

PENGANTAR

Laboratorium Kecerdasan Buatan setiap semester selalu melaksanakan kegiatan praktikum dan pada semester ganjil biasa dilakukan kegiatan praktikum Jaringan Saraf Tiruan dan Fuzzy Logic. Kegiatan praktikum ini dilaksanakan sebagai pendukung matakuliah Jaringan Saraf Tiruan dan Fuzzy Logic. Modul Perceptron ini merupakan salah satu modul yang digunakan untuk kegiatan praktikum Jaringan Saraf Tiruan dan Fuzzy Logic. Pada modul ini berisi materi pengenalan Matlab dan pemanfaatan Matlab untuk simulasi salah satu arsitektur jaringan saraf tiruan yaitu Perceptron. Kasus yang disajikan yaitu penerapan perceptron untuk pengenalan suatu pola. Modul ini telah ditinjau oleh dosen mata kuliah Jaringan Saraf Tiruan dan Fuzzy Logic dan kepala program studi Teknik Informatika Itenas.

Penulis : Dina Budhi Utami, S.Kom., M.T.

Reviewer : M. Ichwan, Ir., M.T.

Asep Nana Hermana, S.T., M.T.

Bandung, September 2016

DAFTAR ISI

1.	PENGENALAN M	ATLAB	3
	PRAKTIKUM 1.1	PENGGUNAAN PERINTAH MELALUI COMMAND WINDOW	4
	PRAKTIKUM 1.2	PENGGUNAAN PERINTAH MELALUI M-FILE	9
2.	PERCEPTRON		. 13
	PRAKTIKUM 2.1	PERCEPTRON UNTUK PENGENALAN FUNGSI LOGIKA	.14
	PRAKTIKUM 2.2	PERCEPTRON UNTUK PENGENALAN POLA KARAKTER	.21
	PRAKTIKUM 2.3	PERCEPTRON UNTUK PENGENALAN HURUF A, B, C	. 28

1. PENGENALAN MATLAB

Pada bab ini anda akan belajar

- 1. Penggunaan kode program melalui Command Window Matlab
- 2. Penggunaan kode program melalui M-file
- 3. Perbedaan script dan fungsi pada M-file
- 4. Penggunaan operasi aritmatik pada Matlab
- 5. Pendefinisian variabel skalar pada Matlab
- 6. Pendefinisian variable array pada Matlab
- 7. Pembuatan deret pada Matlab
- 8. Penggunaan grafik pada Matlab

PRAKTIKUM 1.1 PENGGUNAAN PERINTAH MELALUI COMMAND WINDOW

Tujuan

- : 1. Mahasiswa mampu menggunakan perintah pada Matlab melalui Command Window
- 2. Mahasiswa mampu menggunakan operasi aritmatik, variabel skalar, array, deret, dan grafik pada Matlab
- Mahasiswa mampu menggunakan Matlab sebagai perangkat Simulasi Jaringan Syaraf Tiruan

a. Dasar Teori

MATLAB adalah sebuah bahasa dengan (high-performance) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Hal ini memungkinkan kita untuk memecahkan banyak masalah teknis yang terkait dengan komputasi, kususnya yang berhubungan dengan matrix dan formulasi vektor. Penggunaan Matlab meliputi bidang-bidang:

- Matematika dan Komputasi
- Pembentukan Algorithm
- Akusisi Data
- Pemodelan, simulasi, dan pembuatan prototipe
- Analisa data, explorasi, dan visualisasi
- Grafik Keilmuan dan bidang Rekayasa

Nama MATLAB merupakan singkatan dari *matrix laboratory*. MATLAB pada awalnya ditulis untuk memudahkan akses perangkat lunak matrik yang telah dibentuk oleh

LINPACK dan EISPACK. Saat ini perangkat MATLAB telah menggabung dengan LAPACK dan BLAS library, yang merupakan satu kesatuan dari sebuah seni tersendiri dalam perangkat lunak untuk komputasi matrix.

Dalam lingkungan perguruan tinggi teknik, Matlab merupakan perangkat standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan kelimuan. Di industri, MATLAB merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tingi, pengembangan dan analisanya.

Kelengkapan pada Sistem MATLAB

Sebagai sebuah system, MATLAB tersusun dari 5 bagian utama:

- 1. Development Environment. Merupakan sekumpulan perangkat dan fasilitas yang membantu anda untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah graphical user interfaces (GUI). Termasuk didalamnya adalah MATLAB desktop dan Command Window, command history, sebuah editor dan debugger, dan browsers untuk melihat help, workspace, files, dan search path.
- **2. MATLAB Mathematical Function Library.** Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar sepertri: sum, sin, cos, dan complex arithmetic, sampai dengan fungsi-fungsi yang lebih kompek seperti matrix inverse, matrix eigenvalues, Bessel functions, dan fast Fourier transforms.
- 3. MATLAB Language. Merupakan suatu high-level matrix/array language dengan control flow statements, functions, data structures, input/output, dan fitur-fitur object-oriented programming. Ini memungkinkan bagi kita untuk melakukan kedua hal baik "pemrograman dalam lingkup sederhana " untuk mendapatkan hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memperoleh hasil-hasil dan aplikasiyang komplek.
- **4. Graphics.** MATLAB memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatu grafik. Didalamnya melibatkan high-level functions (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, image processing, animation, dan presentation graphics.
- 5. MATLAB Application Program Interface (API). Merupakan suatu library yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinterakasi dengan MATLAB.

b. Alat dan Bahan

:1. Laptop

2. Aplikasi Matlab yang telah terinstal

c. Kode Program

```
%Awal blok kode variabel skalar dan operasi aritmatik
x = 2
y = 3
z = x + y
Z
х - у
x * y
z = x / y
%Akhir blok kode variabel skalar dan operasi aritmatik
%Awal blok kode variabel array dan operasi aritmatik
x = [1 \ 2 \ 3]
y = [4 \ 5 \ 6]
y(0)
y(1)
y (2)
y (3)
y(4)
x+y
х*у
x.*y
у'
x*y'
A = [1 \ 2 \ 3]
4 5 6
7 8 9];
A*x'
B = [1 \ 2 \ 3 \ 4]
```

```
5 6 7 8
7 6 5 4];
A*B
A+B
zeros(3,6)
zeros(3,6);
ones (3,6)
ones (3, 6);
%Akhir blok kode variabel array dan operasi aritmatik
%Awal blok kode grafik deret 1
x = [1:1:50];
plot(time,x)
xlabel('time (msec)')
ylabel('x(t)')
%Awal blok kode grafik deret 1
%Awal blok kode grafik deret 2
x = [1:50];
plot(time,x)
xlabel('time (msec)')
ylabel('x(t)')
%Awal blok kode grafik deret 2
%Awal blok kode grafik deret 3
x(51:100) = [50:-1:1]
plot(time,x)
xlabel('time (msec)')
ylabel('x(t)')
%Awal blok kode grafik deret 3
```

```
%Awal blok kode grafik cos

x = cos(0.1*pi*(0:99));

time = [0:0.001:0.099];

plot(time,x)

xlabel('time (msec)')

ylabel('x(t)')

%Awal blok kode grafik cos
```

