

# ARTIFICIAL NEURAL NETWORK (JARINGAN SYARAF TIRUAN)

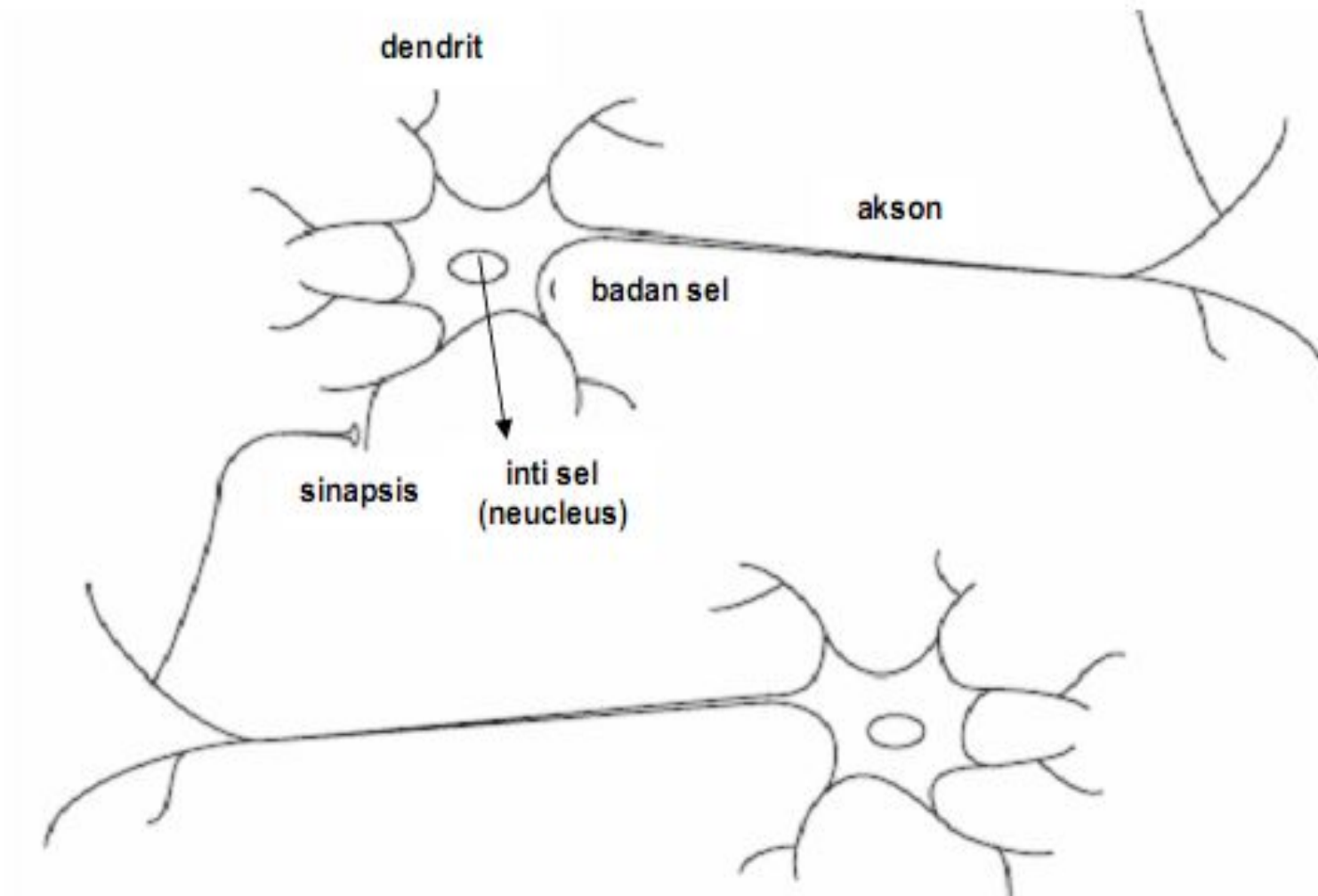
# Outline

1. **Konsep jaringan syaraf tiruan**
2. Model McCulloch – Pitts
3. Model Hebb
4. Model Perceptron

# Jaringan Syaraf Biologis

- Otak manusia berisi jutaan sel syaraf (neuron) yang bertugas memproses informasi
- Neuron saling berinteraksi satu sama lain mendukung kemampuan kerja otak manusia

# Sel Syaraf (Neuron)



# Sel Syaraf (Neuron)

- Komponen utama neuron dapat dikelompokkan menjadi 3 bagian :
  - Dendrit = bertugas menerima informasi = jalur input bagi soma
  - Badan sel (soma) = tempat pengolahan informasi, di badan sel terdapat inti sel yang bertugas mengolah informasi
  - Akson = bertugas mengirimkan impuls-impuls sinyal ke sel syaraf lain = jalur output bagi soma
  - Antar dendrit dipertemukan dengan sinapsis

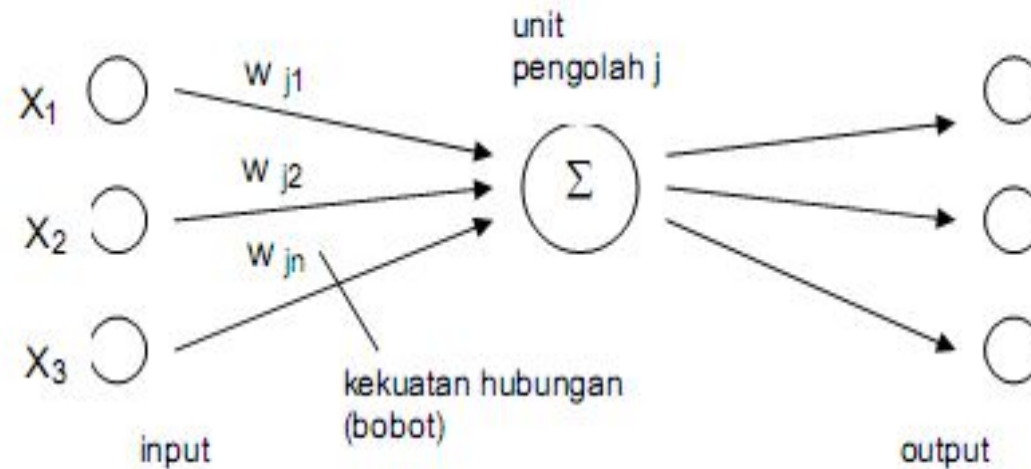
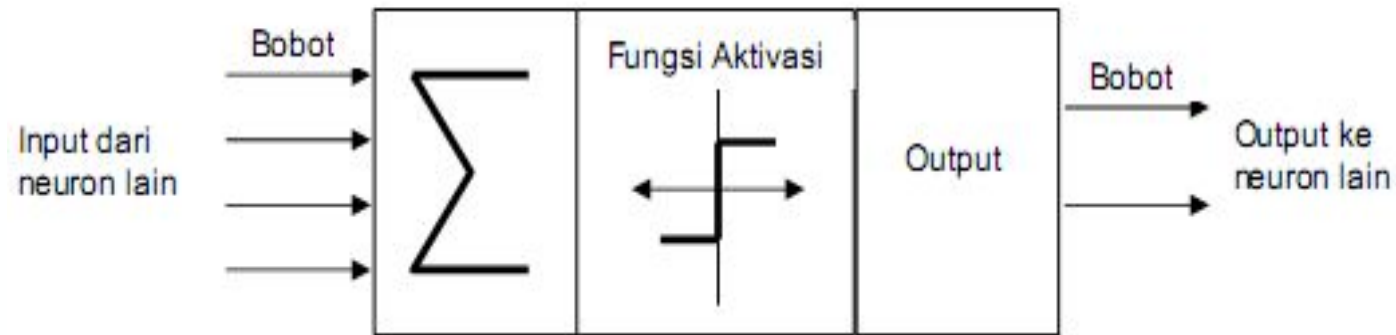
# Analogi JST dengan JSB

