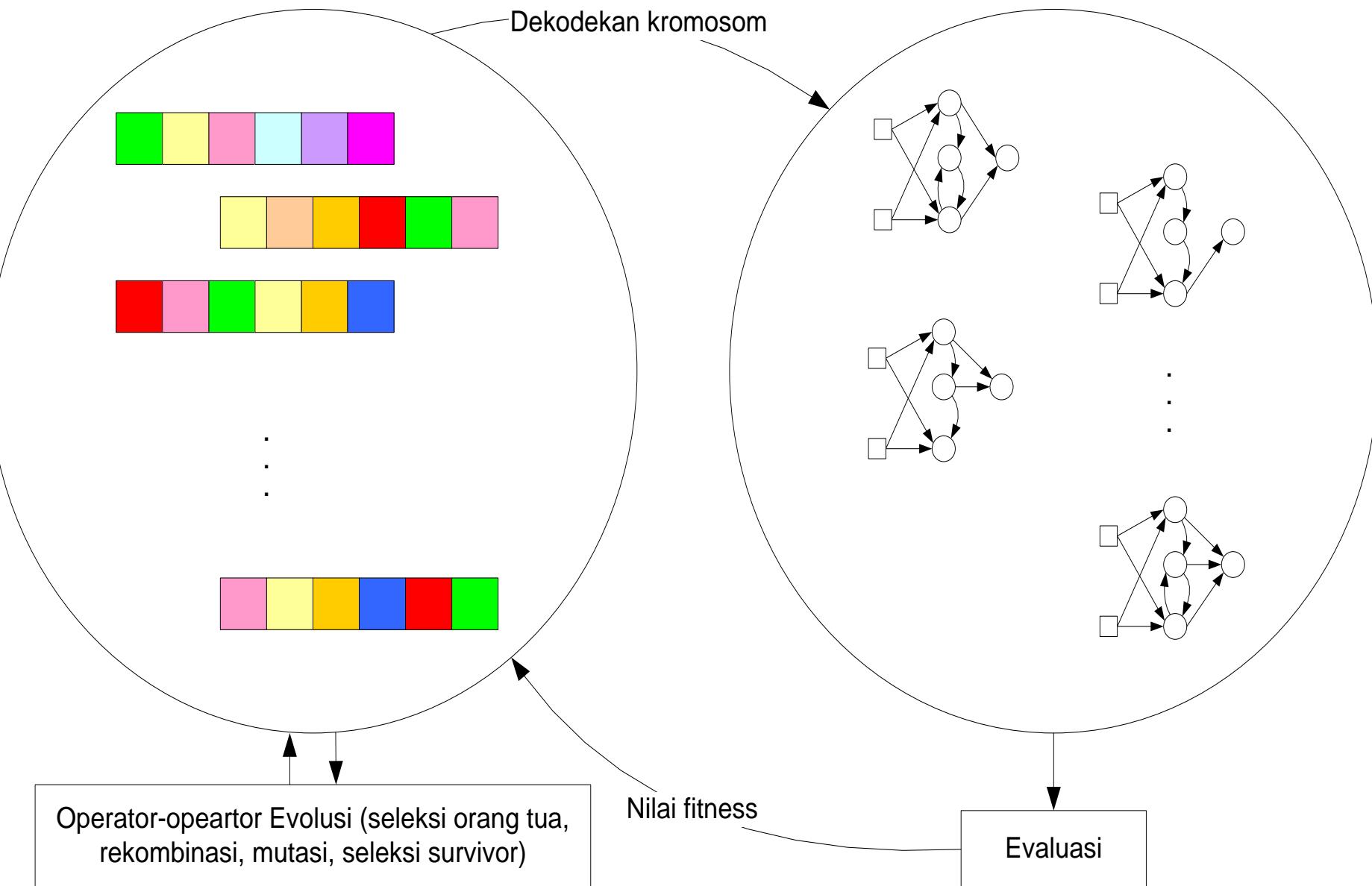
Grammatical Encoding

- Otak manusia merupakan suatu komputer sangat kompleks yang terdiri dari sekitar 10^{11} elemen komputasi (*neurons*).
- Terdapat sekitar 10^{14} sampai 10^{15} koneksi antar *neurons*, atau sekitar 1000 sampai 10000 koneksi per *neuron*.
- Jika setiap koneksi dikodekan ke dalam kromosom, maka informasi yang mengisi kromosom akan sekitar **10⁵ GB**, dimana bobot-bobot sinaptik dikodekan menggunakan hanya 1 *byte*.
- Tetapi, pada kenyataanya ukuran *genome* manusia hanya sekitar **3 GB**.
- Oleh karena itu para peneliti percaya bahwa pengkodean otak manusia tidak menggunakan pengkodean langsung, melainkan pengkodean **prosedur** dimana otak dibentuk.

Grammatical Encoding

- Pada skema ini, kromosom dipandang sebagai kalimat yang diekspresikan menggunakan *grammar* (tata bahasa).
- Ketika sebuah kalimat dibaca (kromosom didekodekan), maka individu dibangkitkan menggunakan *grammar* tsb.
- Contoh: skema Kitano yang digunakan untuk mengkodekan ANN yang berisi maksimum 8 *neurons*





Skema Kitano

S	ACBA	Aadfb	Bbefd	Dfanp	Bahjm	Ckhgf	...
---	------	-------	-------	-------	-------	-------	-----

$$S \rightarrow \begin{pmatrix} A & C \\ B & A \end{pmatrix}$$

$$A \rightarrow \begin{pmatrix} a & d \\ f & b \end{pmatrix} \quad B \rightarrow \begin{pmatrix} b & e \\ f & d \end{pmatrix}$$

$$a \rightarrow \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, b \rightarrow \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, c \rightarrow \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \dots p \rightarrow \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

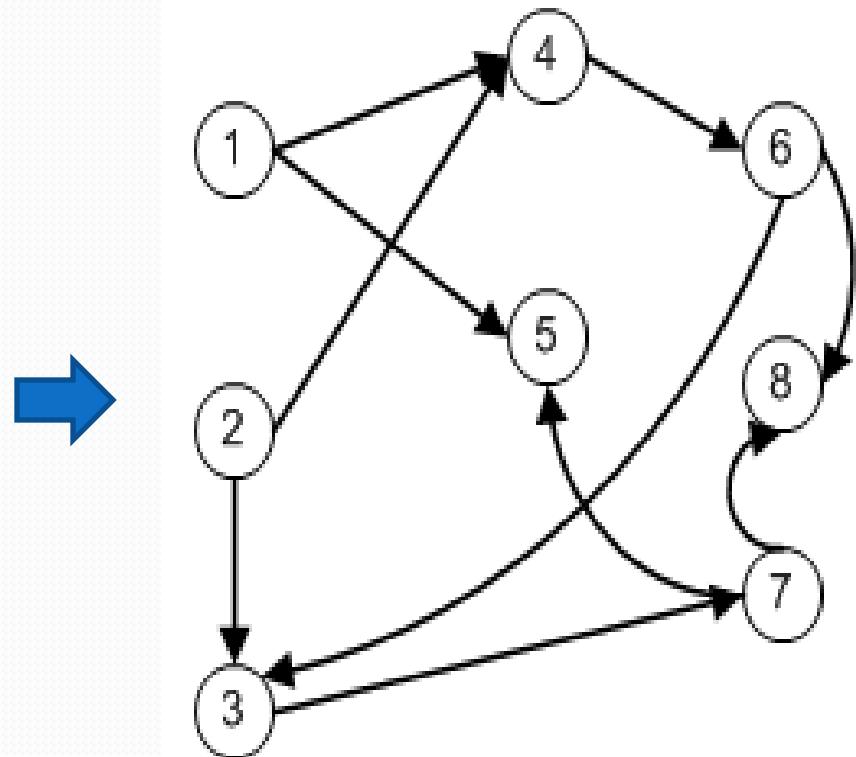
Skema Kitano

S	ACBA	Aadfb	Bbefd	Dfanp	Bahjm	Ckhgf	...
---	------	-------	-------	-------	-------	-------	-----

$$\begin{pmatrix} A & C \\ B & A \end{pmatrix} \xrightarrow{\quad} \begin{pmatrix} a & d & k & h \\ f & b & g & f \\ b & e & a & d \\ f & d & f & b \end{pmatrix} \xrightarrow{\quad} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Skema Kitano

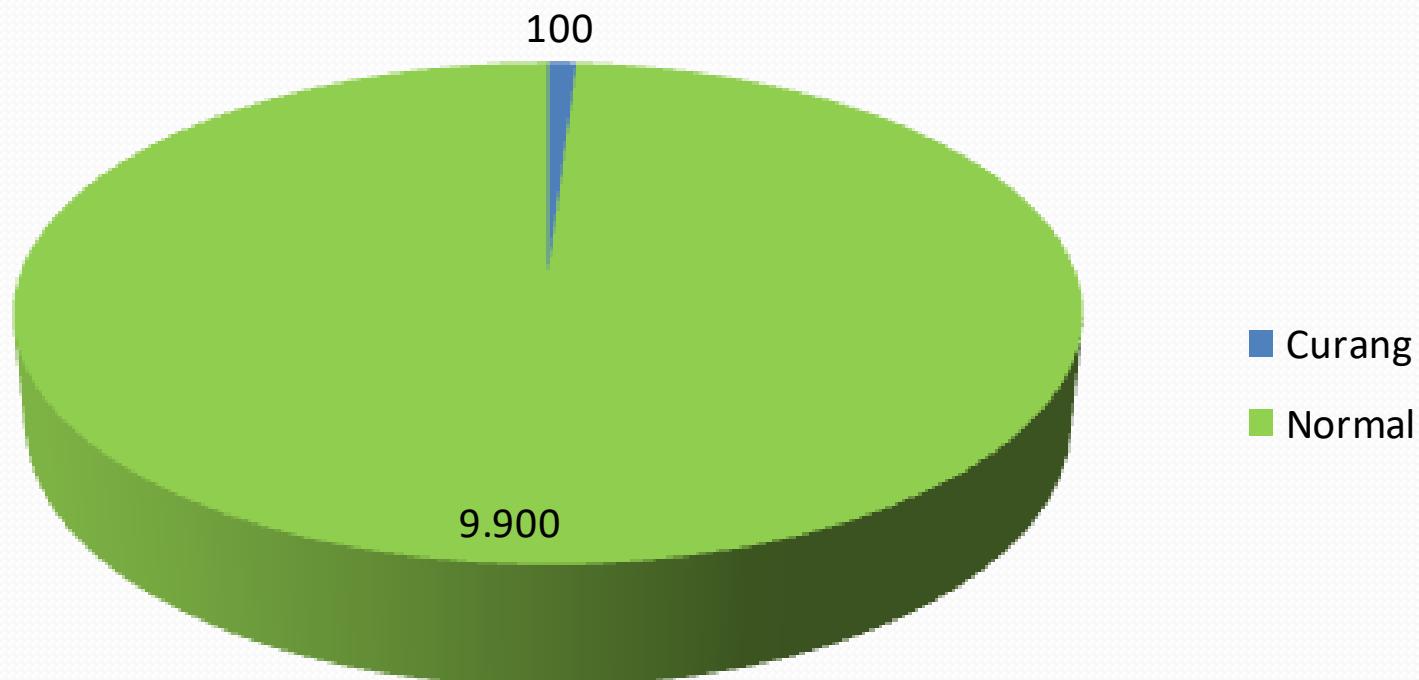
1	0	0	1	1	0	0	0
0	1	1	1	0	0	0	0
0	0	1	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	0	0	1	0	0	0
0	0	1	0	0	1	0	1
0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	1



Masalah Deteksi

- Kecurangan (*fraud*): phone, banking
- Pelanggan potensial
- Penyusupan (intrusion)
- Penyakit Diabetes, Kanker
- Mahasiswa Cumlaude atau DO
- Sistem Keamanan
- Customer Relationship Management (CRM)
- Data Mining 2007: **Couponing**

Imbalance Classification



Kasus 1: DamiCup'07 Couponing

- Mengenali pelanggan yang potensial terhadap *couponing* oleh toko dan pada kupon apakah dia akan merespon?
- Kupon adalah voucher untuk promosi yang diberikan kepada pelanggan melalui berbagai macam media.
- Pelanggan yang membeli barang yang diiklankan di kupon mendapatkan diskon sesuai yang tertera pada kupon.
- Selama ini kupon disebarluaskan melalui majalah atau diberikan pada pelanggan saat berada di kasir.
- Data training: perilaku penebusan kupon oleh pelanggan.
- Pelanggan dibagi menjadi tiga kelas: A, B, dan N.

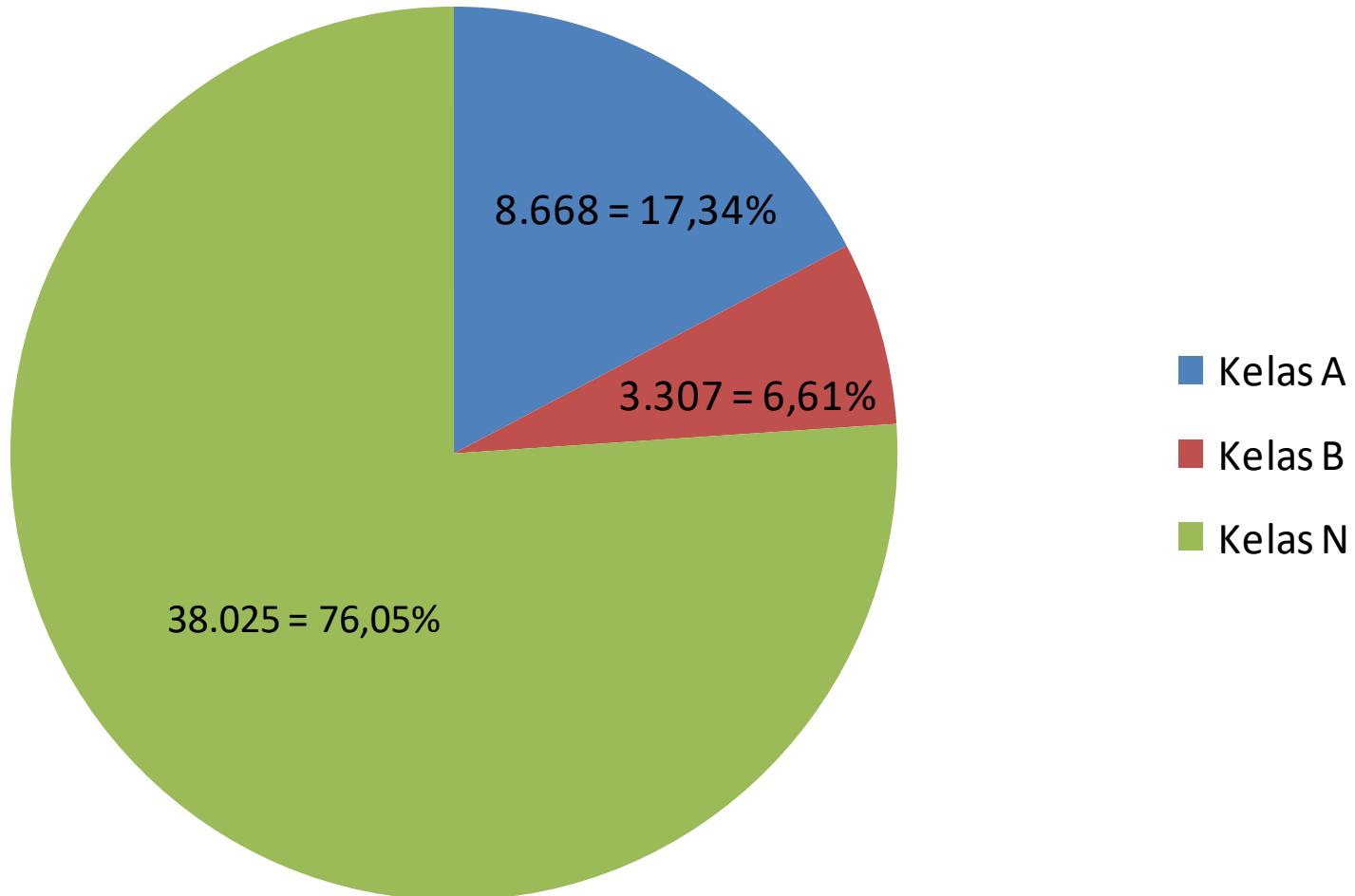
Data training: 20 atribut

- **C10001**: Jumlah penebusan kupon C10001 oleh pelanggan.
- **C10002**: Jumlah penebusan kupon C10002 oleh pelanggan.
- **C10003**: Jumlah penebusan kupon C10003 oleh pelanggan.
- ...
- ...
- **C10020**: Jumlah penebusan kupon C10020 oleh pelanggan.
- **COUPON**: atribut target
- **ID**: Nomor pelanggan

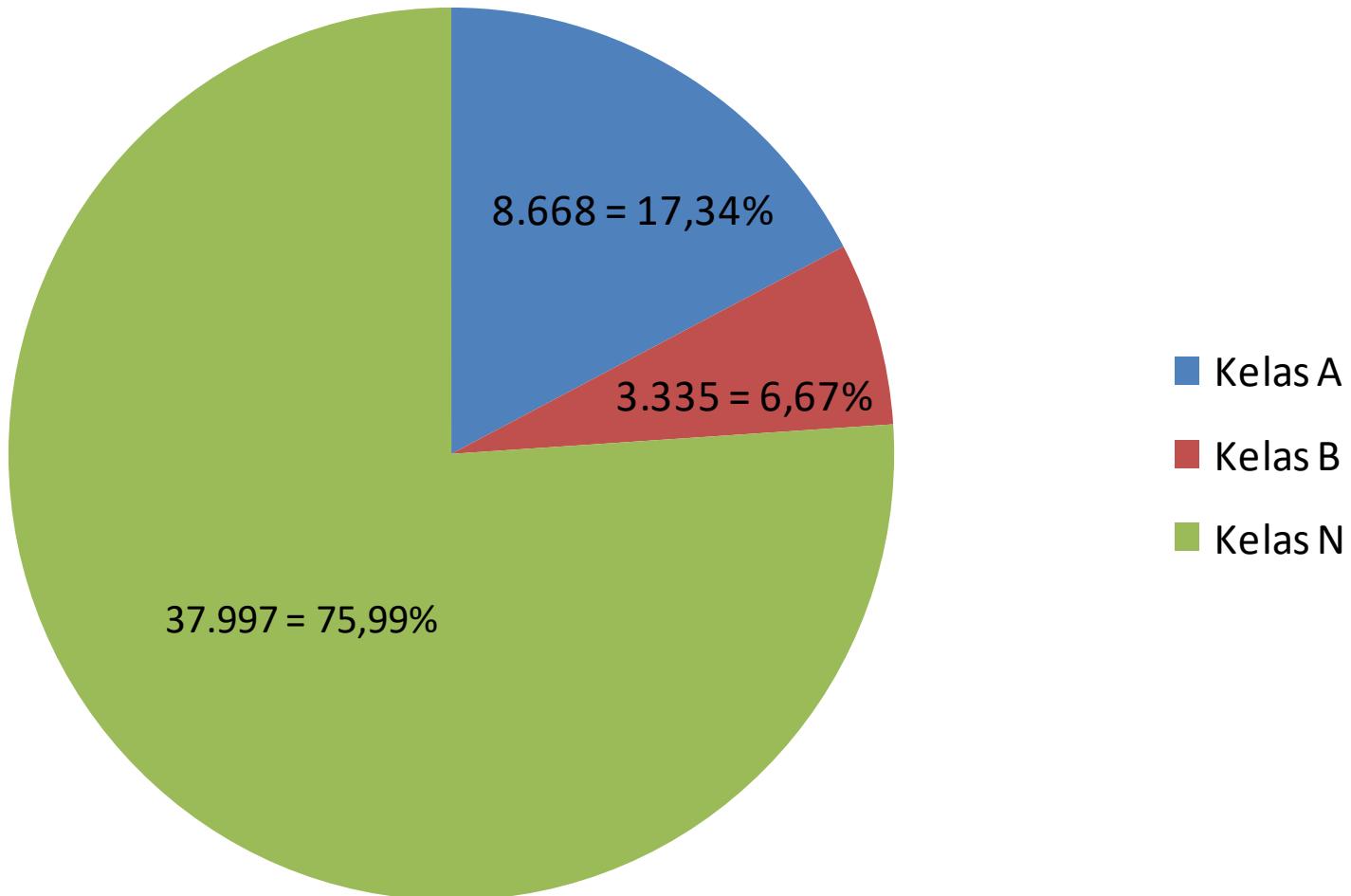
Data training

C10001	C10002	C10003	C10004	C10005	...	C10020	Kelas	ID
0	0	0	0	0	...	0	A	626
0	0	0	0	0	...	0	A	38
0	0	0	0	0	...	0	A	7
0	1	0	0	0	...	0	B	1
0	1	0	0	0	...	0	B	2
0	0	0	0	0	...	0	B	1
0	0	1	1	0	...	0	N	272
0	0	1	1	0	...	0	N	30
0	0	1	1	0	...	0	N	5
0	0	1	1	0	...	0	N	2
1	0	1	1	0	...	0	N	2
0	0	1	1	1	...	0	N	3
0	0	1	1	1	...	0	N	13
0	0	1	1	0	...	0	N	231
0	0	1	1	0	...	0	N	18

Training Set: 50.000 records



Testing Set: 50.000 records



Sampling

- Undersampling
 - Mengurangi distribusi kelas mayoritas
 - Mengambil *record* kelas mayoritas secara acak
 - Record yang memiliki informasi penting tidak terpilih?
- Oversampling
 - Menambah distribusi kelas minoritas
 - Menduplikasi record kelas minoritas
 - Noise yang ada pada kelas minoritas ikut terduplikasi?
 - Data training semakin banyak → komputasi tinggi
- Gabungan undersampling & oversampling

Data training asli

- Kelas A = 8.668
- Kelas B = 3.307
- Kelas N = 38.025

Data Undersampling 1

- Data *undersampling* dengan patokan kelas B
- Data kelas A dan N di-*undersampling* sampai jumlahnya sama dengan data kelas B.
- Kelas A = 3.307
- Kelas B = 3.307
- Kelas N = 3.307

Data Undersampling 2

- Data *undersampling* dengan patokan kelas A
- Data kelas B dan N di-*undersampling* sampai jumlahnya sama dengan data kelas A.
- Kelas A = 8.668
- Kelas B = 8.668
- Kelas N = 8.668

Data Oversampling 1

- Data *oversampling* dengan patokan kelas N
- Data kelas A dan B di- *oversampling* sampai jumlahnya sama dengan data kelas N.
- Kelas A = 38.025
- Kelas B = 38.025
- Kelas N = 38.025

Data Oversampling 2

- Data kelas A di-*oversampling* tiga kali lipat
- Data kelas B di-*oversampling* enam kali lipat
- Kelas A = 26.004
- Kelas B = 19.842
- Kelas N = 38.025

Datasets

Semua data dibagi menjadi dua bagian

- Trainset : 2/3
- Validationset : 1/3

Evolving ANN

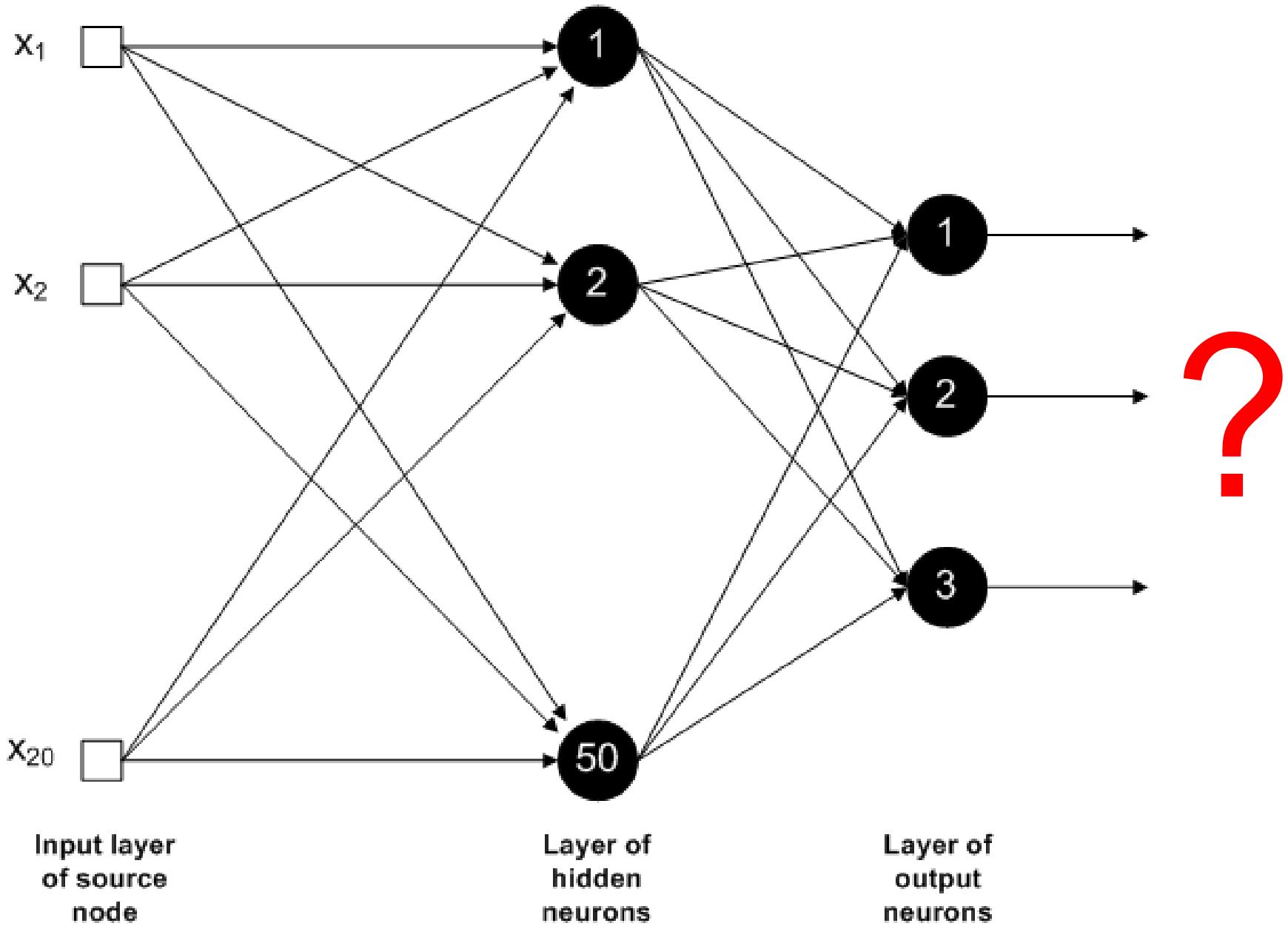
- Evolusi **STRUKTUR & WEIGHTS**
- Struktur ANN?
- Evolusi: GA dan ES?
- Kromosom
- Fitness

Data training

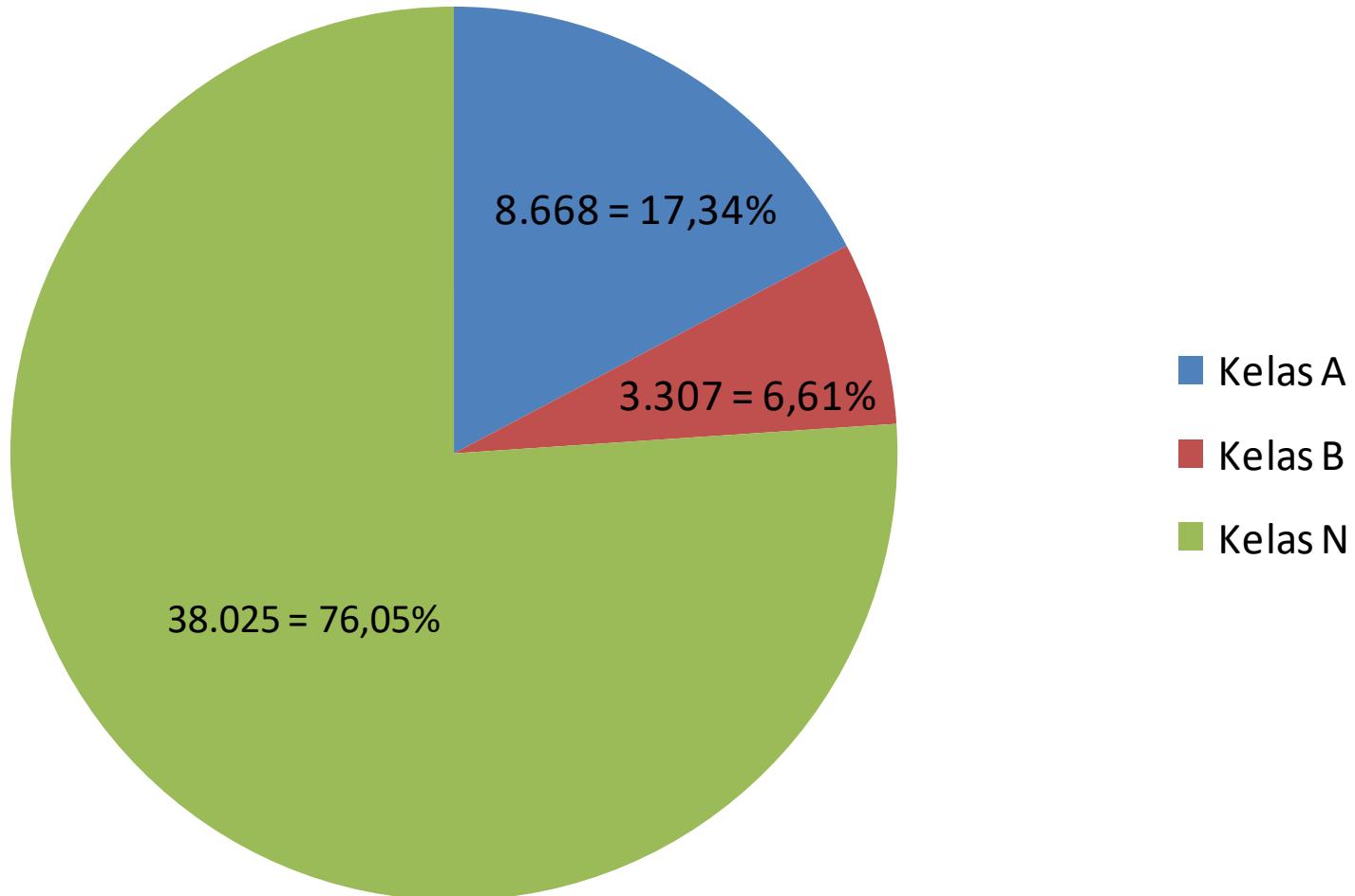
C10001	C10002	C10003	C10004	C10005	...	C10020	Kelas	ID
0	0	0	0	0	...	0	A	626
0	0	0	0	0	...	0	A	38
0	0	0	0	0	...	0	A	7
0	1	0	0	0	...	0	B	1
0	1	0	0	0	...	0	B	2
0	0	0	0	0	...	0	B	1
0	0	1	1	0	...	0	N	272
0	0	1	1	0	...	0	N	30
0	0	1	1	0	...	0	N	5
0	0	1	1	0	...	0	N	2
1	0	1	1	0	...	0	N	2
0	0	1	1	1	...	0	N	3
0	0	1	1	1	...	0	N	13
0	0	1	1	0	...	0	N	231
0	0	1	1	0	...	0	N	18

Struktur ANN

- Arsitektur : Recurrent Network
- Input : 20
- Output neuron : 3
- Hidden neuron : 50
- Masuk akal?

