

# Hebb Net

---

**Dikompilasi Oleh:**

***Agust Isa Martinus***

**BACK**

**Agust Isa Martinus @2008**

**<aimxx@yahoo.com>**

# Karakteristik JST Hebb

---

## Karakteristik Hebb Net.

1. **Aktifasi Bipolar.**
  - Output **+1** (**fires**) atau **-1** (*does not fire*)
2. Setiap neuron dihubungkan dengan sinapsis atau **jalur berarah dan berbobot** (vektor).
  - Positif: excitatory ( $w > 0$ )
  - Negatif: inhibitory ( $-p, p > 0$ )
3. Setiap neuron mempunyai **satu threshold** ( $\theta$ , ambang batas) **yang tetap**.
  - Fires, jika  $net \geq \theta$ .
  - Threshold diset sedemikian rupa sehingga *inhibition* adalah absolut. (any nonzero inhibitory input will prevent the neuron from firing.)
4. Setiap sinyal membutuhkan **satu satuan waktu** untuk mengalir pada jalur koneksi dari satu neuron ke neuron berikutnya.

# Hebbian Synapse

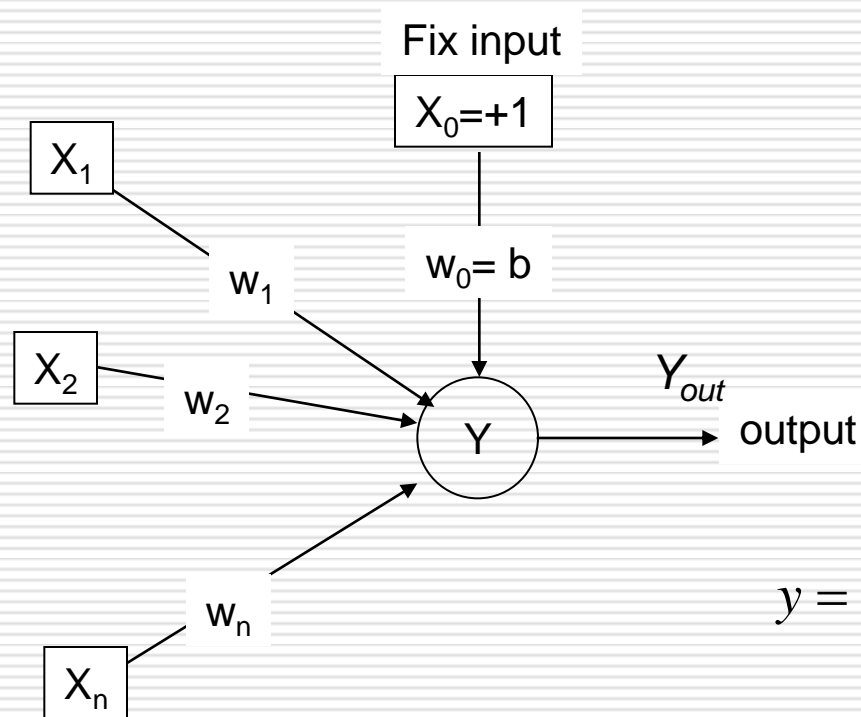
---

- ☐ Time-dependent mechanism
- ☐ Local mechanism
- ☐ Interactive mechanism
- ☐ Conjunctional or correlational mechanism

# Review:

## Arsitektur Neuron Y

---



$$y = F(net) = \begin{cases} +1, & \text{jika } net \geq \theta \\ -1, & \text{jika } net < \theta \end{cases}$$

$$net = \sum_{i=0}^n w_i x_i \quad \text{atau} \quad net = b + \sum_{i=1}^n w_i x_i$$

# Hebb Net

---

## ☐ Arsitektur

- Single Layer, Feed Forward

## ☐ Metode Pembelajaran

- Extended Hebb Rule
  - ☐ (target bipolar)
  - ☐ (input binary or bipolar)

## ☐ Fungsi Aktivasi

- Signum Function
  - ☐ If  $net \geq \theta$  then Output = +1
  - ☐ If  $net < \theta$  then Output = -1