d. Langkah Kegiatan

- 1. Buka aplikasi Matlab.
- Pada Command Window, ketikan baris demi baris kode variabel skalar dan operasi aritmatik seperti pada blok kode variabel skalar dan operasi aritmatik di poin c. Setiap baris kode diakhiri dengan enter. Amati hasil yang tampil pada Command Window untuk setiap barisnya.
- 3. Pada Command Window, ketikan baris demi baris kode variabel array dan operasi aritmatik seperti pada blok kode variabel array dan operasi aritmatik di poin c. Setiap baris kode diakhiri dengan enter. Amati hasil yang tampil pada Command Window untuk setiap barisnya.
- 4. Pada Command Window, ketikan baris demi baris kode grafik deret 1 seperti pada blok kode grafik deret 1 di poin c. Setiap baris kode diakhiri dengan enter. Amati hasil yang tampil pada Command Window untuk setiap barisnya.
- 5. Pada Command Window, ketikan baris demi baris kode grafik deret 2 seperti pada blok kode grafik deret 2 di poin c. Setiap baris kode diakhiri dengan enter. Amati hasil yang tampil pada Command Window untuk setiap barisnya.
- 6. Pada Command Window, ketikan baris demi baris kode grafik deret 3 seperti pada blok kode grafik deret 3 di poin c. Setiap baris kode diakhiri dengan enter. Amati hasil yang tampil pada Command Window untuk setiap barisnya.
- 7. Pada Command Window, ketikan baris demi baris kode grafik cos seperti pada blok kode grafik cos di poin c. Setiap baris kode diakhiri dengan enter. Amati hasil yang tampil pada Command Window untuk setiap barisnya.

e. Pengujian Pemahaman:

- 1. Tunjukan kode yang mendefinisikan sebuah variable array satu dimensi! Berapa panjang array tersebut?
- 2. Perhatikan hasil yang tampil setelah memasuan kode berikut:

```
>> y(0)
```

```
>> y(1)
>> y(2)
>> y(3)
```

>> v(4)

Apa kesimpulan yang anda peroleh?

- 3. Tunjukan kode yang mendefinisikan sebuah variabel array dua dimensi atau matriks dua dimensi! Berapa ordo matriks tersebut?
- 4. Apa perbedaan operasi * dengan operasi .* pada operasi array atau matriks?
- 5. Digunakan untuk apa operasi 'seperti pada y'?
- 6. Jelaskan fungsi dari kode zeros! Digunakan untuk apa kode tersebut?
- 7. Jelaskan fungsi dari kode ones! Digunakan untuk apa kode tersebut?
- 8. Jelaskan maksud kode x = [1:1:50]!
- 9. Jelaskan maksud kode x = [1:50]!
- **10.** Jelaskan maksud kode x(51:100) = [50:-1:1]!
- 11. Jelaskan fungsi dari kode plot, xlabel, dan ylabel!

PRAKTIKUM 1.2 PENGGUNAAN PERINTAH MELALUI M-FILE

b. Tujuan

- : 1. Mahasiswa mampu menggunakan perintah pada Matlab melalui M-File
- 2. Mahasiswa mampu membuat script pada M-File.
- 3. Mahasiswa mampu membuat fungsi pada M-File.
- Mahasiswa memahami perbedaan script dan fungsi pada M-File.
- Mahasiswa mampu menggunakan Matlab sebagai perangkat Simulasi Jaringan Syaraf Tiruan

a. Dasar Teori

Fitur-fitur MATLAB sudah banyak dikembangkan, dan lebih kita kenal dengan nama *toolbox*. Sangat penting bagi seorang pengguna Matlab, toolbox mana yang mandukung untuk *learn* dan *apply* technologi yang sedang dipelajarinya. Toolbox-toolbox ini merupakan kumpulan dari fungsi-fungsi MATLAB (M-files) yang telah dikembangkan ke suatu lingkungan kerja MATLAB untuk memecahkan masalah dalam kelas particular. Area-area yang sudah bisa dipecahkan dengan toolbox saat ini meliputi pengolahan sinyal, system kontrol, neural networks, fuzzy logic, wavelets, dan lain-lain.

M-file adalah sederetan perintah matlab yang dituliskan secara berurutan sebagai sebuah file. hasil dari file yang dibuat akan memiliki ektensi m yang menandakan bahwa file yang dibuat adalah file MATLAB. Sebenarnya, MATLAB memiliki beberapa tipe file lain seperti fig, mat, mex, dan sebagainya. M-file dapat ditulis sebagai sebuah script yang sederhana atau dapat pula ditulis sebagai sebuah fungsi yang menerima argumen atau masukan dan menghasilkan output.

Berikut perbedaan antara m-file sebagai script dan m-file sebagai fungsi.

• Script

Script adalah m-file sederhana karena tidak memerlukan argument input atau output. Script sering digunakan untuk melakukan otomatisasi deret command matlab, misalnya komputasi yang perintahnya dapat diulang-ulang dari baris command. Selanjutnya, script mengoperasikan data yang ada di dalam workspace atau membuat data baru. Semua variable yang telah dibuat oleh script di dalam woarkspace dapat digunakan untuk komputasi selanjutnya setelah script selesai dijalankan. Contoh script sederhana yang digunakan untuk menampilkan grafik fungsi sinus:

```
x=30:180;
alfa=deg2rad(x);
y=sin(alfa);
plot(x,y);
```

Fungsi

Fungsi adalah m-file yang dapat menerima argument input dan menghasilkan argument output. Fungsi mengoperasikan variabel di dalam workspacenya, tetapi terpisah dari workspace yang diakses oleh command matlab. Sebuah fungsi mempunyai beberapa bagian penting di antaranya:

- Nama fungsi MATLAB secara default sama dengan nama varibelnya. Nama fungsi harus diawali dengan huruf, selanjutnya boleh diikuti oleh kombinasi huruf, angka, atau garis bawah (underscore). Sebaiknya, semua karakter menggunakan huruf kecil. Walaupun nama fungsi dapat mempunyai panjang berbeda, MATLAB hanya menggunakan N sebagai karakter pertama, di mana N adalah bilangan yang dihasilkan oleh fungsi namelengthmax().
- o Argument fungsi. Jika sebuah fungsi MATLAB memiliki nilai output lebih dari

satu, maka nilai dapat dituliskan di dalam sebuah kurung siku. Argument input, jika ada, dituliskan di dalam tanda kurung biasa. Untuk memisahkan antara argument yang ada, baik input maupun output, kita dapat menggunakan tanda koma.