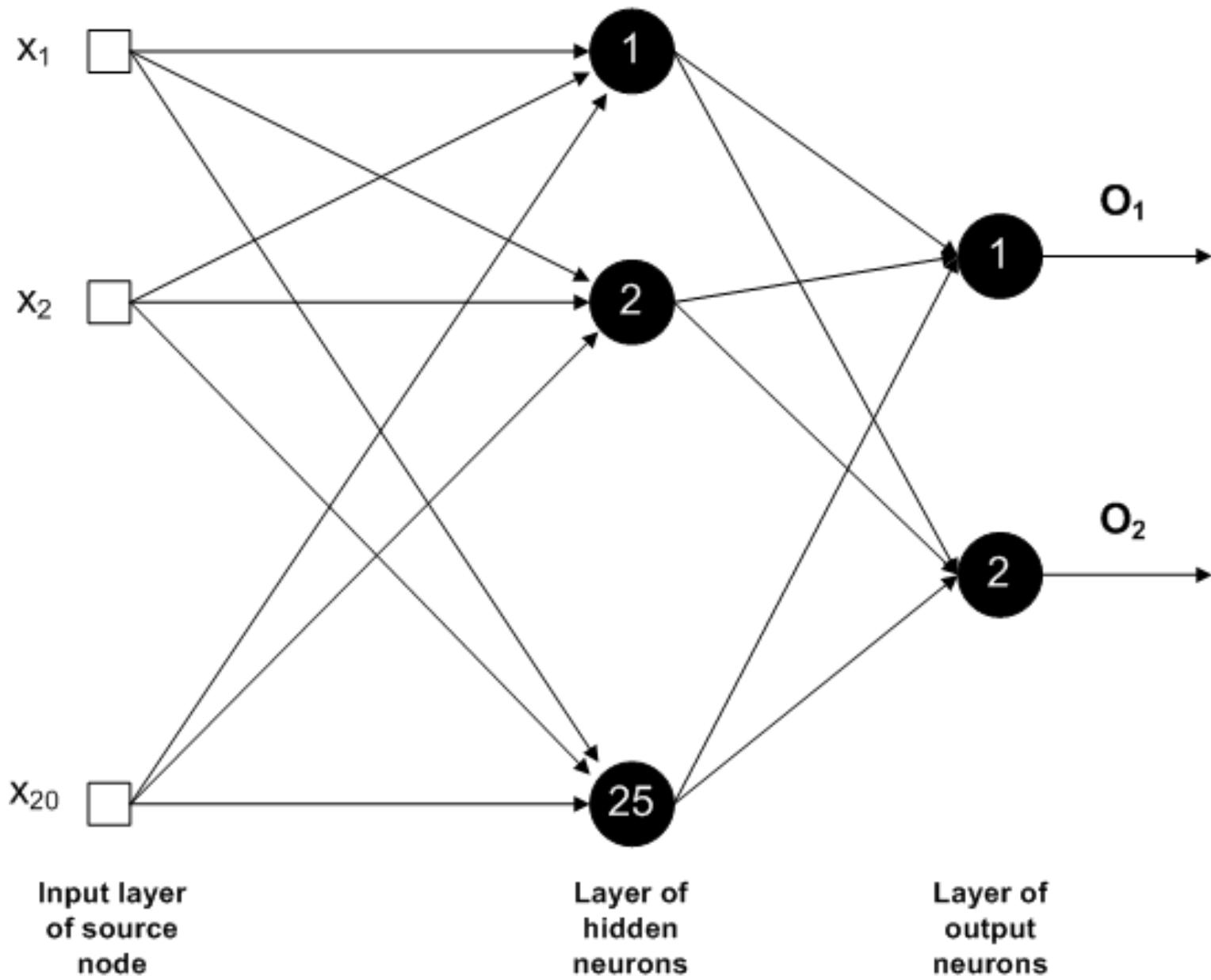


Training Set: 50.000 records



Arsitektur ANN [Danang 2008]

- Recurrent Network
- Input layer : 20 (atribut)
- Hidden layer : maksimum 25 nodes
- Output layer : 2 node (O_1 dan O_2)



Skema klasifikasi

- IF $O_1 \geq 0.5$ AND $O_2 < 0.5$ THEN kelas = A
- IF $O_1 < 0.5$ AND $O_2 \geq 0.5$ THEN kelas = B
- IF $O_1 \geq 0.5$ AND $O_2 \geq 0.5$ AND $O_1 > O_2$ THEN kelas = A
- IF $O_1 \geq 0.5$ AND $O_2 \geq 0.5$ AND $O_1 < O_2$ THEN kelas = B
- Otherwise kelas = N

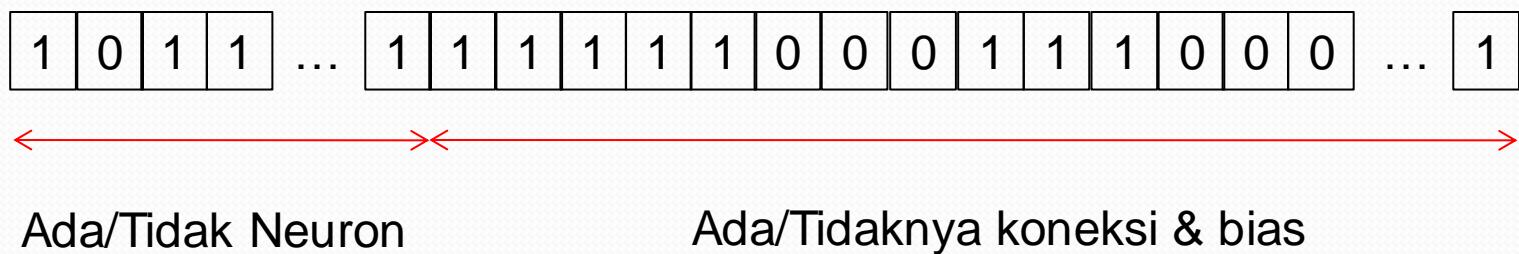
Evolving ANN

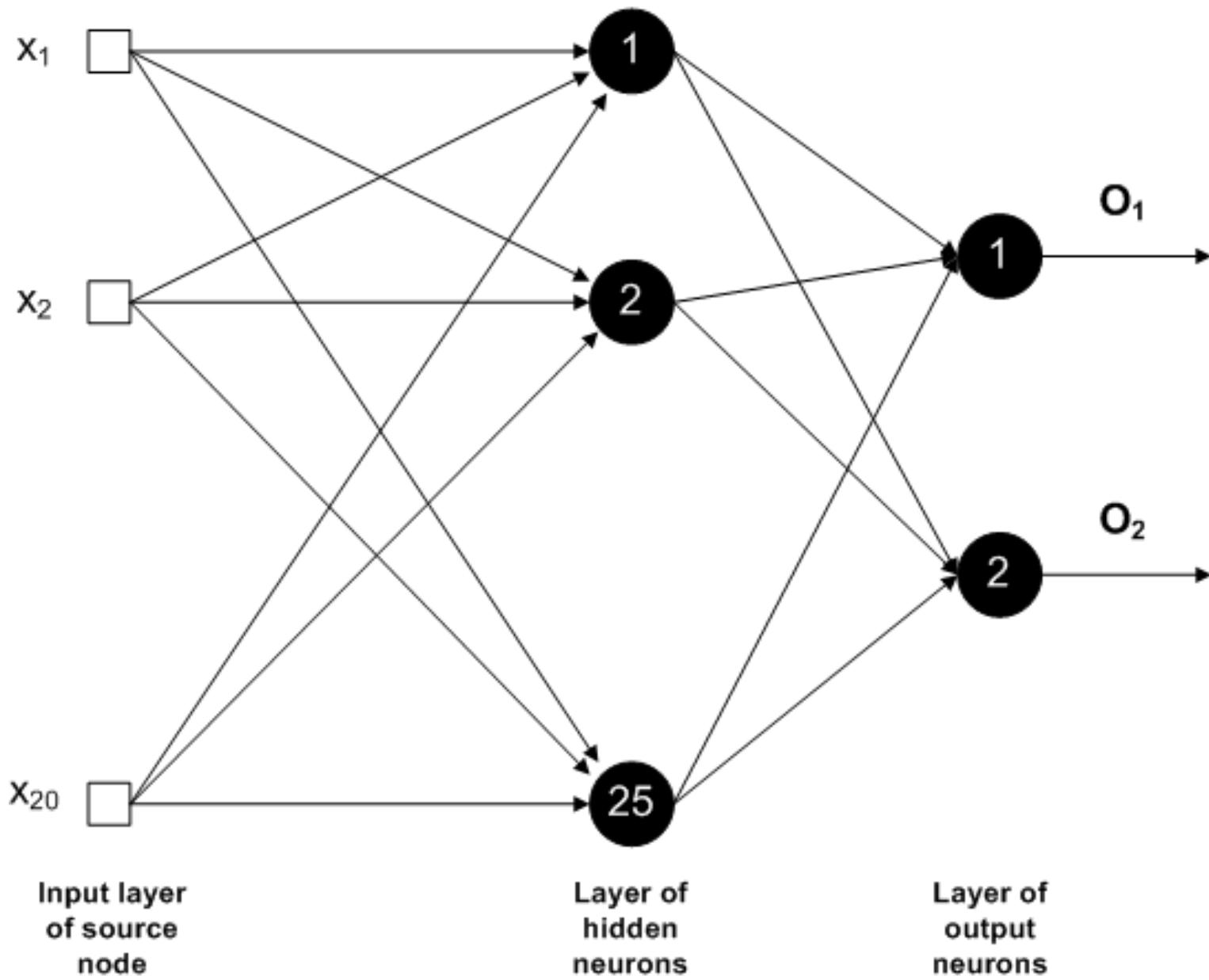
- Evolusi STRUKTUR & WEIGHTS
- Struktur ANN?
- Evolusi: GA dan ES
- Kromosom?
- Fitness?

Kromosom

- Kromosom untuk optimasi struktur
 - Biner
 - Hidden Neuron
 - Koneksi
- Kromosom untuk optimasi weights
 - Real
 - Bobot

Kromosom untuk optimasi struktur





Kromosom untuk optimasi weights

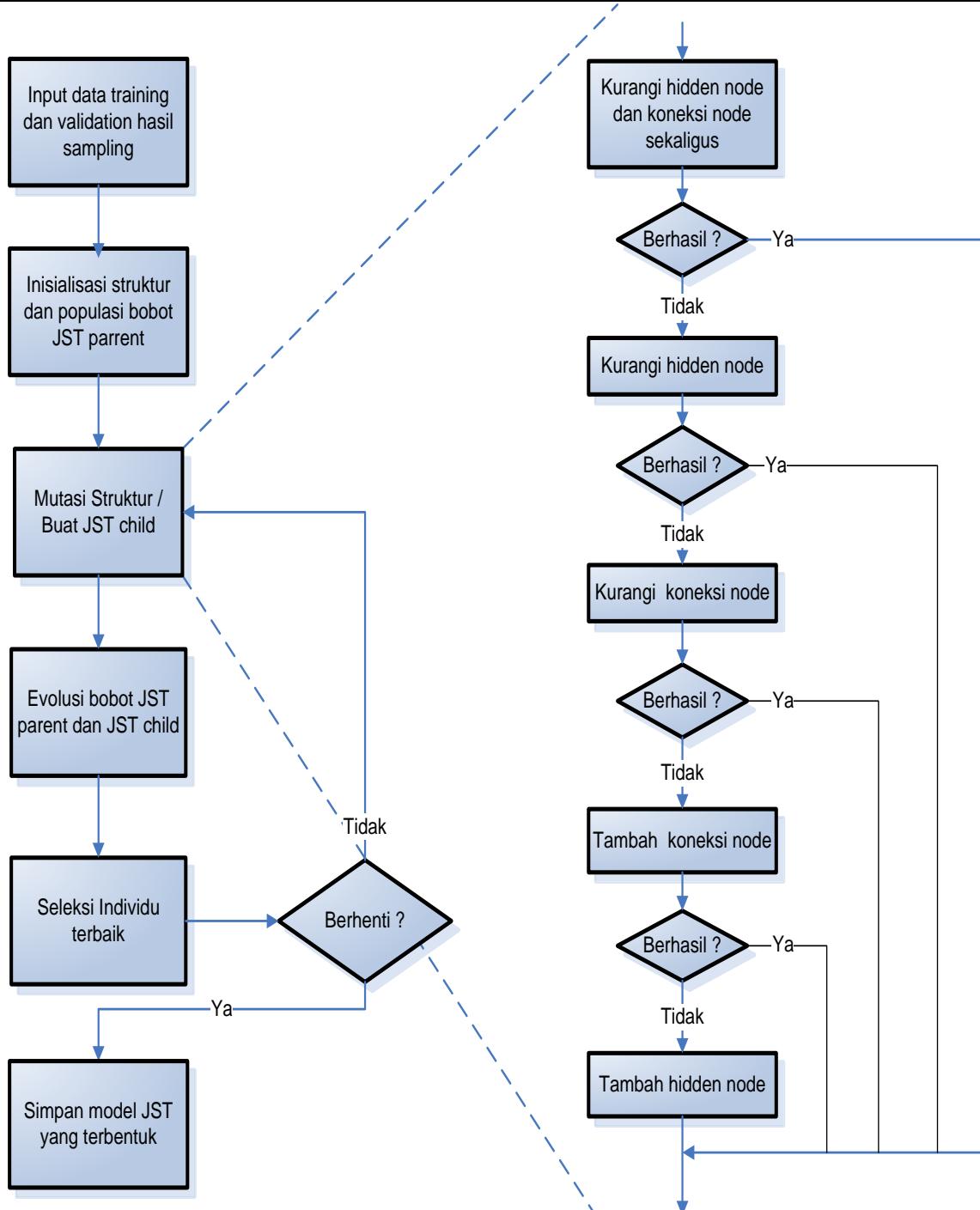
$$\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n, \alpha_1, \dots, \alpha_k \rangle$$

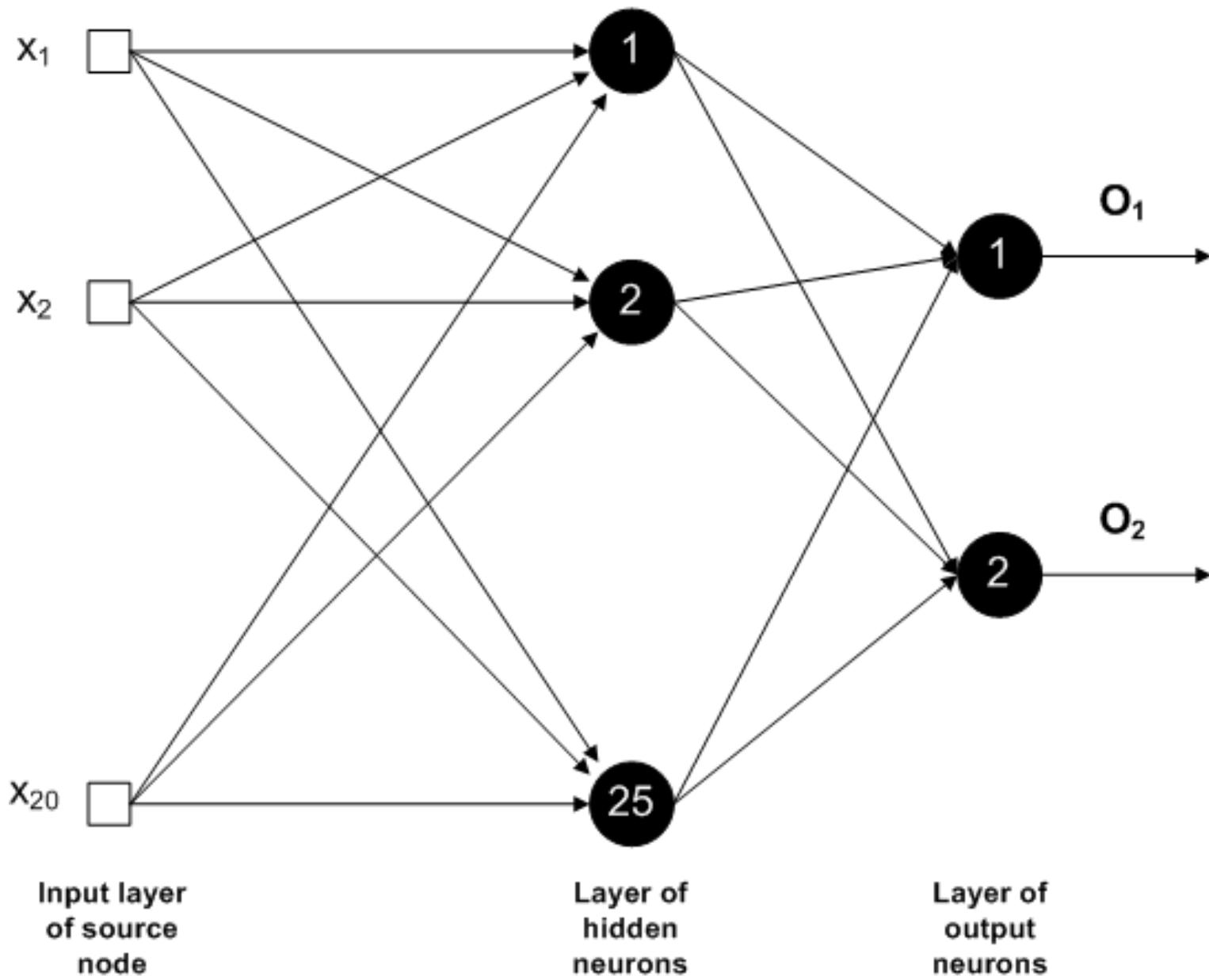
Fitness

- Fitness untuk Struktur
 - Untuk 100 epoch training, hitung MSE-nya
 - $\text{Fitness} = 1 / \text{MSE}_{(100 \text{ epoch})}$
-
- Fitness untuk Weights
 - Untuk semua trainset, lakukan perhitungan maju
 - $\text{Fitness} = 1 / \text{MSE}_{(\text{maju})}$

Evolving ANN berbasis EP

[Danang, 2008]





Skema klasifikasi

- IF $O_1 \geq 0.5$ AND $O_2 < 0.5$ THEN kelas = A
- IF $O_1 < 0.5$ AND $O_2 \geq 0.5$ THEN kelas = B
- IF $O_1 \geq 0.5$ AND $O_2 \geq 0.5$ AND $O_1 > O_2$ THEN kelas = A
- IF $O_1 \geq 0.5$ AND $O_2 \geq 0.5$ AND $O_1 < O_2$ THEN kelas = B
- Otherwise kelas = N

Formulasi EP

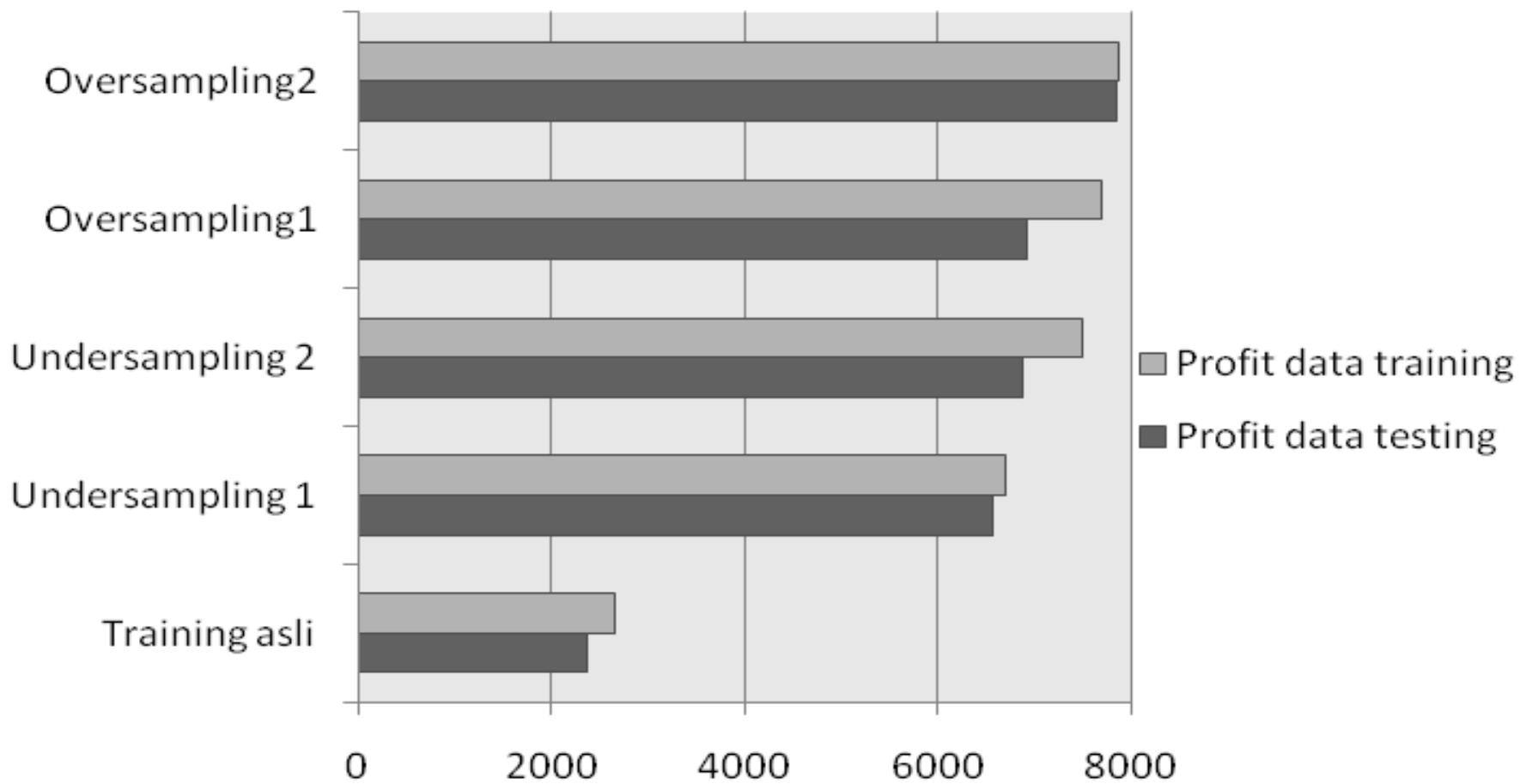
- Fitness = $1 / (\text{MSE dari } O_1 + \text{MSE dari } O_2 + o,1)$
- Nilai *fitness* untuk bobot ANN dihitung menggunakan data *training*.
- Nilai *fitness* untuk struktur ANN dihitung menggunakan data *validation*.

		Prediksi		
		Kelas A	Kelas B	Kelas N
Aktual	Kelas A	AA	AB	AN
	Kelas B	BA	BB	BN
	Kelas N	NA	NB	NN

		Prediksi		
		Kelas A	Kelas B	Kelas N
Aktual	Kelas A	3	-1	0
	Kelas B	-1	6	0
	Kelas N	-1	-1	0

$$\text{Profit} = 3^*AA + 6^*BB - 1^*(NA+NB+BA+AB)$$

Pengaruh data training

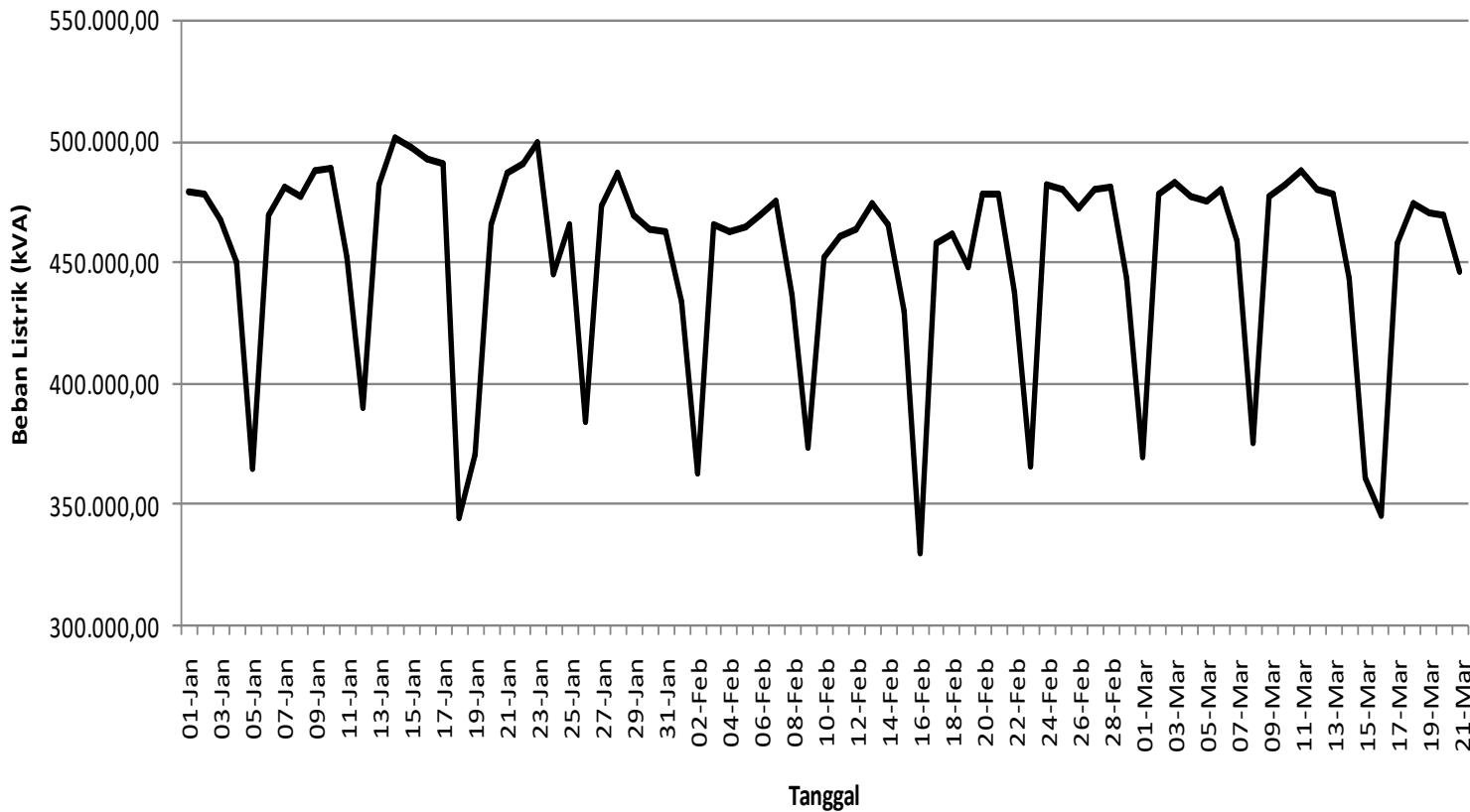


Hasil Akhir

	Training asli	Under Sampling 1	Under Sampling 2	Over Sampling 1	Over Sampling 2
Hidden node	25	17	25	25	25
Koneksi	550	357	502	550	467
Akurasi (%)	77,29	?	66,73	67,40	67,14
Profit data training	2662	6704	7489	7692	7877
Profit data testing	2361	6578	6870	6923	7861

Kasus 2: Prediksi Data Timeseries

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



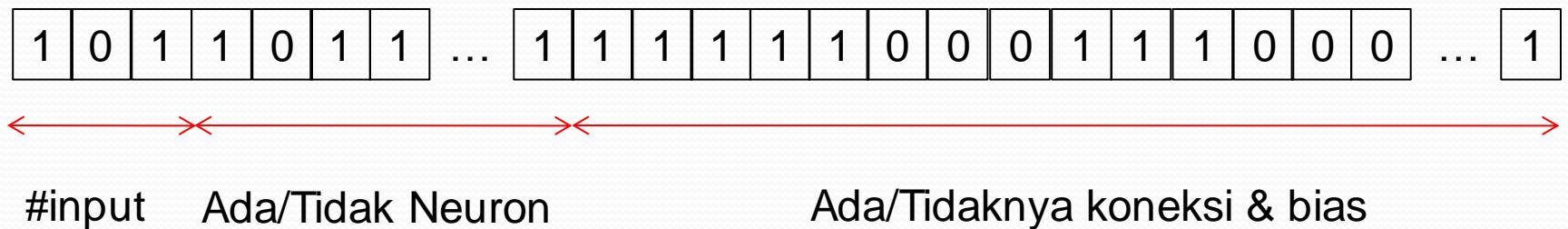
Evolving ANN

- Evolusi STRUKTUR & WEIGHTS
- Struktur ANN?
- Evolusi: GA dan ES
- Kromosom?
- Fitness?