# Pelatihan dan Penggunaan

---

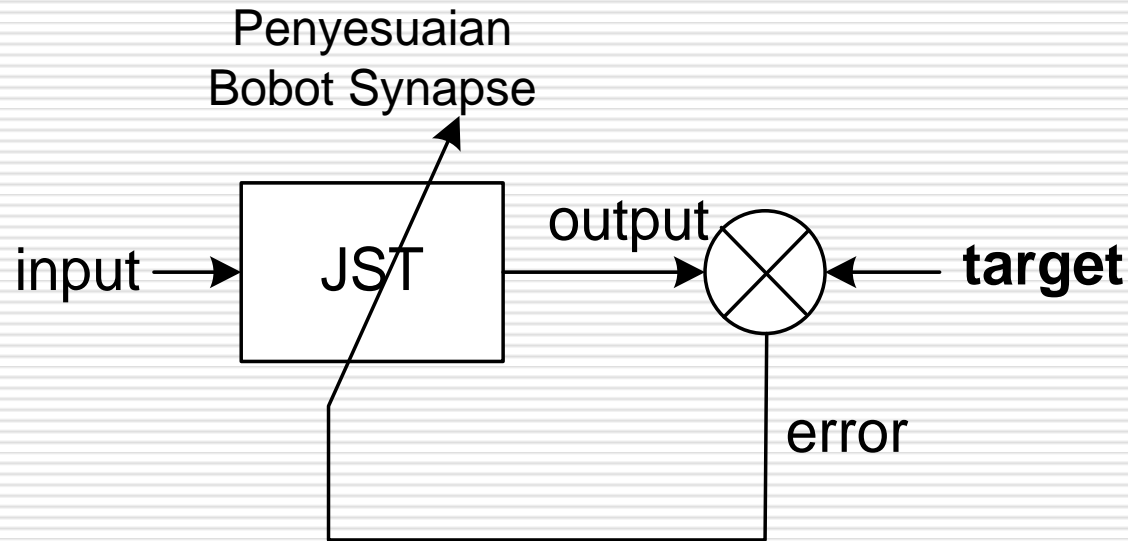
## Pelatihan

- Latih JST menggunakan Algoritma Extended Hebb.
  - Extenden Hebb hanya 1-epoch.

## Penggunaan

- Operasikan JST menggunakan bobot-bobot (pengetahuan) hasil latihan (Algoritma Extended Hebb) yang telah dilakukan.

# Proses Pelatihan dengan Supervisor



Pelatihan dengan Supervisor

# Dasar Algoritma Extended Hebb

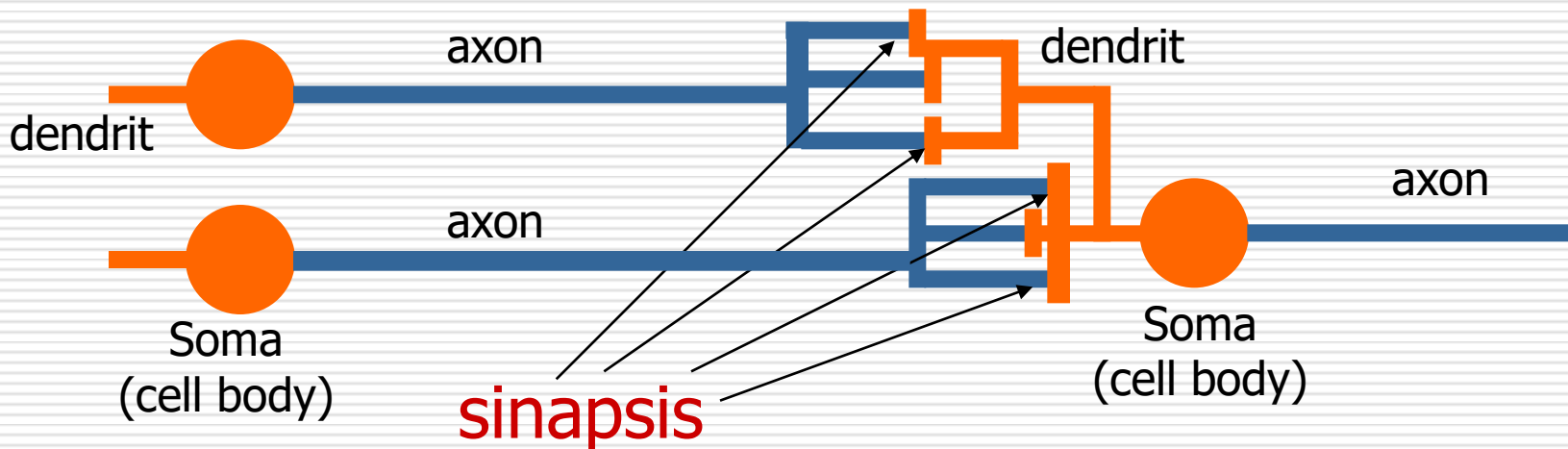
## Original Hebb-rule

- Jika dua buah neuron yang saling terhubung melalui sinapsis, aktif (*fire*) secara bersamaan, maka bobot sinapsisnya bertambah kuat.

