

PENGENALAN KECERDASAN BUATAN

Artificial Intelligence

(2 SKS)

/ AI

Materi ke 8

Ir. Ahmad Haidaroh, M.Kom.
STIKOM Artha Buana.

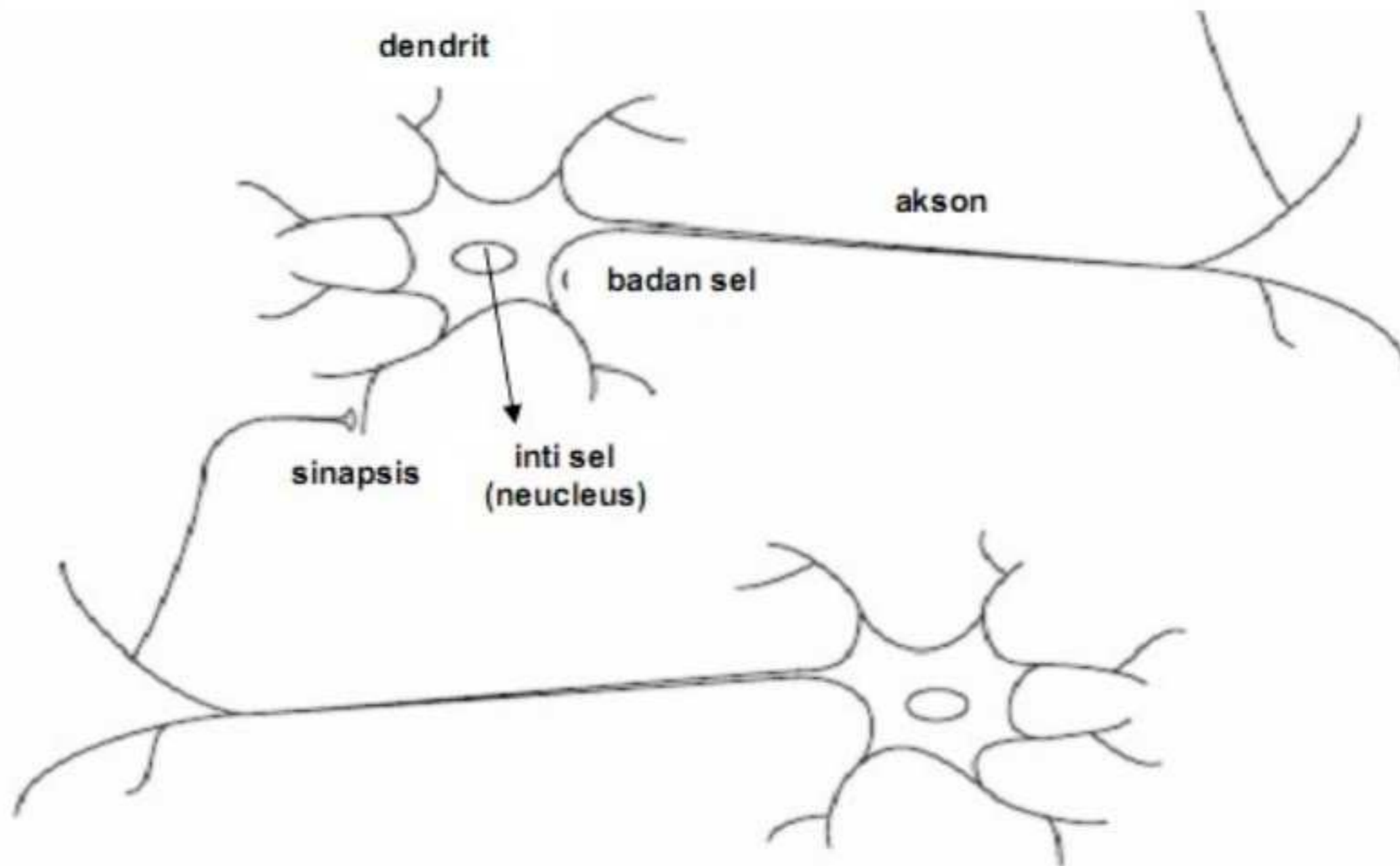


Jaringan Syaraf Tiruan (JST)

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**

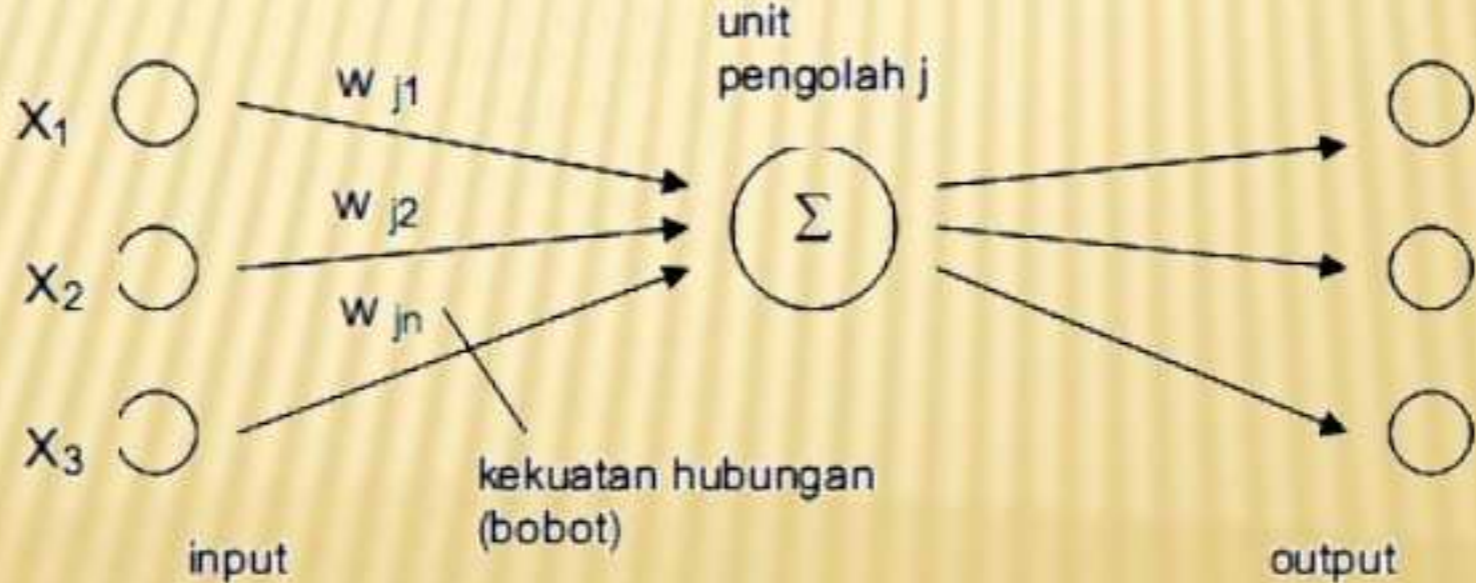
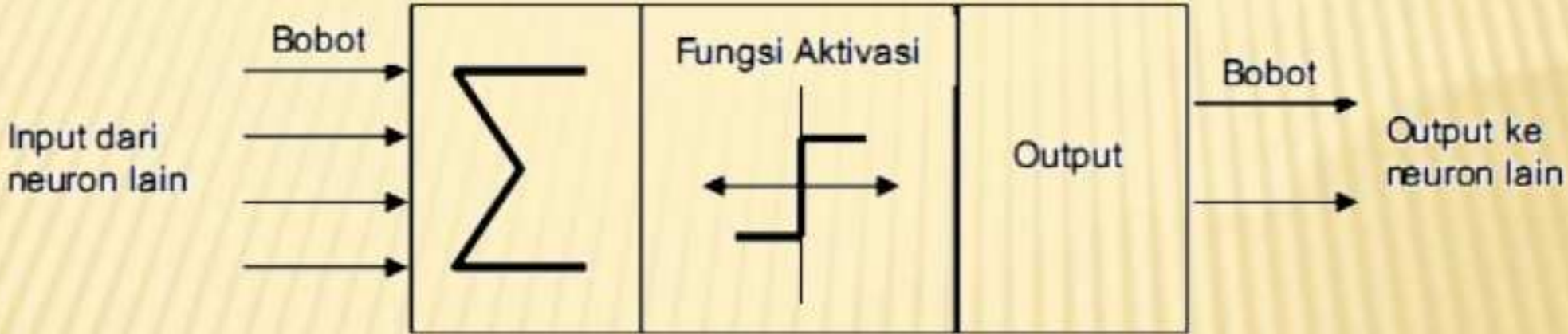
JARINGAN SYARAF TIRUAN

- ▢ 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.