Kromosom

- Kromosom untuk optimasi struktur
 - Biner
 - Hidden Neuron
 - Koneksi
- Kromosom untuk optimasi weights
 - Real
 - Bobot

Kromosom untuk optimasi struktur

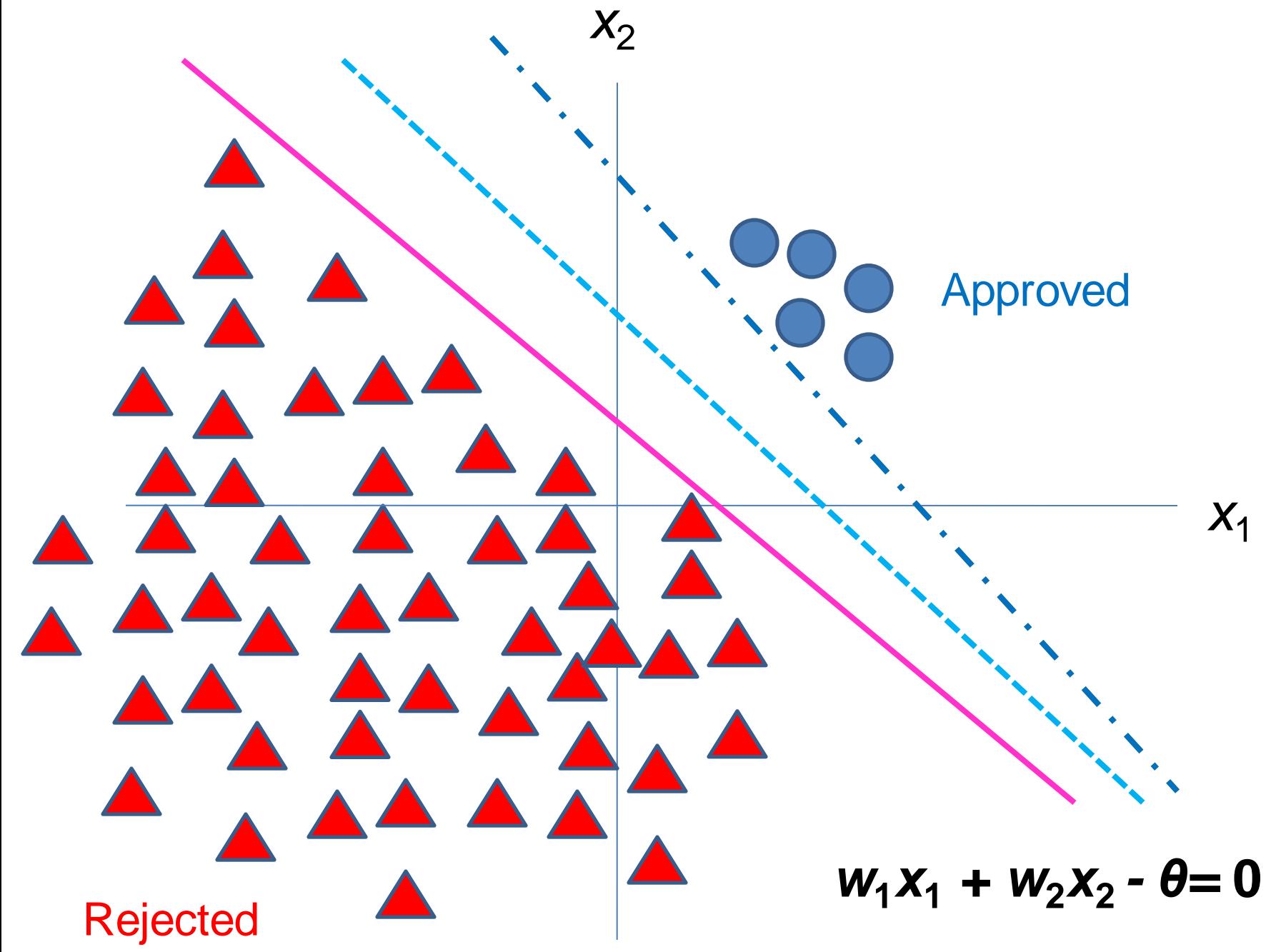


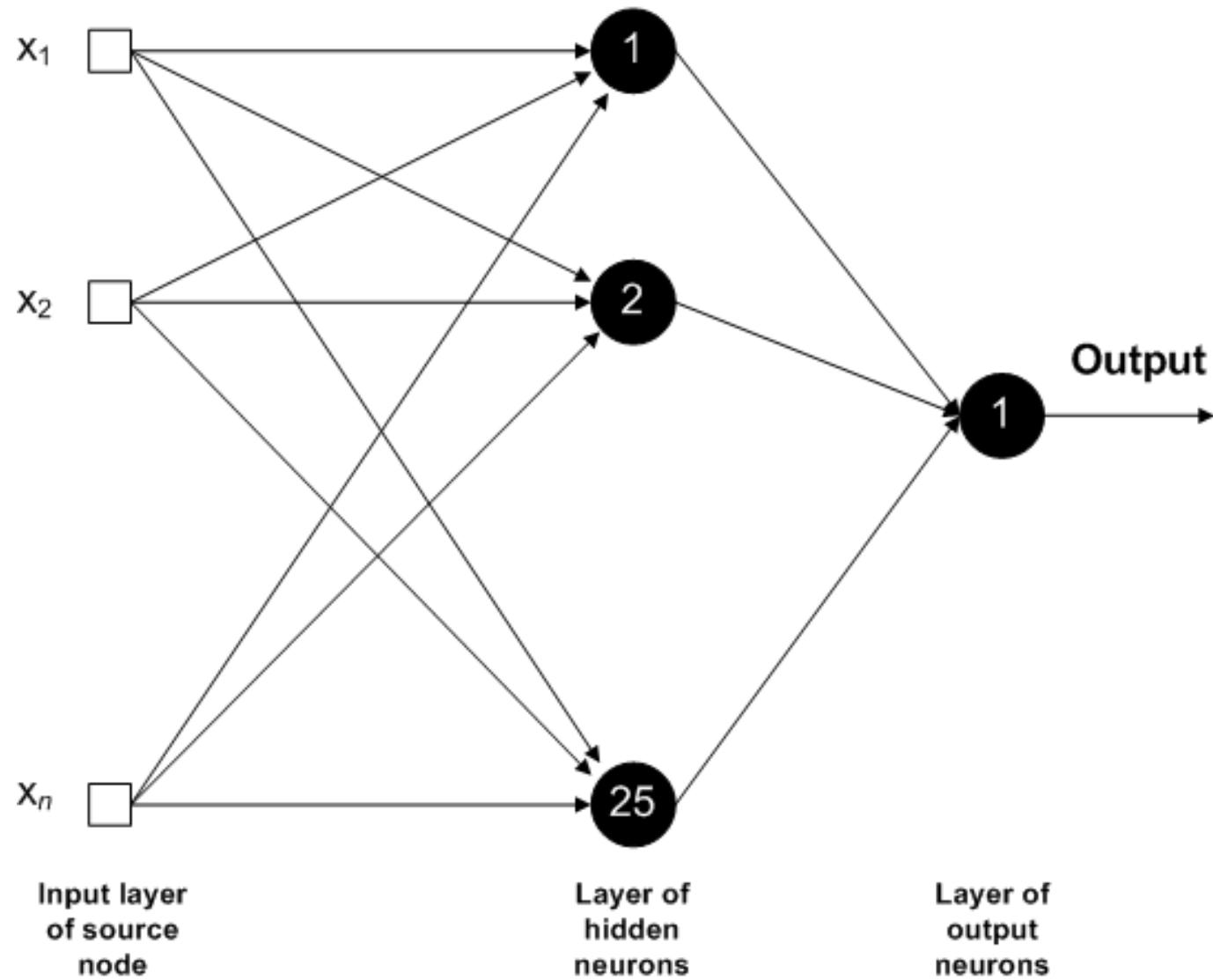
Kromosom untuk optimasi weights

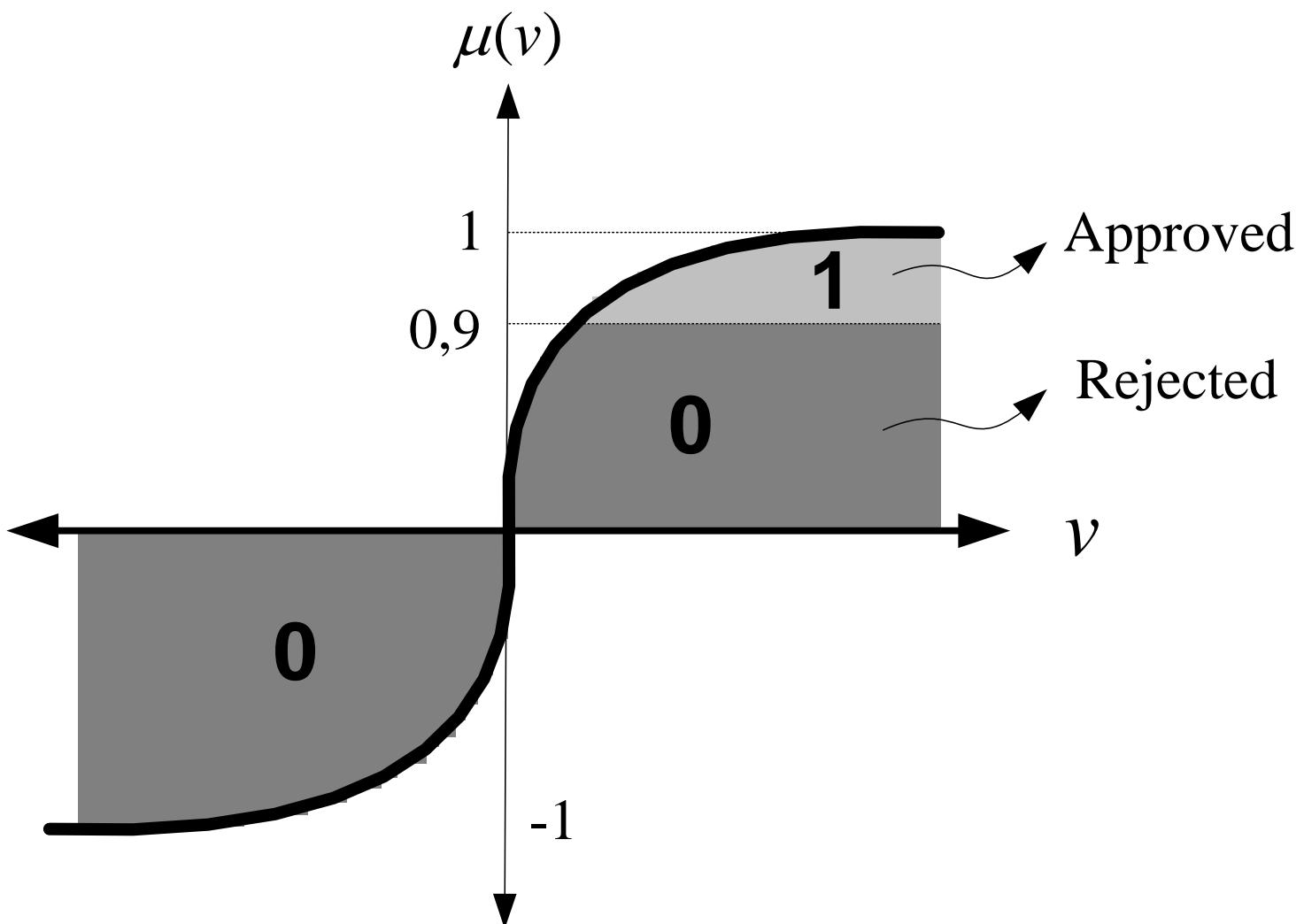
$$\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n, \alpha_1, \dots, \alpha_k \rangle$$

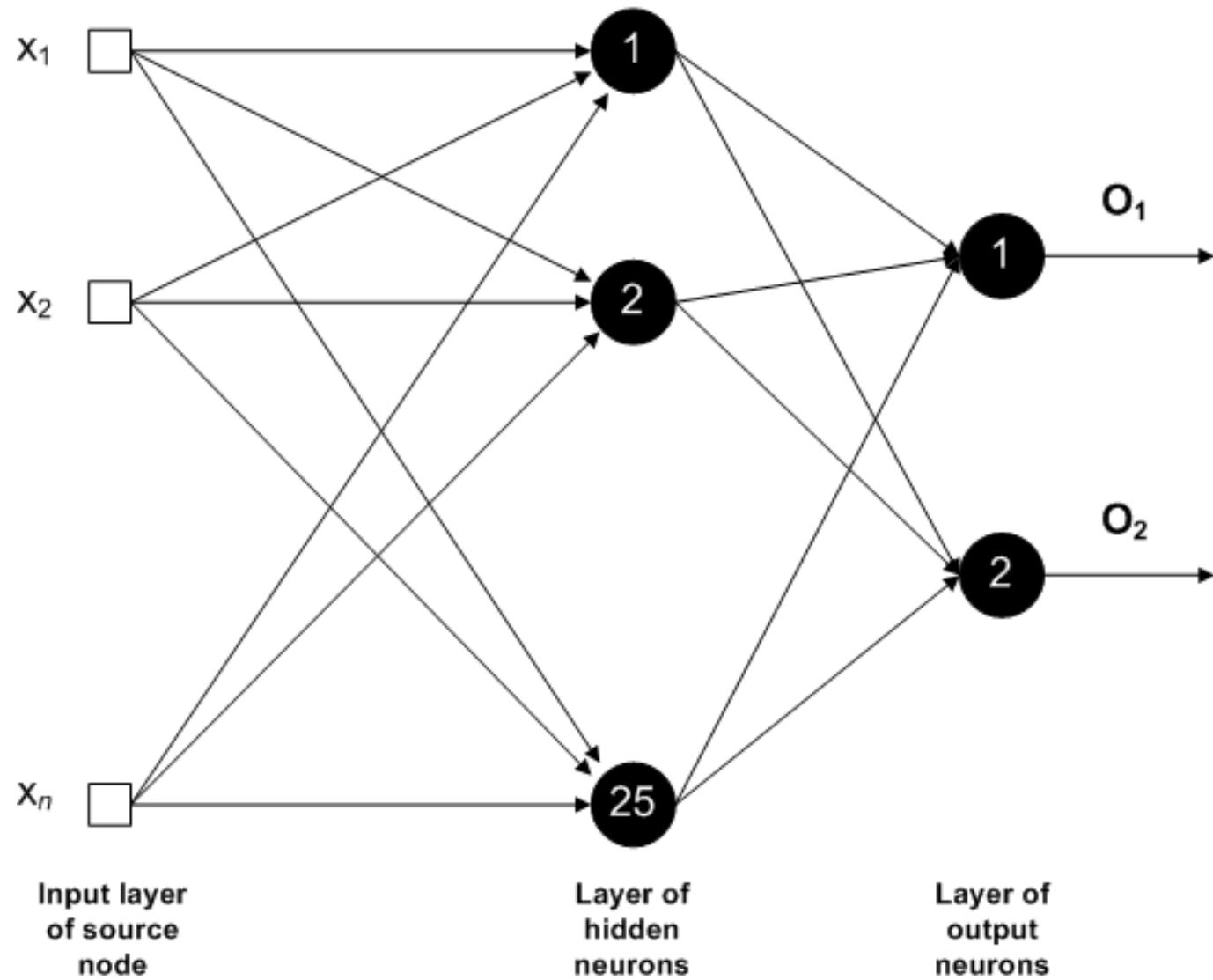
Kasus 3: Security System

- Evolusi STRUKTUR & WEIGHTS
- Struktur ANN?
- Evolusi: GA dan ES
- Kromosom?
- Fitness?







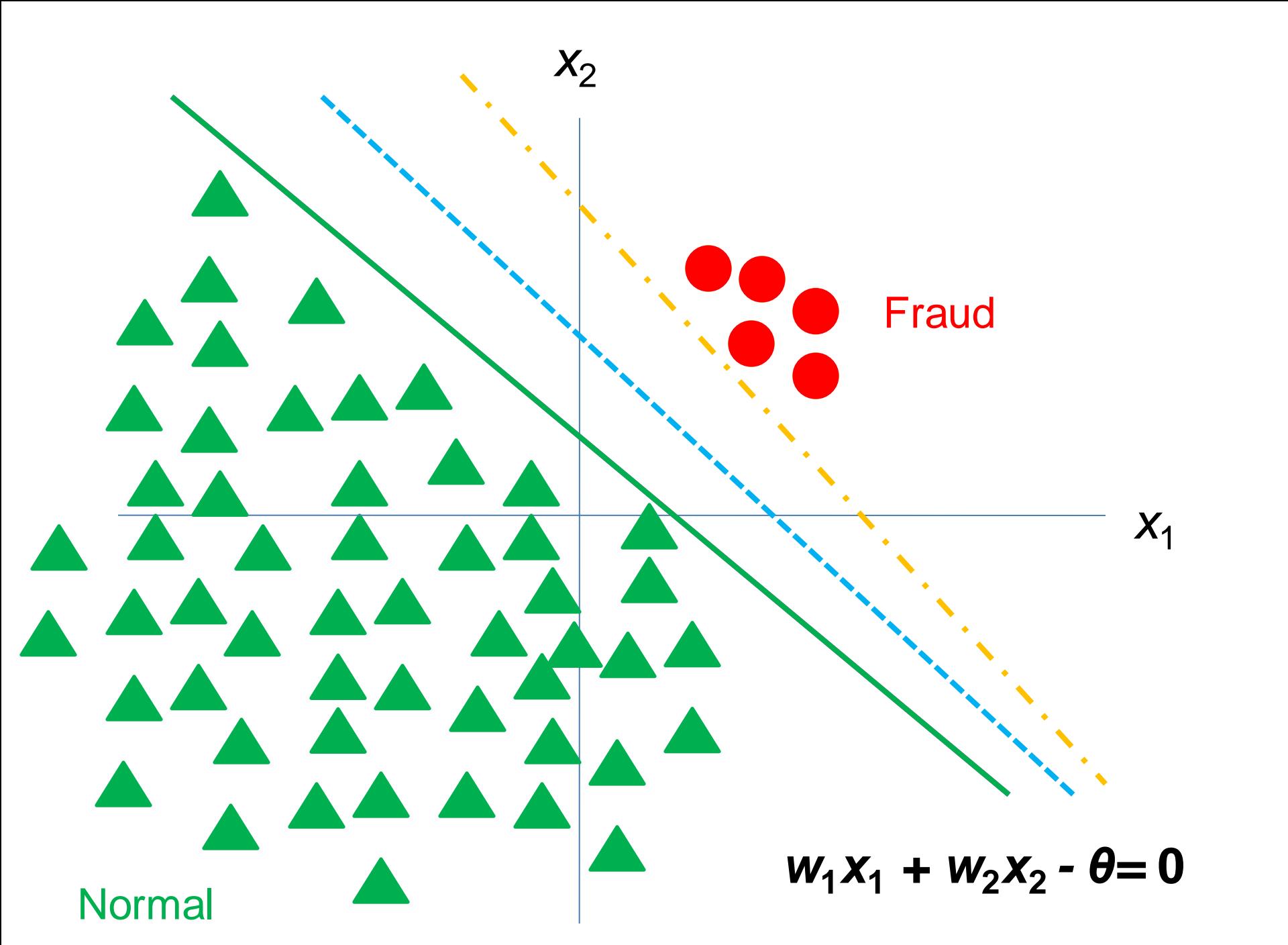


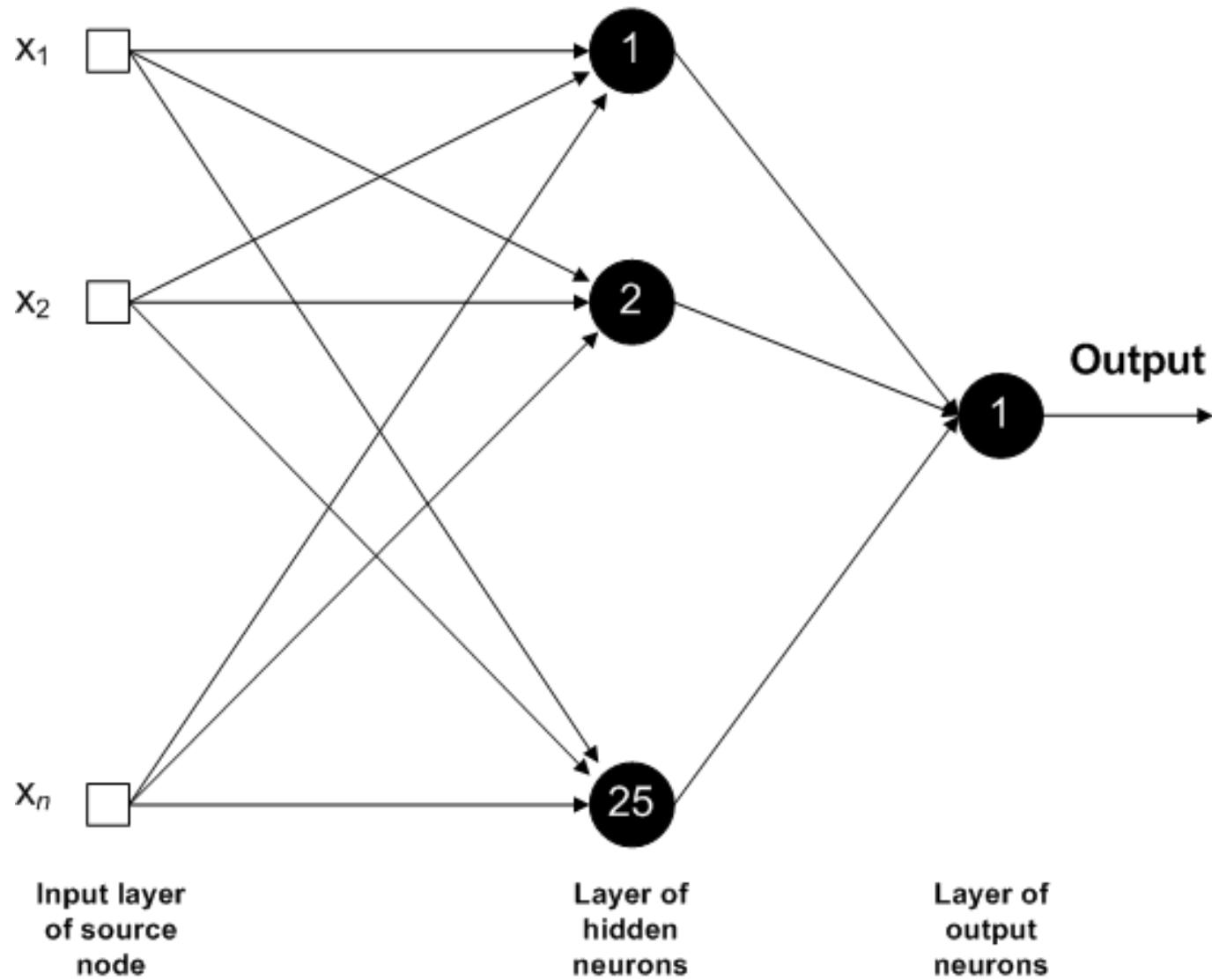
Skema klasifikasi

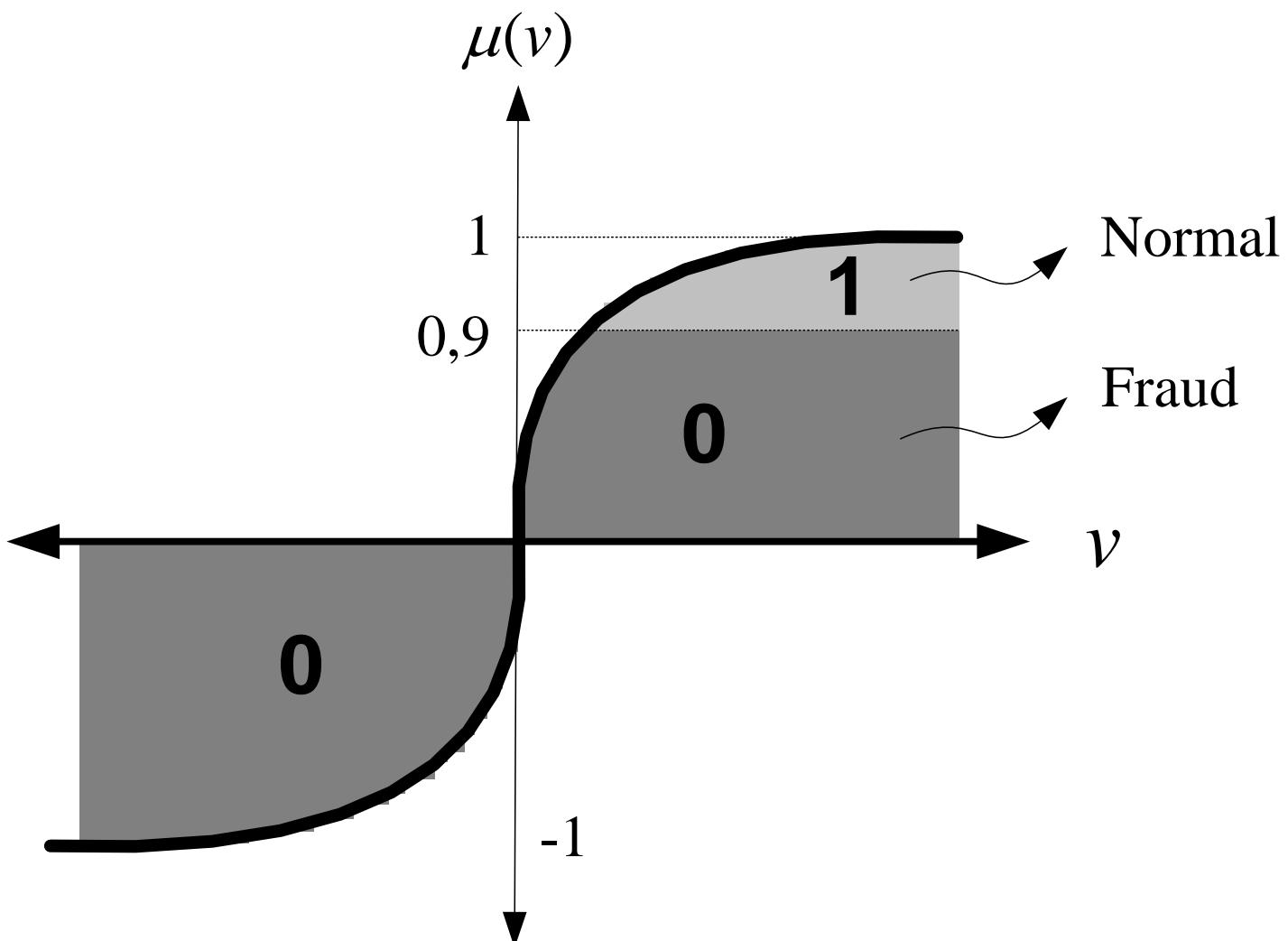
- IF $O_1 \geq 0.5$ AND $O_2 < 0.1$ THEN kelas = Approved
- Otherwise kelas = Rejected

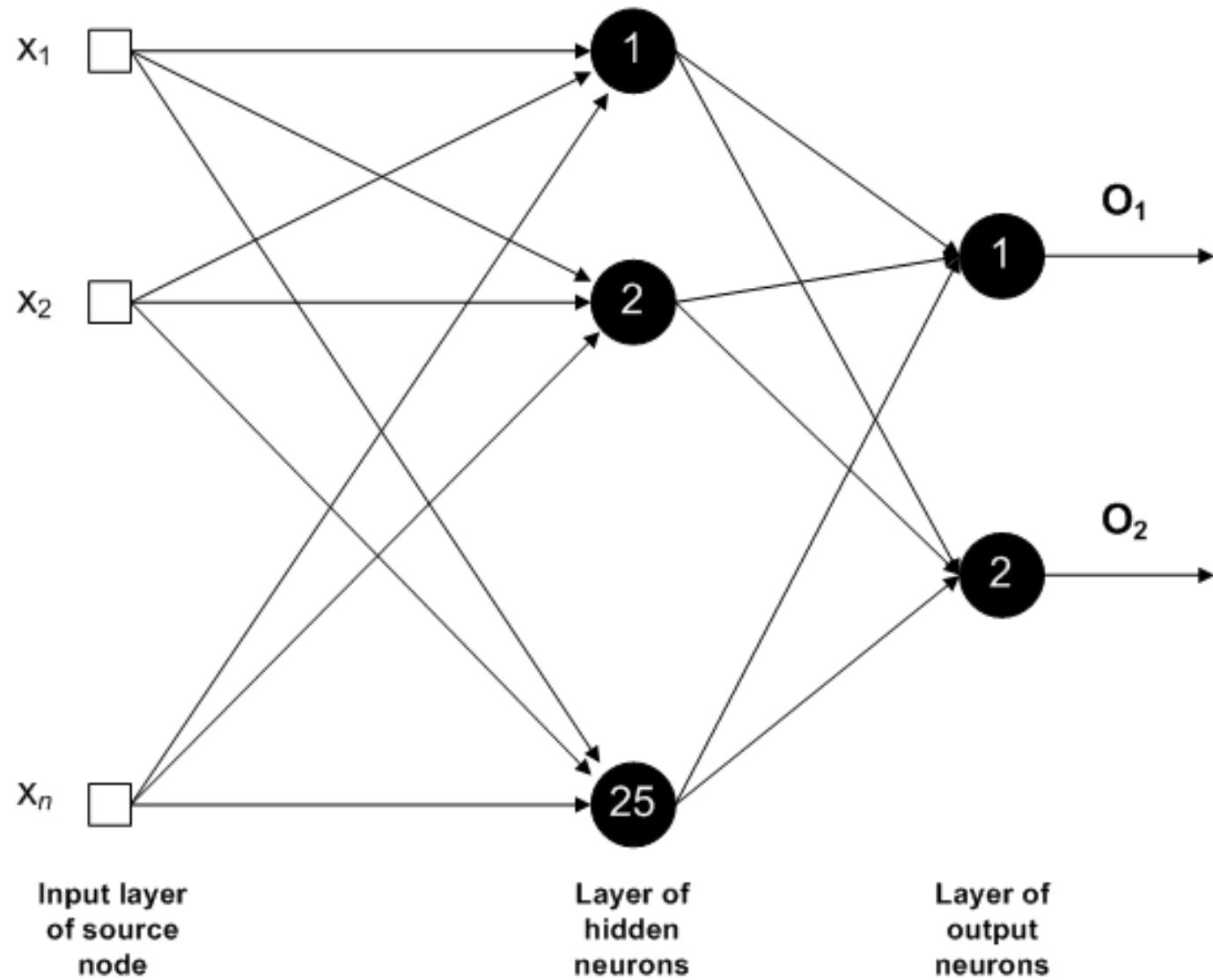
Kasus 4: Fraud Detection

- Evolusi STRUKTUR & WEIGHTS
- Struktur ANN?
- Evolusi: GA dan ES
- Kromosom?
- Fitness?









Skema klasifikasi

- IF $O_1 \geq 0.5$ AND $O_2 < 0.1$ THEN kelas = Normal
- Otherwise kelas = Fraud

Daftar Pustaka

- Suyanto, 2008, “Soft Computing: Membangun Mesin Ber-IQ Tinggi”, Informatika, Bandung Indonesia. ISBN: 978-979-1153-49-2.
- Suyanto, 2008, “Evolutionary Computation: Komputasi Berbasis “Evolusi” dan “Genetika””, Informatika, Bandung Indonesia. ISBN: 978-979-1153-38-6.
- Suyanto, 2005, “Algoritma Genetika dalam MATLAB”, Andi Publisher, Yogyakarta Indonesia, ISBN: 979-731-727-7.

Daftar Pustaka

- [SUY08] Suyanto, 2008, "Soft Computing: Membangun Mesin Ber-IQ Tinggi", Informatika, Bandung Indonesia. ISBN: 978-979-1153-49-2.
- [TET01] 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.
- [DANO8] Danang Risang Djati, 2008, "Analisis dan Implementasi Evolving Artificial Neural Networks dalam Klasifikasi Data Mining yang Memiliki Masalah Imbalance Class dengan Pendekatan Sampling", Tugas Akhir, Informatika, IT Telkom, Bandung.

Evolving Fuzzy Systems

Dr. Suyanto, S.T., M.Sc.
HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

Perbandingan EAs dan Fuzzy Systems

Kriteria	EAs	Fuzzy Systems
Sangat baik untuk masalah dengan informasi yang kurang presisi, tidak lengkap dan memiliki kebenaran parsial?	Tidak	Ya
Sangat baik untuk optimasi, khususnya permasalahan kombinatorial?	Ya	Tidak
Bisa <i>learning</i> ?	Ya	Tidak

Permasalahan *Fuzzy Systems*

- Fungsi keanggotaan: **bentuk & batas2**
- Bagaimana membangun aturan *fuzzy*?

