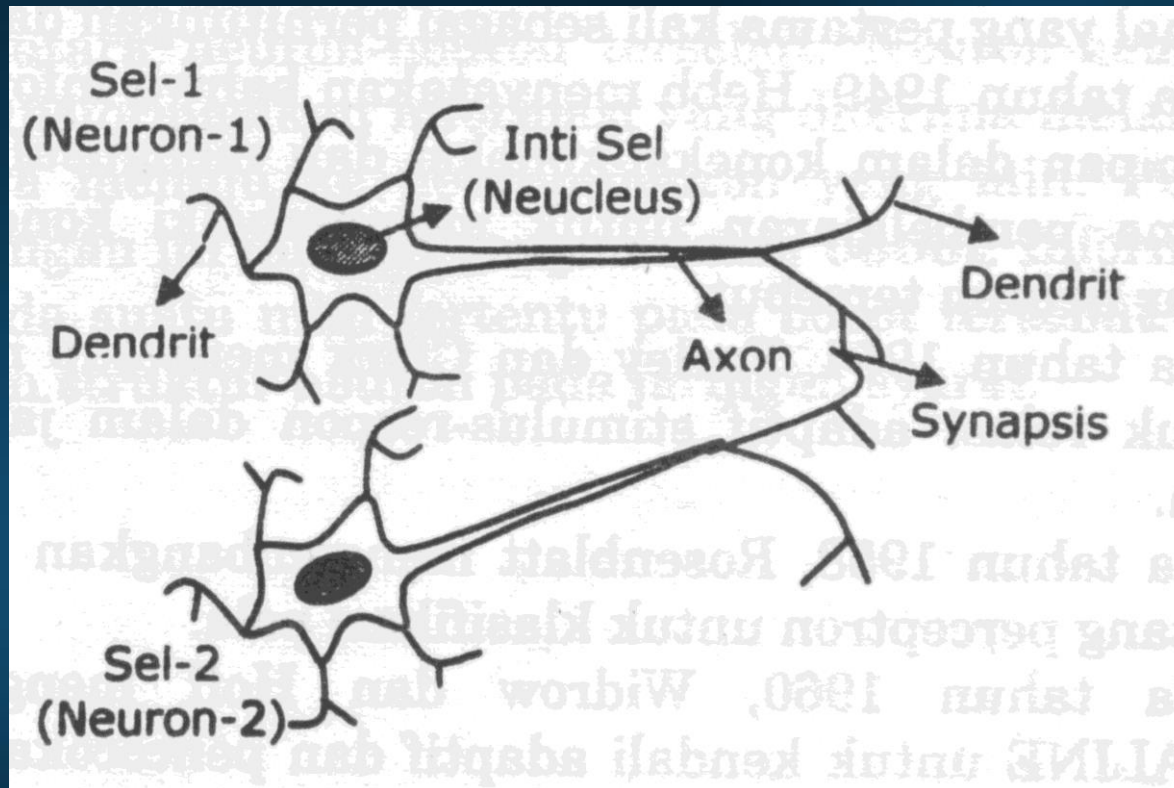




# Jaringan Syaraf Tiruan (JST)

# Susunan syaraf manusia





# Model sel syaraf

- Sel syaraf (*neuron*) adalah unit pemrosesan informasi yang merupakan dasar dari operasi JST.
- Terdapat tiga elemen dasar dari model neuron, yaitu:
  1. Sekumpulan sinapsis atau jalur hubungan, di mana masing-masing sinapsis memiliki bobot atau kekuatan hubungan.
  2. Suatu *adder* untuk menjumlahkan sinyal-sinyal input yang diberi bobot oleh sinapsis neuron yang sesuai. Operasi-operasi yang digambarkan di sini mengikuti aturan *linear combiner*.
  3. Suatu fungsi aktivasi untuk membatasi amplituda output setiap neuron.



# Pengertian JST

- JST merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu.
- JST mempunyai struktur tersebar paralel yang sangat besar dan mempunyai kemampuan belajar, sehingga bisa melakukan generalisasi, yaitu bisa menghasilkan output yang benar untuk input yang belum pernah dilatihkan.



# JST

- Baik tidaknya suatu model JST ditentukan oleh:
  - Pola antar neuron (arsitektur jaringan)
  - Metode untuk menentukan dan mengubah bobot (disebut metode learning)
  - Fungsi aktivasi
- JST disebut juga: brain metaphor, computational neuroscience, parallel distributed processing



# PEMROGRAMAN KONVENSIONAL VS JST

- Pemrograman konvensional :
  - ❑ Menggunakan pendekatan algoritmik, yaitu dengan mengikuti seperangkat instruksi untuk memecahkan masalah.
  - ❑ Kita harus mengerti masalah tsb dan harus mengetahui cara menyelesaikan masalah tsb.



# JST

- JST dapat belajar dari pengalaman!
- Biasanya berhubungan dengan angka (numerik) sehingga data yang tidak numerik harus dibuat ke numerik
- Tidak ada rumus yang tetap (fixed) sehingga disebut dengan free-estimator!
- JST disebut black box atau tidak transparan karena tidak mampu menjelaskan bagaimana suatu hasil didapatkan!
- JST mampu menyelesaikan permasalahan yang tidak terstruktur dan sulit didefinisikan!



# Kelebihan JST

1. Belajar adaptive : kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. Self-Organisation : sebuah JST dapat membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. Real time operation : perhitungan JST dapat dilakukan secara paralel sehingga perangkat keras yg dirancang dan diproduksi sccr khusus dapat mengambil keuntungan dari kemampuan ini.



