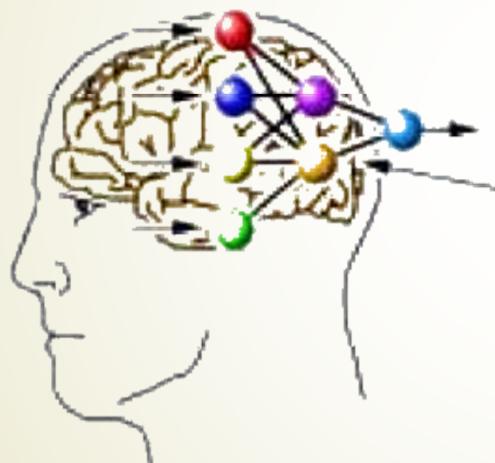
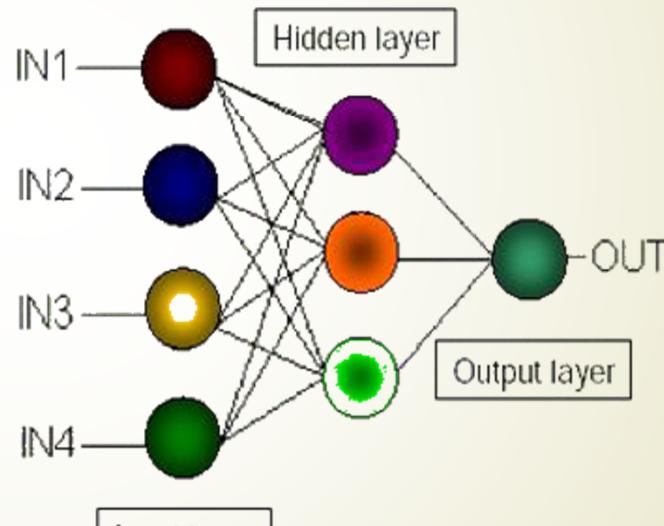


Jaringan Syaraf Tiruan

(a) Otak manusia



(b) Artificial Neural Network (ANN)



Institution
GARUDA CYBER INDONESIA

By : Sendi Novianto



What's in ?

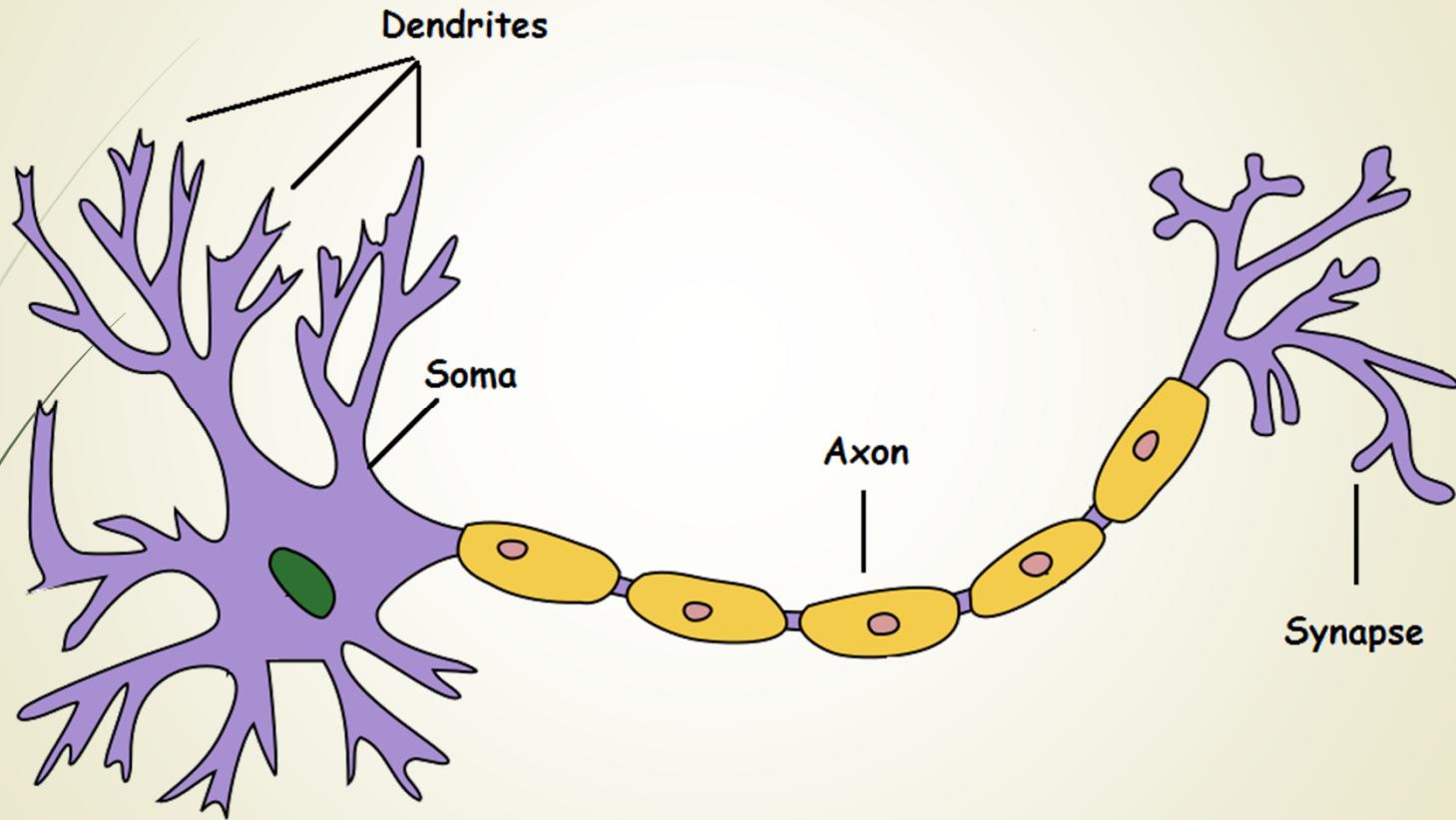
- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron



What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Sel Syaraf (Neuron)



Sel Syaraf (Neuron)

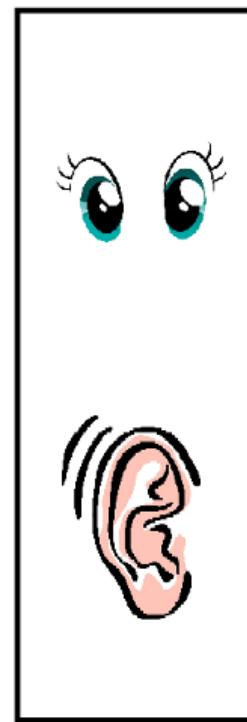


Siapakah Nama Orang INI ?

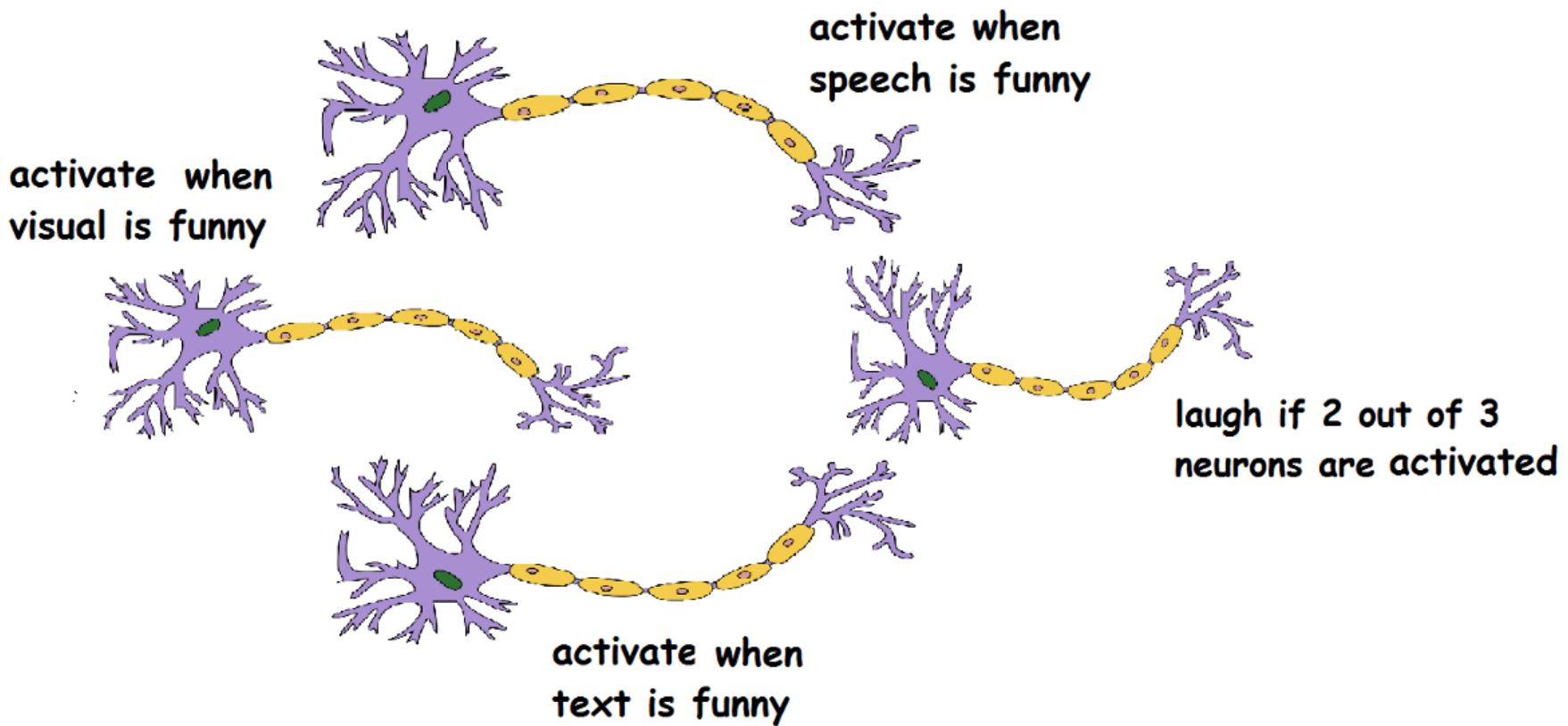
Sel Syaraf (Neuron)