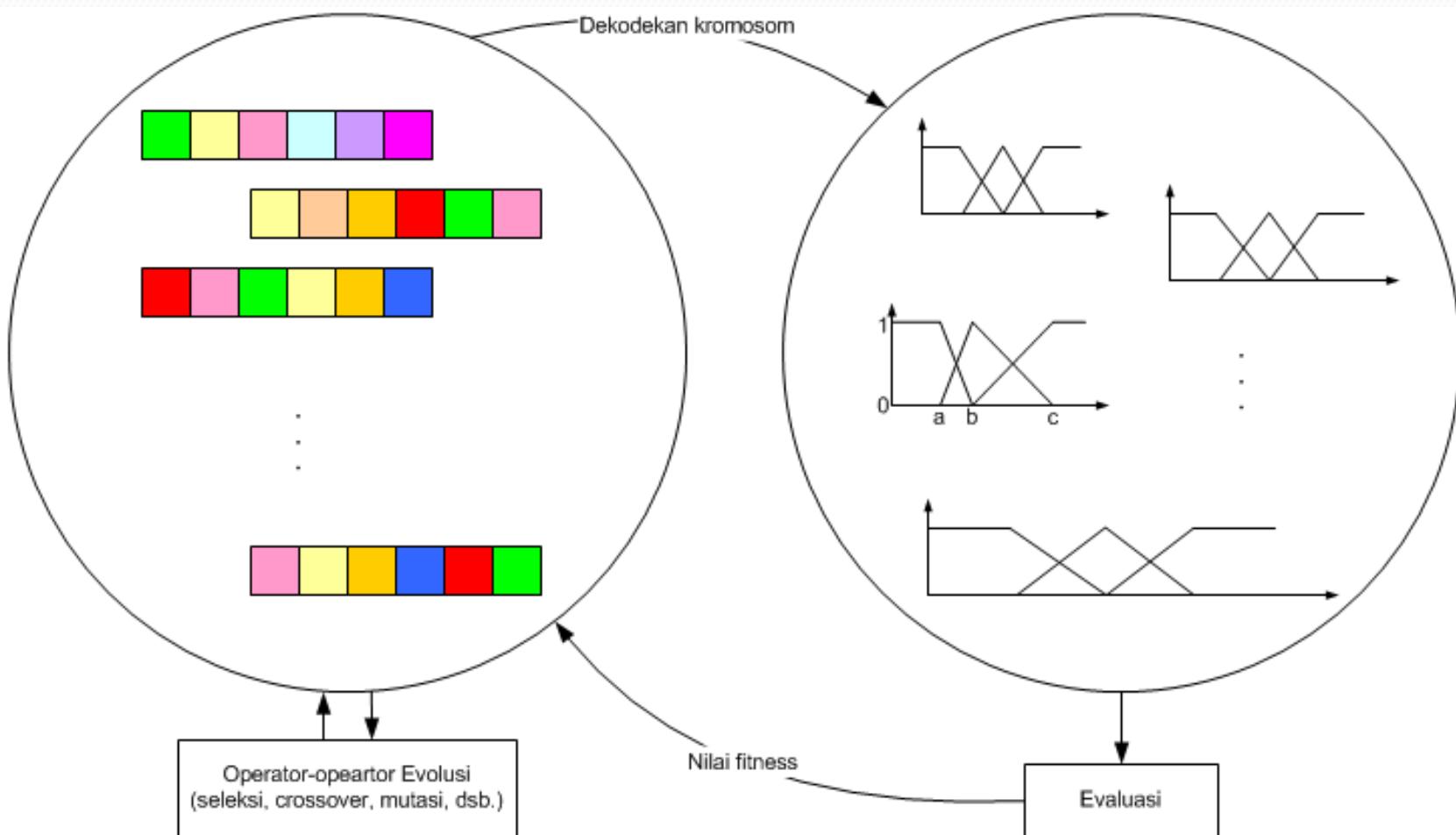
Solusi?

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

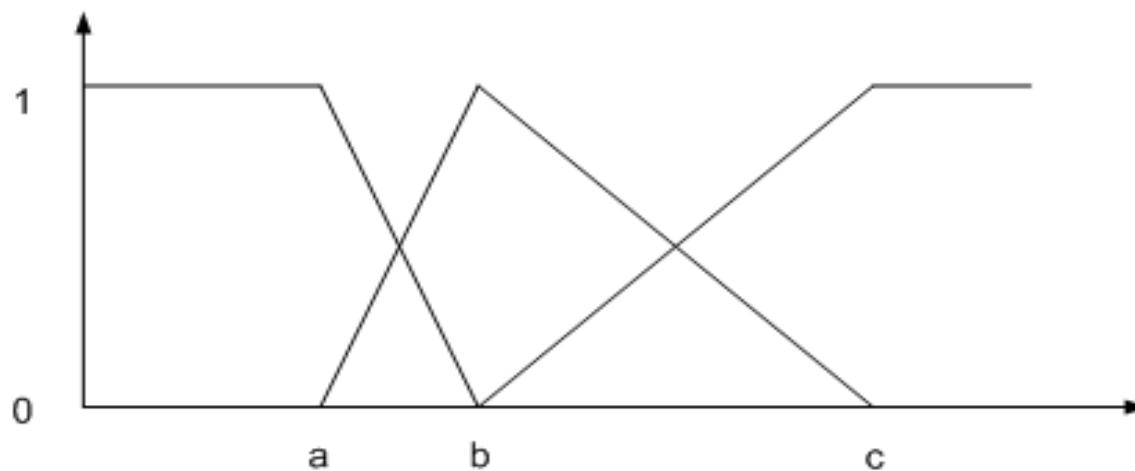
Tuning fungsi keanggotaan

- Bentuk dan batas2 fungsi keanggotaan dibangkitkan berdasarkan data numerik.
- Untuk menghasilkan bentuk dan batas2 fungsi yang optimal, kita perlu melatih sistem dengan *training set* yang representatif.
- Untuk menghindari *overfit*, kita bisa menggunakan *validation set*.

Tuning fungsi keanggotaan (statis)



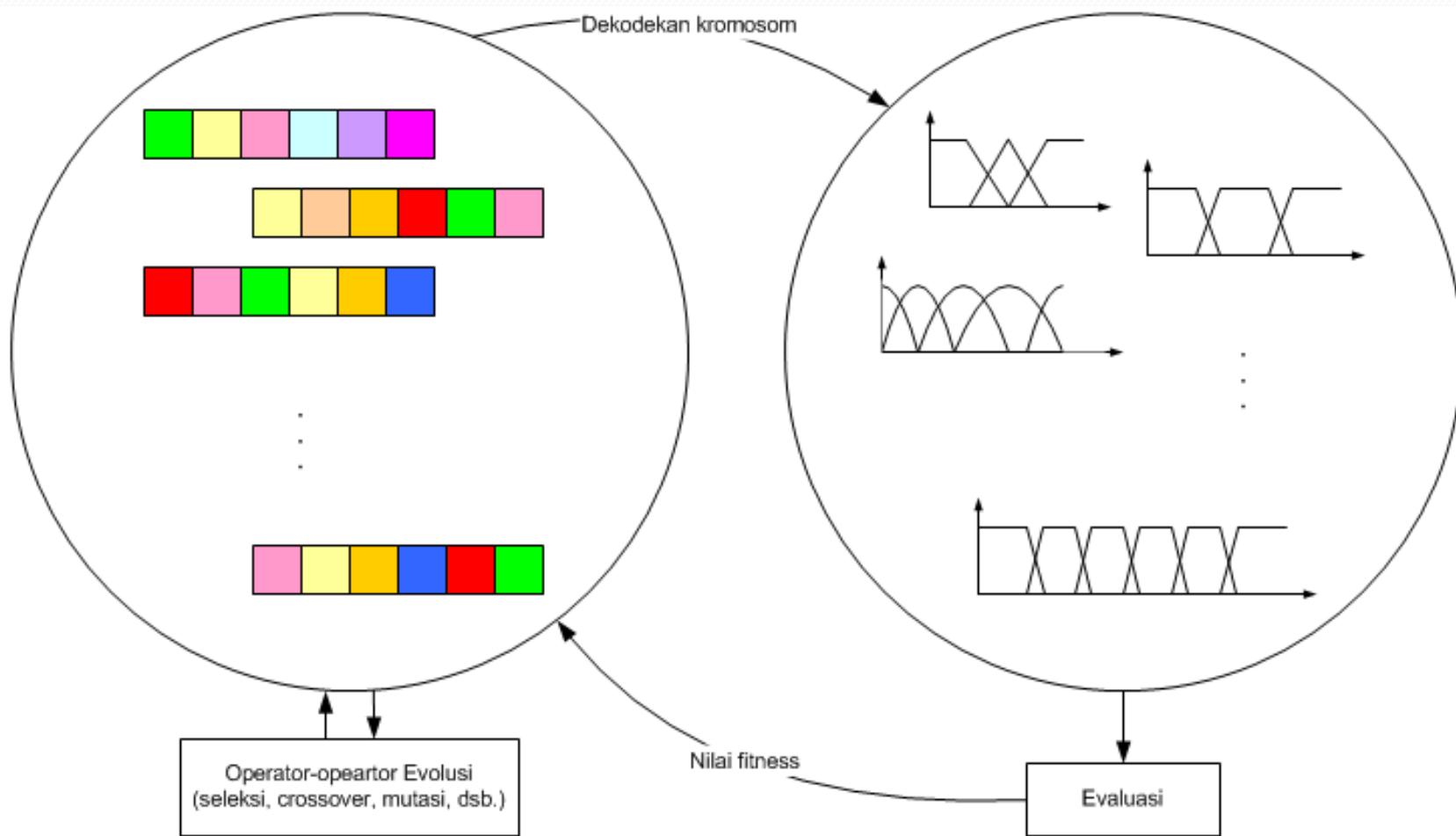
Pengkodean Individu (statis)



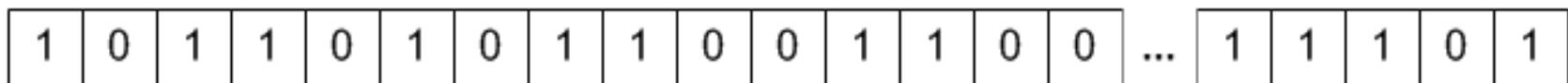
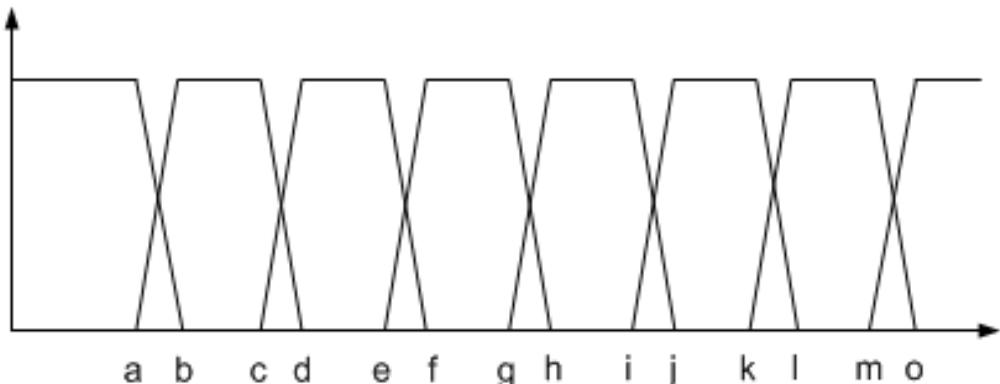
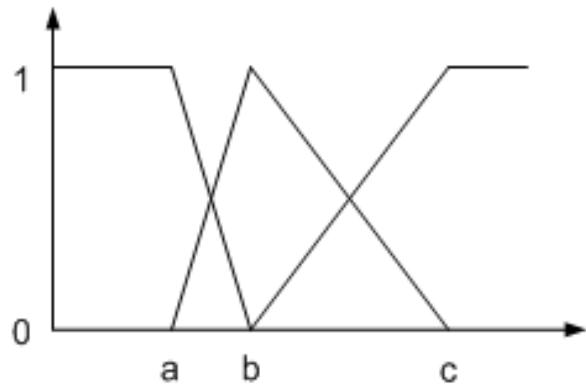
1	0	1	1	0	0	1	1	0	0	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

← a → ← b → ← c →

Tuning fungsi keanggotaan (dinamis)



Pengkodean Individu (dinamis)



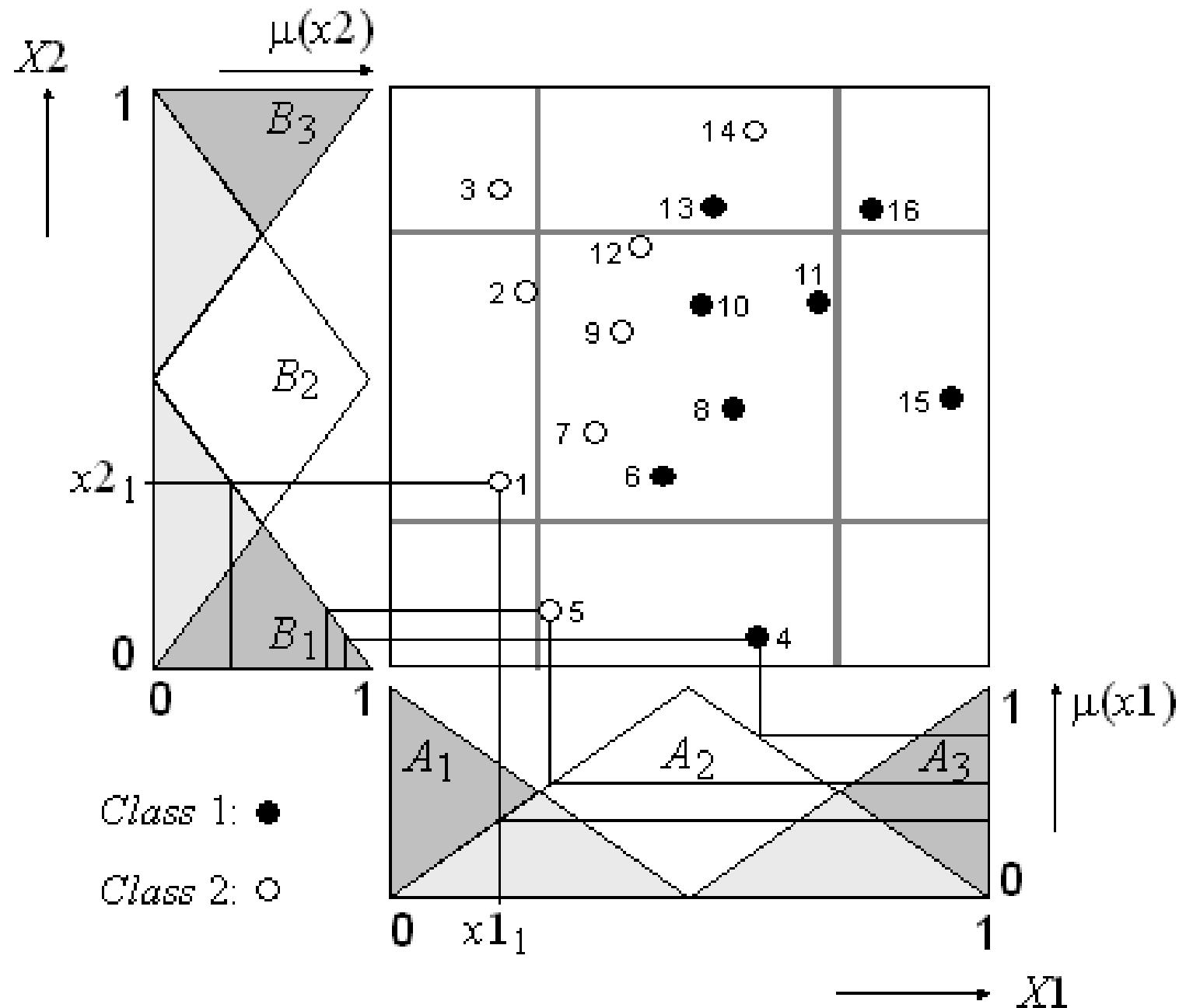
FK: Bentuk Fungsi Keanggotaan (misal: segitiga, trapesium, phi, bell)

JNL: Jumlah Nilai Linguistik (misal maksimum $2^3 = 8$)

a, b, c, ..., o adalah batas2 fungsi keanggotaan

Pemilihan Aturan Fuzzy

- Untuk mendapatkan himpunan aturan fuzzy, kita perlu melatih sistem dengan training set yang representatif dari sekumpulan data-data numerik.
- Untuk menghindari himpunan aturan fuzzy yang overfit, kita bisa menggunakan validation set.



Fuzzy partition

- Titik-titik hitam dan putih menyatakan pola-pola dalam *Training Set* untuk *kelas 1* dan *kelas 2*.
- Kotak-kotak dapat dipandang sebagai **Tabel Aturan**.
- Nilai-nilai linguistik untuk input x_1 (A_1, A_2 and A_3) berada pada garis horizontal, dan nilai-nilai linguistik untuk input x_2 (B_1, B_2 and B_3) berada pada garis vertikal.
- Pada perpotongan baris dan kolom terdapat **rule consequent**.
- Pada Tabel Aturan, setiap sub ruang fuzzy hanya memiliki satu Aturan Fuzzy. Dengan demikian, jumlah aturan yang dapat dibangkitkan dari $K \times K$ grid adalah sebanyak $K \times K$.

Aturan Fuzzy dalam $K \times K$ fuzzy partition dapat direpresentasikan dalam bentuk [NEG02]:

Rule R_{ij} :

IF $x1_p$ is A_i $i = 1, 2, \dots, K$

AND $x2_p$ is B_j $j = 1, 2, \dots, K$

THEN $\mathbf{x}_p \in C_n \left\{ CF_{A_i B_j}^{C_n} \right\}$ $\mathbf{x}_p = (x1_p, x2_p)$, $p = 1, 2, \dots, P$

dimana x_p adalah pola latih pada ruang input $X1 \times X2$, P adalah jumlah pola latih, C_n adalah rule consequent (kelas 1 atau 2), dan $CF_{A_i B_j}$ adalah faktor kepercayaan bahwa suatu pola dalam sub ruang fuzzy $A_i B_j$ termasuk ke dalam kelas C_n .

Penentuan *rule consequent* [NEG02]

Langkah 1: Partisi ruang input ke dalam $K \times K$ sub ruang fuzzy, dan hitung kekuatan masing-masing kelas pada pola-pola latih dalam setiap sub ruang fuzzy.

Kekuatan masing-masing kelas dalam sub ruang fuzzy direpresentasikan oleh jumlah pola-pola latihnya. Makin banyak pola latih, makin kuat kelas tersebut. Artinya, *rule consequent* untuk kelas tersebut menjadi lebih pasti.

Langkah 2: Tentukan *rule consequent* dan *certainty factor* pada setiap sub ruang fuzzy.

Penentuan *rule consequent* [NEG02]

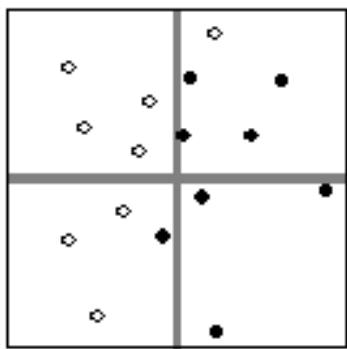
Certainty factor diinterpretasikan sbb:

- Jika semua pola latih dalam sub ruang fuzzy A_iB_j termasuk ke dalam kelas yang sama, maka *certainty factor*-nya adalah maksimum dan setiap pola baru dalam sub ruang tersebut pasti akan termasuk ke dalam kelas ini.
- Jika pola latih termasuk ke dalam kelas2 yang berbeda-beda dan kelas2 tersebut memiliki kekuatan yang sama, maka *certainty factor*-nya adalah minimum dan suatu pola baru akan termasuk ke dalam kelas mana saja. Hal ini berarti bahwa pola2 dalam sub ruang fuzzy dapat diklasifikasikan salah.
- Jika suatu sub ruang fuzzy tidak memiliki pola latih, maka kita tidak dapat menentukan *rule consequent*.
- Jika *fuzzy partition* terlalu kasar, maka banyak pola yang akan diklasifikasikan salah. Sebaliknya, jika *fuzzy partition* terlalu halus, banyak Aturan fuzzy tidak dapat dihasilkan karena kurangnya pola latih dalam masing2 sub ruang fuzzy.

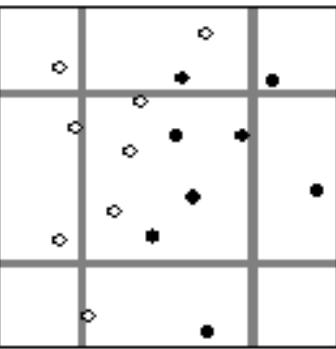
Langkah2 Penentuan *rule consequent* [NEG02]

- Pola-pola latih tidak selalu terdistribusi secara merata pada ruang input.
- Oleh karena itu, seringkali kita kesulitan untuk memilih ukuran *fuzzy grid* yang tepat.
- Untuk mengatasi kesulitan ini, kita bisa menggunakan Tabel Aturan Fuzzy Ganda atau ***multiple fuzzy rule tables***.

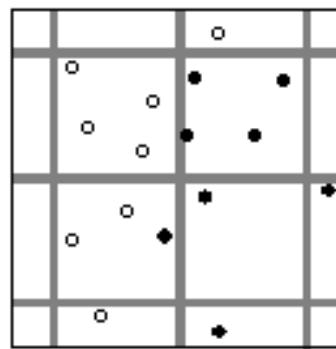
Langkah2 Penentuan *rule consequent* [NEG02]



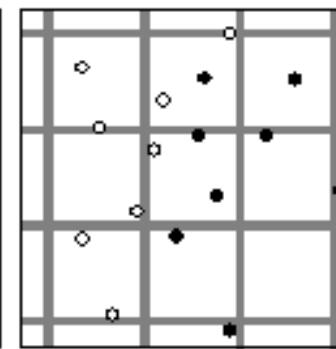
$$K = 2$$



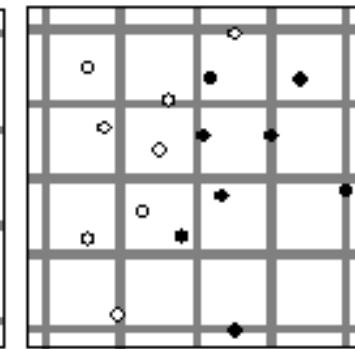
$$K = 3$$



$$K = 4$$



$$K = 5$$



$$K = 6$$

Aturan2 Fuzzy dibangkitkan untuk masing2 sub ruang fuzzy pada *multiple fuzzy rule tables*, sehingga kita mendapatkan suatu himpunan aturan yang lengkap sebanyak:

$$2 \times 2 + 3 \times 3 + 4 \times 4 + 5 \times 5 + 6 \times 6 = 90$$

Langkah2 Penentuan *rule consequent* [NEG02]

Jika himpunan aturan S_{ALL} sudah dibangkitkan, suatu pola baru, $x = (x_1, x_2)$, dapat diklasifikasi menggunakan prosedur sbb:

- **Langkah 1:** Pada masing-masing sub ruang fuzzy pada *multiple fuzzy rule tables*, hitung derajat kompatibilitas pola baru dengan masing2 kelas.
- **Langkah 2:** Tentukan dearajat kompatibilitas maksimum dari pola dengan masing2 kelas.
- **Langkah 3:** Tentukan kelas dimana pola baru memiliki dearajat kompatibilitas tertinggi, dan nyatakan bahwa pola baru adalah kelas tersebut.

Langkah2 Penentuan *rule consequent* [NEG02]

- Jumlah *multiple fuzzy rule tables* yang diperlukan untuk suatu klasifikasi pola yang akurat mungkin sangat banyak
- Akibatnya, himpunan aturan yang dibangkitkan menjadi sangat banyak. Tetapi, aturan2 tersebut memiliki kemampuan klasifikasi yang berbeda2.
- Oleh karena itu, dengan memilih aturan2 yang potensial, kita dapat mengurangi jumlah aturan tetapi tingkat klasifikasi tetap tinggi.
- Dapatkah kita gunakan *Evolutionary Algorithms* (EAs) untuk menyeleksi Aturan Fuzzy tersebut?

EAs untuk menyeleksi Aturan Fuzzy

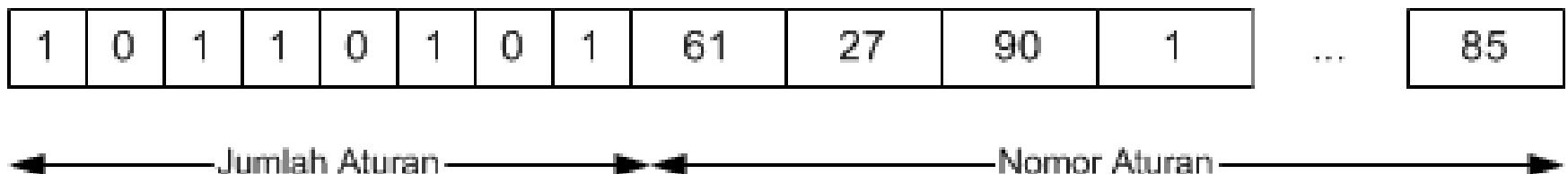
- Masalah pemilihan Aturan Fuzzy dapat dipandang sebagai masalah kombinatorial dengan **dua tujuan**.
- Tujuan pertama (paling penting) adalah memaksimalkan jumlah pola yang diklasifikasi dengan benar.
- Tujuan ke dua adalah meminimalkan jumlah aturan.

EAs untuk menyeleksi Aturan Fuzzy

- Dua masalah paling mendasar dalam penggunaan EAs adalah:
 - Bagaimana bentuk individunya?
 - Bagaimana fungsi *fitness*-nya?
- Jika kedua masalah tersebut berhasil diselesaikan, maka komponen2 EAs yang lain kebanyakan sama dengan EAs untuk menyelesaikan masalah2 lainnya.

Bentuk Individu

- Setiap gen menyatakan aturan2 dalam S_{ALL}
- Jumlah aturan bersifat dinamis
- Terdapat gen yang menyatakan jumlah aturan



Fungsi *Fitness*

Fungsi fitness harus mengakomodasi kedua tujuan tersebut, **memaksimalkan** jumlah pola yang diklasifikasi dengan benar dan **meminimalkan** jumlah aturan.

$$f(S) = w_P \frac{P_S}{P_{ALL}} - w_N \frac{N_S}{N_{ALL}}$$

dimana P_S adalah jumlah pola yang diklasifikasi dengan benar, P_{ALL} adalah jumlah pola latih. N_S dan N_{ALL} secara berturut2 adalah jumlah Aturan Fuzzy dalam himpunan S dan S_{ALL} . Sedangkan w_P dan w_N adalah bobot untuk tujuan pertama dan tujuan ke dua.

Fungsi *Fitness*

Jika memaksimalkan jumlah pola yang diklasifikasi dengan benar **lebih penting 5 kali** dibandingkan dengan meminimalkan jumlah aturan, maka fungsi *fitness*-nya menjadi:

$$f(S) = 5 \frac{P_S}{P_{ALL}} - \frac{N_S}{N_{ALL}}$$

Studi Kasus

- *Sprinkler Control System*
- *Personalized Spam Filtering pada Discovery Challenge di ECML PKDD 2006, the 17th European Conference on Machine Learning and the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*

Sprinkler Control System

Tanggal	Waktu	Suhu (°C)	Kelembaban Tanah (%)	Durasi Penyiraman (menit)
01-02-2006	08:00	20	16	55,7
01-02-2006	13:00	25	12	59,3
01-02-2006	18:00	16	30	5,6
02-02-2006	07:00	15	14	30,1
02-02-2006	12:00	27	10	43,4
02-02-2006	19:00	12	19	18,6
03-02-2006	06:30	19	12	22,1
03-02-2006	10:00	28	8	47,3
03-02-2006	13:30	35	6	76,4
03-02-2006	16:00	20	17	20,9

Sprinkler Control System

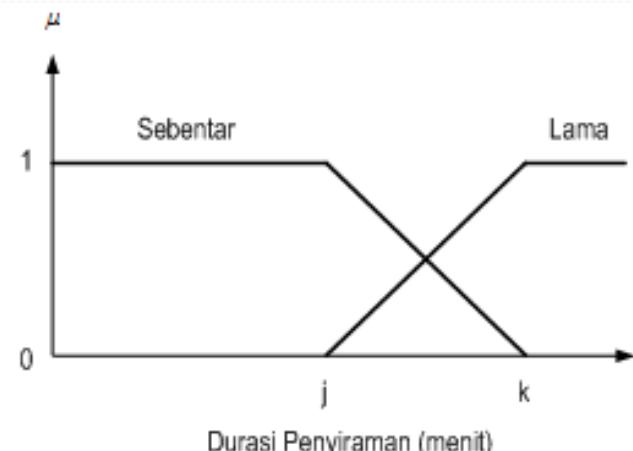
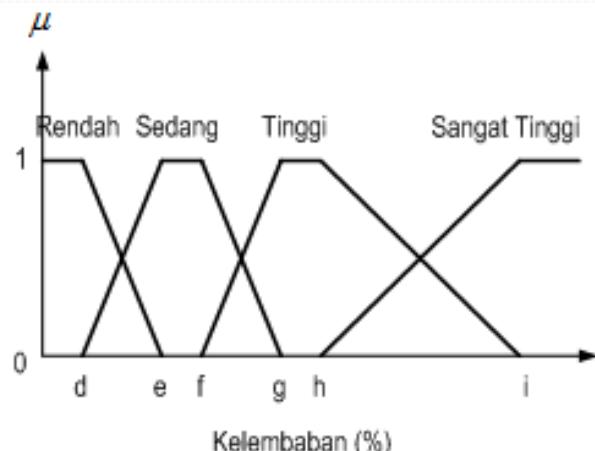
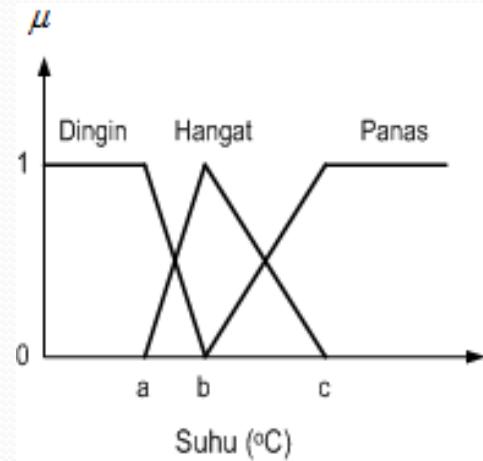
- Bagaimana cara menentukan **Fungsi Keanggotaan** dan/atau **Aturan** untuk *Fuzzy System* menggunakan EAs?
- Buat representasi **Individu**
- Buat **fungsi fitness-nya**
- Jelaskan proses yang terjadi pada EAs

Apa yang di-evolusikan?