# Kelemahan JST

- Kurang mampu untuk melakukan operasi operasi numerik dengan presisi tinggi
- Kurang mampu melakukan operasi algoritma aritmatik, operasi logika dan simbolis
- Lamanya proses training yang mungkin terjadi dalam waktu yang sangat lama untuk jumlah data yang besar



# Aplikasi JST

- Pengenalan pola (pattern recognition)
  - Huruf, tanda tangan, suara, gambar yang sudah sedikit berubah (mengandung noise)
  - Identifikasi pola saham
  - Pendeteksian uang palsu, kanker
- Signal Processing
  - Menekan noise pada saluran telepon
- Peramalan
  - Peramalan saham
- Autopilot dan simulasi
- Kendali otomatis otomotif



# Sejarah

- Model JST formal pertama diperkenalkan oleh McCulloch dan Pitts (1943)
- 1949, Hebb mengusulkan jaringan Hebb
- 1958, Rosenblatt mengembangkan perceptron untuk klasifikasi pola
- 1960, Widrow dan Hoff mengembangkan ADALINE dengan aturan pembelajaran Least Mean Square (LMS)
- 1974, Werbos memperkenalkan algoritma backpropagation untuk perceptron banyak lapisan



# Sejarah

- 1982, Kohonen mengembangkan learning unsupervised untuk pemetaan
- 1982, Grossberg dan Carpenter mengembangkan Adaptive Resonance Theory (ART, ART2, ART3)
- 1982, Hopfield mengembangkan jaringan Hopfield untuk optimasi
- 1985, Algoritma Boltzmann untuk jaringan syaraf probabilistik
- 1987, dikembangkan BAM (Bidirectional Associative Memory)
- 1988, dikembangkan Radial Basis Function



# Model Neuron

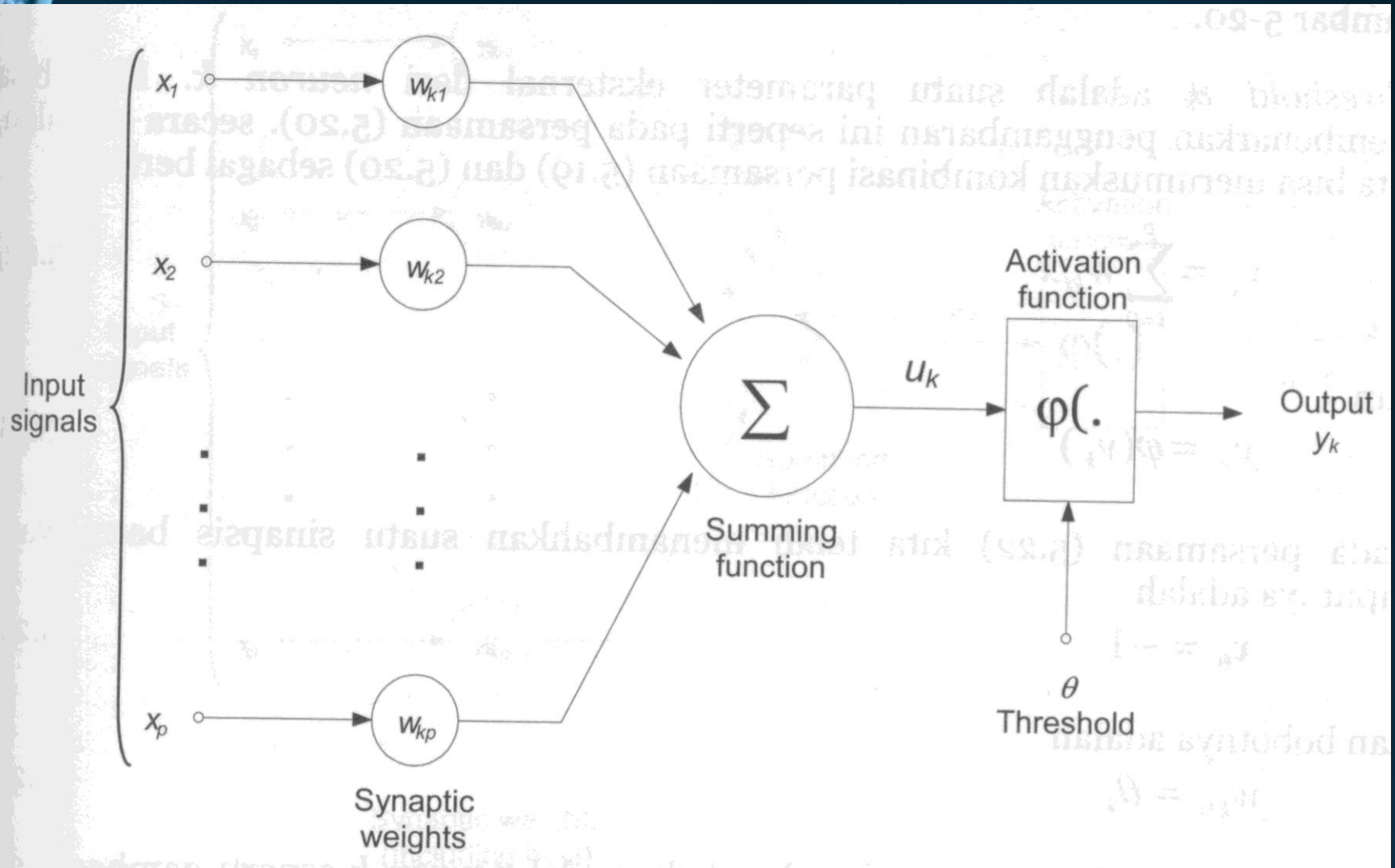
- Pada JST, neuron-neuron akan dikumpulkan dalam lapisan lapisan yang disebut dengan layers
- Neuron dalam satu lapisan akan dihubungkan dengan neuron pada lapisan lainnya
- Kadang muncul juga layer tersembunyi (hidden layer) untuk menambah keakuratan pelatihan
- Informasi tersebut bisa dirambatkan secara forward ataupun backward