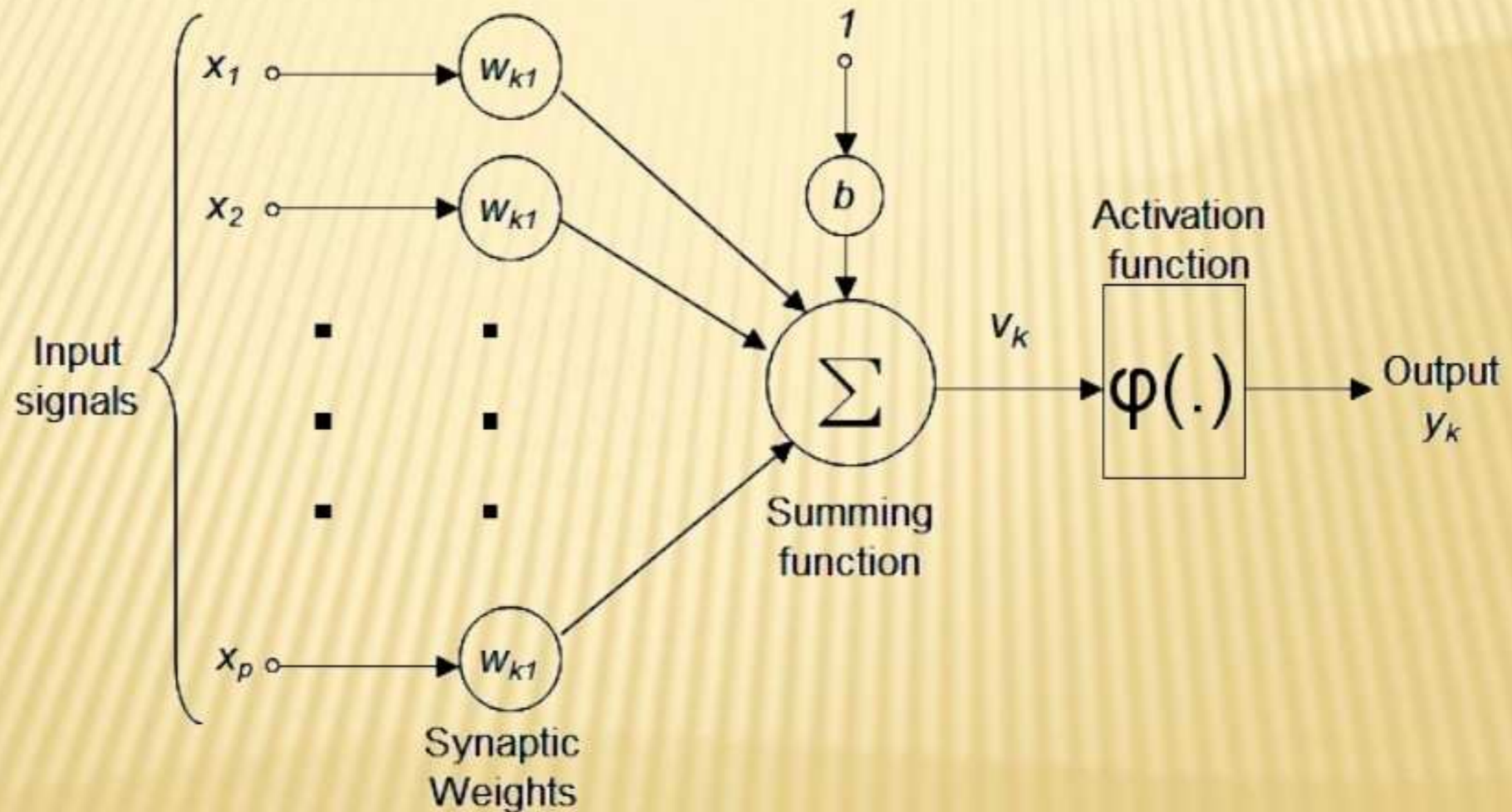
ANALOGI JST DENGAN JSB

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

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_{ij}) 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.

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. Yang populer digunakan adalah fungsi sigmoid yang memiliki beberapa varian : sigmoid logaritma, sigmoid biner, sigmoid bipolar, sigmoid tangen.

FUNGSI AKTIVASI

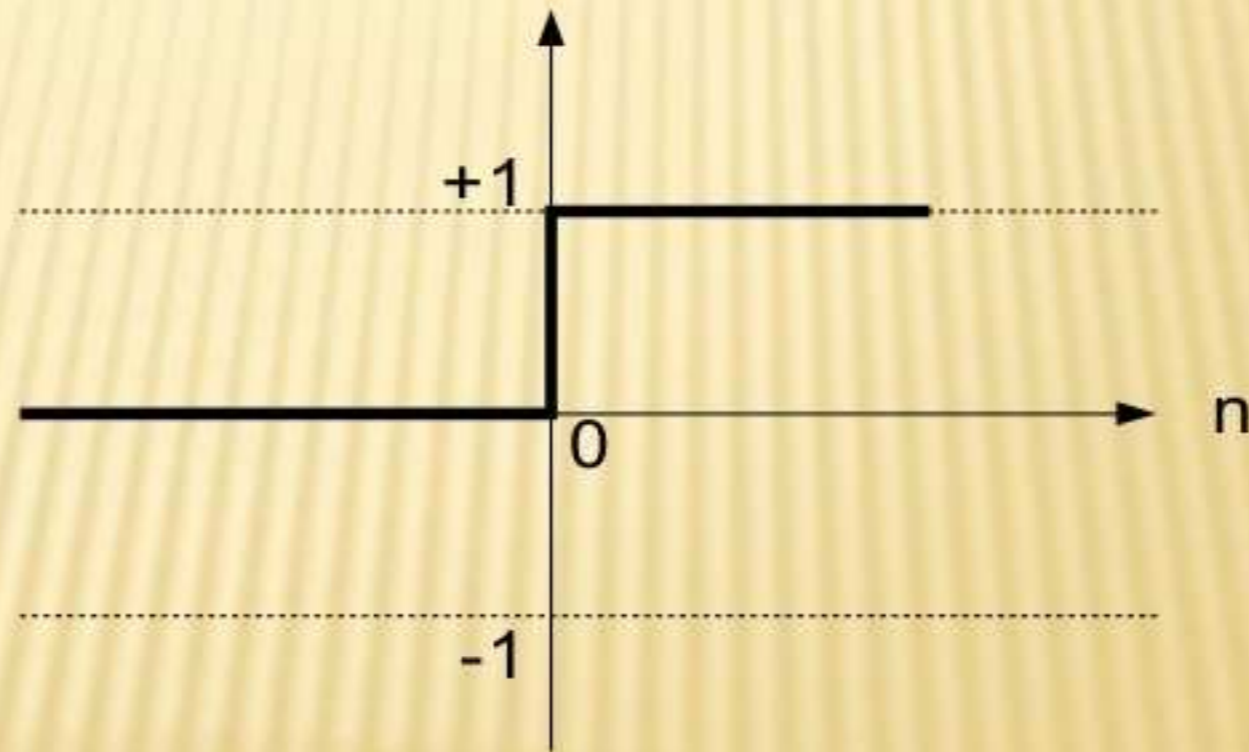
- ◆ Hard limit memberikan batasan tegas 0 atau 1, purelin memisahkan secara linier, sigmoid berupa fungsi smooth bernilai antara 0 sampai dengan 1 (bila biner) atau antara -1 sampai 1 (bila bipolar)

FUNGSI AKTIVASI

Hard limit function

$a = 0$ jika $n < 0$

$a = 1$ jika $n \geq 0$



$a = \text{hardlim} (n)$

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.

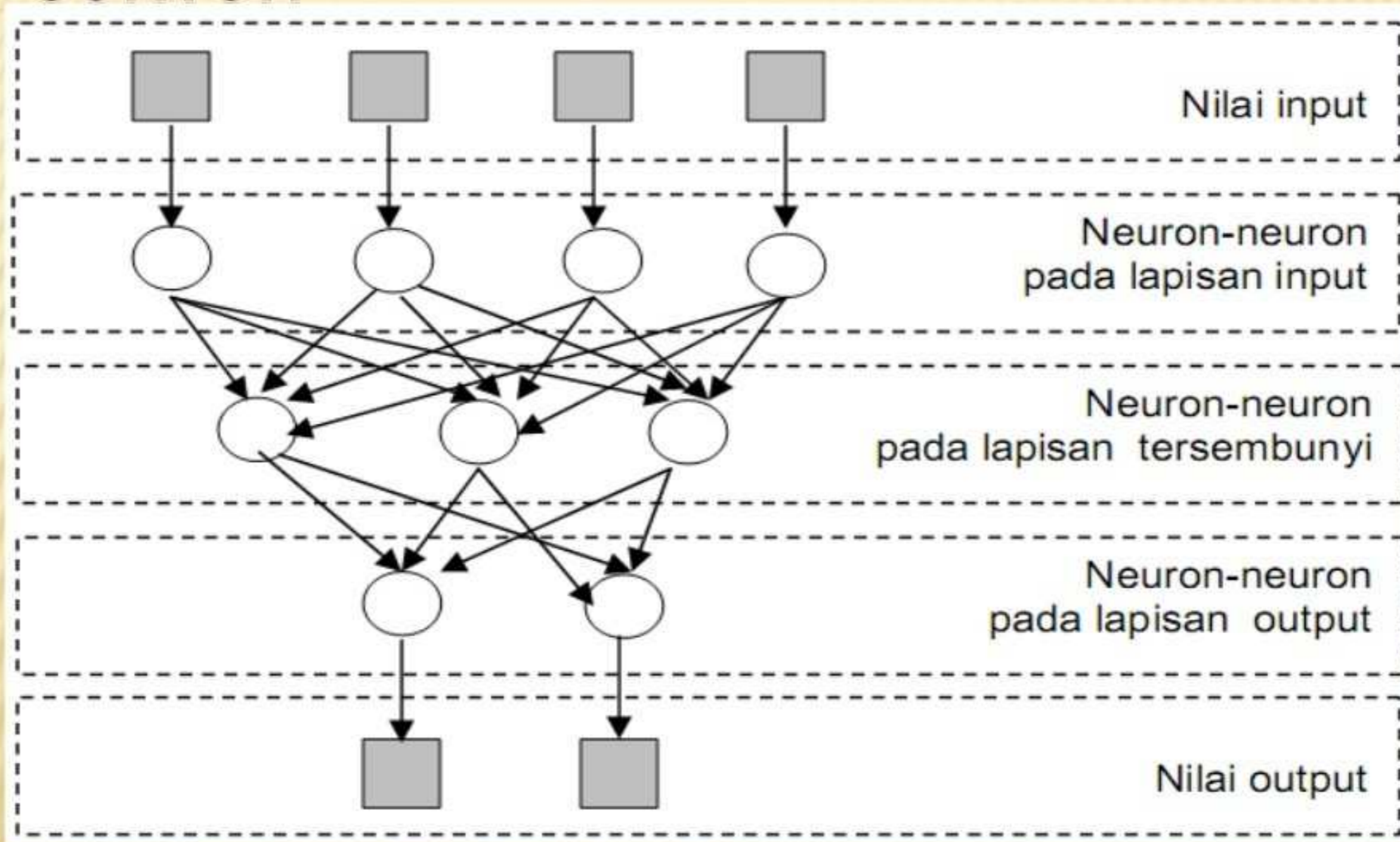
KARAKTERISTIK JST

- Ditetapkan 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)

