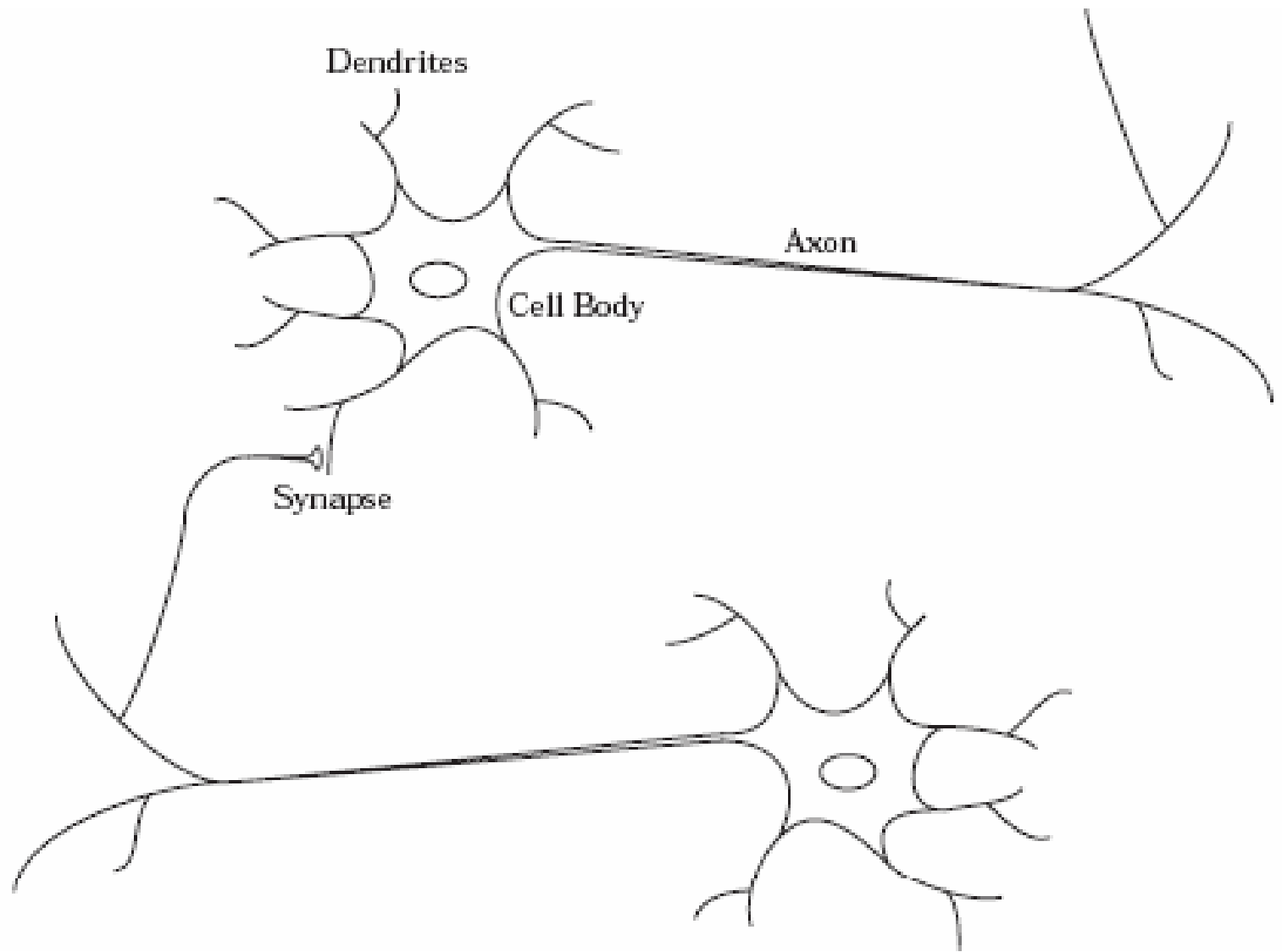


# **PENGANTAR JARINGAN SYARAF TIRUAN (JST)**

# JARINGAN SYARAF BIOLOGIS (JSB)

- Otak manusia berisi sekitar  $10^{11}$  sel syaraf (**neuron**) yang bertugas untuk memproses informasi yang masuk.
- Tiap sel syaraf dihubungkan dengan sel syaraf lain (**sinapsis**).
- Tiap sel bekerja seperti suatu prosesor sederhana.
- Masing-masing sel tersebut saling berinteraksi sehingga mendukung kemampuan kerja otak manusia.