# Istilah dalam JST


- Neuron: sel syaraf tiruan yang merupakan elemen pengolah JST
- Jaringan: bentuk arsitektur JST, kumpulan neuron yang saling berhubungan dan membentuk lapisan
- Input: sebuah nilai input yang akan diproses menjadi nilai output
- Output: solusi dari nilai input
- Hidden layer: lapisan yang tidak terkoneksi secara langsung dengan lapisan input atau output, memperluas kemampuan JST
- Bobot: nilai matematis dari sebuah koneksi antar neuron
- Fungsi aktivasi: fungsi yang digunakan untuk mengupdate nilai-nilai bobot per-iterasi dari semua nilai input.
  - Fungsi aktivasi sederhana adalah mengkalikan input dengan bobotnya dan kemudian menjumlahkannya (disebut penjumlahan sigma)
  - Berbentuk linier atau tidak linier, dan sigmoid
- Paradigma pembelajaran: bentuk pembelajaran, supervised learning, atau unsupervised learning

# Model sel syaraf



- 
- Sebuah neuron bisa memiliki banyak masukan tetapi hanya memiliki satu keluaran yang bisa menjadi masukan bagi neuron-neuron yang lain.
  - Pada gambar terlihat serangkaian sinyal masukan  $x_1, x_2, \dots, x_p$ .
  - Tiap sinyal masukan dikalikan dengan suatu bobot ( $w_{k1}, w_{k2}, \dots, w_{kp}$ ) dan kemudian semua masukan yang telah diboboti tadi dijumlahkan untuk mendapatkan output kombinasi linear  $u_k$ .

$$u_k = \sum_{j=1}^p w_{kj} x_j$$

- 
- Selanjutnya  $u_k$  akan diinputkan ke suatu fungsi aktivasi  $\varphi(\cdot)$  untuk menghasilkan output dari neuron tersebut  $y_k$ .
  - Suatu nilai threshold atau bias ( $\theta$ ) dapat ditambahkan untuk menyesuaikan nilai masukan ke fungsi aktivasi.

$$y_k = \varphi(u_k - \theta_k)$$

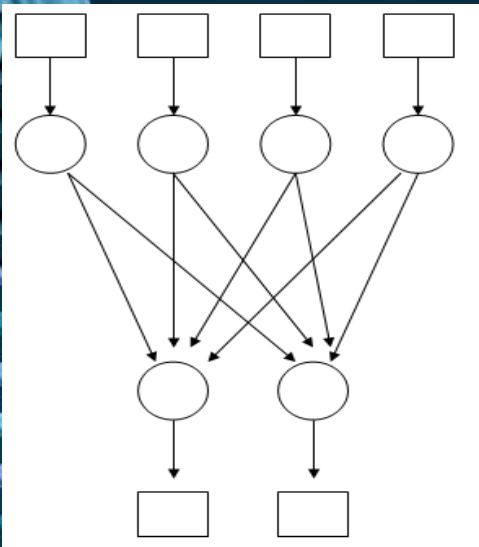


# Arsitektur Jaringan

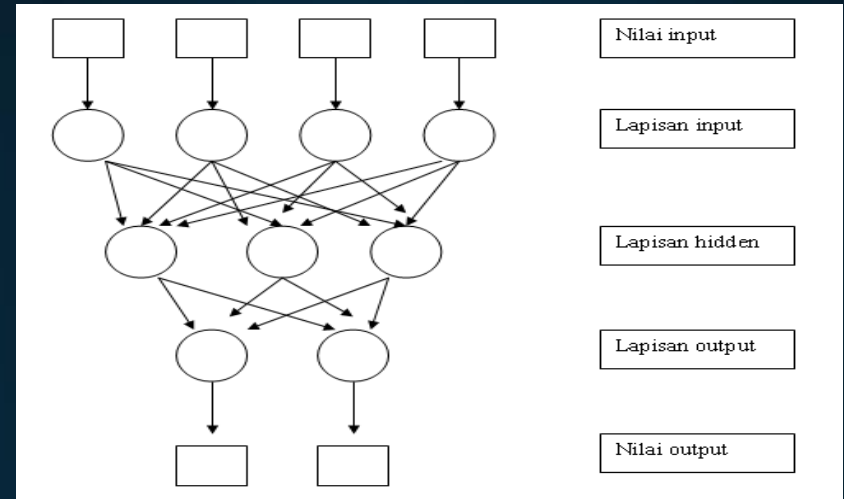
- **Single Layer**
  - Hanya memiliki satu lapisan dengan bobot-bobot terhubung.
  - Langsung menerima input dan mengolahnya menjadi output tanpa menggunakan hidden layer
- **Multi Layer**
  - Memiliki satu atau lebih lapisan input, satu atau lebih lapisan output, dan lapisan tersembunyi
  - Dapat menyelesaikan masalah yang lebih kompleks karena lebih akurat
  - Fungsi pembelajarannya lebih rumit
- **Kompetitive Model / Recurrent Model**
  - Hubungan antar neuron tidak diperlihatkan secara langsung pada arsitektur
  - Hubungan antar neuron dapat digambarkan sebagai jaring yang rumit