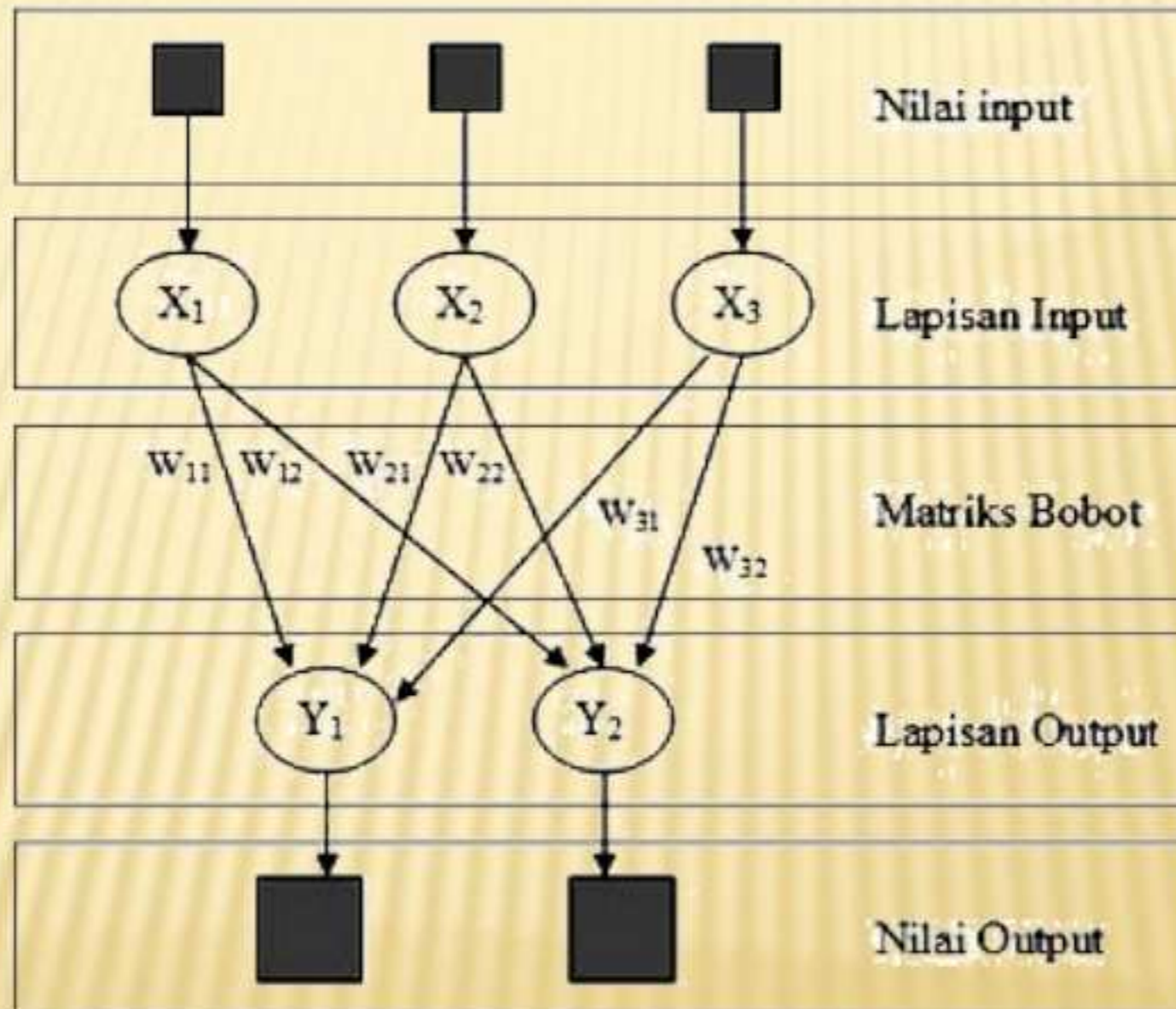
SINGLE LAYER NET

- ◆ 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.

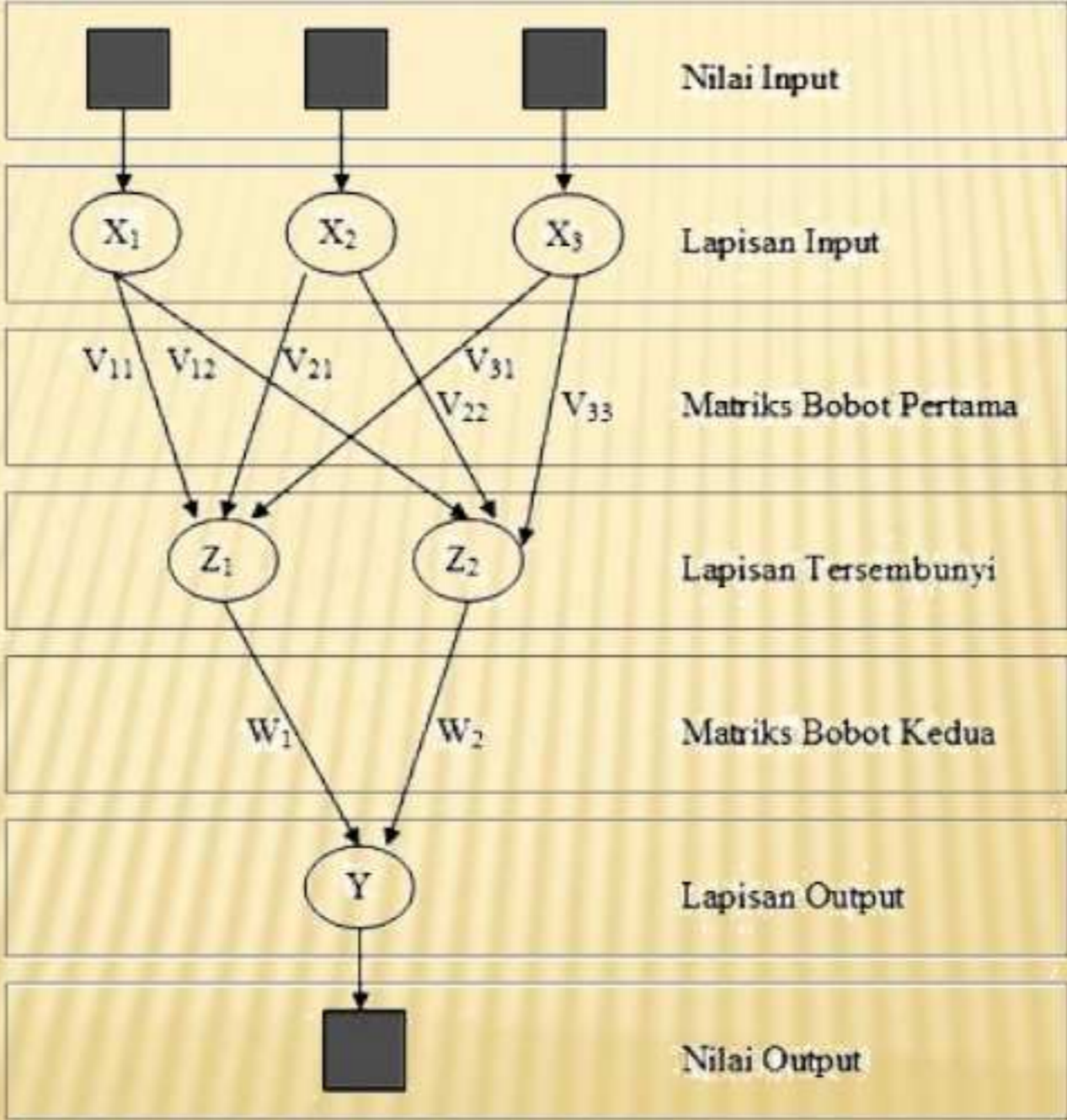
SINGLE LAYER NET



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

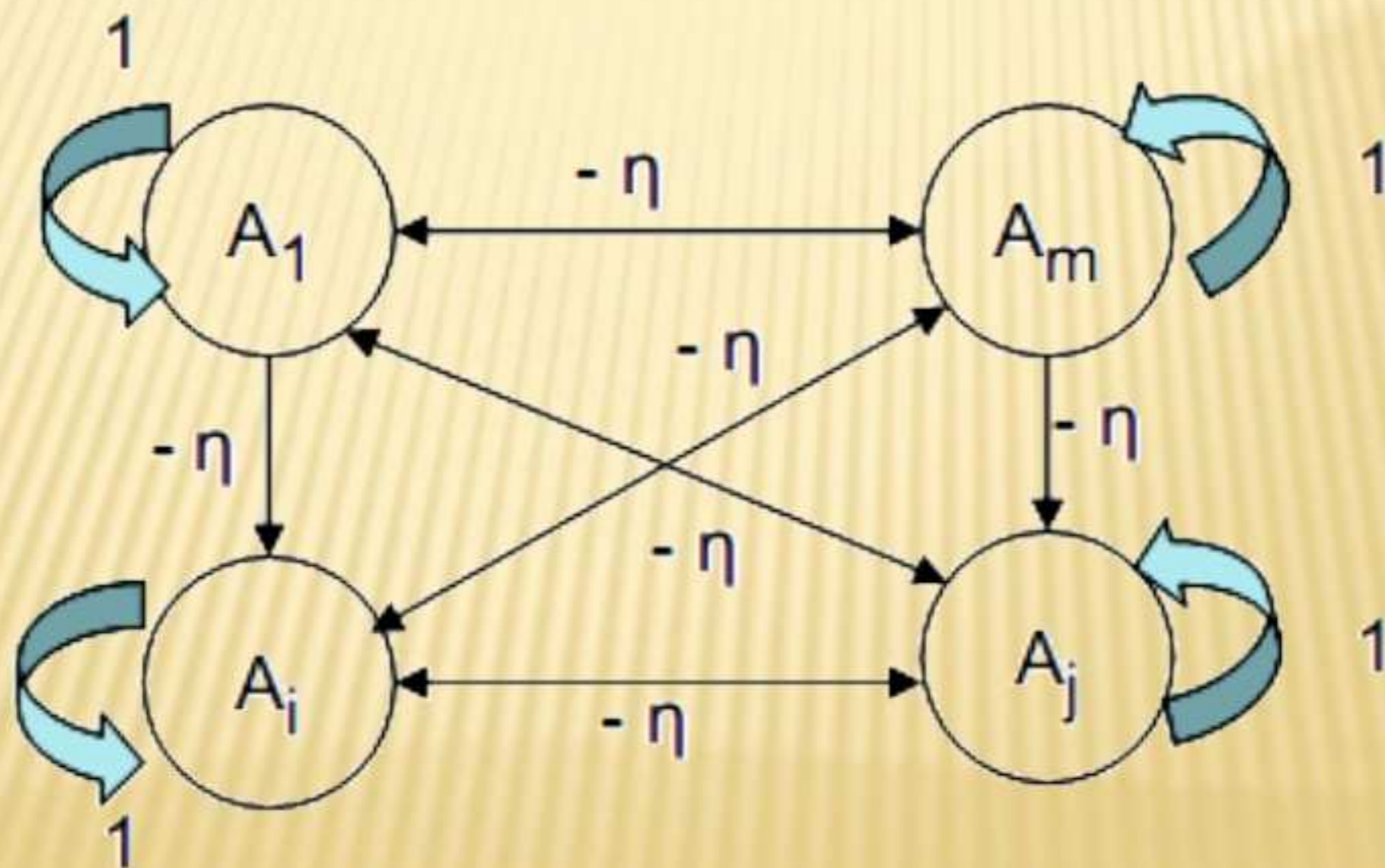
MULTILAYER NET



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

- ◆ Pembelajaran terawasi (**supervised learning**)
- ◆ Pembelajaran tak terawasi (unsupervised learning)
- ◆ Gabungan pembelajaran terawasi dan tak terawasi (**hybrid**)