## extended Hebb-rule

(original Hebb-rule, ditambahkan yang berikut)

- Jika dua buah neuron yang saling terhubung melalui sinapsis, tidak aktif (*not fire*) secara bersamaan, maka bobot sinapsisnya bertambah kuat juga.





# Algoritma Pembelajaran Extended Hebb

## □ Langkah 0 : Inisialisasi

Inisialisasi laju belajar dan semua bobot

$$w_i = 0$$

//(i=0 to n, sebanyak input)

$$\alpha = 1$$

//(0 <  $\alpha$  ≤ 1, laju belajar)

## □ Langkah 1 : Iterasi

*(hanya 1-epoch, 1x iterasi)*

Untuk setiap pasangan masukan dan keluaran, s:t, [s=(s1, s2, s3, ..., sn, s0)] laksanakan langkah 2 – 4,

### ■ Langkah 2

Set aktifasi unit masukan

$$x_i = s_i$$

//(i=0 to n;  $x_0=s_0=+1$ )

### ■ Langkah 3

Set aktifasi unit keluaran

$$y = t$$

### ■ Langkah 4

Hitung semua bobot dan bias

$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha \cdot x_i \cdot y$$

//(i=0 to n;  $x_0=+1$ )

# Prosedur **Penggunaan** Extended Hebb-net

---

## □ **Pelatihan**

Latih dengan algoritma Extended Hebb untuk mendapatkan bobot.

## □ **Penggunaan**

Untuk setiap pasangan masukan dan keluaran,  $s:t$ , yang akan diklasifikasi/dikenali, laksanakan langkah 1 – 2,

### ■ **Langkah 1**

Set aktifasi unit masukan

$$x_i = s_i \quad (i=0 \text{ to } n; x_0=s_0=+1)$$

### ■ **Langkah 2**

Hitung respons unit keluaran,  $y$

$$\text{net} = \sum x_i w_i \quad (i=0 \text{ to } n; x_0=+1)$$

$y =$

- $+1$  jika  $\theta \leq \text{net}$
- $-1$  jika  $\text{net} < \theta$

# Kasus 1 : Fungsi Logika

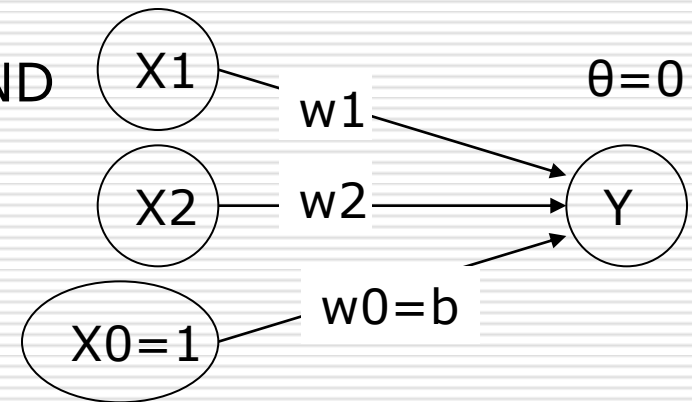
---

- Kasus:
  - Buat jaringan Hebb untuk menyatakan fungsi logika AND
- Representasi masukan/keluaran yang digunakan dicoba dengan:
  - a. Masukan dan keluaran biner  $\{0, +1\}$
  - b. Masukan biner  $\{0, +1\}$  dan keluaran bipolar  $\{-1, +1\}$
  - c. Masukan dan keluaran bipolar  $\{-1, +1\}$
- Melihat masalah yang timbul pada penggunaan fungsi aktivasi threshold.
  - Kadangkala jaringan dapat menentukan pola secara benar jika menggunakan representasi bipolar.

# a. Representasi Biner-Biner

Tabel Masukan dan Target fungsi AND

Pola	Masukan			Target
	s1	s2	s0=1	t
1	1	1	1	1
2	1	0	1	0
3	0	1	1	0
4	0	0	1	0



Arsitektur Jaringan Hebb

$w_i(\text{new}) = w_i(\text{old}) + \Delta w_i$ ,  
dengan  $\Delta w_i = \alpha x_i t$  ( $i=0, \dots, n$ )

$$net = \sum_{i=0}^n w_i x_i \quad \theta=0$$

$$y = \varphi(net) \begin{cases} +1 & \text{jika } net \geq \theta \\ 0 & \text{jika } net < \theta \end{cases}$$