# Lapisan Penyusun JST

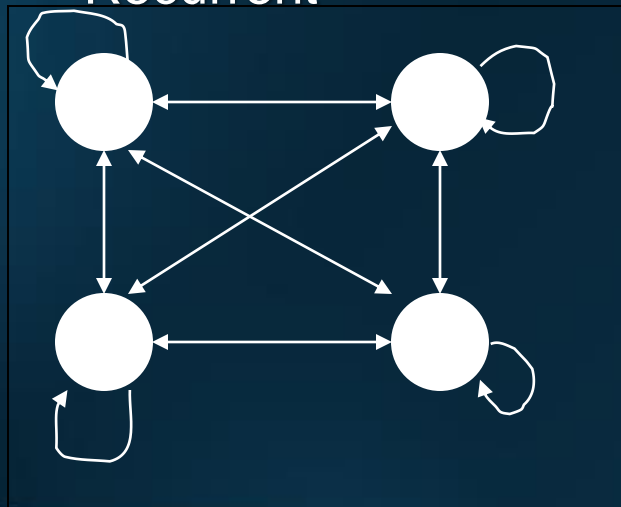
- Single Layer



- Multi Layer



- Competitive Layer / Recurrent





# Pengelompokkan JST

- JST Feed Forward
  - Tidak mempunyai loop
  - Contoh: single layer perceptron, multilayer perceptron, radial basis function
- JST Feed Backward (Recurrent)
  - Memiliki loop, lapisan output akan memberi input lagi bagi lapisan input
  - Contoh: competitive networks, kohonen, hopfield, ART



# Paradigma pembelajaran

- Supervised Learning
  - Kumpulan input berusaha membentuk target output yang sudah diketahui sebelumnya
  - Perbedaan antara output yang masih salah dengan output yang diharapkan harus sekecil mungkin
  - Biasanya lebih baik daripada unsupervised
  - Kelemahan: pertumbuhan waktu komputasi eksponensial, data bnyk berarti semakin lambat
- Unsupervised Learning
  - JST mengorganisasikan dirinya untuk membentuk vektor-vektor input yang serupa tanpa menggunakan data atau contoh-contoh pelatihan, biasanya ke dalam suatu kategori/kelompok2 tertentu

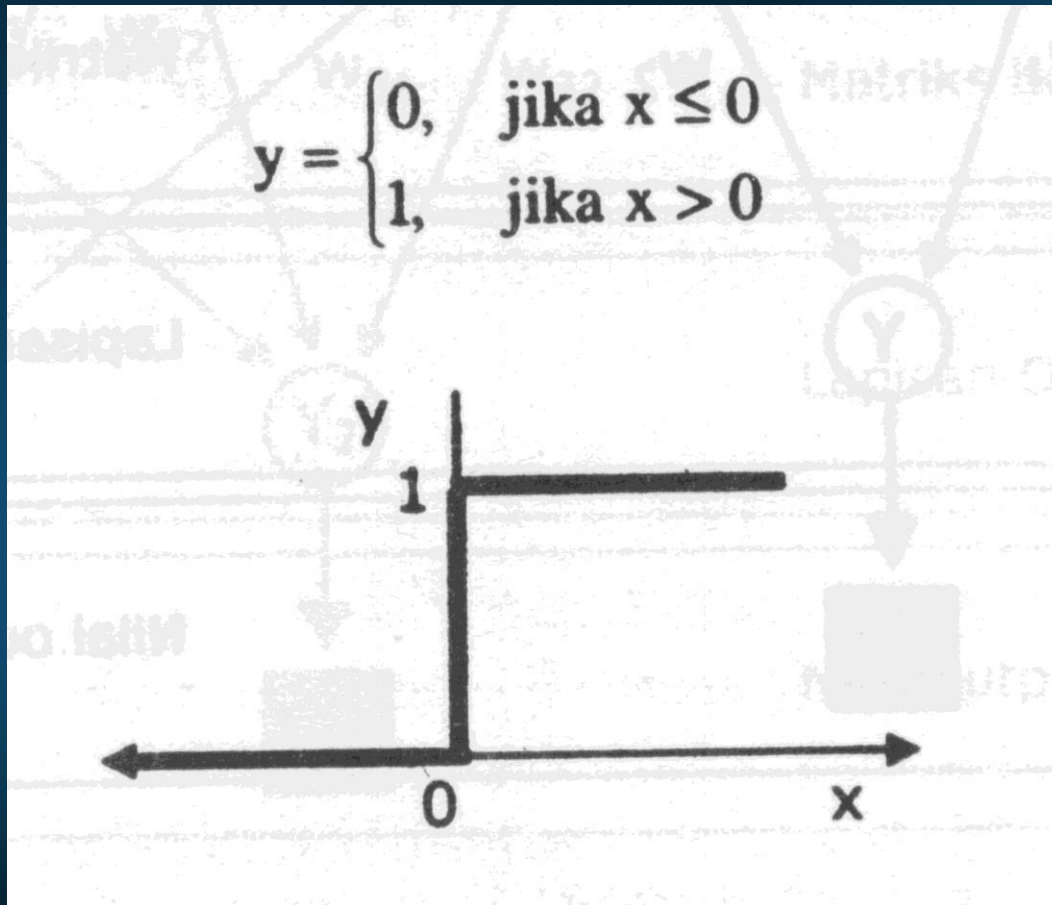


# Fungsi Aktivasi

- Fungsi aktivasi yang dinotasikan dengan  $\phi(\cdot)$  mendefinisikan nilai output dari suatu neuron dalam level aktivitas tertentu berdasarkan nilai output pengkombinasi linier  $u_i$ .
- Ada beberapa macam fungsi aktivasi yang biasa digunakan, di antaranya adalah:
  - Hard Limit
  - Threshold
  - Symetric Hard Limit
  - Fungsi linear (identitas)
  - Fungsi Saturating Linear
  - Fungsi Sigmoid Biner
  - Fungsi Sigmoid Bipolar