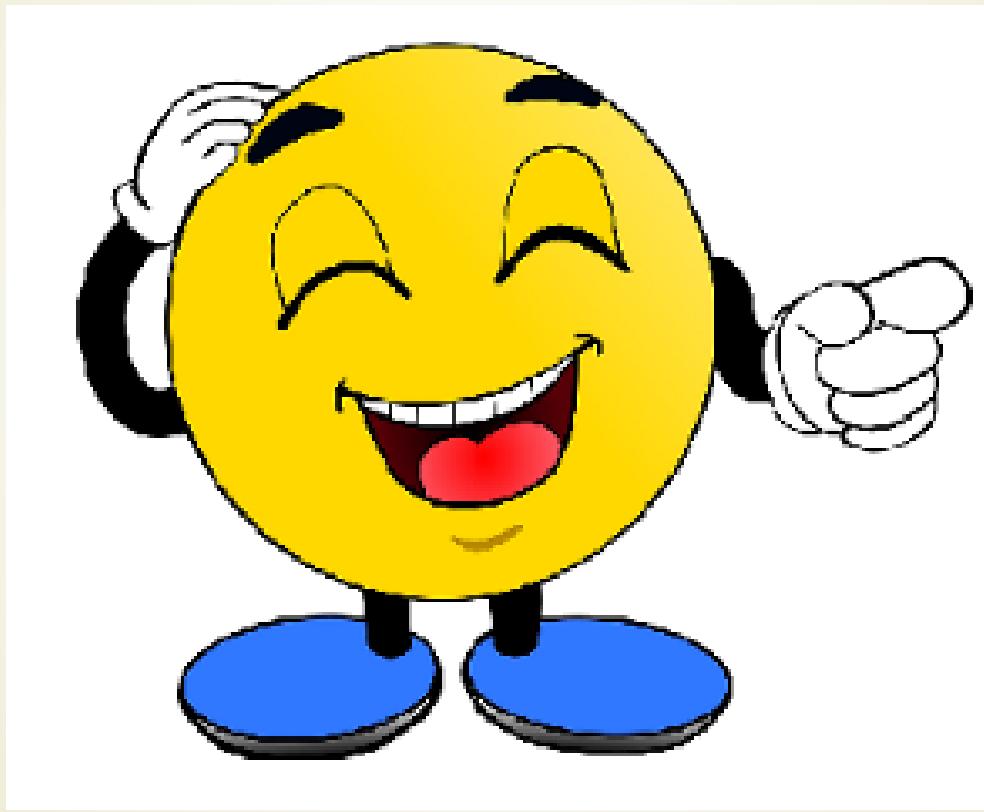
Siapakah Nama Orang INI ?



Sel Syaraf (Neuron)



Sel Syaraf (Neuron)

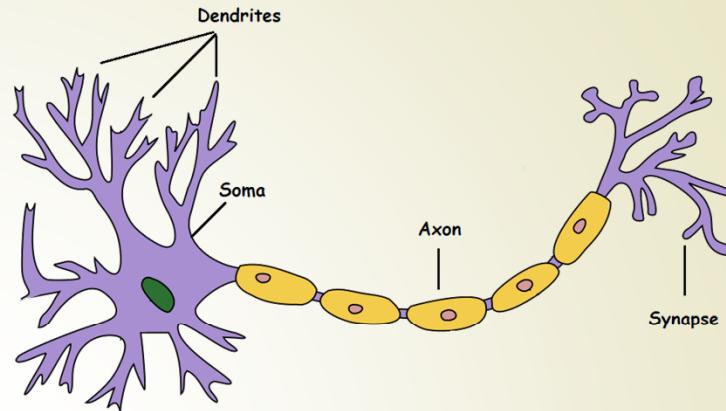




What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Analogi JST Vs JSB



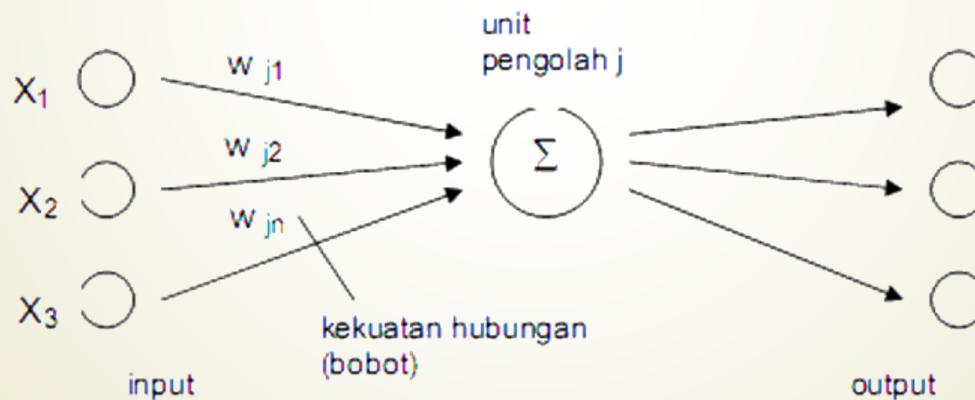
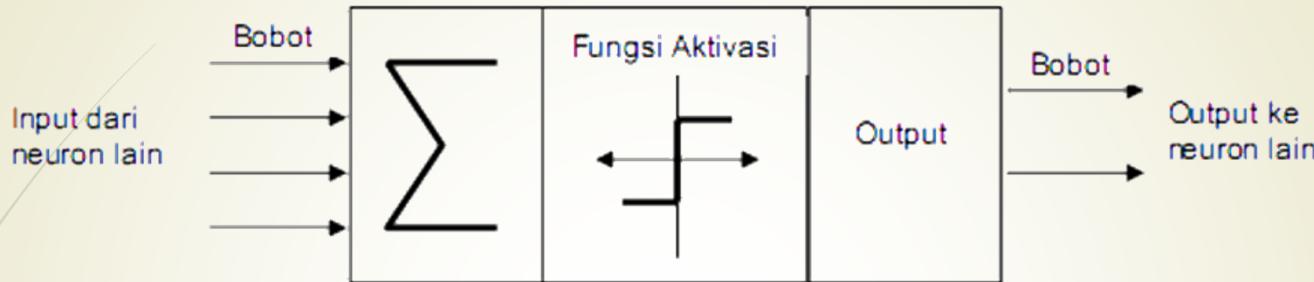
JST	JS Biologis
Node / input	Badan sel (soma)
Input	Dendrit
Output	Akson
Bobot	Sinapsis



What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Model Struktur Neuron JST

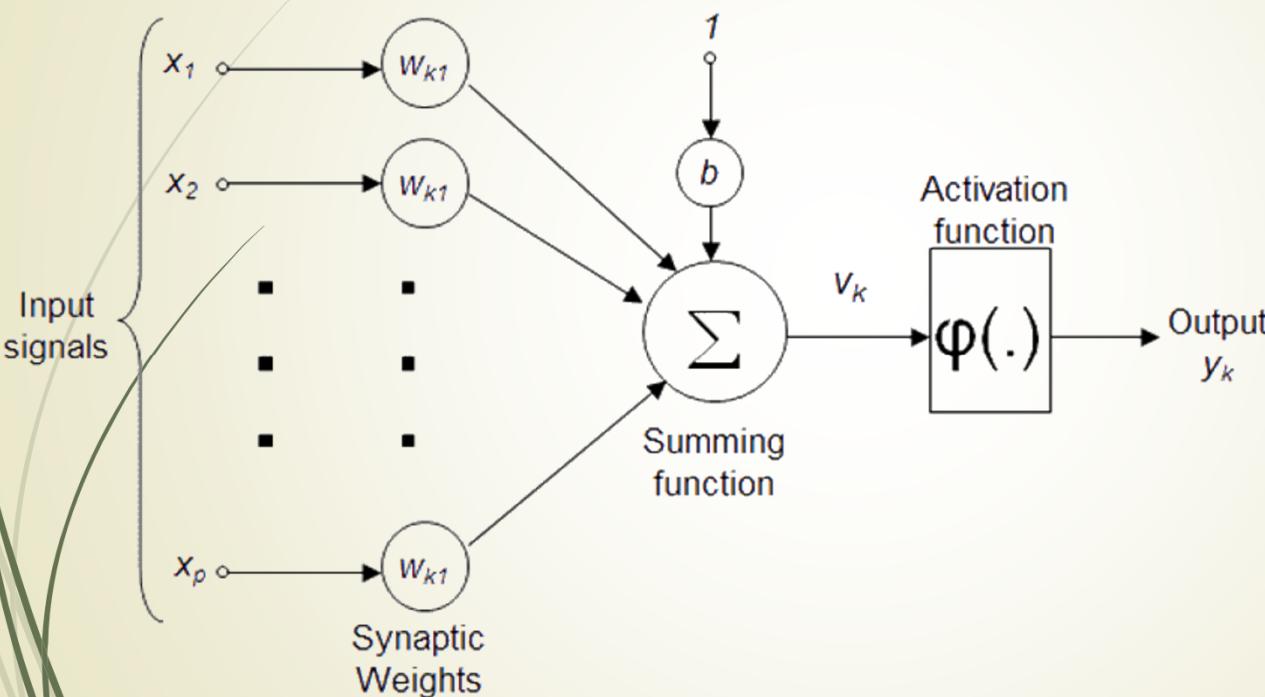




What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Model Sel Syaraf (Neuron)