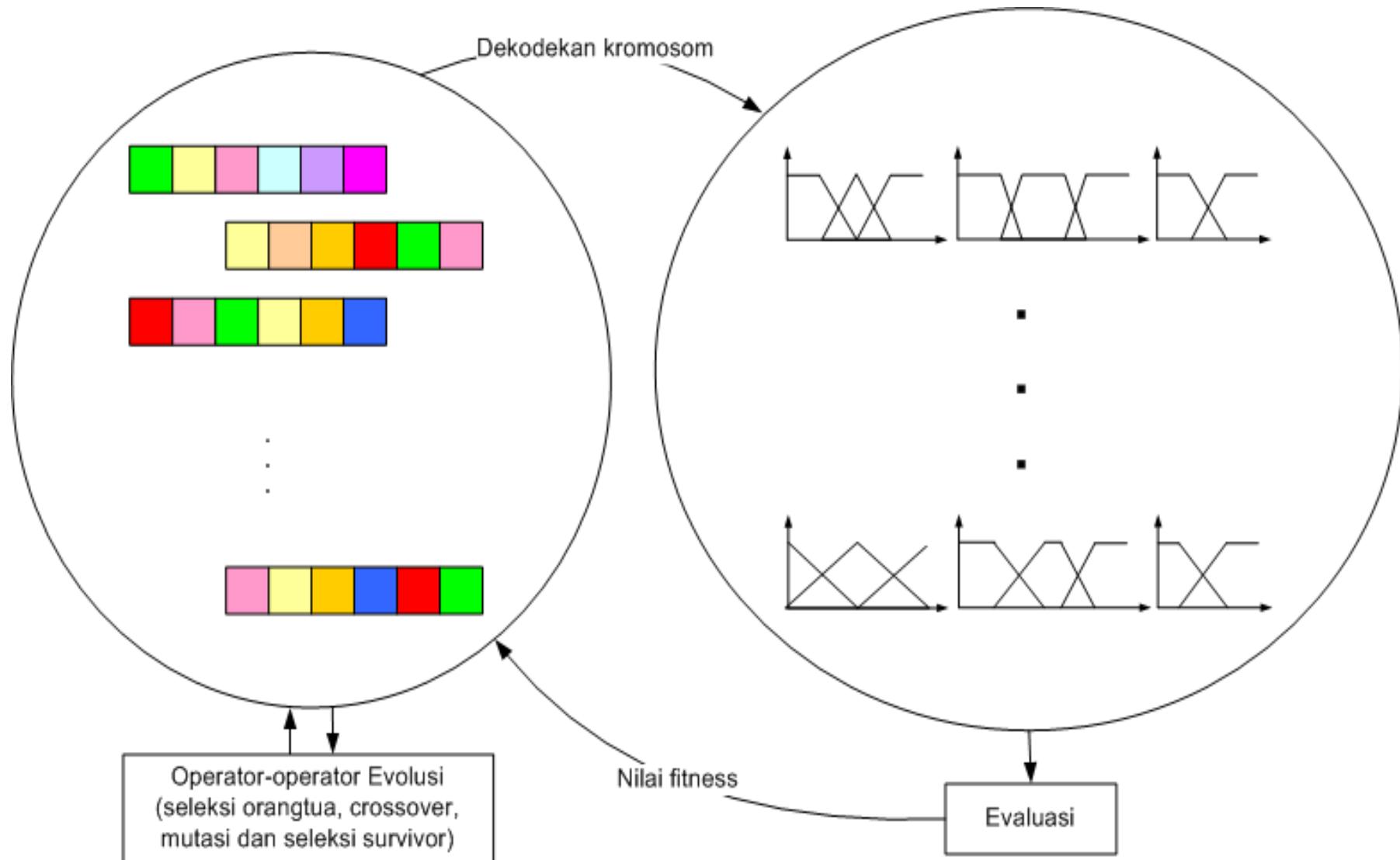
- Batas2 FK
- Jumlah dan Batas2 FK
- Bentuk, Jumlah, dan Batas2 FK
- Bentuk, Jumlah, dan Batas2 FK serta Rule

Evolusi Batas2 FK

Asumsi: Jumlah dan Bentuk FK serta Fuzzy Rule sudah terdefinisi



1,351	2,117	3,015	0,986	1,289	3,201	4,712	5,377	12,078	54,098	80,986
a	b	c	d	e	f	g	h	i	j	k



Evolusi Bentuk, Jumlah, dan batas2 FK serta Rule

- Kromosom sangat panjang
- Ruang solusi sangat besar
- Harus disesuaikan dengan kasus yang dihadapi

Sprinkler Control System

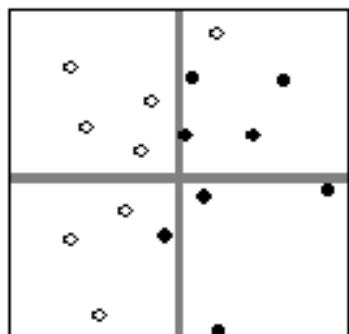
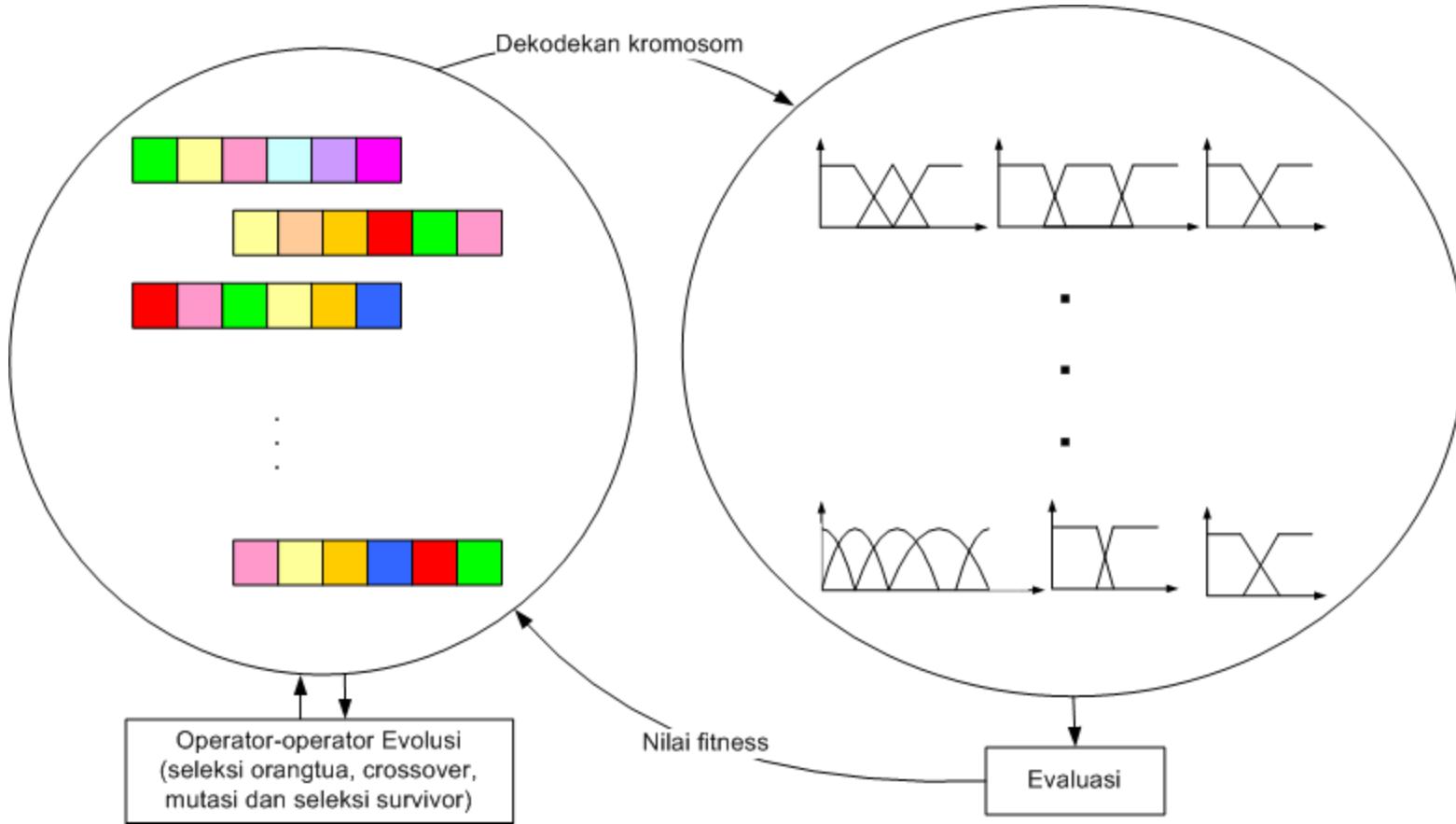
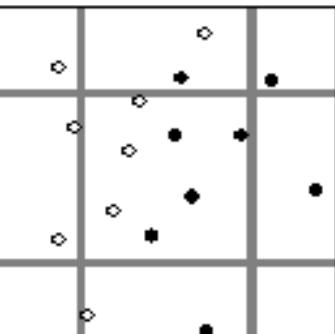
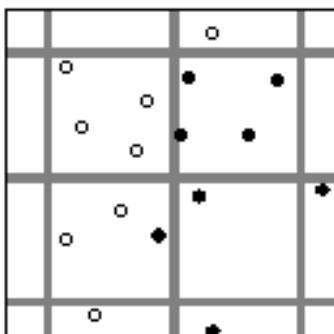
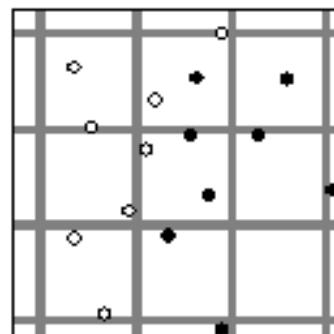
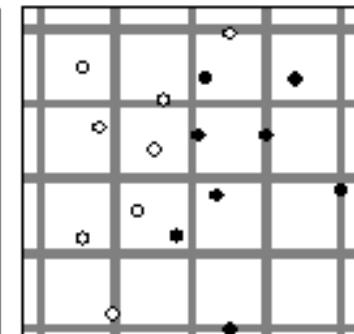
Tanggal	Waktu	Suhu (°C)	Kelembaban Tanah (%)	Durasi Penyiraman (menit)
01-02-2006	08:00	20	16	55,7
01-02-2006	13:00	25	12	59,3
01-02-2006	18:00	16	30	5,6
02-02-2006	07:00	15	14	30,1
02-02-2006	12:00	27	10	43,4
02-02-2006	19:00	12	19	18,6
03-02-2006	06:30	19	12	22,1
03-02-2006	10:00	28	8	47,3
03-02-2006	13:30	35	6	76,4
03-02-2006	16:00	20	17	20,9

Personalized Spam Filtering

Task	File name	Data set size	Description
Task A	task_a_labeled_train.tf	4000 emails	Labeled training emails.
	task_a_u00_eval.tf ... task_a_u02_eval.tf	2500 emails each	Unlabeled evaluation data: 3 inboxes from different users.
	task_a_labeled_tune.tf	4000 emails	Labeled training emails for parameter tuning. Feature representation corresponds only to file task_a_u00_tune.tf.
	task_a_u00_tune.tf	2500 emails	Labeled test emails of one user's inbox for parameter tuning. Feature representation corresponds only to file task_a_labeled_tune.tf.
Task B	task_b_labeled_train.tf	100 emails	Labeled training emails.
	task_b_u00_eval.tf ... task_b_u14_eval.tf	400 emails each	Unlabeled evaluation data: 15 inboxes from different users.
	task_b_labeled_tune.tf	100 emails	Labeled training emails for parameter tuning. Feature representation corresponds only to files task_b_u00_tune.tf and task_b_u01_tune.tf.
	task_b_u00_tune.tf task_b_u01_tune.tf	400 emails each	Labeled test emails from two user's inboxes for parameter tuning. Feature representation corresponds only to file task_b_labeled_tune.tf.

Apa yang di-evolusikan?

- Batas2 FK
- Jumlah dan Batas2 FK
- Bentuk, Jumlah, dan Batas2 FK
- Bentuk, Jumlah, dan Batas2 FK serta Rule


 $K = 2$

 $K = 3$

 $K = 4$

 $K = 5$

 $K = 6$

Personalized Spam Filtering

- Bagaimana cara menentukan Fungsi Keanggotaan dan/atau Aturan untuk *Fuzzy System* menggunakan EAs?
- Buat representasi Individu
- Buat fungsi fitness-nya
- Jelaskan proses yang terjadi pada EAs. Apakah perlu dilakukan *preprocessing* data latih?

Referensi

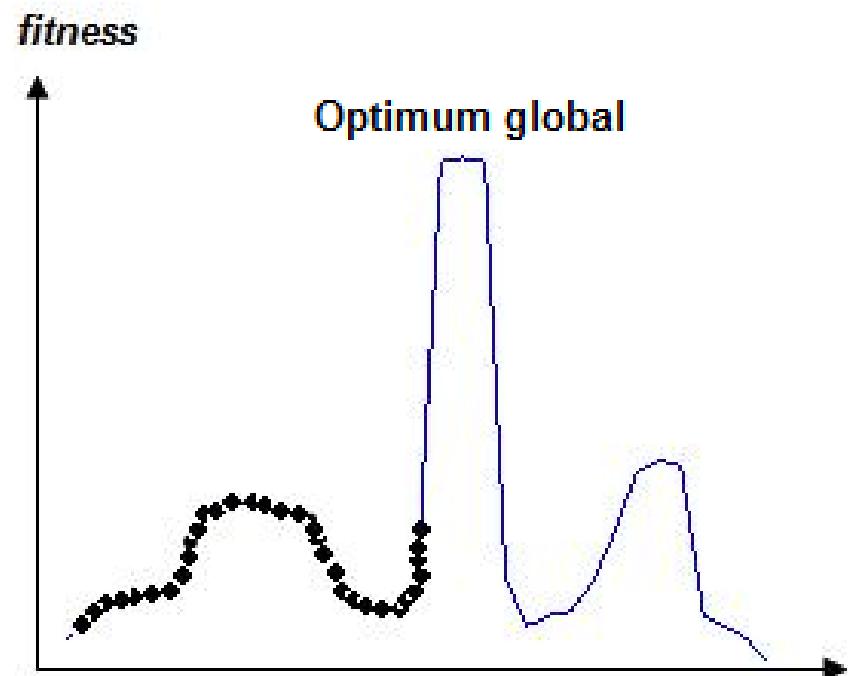
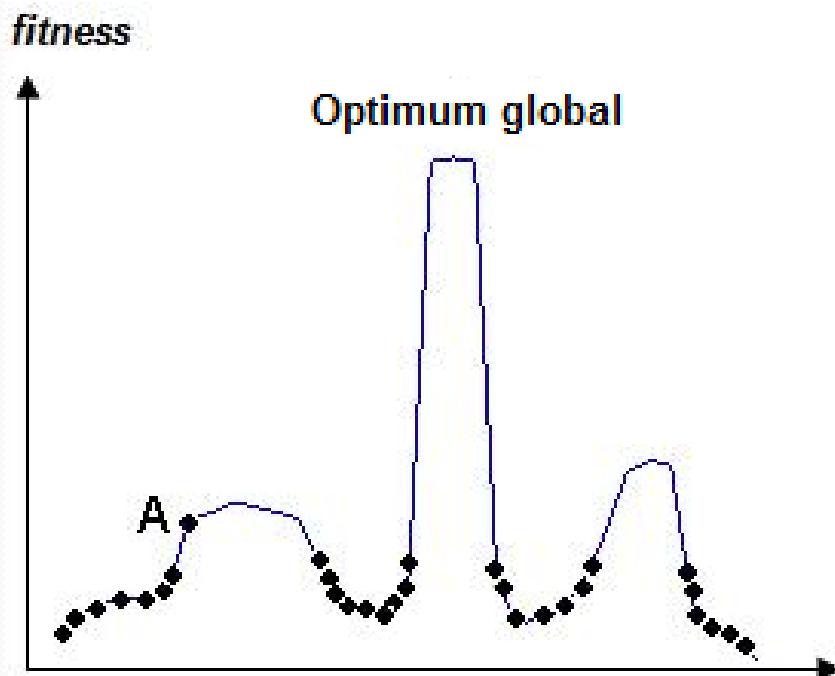
- [NEG02] Negnevitsky, 2002, Hybrid intelligent systems: Evolutionary neural networks and fuzzy evolutionary systems, Pearson Education.
- [SUY08] Suyanto, 2008, "Soft Computing: Membangun Mesin Ber-IQ Tinggi", Informatika, Bandung Indonesia. ISBN: 978-979-1153-49-2.
- [TET01] Tettamanzi A., Tomassini 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.

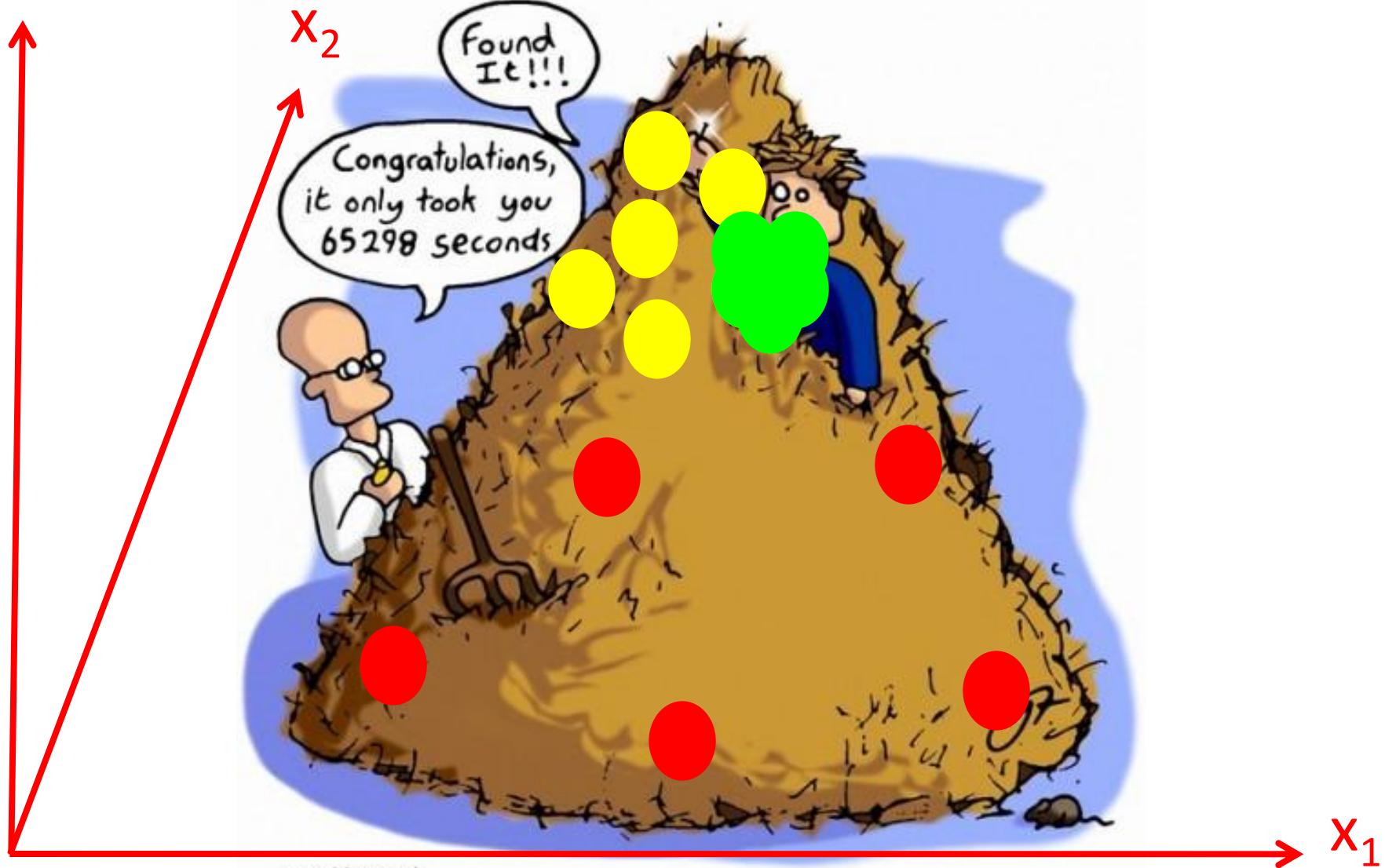
Fuzzy EAs

Dr. Suyanto, S.T., M.Sc.
HP/WA: 0812 845 12345

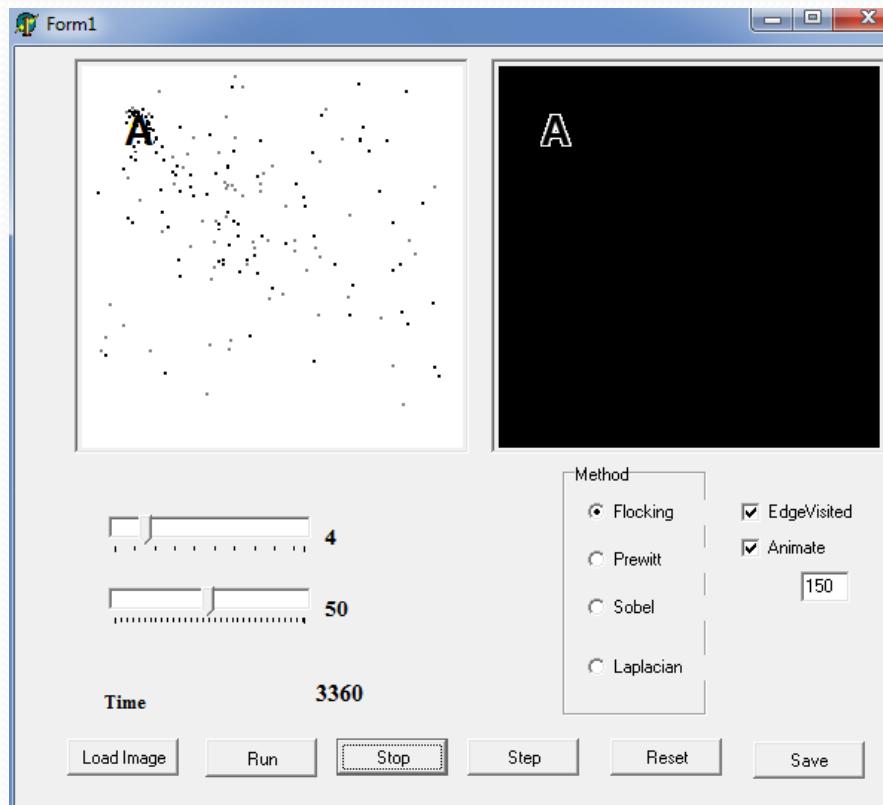
Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

Permasalahan Pada EAs



$f(x_1, x_2)$ 

Edge Detection

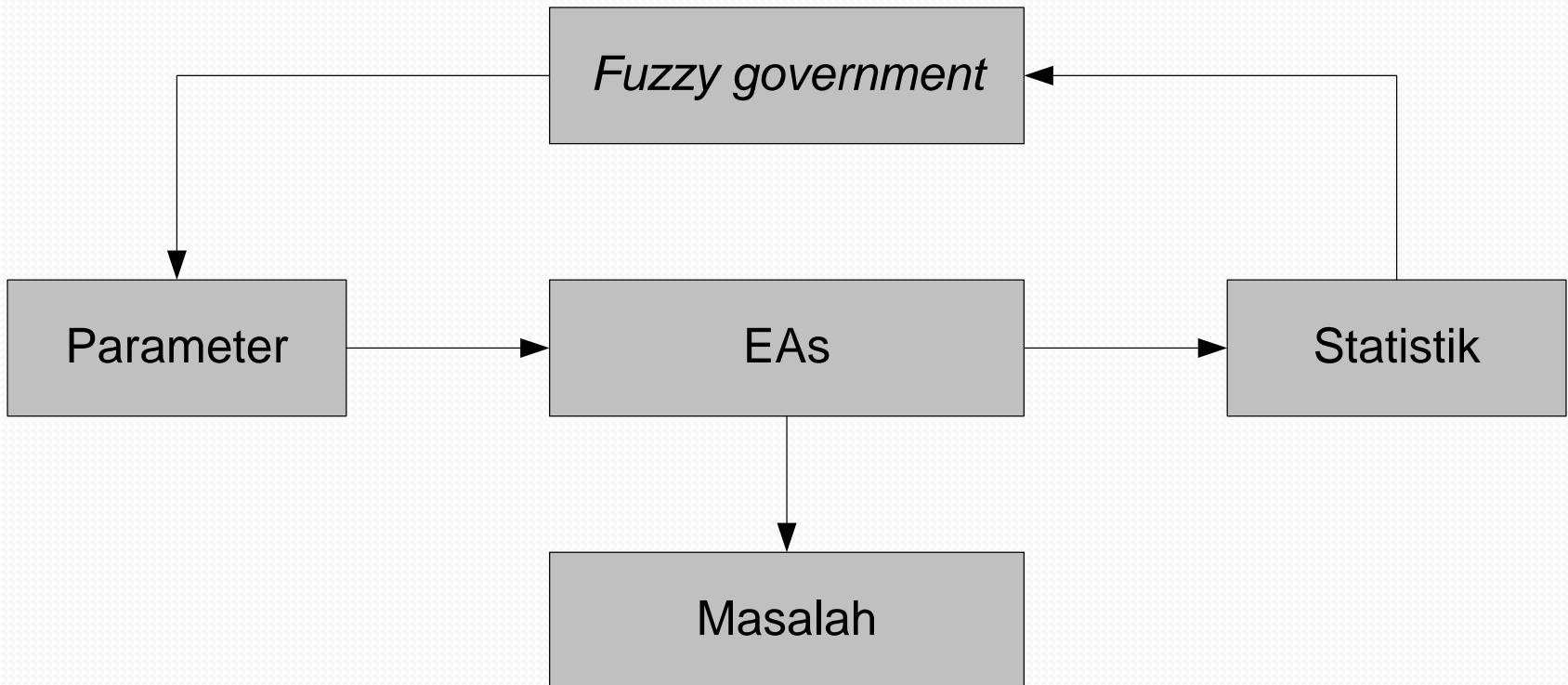


[Addino Yudi Abdal - 113990156 - IMPLEMENTASI PROSES
PENDETEKSIAN SISI DENGAN TEKNIK FLOCKING]

Fuzzy untuk EAs

- *Fuzzy Government* → *Adaptive EAs*
- Memasukkan fuzziness ke dalam EAs
 - *fuzzy fitness*
 - *fuzzy allele*
 - *fuzzy gen*

Adaptive EAs



Adaptive EAs

- *Statistics* diperoleh dari EA pada *sampling rate* tertentu (misal r generasi).
- *Fuzzy Government* adalah kumpulan *fuzzy rules* dan routines yang berfungsi untuk:
 - Mengontrol proses evolusi
 - Mendekksi kemunculan solusi
 - Tuning parameter EA pada saat *running*
 - Mencegah konvergensi prematur

EA Statistics

- ***Genotype statistic***

Kesimpulan atas aspek-aspek yang berhubungan dengan *genotypes* dari individu-individu dalam suatu populasi

- ***Phenotype statistic***

Fokus pada performansi individu (nilai *fitness*) untuk masalah yang dihadapi

Genotype statistic

Yang umum digunakan adalah *diversity measure* yang dihitung berdasarkan *(fuzzy) similarity measure*:

$$\mu_{similar}(A, B) = \frac{1}{1 + (A - B)^2}$$

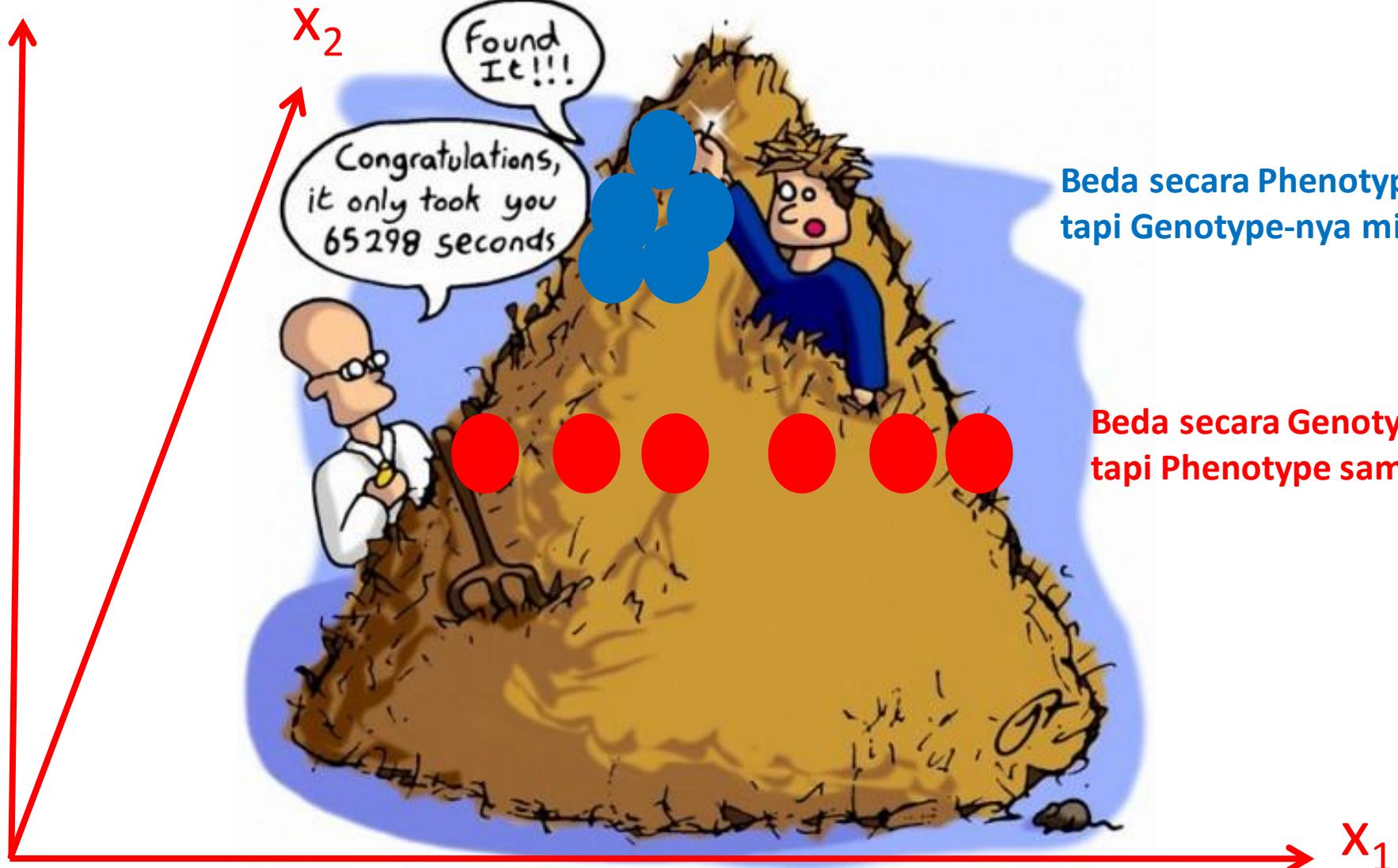
Phenotype statistic

- Fitness range
- Rasio fitness terbaik terhadap fitness rata-rata
- Variansi fitness

Genotype vs Phenotype statistic

- Kenapa harus dipakai keduanya?
- *Genotype* saja atau *phenotype* saja tidak cukup?