Contoh fungsi akar:

```
function [x1,x2]=akar(a,b,c)
x1=(-b+sqrt(b^2-4*a*c))/2*a;
x2=(-b+sqrt(b^2-4*a*c))/2*a;
```

fungsi akar di atas memiliki tiga argument input a,b, dan c serta dua argument output x1 dan x2. Fungsi tersebut dapat dijalanan melalui command MATLAB, dengan menuliskan kode sebagai contoh:

$$>> [x1,x2]=akar(1,5,6)$$

:

- b. Alat dan Bahan
- :1. Laptop
- 2. Aplikasi Matlab yang telah terinstal
- c. Code Program

```
%Awal blok kode script 1
x = [1:1:50];
plot(time,x)
xlabel('time (msec)')
ylabel('x(t)')

%Awal blok kode script 2
x(1:52) = [0 0 1:1:50];
x(53:102) = [50:-1:1];
h = [1 2];
for n = 3:101,
y(n) = 0;
for m = 1:2,
y(n) = y(n) + h(m)*x(n-m);
end
```

```
end
plot(y)

%Awal blok kode fungsi 1
x function [x1,x2]=akar(a,b,c)
x1=(-b+sqrt(b^2-4*a*c))/2*a;
x2=(-b+sqrt(b^2-4*a*c))/2*a;
%Awal blok kode fungsi 2
function [x1,x2] = akar(a,b,c)
if nargin < 3
error('argument input tidak boleh kurang dari 3...');
end;
x1=(-b+sqrt(b^2-4*a*c))/2*a;
x2=(-b+sqrt(b^2-4*a*c))/2*a;</pre>
```

d. Langkah Kegiatan

- 1. Buka aplikasi Matlab.
- 2. Atur direktori tempat bekerja yang berfungsi mengatur tempat penyimpanan M-File dengan cara mengatur **Current Directory** pada toolbar. Kemudian pilih tanda panah ke bawah dan pilih **new directory**.
- 3. Membuat M-File baru dengan cara pilih menu File -> New -> M-file.
- 4. Pada Matlab Editor, ketikan baris demi baris kode script 1 seperti pada blok kode script 1 di poin c.
- 5. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.** Simpan M-File dengan nama file yang anda kehendaki (sebagai contoh "1.2 script 1"). Amati hasil yang tampil.
- 6. Membuat M-File baru dengan cara pilih menu File -> New -> M-file.
- 7. Pada Matlab Editor, ketikan baris demi baris kode script 2 seperti pada blok kode script 2 di poin c.
- 8. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.** Simpan M-File dengan nama file yang anda kehendaki (sebagai contoh "1.2 script 2"). Amati hasil yang tampil.
- 9. Membuat M-File baru dengan cara pilih menu File -> New -> M-file.

- 10. Pada Matlab Editor, ketikan baris demi baris kode fungsi 1 seperti pada blok kode fungsi 1 di poin c.
- 11. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.** Simpan M-File dengan nama yang sama dengan nama fungsi yaitu "akar".
- 12. Pada Command Window, ketikan [x1, x2] = akar(1, 5, 6) kemudian akhiri dengan enter. Amati hasil yang tampil pada Command Window.
- 13. Pada Command Window, ketikan [x1,x2] = akar(1,5) kemudian akhiri dengan enter. Amati hasil yang tampil pada Command Window.
- 14. Ubah fungsi akar pada Matlab Editor menjadi baris kode fungsi 2 seperti pada blok kode fungsi 2 di poin c.
- 15. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.**
- 16. Pada Command Window, ketikan [x1, x2] = akar(1, 5, 6) kemudian akhiri dengan enter. Amati hasil yang tampil pada Command Window.
- 17. Pada Command Window, ketikan [x1, x2] = akar(1, 5) kemudian akhiri dengan enter. Amati hasil yang tampil pada Command Window.

f. Pengujian Pemahaman:

- 1. Berdasarkan hasil percobaan, jelaskan perbedaan script dan fungsi!
- 2. Jelaskan maksud dari kode script 2!
- 3. Buatlah script pada M-File yang menampilkan grafik dari deret x berikut! $x = 0, 100, 200, 300, 400, \dots, 2200, 2300$
- 4. Buatlah script pada M-File yang menampilkan grafik dari deret x berikut! $y = -10, -9.5, -9, -8.5, \dots -0.5, 0, 0.5, \dots, 9, 9.5, 10$
- 5. Buatlah script pada M-File yang menampilkan grafik dari deret x berikut! z = 10, 9.95, 9.9, 9.85, 9.8, 9.75, ..., 1, 0.95, 0.9, ..., 0.05, 0
- 6. Jelaskan perbedaan dari program fungsi 1 dan fungsi 2!
- 7. Jelaskan fungsi dari kode nargin! Digunakan untuk apa kode tersebut?
- 8. Buatlah fungsi untuk menghitung keliling dan luas lingkaran!
- 9. Buatlah fungsi untuk menentukan bilangan Fibonacci ke n (Fn)! dimana bilangan Fibonacci adalah <u>barisan</u> yang didefinisikan secara <u>rekursif</u> sebagai berikut:

$$F(n) = \left\{ egin{array}{ll} 0, & ext{jika } n=0; \ 1, & ext{jika } n=1; \ F(n-1) + F(n-2) & ext{jika tidak}. \end{array}
ight.$$

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765,... Sebagai contoh Bilangan Fibonacci ke 5 (F6) = 8

2. PERCEPTRON

Pada bab ini anda akan belajar

- 1. Arsitektur perceptron
- 2. Pelatihan jaringan saraf tiruan perceptron
- 3. Single perceptron
- 4. Multi perceptron
- 5. Proses pengenalan pola fungsi logika dengan jaringan saraf tiruan
- 6. Proses pengenalan pola karakter dengan jaringan saraf tiruan

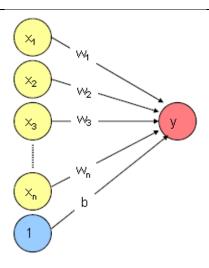
PRAKTIKUM 2.1 PERCEPTRON UNTUK PENGENALAN FUNGSI LOGIKA

a. Tujuan

- : 1. Mahasiswa memahami arsitektur perceptron
- 2. Mahasiswa mampu merancang arsitektur jaringan saraf tiruan untuk pengenalan fungsi logika
- 3. Mahasiswa memahami pengaruh representasi masukan dan keluaran terhadap jaringan saraf tiruan
- 4. Mahasiswa memahami pengaruh leraning rate terhadap proses pelatihan jaringan saraf tiruan
- 5. Mahasiswa memahami pengaruh goal terhadap proses pelatihan jaringan saraf tiruan
- 6. Mahasiswa memahami proses pelatihan jaringan saraf tiruan

b. Dasar Teori

Perceptron adalah salah satu arsitektur jaringan saraf tiruan. Jringan Perceptron menyeruapai arsitektur jaringan Hebb. Jaringan ini diperkenalan oleh Rosenblatt (1962) dan Minsky Papert (1969). Perceptron disebut juga jaringan satu layer (*Single Layer Perceptron*) merupakan jaringan dengan seluruh neuron masukan terhubung langsung ke neuron keluaran dan ditambah dengan satu buah bias. Seperti pada Gambar 1, terdapat dua buah lapisan, yaitu lapisan masukan (x) dan lapisan keluaran (y). Jumlah neuron pada setiap lapisan tidak terbatas. Seluruh neuron pada lapisan masukan terhubung dengan semua neuron pada lapisan keluaran dan memiliki bobot (w). Sebagai contoh, nilai bobot antara neuron satu pada lapisan masukan (x_I) dan neuron dua pada lapisan keluaran (y) dilambangkan dengan w_I . Selain itu setiap neuron pada lapisan keluaran terhubung dengan terdapat bias (b). Nilai bias untuk neuron keluaran satu (y) adalah b.