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

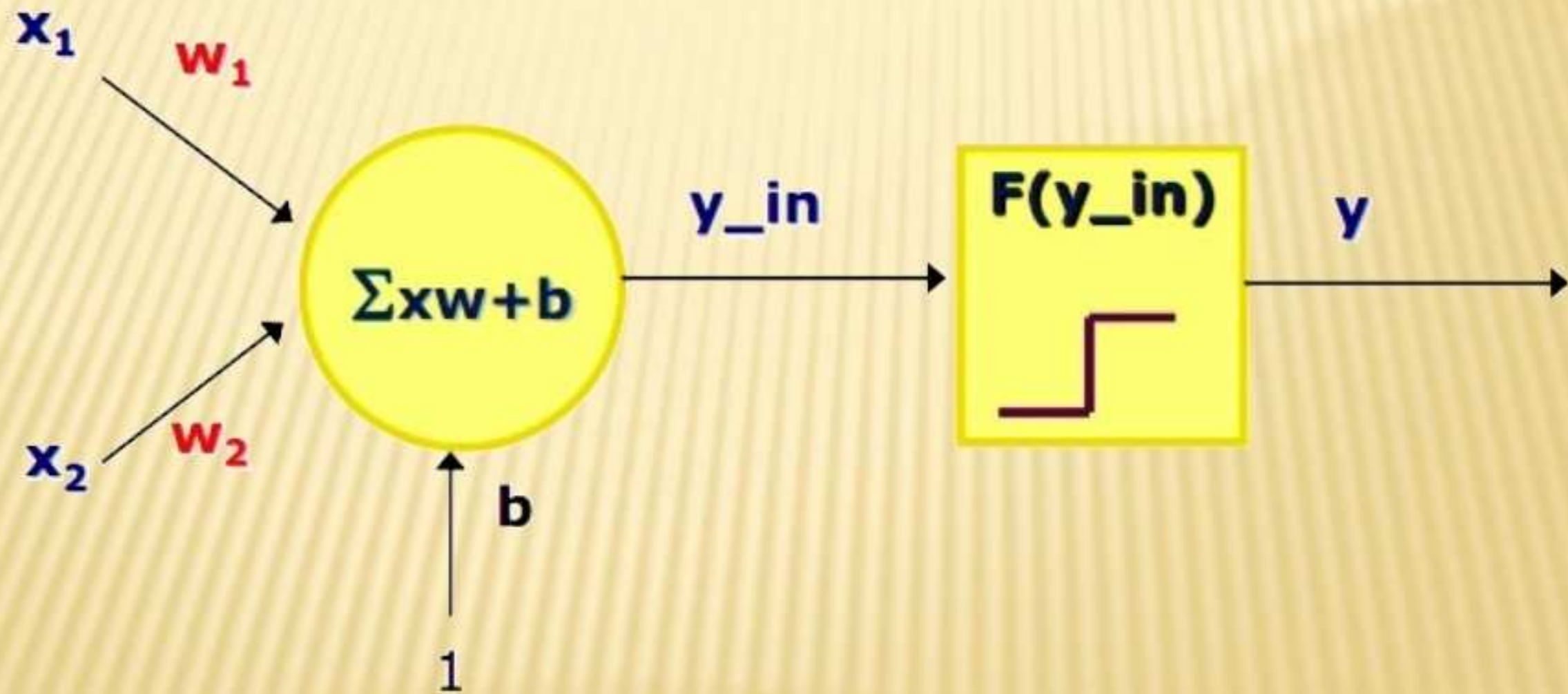
PEMBELAJARAN TAK TERAWASI

- ❑ 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

HYBRID

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

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 (baru) = W_i (lama) + ΔW
 - $\Delta W = X_i \cdot y$
 - Perbaiki bias menurut persamaan :
 - b (baru) = b (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, 0 = -1

X = 1, L = -1

Fungsi aktivasi :

y = 1, jika $y_{in} \geq 0$

y = -1, jika $y_{in} < 0$

0

0 # 0 t=1

0

0 0

0 0 t=-1

#

CONTOH KASUS

Input pertama :

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

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

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

Bias = $b.t = 1.1 = 1$

CONTOH KASUS

Input kedua :

1 -1 -1

1 -1 -1

1 1 1

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

-1 1 1

-1 1 1

-1 -1 -1

Bias = $b.t = 1.-1 = -1$

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

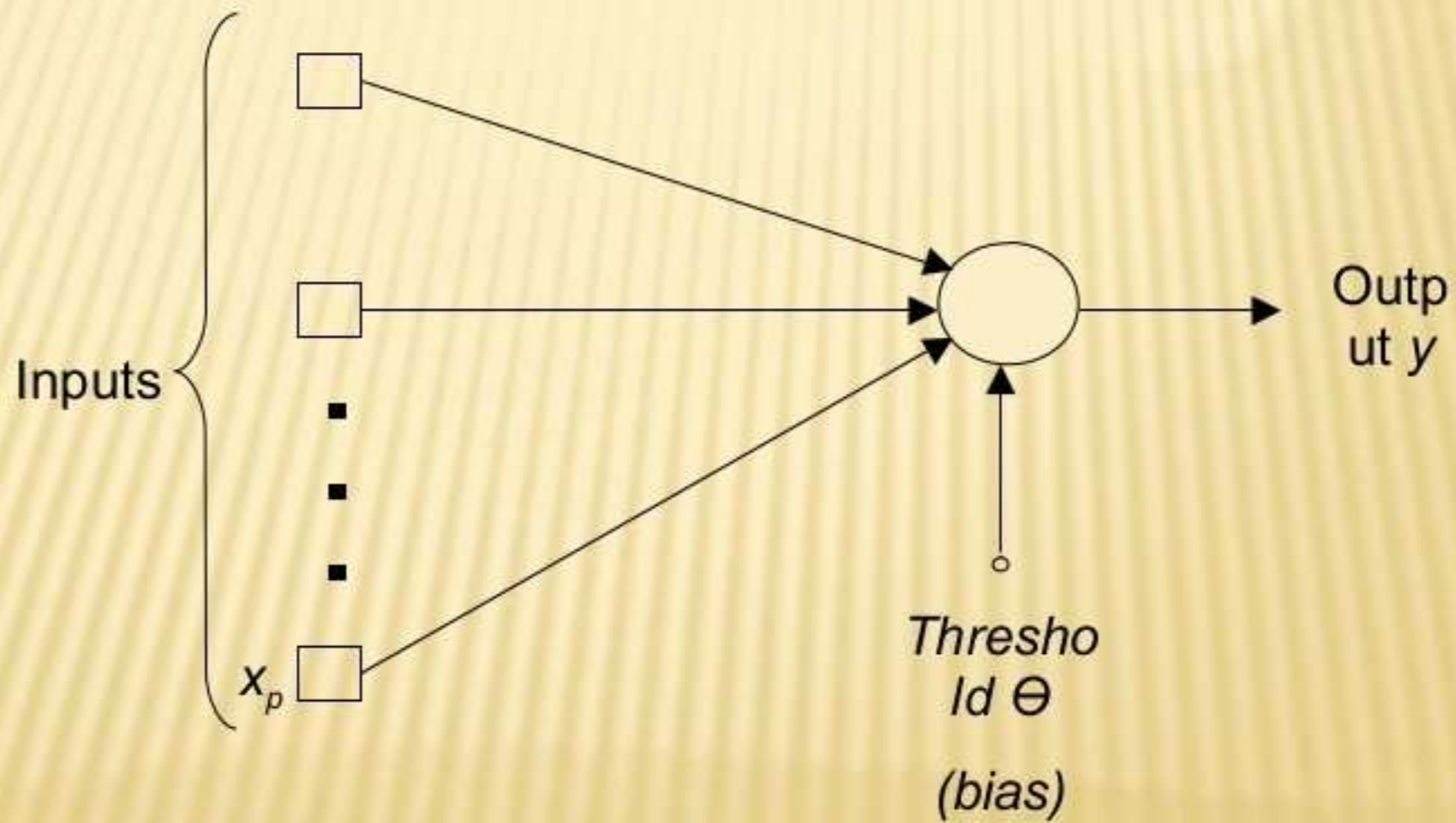
PERCEPTRON

Bentuk paling sederhana dari JST.