$$v_k = \sum_{j=0}^p w_{kj} x_j$$

$$y_k = \varphi(v_k)$$

Summation Function

- Fungsi yang digunakan untuk mencari rata-rata bobot dari semua elemen input.
- Bentuk sederhananya adalah dengan mengalikan setiap nilai input (X_j) dengan bobotnya (W_{ij}) dan menjumlahkannya

$$v_k = \sum_{j=0}^p w_{kj} x_j$$

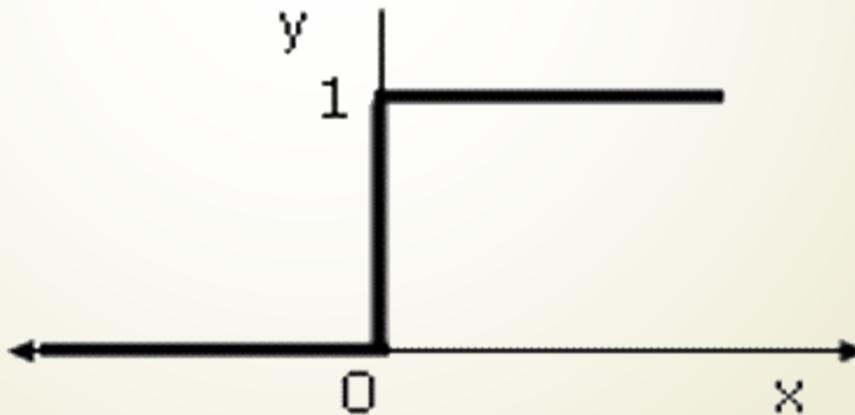
Fungsi Aktivasi

- Fungsi Hard Limit
- Fungsi Threshold
- Fungsi Bipolar Hard Limit
- Fungsi Bipolar Threshold
- Fungsi Linear (Identitas)
- Fungsi Sigmoid

Fungsi Aktivasi

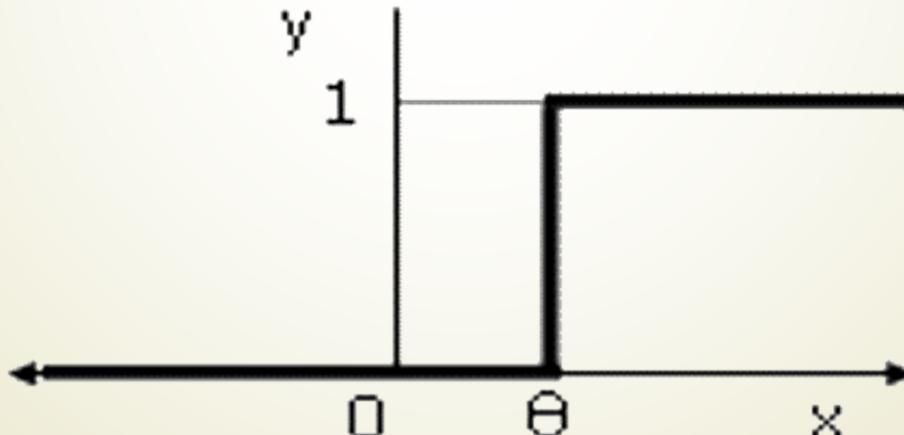
Hard Limit

$$y = \begin{cases} 0, & \text{jika } x \leq 0 \\ 1, & \text{jika } x > 0 \end{cases}$$



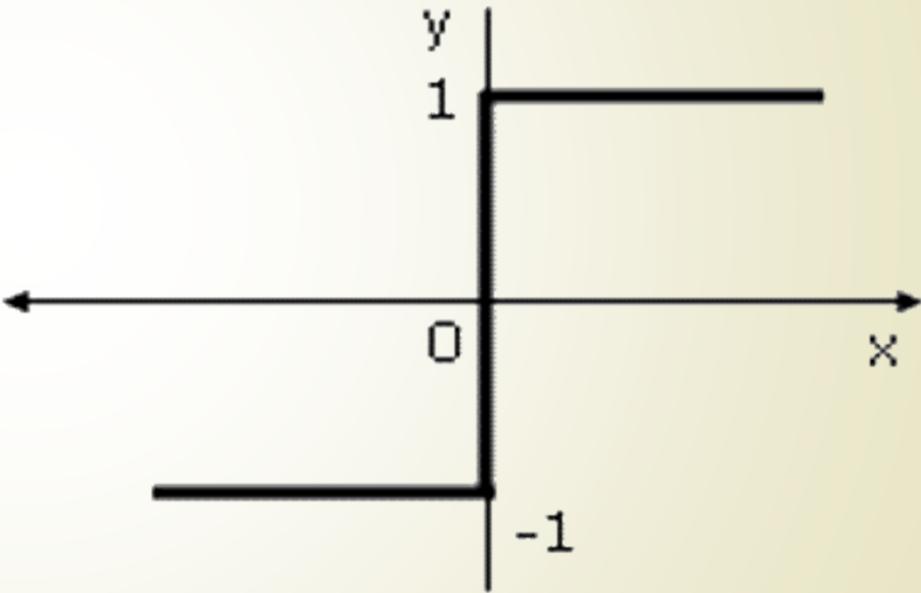
Fungsi Aktivasi Threshold

$$y = \begin{cases} 0, & \text{jika } x < \theta \\ 1, & \text{jika } x \geq \theta \end{cases}$$



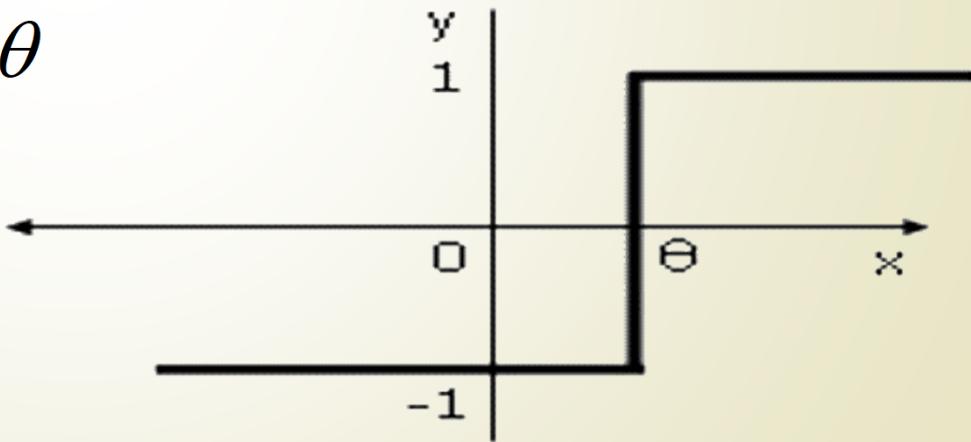
Fungsi Aktivasi Bipolar Hard Limit

$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases}$$



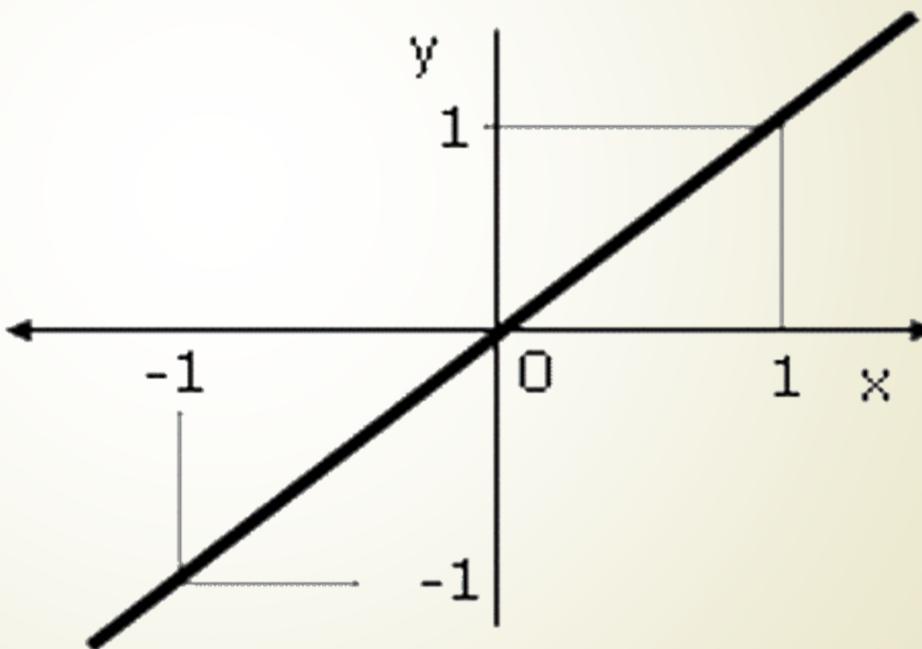
Fungsi Aktivasi Bipolar Threshold

$$y = \begin{cases} 1, & \text{jika } x > \theta \\ 0, & \text{jika } x = \theta \\ -1, & \text{jika } x < \theta \end{cases}$$