$f(x_1, x_2)$



Beda secara Phenotype,
tapi Genotype-nya mirip

Beda secara Genotype,
tapi Phenotype sama

Fuzzy government

- Digunakan untuk mengatasi dua masalah EA:
 - Konvergensi yang sangat lambat
 - Konvergensi prematur
- Kedua masalah EA tersebut terjadi karena:
 - Pemilihan parameter yang kurang tepat pada saat inisialisasi.
 - Parameter EA tidak berubah selama evolusi, padahal kondisi evolusi bisa berubah-ubah.
 - Interaksi antar parameter-parameter yang berbeda adalah kompleks dan sulit dipahami

Pengontrolan parameter EA

- Output dari fuzzy government bisa langsung digunakan untuk mengontrol parameter EA
- Pengontrolan parameter EA bisa dilakukan pada:
 - Ukuran populasi
 - Probabilitas crossover
 - Probabilitas Mutasi
 - *Selective pressure*

Pengontrolan Probabilitas Crossover [Xu & et al]

p_c	Ukuran populasi		
Generasi	Kecil	Sedang	Besar
Singkat	Sedang	Kecil	Kecil
Sedang	Besar	Besar	Sedang
Lama	Sangat Besar	Sangat Besar	Besar

Pengontrolan Probabilitas Mutasi [Xu & et al]

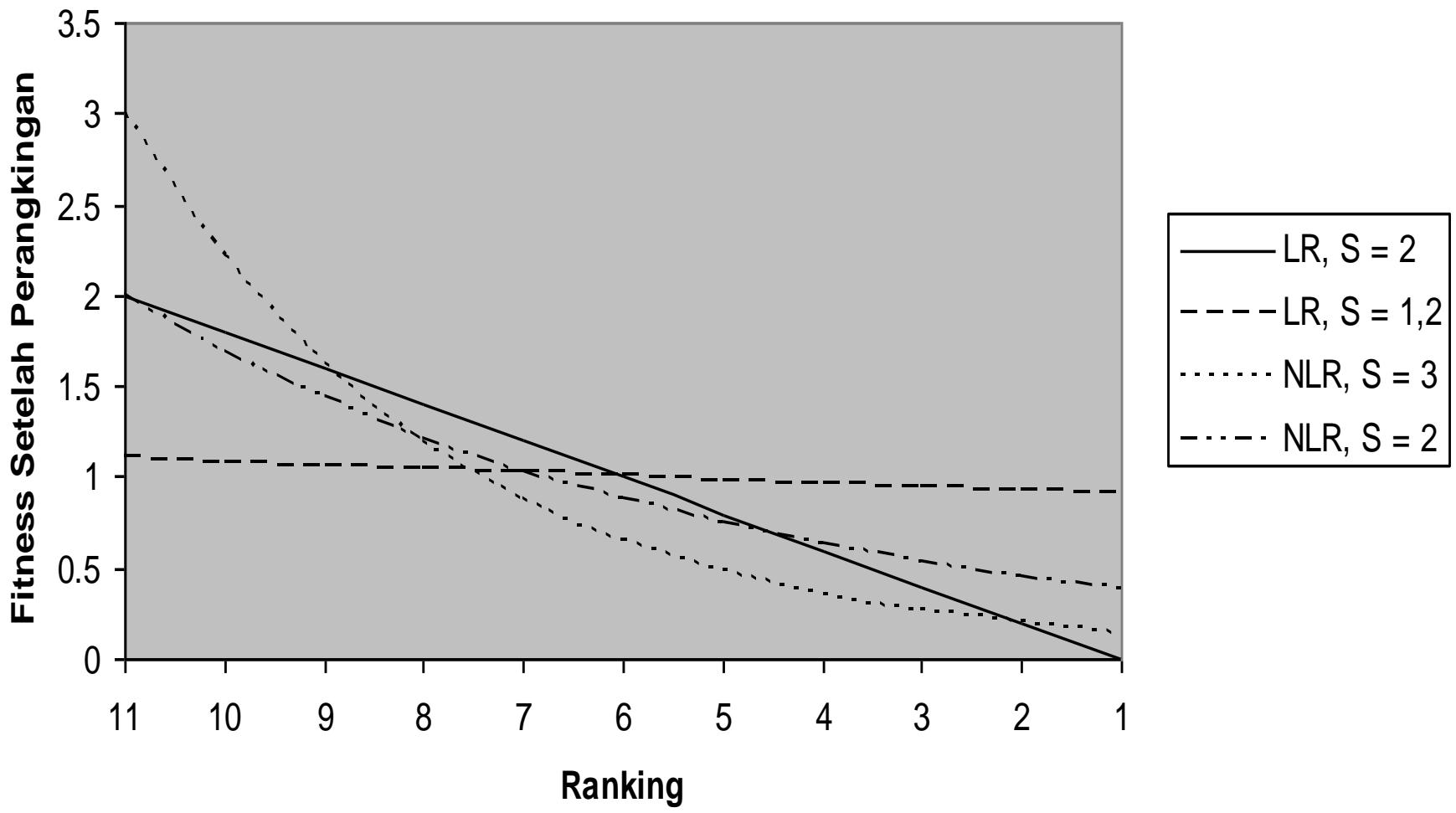
P_m	Ukuran populasi		
Generasi	Kecil	Sedang	Besar
Singkat	Besar	Sedang	Kecil
Sedang	Sedang	Kecil	Sangat Kecil
Lama	Kecil	Sangat Kecil	Sangat Kecil

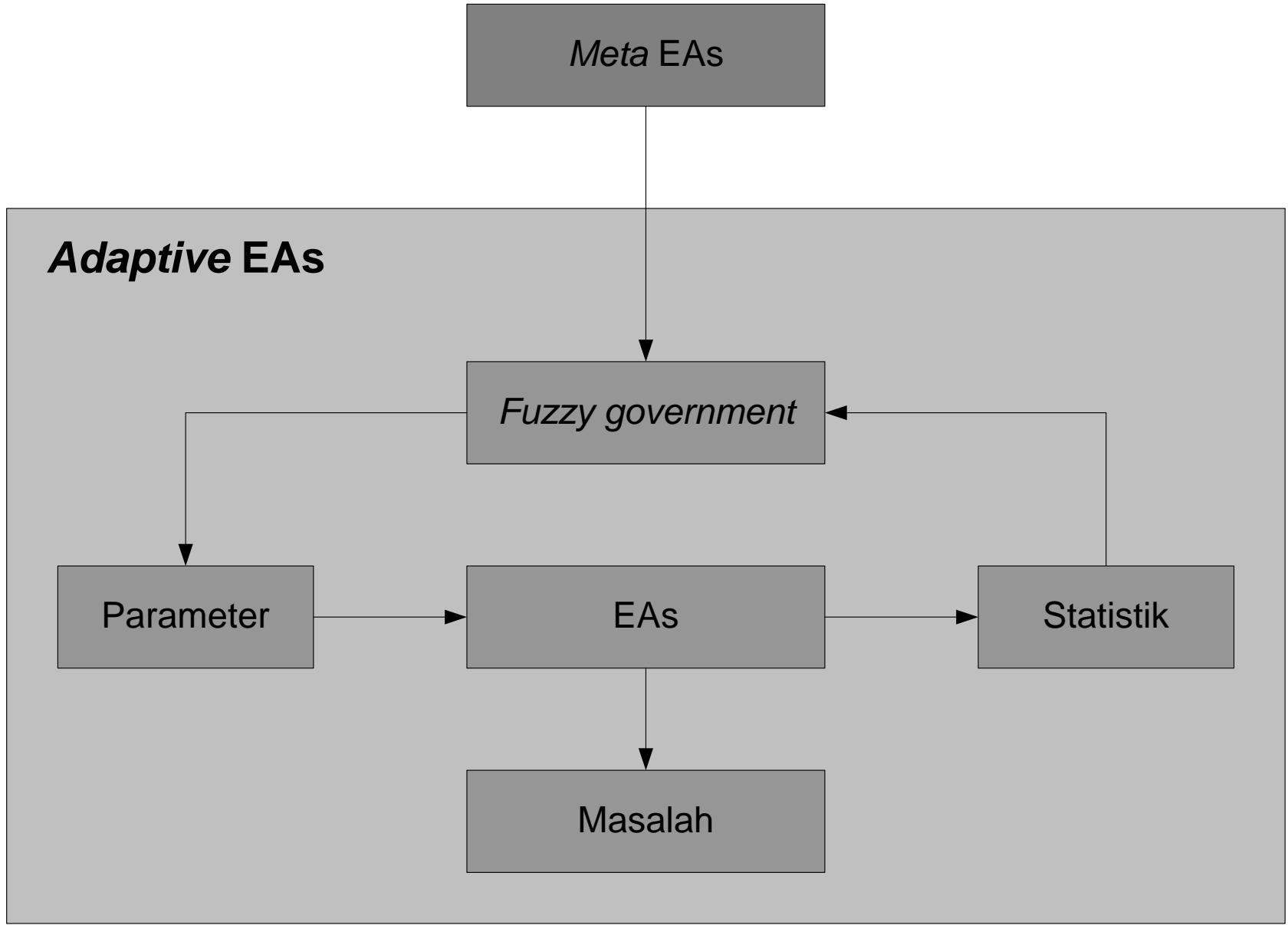
Pengontrolan *exploitation-oriented crossover rate* [Xu & et al]

Δp_e	<i>Phenotype diversity</i>		
<i>Genotype diversity</i>	Rendah	Sedang	Tinggi
Rendah	Sedang	Kecil	Kecil
Sedang	Besar	Besar	Sedang
Tinggi	Besar	Besar	Sedang

Pengontrolan *selective pressure* [Xu & et al]

$\Delta\eta_{\min}$	<i>Phenotype diversity</i>		
<i>Genotype diversity</i>	Rendah	Sedang	Tinggi
Rendah	Kecil	Sedang	Besar
Sedang	Kecil	Besar	Besar
Tinggi	Kecil	Kecil	Besar



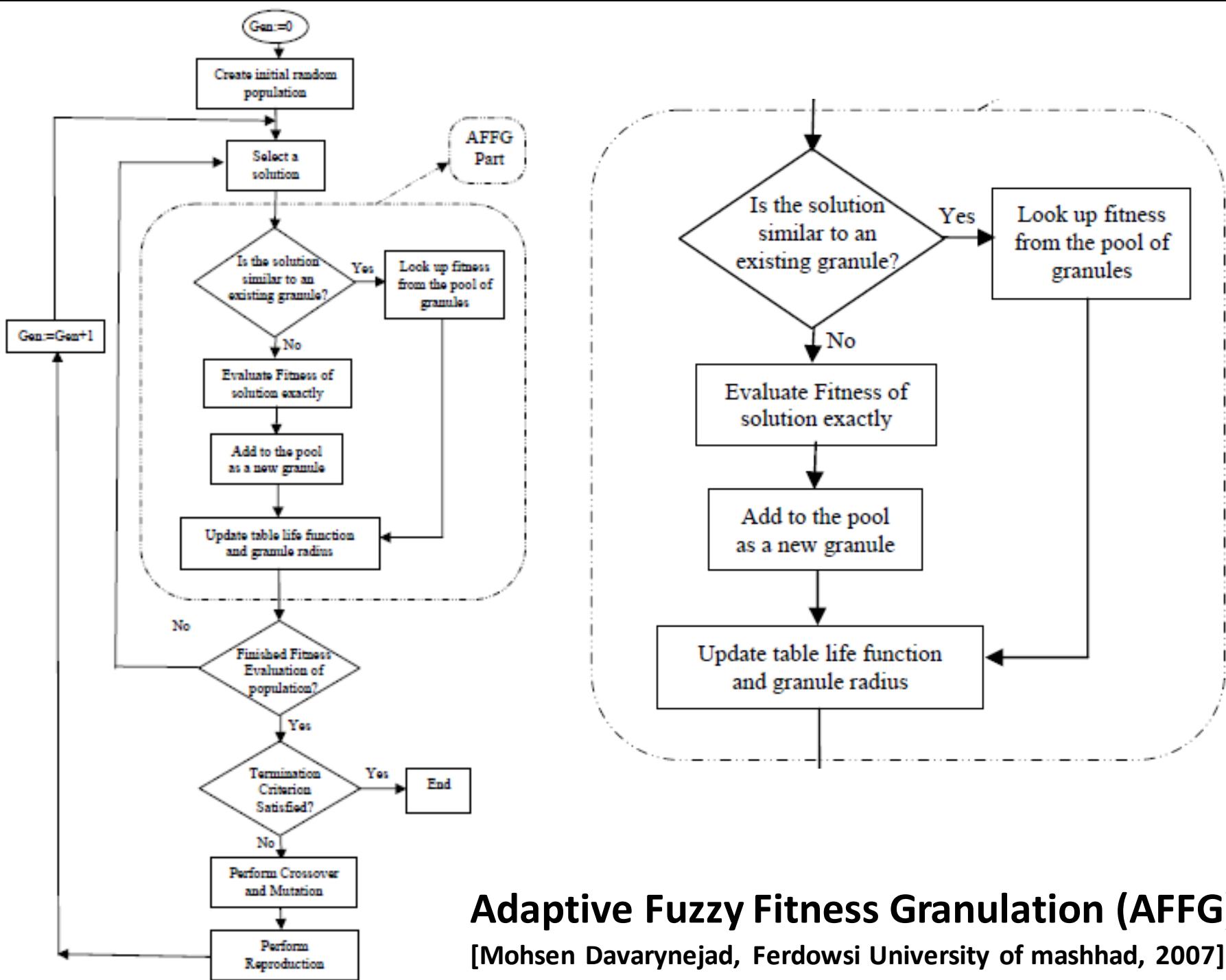


EAs Dengan Komponen-Komponen Fuzzy

- *Fuzzy fitness*
- Rekombinasi berbasis *fuzzy connectives*
- *Soft genetic operators*
- Rekombinasi menggunakan *Templates*

Fuzzy fitness

- Apakah dengan menggunakan *fitness* berpresisi tinggi akan membuat performansi EAs meningkat?
- Dalam banyak kasus, jawabannya tidak.
- Suatu pendekatan yang masuk akal untuk masalah di atas adalah menangani *fitness* secara impresi.
- Penanganan bisa dilakukan menggunakan perangkat-perangkat pada teori himpunan *fuzzy*.
- Adaptive Fuzzy Fitness Granulation (AFFG)



Rekombinasi berbasis *fuzzy connectives*

- Bagi ruang *allele* (nilai gen) ke dalam tiga interval.
- Interval tengah sebagai *exploitation interval* dan interval lainnya sebagai *exploration interval*.

$$L(x, y) \leq \min\{x, y\},$$

$$\min\{x, y\} \leq M(x, y) \leq \max\{x, y\},$$

$$\max\{x, y\} \leq R(x, y).$$

Rekombinasi berbasis *fuzzy connectives*

- Dengan tiga fungsi tersebut, rekombinasi dapat dilakukan dengan tiga cara, yaitu:
 - L-recombination (rekombinasi pada interval kiri $L(x,y)$)
 - M-recombination (rekombinasi pada interval tengah $M(x,y)$)
 - R-recombination (rekombinasi pada interval kanan $R(x,y)$)
- Penggabungan tiga cara rekombinasi tersebut dengan probabilitas yang berbeda-beda menghasilkan kontrol yang baik untuk eksplorasi/eksplorasi.

Soft genetic operators

- Hans-Michael Voigt et. al. menggunakan dua operator (rekombinasi & mutasi) yang bersifat *soft modal*.
- Kedua operator tersebut berdasarkan **distribusi probabilitas segitiga**

Soft Modal Recombination

- Misalkan, ada dua kromosom (x_1, \dots, x_n) dan (y_1, \dots, y_n) .
- Probabilitas bahwa *offspring* memiliki nilai z_i dimana $i = 1, \dots, N$, adalah mengikuti suatu distribusi *bimodal*

$$p(z_i) \in \{\phi(x_i), \phi(y_i)\}$$

dimana $\phi(r)$ adalah distribusi probabilitas segitiga dengan inti r yang didefinisikan dalam interval

$$[r - d|y_i - x_i|, r + d|y_i - x_i|]$$

Soft Modal Recombination

$$\phi(z) = \begin{cases} \frac{z - r + d|y_i - x_i|}{d^2(y_i - x_i)^2}, & z \leq r \\ \frac{r + d|y_i - x_i| - z}{d^2(y_i - x_i)^2}, & z > r \end{cases}$$

$$d \geq 0,5$$

Soft Modal Mutation

- Misalkan, suatu *allele* x bernilai *real* untuk suatu gen didefinisikan dalam interval $[a, b]$.
- *Soft modal mutation* membangkitkan suatu *allele*, dimana nilainya dipilih secara random dari suatu distribusi

$$\{\phi(\pm A\beta^\pi), \phi(\pm A\beta^{\pi+1}), \dots, \phi(\pm A\beta^0)\}$$

- $A << b - a$ merupakan amplitudo mutasi.
- $\pi = \lfloor \log_\beta R_{\min} \rfloor < 0$ dengan $\beta > 1$ disebut sebagai basis mutasi.
- R_{\min} adalah batas bawah dari perubahan mutasi relatif.

Soft Modal Mutation

$$\phi(z) = \begin{cases} \frac{z - r + d|y_i - x_i|}{d^2(y_i - x_i)^2}, & z \leq r \\ \frac{r + d|y_i - x_i| - z}{d^2(y_i - x_i)^2}, & z > r \end{cases}$$

Rekombinasi Menggunakan Templates

- Kita bisa merepresentasikan *genotypes* yang berisi untaian bilangan real dalam interval $[0, 1]$ yang mengkodekan himpunan *fuzzy*.
- Dengan demikian, kita bisa mendefinisikan operator rekombinasi menggunakan *template*.

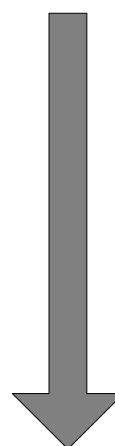
Kromosom g

1	2		5	6	7	8
1	1	0	0	1	1	1

$S_g = \{1,2,5,6,7,8\}$

Kromosom k

1	0	0	1	0	1	1	0
1			4		6	7	



$$T = \{4, 5, 6, 7, 8\}$$

$$\bar{T} = \{1, 2, 3\}$$

$$S_{o1} = (S_g \cap \bar{T}) \cup (S_k \cap T),$$

$$S_{o2} = (S_g \cap T) \cup (S_k \cap \bar{T}),$$

Offspring 1

1	2		4		6	7
1	1	0	1	0	1	1

$S_{o1} = \{1, 2, 4, 6, 7\}$

Offspring 2

1	0	0	0	1	1	1	1
1			5		6	7	8

$S_{o2} = \{1, 5, 6, 7, 8\}$

Kromosom g

0,2	0,1	0,9	0,5	0,8	0,1	0,3	0,4
-----	-----	-----	-----	-----	-----	-----	-----

Kromosom k

0,1	0,7	0,3	0,2	0,1	0,1	0,8	0,9
-----	-----	-----	-----	-----	-----	-----	-----

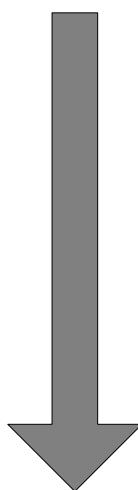
1 5 6 7 8

$$T = \{0, 0, 0, 0, 0.5, 1, 1, 0.5\}$$

$$\bar{T} = \{0.5, 1, 1, 0.5, 0, 0, 0, 0\}$$

$$S_{o1} = \max \left(\min(S_g, \bar{T}), \min(S_k, T) \right)$$

$$S_{o2} = \max \left(\min(S_g, T), \min(S_k, \bar{T}) \right),$$



Offspring 1

0,2	0,1	0,9	0,5	0,1	0,1	0,8	0,4
-----	-----	-----	-----	-----	-----	-----	-----

$$S_{o1} = \{g1, g2, g3, g4, k5, k6, k7, g8\}$$

Offspring 2

0,1	0,7	0,3	0,2	0,8	0,1	0,3	0,4
-----	-----	-----	-----	-----	-----	-----	-----

$$S_{o1} = \{k1, k2, k3, k4, g5, g6, g7, g8\}$$

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.

Studi Kasus

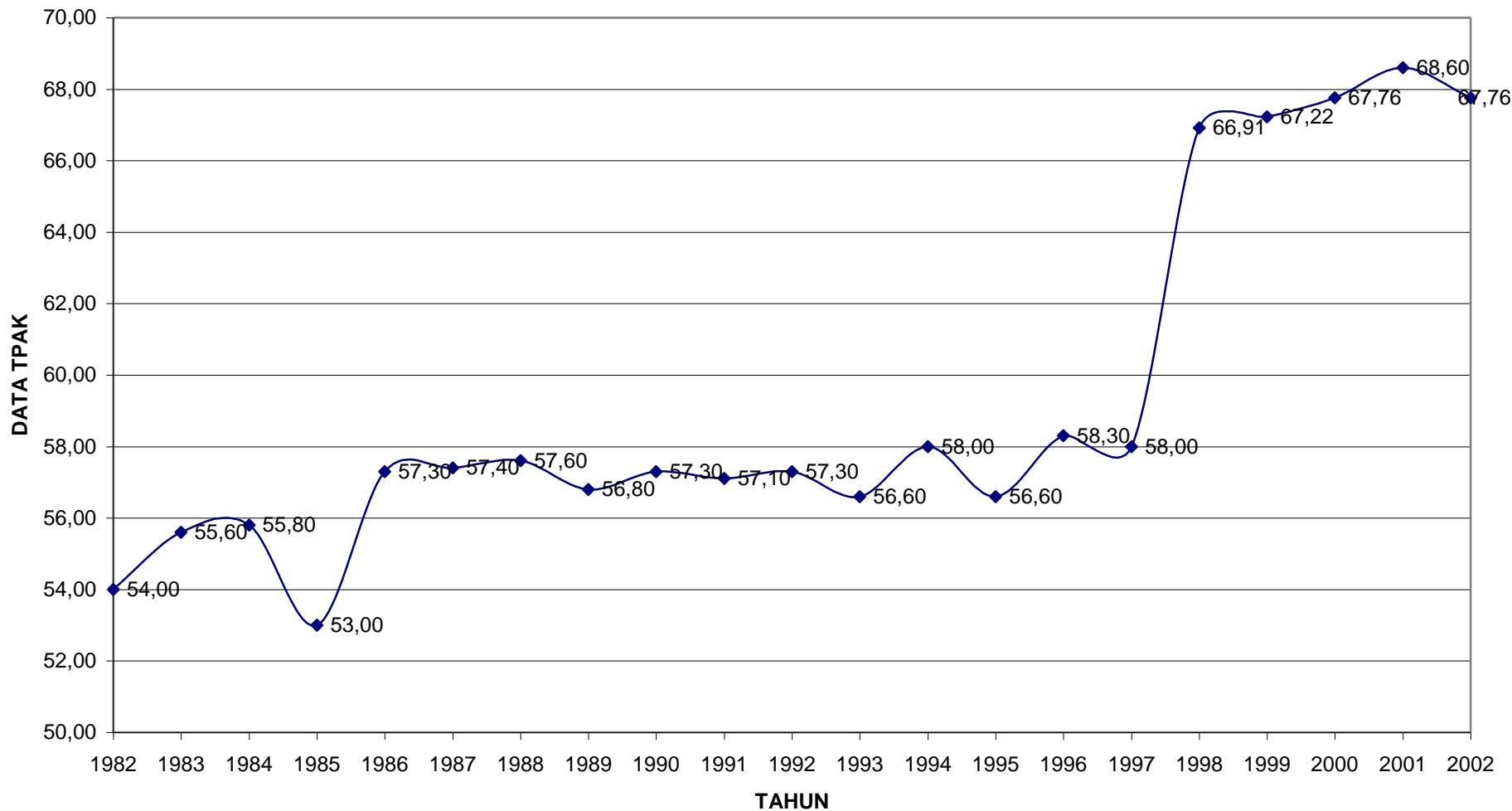
Dr. Suyanto, S.T., M.Sc.
HP/WA: 0812 845 12345

Intelligence Computing Multimedia (ICM)
Informatics faculty – Telkom University

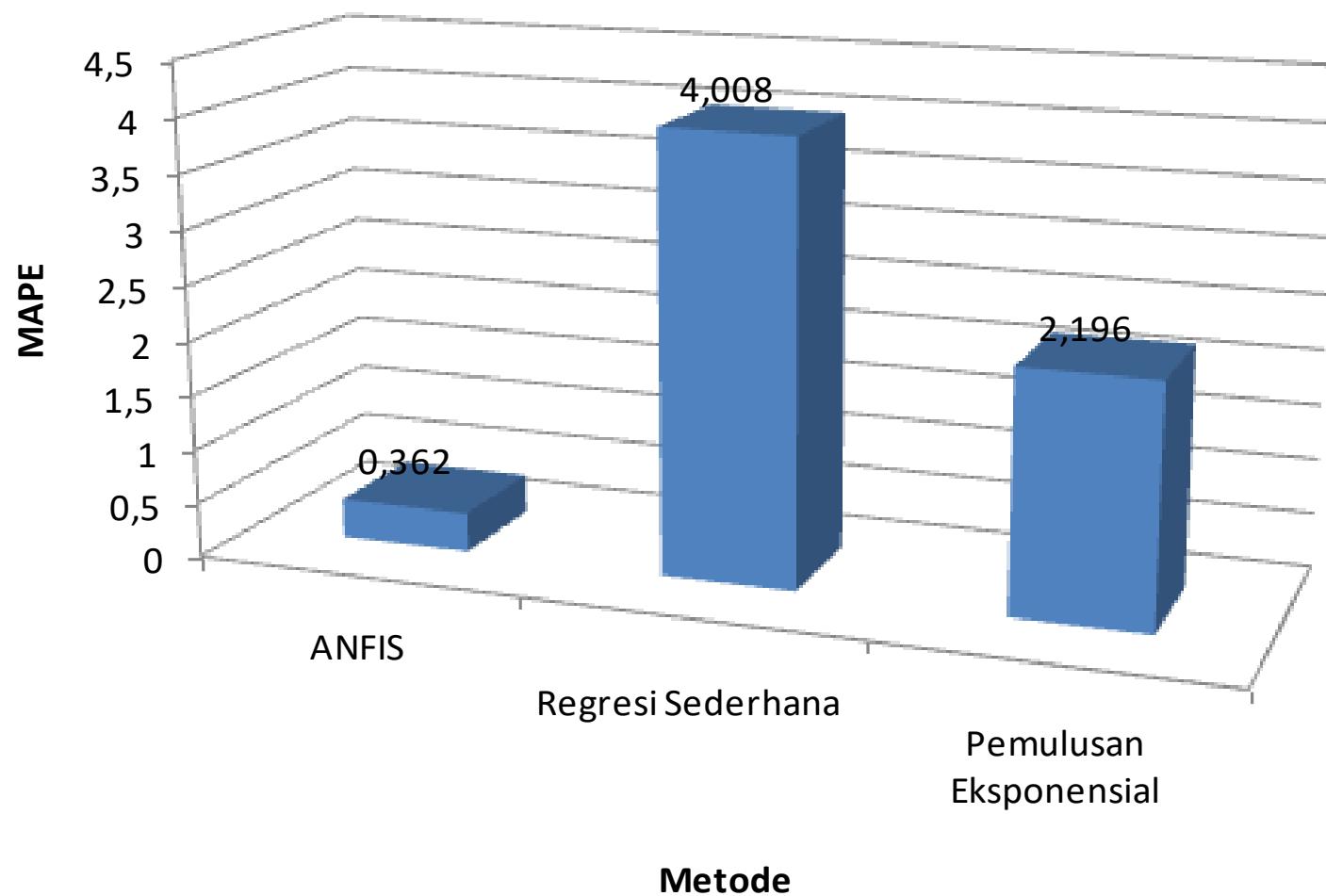
Peramalan Tingkat Partisipasi Angkatan Kerja