Gambar 1 Arsitektur single perceptron

Jaringan saraf tiruan dapat mempelajari masukan-masukan untuk membuat pola. Proses mempelajari masukan-masukan disebut dengan proses pelatihan. Pada proses pelatihan ini, belajar diartikan sebagai menentukan nilai bobot dan bias yang sesuai agar jaringan saraf tiruan dapat mengenali pola dari masukan yang diberikan. Performa jaringan saraf tiruan sangat tergantung pada masukan yang diberikan saat pelatihan. Semakin tinggi variasi masukan maka jaringan saraf tiruan dapat meningkatkan kehandalan jaringan dalam mengenali pola. Algoritma pelatihan jaringan saraf tiruan adalah sebagai berikut:

1. Pertama dilakukan inisialisasi semua bobot (w), bias (b), dan *learning rate* (α) .

$$w_i = 0 \text{ (i=1, ..., n)}$$

 $b = 0$
 $0 \le \text{ learning rate } \le 1$

- 2. Selama ada elemen masukan (s) yang keluarannya tidak sama dengan target (t), lakukan:
 - Set aktifitas unit masukan $x_i = s_i$ untuk i=1, ..., m (m adalah jumlah masukan).
 - Hitung respon unit keluaran *y*

$$y = f(x_1w_1 + \dots + x_mw_m + b)$$

• Bila $y \neq t$, perbaiki bobot dan bias

$$w_i(baru) = w_i(lama) + \Delta w$$
 $i = 1, ..., m$

Dimana $\Delta w = \alpha t x_i$

$$b(baru) = b(lama) + \alpha.t$$

Pada percobaan ini akan dibuat perceptron yang dapat mengenali pola fungsi logika AND dengan representasi masukan dan keluaran bipolar (-1 atau 1). Tabel masukan dan target fungsi logika AND dengan representasi bipolar adalah sebagai berikut:

Ması	ukan	Keluaran
x_1	x_2	у
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Tahapan penyelesaian masalah dengan menggunakan jaringan saraf tiruan adalah sebagai berikut:

Tahap I: Identikasi masalah

Jaringan memiliki dua buah masukan yaitu x_1 dan x_2 , serta memiliki satu keluaran y.

Tahap II: Training data set

Training data set diperlihatkan pada tabel. Terdapat 4 pola masukan dan keluaran yang akan digunakan untuk melatih jaringan. Keempat pola tersebut dimasukkan ke dalam matriks dengan perintah:

$$p = [11-1-1;1-11-1]$$

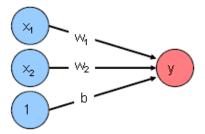
 $t = [1-1-1-1]$

Tahap III: Pemilihan metoda

Pada percobaan ini digunakan jenis jaringan saraf tiruan yaitu perceptron

Tahap IV: Inisialisasi dan pembentukan jaringan

• Arsiterktur jaringan untuk pengenalan fungsi logika AND adalah sebagai berikut:



Karena jaringan memiliki 2 masukan, berarti jaringan terdiri dari 2 neuron pada

layer masukan, 1 neuron pada layer keluaran, dan 1 bias. Pembentukan jaringan perceptron adalah dengan perintah berikut:

```
net=newp(p,t);
```

dengan:

- net : jaringan
- newp : perintah untuk membuat jaringan perceptron
- p : vektor masukan jaringan
- t : vector keluaran jaringan
- Fungsi aktivasi yang digunakan adalah 'hardlims' yaitu

$$y = f(net) \begin{cases} 1, net > 0 \\ 0, net = 0 \\ -1, net < 0 \end{cases}$$

Pengaturan fungsi aktivasi pada jaringan perceptron adalah dengan perintah berikut:

```
net.layers{1}.transferFcn = 'hardlims';
```

• Inisialisasi bobot dan bias awal $w_i = 0$, b = 0 dengan perintah berikut:

```
net.IW{1,1}= [0 0];
net.b{1} = [0];
```

Tahap V: Simulasi jaringan

Sebelum melatih, dilakukan pemeriksaan nilai inisialisasi bobot dan bias awal. Perintah yang digunakan adalah

 $\verb"net.IW{1,1}" : menampilkan bobot dari layer masukan ke kayer keluaran$

net.b{1} : menampilkan bias layer keluaran

Selanjutnya dilakukan simulasi untuk mengetahui keluaran jaringan menggunakan bobot awal ini. Perintahnya:

```
y = sim(net, p);
```

dengan:

- y : keluaran jaringan
- sim : perintah untuk melakukan simulasi
- net : nama jaringan
- p : vektor masukan jaringan

Jika hasil simulasi menunjukan bahwa keluaran jaringan menggunaan bobot dan bias awal belum sesuai dengan target keluaran yang diinginkan, maka jaringan harus dilatih terlebih

dahulu.

Tahap VI: Pelatihan jaringan

Sebelum dilakukan pelatihan, ada beberapa parameter yang harus diatur terlebih dahulu, yaitu:

- Pengaturan ambang batas selisih antara target dengan keluaran. Goal = 1
 net.trainParam.goal=1
- Pengaturan ambang batas iterasi atau epoch=1000

```
net.trainParam.epochs=10000
```

• Learning rate $\alpha = 1$

```
net.trainParam.lr=1
```

Perintah untuk melakukan pelatihan adalah sebagai berikut:

```
net = train(net,p,t);
```

dengan:

- train: perintah untuk melakukan pelatihan
- net : nama jaringan
- p : vektor masukan jaringan
- t : target keluaran jaringan yang diinginkan

Tahap VII: Penggunaan jaringan untuk pengenalan pola

Perintah untuk pengenalan pola adalah printah simulasi jaringan, yaitu

```
y = sim(net, p);
```