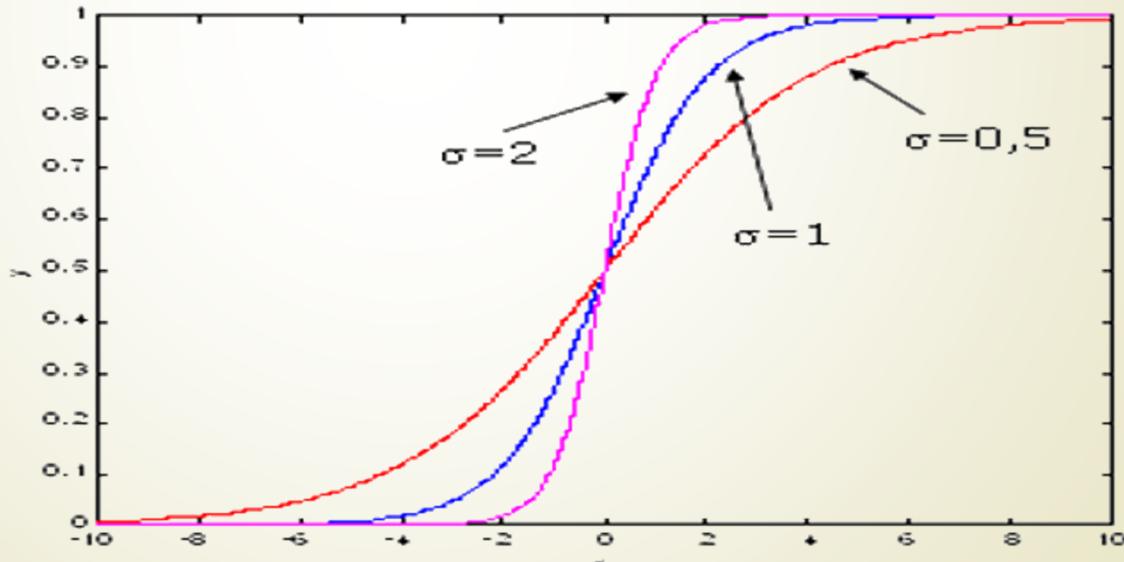
Fungsi Aktivasi Linear (Identitas)

$$y = x$$



Fungsi Aktivasi Sigmoid

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}}$$



Karakteristik JST

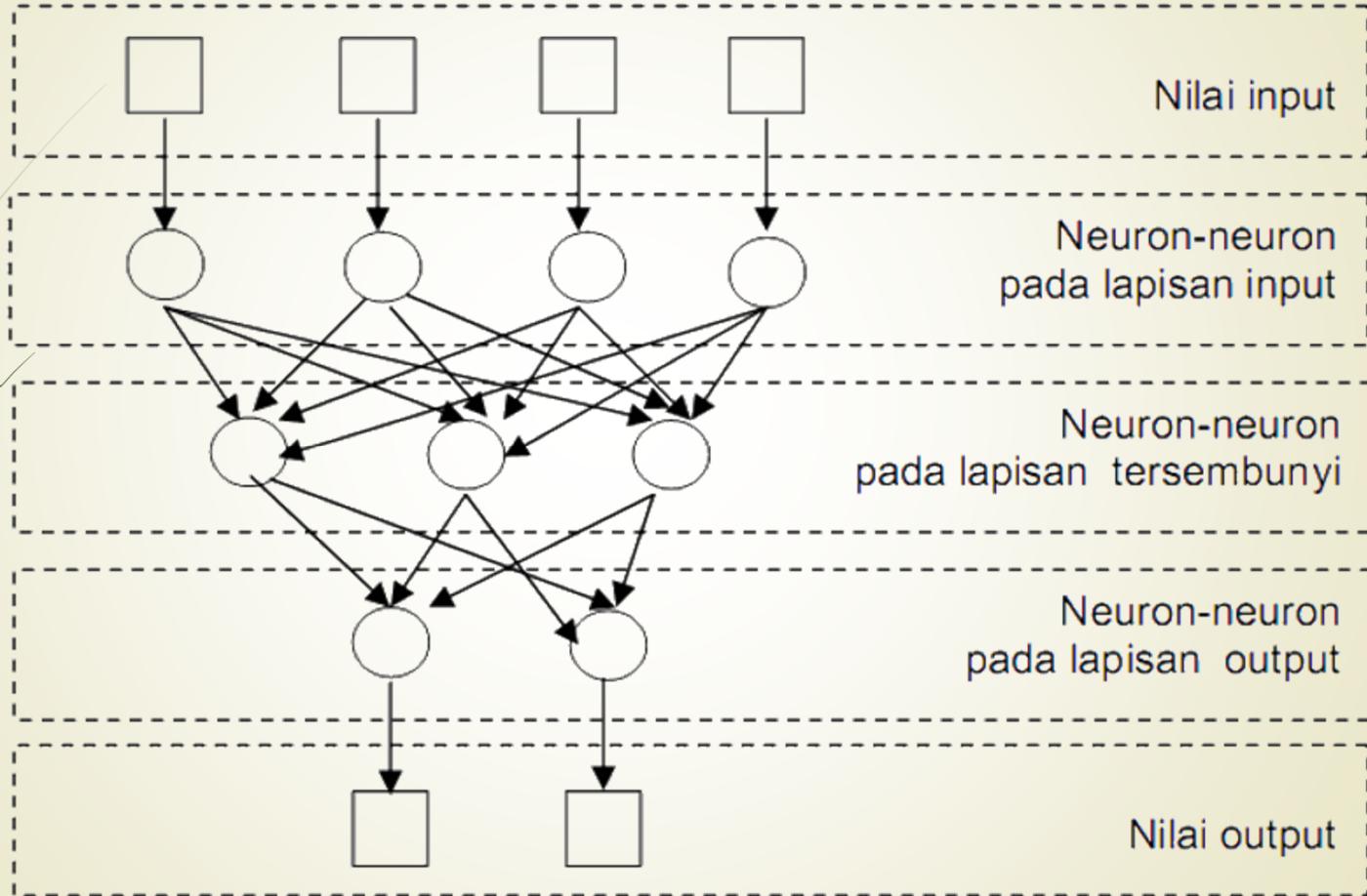
- Dapat belajar dari pengalaman
- Algoritma untuk JST beroperasi secara langsung dengan angka sehingga data yang tidak numerik harus diubah menjadi data numerik.
- JST tidak diprogram untuk menghasilkan keluaran tertentu. Semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran.
- Pada proses pembelajaran, ke dalam JST dimasukkan pola-pola input (dan output) lalu jaringan akan diajari untuk memberikan jawaban yang bisa diterima.