- Masukan biner
- Target keluaran biner

Algoritma

# a. Representasi Biner-Biner

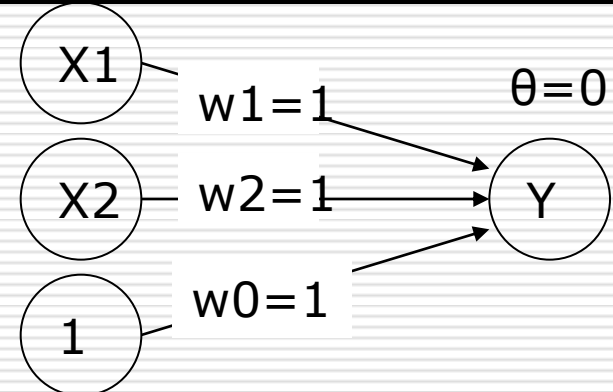
Pola	Masukan			Target	Perubahan Bobot			Bobot		
	x1	x2	x0=1	y=t	$\Delta w_1$	$\Delta w_2$	$\Delta w_0$	w1	w2	w0
								0	0	0
1	1	1	1	1	1	1	1	1	1	1
2	1	0	1	0	0	0	0	1	1	1
3	0	1	1	0	0	0	0	1	1	1
4	0	0	1	0	0	0	0	1	1	1

- ❑ Perhitungan bobot dan bias
  - $w_i(\text{new}) = w_i(\text{old}) + \Delta w_i$ , dimana  $\Delta w_i = \alpha x_i y$  ( $i=0, \dots, n$ )
- ❑ Hasil pembelajaran
  - Bobot berubah akibat pasangan data pertama saja
  - Dari hasil iterasi, diperoleh  $w_1 = 1$ ,  $w_2 = 1$ , dan  $w_0 = 1$

# Penggunaan Representasi Biner-Biner

Pola	x1	x2	$net = \sum_{i=0}^2 x_i w_i$ (w1=1, w2=1, w0=1)	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ 0 & \text{jika } net < 0 \end{cases}$
1	1	1	$1.1 + 1.1 + 1.1 = 3$	1
2	1	0	$1.1 + 0.1 + 1.1 = 2$	1
3	0	1	$0.1 + 1.1 + 1.1 = 2$	1
4	0	0	$0.1 + 0.1 + 1.1 = 1$	1

- ❑ f(net) tidak sama dengan target yang diinginkan pada fungsi AND
- ❑ Jaringan TIDAK DAPAT 'MENGERTI' pola yang dimaksud



# Latihan:

## Fungsi Logika AND, Biner-Biner

- Menggunakan cara yang sama dengan contoh, tetapi dengan urutan (tabel) masukan yang berbeda.

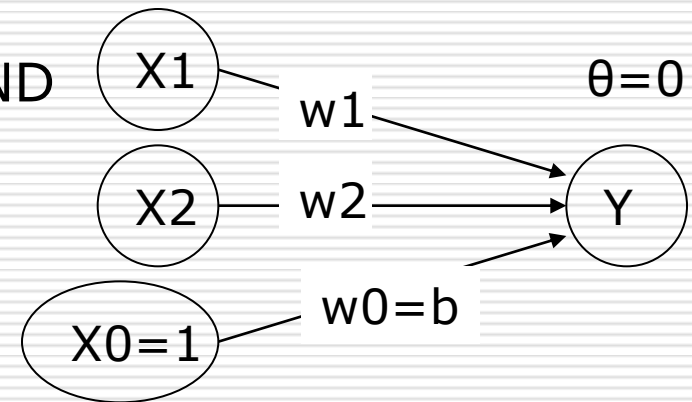
Masukan			Target
s1	s2	s0	t
1	0	1	0
1	1	1	1
0	1	1	0
0	0	1	0

Masukan			Target
s1	s2	s0	t
1	0	1	0
0	1	1	0
0	0	1	0
1	1	1	1

## b. Representasi Biner-Bipolar

Tabel Masukan dan Target fungsi AND

Pola	Masukan			Target
	s1	s2	s0=1	t
1	1	1	1	1
2	1	0	1	-1
3	0	1	1	-1
4	0	0	1	-1



Arsitektur Jaringan Hebb

$w_i(\text{new}) = w_i(\text{old}) + \Delta w_i$ ,  
dengan  $\Delta w_i = \alpha x_i t$  ( $i=0, \dots, n$ )

$$net = \sum_{i=0}^n w_i x_i \quad \theta=0$$

$$y = \varphi(net) \begin{cases} +1 & \text{jika } net \geq \theta \\ 0 & \text{jika } net < \theta \end{cases}$$