**Komponen utama Neuron** dapat dikelompokkan menjadi 3 bagian :

- 1. Dendrit** : bertugas menerima informasi = jalur **input** bagi soma
- 2. Badan sel (soma)** = tempat pengolahan informasi
- 3. Akson** = bertugas mengirimkan impuls-impuls sinyal ke sel syaraf lain = jalur **output** bagi soma

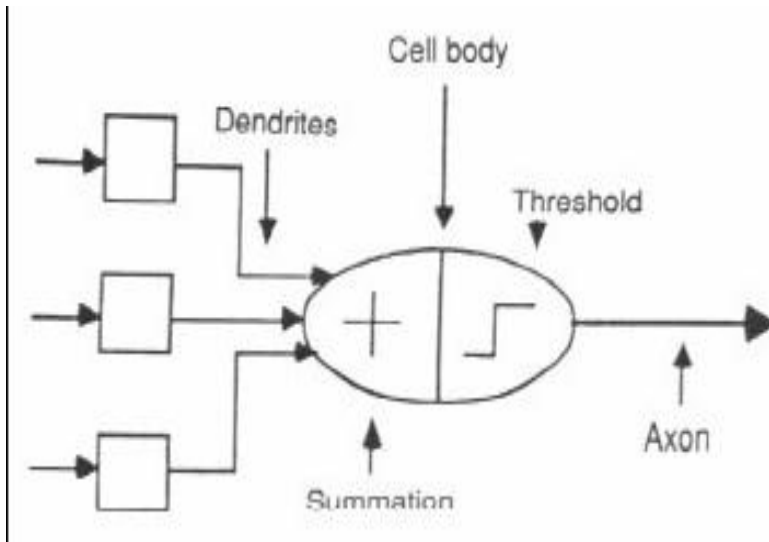
# JARINGAN SYARAF BIOLOGIS (JSB)

- Sebuah neuron menerima impuls-impuls sinyal (informasi) dari neuron lain melalui dendrit dan mengirimkan sinyal yang dibangkitkan oleh badan sel melalui akson.
- Akson dari sel syaraf ini bercabang-cabang dan berhubungan dengan dendrit dari sel syaraf lain dengan cara mengirimkan **impuls** melalui sinapsis.
- Sinapsis adalah unit fungsional antara 2 buah sel syaraf, misal A dan B, dimana yang satu adalah serabut akson dari neuron A dan satunya lagi adalah dendrit dari neuron B.

# JARINGAN SYARAF BIOLOGIS (JSB)

- Kekuatan sinapsis bisa menurun / meningkat tergantung seberapa besar **tingkat propagasi (penyiaran)** sinyal yang diterimanya.
- Impuls-impuls sinyal (informasi) akan diterima oleh neuron lain jika memenuhi batasan tertentu, yang sering disebut dengan **nilai ambang (threshold)**.

**Model  
Struktur  
NEURON JSB**



# JARINGAN SYARAF TIRUAN (JST)

- JST didefinisikan sebagai suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan syaraf manusia (JSB)
- JST tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (human cognition) yang didasarkan atas asumsi sebagai berikut :
  1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron
  2. Sinyal mengalir diantara sel saraf/neuron melalui suatu sambungan penghubung

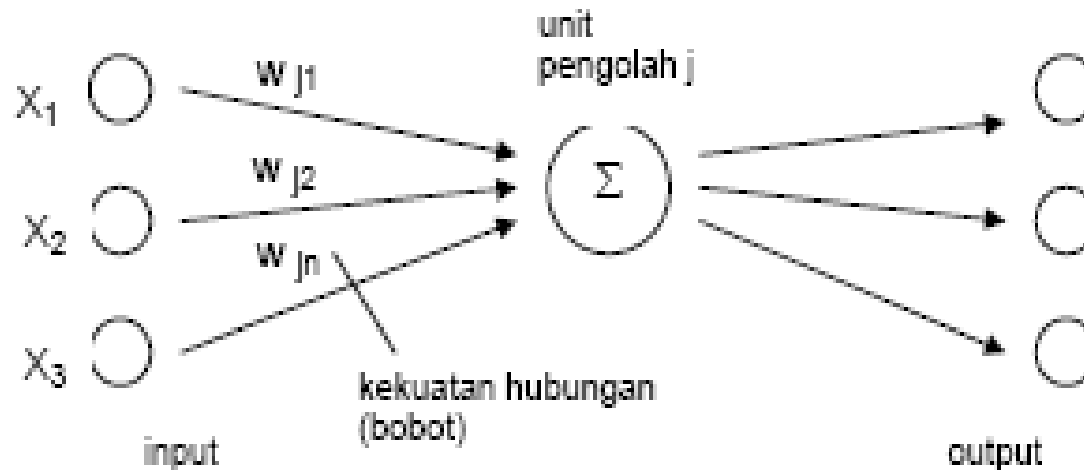
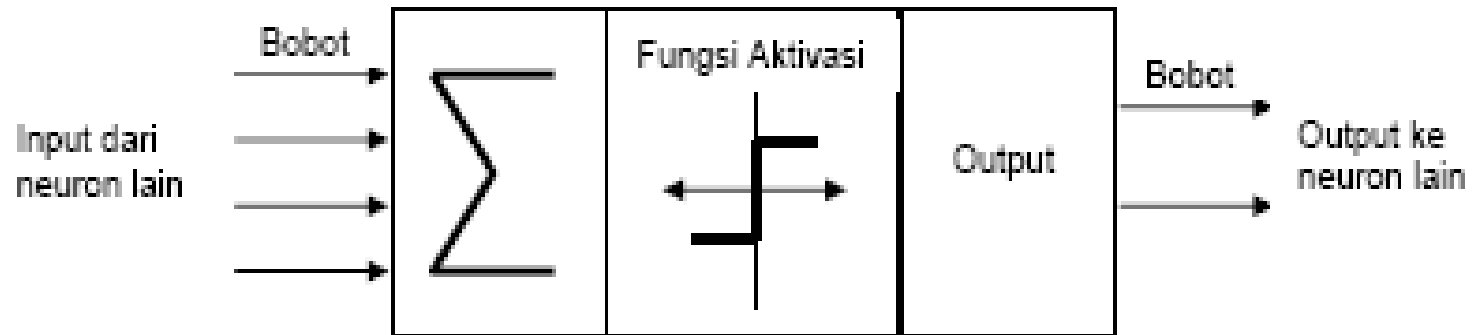
# JARINGAN SYARAF TIRUAN (JST)

3. Setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini akan digunakan untuk menggandakan / mengalikan sinyal yang dikirim melaluinya.
4. Setiap sel syaraf akan menerapkan fungsi aktivasi terhadap sinyal hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan sinyal keluarannya.

# Analogi JST dengan JSB

JST	JS Biologis
Node	Badan Sel (Soma)
Input	Dendrit
Output	Akson
Bobot	Sinapsis

# Model Struktur NEURON JST



# JARINGAN SYARAF TIRUAN (JST)

- Jaringan syaraf tiruan dapat belajar dari pengalaman, melakukan generalisasi atas contoh-contoh yang diperolehnya dan mengabstraksi karakteristik esensial input bahkan untuk data yang tidak relevan.
- 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.

# JARINGAN SYARAF TIRUAN (JST)

- Pada proses pembelajaran, ke dalam JST dimasukkan pola-pola input (dan output) lalu jaringan akan diajari untuk memberikan jawaban yang bisa diterima.
- Pada dasarnya karakteristik JST ditentukan oleh
  1. Pola hubungan antar neuron (disebut arsitektur jaringan)
  2. Metode penentuan bobot-bobot sambungan (disebut dengan pelatihan atau proses belajar jaringan)
  3. Fungsi aktivasi

# Macam arsitektur JST ada 3

- 1. Jaringan dengan lapisan tunggal (single layer net)**
- 2. Jaringan dengan banyak lapisan (multilayer net)**
- 3. Jaringan dengan lapisan kompetitif (competitive layer net)**

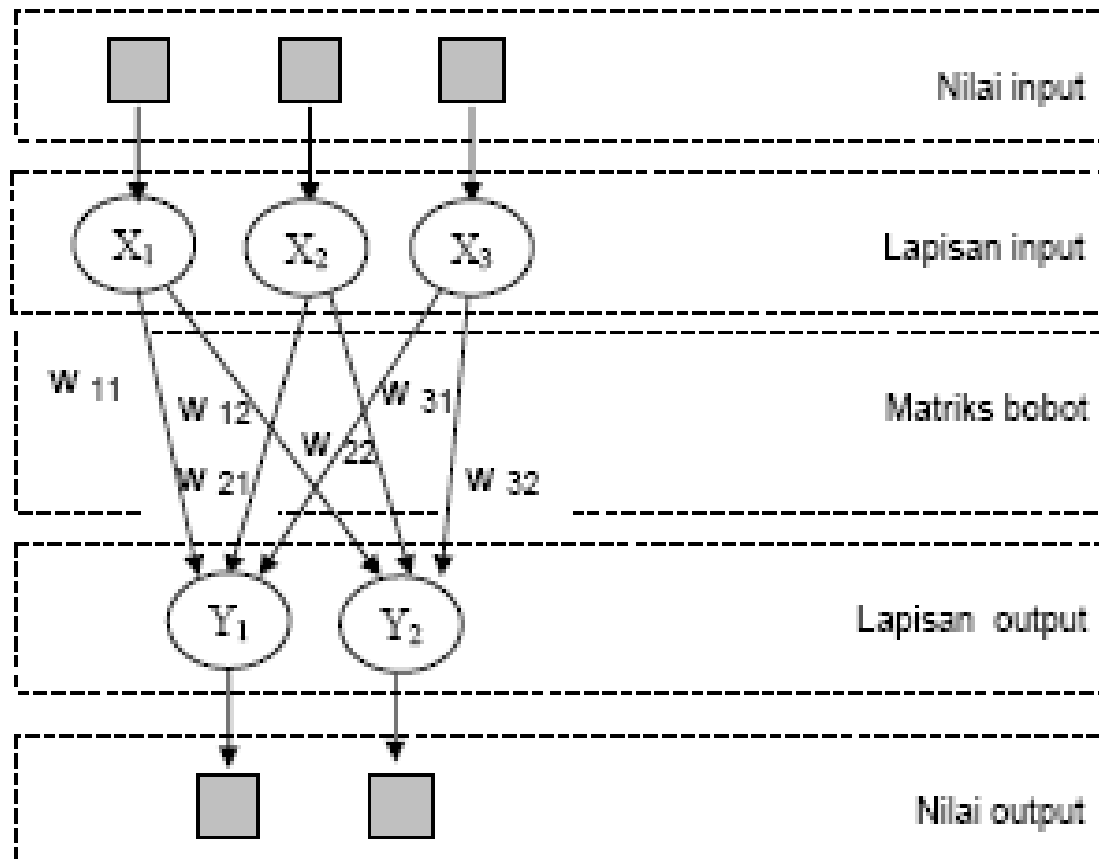
# Jaringan dengan lapisan tunggal (single layer net)

- Hanya memiliki 1 lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.
- Seberapa besar hubungan antara 2 neuron ditentukan oleh bobot yang bersesuaian.
- Semua unit input akan dihubungkan dengan setiap unit output.