What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Arsitektur JST

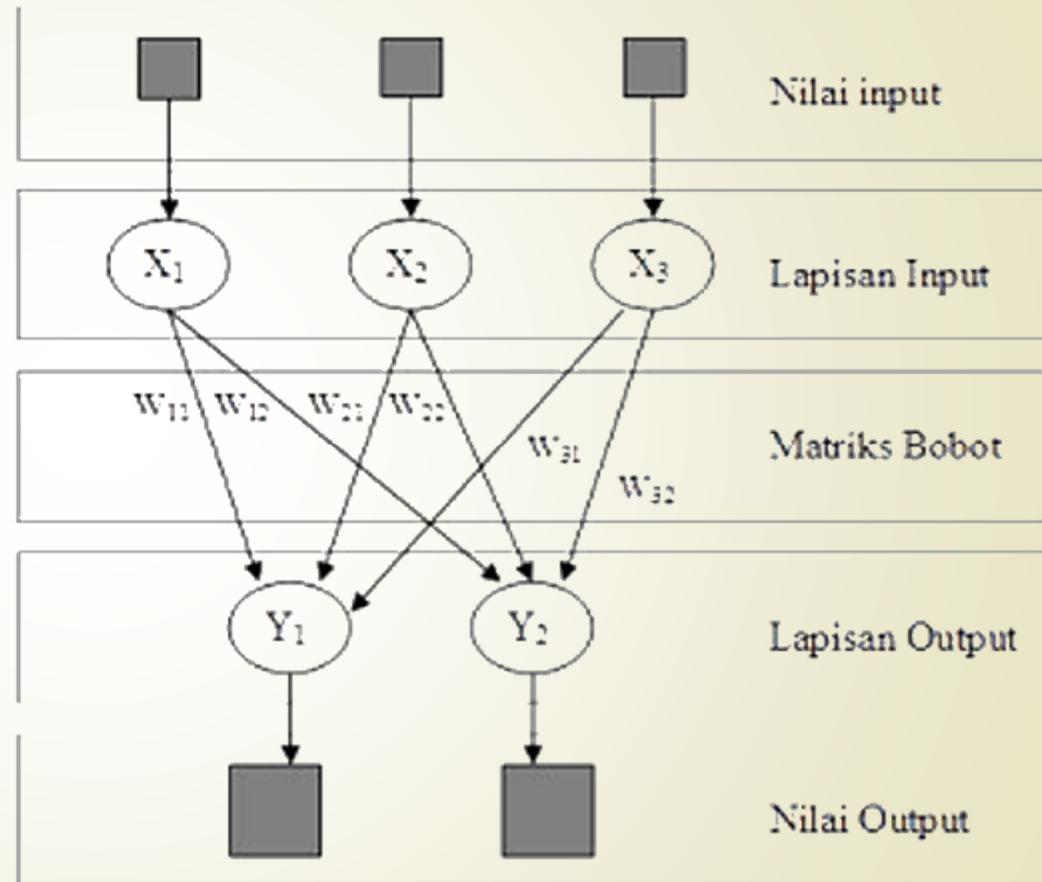


Arsitektur JST

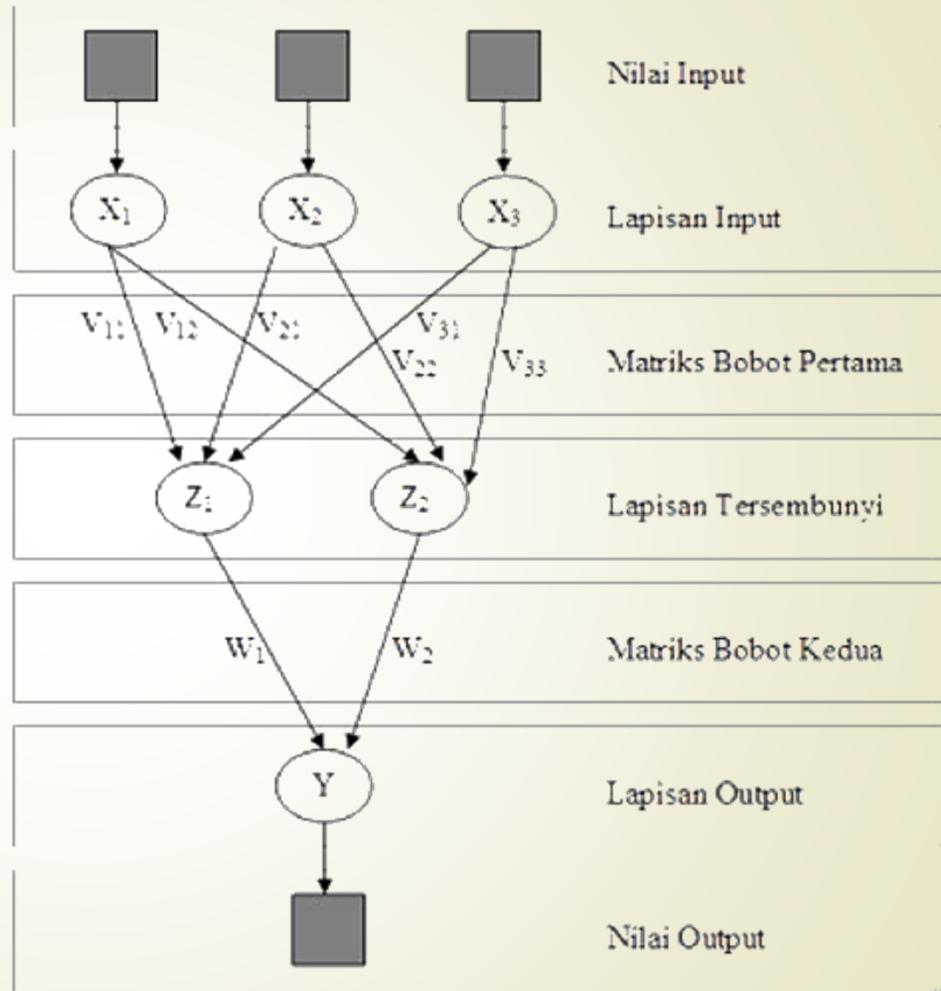
- Ada beberapa arsitektur jaringan syaraf, antara lain:
 - Jaringan dengan lapisan tunggal (single layer net)
 - Jaringan dengan banyak lapisan (multilayer net)
 - Jaringan dengan lapisan kompetitif (competitive net)

Arsitektur JST

Single Layer Net

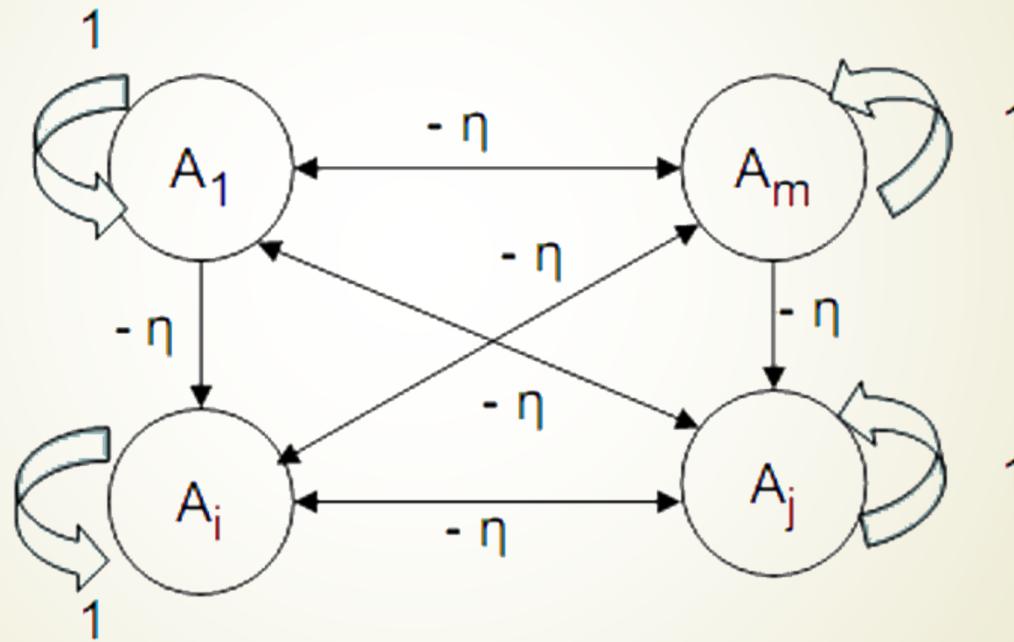


Arsitektur JST Multi Layer Net



Arsitektur JST

Competitive Net



Proses Pembelajaran Jaringan

- ◆ Kemampuan memorisasi = kemampuan JST untuk memanggil kembali secara sempurna sebuah pola yang telah dipelajari.
- ◆ Kemampuan generalisasi = adalah kemampuan JST untuk menghasilkan respon yang bisa diterima terhadap pola-pola input yang serupa (namun tidak identik) dengan pola-pola yang sebelumnya telah dipelajari.



What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Metode Pembelajaran JST

- ▶ Berdasarkan cara memodifikasi bobotnya ada 2 macam pembelajaran :
 - ◆ Pembelajaran terawasi (supervised learning), contoh : Hebb, perceptron, back propagation
 - ◆ Pembelajaran tak terawasi (unsupervised learning), contoh : LVQ

Metode Pembelajaran JST

Terawasi

- Output yang diharapkan telah diketahui sebelumnya
- Contoh : JST untuk mengenali pasangan pola, misalkan pada operasi AND

Input		Target
0	0	0
0	1	0
1	0	0
1	1	1

Metode Pembelajaran JST

Tak Terawasi

- Tidak memerlukan target output
- Tidak dapat ditentukan hasil yang diharapkan selama proses pembelajaran
- Nilai bobot disusun dalam suatu range tertentu tergantung nilai input yang diberikan
- Tujuannya untuk mengelompokkan unit yang hampir sama dalam suatu area tertentu



What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Aplikasi Dengan JST

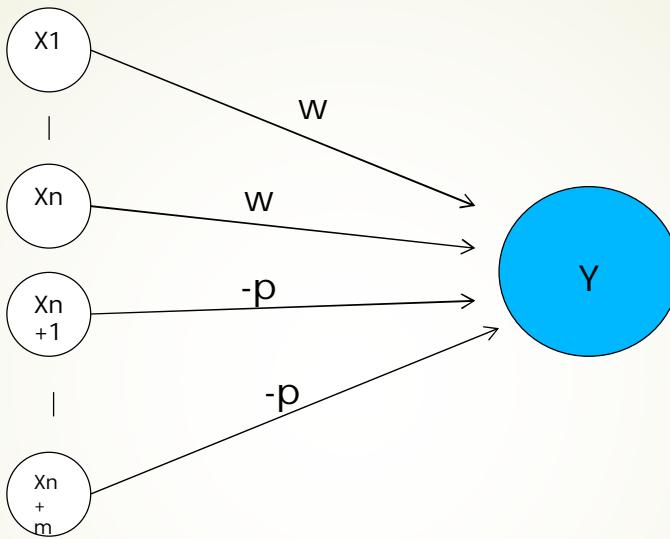
- a. Klasifikasi
- b. Pengenalan pola
- c. Peramalan
- d. Optimisasi