Algoritma

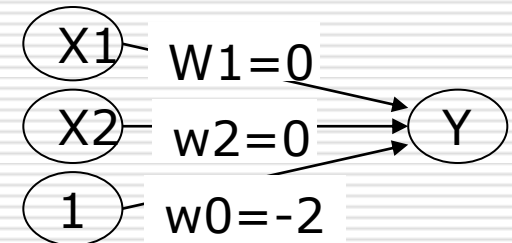
- Masukan biner
- Target keluaran biner



## b. Representasi Biner-Bipolar

Pola	Masukan			Target	Perubahan Bobot			Bobot Baru		
	x1	x2	x0=1	y=t	$\Delta w1$	$\Delta w2$	$\Delta w0$	w1	w2	b=w0
								0	0	0
1	1	1	1	1	1	1	1	1	1	1
2	1	0	1	-1	-1	0	-1	0	1	0
3	0	1	1	-1	0	-1	-1	0	0	-1
4	0	0	1	-1	0	0	-1	0	0	-2

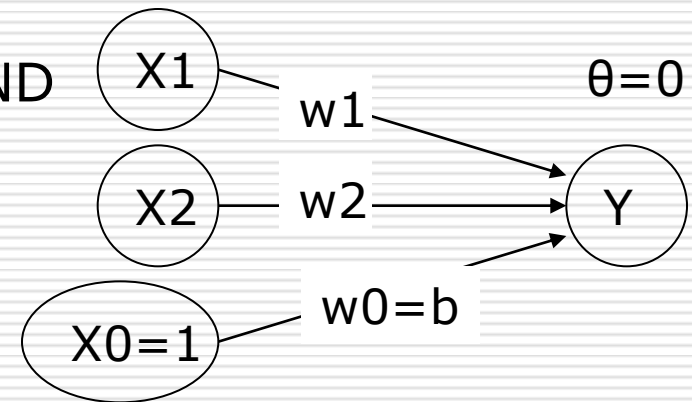
- Perhitungan bobot dan bias
  - $w_i(\text{new}) = w_i(\text{old}) + \Delta w_i$ , dimana  $\Delta w_i = \alpha x_i y$  ( $i=1, \dots, n$ )
- Hasil pembelajaran
  - Bobot berubah akibat pasangan data pertama saja
  - Dari hasil iterasi, diperoleh  $w1 = \dots$ ,  $w2 = \dots$ , dan  $w0 = \dots$



# c. Representasi Bipolar-Bipolar

Tabel Masukan dan Target fungsi AND

Pola	Masukan			Target
	s1	s2	s0=1	t
1	1	1	1	1
2	1	-1	1	-1
3	-1	1	1	-1
4	-1	-1	1	-1



Arsitektur Jaringan Hebb

$w_i(\text{new}) = w_i(\text{old}) + \Delta w_i$ ,  
dengan  $\Delta w_i = \alpha x_i t$  ( $i=0, \dots, n$ )

$$net = \sum_{i=0}^n w_i x_i \quad \theta=0$$

$$y = \varphi(net) \begin{cases} +1 & \text{jika } net \geq \theta \\ -1 & \text{jika } net < \theta \end{cases}$$

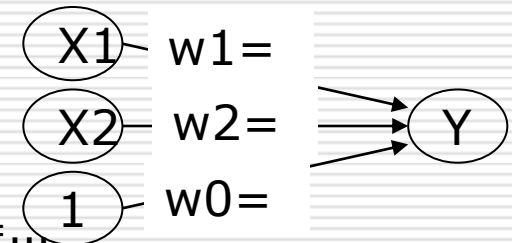
Algoritma

- Masukan bipolar
- Target keluaran bipolar

# c. Representasi Bipolar-Bipolar

Pola	Masukan			Target	Perubahan Bobot			Bobot Baru		
	x1	x2	x0=1	y=t	$\Delta w1$	$\Delta w2$	$\Delta b$	w1	w2	b=w0
								0	0	0
1	1	1	1	1						
2	1	-1	1	-1						
3	-1	1	1	-1						
4	-1	-1	1	-1						

- Perhitungan bobot dan bias
  - $w_i(\text{new}) = w_i(\text{old}) + \Delta w_i$ , dimana  $\Delta w_i = x_i t$  ( $i=1, \dots, n$ )
  - $b(\text{new}) = b(\text{old}) + \Delta b$ , dimana  $\Delta b = t$
- Hasil pembelajaran
  - Bobot berubah akibat pasangan data pertama saja
  - Dari hasil iterasi, diperoleh  $w1 = \dots$ ,  $w2 = \dots$ , dan  $w0 = \dots$