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

# Jaringan Syaraf Tiruan

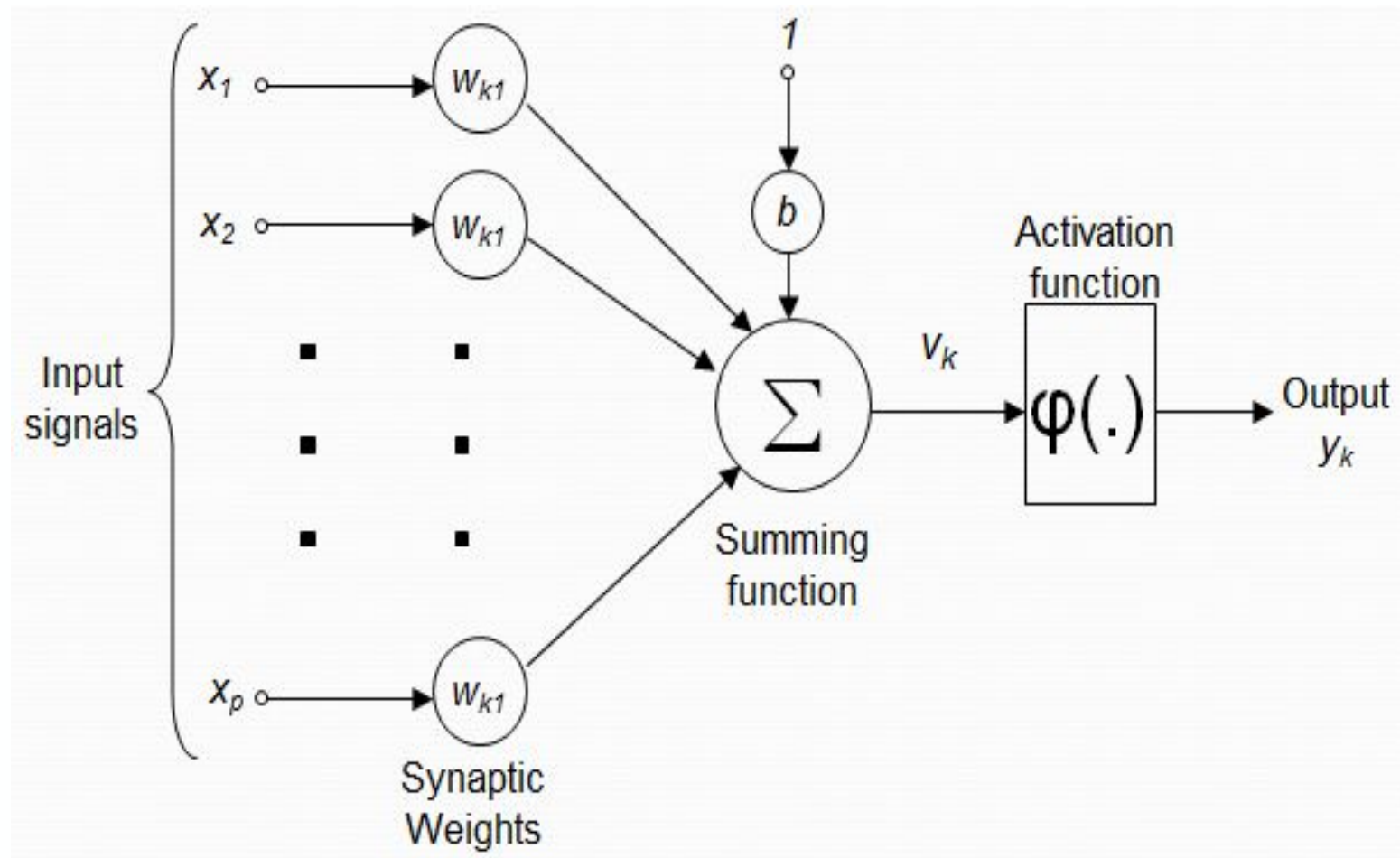
- Meniru cara kerja jaringan syaraf biologis
- Generalisasi model matematis dari pemahaman manusia:
  - Pemrosesan informasi terjadi pada neuron
  - Sinyal mengalir diantara sel saraf/neuron melalui suatu sambungan penghubung
  - Setiap sambungan penghubung memiliki bobot yang bersesuaian.
  - Bobot ini akan digunakan untuk menggandakan / mengalikan sinyal yang dikirim melaluinya.
  - Setiap sel syaraf akan menerapkan fungsi aktivasi terhadap sinyal hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan sinyal keluarannya.

# Model Struktur Neuron JST





# Model Sel Syaraf (Neuron)



# Model Sel Syaraf (Neuron)

- Secara matematis:

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

dan

$$y_k = \varphi(v_k)$$

# 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 ( $w_{kj}$ ) dan menjumlahkannya

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

# SUMMATION FUNCTION

- Diibaratkan dengan sebuah neuron yang memonitor sinyal yang datang dari neuron-neuron lain.
- Neuron ini menghitung penjumlahan berbobotnya dan kemudian menentukan sinyal untuk dikirim ke neuron-neuron lain.