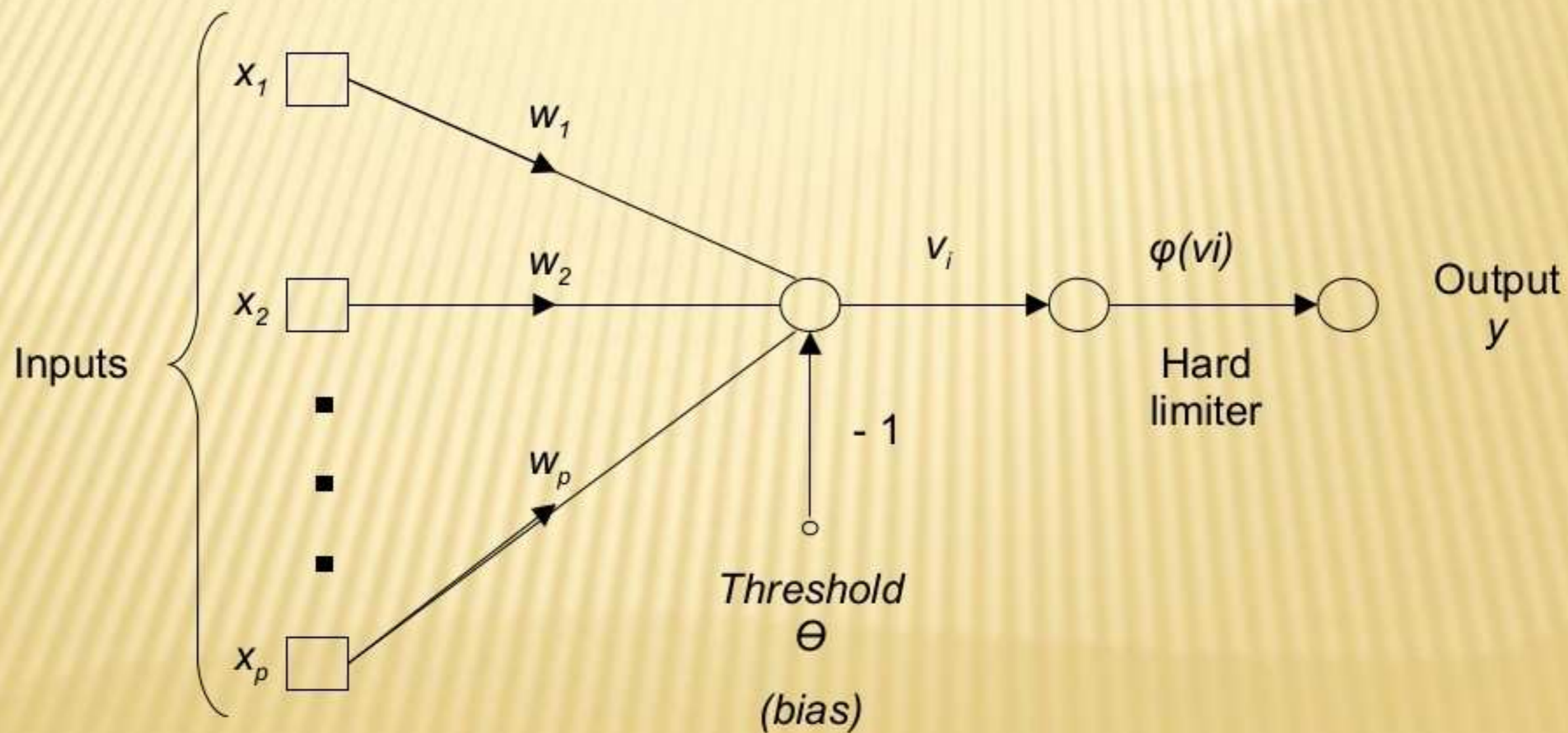
Digunakan untuk pengklasifikasian pola khusus yang biasa disebut *linearable separable*.

Terbatas hanya untuk mengklasifikasikan dua kelas saja.

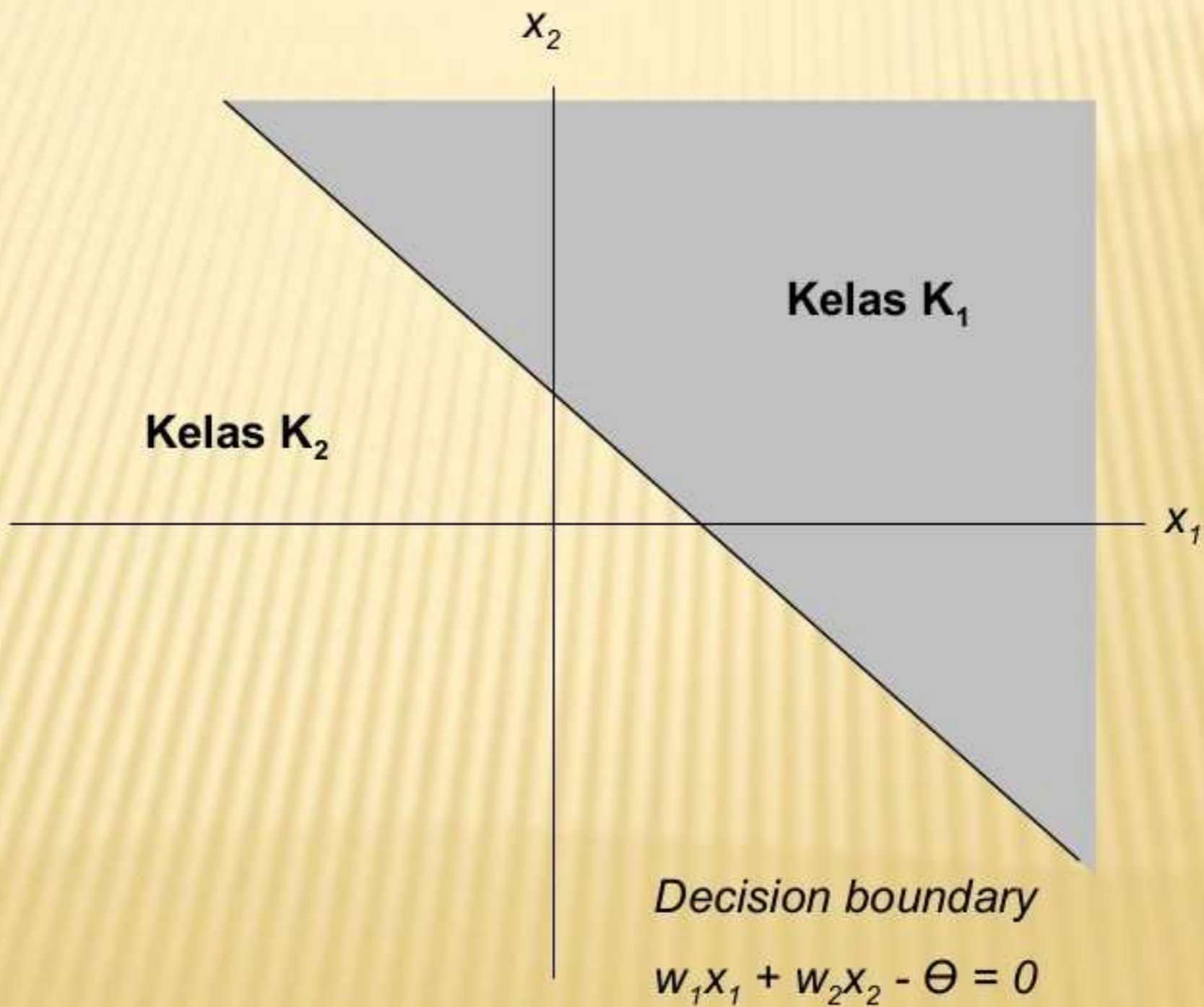
PERCEPTRON



PERCEPTRON



PERCEPTRON



PERCEPTRON

Vektor *input*:

$$x = [x_1 \quad x_2 \quad \dots \quad x_p]^T$$

Vektor bobot:

$$w = [w_1 \quad w_2 \quad \dots \quad w_p]^T$$

Vektor output:

$$v = w^T \cdot x$$

PERCEPTRON

- Aturan Pembelajaran (*Learning Rule*)
Error e (sinyal kesalahan) dinyatakan sbb:

$$e = t - y$$

dimana :

- Jika $e = 1$, maka $w^{\text{baru}} = w^{\text{lama}} + x$ (1)
- Jika $e = -1$, maka $w^{\text{baru}} = w^{\text{lama}} - x$ (2)
- Jika $e = 0$, maka $w^{\text{baru}} = w^{\text{lama}}$ (3)

$$w^{\text{baru}} = w^{\text{lama}} + e.x$$

$$b^{\text{baru}} = b^{\text{lama}} + e$$

KETERANGAN VARIABEL-VARIABEL

e = error

t = target

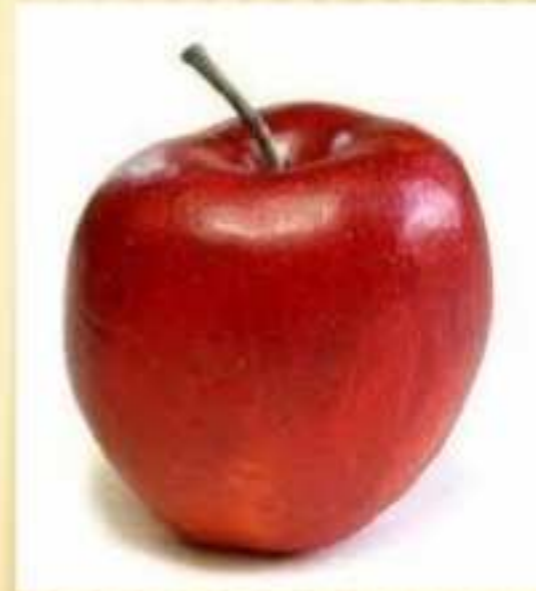
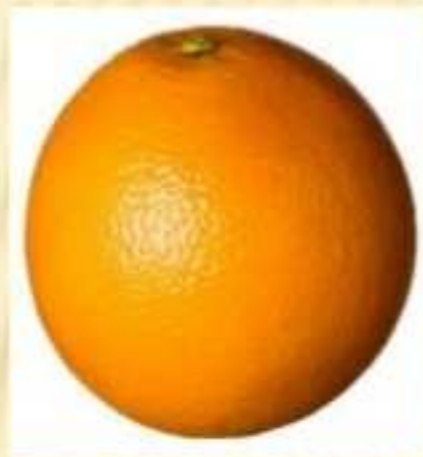
x = input

y = output

w = bobot

CONTOH KASUS

Membedakan :