# Jaringan dengan banyak lapisan (multilayer net)

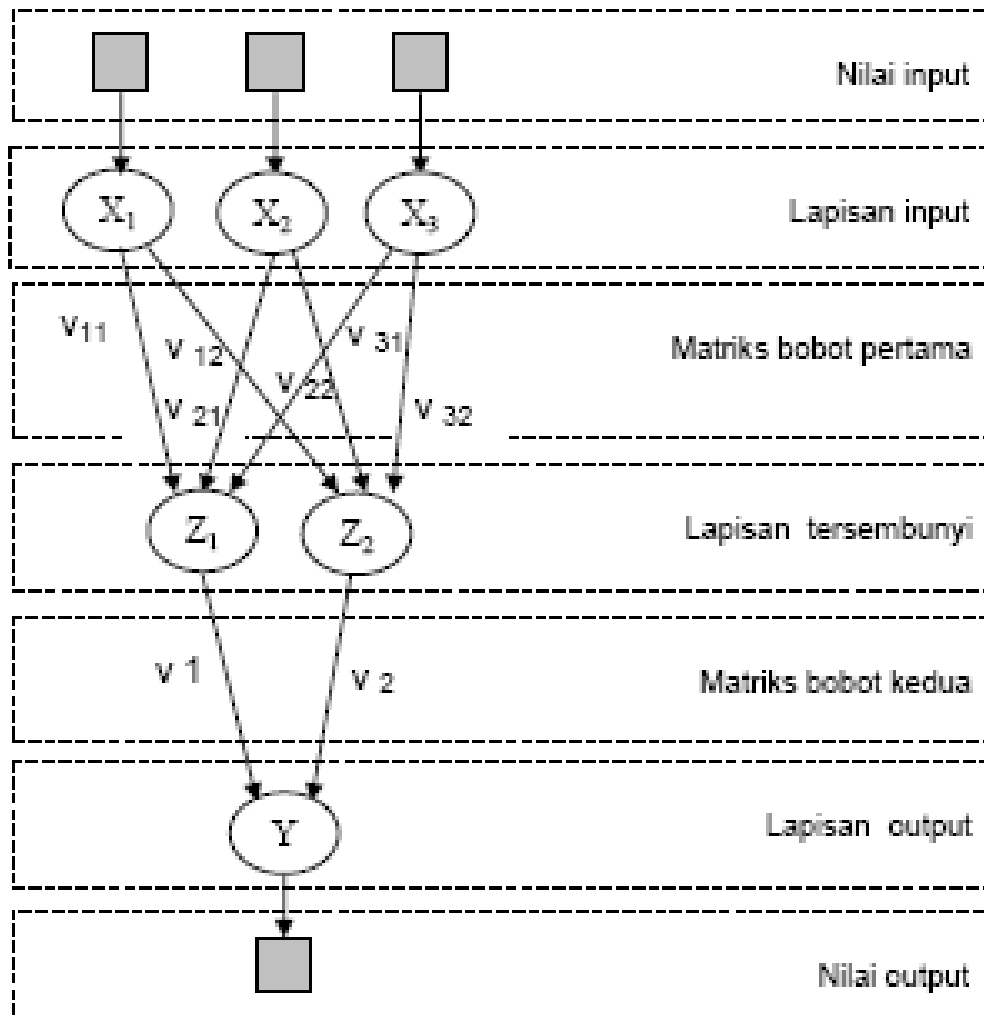
- Memiliki 1 atau lebih lapisan yang terletak diantara lapisan input dan lapisan output.
- Umumnya ada lapisan bobot-bobot yang terletak antara 2 lapisan yang bersebelahan.
- Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit.
- Pada banyak kasus, pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah.

Gambar berikut neuron-neuron pada kedua lapisan saling berhubungan.