- c. Alat dan Bahan : 1. Laptop
 - 2. Aplikasi Matlab yang telah terinstal
- d. Code Program :

```
%pola masukan
p = [ 1 1 -1 -1 ; 1 -1 1 -1]
%pola keluaran
t = [ 1 -1 -1 -1 ]
%pembentukan jaringan perceptron
```

```
net=newp(p,t);
%pemilihan fungsi aktivasi
net.layers{1}.transferFcn = 'hardlims';
%inisialisasi bobot dan bias awal
net.IW{1,1}=[0\ 0];
net.b{1} = [0];
net.IW\{1,1\}
net.b{1}
%simulasi masukan p pada jaringan
y = sim(net,p);
%pengaturan ambang batas selisih antara target dan keluaran
net.trainParam.goal=1
%pengaturan ambang batas iterasi atau epoch
net.trainParam.epochs=10000
%pengaturan nilai learning rate
net.trainParam.lr=1
%pelatihan jaringan
net = train(net, p, t);
net.IW\{1,1\}
net.b{1}
%Pengenalan pola masukan p pada jaringan
```

```
y = sim(net,p);
y
```

e. Langkah Kegiatan

- 1. Buka aplikasi Matlab.
- 2. Atur direktori tempat bekerja yang berfungsi mengatur tempat penyimpanan M-File dengan cara mengatur **Current Directory** pada toolbar. Kemudian pilih tanda panah ke bawah dan pilih **new directory**.
- 3. Membuat M-File baru dengan cara pilih menu File -> New -> M-file.
- 4. Pada Matlab Editor, ketikan baris demi baris kode di poin c.
- 5. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.** Simpan M-File dengan nama file yang anda kehendaki (sebagai contoh "2.1 logika AND"). Amati hasil yang tampil.
- 6. Ubah nilai berikut pada Matlab Editor.
 - Learning rate = 0.5 dan goal = 1
- 7. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.**
- 8. Ubah nilai berikut pada Matlab Editor.
 - Learning rate = 1 dan goal = 0.5
- 9. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.**
- 10. Ubah nilai berikut pada Matlab Editor.
 - Learning rate = 1 dan goal = 0
- 11. Ubah nilai berikut pada Matlab Editor.
 - Learning rate = 0.5 dan goal = 0.5
- 12. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.**
- 13. Ubah nilai berikut pada Matlab Editor.
 - Learning rate = 0.5 dan goal = 0
- 14. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.**

f. Pengujian Pemahaman:

- 1. Berdasarkan hasil percobaan, bagaimana pengaruh perubahan nilai learning rate terhadap performa pengenalan pola?
- 2. Berdasarkan hasil percobaan, bagaimana pengaruh perubahan nilai goal terhadap performa pengenalan pola?
- 3. Apa perbedaan fungsi aktivasi 'hardlims' dan 'hardlim'?

- 4. Apa yang dimaksud dengan epoch?
- 5. Buatlah program pengenalan pola fungsi logika AND dengan representasi masukan dan keluaran biner (0 atau 1). Fungsi aktivasi yang digunakan adalah fungsi biner yaitu 'hardlim'. Tabel masukan dan target fungsi logika AND dengan representasi biner adalah sebagai berikut:

Ması	ukan	Keluaran
x_1	x_2	У
1	1	1
1	0	0
0	1	0
0	0	0

6. Buatlah program pengenalan pola fungsi logika AND dengan representasi masukan biner (0 atau 1) dan keluaran bipolar (-1 atau 1). Fungsi aktivasi yang digunakan] yaitu 'hardlims'. Tabel masukan dan target fungsi logika AND adalah sebagai berikut:

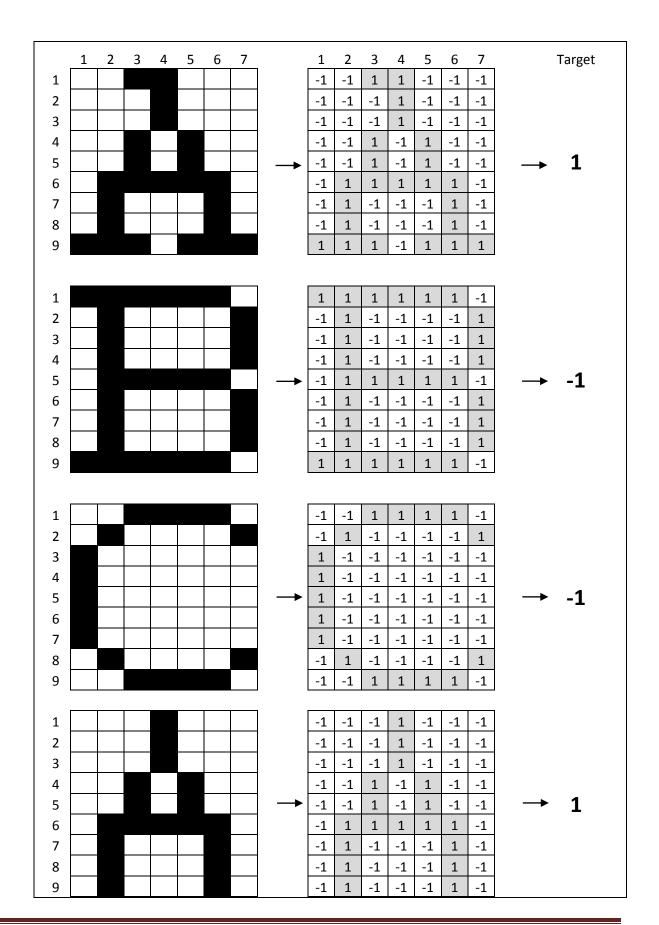
Ması	ukan	Keluaran
x_1	x_2	У
1	1	1
1	0	0
0	1	0
0	0	0

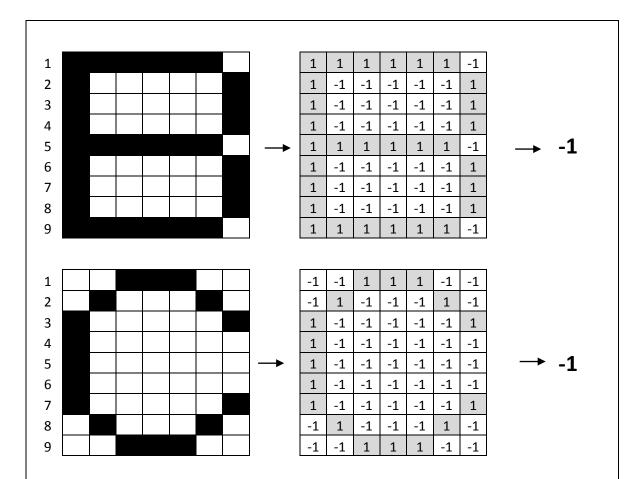
- 7. Berdasarkan percobaan Buatlah program pengenalan pola fungsi logika NOR!
- 8. Buatlah program pengenalan pola fungsi logika XOR!