CONTOH KASUS

Apel, $t=1$; jeruk, $t=0$

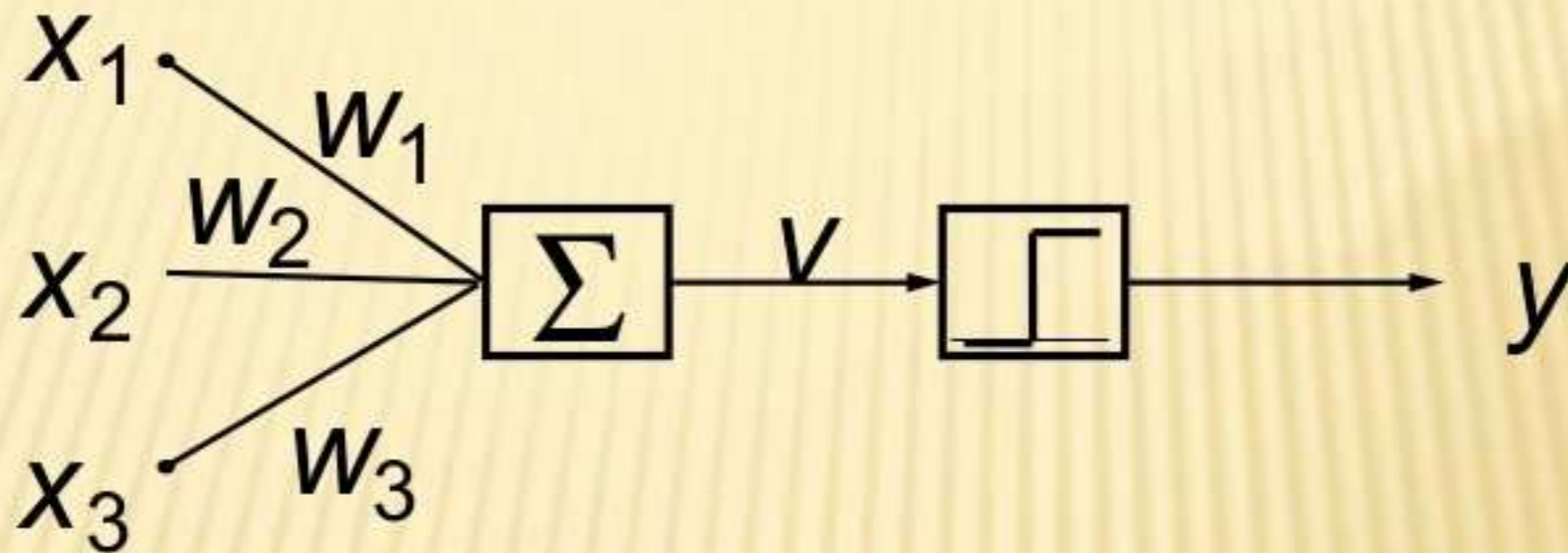
Input berasal dari sensor :

1. Sensor warna : merah=1, tidak merah=0
2. Sensor bentuk : bulat=1, tidak bulat=0
3. Sensor tekstur : kasar=1, tidak kasar=0

Misal terdapat input dan target :

$$\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, t = 1 ; \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, t = 0$$

CONTOH KASUS



No bias neuron

- Ada 3 input yaitu x_1, x_2 dan x_3
- Ada 3 bobot yaitu w_1, w_2 dan w_3
 - Tentukan bobot secara random, misal :
 $w = [0.5 \quad 0.5 \quad 0.5]$

CONTOH KASUS

1. Hitung jumlah input berbobot (v)

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

$$(1*0.5)+(1*0.5)+(0*0.5) = 1$$

CONTOH KASUS

2. Fungsi Aktivasi (menggunakan hardlimit)

$$y = 0 \text{ jika } v < 0$$

$$y = 1 \text{ jika } v \geq 0$$

$$y = \text{hardlimit}(v)$$

$$y = \text{hardlimit}(1)$$

Jadi $y = 1$, sesuai target $t = 1$ untuk input pertama

CONTOH KASUS

3. Cek bobot apakah cocok dengan input kedua [0 1 1]
dengan $t=0$, bobot [0.5 0.5 0.5]

$$v = (0.5 * 0) + (0.5 * 1) + (0.5 * 1) = 1$$

$$y = \text{hardlimit}(1)$$

$y = 1$, tidak sesuai dengan target, $t = 0$, maka harus dilakukan perubahan bobot

CONTOH KASUS

4. Perubahan bobot
dihitung dengan cara :

$$w^{\text{baru}} = w^{\text{lama}} + e.x$$

$$\text{dimana } e = t - y$$

$$\text{jadi } e = 0 - 1 = -1$$

$$\text{maka } w^{\text{baru}} = w^{\text{lama}} + (-1).x$$

$$w^{\text{baru}} = w^{\text{lama}} - x$$

$$w^{\text{baru}} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

CONTOH KASUS

5. Hitung y dengan bobot yang baru

$$v = (0.5 * 0) + (-0.5 * 1) + (-0.5 * 1) = -1$$

$$y = \text{hardlimit}(-1) = 0, \text{ sesuai dengan } t=0$$

6. Cek bobot yang baru dengan input pertama

$$v = (0.5 * 1) + (-0.5 * 1) + (-0.5 * 0) = 0$$

$$y = \text{hardlimit}(0) = 1, \text{ sesuai dengan } t=1$$

PERCEPTRON

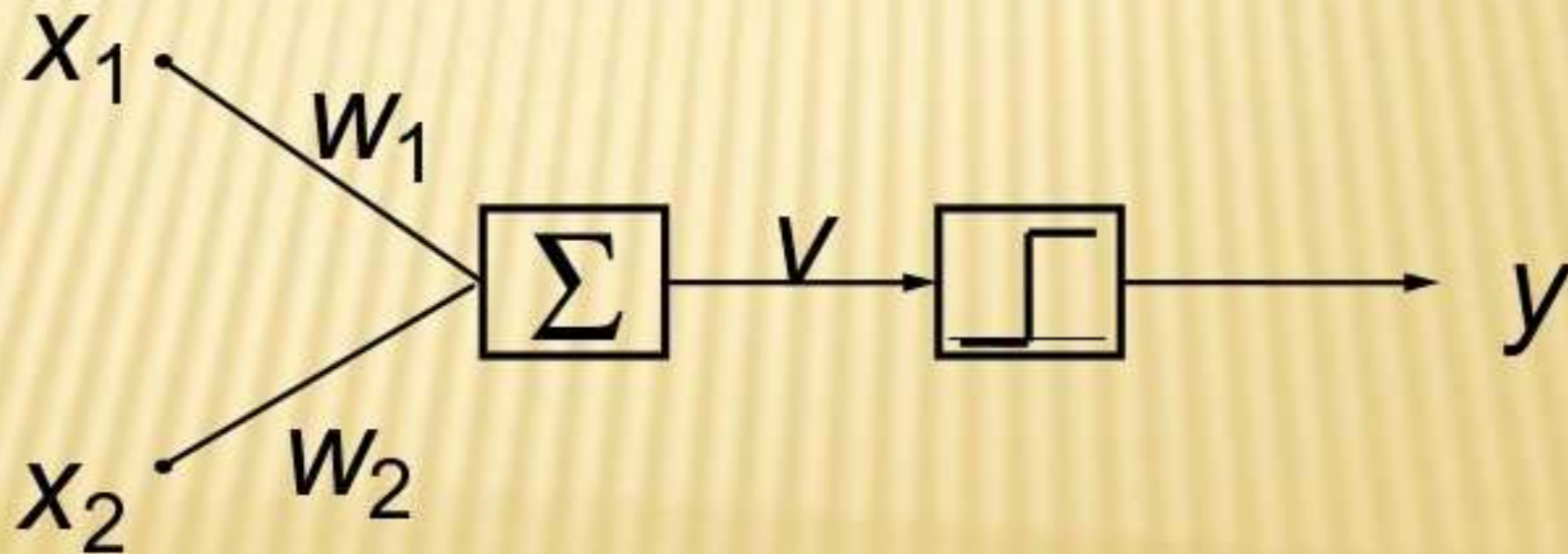
7. Jika semua input sudah diklasifikasi dengan benar, tidak perlu ada perubahan pada bobot sehingga digunakan aturan sbb:

Jika $t = y$, maka $w^{baru} = w^{lama}$

CONTOH KASUS 2

Test Problem

$$\left\{ x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, t_1 = 1 \right\} \quad \left\{ x_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, t_2 = 0 \right\} \quad \left\{ x_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, t_3 = 0 \right\}$$



No bias neuron

CONTOH KASUS 2

CONTOH KASUS 2

Kita pilih bobot secara random, misalnya:

$$w^T = [1 \quad -0.8]$$

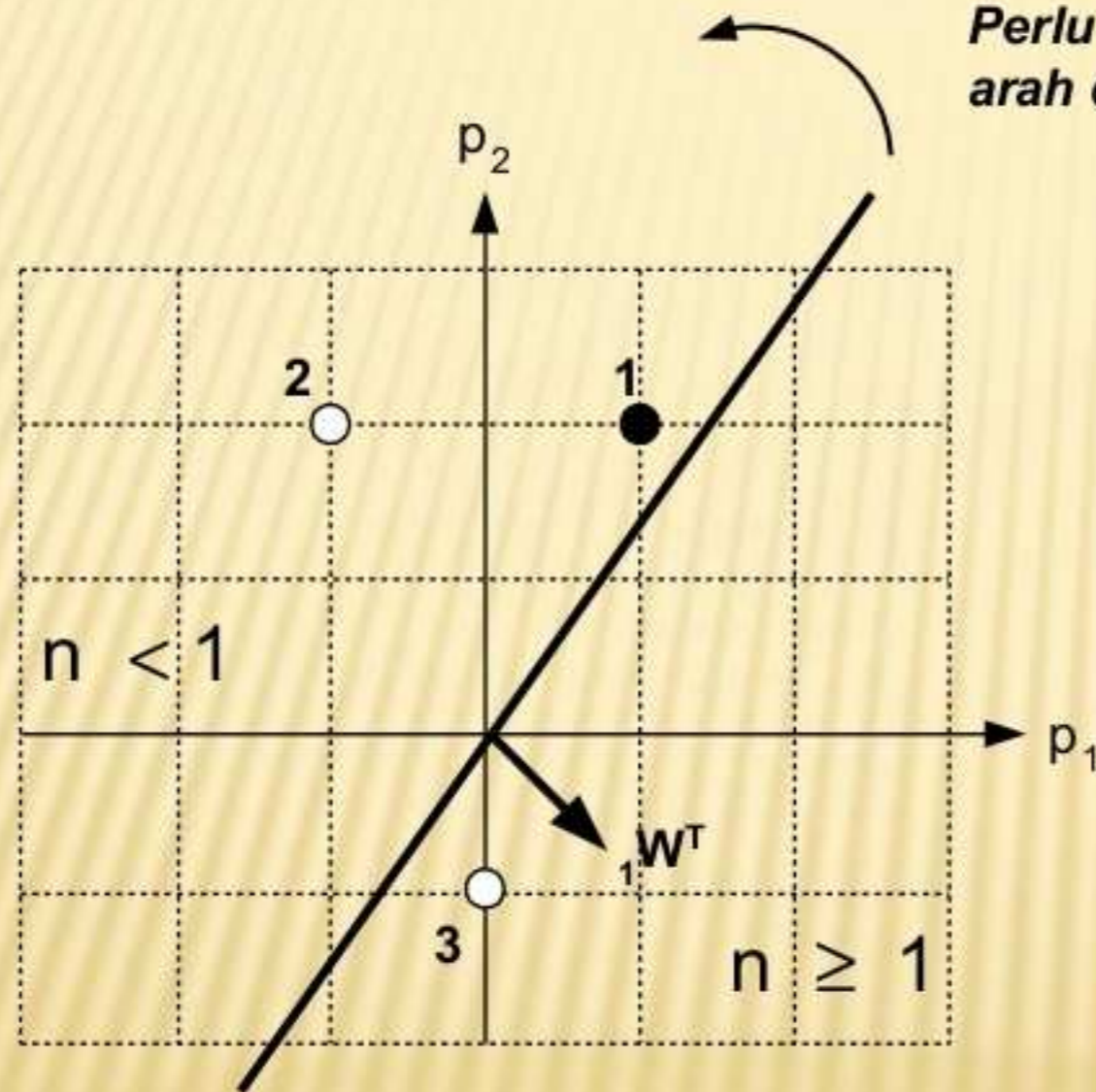
Masukkan input x_1 ke dalam jaringan dan output:

$$y = \text{hardlim} (w^T \cdot x_1)$$

$$y = \text{hardlim} \left([1 \quad -0.8] \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) = \text{hardlim} (-0.6) = 0$$

Input x_1 tidak benar karena $\mathbf{t}_1 = \mathbf{1}$

CONTOH KASUS 2



Perlu diputar berlawanan arah dengan jarum jam

Cara mencari *decision boundary*:

Jika,

$$n = \mathbf{w}\mathbf{p} + b = 0$$

$$n = w_1 \cdot x_1 + w_2 \cdot x_2 = 0$$

Maka,

$$x_1 - 0.8 x_2 = 0$$

$$5x_1 - 4x_2 = 0$$

$$\text{Jika } x_2 = 1 \rightarrow x_1 = 0.8$$

$$\text{Jika } x_1 = 1 \rightarrow x_2 = 1.25$$

CONTOH KASUS 2

Jika $t = 1$, sedangkan $y = 0$, maka:

$$w^{baru} = w^{lama} + x$$

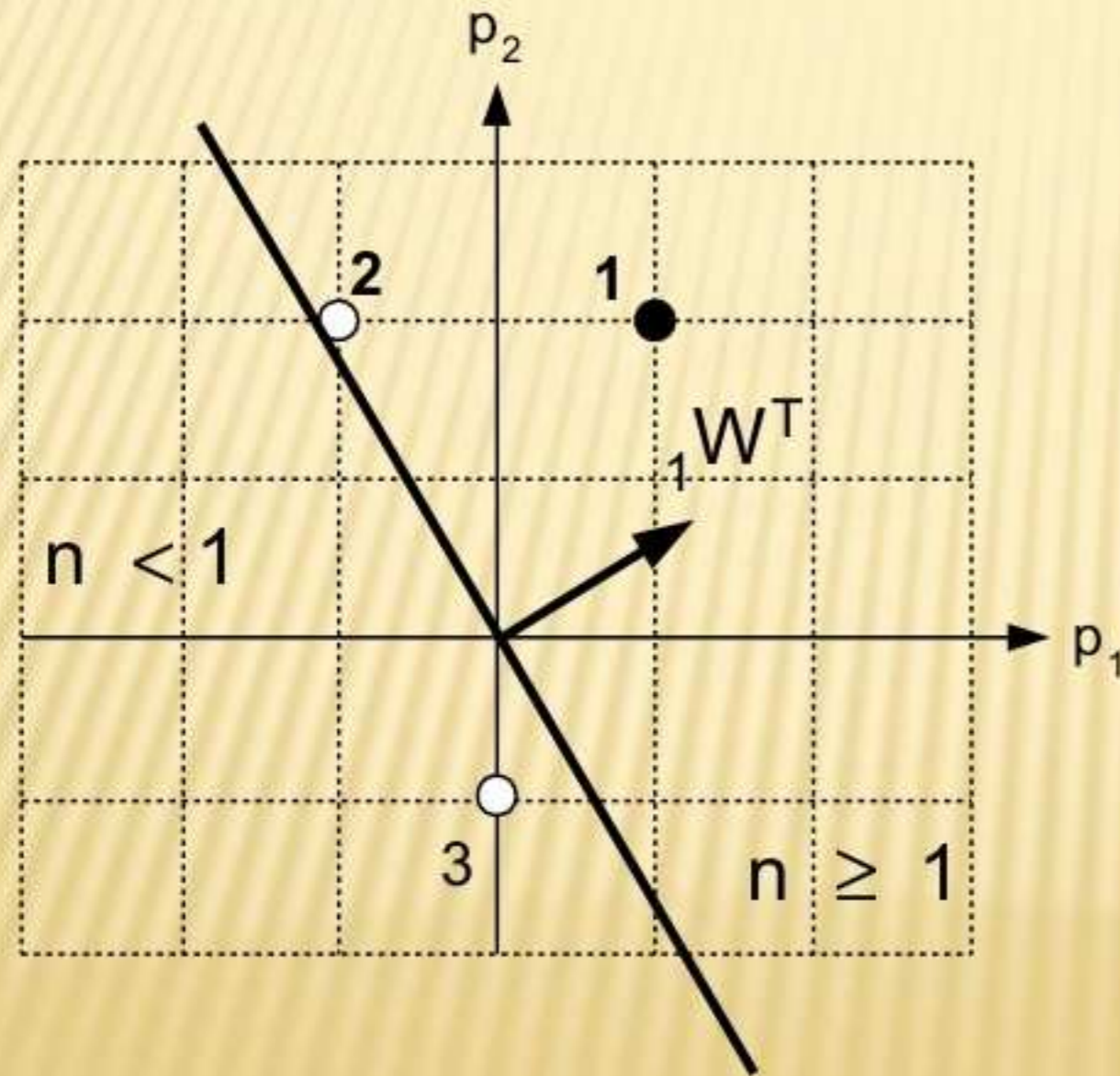
$$w^{baru} = w^{lama} + x_1 = \begin{bmatrix} 1 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1.2 \end{bmatrix}$$

Tes bobot baru ke jaringan:

$$y = \text{hardlim} \left(\begin{bmatrix} 2 & 1.2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) = \text{hardlim} (4.4) = 1$$

Hasil output y sesuai dengan target output = 1

CONTOH KASUS 2



Cara mencari *decision boundary*:

Jika,

$$n = \mathbf{w}\mathbf{p} + b = 0$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 = 0$$

Maka,

$$2x_1 + 1.2x_2 = 0$$

Jika,

$$x_2 = 1 \rightarrow x_1 = -1.2/2$$

$$x_1 = -0.6$$

CONTOH KASUS 2

Tes input kedua:

$$y = \text{hardlim} (w^T \cdot x_2)$$

$$y = \text{hardlim} \left([2 \quad 1.2] \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right) = \text{hardlim} (0.4) = 1$$

Input x_2 tidak benar karena $t_2 = 0$

CONTOH KASUS 2

Jika $t = 0$, sedangkan $y = 1$, maka:

$$w^{baru} = w^{lama} - x$$

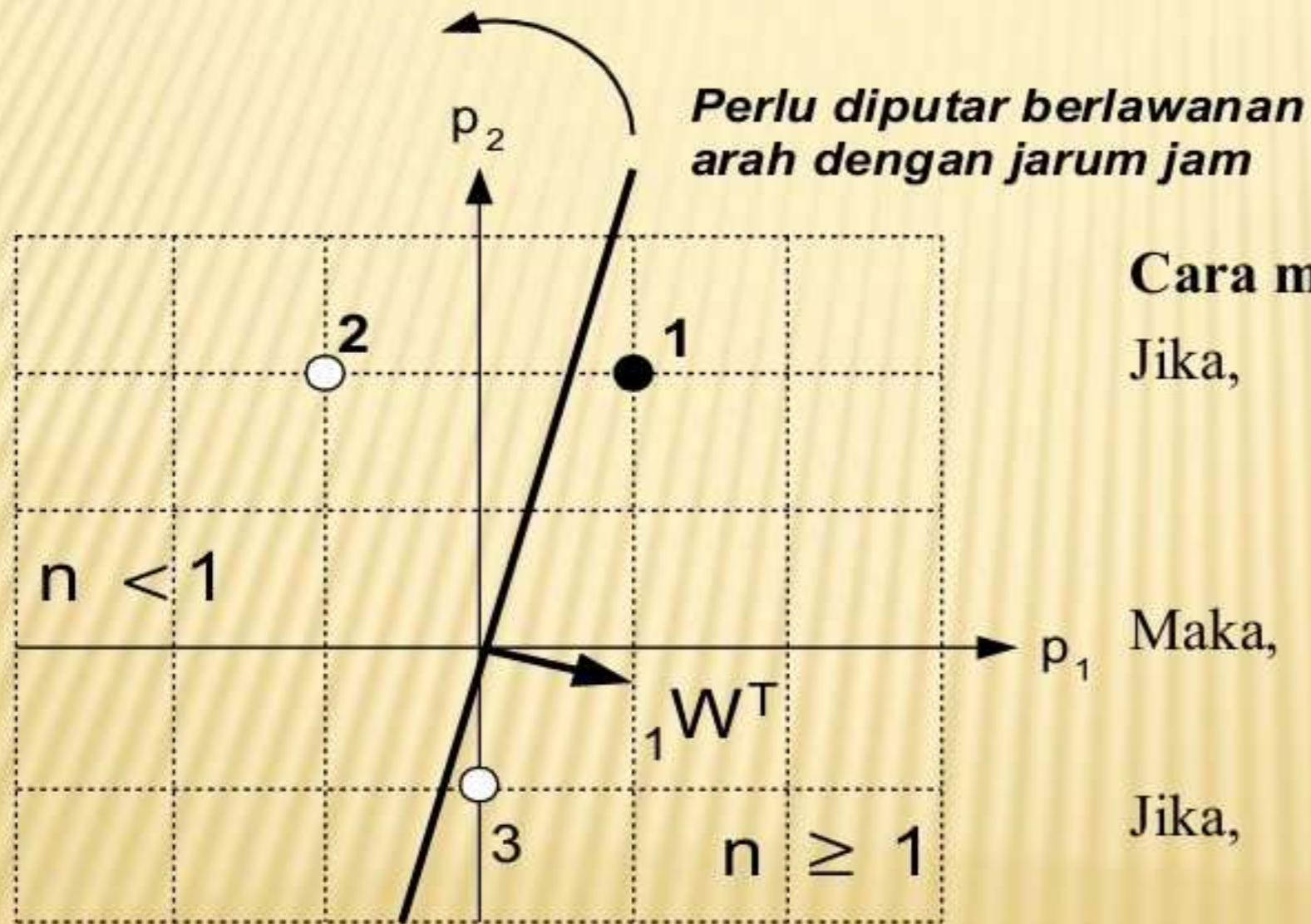
$$w^{baru} = w^{lama} - x_2 = \begin{bmatrix} 2 \\ 1.2 \end{bmatrix} - \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ -0.8 \end{bmatrix}$$