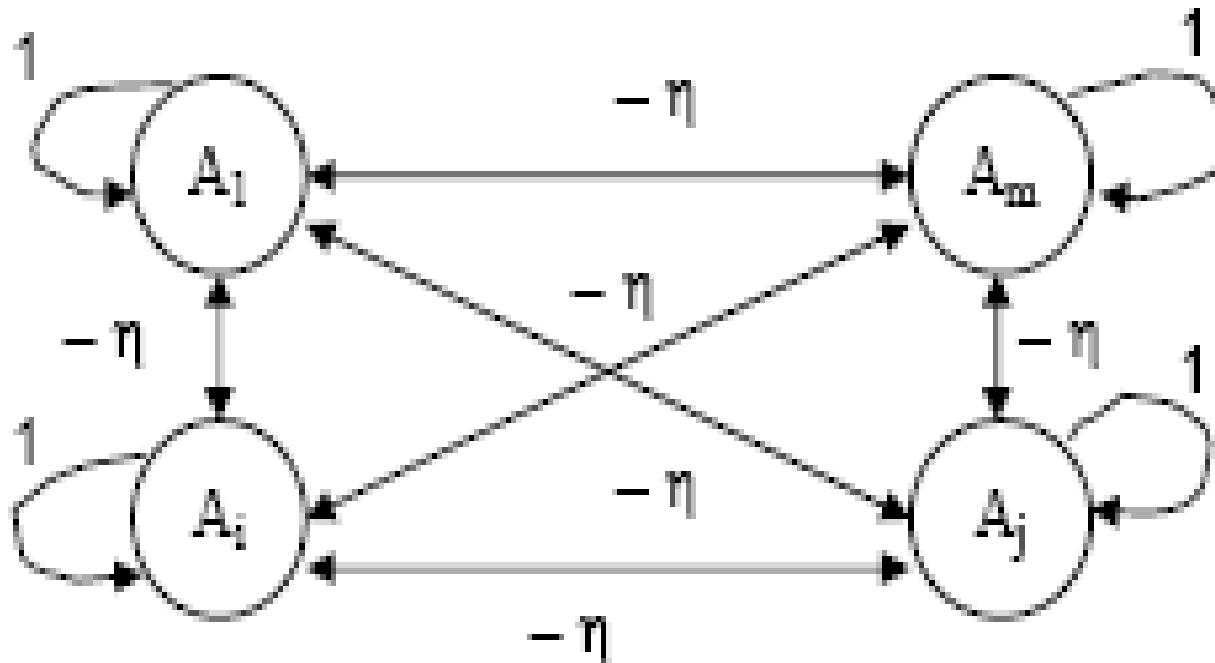


# Jaringan dengan lapisan kompetitif (competitive layer net)

- Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif.
- Umumnya hubungan antar neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur.
- Gambar berikut menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot  $-\eta$



# Jaringan dengan lapisan kompetitif (competitive layer 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.
- 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.

# JARINGAN SYARAF TIRUAN (JST)

- Hal yang ingin dicapai dengan melatih/mengajari JST adalah untuk mencapai keseimbangan antara kemampuan memorisasi dan generalisasi.
- Kemampuan memorisasi = kemampuan JST untuk memanggil kembali secara sempurna sebuah pola yang telah dipelajari.

# JARINGAN SYARAF TIRUAN (JST)

- 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.
- Hal ini sangat bermanfaat bila pada suatu saat ke dalam JST diinputkan informasi baru yang belum pernah dipelajari, maka JST masih akan tetap dapat memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati.

# APLIKASI JARINGAN SYARAF TIRUAN

- Aerospace : autopilot pesawat terbang, simulasi jalur penerbangan, sistem kendali pesawat, perbaikan autopilot, simulasi komponen pesawat
- Otomotif : sistem kendali otomatis mobil
- Keuangan dan perbankan : pendeteksian uang palsu, evaluator aplikasi kredit, pengidentifikasian pola-pola data pasar saham

# APLIKASI JARINGAN SYARAF TIRUAN

- Militer : Pengendali senjata, pendeteksi bom, penelusuran target, pembedaan objek, pengendali sensor, sonar, radar, dan pengolahan sinyal citra yang meliputi kompresi data, ekstraksi bagian istimewa, dan penghilangan derau, pengenalan sinyal atau citra.
- Elektronik : Pembuatan perangkat keras yang bisa mengimplementasikan JST secara efisien, machine vision, pengontrol gerakan

# APLIKASI JARINGAN SYARAF TIRUAN

- Broadcast : pencarian klip berita melalui pengenalan wajah
- Keamanan : JST digunakan untuk mengenali mobil dan mengenali wajah oknum
- Medis : analisis sel kanker
- Pengenalan suara : pengenalan percakapan, klasifikasi suara
- Pengenalan tulisan : pengenalan tulisan tangan, penerjemahan tulisan ke dalam tulisan latin

# APLIKASI JARINGAN SYARAF TIRUAN

- Matematika : alat pemodelan masalah dimana bentuk eksplisit dari hubungan antara variabel-variabel tertentu tidak diketahui
- Pengenalan benda bergerak : selain pola dari citra diam, JST juga bisa digunakan untuk mendeteksi citra bergerak dari video seperti citra orang yang bergerak, dll.
- JST digunakan sebagai detektor virus komputer, penginderaan bau, dll

# Paradigma/metode pembelajaran/pelatihan JST :

- 1. Pembelajaran terawasi (supervised learning)**
- 2. Pembelajaran tak terawasi (unsupervised learning) / pembelajaran tanpa guru**
- 3. Gabungan pembelajaran terawasi dan tak terawasi (hybrid)**