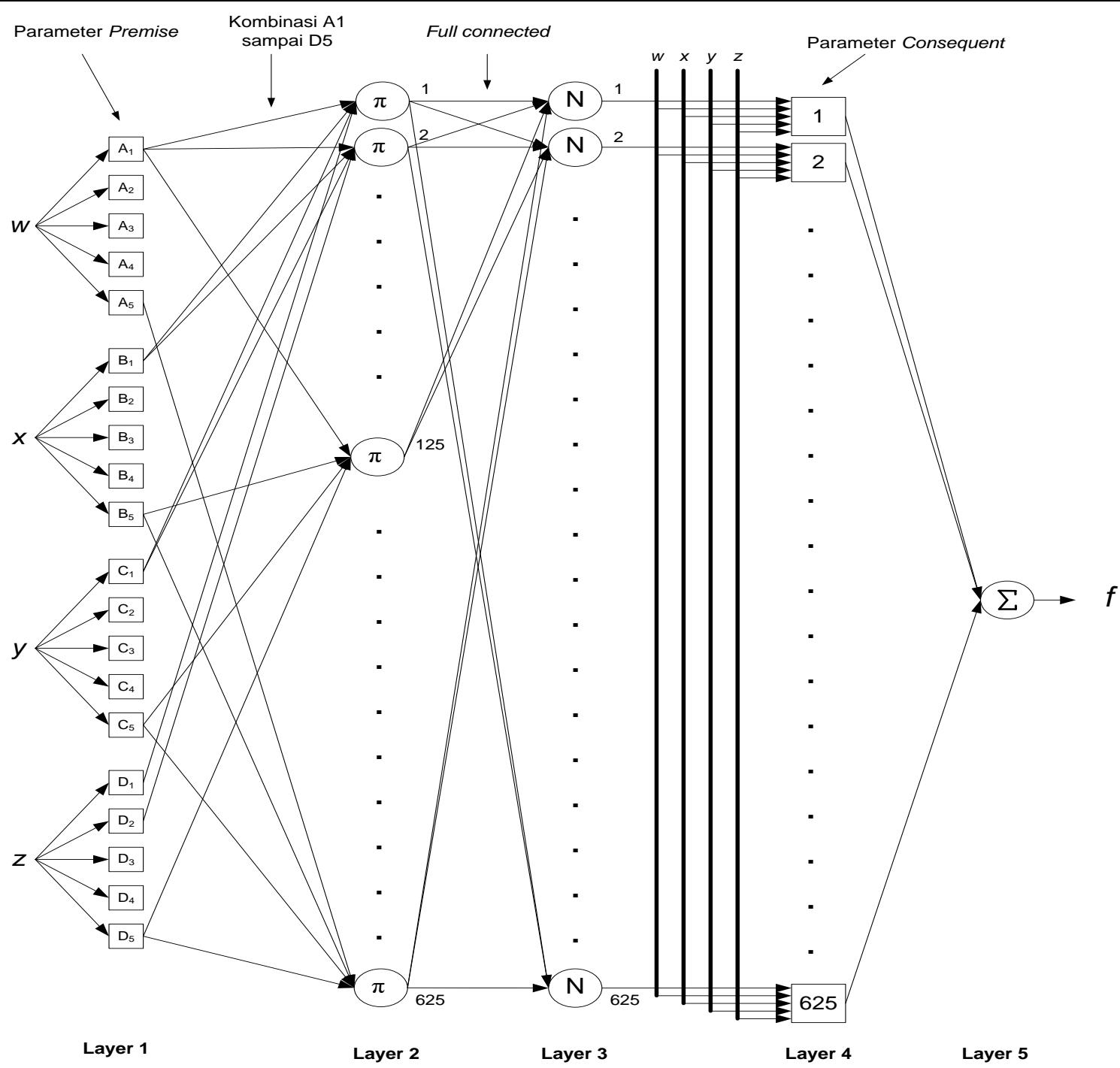
- Peramalan Data Time Series
- ANFIS

GRAFIK POLA DATA TPAK NASIONAL PERIODE 1982-2002



MAPE untuk ANFIS, Regresi Sederhana, dan Pemulusan Eksponensial



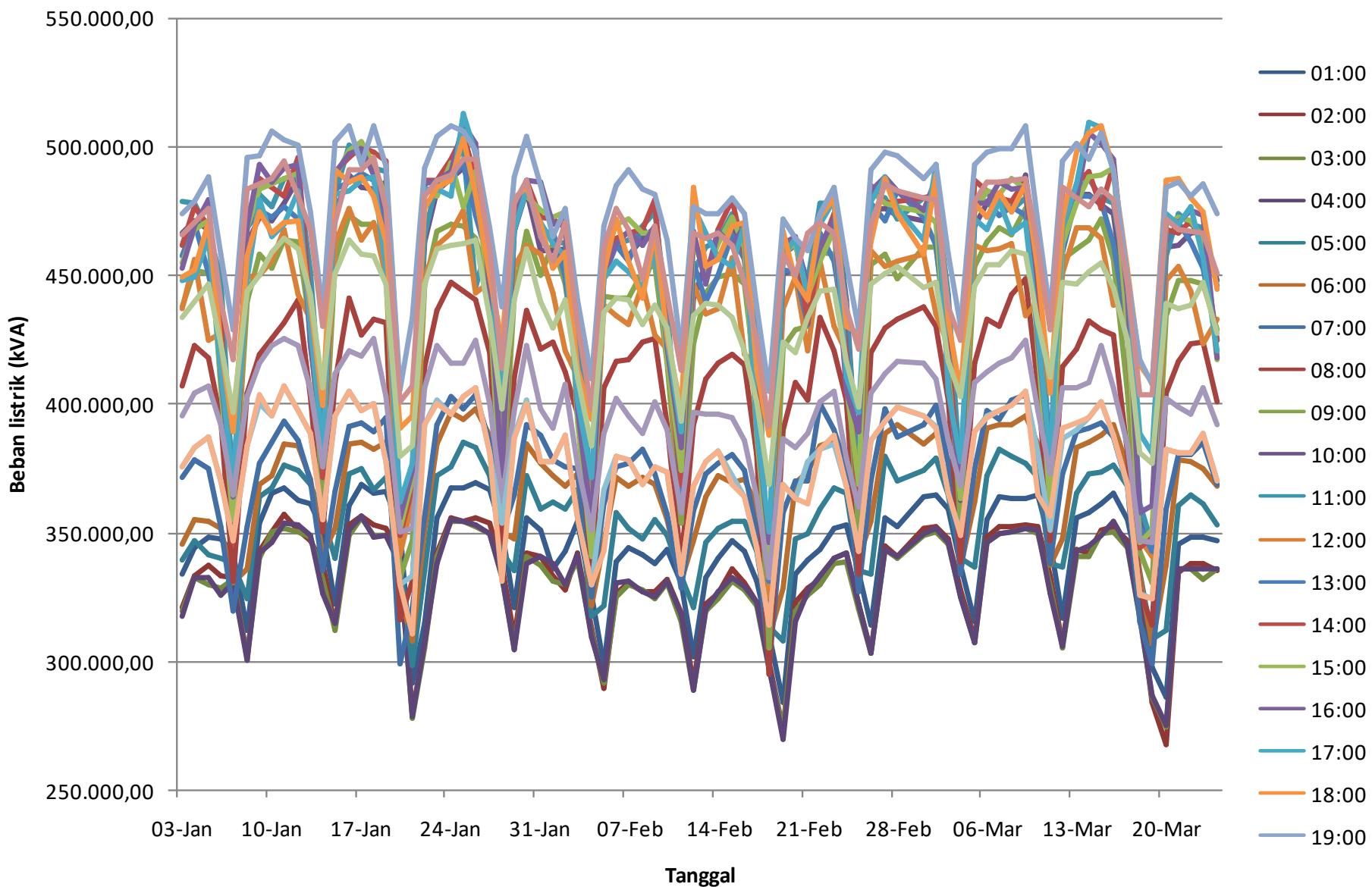


Prediksi Beban Listrik Jangka Pendek

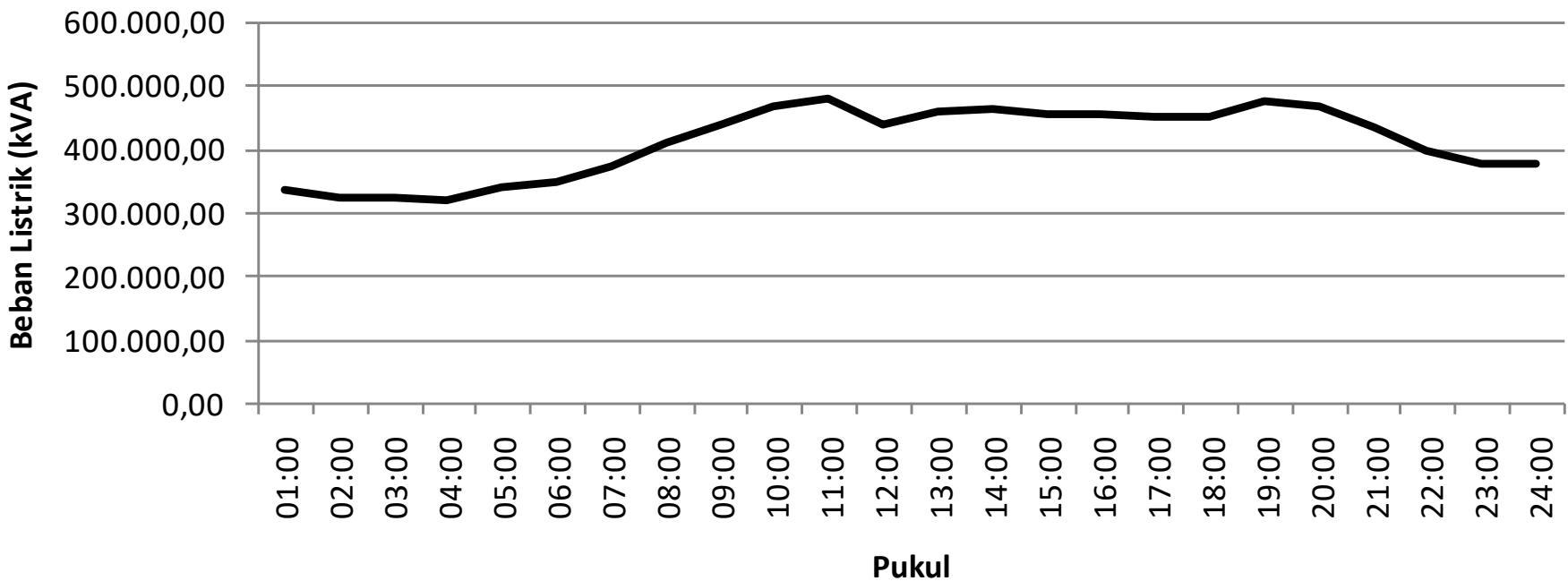
- Peramalan Data Time Series
- NEFPROX

Data Beban Listrik per jam (pukul 01:00 - 24:00)

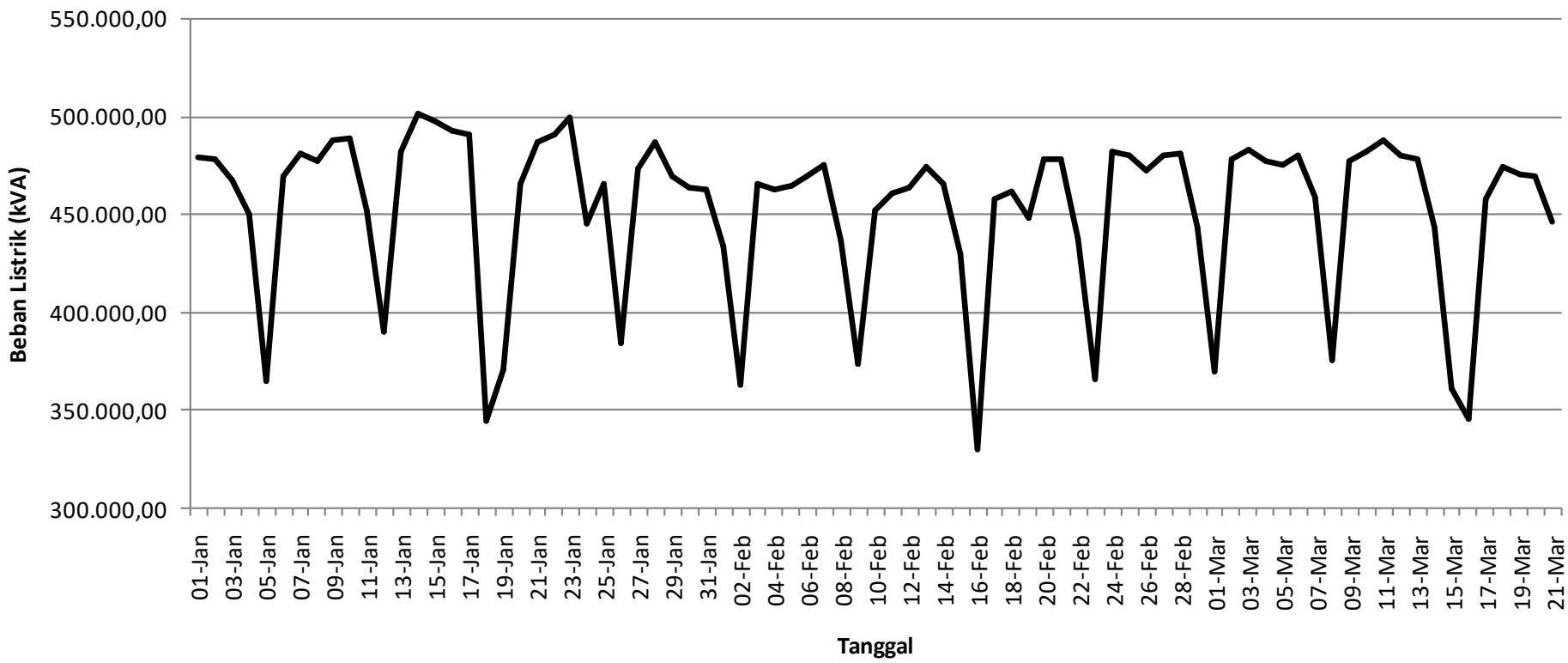
Periode 03 Januari - 24 Maret 2007 di suatu APJ



Beban listrik pada tanggal 03 Jan 2007



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



Parameter NEFPROX

- *fuzzy set* yang digunakan adalah segitiga
- *Jumlah fuzzy set* = 4 atau 5
- *Jumlah input* = 4 atau 5
- *Learning rate (σ)* = 0,0025; 0,006; dan 0,01
- Fungsi kenggotaan memiliki *intersect* 0,5

Domain fungsi keanggotaan untuk data per jam

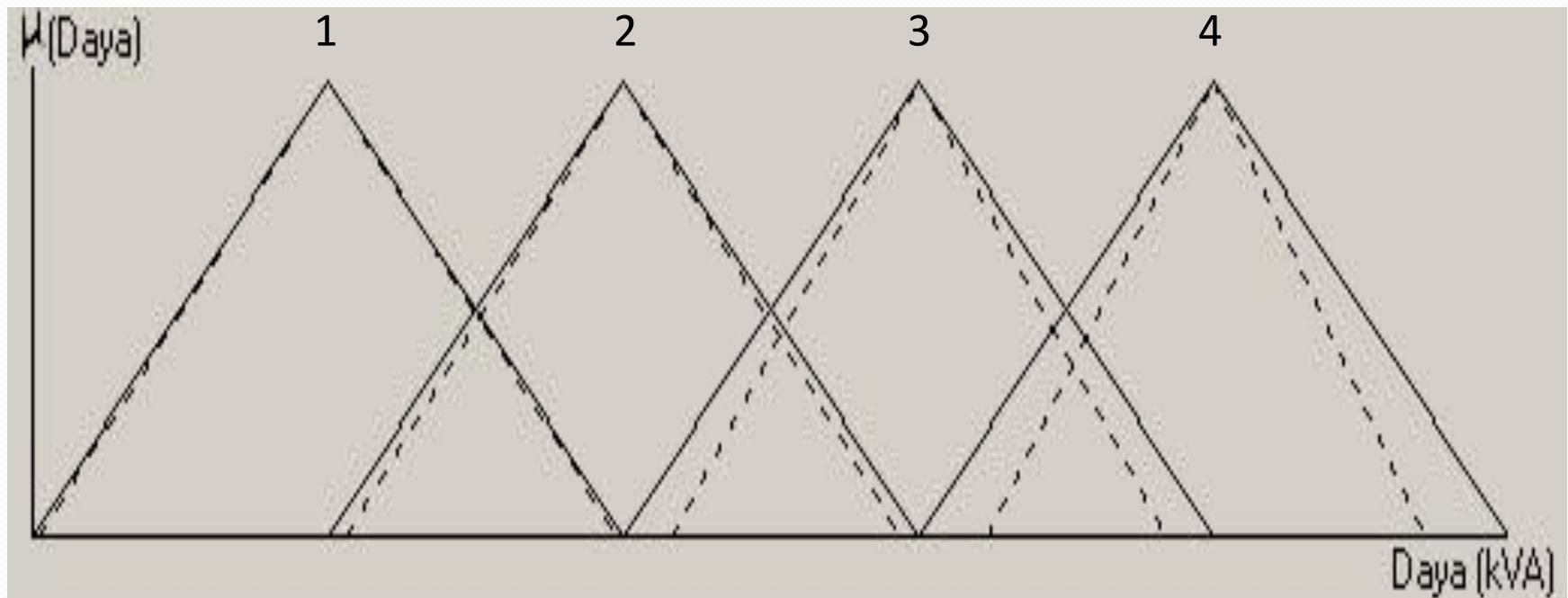
Pukul	Domain (kVA)	Pukul	Domain (kVA)
01:00	270000 - 390000	13:00	320000 - 510000
02:00	250000 - 370000	14:00	320000 - 535000
03:00	250000 - 370000	15:00	320000 - 535000
04:00	250000 - 370000	16:00	330000 - 530000
05:00	280000 - 415000	17:00	340000 - 540000
06:00	275000 - 420000	18:00	375000 - 525000
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

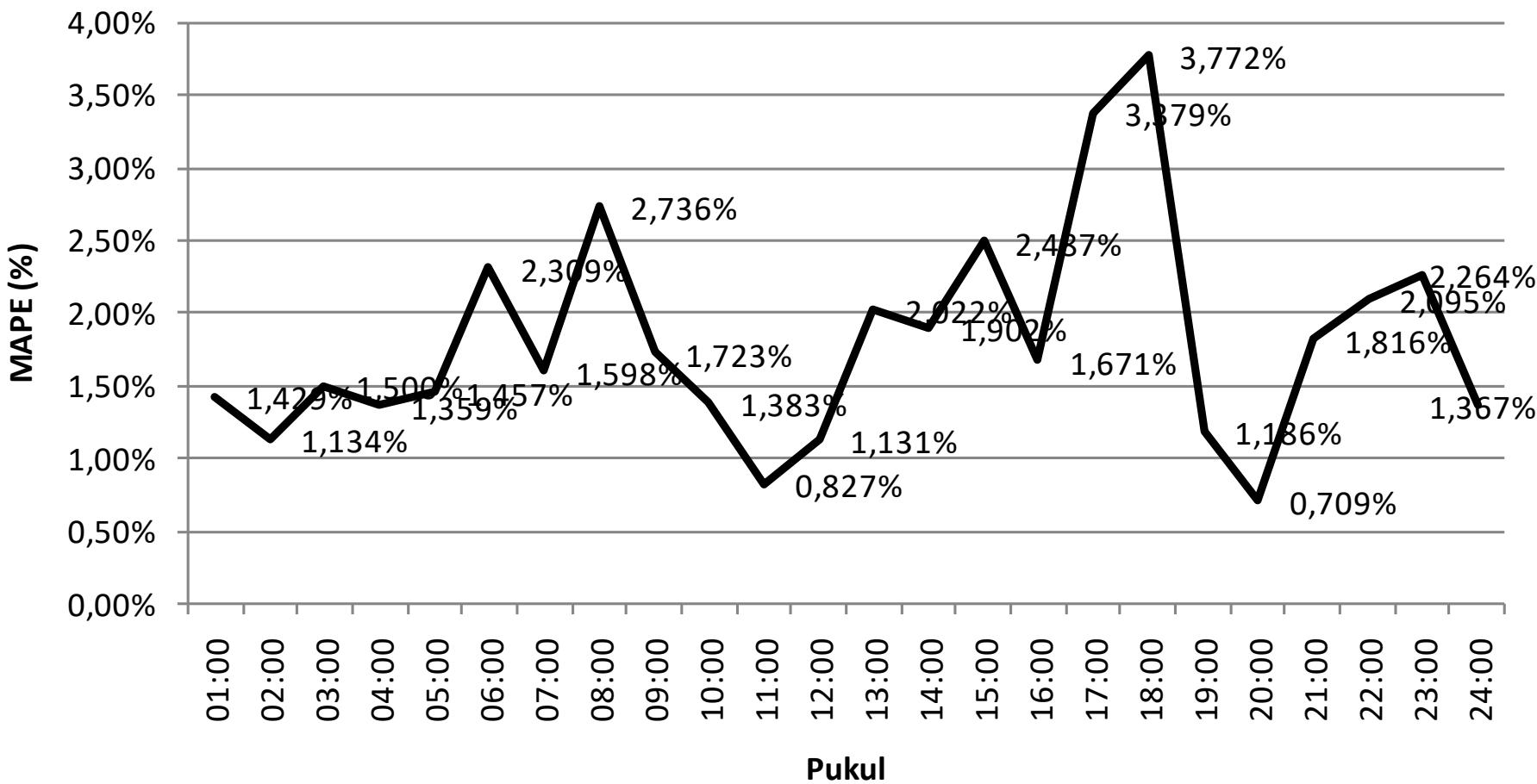
- Untuk menemukan struktur (arsitektur) NEFPROX yang paling optimal.
- Jika *jumlah fuzzy set* (F) = 4 dan *jumlah input* (P) = 4, maka total aturan (*fuzzy rule*) yang harus dibuat pada NEFPROX adalah $F^P = 4^4 = 256$ aturan.
- Jika *jumlah fuzzy set* (F) = 5 dan *jumlah input* (P) = 5, maka total aturan harus dibuat pada NEFPROX adalah $F^P = 5^5 = 3125$ aturan.

Pukul	Jumlah <i>input</i> optimal	Jumlah Fuzzy Set optimal	σ	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

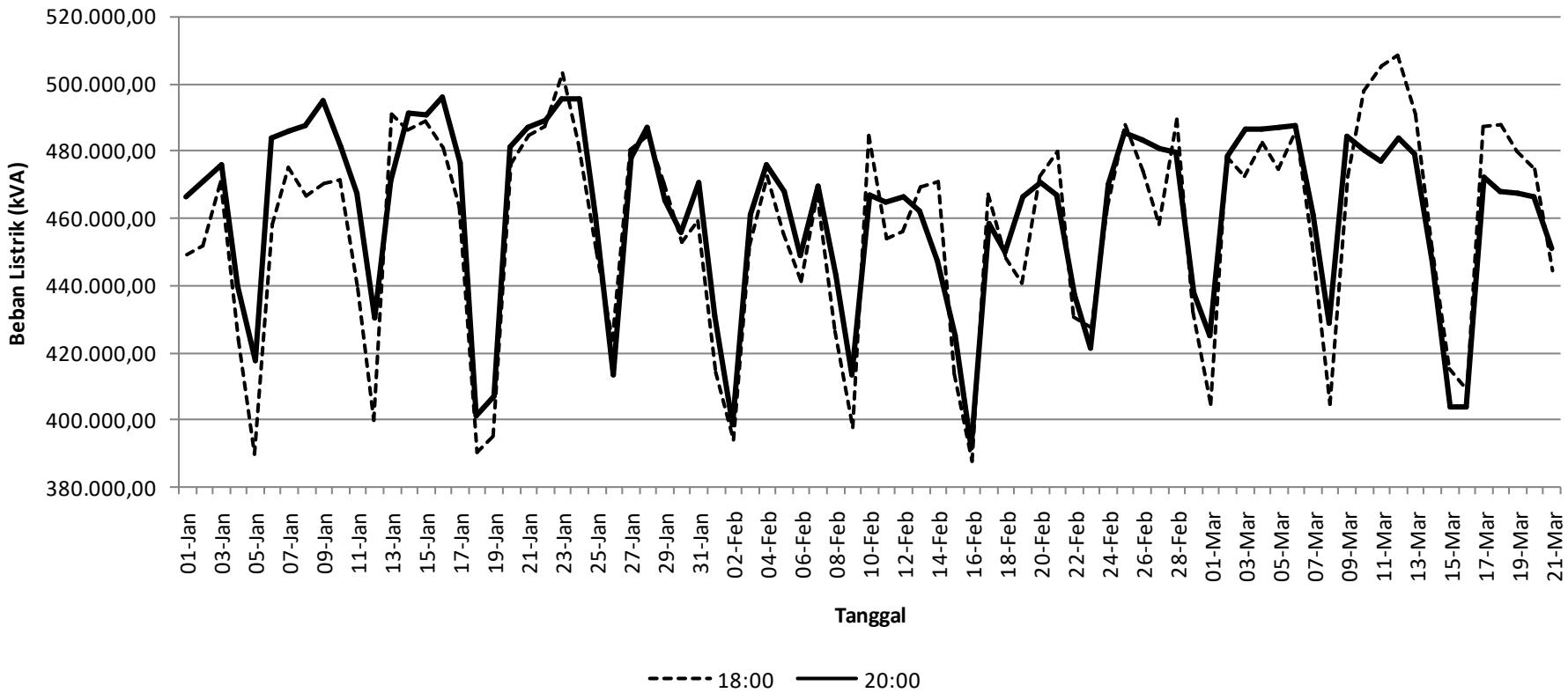
Parameter Learning



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



Beban Listrik pada pukul 18:00 dan 20:00



Klasifikasi Data dengan *Imbalance Class*

- Bagaimana mengenali pelanggan yang potensial terhadap *couponing* oleh toko.
- Kupon adalah voucher untuk promosi yang diberikan pada pelanggan melalui berbagai macam media.
- Seorang pelanggan yang membeli barang yang diiklankan di kupon, akan mendapatkan diskon sesuai yang tertera pada kupon.
- Selama ini, kupon disebarluaskan melalui majalah atau diberikan secara langsung kepada pelanggan ketika berada di kasir akan melakukan pembayaran.

Data

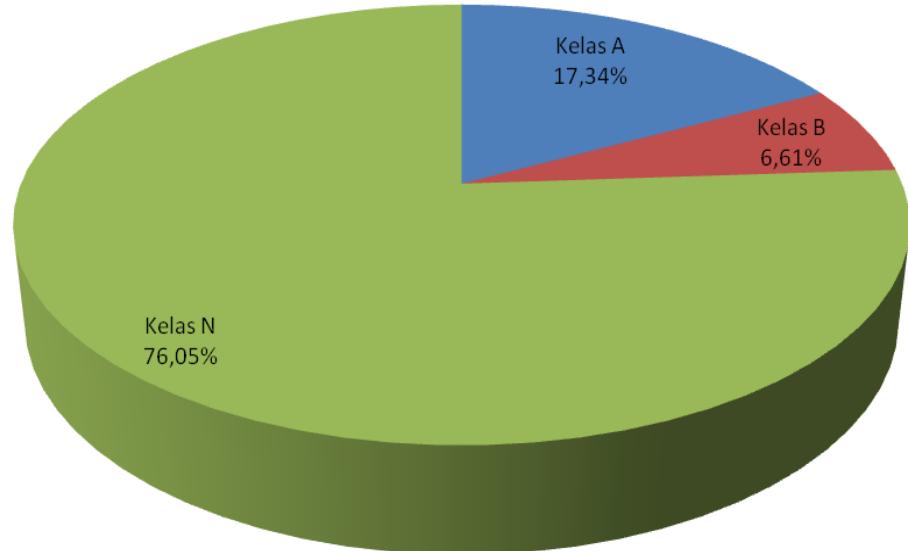
- Data-data untuk pelatihan (*training set*) dan pengujian (*test set*) berupa perilaku penebusan kupon oleh pelanggan.
- Pada data tersebut, pelanggan dibagi menjadi tiga kelas yaitu A, B, dan N.
- Setiap record memiliki 22 atribut.

Data

No	Nama Atribut	Keterangan
1	ID	Nomor pelanggan
2	C10001	Jumlah penebusan kupon C10001 oleh pelanggan
3	C10002	Jumlah penebusan kupon C10002 oleh pelanggan
...
21	C10020	Jumlah penebusan kupon C10020 oleh pelanggan
22	COUPON	atribut target

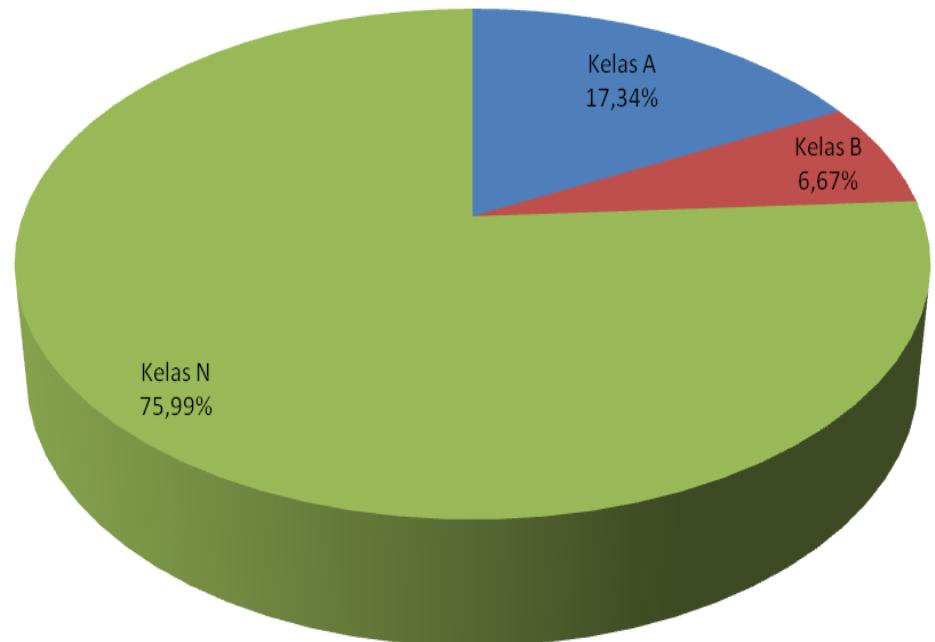
Training Set

- 50 ribu records
- Kelas A berjumlah 8668 record (17,34%)
- Kelas B berjumlah 3307 record (6,61%)
- Kelas N berjumlah 38025 record (76,05%)



Test Set

- 50 ribu records
- Kelas A berjumlah 8668 record (17,34%)
- Kelas B berjumlah 3335 record (6,67%)
- Kelas N berjumlah 37997 record (75,99%)



Prediksi Churn Pelanggan Telepon

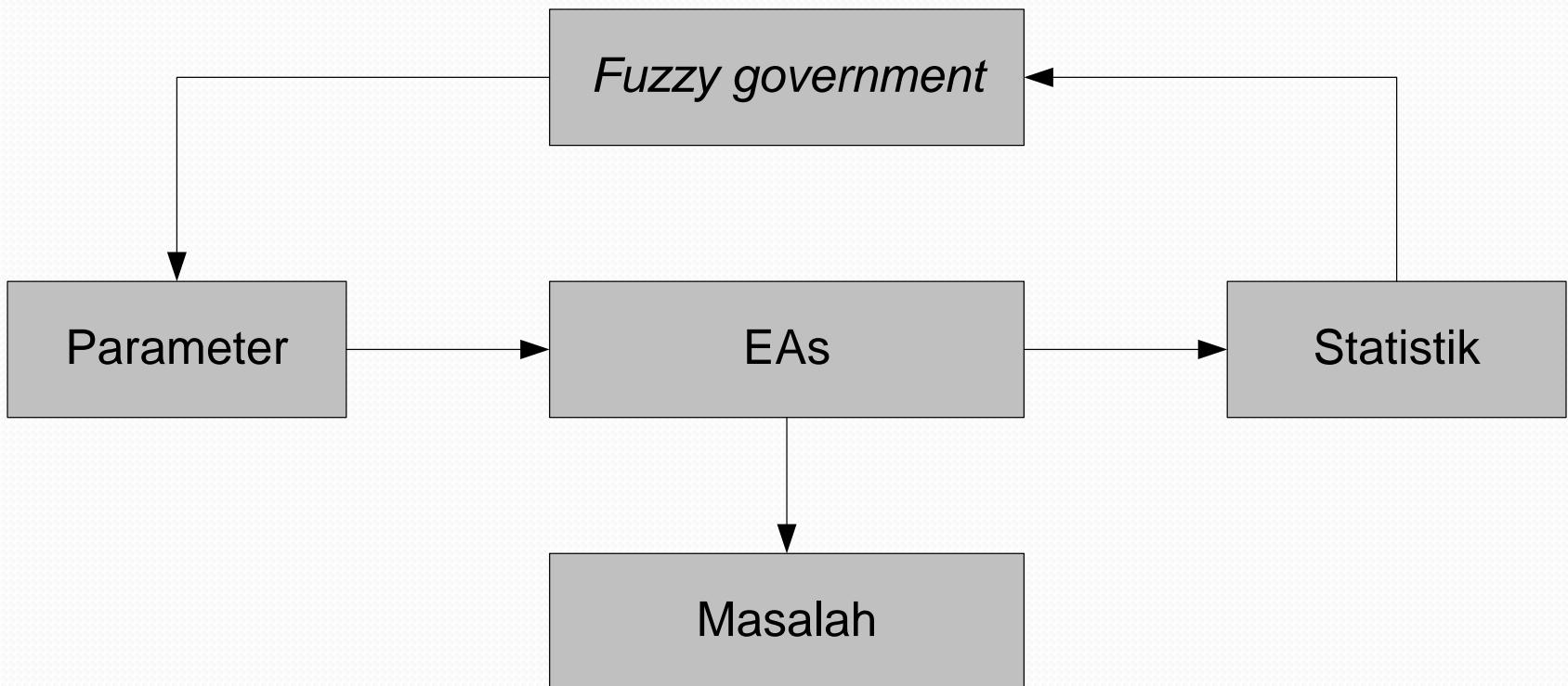
- Input: Data Pelanggan
 - Profil
 - Perilaku pemakaian telefon, dsb.
- Output:
 - Daftar pelanggan yang potensial *churn*

Churn

- *Churn*: pindahnya pelanggan dari satu operator ke operator lain.
- *Voluntary churn*: pindah dengan kemauan sendiri
 - karena kompetitor menawarkan produk yang lebih murah dan berkualitas
- *Involuntary* adalah *churn* bukan karena kemauan sendiri
 - karena pelanggan sudah lama tidak membayar tagihannya sehingga menyebabkan dicabutnya layanan terhadap pelanggan tersebut.