Tes bobot baru ke jaringan:

$$y = \text{hardlim} \left(\begin{bmatrix} 3 & -0.8 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right) = \text{hardlim} (-4.6) = 0$$

Hasil output y sesuai dengan target output = 0

CONTOH KASUS 2



Cara mencari *decision boundary*:

Jika,

$$n = \mathbf{w}\mathbf{p} + b = 0$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 = 0$$

Maka,

$$3x_1 - 0.8x_2 = 0$$

Jika,

$$x_2 = 2 \rightarrow x_1 = 1.6/3$$

$$x_1 = 8/15$$

CONTOH KASUS 2

Tes input ketiga:

$$y = \text{hardlim} (w^T \cdot x_3)$$

$$y = \text{hardlim} \left([3 \quad -0.8] \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right) = \text{hardlim} (0.8) = 1$$

Input x_3 tidak benar karena $t_3 = 0$

CONTOH KASUS 2

Jika $t = 0$, sedangkan $y = 1$, maka:

$$w^{baru} = w^{lama} - x$$

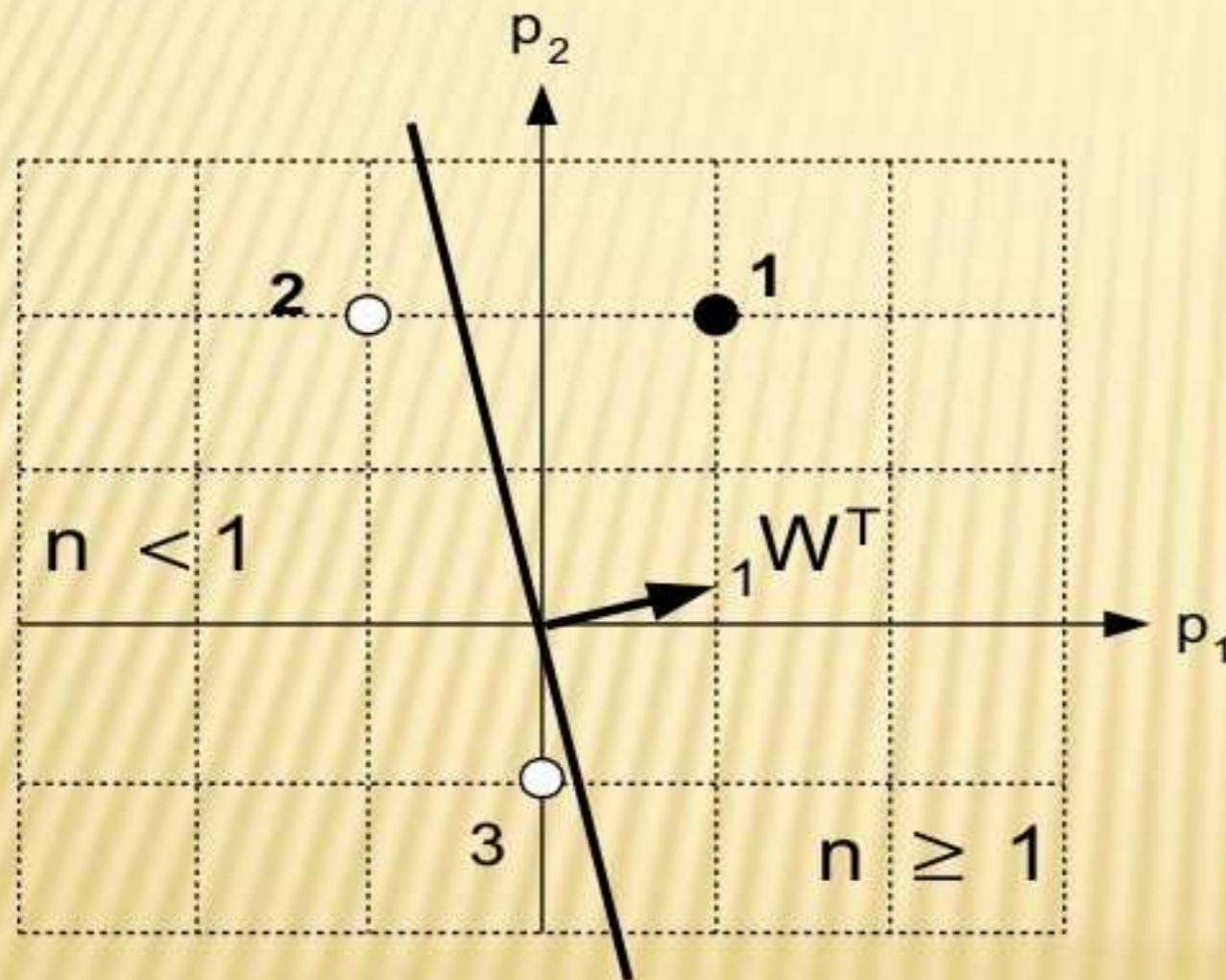
$$w^{baru} = w^{lama} - x_3 = \begin{bmatrix} 3 \\ -0.8 \end{bmatrix} - \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 0.2 \end{bmatrix}$$

Tes bobot baru ke jaringan:

$$y = \text{hardlim} \left(\begin{bmatrix} 3 & 0.2 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right) = \text{hardlim} (-0.2) = 0$$

Hasil output y sesuai dengan target output = 0

CONTOH KASUS 2



Cara mencari *decision boundary*:

Jika,

$$n = \mathbf{w}\mathbf{p} + b = 0$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 = 0$$

Maka,

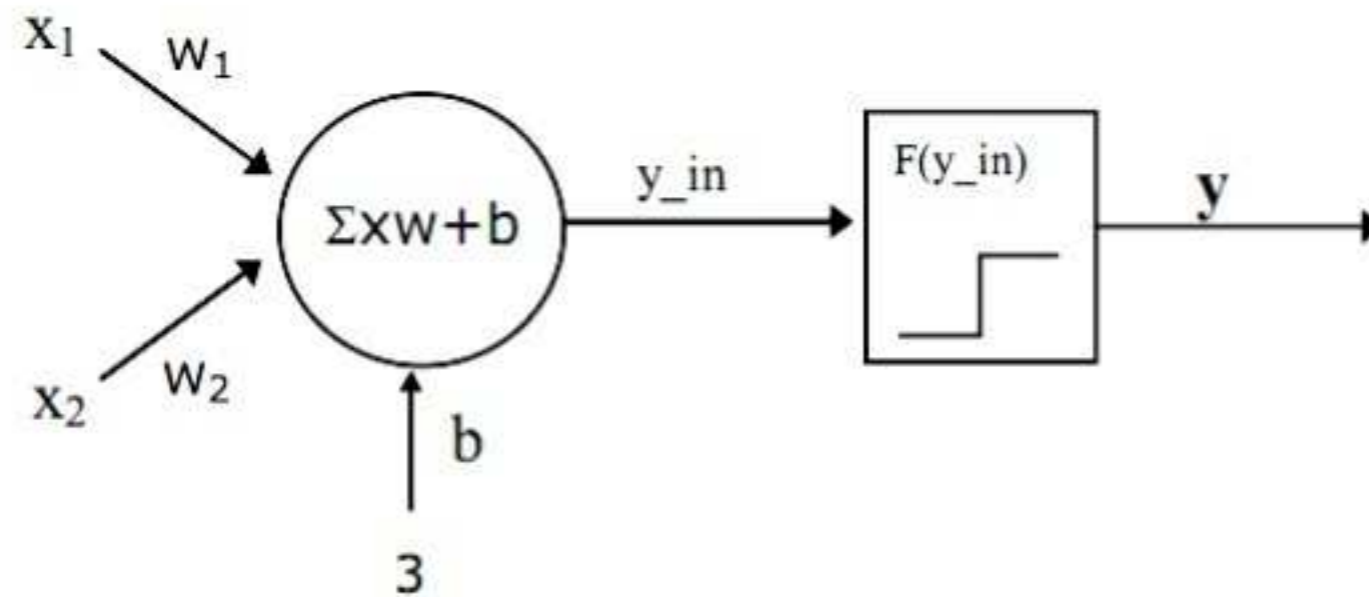
$$3x_1 + 0.2x_2 = 0$$

Jika,

$$x_2 = 1 \rightarrow x_1 = -0.2/3$$

$$x_1 = -1/15$$

CONTOH SOAL



Diketahui nilai-nilai input $x_1 = 2$, $x_2 = -2$, bobot $w_1 = -1$, $w_2 = 1$, bias $b = 3$ dan fungsi aktivasi

$$y = \begin{cases} 0, & \text{jika } x < 4 \\ 1, & \text{jika } x \geq 4 \end{cases}$$

CONTOH SOAL

Pencarian bobot sesuai perhitungan Process Elemen (PE) sederhana: penjumlahan: $\sum_i i \cdot w_i \geq T$
(penjumlahan semua perkalian input(i) & bobot(w) dibandingkan nilai ambang T)
Hitung keluaran dari Process Elemen (PE) berikut ini, jika masukan biner yang diberikan adalah: a: 0,0; b: 0,1; c: 1,0; d: 1,1