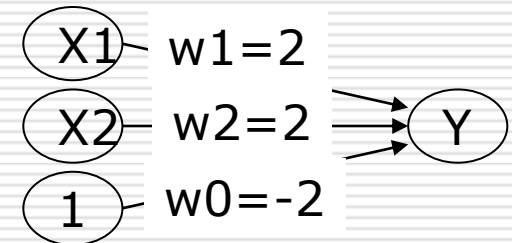


Algoritma

# c. Representasi Bipolar-Bipolar

Pola	Masukan			Target	Perubahan Bobot			Bobot Baru		
	x1	x2	x0=1	y=t	$\Delta w1$	$\Delta w2$	$\Delta w0$	w1	w2	w0
								0	0	0
1	1	1	1	1	1	1	1	1	1	1
2	1	-1	1	-1	-1	1	-1	0	2	0
3	-1	1	1	-1	1	-1	-1	1	1	-1
4	-1	-1	1	-1	1	1	-1	<b>2</b>	<b>2</b>	<b>-2</b>

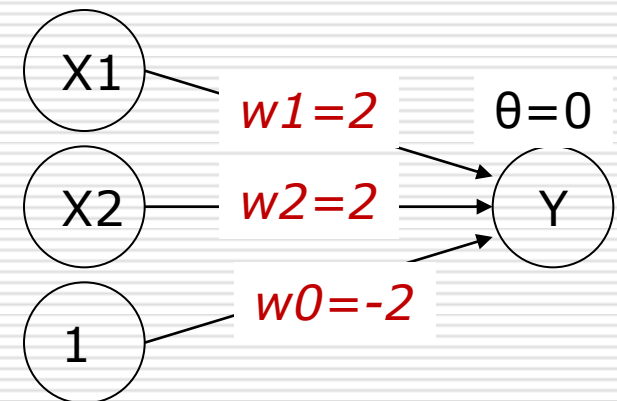
- Perhitungan bobot dan bias
  - $w_i(new) = w_i(old) + \Delta w_i$ , dimana  $\Delta w_i = x_i t$  ( $i=1 \dots n$ ).
  - $b(new) = b(old) + \Delta b$ , dimana  $\Delta b = t$ .
- Hasil pembelajaran
  - Dari hasil iterasi, diperoleh  $w1=2$ ,  $w2=2$ , dan  $b=-2$ .



# c. Representasi Bipolar-Bipolar

Pola	x1	x2	$net = \sum_{i=0}^2 x_i w_i$	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ -1 & \text{jika } net < 0 \end{cases}$
1	1	1	$1.2 + 1.2 + 1.(-2) = 2$	1
2	1	-1	$1.2 + (-1).2 + 1.(-2) = -2$	-1
3	-1	1	$(-1).2 + 1.2 + 1.(-2) = -2$	-1
4	-1	-1	$(-1).2 + (-1).2 + 1.(-2) = -6$	-1

- ❑  $f(net)$  sama dengan target yang diinginkan pada fungsi AND
- ❑ Jaringan SUDAH DAPAT 'MENGETI' pola yang dimaksud



# Kasus 2 : Pengenalan Pola

---

Diberikan 2 pola menyerupai huruf X dan O

```
# . . . #  
. # . # .  
. . # . .  
. # . # .  
# . . . #
```

POLA 1

```
. # # # .  
# . . . #  
# . . . #  
# . . . #  
. # # # .
```

POLA 2

# Representasi Kasus

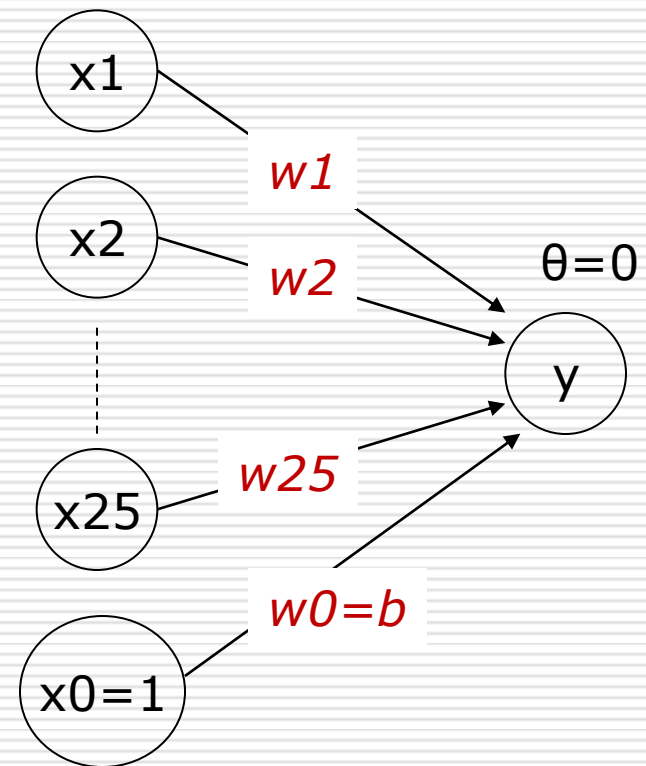
- Setiap karakter pola dianggap sebagai sebuah unit masukan
  - Karakter “#” diberi nilai 1, karakter “.” diberi nilai -1 (minus 1)