PRAKTIKUM 2.2 PERCEPTRON UNTUK PENGENALAN POLA KARAKTER

- a. Tujuan
- : 1. Mahasiswa memahami arsitektur perceptron
- 2. Mahasiswa mampu merancang arsitektur jaringan saraf tiruan untuk pengenalan pola karakter
- 3. Mahasiswa mampu membuat program pengenalan pola karakter dengan jaringan saraf tiruan perceptron
- b. Dasar Teori

Pada percobaan ini akan dibuat perceptron yang dapat mengenali pola karakter "A". Diketahui 6 buah pola masukan gambar biner ukuran piksel 7 x 9 (63 piksel) yang diubah menjadi bipolar. Target keluaran pola huruf "A" adalah 1, dan selain "A" adalah -1.





Tahapan penyelesaian masalah dengan menggunakan jaringan saraf tiruan adalah sebagai berikut:

Tahap I: Identikasi masalah

Jaringan memiliki 63 buah masukan yaitu x_1 - x_{63} , serta memiliki satu keluaran y.

Tahap II: Training data set

Training data set diperlihatkan pada tabel. Terdapat 6 pola masukan dan keluaran yang akan digunakan untuk melatih jaringan. Untuk memudahkan, buatlah file .csv yang berisi masukan dari 6 pola seperti berikut

- 4	Α	В	C	D	Е	F	G	Н	1	J	K	L	M	N	О	P	Q	R	S	Т	U	V	W	X	Υ
1	x1	x2	хЗ	х4	x5	х6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	x22	x23	x24	x25
2	-1	-1	1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	1	-1
3	1	1	1	1	1	1	-1	-1	1	-1	-1	-1	-1	1	-1	1	-1	-1	-1	-1	1	-1	1	-1	-1
4	-1	-1	1	1	1	1	-1	-1	1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1
5	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	1	-1
6	1	1	1	1	1	1	-1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1
7	-1	-1	1	1	1	-1	-1	-1	1	-1	-1	-1	1	-1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1

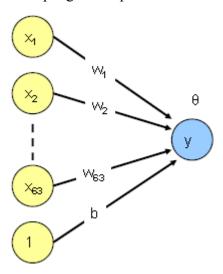
t = [1 - 1 - 11 - 1 - 1]

Tahap III: Pemilihan metoda

Pada percobaan ini digunakan jenis jaringan saraf tiruan yaitu perceptron

Tahap IV: Inisialisasi dan pembentukan jaringan

• Arsiterktur jaringan untuk pengenalan pola karakter "A" adalah sebagai berikut:



Karena jaringan memiliki 63 masukan, berarti jaringan terdiri dari 63 neuron pada layer masukan, 1 neuron pada layer keluaran, dan 1 bias. Pembentukan jaringan perceptron adalah dengan perintah berikut:

dengan:

- net : jaringan
- newp: perintah untuk membuat jaringan perceptron
- p : vektor masukan jaringan
- t : vector keluaran jaringan
- Fungsi aktivasi yang digunakan adalah 'hardlims' yaitu

$$y = f(net) \begin{cases} 1, net > 0 \\ 0, net = 0 \\ -1, net < 0 \end{cases}$$

Pengaturan fungsi aktivasi pada jaringan perceptron adalah dengan perintah berikut:

• Inisialisasi bobot dan bias awal $w_i = 0$, b = 0 dengan perintah berikut:

Tahap V: Simulasi jaringan

Sebelum melatih, dilakukan pemeriksaan nilai inisialisasi bobot dan bias awal. Perintah yang digunakan adalah

net.IW{1,1} : menampilkan bobot dari layer masukan ke kayer keluaran

net.b{1} : menampilkan bias layer keluaran

Selanjutnya dilakukan simulasi untuk mengetahui keluaran jaringan menggunakan bobot awal ini. Perintahnya:

```
y = sim(net, p);
```

dengan:

- y : keluaran jaringan
- sim : perintah untuk melakukan simulasi
- net : nama jaringan
- p : vektor masukan jaringan

Jika hasil simulasi menunjukan bahwa keluaran jaringan menggunaan bobot dan bias awal belum sesuai dengan target keluaran yang diinginkan, maka jaringan harus dilatih terlebih dahulu.

Tahap VI: Pelatihan jaringan

Sebelum dilakukan pelatihan, ada beberapa parameter yang harus diatur terlebih dahulu, yaitu:

- Pengaturan ambang batas selisih antara target dengan keluaran. Goal = 1
 net.trainParam.goal=1
- Pengaturan ambang batas iterasi atau epoch=1000

```
net.trainParam.epochs=10000
```

• Learning rate $\alpha = 1$

```
net.trainParam.lr=1
```

Perintah untuk melakukan pelatihan adalah sebagai berikut:

```
net = train(net,p,t);
```

dengan:

- train: perintah untuk melakukan pelatihan
- net : nama jaringan
- p : vektor masukan jaringan

- t : target keluaran jaringan yang diinginkan

Tahap VII: Penggunaan jaringan untuk pengenalan pola

Perintah untuk pengenalan pola adalah printah simulasi jaringan, yaitu y = sim(net,p);

c. Code Program

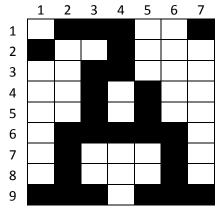
```
%baca pola masukan dari file csv, sesuaikan direktori file
ds=dataset('File','D:\masukan karakter.csv','Delimiter',',')
%konversi masukan ke bentuk double
data= double (ds)
%transpose matrik masukan agar sesuai dengan format masukan
jaringan
p=transpose (data(1:6,1:63))
%pola keluaran
t = [1 -1 -1 1 -1 -1]
%pembentukan jaringan perceptron
net=newp(p,t);
%pemilihan fungsi aktivasi
net.layers{1}.transferFcn = 'hardlims';
%inisialisasi bobot dan bias awal
net.IW\{1,1\}= zeros(1,63);
net.b{1} = [0];
net.IW\{1,1\}
net.b{1}
%simulasi masukan p pada jaringan
```

```
y = sim(net, p);
У
%pengaturan ambang batas selisih antara target dan keluaran
net.trainParam.goal=1
%pengaturan ambang batas iterasi atau epoch
net.trainParam.epochs=10000
%pengaturan nilai learning rate
net.trainParam.lr=1
%pelatihan jaringan
net = train(net,p,t);
net.IW\{1,1\}
net.b{1}
%Pengenalan pola masukan p pada jaringan
y = sim(net, p);
У
```

d. Langkah Kegiatan

- 1. Buatlah file csv yang berisi seluruh pola masukan karakter seperti pada poin b pada bagian Tahap II Training data set.
- 2. Buka aplikasi Matlab.
- 3. Atur direktori tempat bekerja yang berfungsi mengatur tempat penyimpanan M-File dengan cara mengatur **Current Directory** pada toolbar. Kemudian pilih tanda panah ke bawah dan pilih **new directory**.
- 4. Membuat M-File baru dengan cara pilih menu File -> New -> M-file.
- 5. Pada Matlab Editor, ketikan baris demi baris kode di poin c. Pada baris pertama, sesuaikan direktori tempat penyimpanan dan nama file training data set .csv yang dibuat pada langkah 1.