# 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.

# Karakteristik JST

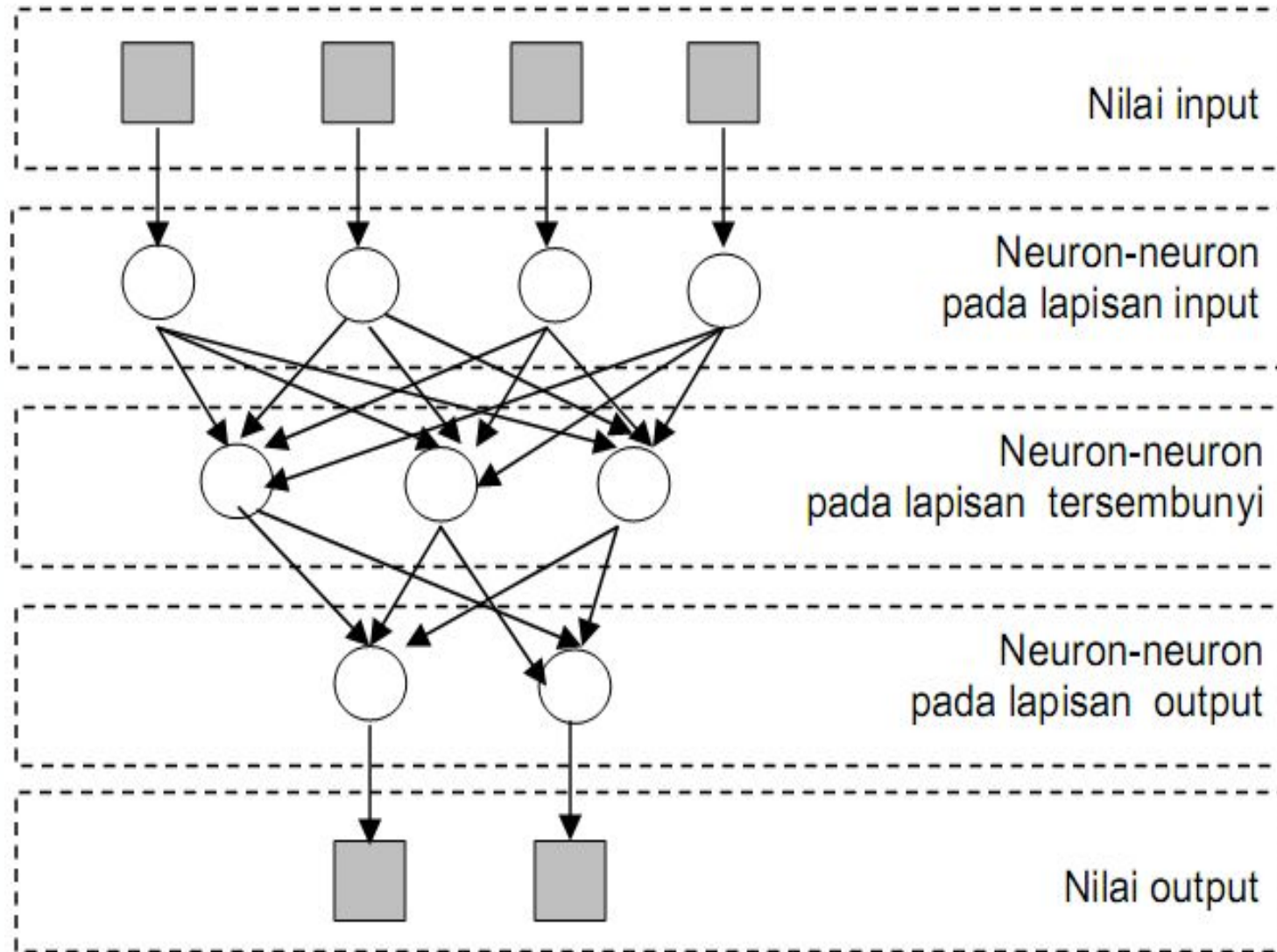
- Ditentukan oleh :
  - Pola hubungan antar neuron (disebut arsitektur jaringan)
  - Metode penentuan bobot-bobot sambungan (disebut dengan pelatihan atau proses belajar jaringan)
  - Fungsi aktivasi

# Arsitektur JST

- Pada JST, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (layer) yang disebut dengan lapisan neuron (neuron layers).
- Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya.
- Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan input sampai ke lapisan output melalui lapisan tersembunyi (hidden layer).



# Contoh





# Arsitektur JST

- ◆ Faktor terpenting untuk menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya.
- ◆ Umumnya neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama → fungsi aktivasi yang sama.
- ◆ Bila neuron-neuron pada suatu lapisan (misal lapisan tersembunyi) akan dihubungkan dengan neuron-neuron pada lapisan lain (misal lapisan output) maka setiap neuron pada lapisan tersebut (lapisan tersembunyi) juga harus dihubungkan dengan setiap neuron pada lapisan lainnya (lapisan output)