	1	2	3	4	5						
1	#	.	.	.	#	→	1	-1	-1	-1	1
2	.	#	.	#	.		-1	1	-1	1	-1
3	.	.	#	.	.		-1	-1	1	-1	-1
4	.	#	.	#	.		-1	1	-1	1	-1
5	#	.	.	.	#		1	-1	-1	-1	1

- Pola terdiri dari 5 baris dan 5 kolom
  - Jaringan Hebb terdiri dari 25 unit masukan (x1 s/d x25) dan sebuah bias bernilai = 1
- Target
  - Keluaran jaringan bernilai 1 jika diberi masukan pola-1 (X) dan bernilai -1 (minus 1) jika diberi masukan pola-2 (O)

# Representasi Masukan dan Target

<i>Pola</i>	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>	<i>x5</i>	<i>t</i>
	<i>x6</i>	<i>x7</i>	<i>x8</i>	<i>x9</i>	<i>x10</i>	
	<i>x11</i>	<i>x12</i>	<i>x13</i>	<i>x14</i>	<i>x15</i>	
	<i>x16</i>	<i>x17</i>	<i>x18</i>	<i>x19</i>	<i>x20</i>	
	<i>x21</i>	<i>x22</i>	<i>x23</i>	<i>x24</i>	<i>x25</i>	
	<i>x26</i>	<i>x27</i>	<i>x28</i>	<i>x29</i>	<i>x30</i>	
<b>1</b>	<b>1</b>	-1	-1	-1	<b>1</b>	<b>1</b>
	-1	<b>1</b>	-1	<b>1</b>	-1	
	-1	-1	<b>1</b>	-1	-1	
	-1	<b>1</b>	-1	<b>1</b>	-1	
	<b>1</b>	-1	-1	-1	<b>1</b>	
<b>2</b>	-1	<b>1</b>	<b>1</b>	<b>1</b>	-1	<b>-1</b>
	<b>1</b>	-1	-1	-1	<b>1</b>	
	<b>1</b>	-1	-1	-1	<b>1</b>	
	<b>1</b>	-1	-1	-1	<b>1</b>	
	-1	<b>1</b>	<b>1</b>	<b>1</b>	-1	



Arsitektur Jaringan Hebb



# Pembelajaran Jaringan

Masukan		Target	Perubahan Bobot		Bobot Baru	
x1 ... x25	x0=1	y=t	$\Delta w_1 \dots \Delta w_{25}$	$\Delta w_0$	w1 s/d w25	w0
Pola 1					0 0 0 0 0 ... 0 0 0 0 0	0
	1	1	1 -1 -1 -1 1	1	1 -1 -1 -1 1	1
	-1 1 -1 1 -1		-1 1 -1 1 -1		-1 1 -1 1 -1	
	-1 -1 1 -1 -1		-1 -1 1 -1 -1		-1 -1 1 -1 -1	
	-1 1 -1 1 -1		-1 1 -1 1 -1		-1 1 -1 1 -1	
	1 -1 -1 -1 1		1 -1 -1 -1 1		1 -1 -1 -1 1	
	1	-1	1 -1 -1 -1 1	-1	2 -2 -2 -2 2	0
	1 -1 -1 -1 1		-1 1 1 1 -1		-2 2 0 2 -2	
	1 -1 -1 -1 1		-1 1 1 1 -1		-2 0 2 0 -2	
	1 -1 -1 -1 1		-1 1 1 1 -1		-2 2 0 2 -2	
	-1 1 1 1 -1		1 -1 -1 -1 1		2 -2 -2 -2 2	

- $w_i(new) = w_i(old) + \Delta w_i$ , dimana  $\Delta w_i = \alpha \cdot x_i \cdot y$  ( $i=0, \dots, n$ ;  $x_0=+1$ )

# Hasil Penggunaan (bobot (terakhir) hasil belajar)

Pola	x1 s/d x25	$net = \sum_{i=0}^{25} x_i w_i$	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ -1 & \text{jika } net < 0 \end{cases}$
<b>1</b>	<b>1</b> -1 -1 -1 <b>1</b> -1 <b>1</b> -1 <b>1</b> -1 -1 -1 <b>1</b> -1 -1 -1 <b>1</b> -1 <b>1</b> -1 <b>1</b> -1 -1 -1 <b>1</b>	$1.2 + (-1).(-2) + \dots + 1.2 = 42$	<b>1</b>
<b>2</b>	-1 <b>1</b> <b>1</b> <b>1</b> -1 <b>1</b> -1 -1 -1 <b>1</b> <b>1</b> -1 -1 -1 <b>1</b> <b>1</b> -1 -1 -1 <b>1</b> -1 <b>1</b> <b>1</b> <b>1</b> -1	$(-1).2 + 1.(-2) + \dots + (-1).2 = -42$	<b>-1</b>