# Fungsi Hard Limit

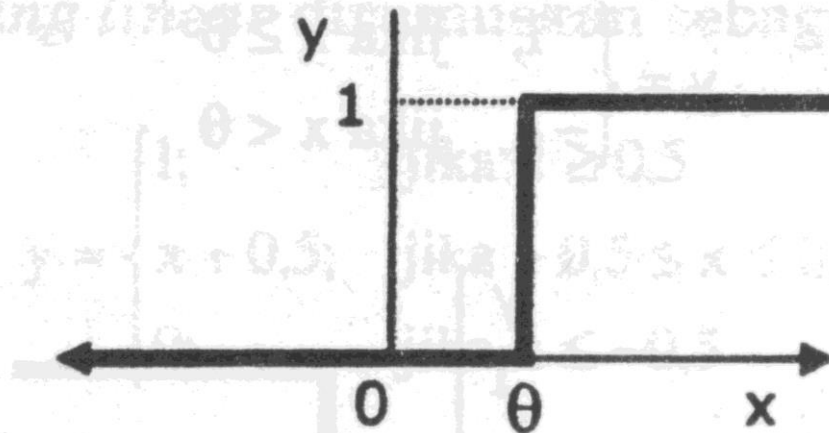
- Fungsi hard limit dirumuskan sebagai:



# Hard Limit (dengan threshold)

- Fungsi hard limit dengan threshold  $\theta$  dirumuskan sebagai:

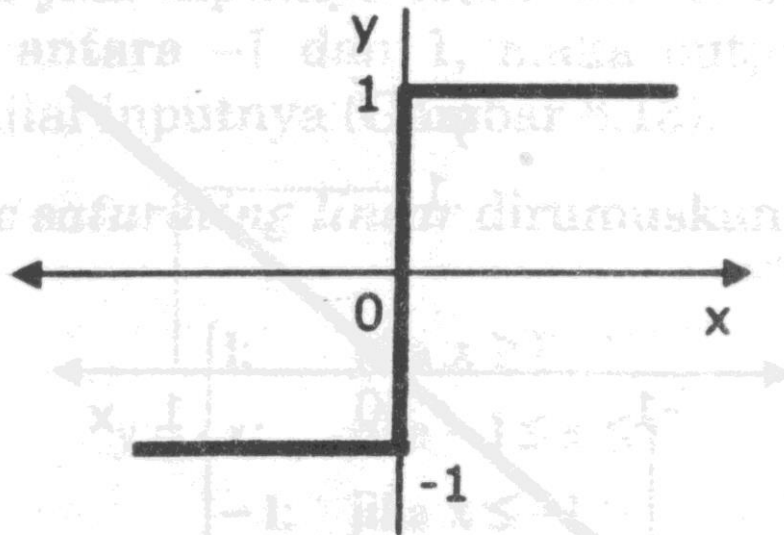
$$y = \begin{cases} 0, & \text{jika } x < \theta \\ 1, & \text{jika } x \geq \theta \end{cases}$$



# Symmetric Hard Limit

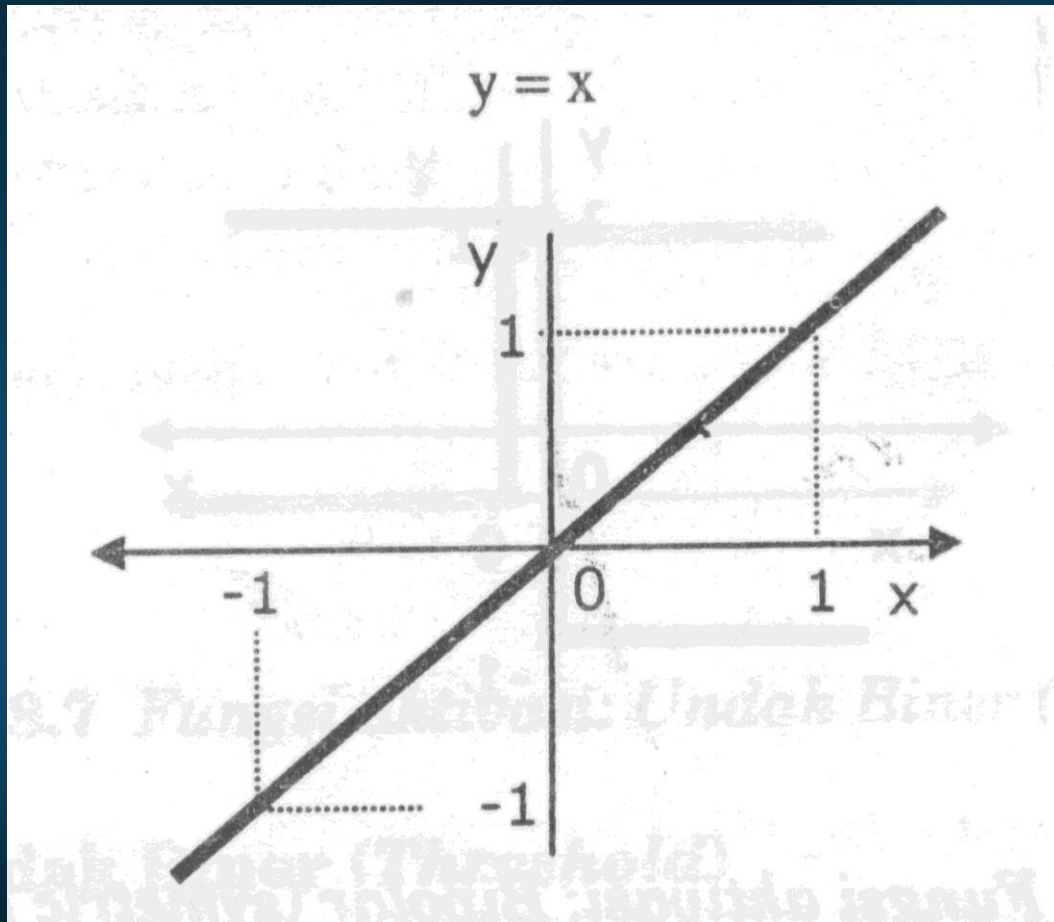
- Fungsi symmetric hard limit dirumuskan sebagai:

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



# Fungsi linear (identitas)

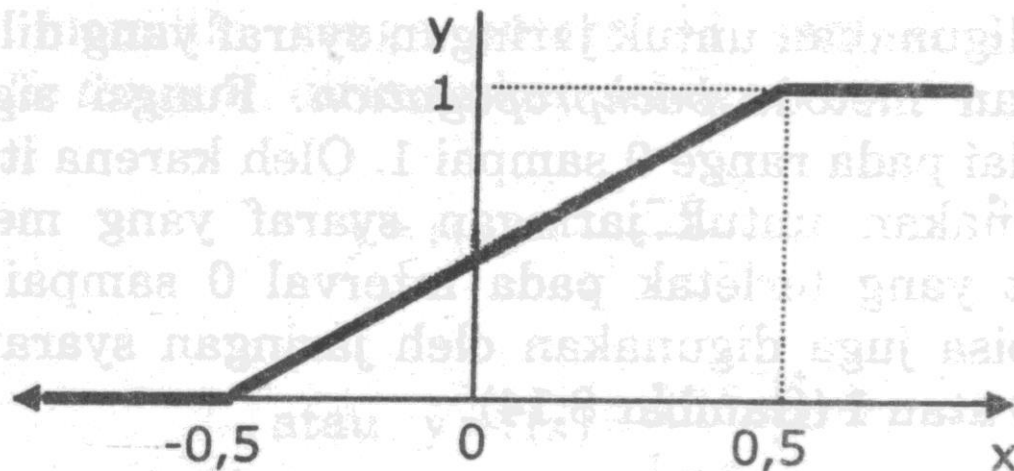
- Fungsi linear dirumuskan sebagai:



# Fungsi Saturating Linear

- Fungsi saturating linear dirumuskan sebagai:

$$y = \begin{cases} 1; & \text{jika } x \geq 0,5 \\ x + 0,5; & \text{jika } -0,5 \leq x \leq 0,5 \\ 0; & \text{jika } x \leq -0,5 \end{cases}$$

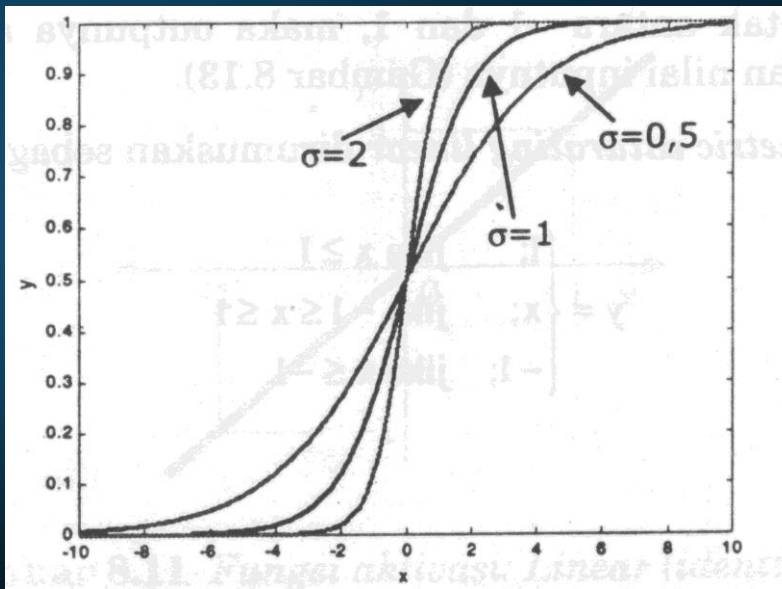


# Fungsi Sigmoid Biner

- Fungsi sigmoid biner dirumuskan sebagai:

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

dengan  $f' = \sigma f(x)[1 - f(x)]$



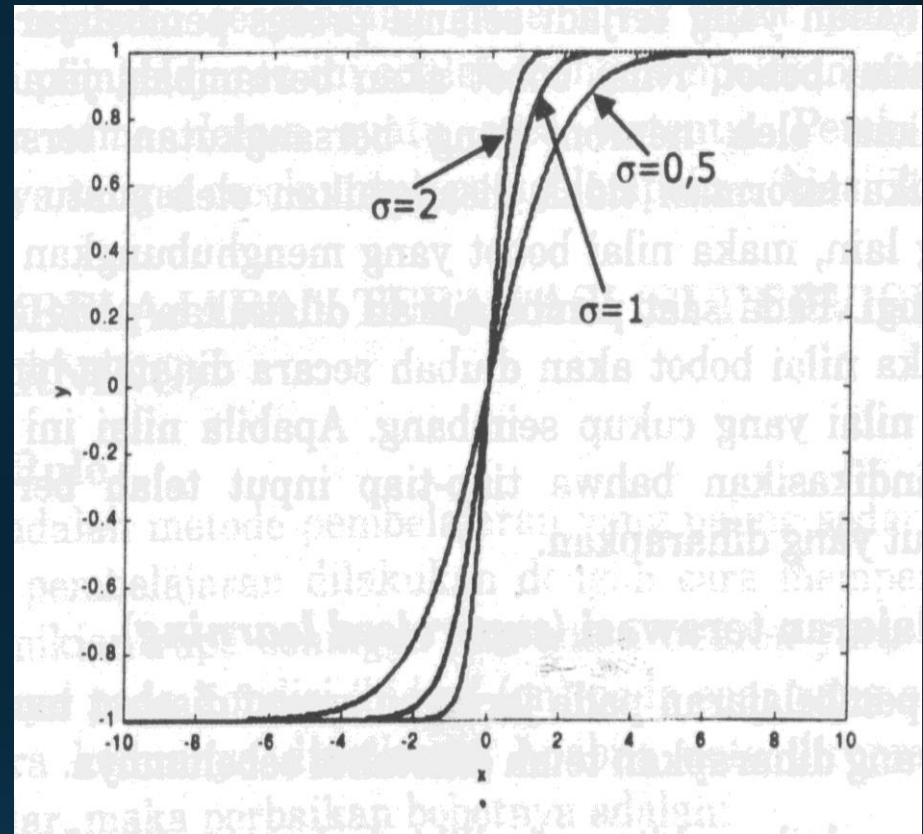
# Fungsi Sigmoid Bipolar

- Fungsi sigmoid bipolar dirumuskan sebagai:

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

dengan :

$$f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)]$$





# McCulloch Pitts

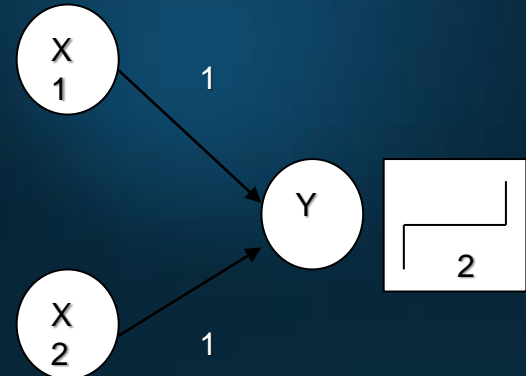
- Fungsi aktivasi biner
- Besar bobotnya sama
- Memiliki threshold yang sama

Contoh buat fungsi logika “and”, input X1 dan X2, dan Y = 1 jika dan hanya jika inputan 1

X1	X2	Y
1	1	1
1	0	0
0	1	0
0	0	0

# Jawab

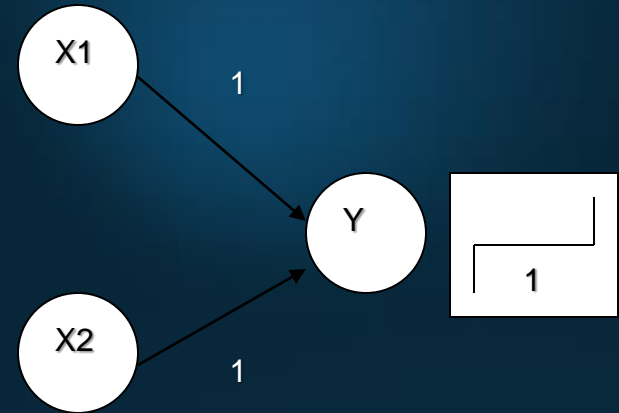
X1	X2	net	Y(net), 1 jika net $\geq 2$ , 0 jika net $< 2$
1	1	$1 \cdot 1 + 1 \cdot 1 = 2$	1
1	0	$1 \cdot 1 + 0 \cdot 1 = 1$	0
0	1	$0 \cdot 1 + 1 \cdot 1 = 1$	0
0	0	$0 \cdot 1 + 0 \cdot 1 = 0$	0



Ternyata BERHASIL mengenali pola

# Problem "OR"

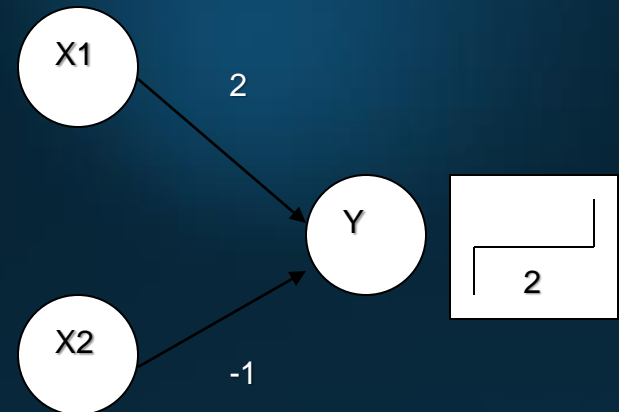
X1	X2	net	Y(net), 1 jika net $\geq 1$ , 0 jika net $< 1$
1	1	$1 \cdot 1 + 1 \cdot 1 = 2$	1
1	0	$1 \cdot 1 + 0 \cdot 1 = 1$	1
0	1	$0 \cdot 1 + 1 \cdot 1 = 1$	1
0	0	$0 \cdot 1 + 0 \cdot 1 = 0$	0



Ternyata BERHASIL mengenali pola

# Problem “X1 and not(X2)”

X1	X2	net	Y(net), 1 jika net $\geq 2$ , 0 jika net $< 2$
1	1	$1 \cdot 2 + 1 \cdot (-1) = 1$	0
1	0	$1 \cdot 2 + 0 \cdot (-1) = 2$	1
0	1	$0 \cdot 2 + 1 \cdot (-1) = -1$	0
0	0	$0 \cdot 2 + 0 \cdot (-1) = 0$	0