Atribut	Keterangan
A1	Kategori Cluster yang didefinisikan oleh carrier
A2	Jenis layanan yang dipakai pelanggan
A3	Jenis Kelamin
A4	Kategori tanggal pembayaran tagihan untuk bulan tagihan M2
A5	Kategori frekuensi pemanggilan dari Domestic to Lokal
A6	Kategori frekuensi pemanggilan dari Domestic to PSTN
A7	Kategori frekuensi pemanggilan dari Lokal to Flexy
A8	Kategori frekuensi pemanggilan dari Lokal to OLO
A9	Kategori frekuensi pemanggilan dari Domestic to PSTN
A10	Kategori frekuensi pemanggilan SMO
A11	Kategori frekuensi pemanggilan SMP
A12	Kategori durasi lamanya nomor telepon tersebut dipakai sampai bulan ke M2
A13	Kategori durasi lamanya nomor telepon tersebut dipakai dr awal
A14	Kategori jumlah revenue dari pemakaian layanan oleh pelanggan pada bulan M2
A15	Trend revenue bulan M2 terhadap bulan M3
A16	Pengkategorian SDEV dari payment
A17	Kombinasi dari variabel kategori panggilan domestic ke seluler (D2C) dan panggilan domestic ke PSTN (D2P) untuk bulan M2
A18	Kombinasi dari variabel kategori panggilan lokal ke flexi (L2F), lokal ke OLO (L2O) dan panggilan lokal ke PSTN (L2P) untuk bulan M2
A19	Kombinasi kategori penggunaan voice per hari selama satu minggu (Minggu - Sabtu) di bulan M2
A20	Kombinasi variabel TREND1-TREND4
A21	Persentase durasi pemakaian jenis voice ke jaringan Telkom dibandingkan seluruh durasi pemakaian jenis voice
A22	Label Kelas (Churn / Active)

Adaptive EAs



Contoh Kasus

Kota	Tipe	Tipe Pembayaran	Tagihan Bulanan	Jumlah Panggilan	Panggilan TidakNormal	Churn
Jakarta	Pemerintah	Cash	Besar	10	Sedikit	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	8	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Kecil	5	Banyak	Tidak
Surabaya	Corporate	Cash	Kecil	3	Banyak	Ya
Surabaya	Corporate	Cash	Kecil	2	Banyak	Ya
Surabaya	Corporate	Kartu Kredit	Besar	1	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	9	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	7	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	6	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	4	Sedang	Tidak
Jakarta	Corporate	Kartu Kredit	Sedang	9	Sedang	Tidak

Diskritisasi dan transformasi

Kota	Tipe	Tipe Pembayaran	Tagihan Bulanan	Jumlah Panggilan	Panggilan TidakNormal	Churn
Surabaya	Corporate	Cash	Kecil	[2...3]	Banyak	Class1
Surabaya	Corporate	Cash	Kecil	[1...2]	Banyak	Class1
Jakarta	Corporate	Kartu Kredit	Sedang	[3...4]	Sedang	Class2
Jakarta	Corporate	Kartu Kredit	Sedang	[5...6]	Sedang	Class2
Jakarta	Corporate	Kartu Kredit	Kecil	[4...5]	Banyak	Class2
Jakarta	Corporate	Kartu Kredit	Sedang	[7...8]	Sedang	Class2
Surabaya	Corporate	Kartu Kredit	Besar	[0...1]	Sedang	Class2
Jakarta	Corporate	Kartu Kredit	Sedang	[8...9]	Sedang	Class2
Jakarta	Corporate	Kartu Kredit	Sedang	[6...7]	Sedang	Class2
Jakarta	Pemerintah	Cash	Besar	[9...10]	Sedikit	Class2
Surabaya	Corporate	Cash	Kecil	[2...3]	Banyak	Class1

APACS

- APACS: *Automatic Analysis And Classification of Conceptual Patterns*
- APACS adalah merupakan teknik induksi probabilistik.
- Diantara beberapa pasangan atribut yang mungkin, APACS dapat mengidentifikasi pasangan yang memiliki hubungan asosiatif bahkan pada basis data yang memiliki data *noise* dan memiliki banyak nilai yang hilang.

APACS → *First Order Rule*

First Order Rule	Nilai Asosiasi terhadap Churn (d)	Nilai Ketidakpastian Asosiasi/ Bobot (ϖ)
Kota = 'Surabaya'	2.41	0.30
Kota = 'Jakarta'	-2.41	-1
Tipe Pembayaran = 'Cash'	2.41	0.30
Tipe Pembayaran = 'Kartu Kredit'	-2.41	-1
Tagihan Bulanan = 'Kecil'	2.41	0.30
Jumlah Panggilan = '[2...3]'	2.10	1
Jumlah Panggilan = '[1...2]'	2.10	1
Panggilan Tidaknormal = 'Banyak'	2.41	0.30

Populasi pada *Second Order rule*

TipePembayaran = 'Kartu Kredit'

TagihanBulanan = 'Kecil'

TipePembayaran = 'Kartu Kredit'

Kota = 'Surabaya'

Kota = 'Surabaya'

TagihanBulanan = 'Kecil'

TipePembayaran = 'Kartu Kredit'

TipePembayaran = 'Kartu Kredit'

JumlahPanggilan = 'Sedikit'

Kota = 'Surabaya'

Crossover

O1 TipePembayaran = 'Kartu Kredit'

TagihanBulanan = 'Kecil'

O2 JumlahPanggilan = 'Sedikit'

Kota = 'Surabaya'

A1 TipePembayaran = 'Kartu Kredit'

Kota = 'Surabaya'

A2 JumlahPanggilan = 'Sedikit'

TagihanBulanan = 'Kecil'

Contoh rule hasil learning

Order	Rule	Weight of Evidence
1st Order	kota = 'Surabaya'	0.3
1st Order	tipepembayaran = 'Cash'	0.3
2nd Order	jumlahpanggilan = '[2...3]' AND panggilantidaknormal = 'Banyak'	1.0
2nd Order	panggilantidaknormal = 'Banyak' AND tipepembayaran = 'Cash'	1.0
3th Order	jumlahpanggilan = '[2...3]' AND panggilantidaknormal = 'Banyak' AND tipepembayaran = 'Cash'	1.0
3th Order	jumlahpanggilan = '[2...3]' AND tagihanbulanan = 'Kecil' AND tipepembayaran = 'Cash'	1.0
4th Order	jumlahpanggilan = '[2...3]' AND panggilantidaknormal = 'Banyak' AND tagihanbulanan = 'Kecil' AND tipepembayaran = 'Cash'	1.0

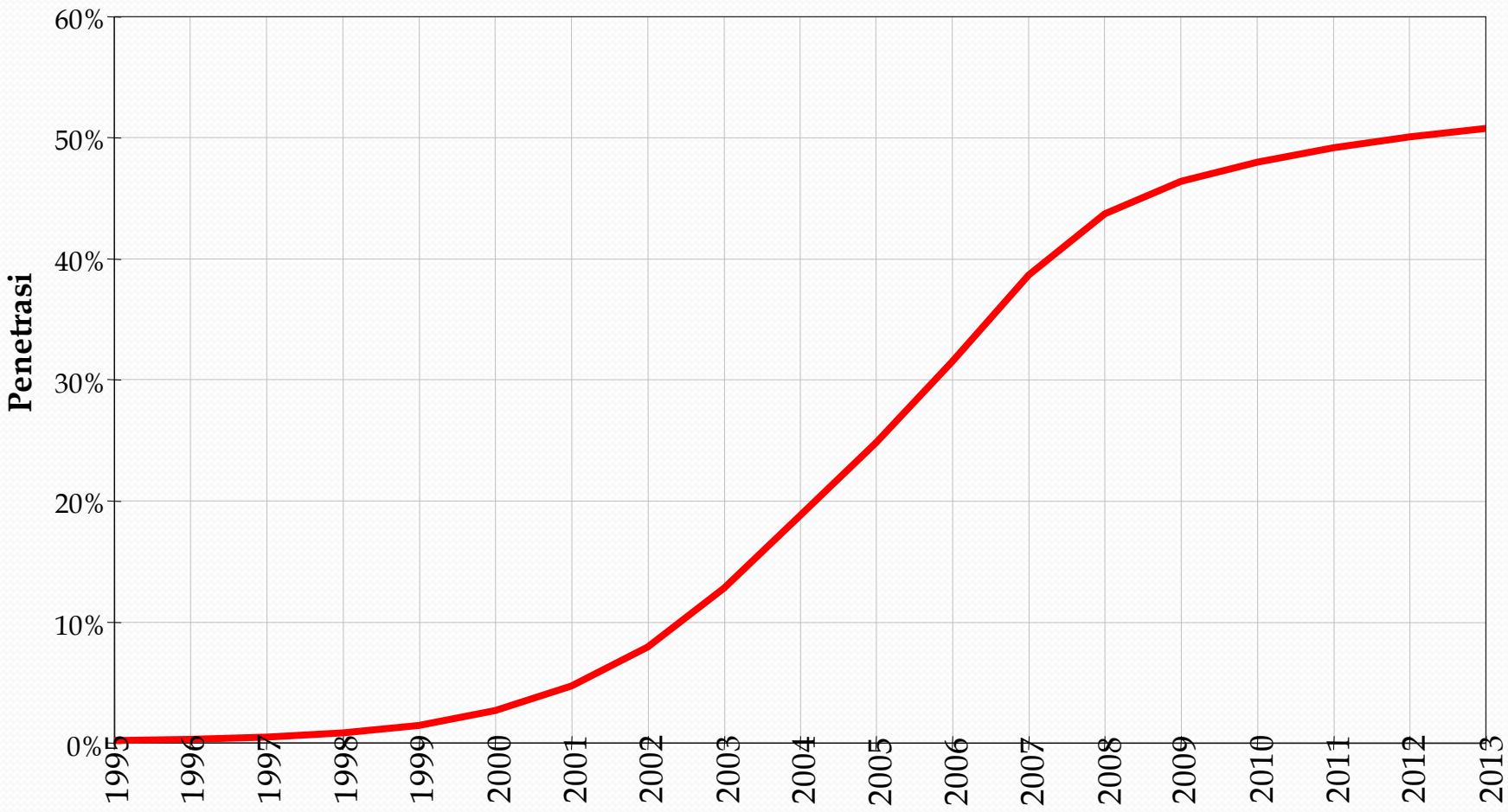
Karakteristik Trafik

- Sangat penting untuk dipahami
- Untuk menentukan dimensi jaringan
- Sistem prediksi berbasis data *time series*

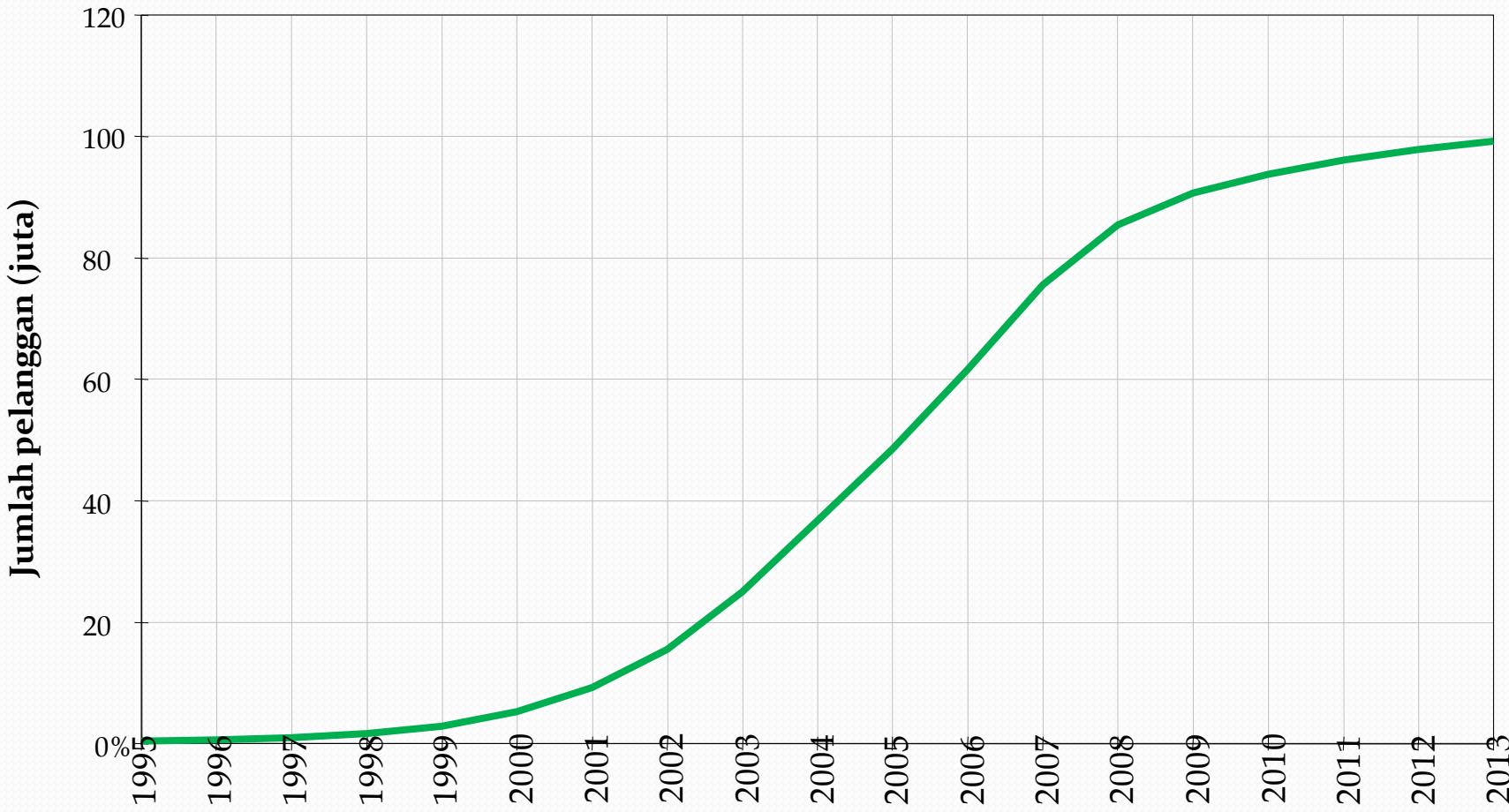
Traffic Forecasting

- Pertumbuhan penetrasi
- Pertumbuhan jumlah pelanggan
- Pertumbuhan trafik

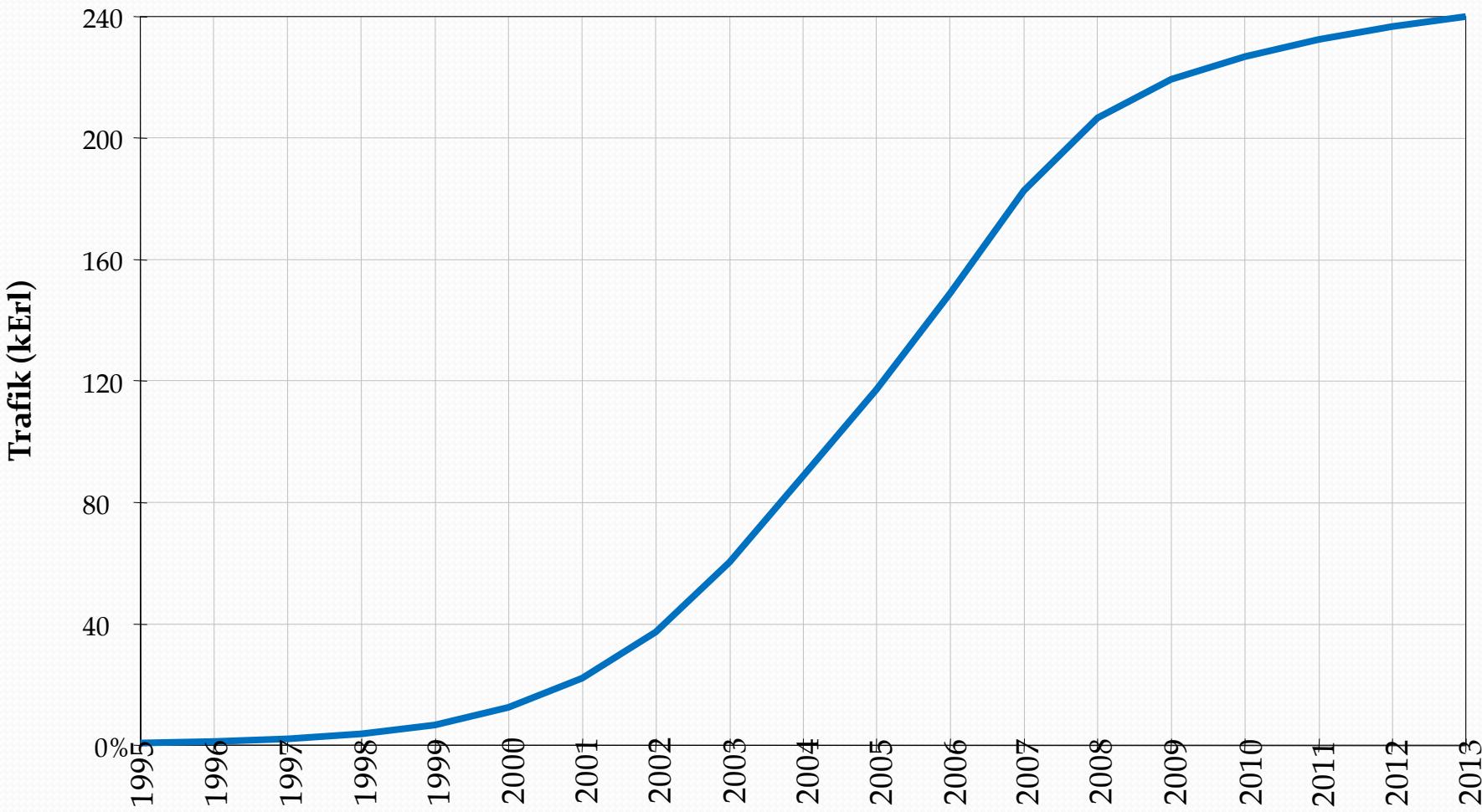
Pertumbuhan Penetrasi



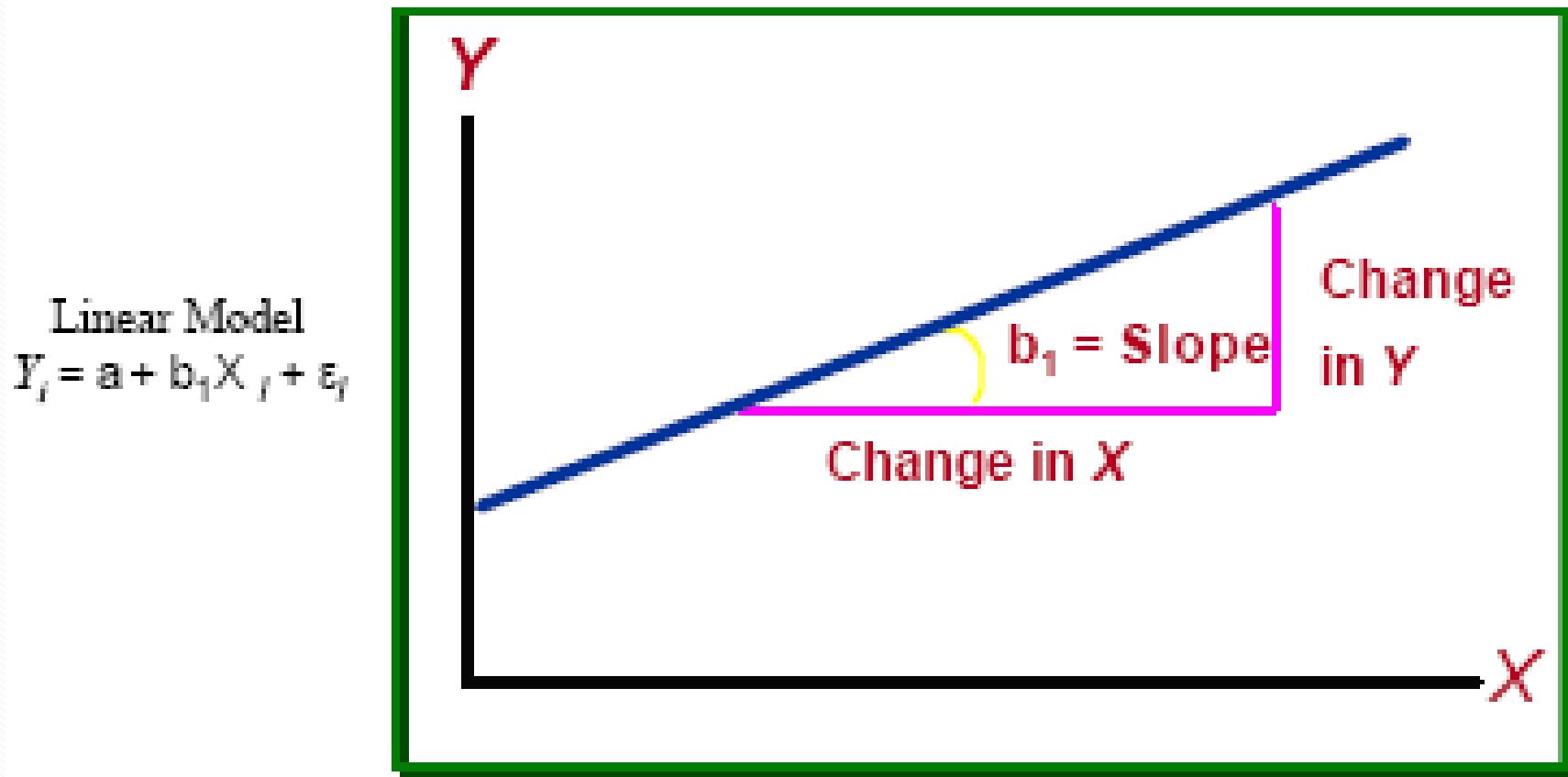
Pertumbuhan jumlah pelanggan



Pertumbuhan trafik



Model Regresi Linear



Prediksi *Time Series*

$$z = a_0 + a_1 y_1 + a_2 y_2 + \dots + a_k y_k$$

- y_1 sampai y_k adalah masukan yang berupa data-data sebelumnya, $H-1, H-2, \dots, H-k$.
- Ukuran *time series* (k) ditentukan secara coba-coba (*trial-and-error*) karena memang sangat sulit mengetahui berapa jumlah data masukan yang tepat.

Representasi individu GA

$$z = 0,1157 + 0,7315y_1 + 0,3995y_2$$



a_0	a_1	a_2
0,1157	0,7315	0,3995

Grammatical Evolution (GE)

- Untuk masalah prediksi, GE adalah algoritma EAs yang akan menghasilkan model terbaik.
- Dengan representasi kromosom yang berupa fungsi atau program, GE bisa melakukan pencarian model prediksi yang lebih bervariasi
- Dengan membangun *grammar* yang luas dalam notasi *Backus Naur Form* (BNF), GE bisa melakukan pencarian untuk sangat banyak kemungkinan model prediksi, baik linier maupun non-linier.

```

N = {expr, op, pre_op}
T = {Sin, Cos, Tan, Log, +, -, /, *, y1, y2, y3, y4, 0,5, 1, 1,5, 2, ()}
S = <expr>
P dapat direpresentasikan sebagai:
(1) <expr> ::= <expr> <op> <expr>           (A)
      | (<expr> <op> <expr>)             (B)
      | <pre_op> (<expr>)               (C)
      | <var>                         (D)

(2) <op> ::= +           (A)
      | -             (B)
      | /             (C)
      | *             (D)

(3) < pre_op > ::= Sin        (A)
      | Cos          (B)
      | Tan          (C)
      | Log          (D)

(4) < var > ::= y1          (A)
      | y2            (B)
      | y3            (C)
      | y4            (D)
      | 0,5           (E)
      | 1              (F)
      | 1,5           (G)
      | 2              (H)

```

Contoh model prediksi GE

$$z = 0,5 + 2y_1 - y_2 + (0,5 * y_3)$$

$$z = 1,5 + \sin(y_1) - \log(y_2)$$

$$z = \frac{0,5 + y_1}{(y_2 + (0,5 * y_3) - \cos(y_4))}$$

Fungsi *Fitness*

$$f = \frac{1}{(K + b)}$$

dimana b merupakan suatu bilangan yang dianggap sangat kecil untuk menghindari pembagian dengan 0, sedangkan K adalah rata-rata kesalahan prediksi untuk semua data penjualan.

Fungsi *Fitness*

Kesalahan prediksi merupakan harga mutlak dari selisih hasil prediksi menggunakan model tersebut (z) dengan data penjualan yang sebenarnya (z^*). Dengan demikian, rata-rata kesalahan prediksi dapat dituliskan sebagai

$$K = \frac{1}{N} \sum_{i=1}^N |z_i - z_i^*|$$

dimana N adalah jumlah semua data prediksi.

Intrusion Detection Systems (IDS)

- IDS adalah usaha mengidentifikasi adanya penyusup yang memasuki sistem tanpa otorisasi atau seorang *user* yang sah tetapi menyalahgunakan *privilege* sumber daya sistem.
- Meskipun teknologi ini belum sempurna dan masih memerlukan perbaikan, namun saat ini *intrusion detection* memegang peranan cukup penting pada keseluruhan arsitektur keamanan sebuah sistem jaringan komputer

Jenis Serangan

- ***Probe/scan***
 - Usaha-usaha yang tidak lazim untuk memperoleh akses ke dalam suatu sistem atau untuk menemukan informasi tentang sistem tersebut.
 - Kegiatan *probe* dalam jumlah besar dengan menggunakan *tools* secara otomatis biasa disebut *scan*.
- ***Denial of Service (DoS)***
 - Usaha yang dilakukan untuk membuat sumber daya jaringan maupun komputer tidak bekerja, sehingga tidak mampu memberikan layanan.
- ***Penetration***
 - Merupakan sebuah usaha untuk mengubah data, *privilege*, atau sumber daya pada sistem. Beberapa jenis gangguannya, antara lain :
 - *User to Root (U2R)*: *user* lokal pada suatu *host* memperoleh hak admin.
 - *Remote to user (R2L)*: pengakses luar dapat *account* lokal di *host* target.

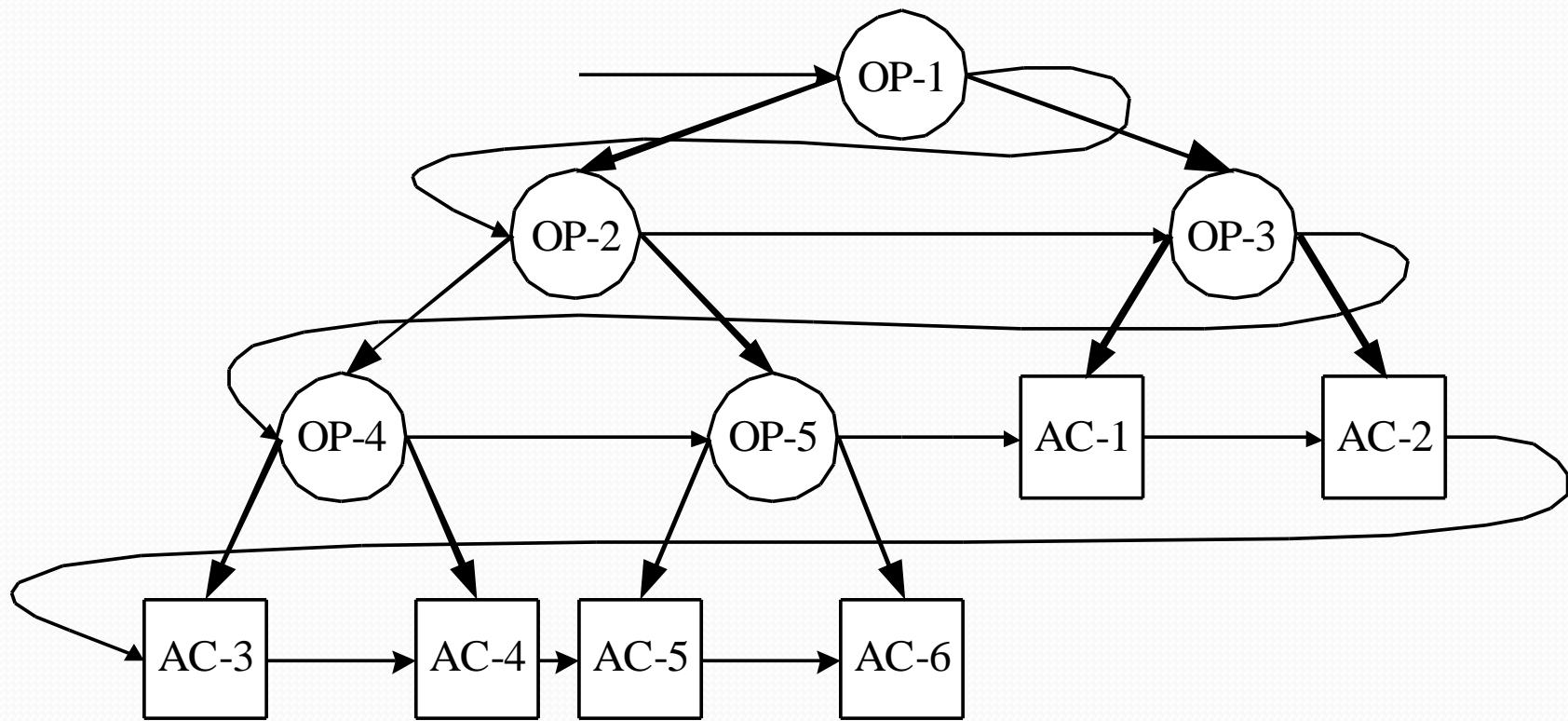
No.	Nama	Keterangan
1	Service	Service yang digunakan pada jaringan.
2	Src_bytes	Jumlah data yang berasal dari <i>node sumber</i> ke <i>tujuan</i> .
3	Dst_bytes	Jumlah data yang berasal dari <i>node tujuan</i> ke <i>sumber</i> .
4	Logged_in	Menunjukkan status <i>login user</i> , jika <i>login</i> sukses maka nilainya TRUE.
5	Count	Jumlah koneksi yang tersambung ke <i>host</i> yang sama dalam 2 detik terakhir.
6	Srv_count	Jumlah koneksi yang tersambung dengan <i>service</i> yang sama dalam 2 detik terakhir.
7	Serror_rate	Prosentase dari koneksi yang memiliki 'SYN' error pada koneksi ke <i>host</i> yang sama.
8	Srv_error_rate	Prosentase dari koneksi yang memiliki 'REJ' error pada koneksi dengan <i>service</i> yang sama.
9	Srv_diff_host_rate	Prosentase dari koneksi tersambung ke <i>host</i> yang berbeda.
10	Dst_host_count	Jumlah koneksi yang tersambung ke <i>node tujuan</i> .
11	Dst_host_srv_count	Jumlah koneksi yang tersambung ke <i>node tujuan</i> dengan <i>service</i> yang sama dalam 2 detik terakhir.
12	Dst_host_diff_srv_rate	Jumlah koneksi yang tersambung ke <i>node tujuan</i> dengan <i>service</i> yang berbeda.
13	Class	Kategori data, yaitu normal atau abnormal.

Struktur Linier Kromosom Rule Classifier

Gen ₁				..	Gen _n				Gen _{n+1}		
ac ₁		op ₁		..	ac _n		op _n		ac _{n+1}		*
var ₁	s ₁	o ₁	prec ₁	..	var _n	s _n	o _n	prec _n	var _{n+1}	s _{n+1}	*

- <ac_i> : *Atomic condition*, terdiri dari: var_i sebagai variabel, dalam hal ini merupakan atribut pada data; s_i sebagai set, yang merupakan label *fuzzy set* untuk nilai suatu data.
- <op_i> : *Fuzzy operator*, terdiri dari o_i , sebagai operator, yaitu AND dan OR; dan prec_i, sebagai *precedence* atau tingkat prioritas.

Struktur Pohon Kromosom Rule Classifier



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.