# **Pembelajaran terawasi (supervised learning)**

- Pada pembelajaran ini kumpulan input yang digunakan, output-outputnya telah diketahui.
- Perbedaan antara output-output aktual dengan output-output yang diinginkan digunakan untuk mengoreksi bobot JST agar JST dapat menghasilkan jawaban sedekat (semirip) mungkin dengan jawaban yang benar yang telah diketahui oleh JST.
- Contoh : Hebbian, Perceptron, Adaline, Back Propagation, LVQ (Learning Vector Quantization)

# **Pembelajaran tak terawasi (unsupervised learning) / pembelajaran tanpa guru**

- Pada pembelajaran ini, JST mengorganisasi dirinya sendiri untuk membentuk vektor-vektor input yang serupa, tanpa menggunakan data atau contoh-contoh pelatihan.
- Struktur menggunakan dasar data atau korelasi antara pola-pola data yang dieksplorasi.
- Contoh : Kohonen, ART

# **Gabungan pembelajaran terawasi dan tak terawasi (hybrid)**

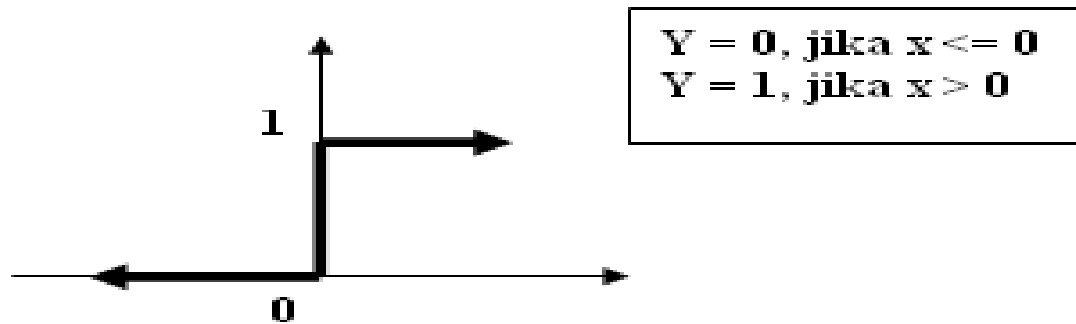
1. Merupakan kombinasi dari kedua pembelajaran tersebut.
2. Sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi.

# FUNGSI AKTIVASI

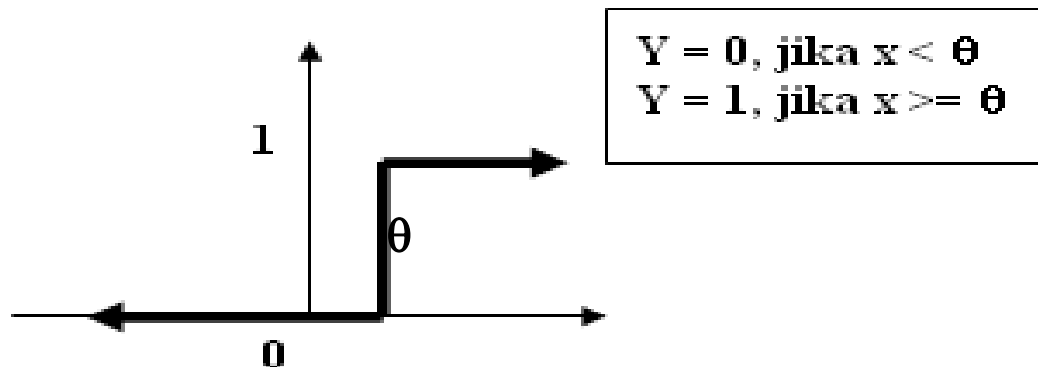
- Dipakai untuk menentukan keluaran suatu neuron
- Merupakan fungsi yang menggambarkan hubungan antara tingkat aktivasi internal (summation function) yang mungkin berbentuk linier atau nonlinear.
- Beberapa fungsi aktivasi JST diantaranya hard limit, purelin, dan sigmoid.

# Fungsi Aktivasi

- Fungsi undak biner (hard limit)

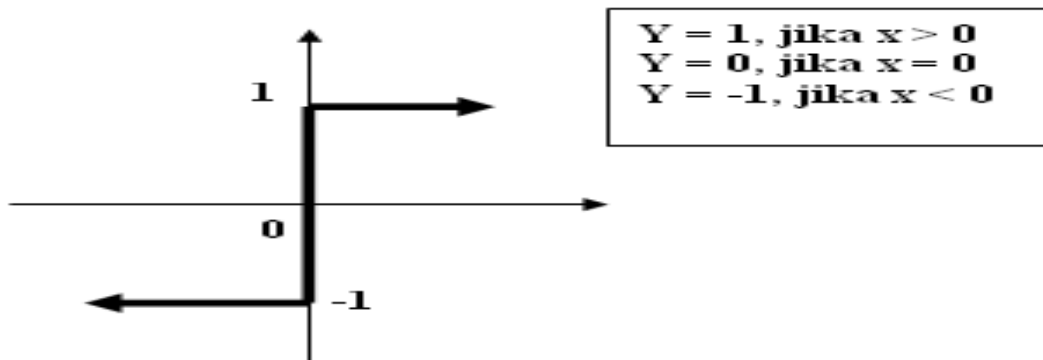


- Fungsi undak biner (threshold)