# 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)

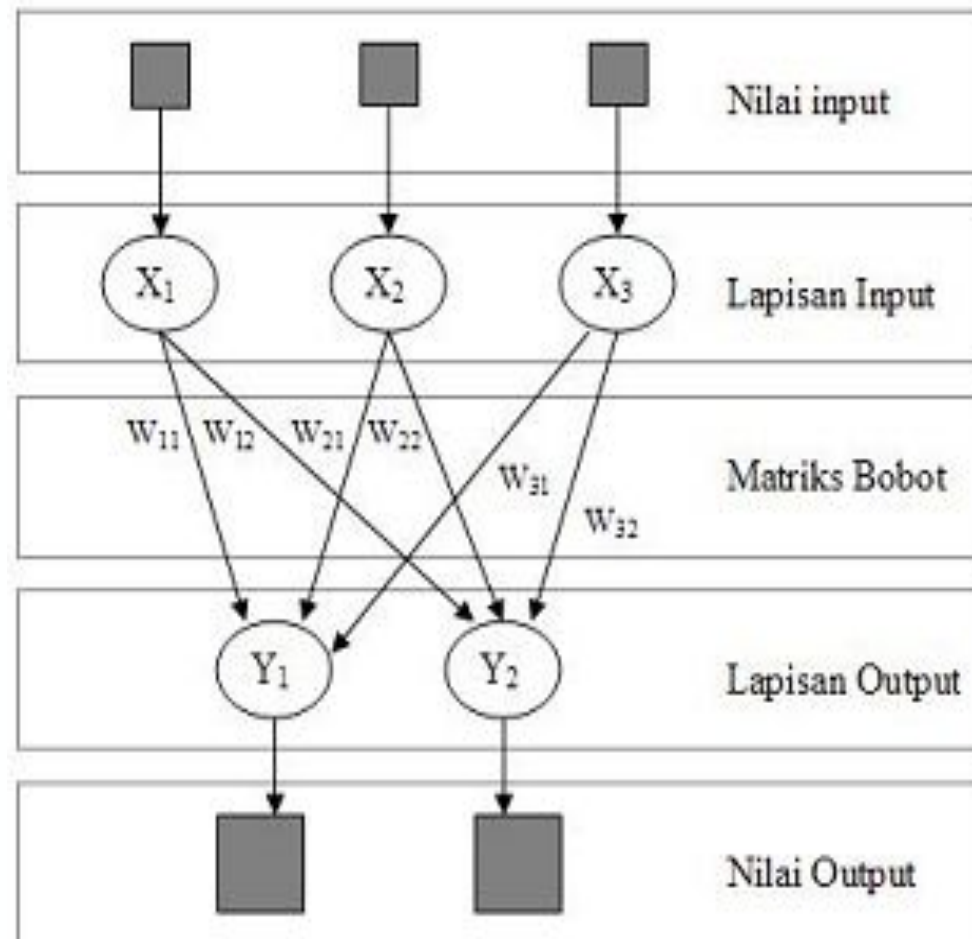
# Single Layer Net

- ◆ Hanya memiliki satu lapisan dengan bobot-bobot terhubung
- ◆ Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.

# Single Layer Net

- Seberapa besar hubungan antara 2 neuron ditentukan oleh bobot yang bersesuaian.
- Semua unit input akan dihubungkan dengan setiap unit output.

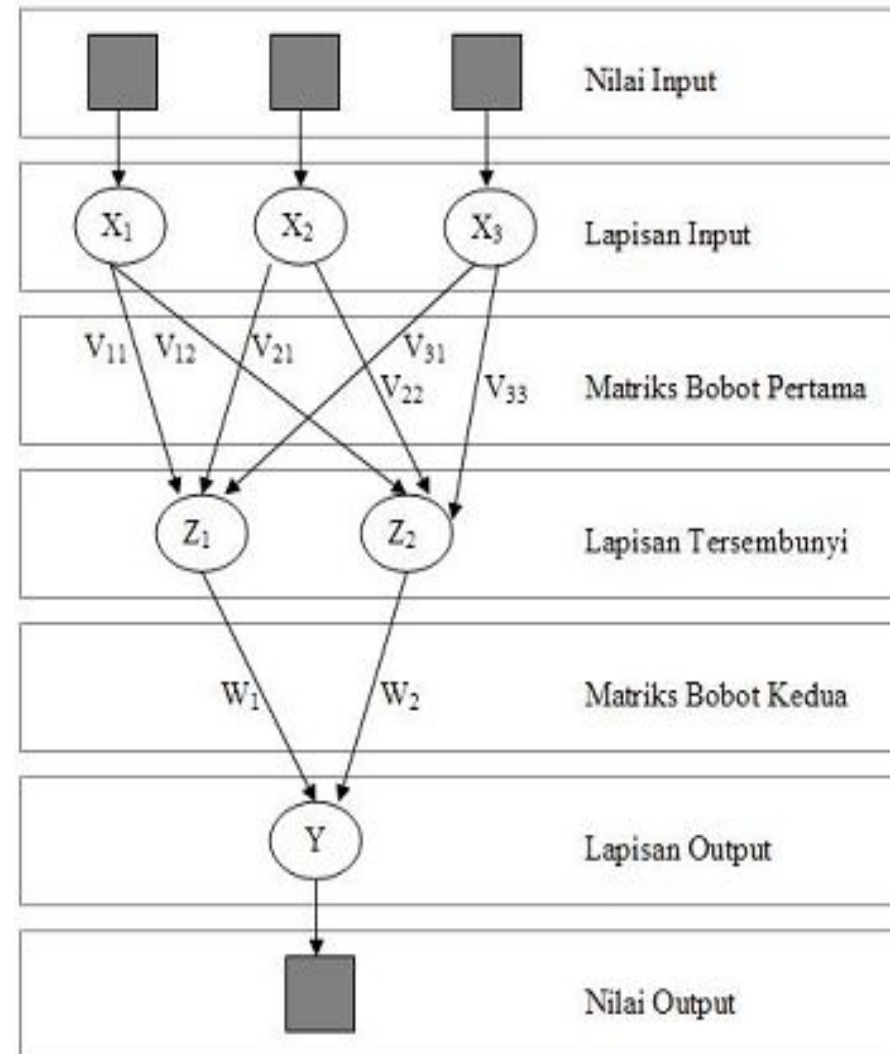
# Single Layer Net



# Multilayer Net

- ◆ Memiliki 1 atau lebih lapisan yang terletak diantara lapisan input dan lapisan output
- ◆ Ada lapisan bobot yang terletak antara 2 lapisan yang bersebelahan
- ◆ Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit

# Multilayer Net

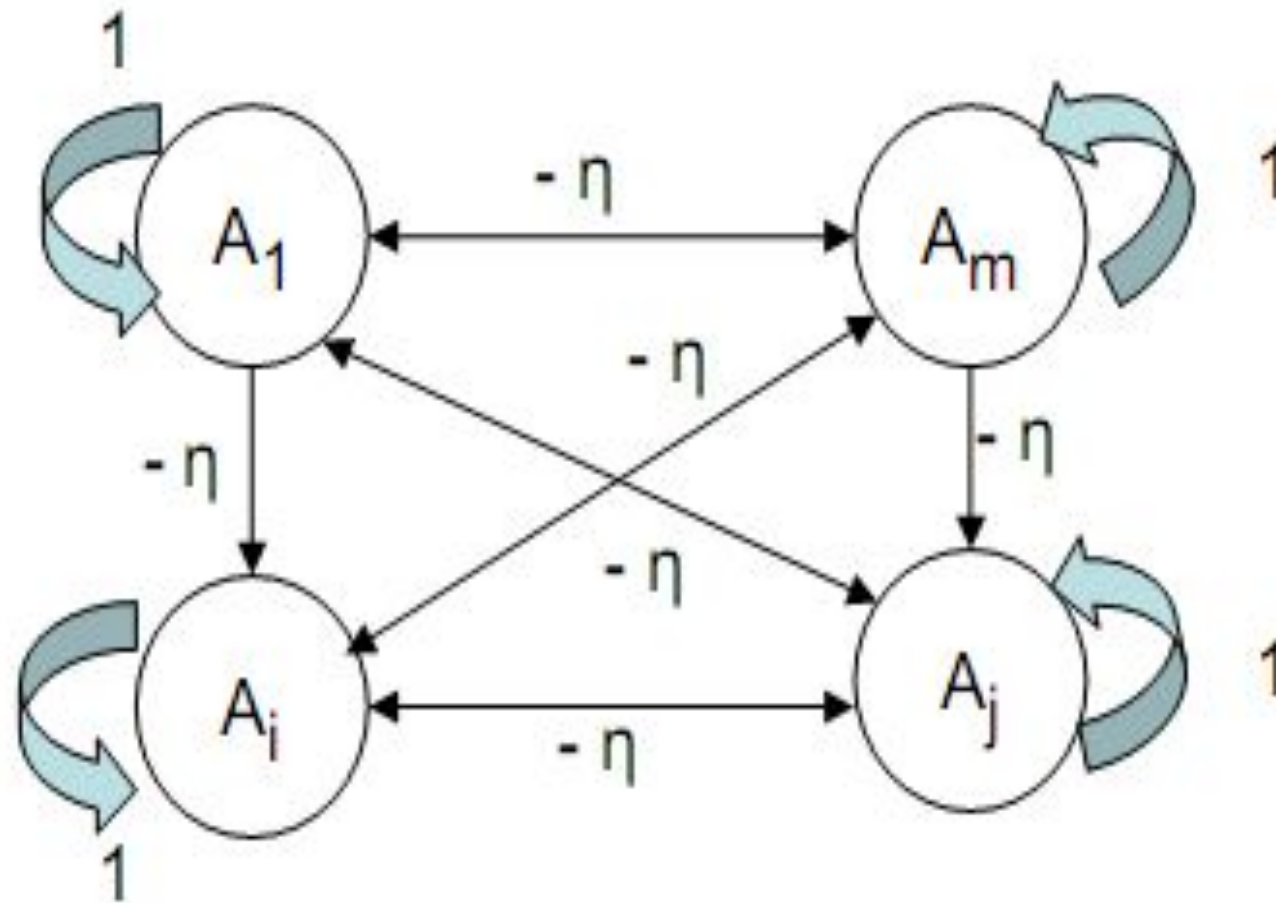


# Competitive Net

- ◆ Sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif
- ◆ Umumnya hubungan antar neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur



# Competitive Net



# Proses Pembelajaran Jaringan

- ◆ Cara belajar JST :
- ◆ Ke dalam JST diinputkan informasi yang sebelumnya telah diketahui hasil keluarannya.
- ◆ Penginputan informasi ini dilakukan lewat node-node atau unit-unit input. Bobot-bobot antarkoneksi dalam suatu arsitektur diberi nilai awal dan kemudian JST dijalankan.

# Proses Pembelajaran Jaringan

- ◆ Bobot-bobot ini bagi jaringan digunakan untuk belajar dan mengingat suatu informasi. Pengaturan bobot dilakukan secara terus-menerus dan dengan menggunakan kriteria tertentu sampai diperoleh keluaran yang diharapkan.
- ◆ Hal yang ingin dicapai dengan melatih/mengajari JST adalah untuk mencapai keseimbangan antara kemampuan memorisasi dan generalisasi.

# 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.

# 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

# Pembelajaran 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

# Pembelajaran Terawasi

- Satu pola input akan diberikan ke satu neuron pada lapisan input
- Pola ini akan dirambatkan di sepanjang jaringan syaraf hingga sampai ke neuron pada lapisan output
- Lapisan output ini akan membangkitkan pola output yang akan dicocokkan dengan pola output targetnya
- Jika berbeda → error
- Jika error terlalu besar, perlu dilakukan pembelajaran lebih banyak

# Pembelajaran 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

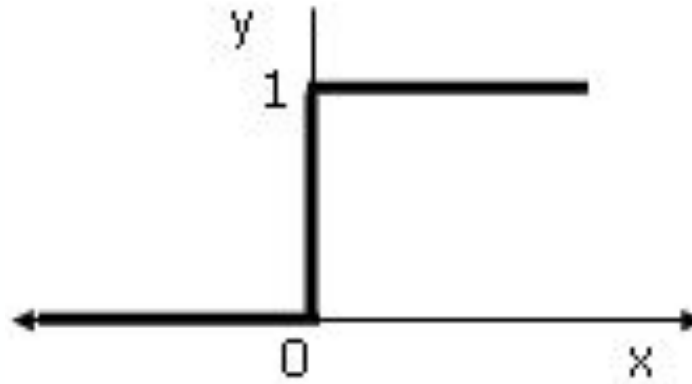


# Fungsi Aktivasi

- ◆ Dipakai untuk menentukan keluaran suatu neuron
- ◆ Merupakan fungsi yang menggambarkan hubungan antara tingkat aktivasi internal (summation function) yang mungkin berbentuk linier atau nonlinear.
- ◆ Beberapa fungsi aktivasi JST diantaranya threshold, hard limit, sigmoid, dan identitas.