- ☐ Keluaran jaringan sama dengan target yang diinginkan
- ☐ Jaringan DAPAT MENGENALI POLA

# Pengenalan Pola

---

Diberikan 3 pola lain

.	.	.	.	.	.	#	#	#	.	#	.	.	.	#
.	#	.	#	.	.	.	#	.	.	#	.	.	.	#
.	.	#	.	.	.	.	#	.	.	#	#	#	#	#
.	#	.	#	.	.	.	#	.	.	#	.	.	.	#
.	.	.	.	.	.	#	#	#	.	#	.	.	.	#

POLA 3

POLA 4

POLA 5

# Hasil Penggunaan

Pola	x1 ... x25	$net = \sum_{i=0}^{25} x_i w_i$	$y = f(net) = \begin{cases} 1 & \text{jika } net \geq 0 \\ -1 & \text{jika } net < 0 \end{cases}$
x (pola 3)	-1 -1 -1 -1 -1 -1 <b>1</b> -1 <b>1</b> -1 -1 -1 <b>1</b> -1 -1 -1 <b>1</b> -1 <b>1</b> -1 -1 -1 -1 -1 -1	$(-1).2 + (-1).(-2) + \dots + (-1).2 + 0$ $= 26$	<b>1</b>
I (pola 4)	-1 <b>1</b> <b>1</b> <b>1</b> -1 -1 -1 <b>1</b> -1 -1 -1 -1 <b>1</b> -1 -1 -1 -1 <b>1</b> -1 -1 -1 <b>1</b> <b>1</b> <b>1</b> -1	$(-1).2 + 1.(-2) + \dots + (-1).2 + 0$ $= -22$	<b>-1</b>
H (pola 5)	<b>1</b> -1 -1 -1 <b>1</b> <b>1</b> -1 -1 -1 <b>1</b> <b>1</b> <b>1</b> <b>1</b> <b>1</b> <b>1</b> <b>1</b> -1 -1 -1 <b>1</b> <b>1</b> -1 -1 -1 <b>1</b>	$1.2 + (-1).(-2) + \dots + 1.2 + 0 =$ $10$	<b>1</b>

# Hasil Penggunaan

---

- ❑ Dari hasil penggunaan hasil pembelajaran JST dengan 2-pola (Pola-1 dan Pola-2), kemudian digunakan untuk mengenali 5-pola, yaitu: Pola-1, Pola-2, Pola-3, Pola-4, dan Pola-5, didapat:
- ❑ Pola-1, termasuk kelompok pola yang dimaksudkan.
- ❑ Pola-2, bukan termasuk
- ❑ Pola-3, termasuk kelompok
- ❑ Pola-4, bukan termasuk
- ❑ Pola-5, termasuk kelompok

# Tugas

1. Buat JST Hebb pengenalan pola untuk mengenali Pola Karakter **Angka Terakhir NIM** anda.
  - Ukuran karakter 5x5 pixel,
  - Representasi input-output: Bipolar-Bipolar,
  - Fungsi aktivasi adalah Signum Function dengan nilai threshold,  $\theta = 0$ ,
  - Algoritma pembelajaran yang digunakan adalah extended-Hebb,
  - Pola karakter (satu angka terakhir NIM) yang betul untuk latihan, boleh lebih dari 1-pola,
  - Pola karakter yang salah untuk latihan, boleh banyak pola dan banyak angka (selain angka terakhir NIM),
  - Buat dan tunjukkan tabel pelatihannya.
2. Gunakan JST Hebb Pengenalan Pola yang terbentuk tadi untuk menilai karakter 0,1,2,...,9.
  - Buat dan tunjukkan tabel penggunaanya.

# Pustaka Acuan

---

- Jong Jek Siang, Drs. M.Sc., *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*, ANDI, Jogjakarta, 2005.
- Laurene Fausett, *Fundamentals Of Neural Networks: Architectures, Algorithms, and Applications*, Prentice-Hall, New Jersey, 1994.
- Simon Haykin, *Neural Networks: A Comprehensive Foundation*, 2<sup>nd</sup>, Prentice-Hall, New Jersey, 1999.

# Sekian

---

**Agust Isa Martinus**

[aimxx@yahoo.com](mailto:aimxx@yahoo.com)

<http://www.GusMartinus.mine.nu>

**BACK**

**Agust Isa Martinus @2008**

**<aimxx@yahoo.com>**