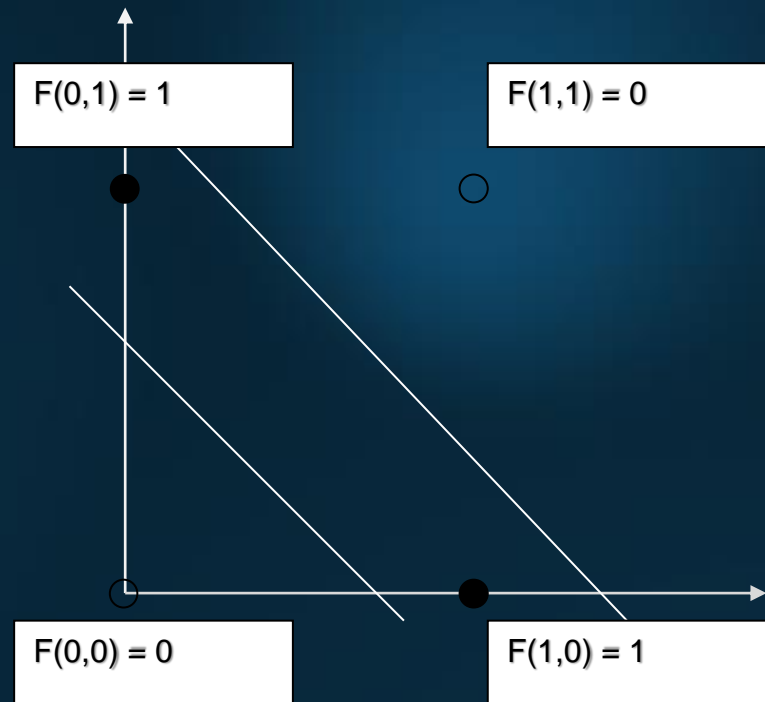


Ternyata BERHASIL mengenali pola

# Problem "XOR"

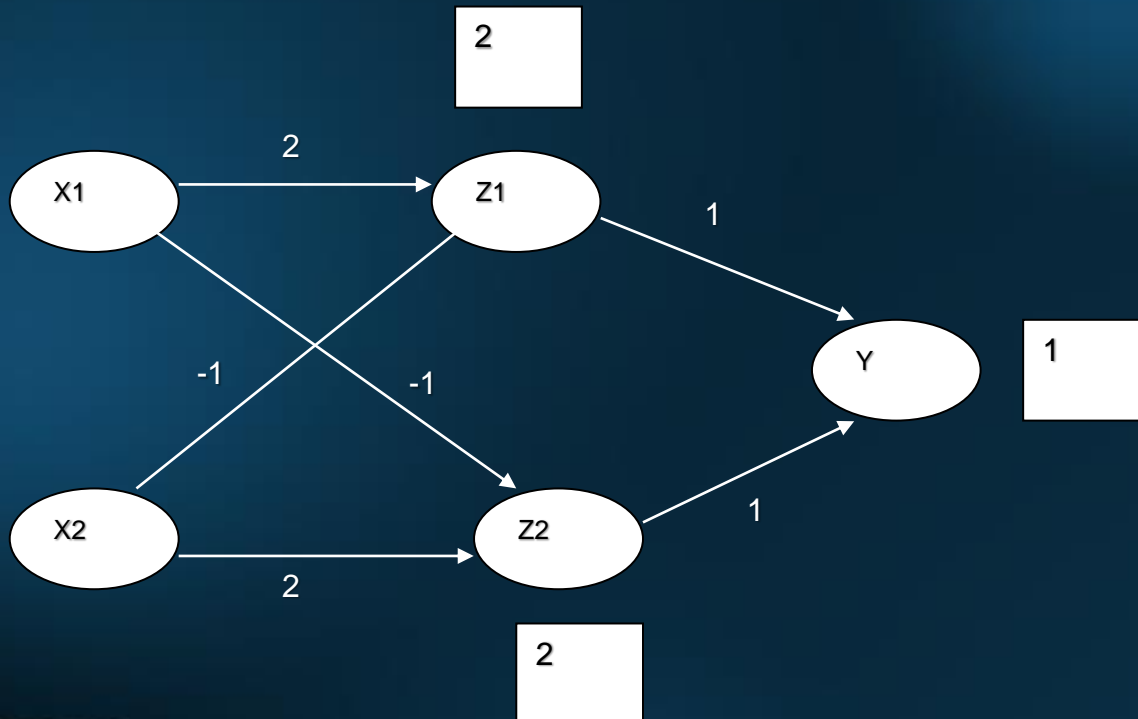
X1	X2	Y
1	1	0
1	0	1
0	1	1
0	0	0

GAGAL!



# Solusi

- $XOR = (x1 \wedge \sim x2) \vee (\sim x1 \wedge x2)$
- Ternyata dibutuhkan sebuah layer tersembunyi



# Tabel

#

X1	X2	Net1	F(net1)	Net2	F(net2)
1	1	$1.2+1.-1=1$	0	$1.-1+1.2=1$	0
1	0	$1.2+0.-1=2$	1	$1.-1+0.2=-1$	0
0	1	$0.2+1.-1=-1$	0	$0.-1+1.2=2$	1
0	0	$0.2+0.-1=0$	0	$0.-1+0.2=0$	0

Kemudian Z1 dan Z2 diteruskan ke Y, dengan bobot Z1=1 dan Z2=1

Z1	Z2	Net	F(net) = Y
0	0	$0.1+0.1=0$	0
1	0	$1.1+0.1=1$	1
0	1	$0.1+1.1=1$	1
0	0	$0.1+0.1=0$	0