- 6. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.** Simpan M-File dengan nama file yang anda kehendaki (sebagai contoh "2.2 karakter A"). Amati hasil yang tampil.
- 7. Uji apakah jaringan dapat mengenali pola 1 dan 4 sebagai karakter A?
- 8. Uji apakah jaringan tidak dapat mengenali pola 2, 3, 5 dan 6 sebagai karakter A?
- 9. Uji apakah jaringan dapat mengenali pola berikut sebagai karakter "A" atau tidak?



- 10. Ubah nilai learning rate dan goal (sesuai dengan keinginan) pada Matlab Editor.
- 11. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.**

e. Pengujian Pemahaman:

- 1. Berdasarkan percobaan pada langkah 7, 8, dan 9, apakah sistem mengenali pola tersebut sebagai pola karakter "A"?
- 2. Berdasarkan hasil percobaan, berapa nilai learning rate dan goal yang diberikan? bagaimana pengaruhnya terhadap performa pengenalan pola karakter "A"?
- 3. Mengapa pada dataset dilakukan proses transpose?
- 4. Apa maksud dari kode p=transpose (data(1:6,1:63))?
- 5. Mengapa pada proses inisialisasi nilai bobot digunakan kode net.IW{1,1}= zeros(1,63)?
- 6. Buatlah program pengenalan pola karakter "B" dengan pola masukan seperti pada percobaan pengenalan pola karakter "A"!
- 7. Buatlah program pengenalan pola karakter "O"!

PRAKTIKUM 2.3 PERCEPTRON UNTUK PENGENALAN HURUF A, B, C

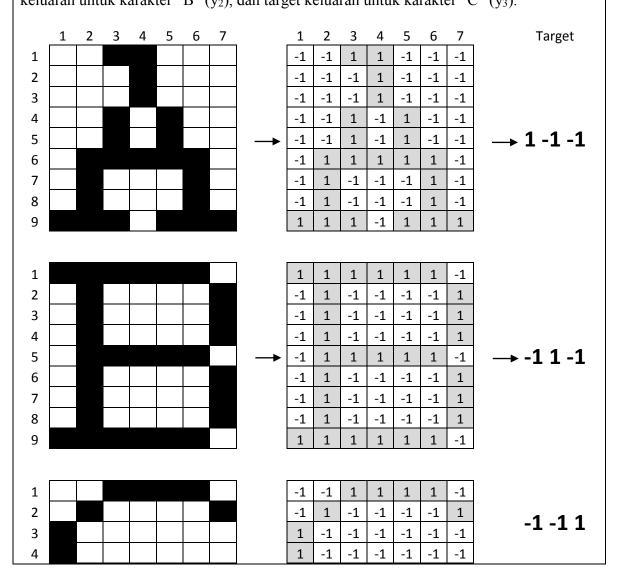
- a. Tujuan
- : 1. Mahasiswa memahami arsitektur perceptron
- 2. Mahasiswa mampu menggabungkan beberapa perceptron (multi perceptron)
- 3. Mahasiswa mampu merancang arsitektur jaringan saraf tiruan untuk pengenalan pola beberapa karakter

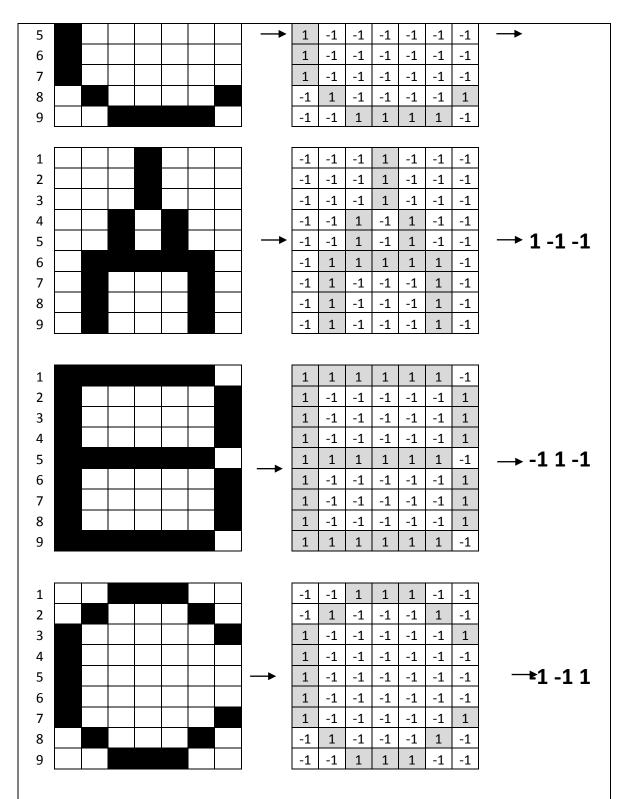
4. Mahasiswa mampu membuat program pengenalan pola beberapa karakter dengan jaringan saraf tiruan multi perceptron

b. Dasar Teori

Pada percobaan ini akan dibuat perceptron yang dapat mengenali pola karakter "A, B, C". Setiap karakter diasumsikan memiliki satu model perceptron, oleh karena itu pada kasus ini dibuat 3 model perceptron yaitu model perceptron untuk karakter "A", "B", dan "C".

Ketiga model perceptron tersebut digabungkan. Diketahui 6 buah pola masukan gambar biner ukuran piksel 7 x 9 (63 piksel) yang diubah menjadi bipolar. Target keluaran pola huruf terdiri dari tiga target keluaran yaitu target keluaran untuk karakter "A" (y₁), target keluaran untuk karakter "B" (y₂), dan target keluaran untuk karakter "C" (y₃).





Tahapan penyelesaian masalah dengan menggunakan jaringan saraf tiruan adalah sebagai berikut:

Tahap I: Identikasi masalah

Jaringan memiliki 63 buah masukan yaitu x_1 - x_{63} , serta memiliki 3 keluaran yaitu y_1 , y_2 , dan y_3 .