What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Model McCulloch-Pitts



$$f(\text{net}) = \begin{cases} 0, & \text{jika } \text{net} < \theta \\ 1, & \text{jika } \text{net} \geq \theta \end{cases}$$

Model McCulloch-Pitts

Buatlah model neuron McCulloch-Pitts untuk mengenali pola fungsi logika “AND” sesuai tabel kebenaran berikut :

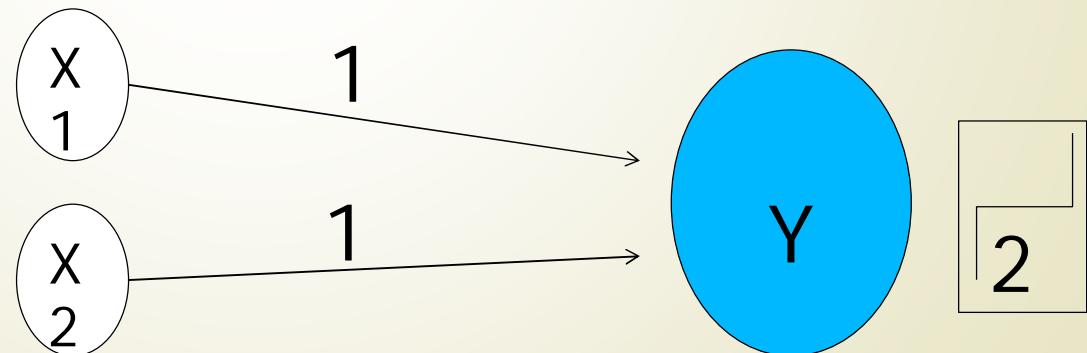
X ₁	X ₂	Y
0	0	0
0	1	0
1	0	0
1	1	1

Model McCulloch-Pitts

- ▶ Model yang pertama ditemukan dengan karakteristik :
 - ▶ 1. Fungsi aktivasi biner threshold
 - ▶ 2. Semua bobot positif (excitatory) memiliki nilai yang sama
 - ▶ 3. Tiap neuron punya threshold yang sama
- ▶ Kelemahan : penentuan bobot secara analitik / trial and error secara manual, sulit untuk masalah yang kompleks

Model McCulloch-Pitts

X ₁	X ₂	Y
0	0	0
0	1	0
1	0	0
1	1	1



2

Model McCulloch-Pitts

terdapat dua input yaitu x_1 dan x_2 dan satu output y . Bila nilai bobot w_1 dan w_2 dibuat sama dengan 1, ($w_1 = 1$ dan $w_2 = 1$), maka kita bisa menghitung jumlah seluruh input yang masuk untuk tiap-tiap data sebagai berikut :

X_1	X_2	$net = \sum x_i w$
0	0	$0.1 + 0.1 = 0$
0	1	$0.1 + 1.1 = 1$
1	0	$1.1 + 0.1 = 1$
1	1	$1.1 + 1.1 = 2$

Model McCulloch-Pitts

nilai bobot w dan nilai ambang θ dapat diubah-ubah sesuai keinginan supaya sesuai dengan yang diharapkan (trial and error)

$$f(\text{net}) = \begin{cases} 0, & \text{jika } \text{net} < \theta \\ 1, & \text{jika } \text{net} \geq \theta \end{cases}$$

X ₁	X ₂	Y
0	0	0
0	1	0
1	0	0
1	1	1

X ₁	X ₂	net = $\sum x_i w$	y(net) = $\begin{cases} 0 & \text{jika } \text{net} < 2 \\ 1 & \text{jika } \text{net} \geq 2 \end{cases}$
0	0	0.1 + 0.1 = 0	0
0	1	0.1 + 1.1 = 1	0
1	0	1.1 + 0.1 = 1	0
1	1	1.1 + 1.1 = 2	1

Jika y(net)=Y
Maka -> (Nilai W
dan T sudah benar)

Jika tidak sama
maka -> (Nilai W
dan T harus
dikoreksi)



What's in ?

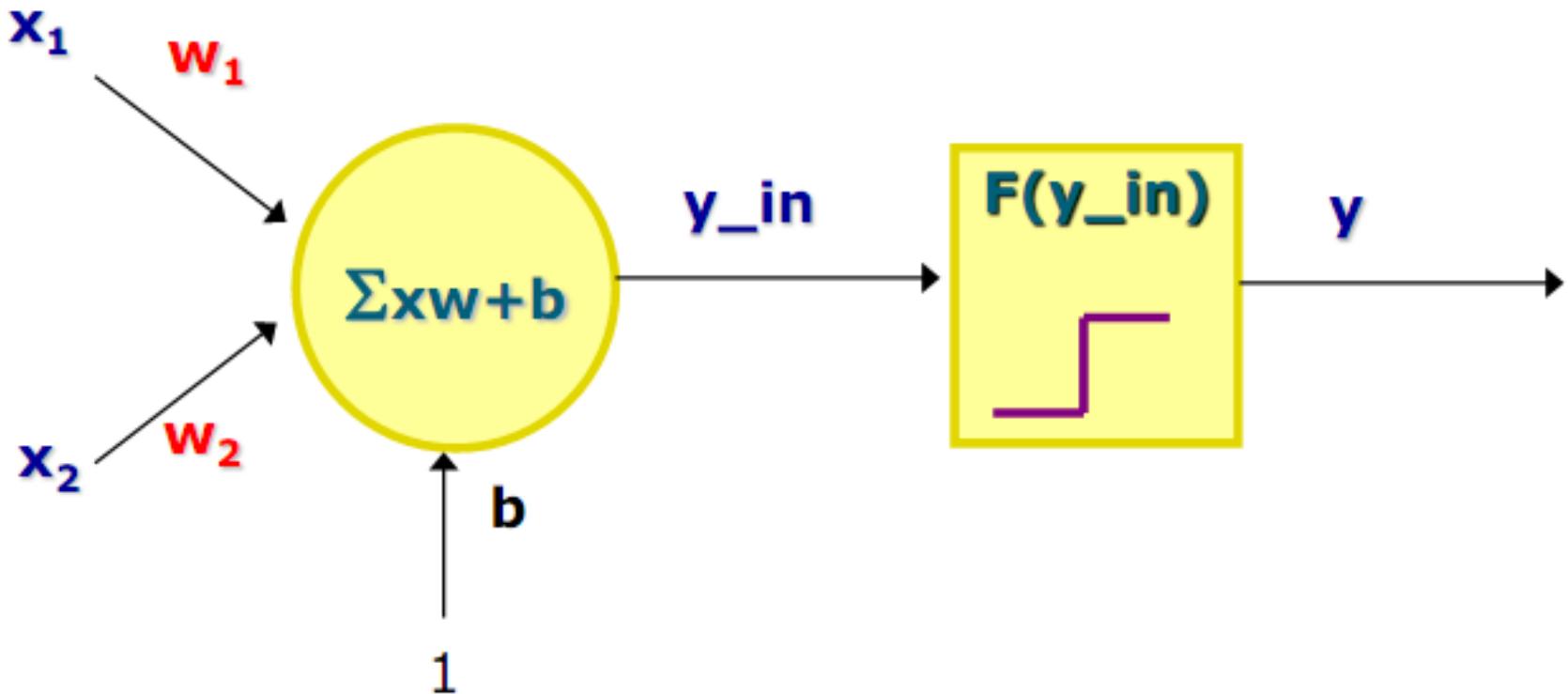
- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Model Hebb

► Langkah-langkah :

1. Inisialisasi semua bobot = $W_i = 0$ ($i=1,\dots,n$)
2. Untuk semua vektor input s dan unit target t , lakukan :
 - Set aktivasi unit masukan $X_i = S_i$ ($i=1,\dots,n$)
 - Set aktivasi unit keluaran $y = \text{target}$
 - Perbaiki bobot menurut persamaan
 - $W_i (\text{baru}) = W_i(\text{lama}) + \Delta W$
 - $\Delta W = X_i \cdot y$
 - Perbaiki bias menurut persamaan :
 - $b(\text{baru}) = b(\text{lama}) + y$

Model Hebb



Model Hebb

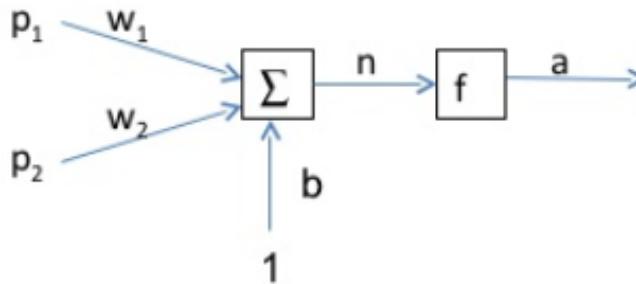
Buat jaringan Hebb untuk menyatakan fungsi logika AND jika representasi yang dipakai adalah :