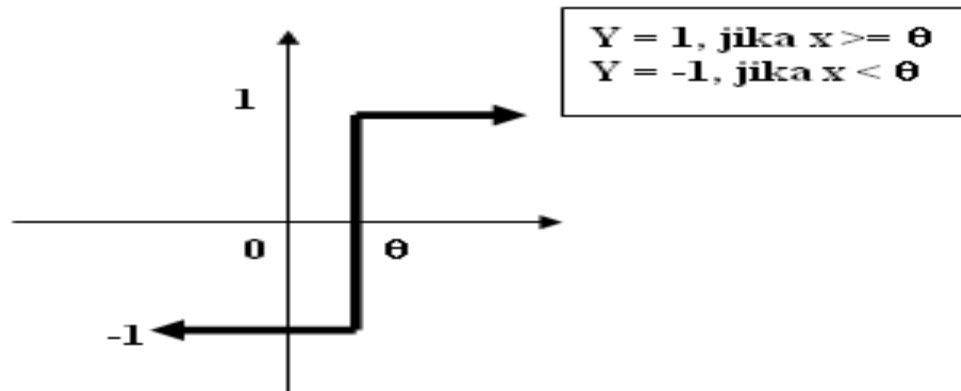


# Fungsi Aktivasi

- Fungsi bipolar

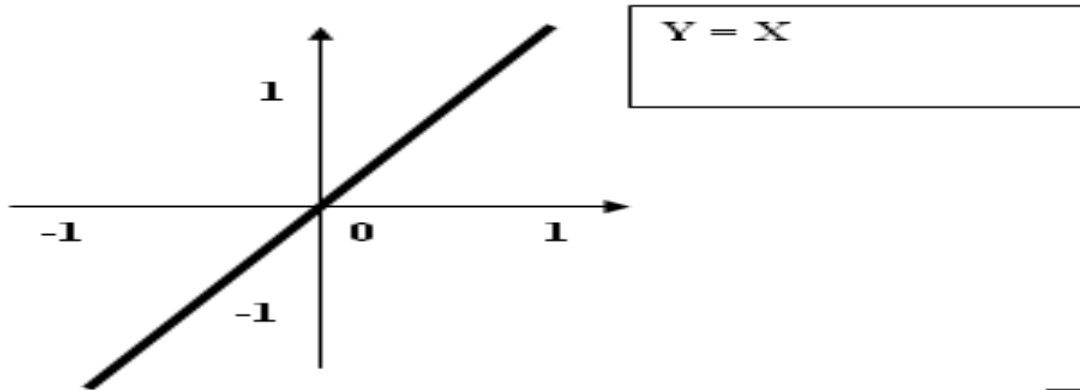


- Fungsi bipolar dengan threshold

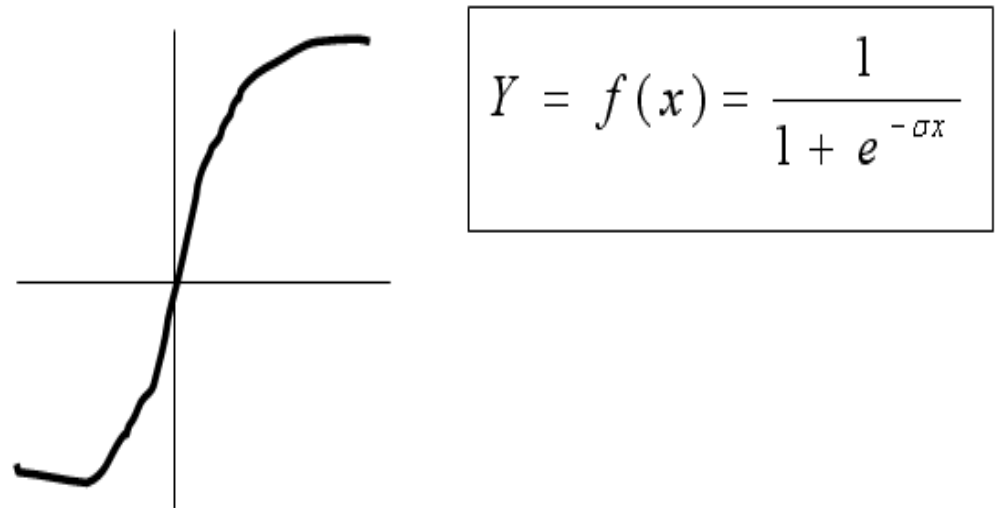


# Fungsi Aktivasi

- Fungsi Linier (identitas)

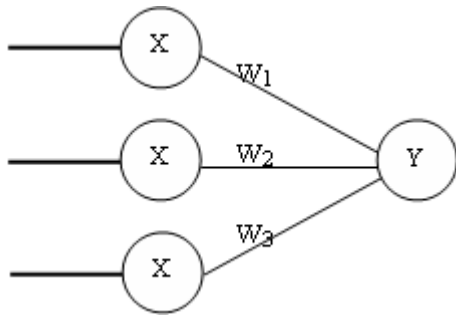


- Fungsi Sigmoid biner



# Model Neuron McCulloch-Pitts

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



- Contoh: 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 Neuron McCulloch-Pitts (untuk mengenali fungsi AND)