Tahap II: Training data set

Training data set diperlihatkan pada tabel. Terdapat 6 pola masukan dan keluaran yang akan digunakan untuk melatih jaringan. Untuk memudahkan, buatlah file .csv yang berisi masukan dari 6 pola seperti berikut

-4	Α	В	C	D	Ε	F	G	Н	1	J	K	L	M	N	O	Р	Q	R	S	T	U	V	W	X	Υ
1	x1	x2	хЗ	х4	x 5	х6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	x22	x23	x24	x25
2	-1	-1	1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	1	-1
3	1	1	1	1	1	1	-1	-1	1	-1	-1	-1	-1	1	-1	1	-1	-1	-1	-1	1	-1	1	-1	-1
4	-1	-1	1	1	1	1	-1	-1	1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1
5	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	1	-1
6	1	1	1	1	1	1	-1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1
7	-1	-1	1	1	1	-1	-1	-1	1	-1	-1	-1	1	-1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1

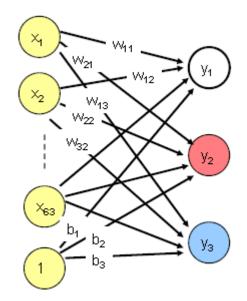
$$t = [1 - 1 - 1 1 - 1 - 1; -1 1 - 1 - 1 1 - 1; -1 - 1 1 - 1]$$

Tahap III: Pemilihan metoda

Pada percobaan ini digunakan jenis jaringan saraf tiruan yaitu perceptron

Tahap IV: Inisialisasi dan pembentukan jaringan

• Arsiterktur jaringan untuk pengenalan pola karakter "A, B, C" adalah sebagai berikut:



Pola		Target								
Masukan	t ₁	t ₂	t₃							
Pola 1	1	-1	-1							
Pola 2	-1	1	-1							
Pola 3	-1	-1	1							
Pola 4	1	-1	-1							
Pola 5	-1	1	-1							
Pola 6	-1	-1	1							

Karena jaringan memiliki 63 masukan, berarti jaringan terdiri dari 63 neuron pada layer masukan, 3 neuron pada layer keluaran, dan 1 bias. Pembentukan jaringan perceptron adalah dengan perintah berikut:

dengan:

- net : jaringan
- newp : perintah untuk membuat jaringan perceptron
- p : vektor masukan jaringan
- t : vector keluaran jaringan
- Fungsi aktivasi yang digunakan adalah 'hardlims' yaitu

$$y = f(net) \begin{cases} 1, net > 0 \\ 0, net = 0 \\ -1, net < 0 \end{cases}$$

Pengaturan fungsi aktivasi pada jaringan perceptron adalah dengan perintah berikut:

```
net.layers{1}.transferFcn = 'hardlims';
```

• Inisialisasi bobot dan bias awal $w_i = 0$, b = 0 dengan perintah berikut:

```
net.IW{1,1}= zeros(3,63);
net.b{1} = zeros(3,1);
```

Tahap V: Simulasi jaringan

Sebelum melatih, dilakukan pemeriksaan nilai inisialisasi bobot dan bias awal. Perintah yang digunakan adalah

 $\verb"net.IW{1,1}" : menampilkan bobot dari layer masukan ke kayer keluaran$

net.b{1} : menampilkan bias layer keluaran

Selanjutnya dilakukan simulasi untuk mengetahui keluaran jaringan menggunakan bobot awal ini. Perintahnya:

```
y = sim(net, p);
```

dengan:

- y : keluaran jaringan
- sim : perintah untuk melakukan simulasi
- net : nama jaringan
- p : vektor masukan jaringan

Jika hasil simulasi menunjukan bahwa keluaran jaringan menggunaan bobot dan bias awal belum sesuai dengan target keluaran yang diinginkan, maka jaringan harus dilatih terlebih dahulu.

Tahap VI: Pelatihan jaringan

Sebelum dilakukan pelatihan, ada beberapa parameter yang harus diatur terlebih dahulu,

yaitu:

- Pengaturan ambang batas selisih antara target dengan keluaran. Goal = 1
 net.trainParam.goal=1
- Pengaturan ambang batas iterasi atau epoch=1000

```
net.trainParam.epochs=10000
```

• Learning rate $\alpha = 1$

```
net.trainParam.lr=1
```

Perintah untuk melakukan pelatihan adalah sebagai berikut:

```
net = train(net,p,t);
```

dengan:

- train: perintah untuk melakukan pelatihan
- net : nama jaringan
- p : vektor masukan jaringan
- t : target keluaran jaringan yang diinginkan

Tahap VII: Penggunaan jaringan untuk pengenalan pola

Perintah untuk pengenalan pola adalah printah simulasi jaringan, yaitu

```
y = sim(net,p);
```

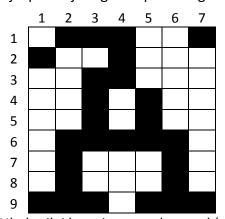
c. Code Program

```
%pembentukan jaringan perceptron
net=newp(p,t);
%pemilihan fungsi aktivasi
net.layers{1}.transferFcn = 'hardlims';
%inisialisasi bobot dan bias awal
net.IW\{1,1\}= zeros(3,63);
net.b{1} = zeros(3,1);
net.IW\{1,1\}
net.b{1}
%simulasi masukan p pada jaringan
y = sim(net, p);
У
%pengaturan ambang batas selisih antara target dan keluaran
net.trainParam.goal=1
%pengaturan ambang batas iterasi atau epoch
net.trainParam.epochs=10000
%pengaturan nilai learning rate
net.trainParam.lr=1
%pelatihan jaringan
net = train(net, p, t);
net.IW\{1,1\}
net.b{1}
```

```
%Pengenalan pola masukan p pada jaringan
y = sim(net,p);
Y
```

d. Langkah Kegiatan

- 1. Buatlah file csv yang berisi seluruh pola masukan karakter seperti pada poin b pada bagian Tahap II Training data set.
- 2. Buka aplikasi Matlab.
- 3. Atur direktori tempat bekerja yang berfungsi mengatur tempat penyimpanan M-File dengan cara mengatur **Current Directory** pada toolbar. Kemudian pilih tanda panah ke bawah dan pilih **new directory**.
- 4. Membuat M-File baru dengan cara pilih menu File -> New -> M-file.
- 5. Pada Matlab Editor, ketikan baris demi baris kode di poin c. Pada baris pertama, sesuaikan direktori tempat penyimpanan dan nama file training data set .csv yang dibuat pada langkah 1.
- 6. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.** Simpan M-File dengan nama file yang anda kehendaki (sebagai contoh "2.2 karakter A"). Amati hasil yang tampil.
- 7. Uji apakah jaringan dapat mengenali pola 1 dan 4 sebagai karakter A?
- 8. Uji apakah jaringan dapat mengenali pola 2 dan 5 sebagai karakter B?
- 9. Uji apakah jaringan dapat mengenali pola 3 dan 6 sebagai karakter C?
- 10. Uji apakah jaringan dapat mengenali pola berikut sebagai karakter "A" atau tidak?



- 11. Ubah nilai learning rate dan goal (sesuai dengan keinginan) pada Matlab Editor.
- 12. Simpan dan jalankan script pada M-File dengan cara pilih menu **Debug -> Save and Run** atau dengan menekan **F5.**

e. Pengujian Pemahaman:

1. Berdasarkan percobaan pada langkah 7, 8, 9, dan 10, apakah sistem mengenali pola pola karakter "A, B, C" dengan tepat?