- a). Masukan dan keluaran biner
- b). Masukan biner dan keluaran bipolar
- c). Masukan dan keluaran bipolar

Model Hebb – masukan dan keluaran biner

a) Pola hubungan masukan-target :

Masukan		Target	
p_1	p_2	1	t
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1



Model Hebb – masukan dan keluaran biner

Hasil pelatihan :

Masukan			Target	Perubahan bobot $\Delta w = p_i t$ $\Delta b = t$		Bobot baru		
p_1	p_2	1	t	Δw_1	Δb	$w_{baru} = w_{lama} + \Delta w$	$b_{baru} = b_{lama} + \Delta b$	
						$w_1 = 1, w_2 = 1, b = 1$		
Inisiasi								
0	0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1

$$w_1 = 1, w_2 = 1, b = 1$$

Model Hebb – masukan dan keluaran biner

Hasil akhir :

$$\text{net} = \sum_{i=1}^2 x_i w_i + b$$

$$w_1 = 1, w_2 = 1, b = 1$$

p_1	p_2	$n = p_1 w_1 + p_2 w_2 + b$	$a = f(n)$
0	0	$0.1+0.1+1 = 1$	1
0	1	$0.1+1.1+1 = 2$	1
1	0	$1.1+0.1+1 = 2$	1
1	1	$1.1+1.1+1 = 3$	1

$$f(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

Masukan		Target	
p_1	p_2	1	t
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1

Keluaran \neq target \rightarrow Jaringan Hebb tidak dapat ‘mengerti’ pola yang dimaksud

Model Hebb – masukan biner keluaran bipolar

b) Pola hubungan masukan-target :

Masukan		Target	
p_1	p_2	1	t
0	0	1	-1
0	1	1	-1
1	0	1	-1
1	1	1	1

Model Hebb – masukan biner keluaran bipolar

Hasil pelatihan :

Masukan			Target	Perubahan bobot $\Delta w = p_i t$ $\Delta b = t$			Bobot baru				
p_1	p_2	1	t	Δw_1	Δw_2	Δb	w_1	w_2	b		
			Inisiasi			0 0 0			0 0 0		
0	0	1	-1	0	0	-1	0	0	-1		
0	1	1	-1	0	-1	-1	0	-1	-2		
1	0	1	-1	-1	0	-1	-1	-1	-3		
1	1	1	1	1	1	1	0	0	-2		

$$w_1 = 0, w_2 = 0, b = -2$$

Model Hebb – masukan biner keluaran bipolar

Hasil akhir :

p_1	p_2	$n = p_1 w_1 + p_2 w_2 + b$	$a = f(n)$
0	0	$0.0+0.0 - 2 = -2$	-1
0	1	$0.0+1.0 - 2 = -2$	-1
1	0	$1.0+0.0 - 2 = -2$	-1
1	1	$1.0+1.0 - 2 = -2$	-1

$$w_1 = 0, w_2 = 0, b = -2$$

$$f(n) = \begin{cases} 1, & n \geq 0 \\ -1, & n < 0 \end{cases}$$

Masukan		Target	
p_1	p_2	1	t
0	0	1	-1
0	1	1	-1
1	0	1	-1
1	1	1	1

Keluaran \neq target \rightarrow Jaringan Hebb tidak dapat ‘mengerti’ pola yang dimaksud

Model Hebb – masukan dan keluaran bipolar

c) Pola hubungan masukan-target :

Masukan		Target	
p_1	p_2	1	t
-1	-1	1	-1
-1	1	1	-1
1	-1	1	-1
1	1	1	1

Model Hebb – masukan dan keluaran bipolar

Hasil pelatihan :

Masukan			Target	Perubahan bobot $\Delta w = p_i t$ $\Delta b = t$			Bobot baru		
p_1	p_2	1	t	Δw_1	Δw_2	Δb	w_1	w_2	b
Inisiasi							0	0	0
-1	-1	1	-1	1	1	-1	1	1	-1
-1	1	1	-1	1	-1	-1	2	0	-2
1	-1	1	-1	-1	1	-1	1	1	-3
1	1	1	1	1	1	1	2	2	-2

$$w_1 = 2, w_2 = 2, b = -2$$

Model Hebb – masukan dan keluaran bipolar

$$w_1 = 2, w_2 = 2, b = -2$$

Hasil akhir :

p_1	p_2	$n = p_1 w_1 + p_2 w_2 + b$	$a = f(n)$
-1	-1	$-1 \cdot 2 + -1 \cdot 2 - 2 = -6$	-1
-1	1	$-1 \cdot 2 + 1 \cdot 2 - 2 = -2$	-1
1	-1	$1 \cdot 2 + -1 \cdot 2 - 2 = -2$	-1
1	1	$1 \cdot 2 + 1 \cdot 2 - 2 = 2$	1

$$f(n) = \begin{cases} 1, & n \geq 0 \\ -1, & n < 0 \end{cases}$$

p_1	p_2	Masukan		Target
		1	1	t
-1	-1	1	1	-1
-1	1	1	1	-1
1	-1	1	1	-1
1	1	1	1	1

Keluaran = target \rightarrow Jaringan Hebb ‘mengerti’ pola yang dimaksud

Keberhasilan jaringan Hebb tergantung pada representasi masukan dan target

Model Hebb

Data Pola



-1	-1	-1
1	-1	1
1	-1	1



x0	x1	x2
x3	x4	x5
x6	x7	x8



1



-1	1	-1
-1	1	-1
-1	-1	-1

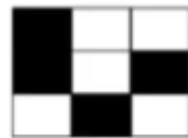


x0	x1	x2
x3	x4	x5
x6	x7	x8



-1

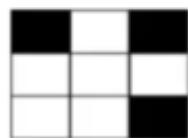
Contoh Pola Rusak



-1	1	1
-1	1	-1
1	-1	1



?



-1	1	-1
1	1	-1
1	1	-1



?



What's in ?

- ▶ Sel Syaraf (Neuron)
- ▶ Analogi JST Vs JSB
- ▶ Model Struktur Neuron JST
- ▶ Model Sel Syaraf (Neuron)
- ▶ Arsitektur JST
- ▶ Metode Pembelajaran JST
- ▶ Aplikasi Dengan JST
- ▶ Model McCulloch-Pitts
- ▶ Model Hebb
- ▶ Model Perceptron

Model Perceptron

- Jaringan terdiri dari beberapa unit masukan (ditambah sebuah bias)
- Memiliki sebuah unit keluaran (y)
$$y_{in} = b + \sum_i x_i w_i$$
- Dengan nilai threshold adalah θ maka fungsi aktivasi :
 - $1, \text{net} > \theta$
 - $F(\text{net}) = 0, -\theta \leq \text{net} \leq \theta$
 - $-1, \text{net} < -\theta$

$$y = \begin{cases} 1, & \text{jika } y_{in} > \theta \\ 0, & \text{jika } -\theta \leq y_{in} \leq \theta \\ -1, & \text{jika } y_{in} < -\theta \end{cases}$$

Model Perceptron

- ▶ Misal : s adalah vektor masukan, t adalah target keluaran, a adalah laju pemahaman, θ adalah threshold.
- 1. Inisialisasi semua bobot dan bias ($w_i = b = 0$)
- 2. Selama output (y) tidak sama dengan target lakukan perbaikan bobot dan bias dengan cara $W(\text{baru}) = W(\text{lama}) + \Delta w$, $b(\text{baru}) = b(\text{lama}) + \Delta b$, dimana $\Delta w = a \cdot t \cdot x_i$, $\Delta b = a \cdot t$.
- 3. Jika output (y) = target maka tidak ada perubahan bobot dan bias, $\Delta w = 0$, Δb

Model Perceptron

- a. Iterasi dilakukan terus hingga semua pola memiliki keluaran jaringan yang sama dengan target.
- b. Perubahan bobot hanya pada keluaran yang tidak sesuai target
- c. Iterasi akan terus berulang pada semua pola sampai keluaran = target (dinamakan epoch)
- d. Kecepatan iterasi ditentukan oleh laju pemahaman (α), semakin besar harga α , semakin sedikit iterasi yang diperlukan, tapi jika terlalu besar akan merusak pola yang sudah benar

MODEL JARINGAN PERCEPTRON

- Model ini ditemukan oleh Rosenblatt (1962) dan Minsky-Papert (1969).
- Algoritma Pelatihan Perceptron:
 1. Inisialisasi semua bobot dan bias (biasanya = 0)
Set learning rate: α ($0 < \alpha \leq 1$). Untuk penyederhanaan set sama dengan 1.
Set nilai threshold (θ) untuk fungsi aktivasi.
 2. Untuk setiap pasangan pembelajaran s-t, kerjakan:
 - (a) Set aktivasi unit input: $x_i = s_i$;
 - (b) Hitung respons untuk unit output:
$$y_{in} = b + \sum_i x_i w_i$$

Algoritma Pelatihan Perceptron:

(c) Masukkan ke dalam fungsi aktivasi:

$$y = \begin{cases} 1, & \text{jika } y_{in} > \theta \\ 0, & \text{jika } -\theta \leq y_{in} \leq \theta \\ -1, & \text{jika } y_{in} < -\theta \end{cases}$$

(d) Bandingkan nilai output jaringan y dengan target t
jika $y \neq t$, lakukan perubahan bobot dan bias dengan cara berikut:

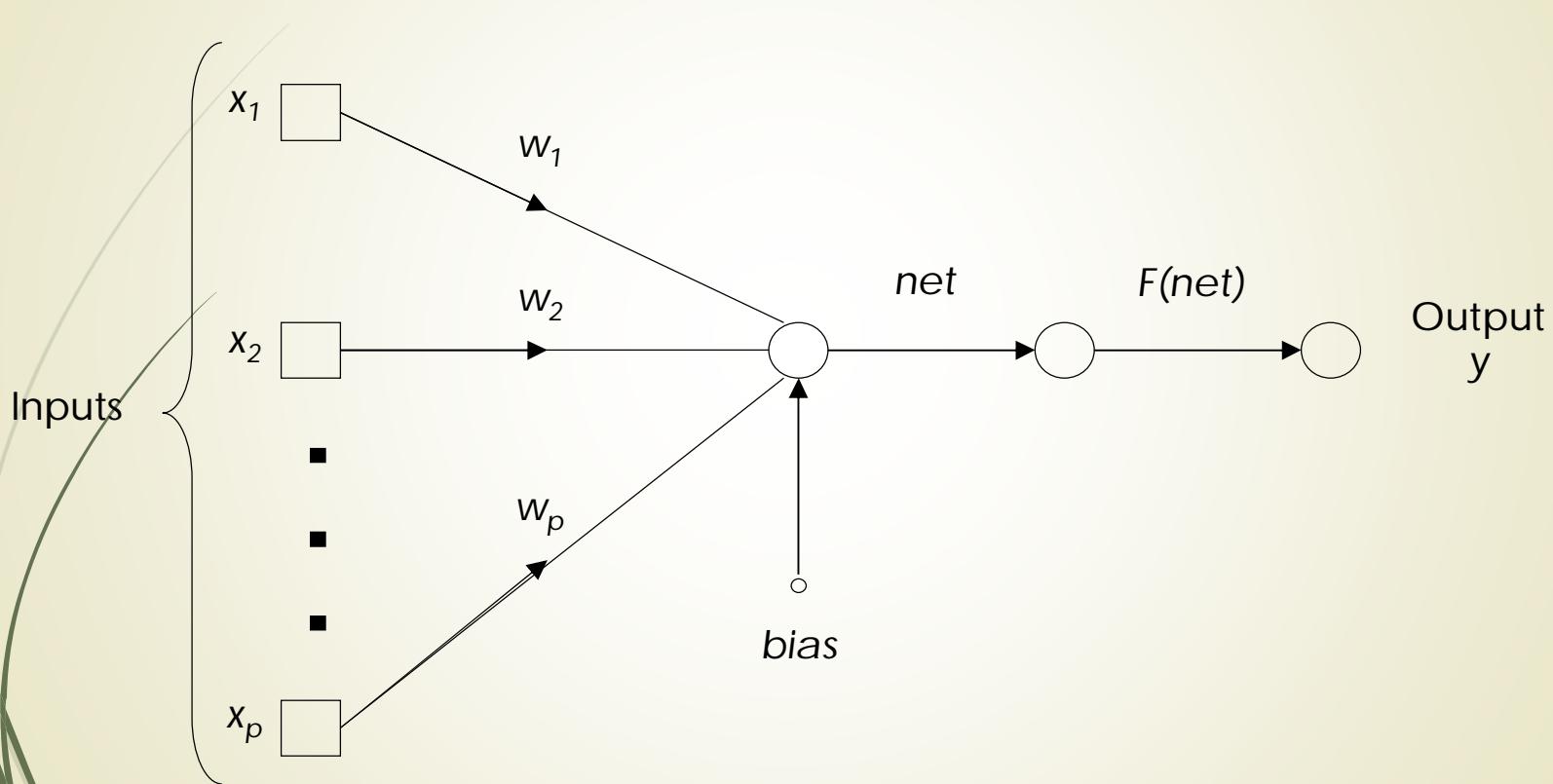
$$\begin{aligned} w_i(\text{baru}) &= w_i(\text{lama}) + \alpha * t * x_i \\ b(\text{baru}) &= b(\text{lama}) + \alpha * t \end{aligned}$$

jika $y = t$, tidak ada perubahan bobot dan bias:

$$\begin{aligned} w_i(\text{baru}) &= w_i(\text{lama}) \\ b(\text{baru}) &= b(\text{lama}) \end{aligned}$$

(3) Lakukan iterasi terus-menerus hingga semua pola memiliki output jaringan yang sama dengan targetnya. Artinya bila semua output jaringan sama dengan target maka jaringan telah mengenali pola dengan baik dan iterasi dihentikan.

Model Perceptron



Model Perceptron

- ▶ Buatlah perceptron untuk mengenali fungsi logika “dan” dengan masukan dan keluaran bipolar seperti pada tabel berikut :

Masukan (x1 dan x2)	Target
1	1
1	-1
-1	1
-1	-1

Model Perceptron

- Inisialisasi : $w=b=0$, $a = 1$, $\theta = 0.5$, jadi fungsi aktivasi :
- $1, \text{net} > 0.5$
- $F(\text{net}) = 0, -0.5 \leq \text{net} \leq 0.5$
- $-1, \text{net} < -0.5$

Model Perceptron

- Epoch 1, $w=b=0$,
- $\alpha = 1, \theta = 0$

- Epoch 2, $w_1=1, w_2=1, b= -1$

- Jaringan sudah mengenali pola

$$b + \sum_i x_i w_i \begin{cases} 1, & \text{jika } y_{in} > \theta \\ 0, & \text{jika } -\theta \leq y_{in} \leq \theta \\ -1, & \text{jika } y_{in} < -\theta \end{cases}$$

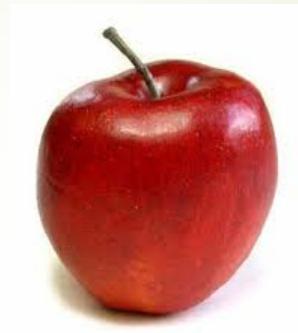
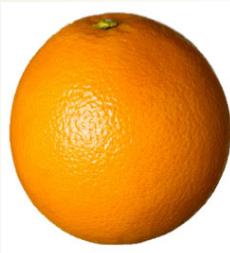
Masukan (x1 dan x2)	Target
1	1
1	-1
-1	1
-1	-1

Masukan				Target		Output	Perubahan bobot $\Delta w = \alpha x_i t$ $\Delta b = \alpha t$			Bobot baru $w_{baru} = w_{lama} + \Delta w$ $b_{baru} = b_{lama} + \Delta b$		
x_1	x_2	a	t	Net	$a=f(n)$	Δw_1	Δw_2	Δb	w_1	w_2	b	
Epoch ke - 1												
1	1	1	1	0	0	1	1	1	1	1	1	
1	-1	1	-1	1	1	-1	1	-1	0	2	0	
-1	1	1	-1	2	1	1	-1	-1	1	1	-1	
-1	-1	1	-1	-3	-1	0	0	0	1	1	-1	

Masukan				Target		Output	Perubahan bobot $\Delta w = \alpha x_i t$ $\Delta b = \alpha t$			Bobot baru $w_{baru} = w_{lama} + \Delta w$ $b_{baru} = b_{lama} + \Delta b$		
x_1	x_2	a	t	n	$a=f(n)$	Δw_1	Δw_2	Δb	w_1	w_2	b	
Epoch ke - 2												
1	1	1	1	1	1	0	0	0	1	1	-1	
1	-1	1	-1	-1	-1	0	0	0	1	1	-1	
-1	1	1	-1	-1	-1	0	0	0	1	1	-1	
-1	-1	1	-1	-3	-1	0	0	0	1	1	-1	

Model Perceptron

Membedakan :



Dengan data input dari sensor adalah sebagai berikut :

X1, Merah=1,tidak merah=-1

X2, Bulat=1,tidak bulat=-1

X3, Kasar=1,tidak kasar=-1

Target : Apel = 1, bukan apel = -1

Model Perceptron

- ▶ Diberikan 2 pola sebagai target :
- ▶ 1
- ▶ 1
- ▶ -1 $t = 1$
- ▶ -1
- ▶ 1
- ▶ 1 $t = -1$
- ▶ Buatlah perceptron untuk mengenali pola itu!

Model Perceptron

► w=b=0, a = 1, θ = 0.5

x1	x2	x3	t
1	1	-1	1
-1	1	1	-1



What's in Next Chapter ?



GA



Contact

Sendi Novianto

Bidang minat utama : game technology, Artificial Intelligence, Image Processing, Pattern Recognition, IOT

Bidang minat sekunder : Computer Graphics, Operating System, Database, Web Programming

- ▶ Email : sendi.novianto@dsn.dinus.ac.id
- ▶ No. HP / Whatsapp : 0813 9010 5422
- ▶ Tempat Ruang H.2.4



Thank You!