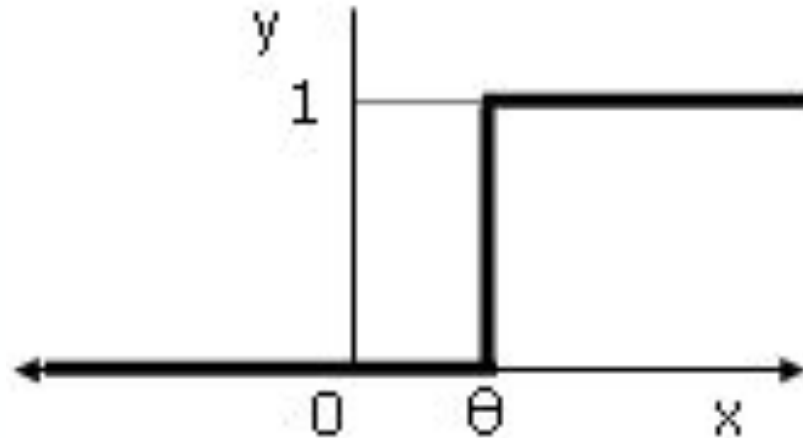
## Fungsi *Hard Limit*

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



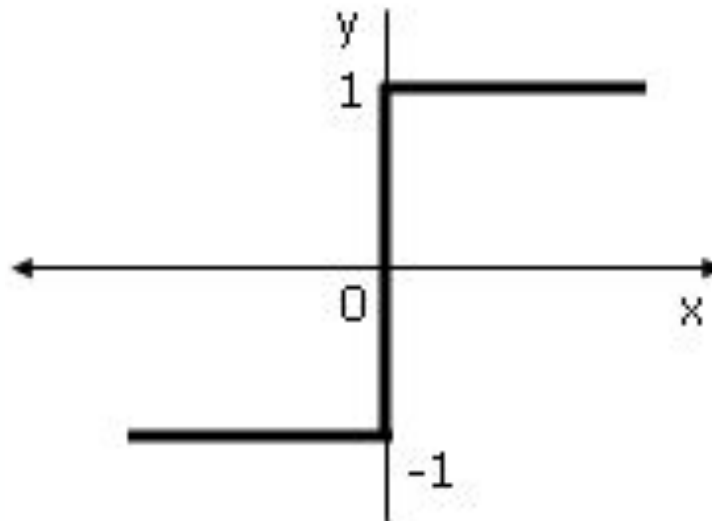
## Fungsi Threshold

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



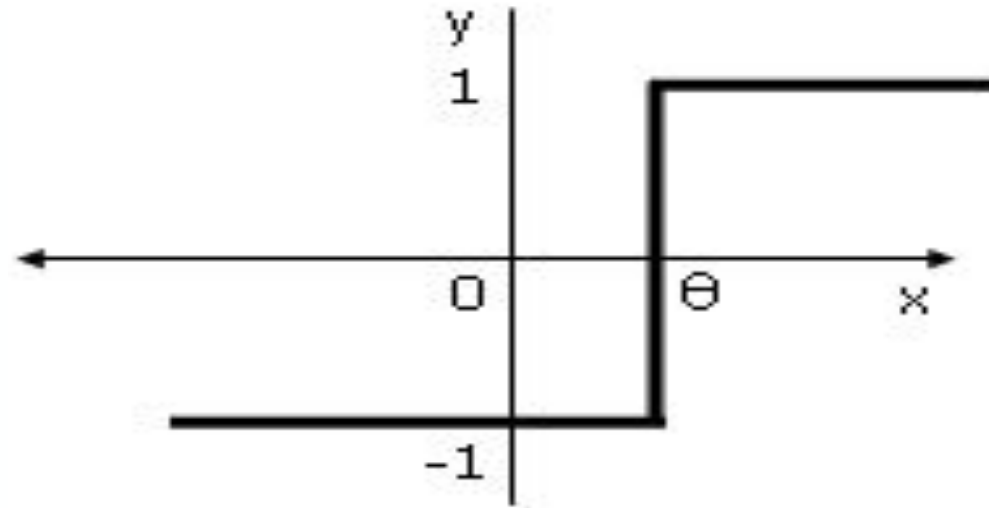
## Fungsi Bipolar *Hard Limit*

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

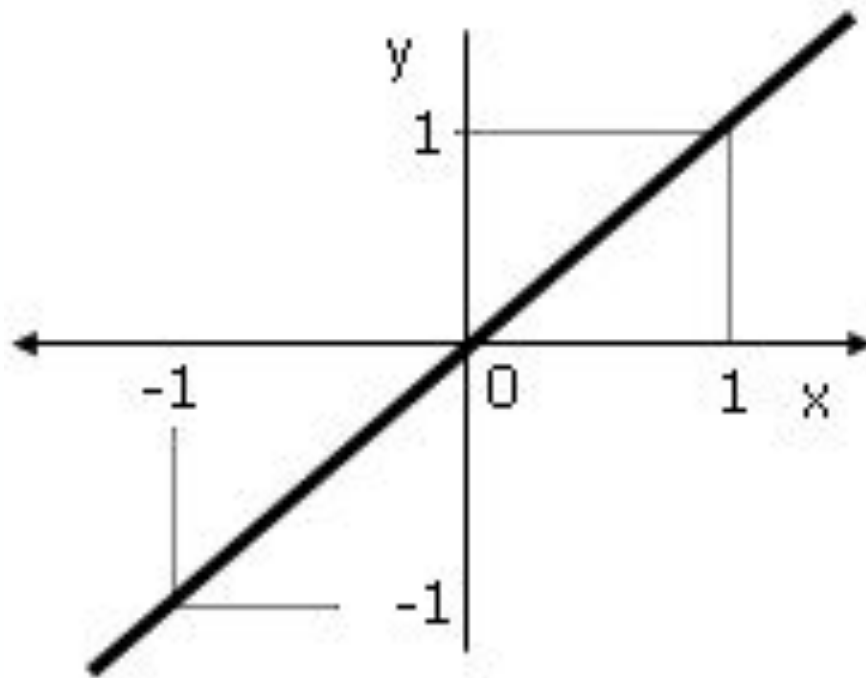


## Fungsi Bipolar *Threshold*

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



## Fungsi Linear (Identitas)

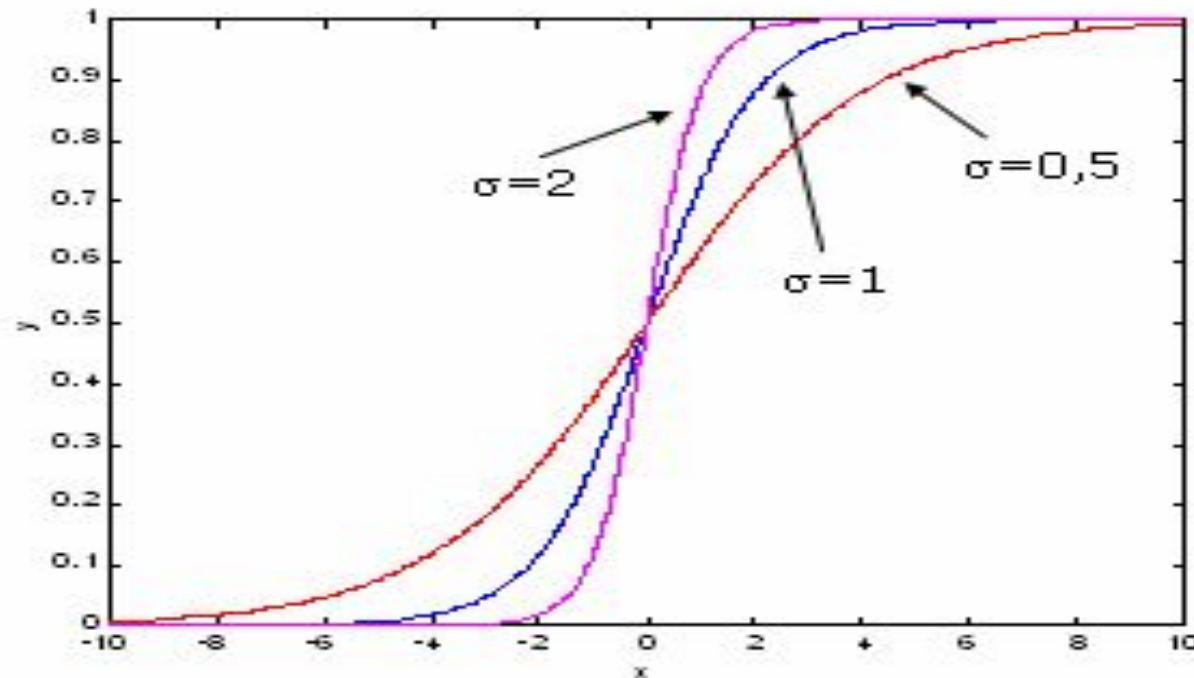


$$y = x$$

$$f(x) = x$$

## Fungsi Sigmoid

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



# Aplikasi Yang Dapat Dibuat dengan JST

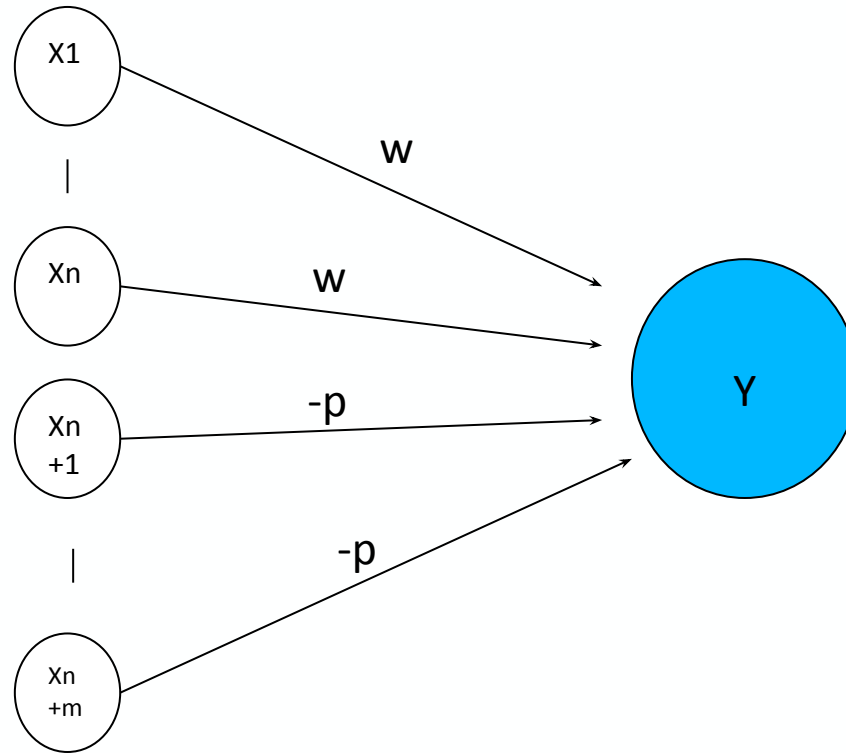
- a. Klasifikasi
- b. Pengenalan pola
- c. Peramalan
- d. Optimisasi



# Outline

1. Konsep jaringan syaraf tiruan
2. **Model McCulloch – Pitts**
3. Model Hebb
4. Model Perceptron