- Input :
  - $x_1$
  - $x_2$
- 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 seperti table di samping :
- Agar  $y(\text{net})$  memenuhi fungsi logika "AND", maka nilai ambang  $\theta$  pada fungsi aktivasi dibuat sama dengan 2 (silahkan coba mengubah nilai  $w$  dan nilai ambang)

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

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

# Model Neuron McCulloch-Pitts (untuk mengenali fungsi XOR)

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

$X_1$	$X_2$	$Y$
0	0	0
0	1	1
1	0	1
1	1	0

- karena fungsi logika “XOR” mempunyai 2 buah output yang bernilai “1”. Untuk menyelesaikan masalah ini, fungsi tersebut harus diubah dahulu menjadi

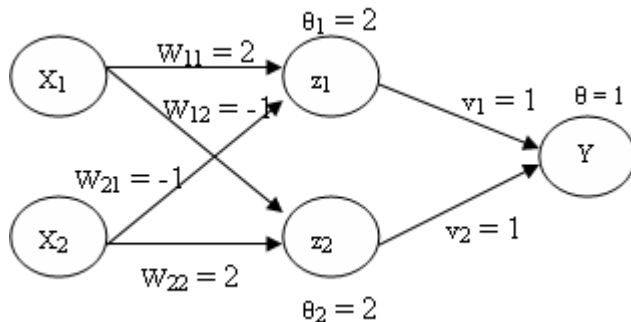
$$Y = X_1 \text{ XOR } X_2 \\ = (\sim X_1 \text{ AND } X_2) \text{ OR } (X_1 \text{ AND } \sim X_2)$$

Bila  $z_1 = (\sim X_1 \text{ AND } X_2)$  dan  $z_2 = (X_1 \text{ AND } \sim X_2)$ , maka

$$Y = z_1 \text{ OR } z_2$$

- Ini berarti unit masukan ( $X_1$  dan  $X_2$ ) harus berhubungan dahulu dengan sebuah layer tersembunyi ( $z_1$  dan  $z_2$ ) kemudian layer tersembunyi tersebut dihubungkan langsung dengan unit keluaran  $Y$ . Bila arsitektur jaringan dibuat seperti berikut

# Model Neuron McCulloch-Pitts (untuk mengenali fungsi XOR)



$X_1$	$X_2$	$net_1 = \sum x_i w$	$z_1(net) = \begin{cases} 0 & \text{jika } net_1 < 2 \\ 1 & \text{jika } net_1 \geq 2 \end{cases}$
0	0	$0.2 + 0.(-1) = 0$	0
0	1	$0.2 + 1.(-1) = -1$	0
1	0	$1.2 + 0.(-1) = 2$	1
1	1	$1.2 + 1.(-1) = 1$	0

- dari skema diatas, maka kita bisa menghitung jumlah seluruh input yang masuk untuk tiap-tiap data pada layar tersembunyi sebagai berikut :

$X_1$	$X_2$	$net_2 = \sum x_i w$	$z_2(net) = \begin{cases} 0 & \text{jika } net_2 < 2 \\ 1 & \text{jika } net_2 \geq 2 \end{cases}$
0	0	$0.(-1) + 0.2 = 0$	0
0	1	$0.(-1) + 1.2 = 2$	1
1	0	$1.(-1) + 0.2 = -1$	0
1	1	$1.(-1) + 1.2 = 1$	0

$$Y = X_1 \text{ XOR } X_2 \\ = (\sim X_1 \text{ AND } X_2) \text{ OR } (X_1 \text{ AND } \sim X_2)$$

Bila  $z_1 = (\sim X_1 \text{ AND } X_2)$  dan  $z_2 = (X_1 \text{ AND } \sim X_2)$ , maka

$$Y = z_1 \text{ OR } z_2$$

$z_1$	$z_2$	$net = \sum z_i v$	$Y(net) = \begin{cases} 0 & \text{jika } net < 1 \\ 1 & \text{jika } net \geq 1 \end{cases}$
0	0	$0.1 + 0.1 = 0$	0
0	1	$0.1 + 1.1 = 2$	1
1	0	$1.1 + 0.1 = 1$	1
0	0	$0.1 + 0.1 = 0$	0

# Model Neuron McCulloch-Pitts

- tampak bahwa untuk mengenali pola, model neuron McCulloch-Pitts, harus menentukan bobot  $w$  dan nilai ambang  $\theta$  secara analitik (dengan cara coba-coba) sehingga model neuron McCulloch-Pitts dapat mengenali pola tersebut.
- Jaringan seperti ini hanya bisa dibuat untuk merepresentasikan fungsi-fungsi yang sederhana. Tetapi untuk masalah-masalah yang kompleks misalnya jumlah input lebih dari 2 atau tidak biner maka cara seperti ini sulit dilakukan.