# Model McCulloch-Pitts



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

# Model McCulloch-Pitts

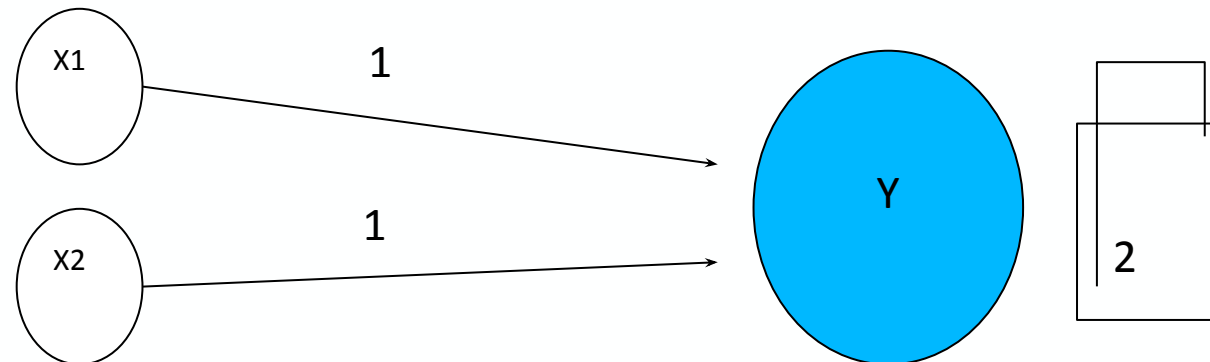
- Model yang pertama ditemukan dengan karakteristik :
  - 1. Fungsi aktivasi biner threshold
  - 2. Semua bobot positif (excitatory) memiliki nilai yang sama
  - 3. Tiap neuron punya threshold yang sama
- Kelemahan : penentuan bobot secara analitik / trial and error secara manual, sulit untuk masalah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

# *Model McCulloch-Pitts*

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 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	$1.1 + 1.1 = 2$

Agar  $y(net)$  memenuhi fungsi logika “AND”, maka nilai ambang  $\theta$  pada fungsi aktivasi dibuat sama dengan 2, sehingga

# Model McCulloch-Pitts

$X_1$	$X_2$	$net = \sum x_i w$	$y(net) = \begin{cases} 0 & \text{jika } net < 2 \\ 1 & \text{jika } net \geq 2 \end{cases}$
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

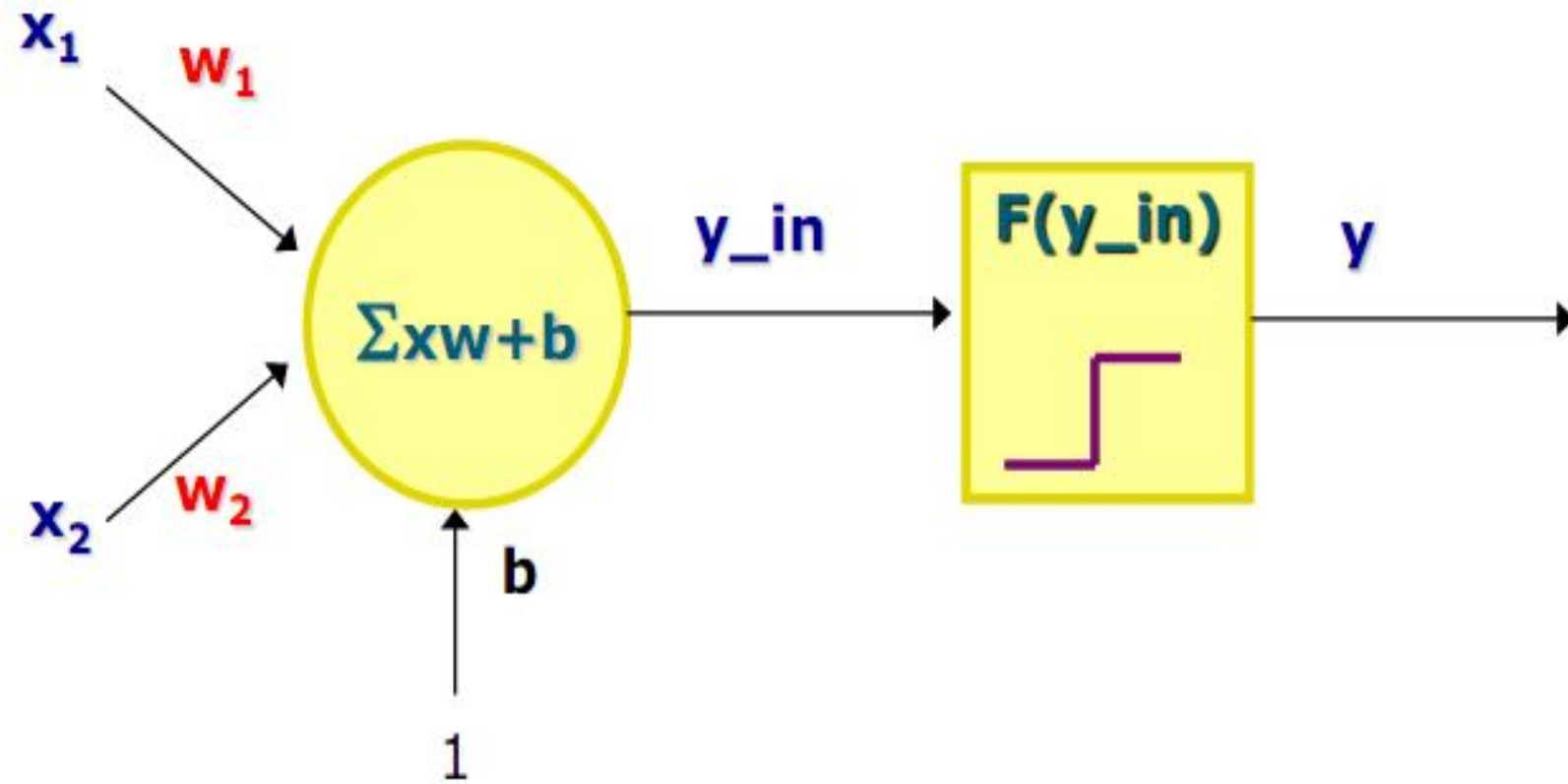
nilai bobot  $w$  dan nilai ambang  $\theta$  dapat diubah-ubah sesuai keinginan supaya sesuai dengan yang diharapkan (trial and error)

# Outline

1. Konsep jaringan syaraf tiruan
2. Model McCulloch – Pitts
3. **Model Hebb**
4. Model Perceptron



# Model Hebb



# 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  $y = t$
    - Perbaiki bobot menurut persamaan
      - $W_i (\text{baru}) = W_i (\text{lama}) + \Delta W$
      - $\Delta W = X_i \cdot y$
    - Perbaiki bias menurut persamaan :
      - $b (\text{baru}) = b (\text{lama}) + y$

# Contoh Kasus

- Membedakan pola :
  - # o #            # o o
  - o # o            # o o
  - # o #            # # #
  - (X)              (L)
- 
- Bagaimana JST mengenali pola berikut :
  - # # o
  - # # o
  - # # #

# Contoh Kasus

- $\# = 1, o = -1$
- $X = 1, L = -1$
- Fungsi aktivasi :
- $y = 1$ , jika  $y_{in} \geq 0$
- $y = -1$ , jika  $y_{in} < 0$
  
- $\# \ o \ \#$                        $\# \ o \ o$
- $o \ \# \ o \ t=1$                  $\# \ o \ o \ t=-1$
- $\# \ o \ \#$                        $\# \ \# \ \#$
-

# Contoh Kasus

- Input pertama :

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

- $\Delta W = x.t$ , maka nilai  $\Delta W$  untuk tiap input:

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

- Bias = t = 1

# Contoh Kasus

- Input kedua :

1 -1 -1

1 -1 -1

1 1 1

- $\Delta W = x.t$ , maka nilai  $\Delta W$  untuk tiap input:

-1 1 1

-1 1 1

-1 -1 -1

Bias = t = -1

# Contoh Kasus

- Bobot baru :

0 0 2

-2 2 0

0 -2 0

Bias = 0

# Contoh Kasus

- Aplikasikan bobot baru ke input 1 :
- $(1.0)+(-1.0)+(1.2)+(-1.-2)+(1.2)+(-1.0)+(1.0)+(-1.-2)+(1.0) = 8$
- Jadi  $y = 1$ , sesuai target ( $t=1$ )
  
- Aplikasikan bobot baru ke input 2 :
- $(1.0)+(-1.0)+(-1.2)+(1.-2)+(-1.2)+(-1.0)+(1.0)+(1.-2)+(1.0) = -8$
- Jadi  $y = -1$ , sesuai target ( $t=-1$ )
  
- Jadi JST sudah bisa mengenali pola



# Contoh Kasus

- Aplikasikan ke pola yang baru :
- 1 1 -1
- 1 1 -1
- 1 1 1
- Beri bobot yang baru :  $(1.0) + (-1.0) + (-1.2) + (1.-2) + (-1.2) + (-1.0) + (1.0) + (1.-2) + (1.0) = -8$
- Jadi  $y = -1$ , dikenali sebagai L